CNC SYSTEMS

OSP-U100L
OSP-U10L

PROGRAMMING MANUAL
(1st Edition)

Pub. No. 4197-E (LE33-010-R1) March 1998

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 Machine Installation

(1) Install the machine at a site where the following conditions (the conditions for achievement of the guaranteed accuracy) apply.

- Ambient temperature: 17 to 25°C
- Factory humidity: 40% to 75% at 20°C (no condensation)
- Site not subject to direct sunlight or excessive vibration; environment as free of dust, acid, corrosive gases, and salt spray as possible.

(2) Prepare a primary power supply that complies with the following requirements.

- Voltage: 200 V
- Voltage fluctuation: ±10% max.
- Power supply frequency: 50/60 Hz
- 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 NC 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).

(3) Installation Environment

Observe the following points when installing the electrical control cabinet.

- Make sure that the NC unit will not be subject to direct sunlight.
- Make sure that the electrical control cabinet will not be splashed with chips, water, or oil.
- Make sure that the electrical control cabinet and operation panel are not subject to excessive vibrations or shock.
- The permissible ambient temperature range for the electrical control cabinet is 0 to 40°C.
- The permissible ambient humidity range for the electrical control cabinet is 30 to 95% (no condensation).
- The maximum altitude at which the electrical control cabinet can be used is 1000 m (3281 ft.).

2. Points to Check before Turning on the Power

(1) Close all the doors of the electrical control cabinet 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 and Continuous Operation

(1) Always follow the instructions in the operating manual.

(2) Do not operate the machine with any of the safety covers (front shield, chuck cover, etc.) removed.

(3) Always close the front shield before starting operation.

(4) Never run a new program without checking its operation. Run the program with no workpiece in the chuck and make sure that there is no interference, then cut a workpiece in the single block mode. If no problems are discovered, automatic operation may be started.

(5) Confirm safety before performing operations involving spindle rotation or axis movement.

(6) Never touch chips or the workpiece while the spindle is rotating.

(7) Do not attempt to stop rotating parts with your hand or any object.

(8) Check the jaw mounting conditions, hydraulic pressure, and maximum allowable spindle speed for the power chuck.

(9) Check the mounting and arrangement of the tools.

(10) Check the tool offset settings.

(11) Check the zero offset settings.

(12) Make sure that the spindle speed and feedrate override settings set on the NC operation panel are 100%.

(13) Before moving the turret, check the software limit setting and the emergency limit LS (limit switch) dog positions for both the X- and Z-axes to ensure that it will not interfere with the chuck or tailstock.

(14) Check the turret index/rotation position.

(15) Check the tailstock body position.

(16) Make sure the cutting operation is within the allowable transmission power and torque ranges.

(17) Make sure the workpiece is securely clamped in the chuck or fixture.

(18) Check that the coolant nozzles are positioned correctly.

4. On Finishing Work

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

(2) Move the turret 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 Applicable 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 maintenance 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 electrical control cabinet 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 electrical control cabinet

a) Switch the main power disconnect switch OFF before opening the electrical control cabinet door.

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

- Servo amplifier

  Discharge the residual voltage one minute after turning off the breaker inside the unit.

- Spindle drive unit

  Discharge the residual voltage one minute after turning off the main power disconnect switch.

c) The electrical control cabinet 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 electrical control cabinet

a) Cleaning the cooling unit

The cooling unit in the door of the electrical control cabinet serves to prevent excessive temperature rise inside the electrical control cabinet 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 machine and area around it 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 gear such as safety shoes and safety 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 imminent hazard which, if not avoided, will result in death or serious injury.

WARNING : Indicates hazards which, if not avoided, could result in death or serious injury.

CAUTION : Indicates hazards which, if not avoided, could result in minor injuries or damage to devices or equipment.

NOTICE : Indicates precautions relating to operation or use.

INTRODUCTION

This programming manual contains instructions for programming the OSP. A careful reading of the manual will be of great assistance in obtaining the full benefit of all the superior functions the machine has to offer. For the most complete understanding, this manual should be read in conjunction with the "Operation Manual for OSP", as the two manuals are very closely related.

TABLE OF CONTENTS

PAGE

PAGE

PAGE

PAGE

PAGE

PAGE

PAGE

SECTION 1 OVERVIEW

Explanation and cautions on programming using this NC unit are given below for the function groups such as program configurations and format, coordinates and coordinate commands, math functions, preparatory functions, fixed cycles, S/T/M functions, and user task functions.

G codes, M codes, etc. are summarized in list in appendix.

Notes on Reading the Manual

The M code numbers used in this manual are the numers assigned as the NC function M codes. There may be cases that the actual M code numbers used in specific machine tools differ from those used in this manual. For the actual M code numbers used by your machine, referred to the manuals of the machine.

SECTION 2 PROGRAM TYPES AND CONFIGURATIONS

Programs used for automatic operation are classified into the following three types: schedule programs, main programs, and subprograms.

1. File Name

1-1. Schedule Program

The schedule program specifies how many times the program is to be executed and the execution order of the main programs, when more than one type of workpiece is being machine using bar feeder or other automatic loading and unloading equipment. This feature allows the realization of continuous machine operation.

A program name may not be used, and the last line of a schedule program must end with the END code.

For details, refer to SECTION 25, "SCHEDULE PROGRAM".

1-2. Main Program

The main program contains a series of commands to machine one type of workpiece. Subprograms can be called from the main program to simplify programming. The program name, which must start with an "O" is required at the beginning of the main program. The end of program M code must be specified at the end of the main program.

1-3. Subprogram

A subprogram can be called from the 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 23. "USER TASK".

An extension is added to the file name when the program is stored to assist the NC in file management. The format is shown below:

<File Name Format>

Main file name ⁎ (The file name consists of a character string of up to 16 characters which begins with an alphabetic character.)	•	Extension (Up to 3 alphabetic characters)

List of extension names:

1) SDF schedule program file
2) MIN main program file
3) SSB system subprogram file
4) SUB user's subprogram file

2. Program Name

A program name or program number is assigned to each program. Operations can be carried out just by calling the program.

If a program name contains only numbers, it is called a program number.

(1) Program Name Designation

(a) Input alphabets (A to Z) or numbers (0 to 9) following after "O".

Note that no space is allowed between "O" and an alphabet or a number. Similarly, no space is allowed between alphabets and numbers.

(b) Up to four characters can be used.

(c) If alphabetic characters are used, the program name must begin with an alphabetic character. If the program name begins with an alphabetic character, it can contain a number. However, if the program name begins with a number, it cannot contain an alphabetic character.

(d) A block which contains a program name must not contain other commands.

(e) The schedule program is not assigned a name.

(f) Although a main program may not be assigned a program name beginning with "O", subprograms must have a program name beginning with "O".

(g) Program names are read in units of characters.

Examples:

1) O0123 is different from O123
2) O00 is different from O0.

(h) Program names must be unique. If the same program name is used for two or more programs, the required program cannot be selected correctly.

3. Sequence Name

A sequence name is defined as a name assigned to a block. Numeric or alphabetic characters following after "N" are designated for a sequence name.

A sequence name makes it possible to use a sequence number search function, and a branching function in a program.

If a sequence name contains only numbers, it is called a sequence number.

3-1. Sequence Name Designation

(a) Input alphabets (A to Z) or Numbers (0 to 9) following after address "N".

(b) Up to four characters can be used.

(c) Both alphabetic and numeric characters may be used.

However, if alphabetic characters are used, the sequence name must begin with an alphabetic character.

(d) A sequence name must be placed at the top of a block. However, a block delete command may be placed preceding a sequence name.

(e) Sequence numbers do no indicate the order of program execution. They may be specified in any order as long as they are unique.

(f) Sequence names are read in units of characters.

Example:

1) N0123 is different from N123.

2) N00 is different from N0.

(g) When a sequence label is used, place a space or a tab following the sequence label.

4. Program Format

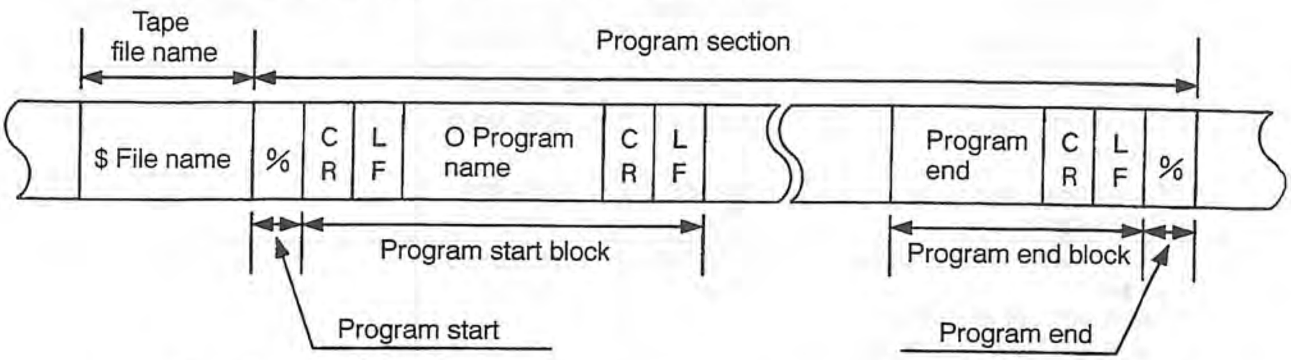
4-1. Program Command Format

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	

SECTION 3 PROGRAM FILE FORMAT

1. File Format

The general format of the NC machining program is shown below:



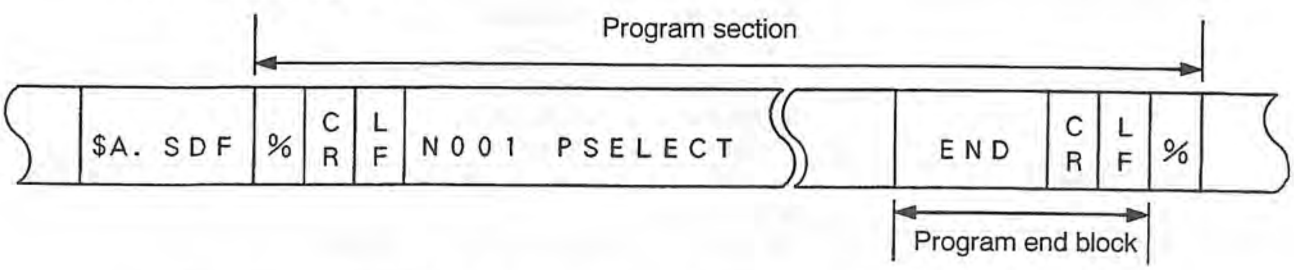
(1) In the EIA code, one EOB (CR) character may be used instead of CR and LF used in the ISO code.

In the ISO code, CR may be omitted.

(2) The program section must begin and end with a percent sign "%".

Three types of programs are handled by the OSP. Their file formats are shown below:

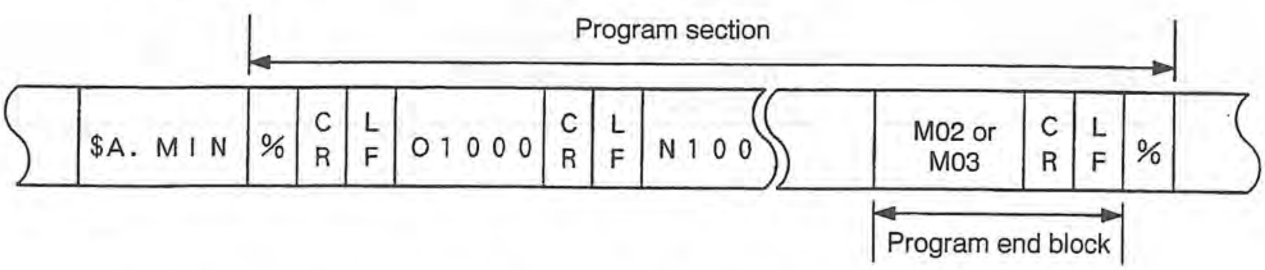
(a) Schedule programs



No program name may be punched in a schedule program.

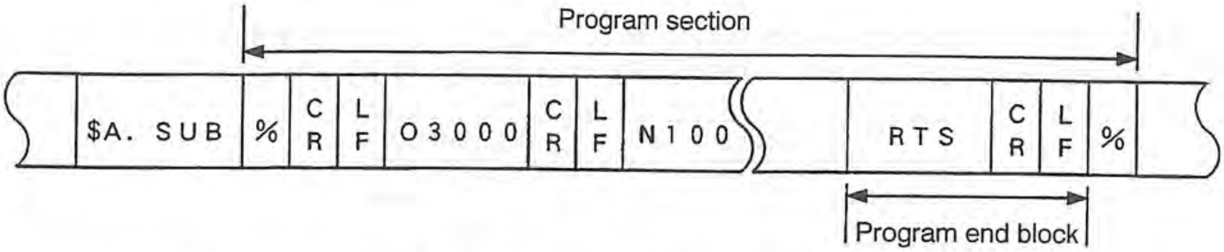
The program end block must contain the "END" code.

(b) Main programs



The program end block must contain the end of program M code.

(c) Subprograms



The RTS command must be specified at the end of the subprogram.

2. Tape Code

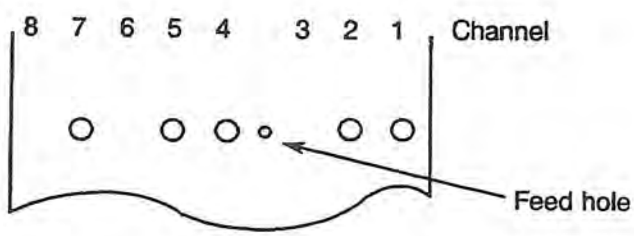
When creating a part program, the NC accepts both the EIA and ISO code. Note that, however, there are characters that can and cannot be used in each of the code. The usable characters are shown in the list in APPENDIX, 1. "EIA/ISO CODE TABLE". The EIA code and the ISO code are not compatible with each other. The processing to be taken if a character not given in the usable character list is specified in a program can be selected from "causing alarm" and "disregarded" and which of the processing should be taken is set for the optional parameter (EXTERNAL I/O).

3. Setting the Special Codes

Among characters and codes used on the NC, those indicated below are not specified in the EIA code. They should be set by NC optional parameter (bit) matching the puncher to be used.

Codes	Parameter No.
=	21
*	22
[	23
]	24
$	25
The code (irregular code) to be replaced with regular code	26
The code (regular code) to be replaced into irregular code	27

Example: Suppose the puncher key "[ ]" is determined for punching the "=" code, and that the arrangement of punched holes by this key operation is as below.



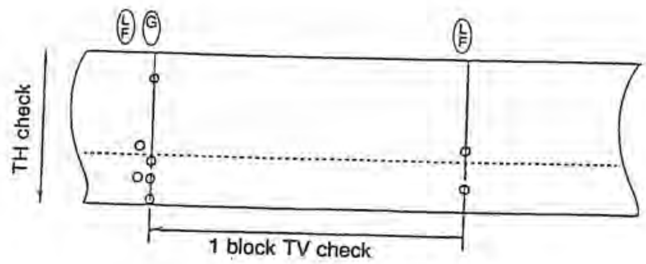
Set this arrangement of punched holes by a "1" and a "0", where "1" indicating a punched hole and "0" a position not punched. Setting will be as below:

0 1 0 1 1 0 1 1

Set this at the No. 21 of optional parameter (bit). Repeating the same operations, set all the codes used on the NC.

4. Parity Check

Two types of parity checks, TV and TH, are conducted during the reading of a part program tape or in a verify operation.



(1) TH check (horizontal parity)

This function checks if the number of holes per character of the machining tape is correct, and also checks for EIA if the number of holes is odd and for ISO if the number of holes is even. The DEL code which has eight holes does not cause a parity error.

(2) Block Configuration

A program is composed of several commands, one unit of which is referred to as a block. An end of block (EOB) code is placed as a delimiter between blocks.

a) The end of block code is different depending on the coding system selected, ISO or EIA:

ISO . . . "LF"
EIA . . . "CR"

b) A block is comprised of several words.

c) Up to 158 characters are allowed in one block.

(3) Word Configuration

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

Examples:

X-1000	Z=100∗SIN[50]	Z=V1+V2
Address Numeric	Address Numeric	Address Numeric
Word	Word	Word

a) An address character may consist of one of the alphabetic characters A through Z to specify respective contents. An extended address character, consisting of two alphabetic characters may also be used.

b) Refer to SECTION 23, 3-2. "Variable Function" for more information on variables.

c) TV check (vertical parity)

This function checks if the number of characters, including LF (ISO) or CR (EIA) which constitute one block, is an even number. It may be selected whether or not this check should be carried out by the optional parameter (bit) No. 1, bit 2.

The check is applied to all the characters including comments.

In the TH check, it may be estimated whether the machining tape to be read is composed of ISO or EIA code characters and it may be selected whether or not automatic recognition should be conducted, by means of the NC optional parameter (bit) No. 1, bits 0 and 1.

SECTION 4 COORDINATES SYSTEMS

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 the coordinate values in the set coordinate system. There are three types of coordinate systems indicated below and a program coordinate system is used for programming.

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

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.

3. Machine Coordinate System

The reference point on the machine is referred to as the machine zero and the coordinate system which has the 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.

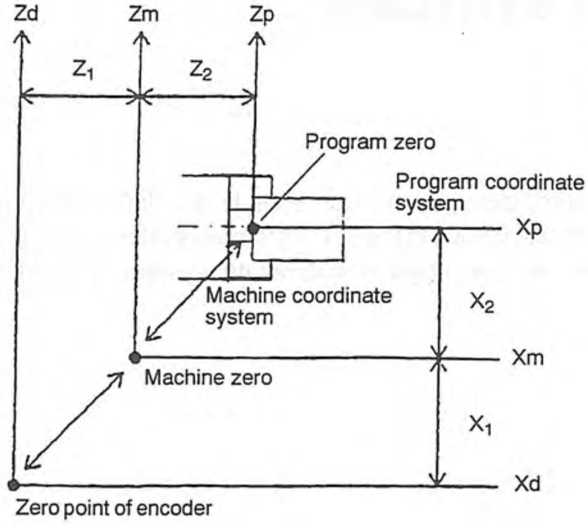
If "0" is set for the encoder zero point offset (system parameter), the machine coordinate system agrees with the encoder coordinate system.

4. Program Coordinate System

The coordinate system used as the reference of program commands is called the program coordinate system.

The position of 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)

X_1, Z_1: Offset amount of position encoder

X_2, Z_2: Zero point offset amount

SECTION 5 COORDINATES AND COORDINATE COMMANDS

1. Controlled Axis

(1) The following table lists the addresses necessary to control the axis.

	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

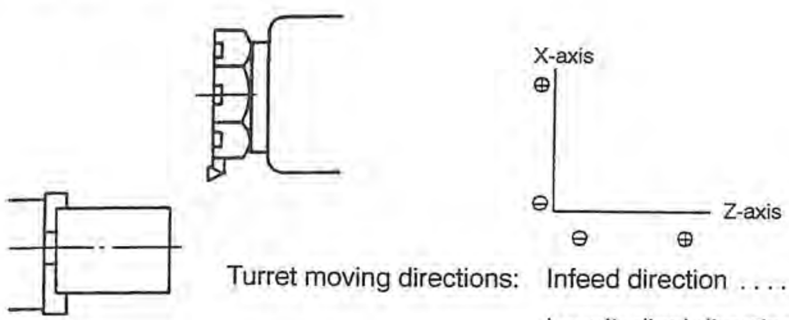
(2) A command used to move an axis consists of an axis address, direction of movement, and a target point.

For the designation of a target point, two different methods are available: absolute command and incremental command. 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 "4. Absolute and Incremental commands".

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

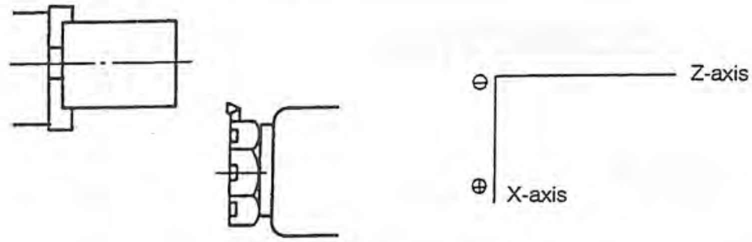
(a) Single-saddle NC lathe



Turret moving directions: Infeed direction .... X-axis

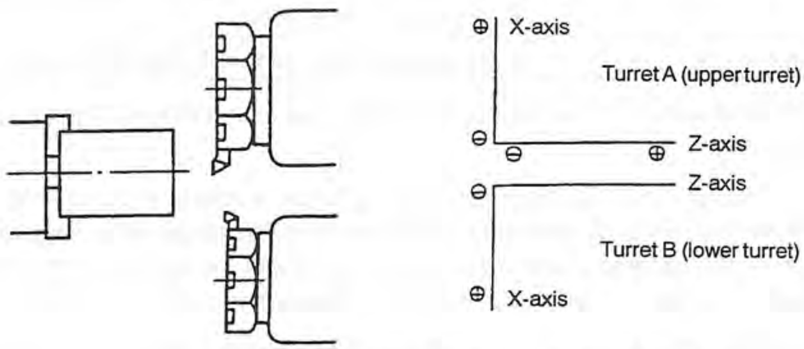
Longitudinal direction .... Z-axis

(b) Single-saddle NC lathe (flat bed)



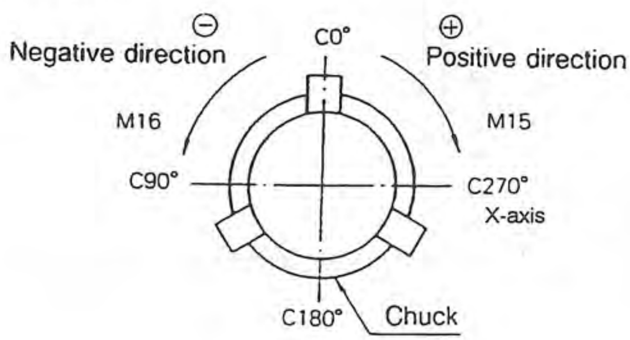
Turret moving directions: Infeed direction . . . . X-axis
Longitudinal direction . . . . Z-axis

(c) Two-saddle NC lathe



Turret moving directions: Infeed direction . . . . X-axis
Longitudinal direction . . . . Z-axis

(d) C-axis coordinate system



(When Viewed from Tailstock)

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 diraction.

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 base of metric system values. The unit system to be selected for data input is determined according to the setting of NC optional parameter (UNIT). The actual unit system for data input can be checked on the NC optional parameter (UNIT) screen.

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

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.

(1) Metric System:

1 μm

10 μm

1 mm

(2) Inch System (Inch/metric switchable specification):

1/10000 inch

1 inch

Unit Data Table (Value for data "1")

Unit / 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} {rpm}: S	1 (min^{-1} {rpm})	1 (min^{-1} {rpm})	1 (min^{-1} {rpm})	1 (min^{-1} {rpm})	1 (min^{-1} {rpm})
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:

1) 0.001 mm movement of X-axis	X0.001
2) 10 mm movement of X-axis	X10
3) 100.00 mm movement of X-axis	X100.01
4) Feedrate of 0.23456 mm/rev.	F0.23456

Following commands are all handed as X1 mm:

X1

X1.0

X1.00

X1.000

Example 2: 10 μm unit system

Commanding:

1) 0.001 mm movement of X-axis	X0.1
2) 10 mm movement of X-axis	X1000
3) 100.010 mm movement of X-axis	X10001
4) Feedrate of 0.23456 mm/rev.	F23.456

Example 3: 1 μm unit system

Commanding:

1) 0.001 mm movement of X-axis	X0.1
2) 10 mm movement of X-axis	X10000
3) 100.010 mm movement of X-axis	X100010
4) Feedrate of 0.23456 mm/rev.	F234.56

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

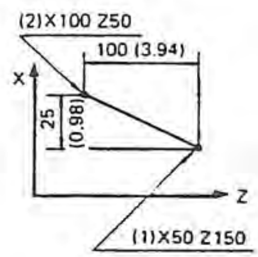
F1.2345678	Acceptable
F100.000001	Alarm (9 digits)

4. Absolute and Incremental Commands

(1) Amount of axis movement can be expressed in either absolute commands or incremental commands.

G Code	Description	Remark
G90	Calls the absolute programming mode (cancels the incremental programming mode.). In the absolute programming mode, the target point of axis movement is defined by the coordinate values in the program coordinate system.	When the control is reset, it is in the G90 mode.
G91	Calls the incremental programming mode. In the incremental programming mode, the target point of axis movement is defined by the amount of distance from the actual position.	

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



Unit: mm (in.)

Absolute

G00 X50 Z150 ..... (1)

X100 Z50 ...... (2)

Incremental

G00 X50 Z150 .... (1)

*G91 X50 Z-100........ (2)

* Designate dimensiona; differennces between points (2) and (1).

NOTICE : (1) In incremental programmig, X word should be expressed in diameter.

(2) It is not allowed to specify both G90 and G91 in the same block.

5. Diametric and Radial Commands

In turning operation, a workpiece is rotated while it is machined. Due to the nature of the turning operation, a tool cuts the circle that has the 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 in 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 6 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 the operation.

Table 1-1 Mathematical Operation Functions

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, 1.7321] (See *1.)
	Square root	SQRT	4 = SQRT [16]
	Absolute value	ABS	3 = ABS [–3]
	Decimal to binary conversion	BIN	25 = BIN [$25] ($ represents hexadecimal number.)
	Binary to decimal conversion	BCD	$25 = BCD [25]
	Integer implementation (rounding)	ROUND	128 = ROUND [1.2763 × 100]
	Integer implementation (truncation)	FIX	127 = FIX [1.2763 × 100]
	Integer implementation (raising)	FUP	128 = FUP [1.2763 × 100]
	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 Note 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 whose rectangular coordinate is (a, b).

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

*3: Spaces 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.)

a) Exclusive OR (EOR) c = a ␣ EOR ␣ b

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

b) Logical OR (OR) c = a ␣ OR ␣ b
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

c) Logical AND (AND) c = a ␣ AND ␣ b

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

d) Negation (NOT) b = NOT ␣ a

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

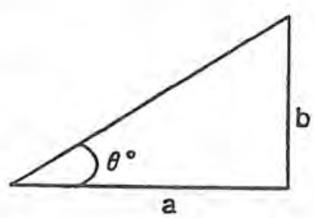
a	b
0	1
1	0

e) Arc tangent (1) (ATAN)

θ = ATAN [b/a]

Arc tangent (2) (ATAN2)

θ = ATAN2 [b, a]



f) Rounding off to integer (ROUND)

ROUND rounds off a specified value into an integer.

g) Truncation into integer (FIX)

FIX truncates (shortens) a specified value into an integer.

h) Raising into integer (FUP)

FUP raises a specified value into an integer.

The units for ROUND, FIX, and FUP functions are microns.

SECTION 7 BLOCK DELETE

(1) Function

(a) 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.

The operator thus has a method to execute or ignore blocks containing the "/" code.

(b) When the block skip function is activated, the entire block will be ignored.

(2) Notes

(a) The slash "/" code must be placed at either the start of a block or immediately after a sequence name (number). If it is placed other position in a block, it will cause an alarm.

(b) The slash "/" may not be contained in the program name block.

(c) Blocks which contain a "/" code are also subjected to TV and TH checks, regardless of the BLOCK DELETE switch position.

(d) Blocks which contain a "/" code are also subjected to the sequence search function, regardless of the BLOCK DELETE switch position.

(e) The block delete function is not possible during SINGLE BLOCK mode. The succeeding block is executed, and then the operation stops.

SECTION 8 COMMENT FUNCTION (CONTROL OUT/IN)

A program may be made easier to understand by using comments in parentheses. The comment must be parenthesized to distinguish it from general operation information.

(1) All information placed in parentheses is regarded by the machine as comments.

(2) Comments are also subject to TV and TH checks and displayed in the normal character size.

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

SECTION 9 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.), for the expansion of program storage capacity.

SECTION 10 TWO TURRETS

With flat bed type machines, there are models which have two turrets mounted on a saddle. Since both turrets are mounted in the same saddle in this configuration, it is not possible to control them independently. For such machines, the turret should be selected first when making a part program.

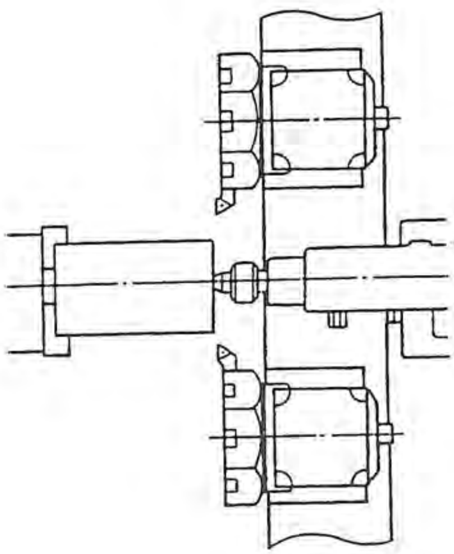
In the two-turret specification machines, front and rear turrets are called turret A and turret B, respectively, and selection of a turret is made by specifying the following G codes.

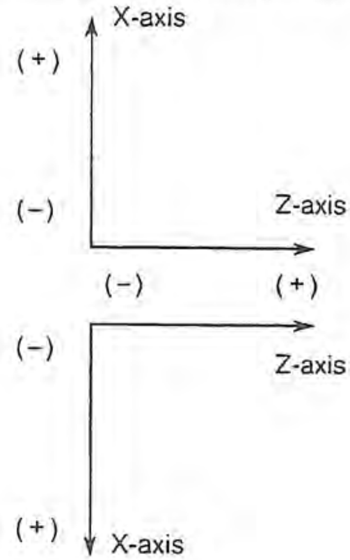
Selection of turret A G13

Selection of turret B G14

Although the numerically controlled axes are only X- and Z-axis since the machine has only one saddle, program zero is set for turrets A and B independently.

It should also be noted that the X-axis direction of coordinate systems is reversed between turrets A and B.





SECTION 11 VARIABLE LIMITS

If an axis movement is executed by the command that specifies 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.

Concerning axis movement by the command specifying a target point beyond the variable limit in the negative direction, axis movement is not executed but an alarm occurs.

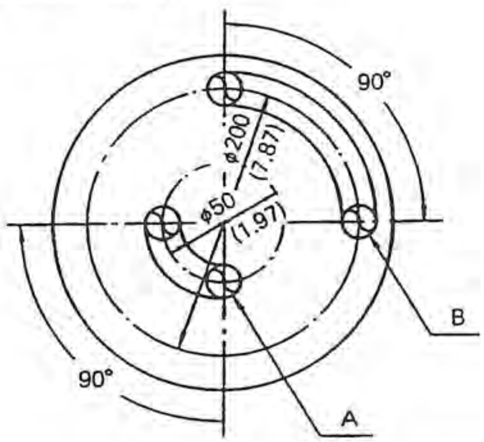
SECTION 12 DETERMINING FEEDRATE FOR CUTTING ALONG C-AXIS

1. Cutting by Controlling Only C-axis

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.

* To match the unit of C-axis feed command with the X- and/or Z-axis command, feedrate command (F) should be calculated by converting 360° into 500 mm. This conversion should also be carried out when only C-axis command is given.

See the example shown below:



Unit: mm (in.)

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

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

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

Along slot A .... $100/39 \times 90 \doteq 230$

Along slot B .... $100/156 \times 90 \doteq 58$

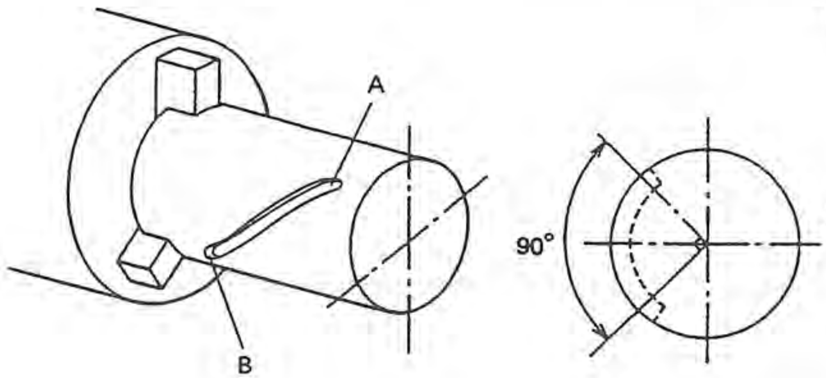
Convert the unit of feed from "deg/min" into "mm/min".

Slot A: $230/360 \times 500 \doteq 320$ (F320)

Slot B: $58/360 \times 500 \doteq 80$ (F80)

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

Example:



Point A coordinate value

X = 80

Z = 100

C = 120

Point B coordinate value

X = 80

Z = 50

C = 210

Fig. 12-1 Simultaneous Control of C- and Z-axis

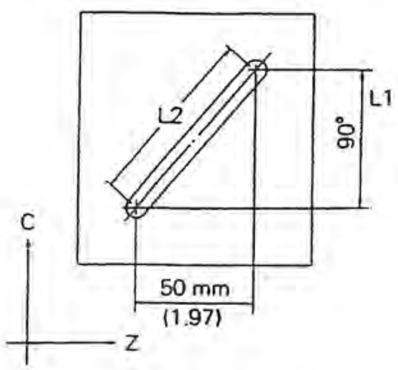
When cutting the spiral from A to B with a two-flute end mill in the following cutting conditions, calculate the feedrate of C-axis as explained below:

Cutting conditions : Feed per tooth .... 0.05 mm

M-tool speed .... 400 min^{-1} {rpm}

(1) Calculate the distance between A and B.

Development of the diagram above is indicated below.



The distance, L1, along circumference:

$$L1 = 80 \times \pi \times \frac{90}{360}$$

$$\doteqdot 63 \text{ (mm)}$$

The distance, L2, between A and B:

$$L2 = \sqrt{63^2 + 50^2}$$

$$\doteqdot 80 \text{ (mm)}$$

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

$$T = \frac{L2}{\text{(Feed per tooth)} \times \text{(No. of teeth)} \times (\text{min}^{-1}\ \{\text{rpm}\})}$$

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

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

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

Z-axis travel : 50 mm

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

(conversion based on 360° = 500 mm)

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

$$L3 = \sqrt{50^2 + 125^2}$$

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

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

$$F = \frac{L3}{T} = \frac{135}{2} \doteq 67.5$$

Specify F67.5 in the program.

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

Example:

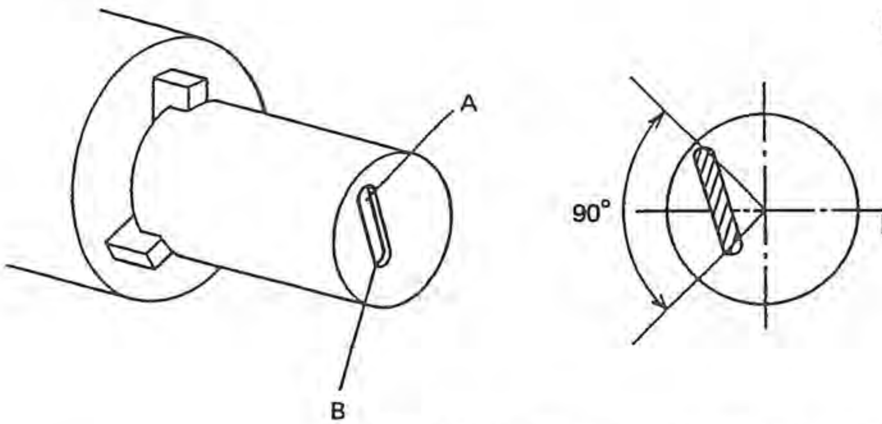
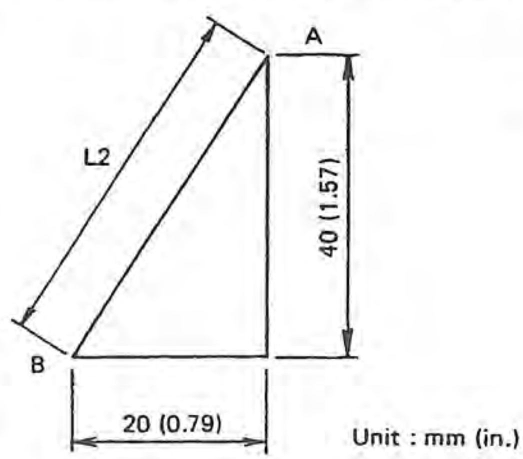


Fig. 12-2 Simultaneous Control of C- and X-axis

The cutting conditions are the same as used in SECTION 12, 1."Cutting by Controlling Only C-axis".

(1) Calculate the distance between A and B.



The distance, L2 betweem A and B:

$$L2 = \sqrt{40^2 + 20^2}$$

$$\doteqdot 44.7 \text{ (mm)}$$

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

$$T = \frac{L2}{(\text{Feed per tooth}) \times (\text{No. of teeth}) \times (\text{min}^{-1}\ \{\text{rpm}\})}$$

$$= \frac{44.7}{0.05 \times 2 \times 400}$$

$$= 1.12 \text{ (min)}$$

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

Z-axis travel : 50 mm

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

(conversion based on 360° = 500 mm)

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

$$L3 = \sqrt{40^2 + 125^2}$$

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

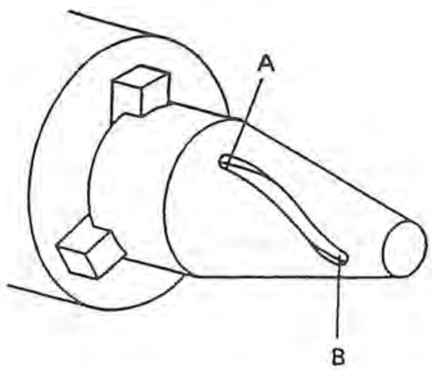
(4) The feedrate to be specified in the program is approximately calculated as below:

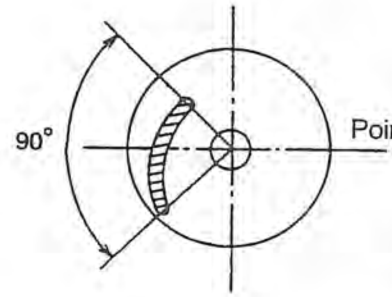
$$F = \frac{L3}{T} = \frac{135}{1.12} \doteqdot 117$$

Specify F67.5 in the program.

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

Example:





Point A coordinate value

X = 80

Z = 50

C = 120

Point B coordinate value

X = 40

Z = 100

C = 120

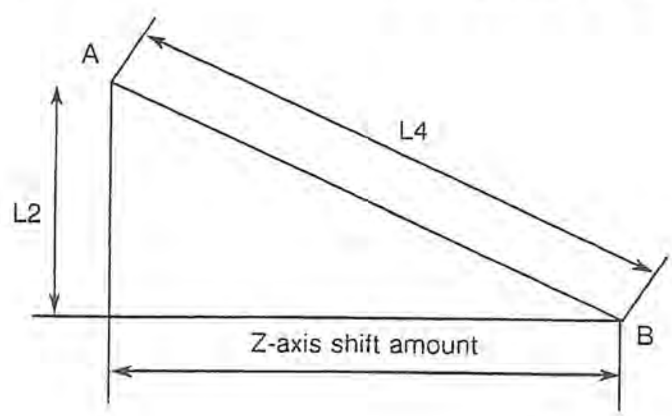
When cutting a slot on the cone as indicated above, simultaneous 3-axis control of X-, Z-, and C-axis becomes necessary. The feedrate to be programmed should be calculated in the following manner. Note that the example below is given assuming the same cutting conditions as used in SECTION 12, 2."Cutting by Controlling Both C-axis and Z-axis".

(1) First, assume the development of the slot on C-axis and X-axis. In this case, calculation of the feedrate is possible in the same manner as in SECTION 12, 3."Cutting by Controlling Both C-axis and X-axis" .

The distance, L2 between A and B:

$$L2 = \sqrt{40^2 + 20^2} \doteqdot 44.7 \text{ (mm)}$$

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



$$L4 = \sqrt{44.7^2 + 50^2} \doteqdot 67.1 \text{ (mm)}$$

Fig. 12-3

(3) Calculate the cutting time T for distance L4 based on the cutting conditions indicated in SECTION 12, 2."Cutting by Controlling Both C-axis and Z-axis".

$$T = \frac{L4}{(\text{Feed per tooth}) \times (\text{No. of teeth}) \times (\text{min}^{-1}\ \{\text{rpm}\})}$$

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

$$= 1.68\ (\text{min})$$

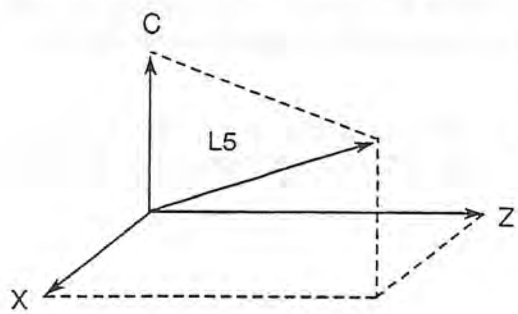
(4) Inside the computer, distance L5 between A and B is calculated in the following manner.

X-axis travel : 40 mm

Z-axis travel : 50 mm

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

(conversion based on 360° = 500 mm)



Distance by coordinated C- and Z-axis movement:

$$L5 \doteq \sqrt{40^2 + 50^2 + 125^2} \doteq 140.4\ (\text{mm})$$

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

$$F = \frac{L5}{T} = \frac{140.4}{1.68} \doteq 83.6$$

Specify F83.6 in the program.

SECTION 13 MATH FUNCTIONS AND AXIS MOVEMENT COMMANDS

1. Positioning (G00)

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.

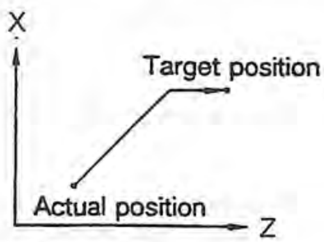
Programming format: G00 X_ Z_ C_

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

In the 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.



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

2. Linear Interpolation (G01)

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.

NOTICE :
(1) The feedrate becomes zero when the NC is reset.
(2) The feedrate for each axis is indicated below. (Calculate the feedrate for X, Z-axis in the incremental value.)

G01 XxZzFf

Calculation of feedrates:

X-axis feedrate: FX = (x/L)f
Z-axis feedrate: FZ = (z/L)f

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

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

3. Circular Interpolation (G02, G03)

Circular interpolation can be used to generate a cutting path which follows an arc.

Programming format:

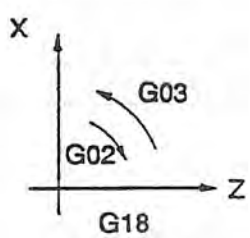
$$\begin{matrix} \text{G02} \\ \text{(G03)} \end{matrix}\ \text{X\_ Z\_} \begin{Bmatrix} \text{L\_} \\ \text{I\_ K\_} \end{Bmatrix} \text{F\_}$$

(1) G Codes and Addresses

The following table presents a summary of the G codes and addresses necessary for circular interpolation.

Item to be Designated		Command	Description
Rotary direction		G02	Clockwise
		G03	Counterclockwise
End point	G90	X, Z	End point in the program coordinate system
	G91	X, Z	End point referenced to the starting point Values should include signs.
Center of arc referenced to starting point		I, K	Center of arc referenced to starting point including signs
Arc radius		L	Radius of arc

(2) Rotary Directions



The two rotary directions, 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.

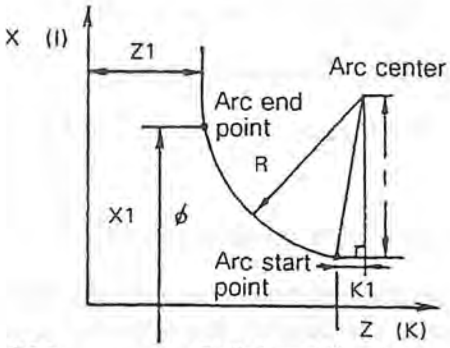
(3) End Points

The end point of an arc is reached depending on the G90/G91 selection.

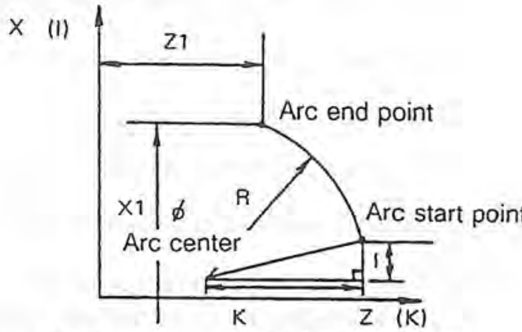
(4) Center Points

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 G90 or G91.



G02 .... Both I and K values are positive



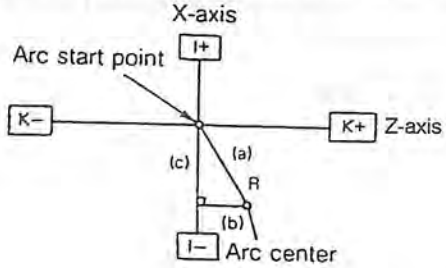
G03 .... Both I and K values are negative

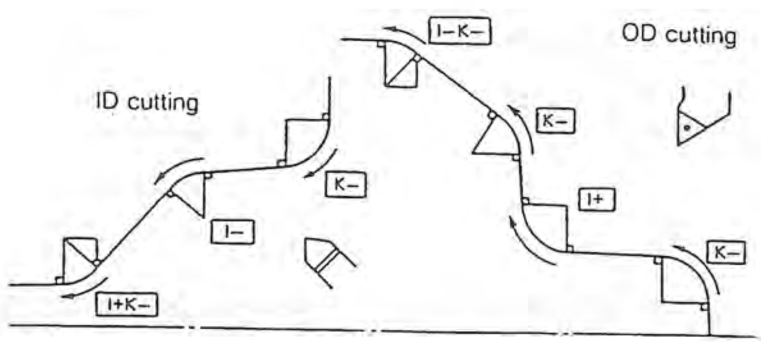
(X1, Z1) indicate the coordinate values of arc end point

Determining Sign and Numeric Value of I and K Words:

See the figure below. Assume the coordinate system having its origin at the arc start point. Draw a right triangle taking the segment connecting the arc center and arc start point as a hypotenuse. Length of the side (b) parallel to Z-axis is the value of K word and that of side (c) parallel to X-axis is the value of 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. In the similar way, the sign of an I word is determined. 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.





(5) 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.

(a) As a G code calling for circular interpolation, G02 or G03 is used as in conventional programming.

(b) Radius of the arc is expressed by an L word which must have a positive value.

(c) Block containing an L word without K or I word needs an arc radius command.

(d) When expressing the arc using its radius, the commands must contain both X and Z words.

If either of them is omitted, an alarm results.

(e) If an L word is specified in a block containing I and/or K word, an alarm results.

(f) If the distance from the current position to the target point (end point) is larger than two times the specified radius, it results and alarm since circular interpolation cannot be performed.

(g) 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 one with central angle larger than 180°, specify "CALRG" in the block commanding circular interpolation.

(h) The direct radius command programming is effective in:

LAP
Tool nose radius compensation mode
Subprogram
Incremental programming mode (G91)

In the direct radius command programming, the control automatically calculates the coordinate 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.

See the example in the following page.

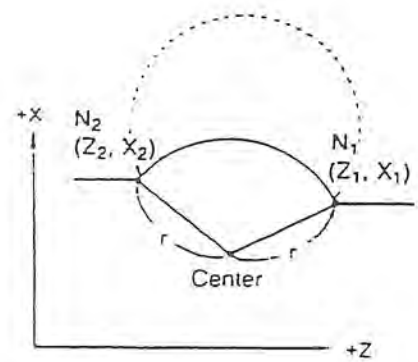
Program:

N1 G01 X_1 Z_1 F_1
N2 G03 X_2 Z_2 L_r

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 L_r



(6) Feedrates

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

NOTICE : (1) Omission of I or K is equal to I0 or K0.

(2) I and K values should be specified in radius.

(3) An arc extending in more than two quadrants can be specified by the commands in a single block.

(4) If either X or Z is omitted, circular interpolation is possible within a 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.

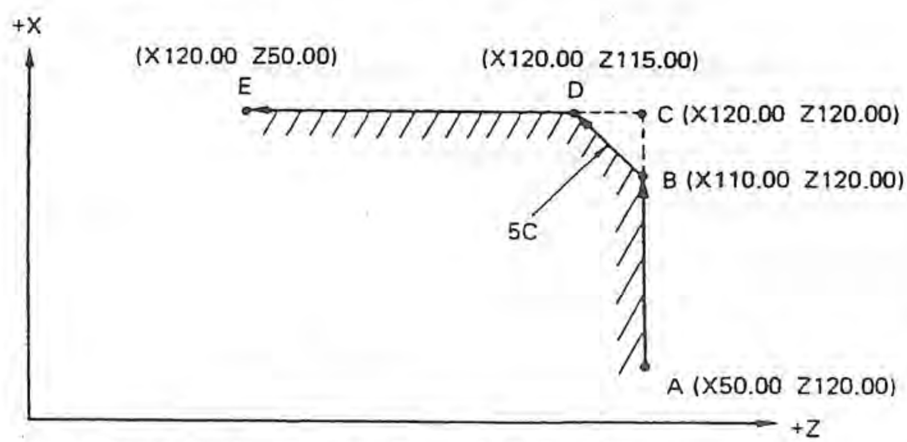
4. Automatic Chamfering

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 in a simple programming.

For chamfering at an arbitrary 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 "0" for this parameter.

4-1. C-chamfering (G75)



To cut the above shown contour along the Points A, B, D and E, program as:

G75 G01 X120 L-5 FOO CR

after positioning the cutting tool to 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 the size of 5 mm.

Explanation of the commands are provided in the following page.

G75: Specifies C-chamfering

X120: X coordinate of Point C

XL-5: Size of chamfered face

Its sign is determined by the direction of axis movement;

"+" ...... when Z-axis (X-axis) moves in the positive direction after X-axis (Z-axis) moved.

"-" ...... when Z-axis (X-axis) moves in the negative direction after X-axis (Z-axis) moved.

By commanding the coordinates of Point E, the cutting tool moves from Point D to Point E.

[Supplement]

1. G75 is effective only in G01 mode. If G75 is specified in other mode, it causes an alarm.
2. G75 is non-modal and active only in the commanded block.
3. If the axis movement dimension specified in the block calling for automatic chamfering is smaller than the absolute value of the L word, and alarm results.
4. 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 block, it results in an alarm.

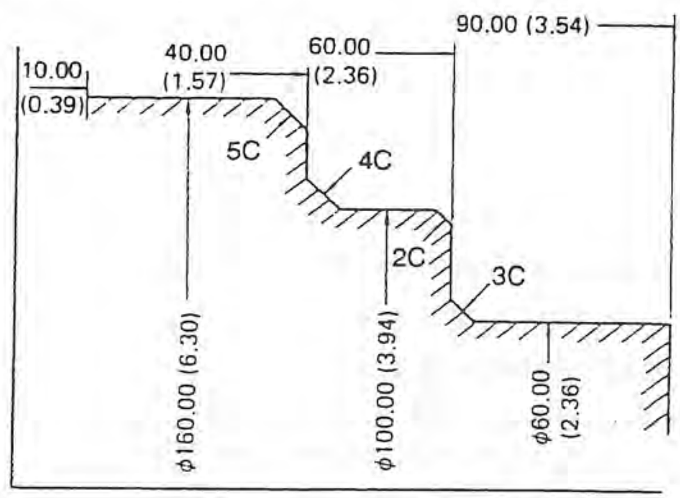
   The block calling for automatic chamfering mode can contain only one dimension word, either X or Z.
5. The automatic chamfering program is effective in:

   LAP

   Tool nose radius compensation mode

See the program example below.

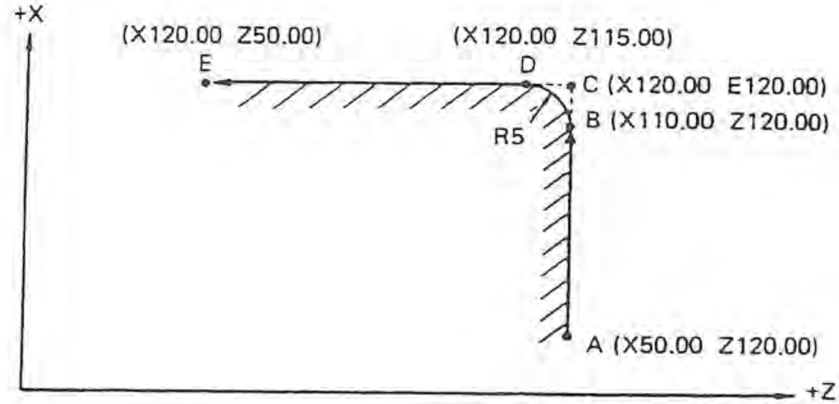
Example (Chamfering at 45°)



Unit: mm (in.)

```
  :
  :
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
  :
  :
```

4-2. Rounding (G76)



To cut the above shown contour along the Points A, B, D and E, program as:

G76 G01 X120 L-5 F○○ CR

after positioning the cutting tool to Point A.

With the commands above, the cutting tool moves from Point A to B and then to D, thus automatically rounding the corner at 5 mm radius.

G76: Specifies rounding of corner

X120: X coordinate of Point C

L-5: Radius of rounding circle

Its sign is determined by the direction of axis movement;

"+" ...... when Z-axis (X-axis) moves in the positive direction after X-axis (Z-axis) moved.

"-" ...... when Z-axis (X-axis) moves in the negative direction after X-axis (Z-axis) moved.

By commanding the coordinates of Point E, the cutting tool moves from Point D to Point E.

[Supplement]

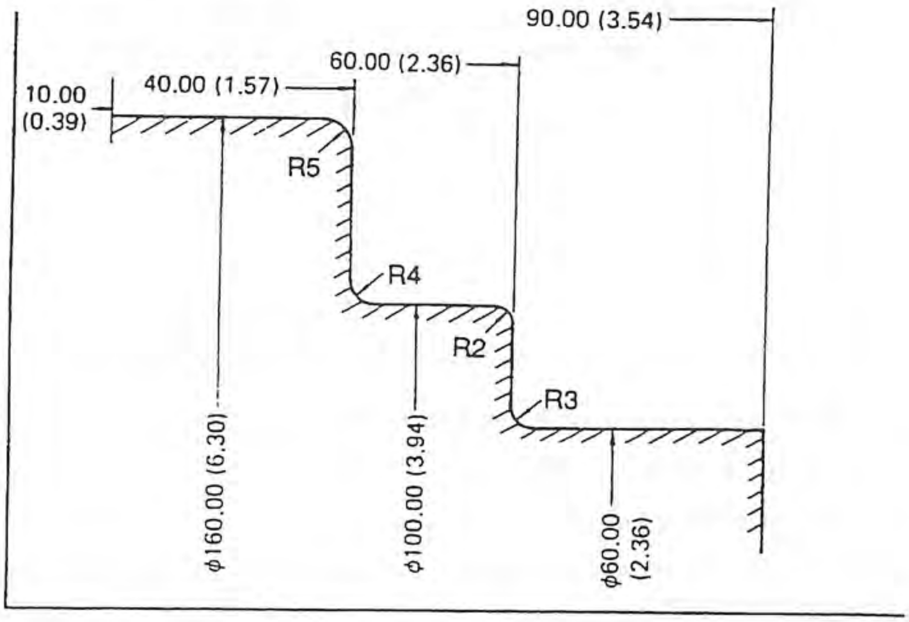
1. G76 is effective only in G01 mode. If G76 is specified in other than G01 mode, it causes an alarm.
2. G76 is non-modal and active only in the commanded block.
3. Rounding is made as 1/4 circle having the radius specified by an L word.
4. If the axis movement dimension specified in the block calling for automatic chamfering is smaller than the absolute value of the L word, an alarm results.
5. 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 block, it results in an alarm.

   The block calling for automatic chamfering mode can contain only one dimension word, either X or Z.
6. The automatic chamfering program is effective in:

   LAP
   Tool nose radius compensation mode

See the program example below.

Example (Rounding corner)



Unit : mm (in.)

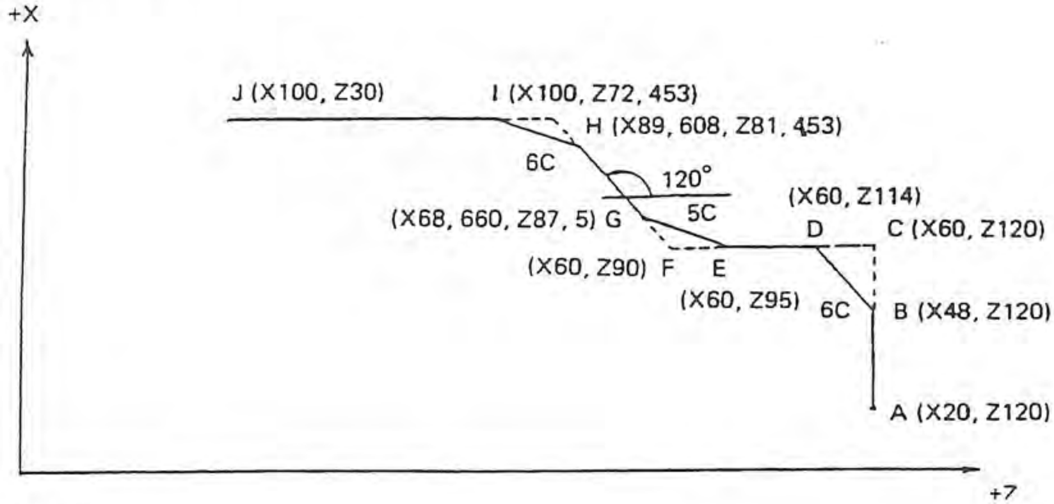
```
   :
N101    G01 X60 Z92 F0.1
N102    G76     Z60   F0.05  L3
N103    G76 X100             L-2
N104    G76     Z40          L4
N105    G76 X160             L-5
N106            Z10
   :
   :
```

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)



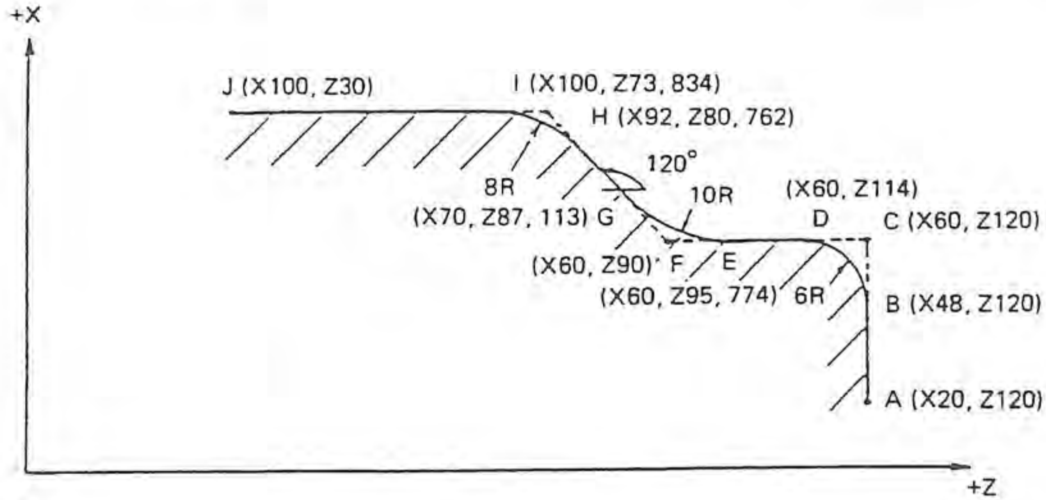
Program:

```
                                  :
                                  :
N1000   G00           X20    Z120
N110    G75    G01    X60           L6          F□□
N120    G75                  Z90    L5
N130    G75    A120   X100          L6
N140                         Z30
                                  :
                                  :
```

With the program above, cutting tool moves from point A to point J in the sequence of 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 Z-axis.

(2) R-Chamfering (G76)



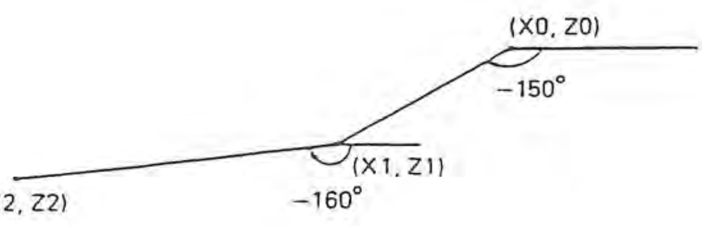
Program:

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

With the program above, cutting tool moves from point A to point J in the sequence of 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  F□      CR
       X2  Z2  A-160  CR
```

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

NOTICE :

(1) Both G75 and G76 are effective only in the G01 mode and if they are designated in other than the G01 mode, an alarm occurs.

(2) If 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 line and arc and between arc and line is impossible. If chamfering at such corners is attempted, an alarm occurs.

(4) Chamfering command is effective both in the LAP and nose radius compensation mode.

(5) If only an angle command A is designated in the G00, G01, G34 and 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

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 □ =

(Designate an axis to be fed, Z or W, for □.)

- The torque limit value is set in percentage, taking the rated torque of the axis feed servomotor as 100%.
- The maximum torque limit value is set at the optional parameter (word) No. 71.

5-2. Torque Skip Cancel Command

The torque skit 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.

Program Format:

G28

5-3. Torque Skip Command

Program Format:

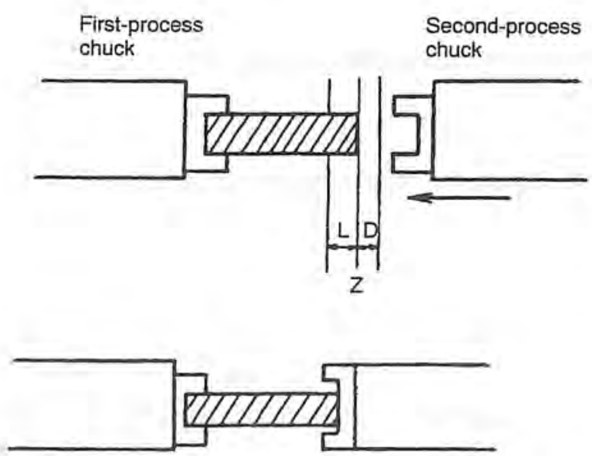
G22 Z __ D __ L __ F __ PZ = __

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

- Designate an axis to be fed for the target point and the set torque value. (
- 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.

When the preset torque value is small, it is reached during approaching motion, resulting in an occurrence of alarm 1219.



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

(1) The second-process chuck approaches to 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.

Feedrate F → $\frac{F}{5}$

Z : Target point

D : Distance between the target point and the approaching point in incremental value

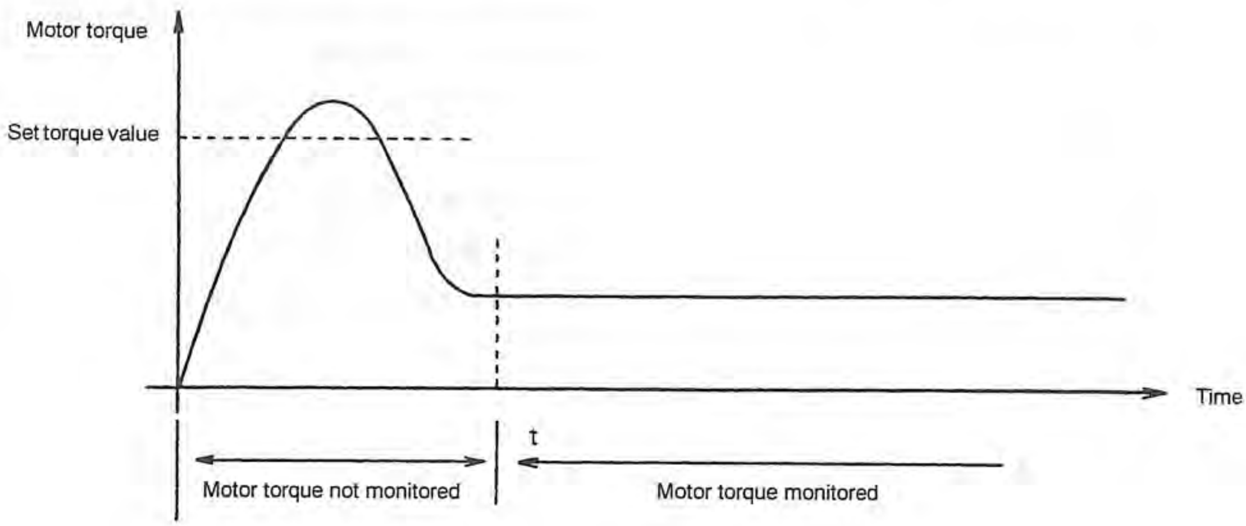
L : Distance between the target point and the virtual approaching point in incremental value

5-4. Parameter Setting

(1) Torque monitoring delay time

If motor torque monitoring is started with the start of torque skip feed designated by G22, there might be cases in which the preset torque value is exceeded at startup of the motor.

To avoid this, set torque monitoring delay time t at the parameter. Motor torque is not monitored for the time duration set at t.



Optional parameter (word) No. 70

Setting unit : 10 (ms)

Setting range : 0 to 9999

Initial setting : 0

(2) Upper limit for the P command value

The upper limit for the P command value in the G29 block can be set.

Optional parameter (word) No. 71

Setting unit : 1 (%)

Setting range : 1 to 100

Initial setting : 0

5-5. Program Example

This is a program example to transfer 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

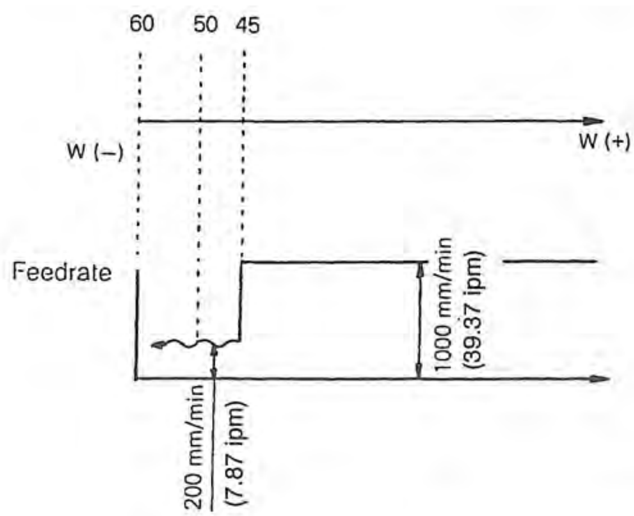
M84 .... Main spindle chuck open

G28 .... Cancels W-axis torque limit.

G90 G00 W300 .... Returns the W-axis to the retract position at the rapid feedrate.

:

:



6. STM Time Over Check Function

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

(1) The check function is set effective or ineffective according to the setting for the machine parameter.

Parameter: 1 .... The check function is effective.

0 .... The check function is ineffective.

(2) 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

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

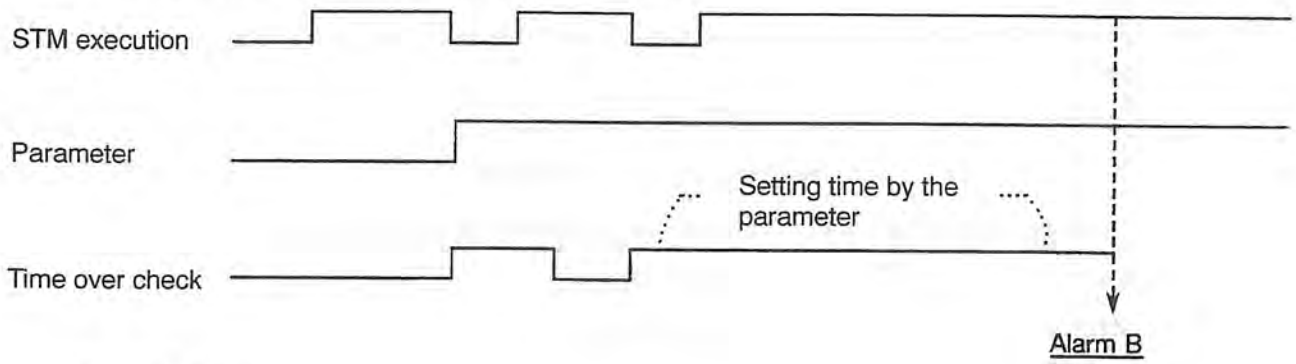
Maximum setting : 600 seconds

6-3. Timing Chart Example

(1) Parameter

Parameter ON ...... STM time over check start

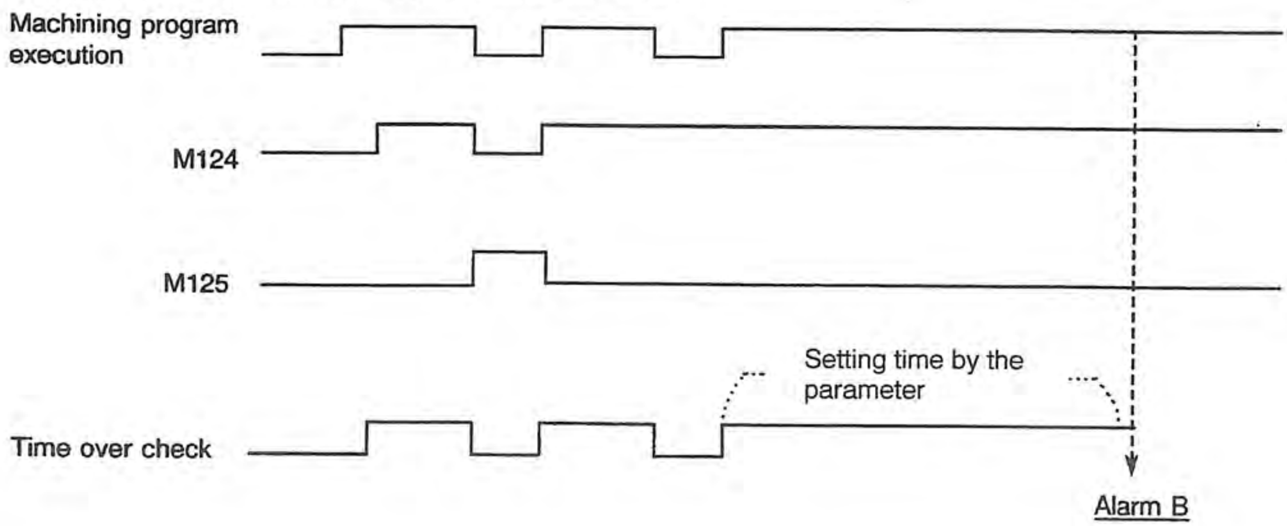
Parameter OFF ..... STM time over check end



(2) M Codes

M124 .... STM time over check start

M125 .... STM time over check end



SECTION 14 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).

(1) Effective G Code Ranges

One-shot : A one-shot G code is effective only in a specified block and is automatically canceled when it moves to the next block.

Modal : A modal G code is effective until it is changed to another G code in the same group.

(2) 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 halfway during a block. Note, however, that a "/" (block delete) and a sequence name may be placed before a special G code.

1. Dwell (G04)

If dwell is specified, execution of the next block is suspended for the specified length of time after the completion of the preceding block.

Program 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 allowable maximum length of dwell period is 9999.99 seconds.

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

2-1. Zero Shift

With G50 code, zero offset value is automatically calculated and zero setting according to the calculated value is made.

This feature is effective when cutting a workpiece having the repetitive same contour on it.

Function:

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

Program Format:

G50 X__ Z__ C__

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

Example:

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

With the program provided above, positioning is made to the coordinate point (X0, Z0) by the commands in N004 block, first. When the commands in N005 are executed, the coordinate system is newly established so that (X0, Z0) where positioning has been made and the axes are now located has the coordinate values (X1, Z1) specified following G50.

In this program, the origin of the coordinate system is shifted:

$X = X0 - X1$

$Z = Z0 - Z1$

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.

NOTICE : (1) Zero offset of the axis not specified in the block containing G50 is not performed.

(2) G50 is non-modal and active only in the programmed block.
(Calculation of zero offset is made 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 G50 code.

2-2. Max. Spindle Speed Set

There are cases that 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, and other factors. This feature allows the setting of the maximum spindle speed in such cases.

Program Format:

G50 S_ _ _ _

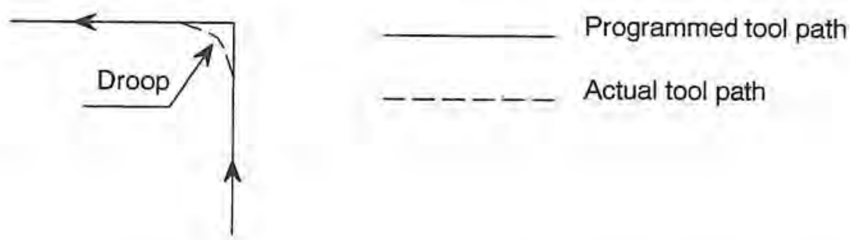
S: Specify the maximum spindle speed.

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

3. Droop Control

The axis movements on NC machines are controlled by the 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 correctly 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 generation of functions (pulses) at the corner until the DIFF reaches the preset permissible droop amount.

Commands Used for Droop Corner Control Function

G64 ...... Droop corner control, OFF

The control is placed in G64 mode, when it is turned ON.

G65 ...... Droop corner control, ON

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.

Permissible droop amount

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)

Specify G95 to control tool movement (feedrate) by "distance per spindle revolution" for performing turning operation.

Program Format:

G95 F__

F: Specify tool movement distance per spindle revolution.

The unit of setting is determined according to the setting for optional parameter (UNIT)

The allowable maximum feedrate depends on the machine specifications.

At turning on the power or after reset, the feed per revolution mode is selected.

5. Feed Per Minute (G94)

Specify G94 to control tool movement (feedrate) by "distance per minute" for performing turning operation.

Program Format:

G94 F__

F: Specify tool movement distance per minute.

The unit of setting is determined according to the setting for optional parameter (UNIT)

The allowable maximum feedrate depends on the machine specifications.

6. Constant Speed Control (G96/G97)

When the constant speed cutting function is selected, cutting at a constant cutting speed can be performed. This feature can reduce cutting time and also assure stable finish in end face cutting operation.

(1) Constant Speed Cutting Command

Program Format:

G96 S_ _ _ _

S: Specify the cutting speed.

Setting unit: m/min

(2) Canceling Constant Speed Cutting

Program Format:

G97 S_ _ _ _

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

(3) Example

```
:
:
N○○○    G96  S100     G96 calls for constant speed cutting mode and the
:                     commands following this block are all executed in
:                     this mode.
:                     S100 ....... 100 m/min.
:
:
:
N○○○○   G97  S500     G97 cancels G96 mode, and cutting after this block is
                      carried out at a spindle speed of 500 min-1 {rpm}.
```

NOTICE :

(1) If the spindle speed exceeds the maximum or minimum speed allowed within the range selected by an M code while 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 X-axis is moved large distance in rapid traverse rate while constant speed cutting mode, from the turret indexing position toward the workpiece for instance, it causes chucking manner.

Therefore, positioning the cutting tool near the workpiece or returning it to the turret indexing position or other operation causing large X-axis travel must be commanded after canceling constant speed cutting mode.

(3) The block containing G96 or G97 must contain an S word.

(4) Thread cutting program cannot be provided in G96 constant speed cutting mode.

(5) To activate constant speed cutting mode on turret B, specify G111 with G96. To restore such mode on turret A again, specify G110.

(6) To execute the commands over two blocks continuously while controlled in constant speed cutting mode without waiting for spindle speed arrived signal, specify M61. To cancel it, specify M60.

SECTION 15 OFFSET FUNCTION

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

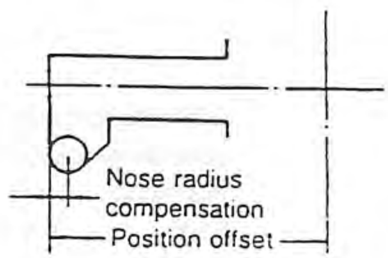
1-1. General Description

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 by simple programming.

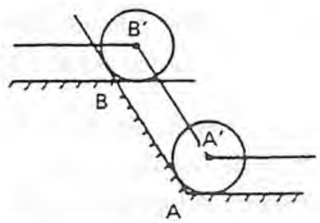
1-2. Tool Nose Radius Compensation for Turning Operation

(1) Tool Offset and Nose Radius Compensation

In turning operation, 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, drill, etc. Accordingly, the tool nose radius compensation function has to be activated simultaneously with the tool offset function.



(2) Tool Nose Radius Compensation at Discontinuous Point

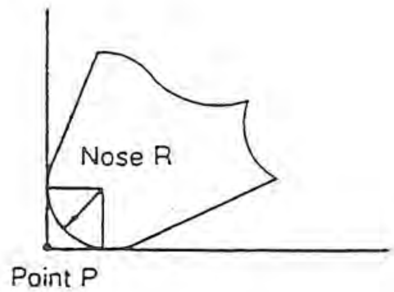


Point A in the figure above constitutes a discontinuous point and an angle less than 180°. By using the tool nose radius compensation function, the tool path shown above can be generated by simply entering the coordinates of points A and B.

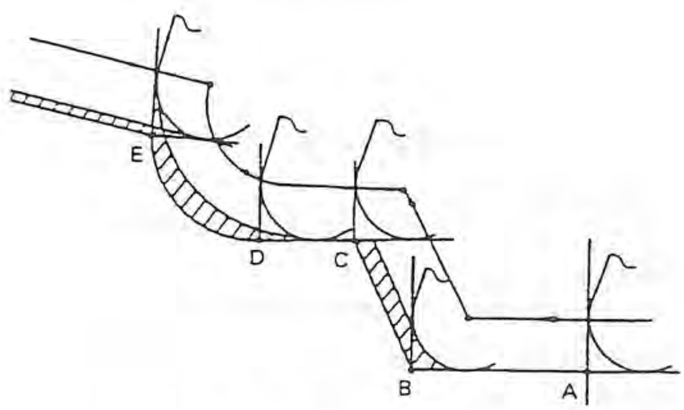
1-3. Compensation Operation

(1) Geometrical Cutting Error due to Tool Nose Radius

If cutting along paths A-B-C-D-E in the figure below is intended without activating the tool nose radius compensation function, the portions indicated by hatching lines will remain uncut and cause a geometrical errors. This is because the tool setting is made to locate the imaginary cutting point P at the datum point and trace 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.



Tool Setting Point

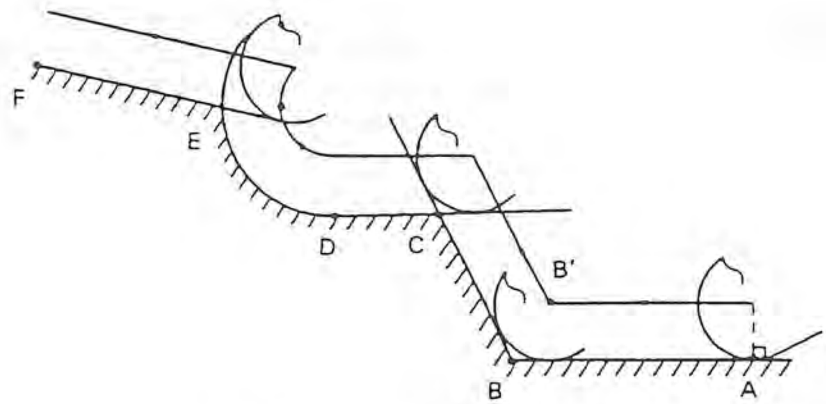


Tool Path and Resulting Error
– Without Tool Nose Radius Compensation

The tool nose radius compensation function automatically compensates the inconsistency between the designated and actual tool paths caused by the tool nose radius.

(2) Compensation Movement

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



Tool Path with Tool Nose Radius Compensation

1-4. Programming

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

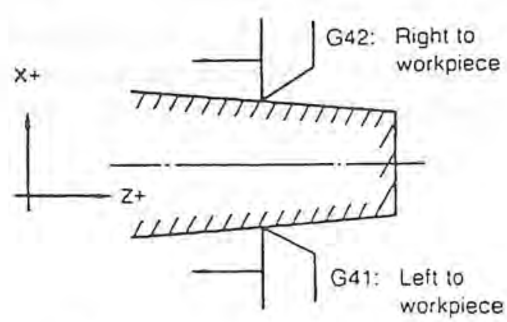
(1) G Codes

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.

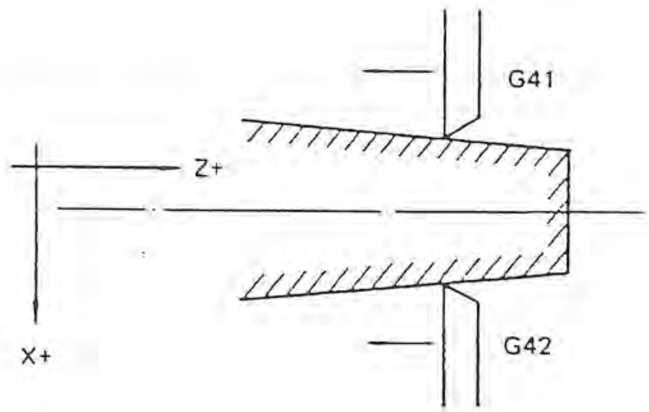
G42: Tool nose radius compensation - Right
Used when the tool moves on the right side of the workpiece.

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



Designation of G41 & G42

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



(2) T Codes

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

T○○△△□□

○○: Tool nose radius compensation number

△△: Tool number

□□: Tool offset number

NOTICE : To change 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	

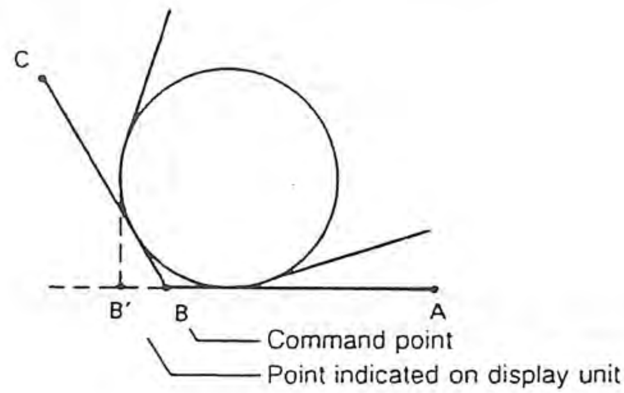
Entry of only the tool offset No. (T01 or T11) in G code command (1) or (2) will cancel the nose radius compensation amount.

1-5. Display

The screen displays various data. The data displayed in the tool nose radius compensation mode are as follows.

(1) Actual Position

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



(2) Alarm Display

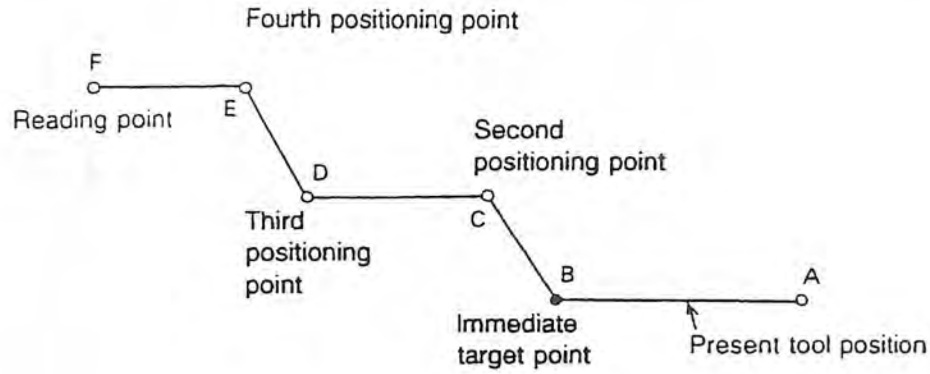
If an alarm in relation to the tool nose radius compensation function occurs, the ALARM light under STATUS DISPLAY goes on and the screen displays the associated alarm message indicating its contents.

1-6. Operation

(1) Conditions of Nose Radius Compensation Function

(a) 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 buffer. This is called the 3-buffer function.

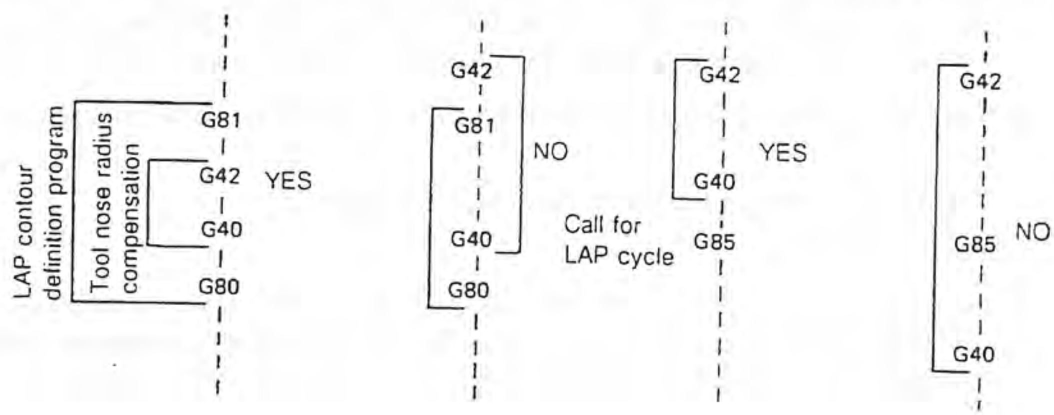
When the tool nose radius function is activated, the target point E is calculated from straight lines DE and EF. This means that the data in the block four blocks ahead the current target point are read if the tool nose radius functions is active.



Data in Buffer

(b) Nose radius compensation during LAP mode

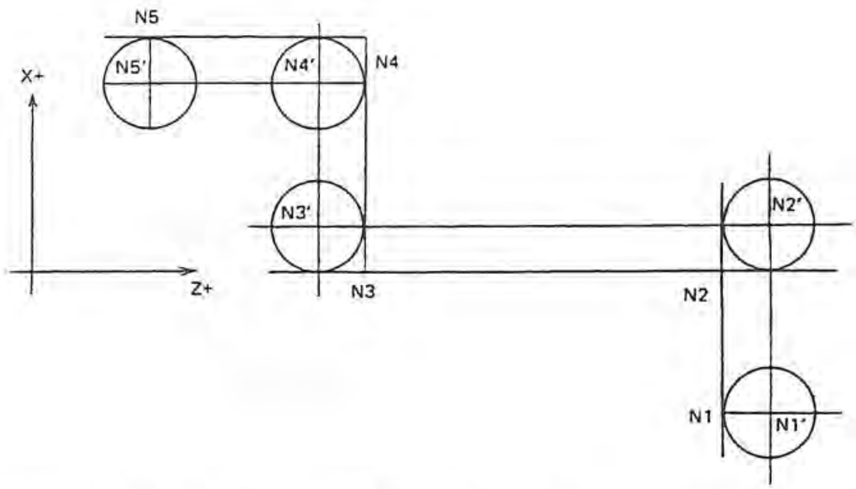
To use the tool nose radius compensation function in the LAP mode, programs for the respective turrets must contain the tool nose radius compensation programs independently as shown below.



(2) 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
```



To obtain point N2' when the center of the tool nose R is at point N1', proceed as follows:

(a) Draw a straight line parallel to the tool advancing direction, 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'.

