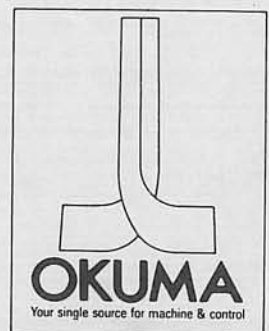


CNC SYSTEMS

OSP500L-G
OSP5000L-G

PROGRAMMING MANUAL (5th Edition)



Publication No. 2452-E-R5
(May 1987)

CNC SYSTEMS
OSP500L-G
OSP500L-G

PROGRAMMING MANUAL (5th Edition)



Publication No. 2452-E-R5
(May 1987)

TABLE OF CONTENTS

	<u>PAGE</u>
SECTION 1 BEFORE YOU BEGIN PROGRAMMING	1
1-1. COORDINATE SYSTEM	2
1-1-1. Model LC1 \emptyset	2
1-1-2. Models LC2 \emptyset /LC3 \emptyset /LC4 \emptyset /LC5 \emptyset	3
1-1-3. Models LS3 \emptyset -N and LH35-N	4
1-1-4. Model LH55-N	5
1-1-5. Models LB1 \emptyset /LB12/LB15	6
1-1-6. LR Series Models	7
1-2. REFERENCE POINT	8
1-3. PROGRAMMING ZERO	9
1-4. DIMENSION WORDS	10
1-5. TURRET SELECTION FOR 2-SADDLE MODELS	13
1-5-1. LC Series	13
1-5-2. LS3 \emptyset -N, LH35-N and LH55-N	14
1-5-3. LR Series	16
1-6. TOOL SELECTION	17
1-7. SPINDLE SPEED SELECTION	27
1-7-1. LC1 \emptyset	27
1-7-2. LC2 \emptyset	31
1-7-3. LC3 \emptyset	32
1-7-4. LC4 \emptyset	34
1-7-5. LC5 \emptyset	35
1-7-6. LS3 \emptyset -N	37
1-7-7. LH35-N	38
1-7-8. LH55-N	39
1-7-9. LB1 \emptyset	40
1-7-10. LB12	41
1-7-11. LB15	43
1-7-12. LR15	44
1-8. SPINDLE ROTATION DIRECTION SELECTION	46
1-9. TAPE FORMAT	47
1-9-1. Program Process Sheet	47
1-9-2. Tape Format	48
1-9-3. Position of Decimal Point	50
1-9-4. Position of Decimal Point for $\emptyset.1 \mu\text{m}$ Specification (optional) ..	52
1-9-5. Tape Code	53
1-9-6. EIA Special Codes	55
1-9-6-1. Setting of EIA Special Characters	56
1-9-6-2. Input/Output of The Special EIA Codes	56

	<u>PAGE</u>
1-10. DETERMINING CUTTING CONDITIONS	58
1-11. PROGRAMMING CODES AND WORD FORMAT (1 mm TAPE INPUT UNIT SYSTEM)	59
1-12. PROGRAMMING FORMAT	72
1-12-1. Model 1S (Single Turret Model)	72
1-12-2. Model 2S (Two-Turret/Two-Saddle Model)	73
 SECTION 2 SIMPLE PROGRAMMING	 75
2-1. POSITIONING (G00)	76
2-2. STRAIGHT-LINE CUTTING (G01)	77
2-3. TAPER CUTTING (G01)	80
2-4. ARC CUTTING (G02/G03)	82
2-5. THREAD CUTTING (STRAIGHT, CONSTANT LEAD THREAD) (G33)	87
 SECTION 3 APPLICATION PROGRAM	 90
3-1. PROGRAMMING SPECIAL THREADS	91
3-1-1. G33 (Fixed Thread Cutting Mode)	91
3-1-2. G32 (Fixed Thread Cutting on End Face Mode)	93
3-1-3. G34 and G35 (Non-Fixed Cycle)	96
3-2. PRECAUTIONS FOR PROGRAMMING THREAD CUTTING CYCLES	97
3-3. PROGRAMMING CONSTANT SPEED CUTTING OPERATION	104
3-4. INCREMENTAL PROGRAMMING	106
3-5. SUPPLEMENT	107
 SECTION 4 PROGRAMMING EXAMPLE	 112
4-1. PROGRAM EXAMPLE FOR CENTER-WORK	112
4-2. PROGRAM EXAMPLE FOR CHUCK-WORK	116
 SECTION 5 PROGRAMMING TAKING TOOL NOSE RADIUS INTO CONSIDERATION	 119
5-1. PROGRAMMING EXAMPLE FOR TAPER CUTTING	121
5-2. PROGRAMMING EXAMPLE FOR ARC CUTTING	122
5-3. PROGRAMMING EXAMPLES WITH TOOL NOSE RADIUS COMPENSATION	124
5-4. GENERAL PROGRAMMING PROCEDURE FOR CIRCULAR INTERPOLATION	125
 SECTION 6 PROGRAMMING FOR SIMULTANEOUS 4-AXIS CUTS (2S Model for LC)	 127
6-1. PROGRAMMING FORMAT	127
6-2. PRECAUTIONS FOR PROGRAMMING SIMULTANEOUS 4-AXIS CUTS	130

	<u>PAGE</u>
6-3. PROGRAMMING EXAMPLE	132
6-3-1. Workpiece Dimensions	132
6-3-2. Tooling and Cutting Conditions	132
6-3-3. Program Process Sheet	133
6-4. WAITING SYNCHRONIZATION M CODE (M100) FOR SIMULTANEOUS CUTS	134
SECTION 7 BARRIER CHECK FUNCTION	135
7-1. GENERAL DESCRIPTION	135
7-2. CHUCK BARRIER AND TAILSTOCK BARRIER	136
7-2-1. Establishing Chuck Barrier	136
7-2-2. Establishing Tailstock Barrier	137
7-2-3. Tool Movements and Alarm	138
7-2-4. Precautions	138
7-3. TOOL INTERFERENCE CHECK	139
7-3-1. Programming	139
7-3-2. Setting of Tool Interference Area	140
SECTION 8 DROOP CONTROL	148
SECTION 9 MM/MIN. MODE FEEDRATE COMMAND	149
SECTION 10 ZERO SET BY G CODE	150
SECTION 11 TAPER ANGLE/ARC RADIUS DIRECT COMMAND	152
11-1. GENERAL DESCRIPTION	152
11-2. DIRECT TAPER ANGLE COMMAND	152
11-3. DIRECT RADIUS COMMAND	154
SECTION 12 AUTOMATIC CHAMFERING	156
12-1. GENERAL DESCRIPTION	156
12-2. 45 DEG. CHAMFERING (G75)	156
12-3. ROUNDING (G76)	159

	<u>PAGE</u>
SECTION 13 COMPOUND FIXED CYCLE (SPECIAL FIXED CYCLE)	162
13-1. GENERAL DESCRIPTION	162
13-2. THREAD CUTTING COMPOUND CYCLE (G71/G72)	163
13-2-1. Longitudinal Thread Cutting Cycle (G71)	163
13-2-2. Transverse Thread Cutting Compound Fixed Cycle (G72)	165
13-2-3. M Code Specifying Thread Cutting Mode and Infeed Pattern	167
13-2-4. P Codes in the G71 and G72 Thread Cutting Cycles	174
13-3. GROOVING/DRILLING COMPOUND FIXED CYCLE (G73/G74)	175
13-3-1. Longitudinal Grooving Fixed Cycle (G73)	175
13-3-2. Transverse Grooving Fixed Cycle (G74)	177
13-3-3. Axis Movements in Grooving Compound Fixed Cycle	179
13-3-4. P Codes in the G73 and G74 Grooving Cycles	180
13-4. TAPPING COMPOUND FIXED CYCLE (G77, G78)	181
13-4-1. Right-hand Tapping Cycle (G77)	181
13-4-2. Left-hand Tapping Cycle (G78)	183
13-4-3. Supplement	184
13-5. APPLICATION OF COMPOUND FIXED CYCLE	185
13-5-1. Application of Longitudinal Thread Cutting Compound Fixed Cycle	185
13-5-2. Application of Longitudinal Grooving Compound Fixed Cycle (G73)	186
13-5-3. Application of Transverse Grooving Compound Fixed Cycle (G74) ..	186
SECTION 14 MIRROR IMAGE FUNCTION (2-Turret Model)	187
14-1. OUTLINE OF MIRROR IMAGE FUNCTIONS	187
14-2. OPERATIONS	188
14-3. PROGRAM AXIS MOTIONS	189
14-3-1. G Codes	189
14-3-2. Cautions on Programming	190
14-3-3. Cutting Program	191
14-3-4. Cutting Operation	192
14-4. MACHINE OPERATION AND DISPLAY	193
14-4-1. Operation Panel	193
14-4-2. Zero Offset	193
14-4-3. Tool Offset	194
14-5. OTHERS	195
14-6. PROGRAM EXAMPLE	196

	<u>PAGE</u>
SECTION 15 USER TASK 1	198
15-1. OUTLINE OF USER TASK FUNCTION	198
15-2. FUNDAMENTAL FUNCTIONS OF USER TASK	199
15-2-1. Control Statement	200
15-2-1-1. GOTO Statement - Unconditional Jump	201
15-2-1-2. IF Statement - Conditional Jump	201
15-2-2. Common Variable	203
15-2-2-1. Common Variable	203
15-2-2-2. Local Variables	204
15-2-2-3. System Variables	205
15-2-3. Arithmetic Operation Function	212
15-2-3-1. Arithmetic Expression	212
15-2-3-2. Comparison Expression	212
15-3. PROGRAM EXAMPLE	213
15-4. COMPARISON OF USER TASK 1 AND USER TASK 2	215
SECTION 16 TAPE CONVERSION FUNCTION (OSP3000L → OSP500L-G/OSP5000L-G) ...	216
16-1. OUTLINE	216
16-2. OPERATION	216
16-3. CONTENTS OF CONVERSION	218
16-3-1. Simultaneous 4-Axis Control Program	218
16-3-2. Constant Cutting Speed Program	219
16-3-3. Address Characters Specifying Variable Pitch	220
16-3-4. Endless Operation	220
16-3-5. Nose Radius Compensation Function	221
16-3-6. LAP	221
16-4. RESTRICTIONS ON TAPE CONVERT FUNCTION (OSP3000L → OSP5000L-G)	224
SECTION 17 TAILSTOCK SPINDLE ADVANCE/RETRACT FUNCTION BY M CODES	227
17-1. M CODES USED FOR ADVANCING/RETRACTING TAILSTOCK SPINDLE	227
17-2. ALARM MESSAGES	227
SECTION 18 SCHEDULE PROGRAM	229
18-1. OUTLINE	229
18-2. PROGRAMMING	229
18-3. PROGRAM EXAMPLE	231

	<u>PAGE</u>
SECTION 19 LATHE AUTO-PROGRAMMING FUNCTION (LAP)	232
19-1. GENERAL DESCRIPTION	232
19-2. CLASSIFICATION OF FUNCTIONS	233
19-2-1. Classification of Cutting Cycle	233
19-2-2. G Code Used to Designate Cutting Mode	234
19-3. PROGRAM FORMAT	235
19-3-1. G Codes	235
19-3-2. M Codes	235
19-3-3. Parameters	236
19-4. EXECUTION MODE OF LAP	238
19-4-1. G85: Bar Turning Cycle	238
19-4-2. G84: Change of Cutting Conditions in Bar Turning Cycle	239
19-4-3. G86: Copy Turning	240
19-4-4. G87: Finish Cut Cycle	241
19-4-5. G88: Continuous Thread Cutting Cycle	242
19-5. EXPLANATION OF LAP FUNCTIONS AND PROGRAM	244
19-5-1. AP Mode I (Bar Turning)	244
19-5-2. AP Mode II (Copy Turning)	256
19-5-3. AP Mode III (Continuous Thread Cutting Cycle)	262
19-6. PRECAUTIONS	275
19-7. APPLICATION OF LAP FUNCTION	276
19-8. SYNCHRONIZATION CODES FOR LAP	278
SECTION 20 TOOL NOSE RADIUS COMPENSATION FUNCTION	280
20-1. OUTLINE	280
20-1-1. General Description	280
20-2. TOOL NOSE RADIUS COMPENSATION FOR TURNING OPERATION	280
20-2-1. Tool Offset and Nose Radius Compensation	280
20-2-2. Tool Nose Radius Compensation at Discontinuous Point	281
20-3. COMPENSATION OPERATION	282
20-3-1. Geometrical Cutting Error Due to Tool Nose Radius	282
20-3-2. Compensation Movement	283
20-4. PROGRAMMING	284
20-4-1. G Codes	284
20-4-2. T Codes	285
20-4-3. Program Example	286

	<u>PAGE</u>
20-5. TOOL NOSE RADIUS COMPENSATION DATA	287
20-5-1. Measuring Nose Radius	287
20-5-2. Setting Compensation Amounts	289
20-5-3. Relationship between Imaginary Tool Tip Orientations and Coded Numbers	292
20-6. DISPLAY	294
20-6-1. Actual Position	294
20-6-2. Alarm Display	294
20-7. OTHER REMARKS	295
20-7-1. Conditions of Nose Radius Compensation Function	295
20-7-2. Path of Tool Nose "R" Center in Tool Nose Radius Compensation Mode	297
SECTION 21 APPLICATION OF TOOL NOSE RADIUS COMPENSATION PROGRAMMING	299
21-1. G41 AND G42	299
21-2. BEHAVIOR ON TOOL NOSE RADIUS COMPENSATION MODE ENTRY	300
21-2-1. General Description	300
21-2-2. Program Examples	301
21-2-3. I and K Command with G41 and G42	304
21-3. BEHAVIOR IN TOOL NOSE RADIUS COMPENSATION MODE	305
21-3-1. Straight Line to Straight Line Cutting	305
21-3-2. Straight Line to Arc Cutting (Arc to Straight Line Cutting)	315
21-3-3. Arc to Arc Cutting	319
21-3-4. Switching from G41 to G42 or from G42 to G41	320
21-4. BEHAVIOR WITH TOOL NOSE RADIUS COMPENSATION MODE CANCEL	323
21-4-1. G40 Given with X- or Z-axis Motion Command	323
21-4-2. I and K Command with G40	326
21-4-3. Independent G40	327
21-5. RELIEVING TOOL TO CHANGE "S" OR "M" CODE DURING CUTTING	328
21-5-1. Program 2	329
21-5-2. Program 3	330
21-5-3. Program 4	331
21-5-4. Program 5	323
21-5-5. Program 6	333
21-6. PRECAUTIONS	334
SECTION 22 USER TASK 2	335
22-1. OUTLINE OF USER TASK FUNCTION	335
22-2. TYPES OF USER TASK FUNCTION	337
22-2-1. Relation between Types of Program Files and User Task Functions	337

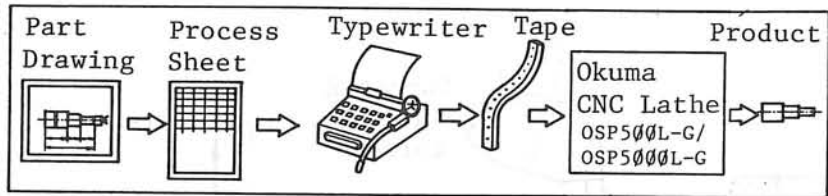
	<u>PAGE</u>
22-3. FUNDAMENTAL FUNCTIONS OF USER TASK	338
22-3-1. Control Statement	339
22-3-2. Variables	354
22-3-3. Arithmetic Operation Function	369
22-3-4. Combination of Operations	372
22-4. PROGRAM EXAMPLES	373
22-5. SUPPLEMENTS	384
22-5-1. Sequence Return in Program Using User Task	384
22-5-2. Simultaneous 4-Axis Control Mode	384
22-5-3. Rules of Operation and Evaluation of Result	385
SECTION 23 INCH/METRIC SWITCHABLE	390
SECTION 24 GRAPHIC CRT DISPLAY	393
24-1. OUTLINE OF GRAPHIC CRT DISPLAY	393
24-2. FUNCTION KEYS USED FOR GRAPHIC DISPLAY OPERATION	394
24-3. GENERAL PRECAUTIONS FOR ANIMATION MODE DISPLAY	401
24-4. HIGH SPEED DRAWING	402
24-5. DISPLAY PAGE	402
24-6. MACHINE TIME DISPLAY	403
24-7. ADDITIONAL FUNCTIONS FOR MULTI-MACHINING MODELS	403
24-8. TOOL FORM SELECTION	408
24-9. DRAWING BLANK SHAPE	411
24-10. HOW TO SET DRAWING POSITIONS OF CHUCK AND TAILSTOCK	423
24-11. SYSTEM VARIABLES FOR ANIMATION DISPLAY	426
24-12. BLANK MATERIAL DRAWING METHOD	430
SECTION 25 CHUCK AUTO OPEN/CLOSE BY M CODES	435
25-1. M CODES USED FOR OPENING/CLOSING CHUCK	435
25-2. ALARM MESSAGES	435

Note: To avoid any confusion over the use of the letter "O (oh)" and figure "0 (zero)" in this manual, the numerical value "0 (zero)" is expressed as "Ø" if there is any possibility of misunderstanding.

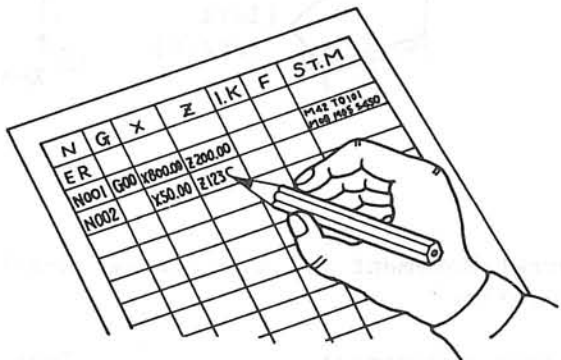
SECTION 1 BEFORE YOU BEGIN PROGRAMMING

On the CNC lathes, turning operations and various miscellaneous functions are performed by the information on a tape instead of manual operation by an operator, and the manual handling of levers and handwheels is performed by motors. In other words, by simply pressing a button, tape-controlled lathes work like automatics.

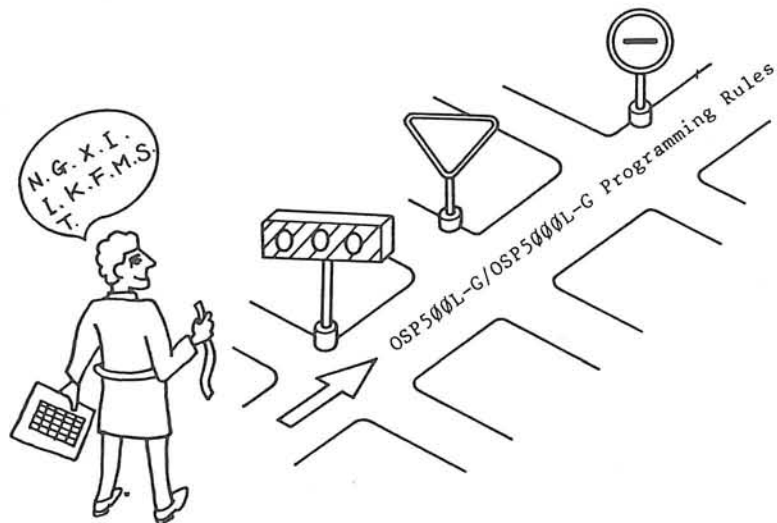
To make a command tape, first complete a program process sheet from a part drawing. After that, punch out a command tape by a tape-puncher.



The program process sheet is a manuscript of listing which details part programming instructions. The procedure to complete a program process sheet is called "PART PROGRAMMING".



There are certain rules in making a process sheet. These rules have to be remembered by the programmer because they are just like traffic rules for drivers.



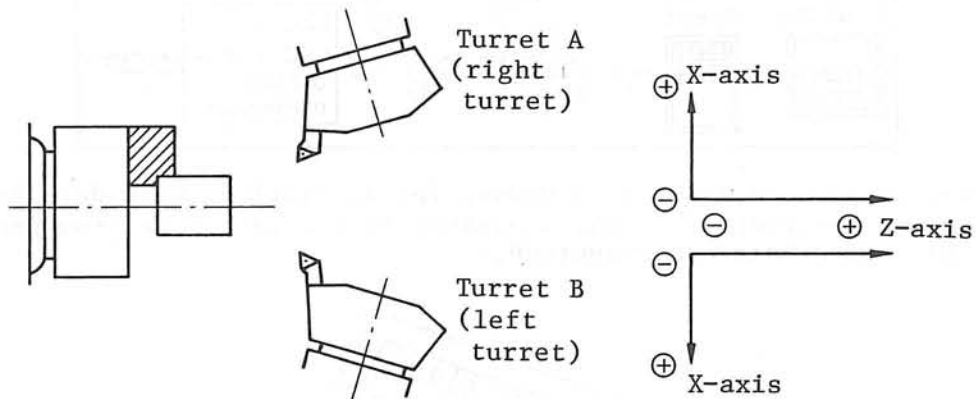
1-1. COORDINATE SYSTEM

The axis nomenclature of individual models is explained in detail in this section.

1-1-1. Model LC1Ø

The axis designation of LC1Ø CNC Lathe is as follows:

2S



Transverse Turret Movement X-axis
(infeeding direction)

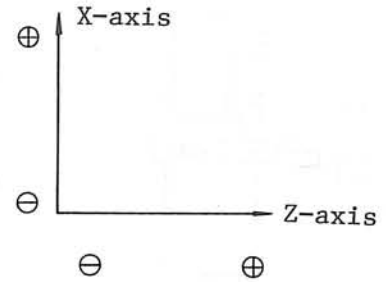
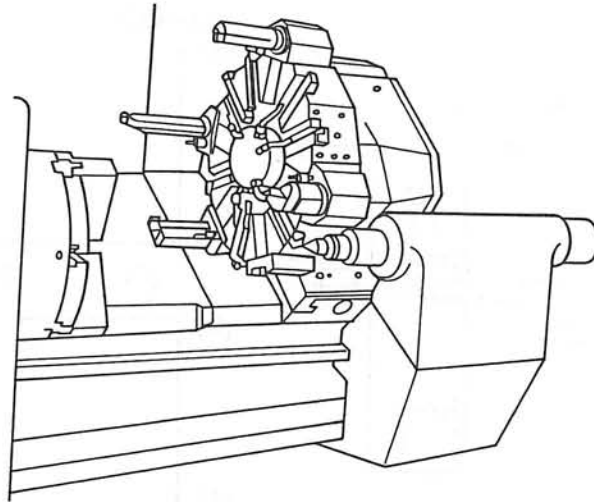
Longitudinal Turret Movement Z-axis

Direction of axis movement is defined by "+" and "-" sign characters.

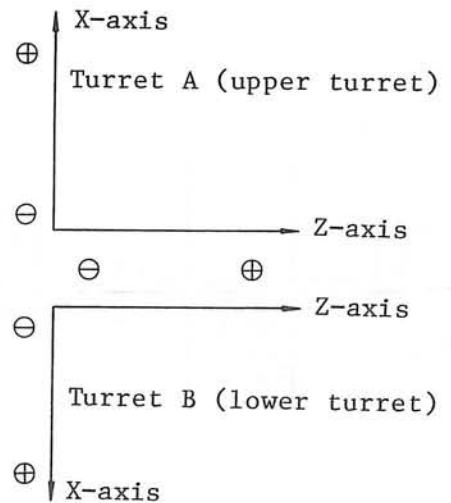
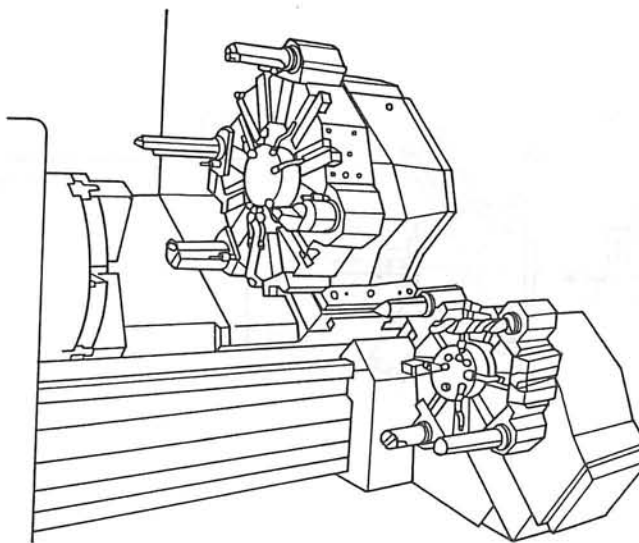
1-1-2. Models LC2Ø/LC3Ø/LC4Ø/LC5Ø

The axis designation of LC2Ø/LC3Ø/LC4Ø/LC5Ø CNC Lathe is as follows:

1S



2S



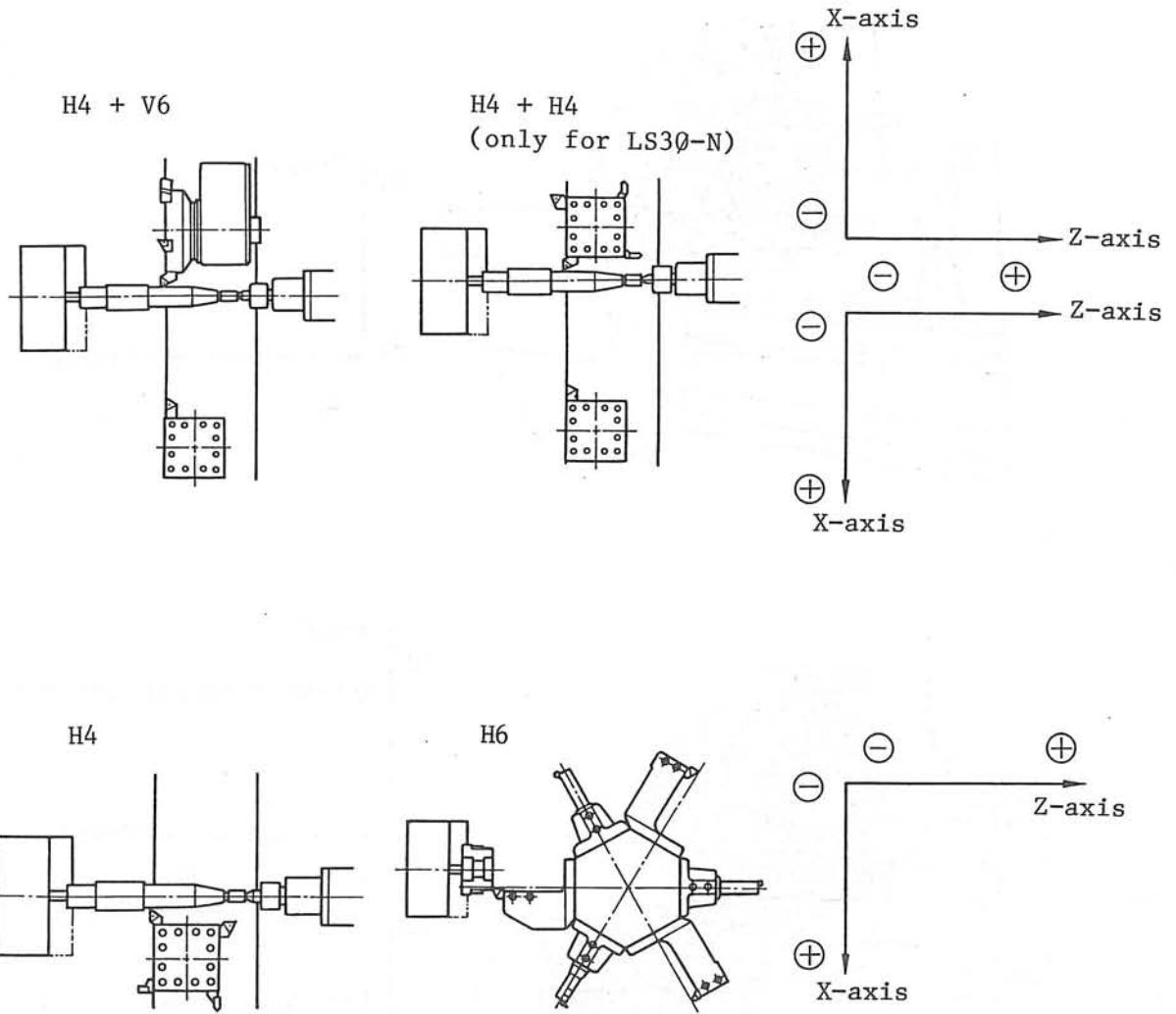
Transverse Turret Movement X-axis
(infeeding direction)

Longitudinal Turret Movement Z-axis

Direction of axis movement is defined by "+" and "-" sign characters.

1-1-3. Models LS30-N and LH35-N

The axis designation of LH35-N and LS30-N CNC Lathe is as follows:



Transverse Turret Movement X-axis
(infeeding direction)

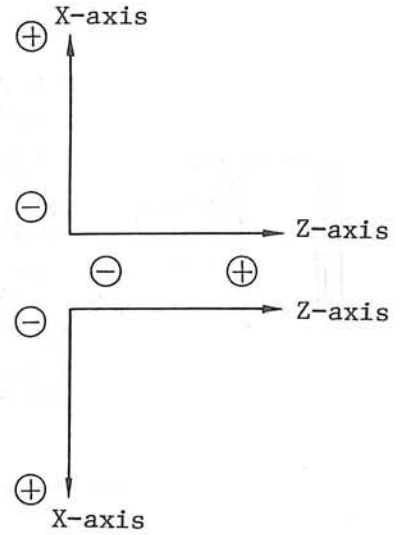
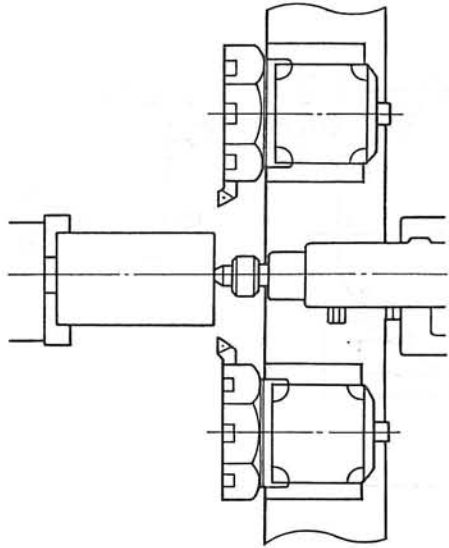
Longitudinal Turret Movement Z-axis

Direction of axis movement is defined by "+" and "-" sign characters.

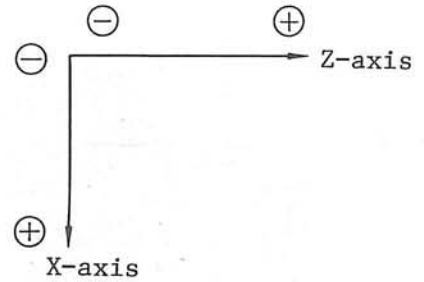
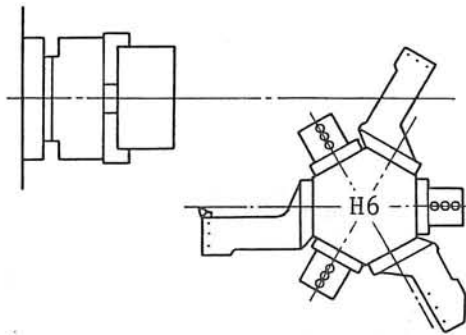
1-1-4. Model LH55-N

The axis designation of LH55-N CNC Lathe is as follows:

V8 + V8



H6



Transverse Turret Movement X-axis
(infeeding direction)

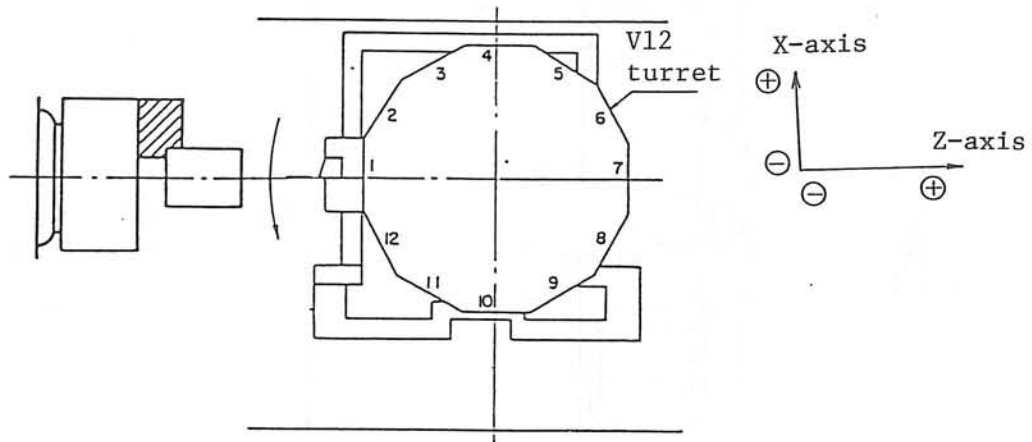
Longitudinal Turret Movement Z-axis

Direction of axis movement is defined by "+" and "-" sign characters.

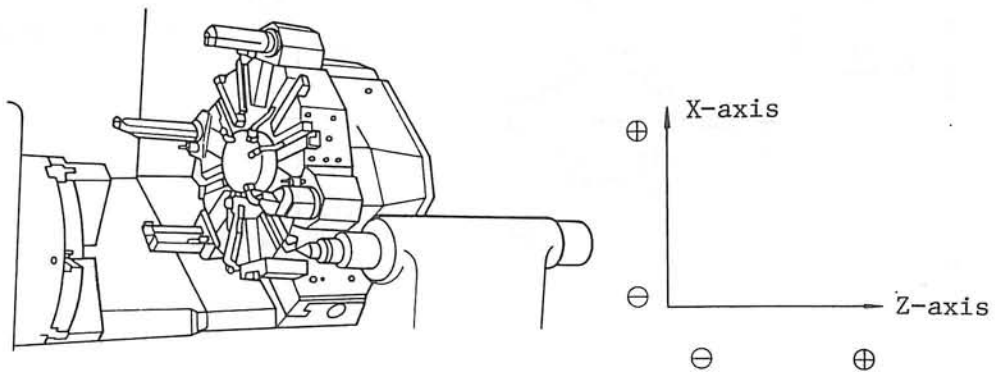
1-1-5. Models LB10/LB12/LB15

The axis designation of LB10/LB12/LB15 CNC Lathe is as follows:

LB10



LB12/LB15



Transverse Turret Movement X-axis
(infeeding direction)

Longitudinal Turret Movement Z-axis

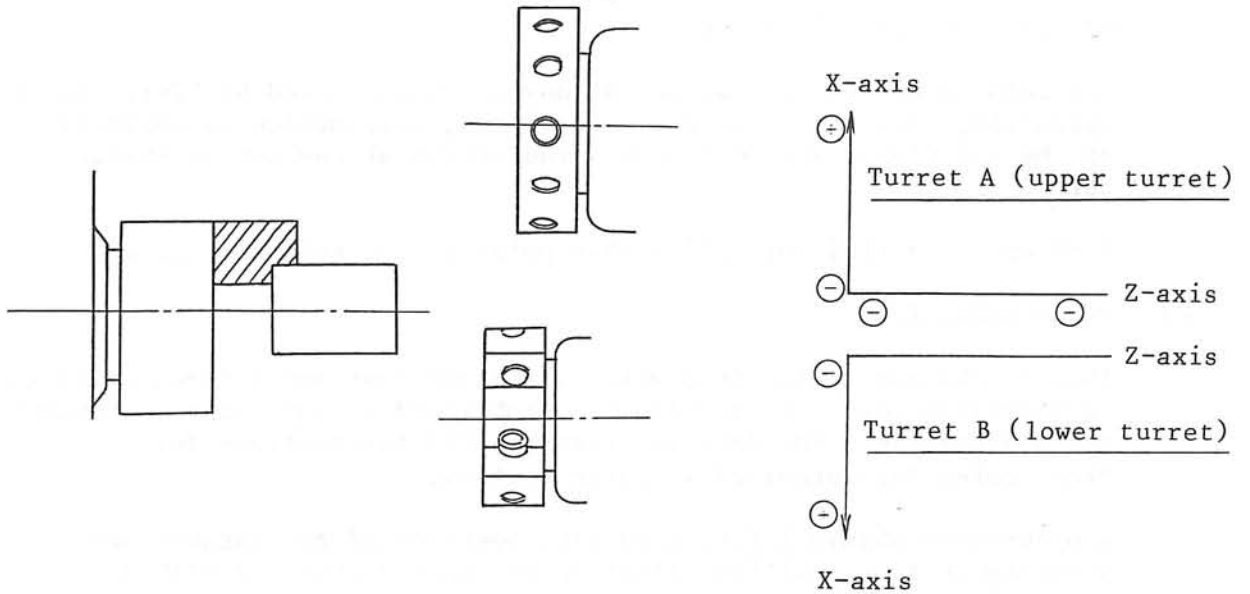
Direction of axis movement is defined by "+" and "-" sign characters.

1-1-6. LR Series Models

The axis designation of LR series CNC Lathe is as follows:

LR15

V10 + V8



Transverse Turret Movement X-axis
(infeeding direction)

Longitudinal Turret Movement Z-axis

Direction of axis movement is defined by "+" and "-" sign characters.

1-2. REFERENCE POINT

On the CNC Lathes, there are three types of reference points as below:

(1) Machine Origin

This is the fixed origin inherent in respective machines, where output values from the OSP position encoders of X- and Z-axis become zero. Axis movement of the machine is referenced to this fixed machine origin.

(2) Workpiece Origin (Machining Zero)

The zero point that can be set at any point as needed by "Zero Offset" operation; once the workpiece zero is set, axis motion is controlled on the coordinate system that has the origin at the set workpiece zero.

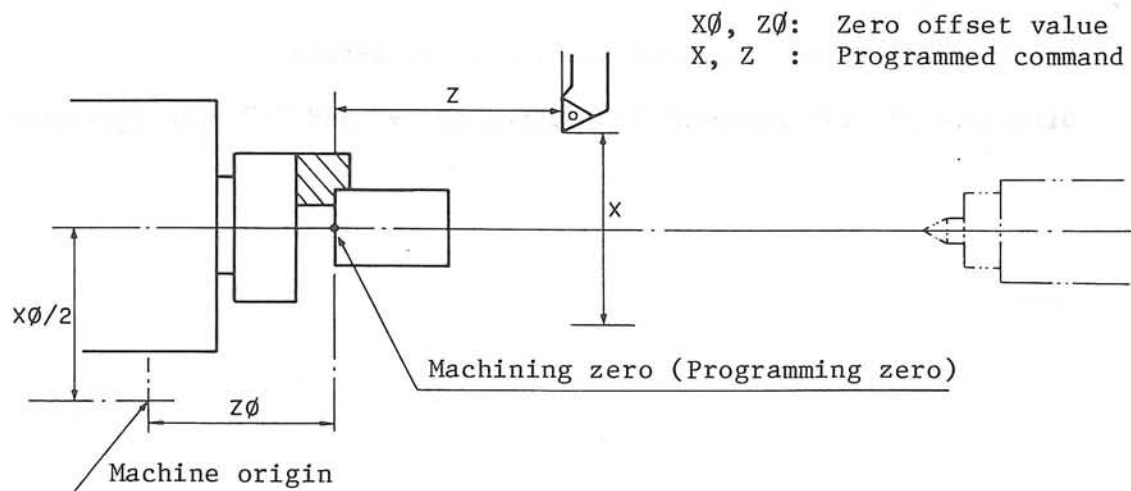
Cutting is carried out taking this point as the reference point.

(3) Programming Zero

This is the zero point from which all dimensions are referenced in the part programming. It can also be established at any point convenient for programming. For details, refer to the instructions for Programming Zero provided in later section.

A programmer should inform a machine operator of the established programming zero position either on the part drawing or with a memorandum.

Note: PROGRAMMING ZERO AND MACHINING ZERO MUST BE MATCHED.



Note 1: On two-saddle models, individual turrets, turret A and turret B, have their own machine origins independently.

- LC series
- LR series

Note 2: On two-turret models, individual turrets, front and rear, have their own machine origins independently (mirror image function).

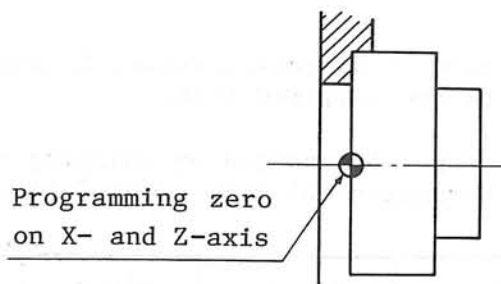
- LS30-N
- LH35-N
- LH55-N

1-3. PROGRAMMING ZERO

The Programming Zero is the reference point for programming and may be established at any required point within a specified range.

Usually, it is taken at the center of the workpiece for X-axis and at the left-hand end of it for Z-axis, and programming is made on the coordinate system which has the origin at such point.

See the figure below showing the preferable zero position.



1-4. DIMENSION WORDS

(1) Programmable Units

Axis movement commands, i.e., dimension words are expressed in the units specified below:

Axis	Programmable Units	
	Metric System	Inch System (optional)
X	1 mm	1 inch
	10 μm	
	1 μm	1/100000 inch
Z	1 mm	1 inch
	10 μm	
	1 μm	1/100000 inch

* X word is expressed in terms of diameter.

Note 1: When programming dimension words, be sure to command numerical data to the selected unit.

Note 2: Unit system may be changed by changing the bit data of optional parameter (bit) No. 3 as follows:

Bit 2	Bit 1	Bit 0	Unit
*	1	0	1 mm
1	0	0	10 μm
0	0	0	1 μm
*	1	1	1 inch
*	0	1	1/100000 inch

* may be either "1" or "0" (usually set at "0").

For details of parameter data setting, refer to the Operation Manual.

Note 3: The 1 mm unit system is set when the control floppy disk is loaded.

(2) Programmable Units for $\emptyset.1 \mu\text{m}$ Specification (optional)

Axis movement commands, i.e., dimension words are expressed in the units specified below:

Axis	Programmable Units	
	Metric System	Inch System (optional)
X	1 mm	1 inch
	$1\emptyset \mu\text{m}$	
	1 μm	$1/1\emptyset\emptyset\emptyset\emptyset$ inch
	$\emptyset.1 \mu\text{m}$	$1/1\emptyset\emptyset\emptyset\emptyset\emptyset$ inch
Z	1 mm	1 inch
	$1\emptyset \mu\text{m}$	
	1 μm	$1/1\emptyset\emptyset\emptyset\emptyset$ inch
	$\emptyset.1 \mu\text{m}$	$1/1\emptyset\emptyset\emptyset\emptyset\emptyset$ inch

Note 1: When programming dimension words, be sure to command numerical data to the selected unit.

Note 2: Unit system may be changed by changing the bit data of optional parameter (bit) No. 3 as follows:

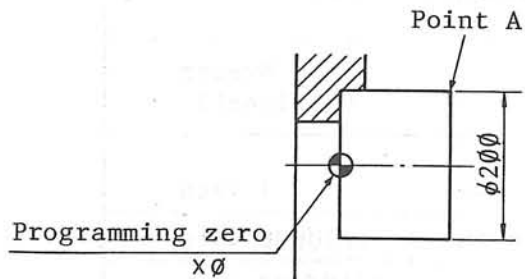
Bit 2	Bit 1	Bit \emptyset	Unit
1	1	\emptyset	$\emptyset.1 \mu\text{m}$
\emptyset	\emptyset	\emptyset	1 μm
1	\emptyset	\emptyset	$1\emptyset \mu\text{m}$
\emptyset	1	\emptyset	1 mm
1	\emptyset	1	$1/1\emptyset\emptyset\emptyset\emptyset\emptyset$ inch
\emptyset	\emptyset	1	$1/1\emptyset\emptyset\emptyset\emptyset$ inch
\emptyset	1	1	1 inch

For details of parameter data setting, refer to the Operation Manual.

Note 3: The 1 mm unit system is set when the control floppy disk is loaded.

(3) X-Axis Command (1 mm unit command)

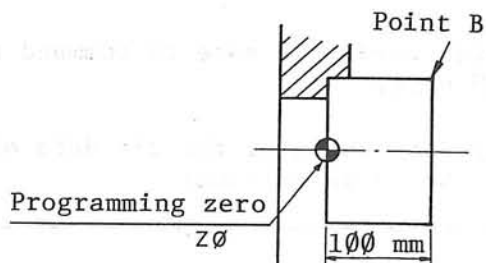
X-axis command is given in terms of diameter as indicated on a part drawing.

Example:

When "X20φ" command is specified, the turret moves along X-axis to position the cutting tool to Point A.

(4) Z-Axis Command (1 mm unit command)

Z-axis command is given in terms of longitudinal dimensions referenced to the programming zero.

Example:

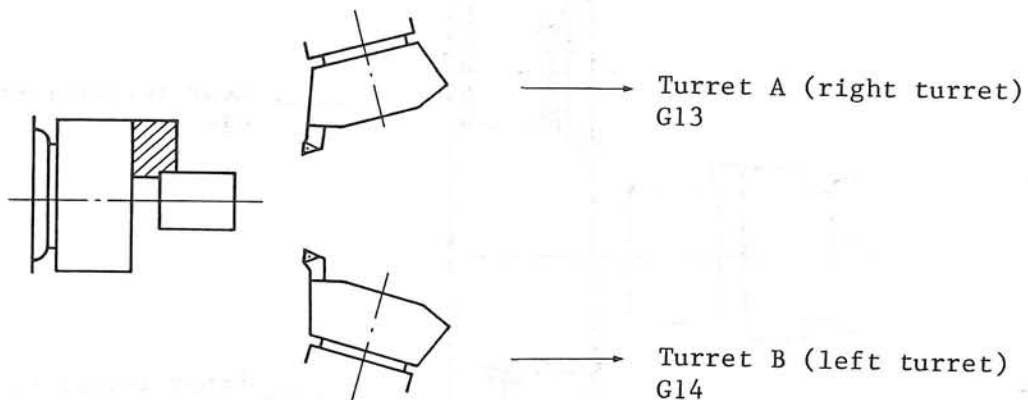
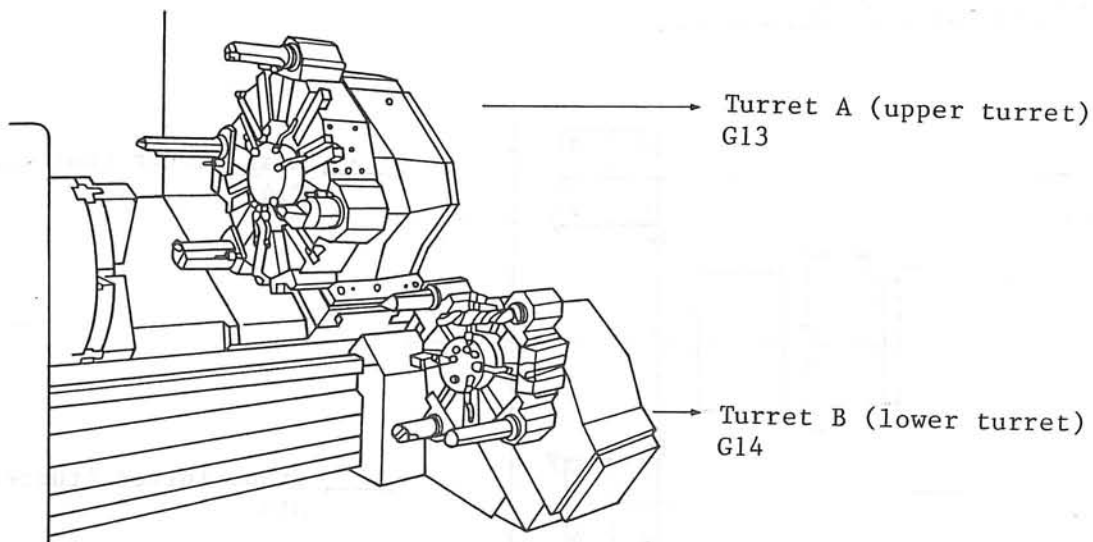
When "Z10φ" command is specified, the turret moves along Z-axis to position the cutting tool to Point B.

1-5. TURRET SELECTION FOR 2-SADDLE/2-TURRET MODELS



1-5-1. LC Series

For two-saddle models, Model 2S, selection of the turret is made by specifying a G code as indicated below:

G13 Turret A
G14 Turret B

(1) LC1 \emptyset (2) LC2 \emptyset , 3 \emptyset , 4 \emptyset and 5 \emptyset 

For the machine equipped with only one saddle, designation of G codes used for turret selection is unnecessary.

In the MDI mode operations, G13 and G14 codes are not used for turret selection but the  and  keys on the operation panel.

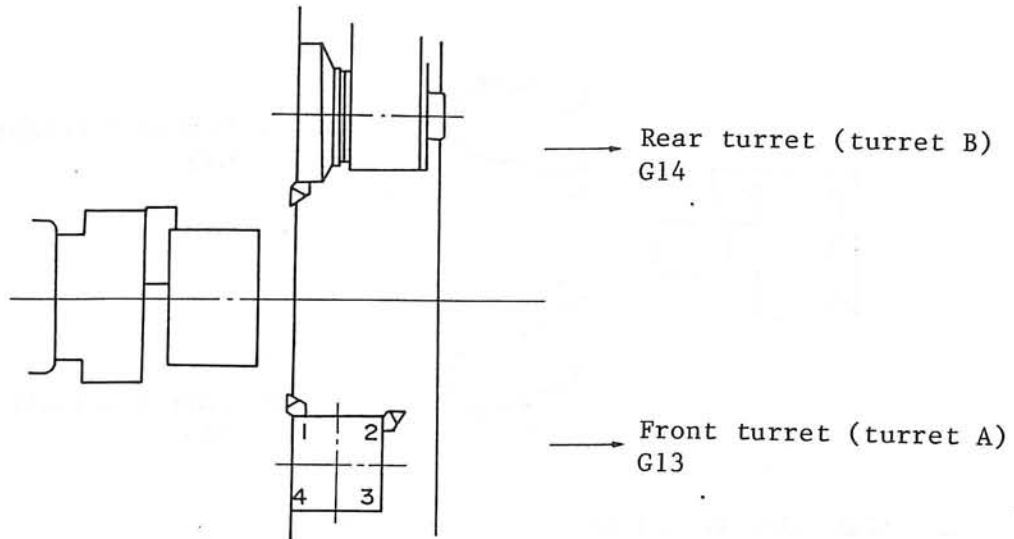
1-5-2. LS30-N, LH35-N and LH55-N

For two-turret models, selection of the turret is made by specifying a G code as indicated below:

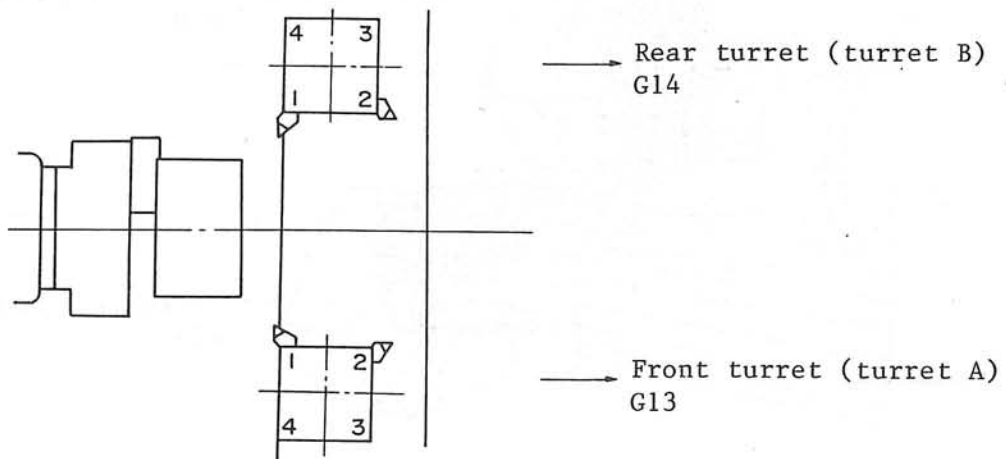
G13 Front turret (turret A)

G14 Rear turret (turret B)

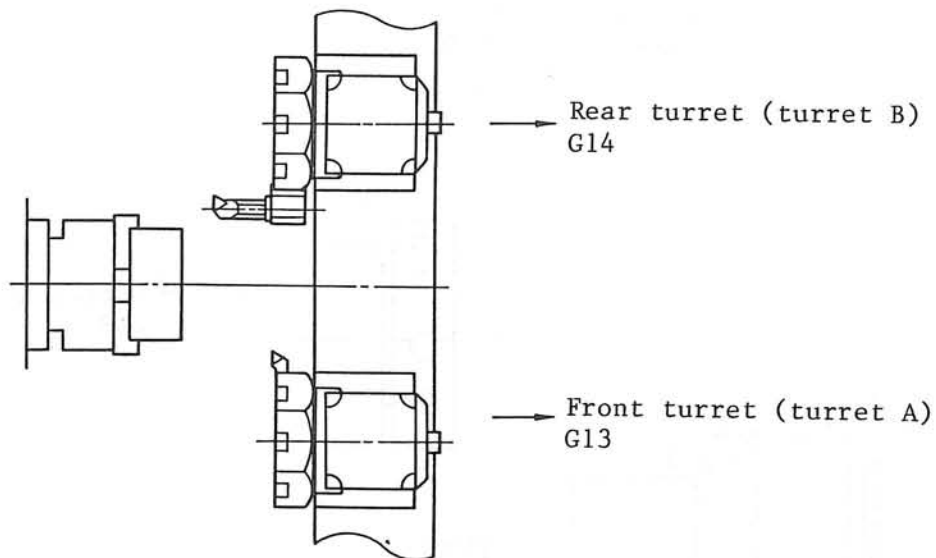
(1) H4 + V6 (LS30-N, LH35-N)



(2) H4 + H4 (LS30-N only)



(3) V8 + V8 (LH55-N only)



For the machine equipped with only one saddle, designation of G codes used for turret selection is unnecessary.

In this Manual, the front turret is called "turret A" and the rear turret "turret B".