(b) Draw a straight line parallel to the tool advancing direction, N2 - N3, offset in the specified direction, (to right 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'.

(c) The nose R center for commanded point N2' in the point of intersection of these two straight lines.

The center of tool nose R advances from point N1' to N2'.

To obtain point N3':

(a) Draw a straight line parallel to the tool advancing direction, N2 - N3, offset in the specified direction, (to the right 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'.

(b) Draw a straight line parallel to the tool advancing direction, 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'.

(c) The nose R center for commanded point N3 is the point of intersection of these two straight lines.

The center of tool nose R advances from point N2' to point N3'.

To obtain point N4':

Follow the same procedure indicated above using points N3, N4 and N5.

1-7. Application of Tool Nose Radius Compensation Programming

(1) G41 and G42

The G41 and G42 codes are used to call out the tool nose radius compensation mode. Since the use 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 advancing direction.

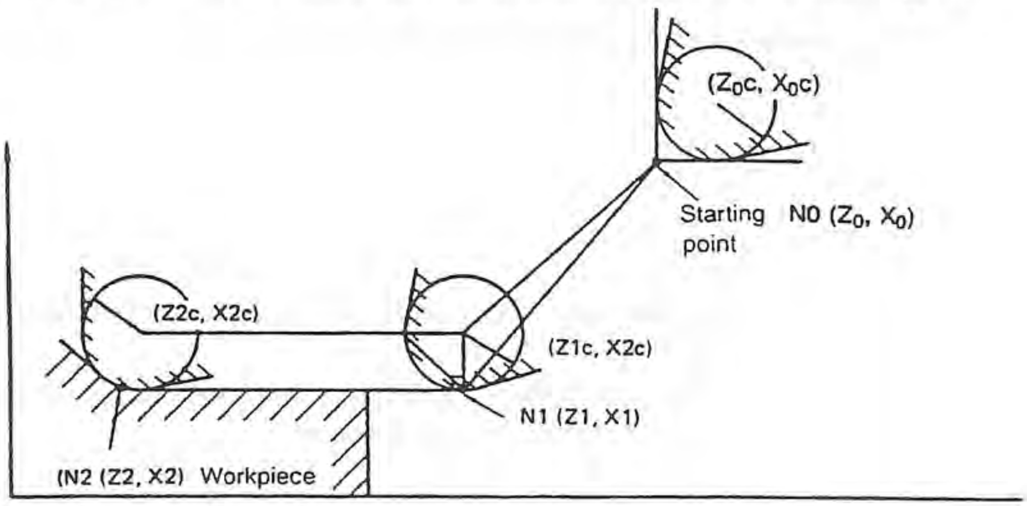
G42: This tool nose radius compensation code is used when the cutting tool moves on the right side of the workpiece in terms of its advancing direction.

(2) Behavior on Tool Nose Radius Compensation Mode Entry

(a) General description

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

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



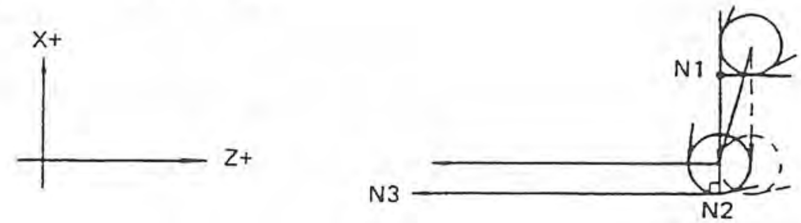
Without the tool nose radius compensation function, positioning is made so that the tool tip reference point is located exactly at the programmed coordinates. At the start up of the tool nose radius compensation mode activated by either G41 or G42, positioning is carried out so that the tool tip circle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

At the start up of the tool nose radius compensation mode, both X- and Z-axis might move even if the block contains only one dimension word, either X or Z. (See Program Example 1.)

(b) Program examples

1) Example 1

```
N1   G00  X100  Z100  S1000  T010101  M03
N2   G42  X80
N3   G01        Z50   F0.2
```



Although the programmer might expect the axis movement indicated by broken lines because 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.

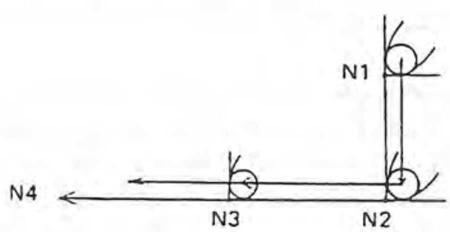
2) Example 2

The desirable program for entry into the compensation mode:

```
N1   G00  X100  Z100  S1000  T010101  M03
N2        X80
N3   G42        Z90
N4   G01        Z50   F0.2
 :
```

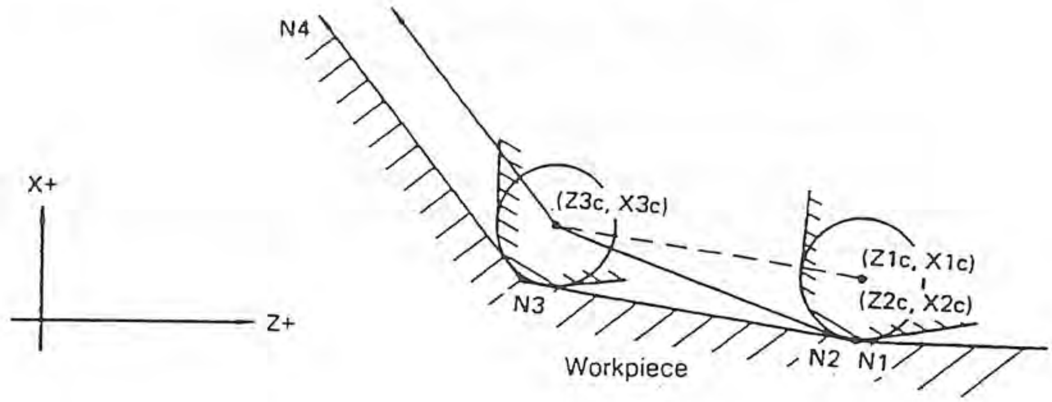
In this program, the G42 block contains only a Z word, and points N2, N3 and N4 are all positioned on the same straight line.

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



3) Example 3

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




```
N1    G01   X50    Z100   F0.2   S1000   T010101   M03
N2    G42
N3          X60    Z80
N4          X100   Z50
```

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 performed 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.

4) Faulty program example 1

```
N1    G01   X50    Z100   F0.2   S1000   T010101   M03
N2    G42
N3          X50    Z100
N4          X60    Z80
N5          X100   Z50
```

Since sequence N3 designates the point identical to the one designated in the start-up sequence N2, an alarm occurs.

5) Faulty program example 2

```
N1   G01   X50   Z100   F0.2   S500    T010101   M03
N2   G4
N3                             S1000
N4                             M08
N5         X50   Z100
N6         X60   Z80
```

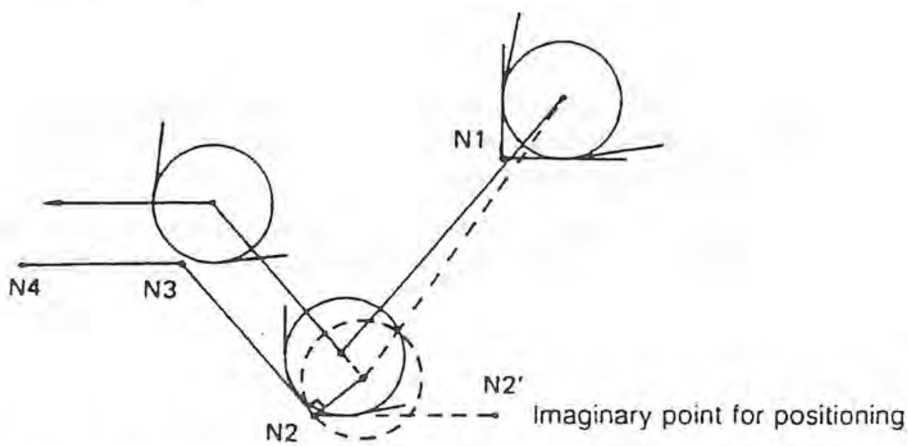
Since sequences N3 and N4, the successive two sequences after the start-up of the tool nose radius compensation mode, do not contain X and Z axis movement commands, an alarm occurs.

(c) 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.

Example:

```
N1   G00   X100   Z100   F0.2   S1000   T010101   M03
N2   G42   X 60   Z 80   K20
N3   G01   X 80   Z65
N4                Z50
 :                 :
```



If block N2 containing G42 has no I and K words, positioning of the cutting tool by the commands in block N2 is executed so that the tool nose R comes into contact with line N2-N3 at designated point N2 and then moves along the path indicated by broken lines at point 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.

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 such word to have "0" value. Therefore, KO in the above program can be omitted.

(3) Behavior in Tool Nose Radius Compensation Mode

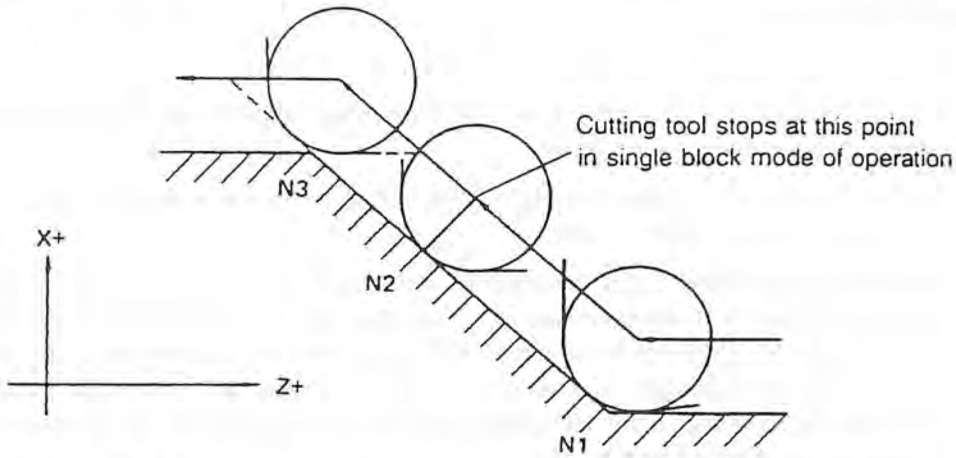
The tool nose radius compensation function provides means to automatically compensate for the tool nose radius in continuous cutting.

Since such compensation is made automatically, there are some restrictions in programming when the tool nose radius compensation function is used.

(a) Straight line to straight line cutting

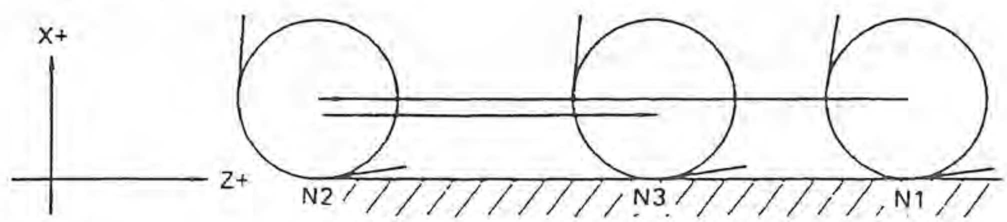
1) Midpoint on a straight line

When specifying a midpoint on a straight line, the point should be commanded carefully.



When point N2 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.

2) Returning along straight line



Such axis movement causes no problem when the program is made without using the tool nose radius compensation function.

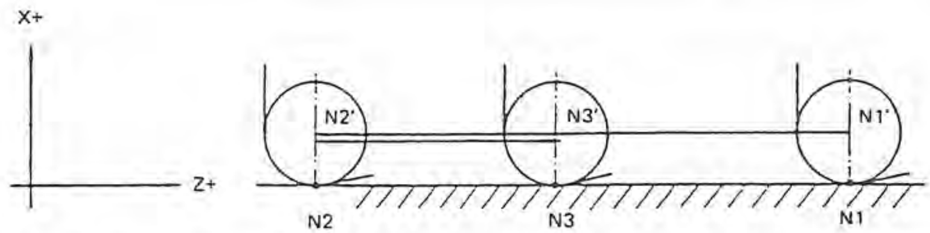
However, with the use of this function the axis movements must be programmed carefully.

Example:

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

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 tool advancing direction. However, in tool returning motion from point N2 to point N3, the cutting tool is on the left side of the workpiece with respect to the tool advancing direction. Therefore, G41 is specified instead of G42.



The axis movements above are possible by the special processing for the tool nose radius compensation function.

When the program above is executed, the actual path taken by the center of the tool nose R is obtained as follows.

The center of the tool nose R (N2') at point N2 is obtained as;

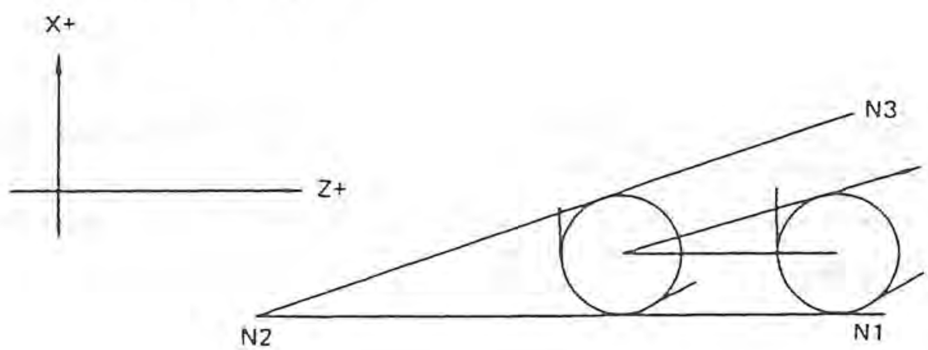
- The line parallel to the straight line N1 - N2 is obtained, with an upward offset (G42) by the tool nose radius amount effective at N1.
- The line parallel to the straight line N2 - N3 is obtained, with an upward offset (G41) by the tool nose radius amount effective at N2.
- The center of the tool 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'.

The center of the tool nose R (N3') at point N3 is obtained in the same manner as in a).

In this way, the program in 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, it will cause the tool path to be shifted considerably from the expected path.

3) Two lines making an acute angle

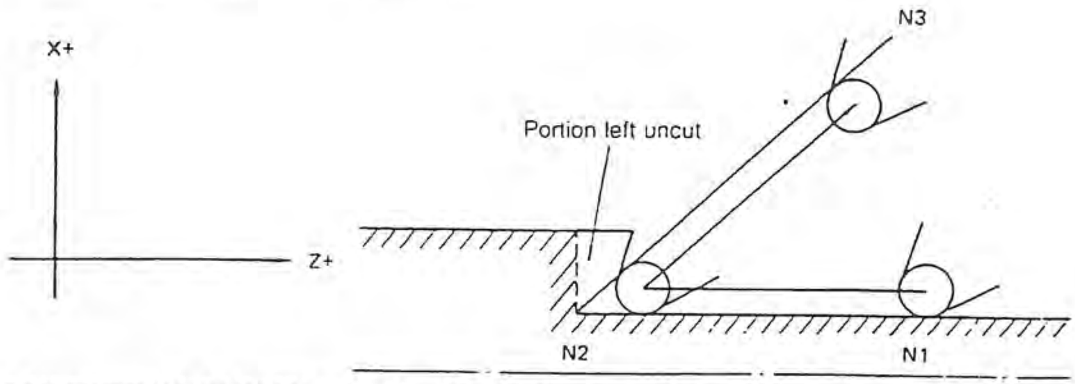


As is evident from the figure above, the cutting tool cannot reach point N2 even when positioning from point N1 to point N2 is intended. This is because the cutting tool can move up only to the point where the tool nose R comes into contact with line N2 - N3.

The example provided in the previous page illustrates a case where programmers are apt to be confused. Another example is provided below:

Example of faulty program 1:

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

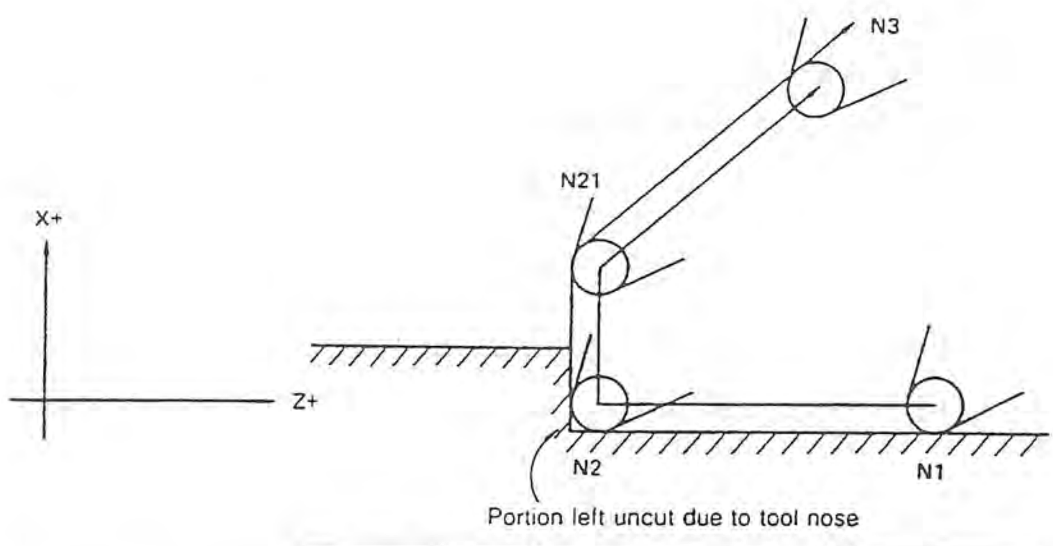


The programmer expected to cut up to point N2, (i.e., up to Z50) allowing a slight uncut portion on the sharp corner due to tool nose R. Contrary to this intention, however, the cutting tool leaves a considerable amount of uncut section since it stops before reaching the desired point.

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  M03
N2                   Z50
N21            X104  [> 100 + 4 × (nose R)]
N3   G00       X300  Z300                        M05
```



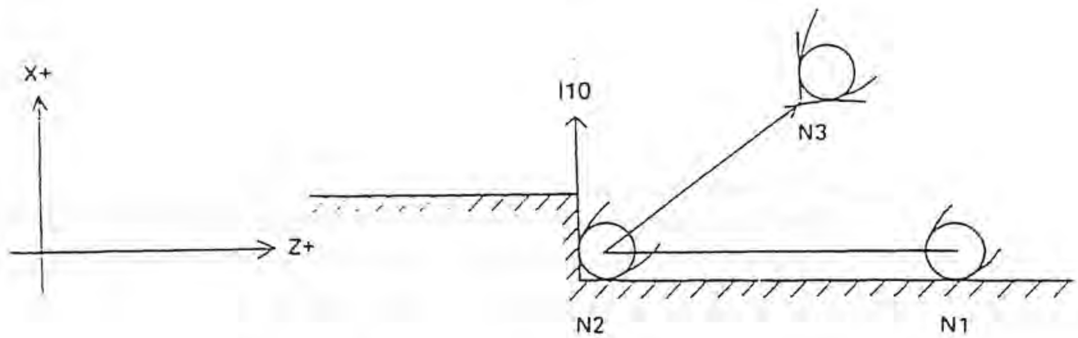
The improved program generates the tool path shown above, and almost all the cutting can be accomplished as expected except for a slight uncut section due to the tool nose R.

For relieving the tool along X-axis in the positive direction in 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, as an X word is expressed in diameter, 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 N1 - N2 surface.

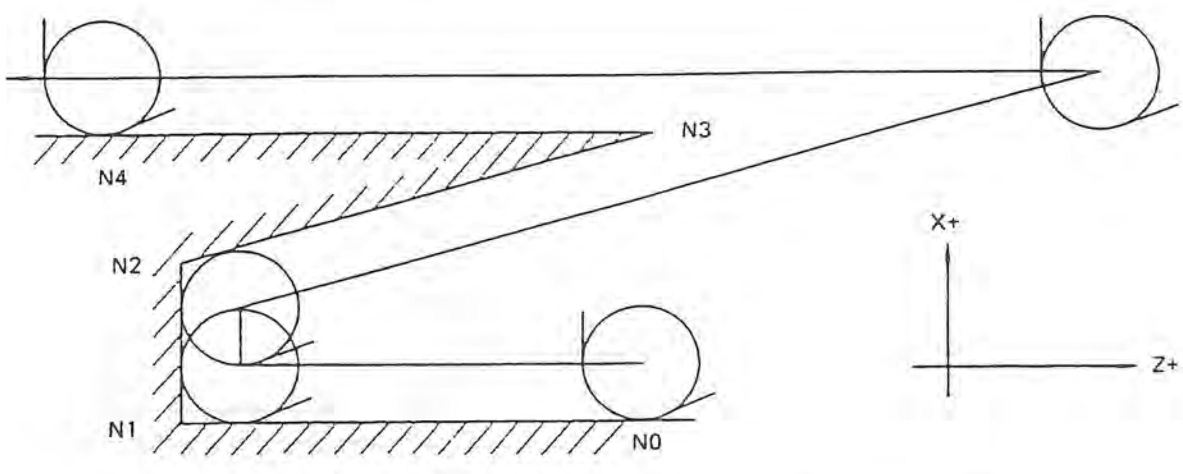
Example of improved program 2:

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



The G40 command in N3 cancels the tool nose radius function. At point N2, the cutting tool moves so that the tool nose R will come into contact with line N1 - N2 and I10 vector standing at point N2.

4) Two lines making an obtrude angle



In this section, program to feed the cutting tool along path N0 - N1 - N2 - N3 - N4.

As shown in the figure, angle N2N3N4 is an acute angle and the cutting tool moves along the line outside of that angle. Therefore, for point N3, the cutting tool moves to a point away from it.

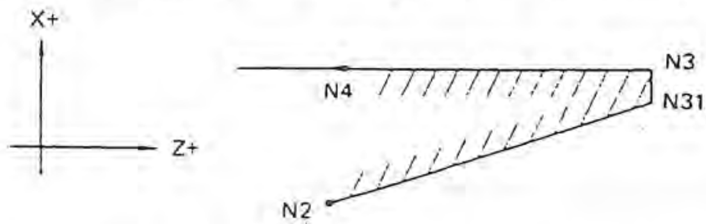
When preparing a program in which cutting similar to that contour is required, it is necessary to check the safety of the tool motion so that it does not strike against obstacles when moving to such a distant point.

Example:

```
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
```

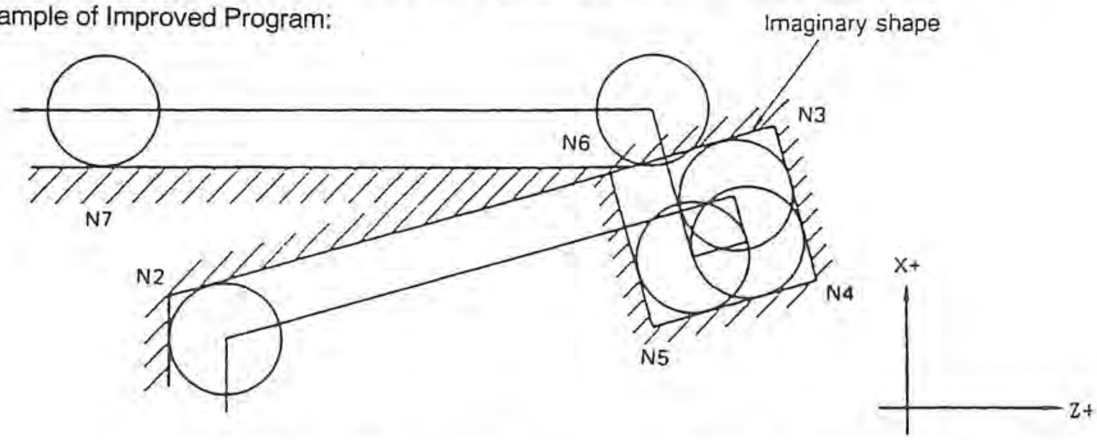
It is advantageous to improve the program and eliminate a positioning sequence to a distant point through commands in N3 block.

If N2N3N4 is not a sharp angle, such a problem would not occur. To eliminate sharp angles from the required contour, interposing a short straight line N3 - N31 is one possible solution.



In some cases, such a modification is not possible; to cut a sharp angle without positioning the cutting tool at a distant point, follow the steps detailed below.

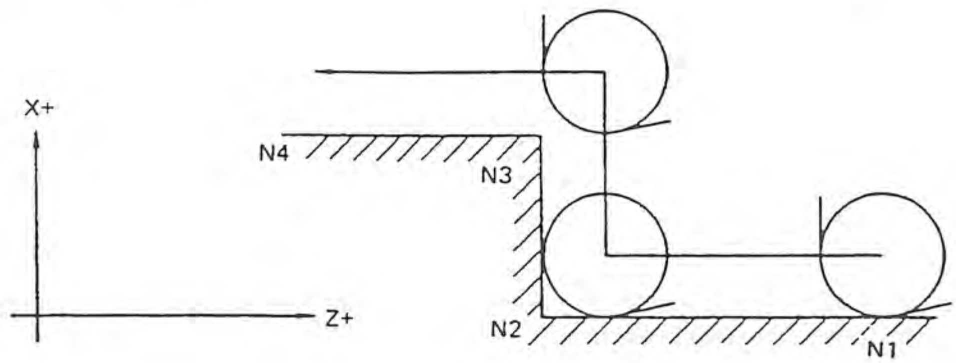
Example of Improved Program:




```
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
```

In this program, the cutting tool moves along 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.

5) Two lines forming a right angle



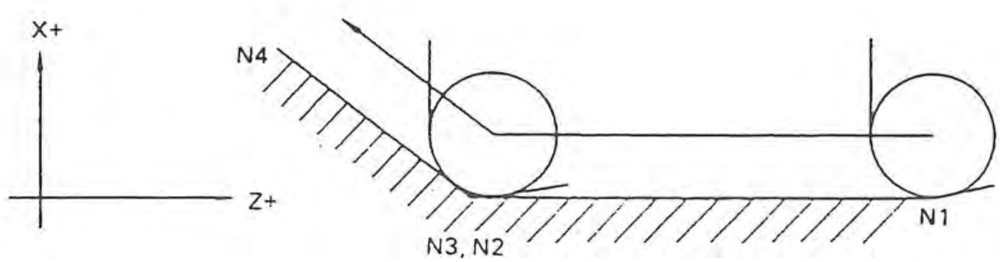
N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M03
N2				Z60				
N3			X150					
N4				Z20				

There are no particular problems in this case.

6) Command of identical point

- If a block without axis movement commands is provided during the tool nose radius compensation mode, the path of the tool nose R is quite the same as the one generated when such a block is not provided.

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



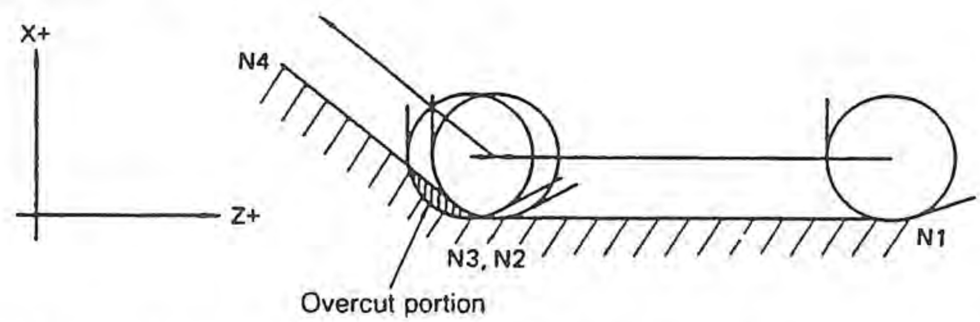
- When two or more blocks without axis movement commands are provided or when the same point as commanded in the preceding sequence is repeatedly provided during the tool nose radius compensation mode:

  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   S100   T010101   M04
N2                     Z80
N3                     Z80                            M08
N4               X60   Z70
```

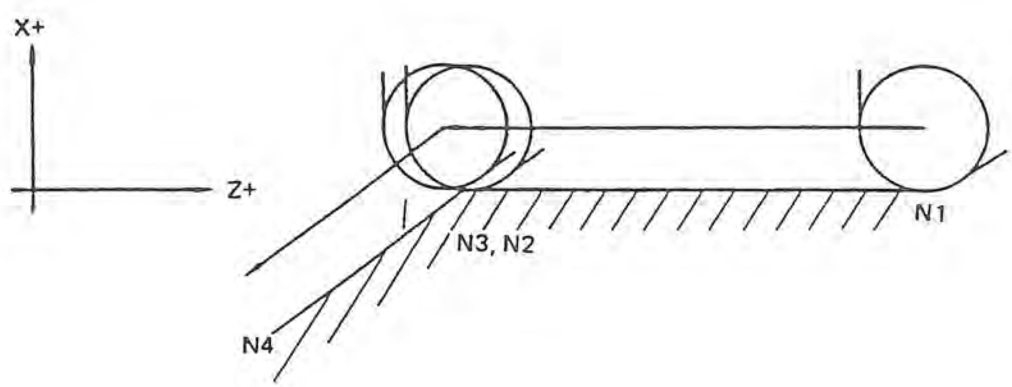
  The program might cause overcutting as below:



  Depending on the contour to be cut, such unexpected motion does not result in overcut as seen in program 2.

  Program 2:

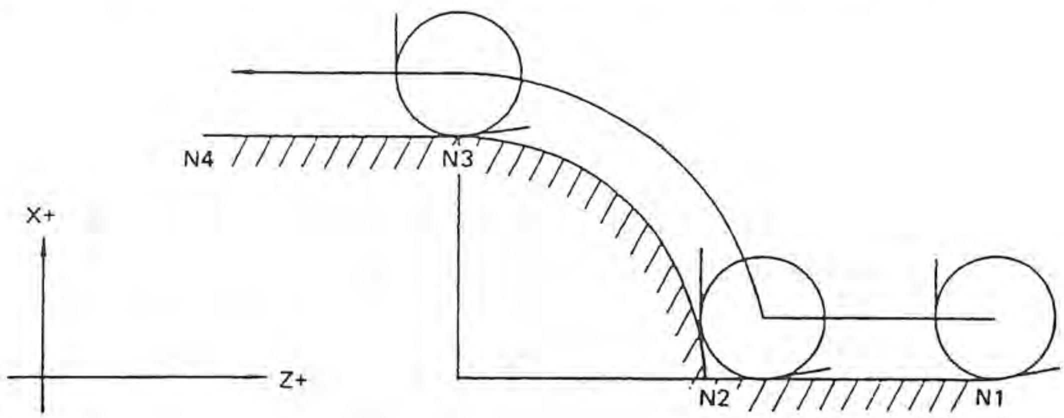
```
N1   G42   G01   X50   Z100   F0.2   S100   T010101   M04
N2                     Z80
N3                     Z80                            M08
N4               X40   Z70
```



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

1) 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 the case from a straight line to a straight line.



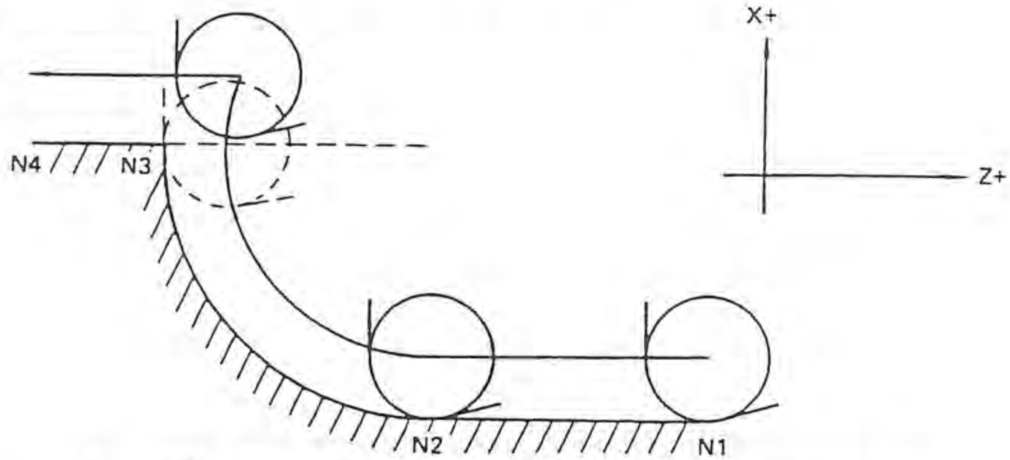
N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3	G03		X140	Z60	K-20			
N4	G01			Z40				

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 comes into contact at point N3.

When the cutting tool moves from point N3 to point N4, 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.

2) Arc in two quadrants

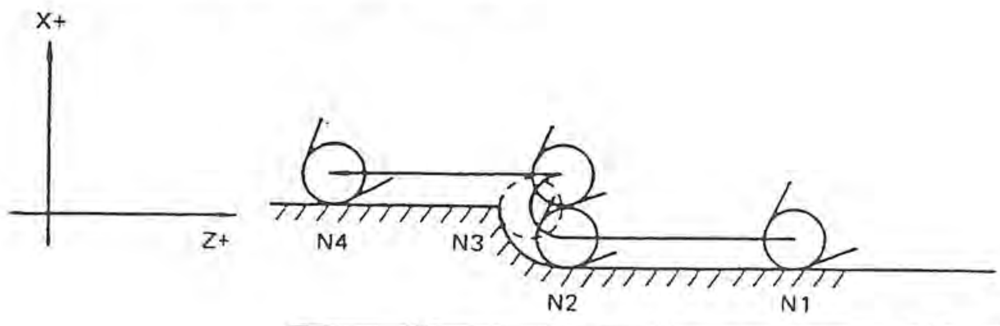
Case where the arc radius is greater than "2 × nose R":




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

The tool position determined by the commands in N2 block is the point where the tool nose R comes into contact with line N1 - N2 at point N2. In 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.

Case where the arc radius is equal to "2 × nose R":

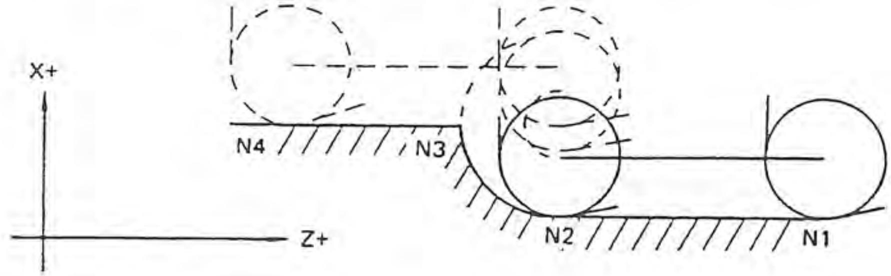


Tool nose R = 0.8 mm

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

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. That is, the cutting tool is positioned right above point N2.

Case where the arc radius is less than "2 × nose R":
(Impossible)



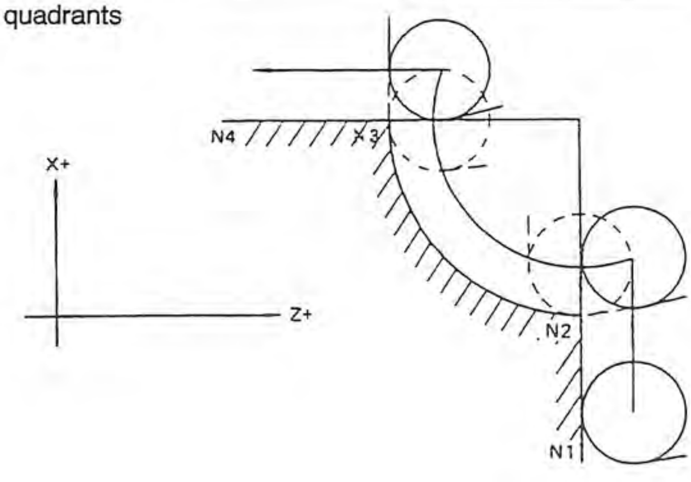

```
N1   G42   G01   X100   Z100   F0.2   S1000   T010101   M04
N2                      Z80
N3   G02         X102   Z79    I1
N4   G01                Z40
```

With the commands in block N3, positioning of the cutting tool is intended 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.

NOTICE : When cutting inside an arc, the radius of the arc must be equal to or greater than twice the tool nose R:

$R \geqq 2 \times R_N$
R = arc radius
R_N = tool nose R

3) Arc in three quadrants



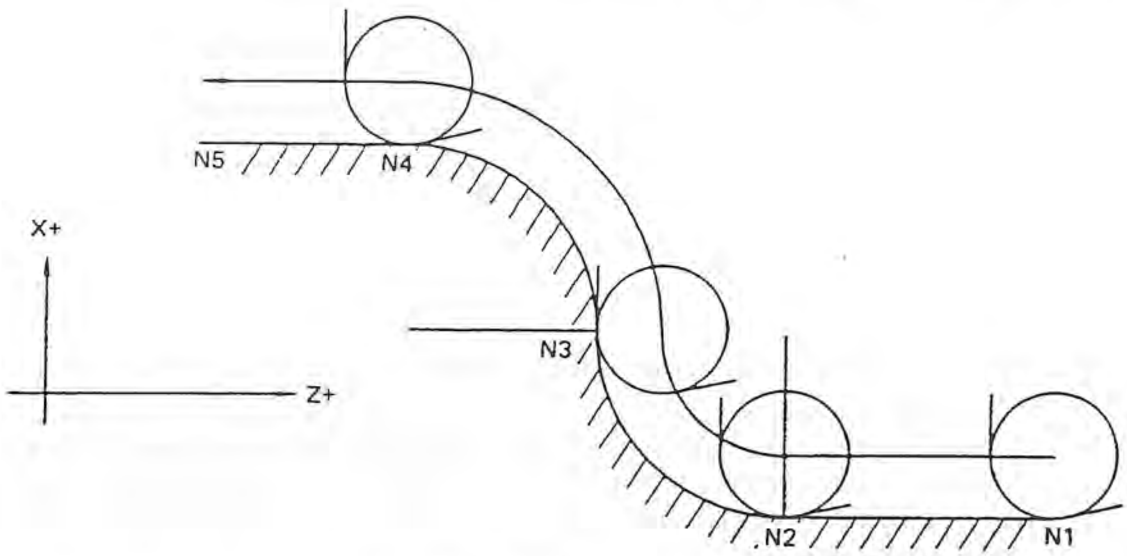

```
N1   G42   G01   X100   Z100   F0.2   S1000   T010101   M04
N2               X120
N3   G02         X160   Z80    I20
N4   G01                Z60
```

Positioning by the commands in block N2 is carried out at 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 explained in 2).

(c) Arc to arc cutting

Arc to arc cutting can be programmed in the same manner as 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				

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.

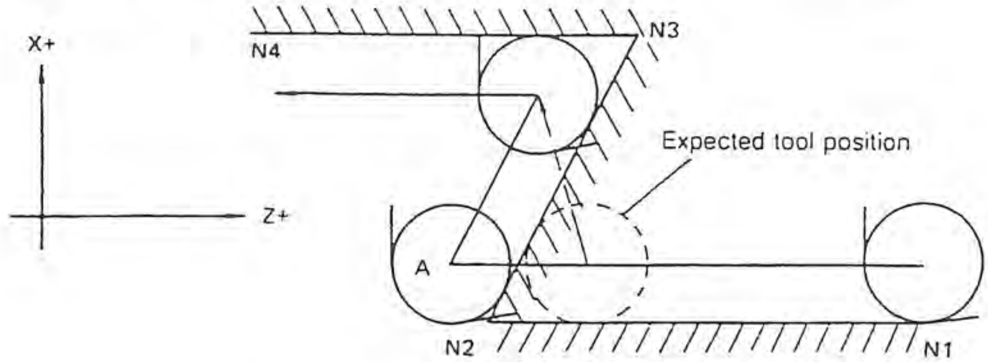
Other motions of the cutting tool are as explained in b), "Straight line to arc cutting".

(d) Switching from G41 to G42 or from G42 to G41

To switch the tool nose radius compensation mode from G41 to G42 or from G42 to G41, it is recommended to cancel the compensation mode once 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.

1) Switch-over in straight line to straight line cutting



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	

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.

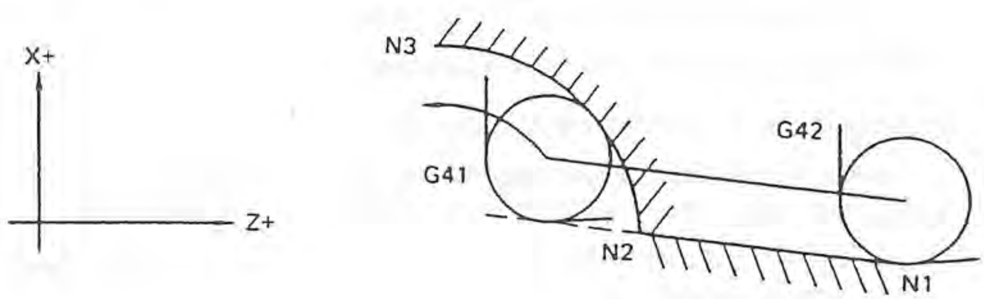
Positioning in block N2 is carried out at the left side of straight line N2 - N3.

2) Switch-over in straight line to arc cutting

The motion of the cutting tool will be easily understood. The concept is the same as in 1).

Example:

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

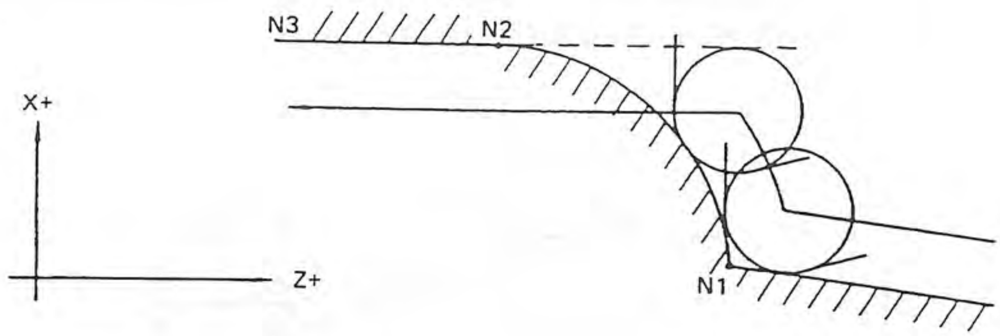


3) Switch-over in arc to straight line cutting

The motion of the cutting tool will be easily understood. The concept is the same as in 1).

Example:

```
N1   G42   G01   X1   Z1   F1   T
N2   G03         X2   Z2   I2   K2
N3   G41   G01   X3   Z3
```

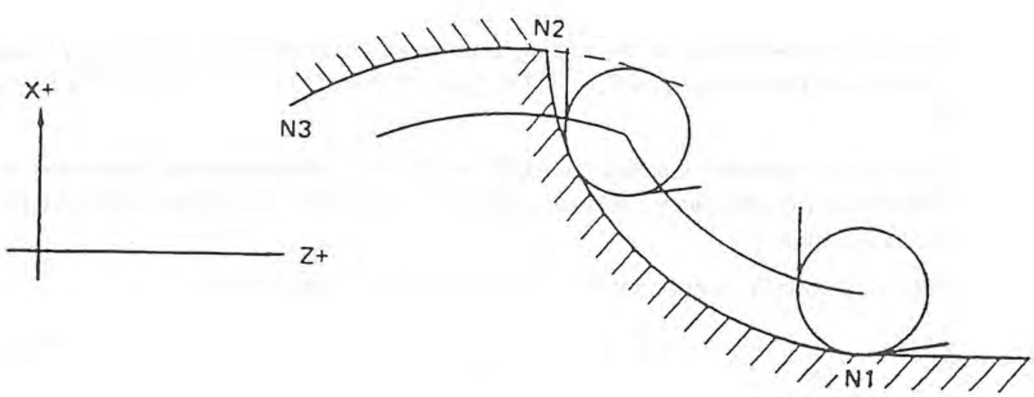


4) Switch-over in arc to arc cutting

The motion of the cutting tool will be easily understood. The concept is the same as in 1).

Example:

```
N1   G42   G01   X1   Z1         F1          T
N2   G02         X2   Z2   I2          Z2
N3   G41         X3   Z3   I3          Z3
```

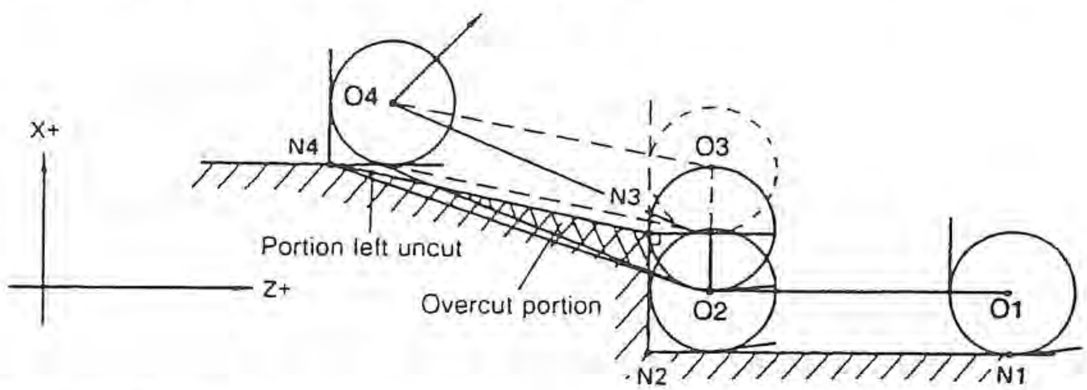


(4) Behavior with Tool Nose Radius Compensation Mode Cancel

(a) 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 resulted from the cancel of the compensation mode to avoid unexpected troubles.

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. Meanwhile 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.



When cutting the contour comprising straight line segments as illustrated above, it is programmed as 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
```

With the commands above, the cutting tool moves along the path indicated by broken lines. That is, positioning for designated Point N3 is made to Point O3 and to Point O4 for Programmed Point N4.

The uncut part parallel to straight line N3 - N4 is left. Therefore the tool nose radius compensation function will be effectively used for cutting such contour accurately. See the programs in the following pages.

When 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
```

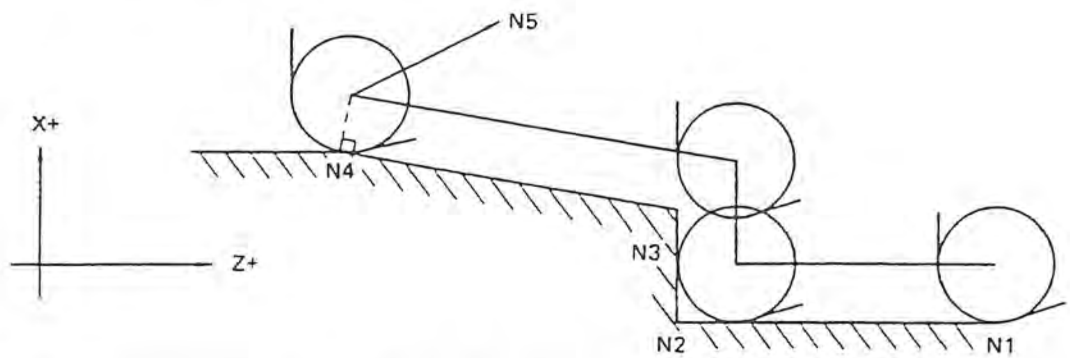
The tool path generated in the above program is shown by solid lines:

Positioning at programmed point N3 is carried out at the point where the tool nose R comes into contact with point N3, and that at programmed point N4 is carried out to 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
```



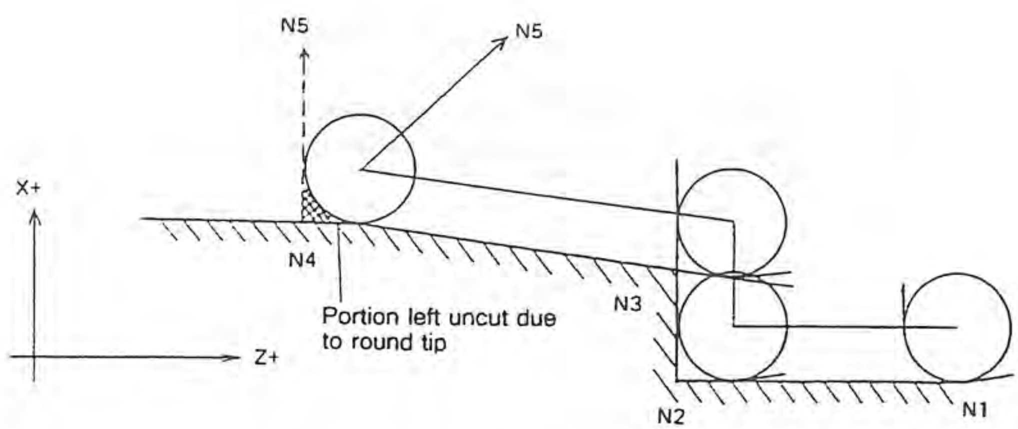
To cut the exact contour up to Point N4, the G40 command which cancels the tool nose radius compensation mode is specified in block N5.

Although the program yields almost the expected contour, the tool nose R goes beyond the designated point N4 along Z-axis since the tool nose R comes into contact with line N3 - N4 at point N4. When such overtravel does not interfere or overcut, there are no problems.

To eliminate such possible overcutting along Z-axis, see the program below.

Program:

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

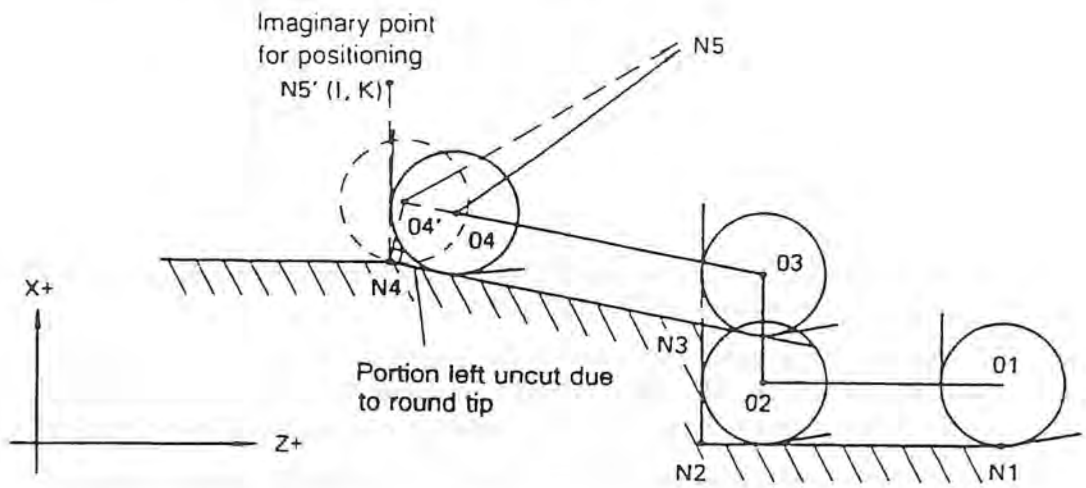


I and K words specified in the G40 block allows the tool to move to the point where the tool nose R is brought into contact with both line N3 - N4 and line N4 - N5.

(b) 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
```



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 at point N5.

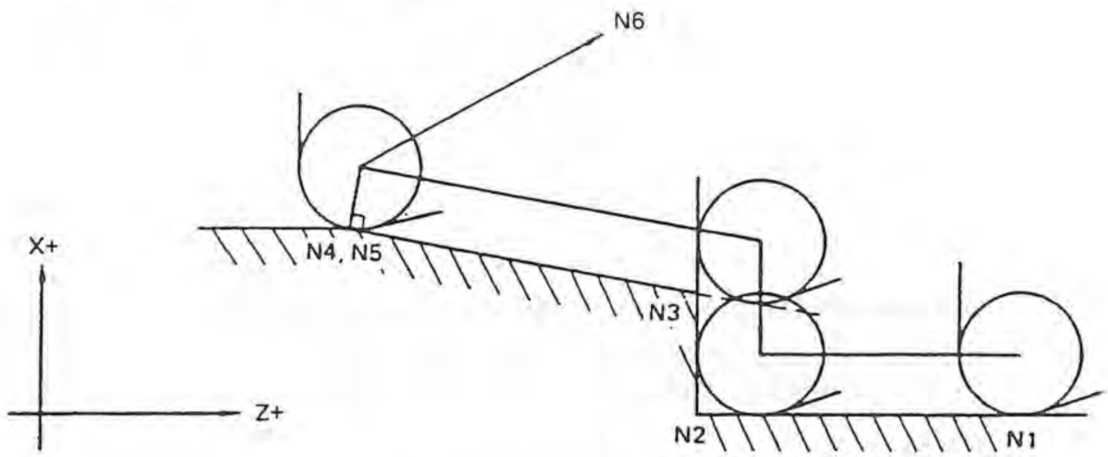
Addition of I and K words in block N5 positions the cutting tool to the point where the tool nose R is brought into contact with straight line N3 - N4 and imaginary straight line N4 - N5' when the commands in block N4 are executed. Execution of the commands in block N5 brings the cutting tool to the programmed point N5 where the tool nose radius compensation is not active.

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

(c) Independent G40

When G40 code is programmed without other commands in one 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
```



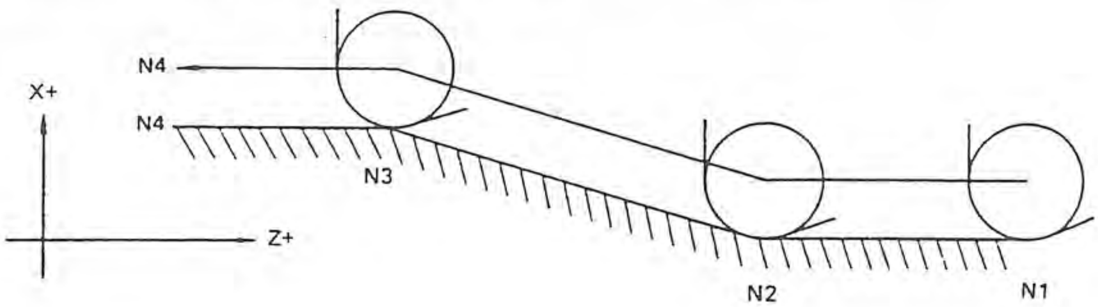
While canceling the tool nose radius compensation mode (G40), the mode of operation must be either G00 or G01. If not, an alarm occurs.

(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 made to finish the programmed dimensions. However, such powerful function requires careful programming while interrupting the continuous cutting 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.

The original contour and its associated program are provided below:



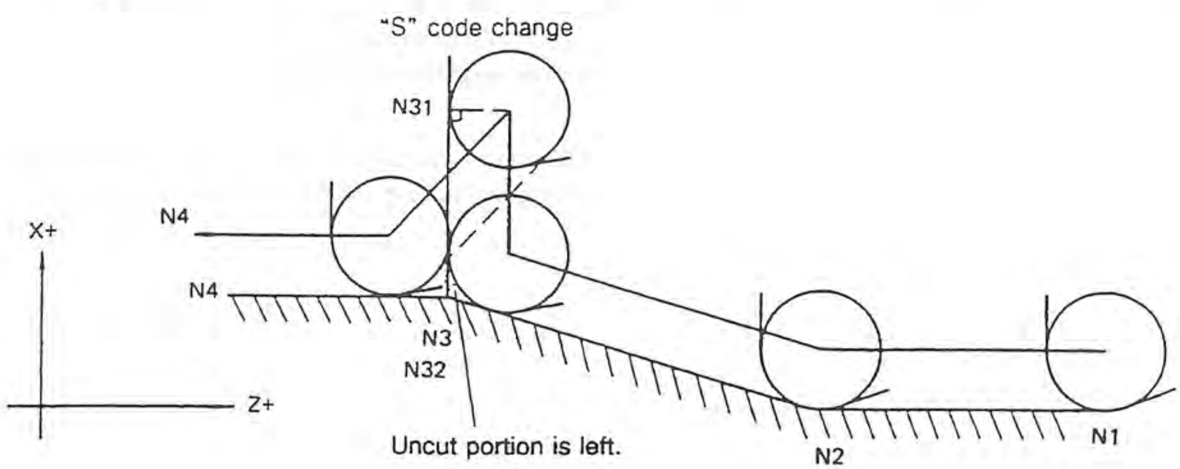

```
N1   G42   G01   X100   Z100   F0.2   S1500   T010101   M03
N2                      Z80
N3               X120   Z40           S1000
N4                      Z20
```

The original contour comprises; straight line - slope - straight line and has no complicated section.

Based on this program, the program to relieve the cutting tool at point N3 in the +X direction in order to change the spindle speed is made. See the programs provided hereafter.

(a) 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
```



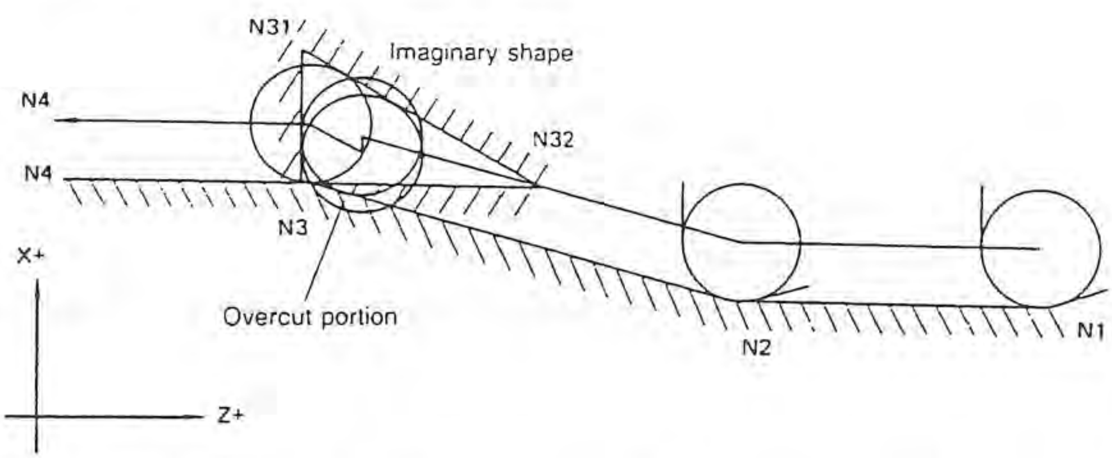
The cutting tool is relieved at point N3 in the +X direction, spindle speed is changed and then continuous cutting is intended.

In this program, 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 side of the positioning with respect to the line is on the right. To the contrary, commands in block N32 position the cutting tool to 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 tool advancing direction. This causes the cutting tool to move not only in X-axis direction but in Z-axis direction although block N32 contains only X word.

Such cutting tool movements leave an uncut portion as shown above.

(b) 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
```



In this program, elimination of the uncut portion caused by Program 2 is intended. Although the uncut portion is eliminated, this program caused overcutting, instead; what caused the overcutting is explained hereafter.

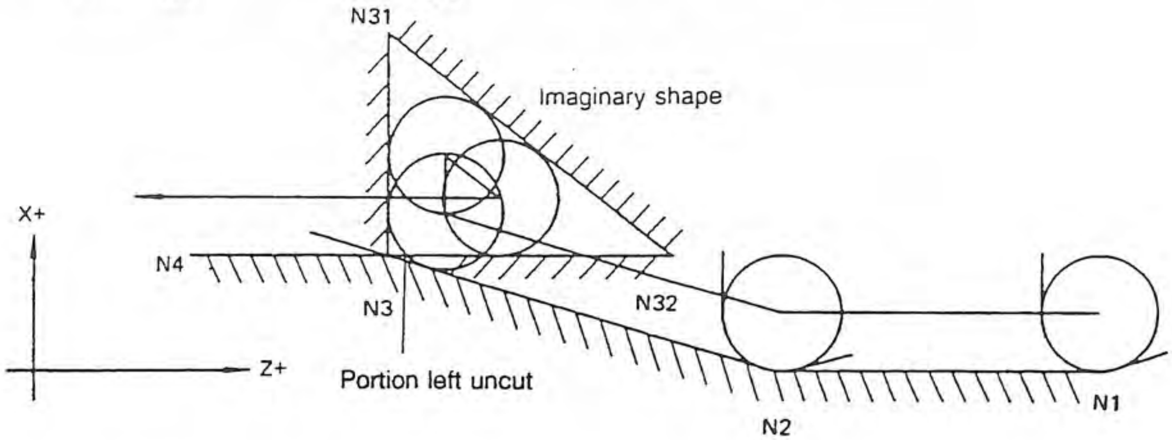
When the control feeds the cutting tool from point N2 to point N3, it reads the position data of point N31 as well as those of point N3. This permits the tool nose R to be positioned at the point where the nose R is in contact with the two straight lines N2 - N3 and N3 - N31.

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 made 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 expected from 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.

(c) Program 4

```
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
```



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

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

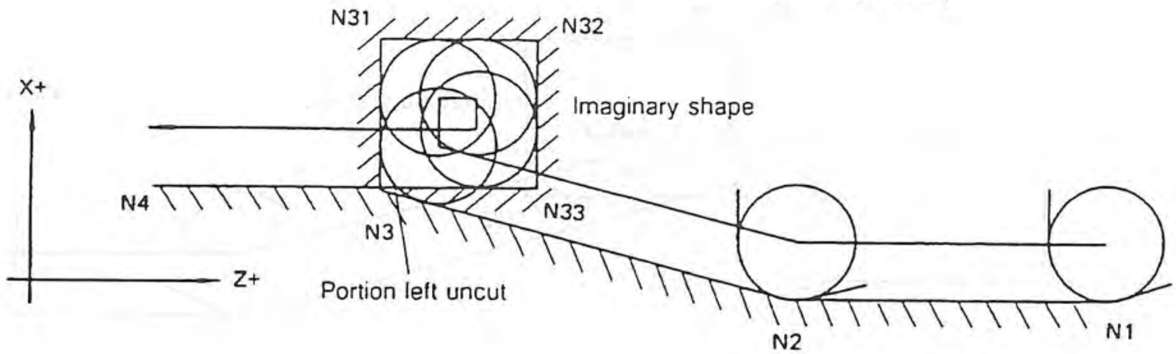
- overcutting is caused depending on the size of the tool nose R
- the length of side N31 - N32 cannot be readily found.

These problems are solved by looping the tool path along a square as explained later.

(d) Program 5

```
N1   G42   G01   X100   Z100   F0.2   S1500   T010101   M03
N2                      Z80
N3               X120   Z40
N31  G00         X124 (*a)
N32                     Z42 (*b)        S1000
N33  G01         X120
N4                      Z20
```

*a: $> 120 + 4 \times$ (nose R)
*b: $> 40 + 2 \times$ (nose R)



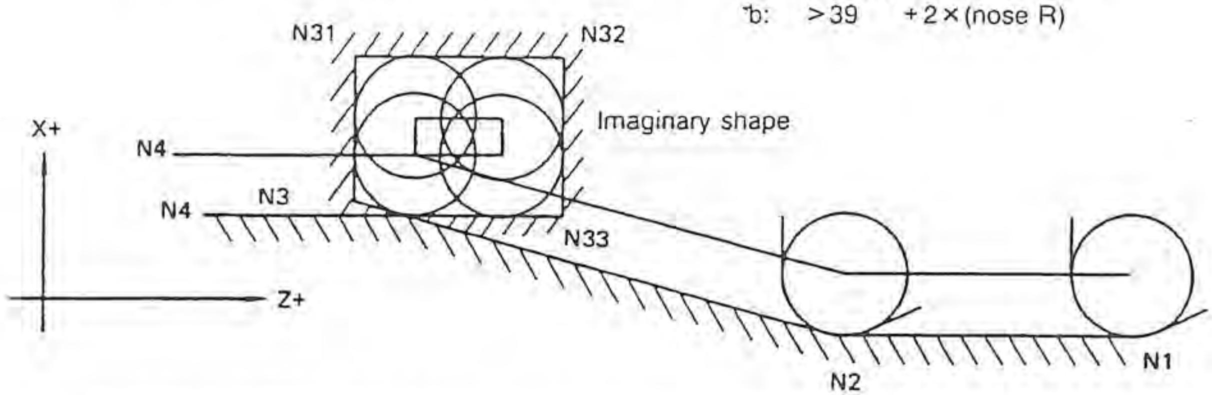
In this looping path, the tool nose R moves inside the programmed rectangle, N3 - N31 - N32 - N33. Therefore, axis behavior can be easily expected if only these respective sides are longer than twice the tool nose R (four times on X-axis).

Since this program leaves an uncut portion as specified, it can be further improved as indicated in Program 6.

(e) Program 6

```
N1    G42   G01    X100    Z100     F0.2   S1500   T010101   M03
N2                         Z80
N3                 X120.5  Z39
N31   G00          X124 (*a)
N32                        Z42 (*b)        S1000
N33                X120
N4    G01                  Z20
```

*a: > 120.5 + 4 × (nose R)
*b: > 39 + 2 × (nose R)



Point N3 is shifted in the –Z direction as much as tool nose R to eliminate the uncut part seen in Program 6.

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.

NOTICE : (1) If either X- or Z-axis exceeds the respective soft-limits, 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 the LAP mode operation, provide G41 or G42 in the block preceded by the one containing G81 or G82 in which dimensions of cutting in LAP are specified. In the LAP mode operation, the 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 tool nose radius compensation mode, the same point should not be commanded in succession. However, one block that does not contain axis motion commands can be provided; 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 CYCLE START button, in 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

(1) Function Overview

This function automatically offsets the tool paths to generate the required shape by programming the final shape.

Using this function, different-diameter cutters can be used to machine the workpieces of the same shape without modifying the program.

(2) Programming

(a) Designation of offset plane (G17, G18, G119)

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 a proper G code.

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)

G17 and G19 are effective only while the C-axis is jointed.

When the C-axis control cancel command (M109) is executed, X-Z plane (G18) is automatically selected.

When the power is turned on or the control is reset, X-Z plane (G18) is selected.

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

G codes are used to turn on and off the cutter radius compensation function.

G40 : Cutter radius compensation function OFF

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

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

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

(c) Cutter radius compensation values

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

T○○△△□□

○○: Tool offset number

△△: Tool number

□□: Tool nose radius compensation number

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 becomes effective.

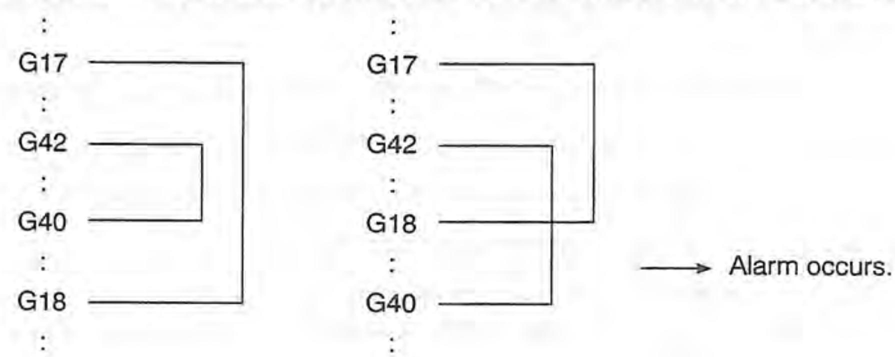
The nose R pattern number is effective only in the G18 (nose R compensation) mode. In the G17 and G119 (cutter radius compensation) modes, it is ignored.

(d) 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.

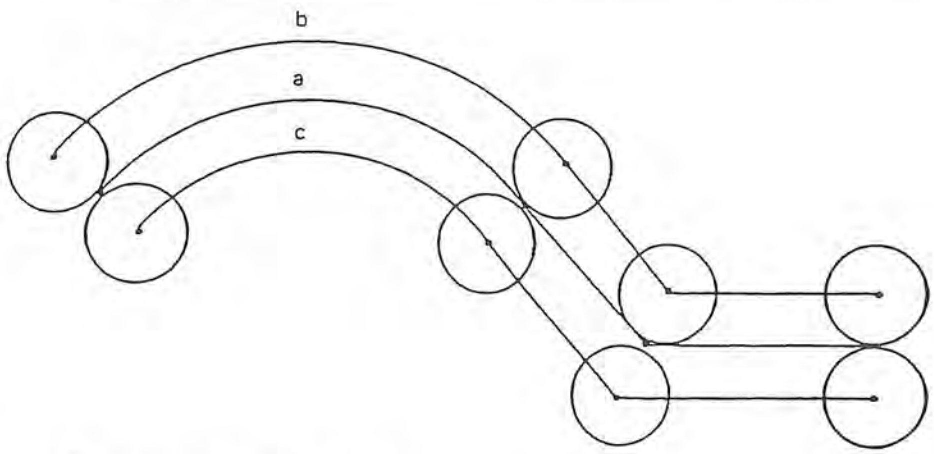
To change the compensation plane, cancel the cutter radius compensation function by designating the G40. If G17, G18 or G119 is designated in the G41 or G42 mode, an alarm occurs.



Correct Wrong

(3) Operations

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



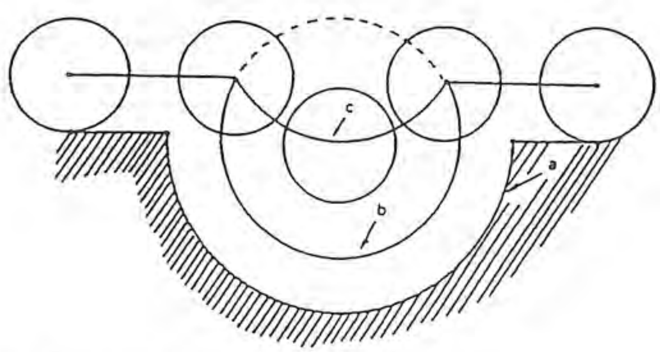
a : Programmed path (final shape)

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".

CAUTION : (1) 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 Fig. 15-1. This is because the contour generation function selects an arc with a center angle of less than 180° from the possible two 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°)

Fig. 15-1

CAUTION

: (2) 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 Fig. 15-2.

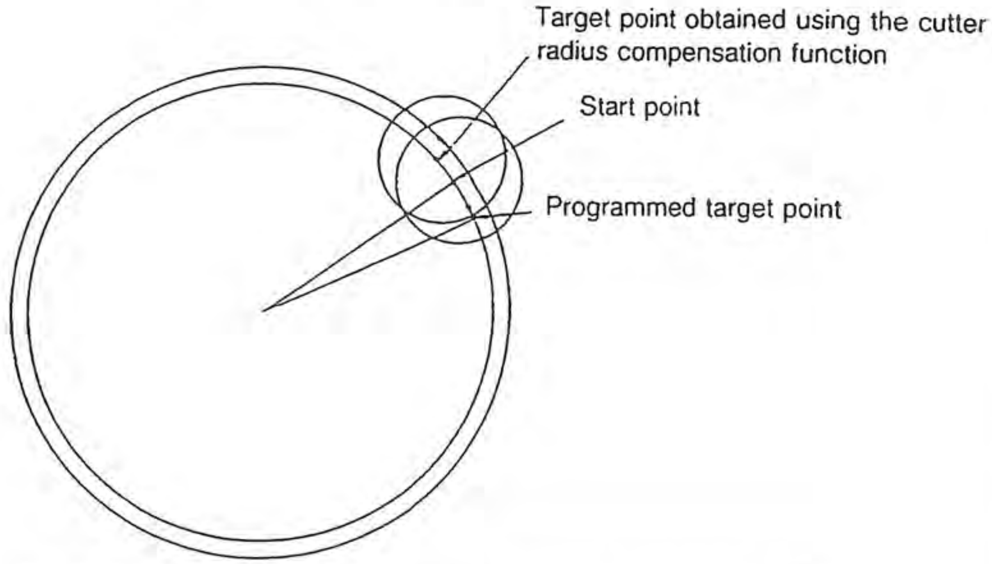


Fig. 15-2 Cutter Radius Compensation for Contour Generation (Face)

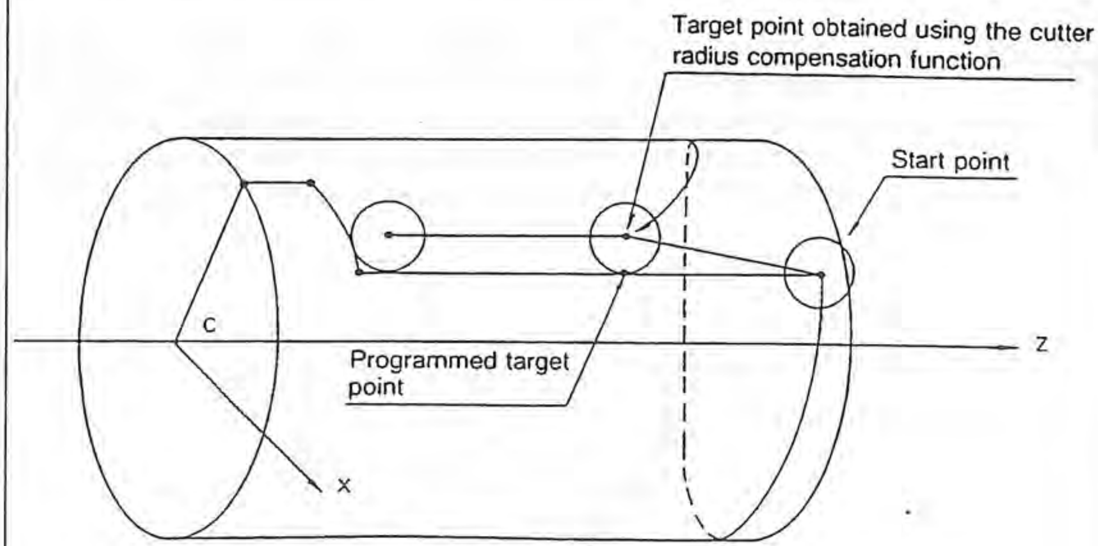


Fig. 15-3 Cutter Radius Compensation for Contour Generation (Side)

In the G00 and G01 modes, rotating direction follows the designated command (M15, M16). In the G101, G102, and G103 modes, rotating direction 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.

CAUTION : (3) 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 Fig. 15-4.

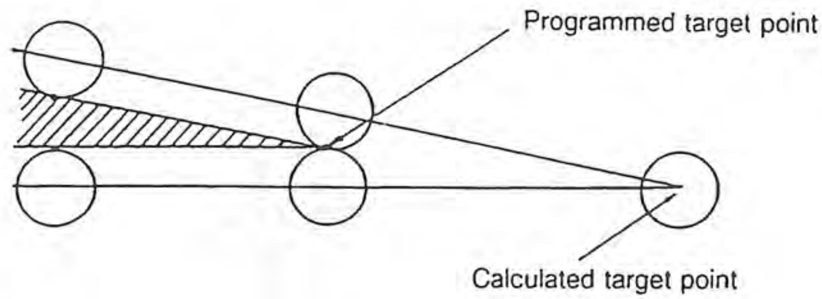


Fig. 15-4

To avoid such a problem, it is necessary to change the program. See Fig. 15-5.

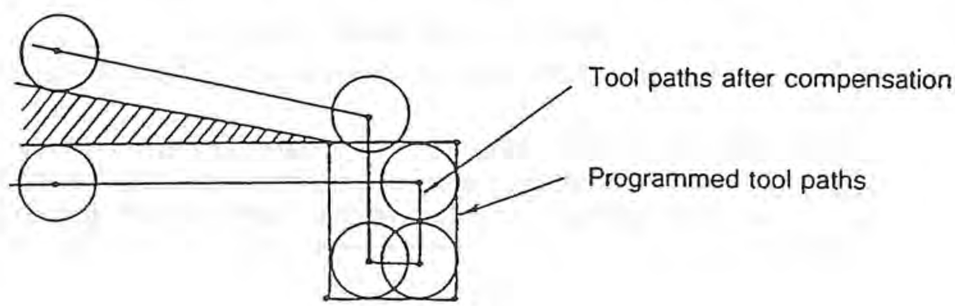
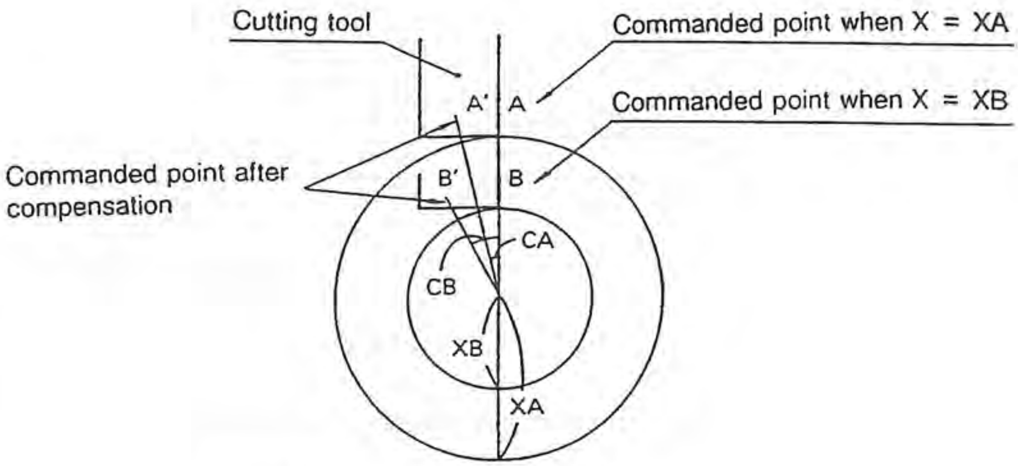


Fig. 15-5

CAUTION : (4) 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 Fig. 15-6.



Machining (Side) Viewed from Front

Fig. 15-6 Cutter Radius Compensation for Contour Generation

The commanded points change as A and B according to the X value (XA, XB) even when the Z-C commands are the same. As illustrated above, the actual compensation values will vary at points A and B even when the same diameter tool is used.

SECTION 16 S, T, AND M FUNCTIONS

The S, T, and M functions specify the necessary machine operations other than axis movement commands.

S : Spindle speed

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)

By specifying number following address S, spindle speed can be specified.

Program Format: S __

(1) S command range: 0 to 65535

(2) 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.

(3) 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.

(4) 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.

NOTICE : (1) For the machine equipped with the transmission gears, the required gear range should be selected by a corresponding M code.

(2) Spindle rotation (forward, reverse) and stop are specified by M codes.

2. SB Code Function

M-tool spindle speed is specified using address SB.

Program Format: SB = __

If an address consisting of two or more characters is used, an equal symbol must be entered before a numeric value.

(1) SB command range: 0 to 65535

(2) M-tool spindle rotation (forward, reverse) and stop are specified by M codes.

(3) 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.

(4) 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.

NOTICE : (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)

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.

Program format: T○○△△□□

○○: Tool offset number

△△: Tool number

□□: Tool nose radius compensation number

(1) Programmable T command range is indicated below.

Tool number: 00 to 99
(the number of turret stations)

<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.)

<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.)

<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.)

(2) 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.

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

4. M Functions (Auxiliary Functions)

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.

(1) Programmable M code range is from 0 to 511.

(2) Concerning the M codes shown below, they are processed as the special function M codes. For the M codes not listed below, refer to APPENDIX 3. "List of M Codes".

- M00 (program stop)

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

- 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.

- 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 the reset processing is executed. The program is rewound to the start of the program. (In the case of a schedule program, execution of M02 or M30 in the main program does not reset the NC.)

- 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).

- 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).

- M15, M16 (C-axis positioning direction)

  These M codes control 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).

- M19 (spindle orientation)

  This controls spindle orientation.

- M20, M21 (tailstock barrier ON, OFF)

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

- M22, M23 (chamfering ON, OFF for thread cutting)

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

- M24, M25 (chuck barrier ON, OFF)

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

- 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).

- 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)).

- 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).

- M48, M49 (spindle speed override ignore)

  In the state the spindle speed override ignore function is valid, the spindle speed override rate is fixed at 100% disregarding 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 code >

  Spindle speed override ignore M49

  Spindle speed override ignore cancel M48

- M55, M56 (tailstock spindle retract, advance)

  These M codes specify tailstock retract/advance operation.

- M60, M61 (fixed surface speed arrival ignore OFF, ON)

  These M codes are used to specify whether or not the program, controlled in the constant surface speed, is executed continuously without waiting for the arrival at the specified surface speed; advances to the next block without waiting for the arrival at the specified surface speed (M61), advances to the next block only after the arrival at the specified surface speed (M60).

- M63 (spindle rotation answer signal ignore)

  The M codes related with 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.

- 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.

- M83, M84 (chuck clamp, unclamp)

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

- M85 (not returning to the start point after the completion of LAP roughing cycle)

  In LAP4, 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, but the next block is executed continuously.

- 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; turret clockwise rotation ON (M86), turret clockwise rotation OFF (M87).

- 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 back to the spindle control mode. Note that M110 must be specified in a block without other commands.

- M124, M125 (STM time-over check ON, OFF)

  These M codes are used to determine whether or not the alarm is generated if the counted STM execution cycle time exceeds the parameter-set time; alarm is generated (M124), alarm is not generated (M125).

- M136 (shape definition for compound fixed cycle)

  This M code is used to specify the shape for compound fixed cycle, 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.

- 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, 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 for the multiple-process specification models.

- M141, M146, M147 (C-axis clamp used/not-used selection, C-axis unclamp, C-axis clamp)

  For a compound fixed cycle, carried out in 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 "C-axis clamp is not used" state, thereby reducing cutting time.

  M146 and M147 are used to control C-axis clamp and unclamp; M146 for C-axis clamp and M147 for C-axis unclamp.

- M156, M157 (center work interlock ON, OFF)

  When center work is selected, the operation is impossible 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]

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

- 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, set at 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 the setting of 100%, and in the M160 mode, the setting of the feedrate override dial is valid.

- 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, set at 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.

- 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 by 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.

- M166, M167 (tailstock spindle advance/retract interlock during spindle rotation ON, OFF)

  While the spindle is rotating, it is not allowed to advance or retract the tailstock spindle to ensure safety. However, tailstock spindle operation is permitted even while the spindle is rotating by turning OFF the interlock.

[Supplement]

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

- M184, M185 (chuck open/close interlock ON, OFF)

  While the spindle is rotating, it is not allowed to open or close the chuck to ensure safety. However, chuck open/close operation is permitted even while the spindle is rotating by turning OFF the interlock.

[Supplement]

1. When the power supply is turned off or after the NC is reset, the NC is in the M184 state.
2. The state selected by these M codes is effective only for MDI and automatic operation modes.
3. The state selected by these M codes is effective only when the door is closed.
4. 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.

- M193, M194 (thread cutting phase matching control OFF, ON)

  In the M194 mode, phase offset amount at the thread cutting start point is calculated and compensation is made 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.

- 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 program for thread cutting phase matching, amount of manual axis move which is made for phase matching is stored. M196 must be specified in a block without other commands.

  After the completion of manual axis move for phase matching, the M196 mode must be canceled by specifying M195 in a block without other commands.

- M197 (clearing thread cutting phase matching amount)

  This M code is used to clear the amount which is stored as the manual axis move amount for phase matching.

- 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; specified cutting amount (M213), equally-divided cutting amount (M214).

- 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   G94   X△△△  Z△△△  C△△△  M15 (M16)  SB=△△△△
X004   G01   X(Z)△△△   F△            M147 M13 (M14)

             Program for cutting with M-tools

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

(1) M110 must be programmed in a block without other commands.

(2) It is recommended to limit the rotating direction of C-axis to either of two directions M15 or M16 for better positioning accuracy.

(3) M110 and M147 cannot be reset or canceled even when the control system is reset. To cancel them, specify M109 and M146, respectively.

(4) If commands related with M-tool are specified while C-axis is not engaged, an alarm occurs.

An alarm does not occur if M-tool spindle interlock (optional) is designated.

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

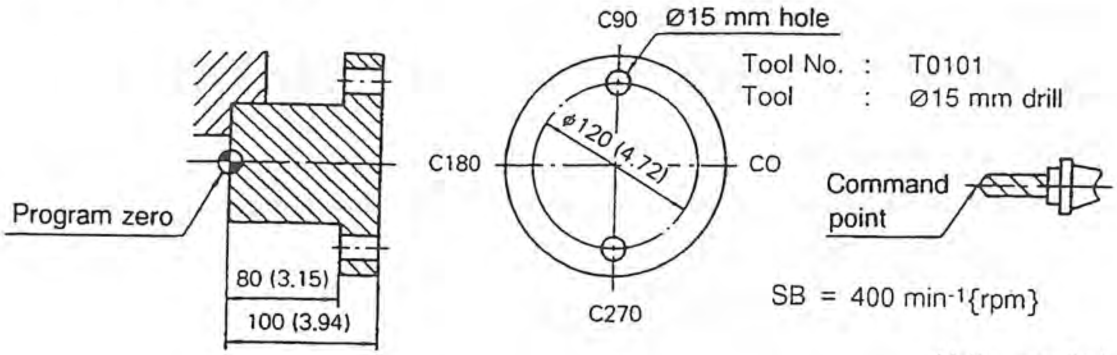
The following codes are necessary for programming C-axis movements.

<table>
<tr><th>Codes</th><th colspan="2">Contents</th></tr>
<tr><td>M110</td><td colspan="2">Used to designate the spindle to be controlled in the C-axis control mode.
When programming C-axis commands, first specify M110 in the block without other commands.</td></tr>
<tr><td>M109</td><td colspan="2">Used for switchover from the C-axis control mode to the spindle control mode.</td></tr>
<tr><td>M147</td><td colspan="2">Used to clamp C-axis.</td></tr>
<tr><td>M146</td><td colspan="2">Used to unclamp C-axis.
The control system automatically selects the M146 mode when the power is turned on.
Program M146 before starting C-axis rotation.</td></tr>
<tr><td>M141</td><td colspan="2">C-axis clamp ineffective
(compound fixed cycle mode)</td></tr>
<tr><td>M15</td><td>Used to rotate the C-axis in the positive direction.</td><td rowspan="2">

Facing Chuck</td></tr>
<tr><td>M16</td><td>Used to rotate the C-axis in the negative direction.</td></tr>
<tr><td>QA =</td><td colspan="2">Used to specify the number of C-axis revolutions.
For example, QA=5 rotates C-axis five times.</td></tr>
</table>

* When the NC is reset, it is placed in the M15 mode.

[Example of Program]



Unit: mm (in.)

When drilling two 15 mm dia. holes, create a program as indicated below:

Continued from turnig operation program					
N099	G00	X1000	Z1000		M05
N100					M01
N101					M110
N102					M15
N103	G94	X120	Z102	C90	T0101 SB=400
N104	G01		Z75	F40	M13 M147
N105	G00		Z102		
N106				C270	M146
N107	G01		Z75		M147
N108	G00		Z102		
N109	G95	X1000	Z1000		M12 M146
N110					M109
N111					M02

Calculate the feedrate (mm/min) for drilling with the equation below:

Feedrate (mm/min) =Tool speed (rpm) Feedrate (mm/rev)

Therefore, when tool speed is 400 min^{-1}{rpm} and feedrate is 0.1 mm/rev, feedrate (mm/min) is calculated as:

F = 400 × 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

Provided the end mill with four blades (flutes) be used at 300 min^{-1}{rpm} and a feedrate of 0.05 mm/blade, the feedrate (mm/min) is

F = 300 × 0.05 × 4 = 60 mm/min

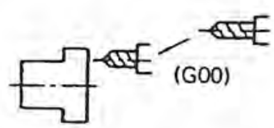
Explanation of Example Program

N101 : Designates the spindle as C-axis.

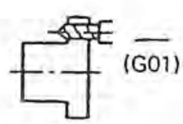
N102 : Indexes C-axis in the positive direction.

N103 : 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 : The drill starts rotation at 400 min^{-1}{rpm} in the leftward direction. After the spindle is clamped, 15 mm dia. hole is drilled at a feedrate of 40 mm/min.



N105 : The drill returns to the commanded point at the rapid feedrate.

N106 : The spindle is indexed at the 270 position after it is unclamped.

N107 : The second hole is drilled after the spindle is clamped.

N109 : The M-tool stops and the turret returns to the turret index position.

SECTION 17 FIXED CYCLES

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

1. Fixed Thread Cutting Cycles

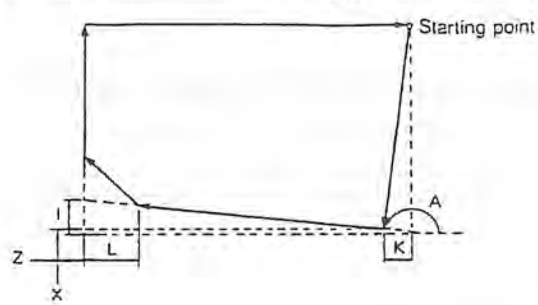
1-1. Fixed Thread Cutting Cycle: Longitudinal (G31, G33)

Format:

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

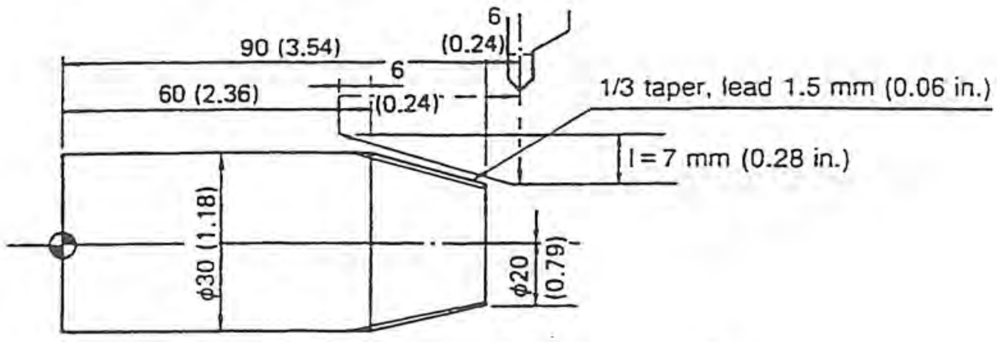
G33	Xoooo.ooo	Zoooo.ooo	(Ioooo.ooo / Aooo.ooo	Eo.ooo)	Fo.ooo	(K, L, J)	Cooo.ooo
	Xoooo.ooo	Cooo.ooo					
	Xoooo.ooo	Cooo.ooo					
	:						

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 I or A word.)
E : Z-axis shift amount of thread cutting starting point
(When no K word is provided, the control assumes K=0.)
L : Chamfering distance
(When no L word is provided, the control assumes L=one lead at thread cutting starting)
L word is effective in the thread chamfering ON (M23) mode.
J : Number of threads within a distance specified by F word
(When no J word is provided, the control assumes J=1.)
C : Phase difference for multi-thread thread cutting (C=0 if omitted.)



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

Example of Program: Constant Lead Taper Thread



N001	G00	X40	Z96		
N002	G33	X17	Z54	I7	F1.5
N003		X16.5			
N004		X16.2			
N005		X16.5			

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

N001 : Positioning to the thread cutting starting point, X = 40 mm (in dia.) and Z = 96 mm, at a rapid feedrate.

N002 : With an I word in G33 block taper thread cutting cycle indicated by 1 through 4 is performed at above drawing.

X, Z and F words can be determined in the same manner as cutting a straight thread.

The value I in this example can be calculated in the following equation:

$$[(96-54) \times 1/3]/2 = 7$$

Note that I word is provided in terms of a difference in radius.

N003 : Program X dimension (in diameter) for each succeeding thread cutting pass.

:

N005

NOTICE : (1) The sign of an I word determines increasing or decreasing taper ("+" for increasing taper and "–" for decreasing taper). Plus sign (+) may be omitted.

(2) Difference of radius between starting point and end point is expressed by an I word.

1-2. Fixed Thread Cutting Cycle: End Face (G32)

Format:

$$[\,G32\ X\_\ Z\_\ \begin{Bmatrix} K\_ \\ A\_ \end{Bmatrix}\ (E\_)\ (I\_)\ (L\_)\ (J\_)\ F\_\ (C\_)\,]$$

```
G32   Xoooo.ooo  Zoooo.ooo  ( Koooo.ooo  Eo.ooo  (k, l, j) )  Fo.ooo   Cooo.ooo
                              Aooo.ooo
                 Zooo.ooo     Cooo.ooo
                 Zooo.ooo     Cooo.ooo
                     :
```

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 I or A word.)

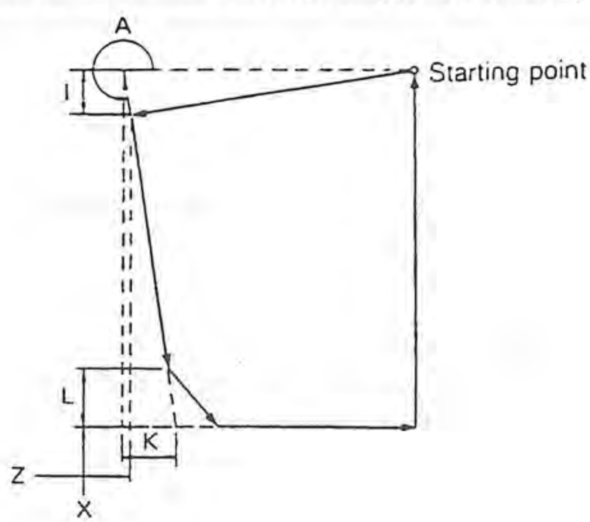
E : Lead variation per lead in cutting variable lead thread
(When no E word is provided, the control assumes E=0.)

I : X-axis shift amount of thread cutting starting point
(When no I word is provided, the control assumes I=0.)

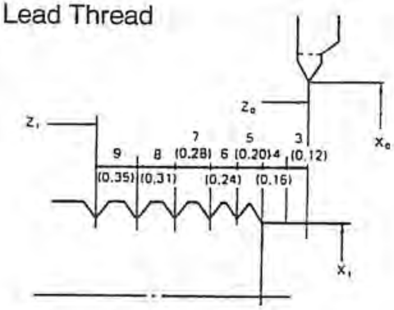
L : Chamfering distance
(When no L word is provided, the control assumes L=one lead at thread cutting starting)

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

C : Phase difference for multi-thread thread cutting (C=0 if omitted.)



Example of Program: Variable Lead Thread



N001	G00	X_0 Z_0
N002	G33	X_1 Z_1 E_1 F2.5
N003		X_2
N004		X_3
:		

N001 : Positioning to thread cutting starting point X_0, Y_0 at a rapid feedrate.

N002 : With an E word in G33 block, variable lead thread cutting cycle is performed along paths 1 through 4.

X and Z words can be determined in the same manner as 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, specify as E_1.

F word specifies the first lead employed in starting the thread cutting cycle.
F2.5 in this example.

N003 : Program Z dimension for each succeeding thread cutting pass.

N004

:

NOTICE : (1) The sign of an E word expresses increasing or decreasing lead.

Increasing lead .... E+
Decreasing lead ...... E–

"+" sign may be omitted.

(2) When determining F word value, use the following equation:

$D = n \times [F_0 \times (n \times E)/2]$

where,

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

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

$42 = 7 \times [F_0 + (7 \times 1)/2]$
$F_0 = 2.5$

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

Format:

$$\left[\begin{matrix}G34\\G35\end{matrix}\right.\ X\_\ Z\_\ (E\_)\ F\_\ (C\_)\ (J\_)\ ]$$

G34 . . . . . . increasing lead

G35 . . . . . . decreasing lead

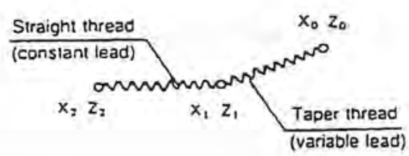
By the commands above, thread cutting cycle with a lead F is carried out from the current tool position to the commanded coordinate point (X, Z).

This programming is conveniently used for special thread cutting such as combined thread, straight and taper thread, or variable lead thread and taper straight thread.

A J word is used to specify the number of threads within the specified lead F. When no J word is provided it is assumed as J=1.

A C word is used to specify the phase difference for multi-thread thread cutting (C=0 if omitted.)

Example:



G00 X_0 Z_0
G34 X_1 Z_1 E F
G34 Z_2 E_0

NOTICE : (1) The lead of the straight thread is the one at the start point (Z_0) of variable lead thread. When the straight thread is required to be cut with a lead obtained at point Z_1, specify an F word again.

(2) To specify the thread lead along the axis parallel to 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 Z-axis, specify M26.

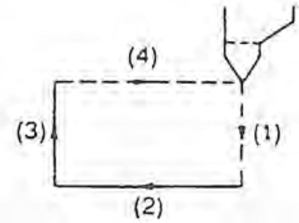
3. Precautions for Programming Thread Cutting Cycles

Observe the following points when programming thread cutting cycles:

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

(2) Motion of Thread Cutting Tool

In thread cutting cycle called for by G31, G32 and G33, tool paths (1) and (4) are executed at a rapid feedrate, (2) at the feedrate specified by an F word, and (3) at the rate determined by parameter setting.



(3) Lead of Taper Thread

Lead of a taper thread is parallel to Z-axis in G31 and G33 mode, and to X-axis in G32 mode.

In G34 and G35 mode, M code is used to designate direction of thread lead:

M26 . . . . . . Cancel of M27, parallel to Z-axis

M27 . . . . . . Parallel to X-axis

If no M code is specified in G34 or G35 mode, the control assumes M26, parallel to Z-axis.

(4) Number of Thread Cutting Passes

Determine the number of thread cutting passes to complete the thread according the workpiece material, thread lead, etc.

(5) Spindle Speed Change During Thread Cutting Cycle

If the spindle speed change is intended while thread cutting cycle, it will shift the starting point of the thread cutting cycle, thus damaging the thread being cut.

Therefore, never change spindle speed during thread cutting cycle.

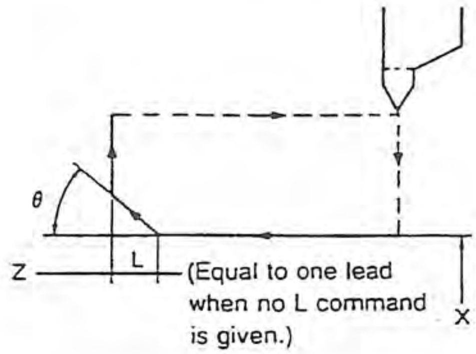
(6) Feedrate Override

The feedrate override dial is inoperative while thread cutting cycle.

(7) Chamfering

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

M22	Chamfering OFF
M23	Chamfering ON



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

Feedrate on the X-axis (mm/min)

$$= \frac{\text{Parameter set value } (\mu)}{10^3} \times \frac{60 \times 10^3 \text{ (msec)}}{12.8 \text{ (msec)}}$$

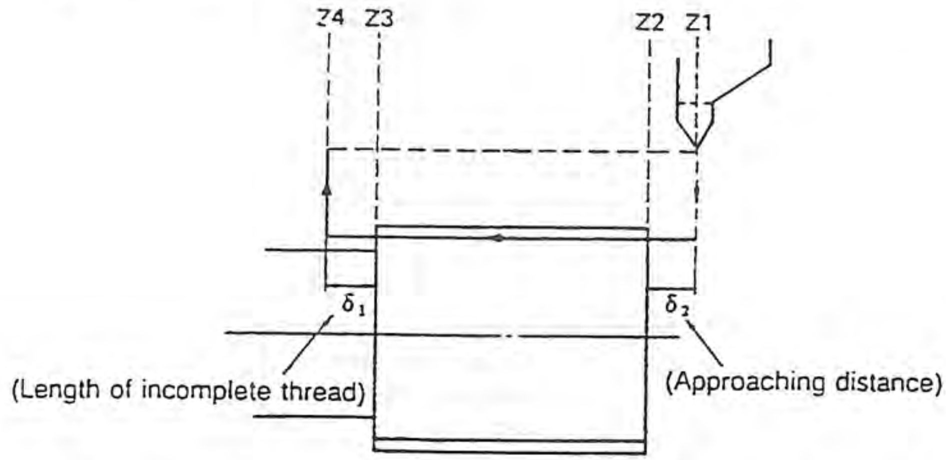
Therefore, the chamfering angle θ is determined by the feedrate in the Z-axis direction (designated in the thread cutting program) and the feedrate in the X-axis.

Example:

N001	G00	X40	Z80	$\boxed{\text{M23}}$
N002	G33	X29	Z50	$\boxed{\text{F1.5}}$
N003		X28.5		
N004		X28.2		
N005		X28.05		
N006	G00	X_1	Z_1	$\boxed{\text{M22}}$

(8) Extra Length in Thread Cutting Program

Since certain length of incomplete thread is usually produced near start and end point of the cut, it is necessary to add proper amount δ_1 and δ_2 to the start and from the end of the thread to be cut for cutting proper shape of thread.



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

$\delta_1 > K \cdot N \cdot P$
$\delta_2 > K \cdot N \cdot P$

where,

N: spindle speed
P: lead
K: machine model dependent constant

Values for constant K for individual models are indicated below:

Model	K [$\times 10^{-3}$]	Model	K [$\times 10^{-3}$]
LB10 II	0.85	LU35	0.85
LB15 II	0.96	LU45	1.28
LB25 II	0.85	LT10	0.85
LB35 II	1.28	LT15	0.85
LB35 II (AW model)	2.56	LT25	1.07
LB45 II	0.96	LCC15	1.07
LB30 II	0.85	LCS10	0.75
LB35 II	1.07	LCS15	0.85
LB55 II	0.96	LCS25	0.64
LU15	1.07	LAW	2.56
LU25	1.07	LAW-F	1.28

Example:

For the LB15-II, with a peripheral speed of 100 m/min, a 10 mm diameter and a thread lead of 1.5 mm, spindle speed and feedrate are calculated as below.

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

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

Since $K = 0.64 \times 10^{-3}$, δ_1 and δ_2 must be greater than 3.05 mm which is calculated as below:

$0.64 \times 10^{-3} \times 4775 = 3.05$ (mm)

(9) Restrictions on Cutting Speed

In thread cutting cycle, following restrictions apply to the relation between spindle speed and thread lead:

Programmable thread lead .... 0.001 to 1000.000 mm

Spindle speed: X-axis .... $N \times P <$ Max. feedrate of X-axis

Z-axis .... $N \times P <$ Max. feedrate of Z-axis

where,

N: spindle speed
P: lead

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

(10) Inch System Thread

When cutting inch threads, metric lead converted from the desired inch lead is used in programming. To cut 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

$25.4/11 \doteqdot 2.309091$

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)

(11) Feed Hold During Thread Cutting Cycle

This function is effective while an axis, X (Z) is moving in G32 (G33) mode. Pressing the SLIDE HOLD pushbutton while thread cutting cycle immediately stops axis movement breaking the thread being cut, thus damaging the workpiece. This function is provided to prevent such trouble.

Activate this function to check dimensions and shape of the threads being cut and also to check the tip point of the thread cutting tool.

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

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

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

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

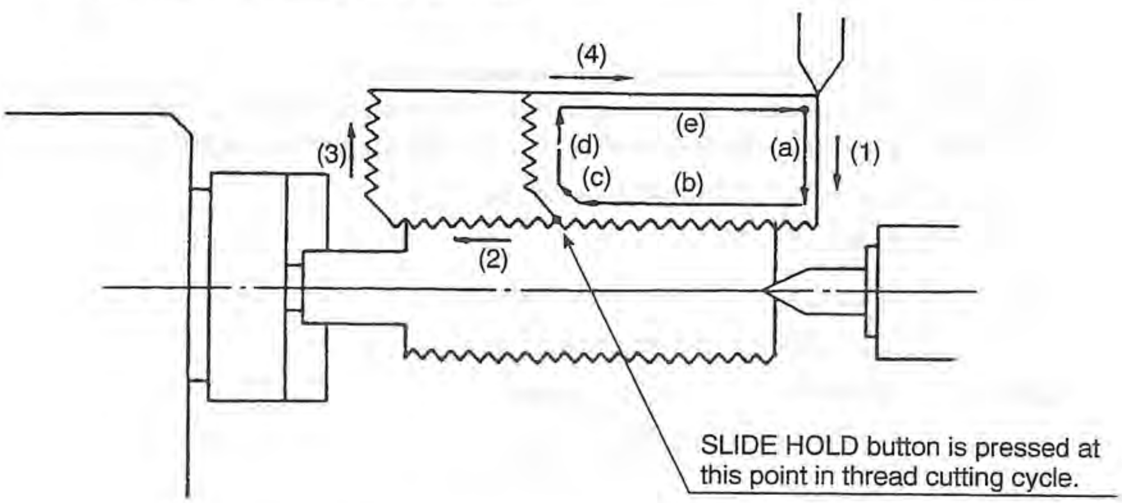
(d) The control is in cycle stop mode waiting for pressing of the CYCLE START button.

When the CYCLE START button is pressed,

(a) 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) 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)

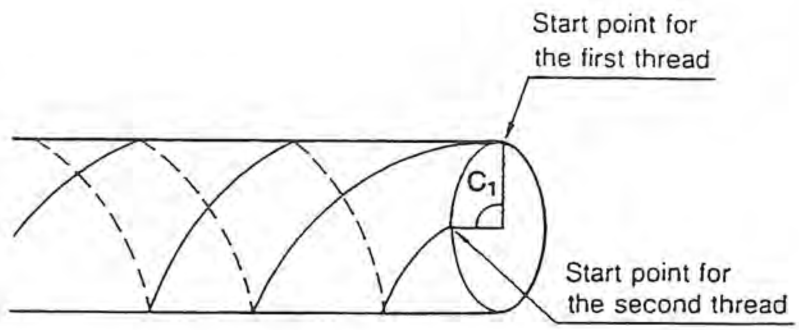
(12) Designation of Phase Difference (Angle) for Multi-thread Thread Cutting

Multi-thread thread can be programmed easily by designating the thread cutting start point.

G33 cycle:

$$\left.\begin{array}{llll} G33 & X_1 & Z_1 & F \\ & X_2 & & \\ & X_3 & & \end{array}\right\} \text{First thread}$$

$$\left.\begin{array}{lllll} G33 & X_1 & Z_1 & F & C_1 \\ & X_2 & C_1 & & \\ & X_3 & C_1 & & \end{array}\right\} \text{Second thread}$$



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

G32 cycle:

$$\begin{array}{lllll} G32 & X_1 & Z_1 & F & C_1 \\ & & Z_2 & & C_1 \\ & & Z_3 & & C_1 \end{array}$$

G34, G35 cycle:

$$\begin{array}{llllll} G34 & X_1 & Z_1 & E_1 & F & C \\ G34 & & Z_1 & E_2 & & \end{array}$$

← The C command is ignored in the sequence other than the first sequence.

NOTICE : (1) Range of 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 succeeding 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 following block, such a C command value is ignored.

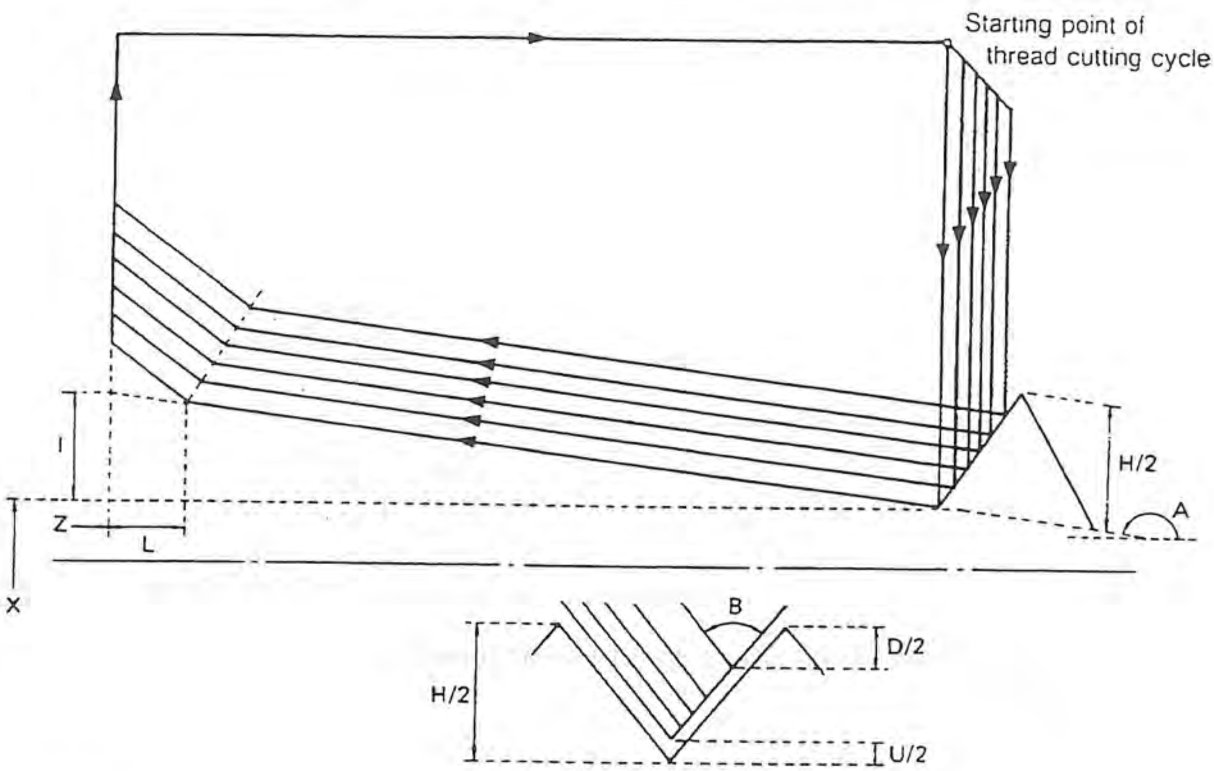
(4) In the G34 and G35 cycles, a C command value can be designated only in the first sequence block; a C command value designated in the second and following sequence block is not acceptable.

For multi-thread thread cutting operation, refer to SECTION 17, 4."Thread Cutting Compound Cycle (G71/G72)".

4. Thread Cutting Compound Cycle (G71/G72)

4-1. Longitudinal Thread Cutting Cycle (G71)

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



Format:

[G71 X_ Z_ $\left\{ \begin{matrix} A\_ \\ I\_ \end{matrix} \right\}$ B_ D_ U_ H_ L_ E_ F_ J_ M_ Q_]

Description of each word:

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 in radius)

For taper thread, use either A or I word.

B : Infeed angle
($0° \leqq B < 180°$; 0° if no B command is provided. Normally it is equal to the cutter tip point angle.)

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

U : Finishing allowance
(Expressed in diameter; no finishing cycle is performed if a U word is not provided.)

H : Thread height
(Expressed in diameter)

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

E : Lead variation rate per lead for variable lead thread

F : Same as in the G33 mode

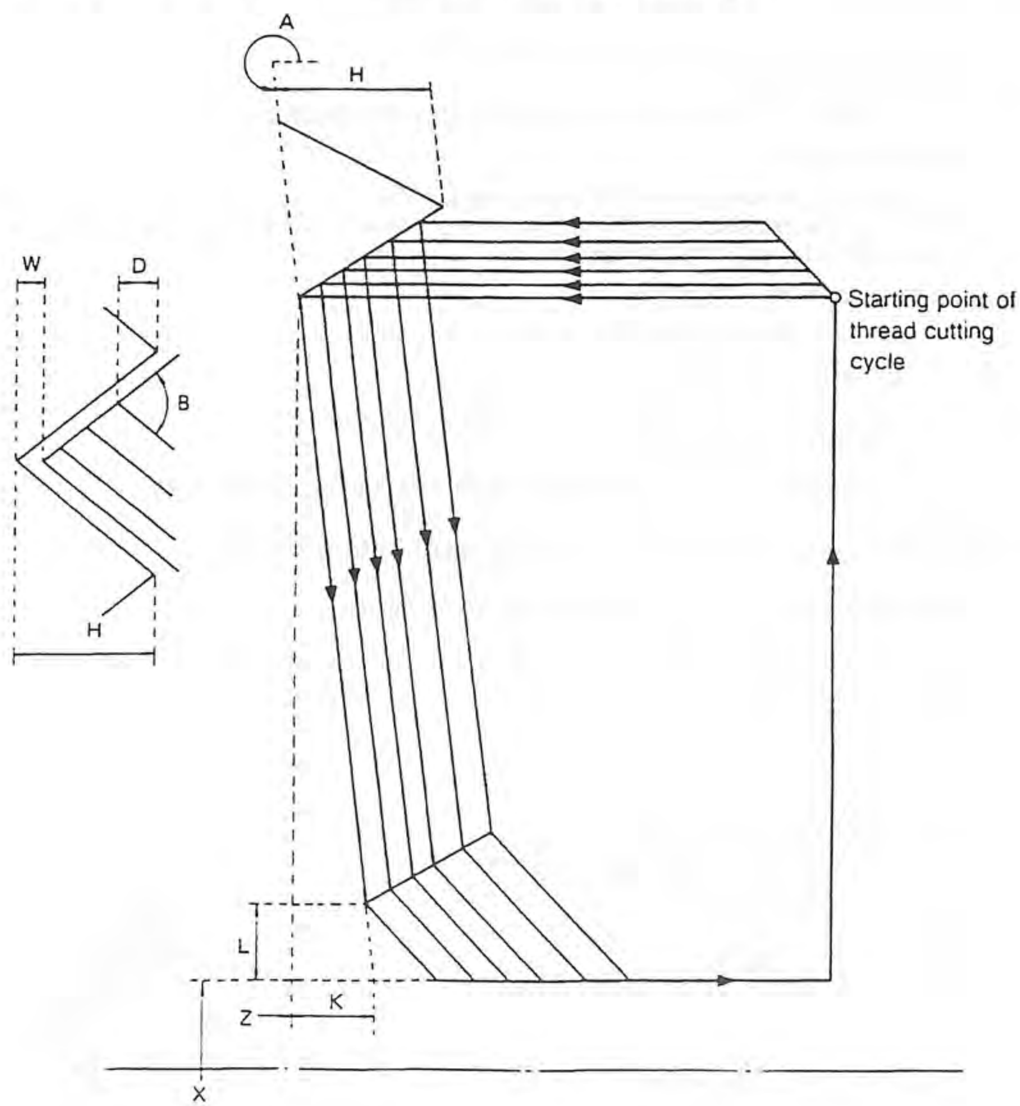
J : Same as in the G33 mode

M : Used to select thread cutting pattern and mode of infeed.
(For details, refer to 4-3.)

Q : The number of threads for multi-thread thread cutting (refer to 4-4.)

4-2. Transverse Thread Cutting Compound Fixed Cycle (G72)

In this fixed cycle, thread cutting cycle as shown below is performed.



Format:

[G72 X_ Z_ $\begin{Bmatrix} A\_ \\ K\_ \end{Bmatrix}$ B_ D_ W_ H_ L_ E_ F_ J_ M_ Q_]

Description of each word:

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 A or K word.

B : Infeed angle
(0° ≦ B < 180°; 0° if no B command is provided.)

D : Depth of cut in the first thread cutting cycle

W : Finishing allowance
(No finishing cycle is performed if a W word is not provided.)

H : Thread height

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

E : Same as in the G32 mode

F : Same as in the G32 mode

J : Same as in the G32 mode

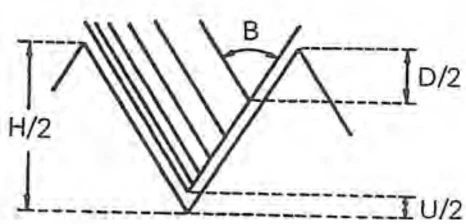
M : Used to select thread cutting pattern and mode of infeed.
(For details, refer to 4-3.)

Q : The number of threads for multi-thread thread cutting (refer to 4-4.)

4-3. M Code Specifying Thread Cutting Mode and Infeed Pattern

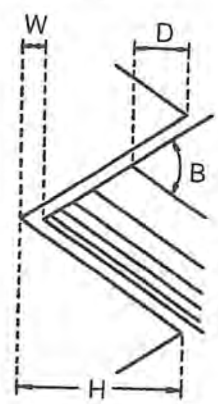
(1) The data necessary for thread cutting cycle are as follows:

Thread Cutting in Longitudinal Direction



B : Tool angle
H : Thread height
D : Depth of cut in the 1st Cycle
U (W) : Stock removal

Thread Cutting in Transverse Direction (on End Face)



(2) M Code Specifying Thread Cutting Mode and Infeed Pattern

(a) M code specifying the thread cutting mode

M32: Straight infeed along thread face (on left face)
M33: Zigzag infeed
M34: Straight infeed along thread face (on right face)

When neither of these M codes is specified, the control automatically selects the M32 mode.

[Supplement] If tool angle B is 0°, the cutting tool is fed straight independent of the designated cutting mode.

(b) M code specifying the infeed pattern

M73: Infeed pattern 1

Infeed is made by D (in diameter) in each thread cutting cycle up to the point D mm away from "H - U (W)" position. After that point is reached, infeed amount is changed as D/2, D/4, D/8 and D/8, leaving stock removal U (W) if specified. And in the finishing cycle, infeed is made as much as the specified amount U (W).

When no U (W)word is specified, finishing cycle is not performed.

M74: Infeed pattern 2

Infeed is made by D (in diameter) in each thread cutting cycle until "H - U (W)" position is reached. After that, finishing cycle is carried out with infeed amount of U (W). If no U (W) word is specified, 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. Whether pattern 3 or pattern 4 is selected is determined by the setting for optional parameter (OTHER FUNCTION 1) Infeed pattern in the M75 mode.

Infeed pattern 3
(M32, M34 is designated)

$\underline{D^2 \geqq \{H^2 - (H - U\ (W))^2\}}$

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 "$\sqrt{2}$ D", and the "n"th path at cutting point "$\sqrt{n}$ D" until the path reaches the cutting point of "H - U (W)". 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.

$D^2 < \{H^2 - (H - U (W))^2\}$

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_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 the number of paths is minimized. 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.

(M33 is designated)

$D^2 \geqq \{H^2 - (H - U (W))^2\}$

Thread cutting cycle is repeated with the cutting point at each even numbered thread cutting path being "$\sqrt{n}$ D" until the cutting point of "H - U (W)" is reached. In each odd numbered tool paths, the cutting point is calculated as $1/2\ (\sqrt{n+1} + \sqrt{n-1}\ )$. 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.

$D^2 < \{H^2 - (H - U (W))^2\}$

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 the number of paths is minimized. 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.

Infeed pattern 4
(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 d_n and S_n 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 the number of paths is minimized. 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.

(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 numbored 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 the number of paths is minimized. 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.

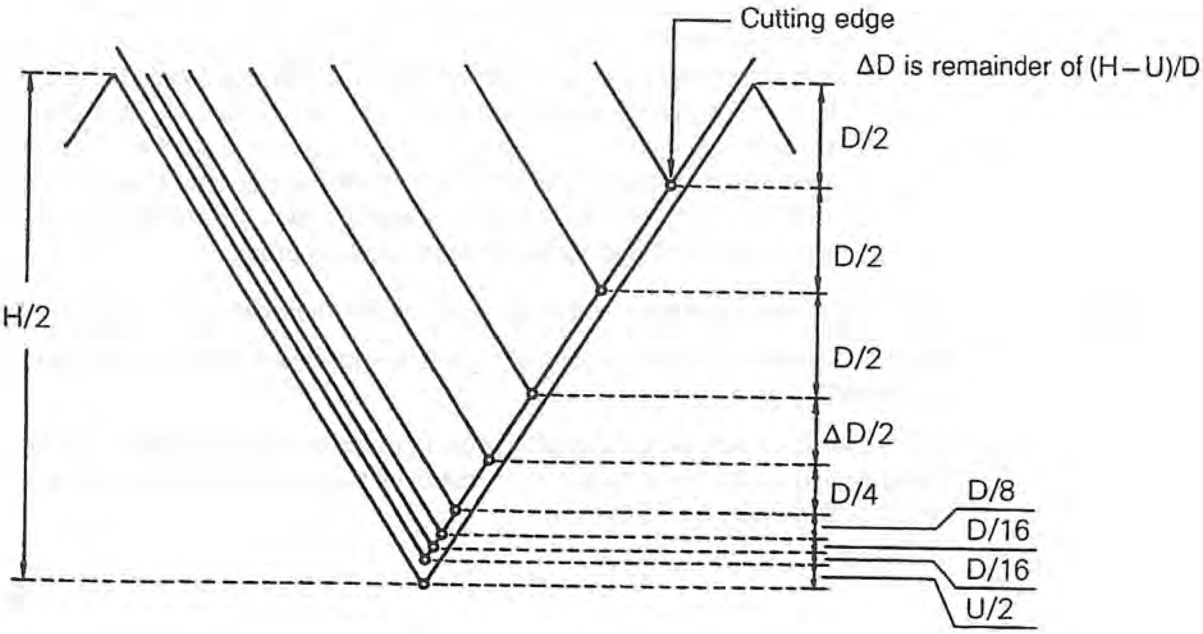
[Supplement] Since X commands are specified in diameter, actual infeed amount is "D/2".

When no infeed pattern designating M code is programmed, the control automatically selects M73.

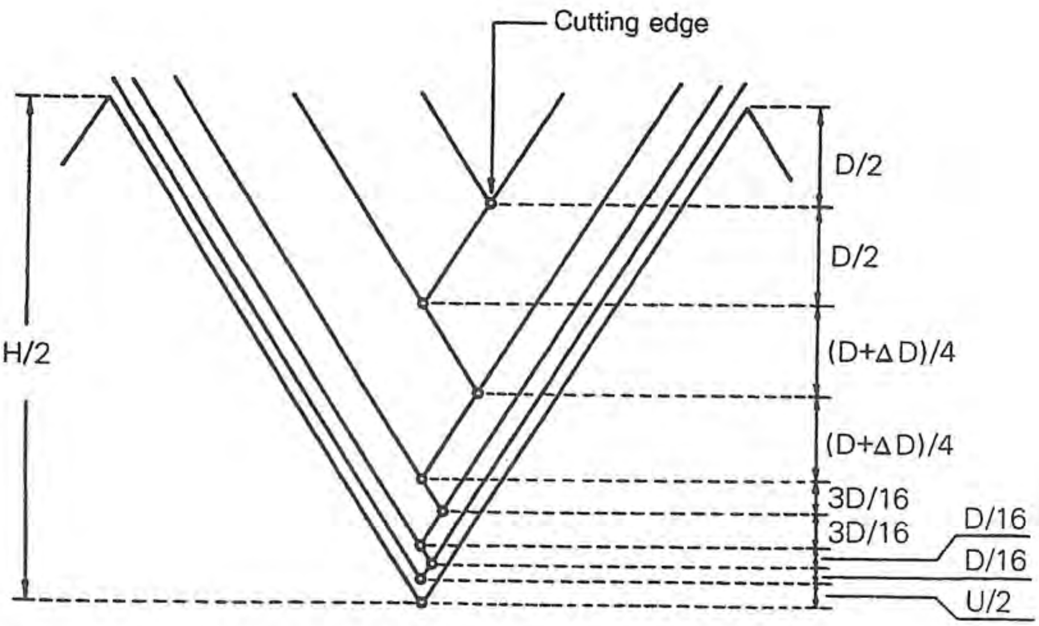
By combining the M codes designating cutting mode and infeed pattern, ten types of thread cutting cycles are available for longitudinal thread cutting cycle and transverse thread cutting cycle, respectively.

Longitudinal Thread Cutting Cycles

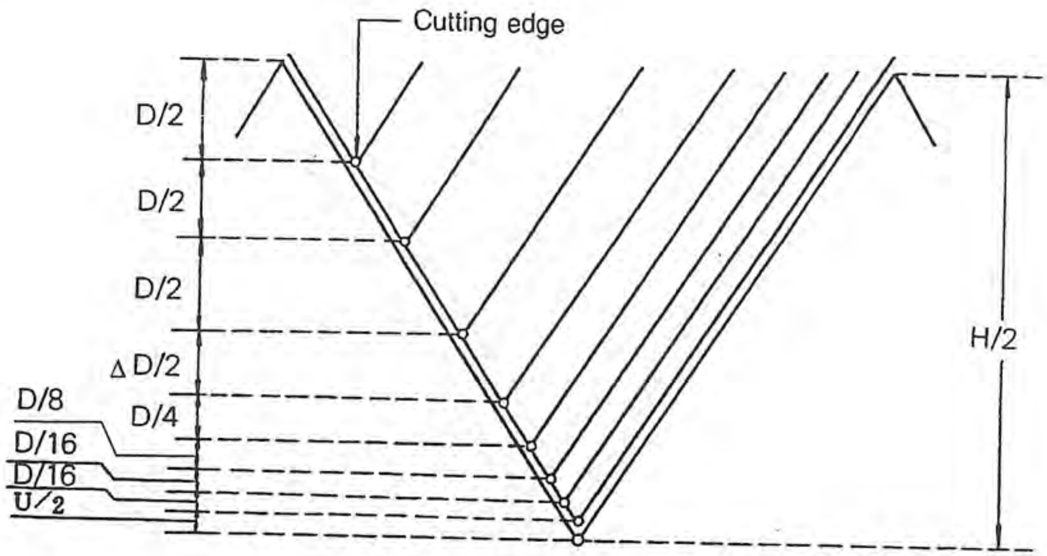
M32 + M73 Mode



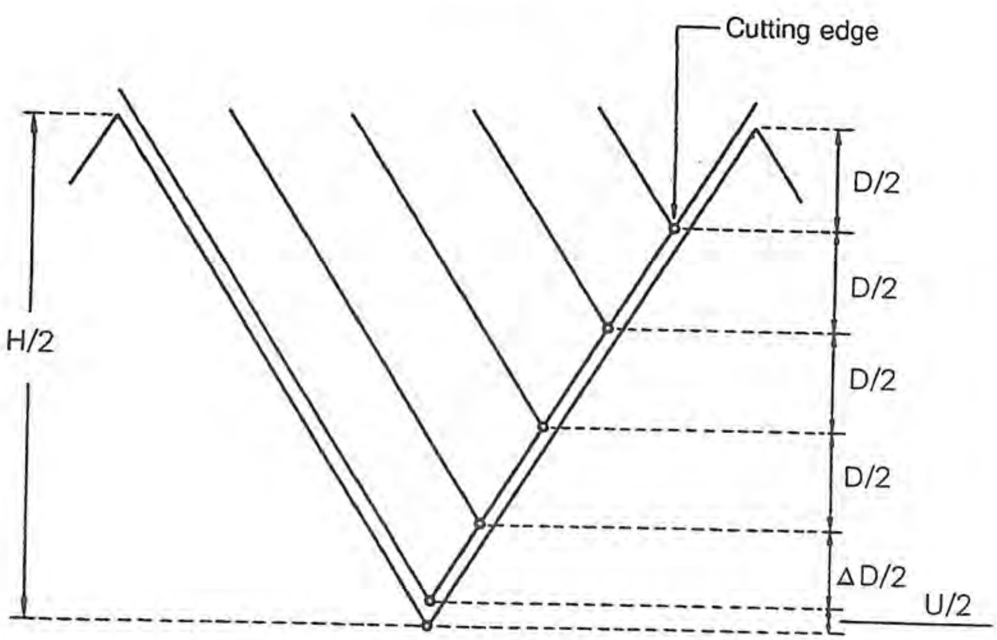
M33 + M73 Mode



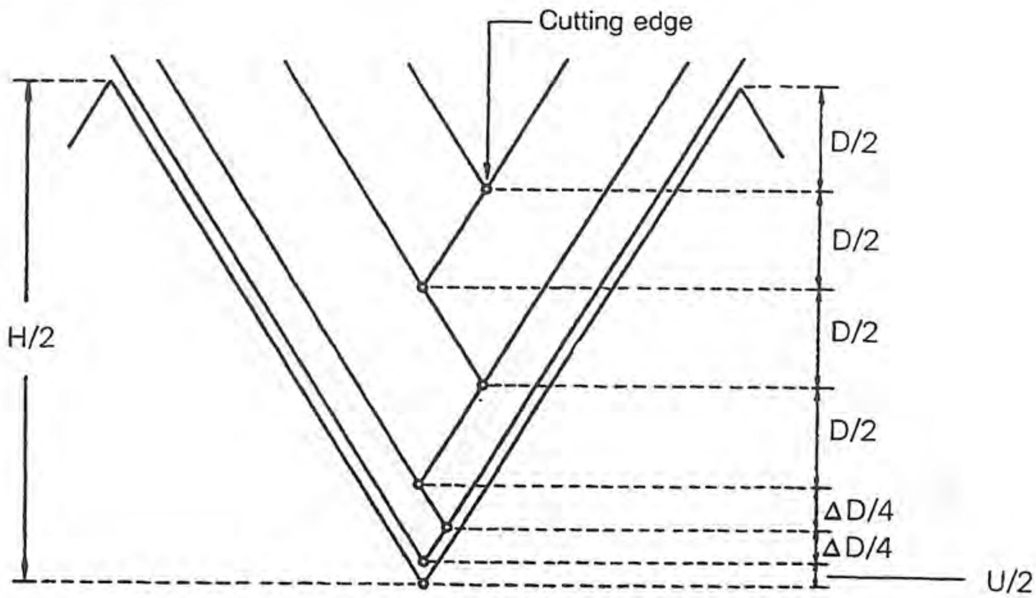
M34 + M73 ModeΔ



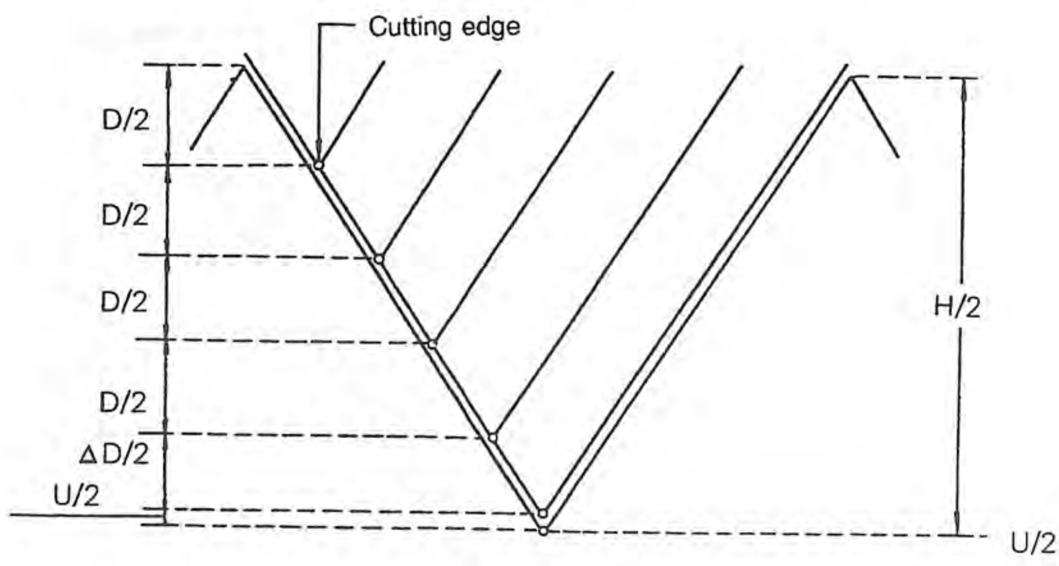
M32 + M74 Mode



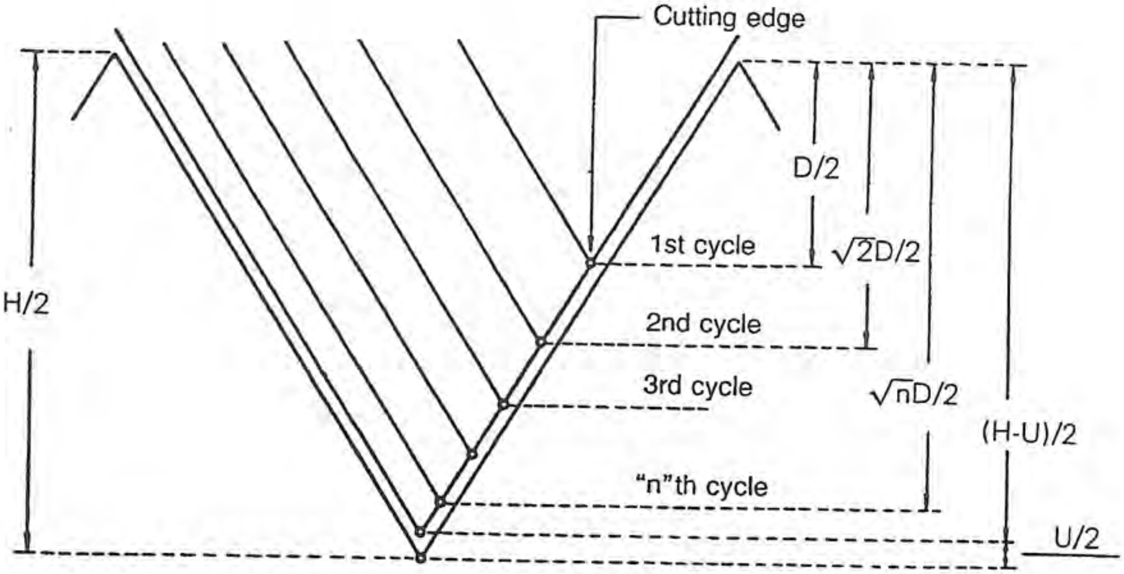
M33 + M74 Mode



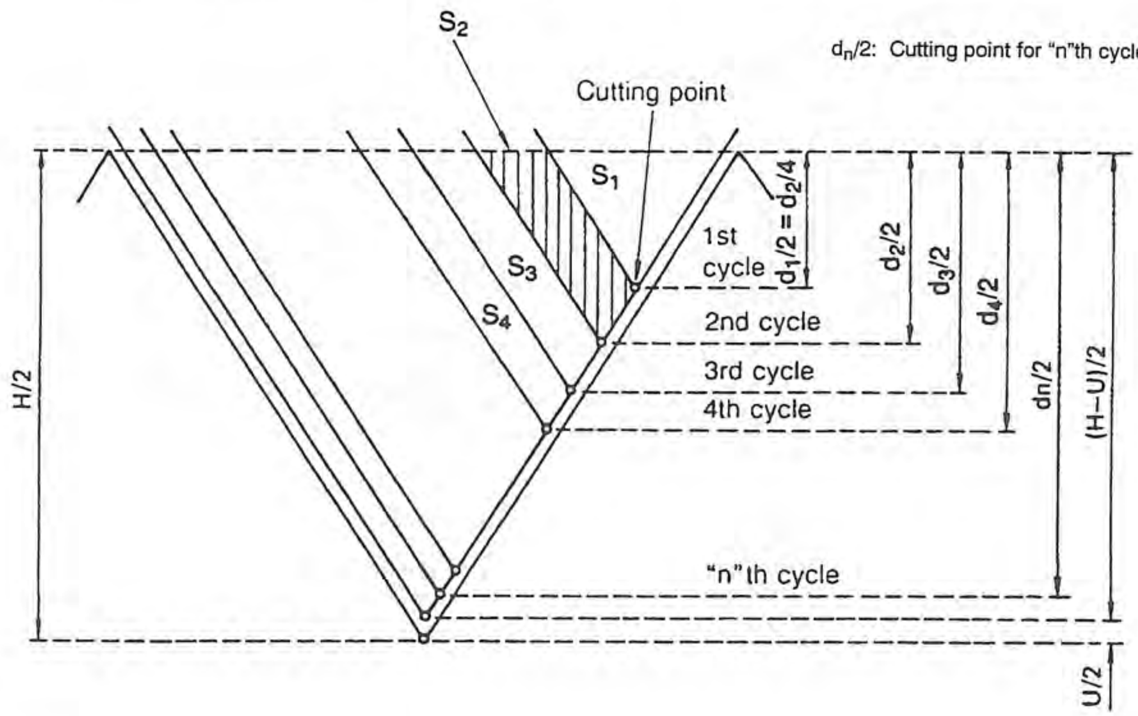
M34 + M74 Mode



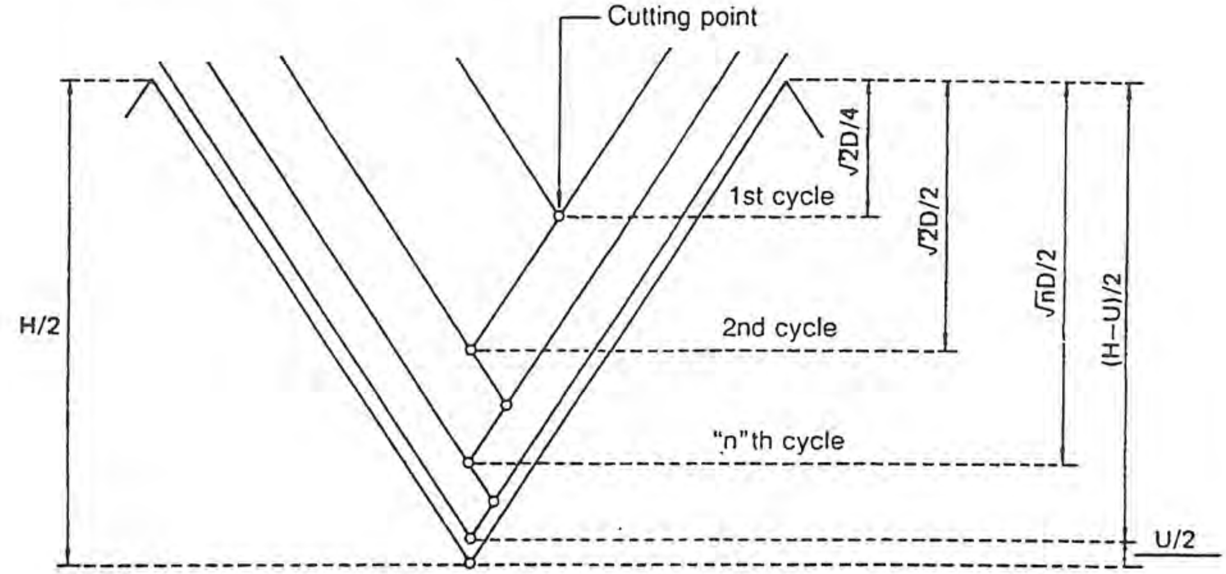
M32 + M75 Mode (infeed pattern 3 $D^2 \geqq \{H^2 - (H - U\ (W))^2\}$)



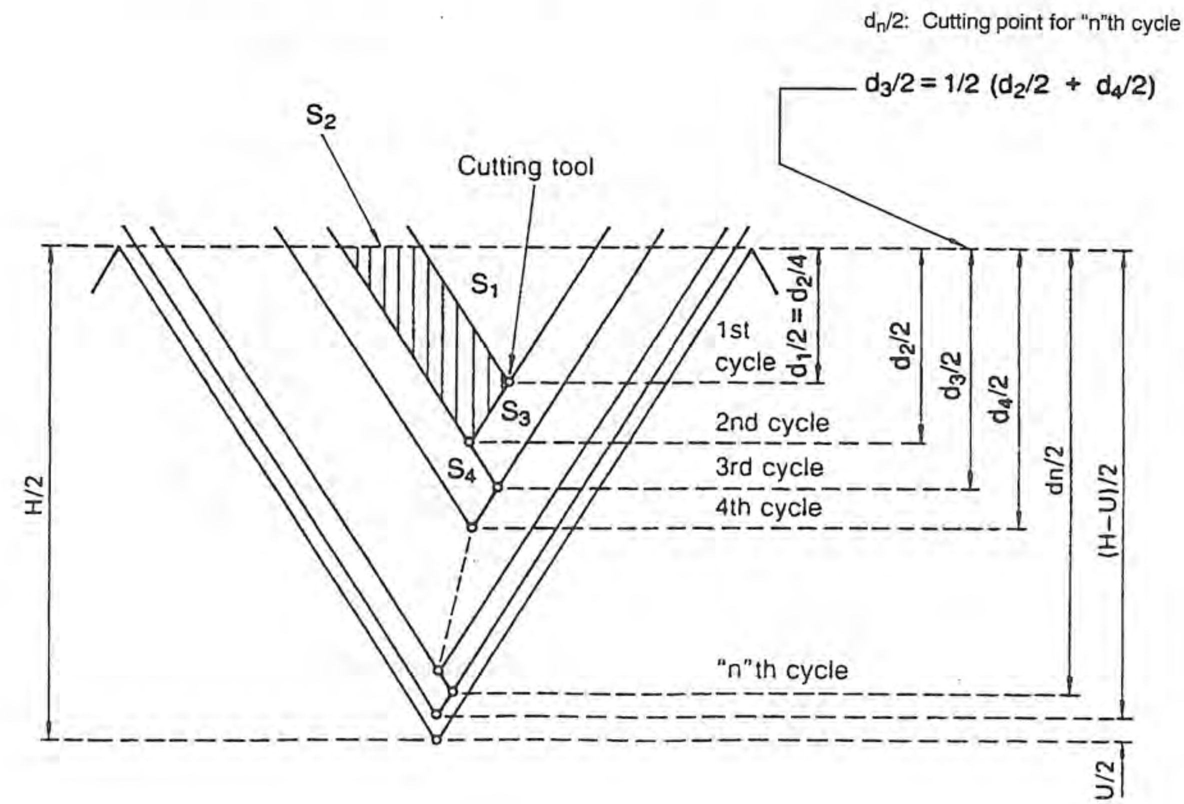
M32 + M75 Mode (infeed pattern 3 $D^2 < \{H^2 - (H - U\ (W))^2\}$ or infeed pattern 4)



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

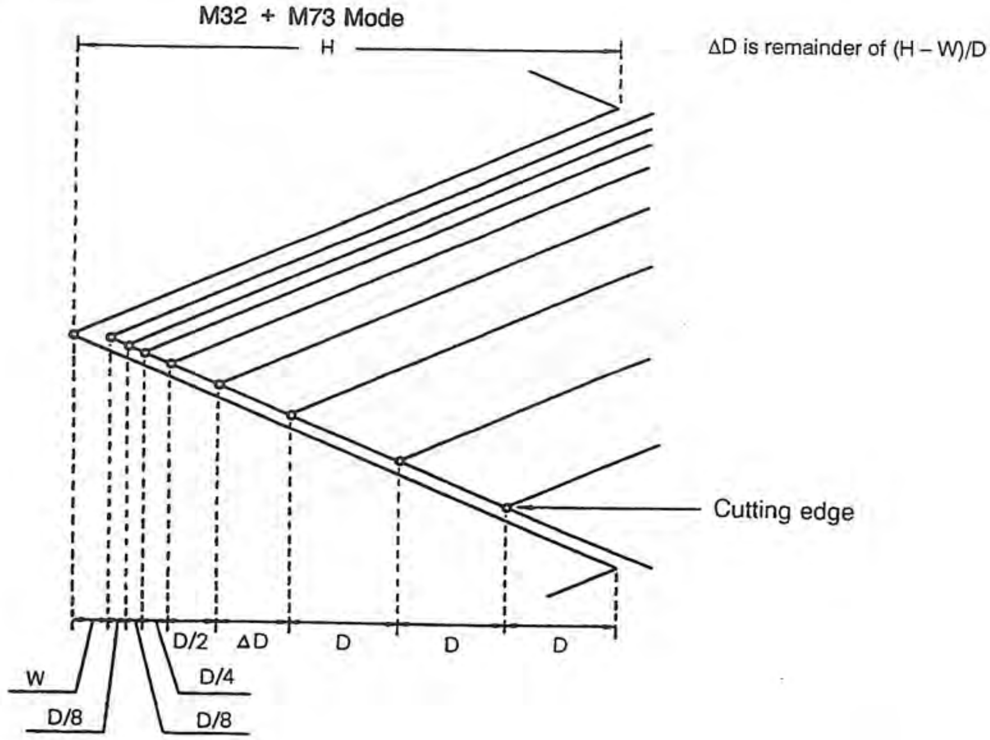


M33 + M75 Mode (infeed pattern 3 $D^2 < \{H^2 - (H - U\ (W))^2\}$ or infeed pattern 4)

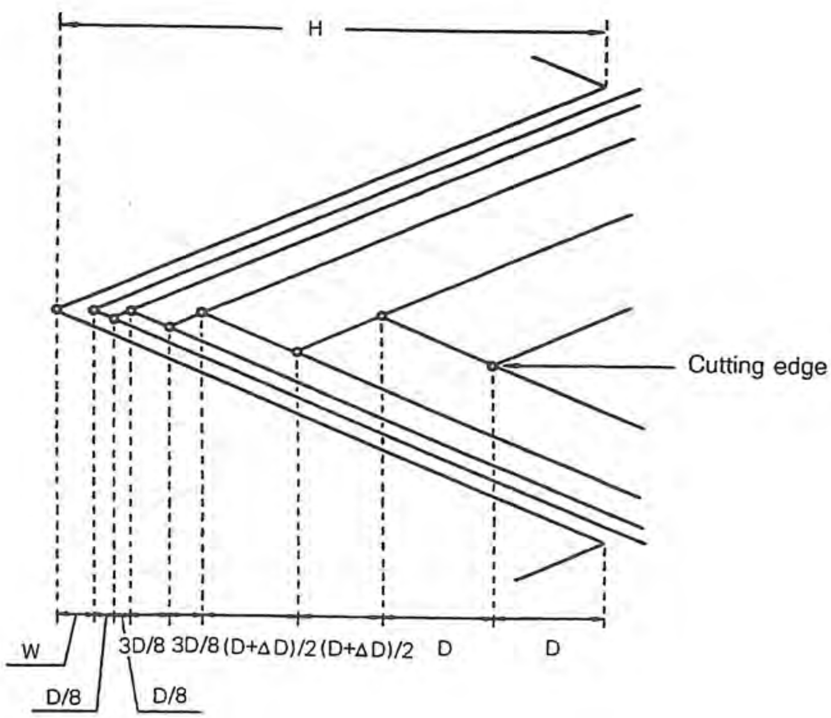


Transverse Thread Cutting Cycles

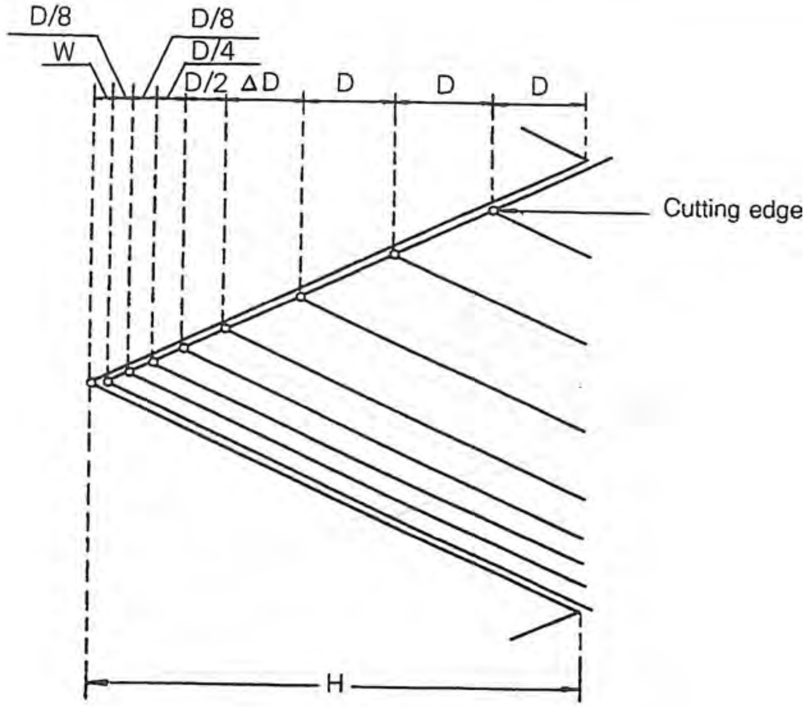
M32 + M73 Mode



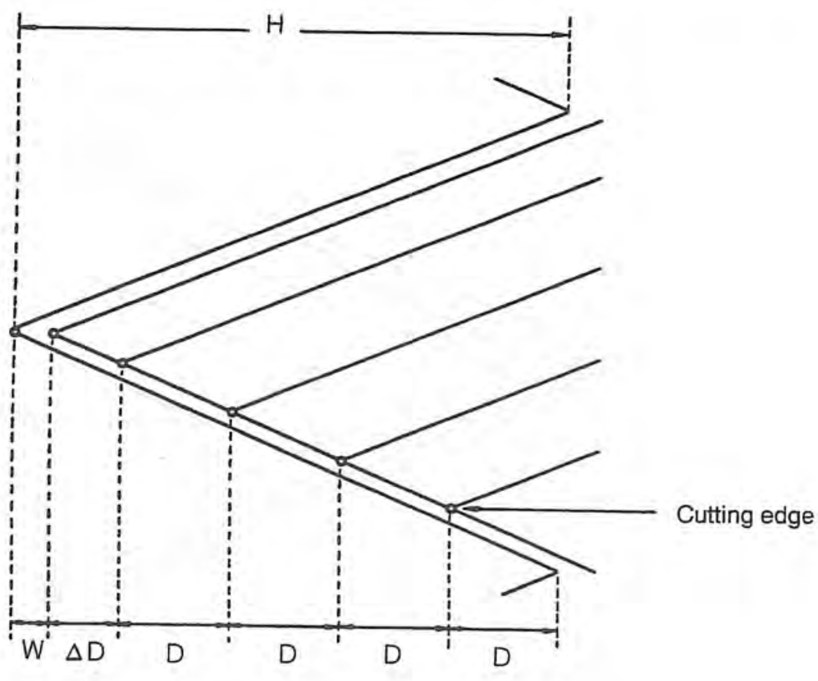
M33 + M73 Mode



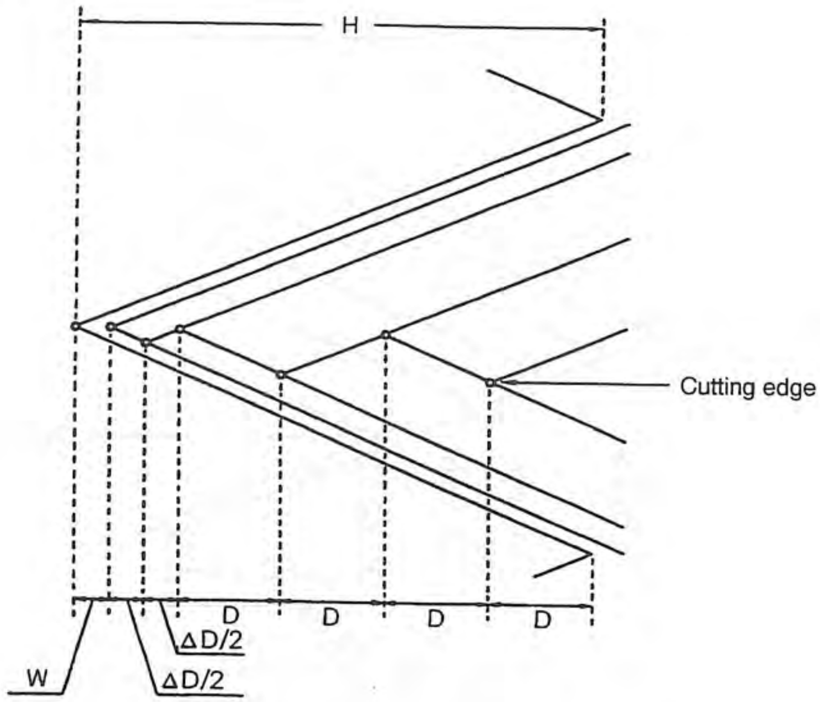
M34 + M73 Mode



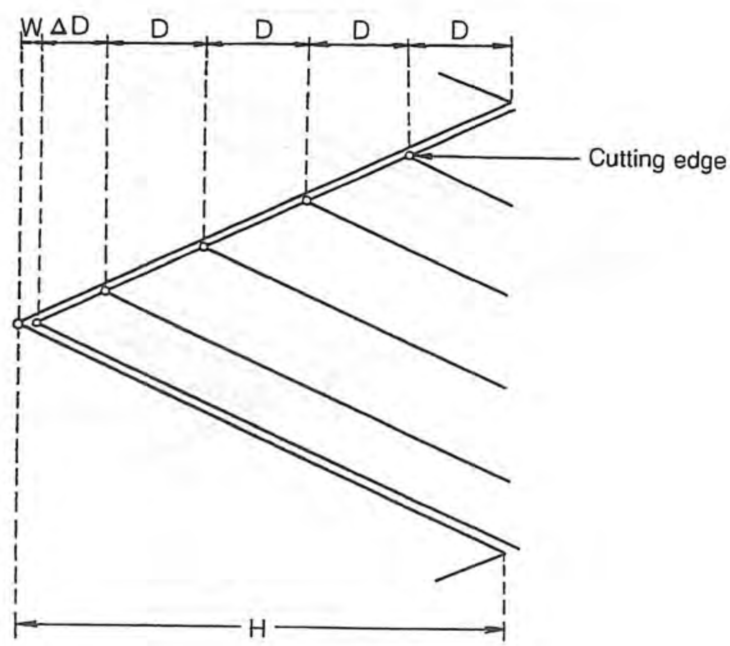
M32 + M74 Mode



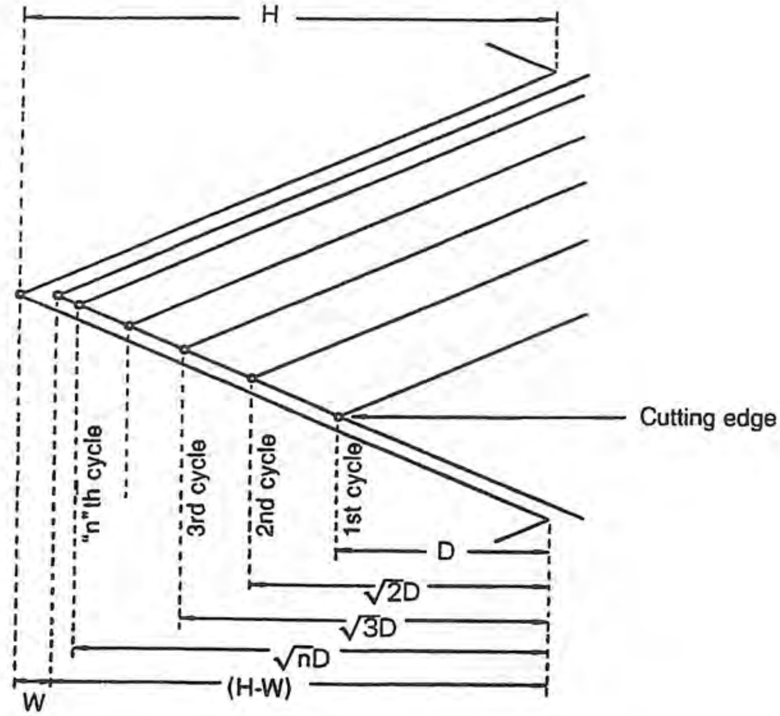
M33 + M74 Mode



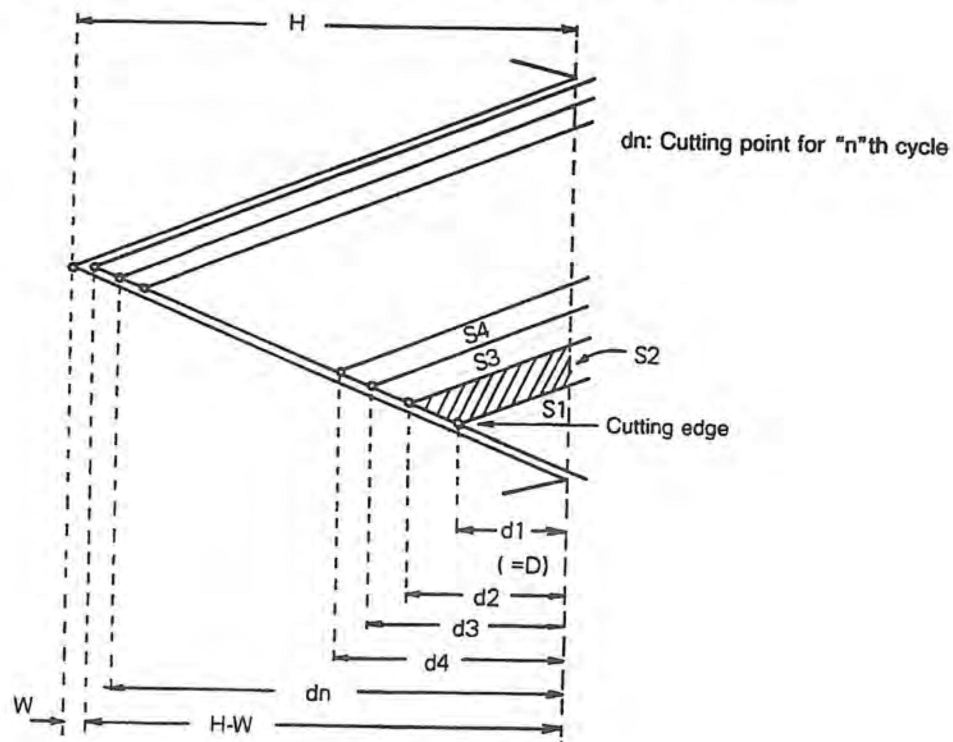
M34 + M74 Mode



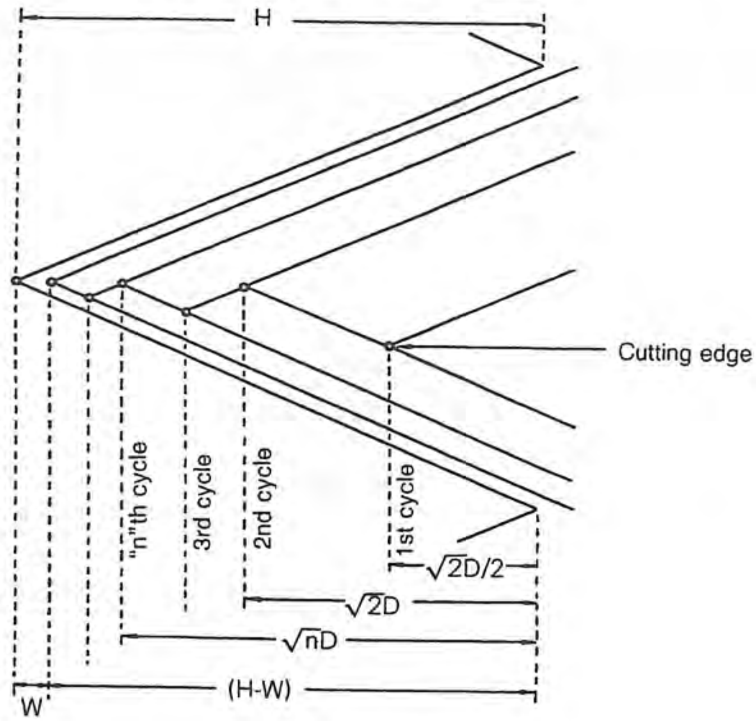
M32 + M75 Mode (infeed pattern 3 $D^2 \geqq \{H^2 - (H - U\ (W))^2\}$)



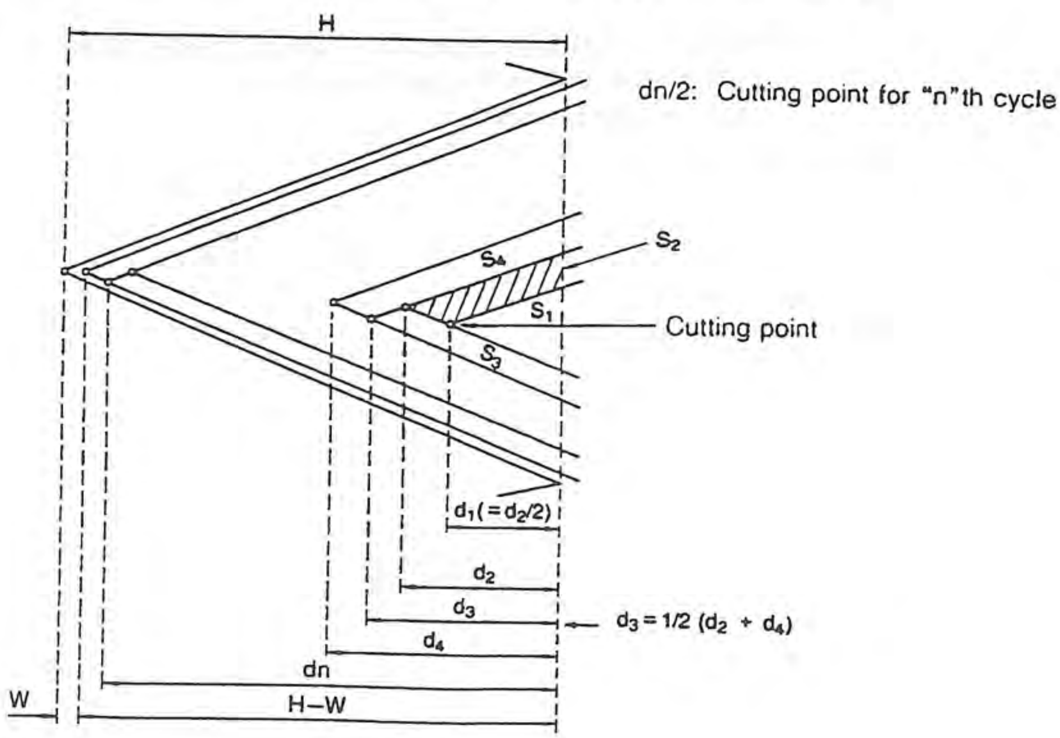
M32 + M75 Mode (infeed pattern 3 $D^2 < \{H^2 - (H - U\ (W))^2\}$ or infeed pattern 4)



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



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

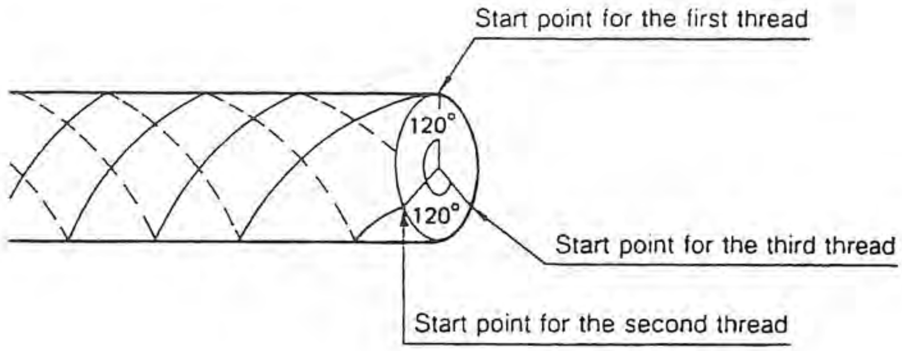


4-4. Multi-thread Thread Cutting Function in Compound Fixed Thread Cutting Cycle

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

Example of Machining Loci:

Q = 3:



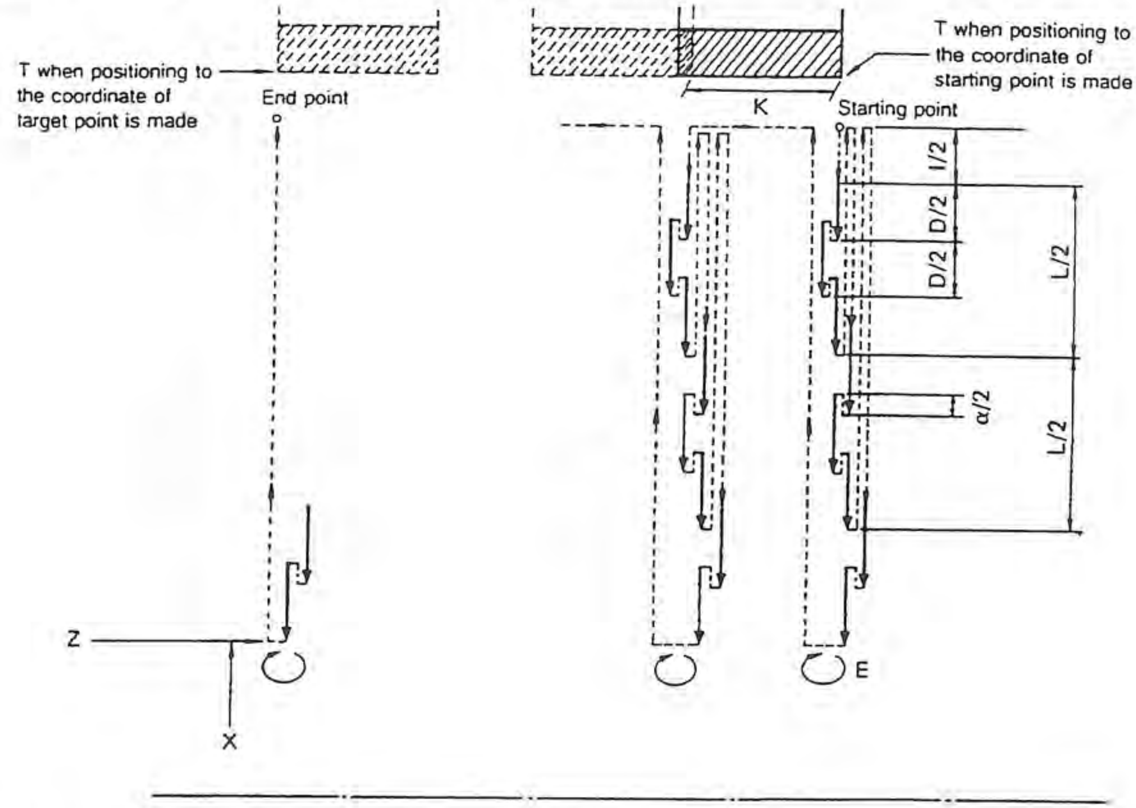
[Supplement]

1. Command range: 0 to 9999
2. Omission of a Q command is regarded as Q1.
3. In 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.

5. Grooving/Drilling Compound Fixed Cycle

5-1. Longitudinal Grooving Fixed Cycle (G73)

In G73 mode, grooving cycle as shown below is performed.



Format:

[G73 X_ Z_ I_ K_ D_ L_ F_ E_ T_]

Description of each word:

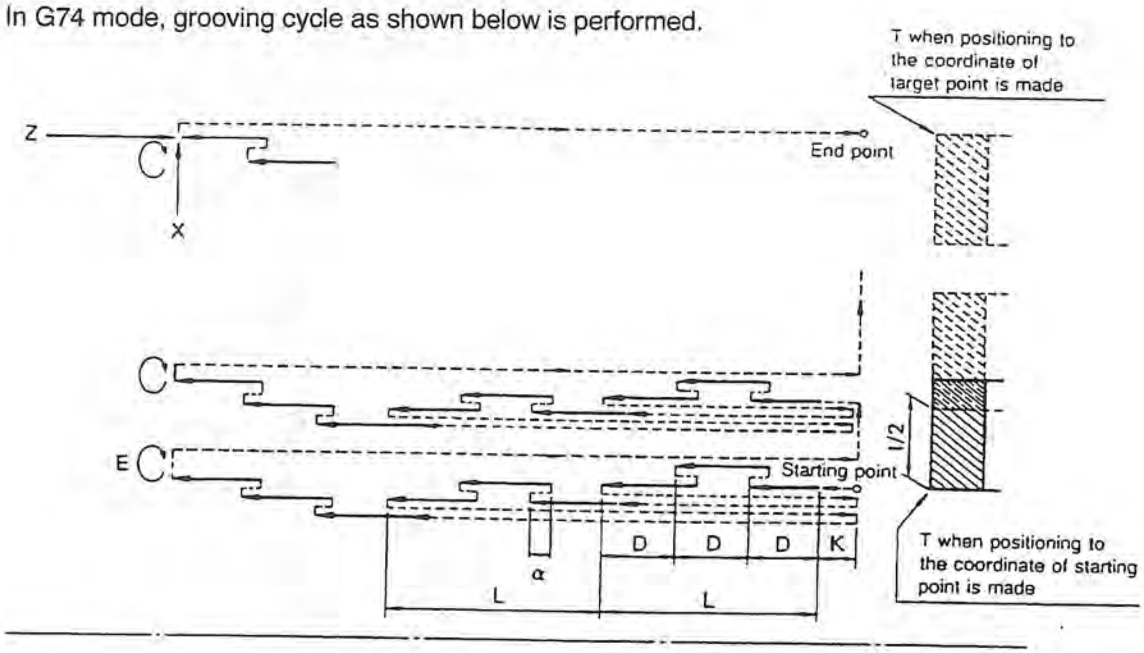
- X : X coordinate of target point
- Z : Z coordinate of target point
- I : Shift amount in X-axis direction (in diameter; 0 if no I word is provided)
- K : Shift amount in Z-axis direction (0 if no K word is provided)
- D : Depth of cut (infeed amount)
- L : Total infeed amount for tool withdrawal motion (in diameter; tool sequence is not performed when L word is not specified.)
- DA: Retraction amount "α" is specified. When no DA word is provided, 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.

  Note that DA command is not effective for A specification.
- E : Duration of dwell motion when target point on X-axis is reached (Command unit is the same as an F word in G04 mode.) (If no E word is provided, this sequence is not performed.)

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

5-2. Transverse Grooving/Drilling Fixed Cycle (G74)

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



Format:

[G74 X_ Z_ I_ K_ D_ L_ F_ E_ T_]

Description of each word:

X : X coordinate of target point

Z : Z coordinate of target point

I : Shift amount in X-axis direction (in diameter; 0 if no I word is provided)

K : Shift amount in Z-axis direction (0 if no K word is provided)

D : Depth of cut (infeed amount)

L : Total infeed amount for tool withdrawal motion
(The sequence is not performed when L word is not specified.)

DA: Retraction amount "α" is specified. When no DA word is provided, the amount set for the optional parameter (OTHER FUNCTION 1) Pecking amount in grooving and drill cycle is used as the retraction amount. This applies both in the G94 and G95 modes.

E : Duration of dwell motion 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 provided, 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 provided, tool offset number selected at positioning to the starting point of grooving cycle is selected. T command after this block is the one designated when positioning to the starting point is performed.)

5-3. Axis Movements in Grooving/Drilling Compound Fixed Cycle

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

If no I or K word is specified, this rapid positioning cycle is not performed.

(2) After the axis is infeed as much as "D", it retracts by "DA" at a rapid traverse rate.

Such peck-feeding cycle is repeated until the programmed target point along infeed axis direction.

(3) When an L word is provided in a 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 along infeed axis direction is reached, dwell motion is activated for the duration commanded in an E word. If no E word is provided, dwell motion is not performed.

After that, the axis returns to the cycle starting point level, and then shift is made along another axis direction by the commanded amount "K" or "I" at a rapid traverse rate.

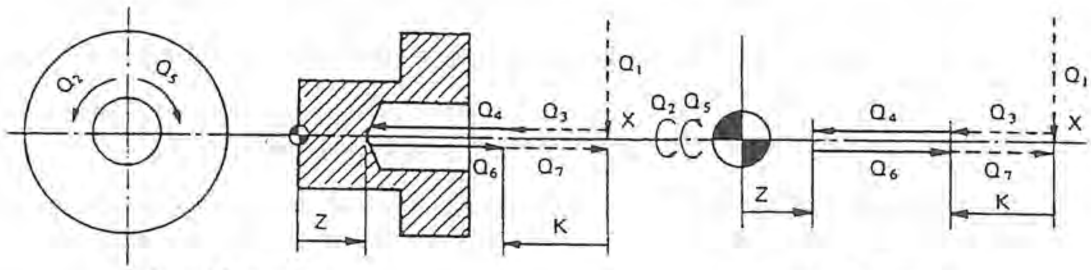
(5) This completes one grooving cycle. The steps (1) through (4) are repeated to machine the desired groove.

(6) When the offset tool position (offset number specified in the same block) reaches or exceeds the target point on X or Z axis direction along shift direction while the grooving cycle with shift is repeated, target point of the shift operation is taken at the final target point of the cycle; 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.

6. Tapping Compound Fixed Cycle

6-1. Right-hand Tapping Cycle (G77)

The compound cycle called out by G77 conducts the tapping cycle as illustrated below.



Actual Example Illustration Example

Format:

[G77 X_ Z_ K_ F_]

Description of each word:

G77: G code to call out tapping compound fixed cycle.
Specify this G code in the next place following 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 : Positioning of X-axis 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 have been positioned at a position where not interfere with the workpiece during this positioning before calling out the G77 cycle.

Q_2 : The spindle rotates clockwise at a speed active before the G77 cycle is called. Therefore, it is necessary to specify required spindle speed 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 : Positioning of Z-axis at a position designated by a K word at a rapid feedrate.

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

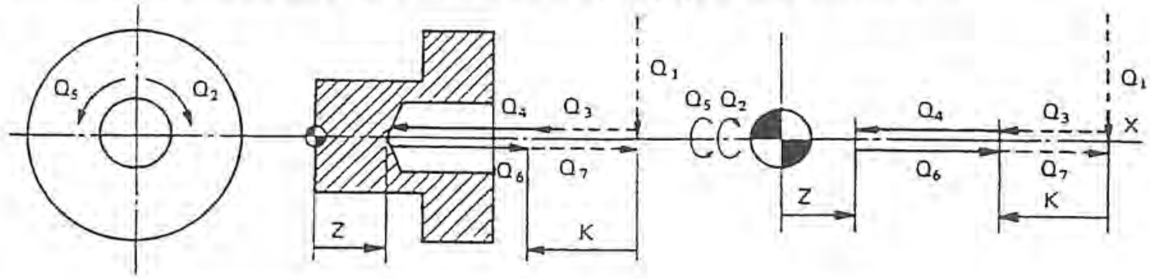
Q_5 : The spindle stops once and then starts in the reverse direction at the same speed as used in infeeding.

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

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

6-2. Left-hand Tapping Cycle (G78)

The compound cycle called out by G78 conducts the tapping cycle as illustrated below.



Actual Example

Illustration Example

Format:

[G78 X_ Z_ K_ F_]

Description of each word:

G78: G code to call out tapping compound fixed cycle.
Specify this G code in the next place following 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 : Positioning of X-axis 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 have been positioned at a position where not interfere with the workpiece during this positioning before calling out the G78 cycle.

Q_2 : The spindle rotates counterclockwise at a speed active before the G77 cycle is called. Therefore, it is necessary to specify required spindle speed 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 : Positioning of Z-axis at a position designated by a K word at a rapid feedrate.

Q_4 : Tapping 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 : Z-axis retraction to a point reached in Q_4 cycle at a cutting feedrate.

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

[Supplement]

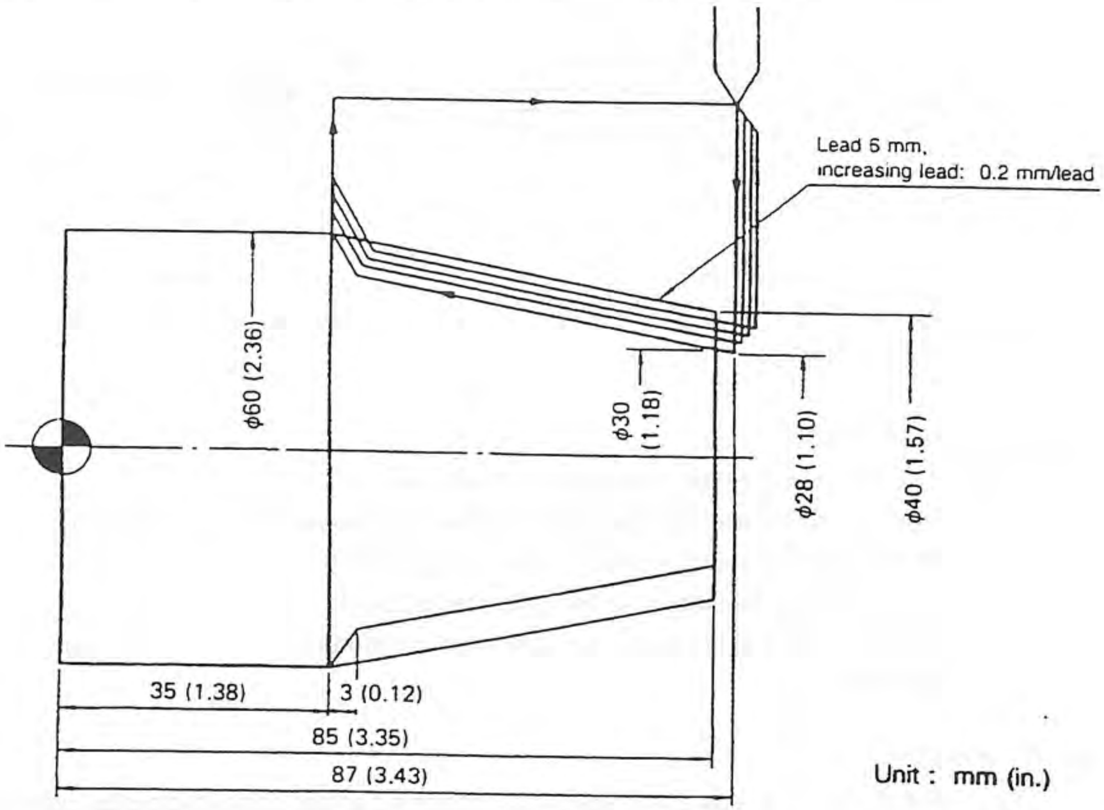
1. While the tapping compound cycle is being executed, the feedrate override is fixed at 100%.

2. 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.

3. After the execution of the tapping compound cycle (G77, G78), the spindle stops and the stop state is remained. When cutting is to be conducted continuously, specify the spindle start command before progressing to the succeeding operation.

7. Application of Compound Fixed Cycle

7-1. Application of Longitudinal Thread Cutting Compound Fixed Cycle (G71)



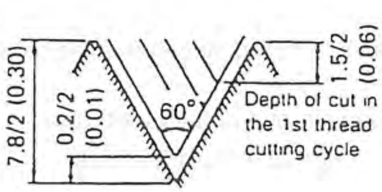
Unit : mm (in.)

M32 + M75:

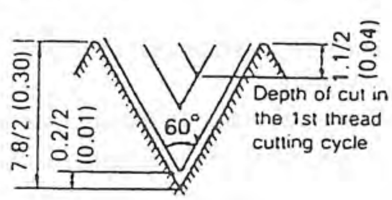
```
N0001  G71  X28  Z35  I11  B60  D1.5  U0.2  H7.8  L3  E0.2  F6
$                                     M23  M32  M75
```

M33 + M74:

```
N0001  G71  X28  Z35  I11  B60  D1.1  U0.2  H7.8  L3  E0.2  F6
$                                     M23  M33  M74
```

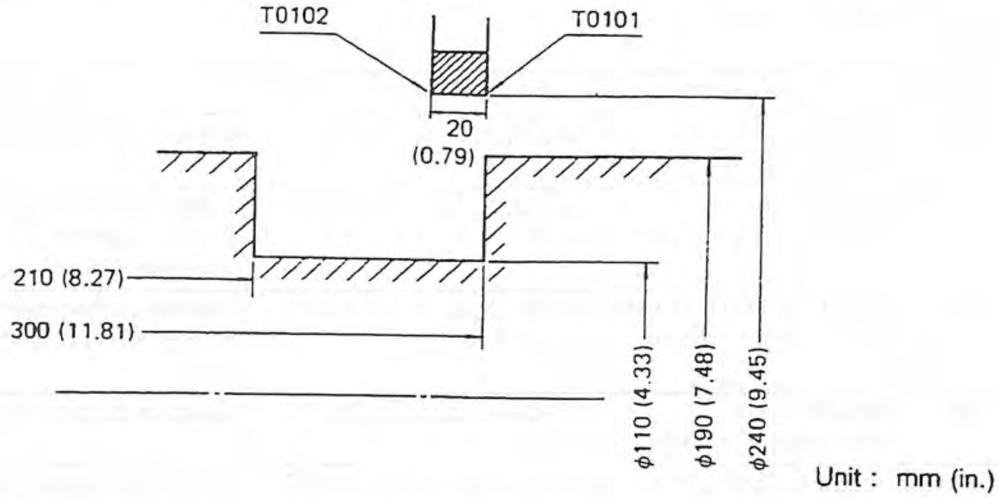


M32 + M75



M33 + M74

7-2. Application of Longitudinal Grooving Compound Fixed Cycle (G73)

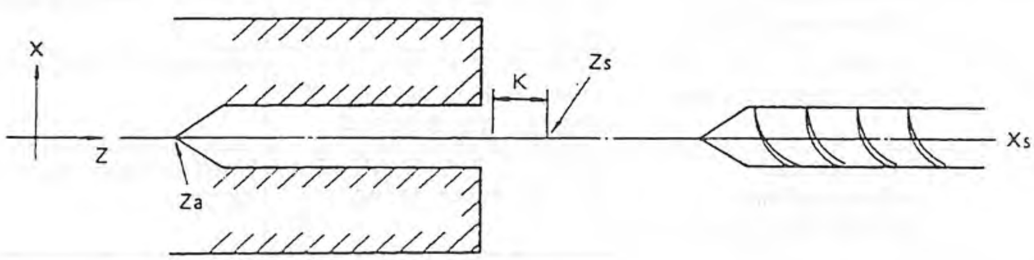



```
N0001  G00  X1000  Z1000  S300  T0101  M03  M42
N0002       X240   Z300
N0003  G73  X100   Z210   I45 K18  D20  E0.2  F0.3  T0102
```

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

7-3. Application of Transverse Grooving/Drilling Compound Fixed Cycle (G74)

Drilling Cycle:




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

NOTICE : G74 block must contain both X and Z words.

8. Compound Fixed Cycle

8-1. Programming Format

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 the 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 the 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.

NOTICE : (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 feedrate per C-axis revolution.

(2) In this G181 through G184, G189, and G190 modes, feedrates can be programmed only in the G94 (mm/min) mode. Feedrate command in units of mm/rev is not accepted.

(3) In G181 through G184, G189, and G190 modes, the control judges cutting direction by programmed I and K words: I for X-axis direction cutting and K for Z-axis direction cutting.

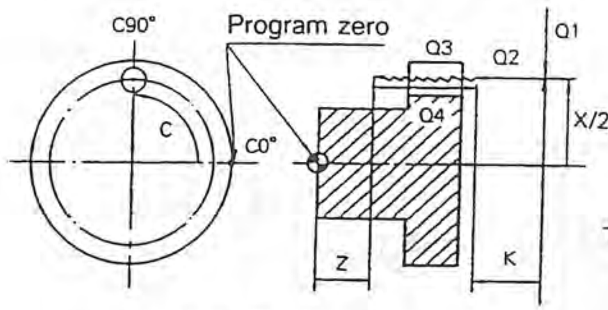
(4) "SA =" command is effective only in G185 through G188 modes.

8-2. Basic Axis Motions

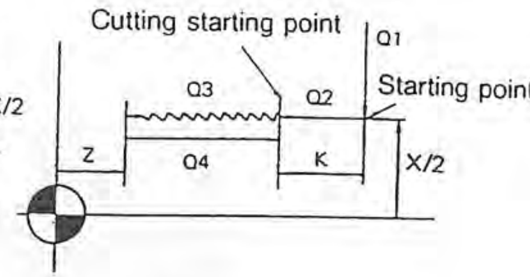
When compound fixed cycle is called for, commands specified in one block result in several steps of axis movement as detailed in this section.

(1) G181, G182, G183, G184, G178, G179 and G189

(a) Face Machining (With K command)

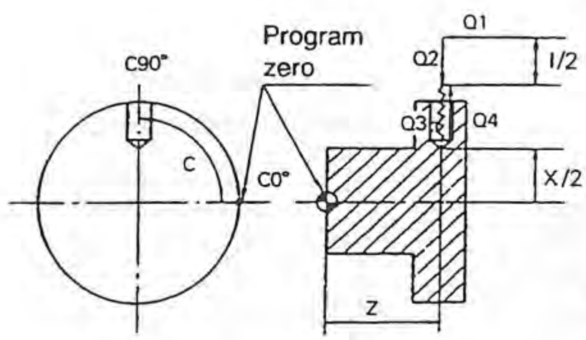


Actual Example

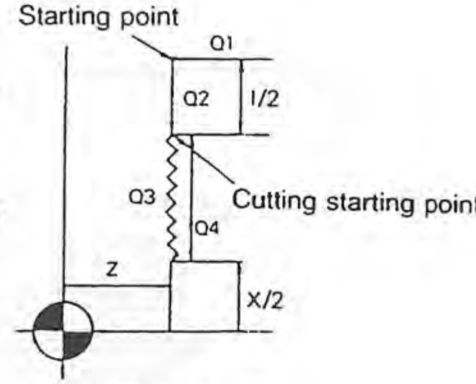


Diagram

(b) Side Machining (With I command)



Actual Example



Diagram

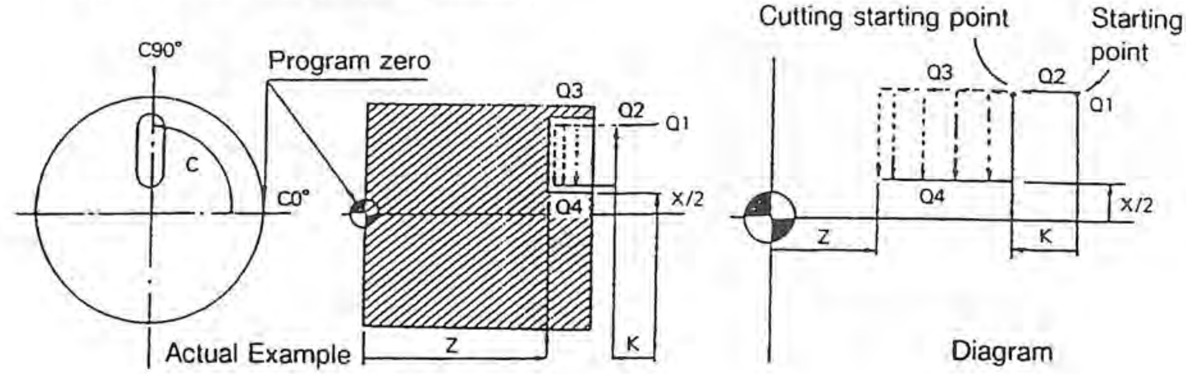
	Face Machining (With K command)	Side Machining (With I command)
Q1	Positioning of X- and C-axis at the rapid feedrate	Positioning of Z- and C-axis at the rapid feedrate
Q2	Positioning of Z-axis at the point "Q1 – K" at the rapid feedrate	Positioning of X-axis at the point "Q1 – K" at the rapid feedrate
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.

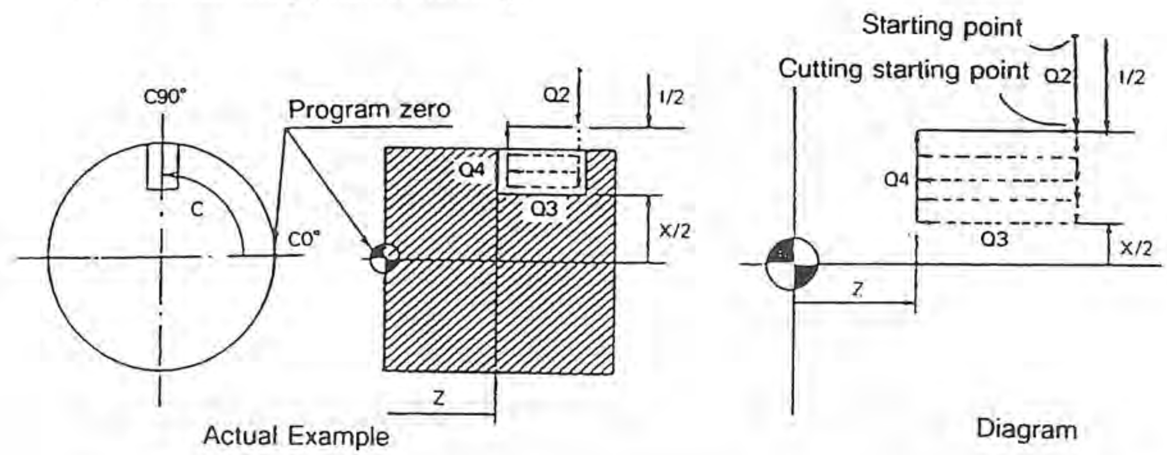
NOTICE : (1) For K or I commands, only positive values are allowed. If a negative value is specified, an alarm occurs.
(2) Axis feed direction is determined automatically. The axis is then fed by amount K or I.
(3) In the Q3 cycle, the end point of cutting may be specified by an R command.

(2) In the G190 mode, the following cycle is carried out in a single block of commands.

(a) Face Machining (With K command)



(b) Side Machining (With I command)



	Face Machining (With K command)	Side Machining (With I command)
Q1	Positioning of C-axis at the rapid feedrate	Positioning of 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 – K" 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 Q3 cycle start point at the rapid feedrate. The cycle is repeated until Z-axis reaches the programmed Z level.	X-axis returns to the Q3 cycle start point at the rapid feedrate. The cycle is repeated until X-axis reaches the programmed X level.

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

NOTICE : (1) For K or I commands, only positive values are allowed. If a negative value is specified, an alarm occurs.

(2) Axis feed direction is determined automatically. The axis is then fed by amount K or I.

- Return point designation for the fixed cycle (G181, G182, G183, G184, G189, G190, G178 or G179):

  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 for optional parameter (MULTIPLE MACHINING) Multi cycle return point or M code specified in a part program.

  Optional parameter = 0 .... Return to the cutting start point (Q3 start point)

  Optional parameter = 1 .... Return to the rapid feed start point (Q2 start point)

  M code M136 .... Designation of shape in compound fixed cycle

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

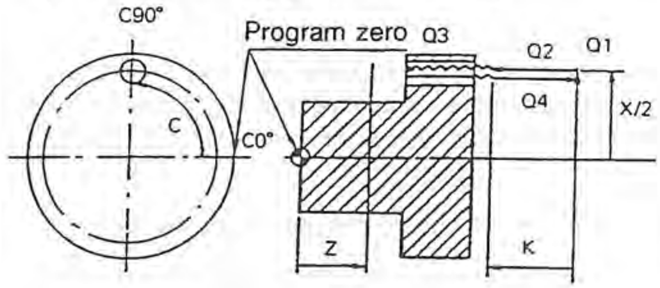
  This M code is cleared by the reset operation and it is effective only in the specified block.

  Note: *M code is given priority over the optional parameter setting.*

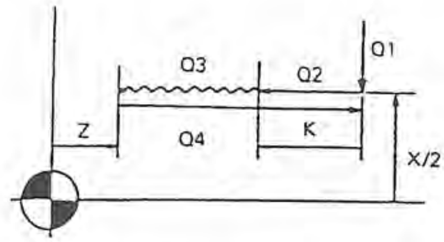
  When no M code is designated, the optional parameter setting becomes effective.

(3) Basic axis motions of 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 M136 command exists in the program.

(a) Face Machining (With K command)

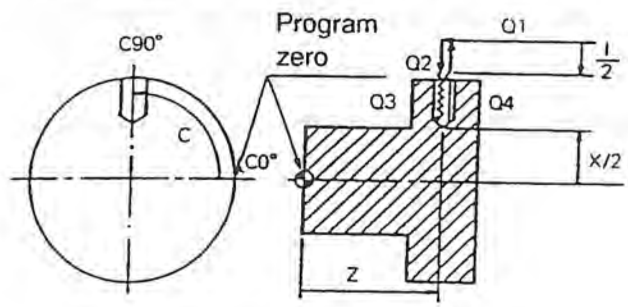


Actual Example

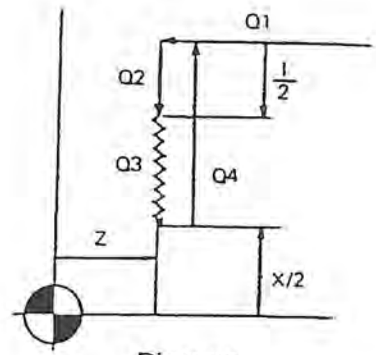


Diagram

(b) Side Machining (With I command)



Actual Example



Diagram

	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 – K" 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	X-axis returns to Q3 where cutting started at a cutting feedrate and then to Q2 at the rapid feedrate.	X-axis returns to Q3 where cutting started at a cutting feedrate and then to Q2 at the rapid feedrate.

- C-axis clamp effective/ineffective command (for G181, G182, G183, G184, G189, G190, G178 or G179)

  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.

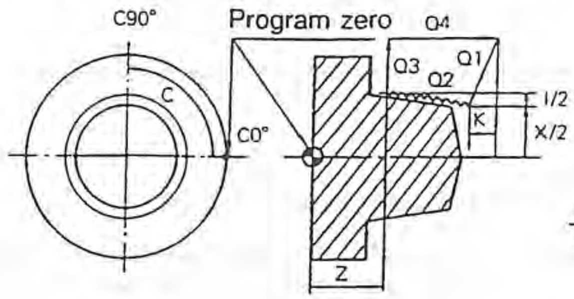
  M169 is only effective within a block.

- Ignoring the M-tool constant speed rotation answer for M140 (tapping cycle)

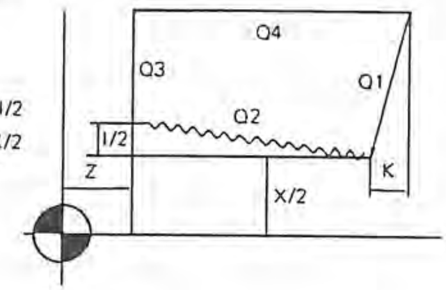
  In the tapping cycle, cutting feed starts after receiving the M-tool constant speed rotation answer. In this case, a time lag occurs between the start of tool rotation and the start of cutting feed. Normally, the time lag is adjusted by a mechanism in the tapping unit. If the time lag cannot be adjusted,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

(4) G185, G186, G187, and G188

(a) Longitudinal Thread Cutting (G185 and G187)

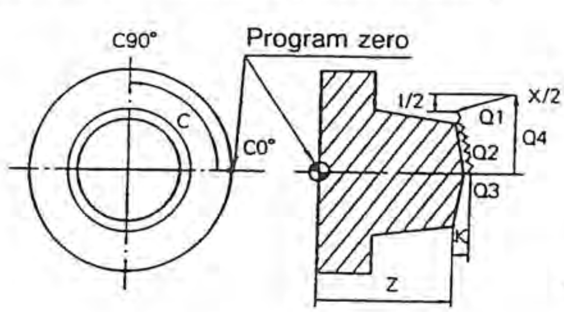


Actual Example

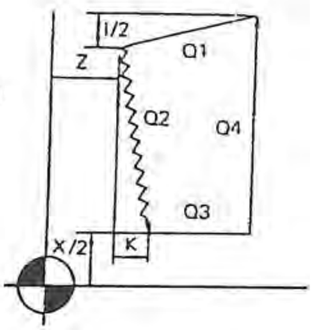


Diagram

(b) Transverse Thread Cutting (G186 and G188)



Actual Example



Diagram

	Longitudinal Thread Cutting (G185 and G187)	Transverse Thread Cutting (G186 and G188)
Q1	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
Q2	Cutting along X- (X±I), Z- and C-axis	Cutting along X-, Z- (Z±K) and C-axis
Q3	Positioning of X-axis at the starting point of sub cycle Q1 at the rapid feedrate	Positioning of Z-axis to the starting point of sub cycle Q1 at the rapid feedrate
Q4	Positioning of Z-axis at the starting point of sub cycle Q1 at the rapid feedrate	Positioning of X-axis to the starting point of sub cycle Q1 at the rapid feedrate

In G188 or G187 mode operation, only sub cycles Q1 and Q2 are carried out.

8-3. Explanations on Address Characters

X : For cutting on end face and longitudinal thread cutting, "X" indicates the X-coordinate of the cycle starting point.

For cutting on 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 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 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 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 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 one peck feeding in deep-hole drilling and key way cutting
Start position of tapping by 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 cycle). C-axis rotation speed command.

This SA= command is programmed to obtain the axis movement amount of C-axis in G185 through G188 thread cutting cycle.

R : Infeed amount for drilling cycle

Specify the distance from the cutting starting point. The sign of the R command indicates the direction of cutting.

An R command in X-axis direction should be given in diametral value.

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

8-4. M Codes

M211 and M212 : Designation of key way cutting direction

M211 .... One-directional cutting

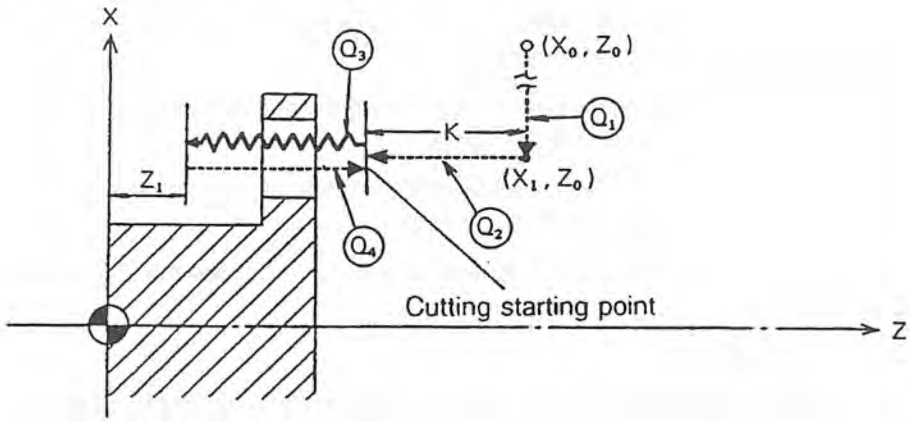
M212 .... Zigzag cutting

M213 and M214 : Designation of key way cutting method

M213 .... Designated infeed

M214 .... Equal infeed

8-5. Drilling Cycle (G181)

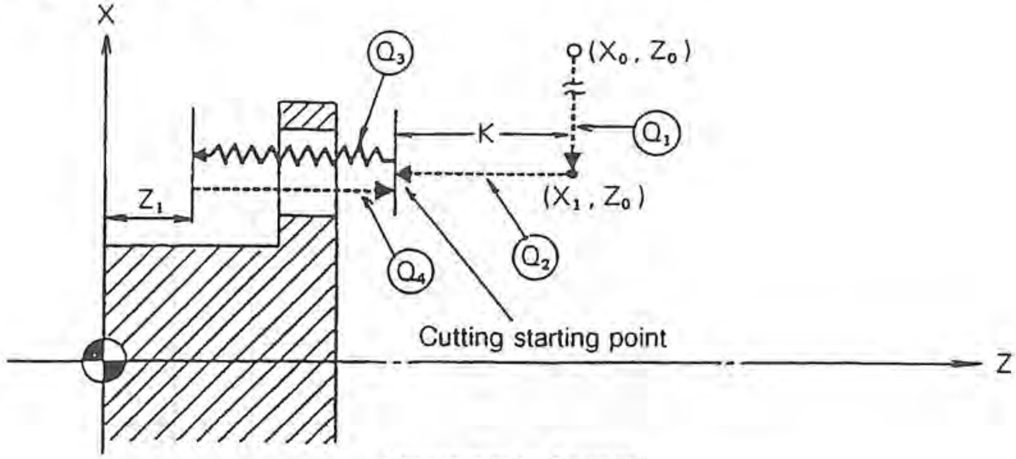


Programming Format

```
              ⁝
N100   G00    X0    Z0
N101   G94                     SB=
N102   G181   X1    Z1    C    K     F
N103   G180
              ⁝
```

	Drilling Cycle (G181)
Q1	Positioning in the G00 mode at the point specified by X1, Z0, and C
	After the completion of positioning, the M-tool spindle starts rotating in the forward direction.
Q2	Positioning of Z-axis at a point "–K" from Z0
	After the completion of positioning, the C-axis is clamped.
Q3	Cutting up to Z1 in the G01 mode
Q4	Positioning at the cutting starting point in the G00 mode
	After the completion of positioning, the C-axis is unclamped.

8-6. Boring Cycle (G182)



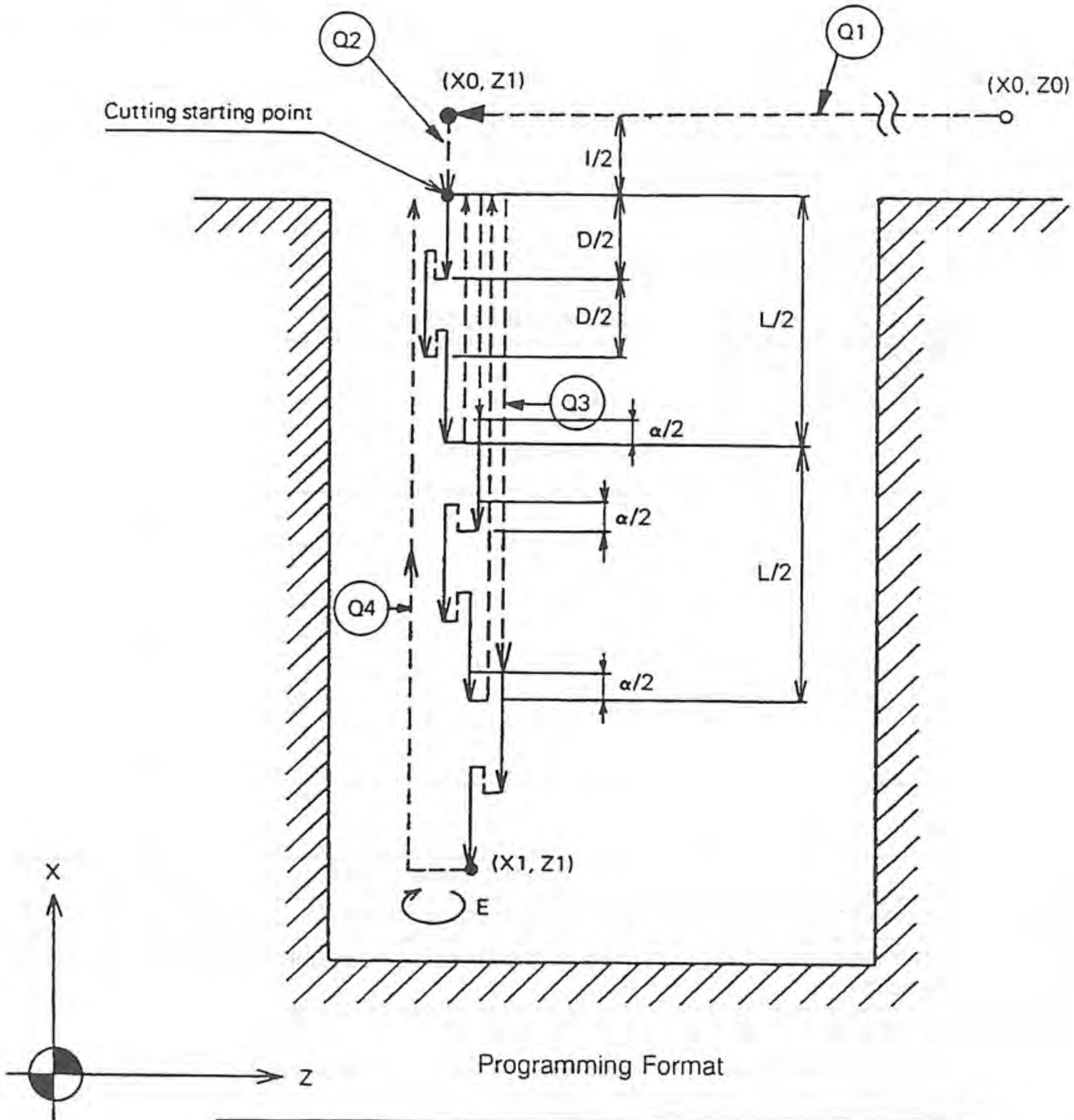
Programming Format

```
                    (
N100    G00   X0     Z0
N101    G94                       SB=
N102    G182  X1     Z1     C     K      F   E
N103    G180
                    (
```

	Boring Cycle (G182)
Q1	Positioning in the G00 mode at the point specified by X1, Z0, and C
	After the completion of positioning, the M-tool spindle starts rotating in the forward direction.
Q2	Positioning of Z-axis at a point "–K" from Z0
	After the completion of positioning, the C-axis is clamped.
Q3	Cutting up to Z1 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 rotating.
Q4	Positioning 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 cycle Q3 should be programmed in the same manner as an F command in the G04 mode.

8-7. Deep Hole Drilling Cycle (G183)



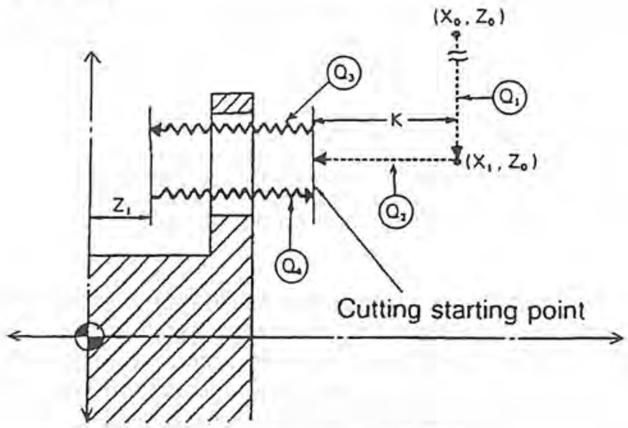
Programming Format

```
                    ⌇
N100    G00   X0    Z0
N101    G94                             SB=
N102    G183  X1    Z1    C   I   F   E   D   L
N103    G180
                    ⌇
```

	Deep Hole Drilling Cycle (G183)
Q1	Positioning in the G00 mode to the point specified by X0, Z1 and C
	After the completion of positioning, the M-tool spindle starts rotating in the forward direction.
Q2	Positioning of X-axis at a point "–I" from X0
	After the completion of positioning, the C-axis is clamped.
Q3	Drilling cycle in step feed mode is carried out up to X1. The step feed means the axis movement as 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 retracting 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 infeed to the previous drilled depth and then the cycle indicated above is repeated up to the target point X1.
	At the bottom of the hole, the dwell function is activated for time duration "E"(omissible).
Q4	Positioning at the cutting starting point in the G00 mode
	After the completion of positioning, the C-axis is unclamped.

[Supplement] For "α", the value set for optional parameter (MULTIPLE MACHINING) Pecking amount in drilling cycle is used.

8-8. Tapping Cycle (G184)

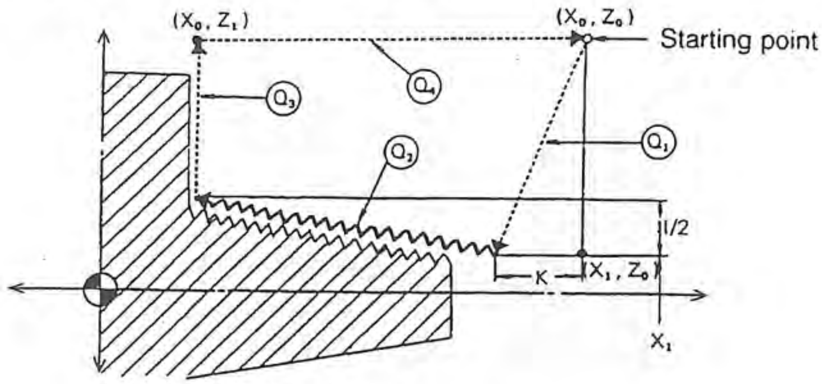


Programming Format

```
                          (
                          )
N100      G00      X0      Z0
N101      G94                         SB=
N102      G184     X1      Z1     C   K        F   E
N103      G180
                          (
                          )
```

	Tapping Cycle (G184)
Q1	Positioning in the G00 mode at the point specified by X1, Z0 and C After the completion of positioning, the M-tool spindle starts rotating in the forward direction.
Q2	Positioning of the Z-axis at a point -K from Z0 After the completion of positioning, the C-axis is clamped.
Q3	Cutting up to Z1 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.
Q4	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)



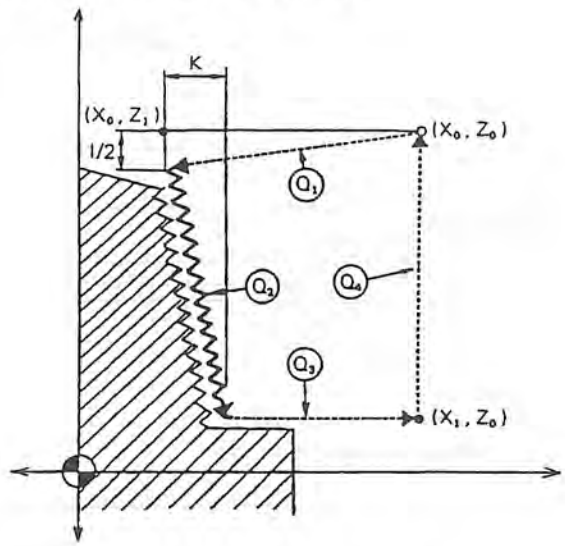
Programming Format

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

	Longitudinal Thread Cutting Cycle (G185)
Q1	Positioning in the G00 mode at the point specified by X1, Z0-K and C
	After the completion of positioning, the M-tool spindle starts rotating in the forward direction.
Q2	The C-axis starts rotation.
	Thread cutting cycle is carried out up to (X1+I, Z1) point in the G01 mode.
	After the completion of thread cutting, the C-axis stops rotation.
Q3	Positioning in the G00 mode at X0
Q4	Positioning in the G00 mode at the starting point

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

8-10. Transverse Thread Cutting Cycle (G186)



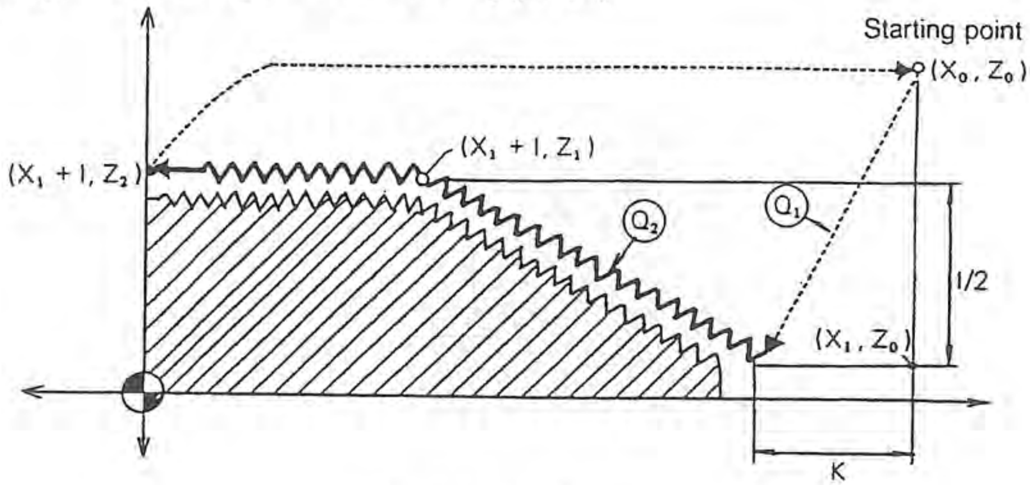
Programming Format

```
                              ∫
N100      G00      X0      Z0
N101      G95                                  SB=
N102      G186     X1      Z1      C           I    K    SA=
N103      G180
                              ∫
```

	Transverse Thread Cutting Cycle (G186)
Q1	Positioning in the G00 mode to the point specified by X0-I, Z1 and C
	After the completion of positioning, the M-tool spindle starts rotation in the forward direction.
Q2	The C-axis starts rotation.
	Thread cutting cycle is carried out up to (X1, Z1+K) point in the G01 mode.
	After the completion of thread cutting, the C-axis stops rotation.
Q3	Positioning in G00 mode at Z0
Q4	Positioning in G00 at starting point

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

8-11. Longitudinal Straight Thread Cutting (G187)



Programming Format

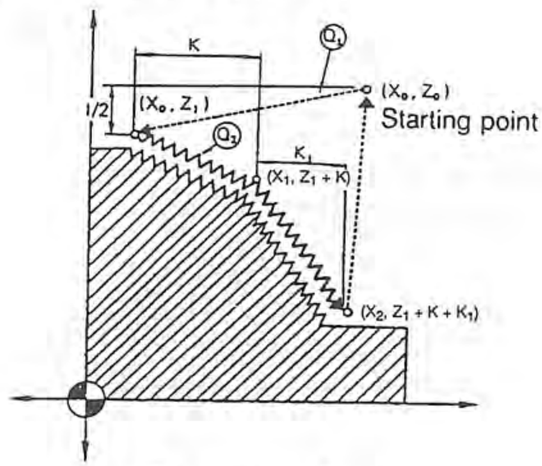
```
                    {
N100    G00    X0   Z0
N101    G95                   SB=
N102    G187   X1   Z1    C   I   K   SA=
N103                Z2                SA=
N104    G180
                    {
```

Since the G187 cycle contains only Q1 and Q2 cycles, designation of G187 in succession can cut the threads continuously.

	Longitudinal Straight Thread Cutting (G187)
Q1	Positioning in the G00 mode to the point specified by X1 and Z0-K
	After the completion of positioning, the M-tool spindle starts rotation in the forward direction.
Q2	The C-axis starts rotation.
	Thread cutting cycle is carried out up to (X1+I, Z) point in the G01 mode.

Thread cutting cycle is carried out by the commands in sequence N103 up to the commands target point (X1+I, Z2). Then, the axes return 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)



Programming Format

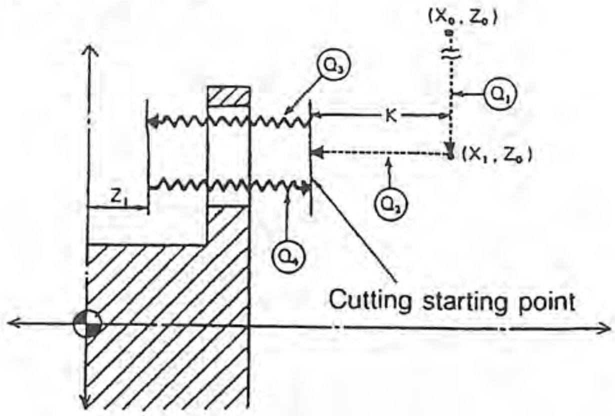
```
                              (
N100      G00       X0      Z0
N101      G95                        SB=
N102      G188      X1      Z1       I   K   F   C   SA=
N103                X2                   K1      C   SA=
N104      G180                (
```

Since the G188 cycle contains only Q1 and Q2 cycles, designation of G188 in succession can cut the threads continuously.

	Transverse Straight Thread Cutting (G188)
Q1	Positioning in the G00 mode to the point specified by X0-I and Z1
	After the completion of positioning, the M-tool spindle starts rotation in the forward direction.
Q2	The C-axis starts rotation.
	Thread cutting cycle is carried out up to (X1, Z+K) point in the G01 mode.

Thread cutting cycle is carried out by the commands in sequence N103 up to the commanded target point (X1+I, Z2). Then, the axes return to the starting point at the rapid feedrate by the command G180 (cancel) specified in the N104 sequence.

8-13. Reaming/Boring Cycle (G189)



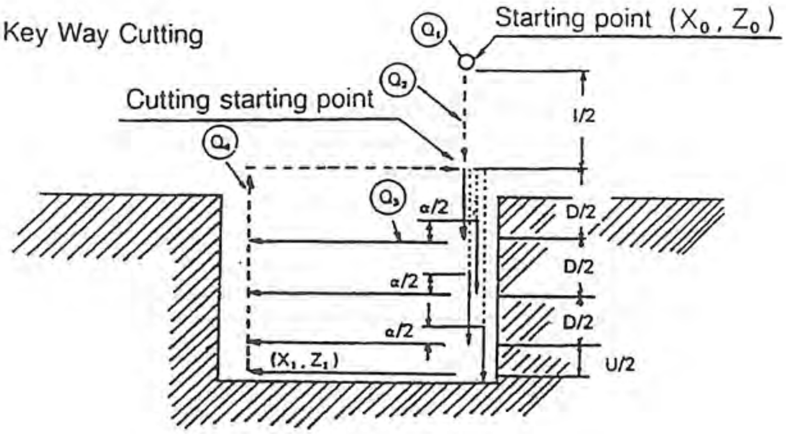
Programming Format

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

	Reaming/Boring Cycle (G189)
Q1	Positioning in the G00 mode to the point specified by X1, Z0 and C
	After the completion of positioning, the M-tool spindle starts rotating in the forward direction.
Q2	Positioning of Z-axis at a point "–K" from Z0
	After the completion of positioning, the C-axis is clamped (omissible).
Q3	Cutting up to Z1 in the G01 mode
	After the completion of cutting, dwell for "E" is carried out (omissible).
Q4	Cutting up to cutting starting point in the G01 mode
	After the completion of axis movement, the C-axis is unclamped.

8-14. Key Way Cutting (G190)

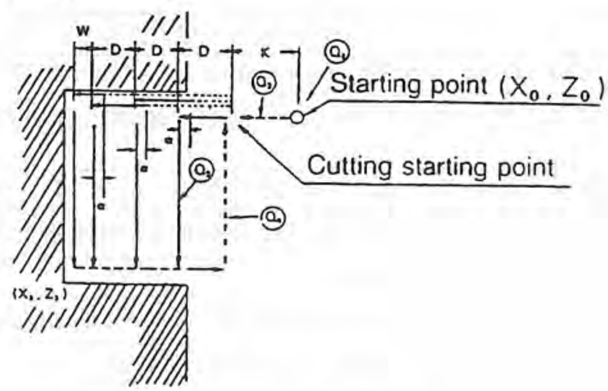
Side Key Way Cutting



Programming Format

```
                              (
N100        G00        X0      Z0
N101        G94                                         SB=
N102        G190  X1   Z1  C   I   D   U   E   F    M211 M213
N103        G180                (
```

Face Key Way Cutting



Programming Format

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

	Deep Hole Drilling Cycle (G183)
Q1	Positioning of X and Z axes at C-axis at the designated position in the G00 mode.
	After the completion of positioning, the M-tool spindle starts rotating in the forward direction.
Q2	Positioning of X-axis (Z-axis for face key way cutting) to a point –I (–K for face key way cutting) from X0 (Z0 for face key way cutting) in the G00 mode.
	After the completion of positioning, the C-axis is clamped.
Q3	Key way cutting is carried in the "one directional, designated infeed" mode.
	For the "one directional, designated infeed" mode, refer to the cutting modes for the key way cutting explained in later pages.
Q4	Positioning at the starting point in the G00 mode.
	After the completion of positioning, the C-axis is unclamped.

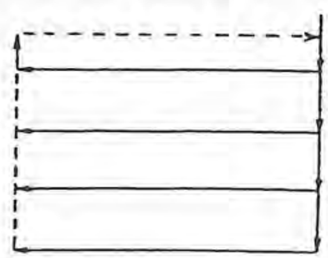
[Supplement] For "α", the value set for optional parameter (MULTIPLE MACHINING) Pecking amount in drilling cycle is used.

Key Way Cutting Modes

The key way cutting modes is selectable from the following four modes by programming the corresponding M code.

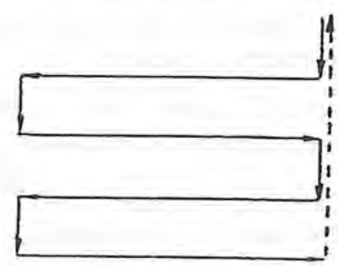
(a) Selection of cutting direction (M211, M212)

One-directional Cutting Mode (M211)



Cutting in one direction

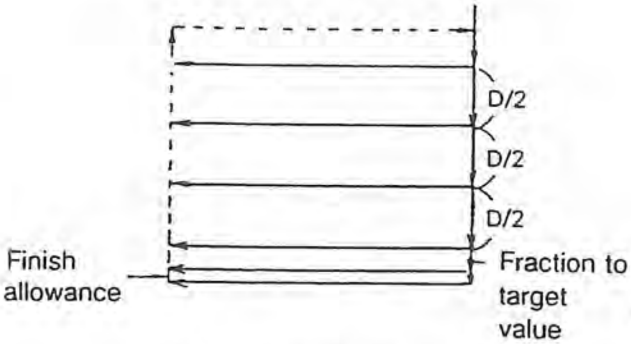
Zigzag Cutting Mode (M212)



Cutting direction varies for each cutting path

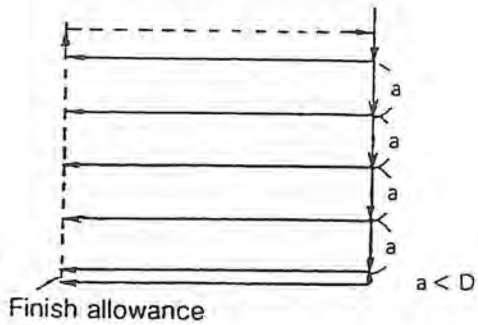
(b) Selection of infeed mode (M213, M214)

Designated Infeed Mode (M213)



Infeed is made by the designated amount "D".
In the final cutting path, the depth equivalent to fraction is cut.

Equal Infeed Mode (M214)



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

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

Note 1: *Before starting fixed cycle mode operation, the C-axis must be placed in unclamped (M146) state.*

Note 2: *In the G181 through G184, G189, G190, G178 and G179 modes, the first cycle is executed in the order of Q1, Q2, Q3 and Q4. However, Q3 and Q4 are repeated after that, when C or Q command is specified.*

The C-axis clamp and unclamp commands (M147 and M146) necessary for repeating sub cycles Q3 and Q4 are automatically generated.

Note 3: *When the called fixed cycle mode is canceled, the control is in the M146 and M13 mode. Specify M147 and M12, if necessary.*

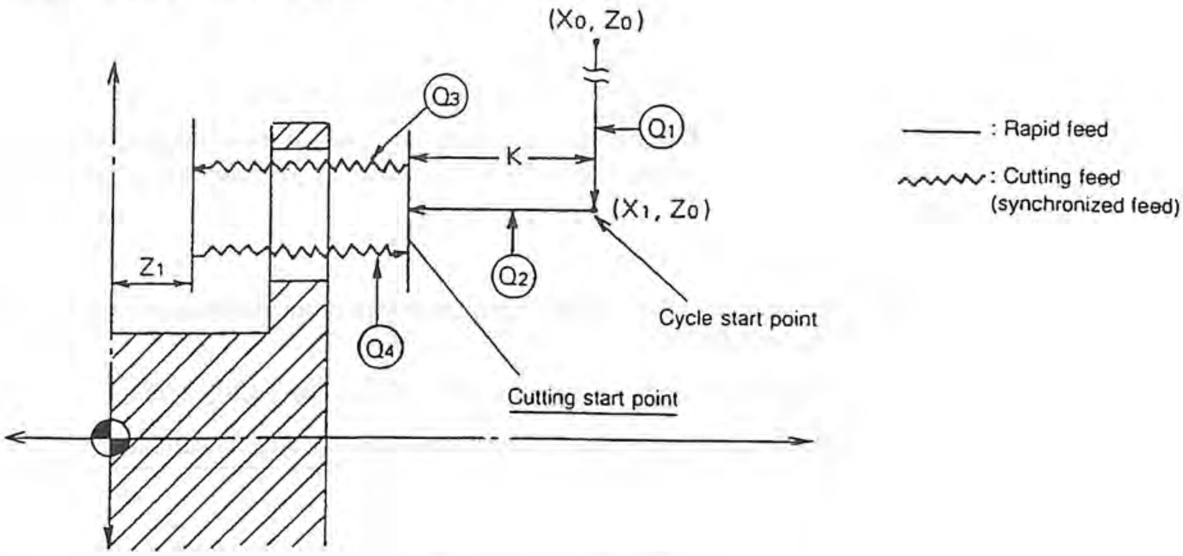
Note 4: *The block right after the one canceling the fixed cycle mode must contain both X- and Z-axis commands.*

8-15. Axis Motion in the Synchronized TApping Cycle

Program format:

G178 ⎤
⎬ X_ Z_ C_ R_ K (I)_ F_ D_ J_ Q_ M141 M136
G179 ⎦

(1) Forward Tapping Cycle (G178)



Programming Format

```
                      (
N100     G00     X0     Z0
N101     G94                        SB=
N102     G178    X1     Z1     C    K        F    D
N103     G180
                      (
```

	Transverse Straight Thread Cutting (G188)
Q1	The axes are positioned at coordinate (x1, C) at the rapid feedrate, and M-spindle rotation is stopped.
Q2	The axes are positioned at the cutting start point (Z0, –K) at the rapid feedrate. With M-spindle one-point clutch models, the M-spindle oscillates after the completion of positioning at (Z0, –K) to engage the clutch. When a D command had been designated, the M-spindle is positioned at point D along the tapping rotation direction.
Q3	After the C-axis had been clamped, the M-spindle is synchronized with the Z-axis to point Z1 while it is rotated in the forward direction. Axis motion is suspended at point Z1 until the M-spindle and the Z-axis come within the droop.
Q4	The M-direction is synchronized with the Z-axis to coordinate (Z0, –K while it is 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 Z0 at the rapid feedrate.

NOTICE : (1) During the execution of steps Q3 and Q4, the M-spindle override and feed axis override are set at 100%.

(2) When the slide hold is designated during the execution of the steps Q3 and Q4, the axes are moved as follows.

Step Q3

The M-spindle and the feed axes are stopped at that position.

The M-spindle is then returned to the cutting start point in synchronized cutting feed while it rotates in the direction opposite to the previous direction and establishes the slide hold state.

Step Q4

The axes are returned to the cutting start point in synchronized feed, and the slide hold states is established.

(3) The dry function is not effective during the execution of steps Q3 and Q4.

(4) The M-spindle is stopped when the synchronized tapping cycle is complete.

(2) Reverse tapping cycle (G179)

Axis motion is the same as in the forward tapping cycles but the M-spindle rotates reversely in steps Q2, Q3, and Q4.

(3) Details of commands

(a) Synchronized tapping command

G178 commands the forwards synchronized tapping cycle while G179 commands the reverse synchronized tapping cycle.

G184 can also be used to command the forward synchronized tapping cycle. In this case, however, "1" must be set for optional parameter (MULTIPLE MACHINING) G184 tapping mode.

(b) Hole location X, C (Z, C)

The position of the tap hole is designated.

X, C .... Tapping on the front surface

Z, C .... Tapping on the side surface

When this commands is omitted, tapping is performed at the position where the previous tapping cycles was carried out.

(c) Hole bottom level Z (X)

The bottom level of the tap hole is designated.

Z .... Tapping on the front surface

X .... Tapping on the side surface

(d) Depth of cut R

The total tap length, from the cutting start point, is designated as R. Cutting direction is indicated by a positive (+) or negative (–) sign proceeding 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.

(e) Shift amount to the cutting start point (I)

The axis is positioned at G00. This point is obtained by shifting the currently effective Z coordinate in the tapping direction by K (I).

K .... Tapping on the front surface

I .... Tapping in the side surface

(f) Cutting feedrate F

The cutting feedrate during tapping is designated as F.

Determine a value so that the M-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-spindle speed (rpm).

G94 mode ..... $F = P \times SB$ (mm/min)

G95 mode ..... $F = P$ (mm/rev)

NOTICE : In the synchronized tapping cycle, G95 can be used.

(g) Tapping start point of the M-spindle D

The position where M-spindle starts tapping is designated as D. The M-spindle is governed by a "constant start point control". This is used when no new D command is assigned, therefore the M-spindle starts tapping at the position defined by the previously designated D command.

NOTICE : When the one-point clutch specification is not selected for the M-spindle clutch, its start point is not guaranteed even when a D command is designated.

(h) Number of threads J

Whenever the number of threads per inch is specified with the inch taps, it is convenient to use J as an indicator, to avoid confusion with metric measurements.

8-16. Repeat Function

When cutting the equally spaced holes, the use of the repeat function simplifies programming

```
N103    G183  X40   Z80  C0  I46  D10  E1  F40  Q6
N109    G180
```

Specify the number of holes to be drilled.

Note that the repeat function is effective for G178, G179 and G181 through G184 and G189, G190 cycles.

(1) When no Q word is specified or Q0 is specified, it is regarded as Q1.

(2) The fixed cycle command associated with Q word is effective only in one block. Be sure to specify G180 in the block following such block.

```
 G181  X100 Z150  C30   I46   E1    Q4
*      X80        C90
 G180
```

The commands in *N051 block cannot be given.

(3) 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.

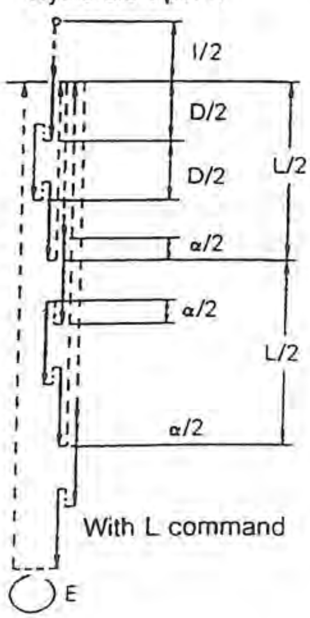
As explained before, in deep-hole drilling cycle (G183), the use of D and E codes breaks the chips in which the drill will tend to get entangled if not properly broken. In addition to such programing, drilling operation with discharging chips outside from the hole being drilled can be also programmed.

Code	Description of Code
L	L: Infeed amount (in 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 – ∝ mm (*1) away (above) from the depth in the previous cut at the rapid traverse; the programmed drilling cycles is repeated until the commanded X (Z) is reached.

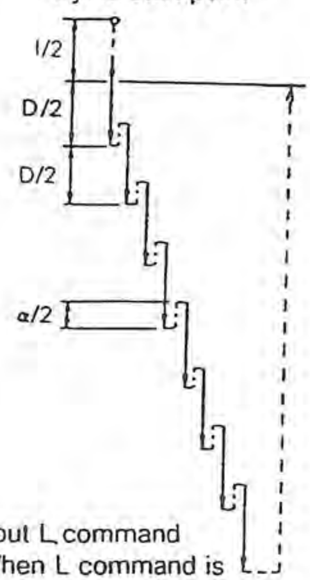
*1 For "α", the value set for optional parameter (MULTIPLE MACHINING) Pecking amount in drilling cycle is used.

To execute chip discharging drilling command in a fixed cycle program, enter the commands in a block as indicated below.