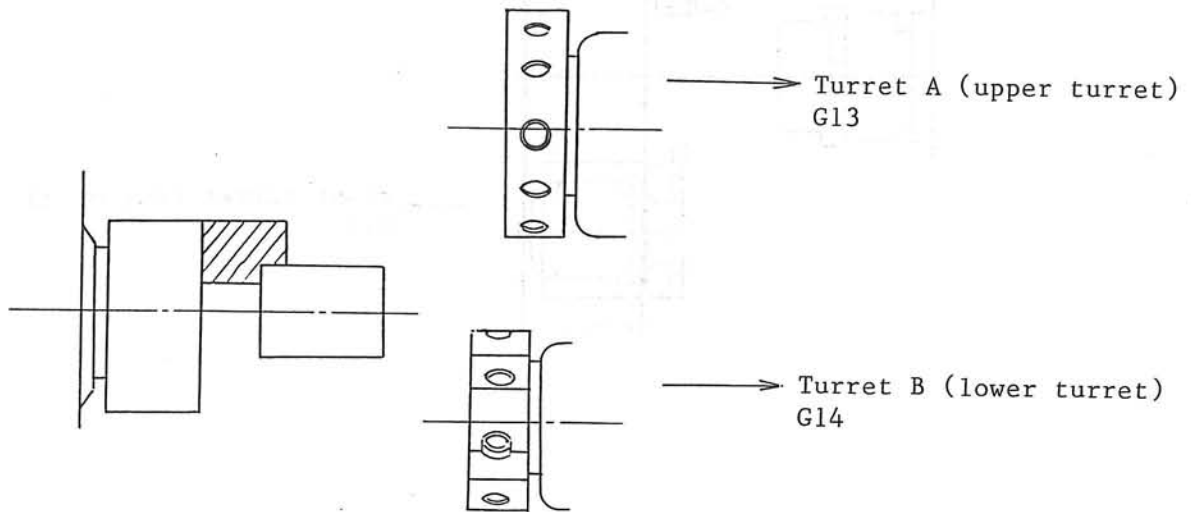
1-5-3. LR Series



For two-saddle models, Model 2S, selection of the turret is made by specifying a G code as indicated below:

G13 Turret A

G14 Turret B

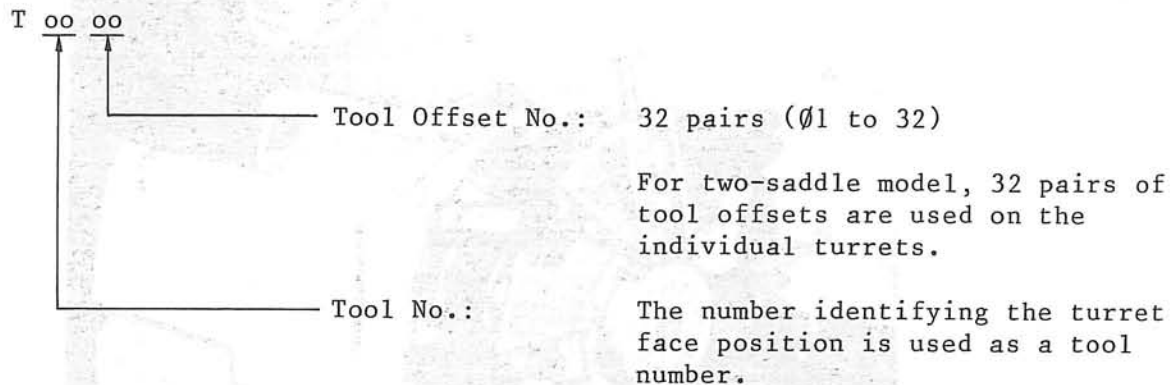
(1) LR15



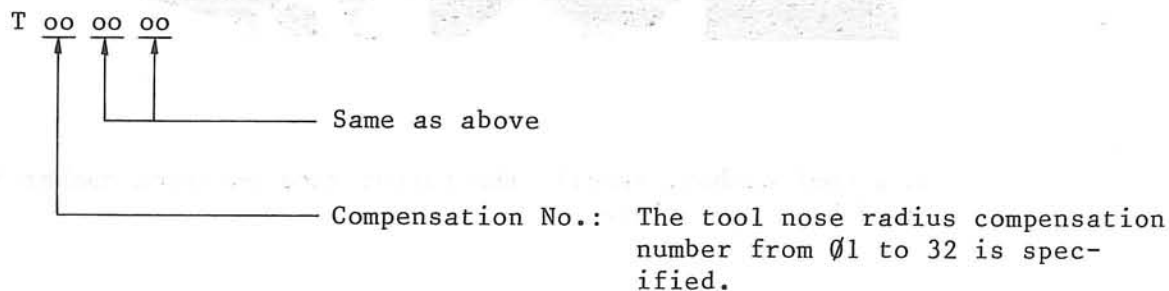
In the MDI mode operations, G13 and G14 codes are not used for turret selection but the  and  keys on the operation panel.

1-6. TOOL SELECTION

Selection of a cutting tool is made by four digit figures following address character T.



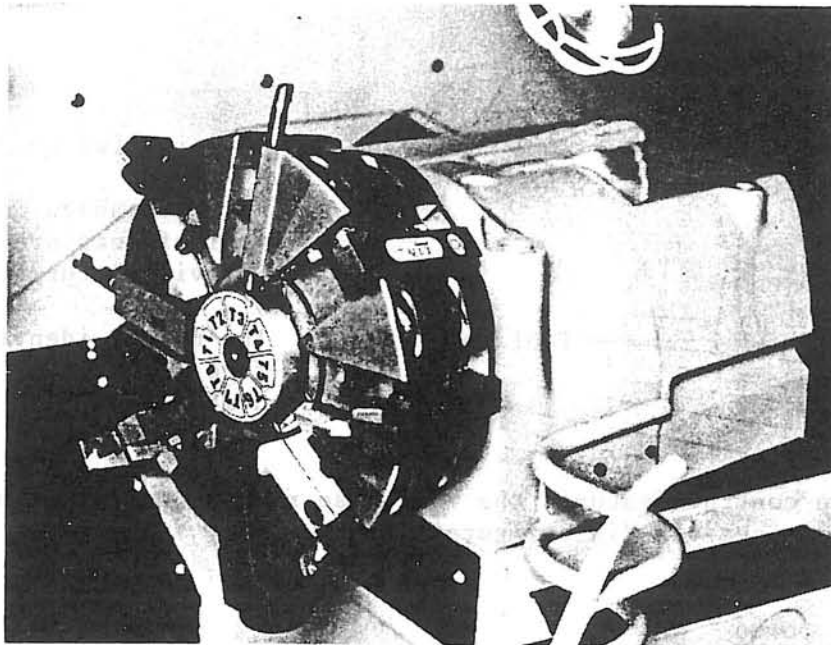
When the control features the tool nose radius compensation function, a T word comprises six digit figures.



For two-saddle (turret) models, tool nose radius compensation numbers 1 through 32 can be used on the individual turrets.

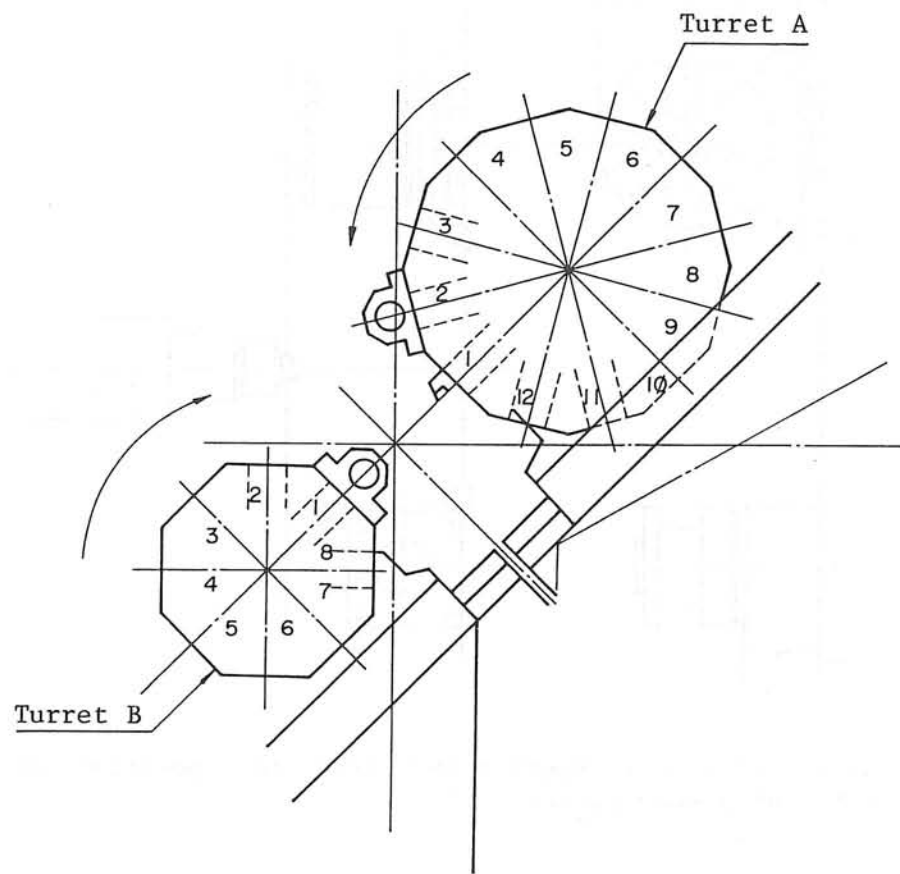
(1) Tool Number

a) LC1Ø



As a tool number, specify the turret face position number indicated by a name plate.

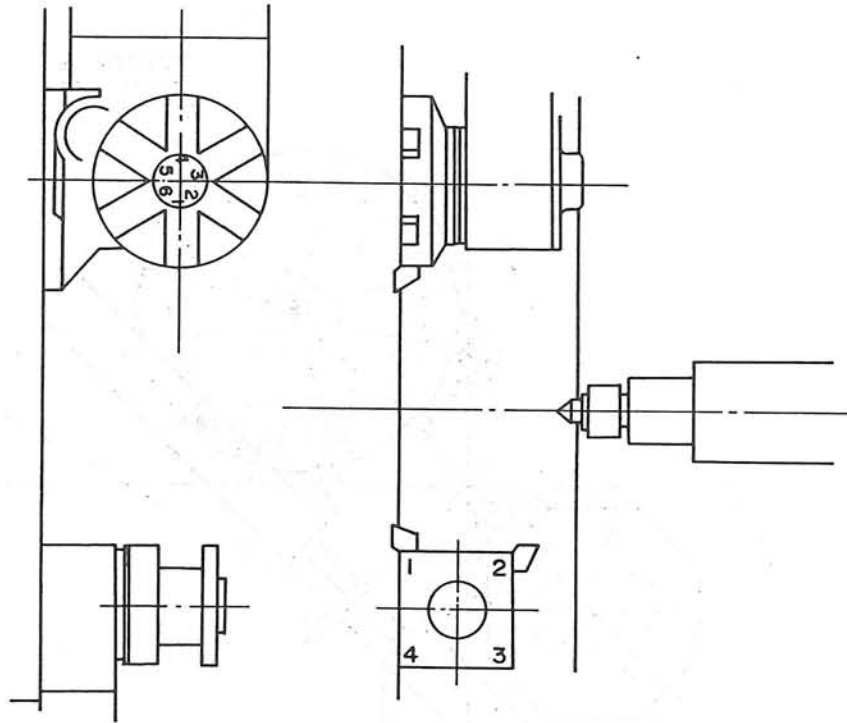
b) LC2Ø/LC3Ø/LC4Ø/LC5Ø



Note 1: The illustration above shows the turrets of Model 2S.
Only turret A is available for Model 1S.

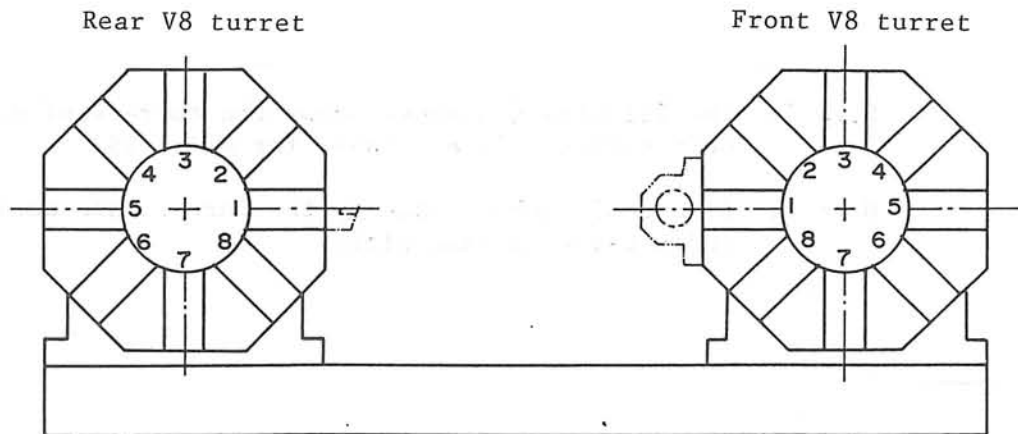
Note 2: As a tool number, specify the turret face position number
indicated by a name plate.

c) LS30-N and LH35-N

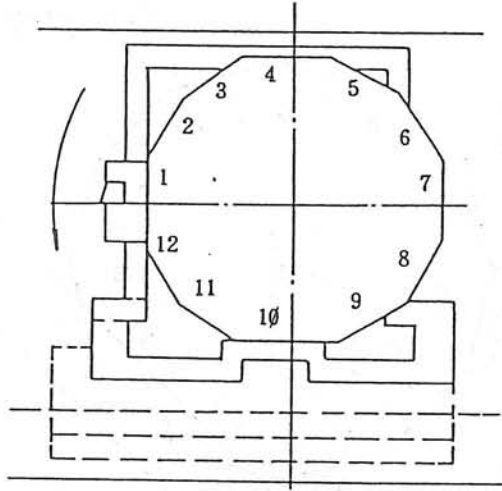


As a tool number, specify the turret face position number indicated by a name plate.

d) LH55-N

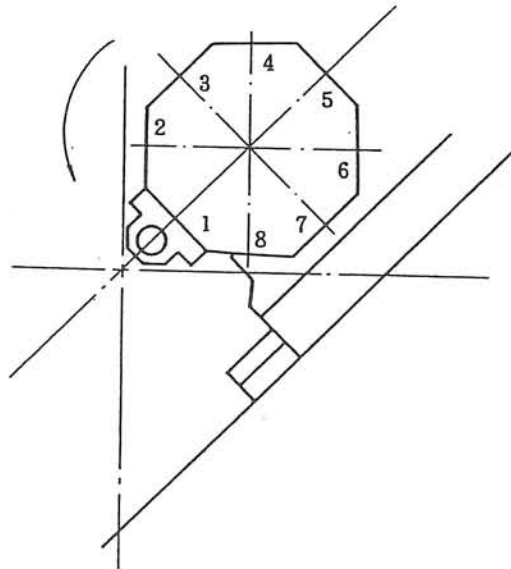


As a tool number, specify the turret face position number indicated by a name plate.

e) LB1 \emptyset 

As a tool number, specify the turret face position number indicated by a name plate.

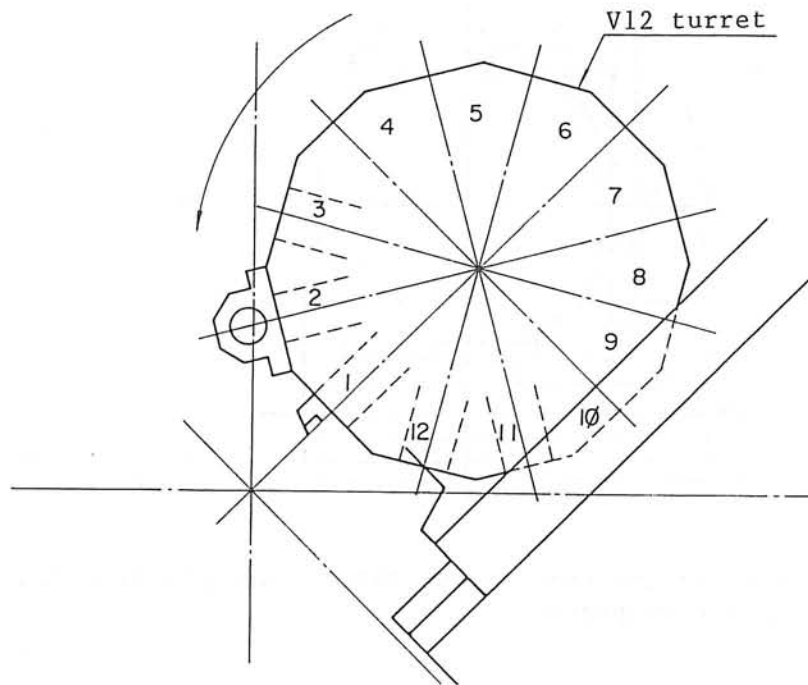
f) LB12



Note 1: As a tool number, specify the turret face position number indicated by a name plate.

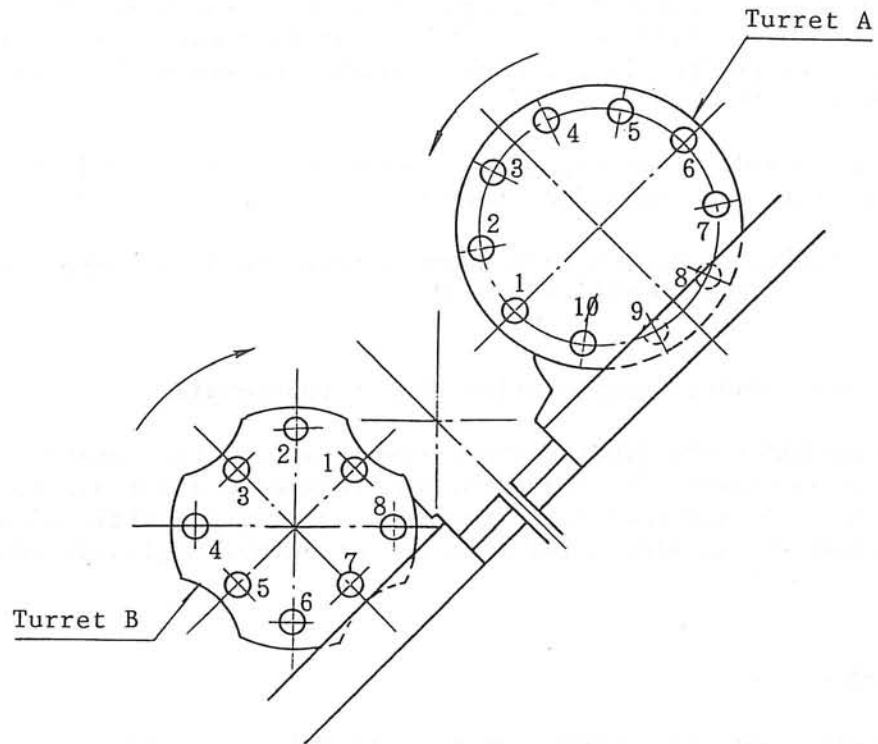
Note 2: Optionally, dodecagonal turret is available.

g) LB15



As a tool number, specify the turret face position number indicated by a name plate.

h) LR15



Note 1: As a tool number, specify the turret face position number indicated by a name plate.

Note 2: Tool numbers on turret A are 1 - 10 and those on turret B are 1 - 8.

(2) Tool Offset Number

The control has a capability to compensate for the tool dimensions between those assumed in programming and those actually set. This function is called the tool offset function and it is called out by two-digit figures in a T code command. Programmable tool offset number is from "1" to "32".

For two-saddle (turret) models, tool offset numbers 1 through 32 can be used on the individual turrets.

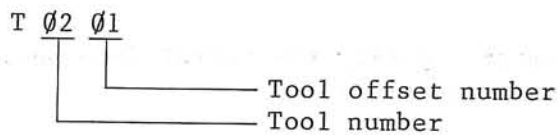
For details of tool offset number, refer to 3-5, "Supplement" or 3-3-2-3 of the Operation Manual.

(3) Tool Nose Radius Compensation Number (optional)

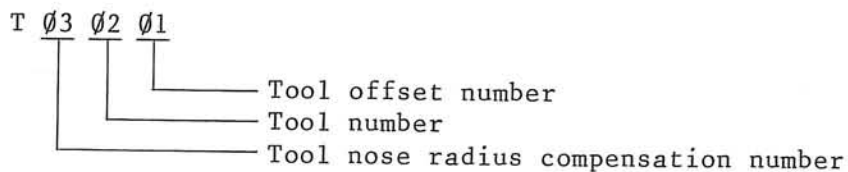
The OSP500L-G/OSP5000L-G CNC system features the function to automatically compensate for the error in finished contour due to tool nose radius. To activate this function, specify two digit figures, 1 through 32, as addressing tool offset number following address character T.

(4) Example

When the tool nose radius compensation function is not used:



When the tool nose radius compensation function is used:



In usual programming, tool offset number and tool nose radius compensation number are taken as the same number as the tool number.

T0101
T020202

- (5) Turret Indexing Direction
 (standard for LC2 \emptyset /LC3 \emptyset /LC4 \emptyset /LC5 \emptyset /LB1 \emptyset /LB12/LB15/LR15, and optional for LH55-N)

M Code	Description	Remarks
M86	The turret rotates in the direction in which the tool number shifts in the descending order as 4 \rightarrow 3 \rightarrow 2 \rightarrow 1. (Reverse)	
M87	The turret rotates in the direction in which the tool number shifts in the ascending order as 1 \rightarrow 2 \rightarrow 3 \rightarrow 4. (Forward)	The control is in the M87 mode when the control software tape is loaded.

For the models featuring bi-directional turret indexing function, the rotation direction of the turret can be specified by M86 and M87.

In addition, the control has the capability to automatically determine the turret rotation direction by setting a flag at a proper parameter. If this automatic bi-directional turret rotation function is set active, codes M86 and M87 are ignored even if they are programmed.

- (6) Turret Shortest Path Rotation Control
 (standard for LC2 \emptyset /LC3 \emptyset /LC4 \emptyset /LC5 \emptyset /LB1 \emptyset /LB12/LB15/LR15, and optional for LH55-N)

In the automatic or MDI mode operation, it is possible to select the turret rotation direction to index the commanded turret position via the shortest path from the current turret position independently of the turret rotation command (M86, M87) designated. For this function, set the parameters indicated below:

- a) Availability of automatic turret shortest path indexing control

Parameter (bit) No. 3, bit 7

1: Available
 \emptyset : Not available

(initial setting: \emptyset)

b) Number of turret faces

Parameter (word) No. 47:

Number of turret faces on turret A

Initial setting: 0

Setting range : 0 - 16

Parameter (word) No. 48:

Number of turret faces on turret B

Initial setting: 0

Setting range : 0 - 16

- Note 1: If parameter (word) No. 47 and No. 48 are set at "0", the shortest path rotation control is not conducted even when bit 7 of parameter (bit) No. 3 is set at "1" and turret rotation direction is determined by the programmed M86 and M87.
- Note 2: While the turret shortest path rotation control is effective, turret rotation by manual operation (manual operation mode or operation during manual interruption mode) is automatically controlled in the forward direction unconditionally.
- Note 3: After the reset of turret shortest path control function, turret rotation direction is controlled by M86 or M87 designated before the shortest path turret rotation control is set active.
- Note 4: If the distance required to reach the commanded turret face is equal when the turret is rotated forward and reverse, then the turret is rotated in the forward direction.

1-7. SPINDLE SPEED SELECTION

The selection of spindle speed range is made by a two-digit M code. The desired spindle speed within the selected spindle speed range is selected by a four-digit S code, expressed directly by the desired spindle speed.

1-7-1. LC10

On Model LC10, there are three different spindle speed ranges as indicated below depending on the selected specifications:

- a) 200 to 4500 rpm
- b) 133 to 3000 rpm
- c) 98 to 2200 rpm (optional specifications)

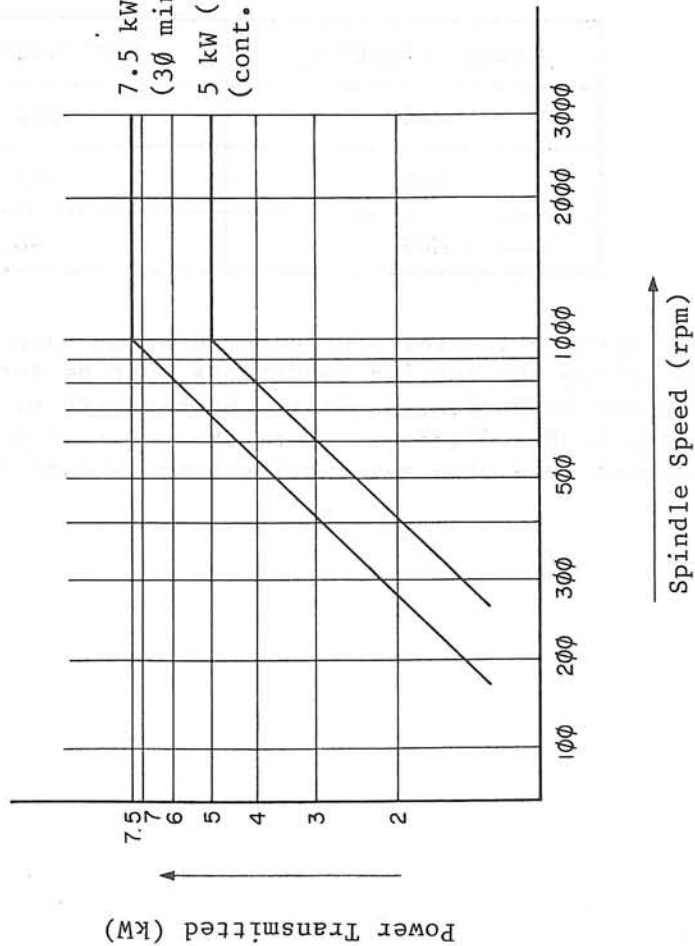
As an optional specification, pulley changeable specification is available, which allows the selection of three different spindle speed ranges as 200 to 4500 rpm, 133 to 3000 rpm and 98 to 2200 rpm by changing the spindle pulley.

In this specification, the following M codes are used to specify the pulleys to be employed. Since these M codes are not cleared when power supply is turned off or the control is reset, program such M code only when pulleys are changed and it is unnecessary to provide the one on individual part program tapes.

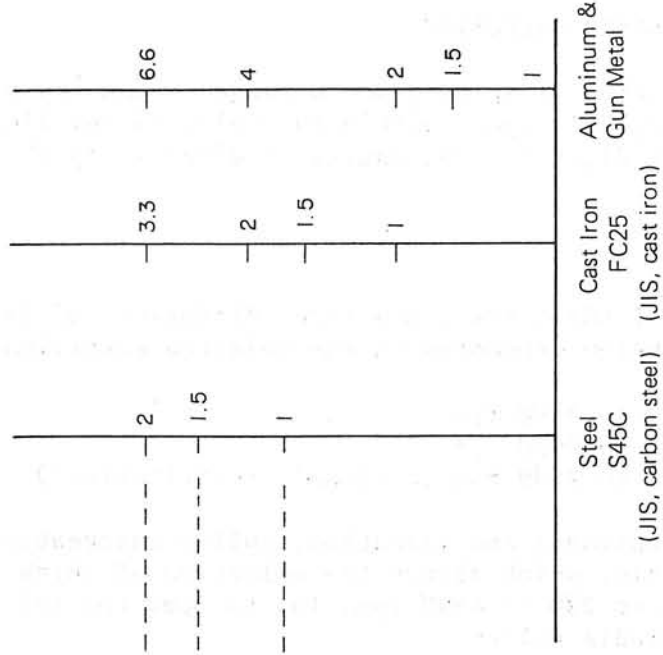
Range Selection	Available Spindle Speeds
M44	200 to 4500 rpm
M43	133 to 3000 rpm
M42	98 to 2200 rpm

Spindle speed - transmission power diagrams are provided in the following pages. Since the cutting conditions must be determined so that intended cutting can be performed within the capacity of LC10, determine them referring to the diagrams. Note the required power will vary from material to material even when the cutting area is identical.

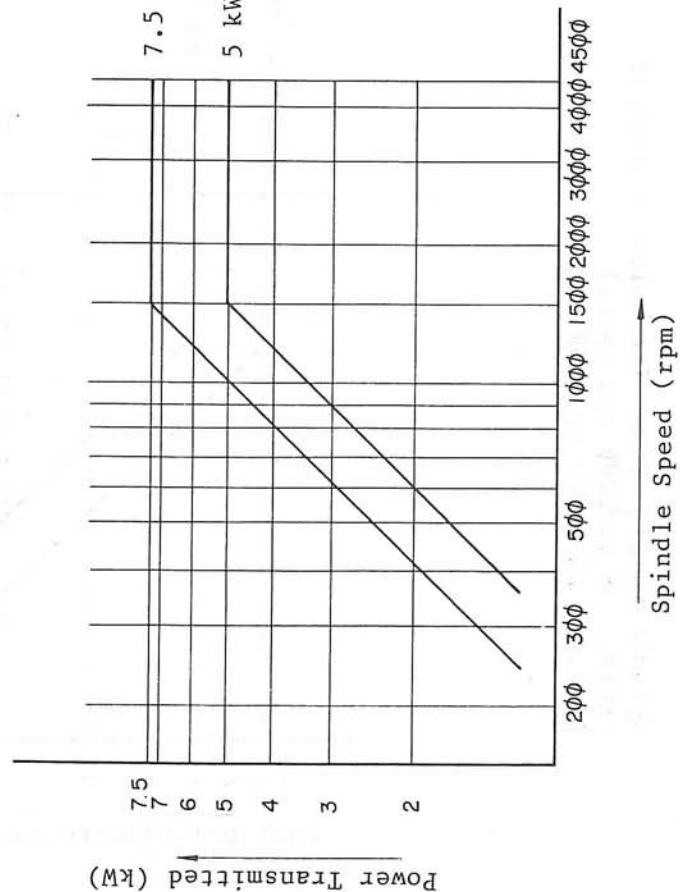
Spindle Speed - Power Transmission Diagram
 (LC10: 130 to 3000 rpm spec.)
 DC 7.5/5 kW (10/7 hp) (30 min./cont. rating)



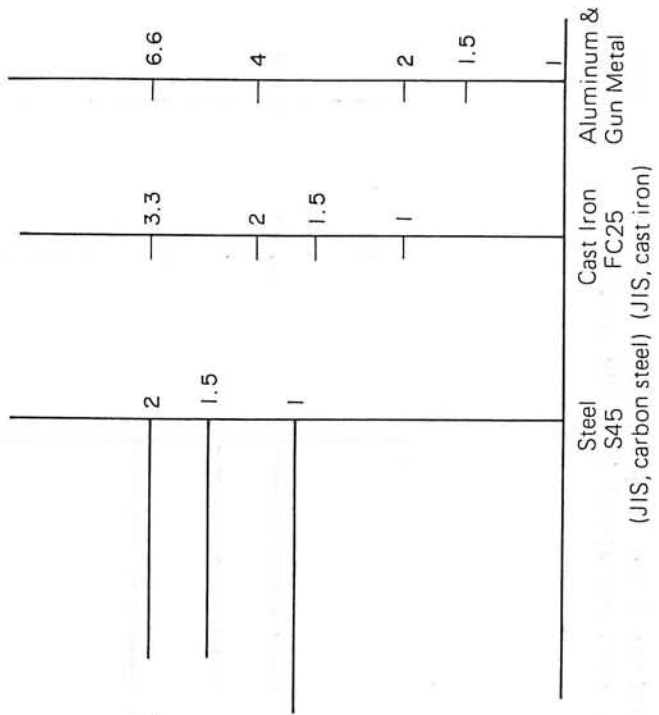
Cutting Area (Depth of cut x Feedrate) (mm²)
 at cutting speed V = 100 m/min.



Spindle Speed - Power Transmission Diagram
 (LC10: 200 to 4500 rpm spec.)
 DC 7.5/5 kW (10/7 hp) (30 min./cont. rating)

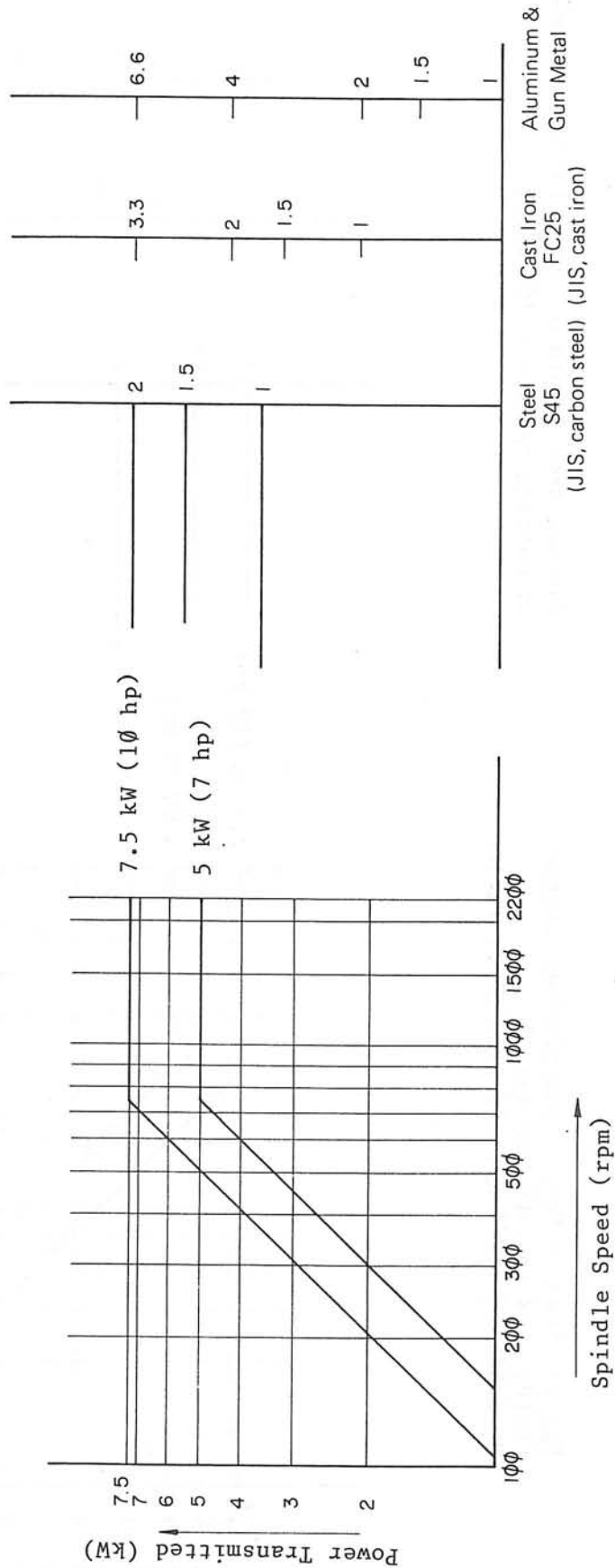


Cutting Area (Depth of cut x Feedrate) (mm²)
 at cutting speed V = 100 m/min.



Spindle Speed - Power Transmission Diagram
 (LC10: 98 to 2200 rpm spec.)
 DC 7.5/5 kW (10/7 hp) (30 min./cont. rating)

Cutting Area (Depth of cut x Feedrate) (mm²)
 at cutting speed V = 100 m/min.



1-7-2. LC2Ø

Range Selection	Available Spindle Speeds
M41	75 to 1Ø82 rpm
M42	3Ø3 to 42ØØ rpm

Example:

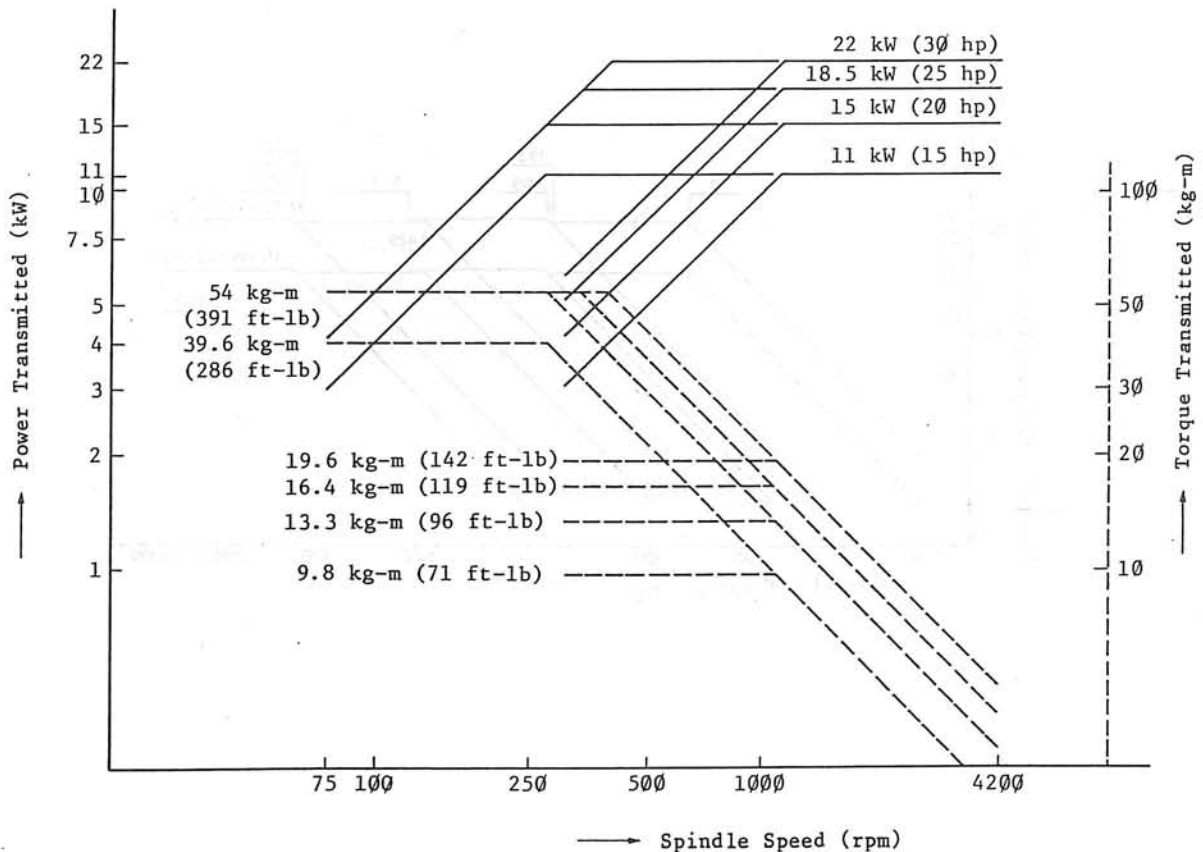
To rotate the spindle at 12Ø rpm, program as:

M41 S12Ø

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagram provided below. And program should be made so that cutting is performed within a constant power range where practicable.

LC2Ø 1S: VAC 15/11 kW (2Ø/15 hp) (3Ø min./cont. rating)

LC2Ø 2S: VAC 22/18.5 kW (3Ø/25 hp) (3Ø min./cont. rating)



1-7-3. LC30

Range Selection	Available Spindle Speeds
M41	15 to 193 rpm
M42	40 to 519 rpm
M43	86 to 1118 rpm
M44	230 to 3000 rpm

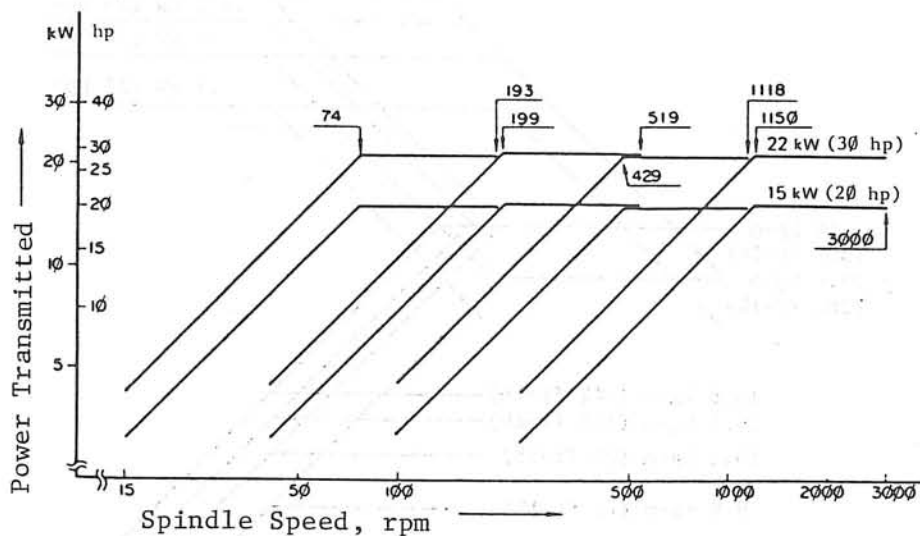
Example:

To rotate the spindle at 120 rpm, program as:

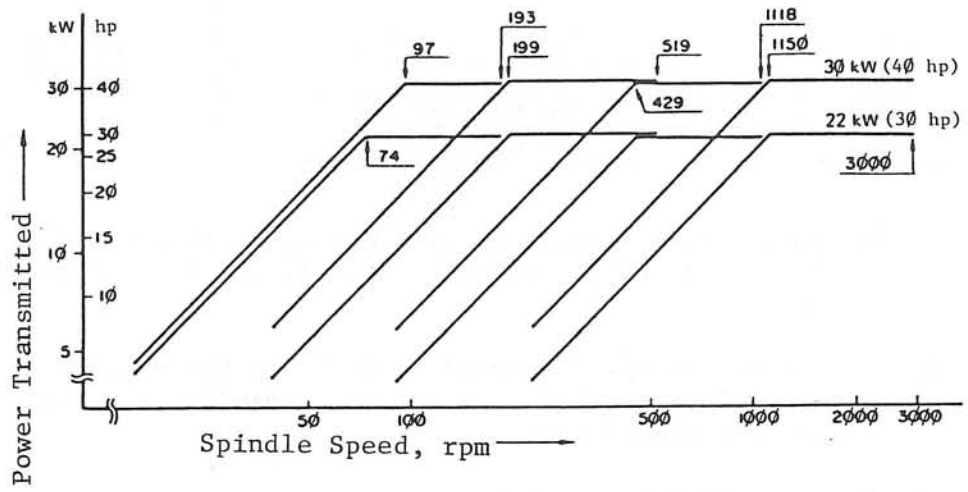
M41 S120

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagrams provided below. And program should be made so that cutting is performed within a constant power range where practicable.

LC30 1S
DC 22/15 kW (30/20 hp) (30 min./cont. rating)



LC30 2S
 DC 30/22 kW (40/30 hp) (30 min./cont. rating)



1-7-4. LC4 \emptyset

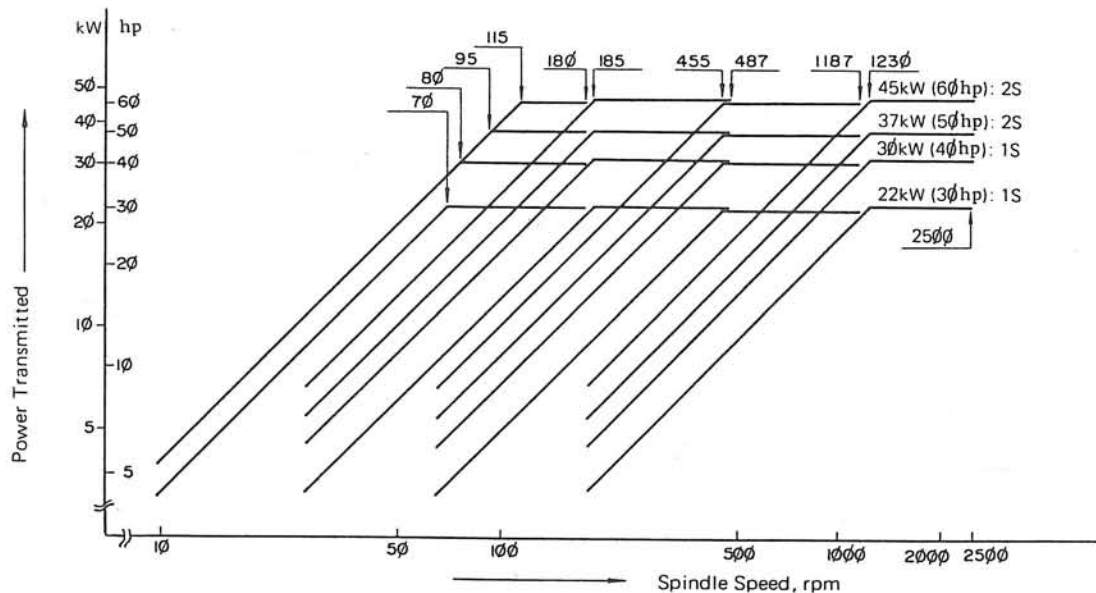
Range Selection	Available Spindle Speeds
M41	1 \emptyset to 18 \emptyset rpm
M42	27 to 487 rpm
M43	66 to 1187 rpm
M44	177 to 25 $\emptyset\emptyset$ rpm

Example:

To rotate the spindle at 12 \emptyset rpm, program as:

M41 S12 \emptyset

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagram provided below. And program should be made so that cutting is performed within a constant power range where practicable.

LC4 \emptyset 1S: DC 3 \emptyset /22 kW (4 \emptyset /3 \emptyset hp) (3 \emptyset min./cont. rating)LC4 \emptyset 2S: DC 45/37 kW (6 \emptyset /5 \emptyset hp) (3 \emptyset min./cont. rating)

1-7-5. LC50

Range Selection	Available Spindle Speeds
M41	5 to 135 rpm
M42	11 to 283 rpm
M43	18 to 487 rpm
M44	38 to 1000 rpm

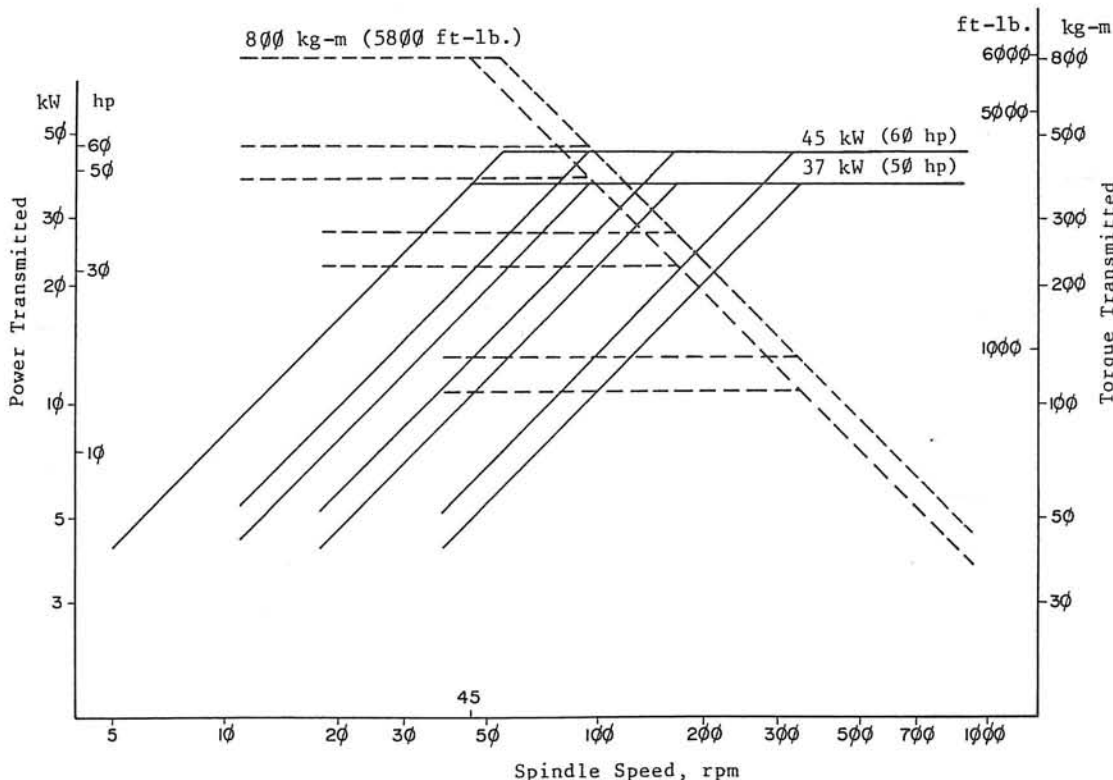
Example:

To rotate the spindle at 120 rpm, program as:

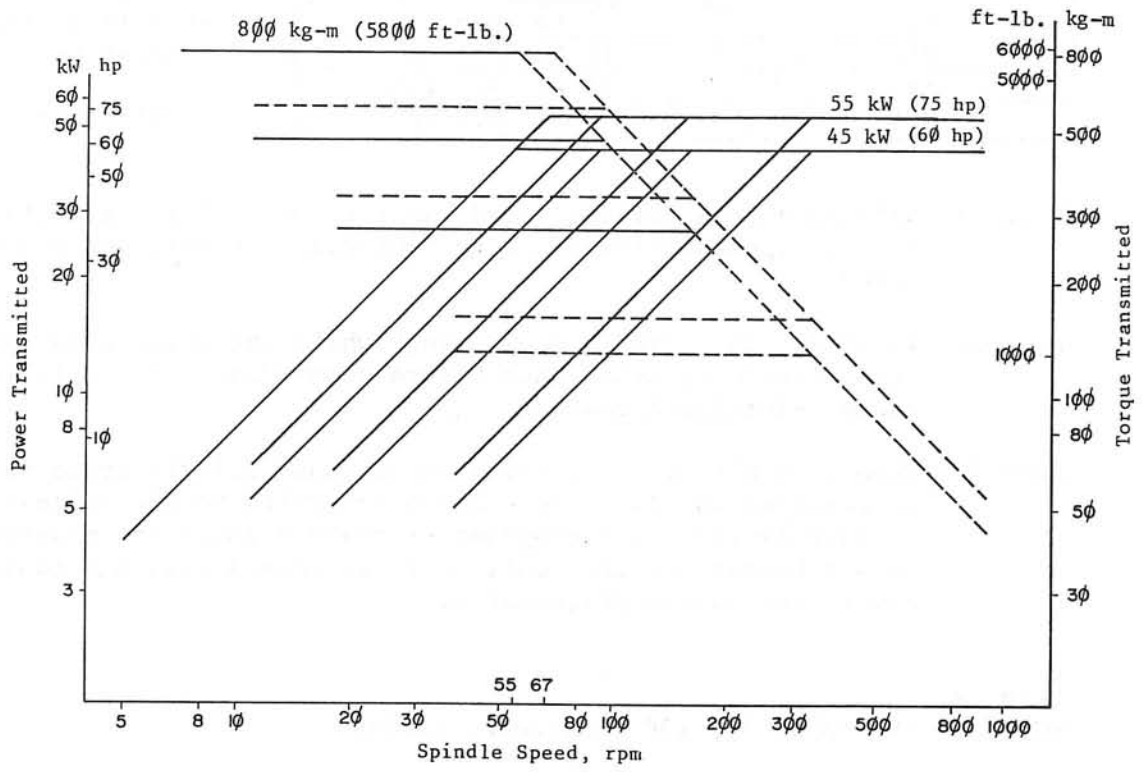
M41 S120

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagrams provided below. And program should be made so that cutting is performed within a constant power range where practicable.

LC50 1S
 DC 45/37 kW (60/50 hp) (30 min./cont. rating)



LC50 2S
 DC 55/45 kW (76/60 hp) (30 min./cont. rating)



1-7-6. LS30-N

Range Selection	Available Spindle Speeds
M41	25 to 165 rpm
M42	69 to 452 rpm
M43	138 to 915 rpm
M44	377 to 2500 rpm

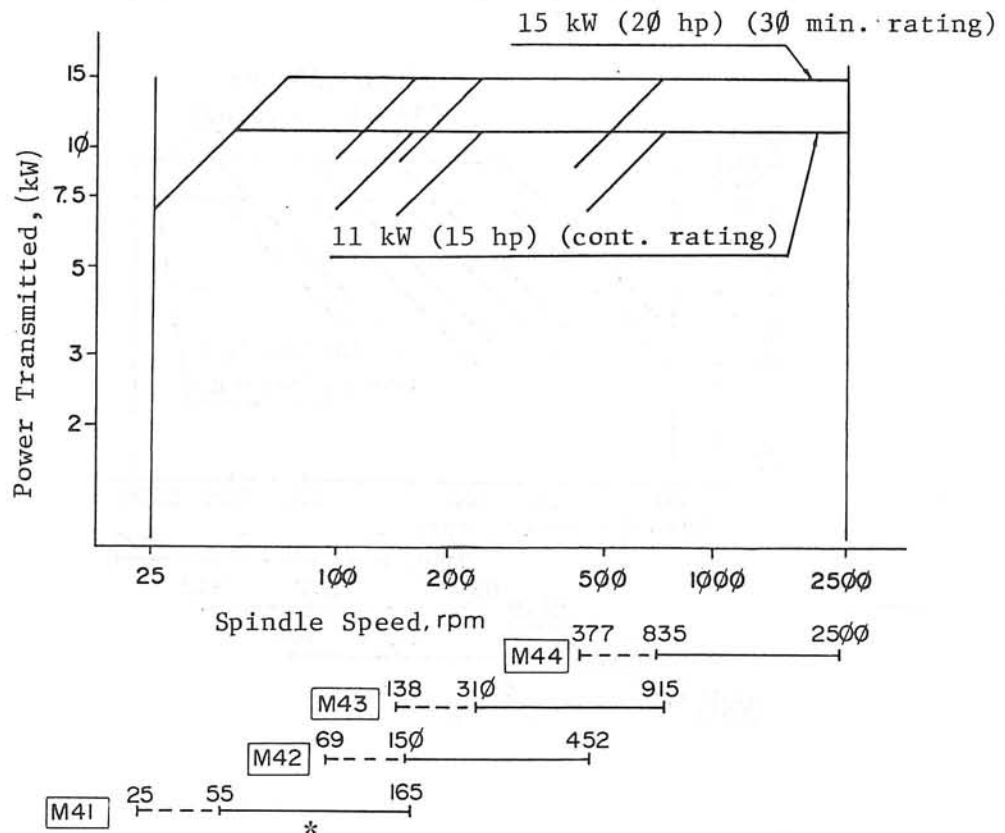
Example:

To rotate the spindle at 120 rpm, program as:

M41 S120

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagram provided below. And program should be made so that cutting is performed within a constant power range where practicable.

DC 15/11 kW (20/15 hp) (30 min./cont. rating)



* Solid lines indicate the spindle speeds range in which constant full power [11 kW (15 hp) or 15 kW (20 hp)] can be obtained. In the area indicated by dashed lines, transmission power varies as the spindle speed changes.

1-7-7. LH35-N

Range Selection	Available Spindle Speeds
M41	20 to 197 rpm
M42	52 to 507 rpm
M43	86 to 840 rpm
M44	224 to 2200 rpm

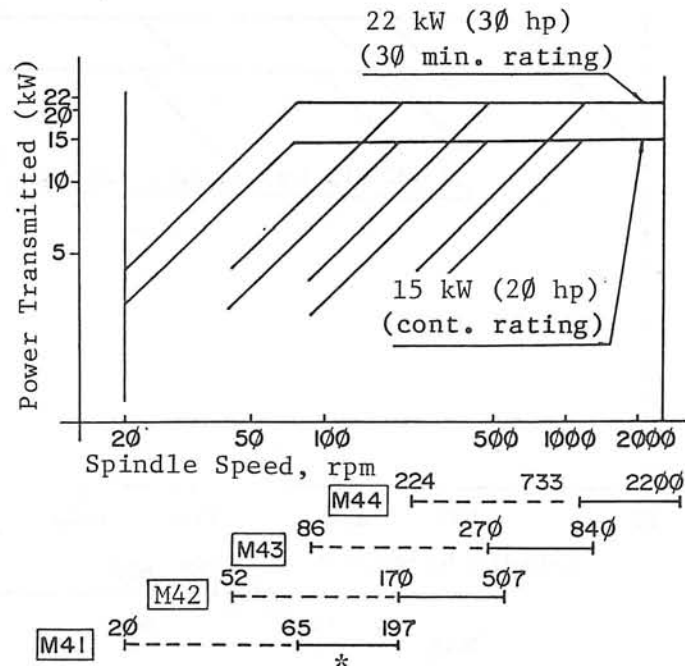
Example:

To rotate the spindle at 120 rpm, program as:

M41 S120

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagram provided below. And program should be made so that cutting is performed within a constant power range where practicable.

DC 22/15 kW (30/20 hp) (30 min./cont. rating)



* Solid lines indicate the spindle speeds range in which constant full power [15 kW (20 hp) or 22 kW (30 hp)] can be obtained. In the area indicated by dashed lines, transmission power varies as the spindle speed changes.

1-7-8. LH55-N

Range Selection	Available Spindle Speeds
M41	10 to 135 rpm (10 to 160 rpm)
M42	21 to 280 rpm (21 to 330 rpm)
M43	36 to 480 rpm (36 to 575 rpm)
M44	75 to 1000 rpm (75 to 1200 rpm)

Example:

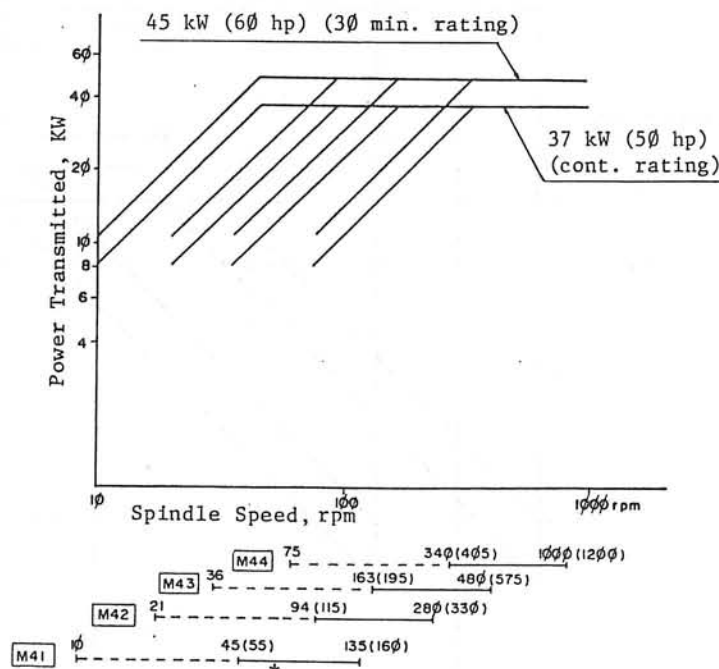
To rotate the spindle at 120 rpm, program as:

M41 S120

1000 rpm spec.
() ... 1200 rpm spec.

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagram provided below. And program should be made so that cutting is performed within a constant power range where practicable.

DC 45/37 kW (60/50 hp) (30 min./cont. rating)



* Solid lines indicate the spindle speeds range in which constant full power [37 kW (50 hp)] can be obtained. In the area indicated by dashed lines, transmission power varies as the spindle speed changes.

1-7-9. LB10

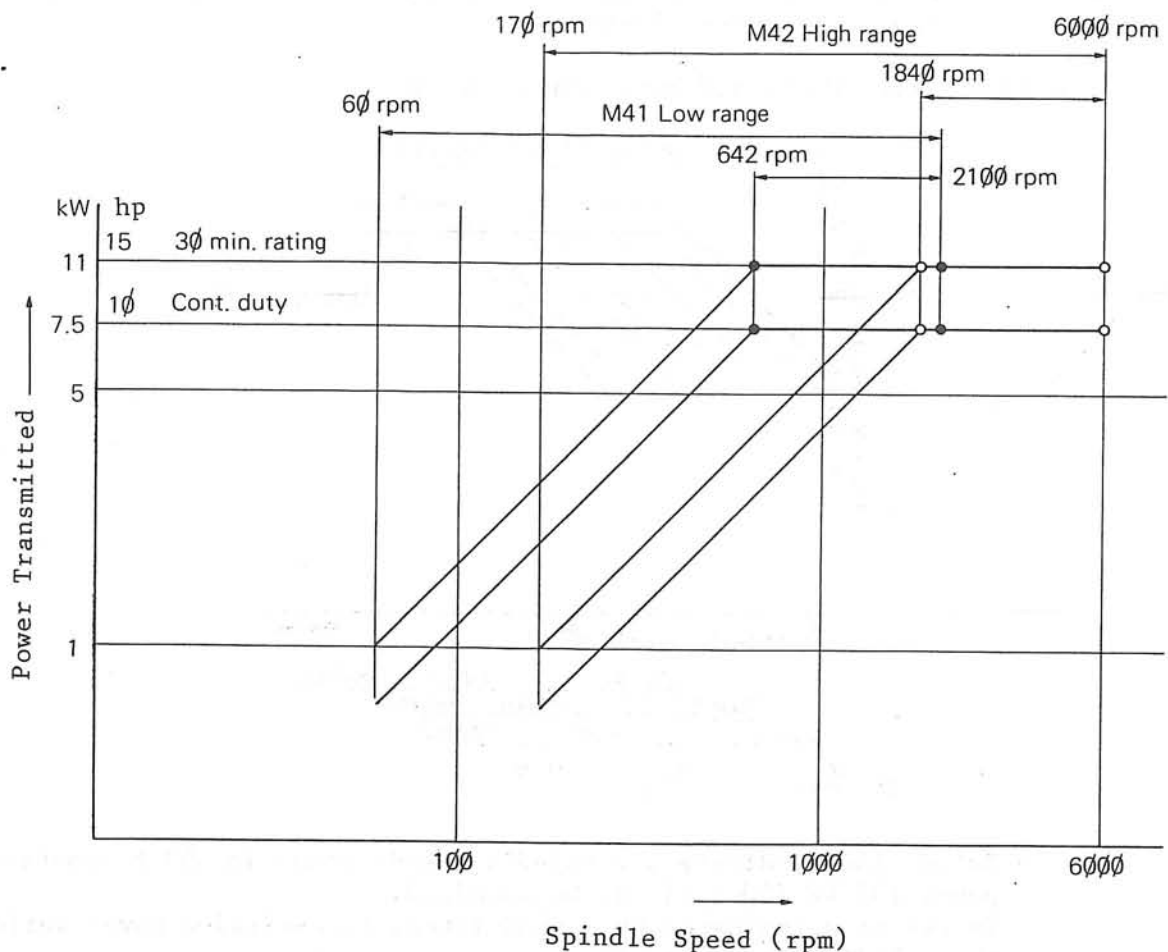
Range Selection	Available Spindle Speeds
M41	60 to 2100 rpm
M42	170 to 6000 rpm

Example:

To rotate the spindle at 120 rpm, program as:

M41 S120

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagram provided below. And program should be made so that cutting is performed within a constant power range where practicable.



1-7-10. LB12

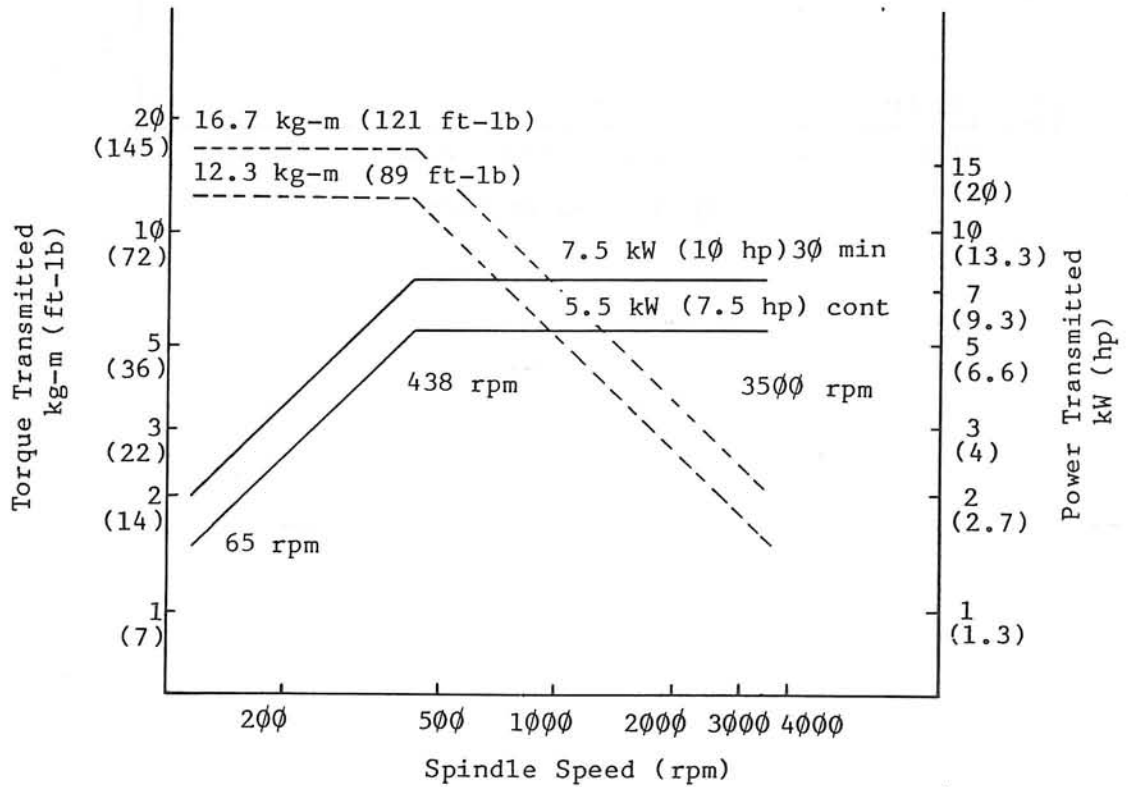
Selection of spindle is possible by directly designating a required speed following address character S. The LB12 does not have the transmission to change the spindle speed range and thus specification of M codes used for selecting the spindle speed range is not required.

Programmable spindle speed range: 65 - 3500 rpm
 75 - 4200 rpm (optional)

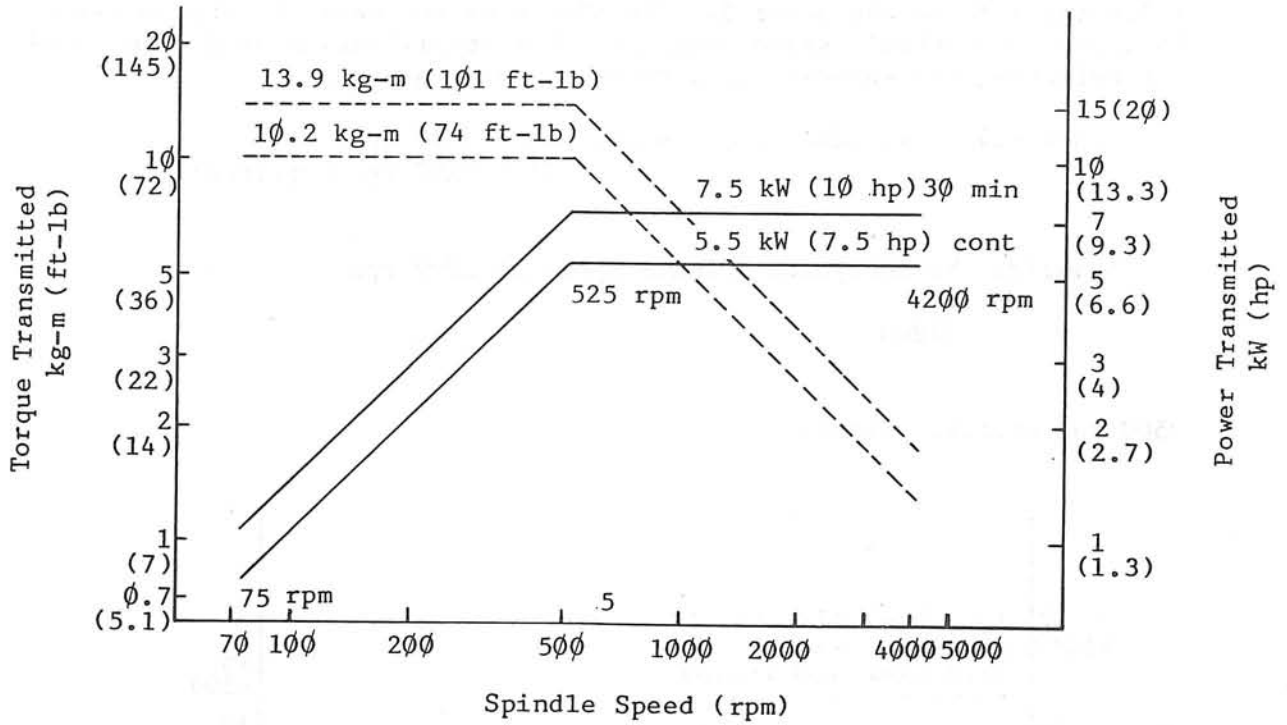
Example: To designate spindle speed of 2000 rpm

S2000

3500 rpm specifications:



4200 rpm specifications:



1-7-11. LB15

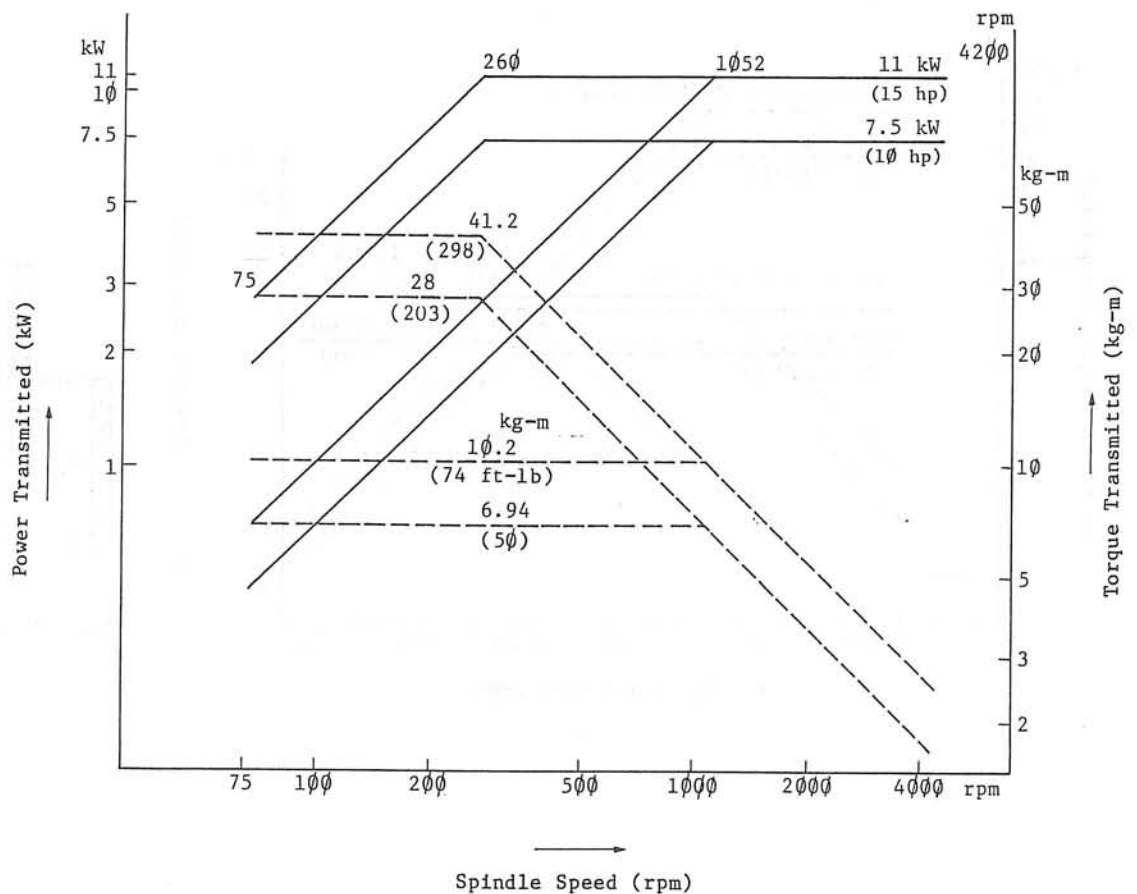
Range Selection	Available Spindle Speeds
M41	75 to 1040 rpm
M42	75 to 4200 rpm

Example:

To rotate the spindle at 120 rpm, program as:

M41 S120

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagram provided below. And program should be made so that cutting is performed within a constant power range where practicable.

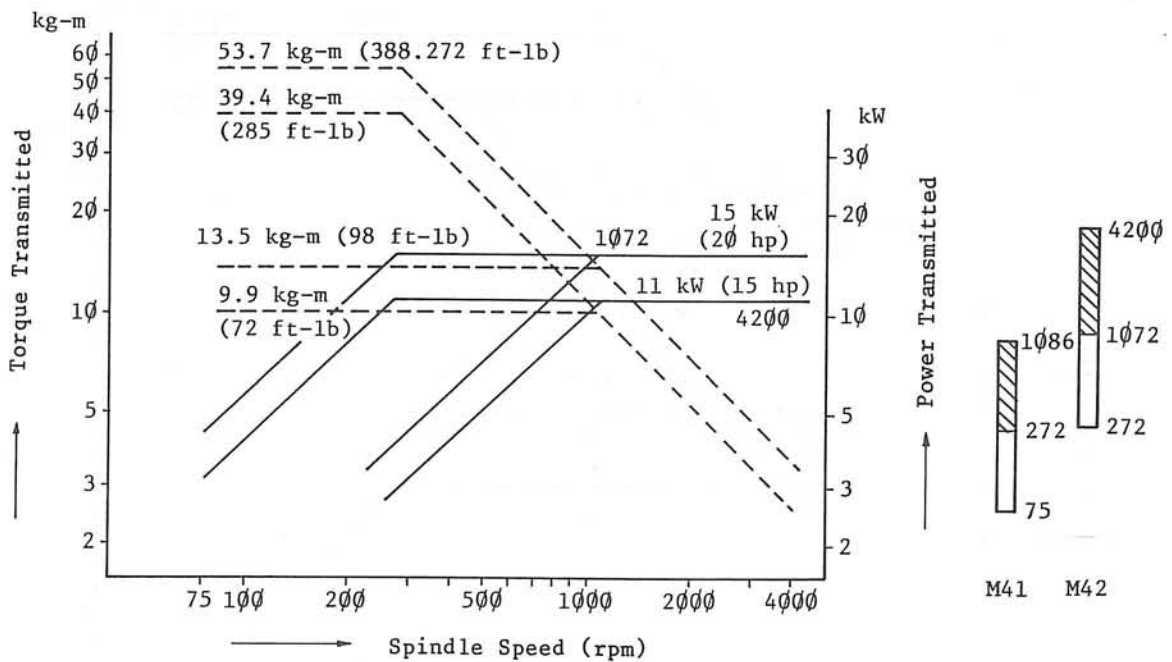


1-7-12. LR15

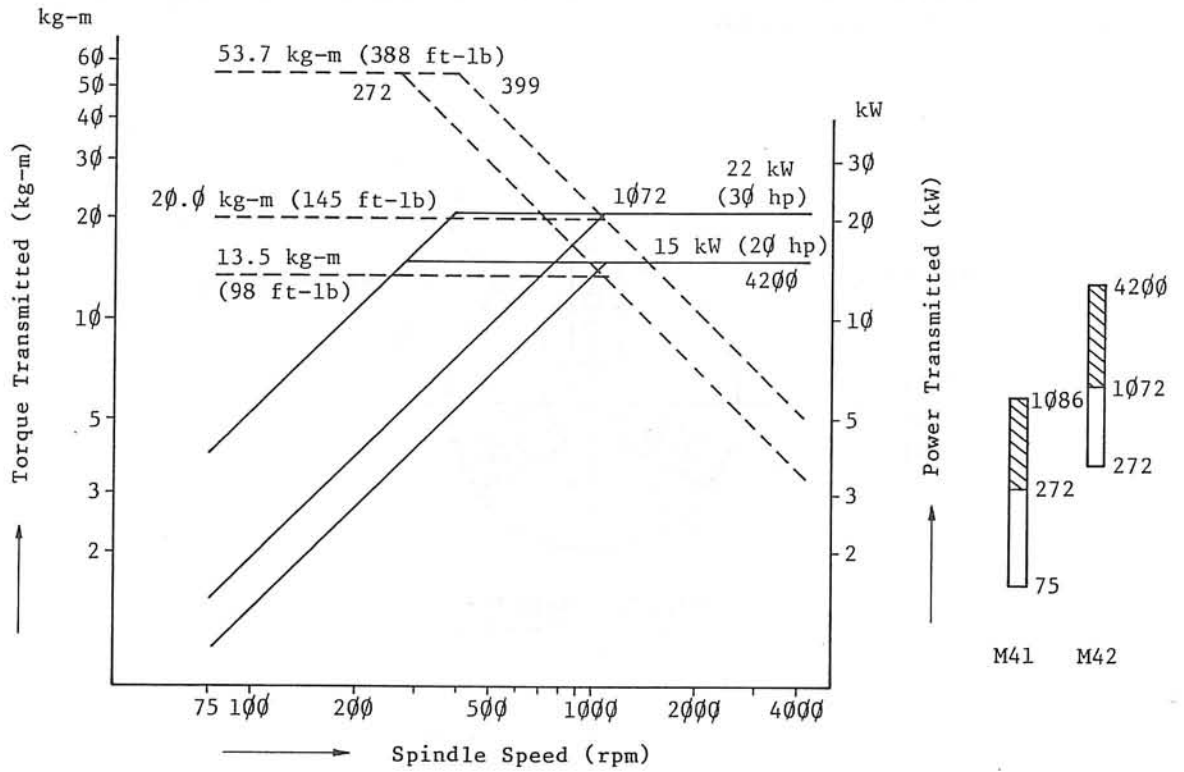
Range Selection	Available Spindle Speeds
M41	75 to 1086 rpm
M42	272 to 4200 rpm

- Note 1: With an M code spindle speed range to be used is selected, and an S code consisting of up to four digits determines the spindle speed.
- Note 2: To change the spindle speed range while the spindle is rotating, stop spindle rotation once before commanding a new spindle speed range selection M code.
- Note 3: When determining the cutting conditions, spindle speed range to be employed should be determined referring to the spindle speed - transmission power diagrams provided below. And program should be made so that cutting is performed within a constant power range where practicable.

Spindle motor: 15/11 kW (20/15 hp)

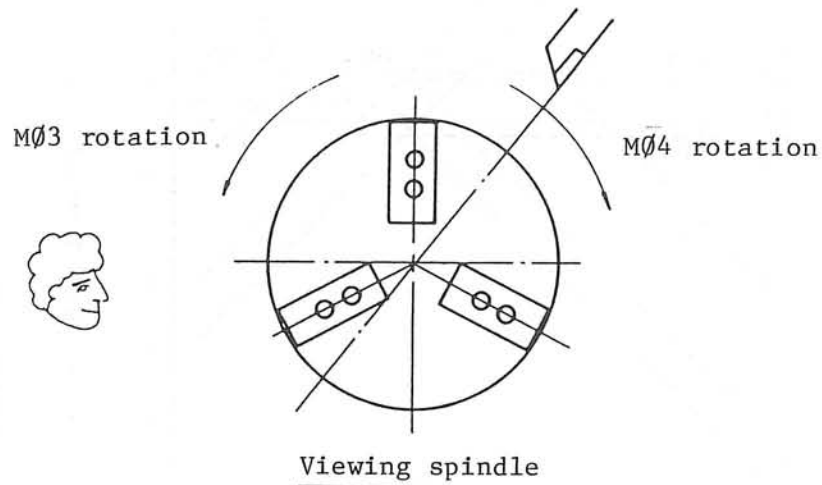


Spindle motor: 22/15 kW (30/20 hp), optional



1-8. SPINDLE ROTATION DIRECTION SELECTION

Spindle rotation direction, either CW or CCW, and spindle stop are commanded by a two-digit M code.



Spindle CW	M03
Spindle CCW	M04
Spindle Stop	M05

Note 1: Clockwise (CW) spindle rotation is in the direction to advance a right-handed screw into the workpiece.

Note 2: Changing the spindle rotation direction directly from CW to CCW or from CCW to CW is not possible for the following models. For these models, it is necessary to stop spindle rotation once by specifying M05 before changing spindle rotation.

1-9. TAPE FORMAT

1-9-1. Program Process Sheet

PROCESS SHEET								
N	G	X(U)	Z(W)	I	K	F	S,T,M	LF
%								LF
O104								LF
N001	G00	X800	Z2000					LF
N002		X250	Z100				M41	LF
N003			Z53				S120 M03 M08	LF
N004			Z43					LF
N005	G01	X205				F0.3		LF
N006	G00	X250	Z50					LF
N007			Z40					LF
N008	G01	X205				F0.25		LF
N009	G00	X230	Z58					LF
N010		X192						LF
N011	G01		Z48			F0.4		LF
N012		X200						LF
N013			Z40					LF
N014	G00	X210	Z49					LF
N015		X196						LF
N016	G01							LF

Shown above is an example of a program process sheet. It is a listing which details part programming instructions comprising alphabetic and numeric characters arranged in a definite programming scheme. With these instructions, operations of the machine are expressed.

Each line of the process sheet starts with Sequence Number (N word) and ends with the LF code. A group of commands written in one line is referred to as a "block" and it describes one machine operation.

Usually, the machine executes the programmed commands block by block in the order as programmed to produce a part.

The steps preparing such program process sheet is called "programming".

Note 1: The first block of each program must contain % (or ER) code which is directly followed by LF (or CR) code.

Note 2: Programs should be identified by program name or number preceded by address character O.

Note 3: Each block of commands is separated by LF (or CR) code. This code must be provided at the end of each block without fail.

1-9-2. Tape Format

When preparing a program process sheet, commands are expressed by the following codes in the specified format.

Item	Code	Number of Digits in Tape Command	
		Metric System	Inch System
Program name	O	4	4
Sequence name	N	4	4
Preparatory function	G	3	3
Axis movement command	X,Z	<u>+5.3</u> (mm)	<u>+4.4</u> (inch)
	I,K	<u>+5.3</u> (mm)	<u>+4.4</u> (inch)
LAP command	D,U,W H,L	5.3 (mm)	4.4 (inch)
	E	<u>+5.3</u> (mm/rev.)	<u>+4.4</u> (inch/rev.)
	A,B	5.3 (deg.)	4.4 (deg.)
Feed command	F	4.3 (mm/rev.)	<u>+3.4</u> (inch/rev.)
Dwell command	F	4.2 (sec.)	4.2 (sec.)
Tool command	T	6 (or 4)	6 (or 4)
Spindle speed command	S	4	4
Miscellaneous function	M	2	2
Synchronizing command	P	<u>+4</u>	<u>+4</u>

Note 1: Both alphabetic and numerical characters may be used in program and sequence names.

Note 2: The word format for tool command is either T6 or T4 according to the availability of the tool nose radius compensation feature:

T6 with tool nose radius compensation function

T4 without such function

Note 3: P word used for synchronizing execution of the commands is effective only for two-saddle specification of LC series lathes.

Note 4: The above indicated figures show the number of digits used in programming to express the command data in the unit indicated in ().

+5.3 (mm) indicates that:

Maximum programmable digits of command data are five left to the decimal point and three right to the decimal point when the data is expressed in the unit of "mm".

Note 5: Maximum programmable command data are:

+99999.999 mm X, Z, I, K, D, E, and L
9999.999 mm/rev. F (feedrate)
9999.99 sec. F (dwell)

1-9-3. Position of Decimal Point

With the OSP500L-G/OSP5000L-G, a decimal point can be used in command data as a standard feature. Units of the data available with the control are shown below and selection of the unit to be employed can be done 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.

Note that for machine data such as zero offset, the data smaller than 1 μm cannot be set.

Metric System

1 μm
 10 μm
 1 mm

Inch System

1/10000 inch
 1 inch

Unit Data Table (Value for data "1")

	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)	0.01 (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): F,E	0.1 (mm/min)	1 (mm/min)	1 (mm/min)	0.01 (inch/min)	1 (inch/min)
Angle: A,B,C	0.001 (deg.)	0.01 (deg.)	1 (deg.)	0.001 (deg.)	1 (deg.)
Time: F,E	0.01 (sec)	0.1 (sec)	1 (sec)	0.01 (sec)	1 (sec)
Spindle rpm: S	1 (rpm)	1 (rpm)	1 (rpm)	1 (rpm)	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) $\emptyset.001$ mm movement of X-axis | X $\emptyset.001$ |
| 2) 1 \emptyset mm movement of X-axis | X1 \emptyset |
| 3) 1 $\emptyset\emptyset.01\emptyset$ mm movement of X-axis | X1 $\emptyset\emptyset.01$ |
| 4) Feedrate of $\emptyset.23456$ mm/rev. | F $\emptyset.23456$ |

Following commands are all handled as X1 mm:

X1
 X1. \emptyset
 X1. $\emptyset\emptyset$
 X1. $\emptyset\emptyset\emptyset$

Example 2: 1 \emptyset μ m unit system

Commanding:

- | | |
|---|----------------------------------|
| 1) $\emptyset.001$ mm movement of X-axis | X $\emptyset.1$ |
| 2) 1 \emptyset mm movement of X-axis | X1 $\emptyset\emptyset\emptyset$ |
| 3) 1 $\emptyset\emptyset.01\emptyset$ mm movement of X-axis | X1 $\emptyset\emptyset01$ |
| 4) Feedrate of $\emptyset.23456$ mm/rev. | F23.456 |

Example 3: 1 μ m unit system

Commanding:

- | | |
|---|---|
| 1) $\emptyset.001$ mm movement of X-axis | X1 |
| 2) 1 \emptyset mm movement of X-axis | X1 $\emptyset\emptyset\emptyset\emptyset$ |
| 3) 1 $\emptyset\emptyset.01\emptyset$ mm movement of X-axis | X1 $\emptyset\emptyset01\emptyset$ |
| 4) Feedrate of $\emptyset.23456$ mm/rev. | F234.56 |

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

F1.2345678 Acceptable
 F1 $\emptyset\emptyset.000001$ Alarm (9 digits)

PROGRAM EXAMPLES PROVIDED IN THIS INSTRUCTION MANUAL ARE ALL PREPARED ASSUMING 1 mm UNIT SYSTEM.

1-9-4. Position of Decimal Point for $\emptyset.1 \mu\text{m}$ Specification (optional)

Units of the data available with the control for $\emptyset.1 \mu\text{m}$ specification are shown below and selection of the unit to be employed can be done 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.

Note that for machine data such as zero offset, data smaller than $1 \mu\text{m}$ cannot be set.

Metric System

$\emptyset.1 \mu\text{m}$
 $1 \mu\text{m}$
 $1\emptyset \mu\text{m}$
 1 mm

Inch System

$1/1\emptyset\emptyset\emptyset\emptyset\emptyset \text{ inch}$
 $1/1\emptyset\emptyset\emptyset\emptyset \text{ inch}$
 1 inch

Unit Data Table (Value for data "1")

	Metric System				Inch System		
	$\emptyset.1 \mu\text{m}$	$1 \mu\text{m}$	$1\emptyset \mu\text{m}$	1 mm	$1/1\emptyset\emptyset\emptyset\emptyset\emptyset \text{ inch}$	$1/1\emptyset\emptyset\emptyset\emptyset \text{ inch}$	1 inch
Length: X,Z,I,K, D,H,L,U,W	$\emptyset.0\emptyset\emptyset\emptyset 1$ (mm)	$\emptyset.0\emptyset\emptyset 1$ (mm)	$\emptyset.0\emptyset 1$ (mm)	1 (mm)	$\emptyset.0\emptyset\emptyset\emptyset\emptyset 1$ (inch)	$\emptyset.0\emptyset\emptyset\emptyset 1$ (inch)	1 (inch)
Feed (/rev): F,E	$\emptyset.0\emptyset\emptyset\emptyset 1$ (mm/ rev)	$\emptyset.0\emptyset\emptyset 1$ (mm/ rev)	$\emptyset.0\emptyset 1$ (mm/ rev)	1 (mm/ rev)	$\emptyset.0\emptyset\emptyset\emptyset\emptyset 1$ (inch/ rev)	$\emptyset.0\emptyset\emptyset\emptyset 1$ (inch/ rev)	1 (inch/ rev)
Feed (/min): F,E	$\emptyset.0\emptyset 1$ (mm/ min)	$\emptyset.1$ (mm/ min)	1 (mm/ min)	1 (mm/ min)	$\emptyset.0\emptyset\emptyset 1$ (inch/ min)	$\emptyset.0\emptyset 1$ (inch/ min)	1 (inch/ min)
Angle: A,B,C	$\emptyset.0\emptyset\emptyset\emptyset 1$ (deg.)	$\emptyset.0\emptyset\emptyset 1$ (deg.)	$\emptyset.0\emptyset 1$ (deg.)	1 (deg.)	$\emptyset.0\emptyset\emptyset\emptyset 1$ (deg.)	$\emptyset.0\emptyset\emptyset 1$ (deg.)	1 (deg.)
Time: F,E	$\emptyset.0\emptyset\emptyset 1$ (sec)	$\emptyset.0\emptyset 1$ (sec)	$\emptyset.1$ (sec)	1 (sec)	$\emptyset.0\emptyset\emptyset 1$ (sec)	$\emptyset.0\emptyset 1$ (sec)	1 (sec)
Spindle rpm: S	1 (rpm)	1 (rpm)	1 (rpm)	1 (rpm)	1 (rpm)	1 (rpm)	1 (rpm)
Surface speed: S	1 (m/min)	1 (m/min)	1 (m/min)	1 (m/min)	1 (feet/ min)	1 (feet/ min)	1 (feet/ min)

1-9-5. Tape Code

The OSP500L-G/OSP500L-G is designed to accept the instructions expressed in both EIA and ISO coding system.

Information punched on a paper tape comprises up to eight holes and combination of punched and non-punched holes expresses characters. One character contains:

Odd number holes for EIA coding system, and
Even number holes for ISO coding system

The control can automatically recognize the coding system when it reads in the taped data.

For details, refer to 4-3-6 in the Operation and Maintenance Manual

CAUTION

COMBINED USE OF ISO AND EIA CODING SYSTEMS IN ONE PART PROGRAM
TAPE IS NOT ALLOWED.

EIA and ISO coding system conforms to the following standard:

EIA EIA RS-244-A
ISO ISO R840

EIA/ISO Code Chart

EIA Code									ISO Code									Remarks
Channel	8	7	6	5	4	3	2	1	Channel	8	7	6	5	4	3	2	1	
Character									Character									
0			○		○				Ø			○	○	○				Numeric Character 0
1					○			○	1	○		○	○	○			○	Numeric Character 1
2					○			○	2	○		○	○	○			○	Numeric Character 2
3				○		○		○	3			○	○	○			○	Numeric Character 3
4					○			○	4	○		○	○	○	○			Numeric Character 4
5				○		○		○	5			○	○	○	○		○	Numeric Character 5
6				○		○		○	6			○	○	○	○	○		Numeric Character 6
7					○			○	7	○		○	○	○	○	○		Numeric Character 7
8					○			○	8	○		○	○	○	○			Numeric Character 8
9				○		○		○	9			○	○	○	○		○	Numeric Character 9
A			○	○				○	A		○			○			○	Alphabetic Character A
B			○	○				○	B		○			○			○	Alphabetic Character B
C			○	○				○	C	○	○			○			○	Alphabetic Character C
D			○	○				○	D	○	○			○			○	Alphabetic Character D
E			○	○				○	E	○	○			○			○	Alphabetic Character E
F			○	○				○	F	○	○			○			○	Alphabetic Character F
G			○	○				○	G	○	○			○			○	Alphabetic Character G
H			○	○				○	H		○			○			○	Alphabetic Character H
I			○	○				○	I	○	○			○			○	Alphabetic Character I
J			○	○				○	J	○	○			○			○	Alphabetic Character J
K			○	○				○	K		○			○			○	Alphabetic Character K
L			○	○				○	L	○	○			○			○	Alphabetic Character L
M			○	○				○	M		○			○			○	Alphabetic Character M
N			○	○				○	N		○			○			○	Alphabetic Character N
O			○	○				○	O	○	○			○			○	Alphabetic Character O
P			○	○				○	P		○	○		○				Alphabetic Character P
Q			○	○				○	Q	○	○			○			○	Alphabetic Character Q
R			○	○				○	R	○	○			○			○	Alphabetic Character R
S			○	○				○	S		○	○		○			○	Alphabetic Character S
T			○	○				○	T	○	○			○			○	Alphabetic Character T
U			○	○				○	U		○	○		○			○	Alphabetic Character U
V			○	○				○	V		○	○		○			○	Alphabetic Character V
W			○	○				○	W	○	○			○			○	Alphabetic Character W
X			○	○				○	X	○	○			○				Alphabetic Character X
Y			○	○				○	Y		○	○		○			○	Alphabetic Character Y
Z			○	○				○	Z		○	○		○			○	Alphabetic Character Z
+			○	○				○	+			○	○	○			○	Plus Sign
-			○	○				○	-			○	○	○			○	Minus Sign
/			○	○				○	/	○	○			○			○	Block Delete or Symbol of Division
BLANK					○			○	NULL					○				Null
SPACE				○				○	SPACE	○	○			○				Space
TAB			○	○				○	HT					○			○	Tab
ER				○				○	%	○	○			○			○	Program Start
CR/EOB	○				○			○	NL/LF					○			○	End of Block
.			○	○				○	CR	○				○			○	Carriage Return *
.			○	○				○	.			○		○			○	Period
.			○	○				○	,	○	○			○			○	Comma
DEL			○	○				○	DEL	○	○			○			○	Delete *
BS			○	○				○	BS	○				○				Back Space *
(○				○	(○		○				Control Out
)			○					○)	○	○			○			○	Control In
									\$					○			○	Dollar Symbol
									*	○	○			○			○	Symbol of Multiplication
									=	○	○			○			○	Equality Sign
									?			○		○			○	Question Mark
									{	○	○			○			○	Brace, Left
									}	○	○			○			○	Brace, Right
									:			○		○			○	Colon
									;	○	○			○			○	Semicolon

* The control ignores when used as cutting data.

1-9-6. EIA Special Codes

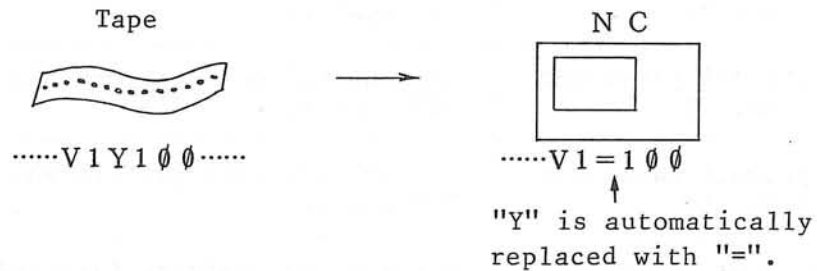
In making a program in the EIA code, some characters cannot be used because it does not have corresponding codes. These characters are "=", "*", etc. Therefore, to make an EIA coded program, it is necessary to replace such characters with other EIA code patterns temporarily.

When such characters are read by the OSP500L-G/OSP5000L-G, they must be converted again to the original characters using the editing function.

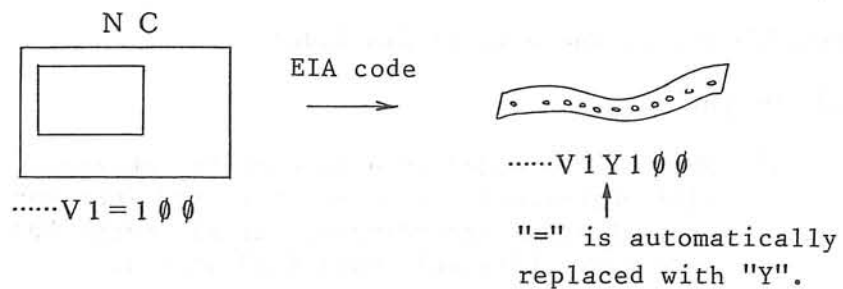
The OSP500L-G/OSP5000L-G has the function to automatically replace the such characters not supported by the EIA coding system in the stage of input to the OSP from paper tape and also output to a tape punch by setting them as the EIA special characters.

Input/Output Example

When EIA code NC tape is punched using character "Y" for symbol "=", it is necessary to set special code "Y" for "=" at the NC. Under such setting, the NC automatically reads character "Y" as symbol "=".



To punch out the program on a tape in the EIA code, symbol "=" is automatically replaced with "Y".



Note: If the special code setting is made as explained above, character "Y" is always used as symbol "=".

1-9-6-1. Setting of EIA Special Characters

The characters which can be set as EIA special characters are the following five characters and a spare character.

=, *, [,], \$

Input and output of the spare character are always made after converting it to the special code which is set for any arbitrary code, disregarding of EIA and ISO code system.

These special codes are set at NC optional parameter (bit) No. 26 through No. 32 in bit pattern.

NC optional parameter (bit) No. 26	Set the EIA code pattern which substitutes "=" symbol.
NC optional parameter (bit) No. 27	Set the EIA code pattern which substitutes "*" symbol.
NC optional parameter (bit) No. 28	Set the EIA code pattern which substitutes "[" symbol.
NC optional parameter (bit) No. 29	Set the EIA code pattern which substitutes "]" symbol.
NC optional parameter (bit) No. 30	Set the EIA code pattern which substitutes "\$" symbol.
NC optional parameter (bit) No. 31	Set the code pattern (irregular code) which replaces the character set at No. 32.
NC optional parameter (bit) No. 32	ASCII code to be replaced with the code set at No. 31. (regular code)

1-9-6-2. Input/Output of The Special EIA Codes

(1) Output

- a) When an EIA coded program contains characters set as EIA special characters, they are converted into and punched as EIA special codes accordingly. If all zeros (0) are set, an EIA code error (illegal character) occurs.
- b) When a program being punched contains the character "\$", and a special code has been set for it, the file name is output in the EIA code. If not, the file name is output in the ISO code.

(2) Input

- a) When special codes for special characters ("=", "*", etc.) are used in the EIA coded program being read in or verified, these special codes are converted into ASCII codes and then read in. This special code conversion is not effective if all the parameters are set at "0".
- b) When the first data of the EIA coded program being read in or verified is a special code for the special character "\$", the character-string following character "\$" is read as the file name.

(3) Input/output of Spare Character (in the EIA/ISO code)

- a) During reading in, data coded in an irregular code is first converted into a regular code and then input.
- b) During punching, data coded in a regular code is first converted into an irregular code and then punched.
- c) The regular/irregular code conversion is not effective if all the parameters are set at "0".

Note 1: Conversion from the special code to the EIA code is carried out for all special codes indiscriminately.

Example: If the "01100001" pattern is set to convert special character "=" into EIA code "A", "A=B" punched in the special code is output as "AAB". Reading this into the control causes "A=B" to be stored.

Note 2: The special code conversion is effective only in the PIP command. It is not effective in the optional DNC mode.

1-10. DETERMINING CUTTING CONDITIONS

Cutting conditions such as spindle speed, feedrate and depth of cut are selected more or less in the same manner as with a conventional lathe. With CNC lathes, these conditions are entered in a program process sheet using alphanumeric in the predetermined format.

Formulas used to determine cutting conditions:

$$V = \frac{\pi \times D \times n}{1000}$$

where, V = cutting speed, m/min
 π = circular constant (3.14)
 D = workpiece diameter, mm
 n = spindle speed, rpm

$$HP = \frac{V \times K \times t \times f}{75 \times 60}$$

$$kW = \frac{V \times K \times t \times f}{60000}$$

where, HP = required horse power, hp

kW = required power, kW

V = cutting speed, m/min

K = cutting resistance, kg/mm²

Example: 200 for S45C (JIS, carbon steel)
 120 for FC20 (JIS, cast iron)

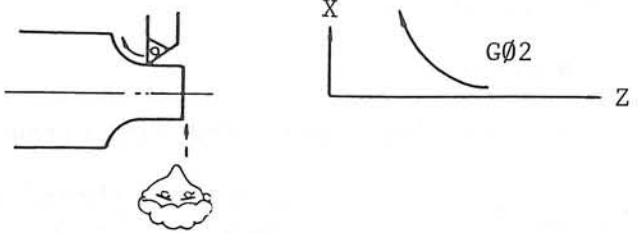
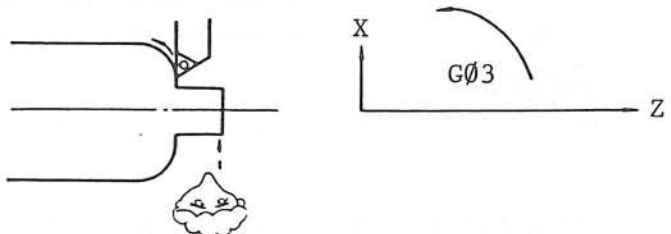
t = depth of cut, mm

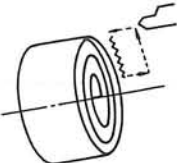
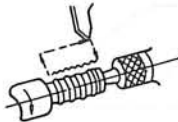
f = feedrate, mm/rev.

1-11. PROGRAMMING CODES AND WORD FORMAT (1 mm TAPE INPUT UNIT SYSTEM)

Word Format or Code	Associated Information and Function
<p style="text-align: center;">% (ER)</p>	<p>Commanded at the beginning of a part program tape.</p> <p style="margin-left: 40px;">ISO % EIA ER</p>
<p style="text-align: center;">LF (CR)</p>	<p>Commanded at the end of each block.</p> <p style="margin-left: 40px;">ISO LF (Line Feed) EIA CR (Carriage Return)</p>
<p style="text-align: center;">Ooooo</p>	<p>Program Number or Program Name</p> <p>Entered at the beginning of a part program to identify respective programs.</p> <p>Program Number: Up to four numeric characters following address character "0" are used to indicate a Program Number.</p> <p>Program Name : When an alpha character appears following address character "0", such expression is referred to as "Program Name". Up to three alphanumerics can be used following the first alphabetic character.</p> <p>Note 1: No spacing or other information is allowed between address character "0" and its following number or name. Program number or name must be expressed in up to four alphanumerics without spacing in between the characters.</p> <p>Note 2: Program number or name must be located at the beginning of a part program; sequence number or name or other information cannot be placed before the program number or name.</p> <p>Note 3: Command such as a G code cannot be provided right after a program number or name.</p>
<p style="text-align: center;">Noooo</p>	<p>Sequence Number or Sequence Name</p> <p>Entered at the beginning of each block to identify respective blocks in a part program.</p> <p>Numbers are usually used to indicate the execution order of blocks. Although program numbers are generally assigned with consecutive numbers, such numbers may not necessarily be consecutive. Sequence name is used to identify a specific block in a part program.</p>

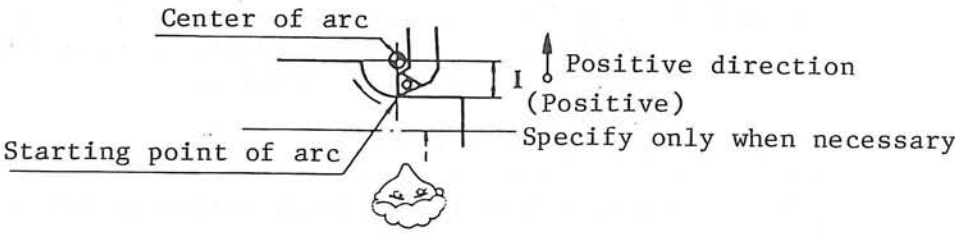
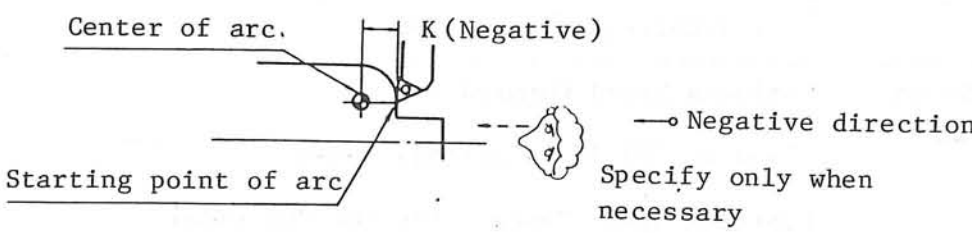
Word Format or Code	Associated Information and Function
	<p>Sequence Number: Up to four numeric characters following address character "N" are used to indicate a Sequence Number.</p> <p>Sequence Name : When an alpha character appears following address character "N", such expression is referred to as "Sequence Name". Up to three alphanumerics can be used following the first alphabetic character.</p> <p>Note 1: No spacing or other information is allowed between address character "N" and its following number or name. Sequence number or name must be expressed in up to four alphanumerics without spacing in between the characters.</p> <p>Note 2: When a sequence name is used, be sure to place a space code or a tab code following the name.</p>
X+000.000	<p>Dimension Word: Diameter</p> <p>Used to specify X-axis coordinate in diameter.</p> <p>For details, refer to 1-4.</p>
Z+000.000	<p>Dimension Word: Longitudinal Dimension</p> <p>Used to specify Z-axis coordinate.</p> <p>For details, refer to 1-4.</p>
G00	<p>G Code: Three numeric characters following address character G establishes the mode of axis movements.</p>
G00	<p>Rapid Feed</p> <p>Used to feed the axes at a rapid feedrate to the commanded coordinate position.</p> <p>For details, refer to 2-1.</p>
G01	<p>Linear Interpolation</p> <p>Used to cut a straight line parallel to X- or Z-axis or a taper. Feedrate to be employed in this mode is commanded by an F word, F0000.</p> <p>For details, refer to 2-2 and 2-3.</p>

Word Format or Code	Associated Information and Function
G02	<p>Circular Interpolation, CW</p> <p>Used to cut an arc in the clockwise direction. Feedrate to be employed is commanded by an F word, Foooo, as in G01 mode.</p> <p>For details, refer to 2-4.</p> 
G03	<p>Circular Interpolation, CCW</p> <p>Used to cut an arc in the counterclockwise direction. Feedrate to be employed is commanded by an F word, Foooo, as in G01 mode.</p> <p>For details, refer to 2-4.</p> 
G04	<p>Dwell</p> <p>Used to activate dwell function which stops axis motion for any required duration of time during a machining cycle.</p> <p>Duration of dwell movements is programmed in an F word: G04 Foooo.</p> <p>G04 F12.3 stops axis motion for 12.3 seconds, for instance.</p>
G13	<p>Turret Selection: Turret A</p> <p>Used to select the turret to be used for two-saddle and two-turret models.</p> <p>For details, refer to 1-5 and 1-6.</p>

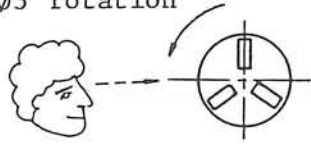
Word Format or Code	Associated Information and Function
G14	<p>Turret Selection: Turret B</p> <p>Used to select the turret to be used for two-saddle and two-turret models.</p> <p>For details, refer to 1-5 and 1-6.</p>
G31	<p>Fixed Thread Cutting Cycle</p> <p>See explanation of G33.</p>
G32	<p>Fixed Thread Cutting Cycle: End Face (Transverse)</p> <div style="display: flex; align-items: center; justify-content: center;">  <div style="margin-left: 20px;"> <p>Automatic thread cutting cycle as shown at the left is executed.</p> </div> </div> <p>For details, refer to 3-1 and 3-2.</p>
G33	<p>Fixed Thread Cutting Cycle: Longitudinal</p> <div style="display: flex; align-items: center; justify-content: center;">  <div style="margin-left: 20px;"> <p>Automatic thread cutting cycle as shown at the left is executed.</p> </div> </div> <p>For details, refer to 2-5, 3-1 and 3-2.</p>
G34	<p>Variable Lead Thread Cutting Cycle: Increasing Lead</p> <p>Used to call out increasing lead thread cutting cycle.</p> <p>For details, refer to 3-1 and 3-2.</p>
G35	<p>Variable Lead Thread Cutting Cycle: Decreasing Lead</p> <p>Used to call out decreasing lead thread cutting cycle.</p> <p>For details, refer to 3-1 and 3-2.</p>
G40	<p>Tool Nose Radius Compensation: Cancel</p> <p>Used to cancel the tool nose radius compensation function.</p>

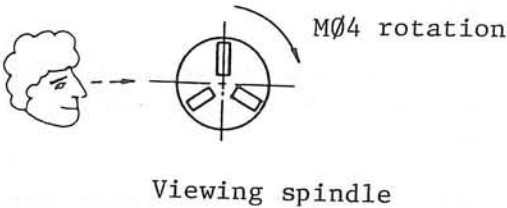
Word Format or Code	Associated Information and Function
G41	<p>Tool Nose Radius Compensation: ID Ordinary Cutting</p> <p>Used to call out the tool nose radius compensation mode for ordinary ID cutting cycle.</p>
G42	<p>Tool Nose Radius Compensation: OD Ordinary Cutting</p> <p>Used to call out the tool nose radius compensation mode for ordinary OD cutting cycle.</p>
G50	<p>Maximum Spindle Speed Designation</p> <p>Used to set the allowable maximum spindle speed.</p> <p>For details, refer to 3-3.</p>
	<p>Zero Offset</p> <p>Used to establish a zero point.</p> <p>For details, refer to 10.</p>
G64	<p>Droop Control OFF</p> <p>Used to turn off droop control mode.</p> <p>For details, refer to 8.</p>
G65	<p>Droop Control ON</p> <p>Used to turn on droop control mode.</p> <p>For details, refer to 8.</p>
G90	<p>Absolute Programming</p> <p>Used to establish absolute programming mode.</p> <p>When the control is reset, it is in the G90 mode.</p> <p>For details, refer to 3-4.</p>
G91	<p>Incremental Programming</p> <p>Used to establish incremental programming mode.</p> <p>For details, refer to 3-4.</p>