```
G183 X40 Z80 C0 I46 D10 E1 F40 L50
```



With L command



Without L command
When L command is programmed as 0, an alarm results.

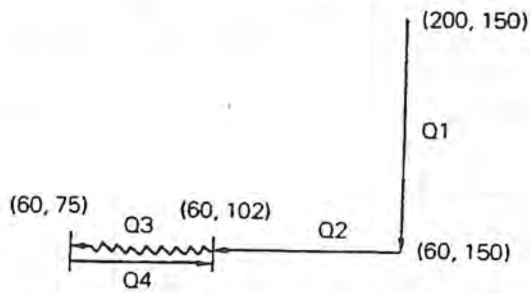
8-18. Drilling Depth Setting (Only for drilling cycles)

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 from the position shifted by I or K, instead of specifying the end point of the drilling cycles.

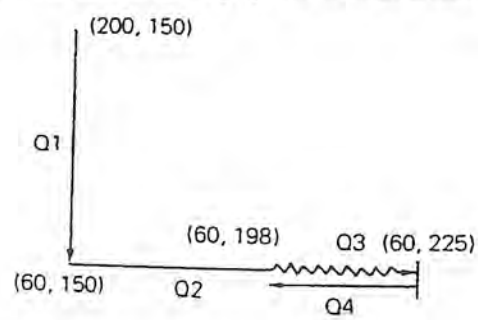
Code	Description of Code
±R	R: Drilling hole depth in drilling cycle For a drilling cycle, the drill hole depth (distance) is specified by an R command. The R command in the X-axis direction must be specified in diameter. The use of an R command allows the drilling cycle to be programmed by the hole depth instead of the end point of the cycle.

If an R command is used in a fixed cycle program, enter the commands in a block as indicated below.

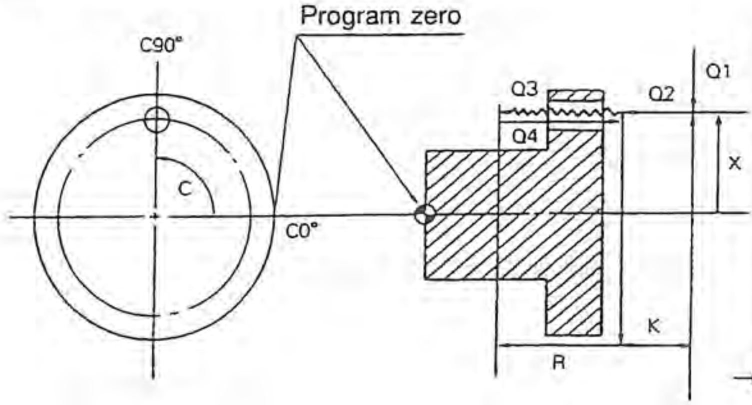
```
N102    G94    X200   Z150     T0101      SB=400
N103    G181   X60    R-27     C0   K48        F40
```



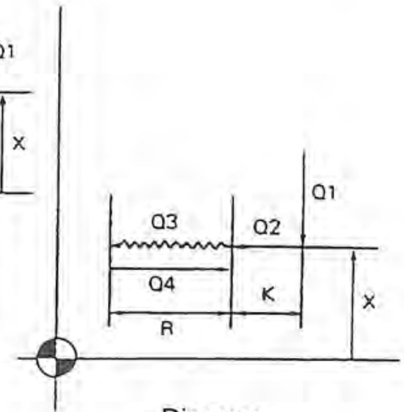
Direction of drilling is determined by the plus or minus of the R command. If R27 is specified instead of R–27 in the program above, direction of drilling cycles is as indicated below.



(1) Face Machining (With K command)

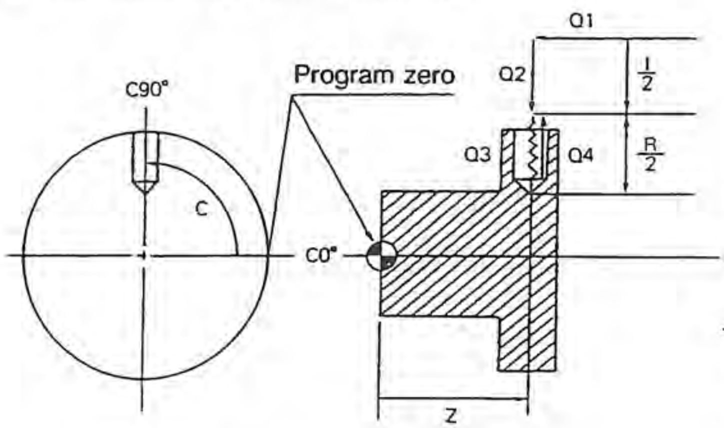


Actual Example

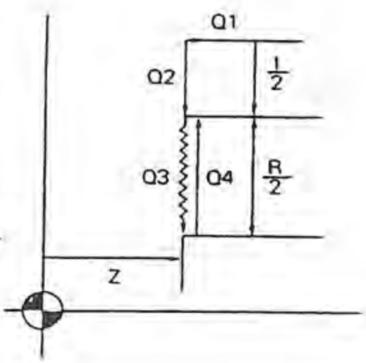


Diagram

(2) Side Machining (With I command)



Actual Example



Diagram

	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 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
Q2	Cutting along Z-axis up to the commanded point Z	Cutting along X-axis up to the commanded point X
Q4	Z-axis returns to the point where cutting started (Q3) 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 (Q3) either at a specified feedrate or the rapid feedrate depending on the called fixed cycle mode.

(a) For I and K commands, only a positive value is allowed. If a negative value is specified, an alarm causes.

(b) The direction of axis feed is automatically determined. The axis is fed in the determined direction by the amount specified by I or K command.

(c) With an R command, 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 is selectable by the parameter data.

Optional parameter (MULTIPLE MACHINING) Multi cycle return point

1 .... Return to the cycle start point

0 .... Return to the cutting start point

(1) Optional parameter (MULTIPLE MACHINING) Multi cycle return point = 1

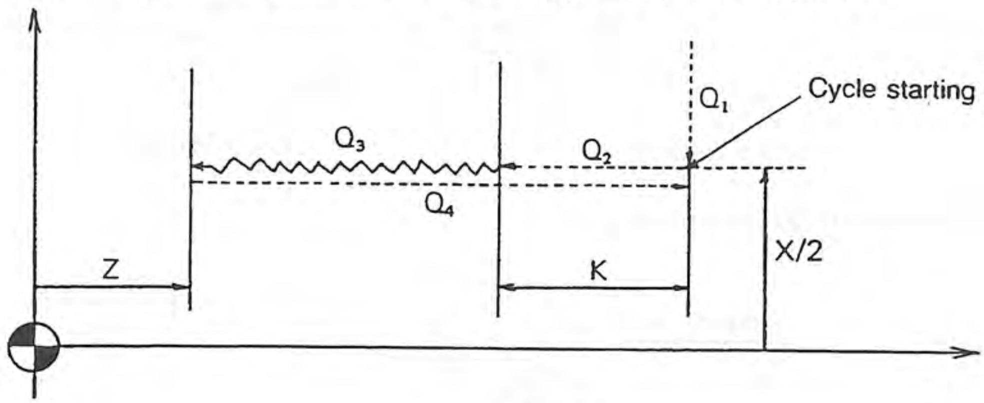


Fig. 17-7

(2) Optional parameter (MULTIPLE MACHINING) Multi cycle return point = 0

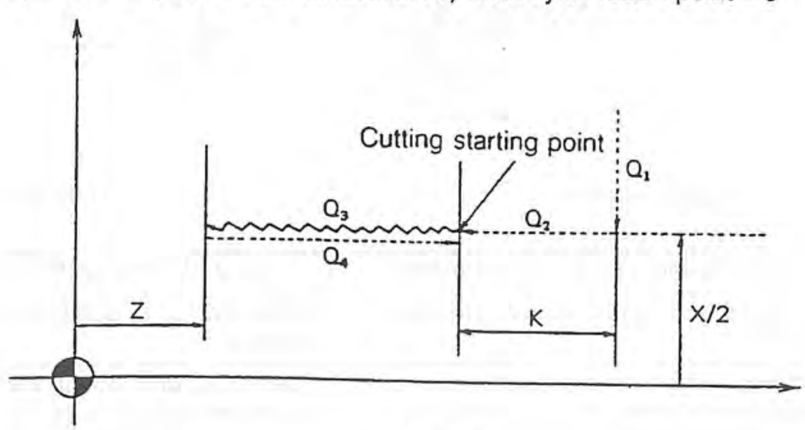


Fig. 17-8

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 joint state, causes an alarm. However, using the M-tool spindle interlock release M code in the optional operation time reduction function allows the rotation of the M-tool even if the C-axis is not in the joint state.

M-tool spindle interlock M code:

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 disregarding of C-axis joint 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

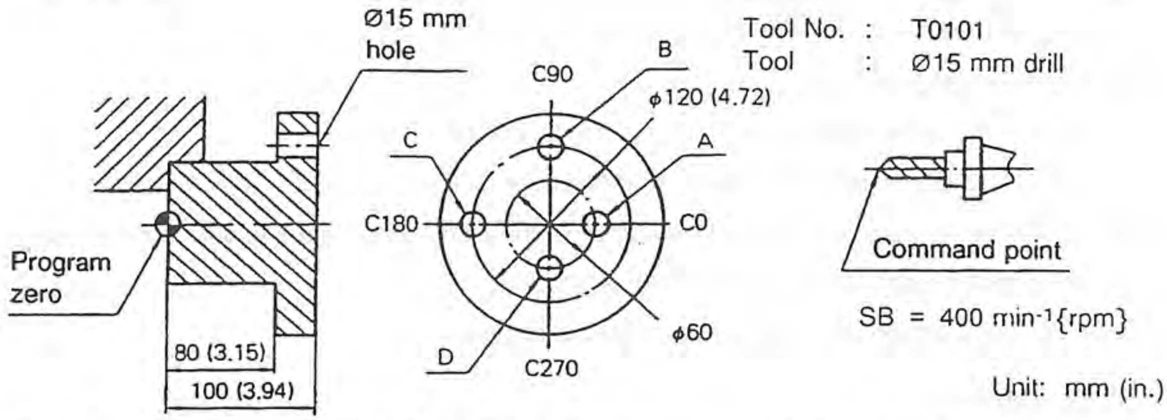
(1) No incremental data can be provided during fixed cycle mode operation.

(2) 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.

(3) If the pattern is repeated by the Q command in a fixed cycle, the Q2 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:



When drilling the four 15 mm dia. holes shown above, program as below using G181 for the drilling cycle.

Continued from turnig operation program							
N099	G00	X1000	Z1000				M05
N100							M110
N101							M15
N102	G94	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		X1000	Z1000				M02

Both X- and Z-axis are programmed.

(1) Tool rotation, and C-axis clamp and unclamp commands in blocks N103 through N106 are unnecessary as they are generated automatically.

(2) In block N104, which calls out drilling cycle on the second hole, program only the commands differing from those provided in the previous block N103. In blocks N105 and N106, the same programming manner applies.

(3) S, M and T codes must not be programmed during fixed cycle mode operation.

Explanation of Example Program

N103 : 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} {rpm}.

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.

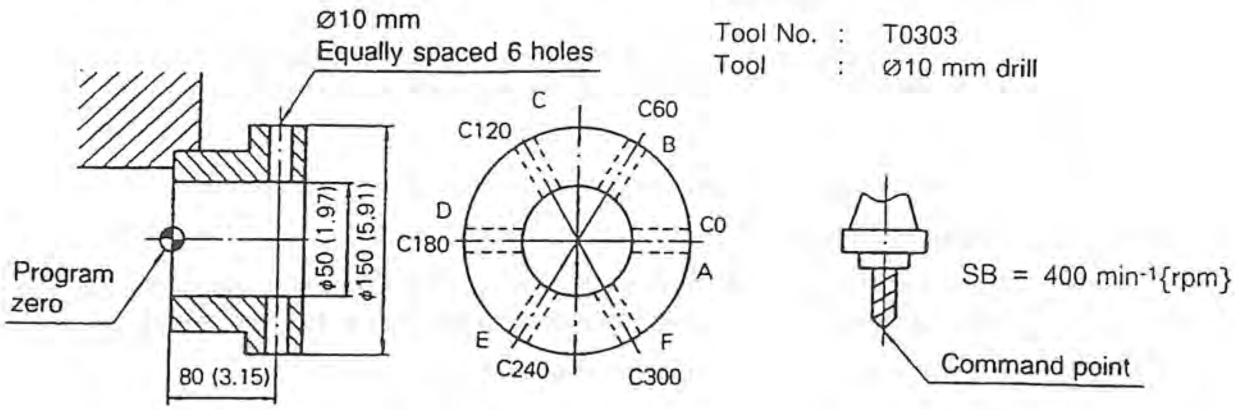
N104 : The spindle indexes to the 90 position. After the drill is positioned at X120 at the rapid feedrate, it drills hole B in the same manner as N103.

N105 : After the spindle indexes to the 180 position, the same drilling cycle is executed for hole C.

N106 : 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.

N107 : G80 cancels the fixed cycle mode.

Example 2:



Unit: mm (in.)

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 turnig operation program									
N099	G00	X00	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									M109
N112									M02

Deep-hole drilling cycle is executed in the peck feed mode.

D word . . . . peck feed stroke (mm)

E word . . . . duration of dwell motion (second)

In the program shown above, peck feed in 10 mm increment (in dia.) is repeated until the programmed depth is reached, where dwell motion is executed for one second.

(1) Tool rotation, and C-axis clamp and unclamp commands in block N103 through N108 are unnecessary as they are generated automatically.

(2) In block N102, which calls out drilling cycle on the second hole, program only the commands differing from those provided in the previous block N103. In blocks N105 and N106, the same programming manner applies.

(3) S, M, and T codes must not be programmed during fixed cycle mode operation.

(4) Value D of peck feed stroke is specified in absolute value and is always preceded by a plus sign.

(5) Peck feed stroke D command is effective for deep-hole drilling cycle. Duration E of dwell motion is effective for deep-hole drilling, boring, and reaming cycles.

Explanation of Example Program

N103 : The spindle indexes to the 0 position. After the drill is positioned at Z80 mm at the rapid feedrate, it starts rotating at 400 min^{-1} {rpm}.

The drill is positioned at X154 at the rapid feedrate.

The drill is fed to X40 at the commanded feedrate 40 mm/min.

The drill is returned to X154 at the rapid feedrate.

This completes drilling of hole A.

N104 : Drilling cycle at point B

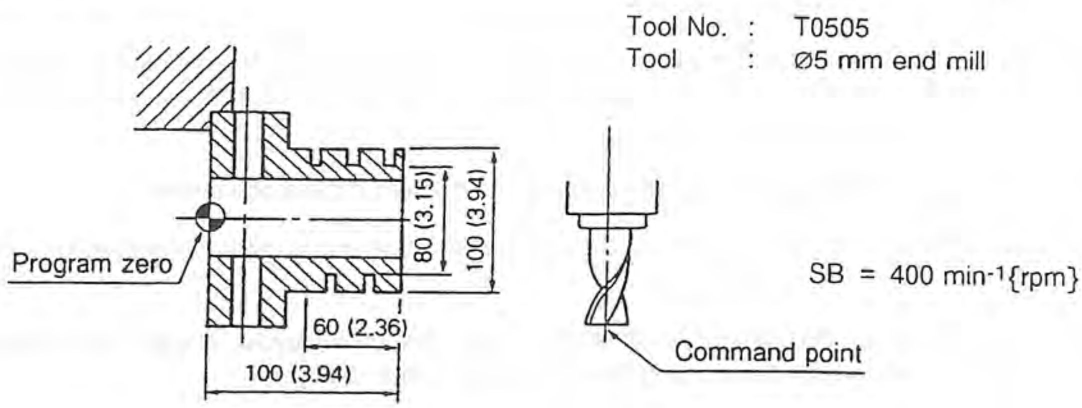
N105 : Drilling cycle at point C

N106 : Drilling cycle at point D

N107 : Drilling cycle at point E

N108 : Drilling cycle at point F

Example 3:



Unit: mm (in.)

When cutting a thread having a width of 5 mm and 60 mm long as shown above, program as below using G185 calling out thread cutting fixed cycle.

Continued from turnig operation program						
N099	G00	X1000	Z1000			M05
N100						M110
N101						M15
N102	G95	X110	Z120			T0101 SB = 400
N103	G185	X95	Z60	C0	F10	SA = 12
N104		X90		C90		
N105		X85		C180		
N106		X80		C270		
N107	G180					
N108	G00	X1000	Z1000			M12
N109	G95					M109
N110						M02

(1) Command SA = 12 in the N103 block specifies the feedrate along the C-axis as 12 min^{-1} {rpm}.

(2) Repeat function is not available for fixed thread cutting cycle G185 through G188.

(3) In the thread cutting fixed cycle called for by G185 through G188, only G95 feedrate mode is selectable. Programming G94 in these modes results in alarm. In this mode, rpm of the C-axis is referred to.

(4) In the thread cutting cycles called by G185 through G188, only constant pitch thread can be cut.

(5) In the G183 mode, an F code specifies thread pitch.

Explanation of Example Program

N103 : 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} {rpm}.

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.

N104 : The spindle indexes to the 0 position. After the end mill is positioned at X90 at the rapid feedrate, thread cutting cycle is executed.

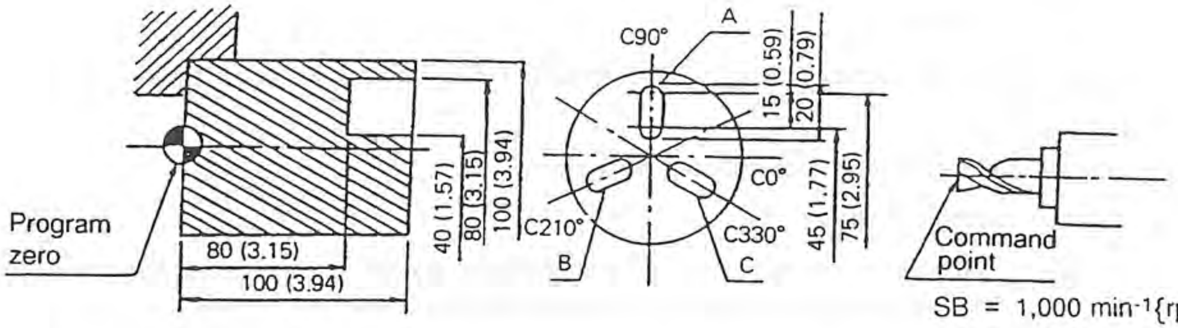
N105 : In the same way, the spindle indexes and thread cutting cycle is executed up to X85.

N106 : In the same way, thread cutting cycle is executed at X80.

N107 : G180 cancels the fixed cycle mode.

Example 4:

Tool No. : T0707
Tool : Ø5 mm end mill



SB = 1,000 min^{-1}{rpm}
Unit: mm (in.)

When cutting a key way having a width of 5 mm and 20 mm long as shown above, program as below using G190 calling out key way cutting fixed cycle.

Continued from turnig 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				X330							
N107	G180										
N108	G00	X1000	Z1000							M12	
N109										M109	
N110										M02	

(1) Tool rotation, and C-axis clamp and unclamp command in blocks N104 through N106 are unnecessary as they are generated automatically.

(2) 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.

(3) S, M, and T codes must not be programmed during fixed cycle mode operation.

Explanation of Example Program

N104 : 1) The spindle is indexed to the 90 position. The end mill is positioned at Z105 mm position at the rapid feedrate and the M-tool spindle starts rotating CCW at 1000 min^{-1} {rpm}.

2) The end mill cuts to Z97 mm position at 15 mm/min.

3) The end mill cuts to X45 mm position at 30 mm/min.

4) The end mill returns to Z105 mm position at the rapid feedrate.

5) The end mill returns to 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 Z-axis reaches 80 mm position.

N105 : Groove B

N106 : Groove C

SECTION 18 LATHE AUTO-PROGRAMMING FUNCTION (LAP4)

1. Overview

LAP (Lathe Auto-Programming) is the function to make full use of high-speed processing capability which characterizes the NC. With this function, the control automatically generates tool path to produce the required part contour.

In this function the program comprising 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 tool path for respective 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; this furthermore facilitates preparatory steps for programming as well program check procedure.

Various cutting modes available with the LAP can cope with any type of cutting intended.

In addition to the features above, LAP4 (*) is also available; with the LAP4, a workpiece can be cut in the most efficient tool paths by simply entering the blank workpiece shape.

Features of LAP are:

(1) No special programming language is needed. The same programming manner as in conventional programming technique can handle the LAP function.

(2) Programming time can be greatly reduced.

(3) Programming for rough cut cycle can be eliminated, and this simplifies manual calculation in programming.

(4) Change of cutting conditions, such as depth of cut and feedrate, is possible during rough cut cycle.

(5) By entering the blank workpiece shape, unnecessary air-cutting tool paths can be eliminated to improve cutting efficiency (LAP4).

*: The LAP4 has been developed by extending the functions of the LAP3.

2. Classification of Functions

2-1. Classification of Cutting Cycle

- 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)
- AP Mode V for bar copy turning (LAP4)

2-2. G Code Used to Designate Cutting Mode (G80, G81, G82, G83)

- G85 : AP Mode I/AP Mode IV

  Used to call out bar rough turning cycle.
- G84 : Change of conditions for bar rough turning
- G86 : AP Mode II/AP Mode V

  Used to call out copy turning cycle.
- G87 : Finish turning cycle

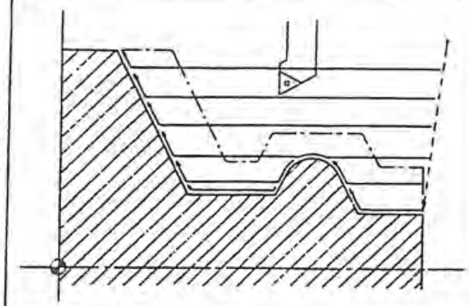
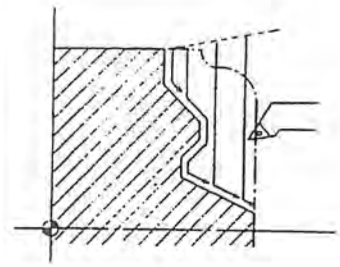
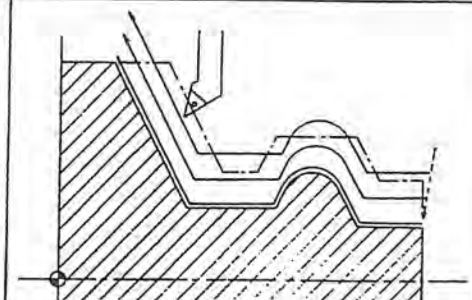
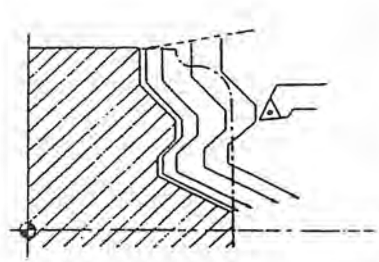
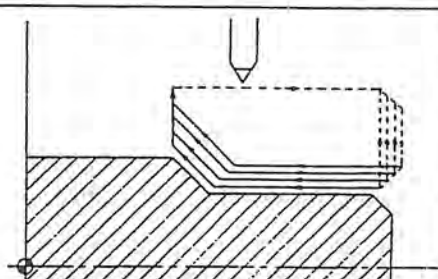
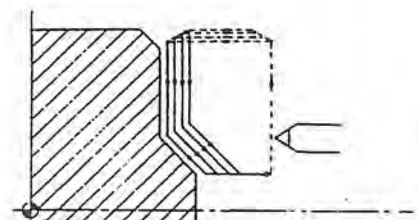
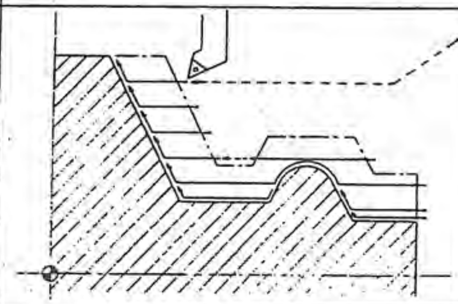
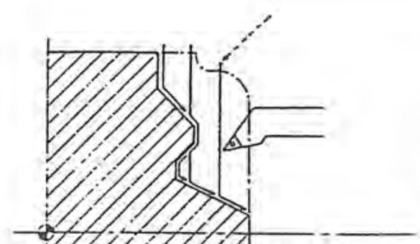
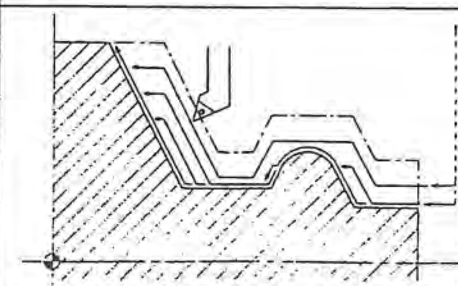
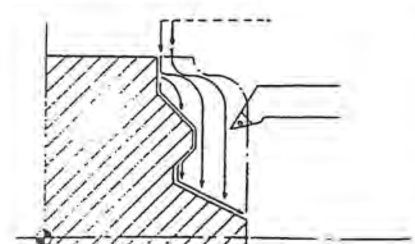
  Used to call out finish turning cycle.
- G88 : AP Mode III

  Used to call out continuous thread cutting cycle.

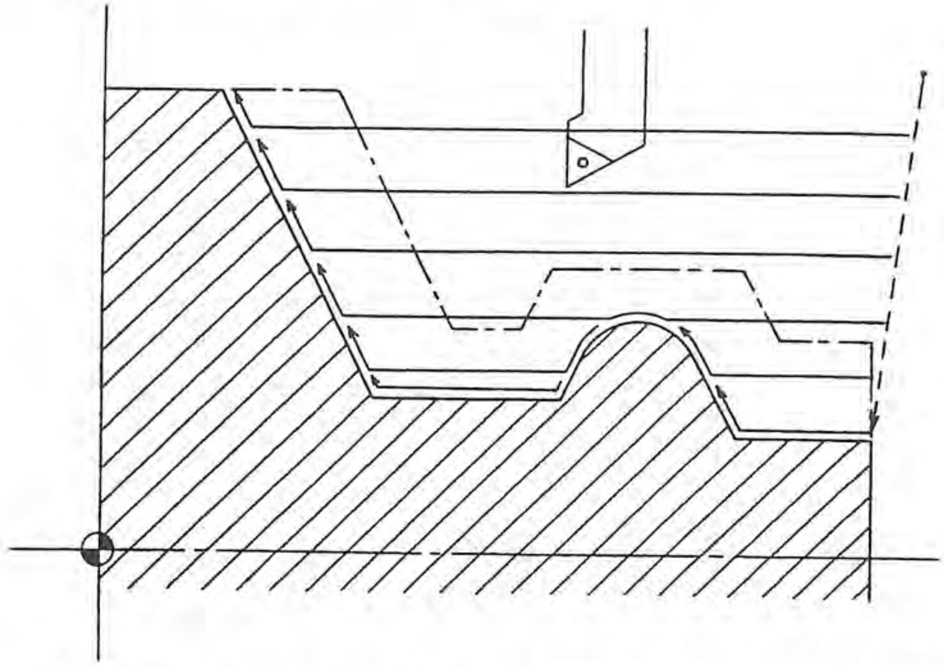
The above indicated G codes calling out AP modes are used in combination with the G codes provided below, which are used to designate the direction of cutting in the AP mode (contour specification):

- G83 : Start of blank shape definition (LAP4)
- G81 : Start of longitudinal contour definition
- G82 : Start of transverse (on end face) contour definition
- G80 : End of contour definition

2-3. List of Cutting Mode

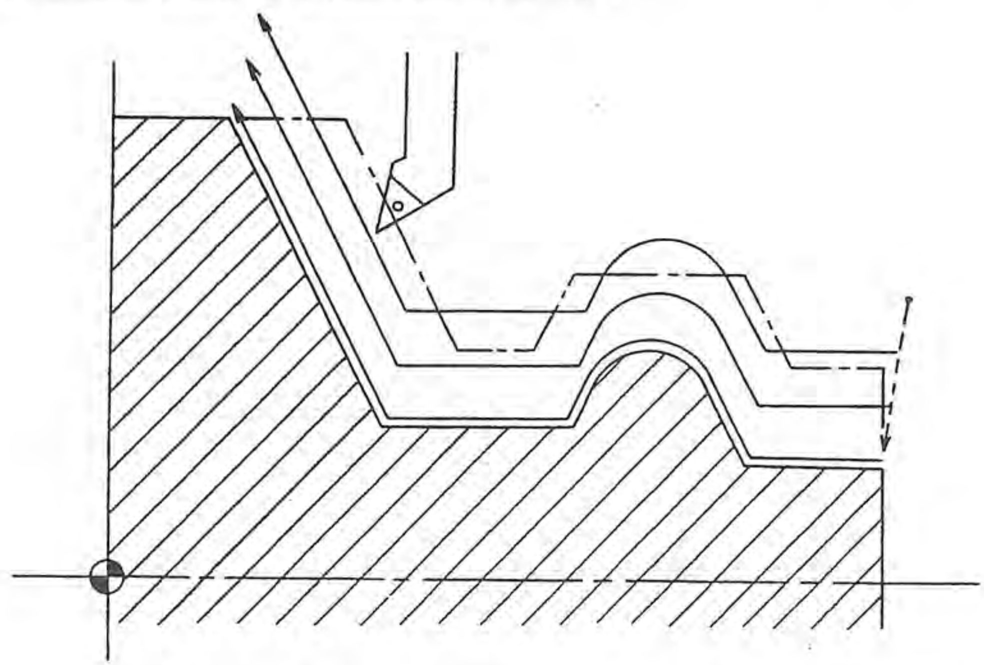
			Longitudinal Cutting Mode	Transverse Cutting Mode
LAP4	LAP3	AP Mode I		
LAP4	LAP3	AP Mode II		
LAP4	LAP3	AP Mode III		
LAP4		AP Mode IV		
LAP4		AP Mode V		

(1) AP Mode I Longitudinal Cutting Mode (G85 + G81 + G80 Mode)



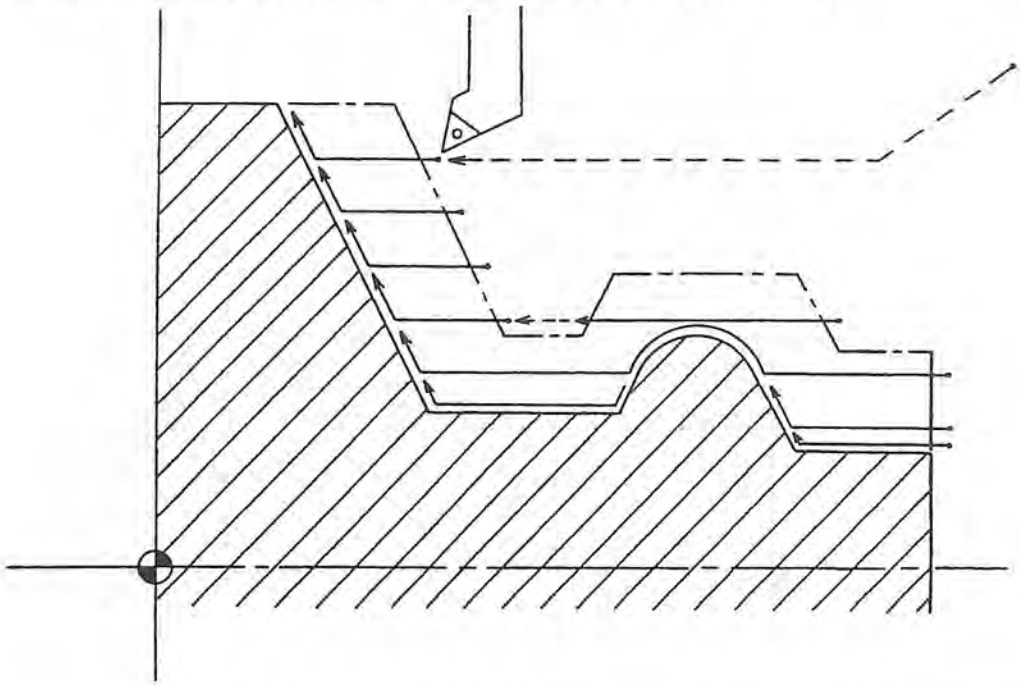
Cutting is executed while shifting the cutting level by the depth of cut. A part program can be created by simply designating the finish contour data.

(2) AP Mode II Longitudinal Cutting Mode (G86 + G81 + G80 Mode)



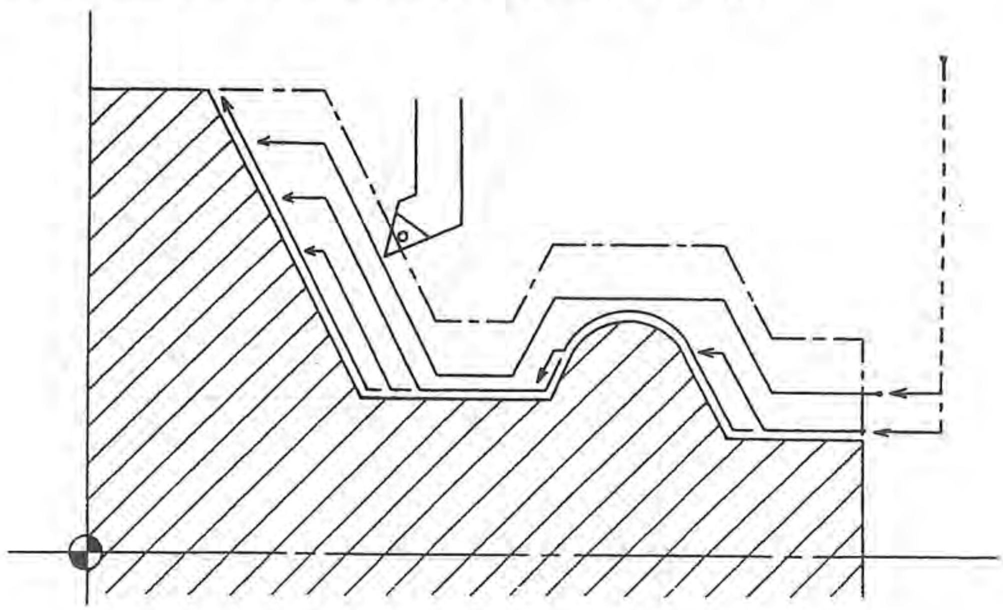
Cutting is executed along the finish contour. The cast-iron workpiece or forged workpiece can be cut at a higher speed than in the AP Mode I since unnecessary tool motion is reduced in this mode.

(3) AP Mode IV Longitudinal Cutting Mode (G85 + G83 + G81 + G80 Mode)



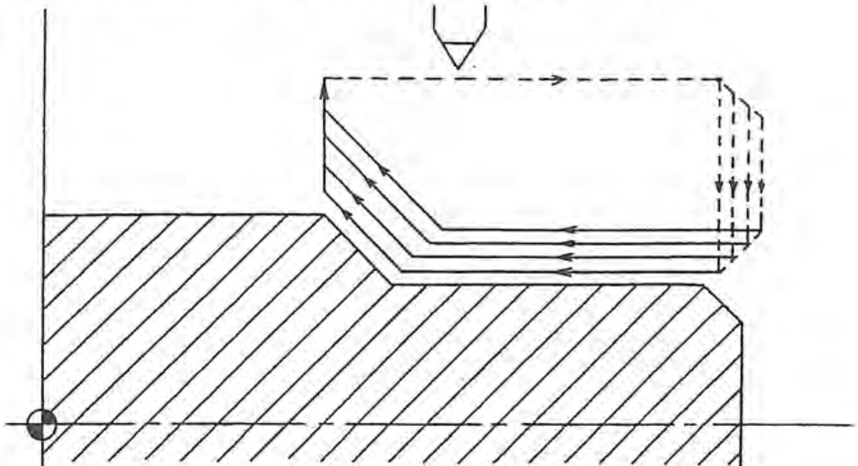
The area between the blank material shape and the finish contour is cut. The cutting tool moves at the rapid feedrate in other area. The time required for cutting is shortest.

(4) AP Mode V Longitudinal Cutting Mode (G86 + G83 + G81 + G80 Mode)

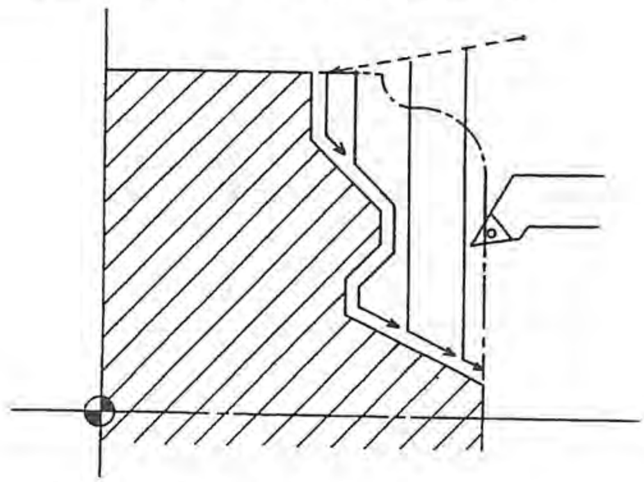


Cutting is carried out along the blank material shape. Since the number of tool collision against the forged workpiece surface is small in this mode, this mode is effective for untended operation in the night in which tool life must be ensured.

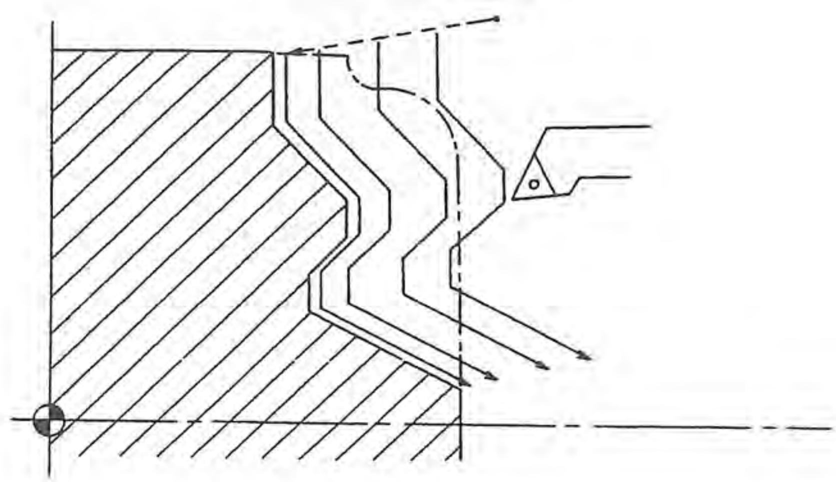
(5) AP Mode III Longitudinal Cutting Mode (G88 + G81 + G80 Mode)



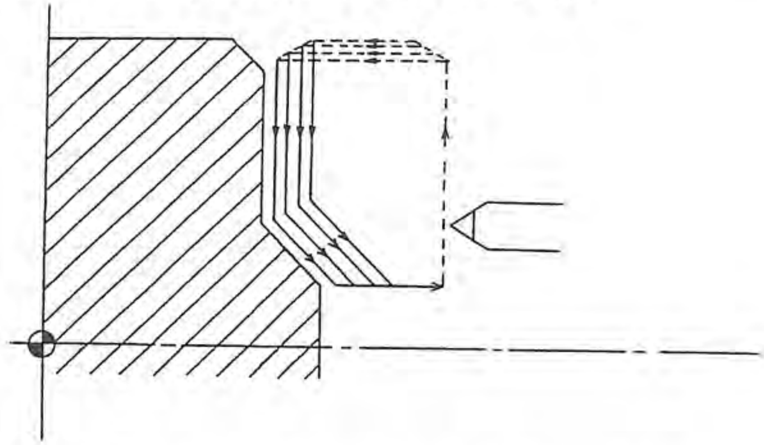
(6) AP Mode I Transverse Cutting Mode (G85 + G82 + G80 Mode)



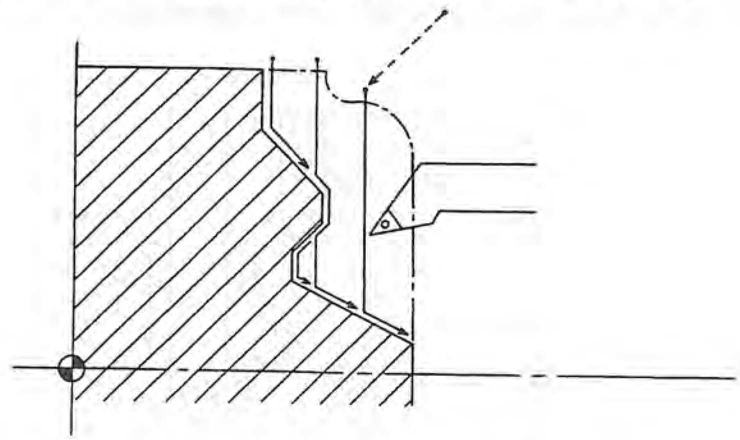
(7) AP Mode II Transverse Cutting Mode (G86 + G82 + G80 Mode)



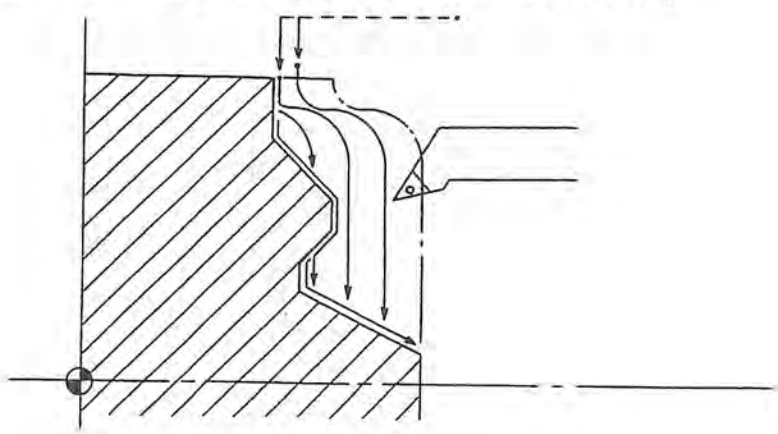
(8) AP Mode III Transverse Cutting Mode (G88 + G82 + G80 Mode)



(9) AP Mode IV Transverse Cutting Mode (G85 + G83 + G82 + G80 Mode)



(10) AP Mode V Transverse Cutting Mode (G86 + G83 + G82 + G80 Mode)



3. Program Format

3-1. 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)
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

3-2. 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)

3-3. Parameters

Parameter	Description	Default	Data Setting Range
D	Depth of cut in rough turning cycle	Alarm	D > 0
DA	Depth of cut after rough turning conditions 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	$\lvert XA \rvert \leqq 99999.999$
XB	X coordinate of rough turning condition change point B	No change of cutting conditions at point B	$\lvert XB \rvert \leqq 99999.999$
ZA	Z coordinate of rough turning condition change point A	No change of cutting conditions	$\lvert ZA \rvert \leqq 99999.999$
ZB	Z coordinate of rough turning condition change point B	No change of cutting conditions at point B	$\lvert ZB \rvert \leqq 99999.999$
U	Stock removal in X-axis direction for finish turning cycle	U = 0	$U \geqq 0$
W	Stock removal in Z-axis direction for finish turning cycle	W = 0	$W \geqq 0$
H	Thread height in G88 thread cutting cycle	Alarm	H > 0
B	Tip point angle of thread cutting tool in G88	B = 0	$0 \leqq B < 180°$

3-4. NC Parameter

Parameter	Contents	Initial Value
Optional parameter (OTHER FUNCTION 1)	Relieving amount in LAP-bar turning (μm)	100
	LAP clearance (LAP 4)	2000
	Infeed pattern in thread cutting cycle	0

[Supplement]

1. The following words must be specified in incremental values.
   D, DA, DB, U, W and H
2. D, DA, DB, XA, XB, U and H words must be commanded in diameter.
3. In thread cutting cycle using the M73 pattern, "H - U" must be greater than or equal to D:
   $H - U \geqq D$
   In the M74 and M75 patterns, it must be positive:
   $H - U \geqq 0$
4. When more than one alphabetic character is used in succession, the control interprets such 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 =

4. Execution Mode of LAP

4-1. Bar Turning Cycle (G85)

Format

N0103 G85 NAT01 D F U W G84

N0103	:	Sequence number
G85	:	G code calling out bar turning cycle To be designated right after sequence number (name).
NAT01	:	Sequence name in the first block of contour defining blocks
Blank	:	Enter either tab or space code.
D	:	Depth of cut in rough turning cycle
F	:	Feedrate in rough turning cycle
U	:	Stock removal in finish turning cycle, X component
W	:	Stock removal in finish turning cycle, Z component
G84	:	Change of rough turning conditions

With the commands above, the control starts searching of contour definition program beginning with the sequence name NAT01. After assigning parameter data of D, F, U, W and G84 for NAT01, the control starts bar turning cycle.

[Supplement]

1. Do not designate S, T, or M code in the G85 block.
2. D word is used to specify depth of cut in rough turning cycle. When the command indicating change of cutting conditions is provided, the D word is effective up to such point, XA and ZA.

   A D word must be designated in the G85 block without fail with a value greater than 0. Illegal designation will cause an alarm.
3. F word is used to specify the feedrate in rough turning cycle. When the command indicating change of cutting conditions is provided, the F word is effective up to such point, XA and ZA.

   If no F word is provided in the G85 block, the feedrate which was effective before the execution of the G85 block is effective.

   F word must be positive. If not, an alarm occurs.
4. When U and/or W word is not designated, U and/or W is assumed "0".

   U and W words must be positive or zero. If not, an alarm occurs.

4-2. Change of Cutting Conditions in Bar Turning Cycle (G84)

Format

N . . .	G85	N		
$	G84	XA = (ZA =)	DA =	FA =
$		XB = (ZB =)	DB =	FB =
(1)		(2)	(3)	(4)

(1) : Indicates that the commands are continuous.
(Must be specified at the beginning of the block.)

(2) : Specifies the point where cutting conditions are changed.

(3) : Depth of cut after cutting condition change point

(4) : Feedrate after cutting condition change point

These commands must be programmed in the block containing G85 calling 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 provided in different lines preceded by "$" which indicates that the commands in these lines belong to the same block.

With these commands, cutting conditions can be changed from the desired point(s) during rough turning cycle from the desired point(s). If change of cutting conditions is not necessary, omit them.

[Supplement]

1. G84 and commands following it must be designated after "N. . . . .G85 N . . . . .".
2. For OD turning, 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 that order.
3. If both the cutting condition change points A and B exist when infeed D is executed, the depth of cut and the feedrate designated for XB = (ZB =) become effective.
4. If tool path exceeds XA when cutting cycle is performed with the depth of cut D from the present position, the cycle is performed with D designated the present position is outside XA, and with DA when the present position is on XA.
5. In longitudinal cutting, ZA = and ZB = commands may not be designated. In transverse cutting, XA = and XB = commands may not be designated, either.

4-3. Copy Turning Cycle (G86)

Format

N0123 G85 NAT02 ␣ D ␣ F ␣ U ␣ W ␣

N0123	:	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

With the commands above, the control starts searching of contour definition program beginning with the sequence name NAT02. After assigning parameter data of D, F, U and W of NAT02, the control starts copy turning cycle.

[Supplement]

1. Do not designated S, T, or M code in the G86 block.
2. D word is used to specify depth of cut in each cycle and must be designated in the G86 block without fail.

   D word value must be positive. If not, an alarm results.
3. 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.

   F word must be positive. If not, an alarm occurs.
4. When U and/or W word is not designated, U and/or W is assumed "0".

   U and W words must be positive or zero. If not, an alarm occurs.

4-4. Finish Turning Cycle (G87)

Format

N0203 G87 NAT03 U W

N0203 : Sequence number

G87 : G code calling out finish turning cycle
To be designated right after sequence number (name).

NAT03 : Sequence name in the first block of contour defining blocks

Blank : Enter either tab or space code.

U : Stock removal in finish turning cycle, X component

W : Stock removal in finish turning cycle, Z component

With the commands above, the control starts searching of contour definition program beginning with the sequence name NAT03. After assigning parameter data of U and W of NAT03, the control starts finish turning cycle.

[Supplement]

1. Do not designate S, T, or M code in the G87 block.
2. As a feedrate, the one provided in the contour definition program is effective.
   If no F word is designated in the contour definition program, the feedrate which was effective before this block becomes effective.
3. When U and/or W word is not designated, in and/or W is assumed "0".
   U and W words must be positive or zero. If not, an alarm occurs.

4-5. Continuous Thread Cutting Cycle (G88)

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
Blank	:	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)	:	Infeed mode

With the commands above, the control starts searching of the contour definition program beginning with sequence name NAT04. After assigning parameter data of D, H, B, U, W, M32 (M33, M34) and M73 (M74, M75) for NAT04, the control starts the thread cycle.

[Supplement]

1. Do not designate S, T, or M code in the G88 block.
2. 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.
3. H word must have a positive value and must be specified in the G88 block without fail. If the numeral data of the D word is not positive, or if it is omitted, an alarm occurs.

   H value must be greater than U and/or W value. If not, an alarm occurs.
4. B word specifying the tip point angle of thread cutting tool must have the value within the following range:

   $0° \leqq B < 180°$

   When no B word is designated, it is assumed "0".
5. M32, M33, and M34 are used to select 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 neither M32, M33 nor M34 is designated, the control selects M32.
6. 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".

   $H - U \geqq D$

   If not, an alarm occurs.

5. Explanation of LAP Functions and Program

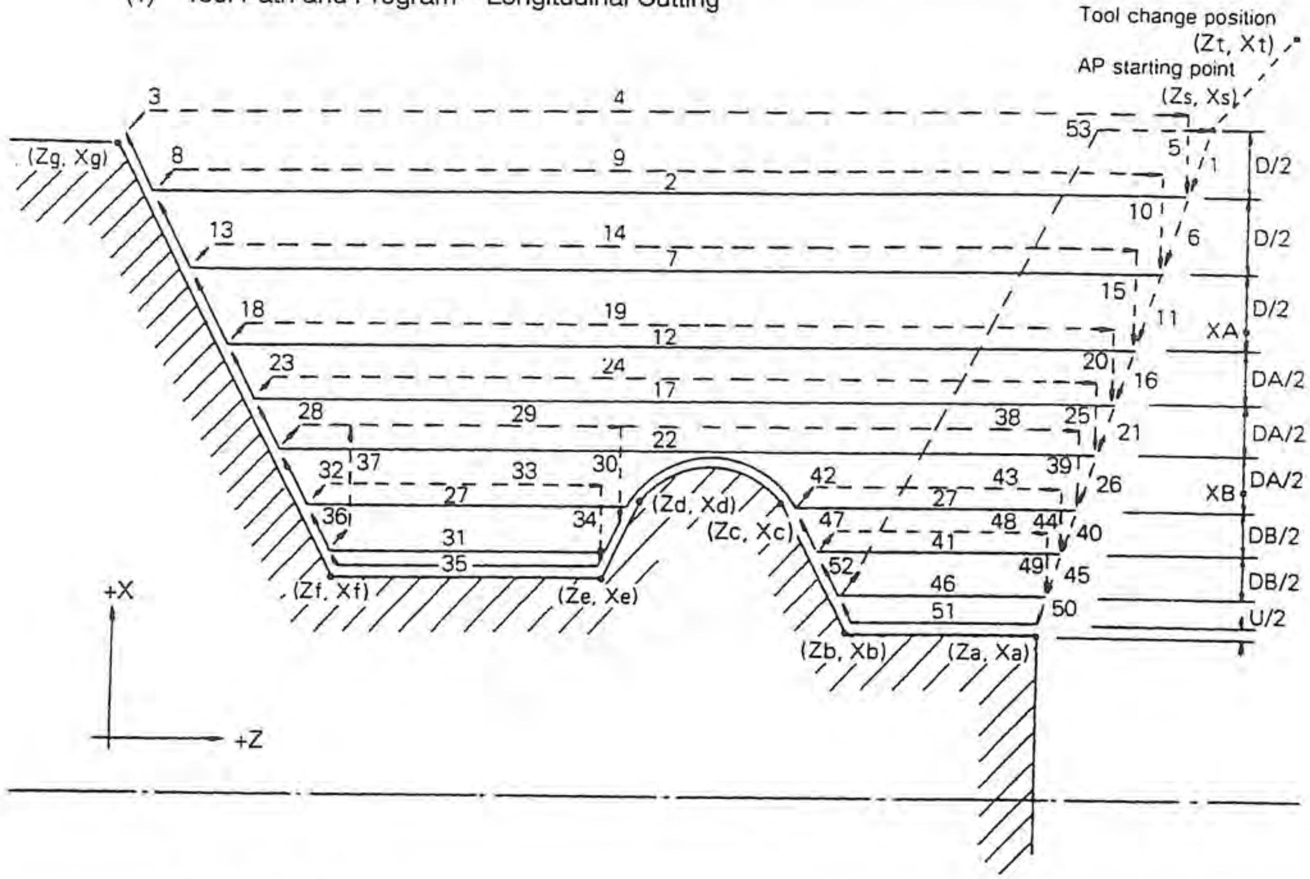
5-1. AP Mode I (Bar Turning)

In the AP Mode I, the area surrounded by the AP starting point 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 in it, the length of program can be reduced.

(1) Tool Path and Program - Longitudinal Cutting



Contour Definition

NAT01	G81								.... Start of longitudinal contour definition G code
N0001	G00	Xa	Za						
N0002	G01	Xb	Zb			Fb	Sb	Eb	
N0003		Xc	Zc			:	:	:	
N0004	G03	Xd	Zd	Id	Kd	Fd	Sd	Ed	Finish contour definition blocks
N0005	G01	Xe	Ze			Fe	Se	Ee	
N0006		Xf	Zf			:	:	:	
N0007		Xg	Zg			Fg	Sg	Sg	
N0008	G80								.... End of contour definition G code

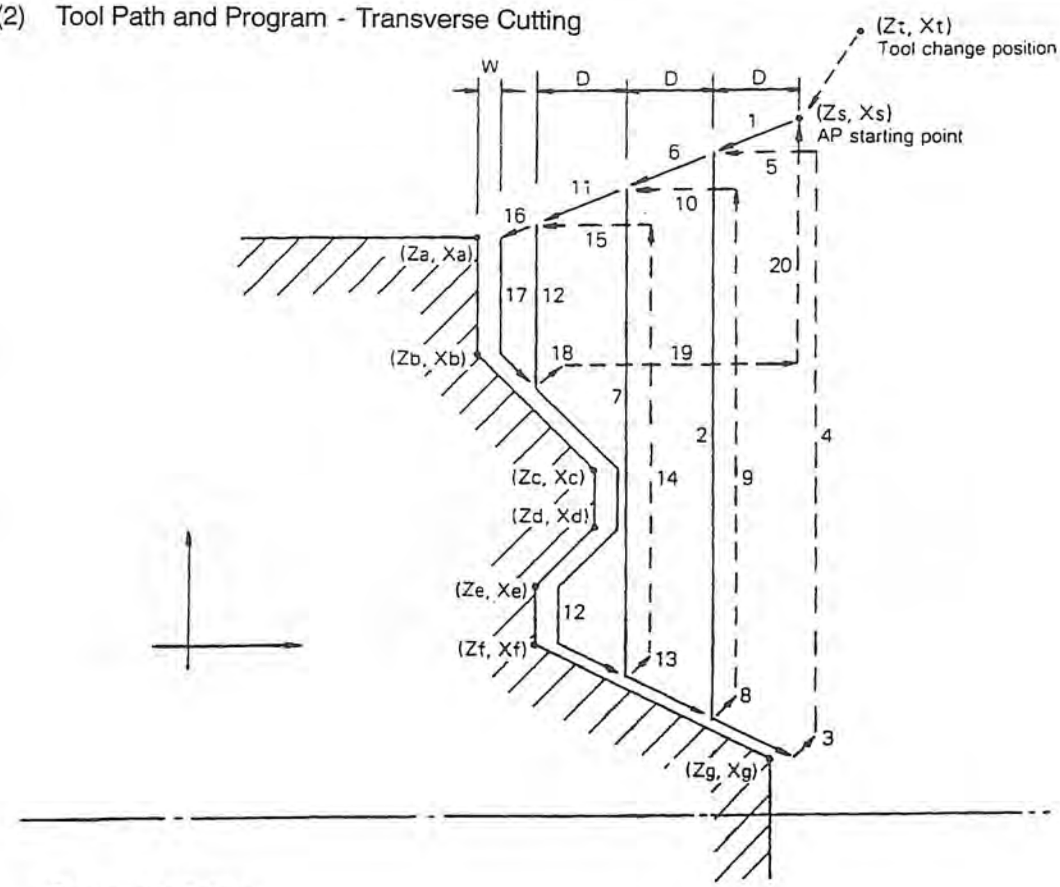
Rough Turning Cycle

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	F	U	W	M85 .	Calls for rough turning cycle
$	G84	XA =	DA =		FA =			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

(2) Tool Path and Program - Transverse Cutting



Contour Definition

NAT10	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	Fg	Eg		
N0018	G80						......	End of contour definition G code

Rough Turning Cycle

N0111	G00	Xt	Zt					Tool change position
N0112		Xs	Zs		STM			Starting point of AP, S, T, and M for rough turning cycle
N0113	G85	NAT10	D	F	U	W	M85 .	Calls for rough turning cycle
$	G84	ZA =	DA =		FA =			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

(3) Outline of Bar Turning Cycle

(a) Rough turning cycle in the longitudinal direction

1) With the commands in block N0101, positioning at the tool change point is performed.

2) With the commands in block N0102, S, T, and M commands for rough turning cycle are selected, and then positioning at the LAP starting point is performed.

When the S, T, or M command is not designated in this block, those selected in the preceding block(s) become effective.

3) With the NAT01 command in block N0103, the control searches the program assigned with the program name NAT01. Rough turning cycle in the bar turning mode is performed on this program.

In the same block, cutting conditions for rough turning cycle are also specified.

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 cutting conditions during 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 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 the line following it continues the preceding line.

When an F word is not designated in this block, the feedrate commanded last becomes effective.

Point data of cutting condition change point(s) must become smaller in the order of the AP starting point, XA and XB when OD turning is intended. In the case of ID turning, they must become larger in that order.

4) Upon reading the commands in block N0001, the control calculates the intersection point of the following two straight lines. The line parallel to the Z-axis running at "Xs - D/2" and the one passing the two points (Xs, Zs) and (Xa + U, Za + W). Positioning at the calculated point A (Xp, Zp) is then performed.

   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.

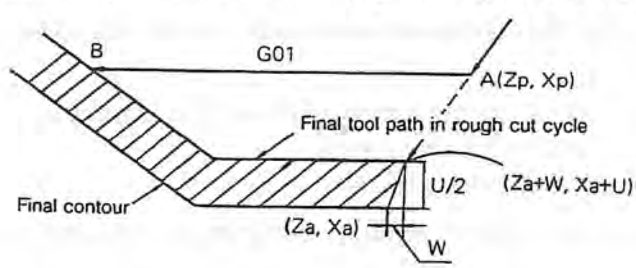
   Select the AP starting point (Xs, Zs) with respect to the coordinated point (Xa, Za) to meet the following requirements:

   Xs < Xa for ID cutting

   Xs > Xa for OD cutting

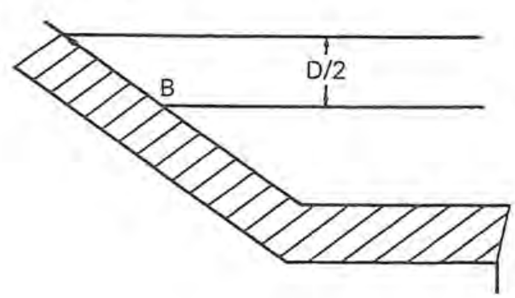
   If too large a finish allowance U is taken to the degree "Xa + U" falls outside "Xs" with respect to the workpiece, it results in an alarm.

5) Cutting is performed in the G01 mode up to point B where the straight line parallel to the Z-axis and passing point A intersects final contour of rough turning cycle. The feedrate in this cutting cycle is as selected by the F word when rough turning cycle is called out.



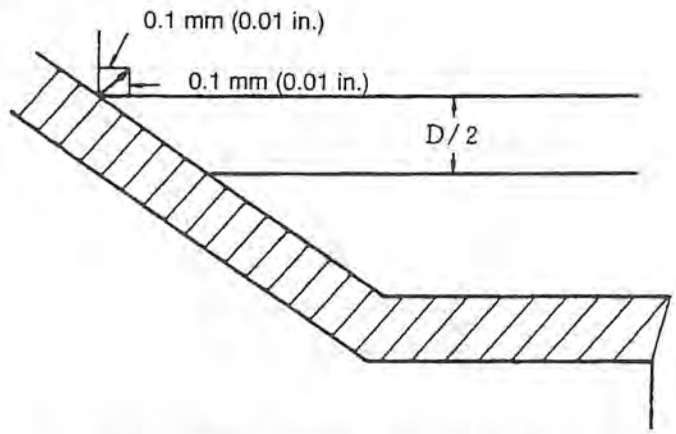
6) After point B has been reached, cutting is performed along the final contour of rough turning cycle up to the point whose X coordinate is Xb + D. If G80 indicating end of contour definition is present before such point is reached, cut along the final rough turning contour is performed up to the point specified in the block preceding the G80 block.

   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 rough turning cycle becomes active.

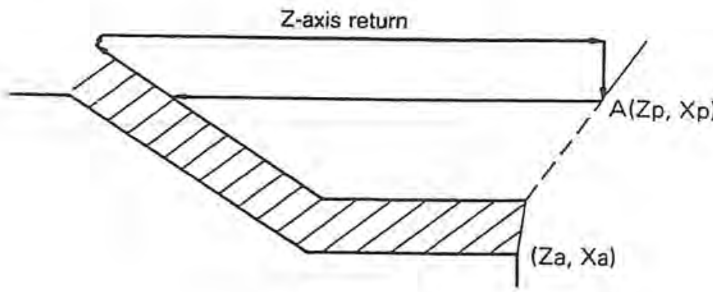


7) After the completion of cutting explained in 6), the cutting tool relieves from the workpiece in the direction opposite to the infeed direction along X-axis and toward Zs along the Z-axis as much as 0.1 mm on the respective axes (in diameter in the case of X-axis).

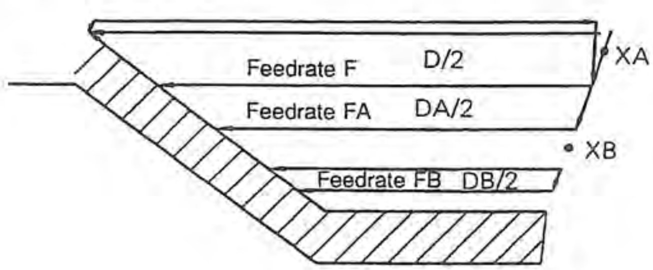
The relieving amount is set at the NC optional parameter (OTHER FUNCTION 1) Relieving amount in LAP-bar turning in units of μm.



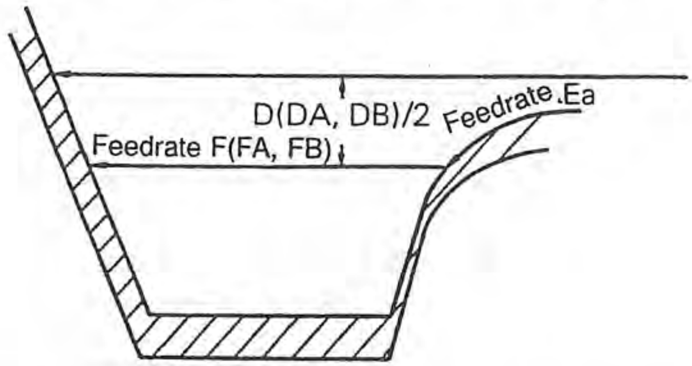
8) This completes the first rough turning cycle. The Z-axis returns to Zp determined in step 4) at the rapid feedrate and then the X-axis to Xp.



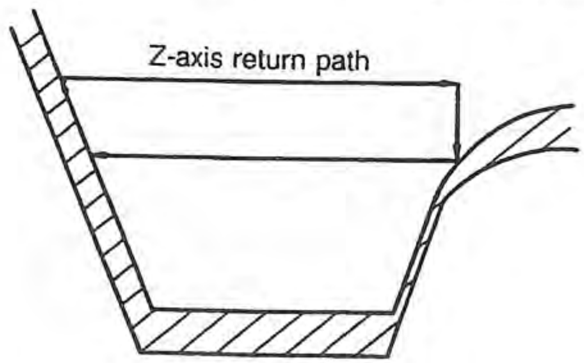
9) The steps 4) through 8) are repeated up to the cutting condition change point. After that point, cutting is continued with the depth of cut (D) and feedrate (F) changed.



10) If cutting along a descending slope is performed in step 6), and when contour to be cut exists below the point of cutting (Xp), cutting is first performed along the contour until the programmed depth of cut is reached and then it is performed in parallel with the Z-axis up to the point where the straight line passing that point intersects the final rough turning contour. Cutting along the parallel line is performed at the feedrate specified by an F word (FA/FB).

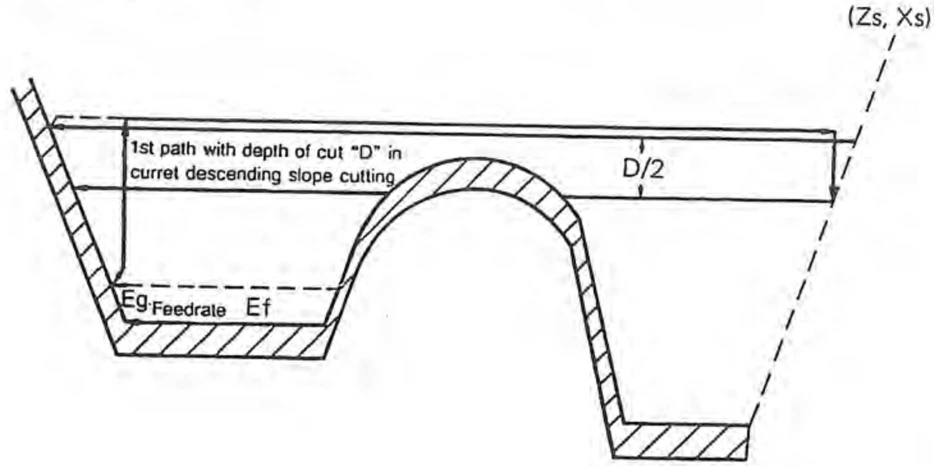


11) Steps 6) and 7) are repeated after that. The Z-axis then returns to the point where cutting along the Z-axis is started in step 10). After the completion of positioning of the Z-axis, the X-axis is positioned at the point where previous cutting cycle has been started.



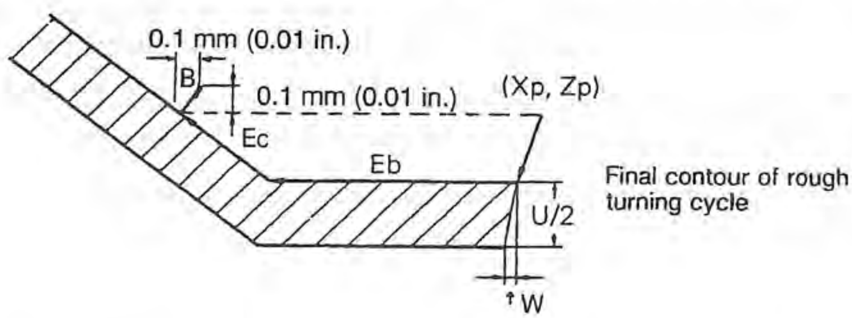
12) Steps 10) and 11) are repeated until the most recessed section along X-axis is cut. After the most recessed section has been cut, both the X- and Z-axis retracts by 0.1 mm (in radius for the X-axis), and the X-axis is positioned at the point having coordinate value "the first cutting level along the descending slope ± (D + 0.2)" mm. The Z-axis returns to the point which has the coordinate value as that of the point from which the cutting cycle of the descending slope has started with depth of cut D. Positioning of the X-axis at that point is then executed.

After the completion of descending slope cutting, cutting before starting it is resumed and steps after 4) are repeated.



13) The steps indicated before are repeated until the X-axis reaches the level where 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 turning contour is the one specified by the E word.



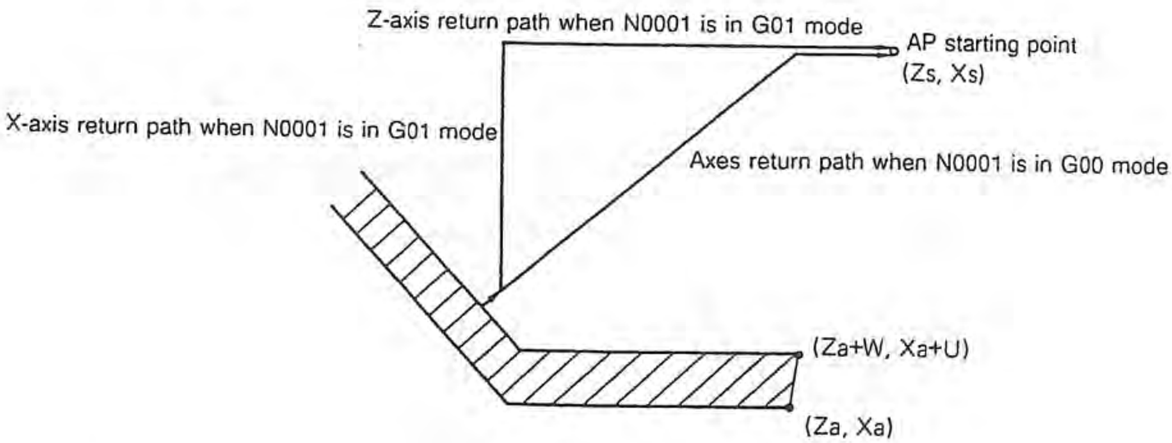
After the completion of the final rough turning step, the X- and Z-axes relieve by 0.1 mm (in diameter for the X-axis). The relieving amount is set at the NC optional parameter (OTHER FUNCTION 1) Relieving amount in LAP-bar turning.

14) At the completion of step 13), the axes return to the AP starting point (Xs, Zs).

There are two patterns of axis return motion:

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 first made and then the Z-axis returns to the AP starting point when G01 is designated in the above indicated block.



This completes rough turning cycle.

(b) Finish in bar turning cycle

1) With the commands in block N0201, positioning at the tool change position is performed.

2) With the commands in block N0202, S, T, and M commands for finish turning cycle are selected.

3) With the NAT01 command in block N0203, the control searches the program assigned with the program name NAT01. Finish bar turning cycle is performed on this program.

4) The finish turning cycle is performed following the data designated in the contour definition program under the cutting conditions specified for the finish turning cycle.

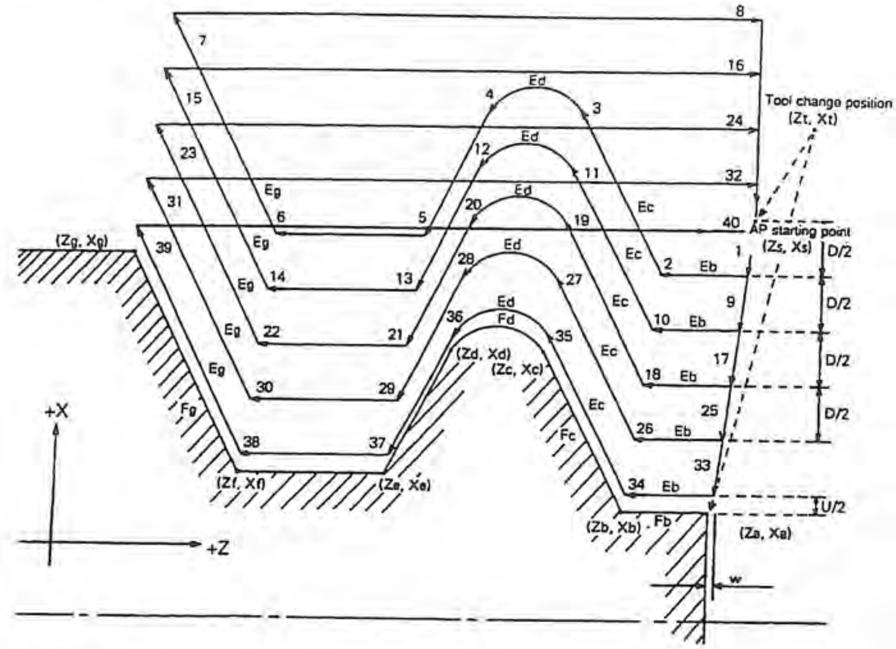
5) After the finish turning cycle is completed, the commands in the block following N0203 are executed.

5-2. AP Mode II (Copy Turning)

In the AP Mode II, the finish contour designated by the contour definition program is shifted parallel, up to the AP starting point. Cutting is executed along the shifted finish contour while shifting the cutting level by depth of cut D.

When this mode is used for cutting the workpiece whose cutting depth is constant, for example the cast-iron workpiece or forged workpiece, it can be cut at a higher speed than in the AP Mode I.

(1) Tool Path and Program - Longitudinal Cutting



Contour Definition

NAT20	G81								...... Start of longitudinal contour definition G code
N0021	G01	Xa	Za						
N0022		Xb	Zb			Fb	Sb	Eb	Finish contour definition blocks
N0023		Xc	Zc			:	:	:	
N0024	G03	Xd	Zd	Id	Kd	Fd	Sd	Ed	
N0025	G01	Xe	Ze			Fe	Se	Ee	
N0026		Xf	Zf			:	:	:	
N0027		Xg	Zg			Fg	Fg	Eg	
N0028	G80								...... End of contour definition G code

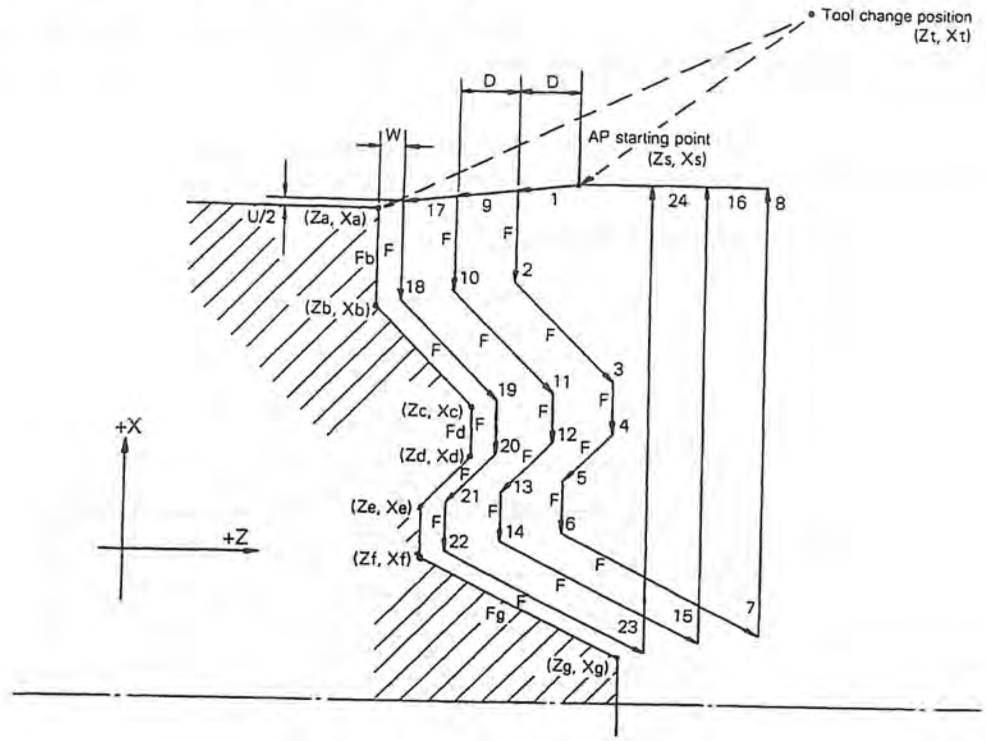
Rough Turning Cycle

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

(2) Tool Path and Program - Transverse Cutting



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	G85	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

(3) Outline of Copy Turning Cycle

(a) Rough turning cycle in the longitudinal direction

1) With the commands in block N0121, positioning at the tool change position is performed.

2) With the commands in block N0122, S, T, and M commands for rough turning cycle are selected, and then positioning at the AP starting point is performed.

When the S, T, or M command is not specified in this block, those selected in the preceding block(s) become effective.

3) With the NAT20 command in block N0123, the control searches the program assigned with the program name NAT20. Rough turning cycle in the copy turning mode is performed on this program.

In the same block, cutting conditions for rough turning cycle are also specified:

D : Depth of cut
U : X component of stock removal in finish turning cycle
W: Z component of stock removal in finish turning cycle

Program also F word if required. When no F word is designated in the contour definition program, the feedrate commanded last becomes effective.

4) Upon reading the commands in block N0201 in the contour definition program, the control calculates the point of intersection of the following two straight lines. The line parallel to the Z-axis running at "Xs - D/2" and the one passing the two points (Xs, Zs) and (Xa + U, Za + W). Positioning at the calculated point A (Xp, Zp) is then performed.

Along with the positioning, the control calculates the distance (XOFF, ZOFF) between these two points (Xp, Zp) and (Xa + U, Za + W).

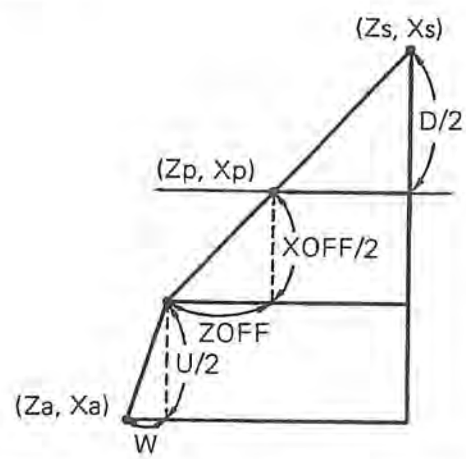
$Xp = Xs - D$

$Zp = Za + W + (Zs - Za - W) (1 - D/(Xs - Xa - U))$

$XOFF = Xp - (Xa + U)$

$ZOFF = Zp - (Za + W)$

See the illustration below:

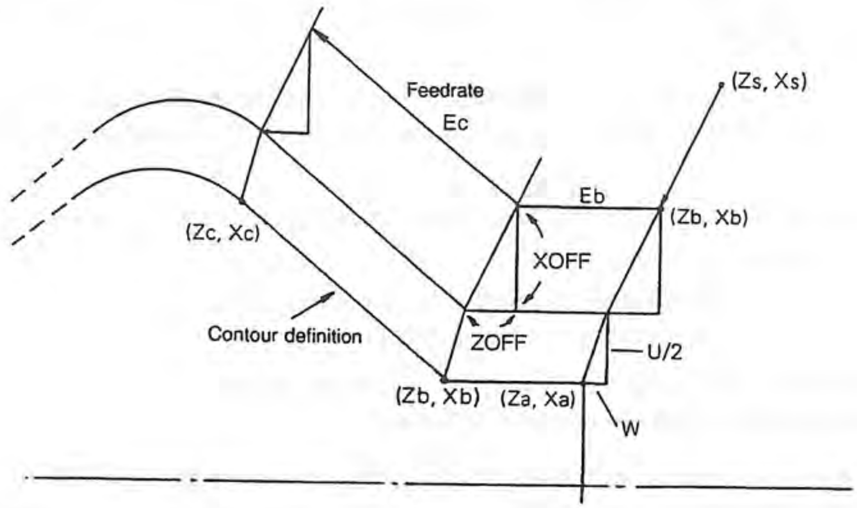


If the valve for U or W is too large and the infeed direction is reversed, an alarm occurs.

5) Cutting is started from (Xp, Zp) to the target point (*1) calculated by the OSP.

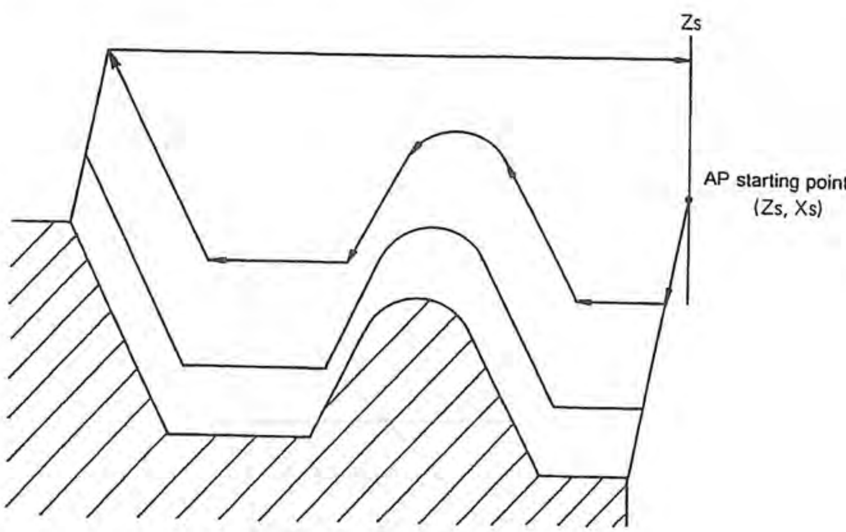
Target point (*1): The point obtained by offsetting the points commanded in the contour definition program in parallel with respective axes as much as (XOFF + U, ZOFF + W).

Cutting is performed at the feedrate specified by an E word in each block of the contour defining blocks.



6) Step 5) is repeated until contour definition ends (G80 active).

The Z-axis then returns to the AP starting point coordinate, Zs.



7) This completes one rough cutting cycle. New XOFF and ZOFF are calculated and steps 4) through 6) are repeated.

The positions for the Nth cycle are calculated as follows.

Longitudinal direction:

$$Xp = Xs - N*D$$
$$Zp = Za + W + (Zs - Za - W)\ (1 - N*D/(Xs - Xa - U))$$

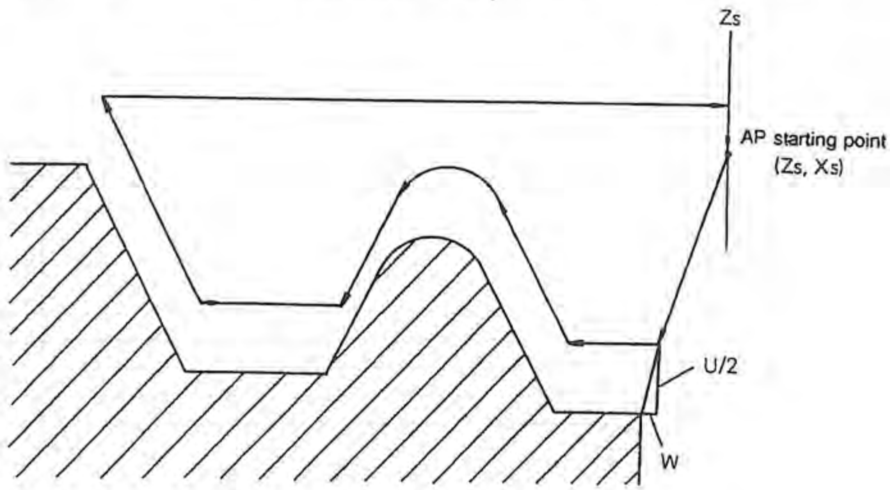
Transverse direction:

$$Xp = Xa + U + (Xs - Xa - U)\ (1 - N*D/(Zs - Za - W))$$
$$Zp = Zs - N*D$$

$$XOFF = Xp - (Xa + U)$$
$$ZOFF = Zp - (Za + W)$$

8) The above indicated steps are repeated until the infeed point reaches or exceeds "Xa + U". At such point, the control takes (0, 0) for (XOFF, ZOFF) and performs cutting along the 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, and then the X-axis moves to the AP starting point.



9) This completes rough turning cycle and commands in the block following N0123 are executed.

(b) Finish cut cycle

1) With the command in block N0221, positioning at the tool change position is performed.

2) With the commands in block N0222, S, T, and M commands for finish turning cycle are selected.

3) With the NAT20 command in block N0223, the control searches the program assigned with the program name NAT20. Finish bar turning cycle is performed on this program.

4) The finish turning cycle is performed following 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.

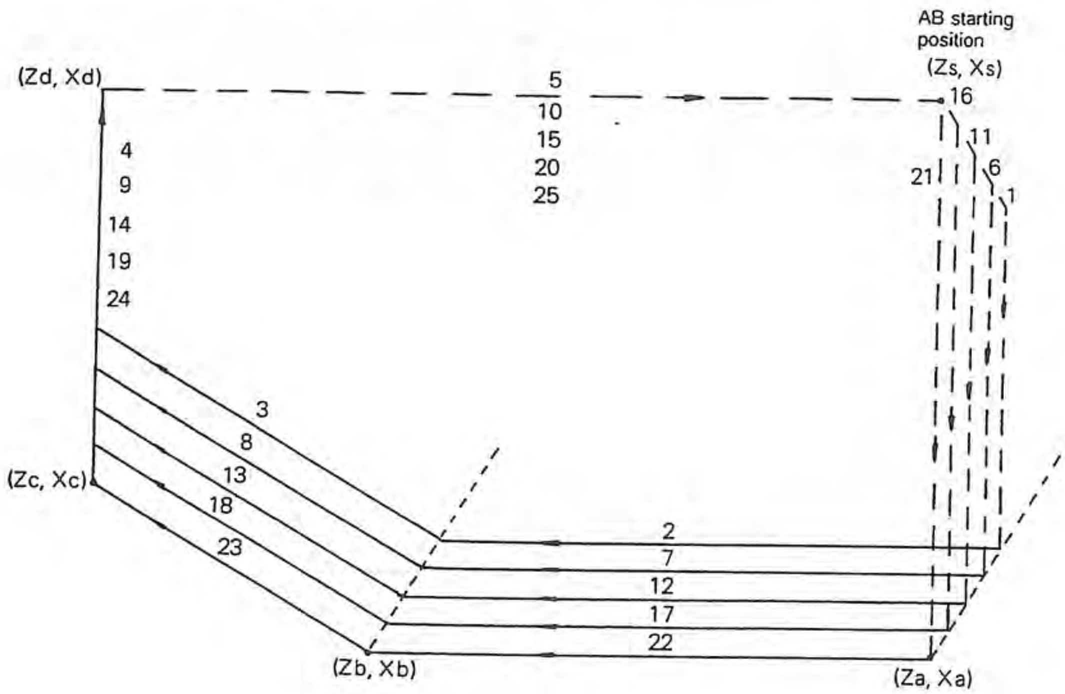
5-3. AP Mode III (Continuous Thread Cutting Cycle)

In the AP Mode III, thread cutting is executed along the contour which is designated by the contour definition program starting 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.

(1) Tool Path and Program - Longitudinal Cutting

G34, G35, G112 and G113 Mode (G112 and G113 can not be designated unless the optional circular thread cutting function is selected.)



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

(2) Outline of Continuous Thread Cutting Cycle

(a) With the commands in block N0141, S, T, and M commands for thread cutting are selected.

(b) Positioning at the AP starting point (Xs, Zs) is carried out with the commands in block N0142.

(c) B, H, D, and U words in block N0143 specify the data necessary for 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 contour definition program and executes the required thread cutting cycle (AP Mode III).

For details of thread cutting cycle, refer to SECTION 17, 4-3."M Code Specifying Thread Cutting Mode and Infeed Pattern"

[Supplement] For thread cutting on the 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 U word.

5-4. AP Mode IV (High-speed Bar Turning Cycle)

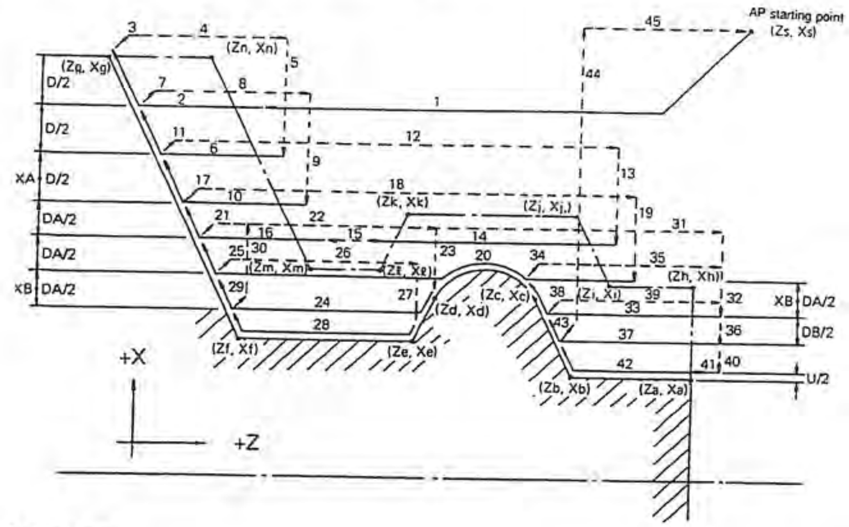
In the AP Mode IV the blank material shape data is input in addition to the finish contour shape data.

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 judges whether or not the blank material resides, and the cutting tool is fed at the rapid feedrate in the area where the blank material does not reside and cutting is not required. This eliminates cutting feed in the unnecessary-to-cut area, which occurs in the AP Mode I , and allows high-speed cutting.

(1) Tool Path and Program - Longitudinal Cutting



Contour Definition