Word Format or Code	Associated Information and Function
G94	<p>Feed per Minute Mode</p> <p>Used to establish mm/min feedrate mode.</p> <p>For details, refer to 9.</p>
G95	<p>Feed per Revolution Mode</p> <p>Used to establish mm/rev. feedrate mode.</p> <p>When the control is reset, it is in the G95 mode.</p> <p>For details, refer to 9.</p>
G96	<p>Constant Speed Cutting ON</p> <p>Used to establish the constant speed cutting mode.</p> <p>For details, refer to 3-3.</p>
G97	<p>Constant Speed Cutting OFF</p> <p>Used to cancel the constant speed cutting mode.</p> <p>For details, refer to 3-3.</p>
G100	<p>Priority Command for Turret A or B Independent Cutting</p> <p>Used to designate the priority of turret for independent cutting. (active only on two-saddle models)</p> <p>For details, refer to 6-2.</p>
G110	<p>Constant Speed Cutting on Turret A</p> <p>Used to establish the constant speed cutting mode for turret A. (active only on two-turret and two-saddle models)</p> <p>When the control is reset, it is in the G110 mode.</p> <p>For details, refer to 3-3 and 6.</p>
G111	<p>Constant Speed Cutting on Turret B</p> <p>Used to establish the constant speed cutting mode for turret B. (active only on two-turret and two-saddle models)</p> <p>For details, refer to 3-3 and 6.</p>

Word Format or Code	Associated Information and Function
<p><u>I</u>+000.000</p>	<p>Used to specify the distance of the center of the arc with respect to the starting point of it in X-axis direction.</p>  <p>Center of arc</p> <p>Starting point of arc</p> <p>Positive direction (Positive)</p> <p>Specify only when necessary</p> <p>For details, refer to 2-4.</p> <hr/> <p>Used to specify the taper amount in radius, in G31 and G33 longitudinal thread cutting fixed cycle.</p> <p>Used to specify the shift amount of thread cutting starting point from the cycle starting point, in G32 transverse thread cutting fixed cycle.</p> <p>For details, refer to 3-1.</p>
<p><u>K</u>+000.000</p>	<p>Used to specify the distance of the center of the arc with respect to the starting point of it in Z-axis direction.</p>  <p>Center of arc.</p> <p>Starting point of arc</p> <p>K (Negative)</p> <p>Negative direction</p> <p>Specify only when necessary</p> <p>For details, refer to 2-4.</p> <hr/> <p>Used to specify the taper amount in G32 transverse thread cutting cycle.</p> <p>Used to specify the shift amount of thread cutting starting point from the cycle starting point, in G33 longitudinal thread cutting cycle.</p> <p>For details, refer to 3-1.</p>

Word Format or Code	Associated Information and Function
F0.000	<p>Used to specify axis feedrate. On lathes, a feedrate is programmed in "mm/rev."</p> <p>F0.5 indicates the feedrate 0.50 mm/rev. To feed the axis at a rate of 0.03 mm/rev., specify F0.03. Feedrate smaller than 1 μm unit can be commanded as F0.0123.</p> <p>For details, refer to 2-2.</p> <hr/> <p>Used to specify duration of dwell motion in G04 mode.</p>
E0.000	<p>Used to specify lead variation rate per lead in variable lead thread cutting.</p> <p>For details, refer to 3-1.</p> <hr/> <p>Used to specify the feedrate in LAP cycle when cutting is performed along the contour. (optional)</p>
T0000 T000000	<p>Tool Selection Command</p> <p>4-digit T command without tool nose radius compensation function</p> <p>6-digit T command with tool nose radius compensation function</p> <p>For details, refer to 1-6.</p>
S0000	<p>Spindle Speed Command</p> <p>Used to specify a spindle speed.</p> <p>Cutting Speed Command (in the G96 mode)</p> <p>Used to specify a cutting speed.</p> <p>For details, refer to 1-7.</p>

Word Format or Code	Associated Information and Function
M00	<p>M Code: Up to three numeric characters following address character M are used for specifying various miscellaneous machine functions such as spindle CW/CCW, coolant ON/OFF, etc.</p>
M00	<p>Program Stop</p> <p>When M00 is executed, machine operation goes into cycle stop state; spindle rotation and coolant supply are also brought to a stop.</p> <p>To continue execution of the part program, press the CYCLE START button.</p> <p>This program stop function is effectively used for measuring finished dimensions and also for removing chips during cycle.</p>
M01	<p>Optional Stop</p> <p>M01 performs the same function as M00 Program Stop, except that the control ignores programmed M01 codes unless the <u>OPTIONAL STOP</u> switch is turned ON.</p>
M02	<p>End of Program</p> <p>M02 provided at the end of a part program resets the control.</p>
M03	<p>CW Rotation</p> <p>M03 starts the spindle rotation to advance a right-handed screw into the workpiece.</p> <div data-bbox="690 1386 1112 1596" style="text-align: center;"> <p>M03 rotation</p>  <p>Viewing spindle</p> </div> <p>For details, refer to 1-8.</p>

Word Format or Code	Associated Information and Function
M04	<p>CCW Rotation</p> <p>M04 starts spindle rotation to retract a right-handed screw from the workpiece.</p> <div data-bbox="634 499 1138 705" style="text-align: center;">  <p>M04 rotation</p> <p>Viewing spindle</p> </div> <p>For details, refer to 1-8.</p>
M05	<p>Spindle Stop</p> <p>M05 stops spindle rotation.</p> <p>For details, refer to 1-8.</p>
M08	Coolant ON
M09	Coolant OFF
M20	<p>Tailstock Barrier OFF</p> <p>For details, refer to 7.</p>
M21	<p>Tailstock Barrier ON</p> <p>For details, refer to 7.</p>
M22	<p>Cancel of M23</p> <p>For details, refer to 3-2.</p>
M23	<p>Chamfering ON</p> <p>M23 executes chamfering using a fixed cycle, in G31 through G33 thread cutting cycle.</p> <p>For details, refer to 3-2.</p>
M24	<p>Chuck Barrier OFF</p> <p>For details, refer to 7.</p>
M25	<p>Chuck Barrier ON</p> <p>For details, refer to 7.</p>

Word Format or Code	Associated Information and Function
M26	<p>Thread Lead along Z-Axis</p> <p>Thread lead is referenced to Z-axis in G34 and G35 thread cutting cycle.</p> <p>When the control is reset, it is in M26 mode.</p> <p>For details, refer to 3-2.</p>
M27	<p>Thread Lead along X-Axis</p> <p>Thread lead is referenced to X-axis in G34 and G35 thread cutting cycle.</p> <p>For details, refer to 3-2.</p>
M30	<p>End of Tape</p> <p>M30 provided at the end of a part program resets the control.</p> <p>For the OSP, usually M02 is used as the program end indicating command. However, M30 can be used instead of M02.</p>
M32	<p>Straight Infeed along Thread Face (on left face)</p> <p>When the control is reset, it is in M32 mode.</p> <p>For details, refer to 13.</p>
M33	<p>Zig-Zag Infeed in Thread Cutting</p> <p>M33 infeeds the thread cutting tool in zig-zag mode.</p> <p>For details, refer to 13.</p>
M34	<p>Straight Infeed along Thread Face (on right face)</p> <p>For details, refer to 13.</p>
M40	<p>Spindle in Neutral State</p> <p>For details, refer to 1-7.</p>

Word Format or Code	Associated Information and Function
M41	Spindle Speed Range Selection For details, refer to 1-7.
M42	Spindle Speed Range Selection For details, refer to 1-7.
M43	Spindle Speed Range Selection For details, refer to 1-7.
M44	Spindle Speed Range Selection For details, refer to 1-7.
M55	Tailstock Spindle Retract
M56	Tailstock Spindle Advance
M60	Cancel of M61 For details, refer to 3-3.
M61	With M61 active, blocks of commands are continuously executed in constant speed cutting mode without waiting for arrival of the spindle speed corresponding to the programmed constant cutting speed. For details, refer to 3-3.
M73	Pattern of Thread Cutting For details, refer to 13.
M74	Pattern of Thread Cutting For details, refer to 13.
M75	Pattern of Thread Cutting For details, refer to 13.
M83	Chuck Close
M84	Chuck Open

Word Format or Code	Associated Information and Function
M86	<p>Turret Indexing Direction: Reverse</p> <p>M86 rotates the turret in the reverse direction as T4 → T3 → T2 → T1.</p> <p>For details, refer to 1-6.</p>
M87	<p>Turret Indexing Direction: Forward</p> <p>M87 rotates the turret in the forward direction as T1 → T2 → T3 → T4.</p> <p>For details, refer to 1-6.</p>
M88	Air Blower OFF (optional)
M89	Air Blower ON (optional)
M90	Door (Cover) Close (optional)
M91	Door (Cover) Open (optional)
M100	Waiting Synchronization Command in the 4-axis simultaneous cut mode
/	<p>Block Delete</p> <p>The block delete slash code is provided either at the beginning of a block or right after sequence name (Noooo): N000/G00</p> <p>When the BLOCK DELETE switch on the operation panel is ON, the control ignores all part program blocks that contain a block delete slash code (/).</p> <p>For details, refer to 3-5.</p> <hr/> <p>When a slash code (/) appears at a position other than specified above, it is handled as the sign of division.</p>
P+oooo	<p>Synchronization Command</p> <p>Used to synchronize execution of blocks of commands in the 4-axis simultaneous cut mode.</p> <p>For details, refer to 6.</p>

1-12. PROGRAMMING FORMAT

1-12-1. Model 1S (Single Turret Model)

N	G	X	Z	I,K,F	S,T,M	
%						
O0100						LF
N0001	G00	X000000	Z000000		S0000 T0000 M00	LF
N0002		X000000	Z000000		M03(M04) M08	LF
N0003						
:		(Cutting Program)				
:						
:						
N0050	G00	X000000	Z000000		M05 M09	LF
N0051					M02	LF

Explanation

% : % code must be provided at the very beginning of a part program.

O0100: Program name

N0001: Dimension words X and Z specify the turret indexing position.

Four-digit S word specifies the spindle speed.

T word with four-digit number when tool nose radius compensation is not used, or with six-digit number when it is used.

Two-digit M code specifies the spindle speed range.

N0002: M03 starts the spindle rotation in the forward direction.
(M04 in reverse)

M08 starts coolant supply.

N0050: Dimension words X and Z return the turret to the indexing position.

M05 stops spindle rotation and M08 coolant supply.

N0051: Provide M02 at the end of a part program.

1-12-2. Model 2S (Two-Turret/Two-Saddle Model)

N	G	X	Z	I,K,F	S,T,M	
%						LF
O0100						LF
N0000	G13					LF
N0001	G00	X000000	Z000000	P10	S0000 T0000 Moo	LF
N0002		X000000	Z000000	P20	M03(M04) M08	LF
:		(Cutting Program for A Turret)				
:						
:						
N0049	G00	X000000	Z000000			LF
N0050	G14					LF
N0051	G00	X000000	Z000000	P110	S0000 T0000 Moo	LF
N0052		X000000	Z000000	P120	M03(M04) M08	LF
:		(Cutting Program for B Turret)				
:						
:						
N0100	G00	X000000	Z000000	P190	M05 M09	LF
N0101					M02	

Explanation

% : % code must be provided at the very beginning of a part program.

O0100: Program name

N0000: G13 selects turret A; cutting is performed with the tools on on turret A until G14 which selects turret B is provided.

N0001: Dimension words X and Z specify the turret indexing position.

Four-digit S word specifies the spindle speed.

T word with four-digit number when tool nose radius compensation is not used, or with six-digit number when it is used.

Two-digit M code specifies the spindle speed range.

N0002: M03 starts the spindle rotation in the forward direction.
(M04 in reverse)

M08 starts coolant supply.

N0049: Dimension words X and Z return the turret to the indexing position.

- N0050: G14 selects turret B.
- N0051: Dimension words X and Z specify the turret indexing position. For S, T and M words, refer to instruction in N0001.
- N0100: Dimension words X and Z return the turret to the indexing position. M05 and M09 stop spindle rotation and coolant supply, respectively.
- N0101: Provide M02 at the end of a part program.

- Note 1: When specifying G13 and G14, G14 may be provided before G13. (Only with two-saddle, two-turret models)
- Note 2: G13 and G14 must be provided in a block without other commands.
- Note 3: For programming instructions on two-turret model, Mirror Image Program, refer to Section 14.
- Note 4: For details of programming in the simultaneous 4-axis control mode, refer to Section 6. (LC series)
- Note 5: Numbers following P should be in the ascending order. (LC series)

SECTION 2 SIMPLE PROGRAMMING

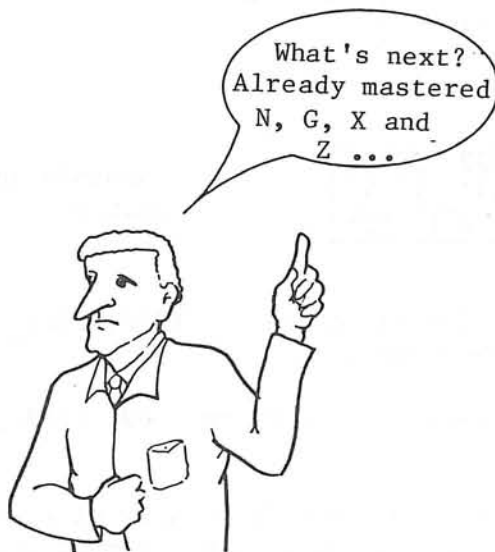
In the previous section, fundamentals of programming are explained:

- Expression and commanding methods of tool movements
- Commanding turret and tool
- Commanding spindle rotation and its direction
- Codes necessary for programming, etc.

This section deals with actual programming taking simple examples.

The examples are:

- 2-1. Rapid Traverse
- 2-2. Straight-line Cutting
- 2-3. Taper Cutting
- 2-4. Arc Cutting
- 2-5. Thread Cutting (Straight, Constant Lead Thread)



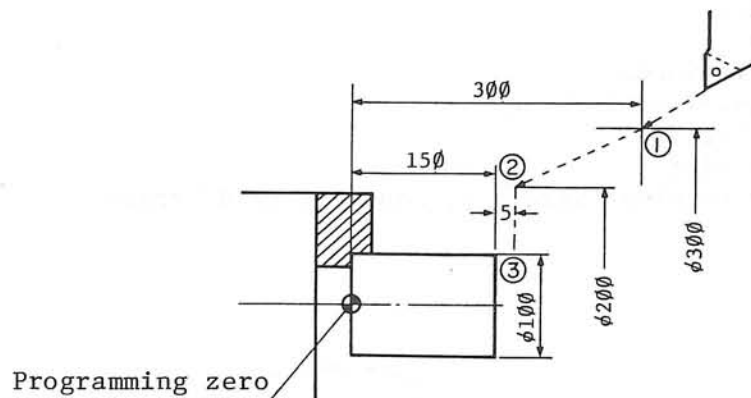
2-1. POSITIONING (G00)

(1) Format

G00 X0000.000 Z0000.000 CR

With the commands indicated above, positioning to the programmed coordinate point is carried out at a rapid traverse rate.

(2) Example Program



N001	G00	X30φ	Z30φ
N002	(G00)	X20φ	Z155
N003	(G00)	X10φ	(Z155)

Commands in () may be omitted.

N001: Positioning is made to X30φ, Z30φ position at a rapid traverse rate.

N002: Positioning is made to X20φ, Z155 position at a rapid traverse rate.

N003: Positioning is made to X10φ along X-axis at a rapid traverse rate. No Z-axis movement occurs.

Note 1: For rapid traverse rates along individual axes, refer to the Operation Manual for individual models.

Note 2: Tool path during positioning is not always straight.

Note 3: In the G00 positioning mode, acceleration and deceleration are automatically activated.

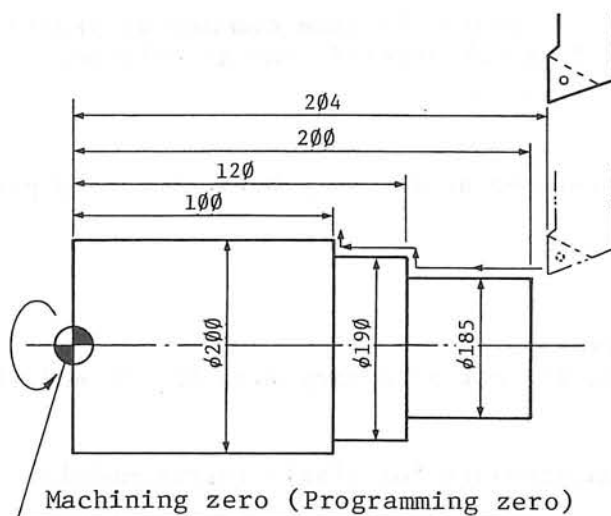
2-2. STRAIGHT-LINE CUTTING (G01)

(1) Format

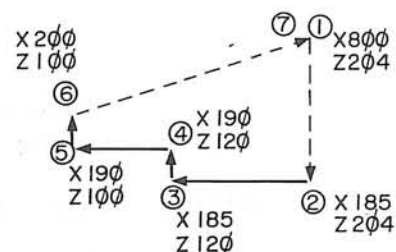
G01 X0000.000 (Z0000.000) F0.000 CR

With the commands above, axis movement from the current position to the commanded position is performed along the straight line parallel to either X- or Z-axis at a feedrate specified by an F word.

(2) Program Example



(Coordinates Commanded on Process Sheet)



The reference point for cutting (programming zero) may be taken at any position convenient for programming. However, it is recommended to take such a point at the center of the workpiece left end face. This will ease programming since such a point permits the programmer to directly enter the dimensions indicated on a part drawing.

Feedrate:

- A desired feedrate is commanded by an F word. With F0.3 command, axis is fed at 0.3 mm/rev. For an F word, numerical data smaller than the selected unit system can be specified.
- To feed the axis at a rate of 0.1 mm/rev., specify F0.1.

N	G	X	Z	I,K	F	S,T,M
%						
O0100						
N0000	G13					
N0001	G00	X800	Z204	P10		M42 S200 T0101
N0002		X185		P20		M03
N0003	G01		Z120		F0.5	M08
N0004	[]	X190				
N0005		[]	Z100			
N0006		X200	[]			
N0007	G00	X800	Z204	P30		M05 M09
N0008						M02

Commands in [], where the same command as provided in the previous block is to be entered, can be omitted.

% : % code must be provided at the very beginning of a part program.

O0100: Program name

N0000: Selection of turret A
(for 2-turret models; not necessary with 2S, 1S and single turret model)

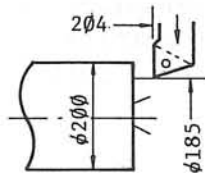
Such G code is unnecessary for single turret models.

N0001: Dimension words X and Z indicate the turret indexing position.

M, S and T commands necessary for cutting are entered.

N0002:

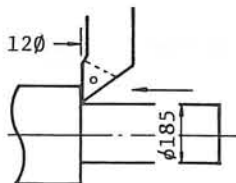
X185 positions the tool at 185 mm diameter position.



M03 starts spindle rotation in the forward direction.

N0003:

G01 Z120 feeds Z-axis up to 120 mm position at a rate commanded by an F word.

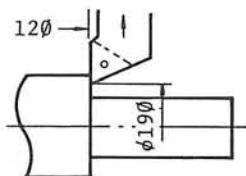


F0.5 determines the feedrate in G01 mode as 0.50 mm/rev.

M08 starts coolant supply.

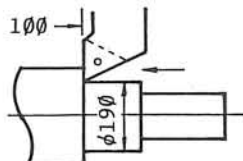
N0004:

G01 X190 feeds X-axis up to 190 mm diameter position at 0.5 mm/rev. which is specified in the preceding block to finish the shoulder.



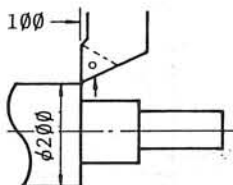
N0005:

G01 Z100 feeds Z-axis up to 100 mm position at 0.5 mm/rev.



N0006:

G01 X200 feeds X-axis up to 200 mm diameter position at 0.5 mm/rev. to finish the shoulder.



N0007: G00 X800 Z204 returns the cutting tool to the starting point.

M05 stops spindle rotation.

M09 stops coolant supply.

N0008: M02 resets the control.

Note 1: G13 selects turret A on two-turret and two-saddle (2S) model. For turret B, specify G14.

Note 2: M42 S200 in block N0001 selects spindle speed range and spindle speed.

Note 3: For a single-turret and 1S model, omit G13 command and all P words* in the program.

*: P words are effective for two-saddle models.

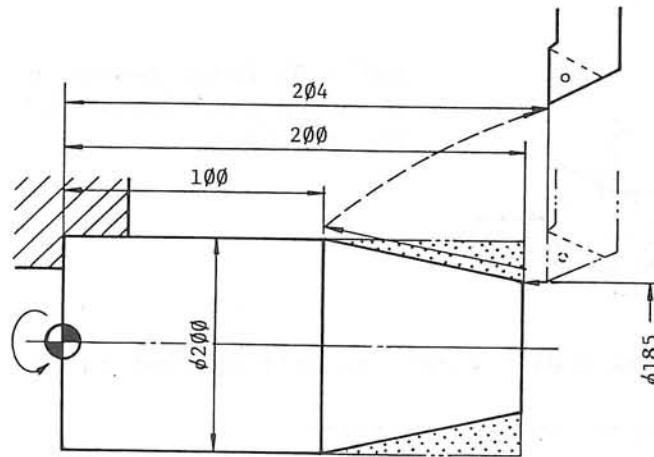
2-3. TAPER CUTTING (G01)

(1) Format

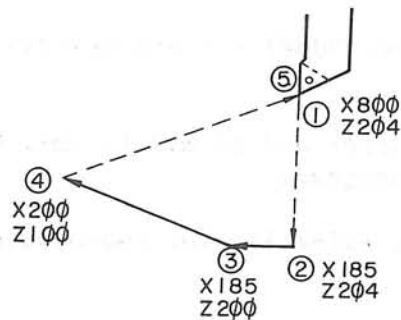
G01 X0000.000 Z0000.000 F0.000

The same commands as straight-line cutting are used for cutting a taper.

(2) Example Program



(Coordinates Commanded on Process Sheet)



N	G	X	Z	I,K	F	S,T,M
%						
O0100						
N0001	G00	X800	Z204			M42 S200 T0101
N0002		X185				M08 M03
N0003	G01		Z200		F0.5	
N0004		X200	Z100			
N0005	G00	X800	Z204			M09 M05
N0006						M02

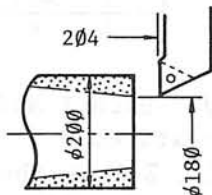
% : % code must be provided at the very beginning of a part program.

O0100: Program name

N0001: Commands in N0001 indicate:

- starting point
- spindle speed
- tool number

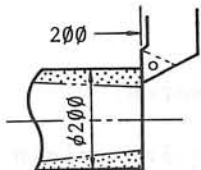
N0002:



X185 positions the tool at 185 mm diameter position.

M03 starts spindle rotation in the forward direction.

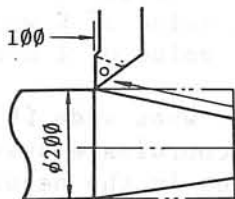
N0003:



G01 Z200 feeds Z-axis up to 200 mm position at a rate commanded by an F word.

F0.5 determines the feedrate in G01 mode as 0.50 mm/rev.

N0004:



With X200 Z100 command, cutting tool is fed to the end point for taper cutting.

N0005: G00 X800 Z204 returns the cutting tool to the starting point.

M05 stops spindle rotation.

M09 stops coolant supply.

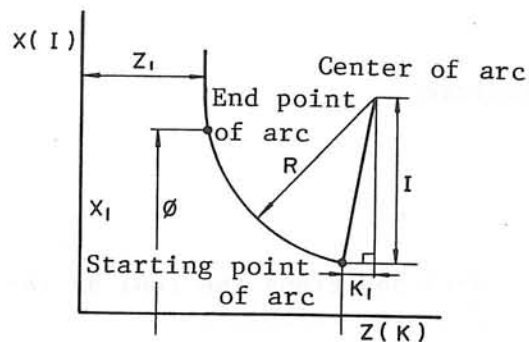
N0006: M02 resets the control.

2-4. ARC CUTTING (G02/G03)

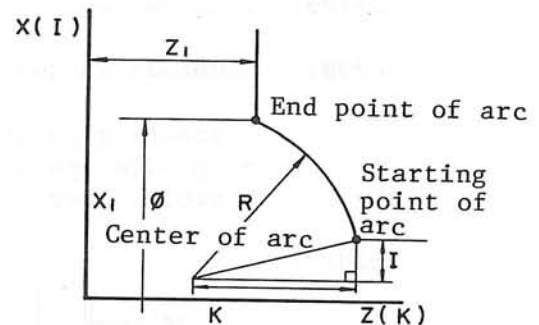
(1) Format

G02(G03) X1 Z1 I K F

G02 and G03 are used to specify the direction of arc. X and Z words indicate the coordinates of end point of arc to be cut and I and K words, the coordinates of the center of arc referenced to the coordinates of the arc starting point. I is used to express X value and K to express Z value.



G02 Both I and K values are positive.
(X1, Z1) is the coordinate of arc end point.



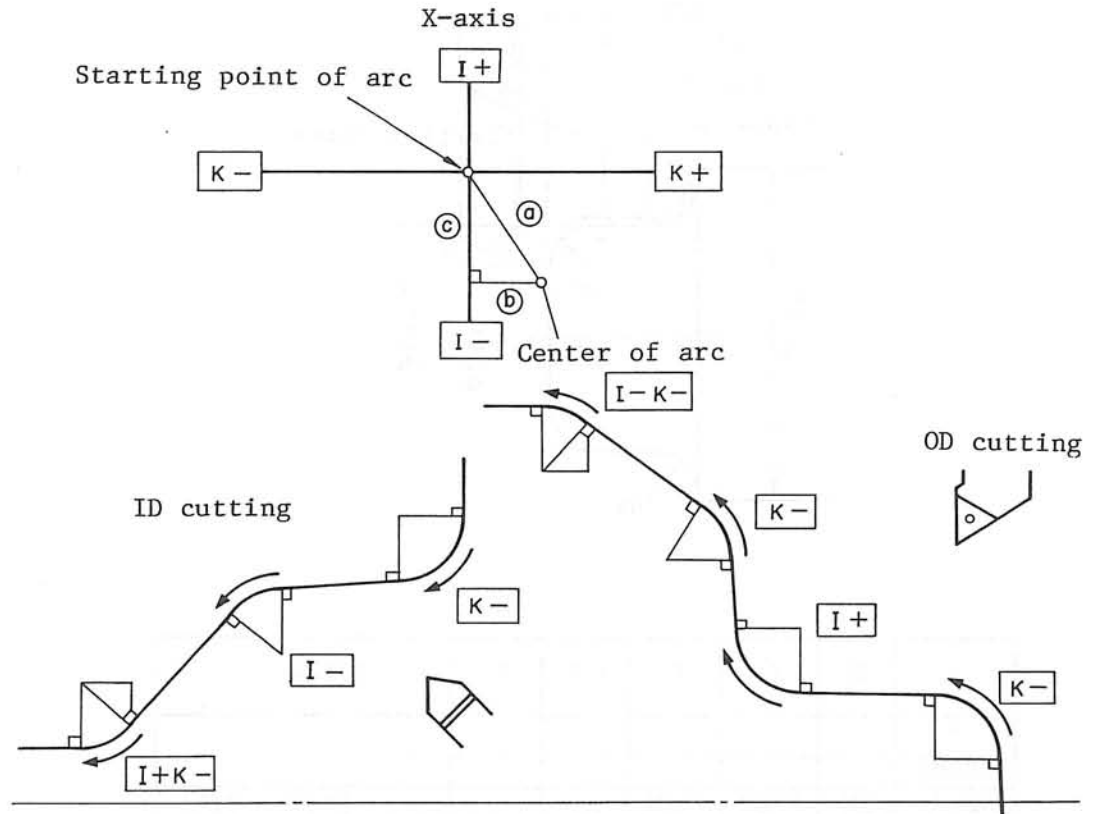
G03 Both I and K values are negative.
(X1, Z1) is the coordinate of arc end point.

Determining Sign and Value of I and K Words:

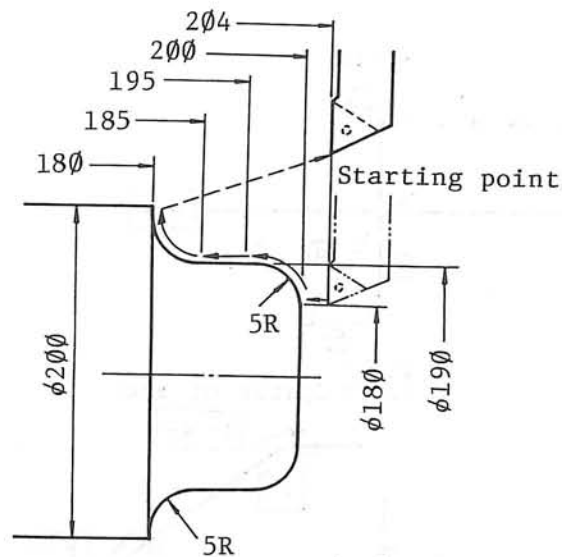
Assume the coordinate system having its origin at the arc start point. Draw a right triangle taking the segment connecting the arc center and arc start point as a hypotenuse. Length of the side (b) parallel to Z-axis is the value of K word and that of side (c) parallel to X-axis is the value of I word.

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.

Refer to the figure provided in the next page.



(2) Example Program



N	G	X	Z	I,K	F	S,T,M
%						
O0100						
N0001	G00	X800	Z204			M42 S200 T0101
N0002		X180				M08 M03
N0003	G01		Z200		F0.2	
N0004	G03	X190	Z195	K-5		
N0005	G01		Z185			
N0006	G02	X200	Z180	I5		
N0007	G00	X800	Z204			M09 M05
N0008						M02

% : % code must be provided at the very beginning of a part program.

O0100: Program name

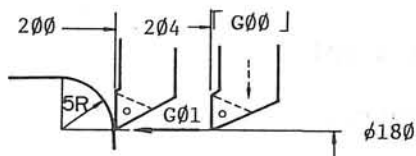
N0001: Commands in N0001 indicate:

- starting point
- spindle speed
- tool number

N0002: X180 moves the cutting tool to X180 position, X coordinate of the starting point of the arc to be cut.

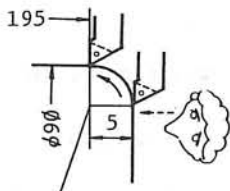
M08 starts coolant supply.

N0003: G01 Z200 feeds the cutting tool to the starting point of the arc to be cut at the specified feedrate.



N0004:

Since the arc is to be cut in the counterclockwise direction, G03 code is provided.



Center of arc

Z coordinate of the arc center is located 5 mm away from the arc starting point in the negative direction: K-5.

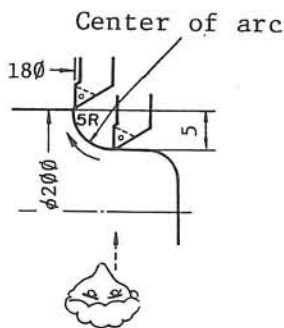
Since X coordinate of the arc center is the same as that of the arc starting point, I word is specified as I0, which may be omitted in the program.

X and Z words are used to specify the coordinates of the end point of the arc.

N0005: Z185 indicates the coordinate of the starting point of the arc.

N0006:

Since the arc is to be cut in the clockwise direction, G02 code is provided.



Center of arc

X coordinate of the arc center is located 5 mm away from the arc starting point in the positive direction: I5.

Since Z coordinate of the arc center is the same as that of the arc starting point, K word is specified as K0, which may be omitted in the program.

X and Z words are used to specify the coordinates of the end point of the arc.

N0007: X800 Z204 returns the cutting tool to the start point.

M05 stops spindle rotation.

M09 stops coolant supply.

N0008: M02 resets the control.

Note 1: When values of I and K words are zero (0), I0 and K0 may be omitted in a part program.

Note 2: In the block containing either G02 or G03 calling for circular interpolation mode,

Either I or K, or both I and K words must be commanded.

Note 3: I and K words must be specified in radius.

Note 4: Even when an arc lies over one quadrant, such arc can be programmed by the commands in one block.

Note 5: When an arc lies within one quadrant, one block may contain only X or Z word.

Note 6: Program to be executed by the tools on turret B may be prepared assuming turret A. (for two-saddle (2S) and two-turret models)

2-5. THREAD CUTTING (STRAIGHT, CONSTANT LEAD THREAD) (G33)

(1) Format

```

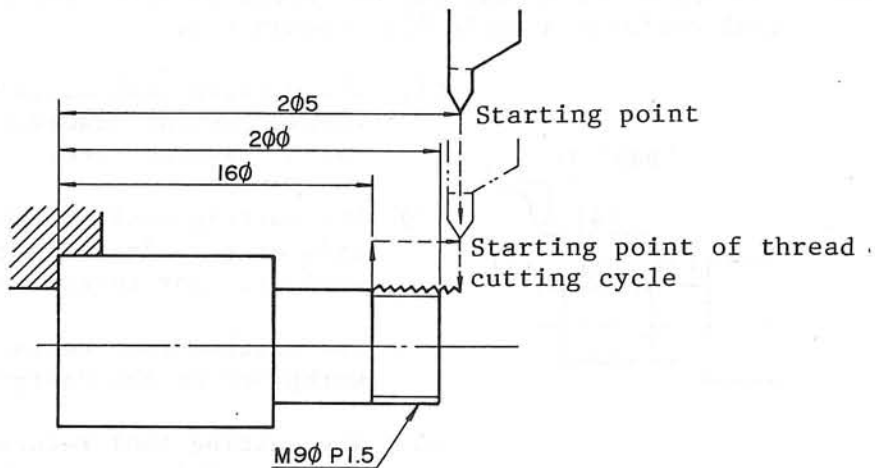
G00 X0000.000 Z0000.000
G33 X0000.000 Z0000.000 Foo.000
     X0000.000
     X0000.000

```

Thread cutting is performed in G33 mode.

X: Diameter of each thread cutting cycle
 Z: End point of thread in longitudinal direction
 F: Thread lead

(2) Example Program



N	G	X	Z	I,K	F	S,T,M
%						
O100						
N001	G00	X800	Z205			M42 S350 T0101
N002		X120				M08 M03
N003	G33	X89	Z160		F1.5	
N004		X88.5				
N005		X88.3				
N006		X88.1				
N007		X88				
N008	G00	X800	Z205			M09 M05
N009						M02

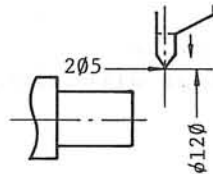
% : % code must be provided at the very beginning of a part program.

O100: Program name

N001: Commands in N001 indicate:

- starting point
- spindle speed
- tool number

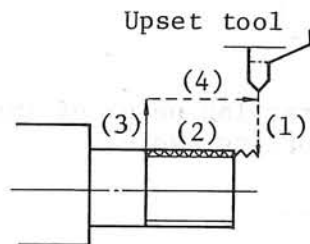
N002:



M03 starts spindle rotation in the normal direction.

X120 indicates the X coordinate of thread cutting cycle starting point.

N003: G33 calls for thread cutting cycle in which the cutting tool performs a cycle (1) through (4).



- (1) The cutting tool moves to the first thread cutting diameter position at a rapid traverse rate.
- (2) The cutting tool is then fed along Z-axis at a feedrate specified by the F word (1.5 for thread lead of 1.5 mm).
- (3) The cutting tool retracts from the workpiece at the designated feedrate.
- (4) The cutting tool returns to the thread cutting cycle starting point at a rapid traverse rate.

Dimension word X indicates the diameter at which the first thread cutting cycle is performed, and Z the end point of the thread. Thread lead is commanded by an F word.

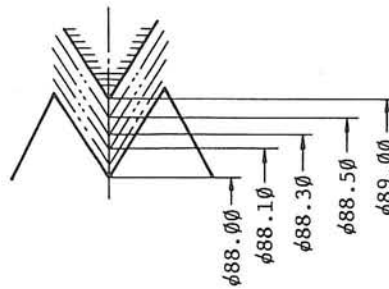
N004: X words in these blocks indicate the diameter of respective
:
N007

N008: G00 X800 Z205 returns the cutting tool to the starting point at a rapid traverse rate.

M05 stops spindle rotation.

N009: M02 resets the control.

X Words in Thread Cutting Cycle



Note 1: Number of infeeds in thread cutting cycle should be selected according to material to be cut, thread lead, etc.

Note 2: NEVER CHANGE SPINDLE SPEED WHILE THREAD CUTTING CYCLE.

Note 3: For chamfering operation, refer to the instruction 3-2.

SECTION 3 APPLICATION PROGRAM

Programming fundamentals detailed in the previous section will be sufficient to program almost all workpieces.

In this section, programming technique taking tool nose radius into consideration and for special thread are detailed.

If you understand thoroughly the instructions in this section, any type of workpieces can be programmed for turning operation.

3-1. PROGRAMMING SPECIAL THREADS

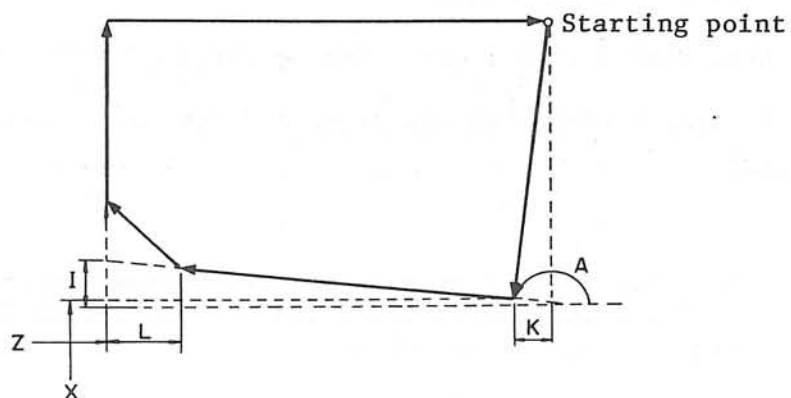
In the previous section, programming for straight, constant lead thread has been explained as G33 thread cutting cycle mode. In addition, the OSP500L-G/OSP5000L-G permits taper thread, thread on end face and variable lead thread can be cut in G32, G33, G34 or G35 mode.

3-1-1. G33 (Fixed Thread Cutting Mode)

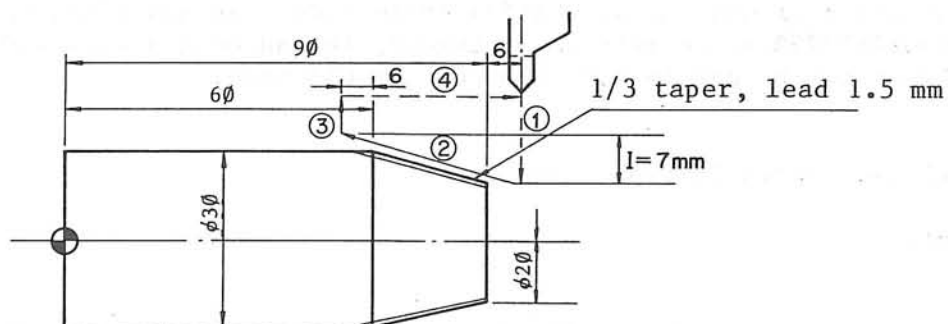
Format:

```
G33 X0000.000 Z0000.000 ( I0000.000 E0.000 ) F0.000 (K, L, J)
      X0000.000
      X0000.000
      :
      :
```

- 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: Lead variation per lead in cutting variable lead thread
 K: 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 M23 mode.
 J: Number of threads within a distance specified by F word
 (When no J word is provided, the control assumes $J=1$.)



Example of Program: Constant Lead Taper Thread



NØØ1	GØØ	X4Ø	Z96
NØØ2	G33	X17	Z54 I7 F1.5
NØØ3		X16.5	
NØØ4		X16.2	
NØØ5		X16.Ø5	

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

NØØ1: Positioning to the thread cutting starting point, $X = 4\phi$ mm (in dia.) and $Z = 96$ mm, at a rapid traverse rate.

NØØ2: With an I word in G33 block taper thread cutting cycle indicated by (1) through (4) is performed.

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.

NØØ3: Program X dimension (in diameter) for each succeeding thread cutting pass.
:
NØØ5

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

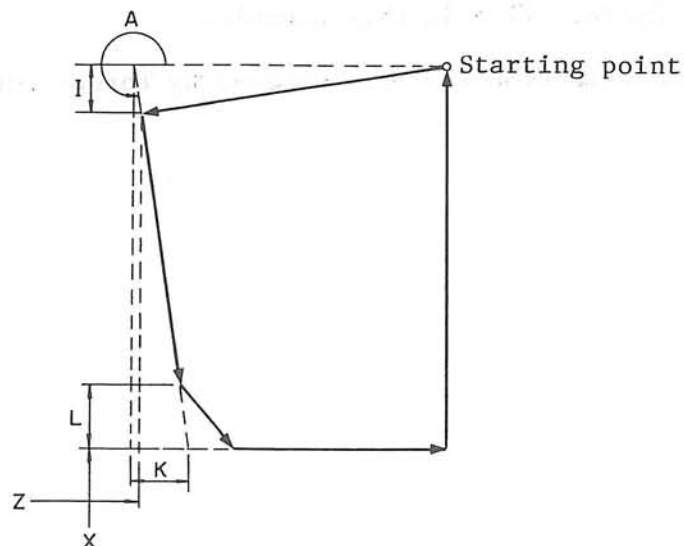
Note 2: Difference of radius between starting point and end point is expressed by an I word.

3-1-2. G32 (Fixed Thread Cutting on End Face Mode)

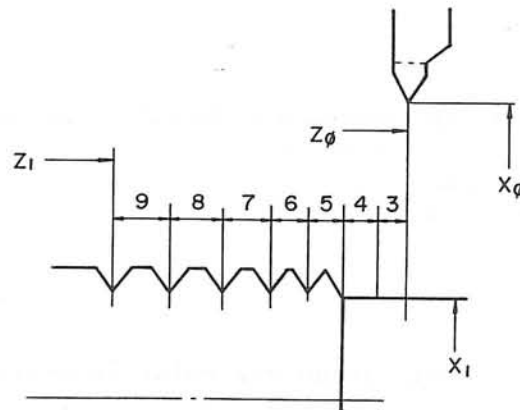
Format:

```
G32 X0000.000 Z0000.000 (K0000.000 E0.000 (I, L, J)) F0.000
      Z0000.000
      Z0000.000
      :
      :
```

- 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 $K=\emptyset$.)
- "+" 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 $E=\emptyset$.)
- I: X-axis shift amount of thread cutting starting point (When no I word is provided, the control assumes $I=\emptyset$.)
- L: Chamfering distance (When no L word is provided, the control assumes $L=\text{one lead at thread cutting starting}$)
- J: Number of threads within a distance specified by F word (When no J word is provided, the control assumes $J=1$.)



Example of Program: Variable Lead Thread



N001	G00	X0	Z0			
N002	G33	X1	Z1	E1	F2.5	
N003		X2				
N004		X3				
:						

N001: Positioning to thread cutting starting point $X0$, $Y0$ at a rapid traverse rate.

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 E1.

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

:

Note 1: The sign of an E word expresses increasing or decreasing lead.

Increasing lead E+

Decreasing lead E-

"+" sign may be omitted.

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

$$D = n \times [F\emptyset + (n \times E)/2]$$

where,

D = displacement after "n" revolutions, mm

n = number of revolutions required for displacement D, rpm

F \emptyset = thread lead at starting thread cutting cycle

E = lead variation amount per revolution

+ = increasing or decreasing lead

"+" increasing lead

"-" decreasing lead

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

$$42 = 7 \times [F\emptyset + (7 \times 1)/2]$$

$$F\emptyset = 2.5$$

3-1-3. G34 and G35 (Non-Fixed Cycle)

Format:

G34(G35) X \emptyset Z \emptyset E F (J)

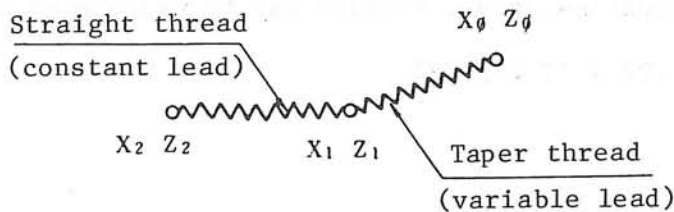
G34 increasing lead

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 \emptyset , Z \emptyset).

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.

Example

```

G00 X0 Z0
G34 X1 Z1 E F
G34 Z2 E0

```

Note 1: The lead of the straight thread is the one at the start point (Z \emptyset) of variable lead thread. When the straight thread is required to be cut with a lead obtained at point Z₁, specify an F word again.

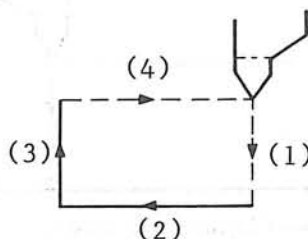
Note 2: To specify the thread lead along the axis parallel to X-axis, command M27. For thread lead along Z-axis, specify M26.

3-2. PRECAUTIONS FOR PROGRAMMING THREAD CUTTING CYCLES

Observe the following points when programming thread cutting cycles:

(1) 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 traverse rate, (2) at the feedrate specified by an F word, and (3) at the rate determined by parameter setting.



(2) 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.

(3) Number of Thread Cutting Passes

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

(4) 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 WHILE THREAD CUTTING CYCLE.

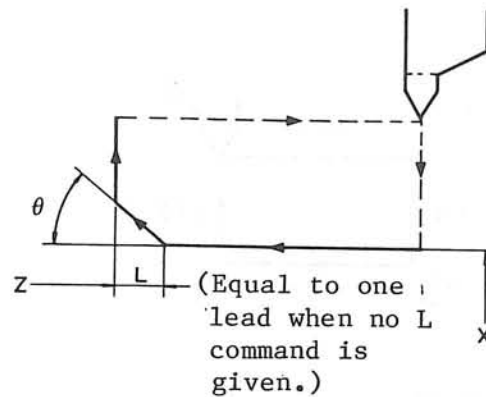
(5) Feedrate Override

The feedrate override dial is inoperative while thread cutting cycle.

(6) 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 of chamfering in the X-axis direction is set in the parameter long (word) No. 8. The set value is calculated by a unit of 12.8 msec.

Feedrate on the X-axis (mm/min)

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

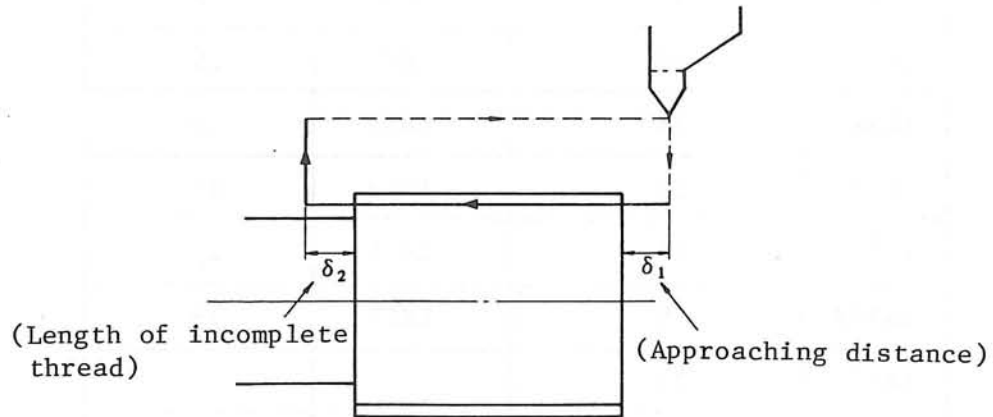
Therefore, the chamfering angle θ is determined by the feedrate in the Z-axis direction (designated in the thread cutting program) and the feedrate in the X-axis.

Example:

N001	G00	X40	Z80	M23
N002	G33	X29	Z50	F1.5
N003		X28.5		
N004		X28.2		
N005		X28.05		
N006	G00	X1	Z1	M22

(7) 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.



Value δ_1 and δ_2 can be obtained from the equations described below.

$$\delta_1 > \frac{N \times P}{KV \times 2\phi}$$

$$\delta_2 > \frac{N \times P}{KV \times 6\phi}$$

where,

N : spindle speed

P : lead

KV: machine model dependent constant

On the OSP500L-G/OSP5000L-G, axis feedrates are controlled according to the follow-up error from the commanded value. The constant used to calculate an axis feedrate from follow-up error is the KV value.

KV values of individual models are indicated below:

Model	KV value (sec/min)	Model	KV value (sec/min)
LC10	35	LB6	35
LC20	30	LB8	35
LC30	30	LB10	35
LC40	25	LB12	30
LC50	25	LB15	25
LS30N	25	LR15	25
LH35	25		
LH55	25		

$\delta 2$ is determined so that actual value (current turret position) will be Z3 when the command value is Z4.

Example 1: With feedrate of 4.8 m/sec., $\delta 2$ is calculated as follows when KV=25.

$$\delta 2 = \frac{4.8 \times 10^3}{25 \times 60} = 3.2 \text{ mm}$$

Example 2: At periphery speed of 1000 m/min., 100 mm diameter and thread lead of 1.5 mm, feedrate and spindle speed are calculated as below.

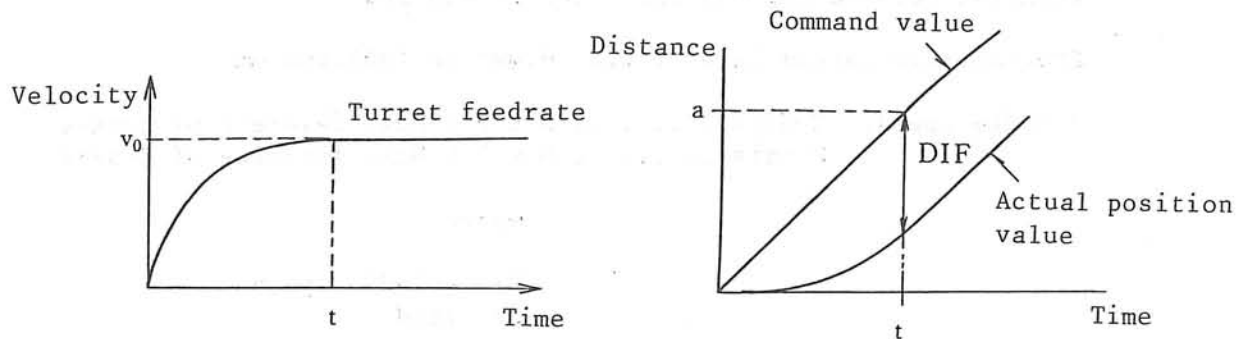
$$N = \frac{1000 \times 10^3}{100 \pi} = 3183 \text{ (rev/min)}$$

$$F = N \times P = 3183 \times 1.5 = 4775 \text{ (mm/min)}$$

If KV=30, $\delta 2$ is calculated as below:

$$\delta 2 = \frac{4775}{30 \times 60} \approx 2.7 \text{ (mm)}$$

For 1, distance to allow the difference (DIF) between the command value and actual position value to be constant is required.



Velocity "v" in relation to time "t" is the function of KV value and time.

Assume that DIF value becomes a constant value at distance "a", δl must be greater than "a".

Turret feedrate is calculated in the following formula.

$$v = v_0 (1 - e^{-kv \cdot t})$$

Value δl at which "v" is 95% of "v₀" is calculated as follows:

$$\delta l = \frac{P \times N}{60} \times t = \frac{P \times N}{60 \times \frac{1}{t}} = \frac{P \times N}{60 \times \frac{1}{3} \times KV} = \frac{P \times N}{20 \times KV}$$

Value δl at which "v" is 98% of "v₀" is calculated as follows:

$$\delta l = \frac{P \times N}{60 \times \frac{1}{4} \times KV} = \frac{P \times N}{15 \times KV}$$

With example 1, δl must be greater than 9.6 mm.

With example 2, δl must be greater than 8.0 mm.

(8) Restrictions on Cutting Speed

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

Programmable thread lead $\emptyset.001$ to 1000.000 mm

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

where,

N: spindle speed

P: lead

Note 1: The same restrictions apply in G01 linear interpolation mode operation.

Note 2: The maximum feedrates vary from model to model.
 Refer to the Operation manual for individual models.

(9) 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, $1 \mu\text{m}$, or use a J word in combination with an F word.

Example: To cut an inch thread of 11 threads per inch

$$25.4/11 \approx 2.309091$$

G34	X0000.000	Z0000.000	F25.4 J11	(1 mm unit system)
G34	X0000.000	Z0000.000	F230.9091	($10 \mu\text{m}$ unit system)
G34	X0000.000	Z0000.000	F2.309091	(1 mm unit system)
G34	X0000.000	Z0000.000	F2309.091	($1 \mu\text{m}$ unit system)

(10) 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.

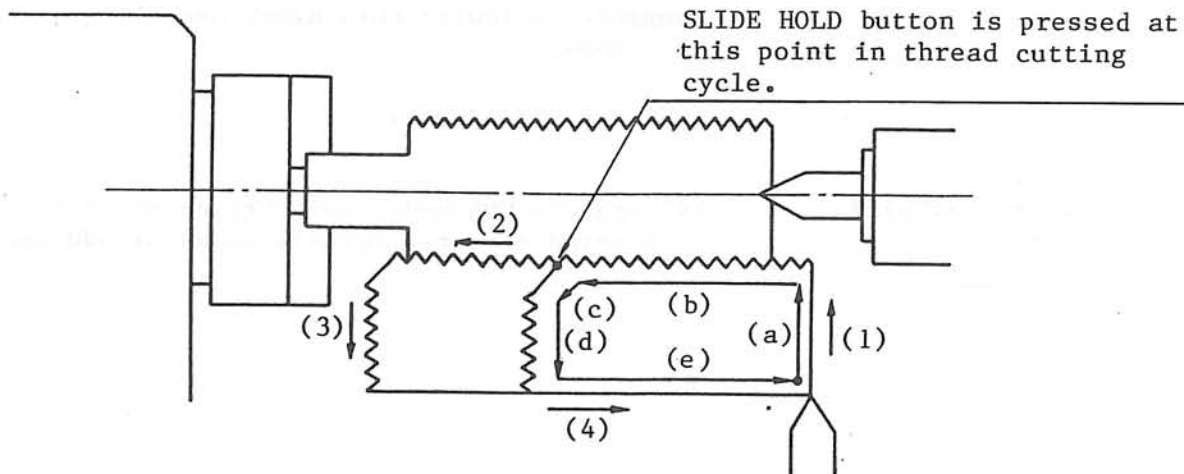
When the SLIDE HOLD pushbutton is pressed during thread cutting cycle:

- 1) Chamfering equivalent to one lead length or length specified by an L command is performed.
- 2) X-axis returns to the thread cutting cycle starting point.
- 3) Z-axis returns to the thread cutting cycle starting point.
- 4) The control is in cycle stop mode waiting for pressing of the CYCLE START button.