```
NAT60  G83 ................................. a) Blank material shape definition start G
                                                code
N0601  G01  Xh   Zh                         ┐
N0602       Xi   Zi                         │
N0603       Xj   Zj                         │
N0604       Xk   Zk                         │ b) Blank material shape definition blocks
N0605       Xl   Zl                         │
N0606       Xm   Zm                         │
N0607       Xn   Zn                         ┘
N0608  G81 ..................................... c) Finish contour definition start G code
N0609  G01  Xa   Za
N0610       Xb   Zb              Fb   Sb   Eb ┐
N0611       Xc   Zc              :    :    :  │
N0612  G03  Xd   Zd   Ib   Kd    :    :    :  │ d) Finish contour definition blocks
N0613  G01  Xe   Ze              :    :    :  │
N0614       Xf   Zf              Ff   Sf   Ef │
N0615       Xg   Zg              Fg   Sg   Eg ┘
N0616  G80 ................................. e) Contour definition end G code
```

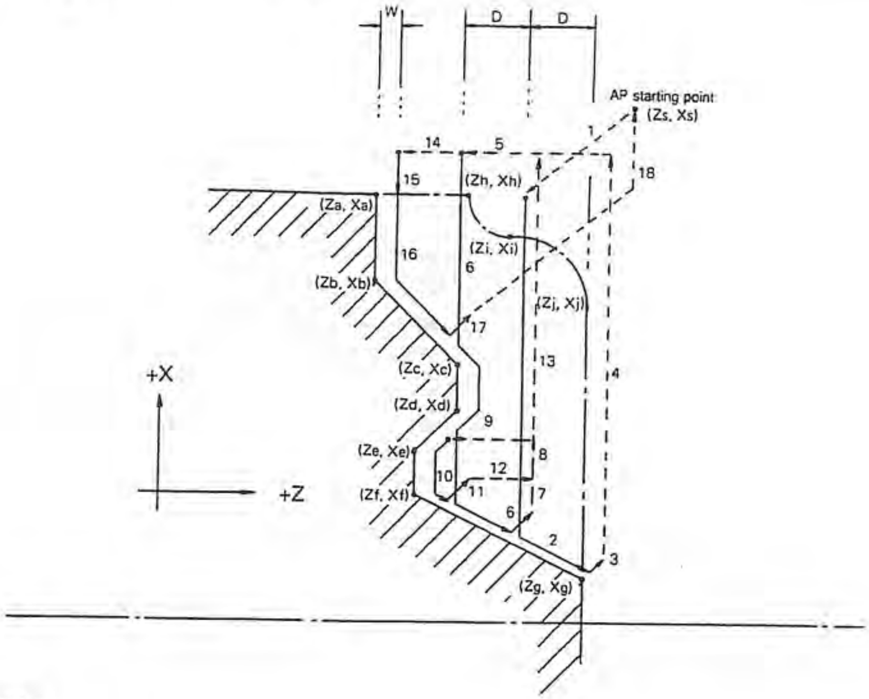
Rough Turning Cycle

```
N0161  G00  Xt   Zt                             Tool change position
N0162       X    Z        STM                   Starting point of AP, S, T, and M for rough
            s    s                              turning cycle
N0163  G85  NAT60  D    F    U    W    M85 ┐
$      G84   XA =  DA =      FA =          │ f) Calls for rough turning cycle
$            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 .......................... g) Calls for finish turning cycle
```

(2) Tool Path and Program - Transverse Cutting



Contour Definition

Program										Description
NAT70	G83									a) Blank material shape definition start G code
N0701	G01	Xh	Zh							b) Blank material shape definition blocks
N0702	G03	Xi	Zi	Ii	Ki					
N0703	G02	Xj	Zj	Ij	Kj					
N0704	G82									c) Finish contour definition start G code
N0705	G00	Xa	Za							
N0706	G01	Xb	Zb				Fb	Sb	Eb	d) Finish contour definition blocks
N0707		Xc	Zc				:	:	:	
N0708		Xd	Zd				:	:	:	
N0709		Xe	Ze				:	:	:	
N0710		Xf	Zf				Ff	Sf	Ef	
N0711		Xg	Zg				Fg	Sg	Eg	
N0712	G80									e) Contour definition end G code

Rough Turning Cycle

Program								Description
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		f) Calls for rough turning cycle
$	G84	ZA =	DA =	FA =				
$		XB =	DB =	FB =				

Finish Turning Cycle

Program					Description
N0271	G00	Xt	Zt		Tool change position
N0272				STM	S, T, and M for finish turning cycle
N0273	G87	N0704			g) Calls for finish turning cycle

(a) Blank material shape definition start G code

- 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.

(b) 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.

(c) 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

(d) 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.
- 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.
- The G00 code can be used only in the first block.

(e) Contour definition end G code

- This code declares the end of contour definition.

(f) Calls for rough turning cycle

- 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)
- When the finish contour definition blocks start with G81 or G82, the AP Mode I (bar turning cycle) is selected.

(g) Calls for finish turning cycle

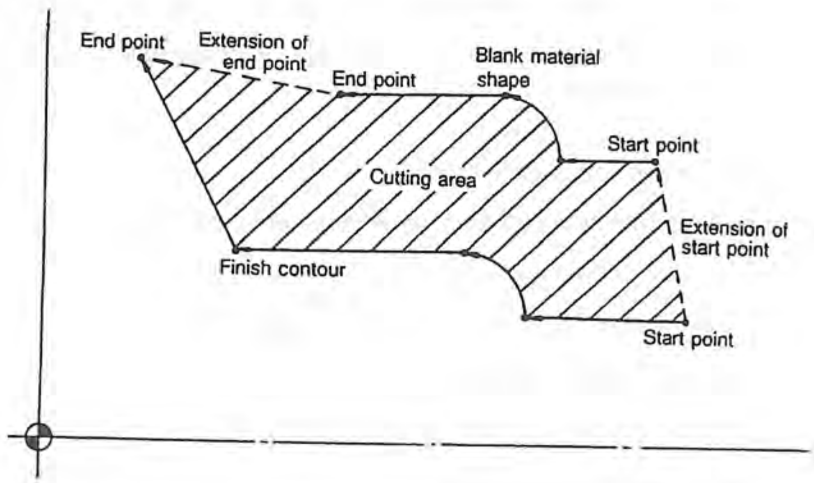
- Finish turning cycle is carried out by designating G87 and calling for the finish contour definition blocks starting with G81 or G82.

NOTICE : (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.



(3) Outline of High-speed Bar Turning Cycle

(a) Rough turning cycle in longitudinal direction

1) With the commands in block N0161, positioning at the tool change point is performed.

2) With the commands in block N0162, S, T and M commands for rough turning cycle are selected, and then positioning at the LAP starting point is performed.

When the S, T, or M command is not designated in this block, those selected in the preceding block(s) become effective.

3) With the NAT60 command in block N0163, the control searches the program assigned with the program name NAT60. Rough turning cycle in the bar turning mode is performed on this program.

When NAT60 is designated in the block starting with G83, high-speed bar turning cycle (LAP4) is carried out.

In the same block, cutting conditions for rough turning cycle are also specified.

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 retracting motion 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 cutting conditions during 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 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 the line following it continues the preceding line.

When an F word is not designated in this block, the feedrate commanded last becomes effective.

Point data of cutting condition change point(s) must become smaller in the order of the AP starting point (Xs), XA and XB when OD turning is intended. In the case of Ib turning, they must become larger in that 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.

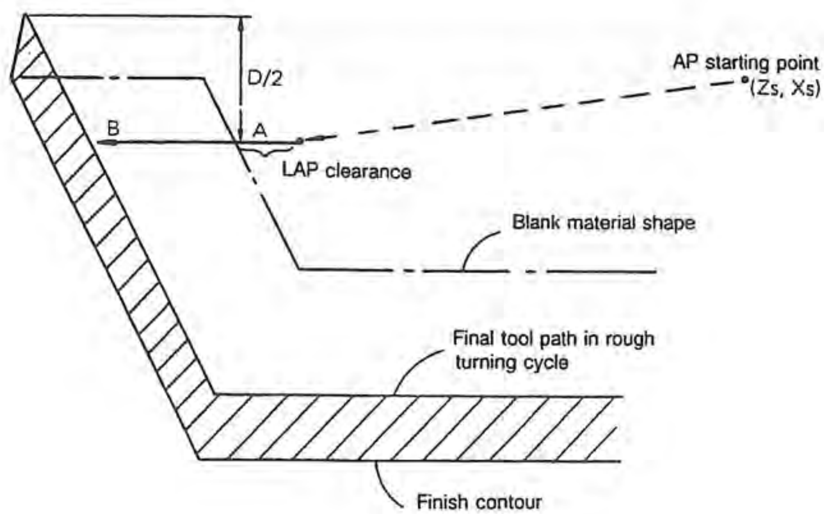
   In the case of OD turning, draw the perpendicular from the point which is obtained by shifting the point on the maximum OD of the blank material shape or final rough turning contour, whichever larger, and obtain point of intersection A where this perpendicular intersects the blank material shape.

   In the case of Ib turning, draw the perpendicular from the point which is obtained by shifting the point on the minimum Ib of the blank material shape or final rough turning contour, whichever smaller.

   The cutting tool is positioned at the point which is away 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 it is performed at a cutting feedrate when G01 is designated in the first block of the finish contour definition blocks.

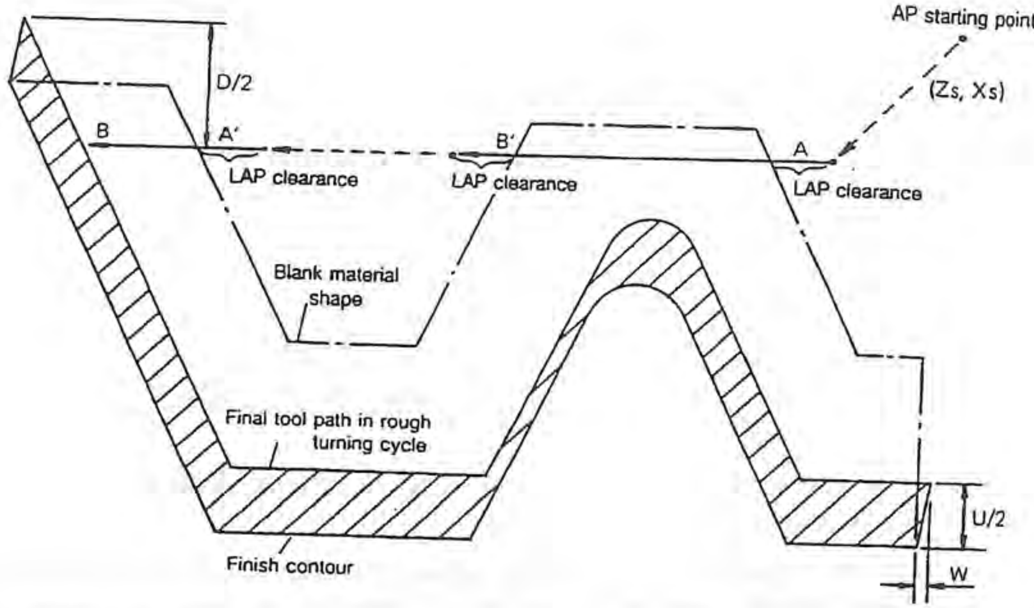
   - The LAP clearance amount (Lc) is set at the optional parameter (OTHER FUNCTION 1) in units of μm.
   - For the relationship between the LAP clearance amount and the AP starting point, refer to (4), b) "How to obtain the infeed starting point".
   - An alarm occurs if the G02 or G03 code is specified 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 point A intersects with the final contour of rough turning cycle. The feedrate in this cutting cycle is as selected by the F word when rough turning cycle is called out.



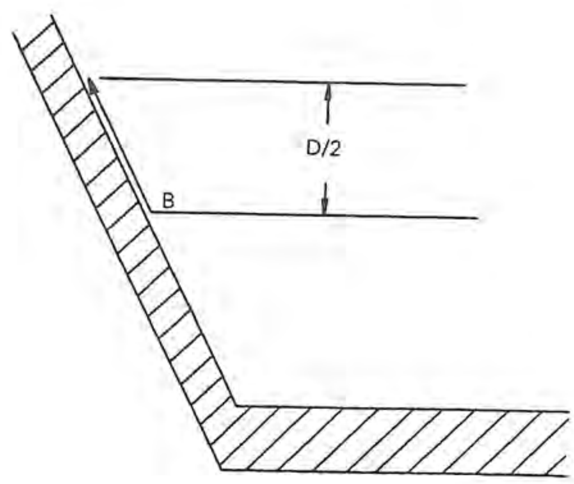
When the straight line intersects the blank material shape at point B' before it intersects the blank material at point B, cutting is executed in the G01 mode up to the point which is away from point B' by the LAP clearance amount (Lc) in the Z-axis direction, and after that the cutting tool is fed in 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 which is away from point A' by the LAP clearance amount (Lc) in the Z-axis direction.



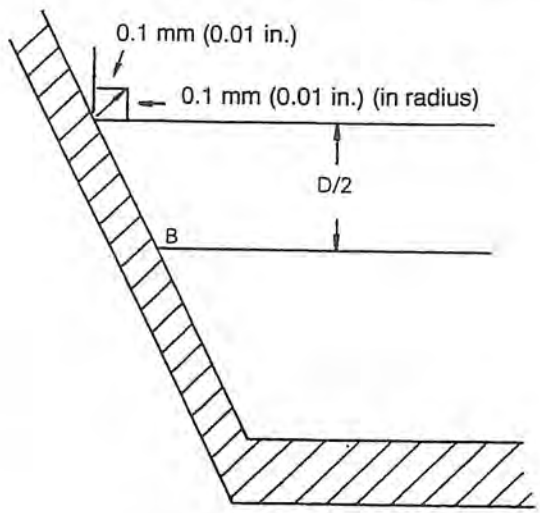
6) After point B is reached, cutting is performed along the final contour of rough turning cycle up to the point whose X coordinate is Xb + D. If G80 indicating end of contour definition is existent before such point is reached, cut along the final rough turning contour is performed up to the point specified in the block preceding the G80 block.

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 becomes effective. When an E word has not been specified, the feedrate specified when calling out rough turning cycle becomes active.



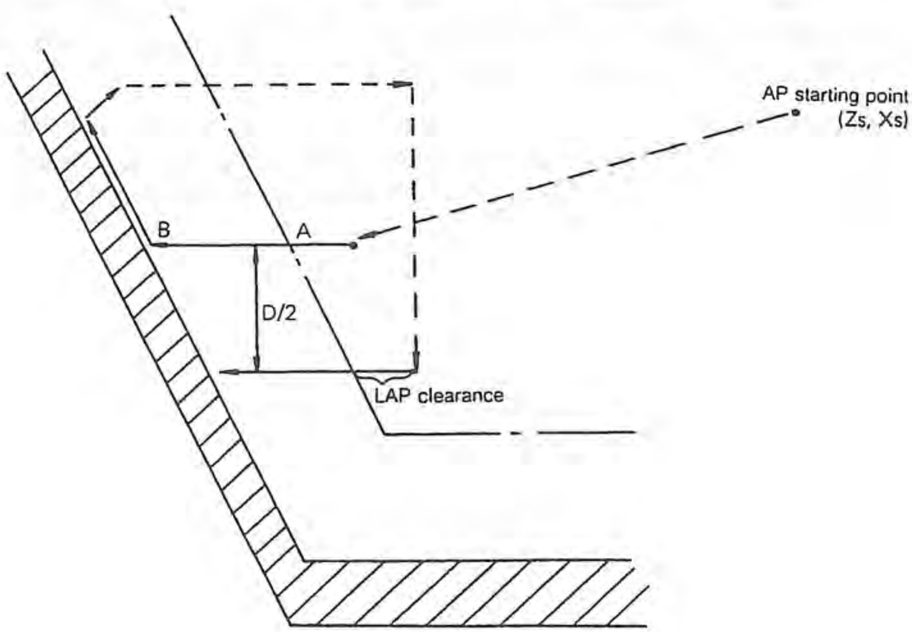
7) After the completion of cutting explained in 6), the cutting tool relieves from the workpiece in the direction opposite to infeed direction along the X-axis and opposite to cutting feed direction along the Z-axis as much as 0.1 mm (0.004 in.) on the respective axes (in diameter in the case of X-axis).

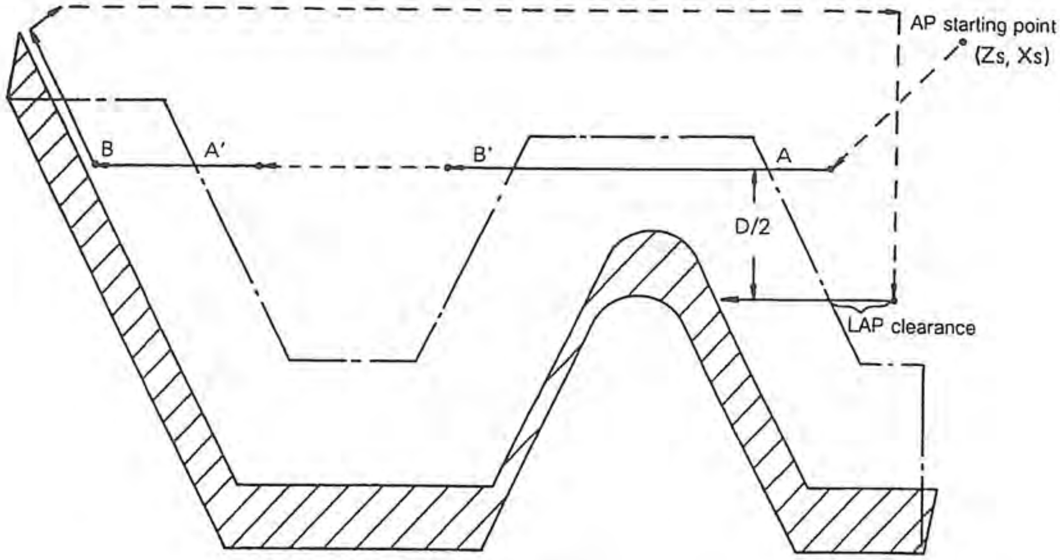
   The relieving amount is set at the optional parameter (OTHER FUNCTION 1) in units of μm.



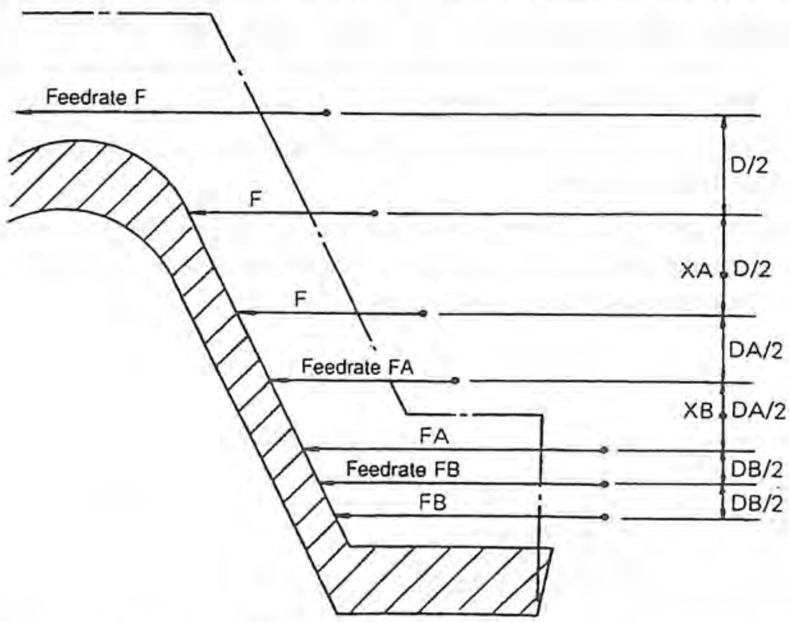
8) This completes the first rough turning cycle. The Z-axis returns to the next infeed point at the rapid feedrate and then X-axis to Xs.

   The next infeed starting point is the point which is away 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).



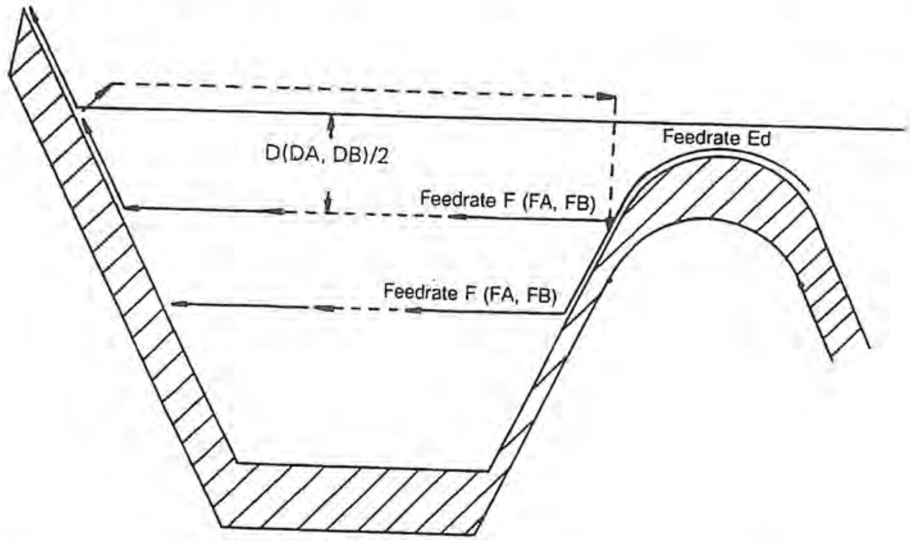


9) The steps 4) through 8) are repeated up to the cutting condition change point. After that point, the same cycle is repeated with the depth of cut (D) and feedrate (F) changed.



10) When 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 along the contour has been 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.

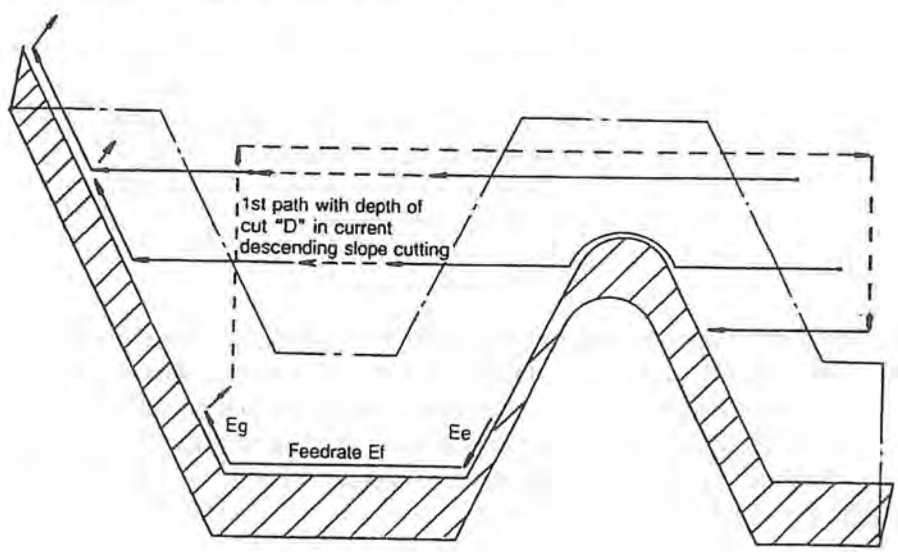
11) Steps 6) and 7) are repeated after that. The Z-axis then returns to the point where cutting along the Z-axis is started in step 10). After the Z-axis has been positioned, X-axis is positioned at the point where previous cutting cycle has been started.



12) Steps 10) and 11) are repeated until the most recessed section along the X-axis is cut. After the most recessed section has been cut, both the X- and Z-axis retracts by 0.1 mm (in radius for the X-axis), and the X-axis is positioned at the point having coordinate value "the first cutting level along the descending slope D + 0.2" mm.

After the completion of descending slope cutting, cutting before starting such cutting is resumed and steps after 4) are repeated.

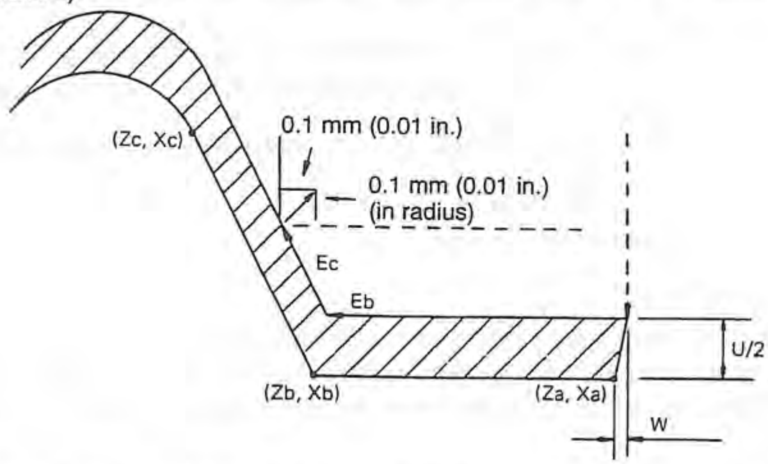
The next infeed starting point is the point which is away from the point of intersection between the line whose X-coordinate is "the first cutting level along the descending slope D - D" and the blank material shape, by the LAP clearance amount (Lc).



13) The steps indicated before are repeated until X-axis reaches the level where tool path is generated below "Xa + U" if ordinary infeed along X-axis is taken. When such level is reached, the final rough cutting is carried out along the contour leaving 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 relieve by 0.1 mm (0.004 in.) (in diameter for the X-axis). The relieving amount is set at the optional parameter (OTHER FUNCTION 1).

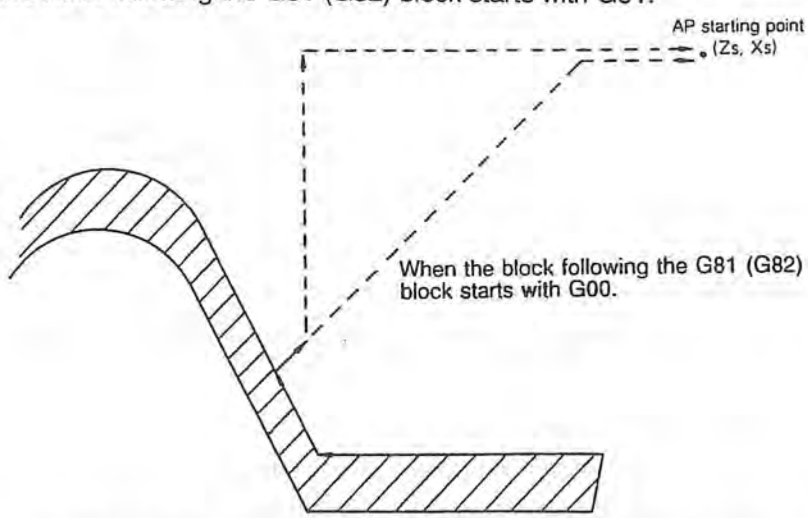


14) At the completion of step 13), the axes return to the AP starting point (Xs, Zs).

There are two patterns of axis return motion:

- 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 first made and then the Z-axis returns to the AP starting point when G01 is designated in the above indicated block.

When the block following the G81 (G82) block starts with G01:



The tool does not return to the AP starting point as explained in step 14) when M85 is designated in the block for calling for rough turning cycle (the block starting with G85).

This completes a rough turning cycle.

(b) Finish turning cycle in high-speed bar turning

1) With the commands in block N0261, positioning at the tool change position is performed.

2) With the commands in block N0262, S, T, and M commands for finish turning cycle are selected.

3) In block N0263, the control searches the program assigned with the program name N0608. Finish turning cycle in the bar turning mode is performed using this program.

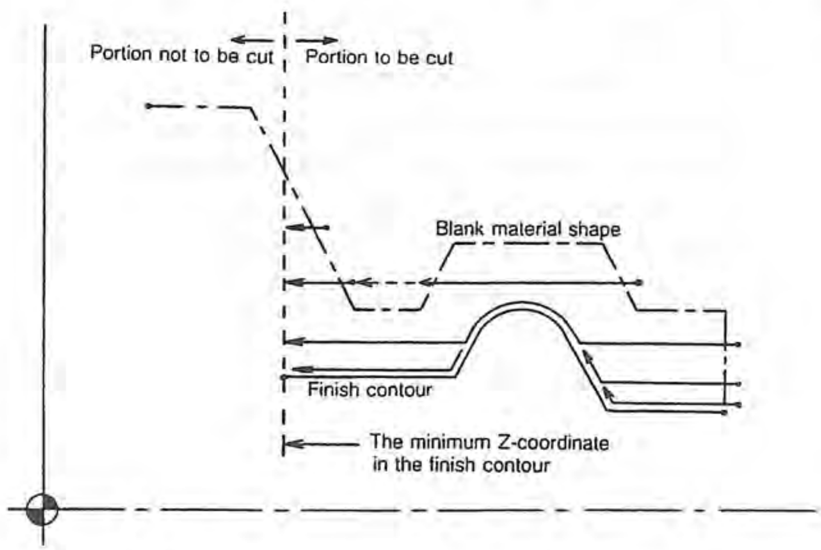
4) The finish turning cycle is perform following the dimension data designated in the contour definition program in the specified cutting conditions for the finish turning cycle.

5) After the finish turning cycle is complete, the commands in the block following N0263 are executed.

(4) Precautions in High-speed Bar Turning

Finish contour end point

In the 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.



(5) How to Obtain the Infeed Starting Point

The infeed starting point in 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 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 is made taking longitudinal cutting in the forward direction as an example.

(a) Cs (Zs) ≧ Bsp (Z) + Lc

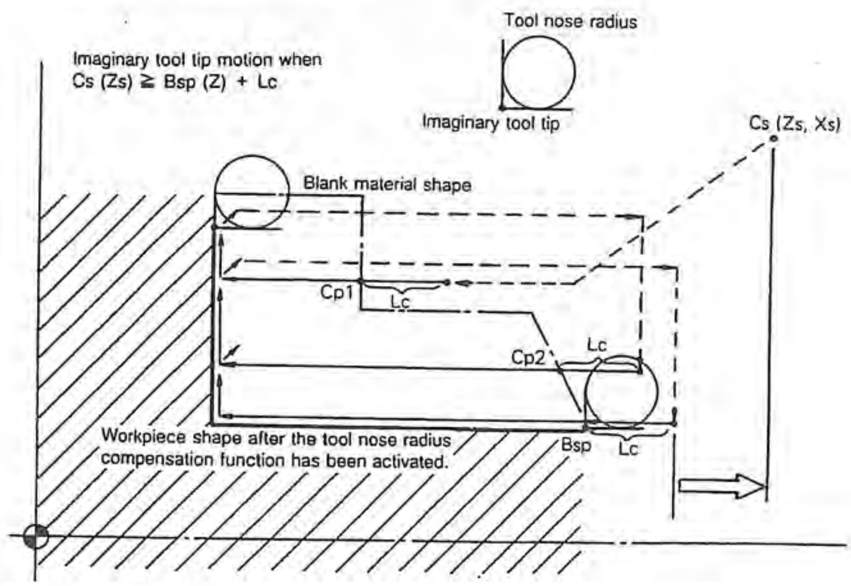
This is the normal positional relationship in bar turning.

1) X-coordinate of the infeed line > Bsp (X)

The infeed starting point is defined at "Cp (Z) + Lc, Cp (X)" which is away from point Cp by the LAP clearance amount (Lc).

2) X-coordinate of the infeed line ≦ Bsp (X)

When cutting is carried out from finish contour start point Bsp along the finish contour, the cutting tool is first positioned at the rapid feedrate at "Bsp (Z) + Lc, Bsp (X)" which is away from point Bsp by the LAP clearance amount (Lc) in the G00 mode, and it is then positioned at point Bsp at a cutting feedrate in the G01 mode.



(b) Cs (Zs) < Bsp (Z) + Lc

The portion that is to the right (in the Z-axis positive direction) of the 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 the above segment is "Xp".

1) Xp (Z) > Cp (Z) + Lc

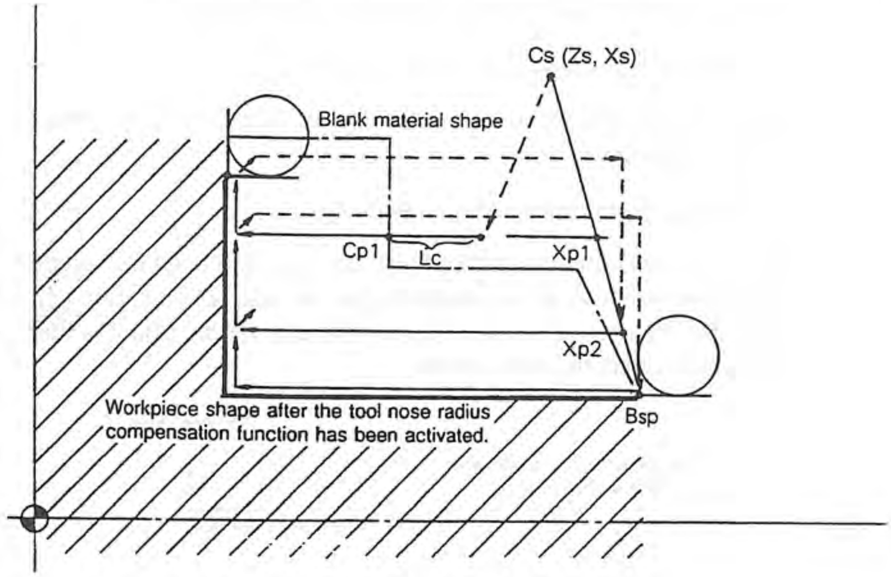
The infeed starting point is defined at "Cp (Z) + Lc, Cp (X)" that is away from point Cp by the LAP clearance amount (Lc).

2) Xp (Z) ≦ Cp (Z) + Lc

The infeed starting point is defined at point Xp (Z, X) where the segment Cs-Bsp intersects the infeed line.

(c) X-coordinate of the infeed line ≧ Bsp (X)

When cutting is carried out from finish contour start point Bsp along the finish contour, the cutting tool is directly positioned at point Bsp at the rapid feedrate.



5-5. AP Mode V (Bar Copying Cycle)

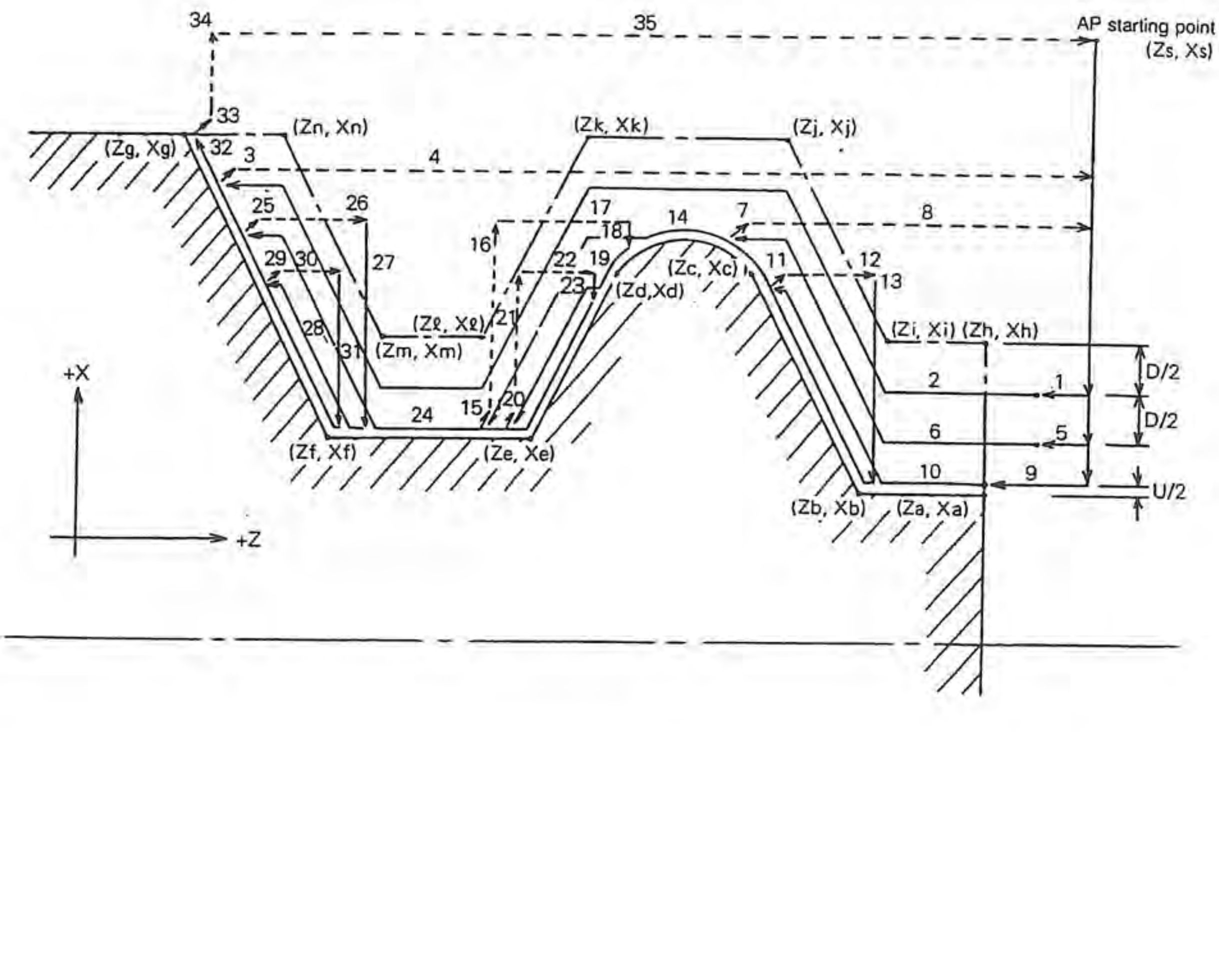
In the AP Mode V, the blank material shape data is input in addition to the finish contour shape data. The blank material shape is programmed in the blocks starting with G83.

Cutting is executed in parallel to the designated blank material shape. Once the cutting tool starts cutting the workpiece, it does not get distant from the blank material until it meets the finish contour.

This feature reduces the number of tool collision against the forged workpiece surface, resulting in longer tool life.

This mode is also effective to turn the ID of the workpiece, which used to be difficult to turn in the AP Mode II.

(1) Tool Path and Program - Longitudinal Cutting



Contour Definition

```
NAT80  G83 ................................. a) Blank material shape definition start G
                                                code
N0801  G01  Xa  Za                          ┐
N0802       Xh  Zh                          │
N0803       Xi  Zi                          │
N0804       Xj  Zj                          │ b) Blank material shape definition blocks
N0805       Xk  Zk                          │
N0806       Xl  Zl                          │
N0807       Xm  Zm                          │
N0808       Xn  Zn                          │
N0809       Xg  Zg                          ┘
N0810  G81 ................................. c) Finish contour definition start G code
N0811  G01  Xa  Za                          ┐
N0812       Xb  Zb            Fb   Sb   Eb  │
N0813       Xc  Zc            :    :    :   │
N0814  G03  Xd  Zd  Ib   Kd   :    :    :   │ d) Finish contour definition blocks
N0815  G01  Xe  Ze            :    :    :   │
N0816       Xf  Zf            Ff   Sf   Ef  │
N0817       Xg  Zg            Fg   Sg   Eg  ┘
N0818  G80 ................................. e) Contour definition end G code
```

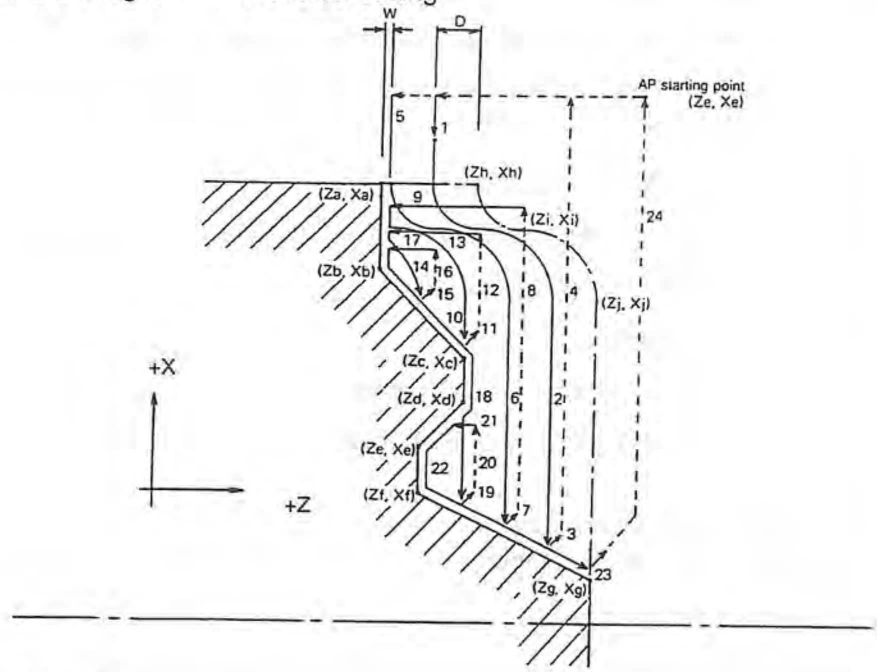
Rough Turning Cycle

```
N0181  G00  Xt  Zt                              Tool change position
N0182       Xs  Zs        STM                   Starting point of AP, S, T, and M for rough
                                                turning cycle
N0183  G86  NAT80   D   F   U   W   M85      f) Calls for rough turning cycle
```

Finish Turning Cycle

```
N0281  G00  Xt  Zt                              Tool change position
N0282                     STM                   S, T, and M for finish turning cycle
N0283  G87  N0810 ........................   g) Calls for finish turning cycle
```

(2) Tool Path and Program - Transverse Cutting



Contour Definition

```
NAT90  G83 .................................. a) Blank material shape definition start G code
N0901  G00  Xa  Za                              ┐
N0902  G01  Xh  Zh                              │ b) Blank material shape definition blocks
N0903  G03  Xi  Zi  Ii  Ki                      │
N0904  G02  Xj  Zj  Ij  Kj                      │
N0905  G01  Xg  Zg                              ┘
N0906  G82 .................................. c) Finish contour definition start G code
N0907  G00  Xa  Za                              ┐
N0908  G01  Xb  Zb           Fb  Sb  Eb         │
N0909  G00  Xc  Zc           :   :   :          │
N0910  G01  Xd  Zd           :   :   :          │ d) Finish contour definition blocks
N0911       Xe  Ze           :   :   :          │
N0912       Xf  Zf           Ff  Sf  Ef         │
N0913       Xg  Zg           Fg  Sg  Eg         ┘
N0914  G80 .................................. e) Contour definition end G code
```

Rough Turning Cycle

```
N0191  G00  Xt  Zt ........................... Tool change position
N0192       Xs  Zs          STM                Starting point of AP, S, T, and M for rough turning cycle
N0193  G86  NAT90  D   F   U   W   M85         f) 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  N0908 ............................ g) Calls for finish turning cycle
```

(a) Blank material shape definition start G code

- 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.

(b) 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.

(c) 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

(d) 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.
- 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.

(e) Contour definition end G code

- This code declares the end of contour definition.

(f) Calls for rough turning cycle

- Rough turning cycle is started by calling the contour definition blocks starting with G86.
- When the contour definition blocks start with G83, the AP Mode V (bar turning cycle) is selected. (LAP4)
- When the finish contour definition blocks start with G81 or G82, the AP Mode II (copy turning cycle) is selected.

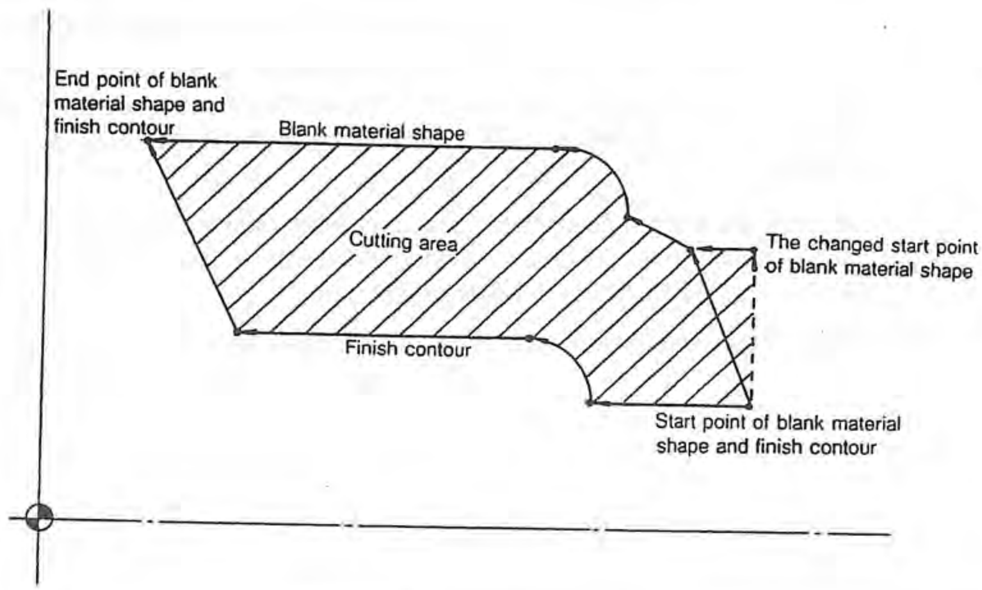
(g) Calls for finish turning cycle

- Finish turning cycle is carried out by designating G87 and calling for the finish contour definition blocks starting with G81 or G82.

NOTICE : (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.



(3) Outline of Bar Copying Cycle

(a) Rough turning cycle in the longitudinal direction

1) With the commands in block N0181, positioning at the tool change point is performed.

2) With the commands in block N0182, S, T, and M commands for rough cut cycle are selected, and then positioning at the LAP starting point is performed.

When no S, T, or M commands are provided in this block, those selected in the preceding block(s) become effective.

3) With the NAT80 command in block N0183, the control searches the program assigned with the program name NAT80. Rough cut cycle in bar turning mode is performed on this program.

   When NAT80 is designated in the block starting with G83, high-speed bar turning cycle (LAP4) is carried out.

   In the same block, cutting conditions for rough turning cycle are also specified.

   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 retracting motion 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 an F word is not designated in this block, the feedrate commanded last becomes effective.

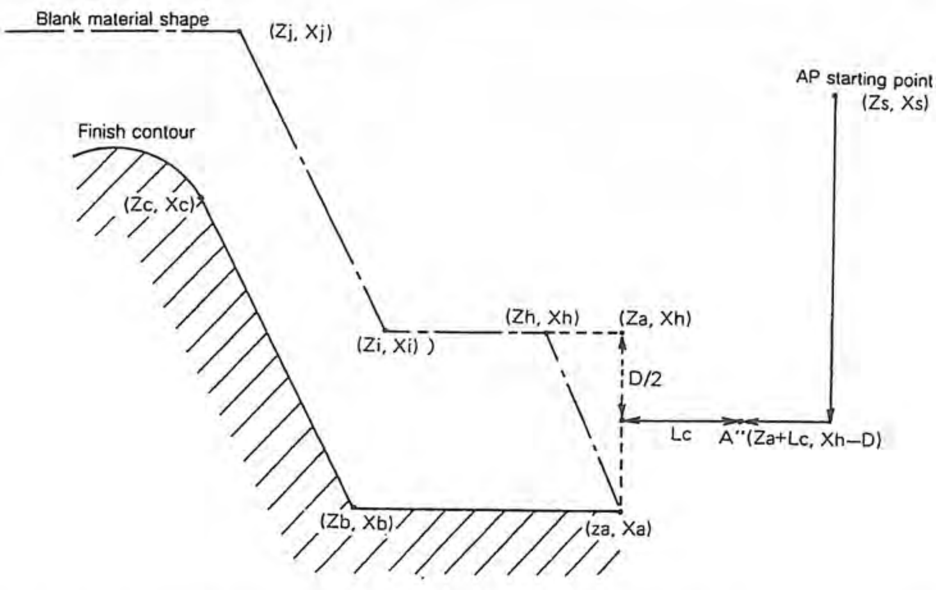
4) The commands between G83 and G81 are taken as the commands to define the blank material shape. And the commands between G81 and G80 are taken as the commands to define the finish contour.

   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.)

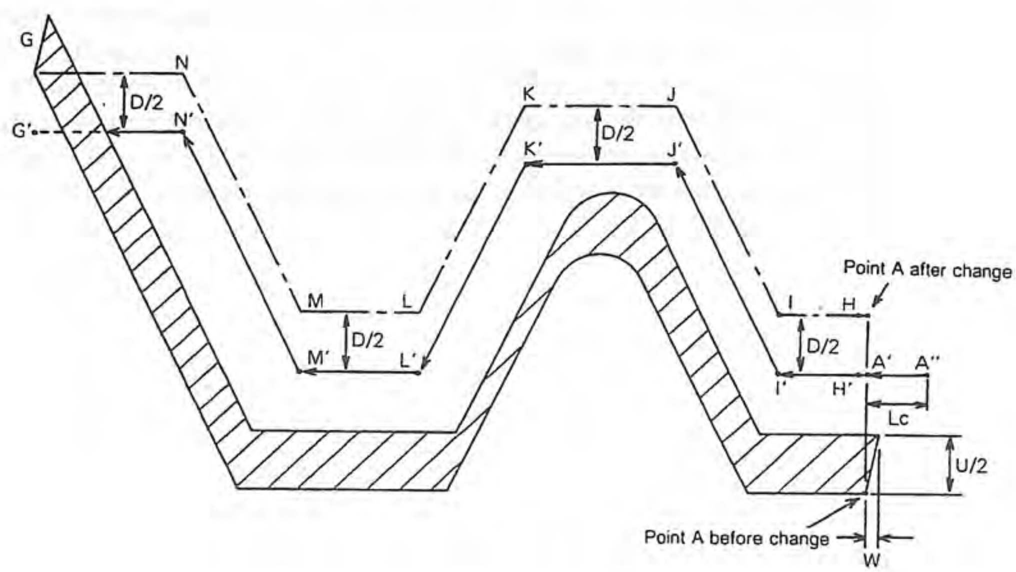
   Then, first the X-axis, then the Z-axis is positioned at a cutting feedrate at point A" which is obtained by shifting the X-coordinate of the first element by depth of cut "D" in the negative direction and then shifting the Z-coordinate by the LAP clearance amount (Lc) in the positive direction.

   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.

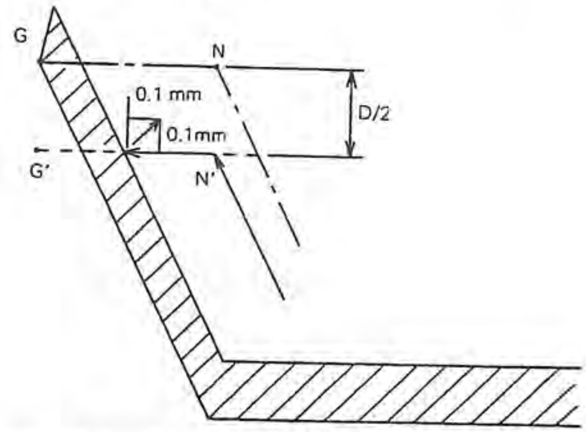
   The LAP clearance amount (Lc) is set at the optional parameter (OTHER FUNCTION 1) in units of μm.



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.

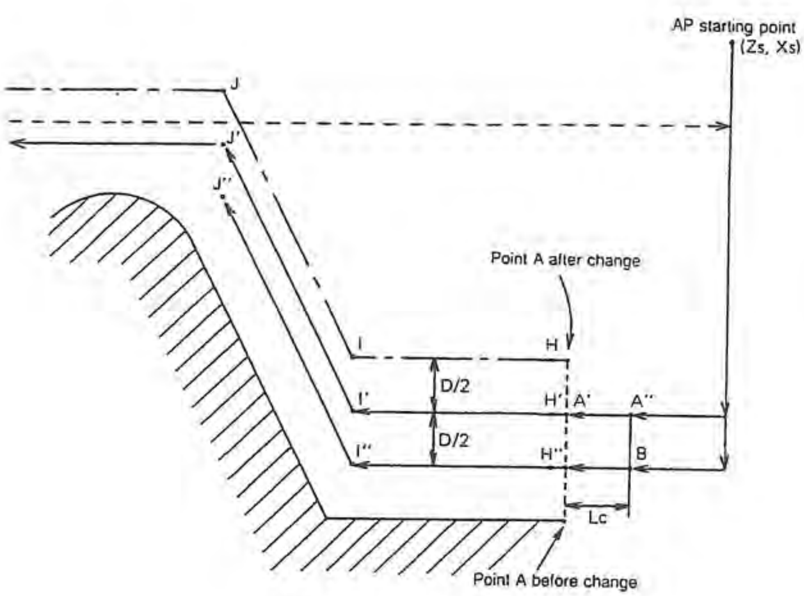


6) When cutting reaches the point where the shifted blank material shape intersects the finish contour, the cutting tool relieves by 0.1 mm (in radius for the X-axis) in the direction opposite to infeed direction along the X-axis and opposite to cutting feed direction along the Z-axis, respectively. The relieving amount is set at the optional parameter (??other function 1??) in units of μm. When stock removal is designated in the program using the U or W command, the cutting tool relieves when cutting reaches the point where the shifted material shape intersects the final rough turning contour.



7) This completes the first rough turning cycle.

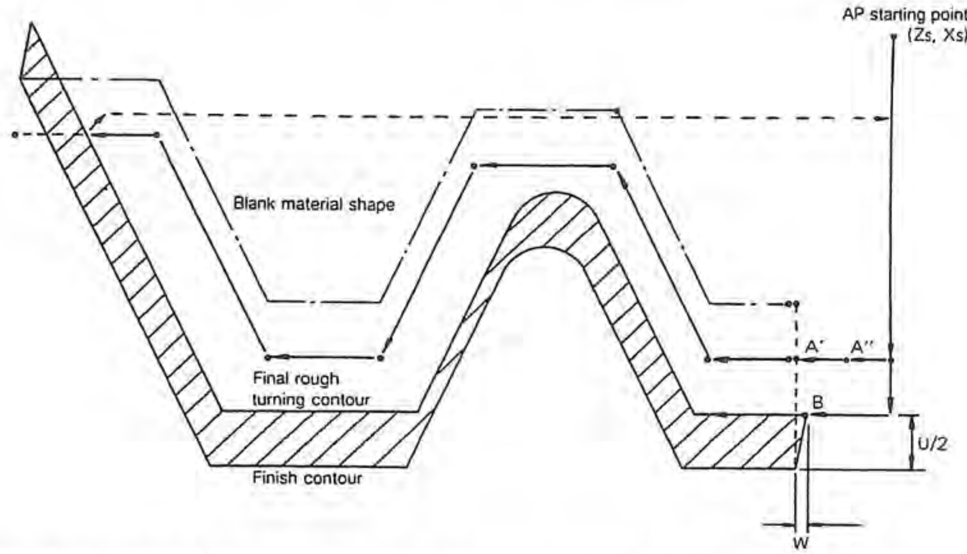
The cutting tool is then positioned at next infeed starting point B at the rapid feedrate. When the X-coordinate at the completion of the first rough turning cycle is smaller than the largest X-coordinate of the next cutting level, the cutting tool moves up to the point "the largest X-coordinate + 0.2 mm" (in diameter) at the rapid feedrate ("the 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 carried out in the same direction as 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)".



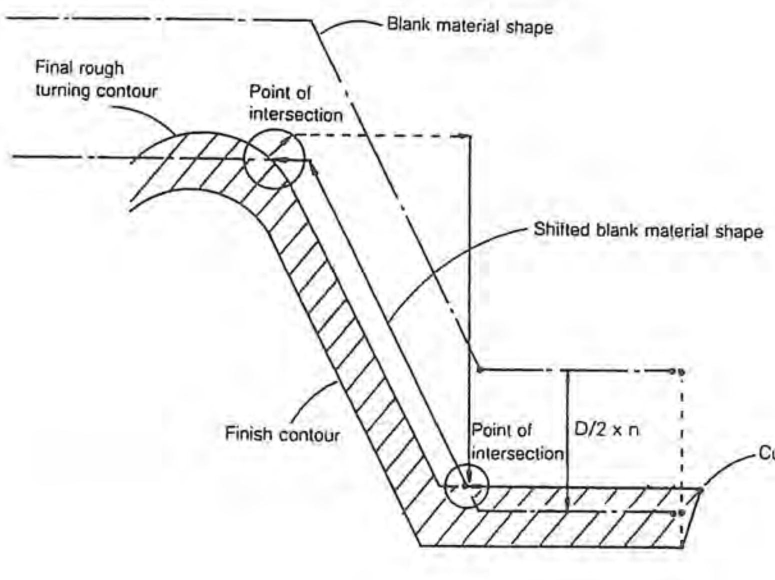
8) When the "Xh - 2D" value is smaller than the Xa value, the finish contour start point is taken as next infeed starting point B.

When the U or W command has been designated, the final rough turning contour is taken as 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 becomes 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 becomes effective.

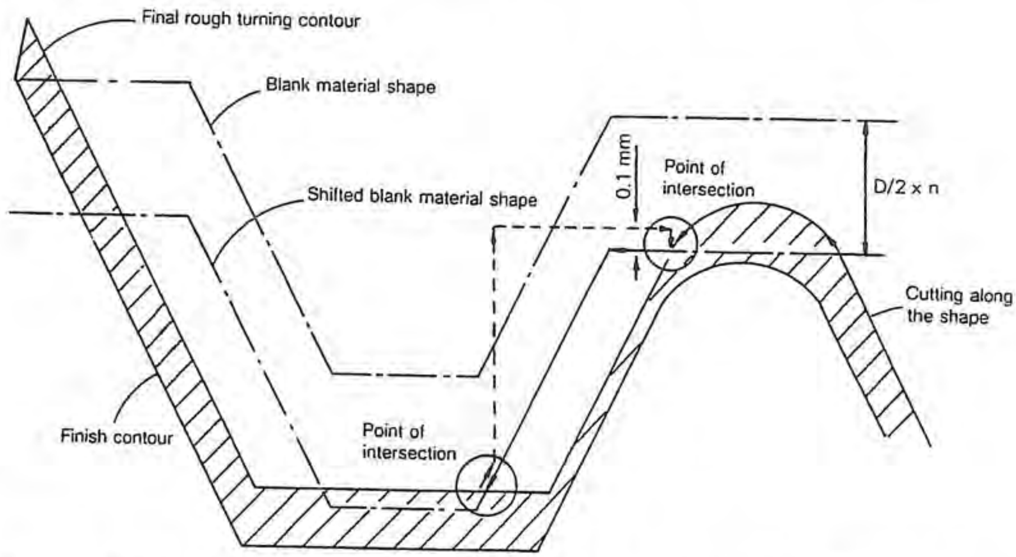


9) When the blank material shape shifted by "D × even number" intersects the contour to be machined (or final rough turning contour) during cutting along the shape, the cutting tool starts cutting along the shifted material shape. When the blank material shape shifted by "D × even number" again intersects 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 the point which is above the point where cutting along the shifted blank material has been started, and then the X-axis is positioned at this point.

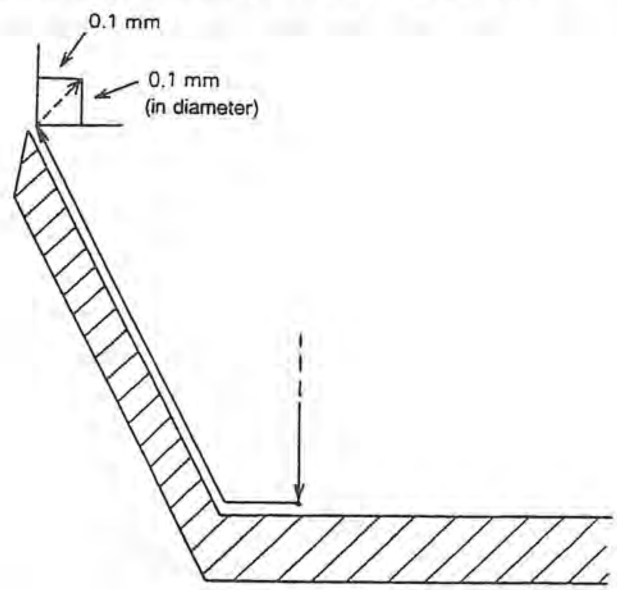


In rough turning cycle in the AP Mode IV, the axes return to the point where cutting along the shifted blank material has been started in the following procedure:

- The X-axis is positioned at the 'largest X-coordinate in that cutting cycle + 0.2 mm (0.008 in.) (in diameter)" point.
- The Z-axis is positioned at the point which is above the point where cutting along the shifted blank material has been started.
- The X-axis is positioned at the point where cutting along the shifted blank material has been started at a cutting feedrate.



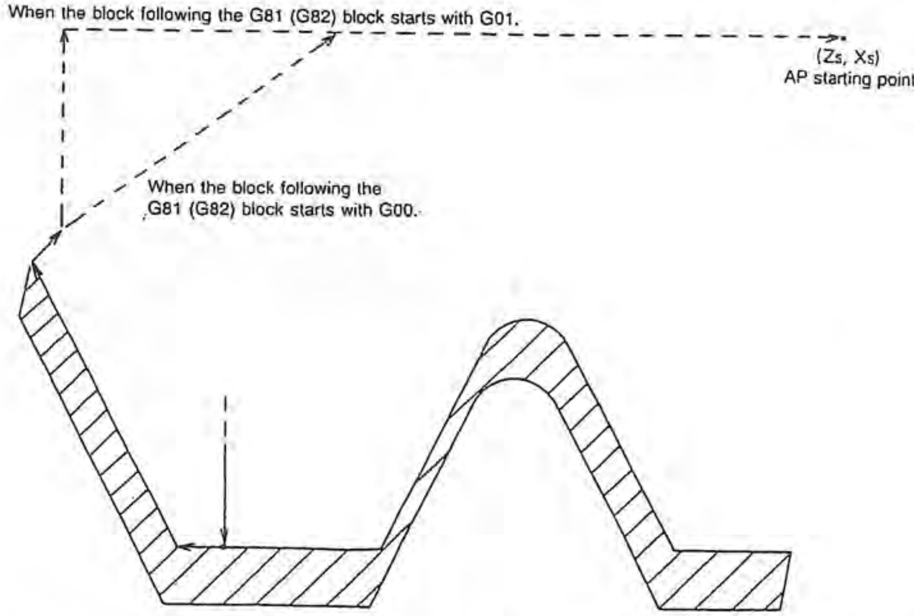
10) Steps 8) through 9) are repeated until the area between the blank material shape and the finish contour (or final rough turning contour) is cut. Then, the cutting tool relieves by 0.1 mm (0.004 in.) (in diameter for the X-axis) in the direction opposite to infeed direction along the X-axis and opposite to cutting feed direction along the Z-axis, respectively. The relieving amount is set at the optional parameter (OTHER FUNCTION 1).



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:

- 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 first made and then the Z-axis returns to the AP starting point when G01 is designated in the above indicated block.



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 rough turning cycle.

(b) Finish turning cycle in the longitudinal direction

1) With the commands in block N0281, positioning at the tool change position is performed.

2) With the commands in block N0282, S, T, and M commands for finish turning cycle are selected.

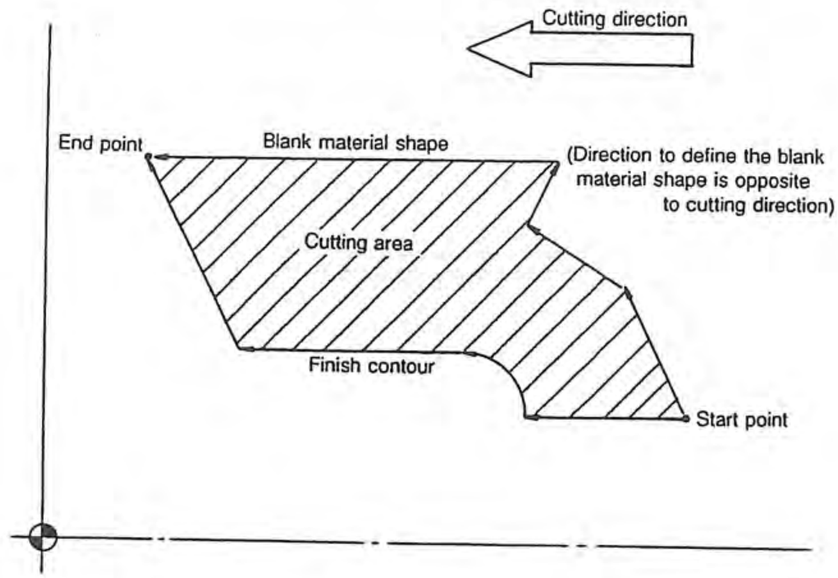
3) In block N0283, the control searches the program assigned with the program name N0810. Finish turning cycle in the bar turning mode is performed using this program.

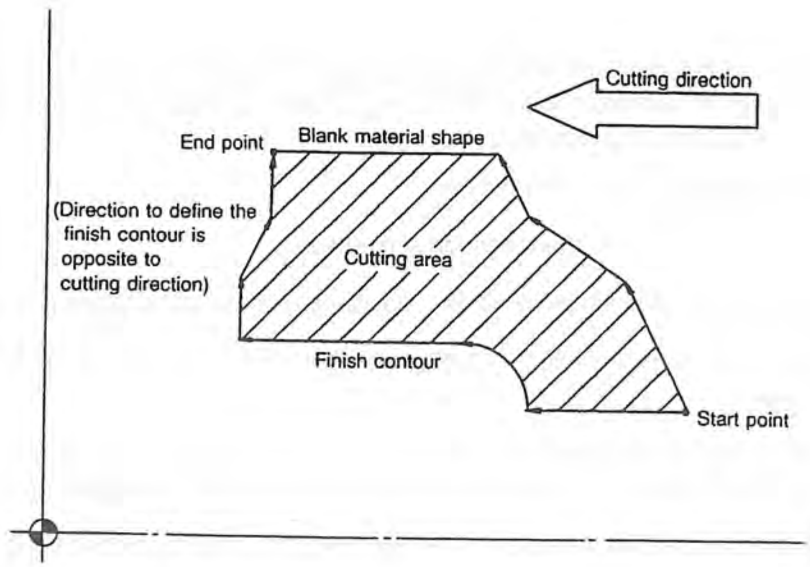
4) The finish turning cycle is performed according to the cutting conditions for finish turning (F command for feedrate, S command for spindle speed) specified in the shape definition program.

5) After the finish turning cycle is complete, the commands in the block following N0283 are executed.

(4) Precautions in Bar Copying Cycle

(a) When the direction to define the blank material shape or finish contour is opposite to cutting direction, an alarm occurs. In such cases, retry shape definition or divide the machining process.





6. Precautions

Precautions in using the LAP function are provided below:

(1) Be sure to designate contour defining sequence name right after the G code calling for execution of LAP program:

G85, G86, G87 and G88

(2) The G83 (G81 or G82) code used to indicate the start of contour definition must be assigned with a proper sequence name.

(3) For absolute or incremental programming, G90 or G91, the mode established when G85, G86, G87 or G88 is commanded becomes effective. However, it will be changed if 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 with X and/or Z commands in the same block.

(4) Concerning G64, G65, G94, G95, G96, and G97, the mode established when G85, G86, G87, or G88 is commanded becomes effective. Once established, it cannot be changed within the contour definition program.

(5) Concerning G00, G01, G02, G03, G31, G32, G33, G34, G35, G64, G65, G94, G95, G96, G97, G112 and G113, those effective when G85, G86, G87 or G88 is commanded become active after the end of the LAP.

(6) Nesting from LAP to LAP is impossible.

(7) If a G code calling for the LAP (G85, G86, G87 and G88) is designated while nose radius compensation mode is active, an alarm results.

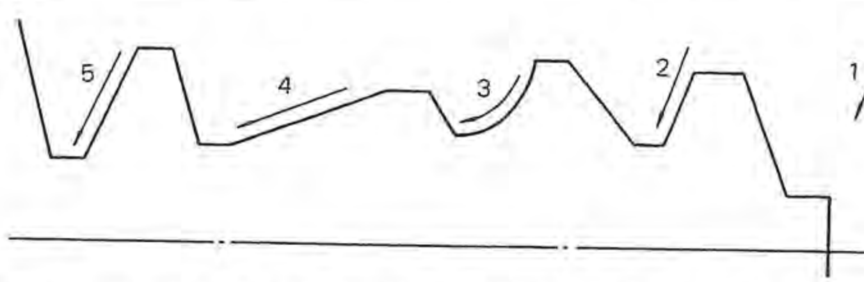
(8) Nose radius compensation can be activated during LAP; however, be sure to cancel the activated nose radius compensation mode before the G80 block which indicates the end of contour definition.

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          Xj   zj   G40
N0021   G80
```

Be sure to activate and cancel the LAP function between G81 (G82) block and G80 block.

(9) The maximum programmable number of descending slopes in the AP Mode I and AP Mode IV is ten (10).



For the shape illustrated above, the number of descending slopes is five. If more than ten descending slopes are programmed, an alarm results.

(10) 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.)

(11) In the AP Mode IV and the 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.

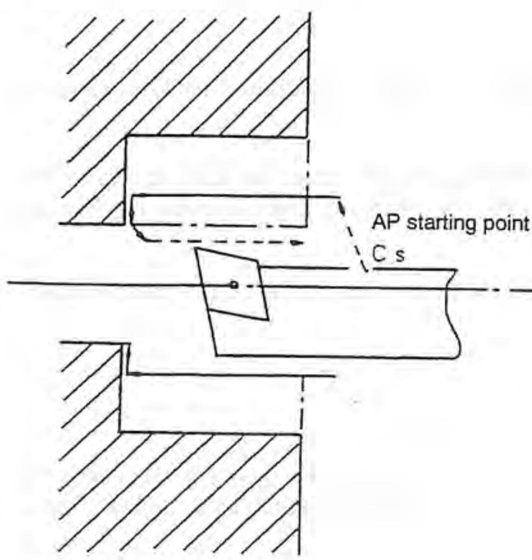
(12) When the blocks which define the blank material shape are deleted from the NC program intended for the AP Mode IV or AP Mode V, the program can be run in the AP Mode I or AP Mode II. To allow this change, call the same sequence number as called in the G81/G82 block in the G85/G86 block.

When the AP Mode has been changed, tool path is changed accordingly. Special care must be taken in the following cases.

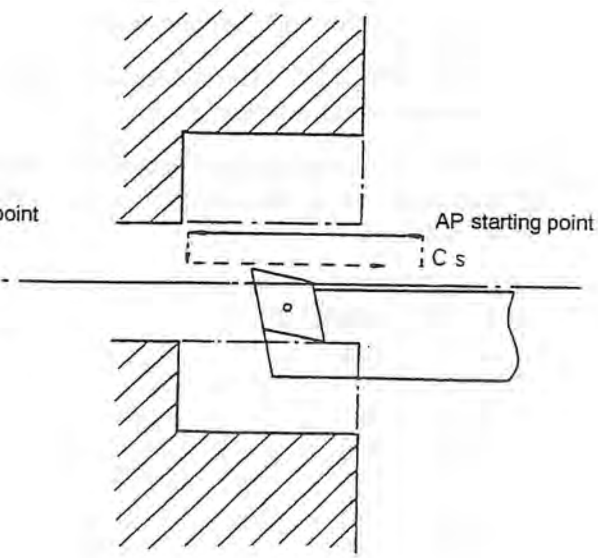
(a) ID machining

The cutting tool may interfere with the workpiece. Correct the program as necessary, for example, change the AP starting point.

1) From AIP Mode IV to AP Mode I

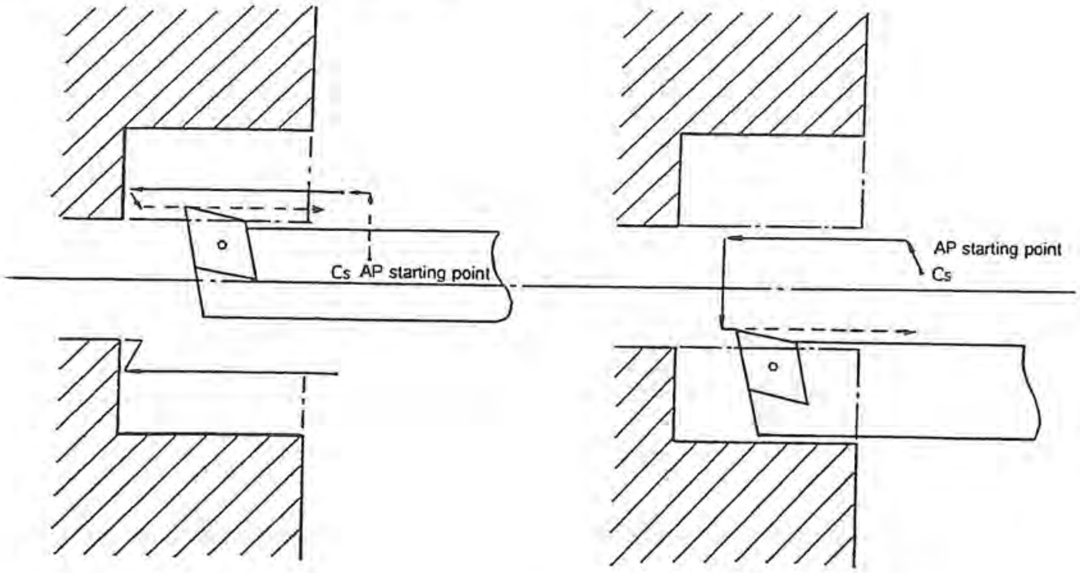


In AP Mode IV



In AP Mode I

2) From AP Mode V to AP Mode II



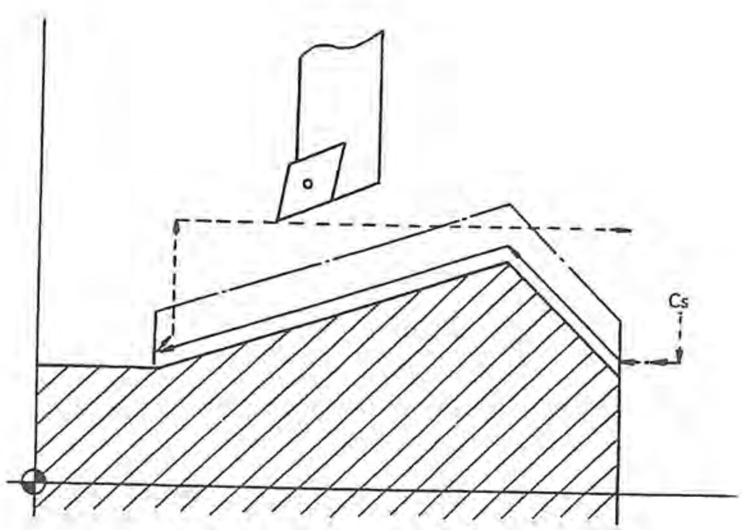
In AP Mode V

In AP Mode II

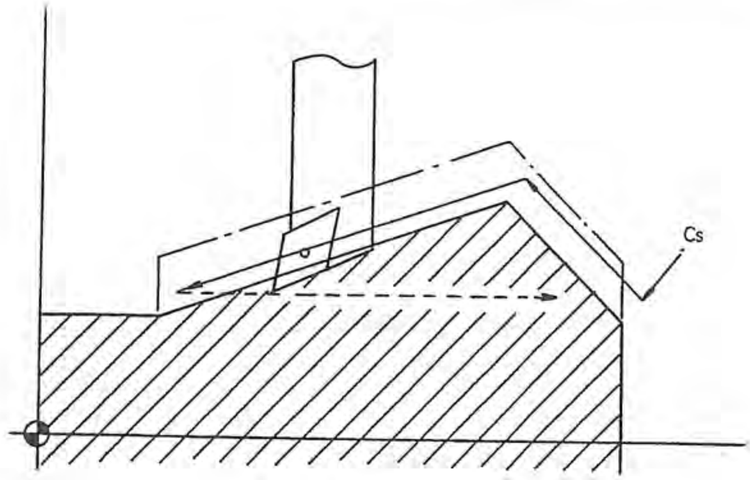
(b) Copy turning in descending slope

In the AP Mode II, the diameter must be largest at the end point of contour definition portion (must be smallest in ID turning). Otherwise, the cutting tool interferes with the workpiece.