When the CYCLE START button is pressed,

- 1) 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)

3-3. PROGRAMMING CONSTANT SPEED CUTTING OPERATION

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.

Format:

G96 S0000

G96 Entry of constant cutting speed mode

S0000 Numerical value in the S word expresses the desired cutting speed.

S1000 means 1000 m/min.

G97 S0000

G97 Cancel of G96

S0000 Numerical value of the S word expresses the desired spindle speed.

When the control is reset, it is in the G97 mode.

Example

N000 G96 S1000

G96 calls for constant speed cutting mode and the commands following this block are all executed in this mode.

S1000 1000 m/min.

N000 G97 S5000

G97 cancels G96 mode, and cutting after this block is carried out at a spindle speed of 5000 rpm.

Precautions for Programming Constant Speed Cutting

- (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 spindle speed to change rapidly, constituting hazardous conditions depending on 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.
- (7) Establishing maximum spindle speed

The allowable maximum spindle speed, restricted due to maximum speed of the chuck, influence of centrifugal force on workpiece gripping force, unbalance in workpiece, etc., can be established in a program with G50 code.

G50 Soooo S word following G50 establishes the allowable maximum spindle speed.

The block establishing the spindle speed restriction cannot contain other commands.

Once established, it is active until another speed is established.

3-4. INCREMENTAL PROGRAMMING

With OSP500L-G/OSP5000L-G, programming is usually prepared in absolute dimensioning system; however, it can accept the commands expressed in incremental dimensioning system. Combined use of absolute and incremental dimension words is also acceptable.

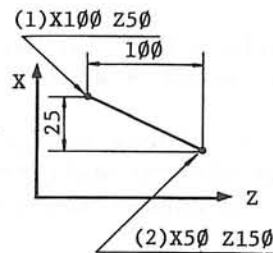
G codes used to select dimensioning system

G90 Absolute programming
(cancel of incremental programming)

When the control is reset, it is in the G90 mode.

G91 Incremental programming

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



Absolute

G00 X50 Z150 (1)
X100 Z50 (2)

Incremental

G00 X50 Z150 (1)
*G91 X50 Z-100 (2)

* Designate dimensional differences between points (2) and (1).

Note: In incremental programming, X word should be expressed in diameter.

3-5. SUPPLEMENT

- (1) S, T and M commands are executed before the dimension words in the same block.

Note that the following M commands are executed after completion of dimension words:

M00, M01, M02 and M30

- (2) "+" sign and leading zeroes may be omitted for the following words:

X, Z, I, K, D, E, F, H, L, U, W, A and B

Example: X+2500 → X2500
Z-0300 → Z-300

- (3) Since G, X, Z and F are all modal, it is unnecessary to repeat the same command in the successive blocks if the contents of these commands do not change.

Example: N002 G01 X100 Z150 F0.3
N003 (G01) (X100) Z100 (F0.3)

Commands in () may be omitted.

Note that following G codes are non-modal:

G04 and G50

Example: G04 F1
[G04] F2 G04 in this block cannot be omitted.

- (4) One block can contain more than one M codes.
(5) Tool Offsets

NC lathes have their own reference points for setting a cutting tool or datum surface for setting a tool holder, which is called "standard tool position" or "preset datum position".

Machining performed on an NC lathe to produce different shapes on workpiece requires various kinds and types of cutting tools and boring bars. It will be virtually impossible to set all cutting tools to their exact datum positions. To solve such difficulty, the NC lathe features "tool offset" function.

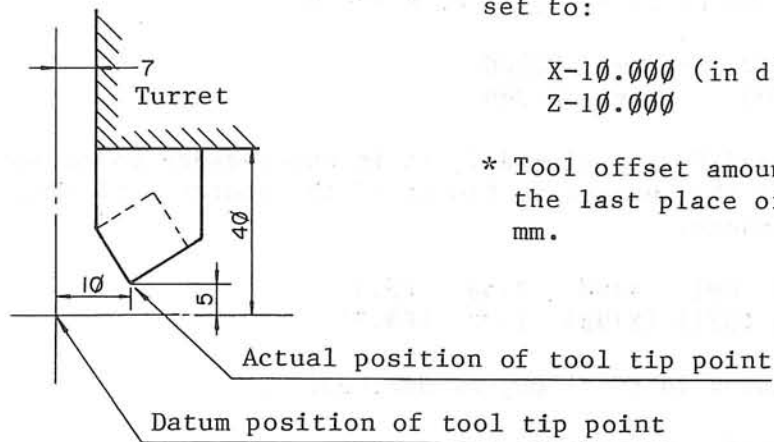
With the tool offset function, tool offset amount between the actually set tool position and the datum position is stored in the NC memory, and machining using such tool is performed with the position adjusted to the correct datum position.

To store tool offset amounts of individual tools, assign the tool offset number for each of them using the MDI switches and then enter the desired amount through them. For details of tool offset amount setting, refer to the Operation Manual.

This feature also permits compensation for tool wear.

See the example below:

Example



Tool offset amounts of the tool left are set to:

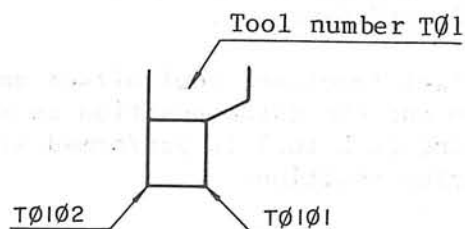
X-10.000 (in diameter)
Z-10.000

* Tool offset amounts must be entered to the last place of input increment, 0.001 mm.

To finish a workpiece $\phi.05$ mm (in diameter) smaller than the programmed diameter using the tool above, enter tool offset amount of X-axis as "-10.050".

The OSP500L-G/OSP5000L-G permits up to 32 pairs of tool offset numbers in one program. For two-saddle/two-turret model, use of 64 pairs, 32 pairs on each turret, of tool offset numbers is allowed.

Effective use of these tool offset numbers can finish workpieces accurately; accurate groove width is assured by using this function, for instance. Assign tool offset number to left and right cutting edge of a grooving tool as shown below can compensate for tool wear and ease dimensional control in grooving.



Range of tool offset amount is as follows:

Metric specification +99999.999 mm
 Inch specification +3937.0078 in.

(6) Block Delete Function

With the BLOCK DELETE switch on the operation panel turned to the ON position, the commands preceded by a slash (/) code are all ignored up to LF (or CR) code in that block.

Example:

```
N010 ... .. LF
/N011 ... .. LF
/N012 ... .. LF
/N013 ... .. LF
N014 ... .. LF
```

BLOCK DELETE switch is ON:

Blocks N011 through N013 are all ignored and execution of commands jumps to block N014 after completion of those in N010.

BLOCK DELETE switch is OFF:

No block is ignored and the control executes the commands normally.

This function can be conveniently used to gauge a part during cutting or cutting workpieces having different stock each other.

(7) Turret Indexing Position

a) LC series

- 1) Turret index is possible when either X-axis or Z-axis is located at the soft-limit in the positive direction.

Select the turret index position taking workpiece shape and tailstock position into consideration to prevent interference.

- 2) Program

X800000 (should be larger than swing)
 Z <arbitrary>

Z800000 (should be greater than Z-axis travel)
 X <arbitrary>

b) LS30-N, LH35-N and LH55-N

The turret can be indexed only at a predetermined position as indicated below:

	X-axis	Z-axis
1) Two-turret model	As desired	As desired
2) H4 (V8) turret model	Variable soft-limit	As desired
3) H6 turret model	As desired	Variable soft-limit

Note:

V8 for LH50-N

- 1) For two-turret model, turret index can be made at any desired position. However, great care must be exercised when indexing the turret so that there is no possibility of colliding the turret mounted tools with workpiece, headstock or tailstock.
- 2) For H4 (or V8) turret, program X1500 mm as turret index position. (It is necessary to program a value larger than the swing of the machine.) Locate the turret at any desired position along Z-axis for turret indexing.
- 3) For H6 turret, program a value larger than the distance between the centers as the turret indexing position on Z-axis.

Program Z2000 mm, for instance, on the machine having the distance between centers of 1000 mm.

For X work, program any desired value which will not cause interference of the turret mounted tools.

Note: For 2) and 3), values larger than axis travel may be programmed. The axis stops at the stroke limit position determined as a soft-limit.

c) LB series

Turret indexing can be performed when X-axis is located at the soft-limit position in the positive direction.

Select the turret index position taking workpiece shape and tailstock position into consideration to prevent interference.

Program:

```
X8000000 (should be larger than swing)
Z <arbitrary>
```

```
Z8000000 (should be greater than Z-axis travel)
X <arbitrary>
```

d) LR series

Turret indexing can be performed when X-axis is located at the soft-limit position in the positive direction.

Select the turret index position taking workpiece shape and tailstock position into consideration to prevent interference.

Program:

```
X8000000 (should be larger than swing)
Z <arbitrary>
```

```
Z8000000 (should be greater than Z-axis travel)
X <arbitrary>
```

- (8) If an illegal character is used in a program, it results in program error and stops machine operation.

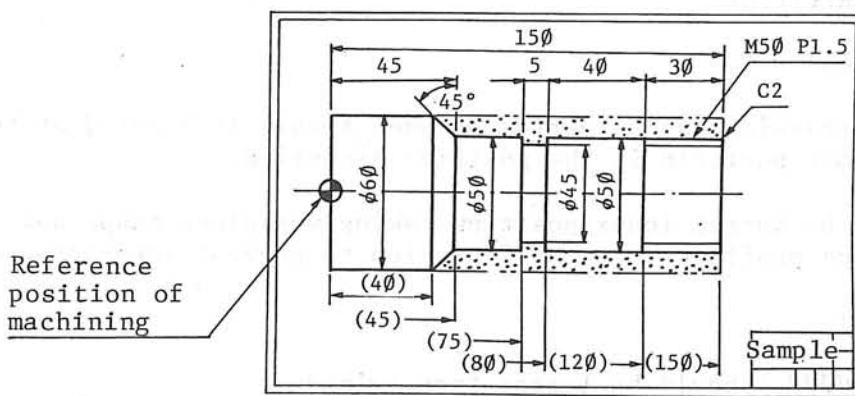
SECTION 4 PROGRAMMING EXAMPLE

This section is prepared to help you review the programming fundamentals detailed in the previous sections.

4-1. PROGRAM EXAMPLE FOR CENTER-WORK

(1) Programming procedure is outlined below:

a) Enter dimensions convenient for programming in the part drawing.



It is recommended to enter the dimensions of a part referenced to the reference position (to be taken as programming zero) as indicated left ... dimensions in ().

b) Determine tooling.

Four types of cutting tools are necessary for finish the sample workpiece.

T0101	T0202	T0303	T0404
Rough Cut	Finish Cut	Grooving	Thread Cutting

c) Determine the cutting conditions.

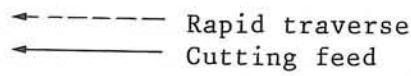
	TØ1Ø1	TØ2Ø2	TØ3Ø3	TØ4Ø4
Depth of Cut	5 (mm)	Ø.25	5	Number of cutting cycles: 4
Feedrate	Ø.4 (mm/rev.)	Ø.2	Ø.1	1.5
Spindle Speed	6ØØ (rpm)	88Ø	6ØØ	6ØØ
Cutting Speed	Cutting speed $\div \frac{3 \times (\text{Diameter to be cut}) \times (\text{Spindle speed})}{1000}$			
(m/min)	8Ø - 12Ø	12Ø - 165	9Ø	85

d) Prepare a program process sheet from the part drawing.

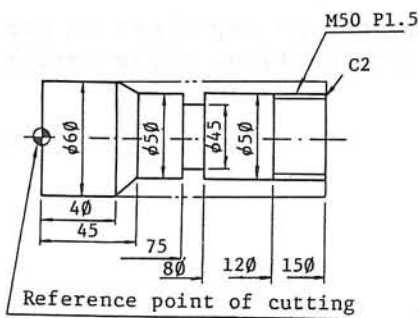
Fill the program process sheet in the manner as explained in previous sections by connecting straight line cutting, taper cutting, and thread cutting commands.

(2) Description of program process sheet for sample workpiece

The sequence of cutting tool motions is expressed by Sequence Numbers.



Contour of Sample Workpiece



<p>OD Roughing</p> <p>Sequence Nos.</p> <p>NO 1 - NO 8</p>	
<p>OD Finishing</p> <p>Sequence Nos.</p> <p>NO 9 - NO 14</p>	
<p>Grooving</p> <p>Sequence Nos.</p> <p>NO 14 - NO 18</p>	
<p>Thread Cutting</p> <p>Sequence Nos.</p> <p>NO 18 - NO 23</p>	<p>Sequence nos. 19 to 22 expresses a thread cutting cycle; number of infeedings in thread cutting is "4".</p>

Be sure to program "X" (end of program) code at the beginning of the program. Tape rewinding stops when this code is read.

Sequence Number
Sequence numbers are used to indicate the order of cutting.

Program the coordinate of the starting point.
Positive stroke end along X-axis

Refer to 3-5, (7) for the details of turret indexing position.

G01 calls for cutting feed mode.

G00 calls for rapid traverse mode.

Blank indicates that the programmed command in the preceding block remains effective. It is not necessary to repeat the same command in the succeeding blocks.

45
40
45

These commands call for taper cutting.

G33 calls for thread cutting mode. The figure shown left indicates the first thread cutting cycle. X and Z words express the end point of the thread cutting.

When depth of cut in the second thread cutting cycle is 0.5 mm in diameter, X value to be programmed is X06.5.

L1135-N

N	G	X	Z	I	K	F	S · T · M	LF
N00100	%							
N00001	G00	X 800	Z 155				S000 (T0101) M43	
N00002		X 50.5					M03 (M08)	
N00003	G01		Z 45			F0.4		
N00004		X 60.5	Z 40					
N00005	G00	X 70	Z 155					
N00006		X 36.5						
N00007	G01	X 50.5	Z 148					
N00008	G00	X 600	Z 155				S800 (T0202)	
N00009		X 36						
N00010	G01	X 50	Z 148			F0.2		
N00011		X 45	Z 45					
N00012		X 60	Z 40					
N00013	G00	X 800	Z 75				S000 T0303	
N00014		X 55						
N00015	G01	X 45				F0.1		
N00016		X 55				F1		
N00017	G00	X 800	Z 155					
N00018		X 70					T0404	
N00019	G33	X 49	Z 120			F1.5		
N00020		X 48.5						
N00021		X 48.15						
N00022		X 48.05						
N00023	G00	X 800	Z 155				M05 (M09)	
N00024							M02	

S000 indicates spindle speed of 600 rpm.

This indicates program name of this program.

This selects tool and tool offset numbers, 01 and 01.

M03 starts the spindle in the forward direction (CW).

M08 starts coolant supply.

M08 or M09 can be programmed in the block containing M03, M04 or M05.

This indicates feedrate of 0.4 mm/rev.

Depth of cut is 4.75 mm in radius.

Note the position of T0202 command; because S, T and M commands are all executed prior to action commands, T0202 is programmed in this block.

S800 indicates the spindle speed of 800 rpm.

One line of a program process sheet is called "one block".

When the desired thread lead is 1.5 mm, program as F1.5.

M05 stops spindle rotation. After that the cutting tool is returned to the starting point at a rapid traverse rate.

Positive stroke end along X-axis.

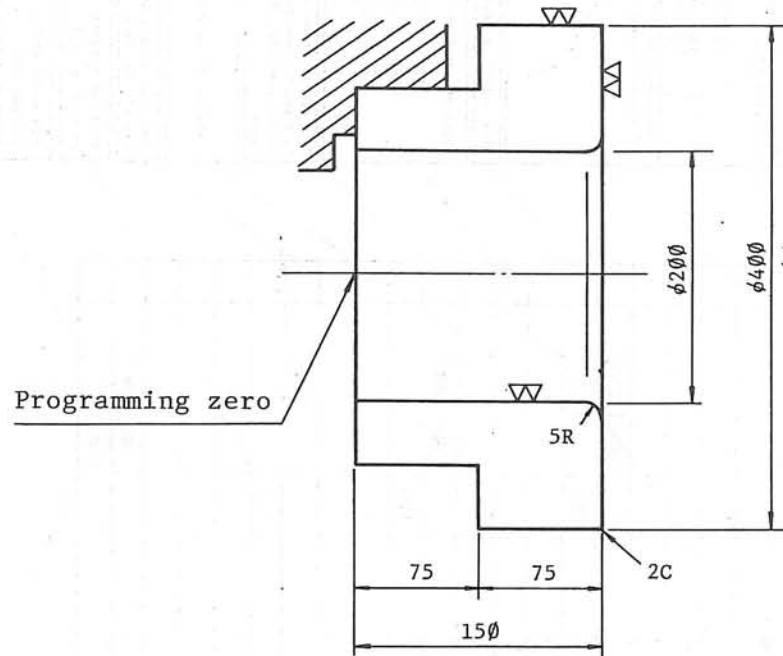
M02 indicates the end of program, of end must be programmed at the end of each program. M02 automatically initiates tape rewinding operation, which terminates when "X" code is read.

M09 stops coolant supply.


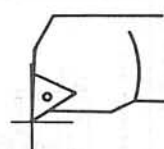
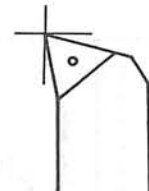
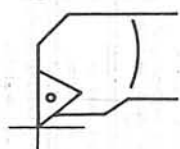
4-2. PROGRAM EXAMPLE FOR CHUCK WORK

(1) Example Workpiece

Material: FC20 (JIS, cast iron)
 Stock : 5 mm (in radius)



(2) Tooling

Turret	H6 Turret			
Tool No.	T0101	T0303	T0505	T0606
Tool Bit				
Cutting Section	OD and End Face ▽	ID ▽	OD and End Face ▽▽	ID ▽▽

(3) Program Process Sheet

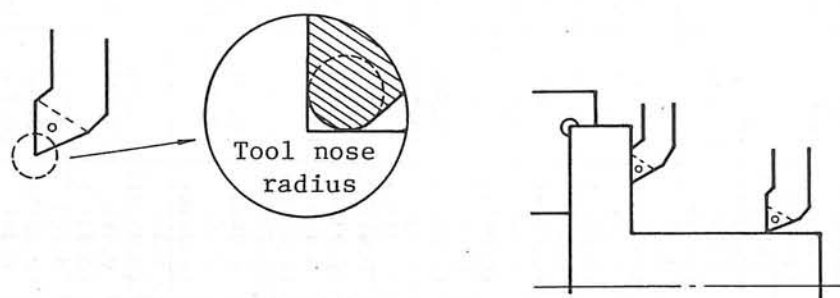
N	G	X	Z	I,K	F	S,T,M	
%							LF
O0100							LF
N0001	G00	X800	Z500			S120 T0101 M42	LF
N0002		X420	Z200				LF
N0003			Z150.1			M03	LF
N0004	G01	X180			F0.5		LF
N0005	G00	X392.5	Z152				LF
N0006	G01	X400.5	Z148				LF
N0007			Z72		F0.6		LF
N0008	G00	X800	Z500				LF
N0009		X199.5	Z153			S200 T0303	LF
N0010	G01		Z-3		F0.4		LF
N0011	G00	X190	Z153				LF
N0012		X209.5					LF
N0013	G01		Z150.1				LF
N0014	G02	X199.5	Z145.1	K-5	F0.3		LF
N0015	G00	X190	Z153				LF
N0016		X800	Z500			M05	LF
N0017		X405	Z200			T0505 M43	LF
N0018	G96		Z150			S200 M03	LF
N0019	G01	X200			F0.2		LF
N0020	G00	X392	Z152				LF
N0021	G01	X400	Z148				LF
N0022			Z72				LF
N0023	G00	X800	Z500				LF
N0024		X210	Z153			S150 T0707	LF
N0025	G01		Z150		F0.15		LF
N0026	G02	X200	Z145	K-5			LF
N0027	G01		Z-3				LF
N0028	G00	X190	Z153				LF
N0029		X800	Z500			M05	LF
N0030						M02	LF

Explanation of Program Process Sheet

- % : % code must be provided at the very beginning of a program.
- O0100: Program name
- N0001: Select spindle speed 120 rpm and tool number 01.
- N0003: Spindle starts in the forward direction.
- N0004: End face cutting at 0.5 mm/rev.
- N0006: Chamfering
- N0007: OD turning at 0.6 mm/rev.
- N0008: Turret returns to the index position.
- N0009: Select spindle speed 200 rpm and tool number 03.
Positioning to X199.5, Z153
- N0010: ID turning at 0.4 mm/rev.
- N0014: Arc cutting at 0.3 mm/rev.
- N0016: Turret returns to the index position after spindle stops.
- N0017: Change spindle speed range and select tool number 05.
- N0018: Enter into constant cutting speed mode; spindle speed
corresponding to cutting speed 200 m/min. Spindle starts in the
forward direction.
- N0019: OD and end face finishing at 0.2 mm/rev.
:
N0022
- N0023: Turret returns to the index position.
- N0024: Select cutting speed 150 m/min. and tool number 06.
- N0025: ID finishing at 0.15 mm/rev.
:
N0027
- N0029: Turret returns to the index position after spindle stops.
- N0030: Reset the control.

SECTION 5 PROGRAMMING TAKING TOOL NOSE RADIUS INTO CONSIDERATION

Cutting edge of the tools used for turning operation is not always sharp but is slightly rounded.



When cutting a straight line along either X- or Z-axis, as in turning outside diameters and end faces of workpieces, they may be cut to the programmed dimensions. However, when slope or arc is cut the workpiece may be oversized or undersized because of a tool nose radius which causes the actual cutting paths to be offset from the programmed paths.

The tool nose radius affects the machining accuracy when circular arc or taper is cut. Thus, it is necessary to compensate for the value of the nose radius to obtain the exact geometry of the profile.

Taper Cutting and Chamfering

For taper cutting or chamfering, the profile obtained by the programming without compensation for the tool nose radius does not give the exact dimensions:

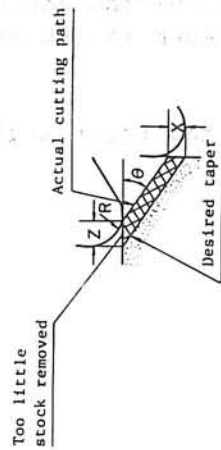
TO OBTAIN THE EXACT DIMENSIONS IN TAPER CUTTING AND CHAMFERING, ADD THE AMOUNTS OF COMPENSATION LISTED IN THE TABLE IN THE NEXT PAGE.

Compensation amount for tool nose radius R and taper angle θ is calculated in the following equations:

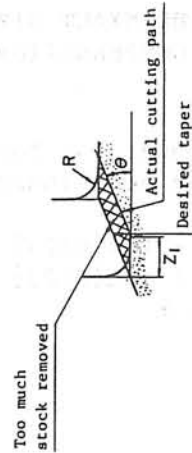
$$\begin{aligned} Z &= R \times [1 - \tan(\theta/2)] \\ Z1 &= R \times [1 + \tan(\theta/2)] \\ X &= Z \times \tan \theta \end{aligned}$$

θ (degrees)	Nose Radius: $\phi.4$ mm ($\phi.016$ in.)		Nose Radius: $\phi.8$ mm ($\phi.032$ in.)	
	[2X] X	Z	[2X] X	Z
5°	$\phi.066$ [($\phi.0026$)]	$\phi.383$ ($\phi.0151$)	$\phi.132$ [($\phi.0053$)]	$\phi.766$ ($\phi.0306$)
	$\phi.033$ ($\phi.0013$)		$\phi.066$ ($\phi.0026$)	
7.5	$\phi.098$ [($\phi.0039$)]	$\phi.374$ ($\phi.0150$)	$\phi.196$ [($\phi.0078$)]	$\phi.748$ ($\phi.0300$)
	$\phi.049$ ($\phi.0020$)		$\phi.098$ ($\phi.0039$)	
10	$\phi.129$ [($\phi.0052$)]	$\phi.365$ ($\phi.0146$)	$\phi.258$ [($\phi.0103$)]	$\phi.730$ ($\phi.0292$)
	$\phi.064$ ($\phi.0026$)		$\phi.129$ ($\phi.0052$)	
15	$\phi.186$ [($\phi.0074$)]	$\phi.347$ ($\phi.0139$)	$\phi.372$ [($\phi.0149$)]	$\phi.694$ ($\phi.0278$)
	$\phi.093$ ($\phi.0037$)		$\phi.186$ ($\phi.0074$)	
30	$\phi.338$ [($\phi.0135$)]	$\phi.293$ ($\phi.0117$)	$\phi.676$ [($\phi.0270$)]	$\phi.586$ ($\phi.0234$)
	$\phi.169$ ($\phi.0068$)		$\phi.338$ ($\phi.0135$)	
45	$\phi.468$ [($\phi.0187$)]	$\phi.234$ ($\phi.0094$)	$\phi.936$ [($\phi.0374$)]	$\phi.468$ ($\phi.0187$)
	$\phi.234$ ($\phi.0094$)		$\phi.468$ ($\phi.0187$)	
60	$\phi.586$ [($\phi.0234$)]	$\phi.169$ ($\phi.0066$)	1.172 [($\phi.0468$)]	$\phi.338$ ($\phi.0135$)
	$\phi.293$ ($\phi.0117$)		$\phi.586$ ($\phi.0234$)	
75	$\phi.694$ [($\phi.0278$)]	$\phi.093$ ($\phi.0037$)	1.388 [($\phi.0555$)]	$\phi.186$ ($\phi.0073$)
	$\phi.347$ ($\phi.0139$)		$\phi.694$ ($\phi.0277$)	
7° 7' 30"	$\phi.094$ [($\phi.0038$)]	$\phi.375$ ($\phi.0150$)	$\phi.188$ [($\phi.0075$)]	$\phi.750$ ($\phi.0300$)
	$\phi.047$ ($\phi.0019$)		$\phi.094$ ($\phi.0038$)	
(1/10 taper) 2° 52' 20"	$\phi.039$ [($\phi.0016$)]	$\phi.390$ ($\phi.0156$)	$\phi.078$ [($\phi.0031$)]	$\phi.730$ ($\phi.0312$)
	$\phi.020$ ($\phi.0008$)		$\phi.039$ ($\phi.0016$)	
Milling taper 8° 17' 8"	$\phi.108$ [($\phi.0043$)]	$\phi.371$ ($\phi.0148$)	$\phi.216$ [($\phi.0086$)]	$\phi.742$ ($\phi.0297$)
	$\phi.054$ ($\phi.0022$)		$\phi.108$ ($\phi.0043$)	

Taper with Increasing Diameters
(Upward Taper)



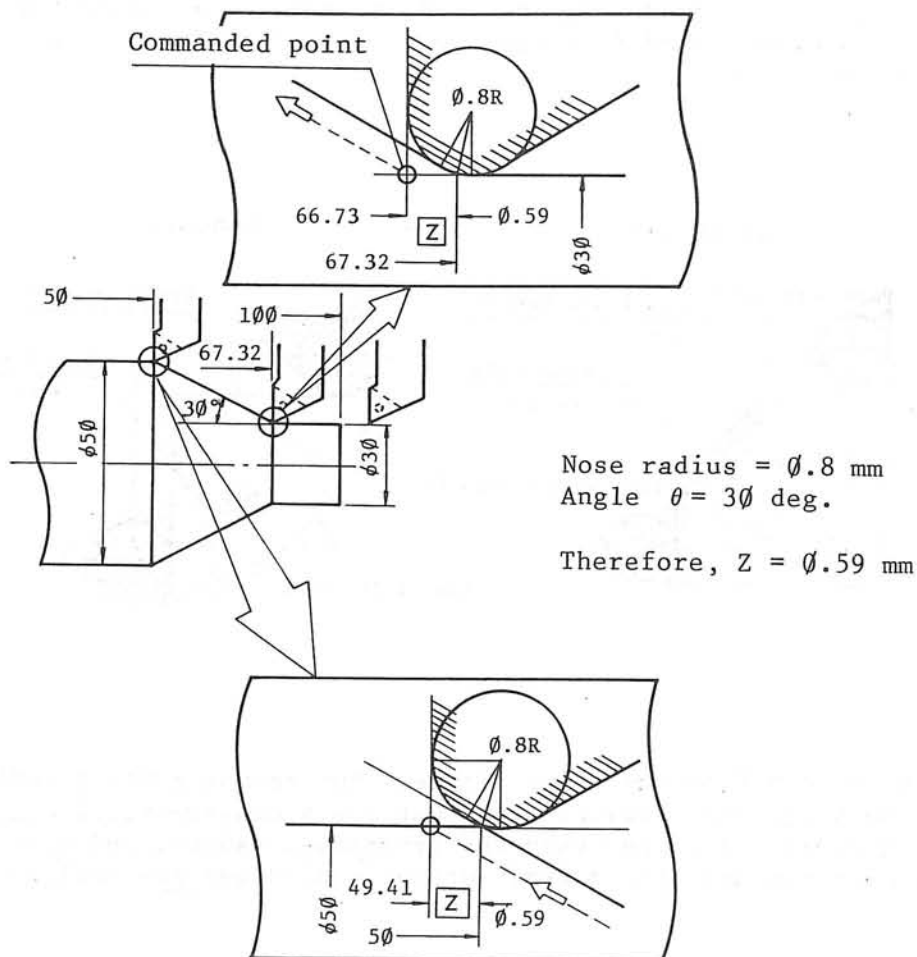
Taper with Decreasing Diameters
(Downward Taper)



*[2X] = Diametral amount

5-1. PROGRAMMING EXAMPLE FOR TAPER CUTTING

The compensation amount for Z value (Z offset dimension) can be found in the table.

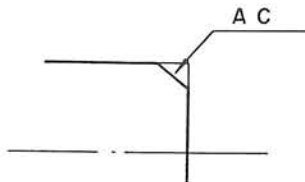


A mm x 45 deg. chamfering:

Chamfering cuts that do not require accurate finish may be programmed as follows.

Command as "A + $\phi 0.25$ " mm when tool nose radius is $\phi 0.4$ mm, or "A + $\phi 0.5$ " mm when tool nose radius is $\phi 0.8$ mm.

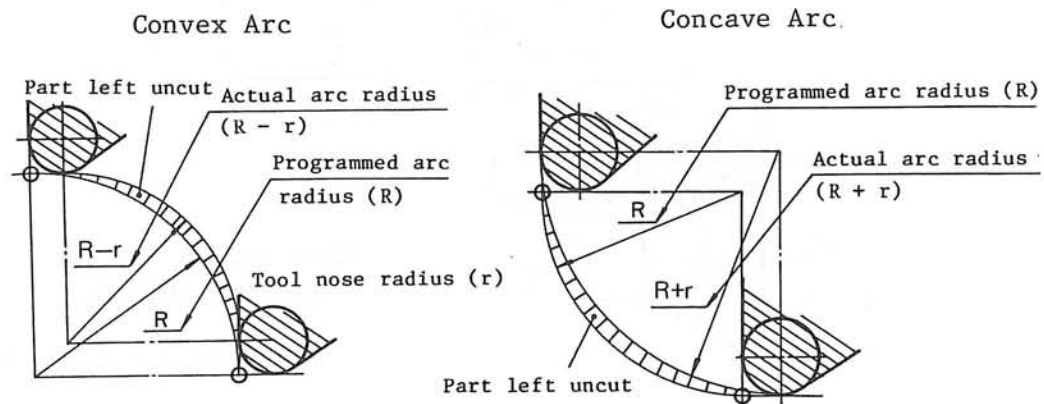
For example, a chamfer of 1 mm may be cut with a tool of $\phi 0.4$ mm radius by programming X and Z dimensions as 1.25 mm.



5-2. PROGRAMMING EXAMPLE FOR ARC CUTTING

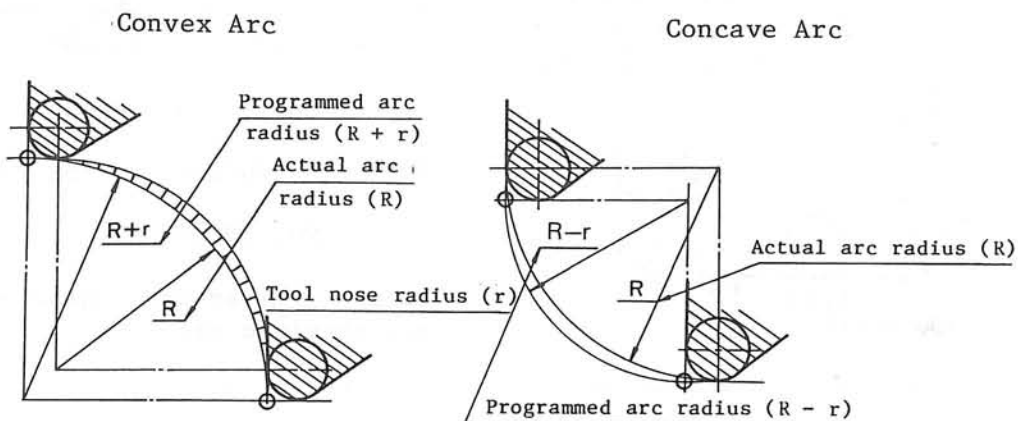
The following deals with arc cuts where an arc travels a full 90° deg. for simplicity of the specifications of the coordinates as well as for the brevity of explanation.

If the programming is made without proper offsets for a tool nose radius in cutting an arc, the arc required and the actual cut would NOT be the same, and the material (shaded in the figure below) would be undercut leaving portion to be cut.



As can be seen from the above figures, the actual cutting radius differs from the programmed radius. When cutting a convex-shaped arc, the actual cutting radius is larger than the programmed radius, and when cutting a concave-shaped arc, the actual cutting radius becomes smaller.

To cut the arc exactly as specified in the engineering drawing, it is necessary to add or subtract the size of tool nose radius to or from the actual radius of the arc to be cut so that the tool may follow the desired path as shown below.

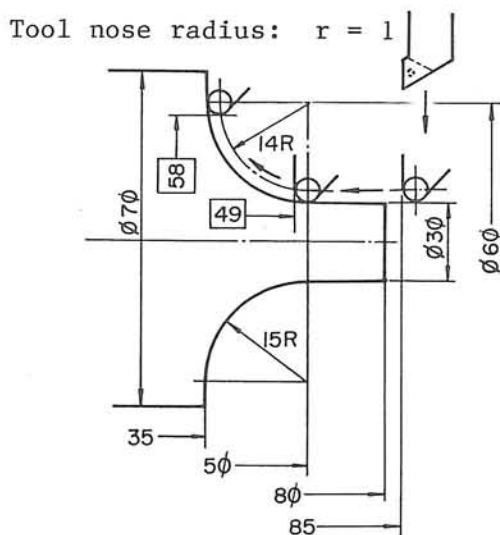


Convex Arc Program $(R + r)$ for radius of arc, R
 Concave Arc Program $(R - r)$ for radius of arc, R

r : tool nose radius

When cutting a 90° convex arc, the arc to be programmed should be the actual arc plus the size of the tool nose radius, r . When cutting a 90° concave arc, the arc to be programmed should be the actual arc minus the size of the tool nose radius, r .

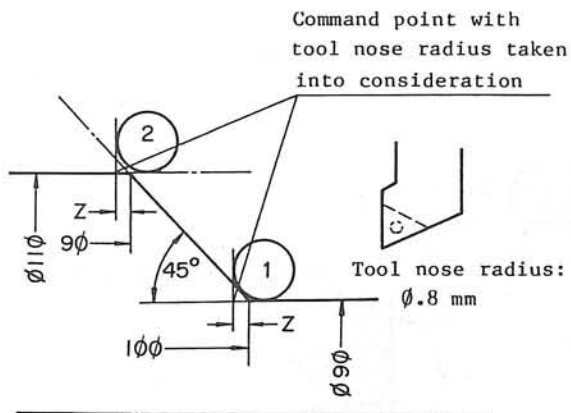
Example:



N	G	X	Z	I	K	F
%						
O100						
N001	G13					
N002	G00	X800	Z400			
N003		X40	Z85			
N004		X30				
N005	G01		Z49			F0.2
N006	G02	X58	Z35	I14		
N007	G01	X70				

5-3. PROGRAMMING EXAMPLES WITH TOOL NOSE RADIUS COMPENSATION

Example 1: Taper Cutting



The tool offset dimension Z with the tool nose radius taken into account may be obtained from the table:

$$Z = 0.468 \approx 0.47$$

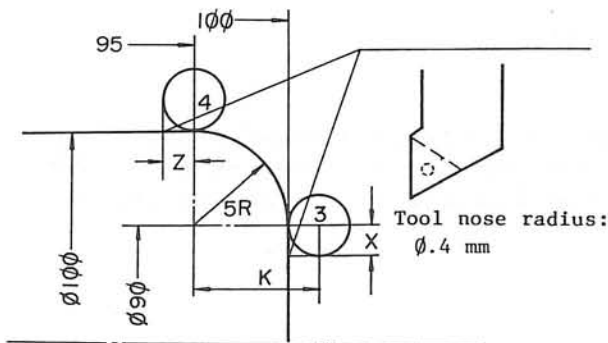
Thus, the coordinates of point 1 are;

$$(X90, Z99.53)$$

In the similar way, those of point 2 are obtained as;

$$(X110, Z89.53)$$

Example 2: Arc Cutting



The tool offset amount X with a tool nose radius r , 0.4 mm, taken into account at point 3 is equal to the tool nose radius.

Thus, the coordinates of point 3 are;

$$(X89.2, Z100)$$

In the similar way, those of point 4 are;

$$(X100, Z94.6)$$

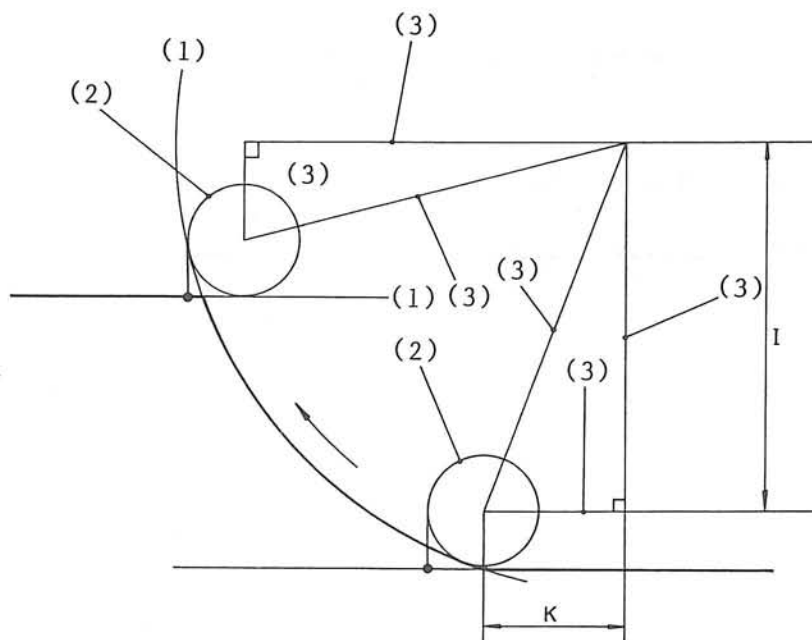
The arc cut program for example 2 is:

```
G01 X89.2 Z100 F0.3
G03 X100 Z94.6 F-5.4
```

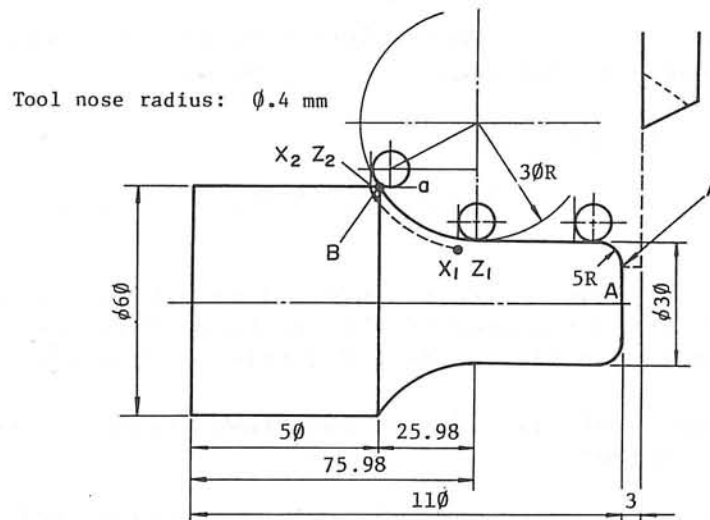
5-4. GENERAL PROGRAMMING PROCEDURE FOR CIRCULAR INTERPOLATION

To obtain the command values in circular interpolation programming, follow the steps below to assure correctness of the commands.

- (1) Extend arcs and straight lines.
- (2) Draw circles tangent to two lines, with a radius equivalent to tool nose radius.
- (3) Draw right-angled triangle with its hypotenuse between the center of the circle drawn and the center of the arc to be cut. Two sides of the triangle should parallel to X- and Z-axis, respectively.
- (4) Calculate the length of all sides of the drawn triangle with the trigonometric technique.
- (5) With the values obtained in step (4), calculate the coordinate values of the circles.
- (6) Find the coordinate values of starting point and end point of the arc to be cut taking the tool nose radius into account.
- (7) Dimension words I and K are determined referenced to the center of the tool nose circle with respect to arc starting point.



Example:



$$X1 = 30.000$$

$$Z1 = 75.98 - 0.4 = 75.58$$

$$X2 = 60.000$$

$$a = \sqrt{(30 - 0.4)^2 - 14.6^2} = 25.75$$

$$Z2 = 75.98 - a - 0.4$$

$$= 49.83$$

Program to cut from point A to point B:

```
G00 X19.2
G01 Z110
G03 X30 Z104.6 K-5.4
G01 Z75.58
G02 X60 Z49.83 I29.6
```

SECTION 6 PROGRAMMING FOR SIMULTANEOUS 4-AXIS CUTS (2S Model for LC)

This section explains how to program simultaneous 4-axis cut operations using both turrets A and B simultaneously.

The simultaneous 4-axis cuts will double the machining efficiency if programmed correctly.

6-1. PROGRAMMING FORMAT

```

N0000 G13
N0001 G00 X00000 Z00000 P0000 S0000 T0000 M00
      :
      :   Cutting program for turret A
      :
N0049
N0050 G14
N0051 G00 X00000 Z00000 P0000 S0000 T0000 M00
      :
      :   Cutting program for turret B
      :
N0099

N0100 G13
N0101 G00 X00000 Z00000 P0000 S0000 T0000 M00
      :
      :   Cutting program for turret A
      :
N0150 G14
N0151 G00 X00000 Z00000 P0000 S0000 T0000 M00
      :
      :   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 .....
:
N0150 G01 X4 Z4 P30 F0.3
:
:
N0200 G00 X800 Z800
N0201 P40 M05

```

P10 in N0001 and in N0101 synchronizes execution of M41 S200 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

6-2. 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.

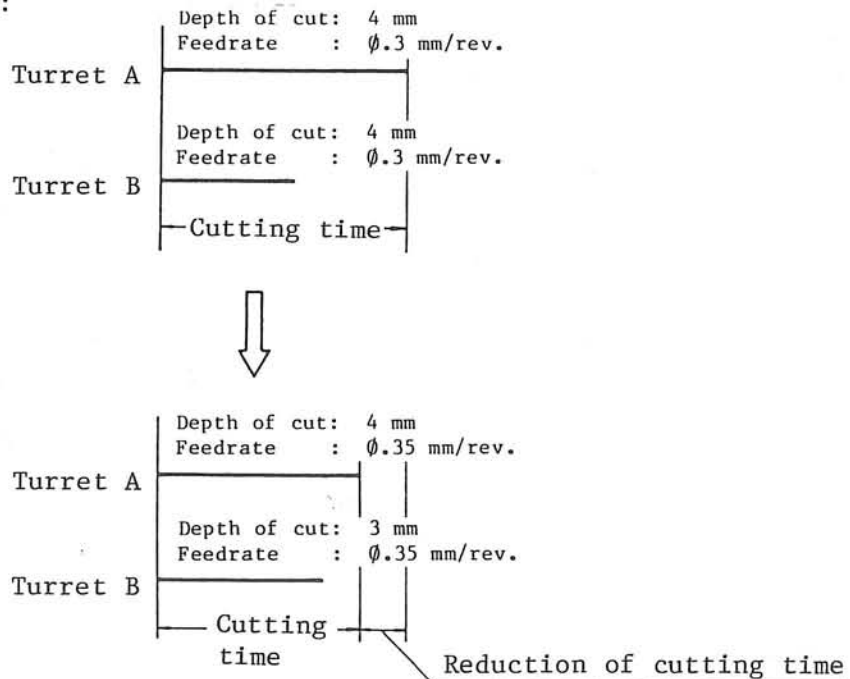
When programming simultaneous 4-axis cuts, observe the following points carefully:

- (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) Other remarks

- 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
:

```

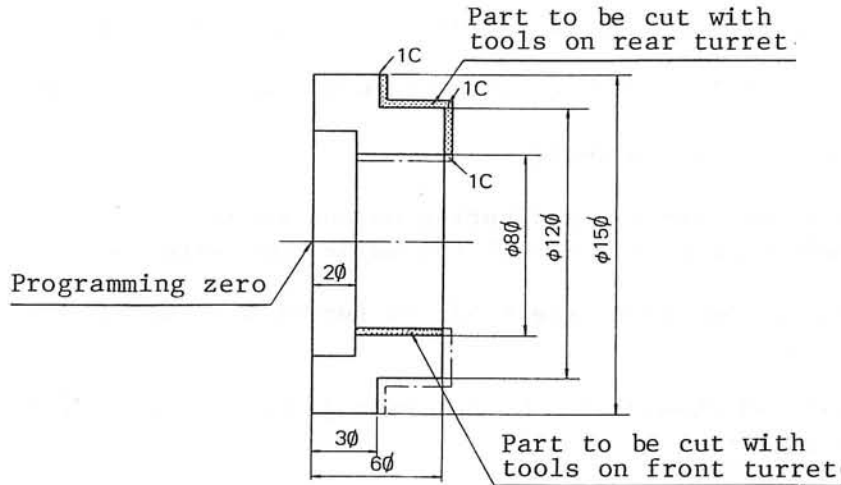
For turrets A and B, G96, S, M and P commands must match.

Even in constant speed cutting mode, it is active only on turret A and turret B is not in such mode.

6-3. PROGRAMMING EXAMPLE

6-3-1. Workpiece Dimensions

Material: S45C (JIS, carbon steel)
 Stock : 3 mm (in radius)



6-3-2. Tooling and Cutting Conditions

Machine: LC4φ-2S

Turret	Tool No.	Cutting Tool	Cutting Conditions	Cutting Time
A	Tφ1φ1	Upset tool Facing	Cutting speed: 12φ to 65 m/min. Depth of cut: 3 mm Feedrate: φ.35 mm/rev.	 4φ sec.
	Tφ2φ2	Upset tool OD turning	Cutting speed: 95 m/min. Depth of cut: 3 mm Feedrate: φ.4 mm/rev.	 28 sec. 68 sec.
B	Tφ1φ1	Normal tool ID turning	Cutting speed: 65 m/min. Depth of cut: 3 mm Feedrate: φ.25 mm/rev.	 63 sec.

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.

6-3-3. 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

:

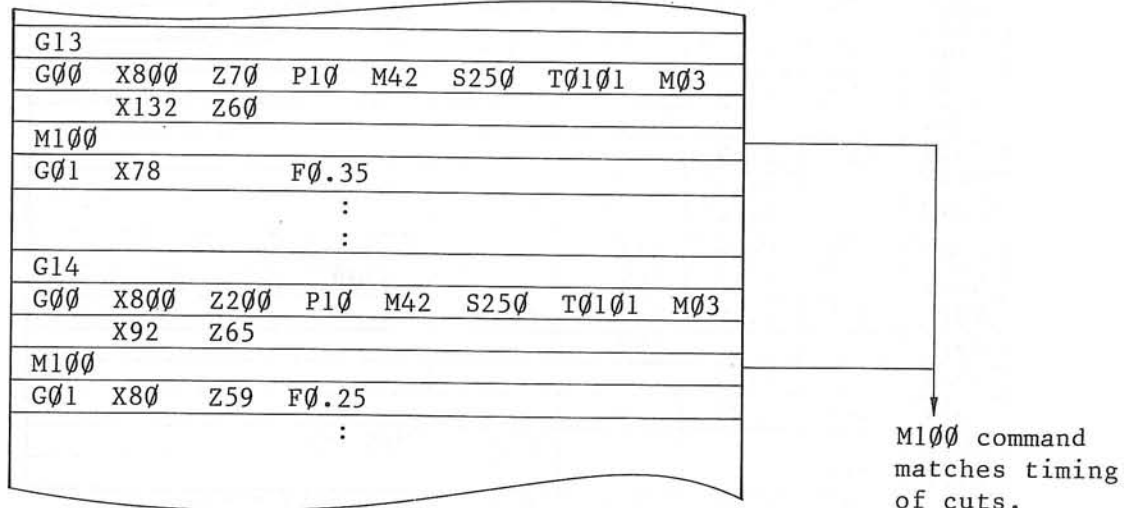
N106

With the program above, simultaneous cut on end face and OD turning by turret A and ID turning by turret B is performed.

6-4. 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:



Points to consider when Using the M100 Command:

The following points should be considered when synchronizing operations with the M100 command.

- (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.

* Nose R compensation (optional): See Optional Specifications Instruction Manual No. 1.

- (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.

SECTION 7 BARRIER CHECK FUNCTION

This safety feature is incorporated into the OSP5000L-G CNC System as one of the self-diagnostic functions.

7-1. GENERAL DESCRIPTION

With the advance of automation in parallel with wider use of numerical control for turning machines, rapid traverse, cutting speeds and spindle speeds are all stepped up to attain a reduction in cutting time. With their performance capabilities upgraded, however, there is an increasing possibility of every more serious accidents caused by tape program errors, operating failures and other faulty procedures.

The BARRIER CHECK Function has been developed specially for use with the OSP5000L-G/OSP5000L-G controlled turning machine applications to provide a reliable means of averting such trouble.

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 on a command tape or those entered through MDI switches. If a tool is commanded to move into the barrier, the control automatically detects this alarm condition and stops the machine operation.

For the two-saddle model, barrier is set for both A and B turret tools, to detect entry of tools into the set barrier.

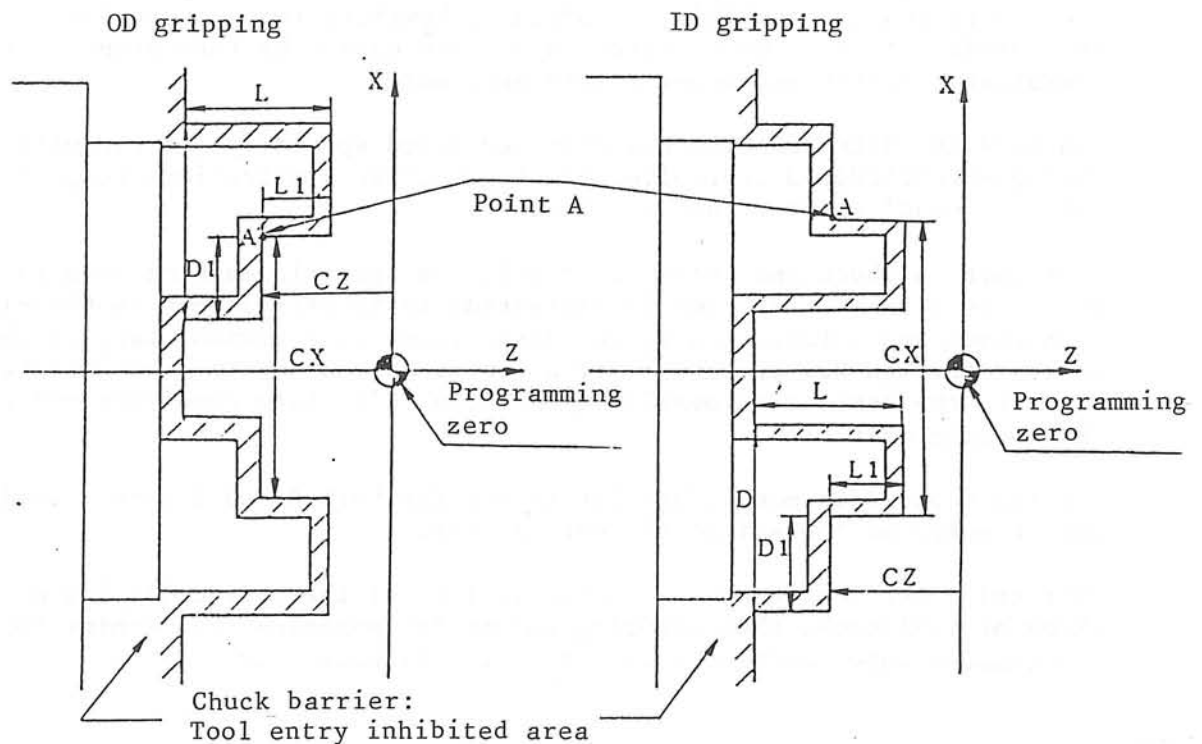
With this function, there is no possibility of the tool colliding with the chuck or tailstock, thus assuring safety in operating the machine to check the command tape, cutting a workpiece in MDI mode, etc.

7-2. CHUCK BARRIER AND TAILSTOCK BARRIER

7-2-1. 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.

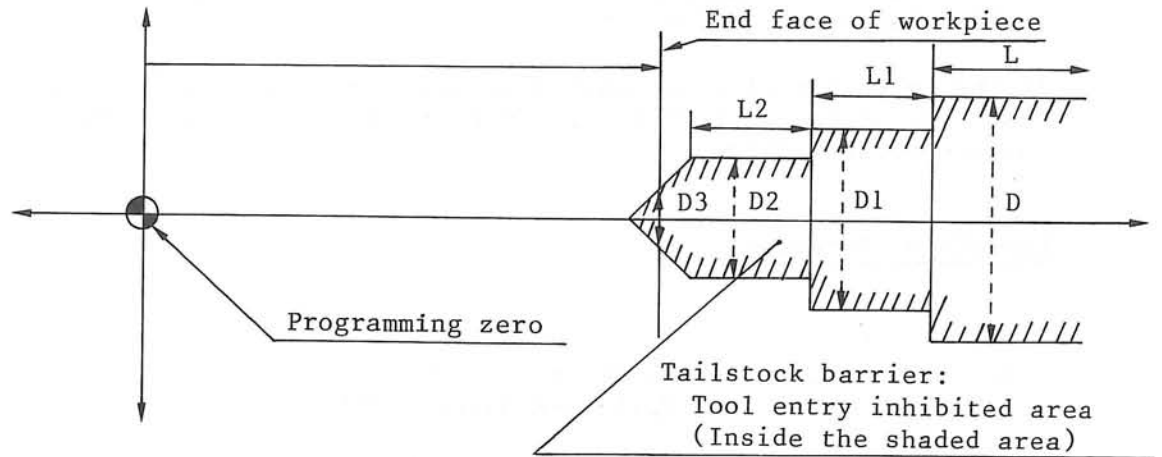


Symbol	Description	Method
L	Chuck jaw length	Optional parameter (long word) No. 18
D	Chuck jaw size	Optional parameter (long word) No. 19
L1	Gripping length of chuck jaw	Chuck/tailstock barrier
D1	Chuck jaw gripping face width	Chuck/tailstock barrier
CX	Chuck gripping diameter	Chuck/tailstock barrier
CZ	Distance from programming zero	Chuck/tailstock barrier

For details of procedure to establish the chuck barrier, refer to 4-3-1, (2) of Operation Manual.

7-2-2. 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	Optional parameter (long word) No. 20
D	Tailstock spindle diameter	Optional parameter (long word) No. 21
L1	Tailstock spindle length (1)	Optional parameter (long word) No. 22
D1	Tailstock spindle diameter (1)	Optional parameter (long word) No. 23
L2	Tailstock spindle length (2)	Chuck/tailstock barrier
D2	Tailstock spindle diameter (2)	Chuck/tailstock barrier
D3	Tailstock spindle center diameter	Chuck/tailstock barrier
WR	Tailstock spindle position (Z)	Chuck/tailstock barrier

For details of procedure to establish the tailstock barrier, refer to 4-3-1, (3) of Operation Manual.

7-2-3. 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
M2Ø   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:

```

:
:
N000   M25   Chuck barrier ON
:      (M21) (Tailstock barrier ON)
:
N000   M24   Chuck barrier OFF
:      (M2Ø) (Tailstock barrier OFF)
:

```

The barrier function is active for the blocks of commands from the block containing M25 (M21) to the one containing M24 (M2Ø).

7-2-4. Precautions

- (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 M2Ø 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 if M25 and M21 are effective.
- (3) The barrier is renewed when new barrier setting data is entered.
- (4) When the Control tape No. 3 is loaded, it automatically clears the barrier data. Be sure to set the point data establishing the chuck barrier after loading the control tape No. 3.

7-3. TOOL INTERFERENCE CHECK

With the two-saddle model of CNC lathes, there is a possibility of tools on A-turret and B-turret to interfere each other since these turrets operate independently. Interference between the tools is checked using the tool interference area set for individual tools and if an interference is detected, an alarm occurs (level A) and the machine is stopped.

7-3-1. Programming

[M28 Tool interference check OFF
 [M29 Tool interference check ON

```

%
O1000
:
:
N0100 G13
N0101 G00 X500 Z500 P10 M42 S250 T0101 M03
      M29
:
:
      M28
N0200 G14
N0201 G00 X400 Z400 P10 M42 S250 T0101 M03
      M29
:
:
N0400 M28
:
N0500 M02
  
```

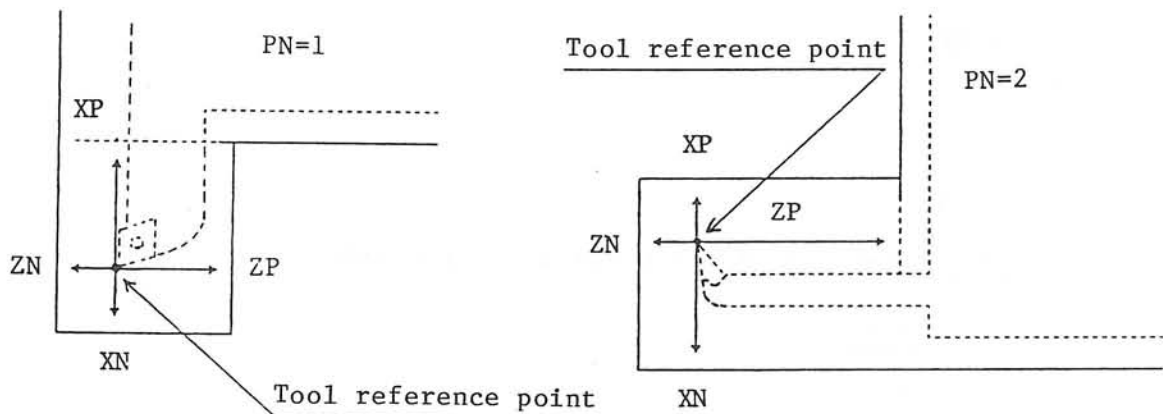
7-3-2. Setting of Tool Interference Area

The procedure to set the tool interference area differs depending on whether the graphic specification is selected or not.

- (1) For the control provided with monochrome/character specification

The data to define the tool interference area is set in the TOOL DATA setting mode.

Enter the tool interference area pattern and tool interference area setting data for each of the tools. The tool interference area pattern is selected from the OD tool pattern and the ID tool pattern. The tool interference area is defined by four lines. See Fig. 7-1 below.



Pattern No. 1 (for OD Tools)

Pattern No. 2 (for ID Tools)

Fig. 7-1 Tool Interference Area Pattern and Area Defining Data

PN: Tool interference area pattern selection number

1 = for OD tools

2 = for ID tools

ZN: Position of the area defining line in the negative direction of Z-axis in reference to the tool reference point






ZP: Position of the area defining line in the positive direction of Z-axis in reference to the tool reference point

XN: Position of the area defining line in the negative direction of X-axis in reference to the tool reference point

XP: Position of the area defining line in the positive direction of X-axis in reference to the tool reference point

TOOL DATA SET		N	0
Page 1	A turret	UNIT 1mm	
* TOOL INTERFERENCE CHECK DATA *			
TOOL NO. 1			
PATTERN NO.	PN= 1	PN 1	
POINT	ZN= 5.000		
	ZP= 37.000		
	XN= 5.000		
	XP= 5.000		
=IF =IF =IF =			
SET	ADD	ITEM1	ITEM1
<div style="display: flex; justify-content: space-around;"> [F 1] [F 2] [F 3] [F 4] [F 5] [F 6] [F 7] [F 8] </div>			

Procedure for setting tool interference area in MDI operations:

- a) Press the TOOL DATA key to set the TOOL DATA SET mode.
- b) Press the function key [F7] (ITEM ↓) to display the *TOOL INTERFERENCE CHECK DATA* page. See the CRT above.
- c) Press the  key to select A-turret.
- d) Press the  or  key to display the page listing the tool number for which the tool interference area is to be set.
- e) Press the  or  key to designate the data to be set.
- f) Press the function key [F1] (SET) and key in the desired data.
- g) Press the WRITE key.

Repeat steps c) through g) to set the necessary tool interference area setting data.

For the tools on B-turret, set the data in the similar manner.

Procedure for setting tool interference area using program:

Designate tool interference area setting data for individual tools using the system variables indicated below:

VTIPN[I] = Pattern No. (1 or 2)
 VTIZN[I] = Value ZN in Fig. 7-1 (\emptyset - 9999.999 mm)
 VTIZP[I] = Value ZP in Fig. 7-1 (\emptyset - 9999.999 mm)
 VTIXN[I] = Value XN in Fig. 7-1 (\emptyset - 9999.999 mm)
 VTIXP[I] = Value XP in Fig. 7-1 (\emptyset - 9999.999 mm)

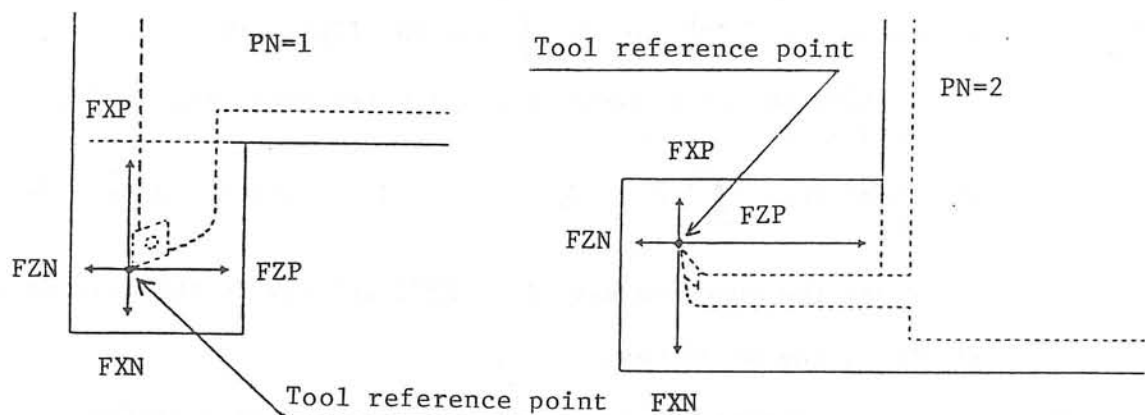
I indicates a tool number.

Note: For the color/monochrome graphic specification, the interference area data set for individual tool kind number is automatically used as the interference data of them and thus these system variables cannot be used.

- (2) For the control provided with the color/monochrome graphic specification

No special operations for setting the tool interference area data is required since the tool interference area data is registered with the tool shape data. Selection of the tool shape in the TOOL DATA SET mode automatically selects the interference area setting data. However, if the change of the registered data is required, it is possible in the parameter setting mode.

Tool interference pattern and tool interference area setting data are registered with the tool kind number. The tool interference area pattern is selected from the OD tool pattern and the ID tool pattern. The tool interference area is defined by four lines. See Fig. 7-2 below.

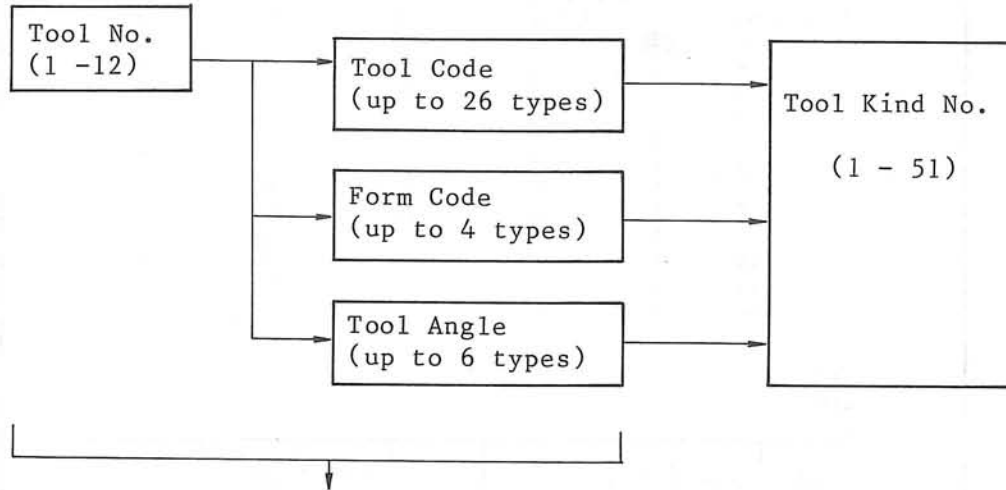


Pattern No. 1 (for OD Tools)

Pattern No. 2 (for ID Tools)

Fig. 7-2 Tool Interference Area Pattern and Area Defining Data

The tool kind number is determined from the tool code and from code for individual tool numbers, and the tool angle. See Tables 7-1 and 7-2.









These are set in the tool form selection step.

For details of the selection of the tool shape, refer to Section 24, "Color Graphic CRT Display" in this Programming Manual.

PARAMETER SET TESTE.MIN						N	3
Page 1						UNIT 1mm	
* TOOL INTERFERENCE PARAMETER *							
TL KIND NO.	PN	FZN	FZP	FXN	FXP		
1	1	5	37	5	5		
2	1	5	37	5	5		
3	1	5	37	5	5		
4	1	5	37	5	5		
5	1	5	37	5	5		
6	1	5	37	5	5		
7	1	18	18	5	5		
8	1	18	18	5	5		
9	1	18	18	5	5		
10	1	37	5	5	5		
11	1	37	5	5	5		
12	1	37	5	5	5		
-IF							
-IF							
-IF							
.							
SET	ADD	CAL		ITEMI	ITEMI		

Procedure for changing the tool interference area setting data in the MDI mode:

- a) Press the PARAMETER SET key to set the PARAMETER SET mode.
- b) Press the function key [F7] (ITEM ↓) to display the TOOL INTERFERENCE PARAMETER page.
- c) Press the  or  key to display the page which contains the desired tool kind number. One page contains 12 tool type numbers and a total of five pages are available (1 to 12, 13 to 24, 25 to 36, 37 to 48 and 49 to 60).
- d) Use the  or  key to locate the cursor on the desired tool kind number.
- e) Use the  or  key to locate the cursor on the data to be changed.
- f) Press the function key [F1] (SET) and key in the desired data.
- g) Press the WRITE key.

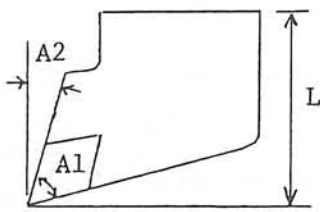
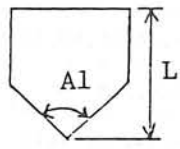
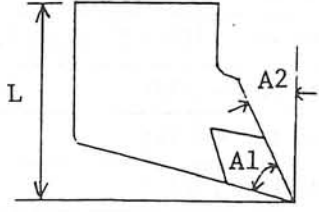
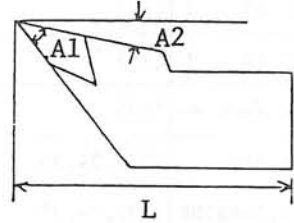
Repeat steps c) through g) to change the tool interference area setting data. In this operation, A and B TURRET selection may be ignored since the data is registered with the tool kind number.

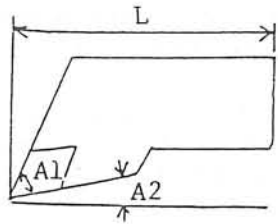
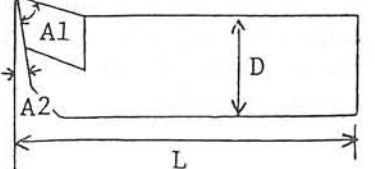
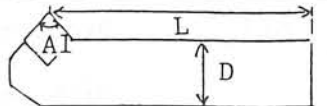
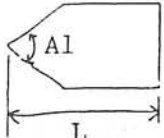
Table 7-1 Tool Kind Number Selection Chart by Tool Code and Form Code

Form Code (1-4)		1	2	3	4
1	ROUGH OD ←	(1), 2, 3, 4, 5, 6	(7), 8, 9	(22), 23, 24, 25, 26, 27	
2	ROUGH ID ←	(36), 37, 38, 39, 40, 41	(42)	(46), 47, 48	
3	ROUGH FACE ↓	(1), 2, 3, 4, 5, 6	(22), 23, 24, 25, 26, 27	(49), 50, 51	
4	ROUGH OD →	(10), 11, 12, 13, 14, 15	(7), 8, 9		
5	ROUGH ID →	(43)	(46), 47, 48		
6	ROUGH FACE ↑	(36), 37, 38, 39, 40, 41	(16), 17, 18, 19, 20, 21	(49), 50, 51	
7	FINISH OD ←	(1), 2, 3, 4, 5, 6	(7), 8, 9	(22), 23, 24, 25, 26, 27	
8	FINISH ID ←	(36), 37, 38, 39, 40, 41	(42)	(46), 47, 48	
9	FINISH FACE ↓	(1), 2, 3, 4, 5, 6	(22), 23, 24, 25, 26, 27		
10	FINISH OD →	(10), 11, 12, 13, 14, 15	(7), 8, 9		
11	FINISH ID →	(43)	(46), 47, 48		
12	FINISH FACE ↑	(36), 37, 38, 39, 40, 41	(16), 17, 18, 19, 20, 21	(49), 50, 51	
13	THREAD OD ←	(30)	(7), 8, 9		
14	THREAD ID ←	(44)			
15	THREAD FACE ↓	(31)	(49), 50, 51		
16	THREAD OD →	(30)	(7), 8, 9		
17	THREAD ID →	(44)			
18	THREAD FACE ↑	(31)	(49), 50, 51		
19	GROVE OD ↓	(28)	(7), 8, 9	(30)	
20	GROVE ID ↑	(45)	(44)		
21	GROVE FACE ←	(29)	(31)		
22	DRILL, HSS	(33), 34, 35			
23	DRILL, CARBIDE	(33), 34, 35			
24	DRILL, CENTER	(32)			
25	RECESS OD	(1), 2, 3, 4, 5, 6	(22), 23, 24, 25, 26, 27		
26	RECESS ID	(36), 37, 38, 39, 40, 41			

Numbers in () in the table above are the tool kind number displayed as a default on the guidance page for the tool for selection in TOOL DATA setting operations.

Table 7-2 Tool Kind Number Selection Chart by Tool Angle

Tool Kind No.	Conditions	Remarks
1	$78^\circ \leq A1 \leq 90^\circ$	 <p>A1: Tool angle A2: Edge angle L: Sticking out</p>
2	$60^\circ \leq A1 < 78^\circ$	
3	$54^\circ \leq A1 < 60^\circ, 3^\circ < A2 < 90^\circ$	
4	$54^\circ \leq A1 < 60^\circ, 0^\circ \leq A2 \leq 3^\circ$	
5	$0^\circ < A1 < 54^\circ, 3^\circ < A2 < 90^\circ$	
6	$0^\circ < A1 < 54^\circ, 0^\circ \leq A2 \leq 3^\circ$	
7	$60^\circ \leq A1 < 180^\circ$	 <p>A1: Tool angle L: Sticking out</p>
8	$54^\circ \leq A1 < 60^\circ$	
9	$0^\circ < A1 < 54^\circ$	
10	$78^\circ \leq A1 \leq 90^\circ$	 <p>A1: Tool angle A2: Edge angle L: Sticking out</p>
11	$60^\circ \leq A1 < 78^\circ$	
12	$54^\circ \leq A1 < 60^\circ, 3^\circ < A2 < 90^\circ$	
13	$54^\circ \leq A1 < 60^\circ, 0^\circ \leq A2 \leq 3^\circ$	
14	$0^\circ < A1 < 54^\circ, 3^\circ < A2 < 90^\circ$	
15	$0^\circ < A1 < 54^\circ, 0^\circ \leq A2 \leq 3^\circ$	
16	$78^\circ \leq A1 \leq 90^\circ$	 <p>A1: Tool angle A2: Edge angle L: Sticking out</p>
17	$60^\circ \leq A1 < 78^\circ$	
18	$54^\circ \leq A1 < 60^\circ, 3^\circ < A2 < 90^\circ$	
19	$54^\circ \leq A1 < 60^\circ, 0^\circ \leq A2 \leq 3^\circ$	
20	$0^\circ < A1 < 54^\circ, 3^\circ < A2 < 90^\circ$	
21	$0^\circ < A1 < 54^\circ, 0^\circ \leq A2 \leq 3^\circ$	

Tool Kind No.	Conditions	Remarks
22	$78^\circ \leq A1 \leq 90^\circ$	 <p>A1: Tool angle A2: Edge angle L: Sticking out</p>
23	$60^\circ \leq A1 < 78^\circ$	
24	$54^\circ \leq A1 < 60^\circ, 3^\circ < A2 < 90^\circ$	
25	$54^\circ \leq A1 < 60^\circ, 0^\circ \leq A2 \leq 3^\circ$	
26	$0^\circ < A1 < 54^\circ, 3^\circ < A2 < 90^\circ$	
27	$0^\circ < A1 < 54^\circ, 0^\circ \leq A2 \leq 3^\circ$	
33	$120^\circ < A1 \leq 180^\circ$	
34	$90^\circ < A1 \leq 120^\circ$	
35	$0^\circ < A1 \leq 90^\circ$	
36	$78^\circ \leq A1 \leq 90^\circ$	 <p>A1: Tool angle A2: Edge angle L: Holder length D: Holder dia.</p>
37	$60^\circ \leq A1 < 78^\circ$	
38	$54^\circ \leq A1 < 60^\circ, 3^\circ < A2 < 90^\circ$	
39	$54^\circ \leq A1 < 60^\circ, 0^\circ \leq A2 \leq 3^\circ$	
40	$0^\circ < A1 < 54^\circ, 3^\circ < A2 < 90^\circ$	
41	$0^\circ < A1 < 54^\circ, 0^\circ \leq A2 \leq 3^\circ$	
46	$60^\circ \leq A1 < 180^\circ$	 <p>A1: Tool angle D: Holder dia. L: Holder length</p>
47	$54^\circ \leq A1 < 60^\circ$	
48	$0^\circ < A1 < 54^\circ$	
49	$60^\circ \leq A1 < 180^\circ$	 <p>A1: Tool angle L: Sticking out</p>
50	$54^\circ \leq A1 < 60^\circ$	
51	$0^\circ < A1 < 54^\circ$	

SECTION 8 DROOP CONTROL

The axis movements on NC machines are controlled by the servo system in which the axis moves to eliminate the lag (termed DIF or droop) between the actual tool position and the commanded coordinate.

Due to existence of DIF (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 DIF 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 DIF amount becomes smaller than the permissible droop amount.

Droop Amount

Droop amount can be set within a range from 0 to 1.000 mm through the operation panel using a parameter. For details of setting procedure, refer to Operation Manual. The control is delivered with 10 μ m setting as a standard amount.

SECTION 9 MM/MIN. MODE FEEDRATE COMMAND

Axis feedrate can be commanded in terms of travel per minute (mm/min.) instead of travel per revolution (mm/rev.). When an F word is provided in "mm/rev." mode, the spindle must be rotating for the axis to move. While, the spindle is not necessarily be rotating for the axis feed if the feedrate is provided in "mm/min." mode. Of course, the spindle may be rotating for axis movement in such feedrate command mode.

Programming Format

G94 mm/min. mode, ON

G95 mm/min. mode, OFF (mm/rev. mode ON)

In G94 mode, feedrate is specified in F0000 mm/min. The axis moves F mm per minute with this command.

F10 command, for instance, feeds the axis at a rate of 10 mm/min.

When the minimum setting unit is 0.001 mm, specify feedrates (F words) in units of 0.1 mm/min.

Programming Unit System - Unit of F Command - Max. F Value

Unit System	Unit of F	Maximum Value of F
Metric		
1 mm	1 mm/min.	99999.999 mm/min.
1 μ m	0.1 mm/min.	99999.999 mm/min.
10 μ m	1 mm/min.	99999.999 mm/min.
Inch		
1/100000 inch	0.01 inch/min.	9999.9999 inch/min.
1 inch	1 inch/min.	9999.9999 inch/min.

Actually, the maximum cutting feedrate on each model is limited at values smaller than the above indicated maximum F value.

For maximum programmable feedrate for each model, refer to the Operation Manual for individual models.

SECTION 10 ZERO SET BY G CODE

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, for tooling having large tool offset value, or other cases.

Axis Movement

The active coordinate point of X- and Z-axis have the coordinate values commanded on a tape following G50. That is, the origin of the coordinate system is shifted by "G50 X0000 Z0000" command.

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:

$$\begin{aligned}
 X &= X0 - X1 \\
 Z &= Z0 - Z1
 \end{aligned}$$

Provided, X0 = 100 mm and X1 = 200 mm, zero offset amount is calculated as;

$$100 - 200 = -100 \text{ mm}$$

This amount can be checked on the CRT.

Dimension words in sequences N006 and after that are all referenced to the origin newly established by the commands in N005.

- Note 1: Zero offset of the axis not specified in the block containing G50 is not performed.
- Note 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.)
- Note 3: When the control is reset, all zero set data are cleared and the initial zero offset data become effective.
- Note 4: No tool offset number entry is allowed in the block containing G50 code.

SECTION 11 TAPER ANGLE/ARC RADIUS DIRECT COMMAND

11-1. GENERAL DESCRIPTION

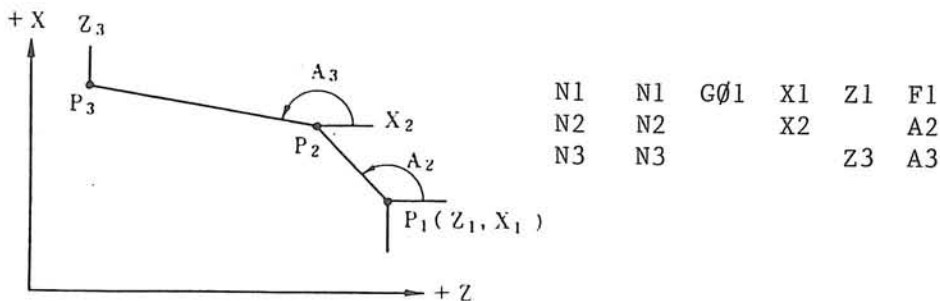
Dimension words in a conventional NC program are all expressed using the coordinate values of respective points although the dimensions are not always given in such manner on a part drawing; for a taper, it is usually specified by its angle to the datum line and a circle or an arc is specified by the radius.

This feature permits the use of these data, i.e., taper angle and arc radius, in a program to facilitate programming.

11-2. 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).



(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".

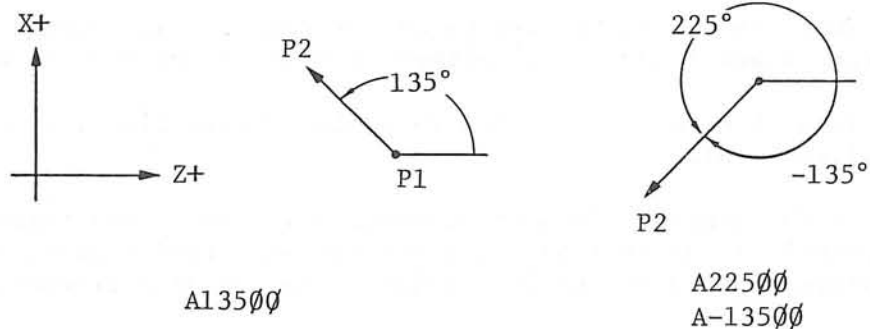
(3) Unit of angle command is:

Metric	{	1 μm unit system	0.001 deg.
		10 μm unit system	0.01 deg.
		1 mm unit system	1 deg.
Inch	{	1/100000 inch unit system	0.0001 deg.
		1 inch unit system	1 deg.

- (4) The control interpretes 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
 Simultaneous 4-axis cuts on 2S model
 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.



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.

11-3. DIRECT RADIUS COMMAND

In conventional programming, circular interpolation is activated by G02 and G03 codes associated with I and K words used to indicate the coordinates of the center of the arc to be cut along with the X and Z words which indicate the coordinate of the end point of the arc.

This feature permits circular interpolation by simply providing the radius of the arc along with the coordinates of the end point with X and Z words.

- (1) As a G code calling for circular interpolation, G02 or G03 is used as in conventional programming.
- (2) Radius of the arc is expressed by an L word which must have a positive value.
- (3) Block containing an L word without K or I word needs an arc radius command.
- (4) 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.
- (5) If an L word is specified in a block containing I and/or K word, an alarm results.
- (6) If the distance from the current position to the target point (end point) is larger than two times the specified radius, it results an alarm since circular interpolation cannot be performed.
- (7) In direct arc command programming, one arc command yields two arcs; one with central angle less than 180 deg., and another larger than 180 deg.

The arc with central angle less than 180 deg. is selected.

To obtain the one with central angle larger than 180 deg., specify "CALRG" in the block commanding circular interpolation.

- (8) The direct radius command programming is effective in:

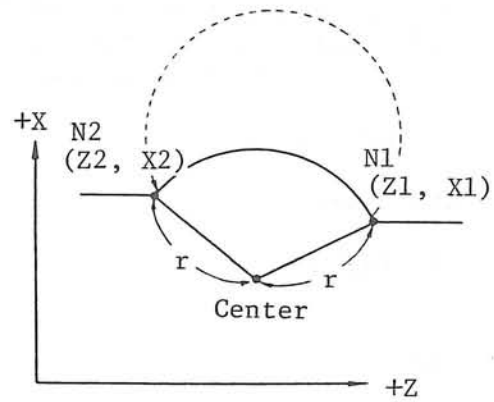
LAP
 Tool nose radius compensation mode
 Subprogram
 Simultaneous 4-axis cuts on 2S model
 Incremental programming mode (G91)

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 X1 Z1 F1
N2 G03 X2 Z2 Lr
```



With the commands above, the arc indicated by a thick solid line is obtained. To move the tool along the arc indicated by dashed lines, program as follows:

```
N1 G01 X1 Z1 F1
N2 G03 CALRG X2 Z2 Lr
```

SECTION 12 AUTOMATIC CHAMFERING

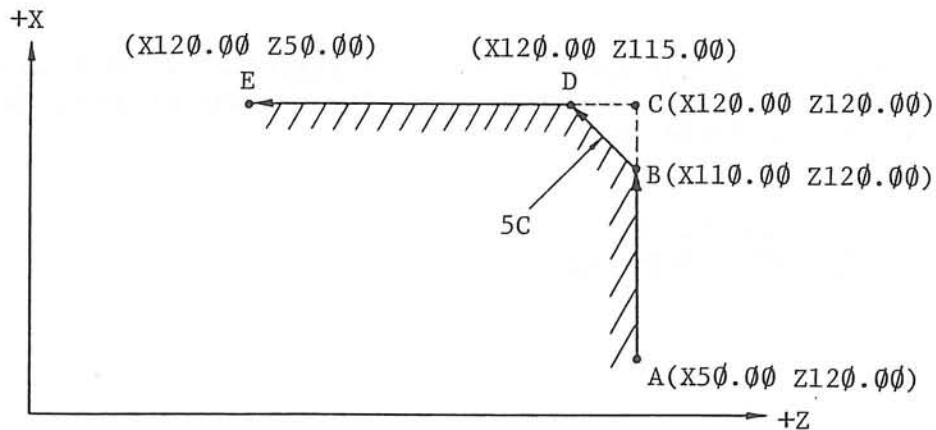
12-1. GENERAL DESCRIPTION

When cutting a workpiece, it is often necessary to chamfer the sharp edge (either 45 deg. chamfering or rounding). Such chamfering can be accomplished using conventional taper and circular interpolation G codes such as G01, G02 and G03. However, this Automatic Chamfering Function permits chamfering to be done in a simple programming.

Commands used in this feature are:

G75: Chamfering at 45 deg.
 G76: Rounding
 L : Size of chamfering

12-2. 45 DEG. CHAMFERING (G75)



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

```
G75 G01 X120 L-5 F00 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 deg. with the size of 5 mm.

Explanation of the commands are provided in the following page.

G75 : Specifies chamfering at 45 deg.

X12 \emptyset : X coordinate of Point C

L-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.

Note 1: G75 is effective only in G \emptyset 1 mode. If G75 is specified in other mode, it causes an alarm.

Note 2: G75 is non-modal and active only in the commanded block.

Note 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.

Note 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.

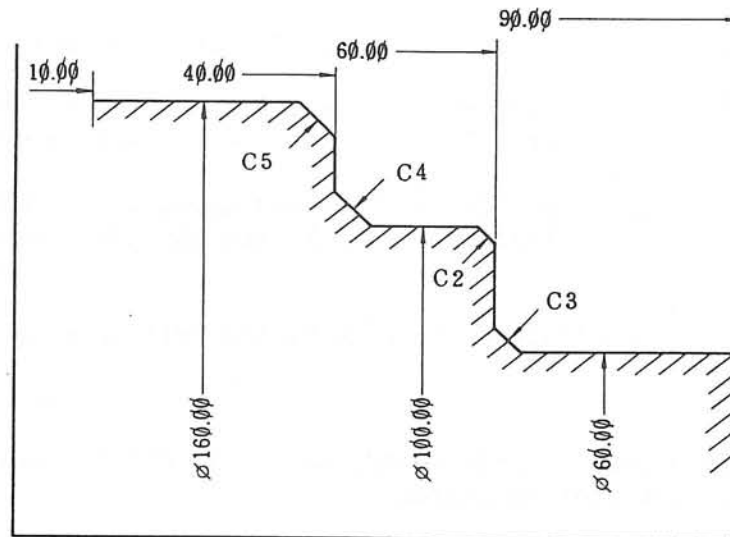
Note 5: The automatic chamfering program is effective in:

LAP

Tool nose radius compensation mode

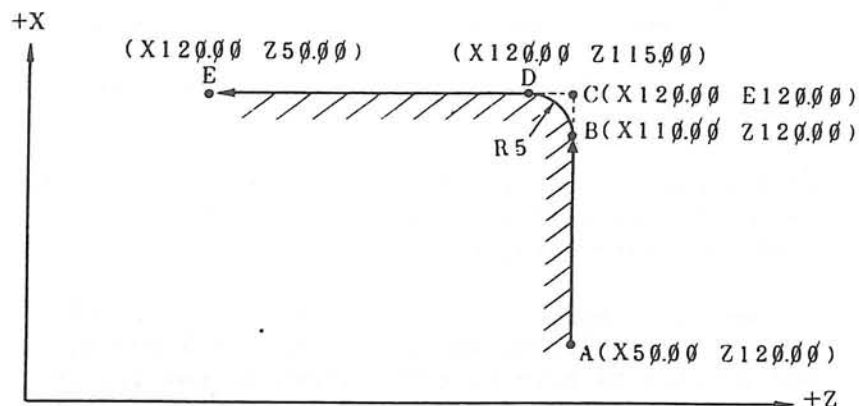
See the program example in the following page.

Example (Chamfering at 45°)



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

12-3. ROUNDING (G76)



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

```
G76 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 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.

- Note 1: G76 is effective only in G01 mode. If G76 is specified in other than G01 mode, it causes an alarm.
- Note 2: G76 is non-modal and active only in the commanded block.
- Note 3: Rounding is made as 1/4 circle having the radius specified by an L word.
- Note 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.
- Note 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.

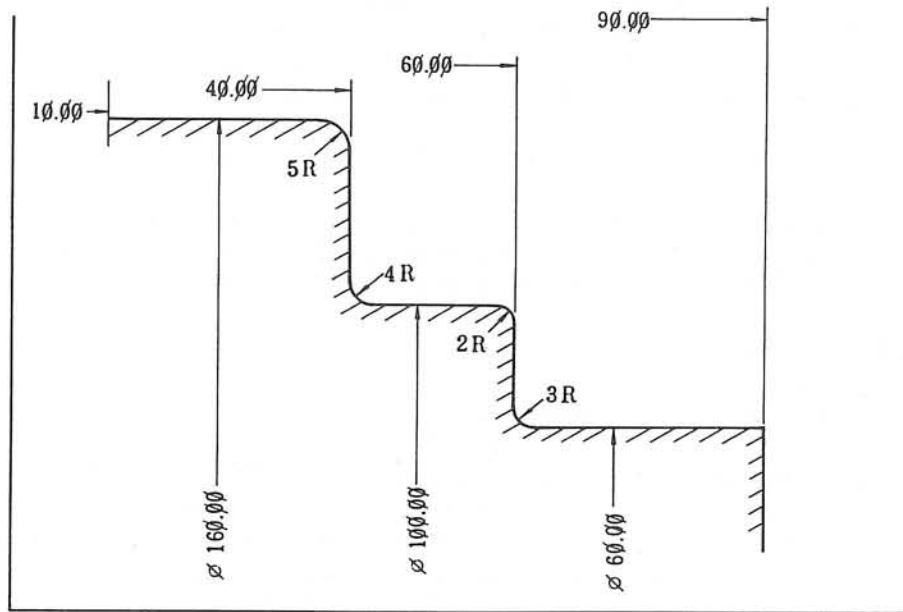
- Note 6: The automatic chamfering program is effective in:

LAP

Tool nose radius compensation mode

See the program example in the following page.

Example (Rounding corner)



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

SECTION 13 COMPOUND FIXED CYCLE (SPECIAL FIXED CYCLE)

13-1. GENERAL DESCRIPTION

This feature allows a series of cyclic operation, which usually requires commands over several to more than ten blocks, to be specified by the commands in one block making the most of one of the features inherent in OSP5000L-G/OSP5000L-G, high processing speed.

There are three types of compound fixed cycle as:

- (1) Thread cutting compound fixed cycle (G71, G72)
- (2) Grooving/drilling compound fixed cycle (G73, G74)
- (3) Tapping compound fixed cycle (G77, G78)

Outline of individual fixed cycle

- (1) Thread cutting compound fixed cycle

Two modes of thread cutting cycles as G71 longitudinal thread cutting cycle and G72 transverse (end face) thread cutting cycle are available. In addition, combination of M code designating cutting mode and the one selecting infeed pattern permits the programmer to select the most desirable mode of thread cutting from available six types of thread cutting cycles.

- (2) Grooving/drilling compound fixed cycle

Two modes of grooving/drilling cycles as G73, cutting in longitudinal direction, and G74, cutting in transverse (end face) direction are available; this cycle simplifies the programming of grooving and parting-off on OD turning and deep hole drilling on end face cutting.

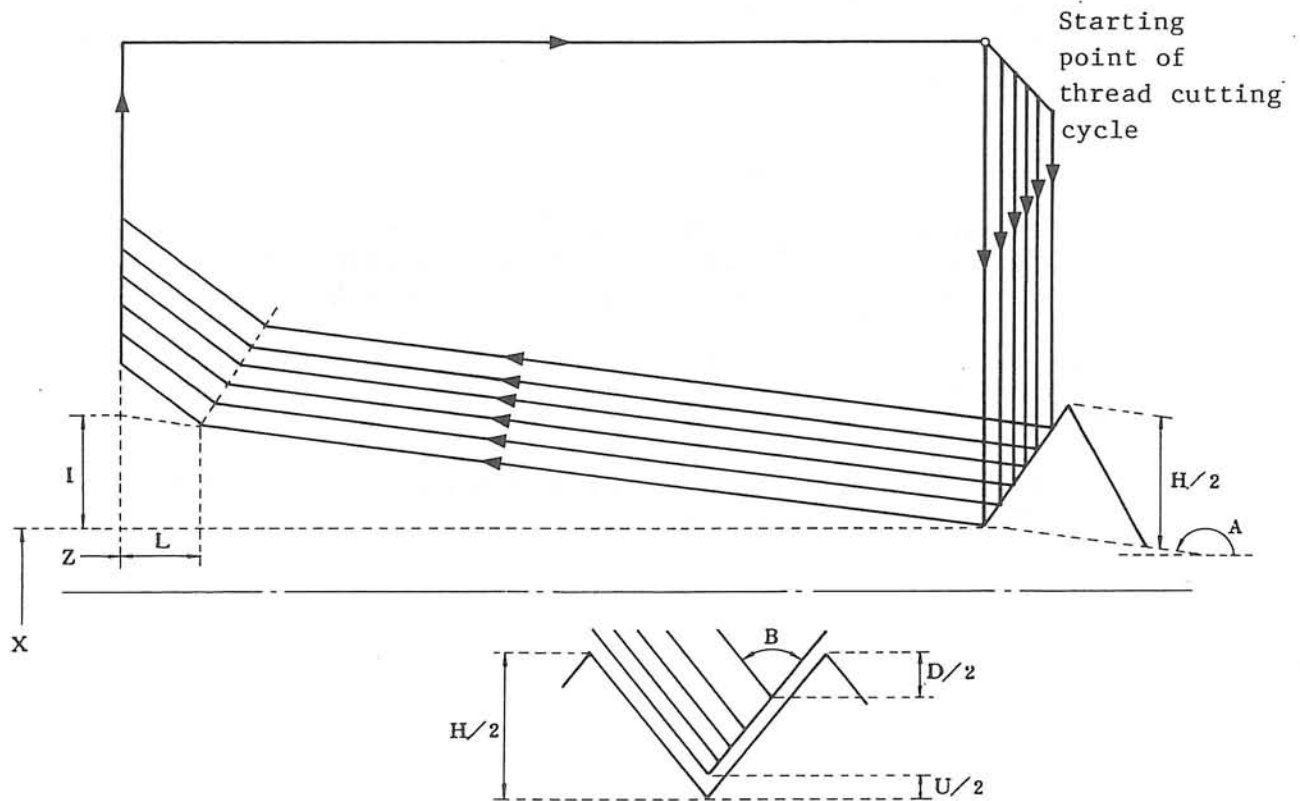
- (3) Tapping compound fixed cycle

Two modes of tapping cycles as G77 right-hand tapping cycle and G78 left-hand tapping cycles are available. This cycle simplifies the programming for both right-hand and left-hand tapping operations.

13-2. THREAD CUTTING COMPOUND CYCLE (G71/G72)

13-2-1. Longitudinal Thread Cutting Cycle (G71)

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



Format:

`N0001 G71 X Z A (I) B D U H L E F J M`

Description of each word:

X: Final diameter of thread

Z: Z coordinate of end point of thread

A: Taper angle

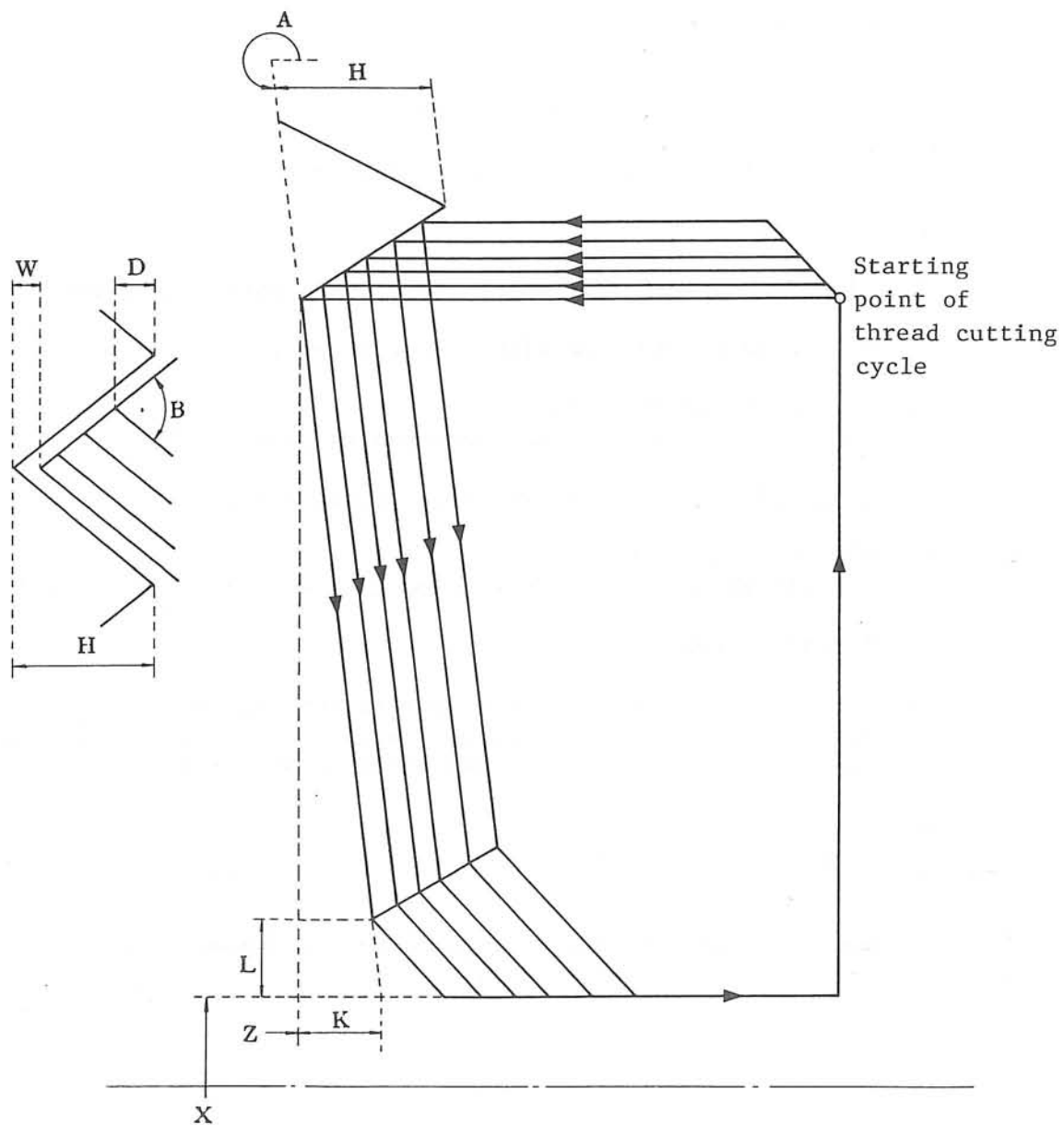
I: Difference in radius between starting point and end point for taper thread (expressed in radius)

For taper thread, use either A or I word.

- B: Cutter tip point angle
($\emptyset^\circ \leq B < 18\emptyset^\circ$; \emptyset° if no B command is provided.)
- 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:
- F: Same as in G33 mode
- J:
- M: Used to select thread cutting pattern and mode of infeed.
(For details, refer to 2-3.)

13-2-2. Transverse Thread Cutting Compound Fixed Cycle (G72)

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



Format:

NØØØ1 G72 X Z A (K) B D W H L E F J M

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: Cutter tip point angle
($0^\circ \leq B < 180^\circ$; 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 M23 mode; if no L word is provided in M23 mode, L is assumed to the distance equivalent to one lead.)
- E:
- F: Same as in G32 mode
- J:
- M: Used to select thread cutting pattern and mode of infeed.
(For details, refer to 2-3.)

13-2-3. M Code Specifying Thread Cutting Mode and Infeed Pattern

In the block calling for thread cutting compound fixed cycle, M codes specifying thread cutting mode and infeed pattern should be presented.

(1) M code specifying cutting mode

M32 straight infeed along thread face (on left face)

M33 zigzag infeed

M34 straight infeed along thread face (on right face)
(not available with A specification)

When either of these M codes is not provided, the control automatically selects M32 mode.

Note: If B word has "0" value, infeed is made parallel to the axis disregarding of selected mode.

(2) M code specifying infeed pattern

M73 Infeed pattern 1

Infeed is made by D (in diameter) in each thread cutting cycle until 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 finishing allowance 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 provided, 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 made with infeed amount of U (W). If no U (W) word is specified, finishing cycle is not performed.

M75 Infeed pattern 3

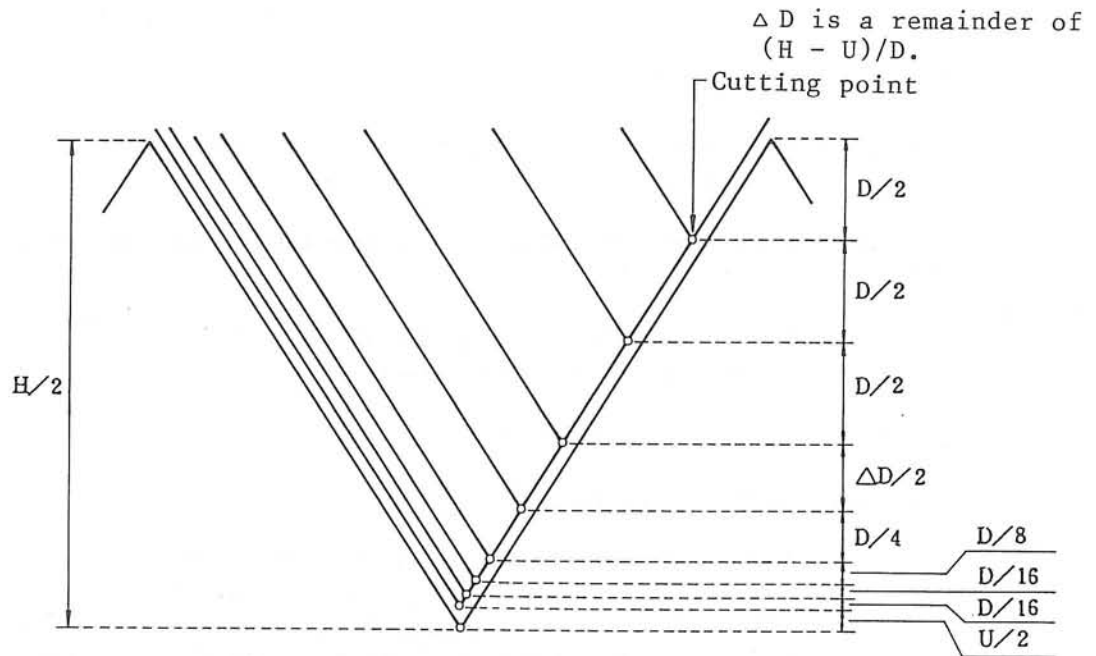
Infeed is made to D in the first cycle, to $\sqrt{2} D$ in the second cycle, and to $\sqrt{n} D$ in the "n"th cycle consecutively until "H - U (W)" position is reached. When $(\sqrt{n} - \sqrt{n-1}) \times D$ becomes smaller than U (W) during the cycle, infeed amount to be taken after that becomes U (W). Finally, the finishing cycle is performed with infeed amount of U (W). If no U (W) word is provided, finishing cycle is not performed.

When no M code specifying infeed pattern is provided, the control automatically selects M73.

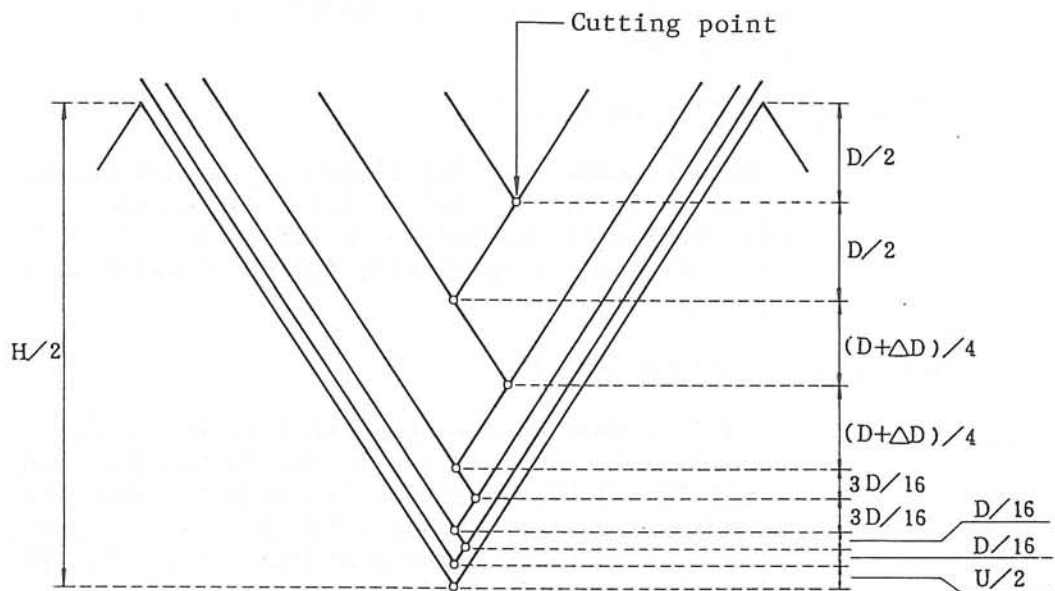
With combining the M codes specifying cutting mode and infeed pattern, six types of thread cutting cycle are available for longitudinal thread cutting cycle and transverse thread cutting cycle, respectively.