1) From AP Mode V to AP Mode II



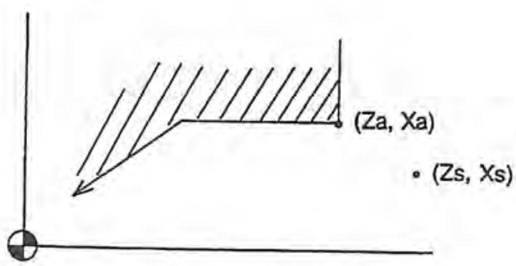
In AP Mode V



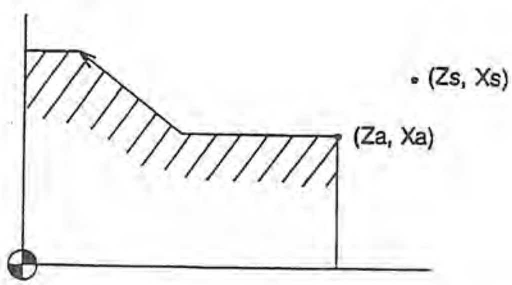
In AP Mode II

(13) The relation between the AP starting point (Zs, Xs) and the cutting start point (Za, Xa) must satisfy the following.

For ID cutting: Zs > Za, Xs < Xa

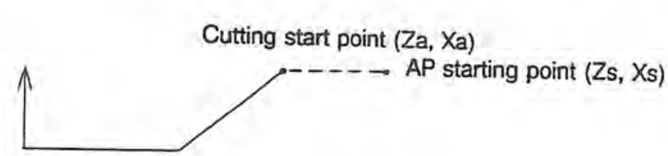


For OD cutting: Zs > Za, Xs > Xa



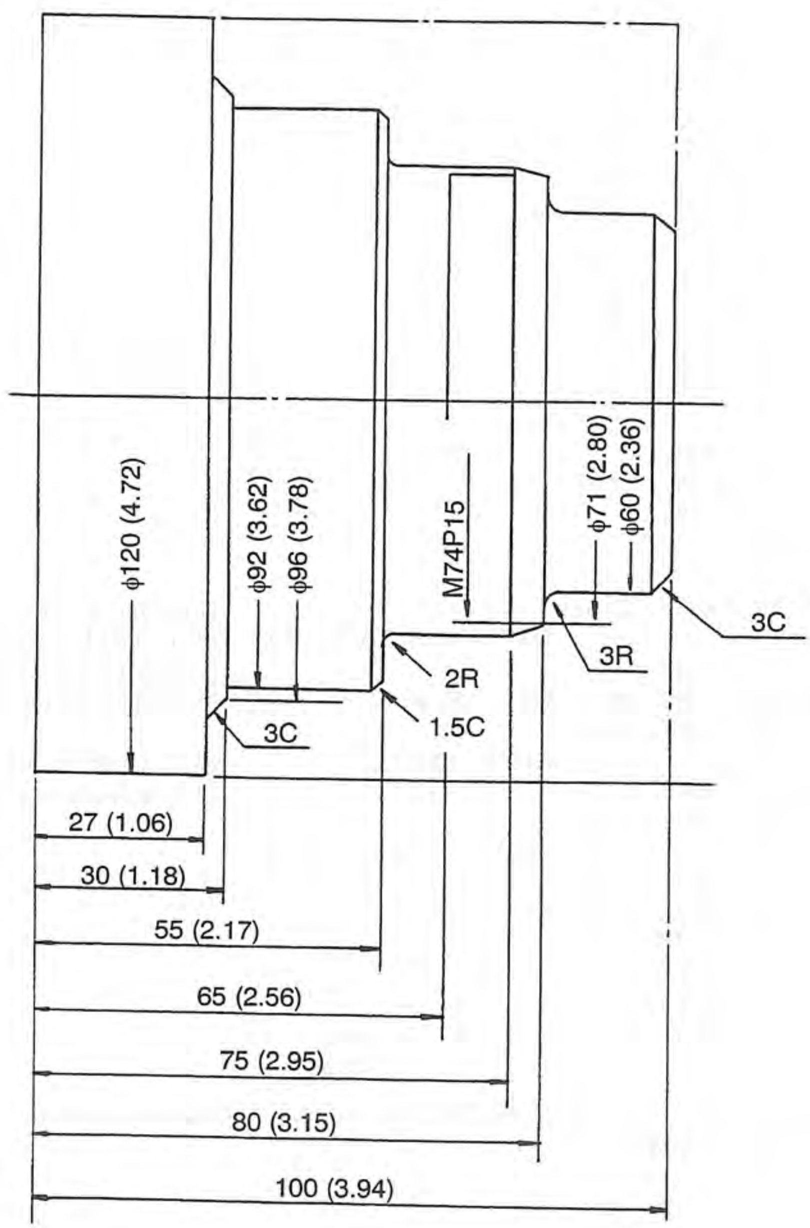
Bear the above relations in mind when designating the AP starting point and the cutting start point.

Example:



When the cutting start point and the AP starting point have been designated as illustrated above (where Xs = Xa), a cycle operation error will occur.

7. Application of LAP Function



Unit: mm (in.)

(1) 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                     │
N008              Z57                                   │
N009  G02  X78    Z55    I2    E0.45                    │  (Contour Definition)
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  X800   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
H113       X71.73
N114  G00  X800   Z102                 M05
N115                                   M02
```

*: Contour defining program beginning from G81 and ending with G80 may be entered at any position within the above indicated program.

(2) Machining Example using the AP Mode IV

Program Example:

```
O0002
NAT1  G83
N001  G01  X54    Z102
N002       X122
N003              Z27
N004  G81
N005  G00  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
```

*: Contour defining program beginning from G81 and ending with G80 may be entered at any position within the above indicated program.

SECTION 19 CONTOUR GENERATION

1. Contour Generation Programming Function (Face)

(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.

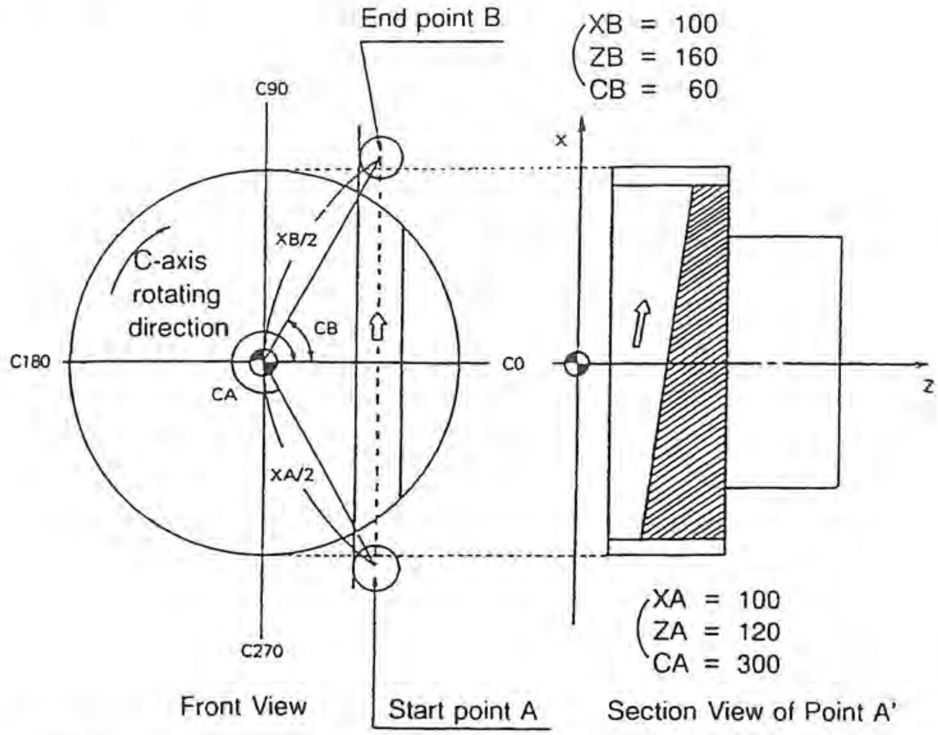
Note that simultaneous three-axis control of X, Z, and C axes are possible for straight line cutting on a plane.

(2) Programming Format

Machining	Format	
Straight line cutting	G101 X Z C F	X : Z : C : Coordinate values of the target point in straight line generation F : Feedrate (mm/min)
Arc cutting	G102 X C L F	X : C : Coordinate values of the end point of arc in CW arc generation L : Arc radius F : Feedrate (mm/min)
	G103 X C L F	X : C : Coordinate values of the end point of arc in CCW arc generation L : Arc radius F : Feedrate (mm/min)

(3) Programming Examples

(a) Straight line cutting

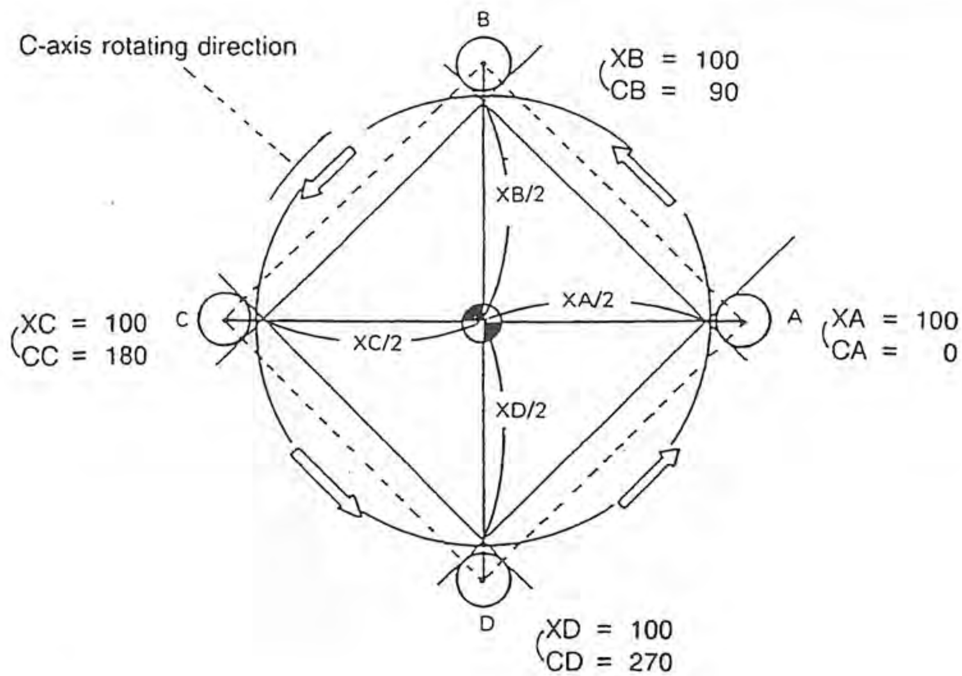


Program 1: Simultaneous 2-axis control of X and C axes

N101	M110					C-axis joint
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

Program 2: Simultaneous 3-axis control of X, Z and C axes

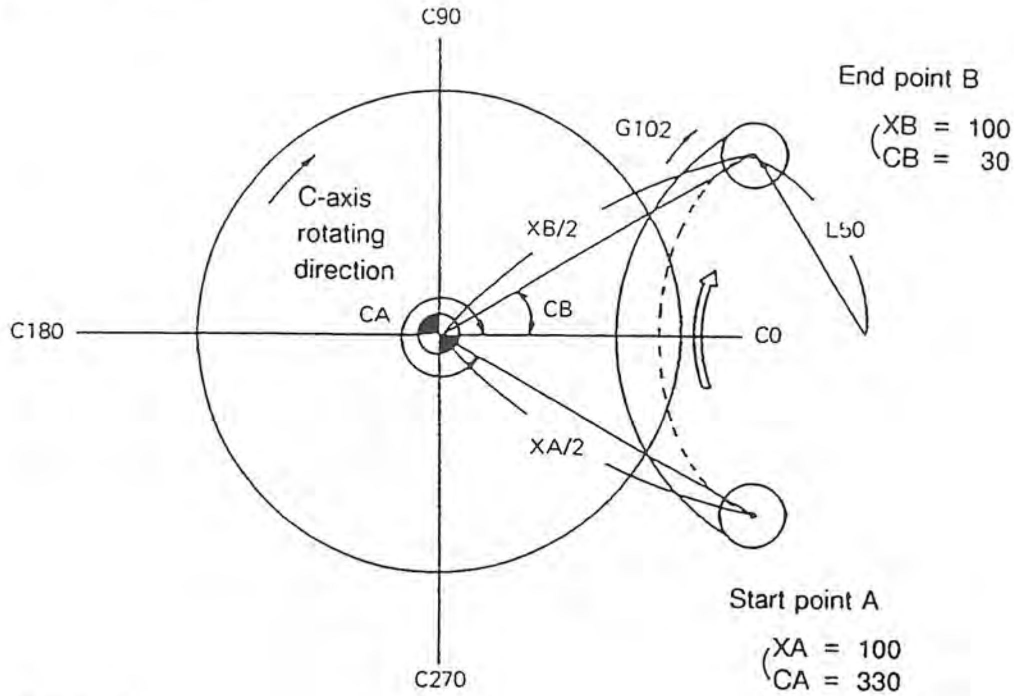
N101	M110					C-axis joint
N102	M146	M15				C-axis unclamp
N103	G00	X100	C300	T0101	SB=250	Positioning
N104	G94		Z120		M13	Start point A
N105	G101		Z160	C60	F30	End point B



Program 3:

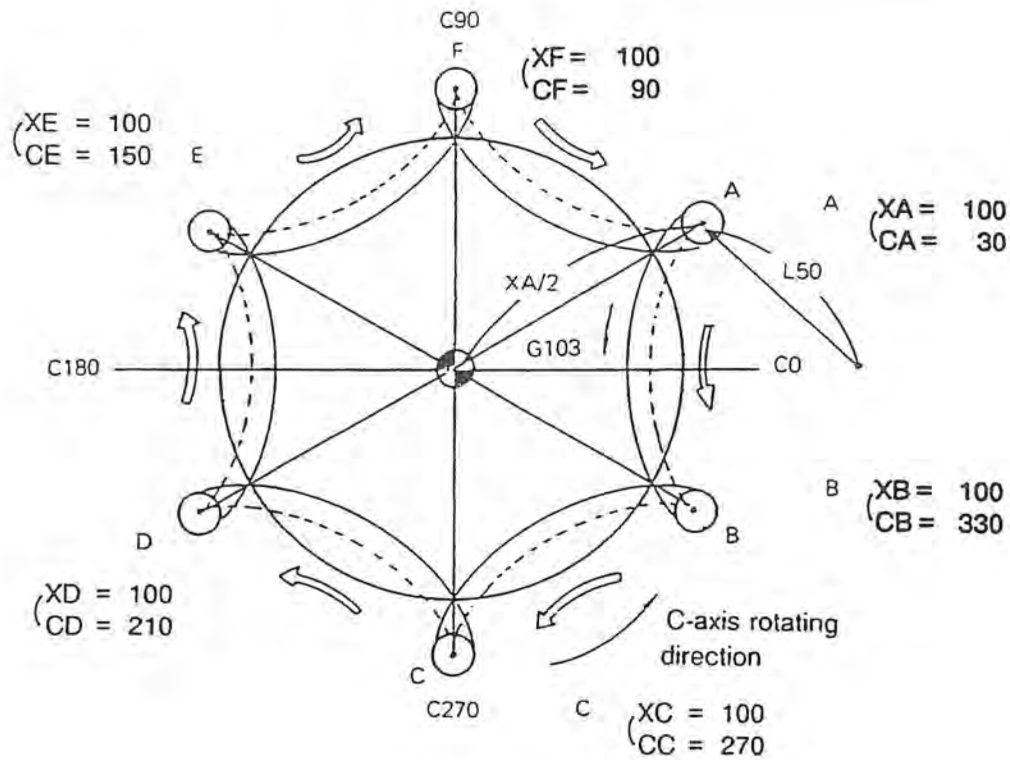
N101	M110					C-axis joint
N102	M146	M15				C-axis unclamp
N103	G00	X100	C0	T0101	SB=250	Positioning
N104	G94		Z120		M13	Start point A
N105	G101			C90	F30	End point B
N106				C180		End point C
N107				C270		End point D
N108				C0		End point A

(b) Arc cutting



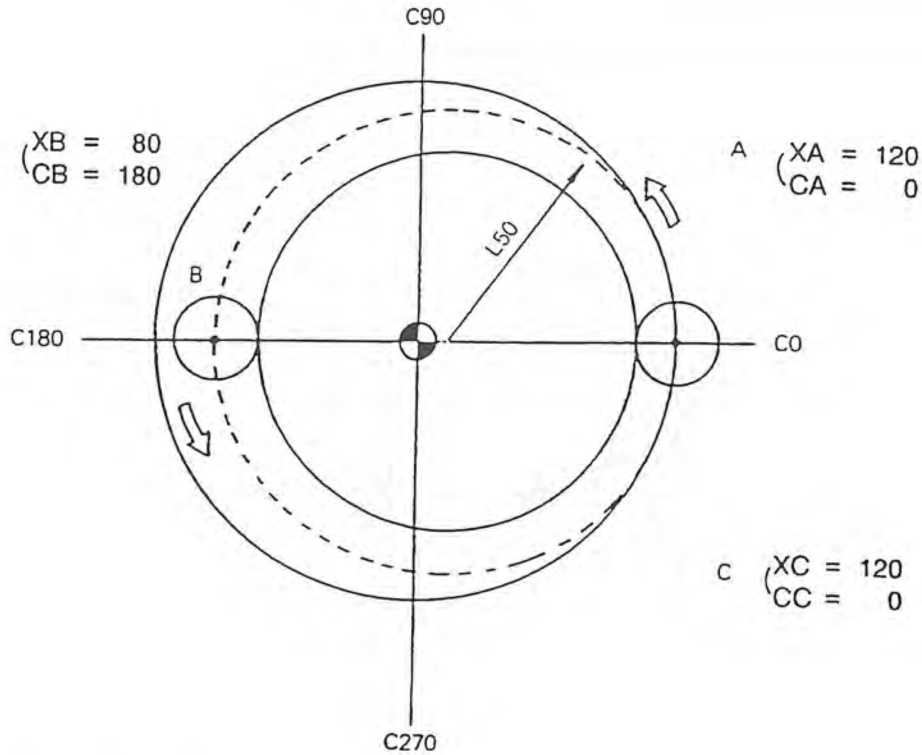
Program 1:

N101	M110							C-axis joint
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



Program 2:

N101	M110						C-axis joint
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



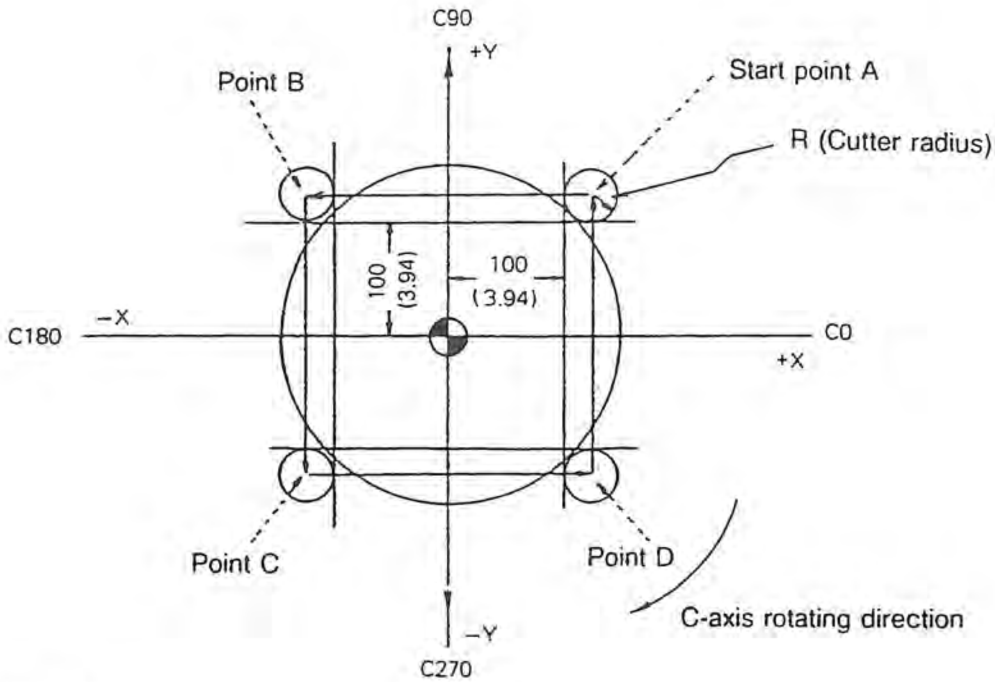
Program 3:

N101	M110							C-axis joint
N102	M146	M15						C-axis unclamp
N103	G00	X120	C0	T0101			SB=250	Positioning
N104	G94		Z120	M13				Start point A
N105	G103		X80	C180	L50	F30		End point B
N106			X120	C0	L50			End point C

(4) Programming Example 2

Combination with Coordinate System Conversion Function

Example 1:



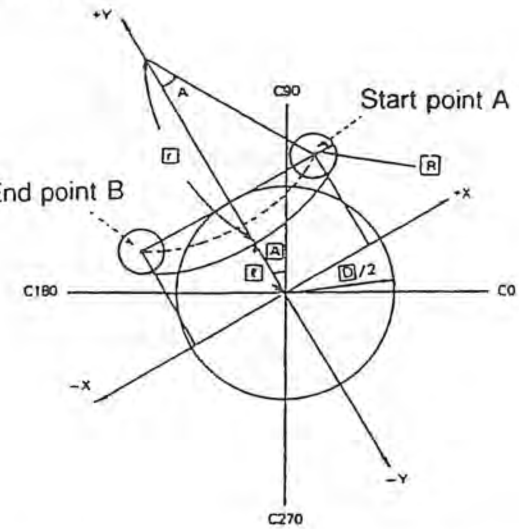
V1 = R

Cutter radius should be set at common variable V1 beforehand.Program:N101

Program:

N101	M110					C-axis joint
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	B101	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

Example 2:



Data to be given:

- r = radius of arc to be cut
- ℓ = depth of cut
- θ = angle
- R = cutter radius
- D = workpiece diameter

X and Y coordinate values of the start point can be calculated as follows:

$$X = (r - R)\sin A$$

$$Y = r + \ell - (r - R)\cos A$$

where,

$$A = \cos^{-1}\frac{(\ell + r)^2 + (r-R)^2 - (D/2 + R)^2}{2(\ell + r)(r - R)} \quad \cdots\cdots\cdots (1)$$

Assume r = 220 mm, ℓ = 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 R should be set at common variable V1 in advance.

Program:

∫

N101	M110			C-axis joint
N102	M146	M15		C-axis unclamp
N103	G137	C30		Start of coordinate system conversion
N104	G00	X[200–V1] *sin [35] Y220 + 60 – [220–V1] *cos [35]		
	T0101	SB=250		
N105	G94	Z100	M13 M13	Positioning at start point A
N106	G102	X – [200–V1] *sin [35] Y220 + 60 – [220–V1] *cos [35]		
	L220 – 20		F30	Cutting up to point B
N107	G136			End of coordinate system conversion

∫

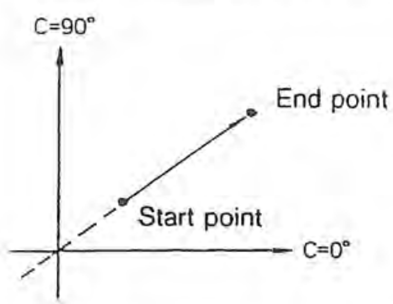
Note: *If the control is not supported by the user task 2 (optional), trigonometric function calculation cannot be carried out by the controller. Therefore, programming must be made by directly entering numeric values.*

(5) Supplementary Information

(a) Special operation in the G101 mode

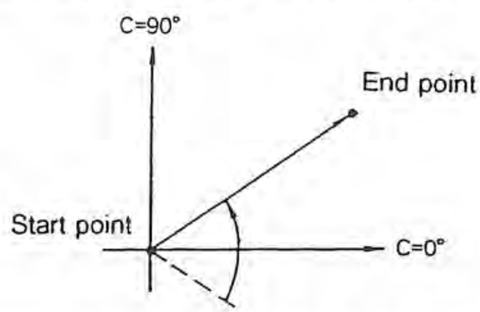
If the tool paths commanded without the cutter radius compensation function or the calculated tool paths as the results of activation of the cutter radius compensation function are the straight lines passing the center of the X-C coordinate, the following special operation occurs.

1) When the C commands of the start and end points are the same:



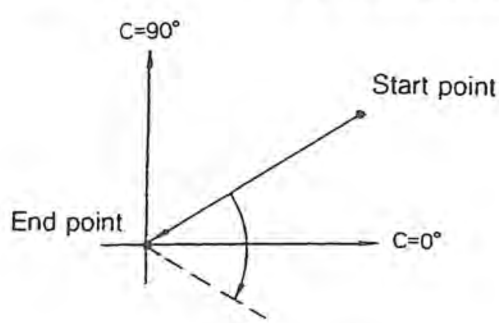
Although the G101 command calls for compound X- and C-axis motion, only the X-axis moves in this case (the same as G01 motion).

2) When the start point lies at the center and the C commands of the start and end points differ:



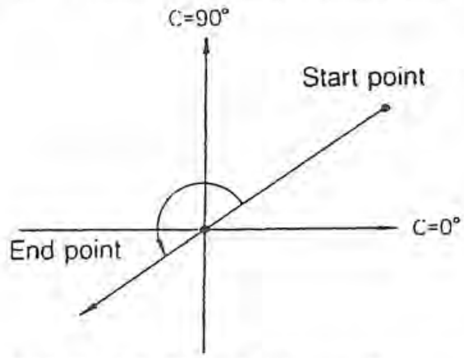
In this case, only the C-axis moves first until the commanded value is reached; then X-axis motion occurs.

3) When the end point lies at the center and the C commands of the start and end points differ.



In this case, conversely from case 3) above, only the X-axis moves first until the commanded value is reached; then C-axis motion occurs.

4) When the start and end points lie at the opposite side in reference to the C-axis center with the C-axis commands at these points apart by 180°:



In this case, only the X-axis moves first until it reaches "0". Then, the C-axis moves by 180 degrees; after the completion of 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 feedrate override setting for this with the parameter setting.

Optional parameter (MULTIPLE MACHINING) C-axis center override (%)

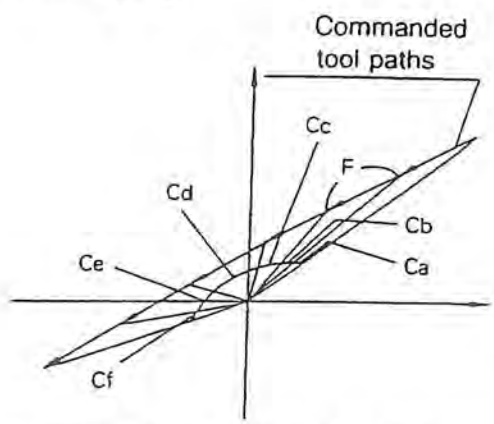
Override value for C-axis feed during G101 mode operation is set.

Setting range : 1 – 1000 (Unit: %)

Initial value : 100 (%)

(b) 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.



For the commanded feedrate F, C-axis feedrates change from Ca, Cb, Cc and to Cf. In this case, the C-axis feedrate is the maximum at Cd.

The 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 setting proper data for the parameter.

Optional parameter (MULTIPLE MACHINING) Auto limit for C-axis feedrate

0 . . . . Automatic limitation function effective (feedrate changes)

1 . . . . Automatic limitation function ineffective

Initial setting : 0

(c) In the G101, G102, and G103 mode, C-axis rotating direction is determined by the control according to the programmed shape, disregarding of M15 or M16.

(d) An alarm occurs if a C-axis command is designated in the M109 or M147 mode.

(e) 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 commands. In this case, divide an arc to make a program.

If the G102 or G103 block does not contain an L command, L value is not positive, or L is too small to define an arc, an alarm occurs.

(f) In the G102 or G103 mode, Z-axis control is not possible.

An alarm occurs if a Z-axis command is specified.

(g) 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.

(h) 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 the following alarm message is displayed.

2. Contour Generation Programming Function (Side)

(1) Function 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. 19-1 and 19-2, and the other is the "inner plane" as shown in Figs. 19-3 and 19-4.

The plane used for programming, that is, the outer plane or inner plane, can be selected optional parameter (MULTIPLE MACHINING) Z-CE coordinate screen

To select the "outer plane", set "0".

To select the "inner plane", set "1".

<Outer Plane>

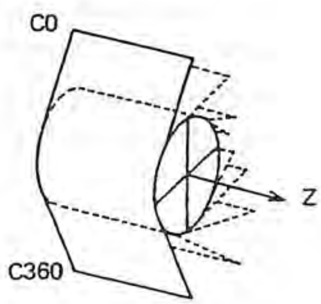


Fig. 19-1

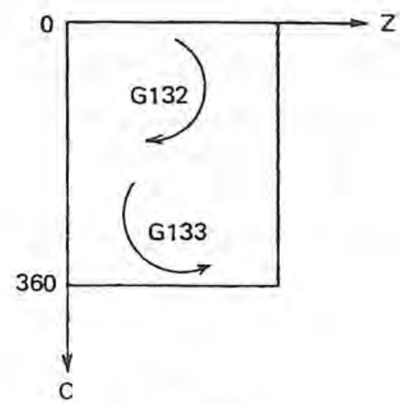


Fig. 19-2

<Inner Plane>

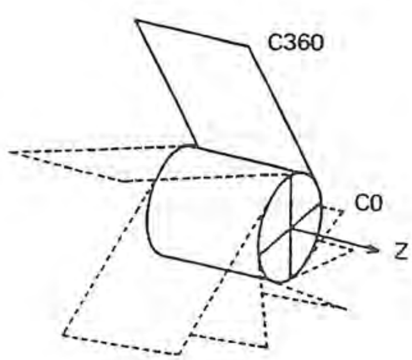


Fig. 19-3

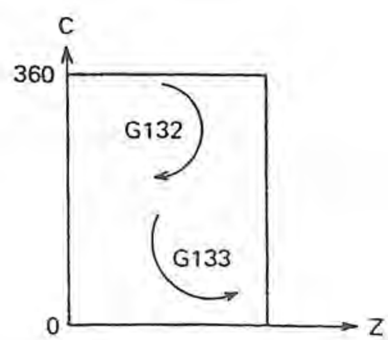


Fig. 19-4

The circular interpolation direction, tool nose radius compensation direction, and others are determined based on the selected plane.

(2) Programming

Format: G132 Z C L F

Z :
C : } End point for circular interpolation (CW) on contour generation side face

L : Radius of arc on side face

F : Cutting feedrate (mm/min)

Format: G133 Z C L F

Z :
C : } End point for circular interpolation (CCW) on contour generation side face

L : Radius of arc on side face

F : Cutting feedrate (mm/min)

(3) Cautions

(a) 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.

(b) For circular interpolation between two points A and B on the side face, there are two possible paths which have the same radius. (The interpolation arcs having the center angle of less than 180°.)

In Fig. 19-5 below, the arc "a" is generated.

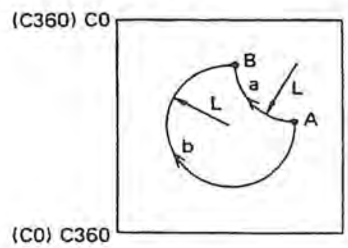


Fig. 19-5

The values in parentheses are for the inner plane.

(c) 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 an rotary axis and the coordinate value is continuous at every 360 degrees.

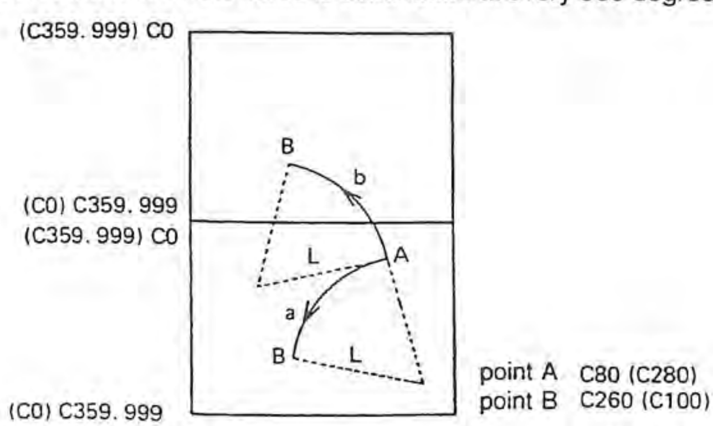


Fig. 19-6

In such a case, an arc is generated according to M15/M16 (C-axis forward/reverse rotation command) designated preceding the arc command.

Arc 2) is generated when M15 is designated.

Arc 1) is generated when M16 is designated.

NOTICE : When the C-axis is connected, machining is possible within the range the C-axis is rotated 5965 turns (5965 turns in the case of 0.1 μm specification) in one direction. If side contour generation machining is carried out exceeding this limit, the following alarm message is displayed.

Alarm B 2480 Profile generation calculation.

If this alarm message is displayed, write the program taking the following into consideration.

Use the side contour generation programming mode function.

(1) Function Overview

The side contour generation programming mode function is valid when "1" is set for optional parameter (bit) No. 39 bit 0.

Optional parameter (bit) No. 39 bit 0.

1 .... Side contour generation programming mode function Valid

0 .... Side contour generation programming mode function Invalid

Initial setting is "0".

(2) Side Contour Generation Programming Function 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 used as the cancel code for G137 (coordinate conversion ON)
- Designation of M109 (C-axis control OFF)
- Reset

(3) Restrictions

When the side contour generation programming mode function is set 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 in other than the side contour programming mode, the following alarm message is displayed.

Alarm B 2224 UNUSABLE contour generation command

SECTION 20 COORDINATE SYSTEM CONVERSION

(1) Function Overview

Multiple-machining models are provided with the function which converts the program commands designated in the Cartesian coordinate system into * and C-axis data in the polar coordinate system on-line

This function is provided in order to simplify 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.

Program format

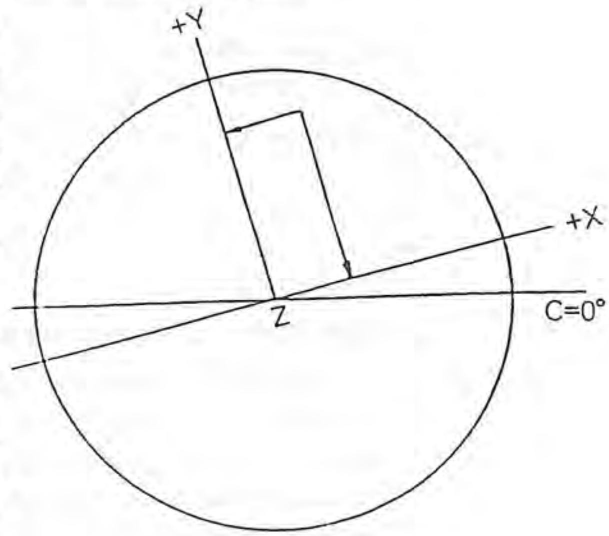
(a) Start of coordinate system conversion

C: Angle of C-axis that defines the orthogonal coordinate system

(b) Cancel of coordinate system conversion

When G137 has been 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 given 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 in radius. Give a plus (+) or minus (–) sign preceding the X and Y words of the specified Cartesian coordinate system.



(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.

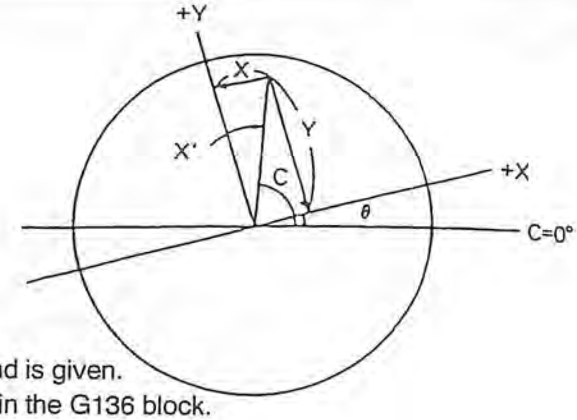
(2) Conversion Format

X' : Radius vector

C : Angle

$X' = \sqrt{X^2 + Y^2}$

C : $\tan\left(\frac{-1Y}{X}\right) + \theta$



(3) Program Examples

G136 : G137 is effective, until this command is given.
Do not designate other commands in the G136 block.

G137 : G137 C ****
Designate the angle in reference to the C-axis zero point as angle C. This angle is equivalent to "θ" in the illustration above.

After designation of G137, use X and Y words as positioning commands instead of X and C words until G136 is designated.

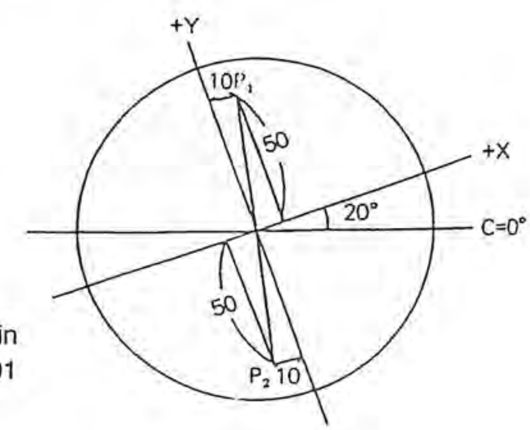
Program 1:

```
      ∫
N01   G137   C20
N02   G181   X10   Y50         Z125   Q2   F30   K10
N03   G180
N04   G136
```

The above is a programming example in which a compound fixed machining cycle is performed.

Program 2:

```
N011   G37   C20
N012   G00   X-10   Y-50
N013   G94
N014   G01   Z125   F30
N015   G00   Z150
N016   G136
```



The above is a programming example in which machining is performed in the G01 mode at P2.

Note: Use X and Y words only for positioning.

(4) Supplement

It is possible to select whether or not the C-axis zero shift is included when creating the orthogonal coordinate system by the coordinate system conversion command.

OPTIONAL PARAMETER (MULTIPLE MACHINING) C-axis zero shift in G137

0 . . . . . C-axis zero shift data is invalid

1 . . . . . C-axis zero shift data is valid

CAUTION : (1) Designate both X and Y words in the first block following the G137 block. When only 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.

1) Cancel the incremental programming mode in the block before the G137 block.

2) Designate X and Y words in the absolute programming mode in the first block following the G137 block.

3) Designate the incremental programming mode.

Program example

```
G91                 ....... Incremental programming mode ON
 ∫                          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 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 21 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 of turret A (upper turret) and turret B (lower turret), the turret for which a 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

A G code used for selecting a 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 the other turret 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
```

Turret selection G codes may be specified in a program as many times as necessary. Programs written under each turret selection G code are separated into the program for G13 and those for G14 to be executed.

1-2. Synchronization P Code

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.

Program Format P_ _ _ _

P: Integer (up to four digits)

P codes control the execution order of the G13 side and G14 side programs. Program is executed in order assigned with a smaller P code number. 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 other side program, P code numbers are compared and the program of a smaller P code number is executed. If the P code numbers are same, both programs are executed. If execution of program for one side is completed while the other side program is suspended, execution of that program starts.

Example:

Spindle rotation commands, spindle speed commands, and gear range selection commands that must be synchronized between the programs of 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
                  S1500  M42    M03    P10   ...... A
G00  X130  Z150
G01  X20          F2.5                 P30   ...... C
                         M05           P40   ...... D
M02
```

In the example program above, blocks A are 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.

Example Program:

```
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
     X92   Z65
M100
G01  X80   Z59   F0.25
                   :
```

M100 command matches timing of cuts.

Points to consider when Using the M100 Command:

The following points should be considered when synchronizing operations with the M100 command.

NOTICE :

(1) The synchronization of S and M commands cannot be performed 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 into 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

```
N0000  G13
N0001  G00   XOOOOO  ZOOOOO  POOOO  SOOOO  TOOOO  MOO
        :
        :    Cutting program for turret A
        :

N0049
N0050  G14
N0051  G00   XOOOOO  ZOOOOO  POOOO  SOOOO  TOOOO  MOO
        :
        :    Cutting program for turret B
        :

N0099
N0100  G13
N0101  G00   XOOOOO  ZOOOOO  POOOO  SOOOO  TOOOO  MOO
        :
        :    Cutting program for turret A
        :

N0150  G14
N0151  G00   XOOOOO  ZOOOOO  POOOO  XOOOO  TOOOO  MOO
        :
        :    Cutting program for turret B
        :
```

G13 selects the turret A and G14 the turret B.

Program from block N0001 to N0049 is for cutting to be performed by the tools on the turret A, and the one from block N0051 to N0099 is for the tools on the turret B.

Note 1: *While simultaneous 4-axis control mode, S command, M commands related to spindle rotation (M00, M01, M03, M04, M05 and M41 through M44), and G96 calling for constant speed cutting mode must match for turrets A and B. Otherwise, an alarm results.*

Note 2: *If G13 and G14 code selecting the turret are not specified, the machine fails to perform the operation intended.*

Note 3: *The blocks dominated by respective G codes, G13 and G14, are continuous as a program. That is, N0101 directly follows N0049 and N0151 follows N0099. Therefore, when S, T, and M commands in those successive blocks are the same as provided in N0001 and N0051, respectively, they can be omitted.*

Note 4: *The block containing S and M commands (M41 through M44, M00, M01, M03, M04 and M05) of turrets A and B, or G96 code must be provided with a P command having the same number (up to four digits) to synchronize the execution of the commands in those blocks on turrets A and B.*

When synchronization of command execution on the two turrets is required, use the P command.

Example:

```
%
O0100
N0000  G13
N0001  G00  X800    Z800    P10     M41   S120  T0101
N0002       X1      Z1      P20     M03
N0003       ......  ......  ......
  :
N0050  G01  X2      Z2      P30     F0.4
  :
  :
N0098  G00  X800    Z800
N0099                       P40     M05
N0100  G14
N0101  G00  X800    Z800    P10     M41   S120  T0101
N0102       X3      Z3      P20     M03
N0103       ......  ......  ......
  :
N0105  G01  X4      Z4      P30     F0.3
  :
  :
N0200  G00  X800    Z800
N0201                       P40     M05
```

P10 in N0001 and in N0101 synchronizes execution of M41 S120 in those blocks.

P20 in N0002 and in N0102 synchronizes execution of M03.

P30 in N0050 and in N0150 synchronizes start of cutting.

P40 in N0090 and in N0201 synchronizes execution of M05.

If P number in block N0002 is taken as P200, i.e., if 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 with 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, the P number must be assigned orderly as P10, P20 and P30 according to the order of the command execution.

It is recommended to use two or three digits as a P number instead of using one digit for ease of correction of programs.

P10 instead of P1

P20 instead of P2

3. Precautions for Programming Simultaneous 4-axis Cuts

The key to efficient simultaneous 4-axis cuts on 2S model is if intended cutting is performed with well balanced.

When programming simultaneous 4-axis cuts, observe the following points carefully:

(1) Determine the extent of operations to be performed by turrets A and B, respectively.

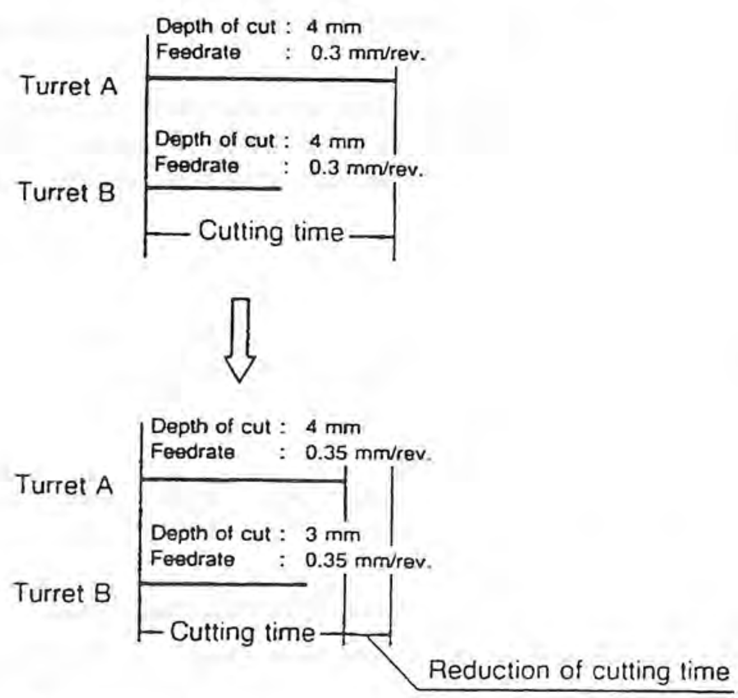
Cutting times required for these two turrets should be well matched when determining sections to be cut by respective turrets.

(2) Determine optimum cutting conditions.

(a) Since spindle change cannot be performed while simultaneous 4-axis cut, cutting speed will vary according to the diameters being cut. Select the tip material carefully to meet the workpiece material to be cut.

(b) Select feedrate and depth of cut taking cuttings on two turrets into account:

Example:



(c) Determine the cutting conditions so that a total of cutting power required by the two turrets will not exceed the capacity of the machine.

(3) Others

(a) The use of the INDIVIDUAL switch allows the independent operation of the turret facilitating check of trial cut.

(b) Care should be exercised on interference.

- Interference between the boring bar and the chuck
- When end face cutting is performed with the tools on turret A:

  Interference between;

  Tools on turret A and boring bar on turret B,

  Tools on turret A and ID toolholder on turret B

(c) Program movements of the tools on turret B assuming those on turret A.

G02 and G03 should also be determined assuming cutting with the tools on turret A.

(d) In constant speed cutting mode operation called for by G96, G110 and G111 selects 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 to select constant speed cutting mode on turret A.

This feature will generate large difference in cutting speeds for the tools on turrets A and B when cutting a workpiece having large difference in the diameter, for instance. Therefore, cutting portion for respective turrets and cutting tip material should be selected very carefully.

Example:

```
        :
G13
G00                X1   Z1   P10   S120   M41   M03   T0101
G96   G01   G110   X2   Z2   P20   S100
        :
        :
G14
G00                X3   Z3   P10   S120   M41   M03   T0101
G96   G01   G110   X4   Z4   P20   S100
        :
```

For turrets A and B, G96, S, M and P commands must match.

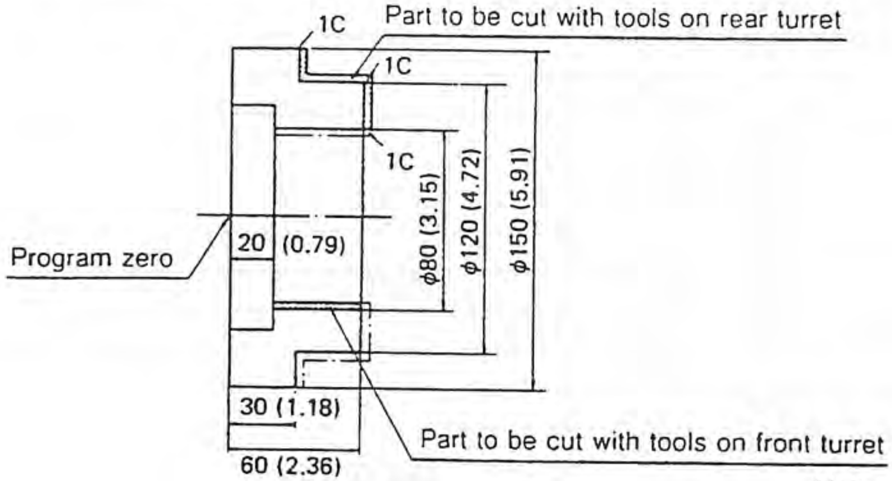
Even in constant speed cutting mode, it is active only on turret A and turret B is not in such mode.

4. Programming Example

4-1. Workpiece Dimensions

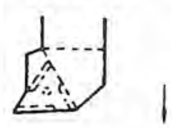
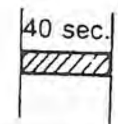
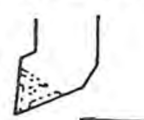
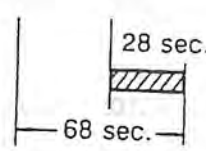
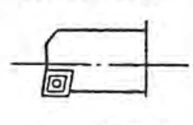
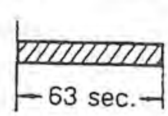
Material : S45C (JIS, carbon steel)

Stock : 3mm (in radius)



Unit: mm (in.)

4-2. 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.	
	T0202	Upset tool  OD turning	Cutting speed: 95 m/min. Depth of cut: 3 mm Feedrate: 0.4 mm/rev.	
B	T0101	Normal tool  ID turning	Cutting speed: 65 m/min. Depth of cut: 3 mm Feedrate: 0.25 mm/rev.	

The net cutting time per piece is 68 seconds when the part is cut in 4-axis simultaneous cut mode. It will be 131 seconds if the part is cut in conventional manner. This means that simultaneous 4-axis cut yields nearly 48% saving on cutting time.

4-3. Program Process Sheet

%					
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		T0202
N011	G01	X120	Z59	F0.4	
N012			Z30		
N013		X130			
N014	G00	X800	Z70		
N015				P20	M09 M05
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

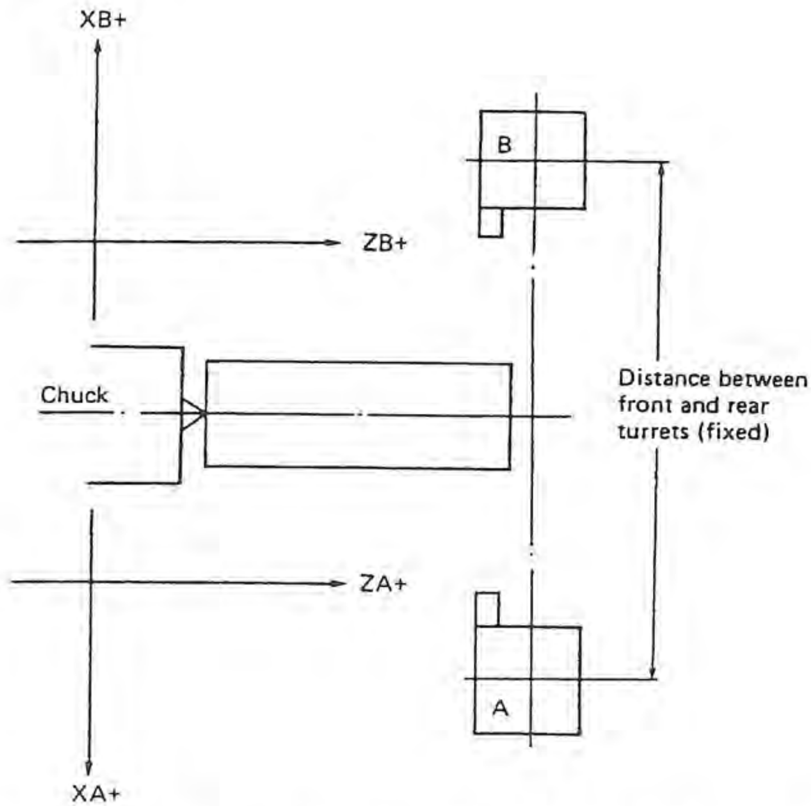
O100: Program name
N000: Selection of turret A
N001: End face cutting with the tool on turret A
:
N009
N010: OD turning with the tool on turret A
:
N014
N015: Since no P20 command is presented in a program executed by the tool on turret B, this block is executed by turret A only.
N100: Selection of turret B
N101: ID turning with the tool on turret B
:
N106

With the program above, simultaneous cut on end face and OD turning by turret A and ID turning by turret B is performed.

SECTION 22 MIRROR IMAGE FUNCTION (2-Turret Model)

1. Outline of Mirror Image Functions

The mirror image function allows the programmer to program cutting with turret B in the same manner as programming with turret A.



With the two-turret specification machines, turret A and turret B are arranged in both sides of the spindle centerline. Since these turrets are mounted on the same one saddle, they cannot be controlled independently, but move simultaneously with separated by the fixed distance.

It is the Mirror Image function that can simplify programming for two-turret model and help the machine to demonstrate its best performance.

2. Operations

If the programmer programmed a part assuming turret A, the mirror image function automatically converts the X-axis moving direction, spindle rotation direction and direction of arc cutting when turret B is selected. Therefore, the same workpiece can be cut as finished by turret A tools.

3. Program Axis Motions

3-1. G Codes

(1) Selecting Turret

(a) Selecting turret

G codes are used to select the turret.

G13	Turret A
G14	Turret B

(b) Selecting tool

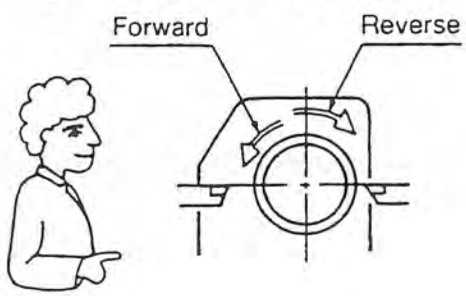
When selecting #4 tool on turret B, program:

```
G14
G00    X Z T0404
```

When selecting #3 tool on turret A, program:

```
G13
G00    X Z T0303
```

(2) Programming Spindle Rotating Direction



When turret B is selected, the mirror image function starts the spindle in the reverse direction when M03 "spindle forward" command is executed.

Turret A: G13 M03

Turret B: G14 M03

When the turret is changed, while the spindle rotation does not change until a new spindle rotation command is provided.

3-2. Cautions on Programming

(1) G codes selecting the turret, G13 and G14, must be designated in a block without other commands.

If other commands are designated in the block containing G13 or G14, such commands are ignored.

(2) Resetting the control automatically selects G13 and cancels G14.

(3) Switching from G13 to G14 or from G14 to G13 is ineffective in the following cases.

If intended, ALARM-B "change timing G13/G14" results.

(a) While G41 or G42 active (tool nose radius compensation mode)

(b) While G91 active (incremental programming mode)

(c) While LAP2 active

(d) While G96 active (constant cutting speed mode)

If the change of the turret is required while the operation mode as indicated above, cancel the mode once before programming turret selecting G code.

(4) When switching of the turret is executed, commands other than X and Z words are all canceled. However, tool number and tool offset number that are active when switching is executed remain active.

(5) G and M codes that are reversed by mirror image function are:

G02 → G03

G41 → G42

M03 → M04

Note that these codes remain effective after switchover between G13 and G14 has been executed (switchover is impossible while G41 or G42 is active).

<Program Example>

N001	G14			
N002	G00	X_2	Z_2	M03
N003	G13			
N004	G00	X_2	Z_2	M03
:	:	:	:	:

With the program above, the spindle rotates in the reverse direction when M03 in N002 is executed. M03 in N004 calling for spindle forward direction rotation causes ALARM-B "change timing G13/G14".

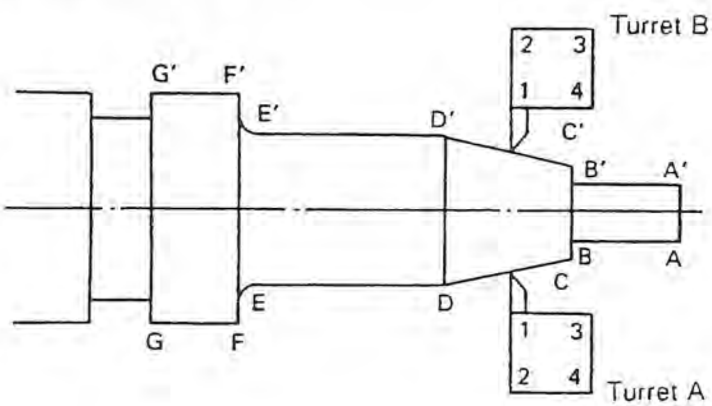
(6) When a programmed cycle is to be executed repeatedly using the "GOTO" statement, care should be exercised on the following point:

<Program Example>

N001	G00	X	Z_1	
N002	G01	X_2	Z_2	F.S.T.M
N003	G14			
N004	G00	X_3	Z_3	
N005	GOTO			N001

When the program above is executed, the first cycle of the commands in blocks N001 and N002 are executed with the tool on turret A (G13): in the second and later cycles, those commands are executed with the tool on turret B (G14). This is because GOTO N001 does not reset the control and G14 designated in N003 remains effective for the second and later cycles. It is necessary, therefore, to designate G13 or G14 in the first block of repetition when the cycle is repeated using GOTO statement.

3-3. Cutting Program



An example of a program for the workpiece as shown above is provided.

When cutting is carried out by the tool on turret A, cutting edge of the tool traces the path A-B-C-D-E-. To the contrary, when cutting is carried out by the tool on turret B, it traces the path A'-B'-C'-D'-E'-.

<Program Example>

N001	G14						
N002		G00	XA	ZA	S300	T0101	G14 is necessary when turret B is used.
N003		G01		ZB	F0.2	M03	
N004			XC				
N005			XD	ZD			
N006				ZE			
N007		G02	XF	ZF	I	K	
N008		G01		ZG			
N009		G00	XH	ZH		M05	

3-4. Cutting Operation

Axis and turret operations are explained below for turret A and turret B, respectively.

(1) Turret A

When the program does not contain G14, cutting program is executed by the tool(s) on turret A (G13).

The commands in N002 select tool number 01 and tool offset number 01 of turret A. N03 in N003 rotates the spindle in the forward direction and cutting is performed along the path A-B-C ... as shown in the figure in 3-3.

(2) Turret B

When the program contains G14, cutting program is executed by the tool(s) on turret B.

The commands in N002 select tool number 01 and tool offset number 01 of turret B, M03 in N003 rotates the spindle in the reverse direction and cutting is performed along the path A'-B'-C'... as shown in figure in 3-3. For arc cutting called for by the commands in N007, direction of cutting is also reversed.

4. Others

(1) Designation of Turrets A and B

On lathes having flat bed, models LS30-N, LH35-N and LH55-N, the front turret is the turret A and the rear the turret B.

On lathes having slant bed, the upper turret is the turret A and the lower the turret B.

(2) Sequence Re-start

On two-turret model, there are following restrictions on the turret for sequence re-start operation.

The turret on which sequence re-start is active is turret A when the block where sequence re-start is to be performed is in G13 mode, and turret B if it is in G14 mode. Sequence re-start is not active on the other turret.

It will be very dangerous if sequence re-start is activated on both turrets on two turret model.

Therefore, it is necessary to follow the steps below to activate sequence re-start on both turrets.

<Program Example>

N001	G13			
N002	G00	X_1	Z_1	T0101
N003	G14			
N004	G00	X_2	Z_2	T0101
N005	G13			
N006	G00	X_3	Z_3	
N007		X_4	Z_4	T0202
N008	G14			
N009	G00	X_5	Z_5	
N010		X_6	Z_6	

When sequence re-start is activated from N007, T1 on turret A is indexed since block N006 is in G13 mode.

When sequence re-start is activated from N010, T1 on turret B is indexed since block N009 is in G14 mode.

To activate sequence re-start from N010 with proper tool number on turret A indexed, index the turret A to the desired position by entering the proper T command through the MDI keyboard and then perform sequence re-start operation from N010.

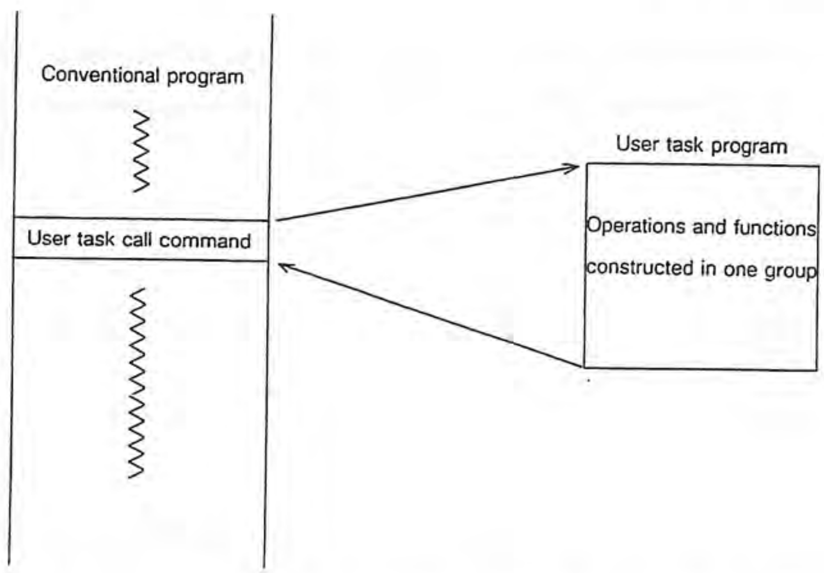
Note: *Turret indexing position is different from the programmed position.*

SECTION 23 USER TASK

1. Overview

Operations and functions constructed as one group of instructions are stored in the memory when assigned with a program name like a subprogram. The stored subprogram can be accessed from the main program by specifying the program name representing a group of instructions and the operations and functions in that program can be executed.

A group of operations and functions to be stored is called User Task Program and the command to call it is User Task Call Command.



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 assure versatility in the user task function.

There are many fields where user task function can be effectively used; among them, the user task function in the following fields is very advantageous.

(1) The same contour is repeatedly specified in cutting a part such as a pulley.

(2) Gears and Flanges which have Similar Contours

By picking up common and similar contour elements of parts to be cut according to Group Technology, express those elements using variables. User task program is programmed using the variables and actual dimensions of a specific part to be cut are provided in a cutting (or main) program. Thus the parts having the similar contour can be machined using one user task program.

(3) Automatic Cycle Involving Peripheral Equipment and/or Functions

Instructions necessary to interlock the machine cycle with bar feeder or loader cycle, or work loader/unloader, work gauging cycle commands, instructions for machine interlocked operation with robot or other peripheral equipment are programmed as user task programs. Activate this user task function when operating the machine with peripheral equipment and/or functions.

Peripheral equipment and functions indicated above are possible for a user a special function or operation by using the user task function.

(4) Parts having 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 if those points can be calculated by several expressions, user task program for these parts can be programmed using expressions.

(5) Special Fixed Cycle for a User

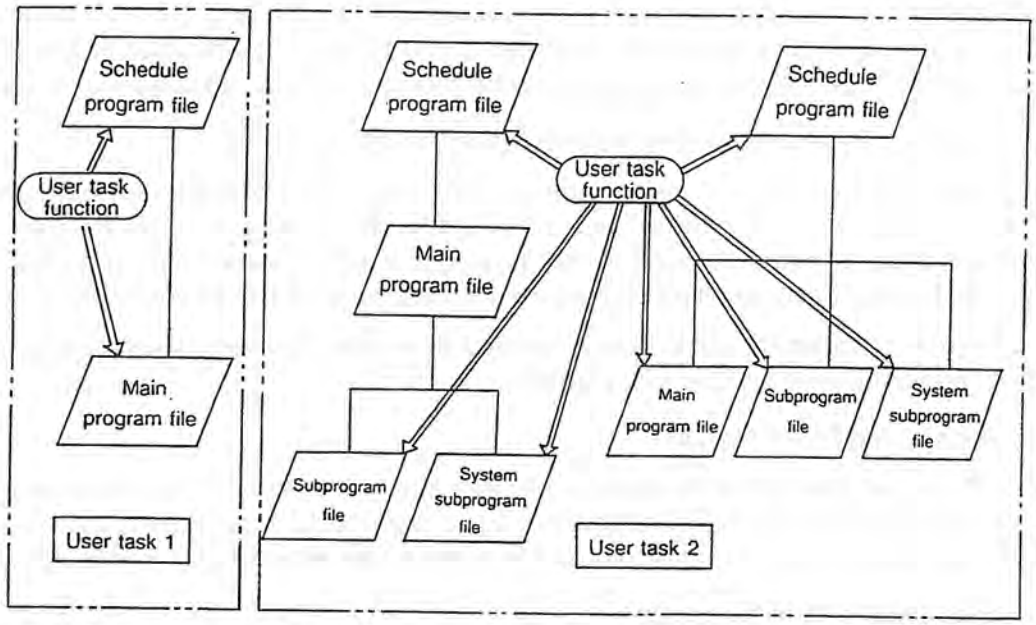
When special fixed cycles or custom cycles are required to be made by users.

As stated above, effective use of user task function can realize 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. Relation between Types of Program Files and User Task Functions

Types of program files are discussed in II OPERATION. Shown below summarizes the relation between the types of program files and user task functions.



Programs in the left consist of schedule program and main program; User Task function can be used in these two types of programs. User Task function applicable in such construction is called "User Task 1".

The case shown in the right is made from all types of program files and User Task function can be used in every type of program. User Task function applicable in such construction 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 variable 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'
	Function		'SIN', 'COS', 'TAN', 'ATAN', 'ATAN2', 'SQRT', 'ABS', 'BIN' 'BCD', 'ROUND', 'FIX', 'FUP', 'DROUND', 'DFIX', 'DFUP', 'MOD'

[Supplement] Program either a space or a tab code following control statements indicated below.

'GOTO', 'CALL', 'RTS', 'MODIN', 'MODOUT'

3. Fundamental Functions of User Task

There are three fundamental user task functions as indicated below:

(1) Control Statement Functions

The statements to control the execution order of programmed sequences, such as "IF", "GOTO" and "CALL" which are easily understood, can be used.

(2) Variables

In normal programming, numerical data directly follows the address characters such as A through Z. Instead of numerical data, a variable expressed by alphanumerics can be assigned to the address characters, and actual numerical data are assigned to the variables in respective programs. This feature provides program versatility and flexibility.

Example:

Word used in conventional program:	When a variable is used:	
X135	X = XP1	XP1 = 135
	Variable name (alphanumeric)	Numerical data is assigned.

(3) Arithmetic Operation Function

This function allows arithmetic expressions to be directly programmed as a word data.

Example:

Word used in conventional program:	When arithmetic operation function is used:
X135	X = 100 + XP2 XP2 = 35

3-1. Control Statement

User Task 1 can use following six control statements:

Statement		
IF statement	User Task 1	User Task 2
GOTO statement		
CALL statement		
RTS statement		
MODIN statement		
MODOUT statement		

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 provide either a space or a tab code following the sequence name or a control statement as a delimiter. If not, an alarm occurs.

For "IF" statement, it is not necessary to provide a space or a tab since it is followed by "left bracket - [".

Example:

N1001 GOTO␣ N2000

A space or a tab code

NLAP1␣ GOTO␣ NLAP2

The element consisting of more than one address characters (A through Z) such as a sequence name and a control code must be followed by either a space or a tab code.

*1: Sequence Name

A sequence name is the code to identify respective blocks in a program, and it consists of four alphanumerics following address character N.

There are two types of sequence names:

<N> <four numerics>, and

<N> <letter> <three alphanumerics>

"Sequence name" handled in this manual refers to both types of sequence names.

(1) GOTO Statement - Unconditional Jump

(a) Program format

<N0> ␣ <GOTO> ␣ <N1>

- <N1>: Sequence name of the block where jump is to be made and it must be specified without fail.
- <GOTO>: Indicates "GOTO" statement.
- <N0>: Sequence name of this block and it can be omitted.

- Sequence name <N1> for jump must be the one in the program containing such control statement.

(b) Function

When the control statement is executed, jump to the programmed block, <N1> is made unconditionally.

(2) IF Statement - Conditional Jump

(a) Program format

1) <N0> ␣ <IF> ␣ <Conditional expression (qualification)> ␣ <GOTO> ␣ <N1>

- <N1>: This is the <sequence name> to which jump is made when <qualification> is true. This must be programmed without fail.
- <GOTO>: Jump instruction to be executed when <qualification> is true. May be omitted.
- <Conditional expression (qualification)>: There are two possible states for the result of the <qualification>, "true" and "false".

  <qualification> must be expressed in [ ].
- <IF>: Indicates "IF" statement.
- <N0>: Sequence name of this block and it can be omitted.

1) <N0> ␣ <IF> ␣ <Local variable> ␣ <GOTO> ␣ <N1>

- <N1>: The <sequence name> of the destination block of jump to be executed when the local variable is defined.
- <GOTO>: Jump instruction to be executed when the local variable is defined.
- <Local variable>: This can be used instead of <qualification>.

  Jump does or does not occur depending on whether the local variable is defined or not.

(b) Function

1) When the <qualification> is true (Example 1) or when the local variable is defined (Example 2), execution of the sequence jumps to the sequence <N1>.

2) When the <qualification> is false (Example1) or when the local variable is not defined (Example 2), the following sequence is executed.

Example 1:

N1000␣ IF [V1␣ EQ␣ 10]␣ GOTO␣ N2000 or
N1000␣ IF [V1␣ EQ␣ 10]␣ N2000

Means "equal (=)".

Jump is made to N2000 when 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 or
N1000␣ IF␣ ABC␣ N2000

Local variable

If Jump is made to N2000 when local variable ABC has been defined. If not, the following block is executed.

(3) Call Statement - Calling Program

(a) Program format

<N0>␣ <CALL>␣ <O1>␣ <Q1>␣ <Variable setting section>

Assign numerical data to the variables used in the subprogram to be called.

Indicates the number of repetitions of the subprogram to be called.
(1 through 9999 following address character Q)
When omitted, the control assumes Q1.

Program name of the subprogram to be called.
This must be specified without fail.

Indicates "CALL" statement.

Seauence name of this block and it can be omitted.

(b) 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

With the designated commands above, the subprogram O1234 is called and executed. At the same time, variables XP1 and ZP1 are registered.

(4) RTS Statement - End of Subprogram

(a) Program Format

<N0> ⌴ <RTS>

- <RTS>: Indicates "RTS" statement.
- <N0>: Sequence name of this block and it can be omitted.

(b) Function

This RTS statement must always be specified at the end of the subprogram; this declares the end of the called subprogram and execution sequence jumps to the block right after the one containing CALL statement. The variables registered in the block containing CALL statement and the variables in the subprogram are all cleared.

<N2> ⌴ <IF> ⌴ <Local variables name> ⌴ <GOTO> ⌴ <N3>

- <N3>: This is the <sequence name> to which jump is made when the designate variable has been defined.
- <GOTO>: This can be omitted.
- <Local variables name>: Local variable name to be judged whether it has been defined or not. It is not necessary to use [ ] to express a local variable name.
- <IF>: Indicates "IF" statement.
- <N2>: Sequence name of this block.

This checks whether the local variable name designated is defined or not. Jump is made to the designated sequence name, N3, when it is defined and the block which follows this N2 block will be executed if the designated local variable is not defined.

Example:

N1000 ⌴ IF ⌴ ABC ⌴ N2000

If local variable ABC has been defined, jump to N2000 sequence is made. If it has not been defined, the next block is executed.

Example:

```
Main Program
    N1000 CALL O1234 XP1 = 150 ZP1 = 10
    N1001
      :
Subprogram
    O1234
    N001  G00  X Z
      :    :
    N050  RTS
```

When block N1000 in the main program is executed, jump to subprogram O1234 is made.

The subprogram is executed from N001 and when the control reads the RTS statement in N050, jump is then made to N1001 of the main program and the commands in that block and those after that are executed. At the instance the jump from the subprogram to the main program is made, variables XP1 and ZP1 are cleared.

Order of program execution:

$\boxed{\text{N1000}} \rightarrow \boxed{\text{O1234}} \rightarrow \boxed{\text{N001}}$ .... $\boxed{\text{N050}} \rightarrow \boxed{\text{N1001}}$ ....

(5) MODIN Statement

(a) Program Format

<N0> ⌴ <MODIN> ⌴ <O1> ⌴ <Q1> ⌴ <Variable setting section>

- <Variable setting section>: Assign numerical data to the variables used in the subprogram to be called.
- <Q1>: Indicates the number of repetitions of the subprogram to be called. (1 through 9999 following address character Q)
  When omitted, the control assumes Q1.
- <O1>: Program name of the subprogram to be called. This must be specified without fail.
- <MODIN>: Indicates "MODIN" statement.
- <N0>: Sequence name of this block and it can be omitted.

(b) Function

The designated subprogram is called each time axis motion command(s) is executed. That is, the designated subprogram is called and executed each time axis motion command (*1) in the program calling that subprogram is executed. This function remains active until the MODOUT statement detailed in (6) is read.

*1: Axis Motion command means the command in G00, G01, G03 and G31 through G35 modes which contain either or both of X and Z words.

(6) MODOUT Statement

(a) Program Format

<N0> ⌴ <MODOUT>

- <MODOUT>: Indicates "MODOUT" statement.
- <N0>: Sequence name of this block and it can be omitted.

(b) Function

This is the statement to cancel the MODIN mode.

(c) Program example of MODIN and MODOUT statements

```
        Main Program                          Subprogram
N001    G00    X  Z
  :
  :
N030
N031    MODIN    O1000                        O1000
N032    X        Z                            N1001  G01      X  Z  F
N033    X        Z                            N1002
N034             Z                              :
N035    X        Z                              :
  :                                             :
  :                                           N1040  RTS
N040    MODOUT
  :
  :
N060    M02
```

Order of program execution:

N001 N030 N031 N032 O1000 N1040

N033 O1000 N1040 N034 O1000 N1040

N035 O1000 N1040

Main and subprograms are executed repeatedly.

N040 Main programs is executed. N060

The program is started from N001 of the main program and the commands up to N030 are executed in ordinary manner. With the commands in N031 executed, the control is in the MODIN mode calling the subprogram O1000. In this block, however, the subprogram is not executed. When the axis motion commands in block N032 are completed, the subprogram O1000 is called and executed up to N1040 in that subprogram. The RTS statement causes jump to the main program and N033 in the main program 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 ordinary manner.

[Supplement] 1. Nesting and effective range of MODIN/MODOUT mode

Permissible nesting level in MODIN mode is eight.

Example:

```
        Main Program                              Subprogram

N001    MODIN       O1000                         O1000
 :                                                N1001   G00 X   Z
 :                                                  :
N010    MODIN       O2000                         N1010   RTS
N011    G00 X   Z
N012      :                                       O2000
 :        :                                       N2001   G00 X   Z
 :                                                  :
N020    MODOUT                                    N2010   RTS
 :
N030    MODOUT
```

The example above shows the nesting of two levels.

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 axis motion command is executed.

2) In the blocks from N010 to N020, subprogram O1000 is called and executed first after axis motion command is executed. Then, subprogram O2000 is called and executed in succession. In case the subprogram O2000 contains axis motion commands, N2001 in this example, O1000 is executed after such 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 axis motion commands is executed.

[Supplement] 2. The MODIN mode must be canceled by the MODOUT statement designated in the same program. That is, the MO DIN mode activated in the main program cannot be canceled by the MODOUT statement in the subprogram, or MODIN mode active in the subprogram cannot be canceled by the MODOUT statement in the main program.

3. Maximum number of subprograms usable in a single program is 125.

(7) READ/WRITE Statement

READ and WRITE statements are used for communications with external devices through the RS232C interface.

They are used in conjunction with GET and PUT statement explained in (8) below.

(a) Program format

<NO> <READ> ␣ <n>

- <n> — Channel designation for RS232C interface

  n = 0 ... CN0: or TT:
  = 1 ... CN1:
  = 2 ... CN2:
  = 3 ... CN3:
  Default: .. n = 0
- <READ> — Indicates "READ" statement.
- <NO> — Sequence name of this block and it can be omitted.

<NO> <WRITE> ␣ <n>

- <n> — Channel designation for RS232C interface Same as READ statement
- <WRITE> — Indicates "WRITE" statement.
- <NO> — Sequence name of this block and it can be omitted.

(b) Function

READ ..... Receives the data from the external device connected to the channel designated by <n> and stores it in read area (max. of 160 characters) in the JIS8 code.

WRITE .... Sends the data stored in the write area (max. of 160 characters) in the JIS8 code to the external device connected to the channel designated by <n>.

(c) Transmission procedure

It is necessary to change the baud rate and signal construction in advance for the RS232C interface which is going to be used for the communication through RS232C interface.

These communication parameters are set on the optional parameter (RS232C) PARAMETER SET MODE.

(d) Transmission code

For transmission codes, JIS 8 bit code or JIS 7 bit code (in this case, use even parity bit) are used.

End of data transmission code is either NULL or %. Which of these codes is used is determined by the setting for optional parameter (RS232C) RS232C stop bit check.
(% is selected by "1" and NULL is selected by "0".)

(e) Others

1) The following situations 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.
- An alarm has occurred in the RS232C interface during transmission.
- RS232C is no longer in ready status.

2) Do not execute list output or punch out using the same channel of RS232C interface during the execution of the READ/WRITE command.

3) Execution of the READ/WRITE command and printing out of measured data cannot be executed at the same time.

While either of them is being execution of the other command is suspended.

4) Since the areas for READ and WRITE are different, execution of the READ command does not change the WRITE area.

5) Nothing but a sequence number may be placed before the READ/WRITE command.

6) 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.

(8) GET/PUT Statement

(a) Program format

<N0> <GET> ⌴ <s>,<ℓ>

- <ℓ>: This indicates the number of digits to be read out from the read area where the data has been stored by the READ statement. (Max. of 10 digits)

  If "s" equals "0", this indicates the number of characters to be skipped. (Max. of 160 characters)
- <s>: The variable name used to store the data in the read area where the data have been stored by the READ statement. (common variable, system variable, local variable output variable)
- <GET>: Indicates "READ" statement.
- <N0>: Sequence name of this block and it can be omitted.

<N0> <PUT> ⌴ <s>,<ℓ>

- <ℓ>: This indicates the number of digits to be written to the Write area using the WRITE statement. (Max. of 10 digits)

  If "S" equals "0", this indicates the number of space codes. (Max. of 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>: Indicates "PUT" statement.
- <N0>: Sequence name of this block and it can be omitted.

(b) Function

GET . . . . . 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 to the variable designated.

PUT . . . . . This stores the numerical data of the variable and a character string designated in the write area output by the WRITE statement. The data is stored in the JIS8 code.

(c) Function details

GET . . . . . First, the code read out in the READ area by the READ statement is read out as many as the number of the characters designated in <ℓ> from the position of the area read pointer (to be referred to as RRP later). Then, the read out data is converted into numerical values and set in the type of variable designated in <s>. At this moment, RRP is added by <ℓ>.

RRP is set at the beginning of the READ area when the READ statement is executed or the NC is reset, and the data is added when GET is executed. Returning is impossible.

Alarm B occurs in the following cases:

- When RRP exceeds the number of codes read by the READ statement

- When conversion into numerical data is impossible

Example: When there are more than two decimal points or code other than 0 through 9 exists

PUT . . . . . When common variable or local variable is designated in <s>, the real type is used, and when input variable is designated, the integer type is used, and 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 in the WRITE area for as many as the number of digits designated in <ℓ> from the position of the write-in pointer of the WRITE area (to be referred to as WWP later). At this moment, WWP is added by <ℓ>.

WWP is set at the beginning of the WRITE area when the WRITE statement is executed or the NC is reset, and the data is added when PUT is executed. Returning is impossible.

Character string or hexadecimal number can be used for <s> in PUT statement. However, the limit for the number of characters to be designated is 16. <ℓ> must not be designated.

Hexadecimal number:

```
PUT    $0D0A      Follow $
```

Capital letters:

```
PUT    'ABCD'     Finish with'
```

Small letters:

```
PUT    'ABCD'     Starts with'
```

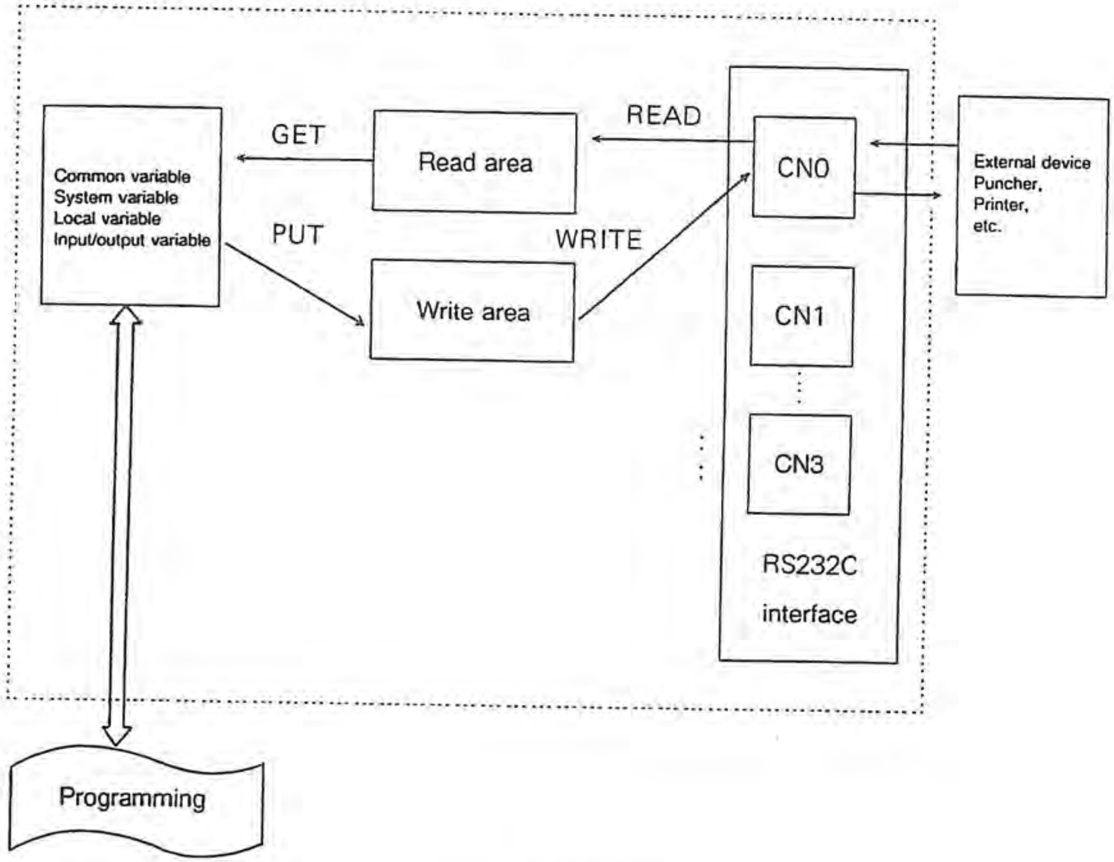
Alarm B occurs in the following cases:

- When WWP exceeds 160

- When NULL or % exists in hexadecimal number

(d) Relationship between READ/WRITE and GET/PUT

Data communications are made using the variables (common variables, system variables, local variables, input/output variables) and external devices through the read/write area.



N1 READ 0 .... Data is read from CN0:.

N2 GET␣ ,10 .... Message (a) (10 letters) is skipped.

N3 GET V1,1 .... 11th letter is read in 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

Transmission data

```
1       5          10     15       20     25    30     35   40
A Compensation Yes/No = 1 Offset No. = ␣3␣OX = 0.02␣OZ = –0.31
└──────────────────────┘ └──────────┘ └────────┘ └────────┘
          (a)                (b)          (c)        (d)
```

The result of the preceding program:

V1 = 1
V2 = 3
VTOFX[3] = 0.02
VTOFZ[3] = –0.31

Printer is supposedly 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
```

The following will be printed when the preceding program is executed.

```
␣␣␣␣**␣TOOL␣OFFSET␣DATA␣**
N = ␣3␣X = ␣␣␣5.412␣Z = ␣␣14.339
```

3-2. Variables

There are three types of variables used with User Task 2:

- Common variable
- Local variable
- System variable

These three types of variables differ in their use and characteristics.

(1) Common Variable

The term "common" of common variables can be literally understood as common; they can be used in common to 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, the common variables, the result of calculation in one program, can be referred to in other programs.

(a) Program format

<V> <a number of up to three digits> = numerical data or expression

Designation of a common variable is made with up to three digits following "V". Usable common variable is V1 through 200.

In a program, a common variable is used as:

N101 ␣ V5 = 10, or N101 ␣ V5 = V5 + 1

[Supplement]

1. Common variables are effective both in main and subprogram.
2. Common variables are not affected by control reset or turning power off. That is, the data are retained unless they are re-set or a control software is installed.
3. Common variables can be set or changed by setting a parameter besides setting or changing them in a program. For detailed information of parameter setting, refer to 3. "OPERATION IN PARAMETER MODE" of III PARAMETERS.

(2) Local Variables

As may be known form the term "local", local variables are the variables that a user can set as desired with a name easy to distinguish from each other. Up to 127 local variables can be used.

(a) Program format

<Letter> <Letter> <two alphanumerics> = Numerical data or expression

"O", "N" and "V" cannot be used.

Example: 'DIA1' 'ITH5'

The same name as used for <function name>*1, <comparison operator>*1, <boolean operator>*1, and <extended address character>*2 cannot be used.

*1: Explained in 3-3. "Arithmetic Operation Function"

*2: Extended address characters are provided to realize the LAP, pattern processing, user-specific fixed cycles. The following extended characters are used presently.
<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>

(b) Characteristics of Local Variables

1) Local variables are cleared when the control is reset.

2) 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.

NOTICE : If a local variable name is used without assigning the data, an alarm results.

3) When new data is assigned to the local variable already registered with another data, that old data is renewed.

Main program

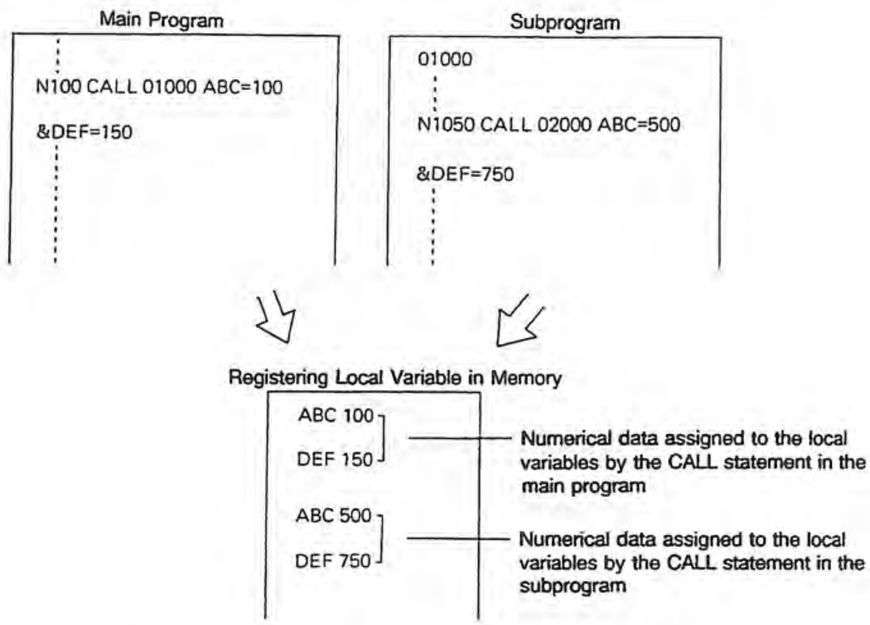
```
  :
  :
N0010    DIA1 = 160
  :
N0049
N0050    DIA1 = 200
```

In N0010, numerical data "160" is assigned to local variable name "DIA1", which remains effective up to sequence N0049. In N0050, new numerical data "200" is assigned to the same local variable name "DIA1". This clears the old data "160" and it is substituted with the new data "200"

4) Up to 127 local variables can be used.

5) When a subprogram call command (CALL statement) is programmed in the main and the subprogram with local variables set in the block containing CALL statement, the variables assigned with the numeral values in such block are all registered as local variables and their numeral values are stored in memory.

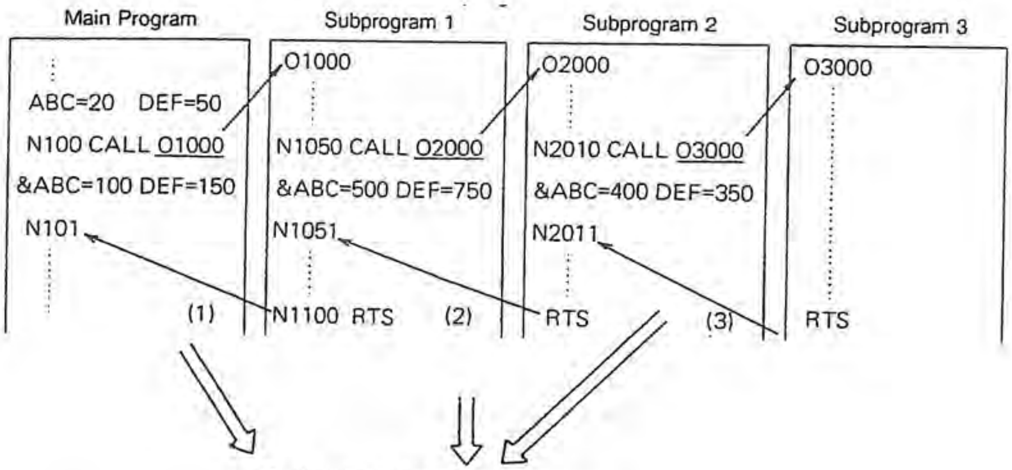
Even when the local variable having the same name as the one already registered before the call statement is programmed is designated, it is newly registered as a new variable.



As shown above, the variables having the same name as those already registered are newly registered as different variables.

6) When using local variables in the subprogram called the numerical data assigned last to the local variable is used when there are several local variables assigned with the same name registered in the memory.

   The local variables set in the block containing CALL statement are all cleared when the RTS statement in the called subprogram is executed.

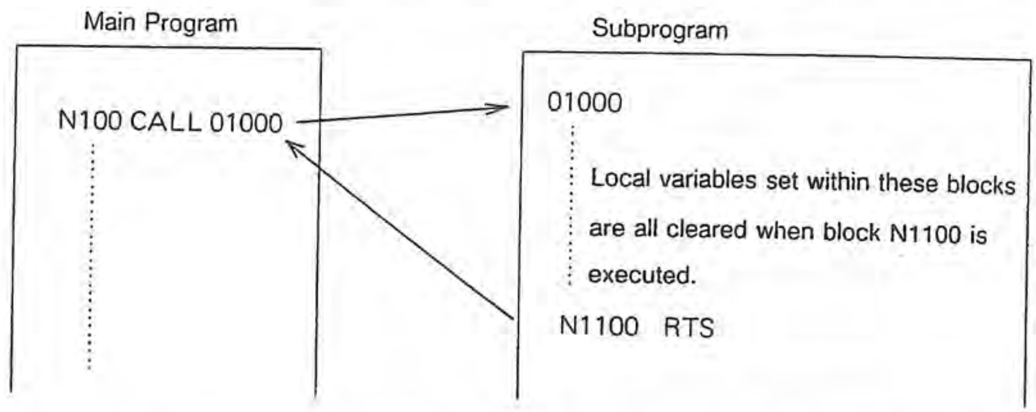


Registration of Local Variables

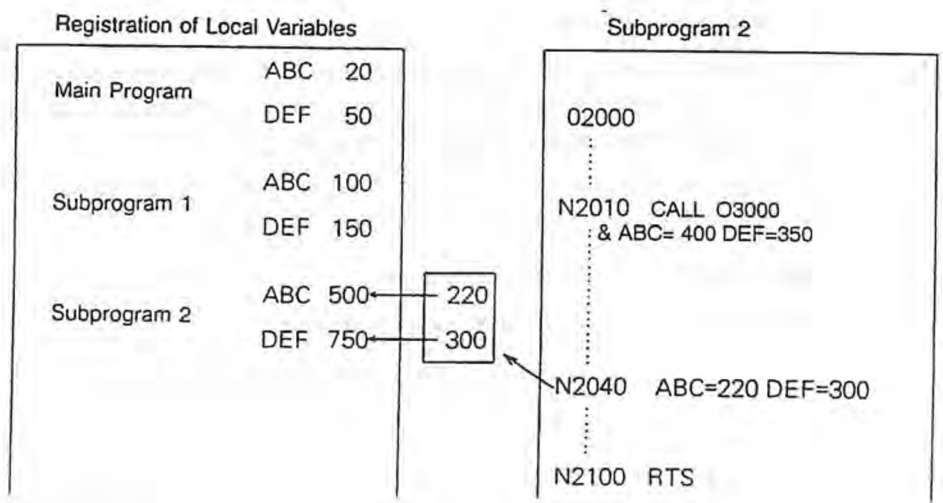
Main Program	ABC	20	
	DEF	50	
Subprogram 1	ABC	100	
	DEF	150	·········· Cleared in (1)
Subprogram 2	ABC	500	
	DEF	750	·········· Cleared in (2)
Subprogram 3	ABC	400	
	DEF	350	·········· Cleared in (3)

In the example above, execution of N201 0 in the subprogram 2 registers 4 kinds of local variables ABCs and DEFs and the subprogram O3000 is executed. If the subprogram O3000 contains local variable names ABC and DEF, the numeral data registered last, i.e., ABC = 400 and DEF = 350 are called for. At the end of the 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.

7) 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.



8) When numerical data is assigned to the local variable name which is already assigned with another numerical data while a subprogram is executed, the numerical data is renewed. If several local variables having the same variable name are registered in the memory, the numerical data of the local variable registered last is renewed.



While the subprogram 2 is executed, local variables ABC = 400 and DEF = 350 are registered in the memory but they are cleared by executing subprogram 3 RTS. Therefore, the variables registered when subprogram 2 is called in N 1050 are effective in the blocks up to N2039. Therefore, when block N2040 is executed, the numerical values of local variables ABC and DEF registered in the subprogram 2 are renewed to 220 and 300, respectively, and those registered in the subprogram 1 and the main program are not renewed.

(3) System Variables

System variable means the variable specific to respective systems and its name is fixed.

The system variables are not cleared when the control is reset.

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 detailed later. Therefore, they can be effectively used in programs requiring them, such as work gauging program, tool gauging program, and post-process gauging program.

They, of course, can be set by selecting ZERO SET, TOOL DATA or PARAMETER mode. For details of setting procedure, refer to III PARAMETERS.

System variables are detailed hereinafter:

Fundamental Program Format of System Variable

System variable . . . . . . <V> <Letter> <Three alphanumerics>

Character string in this part is determined and the use of illegal character string causes alarm.

All the system variables are preceded by character "V".

(a) Zero offset variable

Variable name

<VZOFZ> . . . . Zero OFfset of Z-axis

<VZOFX> . . . . Zero OFfset of X-axis

<VZOFC> . . . . Zero OFfset of C-axis (for multi-machining model)

When setting a variable, designate as VZOFZ = 12364.256.

(b) Zero shift variable

Variable name

<VZSHZ> .... Zero SHift of Z-axis

<VZSHX> .... Zero SHift of X-axis

<VZSHC> .... Zero SHift of C-axis (for multi-machining model)

When setting a variable, designate as VZSHZ = 50.

These zero shift variables deal with shift amount set by zero shift operation called by G50, and the set shift amount is cleared when the control is reset.

(c) Tool offset variable

Variable name

<VTOFZ> [Tool Offset No.] .... Tool OFfset of Z-axis

<VTOFX> [Tool Offset No.] .... Tool OFfset of X-axis

When setting a variable, designate as VTOFZ [5] = 2.634.

This indicates that tool offset amount of Z-axis for #5 is set 2.634.

(d) Nose radius compensation variable

Variable name

<VNSRZ> [Nose R Compensation No.] .... NoSe Radius compensation of Z-axis

<VNSRZ> [Nose R Compensation No.] .... NoSe Radius compensation of Z-axis

When setting a variable, designate as VNSRZ [4] = 0.8

This indicates that nose radius (in Z-axis) of the tool assigned with nose radius compensation no. 4 is set to 0.8 mm.

(e) Variable soft limit variable

Variable name

<VPVLZ> .... PositiVe Limit on Z-axis

<VPVLX> .... PositiVe Limit on X-axis

<VNVLZ> .... NegatiVe Limit on Z-axis

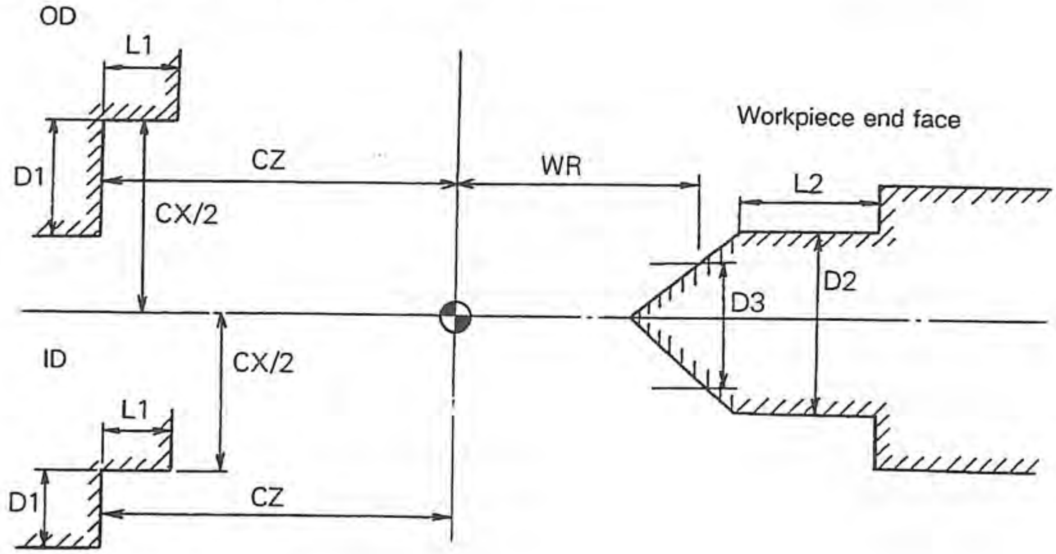
<VNVLX> .... NegatiVe Limit on X-axis

When setting a variable, designate as VPVLZ = 2352.168.

This indicates that variable soft limit of Z-axis in positive direction is set at Z = 2352.168 mm.

Numerical data of these variables are referenced to the origin of the machine coordinate system (machine origin).

(f) Chuck barrier/tailstock barrier variable



Variable name

<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

Numerical data of these variables are referenced to the origin of the program coordinate system (programming zero).

(g) Droop variable

Variable name

<VINPZ> .... Droop amount in Z-axis
IN Position Z-axis

<VINPX> .... Droop amount in X-axis
IN Position X-axis

<VINPC> .... Droop amount in C-axis
IN Position C-axis

(h) Pitch error compensation variable

Variable name

<VPFVZ> .... Pitch compensation amount in Z-axis
Pitch Fillup Value Z-axis

<VPFVX> .... Pitch compensation amount in X-axis
Pitch Fillup Value X-axis

<VPFVC> .... Pitch compensation amount in C-axis
Pitch Fillup Value C-axis

<VPCHZ> .... Pitch amount in Z-axis
PitCH Z-axis

<VPCHX> .... Pitch amount in X-axis
PitCH X-axis

These variables are used only for the pitch error compensation specification.

(i) Restart variable

Variable name

<VRSTT> .... ReSTarT

This indicates the restart state in the sequence restart operation.
#00 = Not in restart
#80 = In restart

Example: N100 IF [VRSTT␣NE␣0]␣N200

This indicates the restart state in the sequence restart operation.
#00 = Not in restart
#80 = In restart

Example: N100 IF [VRSTT NE 0] N200

(j) Alarm comment variable

Variable name

<VUACM> .... User Alarm CoMment

User alarm comments up to 16 characters can be designated.

VUACM[1] - VUACM[16]

This variable is cleared when the NC is reset.

For the alarm variable, character-string or a hexadecimal code (preceded by the $ symbol) in quotation marks (' ') can be used. As character-string, alphabets (upper case and lower case) can be used. For the procedure refer to "GET/PUT Statement".

As hexadecimal code, setting is possible up to four characters.

Output of a comment to the display screen is designated with output variable VDOUT [∗] =code number. See below.

∗ = 991: 3202 Alarm C

∗ = 992: 2292 Alarm B

∗ = 993: 1213 Alarm A

As a code number, numbers 1 - 9999 can be used.

Program example 1:

```
N202 VUACM[1] = 'ABC= 100␣'
N203 VUACM[9] = ' ABC'
N204 VUACM[12] = '=200'
N205 VDOUT[991] = 999
```

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

Program example 2:

```
N202 VUACM[1] = 'ABC'
N203 VUACM[5] = '= 123'
N204 VDOUT[991] = 999
   :
   :
```

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]'
N102 VUACM[5] = '=GEAR"
N103 VDOUT[992] = 1000
   :
   :
```

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 1000 PART = GEAR

Program example 4:

```
N301 VUACM[1] = $41424344
N302 VDOUT[992] = 2000
   :
   :
```

Screen display:

2288 Alarm B User reserve code 2000 ABCD

(k) Arithmetic variable

Variable name

<VPAI> .... π (Ratio of circumference of circle to its diameter)

(l) NOEX command

Designated at the beginning of a variable setting sequence to speed up the "program check" by eliminating single-block processing. (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 for optional parameter (OTHER FUNCTION 1) NOEX command ignore.

(4) I/O Variables

I/O variables are the variables effective for signal transmission between the control and peripheral equipment.

Input variables:

The variable 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 variable representing signals outputted 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".

(a) Input variable

Variable name

<VDIN> [Input variable no.] .... Data INput

Left bracket Right bracket

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 bits data of hexadecimal number, S0 through S1F	
11 - 18	Bit data: 0 (OFF), 1 (ON)	EC input
19	1 byte data in which data of variables #11 through #18 are corresponded 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 are corresponded to bit 0 through bit 7.	
1000 - 1004	Timers: 1 msec., 1 sec., 1 min., 1 hour, 1 day (with power turned on, timers are reset to "0" and starts counting up after that: 4-byte counter)	
1235 - 1250	Condition of specification codes: #1 through #16 (in units of byte)	

Details of specifications must be discussed.

(b) Output variable

Variable name

<VDOUT> [Output variable no.] .... Data OUTput

Left bracket Right bracket

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 are corresponded 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 are corresponded 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 are corresponded 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 are corresponded 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.

(5) I/O read variables

I/O read variables are system variables that can read the status of panel inputs/outputs and EC inputs/outputs.

The system variable used for reading the input status is "VIRD" and the one used for reading the output status is "VORD".

It is possible to read the information of an arbitrary bit by specifying the I/O bit address for a suffix.

The read data is: "1" if the bit is ON and "0" if the bit is OFF.

Reading the input bit:

VIRD[***#]

*** : I/O check number
# : Bit position

Reading the output bit:

VORD[***#]

*** : I/O check number
# : Bit position

Suffix address is expressed by the I/O check number and the bit position.

For the address of individual bits, refer to the Maintenance Manual.

3-3. Arithmetic Operation Function

Arithmetic operation using variables can be performed. Its programming may be done in the same manner as general arithmetic expression.

<Address character>, <Extended address character>, <Variable> = <Expression>

The expression in the right-hand side, requesting arithmetic operation, is made from constants, variables, functions, boolean expressions, comparison expression, and operators.

Types and contents of them are detailed on the following page.

(1) Arithmetic Expression

Operator	Meaning	Example	Rule and Remarks
+	Positive sign	+1234	
–	Negative sign	–1234	
+	Sum (addition)	X = 12.3+V1	
–	Difference (subtraction)	X = 12.3-V1	
∗	Product (multiplication)	X = V10∗10	
/	Quotient (division)	X = V11/10	

(2) Comparison Expression

Operator	Meaning	Example	Contents	Rule
LT	(Less Than, <)	IF [V1 LT 5] N100	Jump to N100 when V1 is less than 5.	Provide a space on either side of the operator.
LE	(Less than or Equal to, ≦)	IF [V1 LE 5] N100	Jump to N100 when V1 is less than or equal to 5.	
EQ	(EQual to, =)	IF [V1 EQ 5] N100	Jump to N100 when V1 is equal to 5.	
NE	(Not Equal to, ≠)	IF [V1 NE 5] N100	Jump to N100 when V1 is not equal to 5.	
GT	(Greater Than, >)	IF [V1 GT 5] N100	Jump to N100 when V1 is greater than 5.	
GE	(Greater than or Equal, ≧)	IF [V1 GE 5] N100	Jump to N100 when V1 is greater than or equal to 5.	

(3) Boolean Expression

Operator	Meaning	Example	Rule
OR	Logical sum	VDIN [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		

(4) Functions

Operator	Meaning	Example	Rule and Remark
SIN	Sine	X = 15∗SIN [22.5]	The numbers after the function operation symbols must be enclosed using '[' and ']'. In addition, they are used for specifying the priority of operation execution order. When two elements are specified in "[␣]", place a comma"," between them. The position of a decimal point is determined in accordance with the unit system selected. Unit systems for angle commands are: 1 deg. for 1 mm and 1 inch unit system, 0.001 deg. for 1 μm unit system, 0.0001 deg. for 0.0001 inch unit system.
COS	Cosine	Z = 15∗COS [22.5]	
TAN	Tangent	Z = 15∗TAN [12.5]	
ATAN	Arctangent (1) Value rang: −90° to 90°	X = 15∗ATAN [22.5]	
ATAN2	Arctangent (2)	ATAN2 [10, 15]	
SQRT	Square root	X = 15∗SQRT [224.5]	
ABS	Absolute value	ABS [V15]	
BIN	Decimal to binary conversion	BIN [V15] 4 BYTE	
BCD	Binary to decimal conversion	BCD [V20] 4 BYTE	
ROUND	Rounding off fractions	ROUND [V5]	
FIX	Cutting off fractions	FIX [V7]	
FUP	Counting fractions as a whole number	FUP [V15]	
DROUND	Rounding off the fractions to three decimal places (metric system) or to four decimal places (inch system)	DROUND [V20]	
DFIX	Cutting off the fractions blow the third place of decimals (metric system) or below the fourth place of decimals (inch system)	DFIX [V20]	
DFUP	Count the figures below the third place of decimals (metric syst em) or below the fourth place of decimals (inch system) as a whole number	DFUP [V21]	
MOD	Remainder (a - fix[a/b]∗b)	MOD [a, b]	

* The value of ATAN2 [a, b] represents the angle of the point defined by the coordinate values (a, b). Its range is from −180° to 180°.

3-4. Combination of Operations

(1) Operations and functions explained in the previous page can be combined as needed.

X = V1 + V2 - V3 + V4∗COS [30]

(2) Designating operator precedence by [ ]

Operator precedence can be determined with the use of [ ].

Example:

V1 = [V2 + V3]∗V4 + SIN [[V5 - V6]/V7] + V8]

① ②

③ ④

⑤

4. Supplemental Information on User Task Programs

4-1. Sequence Return in Program Using User Task

Basically, sequence return can be performed in the same manner as in conventional program and there are no restrictions to activate sequence return function.

When variables are set in a block preceding the one where the sequence return is made, the set data are all registered in the memory.

4-2. Rules of Operation and Evaluation of Result

(1) Type of Data

There are three types of data as "integer type", "real type" and "logical type".

(a) Integer type

The data of integer type accurately express integer values and can be zero, positive integer or negative integer.

(b) Real type

The data of real type accurately express real values and can be zero, positive real number or negative real number.

(c) Logical type

The data of logical type may be either true (I) or false (0).

(2) Constant

There are two types of constants as "integer constant" and "real constant". The constants preceded by either "plus" or "minus" sign are called "signed constant". The constants comprising those are called "generic constant".

(a) Integer constant

Integer constants are the constants of integer type. They are expressed by up to eight digits and are interpreted as a decimal number.

(b) Real constant

Real constants are the constants of real type. They are exPressed by up to eight digits including a decimal point and are interpreted as a decimal number.

(3) Types of Variables and Evaluation of its Value

When setting a variable, assignment statement is used:

V = e
where, V = variable name
e = constant, variable name, expression, and function

With the setting indicated above, value of "e" is evaluated, and the value of "V" is changed according to the rule.

Abbreviation symbol

[I] .... Integer type
[R] .... Real type
[I] → [R] .... Change to real type
[R] → [I] .... Change to integer type

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 off fractions)
	mm inch	[I]	[I] *1000 (metric system) [I] * 10000 (inch system)
		[R]	[R] *1000 → (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)

(4) Rules of Operation Expression and Evaluation of Values

Example:

Expression .... C = A + B

Element 2
Element 1
Result of operation

Abbreviation symbol

[I] .... Integer type
[R] .... Real type
[I] → [R] .... Change to real type
[R] → [I] .... Change to integer type
[b] .... Logical type

Type of Expression	Operator	Meaning	Type of Element 1 "A"	Type of 55 Element 2 "B"	Type of Result of Operation "C"
Arithmetic expression	+	Addition	[I]	[I]	[I]
			[I] → [R]	[R]	[R]
	−	Subtraction	[R]	[I] → [R]	[R]
			[R]	[R]	[R]
	+	Positive sign	[I]		[I]
	−	Negative sign	[R]		[R]
	*	Multiplication	[I] → [R]	[I] → [R]	[R]
	/	Division	[R]	[R]	
Comparison expression	LT	Less than <	[I]	[I]	[b]
	LE	Less than or equal to ≦			
	EQ	Equal to =	[I] → [R]	[R]	
	NE	Not equal to ≠	[R]	[I] → [R]	
	GT	Greater than >	[R]	[R]	
	GE	Greater than or equal to ≧			
Logical expression	EOR	Exclusive OR	[I]	[I]	[I]
	OR	Logical sum	[R] → [I]	[R] → [I]	
	AND	Logical prod-uct	(Cutting off fractions)	(Cutting off fractions)	
	NOT	Negation	[I] or [R] → [I] (Cutting off fractions)		[I]

(5) Operation Rule of Functions and Evaluation of Value

Abbreviation symbol

[I] .... Integer type
[R] .... Real type
[I] → [R] .... Change to real type
[R] → [I] .... Change to integer type

Function Name	Meaning	Unit System	Type of Element 1	Type of Element 2	Type of Result of Operation
SIN COS	Sine Cosine	1 mm	[I] → [R]	[R]/1000 deg. (metric system)	[R]
		1/10000 inch	[R]	[R]/1000 deg. (inch system)	[R]
TAN	Tangent	mm inch	[I] → [R] [R]	(degree)	[R] [R]
ATAN ATAN2	Arctangent	1 mm	[I] → [R]		[R] ∗1000 (1/10000 degree) (metric system)
		1/10000 inch	[R]		[R] ∗1000 (1/10000 degree) (inch system)
SQRT	Square root	–	[I] → [R] [R]	–	[R]
ABS	Absolute value	–	[I] → [R] [R]	–	[R]
BIN	BCD → BIN	–	[I] [R] → [I] (Cutting off fractions)	–	[I]
BCD	BIN → BCD	–	[I] [R] → [I] (Cutting off fractions)	–	[I]
ROUND	Rounding off fractions	–	[I] [R]		[I] (Not changed) [I]
FIX	Cutting off fractions	–	[I] [R]	–	[I] (Not changed) [I]
FUP	Counting fractions as a whole number	–	[I] [R]	–	[I] (Not changed) [I]

5. Program Examples

Three typical program examples are provided in the following pages. They are prepared only for your reference purpose and they do not cover all the functions available with User Task 2.

Please refer to the examples so that you can make the most of User Task function for preparing programs.

<Program Example 1> .... [Shaft work with similar contour]

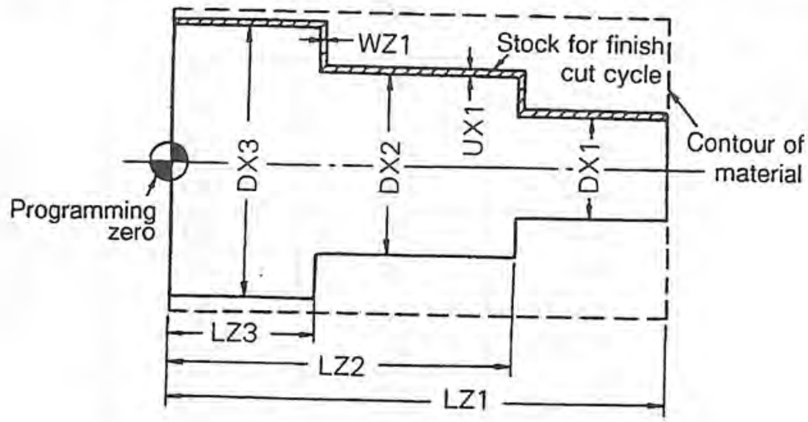


Fig. 23-1

Assuming three different workpieces having similar contours as shown above are to be cut, programs are prepared using user task function.

(1) Assign respective workpieces with file names:

SHAFT-A

SHAFT-B

SHAFT-C

(2) Since these workpieces have similar contours, their contours are defined with a subprogram. 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

(4) For cutting a workpiece, LAP mode is used.

(5) Described above are summarized in Table 2-1.

Table 1-2

		Work A SHAFT-A	Work B SHAFT-B	Work C SHAFT-C
Roughing tool	V1	0101	0303	0505
Finishing tool	V2	0202	0404	0606
Cutting speed in roughing cycle	V3	100	110	90
Cutting speed in finishing cycle	V4	120	130	150
LZ1	LZ1	200	250	300
LZ2	LZ2	150	170	200
LZ3	LZ3	80	100	120
DX1	DX1	30	40	50
DX2	DX2	50	70	90
DX3	DX3	80	120	150
WZ1	WLZ1	0.1	0.15	0.2
UX1	UDX1	0.2	0.25	0.3
XS	XS	100	140	170
ZS	ZS	210	260	300

The subprogram defining the contour, prepared using local and common variables, can be programmed as below according to Table 2-1.

$ 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	Z = ZS	S = V3	T = V1	M03	M08
N1012	G85	NLAP1	D4	F0.35	U = UDX1	W = WLZ1	
N1013	G97	S = 1000/3.14*V3/XS					
N1020	G00	X = 800	Z = 400				
N1021	G96	X = XS	Z = ZS	S = V4	T = V2		
N1022	G87	NLAP1					
N1023	G00	G97	X = XS	Z = ZS	S = 1000/3.14*V4/XS		
N1024	RTS						

The main program for each workpiece is programmed as below using the subprogram.

Workpiece A	
$ SHAFT-A.MIN	.... a)
%	.... b)
0100	.... c)
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 ZS = 210	.... d)
N103 M02	

(a) File name of the main program .... Program $ proceeding the file name.

(b) Punch the machining program (main program) in the following order.

$, program name, MIN, feed holes, %, CR, LF.

(c) Program name .... O100 in this example

(d) With the commands in N102, the subprogram O1000 is called and executed. Numerical data for the variables used in the subprogram to be called are all set. When the commands in this block cannot be written in one line, separated them in several lines with placing "$" code at the beginning of the 2nd and later lines. The control recognize those commands as programmed in one block.

Workpiece B
$ SHAFT-B.MIN
%
0101
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 = 0.15 UDX1 = 0.25 XS = 140 ZS = 260
N103 M02

Workpiece C
$ SHAFT-C.MIN
%
0102
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 ZS = 310
N103 M02

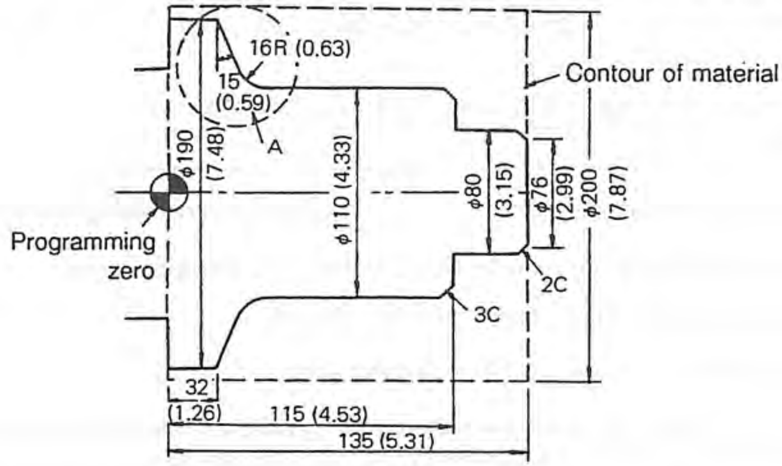
Note: To call the machining program, follow the steps below:

(1) Press the AUTO key to select the automatic operation mode.

(2) Press function key [F1] (PROGRAM SELECT).

(3) Key in the main program name and subprogram name, and press the WRITE key.

<Program Example 2> . . . . (Cutting contour requiring calculation for defining 1)



When cutting the contour containing a circular arc and a taper and when the point(s) of intersection is not indicated on the part drawing, the user task function featuring operation function is effectively used for preparing a program for such contour.

(1) With enlarging section A, have the control calculate the points of intersection using the variables and the operation function of the user task.

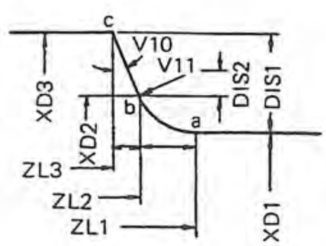


Fig. 23-2

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.

Set using common variables

V10 = Taper angle (15°)

V11 = Arc radius (16 mm)

Set using local 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)

The point of intersection can be calculated in the following equations.

DIS1 = [XD3 - XD1]/2

DIS2 = V11∗SIN [V10]

DIS3 = V11∗COS [V10]

DIS4 = [DIS1 + DIS2 - V11]∗TAN [V10]

XD2 = XD1 + 2∗[V11 - DIS2]

ZL2 = ZL3 + DIS4

ZL1 = ZL2 + DIS3

(2) Since the pattern appearing in Section A can be used in common with other workpieces, it is advisable to program such contour as a subprogram. Name such subprogram "RADIUSTAPER.SUB" (filename). Setting of variables <XD2>, <ZL1>, and <ZL2> is made within this subprogram, and other variables are set in the main program.

Subprogram RADIUS-TAPER.SUB
%
ORT01
N1000 XD2 = XD1 + 2∗[V11 - DIS2] ZL2 = ZL3 + DIS4 $ ZL = AL2 + DIS3
N1001 G01 Z = ZL1
N1002 G02 X = XD2 Z = ZL2 L = V11
N1003 G03 X = XD3 Z = ZL3
N1004 RTS

- Variables are set in block N1000.
- Z coordinate of point "a" is commanded in block N1001.
- X and Z coordinates of point "b" and arc radius are commanded in block N1002.
- X and Z coordinates of point "c" are commanded in block N1003.
- RTS in block N1004 indicates the end of the subprogram.

(3) Prepare the cutting program as a main program.

File name of the main program is "FLANGE-1.MIN". LAP and nose radius compensation function are used in the main program.

The main program is shown below.

Main Program $ FLANGE-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	L2				
N106	G01		Z115				
N107	G75	X = XD1 – 6					
N108		X = XD1	Z112				
N109	CALL	ORT01	DIS1 = [XD3 – XD1]/2		DIS2 = V11*SIN [V10]		
$ DIS3 = V11*COS[V10]		DIS4 = [DIS1 + DIS2 – V11]*TAN [V11]					
N110			Z-2				
N111	G40	G00	X200	I10			
N112	G80						
N120	G00	X800	Z300				
N121	G96	X202	Z137	S110	T010101	M03	M08
N122	G85	NLAP1	D6	F0.35	U0.2	W0.1	
N123	G00	X800	Z300				
N124			S130	T020202			
N125	G87	NLAP1					
N126	G00	X800	Z300				
				M05	M09		
N128	M02						

Note: *The subprogram ORT01 is called for by the command in block N109 for defining the contour consisting of circular arc and tape. Variables used for defining such contour are all set in this block.*

<Program Example 3> . . . . (Cutting contour consisting of repetitive contour)

In this example, the contour having the repeated same contours in it such as a pulley is programmed.

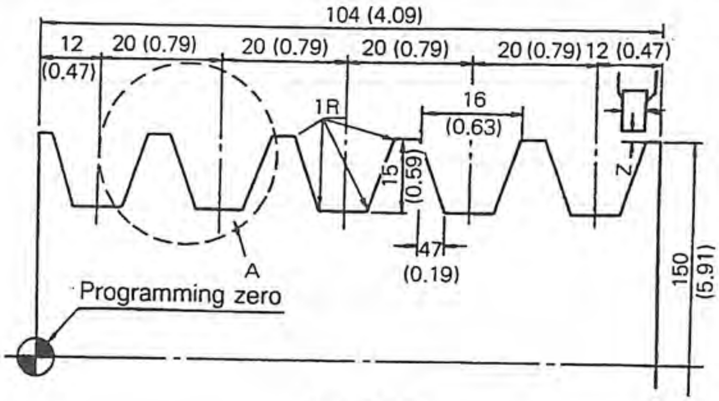


Fig. 23-3

(1) Assume that there are different pulleys having the similar contour as above. For simplify the programs of those pulleys, express the contour of part A using variables.

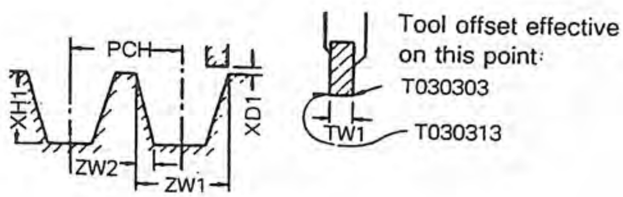


Fig. 23-4

Variable Name	Contents	Numerical Value for This Example (Fig. 2-6)
PCH	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	5
DI	Value of I of IR	0.299
DK	Value of K of KR	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 at any position wherever called.

Subprogram file name is "PULL-PTTN1.SUB".

Subprogram $ PULL-PTTN1.SUB			
%			
OPR 1			
N1	G91	G00	Z = [ZW1/2] + [TW1/2]
N2	G01		X = –XD1 + 0.2 F0.2
N3			X = –[XH1∗2] – 0.2 F0.05
N4	G00		X = [XH1∗2] + XD1
N5	G42		Z = ZW2
N6	G01		X = –XD1 F0.2
N7			X = –[XH1∗2] Z = –ZW2 F0.05
N8	G40		K–1
N9	G00		X = [XH1∗2] + XD1
N10	G41		Z = –ZW2 – TW1 T130313
N11	G01		X = –XD1 F0.2
N12			X = –[XH1∗2] Z = –ZW2 F0.05
N13	G40		K1
N14	G00		X = [XH1∗2] + XD1
N15	G41		Z = –ZW2 – [DK – [1 – DI]∗4.7/15] – 0.8
N16	G01		X = –XD1 F0.2
N17	G02		X = –[1 – DI]∗2 Z = DK I = –1 F0.05
N18	G01		X = [XH1∗2] + [1 – D1]∗4 Z = ZW2 – [1 – DI]∗4.7/15∗2
N19	G03		X = –[1 – D1]∗2 Z = DK I = DI K = DK
N20	G01	G40	K1
N21	G00		X = [XH1∗2] + XD1
N22	G42		Z = ZW2 + [DK – [1 – DI]∗4.7/15] + [0.8∗2] – [DK – [1 – DI]∗4.7/15] + [0.8∗2] + TW1 T030303
N23	G01		X = –XD1 F0.2
N24	G03		X = –[1 – D1]∗2 Z = –DK I = –1 F0.05
N25	G01		X = –[XH1∗2] + [1 – D1]∗4 Z = –XW2 + [1 – DI]∗4.7/15∗2
N26	G02		X = –[1 – D1]∗2 Z = –DK I = DI K = –DK
N27	G01	G40	K–1
N28	G00		X = [XH1∗2] + XD1
N29			Z = ZW2
N30	G90		
N31	RTS		

(3) In step (2), the program for cutting one pulley groove is made. Using that subprogram, the program to cut the pulley shown in Fig. 2-6 can be prepared.

Make this program as a main program: Program file name is "PULLY-1.MIN".

Main Program $ PULLY-1.MIN							
%							
OPLY1							
N001	G13						
N002	G00	X800	Z300				
N003	G50	S1500					
N004	G96			S70	T030303	M03	
N005		PCH = 20	XD1 = 2				
N006			Z100				
N007	MODIN D1 = 0.299	OPP1	XH1 = 15 DK = 0.954	ZW1 = 16	ZW2 = 4.7	TW1 = 5	
N008	G00	X = 150 + XD1		Z100			
N009	G00		Z = 100 – PCH				
N010	G00		Z = 100 – [2*PCH]				
N011	G00		Z = 100 – [3*PCH]				
N012	G00		Z = 100 – [4*PCH]				
N013	MODOUT						
N014	G40	G00	X800	Z300	M05		
N015	M02						

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 command(s) is completed. In this block, variables used in the subprogram OPP1 are also set.

In blocks N008 through N012, the subprogram OPP1 is called and executed every time the axis motion command(s) in those blocks is completed, thus cutting the pulley grooves.

The use of the CALL statement instead of the MODIN and MODOUT statements also allows the pulley grooves to be cut. However, when the CALL statement is used instead of them, that statement must be repeated every time the subprogram is to be called.

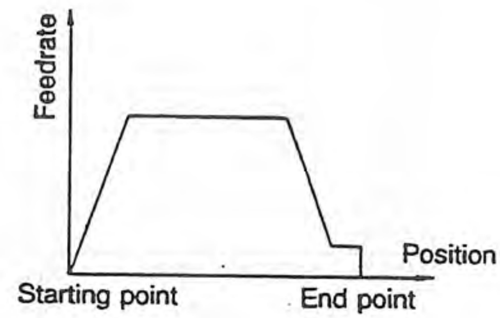
SECTION 24 OTHER FUNCTIONS

1. Automatic Acceleration and Deceleration

Acceleration at the start of an axis movement and deceleration at the end of an axis movement is automatically incorporated into axis movements.

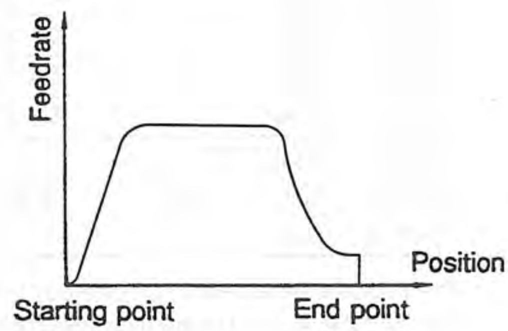
(1) Positioning Mode and Manual Feed Mode

During positioning mode and manual feed mode, automatic acceleration and deceleration is linear, as shown at right.



(2) Cutting Mode (G01, G02, G03)

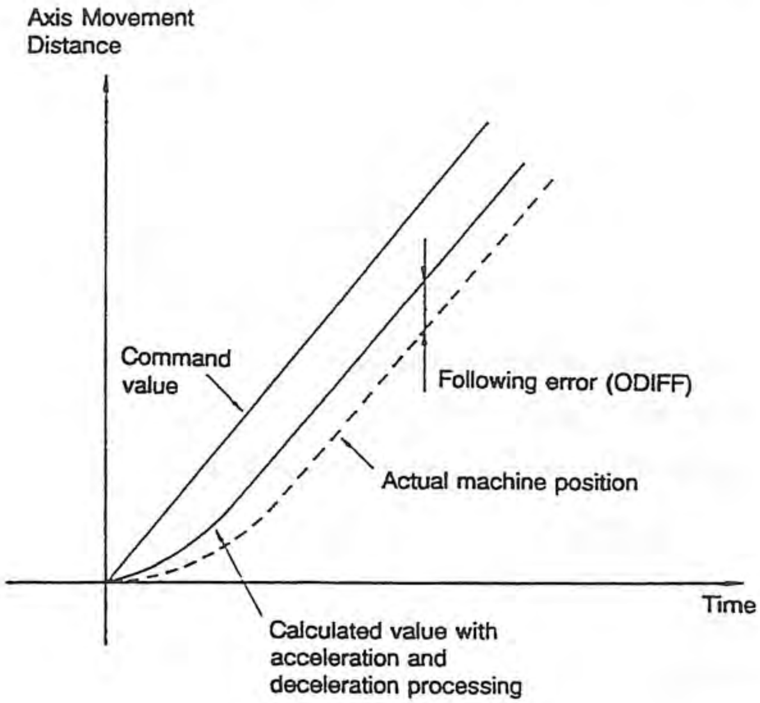
During cutting mode, automatic acceleration and deceleration is competent, as shown at the right.



2. Following Error Check

Following error is defined as the difference between the command value from the NC and the detected position value.

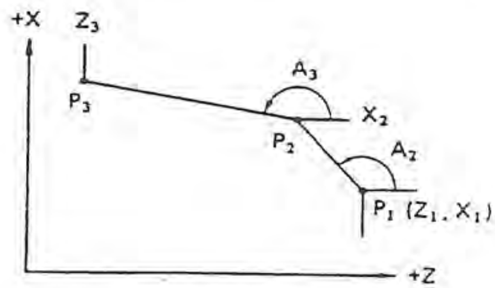
The DIFF-over alarm will be activated if a following error (ODIFF) reaches a certain value during rapid or cutting feed of an axis.



3. 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.

Using this feature, it is commanded simply by entering either X or Z coordinate point of the end point of the taper along with the angle referenced to Z-axis (measured in counterclockwise direction).



N_1	G01	X_1	Z_1	F_1
N_2		X_2		A_2
N_3			Z_3	A_3

(1) Angle command in taper definition is effective in:

G00, G01, G31, G32, G33, G34, and G35

(2) The angle is expressed following an address character "A".

(3) Unit of angle command is:

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°

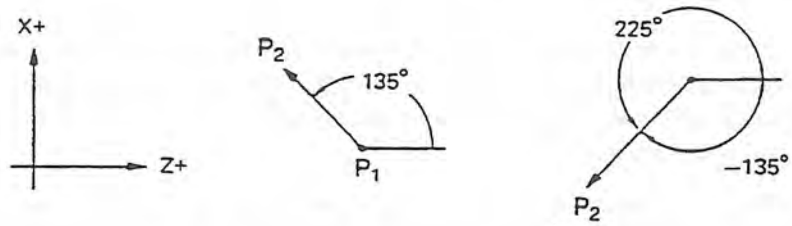
(4) The control interprets the commands as a taper command when the commands contain either of X or Z word along with an A word.

(5) If an A command is provided with both X and Z words, or if it is provided without associated with X or Z word, an alarm results.

(6) The direct taper angle command is effective in:

LAP
Tool nose radius compensation mode
Incremental programming mode
Subprogram

(7) Angle is measured on 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.



A135

A225
A–135

In the figure above, the angle is expressed as A135 on 1 mm unit system control since the angle is taken in the counterclockwise direction.

For the angle shown right, A225 and A–135 result the same taper.

(8) 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

4. Barrier Check Function

4-1. General Description

It permits a chuck and tailstock barrier (a specific machine area in which any entry of a cutting tool is inhibited) to be established in the vicinity of a chuck and a tailstock by the data either in a program or those entered through MDI switches. If a tool is commanded to move into the barrier, it causes an alarm and stops the machine operation.

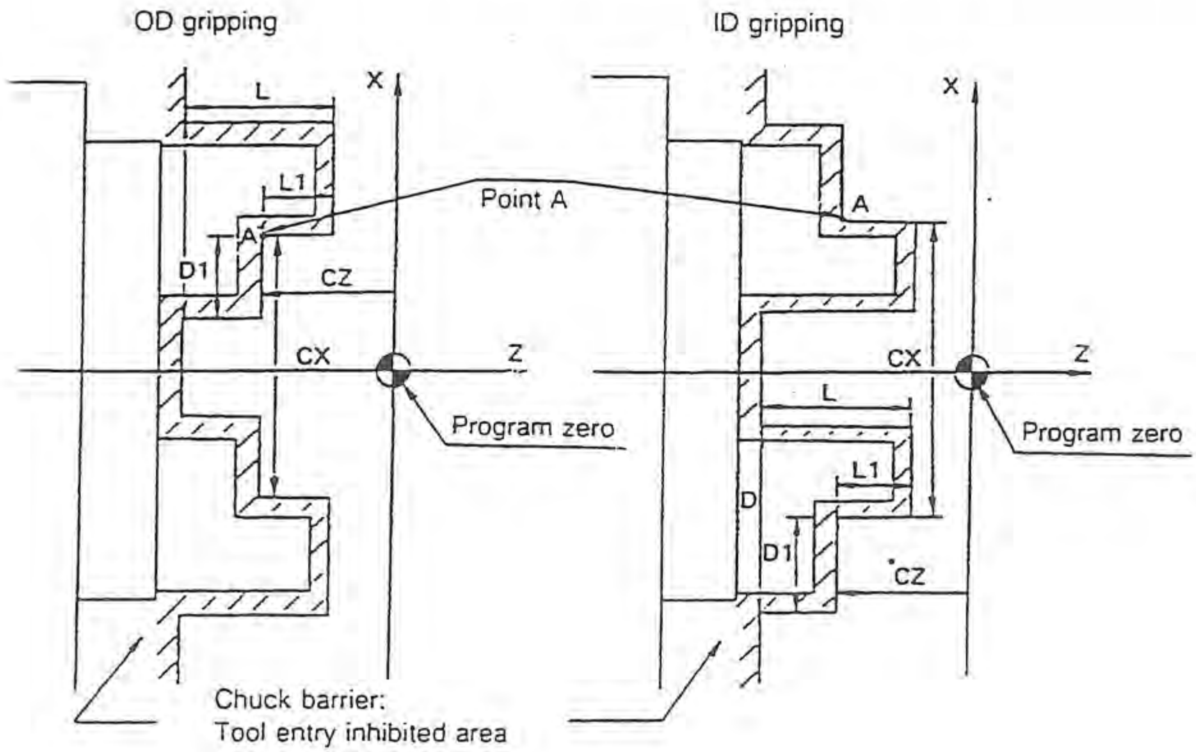
With this function, it is possible to carry out trial operation in the automatic mode or cutting in the MDI mode without worrying about collision of a cutting tool. In addition, this function prevents accidents that a cutting tool is hit against the chuck or tailstock due to unexpected problem during automatic operation.

4-2. Chuck Barrier and Tailstock Barrier

(1) Establishing Chuck Barrier

The chuck barrier function can set the inhibited area for tool entry around the chuck, that cannot be set by variable soft-limit position data.

Activation or deactivation of the chuck barrier function can be selected by programming a proper M code. Therefore, check of tool motion using the chuck barrier function can be made only when required.

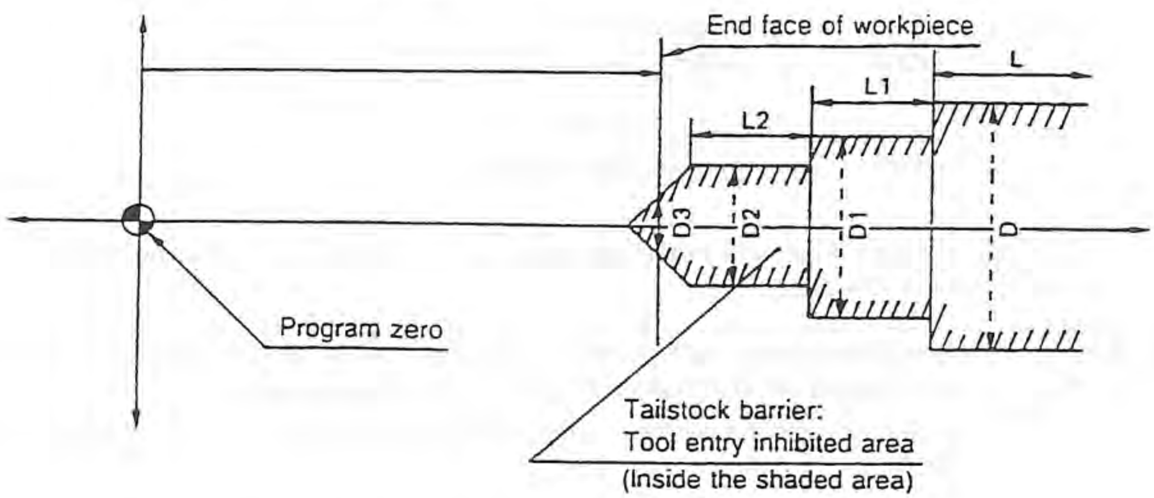


Symbol	Description	Method
L	Chuck jaw length	Chuck/tailstock axis
D	Chuck jaw size	Chuck/tailstock axis
L1	Gripping length of chuck jaw	Chuck/tailstock axis
D1	Chuck jaw gripping face width	Chuck/tailstock axis
CX	Chuck gripping diameter	Chuck/tailstock axis
CZ	Distance from programming zero	Chuck/tailstock axis

For details of procedure to establish the chuck barrier, refer to III PARAMETERS.

(2) Establishing Tailstock Barrier

The tailstock barrier can set the inhibited area for tool entry around the tailstock that cannot be set by variable soft-limit position data.



The tailstock barrier is established by setting required data at parameters.

Symbol	Description	Method
L	Tailstock spindle length	Chuck/tailstock axis
D	Tailstock spindle diameter	Chuck/tailstock axis
L1	Tailstock spindle length (1)	Chuck/tailstock axis
D1	Tailstock spindle diameter (1)	Chuck/tailstock axis
L2	Tailstock spindle length (2)	Chuck/tailstock axis
D2	Tailstock spindle diameter (2)	Chuck/tailstock axis
D3	Tailstock spindle center diameter	Chuck/tailstock axis
WR	Tailstock spindle position (Z)	Chuck/tailstock axis

For details of procedure to establish the tailstock barrier, refer to III PARAMETERS.

(3) Tool Movements and Alarm

Once the chuck barrier is established, the barrier is activated or deactivated by programming a proper 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 into the barrier while the chuck and/or the tailstock barrier function is active, it causes an alarm and stops the machine.

Example of Program:

```
   :
   :
NOOO    M25          Chuck barrier ON
   :    (M21)        (Tailstock barrier ON)
   :
NOOO    M24          Chuck barrier OFF
   :    (M20)        (Tailstock barrier OFF)
   :
```

The barrier function is active for the blocks of commands from the block containing M25 (M21) to the one containing M24 (M20).

NOTICE : (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 desired to be active, command M25 and M21.

(2) The chuck and the tailstock barrier function is active for manual pulse handle mode operation or jog feed operation.

(3) The barrier is renewed when new barrier setting data is entered.

5. Operation Time Reduction Function

5-1. Spindle Rotation Answer Signal Ignore (M63)

This function ignores the answer signal for the spindle rotation.

For the following commands related with spindle control, the individual answer signals are usually checked before starting axis movement. However, this function allows the axis move commands specified in the same block with such spindle control commands to be executed before the checking of the answer signals. The answer signals are checked after the completion of axis movement. Therefore, the axis move commands can be executed simultaneously with the spindle control commands so that cycle time is reduced.

Commands for which answer signals are ignored:

- Spindle start/stop command (M03, M04, M05)
- Gear change command (M40 - M44)
- Spindle orientation command (M19)
- Spindle speed command (S command)

(1) Commands

Enter the M63 command in the block where the spindle rotation answer signals should be ignored.

M63: Spindle rotation answer signal ignore

(2) Example of programming

```
N100 G00 X50 Z50 M03 S1000 M63
N101          Z10
```

In the N100 sequence, "X50 Z50" commands are executed simultaneously with the spindle start commands "M03 S1000" and the answer signals for spindle start and spindle speed arrived are checked after the completion of axis movement.

6. Turret Unclamp Command (for NC Turret Specification)

The NC unclamps the turret and travels the axis at the same time by receiving 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     <=   Simultaneous movements of X-axis from
X220  Z300  T020202               220  to 500 and Z-axis from 300 to 800.
```

Note that the M203 command, if specified in the same block with G00, unclamps the turret without regard to the present turret position.

If the M203 command is preceded by the 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.

SECTION 25 SCHEDULE PROGRAMS

The schedule program permits different types of workpieces to be machined continuously without any operator intervention by using a bar feeder, or loader.

(1) Several main programs can be selected and executed in the specified order by the schedule program.

(2) A schedule program is a set of the following five blocks:

If other blocks are specified, and 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.

(3) These commands must be specified at the start of, or immediately after, the sequence name.

(4) Although comments given between '('and ')' and continuous lines identified by '$' are valid, the block delete (/) is invalid.

(5) Total tape length for the schedule, main, and sub is up to the maximum size of operation buffer area which is selected by the specification.

1. PSELECT Block

This block selects and executes main programs for a workpiece to be machined.

(1) This function searches a specified main program file for a specified main program to be selected as a machining program. This function also searches a specified subprogram file, or system subprogram file, and manufacturer subprogram file for the required subprograms and selects them automatically.

(2) It repeats selected programs as specified.

(3) Programming Format

The command must be specified in the following order:

[ ] may be omitted.

{PSELECT} ␣[fm],[pm],[fs],[n] CR/LF

(a) fm: Main program file name

[ ] may be omitted.

[3 characters :][16 characters or less][. 3 characters]

device name **file name** **extension**

1) The defaults of the device, file, and extension (A comma (,) can be omitted) are 'MD1', 'A' and 'MIN', respectively.

2) The default of fm is MD1:A.MIN.

3) If '*' or '?' is used in a main program file name, an alarm will occur.

4) If the specified file does not exist, an alarm will occur.

(b) pm: Main program name

O

5 characters or less

1) The default of Pm is the name of the top program of the file specified by fm.

2) If the specified file odes not exist in fm, an alarm occurs.

3) The absence of M02 or M30 indicating the end of a program cause an alarm.

(c) fs: Subprogram file name

[ ] may be omitted.

[3 characters :][16 characters or less][. 3 characters]

device name **file name** **extension**

1) fs may be omitted when:

- no subprogram is called in the main program,
- the subprogram called from a main program or subprogram exists in MD1:∗.SSB (system subprogram) or in MD1:∗.MSB (manufacturer subprogram), and
- required subprograms other that SSB and MSB are contained in the main program file.

If fs is specified, the device name and extended name may be omitted. Their defaults are 'MD1' and 'SUB' respectively. If all of them are omitted, it is assumed that no file has been specified.

2) If the total number of used subprograms exceeds 126, an alarm will occur.

3) If 'RTS', which means the end of the subprogram, does not exist, an alarm will also occur.

*: If the file specified by fs does not exist, an alarm occurs.

(d) n: Repetition count

Q	=	Expression

1) The repetition count (the number of the called program to be repeated), which must range from 1 through 9999, may be specified using the address 'Q'. Its default is 1.

2) If other that 1 to 9999 is specified, and alarm will occur. " substitutes for '='. If '=' is followed by a numeric value, it may be omitted.

2. Branch Block

The branching function of the schedule program, which is identical to SECTION 23, 3-3. "Arithmetic Operation Function", falls into GOTO and IF blocks, which provide unconditional branching and conditional branching, respectively.

(1) GOTO Block

The GOTO block unconditionally changes program sequences. The destination of a jump is to be 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

(2) IF Block

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 [$\boxed{\text{Expression}}$ ␣ $\boxed{\text{Comparison operator}}$ ␣ $\boxed{\text{Expression}}$ **] N______**

Specifies the destination of a jump

The comparison operators include LT (<), LE (≦), EQ (=), NE (≠), GE (≧), GT (>).

3. Variables Setting Block

'VSET' must be specified to set variables using the schedule program.

Programming format:

Commands must be specified in the following order:

VSET ␣ $\boxed{\text{Variable}}$ = $\boxed{\text{Expression}}$ ␣ $\boxed{\text{Variable}}$ = $\boxed{\text{Expression}}$

If any variable other than a common variable, system variable, or I/O variable, is specified at the left part, an alarm will occur.

Additionally, if an I/O variable is specified at the right part, an alarm will also occur.

4. Schedule Program Termination Block

The schedule program must be terminated with an 'END' block. All the blocks form the 'END' block on are invalid.

Programming format:

END

5. Program Examples

The procedure to create a schedule program is explained below using an easy example.

Assume that the NC lathe is equipped with a bar feeder and three different workpieces are machined according to the programmed schedule.

(1) Assign respective workpieces with file name and program name (or no.):

Workpiece A .... A. MIN, O100

Workpiece B .... B. MIN, O200

Workpiece C .... C. MIN, O300

(2) After determining the file name and the program name (no.), prepare the program for the workpiece according to the part drawing.

(3) Before actually machining them on the machine, it is necessary to determine the number of workpieces to be machined and the machining sequence of them - scheduling.

No. of workpieces:

A .... 20 pcs.

B .... 15 pcs.

C .... 25 pcs.

Machining sequence: A, B and C

(4) Now the programmer can make the schedule program.

Determine the file name of the schedule program. The file name of this example is determined as "SHAFT-1. SDF".

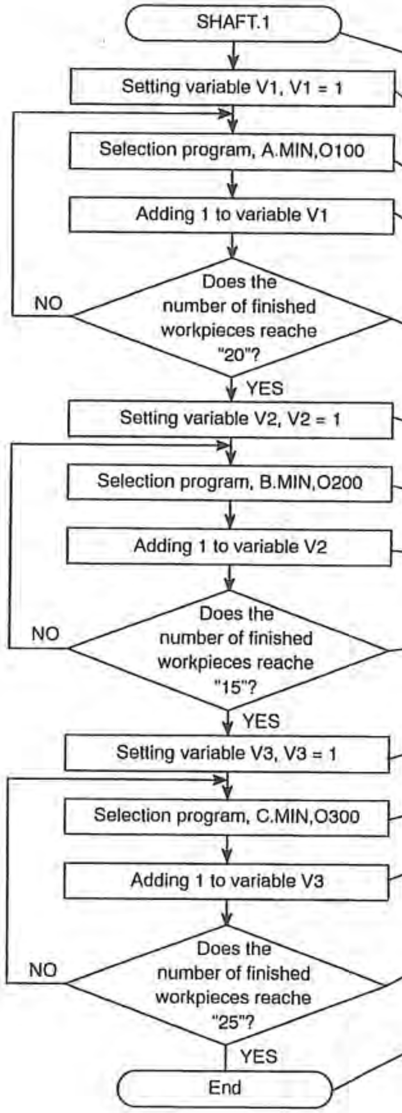
(5) Counting of the finished workpieces can be made using the common variables.

Workpiece A V1

Workpiece B V2

Workpiece C V3

(6) Prepare the schedule program according to the flow chart on the following page.



Schedule Program	
$ SHAFT-1. SDF	File name of schedule program
%	Always program "%" at the beginning of a program.
N1000_VSET_V1 = 1 (*1)	Initializes the work counter for Workpiece A to "1"
N1001_PSELECT_A.MIN,O100	Part program for Workpiece A is called and executed. (Note: PSELECT means Program SELECT.)
N1002_VSET_V1 = V1 + 1 (*1)	Adds "1" to the work counter.
N1003_IF (V1_LE_20) _N1001	While the counter data is less than or equal to 20, sequence jumps to N1001 and the same machining cycle is repeated. After 20 pcs. of workpieces have been machined, block N1004 is executed.
N1004_VSET_V2 = 1 (*1)	Initializes the work counter for Workpiece B to "1"
N1005_PSELECT_B.MIN,O200	Part program for Workpiece B is called and executed.
N1006_VSET_V2 = V2 + 1 (*1)	Adds "1" to the work counter.
N1007_IF (V2_LE_15) _N1005	While the counter data is less than or equal to 15, sequence jumps to N1005 and the same machining cycle is repeated. After 15 pcs. of workpieces have been machined, block N1008 is executed.
N1008_VSET_V3 = 1 (*1)	Initializes the work counter for Workpiece C to "1"
N1009_PSELECT_C.MIN,O300	Part program for Workpiece C is called and executed.
N1010_VSET_V3 = V3 + 1 (*1)	Adds "1" to the work counter.
N1011_IF (V3_LE_25) _N1009	While the counter data is less than or equal to 25, sequence jumps to N1009 and the same machining cycle is repeated. After 25 pcs. of workpieces have been machined, block N1012 is executed.
N1012_END	End code of schedule program

Main Program (Machining Program)

A.MIN,O100	B.MIN,O200	C.MIN,O300
%	%	%
N100 G00 X Z STM	N200 G00 X Z STM	N300 G00 X Z STM
:	:	:
N150 M02	N230 M02	N340 M02

(*1): To set a common variable in the schedule program, be sure to program VSET command which declares variable setting.

APPENDIX

1. EIA/ISO Code Table

EIA Code										ISO Code										Remarks	HEX
Channel / Character	8	7	6	5	4		3	2	1	Channel / Character	8	7	6	5	4		3	2	1		
0			O			o				0			O	O		o				Numerical character 0	30
1						o			O	1	O		O	O		o			O	Numerical character 1	B1
2						o		O		2	O		O	O		o		O		Numerical character 2	B2
3				O		o		O	O	3			O	O		o		O	O	Numerical character 3	33
4						o	O			4	O		O	O		o	O			Numerical character 4	B4
5				O		o	O		O	5			O	O		o	O		O	Numerical character 5	35
6				O		o	O	O		6			O	O		o	O	O		Numerical character 6	36
7						o	O	O	O	7	O		O	O		o	O	O	O	Numerical character 7	B7
8					O	o				8	O		O	O	O	o				Numerical character 8	B8
9				O	O	o			O	9			O	O	O	o			O	Numerical character 9	39
A		O	O			o			O	A		O				o			O	Alphabetical character A	41
B		O	O			o		O		B		O				o		O		Alphabetical character B	42
C		O	O	O		o		O	O	C	O	O				o		O	O	Alphabetical character C	C3
D		O	O			o	O			D		O				o	O			Alphabetical character D	44
E		O	O	O		o	O		O	E	O	O				o	O		O	Alphabetical character E	C5
F		O	O	O		o	O	O		F	O	O				o	O	O		Alphabetical character F	C6
G		O	O			o	O	O	O	G		O				o	O	O	O	Alphabetical character G	47
H		O	O		O	o				H		O			O	o				Alphabetical character H	48
I		O	O	O	O	o			O	I	O	O			O	o			O	Alphabetical character I	C9
J		O		O		o			O	J	O	O			O	o		O		Alphabetical character J	CA
K		O		O		o		O		K		O			O	o		O	O	Alphabetical character K	4B
L		O				o		O	O	L	O	O			O	o	O			Alphabetical character L	CC
M		O		O		o	O			M		O			O	o	O		O	Alphabetical character M	4D
N		O				o	O		O	N		O			O	o	O	O		Alphabetical character N	4E
O		O				o	O	O		O	O	O			O	o	O	O	O	Alphabetical character O (program number or program name)	CF
P		O		O		o	O	O	O	P		O		O		o				Alphabetical character P	50
Q		O		O	O	o				Q	O	O		O		o			O	Alphabetical character Q	D1
R		O			O	o			O	R	O	O		O		o		O		Alphabetical character R	D2
S			O	O		o		O		S		O		O		o		O	O	Alphabetical character S	53
T			O			o	O		O	T	O	O		O		o	O			Alphabetical character T	D4

EIA Code										ISO Code										Remarks	HEX
Character \ Channel	8	7	6	5	4		3	2	1	Character \ Channel	8	7	6	5	4		3	2	1		
U			O	O		o	O			U		O		O		o	O		O	Alphabetical character U	55
V			O			o	O		O	V		O		O		o	O	O		Alphabetical character V	56
W			O			o	O	O		W	O	O		O		o	O	O	O	Alphabetical character W	D7
X			O	O		o	O	O	O	X	O	O		O	O	o				Alphabetical character X	D8
Y			O	O	O	o				Y		O		O	O	o			O	Alphabetical character Y	59
Z			O		O	o			O	Z		O		O	O	o		O		Alphabetical character Z	5A
+		O	O	O		o				+			O		O	o		O	O	Plus Sign	2B
–		O				o				–			O		O	o	O		O	Minus Sign	2D
/			O	O		o			O	/	O		O		O	o	O	O	O	Block Delete or Division Symbol	5F
BLANK						o				NULL						o				Null	00
SPACE				O		o				SPACE	O		O			o				Space*	C0
TAB			O	O	O	o	O	O		HT					O	o			O	Tab	09
ER					O	o		O	O	%	O		O			o	O		O	Program Start	C5
CR/ EOB	O					o				NL/LF					O	o		O		End of Block	0A
						o				CR	O				O	o	O		O	Carriage Return*	8D
.		O	O		O	o		O	O	.			O		O	o	O	O		Period	2E
,			O	O	O	o		O	O	,	O		O		O	o	O			Comma	AC
DEL		O	O	O	O	o	O	O	O	DEL	O	O	O	O	O	o	O	O	O	Delete*	FF
BS			O		O	o		O		BS	O				O	o				Back Space*	88
&					O	o	O	O		&	O		O			o	O	O		Ampersand	A6
						o										o					
(				O	O	o		O		(			O		O	o				Control Out (comment start)	28
)		O			O	o		O		)	O		O		O	o			O	Control In (comment end)	A9
										$			O			o	O			Dollar Sign (comment block)	24
										*	O		O		O	o		O		Multiplication Symbol	AA
										=	O		O	O	O	o	O		O	Equal Sign	BD
										?			O	O	O	o	O	O	O	Question Mark	3F
										[	O	O		O	O	o		O	O	Bracket, Left	DB
										]	O	O		O	O	o	O		O	Bracket, Right	DD
										:			O	O	O	o		O		Colon (program number or program name)	3A
										;	O		O	O	O	o		O	O	Semicolon	BB

* Ignored when used as cutting data.

2. List of G Codes

◎: 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 command	◎
G21	ATC home position command	◎
G22	Torque skip command	◎
G23		
G24		
G25		
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 rod synchronized feeding (forward)	◎
G37	M-tool spindle - feed rod synchronized feeding (reverse)	◎
G38		
G39		
G40	Tool nose radius compensation: Cancel	
G41	Tool nose radius compensation: Left	
G42	Tool nose radius compensation: Right	
G43		
G44		
G45		
G46		
G47		
G48		
G49		
G50	Zero offset, Maximum spindle speed designation	
G51		
G52		
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	

G Code	Contents	
G77	Tapping compound fixed cycle	
G78	Tapping cycle reverse thread	
G79		
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 command 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		

G Code	Contents	
G118		
G119	Cutter radius 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)	◎
G124		
G125		
G126		
G127		
G128		
G129		
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 made 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 pickoff spindle	◎
G143	Designation of machining mode using pickoff spindle and 3rd turret	◎
G144		
G145		
G146		
G147		
G152	Programmable tailstock positioning (tow-along tailstock)	◎
G153		
G154		
G155		
G156		
G157		
G158		
G159		
G160		
G161	G code macro function MODIN	◎
G162	G code macro function MODIN	◎

G Code	Contents	
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	◎
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 (forward)	◎
G179	Synchronized tapping cycle (reverse)	◎
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		

G Code	Contents	
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	◎
G212	G code macro function CALL	◎
G213	G code macro function CALL	◎
G214	G code macro function CALL	◎

3. List of M Codes

◎: Optional
Others: Standard

G Code	Contents	
M00	Program stop	
M01	Optional stop	
M02	End of program	
M03	Spindle CW	
M04	Spindle CCW	
M05	Spindle stop	
M06	Tool change	◎
M07		
M08	Coolant ON	
M09	Coolant OFF	
M10		
M11		
M12	M-tool spindle stop	◎
M13	M-tool spindle CW	◎
M14	M-tool spindle CCW	◎
M15	C-axis positioning (positive direction)	◎
M16	C-axis positioning (negative direction)	◎
M17	CEJ MATIC : Request for data transfer	◎
M18	Post-process gauging : RS232C request for data transfer	◎
M19	Oriented spindle stop	◎
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 (on left face)	
M33	Zig-zag infeed in thread cutting	
M34	Straight infeed along thread face mode (on right face)	

G Code	Contents	
M35	Loader gripper Z slide retract	◎
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 coil selection	
M42	Spindle gear range 2 or high-speed coil 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		
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	◎

G Code	Contents	
M76	Parts catcher retract	◎
M77	Parts catcher advance	◎
M78	Steady rest unclamp	◎
M79	Steady rest clamp	◎
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 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 pickoff 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	◎

G Code	Contents	
M116	Unloader close	◎
M117	Sensor head advance	◎
M118	Sensor head retract	◎
M119	Work count special	◎
M120	No work	◎
M121		
M122	Steady rest retract	◎
M123	Steady rest forward	◎
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	Designation of multiple fixed cycle configuration	◎
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	Pickoff spindle CW rotation	◎
M149	Pickoff spindle CCW rotation	◎
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)	◎

G Code	Contents	
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	
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 no clamp	◎
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	M-tool spindle orientation direction specified CW	◎
M192	M-tool spindle orientation direction specified CCW	◎
M193	Cancel of M194	◎
M194	Phasing for thread cutting	◎
M195	Cancel of M196	◎
M196	Thread cutting phasing stroke effective	◎

G Code	Contents	
M197	Thread cutting phasing stroke clear	◎
M198		
M199		
M200	Z-axis synchronized feeding cancel	◎
M201	Z-axis synchronized feeding G13	◎
M202	Z-axis synchronized feeding 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	
M210		
M211	Keyway cutting cycle: Minus direction	◎
M212	Keyway cutting cycle: Zigzag	◎
M213	Keyway cutting cycle: Designated depth infeed	◎
M214	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	◎

G Code	Contents	
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	M-tool spindle: 3rd range	◎
M244	M-tool spindle: 4th range	◎
M245		
M246	Pick-off interlock ON	◎
M247	Pick-off interlock OFF	◎
M248	Pick-off close	◎
M249	Pick-off open	◎
M250	Work pusher retract	◎
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	Restriction on overlap mode rapid feed release OFF	
M265	Restriction on overlap mode rapid feed release ON	
M266		
M267		
M268		
M269		
M270		
M271		
M272		
M273		
M274		
M275		
M276		
M277		

G Code	Contents	
M278		
M279		
M280		
M281		
M282		
M283		
M284		
M285		
M286		
M287		
M288	Air blow for opposite chuck OFF	
M289	Air blow for opposite chuck ON	
M290	Ceiling door close	
M291	Ceiling door open	
M292		
M293		
M294		
M295		
M296		
M297		
M298		
M299		
M300		
M301		
M302		
M303		
M304		
M305		
M306		
M307		
M308	Robot/loader interface interlock release OFF	
M309	Robot/loader interface interlock release ON	
M310		
M311		
M312		
M313		
M314		
M315		
M316		
M317		
M318		

G Code	Contents	
M319		
M320		
M321	M code macro function CALL (fixed sub-program)	
M322	M code macro function CALL (fixed sub-program)	
M323	M code macro function CALL (fixed sub-program)	
M324	M code macro function CALL (fixed sub-program)	
M325	M code macro function CALL (fixed sub-program)	
M326		
M327		
M328		
M329		
M330		
M331		
M332		
M333		
M334	Machine interference OFF	
M335	Machine interference ON	
M336		
M337		
M338		
M339		
M340		
M341	M code macro function CALL	
M342	M code macro function CALL	
M343	M code macro function CALL	
M344	M code macro function CALL	
M345	M code macro function CALL	
M346	M code macro function CALL	
M347	M code macro function CALL	
M348	M code macro function CALL	
M349	M code macro function CALL	
M350	M code macro function CALL	
M351	M code macro function CALL	
M352	M code macro function CALL	
M353	M code macro function CALL	
M354	M code macro function CALL	
M355	M code macro function CALL	
M356	M code macro function CALL	
M357	M code macro function CALL	
M358	M code macro function CALL	
M359	M code macro function CALL	

G Code	Contents
M360	M code macro function CALL
M361	Work stopper retract
M362	Work stopper advance

4. List 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
VTOFY	Z-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 rpm)	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	Active 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	
VTLOB	Tool offset number (group 2)	0 to 64	
VTLOC	Tool offset number (group 3)	0 to 96	
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	
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	
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		

Variables	Contents	Setting Range	Suffix
VTOAB	Tool offset number A of tool at ATC 2nd position	0 to 96	1 to 96
VTOBB	Tool offset number B of tool at ATC 2nd position		
VTOCB	Tool offset number C of tool at ATC 2nd position		
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	None
VBNCT	Block number count or not count		
VPWSP	Parts catcher work chute position	0 to ±99999.999	
VPWTP	Parts catcher work transfer position		
VTLIN	Tool classification code number	1 to 38	1 to 12 1 to 20 1 to 96
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	0 to 9999.999	
VTLD	Tool holder dia./drill dia.		
VTLW	Tool width		
VTIZN	Tool interference point; ZN		
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	1 to 12 1 to 20 1 to 24
VGRFN	Tool form code number	0 to 4	
VGRA1	Tool nose angle	0 to 360.000	
VGRA2	Cutting edge angle	0 to ±360.000	
VGRL	Tool holder length/projection/drill length	0 to 9999.999	
VGRD	Tool holder dia./drill dia.		
VGRW	Tool width		
VSIDC	Spindle orientation (pin type/electric type)	0/1	None
VEXPO	RS232C post-process gauging point	0 to 9	1 to 12
VEXTR	RS232C post-process gauging turret	0/1	
VEXAX	RS232C post-process gauging axis		
VEXGF	RS232C post-process gauging group flag		
VEXTO	RS232C post-process gauging tool offset number	0 to 32	
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 active point (machine coordinate system)		
VAPAX	X-axis active point (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)		1/2
VETFZ	Tool offset amount for active tool in Z-axis		None
VETFY	Tool offset amount for active tool in Y-axis		
VETFX	Tool offset amount for active tool in X-axis		
VDIFZ	DIF in Z-axis		
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		
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 protrusion L2	0 to 9999.999	
VTSDA	Tailstock spindle diameter D2		
VTSDB	Tailstock center diameter D3		
VWKR	Workpiece end face	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	ZA-axis designation position (parameter long-word No. 68)	0 to 99999999	
VZBRP	ZB-axis designation position (parameter long-word No. 69)		
VZCRP	ZC-axis designation position (parameter long-word No. 70)		
VXARP	XA-axis designation position (parameter long-word No. 74)		
VXBRP	XB-axis designation position (parameter long-word No. 75)		
VWAP	W-axis designation position (parameter long-word No. 65)		

Variables	Contents	Setting Range	Suffix
VSNWD	Dislocation between the sensor center and the sensor head in rotating the C-axis forward rotation	READ ONLY	None
VSNTU	Dislocation between the sensor center and the sensor head in rotating 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)

REVISION HISTORY

Publication No.	Date	Edition
4197-E	March 1998	1st