(1) Longitudinal Thread Cutting Compound Fixed Cycle

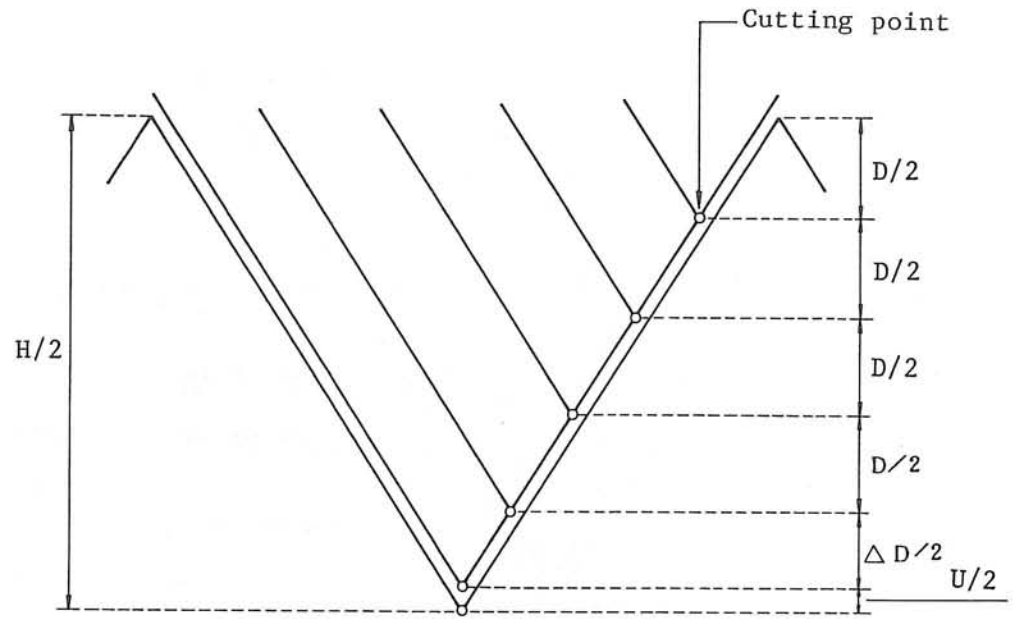
M32 + M73 Mode



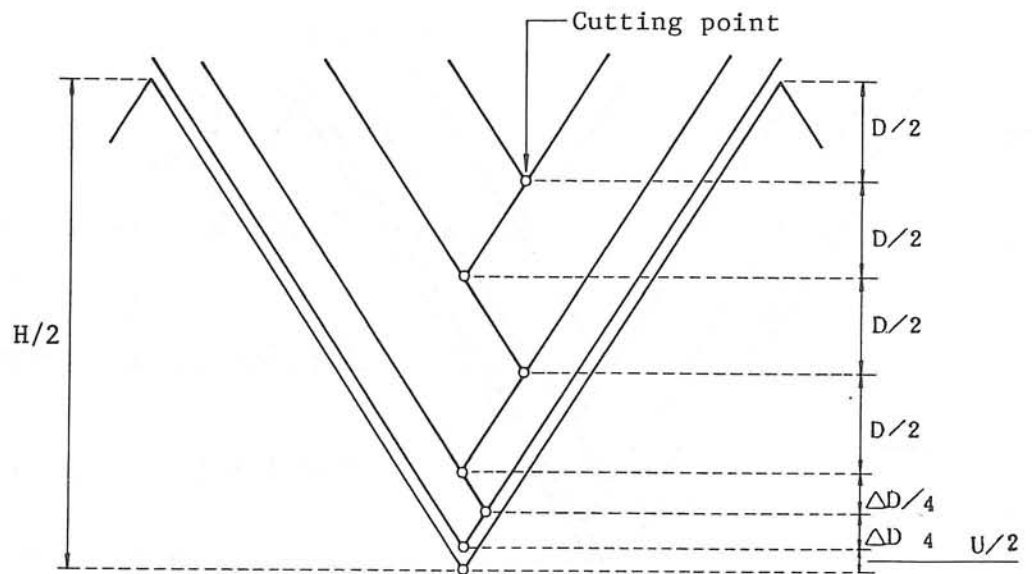
M33 + M73 Mode



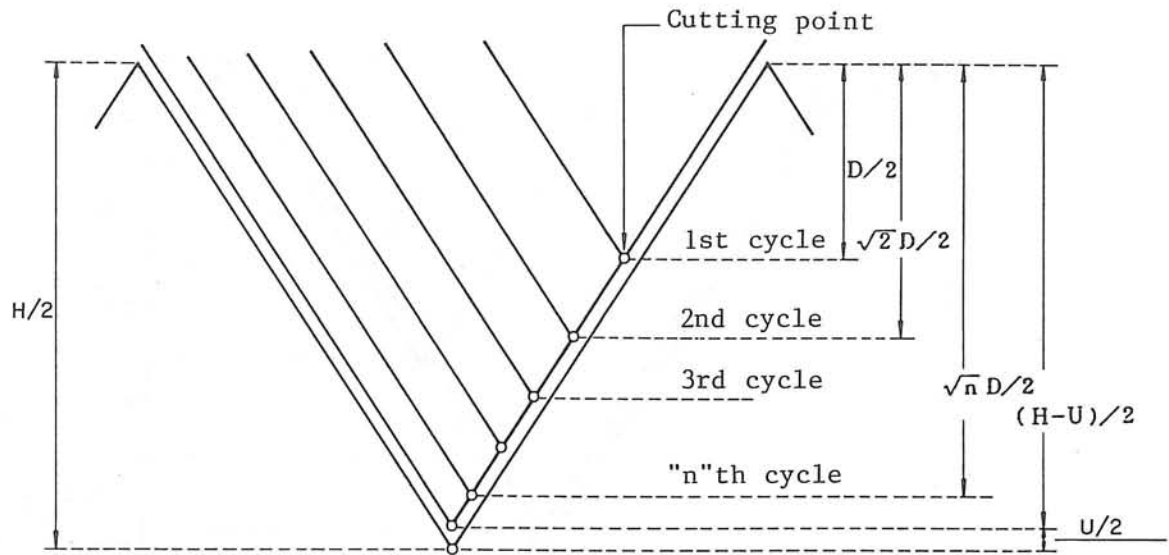
M32 + M74 Mode



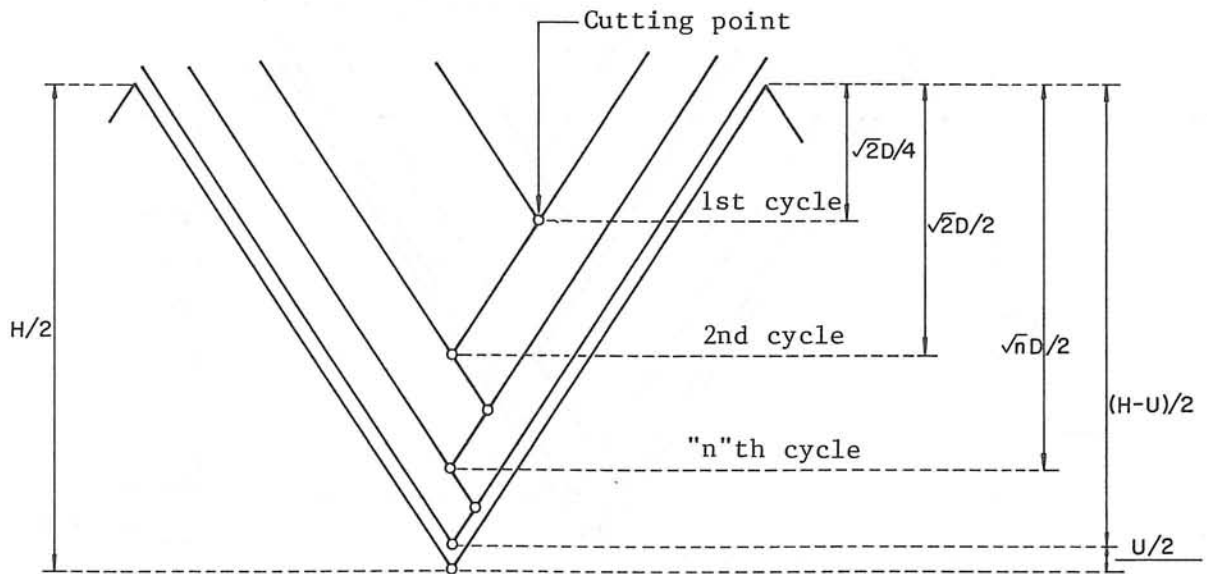
M33 + M74 Mode



M32 + M75 Mode

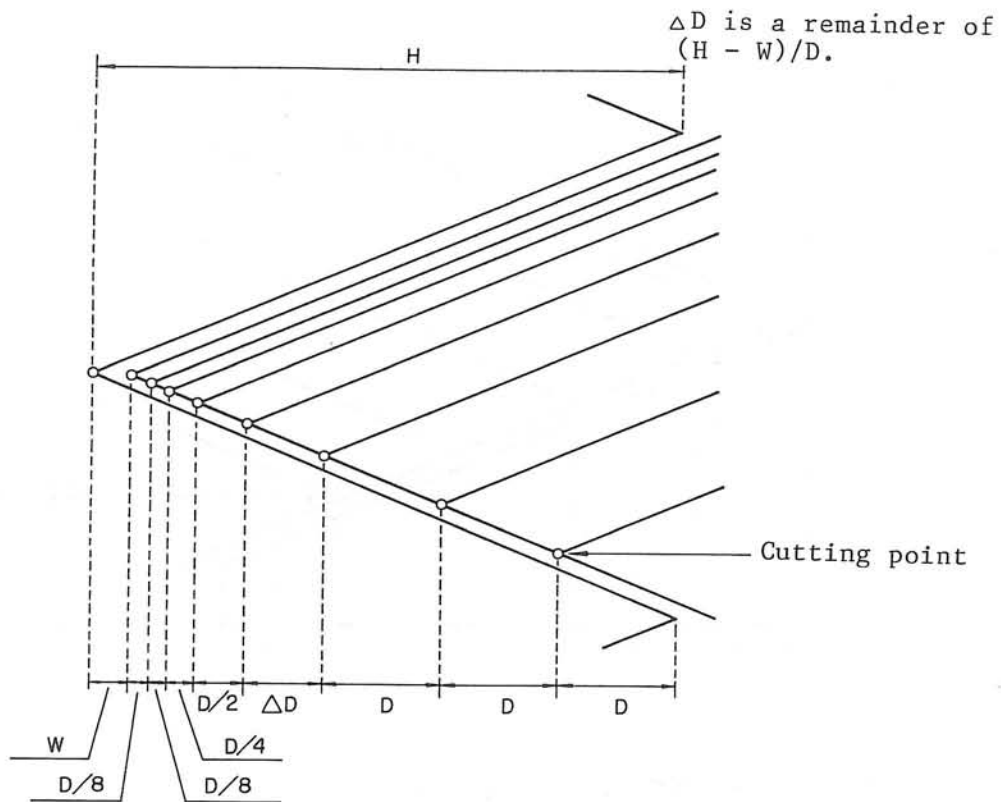


M33 + M75 Mode

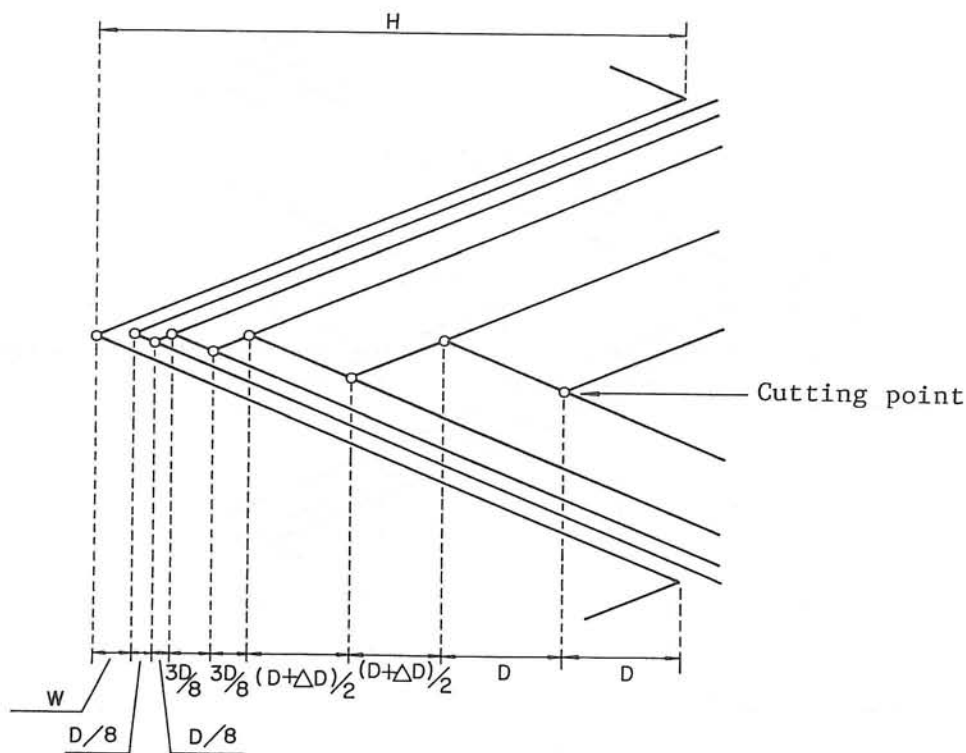


2) Transverse Thread Cutting Compound Fixed Cycle

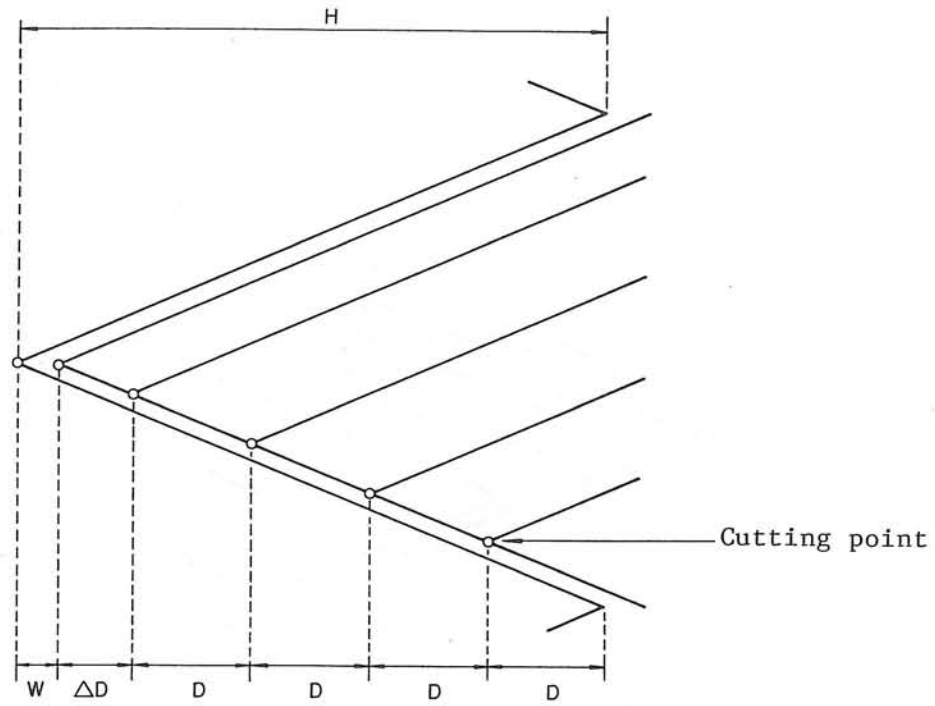
M32 + M73 Mode



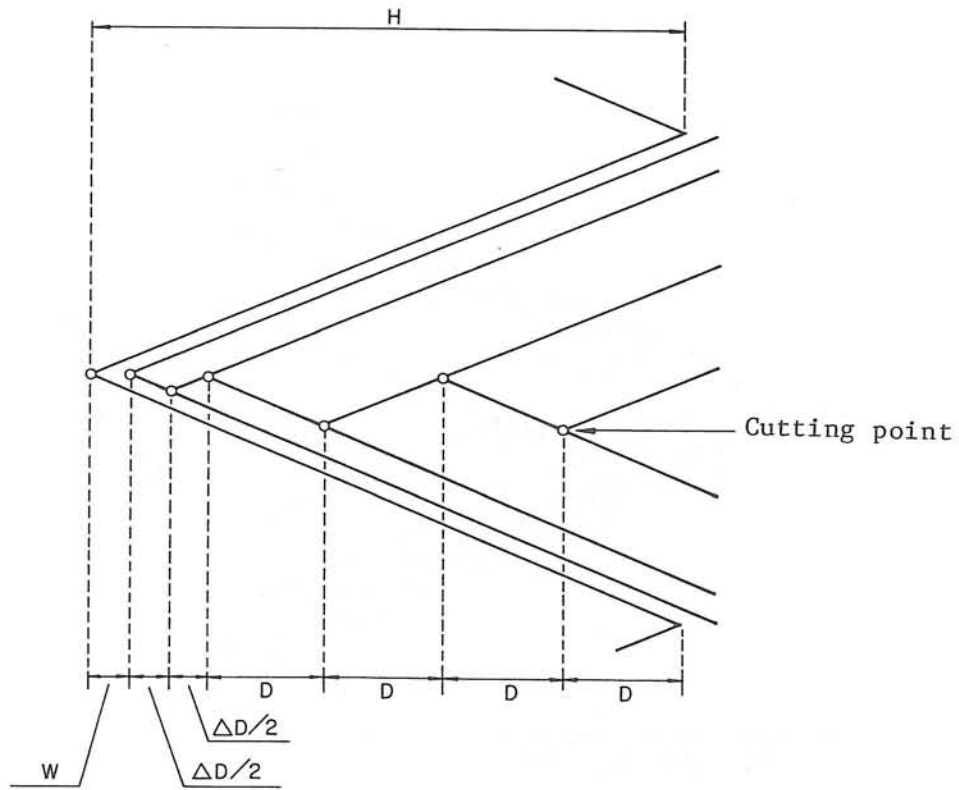
M33 + M73 Mode



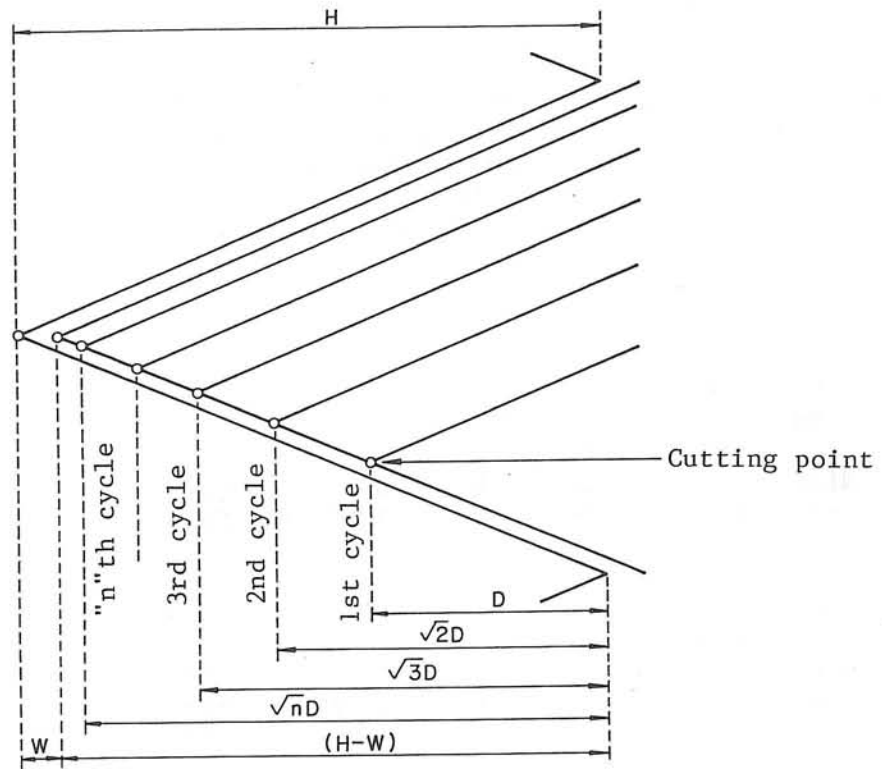
M32 + M74 Mode



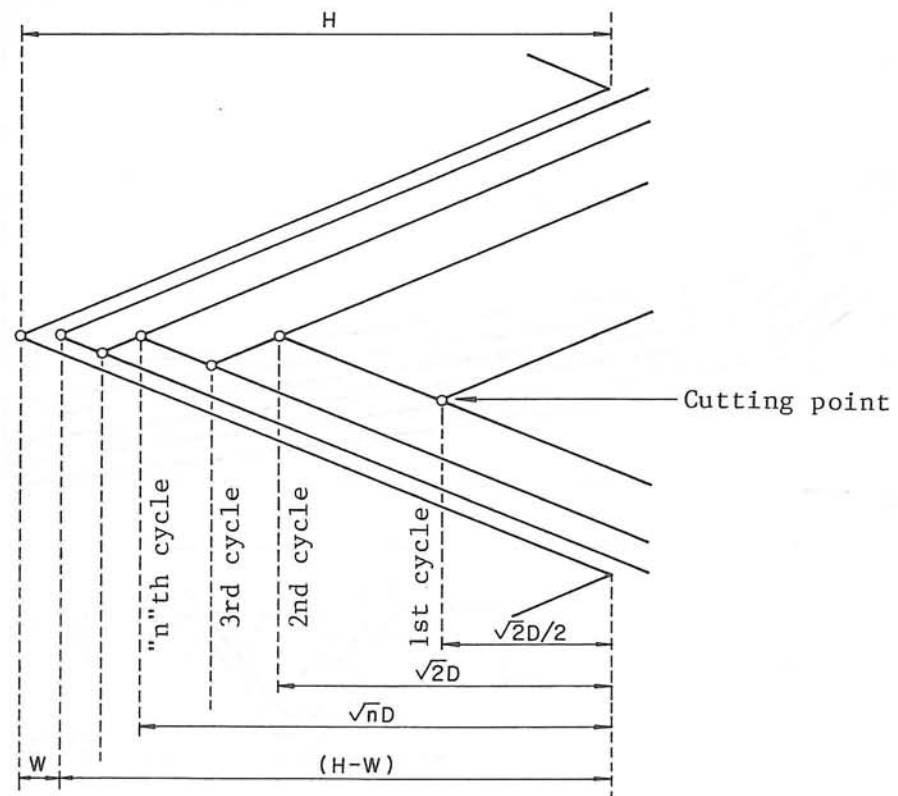
M33 + M74 Mode



M32 + M75 Mode



M33 + M75 Mode



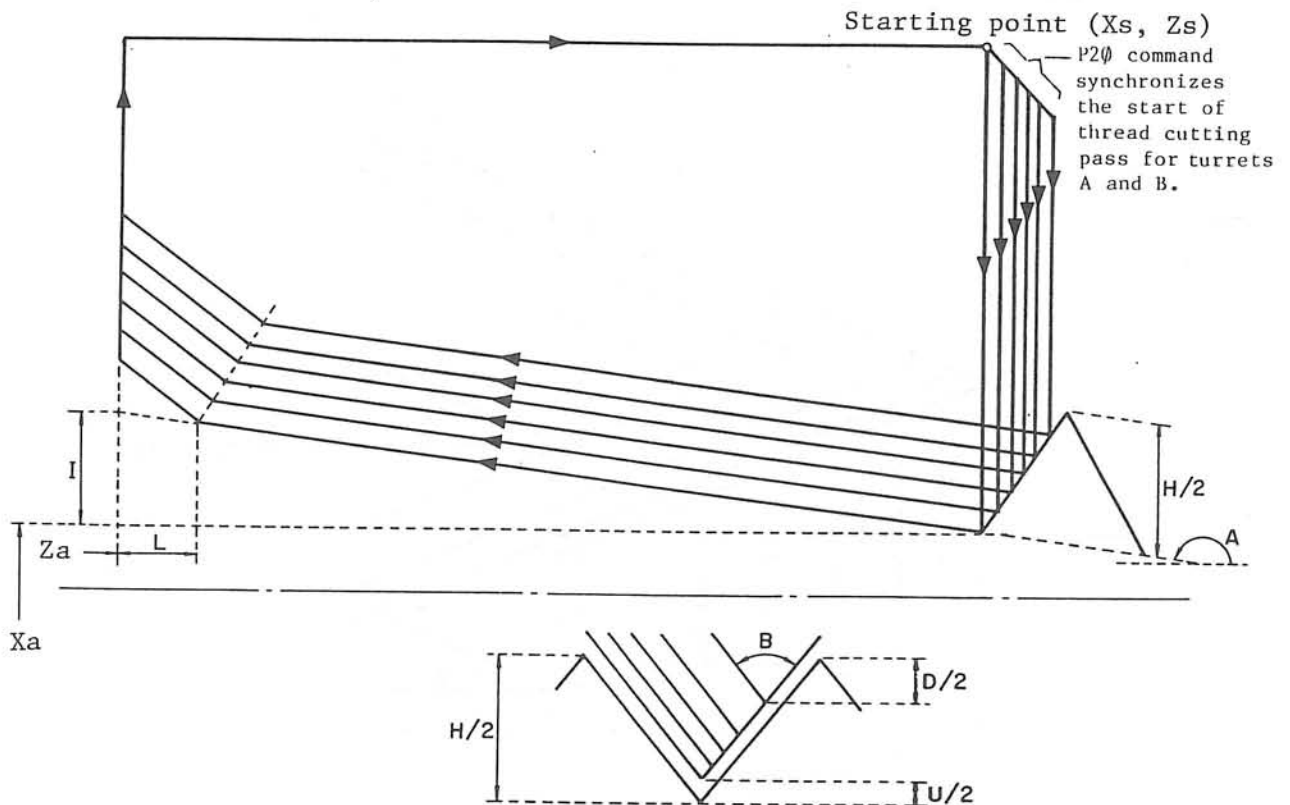
13-2-4. P Codes in the G71 and G72 Thread Cutting Cycles
(2-Saddle Specification)

P codes in a compound fixed thread cutting cycle synchronize the start of turrets A and B at each thread cutting starting point. If the number of thread cutting passes differ between the two turrets, the turret which has completed the programmed thread cutting cycle first will not proceed to the next sequence and waits for the completion of the thread cutting cycle being carried out by the other turret.

Example:

```

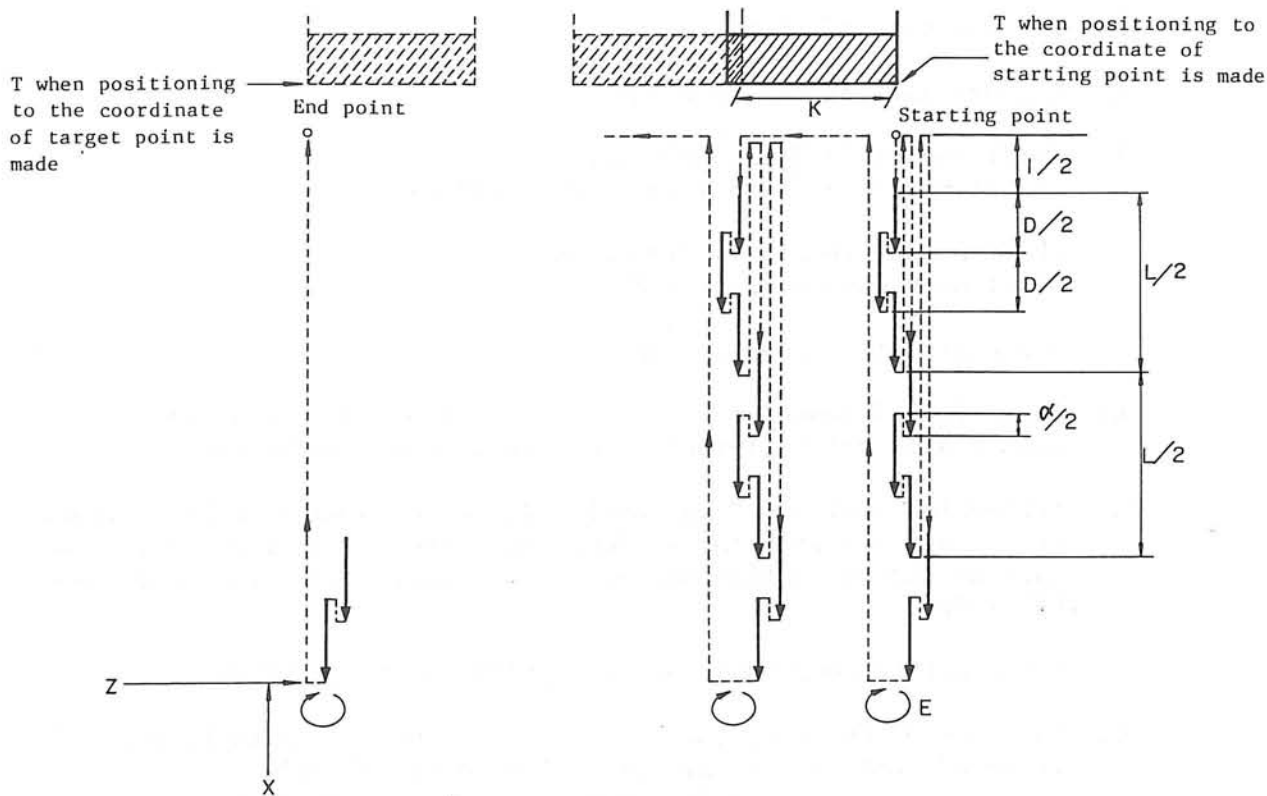
OAPT2
N100 G13
N101 G00 Xt Zt
N102 Xs Zs S T M P10
N103 G71 Xa Za I B D H F M P20
:
:
N200 G14
N201 G00 Xt Zt
N202 Xs Zs S T M P10
N203 G71 Xa Za I B D H F M P20
:
:
    
```



13-3. GROOVING/DRILLING COMPOUND FIXED CYCLE (G73/G74)

13-3-1. Longitudinal Grooving Fixed Cycle (G73)

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



Format:

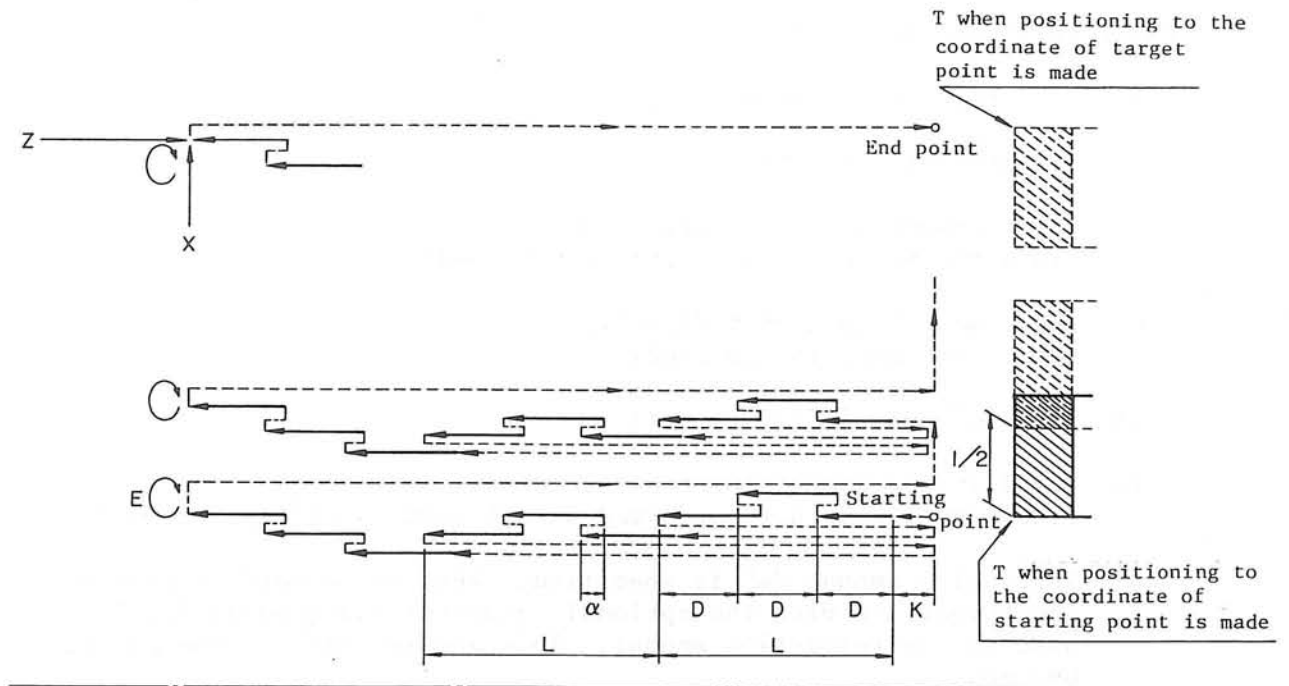
NØØØ1 G73 X Z I K D L F E T DA

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; Ø if no I word is provided)
- K: Shift amount in Z-axis direction
(Ø 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 GØ4 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.)

13-3-2. Transverse Grooving Fixed Cycle (G74)

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



Format:

N0001 G74 X Z I K D L F E T DA

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; \emptyset if no I word is provided)
- K: Shift amount in Z-axis direction
(\emptyset 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 "a" 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.
- 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.)

13-3-3. Axis Movements in Grooving 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.

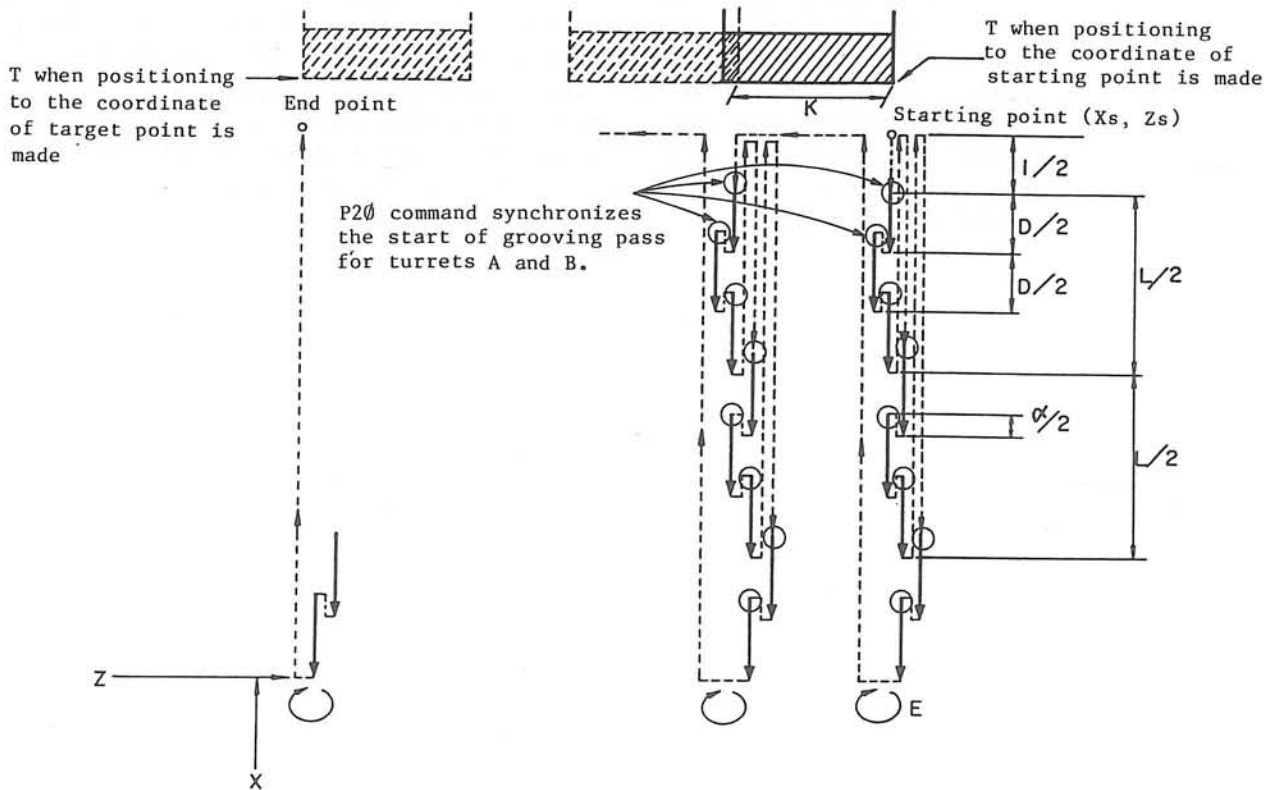
13-3-4. P Codes in the G73 and G74 Grooving Cycles
(2-Saddle Specification)

P codes in a compound fixed grooving cycle synchronize the start of turrets A and B at each groove cutting starting point. If the number of grooving passes differ between the two turrets, the turret which has completed the programmed grooving cycle first will not proceed to the next sequence and waits for the completion of the grooving cycle being carried out by the other turret.

Example:

```

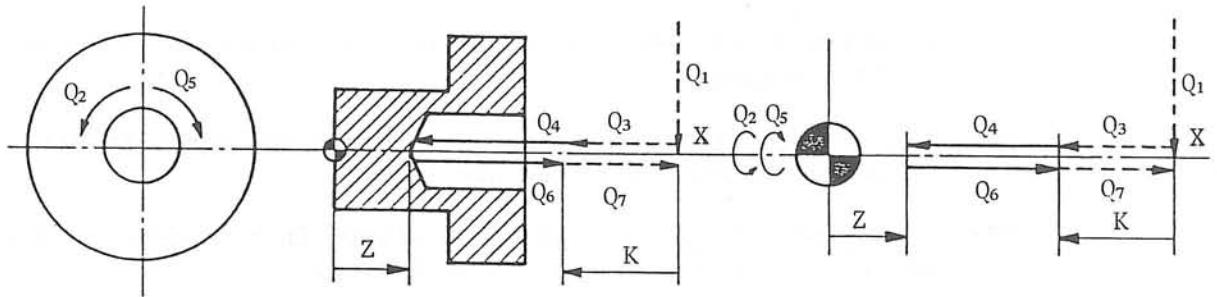
OAPT3
N100 G13
N101 G00 Xt Zt
N102 Xs Zs S T M P10
N103 G73 Xa Za I K D L F E T P20
N104 G00 Xt Zt
:
:
N200 G14
N201 G00 Xt Zt
N202 Xs Zs S T M P10
N203 G73 Xa Za I K D L F E T P20
N204 G00 Xt Zt
:
:
    
```



13-4. TAPPING COMPOUND FIXED CYCLE (G77, G78)

13-4-1. Right-hand Tapping Cycle (G77)

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



Actual Example

Illustration Example

Format:

```
N001 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:

Q1: 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 work-piece during this positioning before calling out the G77 cycle.

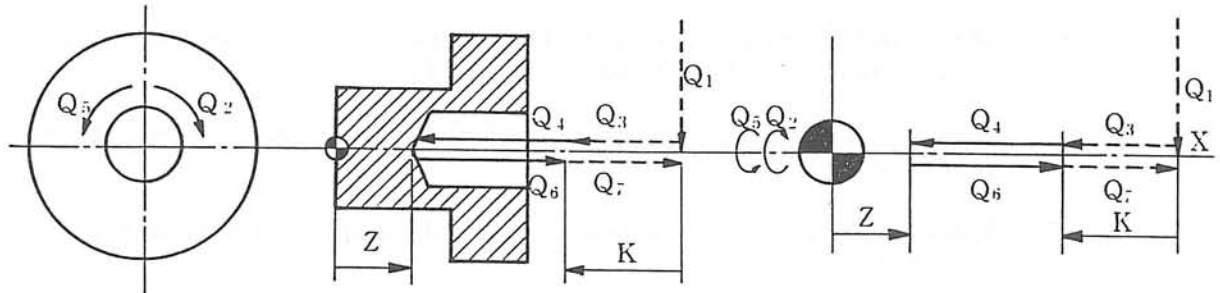
Q2: The spindle rotates clockwise at a speed active before the G77 cycle is called. Therefore, it is necessary to specify required spindle speed before calling the G77 cycle.

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.

- Q3: Positioning of Z-axis at a position designated by a K word at a rapid feedrate.
- Q4: Tapping from the point reached in Q3 to the depth specified by a Z word at a specified feedrate (F).
- Q5: The spindle stops once and then starts in the reverse direction at the same speed as used in infeeding.
- Q6: Z-axis retraction to a point reached in Q4 cycle at a cutting feedrate.
- Q7: Z-axis retraction to a point reached in Q3 cycle at a rapid feedrate.

13-4-2. Left-hand Tapping Cycle (G78)

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



Actual Example

Illustration Example

Format:

```
N001 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:

Q1: 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 work-piece during this positioning before calling out the G78 cycle.

Q2: 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.

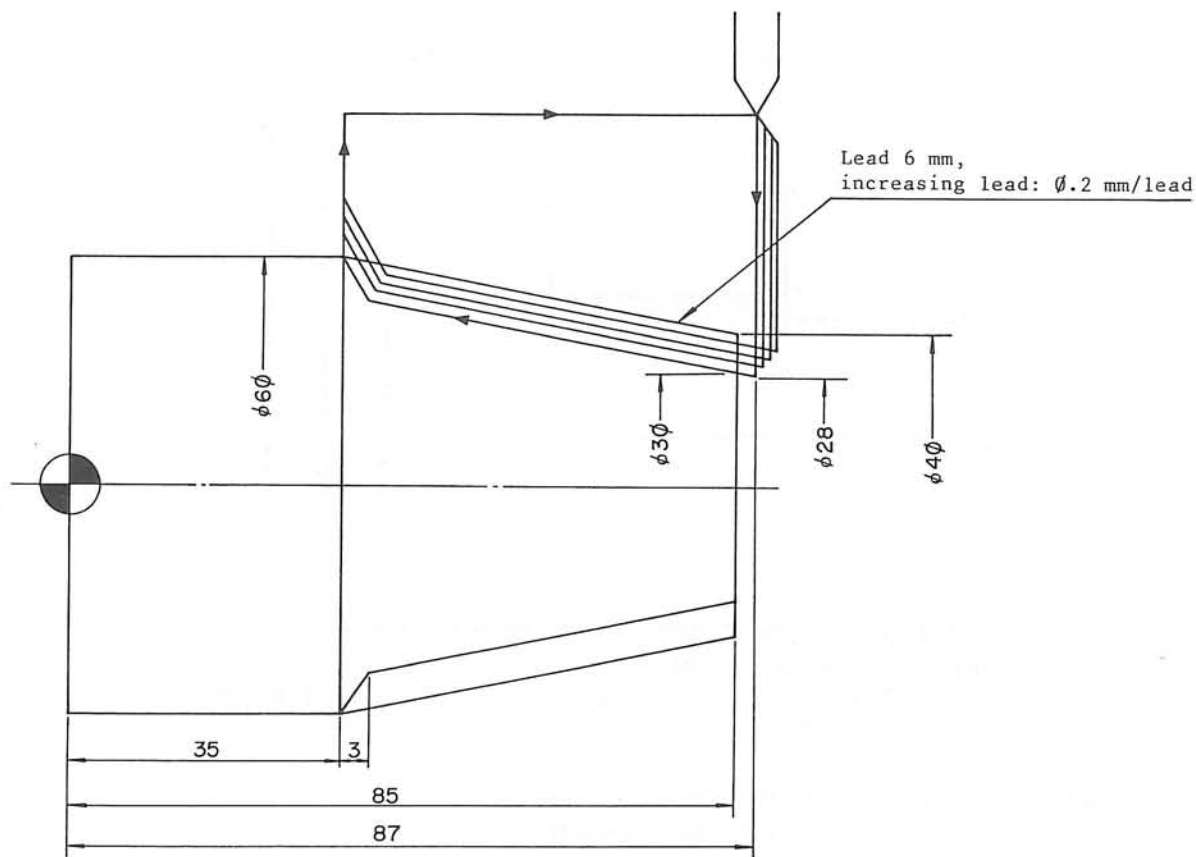
- Q3: Positioning of Z-axis at a position designated by a K word at a rapid feedrate.
- Q4: Tapping from the point reached in Q3 to the depth specified by a Z word at a specified feedrate (F).
- Q5: The spindle stops once and then starts in the forward direction at the same speed as used in infeeding.
- Q6: Z-axis retraction to a point reached in Q4 cycle at a cutting feedrate.
- Q7: Z-axis retraction to a point reached in Q3 cycle at a rapid feedrate.

13-4-3. 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.

13-5. APPLICATION OF COMPOUND FIXED CYCLE

13-5-1. Application of Longitudinal Thread Cutting Compound Fixed Cycle

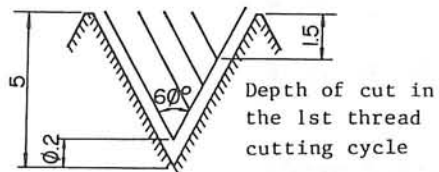


M32 + M75:

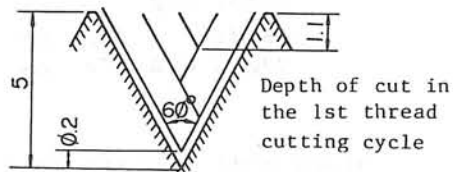
N0001 G71 X28 Z35 I11 B60 D1.5 U0.2 H5 L3 E0.2 F6
 \$ M23 M32 M75

M33 + M74:

N0001 G71 X28 Z35 I11 B60 D1.1 U0.2 H5 L3 E0.2 F6
 \$ M23 M33 M74

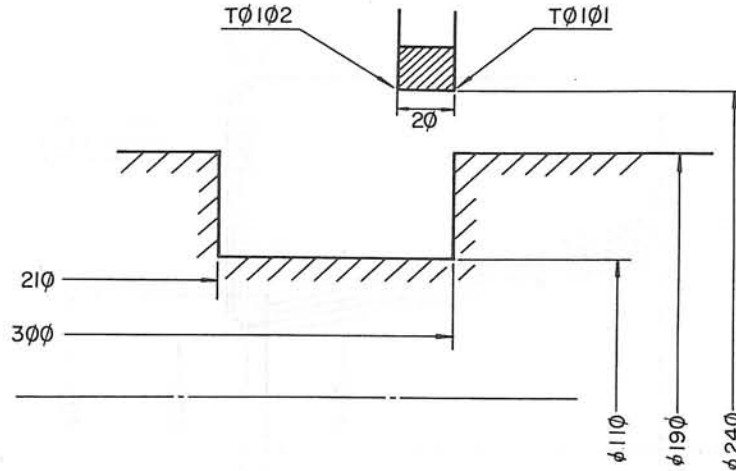


M32 + M75



M33 + M74

13-5-2. Application of Longitudinal Grooving Compound Fixed Cycle (G73)

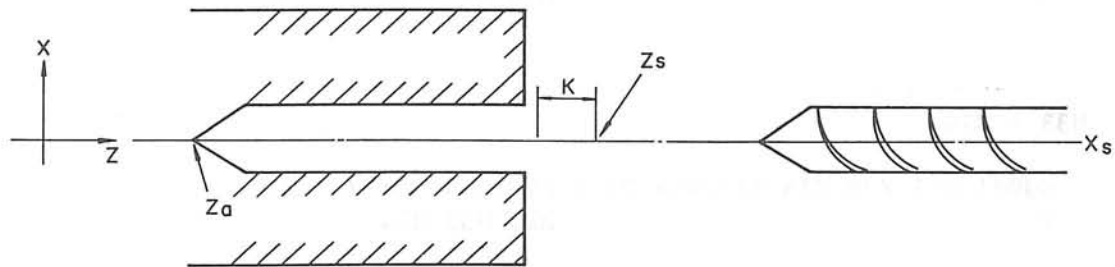


```

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

Note: (Z-axis tool offset amount of #2)
 - (Z-axis tool offset amount of
 = 2φ

13-5-3. Application of Transverse Grooving Compound Fixed Cycle (G74)
 - Drilling Cycle -



```

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

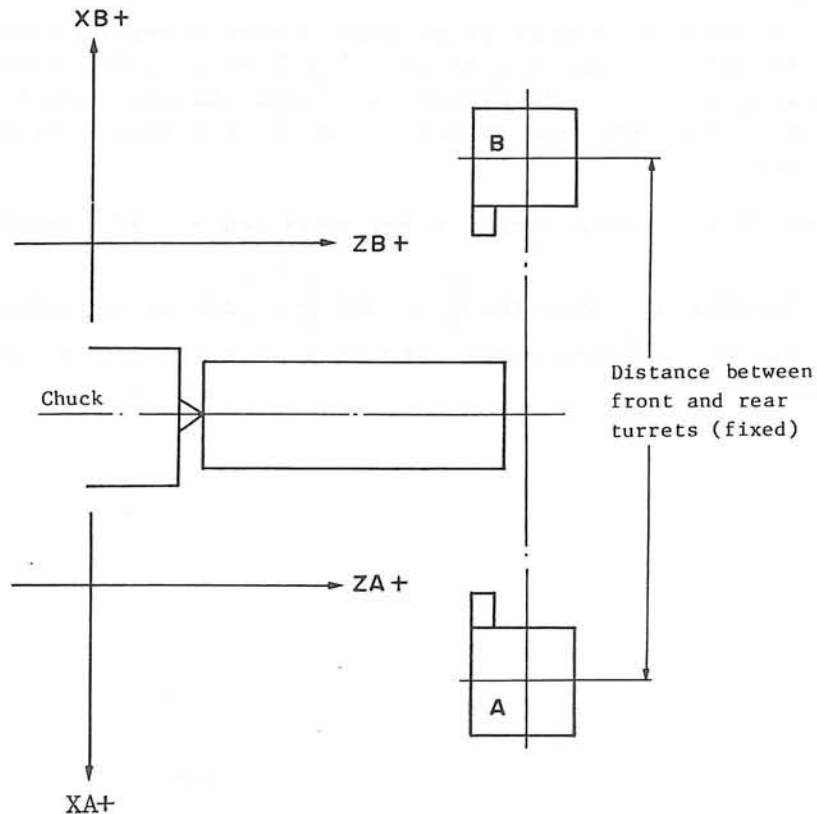
Note: G74 block must contain both X and Z words.

SECTION 14 MIRROR IMAGE FUNCTION (2-Turret Model)

14-1. 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.

As the number of the tools to be mounted on the turret (B), the CNC lathe having two turrets, rear and front, is becoming more popular.



As shown above two turrets, rear (B) and front (A), 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 distance indicated above.

It is the Mirror Image function that can simplify programming for two-turret model and help the machine to demonstrate its best performance.



14-2. OPERATIONS

When making a part program for two-turret model, there lies a program - although the conventional programming technique applies to the program for turret A, how to the program is to be prepared when turret B is selected.

The Mirror Image function for X-axis can solve the difficulties when programming for cutting cycle with turret B. For the machine equipped with this function, the programmer should prepare a part program only taking the part contour into consideration and disregarding the turret to be selected.

For instance, if the programmer programmed a part assuming turret A, the mirror image function automatically converts X-axis moving direction, spindle rotation direction and direction of arc cutting when turret B is selected. Therefore, the same workpiece can be cut when turret as finished by turret A tools.

Details of the Mirror Image function are provided in this section.

The main differences are that the  and  are used to select the turret, and that G13 and G14 codes used to select the turret are necessary in a program.

14-3. PROGRAM AXIS MOTIONS

This section deals with the actual programming procedures and axis motions resulted from such programmed commands.

14-3-1. 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

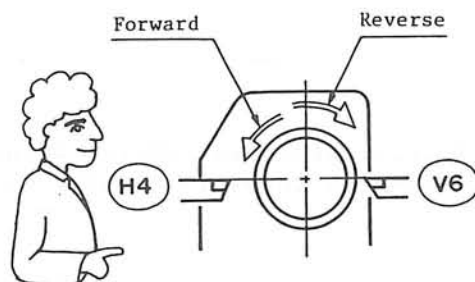
When selecting #4 tool on turret B, program:

G14					LF
G00	X	Z	T0404		LF

When selecting #3 tool on turret A, program:

G13					LF
G00	X	Z	T0303		LF

(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.

14-3-2. 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, NC alarm "G13/G14 CHANGE" 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 active when G13 and G14 are switched remain active since they are stored when executed.
- (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  X2  Z2  M03
N003  G13
N004  G00  X2  Z2  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 NC alarm "M03/M04 CHANGE".

- (6) When a programmed cycle is to be repeatedly executed using "GOTO" statement, care should be exercised on the following point:

<Program Example>

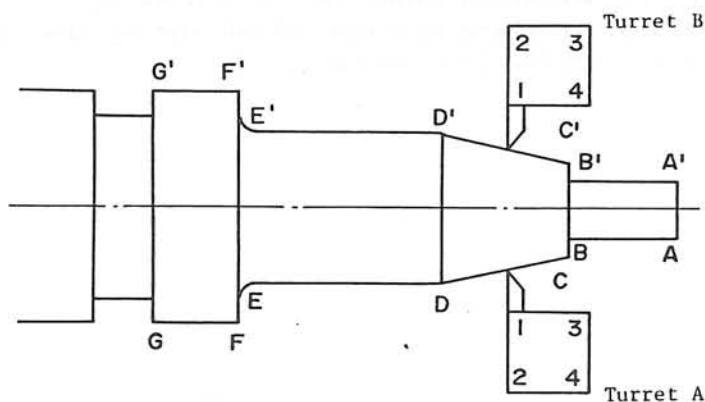
```

N001 G00 X Z1
N002 G01 X2 Z2 F.S.T.M
N003 G14
N004 G00 X3 Z3
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.

14-3-3. 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 LF
N003 G01 ZB F0.2 M03 LF
N004 XC LF
N005 XD ZD LF
N006 ZE LF
N007 G02 XF ZF I K LF
N008 G01 ZG LF
N009 G00 XH ZH M05 LF

```

G14 is necessary when turret B is used.

14-3-4. 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 14-3-3.



(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 14-3-3. For arc cutting called for by the commands in N007, direction of cutting is also reversed.

14-4. MACHINE OPERATION AND DISPLAY

14-4-1. Operation Panel



Selection of the turrets, A and B, for setting and display of actual position, zero offset and tool offsets can be made by pressing the  or  key.

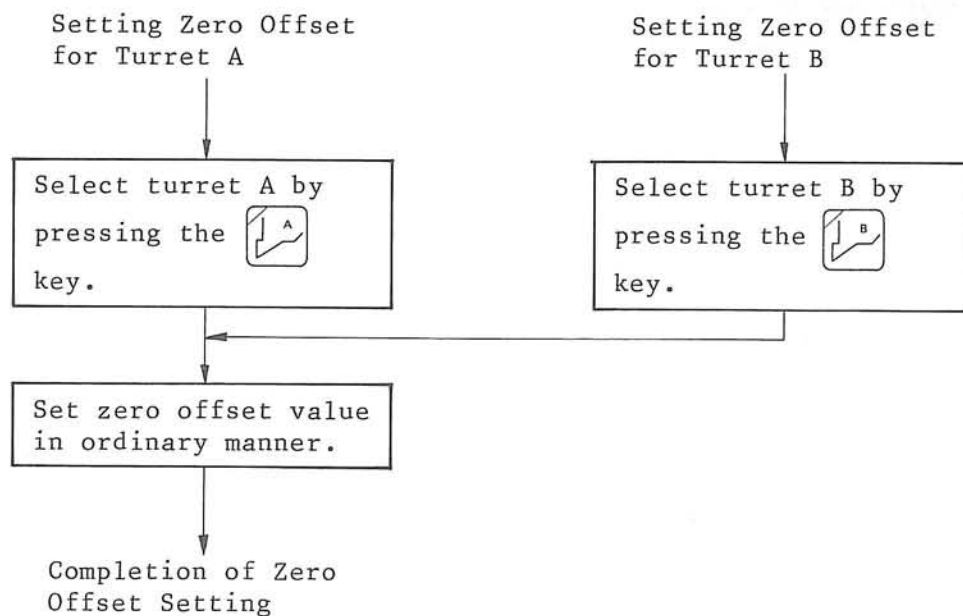
Display of Mirror Image:

While the mirror image function is active, message "MIRROR IMAGE" appears at the top of the CRT.

14-4-2. Zero Offset

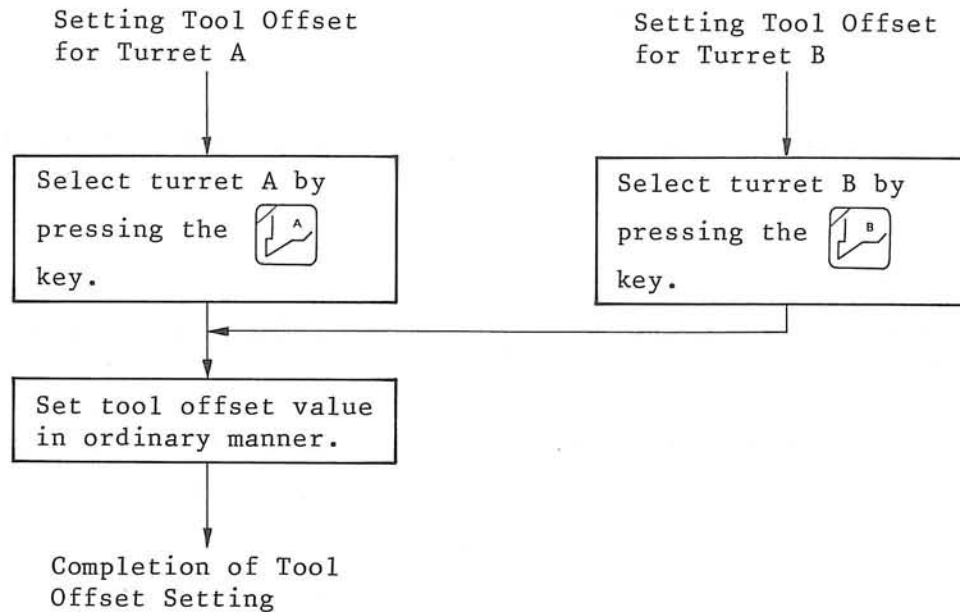
For details of Zero Offset, refer to the Operation Manual for OSP500L-G/OSP500L-G.

For mirror image, select turret A or B first by using the  and  keys. After that, set the zero position for respective turrets.



14-4-3. Tool Offset

For setting tool offset, also refer to the Operation Manual for OSP500L-G/OSP5000L-G as in setting zero offset. Selection of the turret is made using the proper turret selection key as in setting zero offset.



14-5. 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, model LC and LR, 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  X1  Z1  T0101
N003  G14
N004  G00  X2  Z2  T0101
N005  G13
N006  G00  X3  Z3
N007      X4  Z4  T0202
N008  G14
N009  G00  X5  Z5
N010      X6  Z6

```

When sequence re-start is activated from N007, T1 on turret A is indexed since block N006 is in G13 mode.

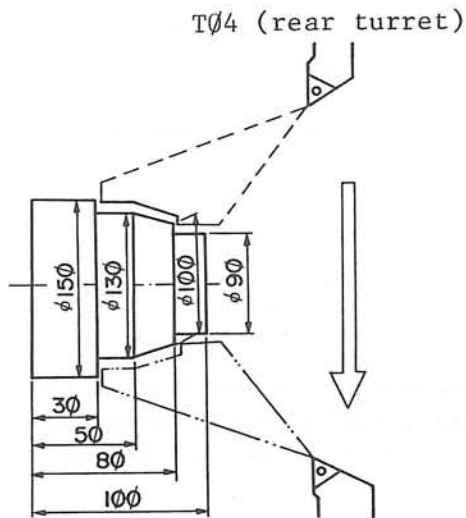
When sequence re-start is activated from N010, T1 on turret B is indexed since block N009 is in G14 mode.

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.

CAUTION

Turret indexing position is different from the programmed position.

14-6. PROGRAM EXAMPLE

Programming Example with T04 Tool
on Rear Turret (LH35-N)

N	G	X	Z	I,K	F	S,T,M	LF
%							LF
O1000							LF
N001	G14						LF
N002	G00	X335 (275)	Z200			S400 T0404	LF
N003		X90	Z103			M08 M03	LF
N004	G01		Z80		F20		LF
N005		X100					LF
N006		X130	Z50				LF
N007			Z30				LF
N008		X155					LF
N009	G00	X335 (275)	Z200			M05 M09	LF
N010						M02	LF

Note: For LS-N, program the dimensions in () in N002 and N009.

- (1) Program assuming T04 is on the front turret.
- (2) Select the turret by G14.
- (3) Spindle rotation command must be M03.
- (4) Program should be made in ordinary manner.

Other cautions:

- (1) Select the turret by G13 and G14 for arc and thread cutting operations as in straight line and taper cutting operation.
- (2) G13 and G14 must be programmed in a block without other commands.

N001	G14							LF
N002	G00	X335	Z300			S400 T0404	M03	LF
N003		X100	Z50					LF

	G00	X335	Z300			M05		LF
N050*	G13							LF
N051*	G00	X270	Z300			S400 T0303	M03	LF

* If T0303 in N051 is not programmed, the tool number and tool offset number effective in the preceding G13 mode automatically becomes effective.

* When N, G, F, S and/or M command is not programmed, that programmed previously becomes effective. (X and Z words become ineffective.)

- (3) When neither G13 nor G14 is commanded, the control automatically selects turret A.
- (4) Turret indexing can be made at any desired position both on X- and Z-axis.

Select the turret index position where interference will not occur.

- (5) If G13 or G14 calling for turret change is programmed while incremental programming mode, constant cutting speed mode, LAP mode and tool nose radius compensation mode, an alarm results and machine operation stops.

SECTION 15 USER TASK 1

15-1. OUTLINE OF USER TASK FUNCTION

After years of work in data processing, we at Okuma are now able to provide our customer with the most efficient use of high data processing capacity in OSP500L-G/OSP5000L-G. We have accomplished this in the development of User Task Function.

The biggest advantage of the user task function is that various operation functions and variables can be used. In addition, the use of control statements assure versatility in the user task function.

User task function has two kinds:

- 1) User Task 1 (standard)
- 2) User task 2 (optional)

"User Task 1" as detailed in this manual, permits the use of arithmetic operations, variables, and control statements. However, User Task 1 is mainly provided to enable continuous automatic operation by controlling peripheral equipment such bar feeders, loaders and robots. We have also included functions to control scheduled machine operations which allow various workpieces to be machined according to the programmed schedule.

"User Task 2" is the authentic user task function with arithmetic operations, variables and all control statements. Functions available with User Task 1 are all contained in User Task 2.

15-2. 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" and "GOTO" which are easily understood, can be used.

(2) Variables

In normal programming, numerical data directly follows the address characters such as X, Y, and 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

X 1 3 5

When a variable is used:

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:

X 1 3 5

When arithmetic operation
function is used:

X = 100 + XP2 XP2 = 35

Provided in later pages are the detailed descriptions of respective fundamental user task functions.

15-2-1. Control Statement

User Task 1 can use following two control statements:

IF
GOTO

Program these control statements either at the beginning of a block or right after a sequence name designated at the beginning of a block. 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 results.

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
NLAP1 GOTO NLAP2

A space or a tab code

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.

Note 1: Sequence Name

A sequence name is the code to identify respective blocks in a program, and it consists of four alphanumeric characters following address character N.

There are two types of sequence name:

<N> <four alphanumeric>, and
<N> <letter> <three alphanumeric>

"Sequence name" handled in this manual refers to both types of sequence names.

15-2-1-1. GOTO Statement - Unconditional Jump

(1) Program Format

<N0> <GOTO> <N1>

Sequence name of the block where jump is to be made and it must be programmed without fail.

Indicates "GOTO" statement.

Sequence name of this block and it can be omitted.

Note: Sequence name <N1> for jump must be the one in the program containing such control statement.

(2) Function

When the control statement is executed, jump to the programmed block, <N1> is made unconditionally.

Note: On two-saddle models, jump from the program for A saddle to the program for B saddle or vice versa is not allowed.

15-2-1-2. IF Statement - Conditional Jump

(1) Program Format

a) <N0> <IF> <Conditional expression (qualification)> <GOTO> <N1>

This is the <sequence name> to which jump is made when <qualification> is true. This must be programmed without fail.

Jump instruction to be executed when <qualification> is true. May be omitted.

There are two possible states for the result of the <qualification>, "true" and "false".

<qualification> must be expressed in [].

Indicates "IF" statement.

Sequence name of this block and it can be omitted.

b) $\langle N0 \rangle _ \langle IF \rangle _ \langle \text{Local variable} \rangle _ \langle GOTO \rangle _ \langle N1 \rangle$

The $\langle \text{sequence} \rangle$ name of the destination block of jump to be executed when the local variable is defined.

Jump instruction to be executed when the local variable is defined.

This can be used instead of $\langle \text{qualification} \rangle$. Jump does or does not occur depending on whether the local variable is defined or not.

(2) Function

- a) When the $\langle \text{qualification} \rangle$ is true (Example 1) or when the local variable is defined (Example 2), execution of the sequence jumps to the sequence $\langle N1 \rangle$.
- b) When the $\langle \text{qualification} \rangle$ is false (Example 1) or when the local variable is not defined (Example 2), the following sequence is executed.

Example 1:

$N1000 _ IF [V1 _ EQ _ 10] _ N2000$ 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.

15-2-2. Common Variable

There are two types of variables used with User Task 1:

- Common variable
- Local variable
- System variable

These three types of variables differ in their use and characteristics.

15-2-2-1. Common Variable

The term "common" of common variables can be literally understood as common; they are common to main and sub programs. When the same variable is used in two or more programs, the variable number used in those programs must be identical. Therefore, the common variable, the result of calculation in one program, can be referred to from other programs.

(1) Program Format

<V> <one or two digits> = numerical data or expression

Designation of a common variable is made with up to two digits following "V". Usable common variable is V1 through V32.

In a program, a common variable is used as:

$$N101\text{V}5 = 10, \text{ or } N101\text{V}5 = V5 + 1$$

Note 1: Common variables are used in common to A-saddle and B-saddle.

Note 2: Common variables are effective both in a main and a sub program.

Note 3: Common variables are not affected by control reset or power turning off operation. That is, the data are retained unless they are reset or a control program tape is loaded.

Note 4: 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 Instruction Manual for OSP500L-G/OSP500L-G.

15-2-2-2. Local Variables

As may be known from 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 for A-saddle and B-saddle, respectively.

(1) Program Format

<Letter><Letter><two alphanumerics> = Numerical data or expression

← "O", "N" and "V" cannot be used.

Example: 'DIA1' 'ITH5'

The same name as relative operators, detailed in 15-2-3, and extended address characters(*1) cannot be used.

*1: Extended address characters are used for LAP, pattern processing, user fixed cycle and other extended functions which cannot be covered with conventional address characters A through Z. Extended address characters are:

<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>,

(2) Characteristics of Local Variables

- a) Local variables can be used independently on A-saddle and B-saddle, and are cleared when the control is reset.
- b) 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.

If a local variable name is used without assigning the data, an alarm results.

- c) 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 DIA2 = 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".

- d) Up to 127 local variables can be used on A-saddle and B-saddle, respectively.

15-2-2-3. System Variables

System variable means the variable specific to respective systems and its name is fixed.

The system variables are set for respective turrets independently and 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
- Tailstock barrier variable
- Droop 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 the Operation Manual for OSP500L-G/OSP5000L-G.

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
 <VZOFW> Zero Offset of W-axis
 (only for programmable tailstock specification)
 <VZOFC> Zero Offset of C-axis
 (only 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
 <VXSHX> Zero Shift of C-axis
 (only 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

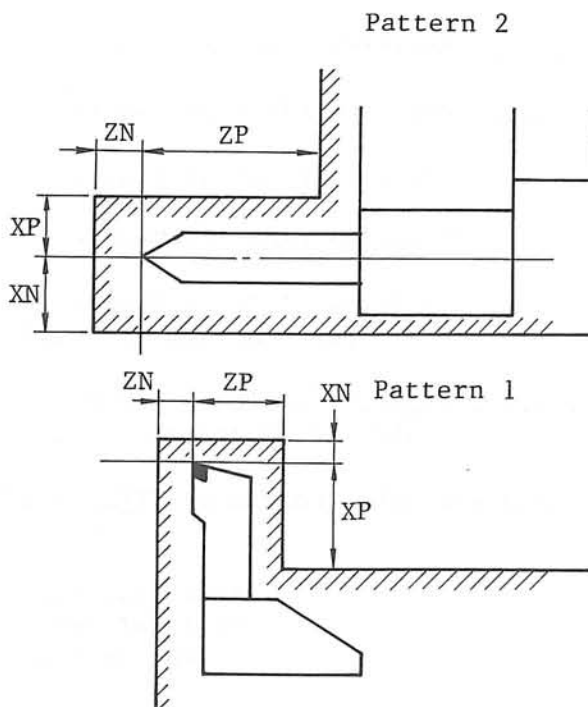
<VNSRX> [Nose R Compensation No.]

NoSe Radius compensation of X-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) Tool interference data



Variable name

<VTIZN> [Tool No.] Tool Interference on Z-axis at Negative end; ZN

<VTIZP> [Tool No.] Tool Interference on Z-axis at Positive end; ZP

<VTIXN> [Tool No.] Tool Interference on X-axis at Negative end; XN

<VTIXP> [Tool No.] Tool Interference on X-axis at Positive end; XP

<VTIPN> [Tool No.] Pattern Number

Note: For the control equipped with the graphic specification, tool interference data is automatically determined from the projection amount, tool diameter set on the TOOL SHAPE selection page and the tool interference variable data. Therefore, if tool interference data is changed using the tool interference data variables, the data is not effective.

f) 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

<VPVLW> Positive Limit on W-axis
(only for programmable tailstock specification)

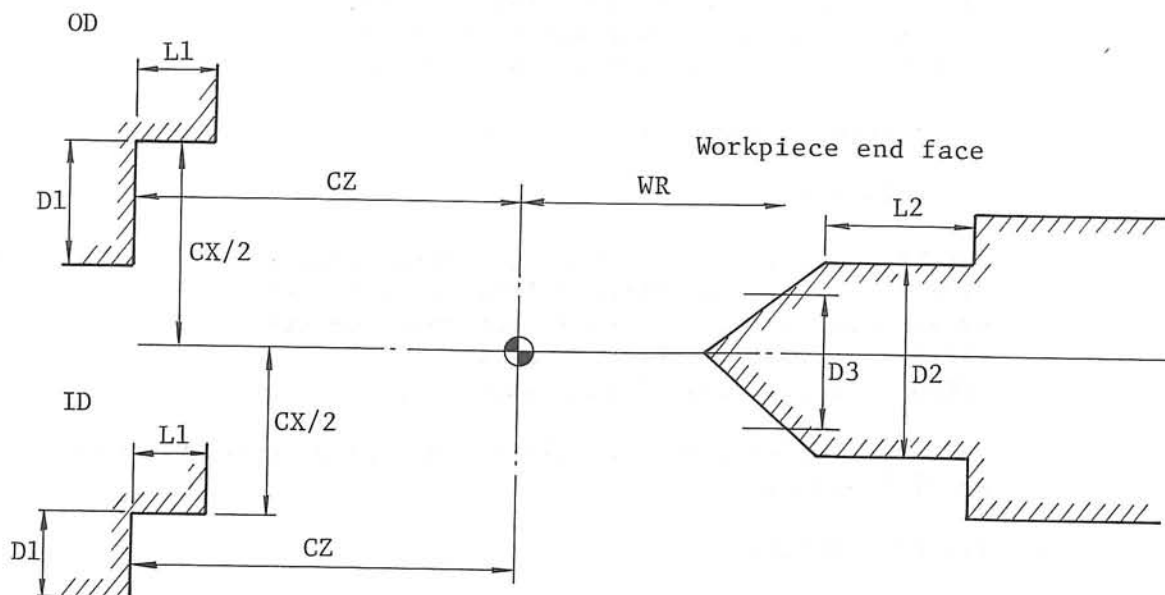
<VNVLW> Negative Limit on W-axis
(only for programmable tailstock specification)

When setting a variable program as VPVLZ = 2352.168.

This indicates that variable soft-limit of Z-axis in positive direction is set at Z = 2353.168 mm.

Numerical data of these variables are referenced to the origin of the machine coordinate system (machine origin).

g) Chuck barrier and 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
<VTSDA>	Center position D3
<VWKR>	Workpiece end face

These variables should be set in reference to be programming zero.

h) Droop variable

Variable name

<VINPZ> IN Position of Z-axis
 <VINPX> IN Position of X-axis
 <VINPC> IN Position of C-axis

i) Pitch error compensation variable

Variable name

<VPFVZ> Pitch Fillup Value Z-axis
 <VPFVX> Pitch Fillup Value X-axis
 <VPFVC> Pitch Fillup Value C-axis
 <VPCHZ> PitCH Z-axis
 <VPCHX> PitCH X-axis

These variables are used only for the pitch error compensation specification.

j) 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

k) Alarm comment variable

Variable name

<VUACM> User Alarm CoMment

User alarm comments up to 16 characters can be designated.

VUACM[1] - VUACM[16]

Example:

N202 VUACM[1] = 'ABC'

N203 VUACM[4] = '123'

N204 VDOUT[991] = 999

:

An alarm with the comment is generated as indicated below when the program shown above is executed.

Alarm C 909

User reserve code 999 ABC=123

VDOUT[*] is an output variable and level of alarm to be generated is controllable by the number to be set at *.

*=991: Alarm C

*=992: Alarm B

*=993: Alarm A

15-2-3. 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>, <Variables>
= <Expression>

The expression in the right-hand side, requesting arithmetic operation, is made from constants, variables, comparison expression, and operators.

15-2-3-1. 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=V1Ø * 1Ø	
/	Quotient (division)	X=V11 / 1Ø	

15-2-3-2. Comparison Expression

Operator	Meaning	Example	Contents	Rule
LT	(<u>L</u> ess <u>T</u> han, <)	IF [V1 <u>L</u> LT <u>L</u> 5] <u>L</u> N1ØØ	Jump to N1ØØ when V1 is less than 5.	Provide a space on either side of the operator.
LE	(<u>L</u> ess than or <u>E</u> qual to, ≤)	IF [V1 <u>L</u> LE <u>L</u> 5] <u>L</u> N1ØØ	Jump to N1ØØ when V1 is less than or equal to 5.	
EQ	(<u>E</u> qual to, =)	IF [V1 <u>L</u> EQ <u>L</u> 5] <u>L</u> N1ØØ	Jump to N1ØØ when V1 is equal to 5.	
NE	(<u>N</u> ot <u>E</u> qual to, ≠)	IF [V1 <u>L</u> NE <u>L</u> 5] <u>L</u> N1ØØ	Jump to N1ØØ when V1 is not equal to 5.	
GT	(<u>G</u> reater <u>T</u> han, >)	IF [V1 <u>L</u> GT <u>L</u> 5] <u>L</u> N1ØØ	Jump to N1ØØ when V1 is greater than 5.	
GE	(<u>G</u> reater than or <u>E</u> qual, ≥)	IF [V1 <u>L</u> GE <u>L</u> 5] <u>L</u> N1ØØ	Jump to N1ØØ when V1 is greater than or equal to 5.	

15-3. PROGRAM EXAMPLE

A simple program example is provided below to help you understand contents of User Task 1.

Assume the NC lathe is equipped with the bar feeder. Three types of workpieces are machined on the NC lathe according to the programmed schedule.

- (1) Assign respective workpieces with file name and program name (or no.):

```
Workpiece A ..... A. MIN, 0100
Workpiece B ..... B. MIN, 0200
Workpiece C ..... C. MIN, 0300
```

- (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 order of them - scheduling.

```
No. of workpieces:  A ..... 20 pcs.
                   B ..... 15 pcs.
                   C ..... 25 pcs.
```

Sequence of machining: A, B and C

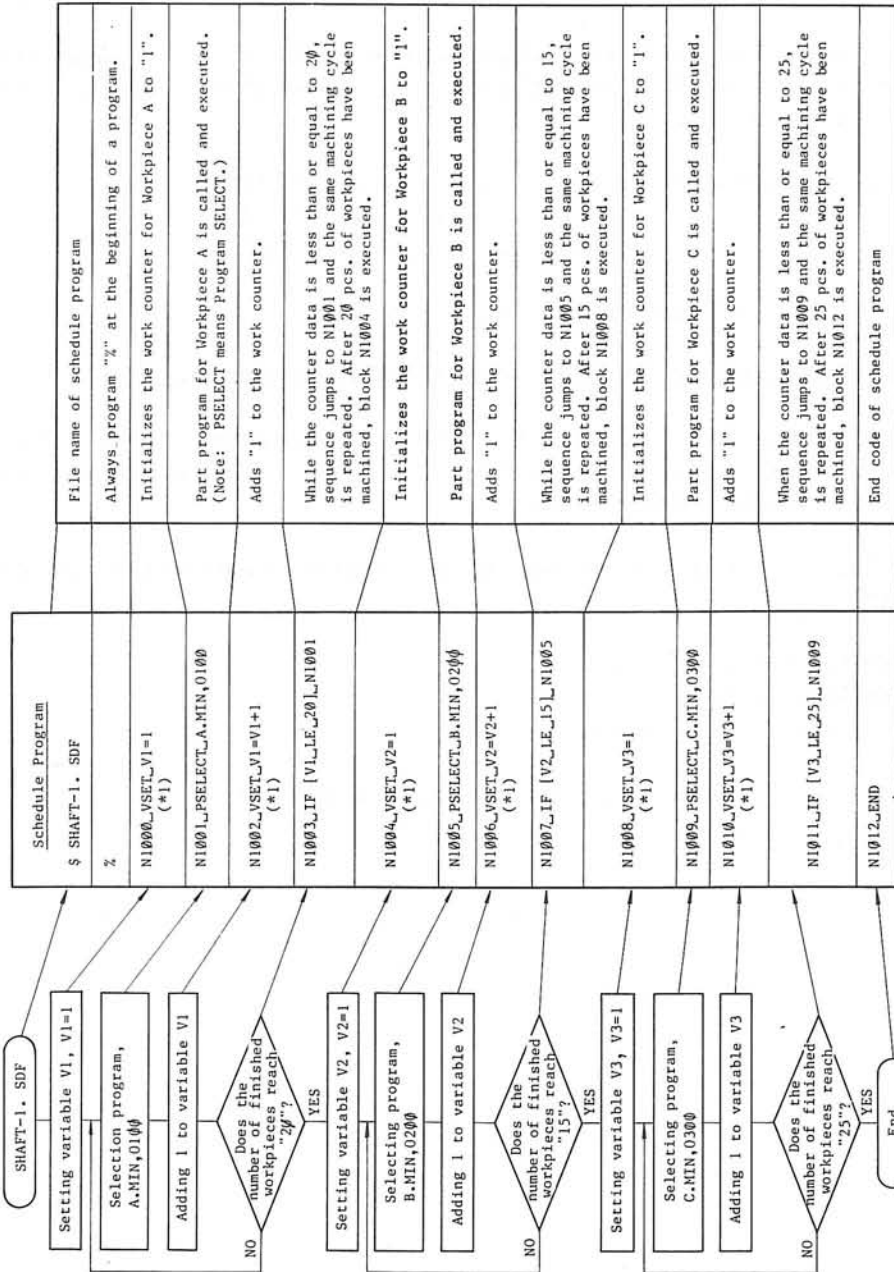
- (4) Now the programmer can make the schedule program.

Determine the file name of the schedule program, first; the file name of this example is determined as "SHAFT-1. SDF" since the intended machining is of shaft work.

- (5) Counting of the finished workpieces can be made using the common variable.

```
Workpiece A ..... V1
Workpiece B ..... V2
Workpiece C ..... V3
```

- (6) Prepare the schedule program according to the flow chart in the following page.



Main Program (Machining Program)	
A.MIN,0100	B.MIN,0200 X.MIN,0300
Z	Z
N100 G90 X Z STM N200 G90 X Z STM N300 G90 X Z STM	
:	:
:	:
:	:
:	:
N150 M02	N230 M02 N340 M02

Schedule Program	File name of schedule program
\$ SHAFT-1. SDF	Always program "Z" at the beginning of a program.
%	Initializes the work counter for Workpiece A to "1".
N1000_VSET_V1=1 (*1)	Part program for Workpiece A is called and executed. (Note: PSELECT means Program SELECT.)
N1001_PSELECT_A.MIN,0100	Adds "1" to the work counter.
N1002_VSET_V1=V1+1 (*1)	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.
N1003_IF [V1_LE_20]_N1001	Initializes the work counter for Workpiece B to "1".
N1004_VSET_V2=1 (*1)	Part program for Workpiece B is called and executed.
N1005_PSELECT_B.MIN,0200	Adds "1" to the work counter.
N1006_VSET_V2=V2+1 (*1)	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.
N1007_IF [V2_LE_15]_N1005	Initializes the work counter for Workpiece C to "1".
N1008_VSET_V3=1 (*1)	Part program for Workpiece C is called and executed.
N1009_PSELECT_C.MIN,0300	Adds "1" to the work counter.
N1010_VSET_V3=V3+1 (*1)	When 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.
N1011_IF [V3_LE_25]_N1009	End code of schedule program
N1012_END	

(*1): To set a common variable in the schedule program, be sure to program VSET command which declares variable setting.

15-4. COMPARISON OF USER TASK 1 AND USER TASK 2

Function and Contents	User Task 1	User Task 2
Usable Programs	Main program Schedule program	Main program Subprogram Schedule program System subprogram
Control Statement Function	<GOTO statement> <IF statement>	<GOTO statement> <IF statement> <CALL statement> <RTS statement> <MODIN statement> <MODOUT statement> <GET/PUT statement> <READ/WRITE statement>
Variable Function	Common variables Local variables System variables	Common variables Local variables System variables I/O variables
<u>Operation Function</u>		
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>, <BCO>, <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>

SECTION 16 TAPE CONVERSION FUNCTION (OSP3000L → OSP500L-G/OSP5000L-G)

16-1. OUTLINE

This function is used to convert programs prepared for OSP3000L and stored in a bubble memory into those applicable to OSP500L-G/OSP5000L-G.

Items converted are:

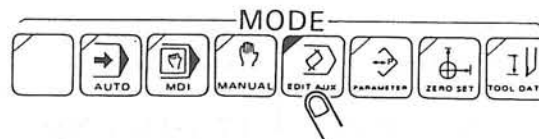
- (1) Simultaneous 4-axis control program
- (2) Constant speed cutting program
- (3) Address characters specifying variable pitch
- (4) Endless operation
- (5) Nose radius compensation (end up) function
- (6) LAP

16-2. OPERATION

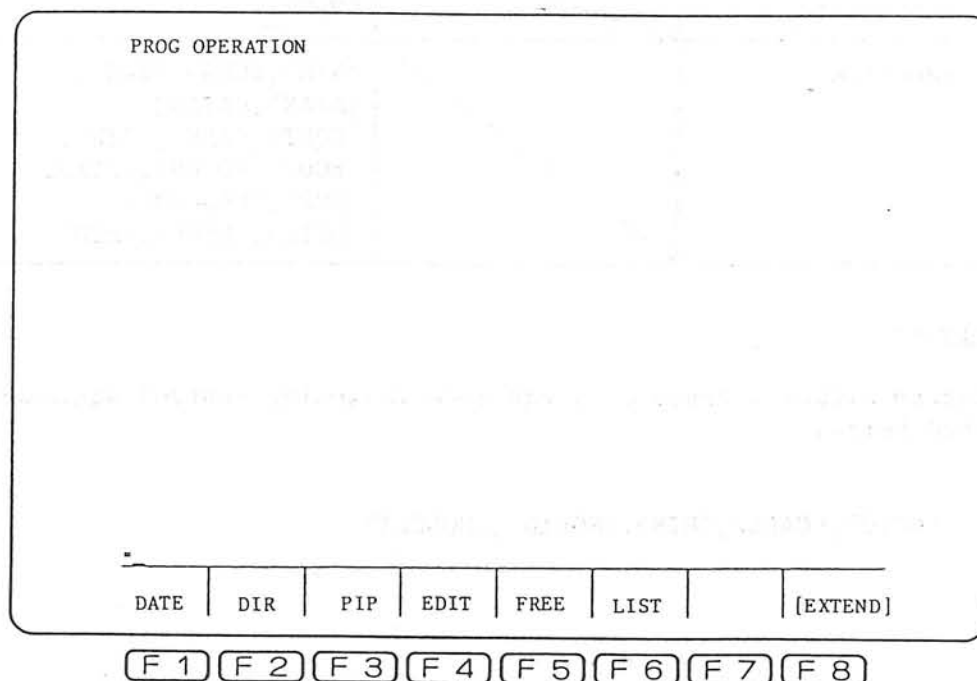
For the store programs prepared for OSP3000L, refer to the instructions provided in 4-2-1 (1) of the Operation Manual for OSP500L-G/OSP5000L-G.

Conversion of the program can be made in the following procedure:

- 1) Press the EDIT AUX key under MODE.



- 2) The CRT display is shown below:



- 3) Press the function key [F8] (EXTEND) two times and the CRT display on the function line changes as shown below:

PROG OPERATION							
= EX							
= EX							
= _							
TAPE CONVERT							[EXTEND]
[F 1]	[F 2]	[F 3]	[F 4]	[F 5]	[F 6]	[F 7]	[F 8]

- 4) Press the function key [F1] (TAPE CONVERT).
The CRT shows "C=".
- 5) Enter "input-file-name" and "output-file-name" through the keyboard.
<input-file-name>,<output-file-name>
- 6) Press the WRITE key. The CRT display changes as shown below:

PROG OPERATION CONVERT							
[OUTPUT FILE NAME]							
= EX							
= EX							
= C [INPUT FILE NAME], [OUTPUT FILE NAME]							
CONVERT 3000 to 5000 (Y/N) ! _							
TAPE CONVERT							[EXTEND]
[F 1]	[F 2]	[F 3]	[F 4]	[F 5]	[F 6]	[F 7]	[F 8]

- 7) When conversion is to be made, key in "Y" then press the WRITE key. When conversion is not required, key in "N" instead of "Y" then press the WRITE key.

CAUTIONS

- (1) When only "input file name" is designated, a program for OSP500L-G/OSP5000L-G assigned with the same file name is created and the program for OSP3000L is erased.
- (2) For other cases, designate both "input-file-name" and "output-file-name".
- (3) When the input file assigned with the designated file name is not found, message "file not found" appears on the CRT and tape conversion operation terminates.
- (4) If the output file assigned with the file name presently designated already exists, message "file exist over write (Y/N)!" appears. Pressing the WRITE key after keying in "Y" erases the old file and tape conversion is executed.

Pressing the WRITE key after keying in "N" terminates the tape conversion operation without executing conversion.

- (5) To check the results of the converted program(s), refer to the instruction in 4-2-3 (4) of the Operation Manual for OSP500L-G/OSP5000L-G.
- (6) When a new command value to be generated in tape conversion process cannot be calculated, "?" is entered at the location where such new data is to be entered.

16-3. CONTENTS OF CONVERSION

16-3-1. Simultaneous 4-Axis Control Program

- (1) Simultaneous 4-axis control part
 - a) M codes calling for start and end of simultaneous 4-axis control, M11 and M10, are deleted.
 - b) For Program Stop and Optional Stop M codes, M00 and M01, P code with the numeral value "1" larger than the value of the P code just preceding it is added for respective turrets to synchronize execution of these M codes.
 - c) P codes already designated for respective turrets are renewed in the manner that the synchronization is maintained.

(2) Independent control part

- a) For G codes selecting turret, G13 and G14, P code with the numeral value "10" larger than the value of the P code just preceding it is added.
- b) The first P code, appearing in the independent cutting mode after completing simultaneous 4-axis control, is assigned with the numeral value "10" larger than the value of the P code having the largest value in the preceding simultaneous 4-axis control program.

Note 1: G13 and G14 should be designated in one block without other commands.

Note 2: M00 and M01 should be designated for both turrets A and B.

Example:

<u>OSP3000L</u>		<u>OSP500L-G/OSP5000L-G</u>	
N001 M11	CR	N001	CR
N002 G13	CR	N002	CR
N003 G00 X1 Z1 P1 M1 T1 S1	CR	N003 G00 X1 Z1 P10 M1 T1 S1	CR
}		}	
N014 M01	CR	N014 M01 P11	CR
N015 G14	CR	N015 G14	CR
N016 G00 X1 Z1 P1 M1 T2 S1	CR	N016 G00 X1 Z1 P10 M1 T2 S1	CR
}		}	
N024 M01	CR	N024 M01 P11	CR
N025 M10	CR	N025	CR
N026 G14	CR	N026 G14	CR
N027 G00 X2 Z2 T3 S2 M2	CR	N027 G00 X2 Z2 T3 S2 M2 P21	CR

16-3-2. Constant Cutting Speed Program

In simultaneous 4-axis control program, the turret on which constant cutting speed (CCS) is effective is determined by the designated turret selection command effective just before entering simultaneous 4-axis control and G code indicated below is added following G96 calling for the CCS operation.

G110 CCS on turret A (G13)
 G111 CCS on turret B (G14)

Example:

<u>OSP3000L</u>					<u>OSP500L-G/OSP5000L-G</u>			
N001	M10		CR		N001			CR
N002	G13		CR		N002	G13		CR
	{					{		
N009	M11		CR		N009			CR
N010	G13		CR		N010	G13		CR
N011	G00	X1 Z1 P10 S1 T1 M1	CR		N011	G00 X1 Z1 P10 S1 T1 M1		CR
N012	G96	X2 Z2 P20 S2	CR	→	N012	F96 G110 X1 Z2 P20 S2		CR
	{					{		
N020	M14		CR		N020	G14		CR
N021	G00	X3 Z3 P10 S1 T1 M1	CR		N021	G00 X3 Z3 P10 S1 T1 M1		CR
N022	G96	X4 Z4 P20 S2	CR		N022	G96 G110 X4 Z4 P20 S2		CR
	{					{		
N030	M10		CR		N030			CR

16-3-3. Address Characters Specifying Variable Pitch

The address character used to command pitch variation amount per pitch in cutting variable pitch thread is converted from "K" to "E".

Example:

<u>OSP3000L</u>					<u>OSP500L-G/OSP5000L-G</u>			
N001	G00	X1 Z1	CR		N001	G00	X1 Z1	CR
N002	G34	X2 Z2 K F	CR		N002	G34	X2 Z2 E F	CR
N003	G34	Z3	CR		N003	G34	Z3	CR
	{					{		
N007	G00	X7 Z7	CR		N007	G00	X7 Z7	CR
N008	G33	X8 Z8 K F	CR		N008	G33	X8 Z8 E F	CR

16-3-4. Endless Operation

In tape conversion, code M31 is converted into code M02.

Example:

<u>OSP3000L</u>					<u>OSP500L-G/OSP5000L-G</u>			
N001	G13		CR		N001	G13		CR
	{					{		
N009	M31		CR	→	N009	M02		CR

16-3-5. Nose Radius Compensation Function

For G40 G00 calling for end up of the nose radius compensation function programmed for the OSP3000L, imaginary point (I, K) is created from X2, Z2 commanded in G40 G00 block and X1, Z1 provided in the preceding block with tape conversion.

$$I = X2 - X1$$

$$K = Z2 - Z1$$

Example:

OSP3000L		OSP500L-G/OSP5000L-G
N009 G01 X1 Z1 CR	→	N009 G01 X1 Z1 CR
N010 G40 G00 X2 Z2 CR		N010 G40 G00 X2 Z2 I K CR

Note: G40 and G00 must be designated in the same block.

16-3-6. LAP

- (1) The address character "L" expressing stock amount is converted into "U" or "W": "U" in longitudinal cutting and "W" in transverse (end face) cutting.
- (2) In a program for OSP3000L, cutting conditions for roughing cycle are specified along with the commands specifying final contour. For OSP500L-G/OSP5000L-G, contour defining commands containing both final contour and rough cut conditions are separately provided from the finish cut conditions; in actual cutting stage, finish cut conditions are called for from the contour defining commands.

Example:

OSP3000L		OSP500L-G/OSP5000L-G
N001 G00 Xt Zt CR	→	N001 G00 Xt Zt CR
N002 Xs Zs Fr Sr Tr Mr CR		N002 Xs Zs Fr Sr Tr Mr CR
N003 G85 CR		N003 G85 NLA01 U D CR
N004 G00 Xt Zt L (U W) D CR		N004 G00 Xt Zt CR
N005 Xa Za Ff Sf Tf Mf CR		Sf Tf Mf CR
N006 G01 Xb Zb CR		G87 NLA01 CR
N007 Xc Zc CR		NLA01 G81 CR
N008 G80 CR		N005 Xa Za Ff CR
N009 G00 Xt Zt CR		N006 G01 Xb Zb CR
		N007 Xc Zc CR
		N008 G80 CR
		N009 G00 Xt Zt CR

- (3) Thread cutting cycle commanded by G33 is performed in a compound fixed cycle and continuous thread cutting cycle by G34 and G35 is performed in LAP.

For G33 thread cutting cycle which is converted into compound thread cutting cycle, G71 or G72 is added depending on thread cutting direction, longitudinal or transverse.

Example:

<u>OSP3000L</u>					<u>OSP500L-G/OSP5000L-G</u>			
N001	G00	S1 T1 M1	CR		N001	G00	S1 T1 M1	CR
N002	Xs Zs		CR		N002	Xs Zs		CR
N003	G86		CR		N003	G88 NLA01 H D		CR
N004	D E		CR		NLA01	G81		CR
N005	G00 Xa Za		CR	→	N005	G00 Xa Za		CR
N006	G34 Xb Zb K F		CR		N006	G34 Xb Zb E F		CR
N007	Xc Zc		CR		N007	Xc Zc		CR
N008	G00 Xd Zd		CR		N008	G00 Xd Zd		CR
N009	G80		CR		N009	G80		CR

Example:

<u>OSP3000L</u>					<u>OSP500L-G/OSP5000L-G</u>			
N001	G00	S T M	CR		N001	G00	S T M	CR
N002	Xs Zs		CR		N002	Xs Zs		CR
N003	G86		CR		N003	G71 Xb Zb I D H E1 F		CR
N004	D E0		CR	→	N007	Xt Zt		CR
N005	G33 Xb Zb (I K) F		CR					
N006	G80		CR					
N007	Xt Zt		CR					

Example:

<u>OSP3000L</u>					<u>OSP500L-G/OSP5000L-G</u>			
N001	G00	S T M	CR		N001	G00	S T M	CR
N002	Xs Zs		CR		N002	Xs Zs		CR
N003	G86		CR		N003	G72 Xb Zb K D H E1 F		CR
N004	G82		CR	→	N008	Xt Zt		CR
N005	D E0		CR					
N006	G33 Zb Xb (I K) F		CR					
N007	G80		CR					
N008	Xt Zt		CR					

- (4) With OSP500L-G/OSP5000L-G, grooving and drilling cycles are performed not in LAP but in compound fixed cycle.

When grooving and drilling cycles are performed in compound fixed cycle, G code is converted into G73 or G74 depending on direction of cutting, longitudinal or transverse.

Example:

<u>OSP3000L</u>							<u>OSP500L-G/OSP5000L-G</u>								
N001	G00	Xt	Zt	Sl	Tl	Ml	CR	N001	G00	Xt	Zt	Sl	Tl	Ml	CR
N002		Xs	Zs				CR	N002		Xs	Zs				CR
N003	G83						CR	N003	G73	Xa	Za	K	D	Fr	CR
N004		D	E				CR	N007	G00	Xt	Zt	K	D	Fr	CR
N005	G01	Xa	Za	Fr			CR								
N006	G80						CR								
N007	G00	Xt	Zt				CR								

Example:

<u>OSP3000L</u>							<u>OSP500L-G/OSP5000L-G</u>								
N001	G00	Xt	Zt	Sl	Tl	Ml	CR	N001	G00	Xt	Zt	Sl	Tl	Ml	CR
N002		Xs	Zs				CR	N002		Xs	Zs				CR
N003	G83						CR	N003	G74	Xa	Za	I	D	Fr	CR
N004	G82						CR	N004	G00	Xt	Zt				CR
N005		D	E				CR								
N006	G01	Xa	Za	Fr			CR								
N007	G80						CR								
N008	G00	Xt	Zt				CR								

Note: When stock "L" or "U" or "W" is "0", finish cut cycle (G87) is not carried out.

16-4. RESTRICTIONS ON TAPE CONVERT FUNCTION (OSP3000L → OSP5000L-G)

The OSP5000L-G/OSP5000L-G has a tape convert function that converts a part program for OSP3000L into one compatible to OSP5000L-G/OSP5000L-G. However, this function cannot convert all OSP3000L part programs; some minor (not explained in the Programming Manual) or special specifications cannot be converted.

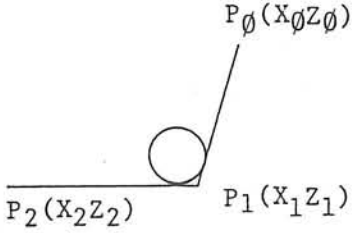
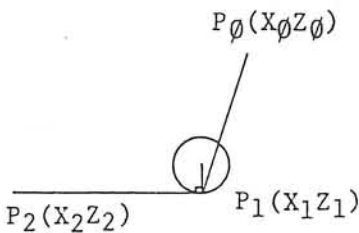
Generally, program items not explained in this section are output in the original OSP3000L form even after passing through the tape convert.

Therefore, specifications not listed as convertible among those modified according to the change of the control from OSP3000L to OSP5000L-G/OSP5000L-G are not accepted or executed by OSP5000L-G/OSP5000L-G.

Inconvertible Specifications:

Specifications	OSP3000L	OSP5000L-G/OSP5000L-G
Programming Format		
Command value "0"	Value "0" is omissible and just "X" or "Z" is acceptable.	Value "0" is not omissible and program must be made as "0" and "Z0".
+/- sign	"+" and "-" signs may be placed at any position before the next address character. See example below: X100-Z1+00	The signs must be placed right before the numeral value as below: X-100 Z+100
Space	Spaces may be placed at any position. See example below: X10 0 Z 1 00	Spaces must not be placed between numerals or between numerals and characters: X10 0Z 100 (not permissible)
M02	Leading zero is omissible and "M2" is acceptable.	Leading zero is not omissible and the code must be given as "M02".
/ (slash)	The slash code may be placed at any position in a block. The commands after the slash code are ignored. N001 G00 X100 /Z100	The slash code must be placed either at the beginning of each block or right after a sequence number. Otherwise, it is interpreted as a division calculation. N001 /G00 X100 Z100

Specifications	OSP3000L	OSP500L-G/OSP5000L-G
Programming Format % code	Commands may be programmed in the first sequence following the % code. % N001 G00 ...	The % code must be followed by an end of block command, LF or CR. % CR (LF) N001 G00 ...
Thread Cutting 8 times F command	Combined with M27, a thread lead is programmed following address character F in 8 times the actual lead. F000 M27	Codes M27 and M28 cannot be programmed since numerical values smaller than 0.001 mm are programmable. F0.00000
10 times F command	Combined with M28, a thread lead is programmed following address character F in 10 times the actual lead. F000 M28	
Designation of reference axis of thread lead	For a thread having an angle of 45 deg. or larger to Z-axis, its lead is referenced to X-axis.	Code M27 is used to change reference axis of thread lead from Z-axis to X-axis.
Simultaneous 4-axis Cutting G13 and G14	Axis movement commands and other commands can be programmed in the same block containing G13 or G14 selecting the turret used for cutting.	Both G13 and G14 must be specified in a block independently.
M00 and M01	These M codes may be programmed only for one turret.	These M codes must be programmed for both turrets A and B.
Mirror Image G13 and G14	Axis movement commands and other commands can be programmed in the block containing G13 or G14.	Both G13 and G14 must be specified in a block independently.

Specifications	OSP3000L	OSP500L-G/OSP5000L-G
<p>Nose Radius Compensation</p> <p>Start-up</p> <p>N0 G00 X0 Z0 N1 G42 X1 Z1 N2 G01 X2 Z2 F1</p>	 <p>With the commands in block N1, the cutting tool is positioned so that the tool nose comes to contact both straight lines P0P1 and P1P2.</p>	 <p>With the commands in block N1, the cutting tool is positioned so that the tool nose comes to contact the straight line P1P2 at point P1.</p>
<p>LAP</p> <p>Sequence branch</p>	<p>Sequence to which jump or branch is made after the completion of rough or finish cutting cycle is specified following address character Q.</p>	<p>Since commands of contour or shape definition, rough cutting cycle, and finish cutting cycle are independently programmed, no sequence branch function is available.</p>

Inconvertible special specifications of OSP3000L

- Parameter program (user task)
- Tool life management
- Gauging functions (work gauging, tool gauging, post-process gauging)
- Robot
- Programmable tailstock function
- Bar feeder
- Multiple machining model (C-axis linkage, C-axis commands, fixed cycle)
- Scheduled operation
- Overload detection on Z-axis

Other functions available only on a specific model and machine designed for custom-use generally cannot be converted.

SECTION 17 TAILSTOCK SPINDLE ADVANCE/RETRACT FUNCTION BY M CODES
 - With Confirmation Limit Switch -

17-1. M CODES USED FOR ADVANCING/RETRACTING TAILSTOCK SPINDLE

M55 Tailstock spindle retract

M56 Tailstock spindle advance

Selected mode is not influenced by turning on the control or by resetting it.

For the two-saddle models, the M codes above may be designated at either G13 or G14 mode.

The time duration in which the tailstock spindle advance should be completed after the tailstock in-position signal has been input during the tailstock spindle advance movement, commanded by the M56 code, is set by parameter (word) No. 14 in increments of 0.01 seconds. The loading of the NC control software sets this time at 0.8 seconds.

17-2. ALARM MESSAGES

138 ALARM-A Tailstock spindle advance answer

In-position answer signal is not input within 5 seconds after the tailstock spindle advance command M56 has been executed.

Index : None
 Character-string: None
 Code : 2

139 ALARM-A Tailstock spindle over advance

The tailstock spindle over-advanced during program execution.

EC input #6

Bit 5	Bit 4	Bit 3
0	0	1

Index : None
 Character-string: None
 Code : 1

464 ALARM-B Tailstock spindle over-advance

The tailstock spindle over-advanced.

EC input #6

Bit 5	Bit 4	Bit 3
0	0	1

Index : None
 Character-string: None
 Code : 1

140 ALARM-A Tailstock spindle condition illegal

The spindle rotation command is designated while the tailstock spindle is not advanced for shaft work, or not retracted for chuck work.

The tailstock spindle is shifted from the correct position in shaft work while the spindle is rotating.

The tailstock spindle is shifted from the retraction end position in chuck work while the spindle is rotating.

The tailstock spindle advance or retraction command (M55 or M56) is designated for chuck work or while the spindle is rotating.

The alarm does not take place when the interlock function is canceled by M157.

Index : TURRET or None
 Character-string: None
 Code : 1 Spindle rotation command is designated. M55 or M56 is designated for chuck work.
 2 The tailstock spindle is shifted from the correct position. M55 or M56 is designated while the spindle is rotating.

Note 1: The confirmation limit switch is not used for the U.S.A. specification.

Note 2: The LH55-N is not provided with this function to assure safety of operations.

SECTION 18 SCHEDULE PROGRAM

18-1. OUTLINE

The schedule program is used for performing machining of various types of workpieces continuously and automatically when these workpieces can be loaded and unloaded automatically by the robot, the autoloader or other material handling equipment. By combining the schedule program with the machining programs, continuous and automatic machining can be accomplished.

The following statements are included in a schedule program:

- (1) Statement for selection the main program PSELECT
- (2) Statement for specifying the order of schedule program block execution IF, GOTO
- (3) Statement for setting variables VSET

A schedule program can be made using the statements indicated above, and various types of workpieces can be machined automatically and continuously by executing the schedule program.

For operation procedures on schedule program execution, please refer to Section 4-1-6 in the Operation Manual.

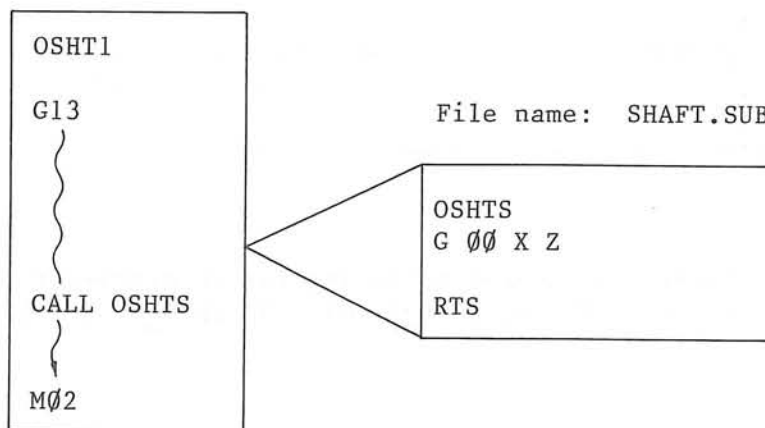
18-2. PROGRAMMING

- (1) Statement for selecting the main program for execution (PSELECT):

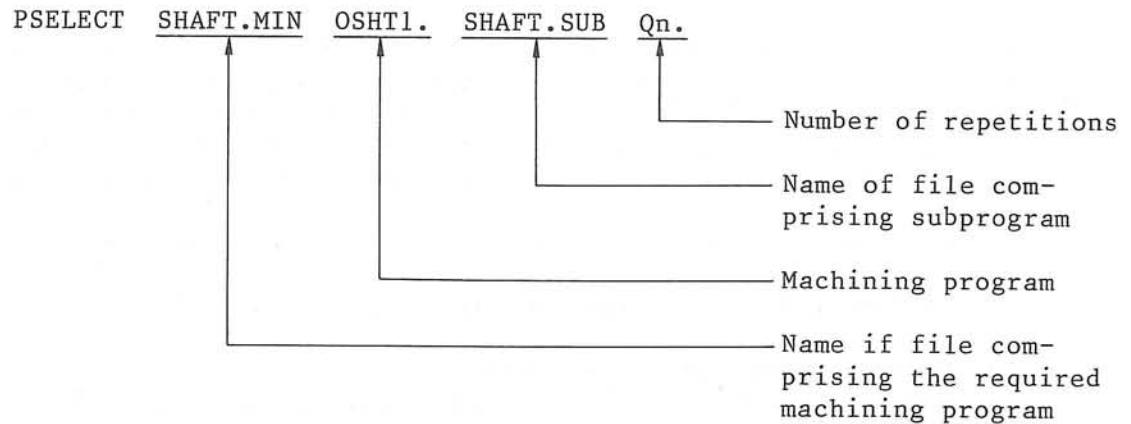
Using the statement PSELECT, program the main program name and the subprogram name to be executed along with the number of repetitions of the called program.

As shown below, the SHAFT.MIN file contains an OSHT1 machining program, and a SHAFT.SUB file is provided for the subprogram called from that program.

File name: SHAFT.MIN



Program as follows for executing the machining program "n" times.

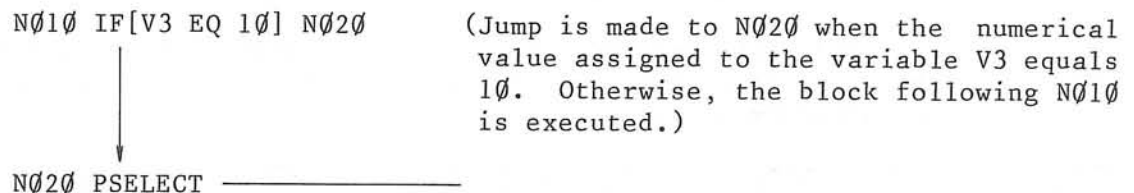


Note: When the program is to be executed one time, it is not necessary to program a Q command.

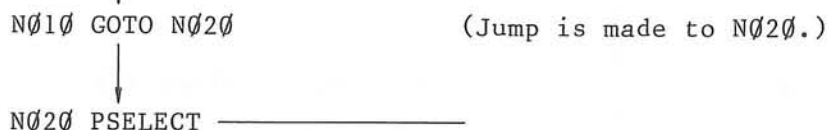
Command Q9999 causes the program to be repeated infinitely.

- (2) Statement for specifying the order of schedule program block execution (IF, GOTO)

The IF statement causes a jump to the specified sequence number or name if the indicated conditions are fulfilled.



The GOTO statement causes a jump to the specified sequence number or name.



Note: Conditions stated following the IF statement can contain EQ (=), NE (≠), GE (>=), GT (>), LE (<=), and LT (<).

(3) Statement for setting variables (VSET)

This is the statement for assigning common variables V1 through V32.

V7 = 1 Assignment of numerical value 1 to variable V7

V1 = V7 + 1 Assignment of numerical value of "V7+1" to
variable V1

V6 = V10 + V11 Assignment of numerical value of "V10+V11" to
variable V6

(4) End of schedule program (END)

Program the END statement at the end of a schedule program.

```

N001 PSELECT
      ↓
N020 END

```

18-3. PROGRAM EXAMPLE

```

N001 VSET V1=0
N002 PSELECT SHAFT.MIN,OSHT1,SHAFT.SUB,Q100
N003 VSET V1=V1+1
N004 IF [V1 GT 1] N007
N005 PSELECT SHAFT.MIN,OSHT2,SHAFT.SUB,Q50
N006 GOTO N008
N007 PSELECT SHAFT.MIN,OSHT3,SHAFT.SUB,Q200
N008 IF [V1 GT 2] N010
N009 GOTO N002
N010 END

```

Explanation of example program:

N001: Variable V1 is assigned with numerical value zero (V1=0).

N002: Machining program OSHT1 is executed 100 times.

N003: Variable V1 is set to "V1+1" (V1=V1+1).

N004: If V1 is greater than 1 (V1 > 1), the sequence jumps to N007.

N005: Machining program OSHT2 is executed 50 times.

SECTION 19 LATHE AUTO-PROGRAMMING FUNCTION (LAP)

19-1. GENERAL DESCRIPTION

LAP (Lathe Auto-Programming) is the function to make full use of high-speed processing capability which characterizes the OSP500L-G/OSP5000L-G series. 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 finish the workpiece to the programmed dimensions.

This feature permits the programmer to complete the part program simply by picking up the dimensions specified in an engineering drawing and, therefore, it simplifies programming as well as it reduces programming time; this furthermore facilitates tape check procedure and also tape punch procedure.

Various cutting modes available with the LAP can cope with any type of cutting intended.

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) Tape preparation time can be greatly reduced.
- (3) Programming for rough cut cycle can be eliminated, and this simplifies manual calculation while programming.
- (4) Change of cutting conditions such as depth of cut and feedrate is possible during rough cut cycle.

19-2. CLASSIFICATION OF FUNCTIONS

19-2-1. Classification of Cutting Cycle

- AP Mode I for bar turning
 AP Mode II for copy turning on forged workpieces
 AP Mode III for thread cutting

	Longitudinal Cutting Mode	Transverse Cutting Mode
AP Mode I		
AP Mode II		
AP Mode III		

19-2-2. G Code Used to Designate Cutting Mode

G85: AP Mode I

Used to call out bar turning rough cut cycle.

G84: Change of rough cut conditions for bar turning

G86: AP Mode II

Used to call out copy turning mode.

G87: Finish cut cycle

Used to call out finish cut 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 indicate direction of AP mode cycle cutting execution (contour specification):

G81: Start of longitudinal contour definition

G82: Start of transverse (on end face) contour definition

G80: End of contour definition

19-3. PROGRAM FORMAT

19-3-1. G Codes

G Code	Description
G80	End of contour definition
G81	Start of contour definition, longitudinal
G82	Start of contour definition, transverse
G84	Change of rough cut conditions, bar turning
G85	Bar turning rough cut cycle
G86	Copy turning cycle
G87	Finish cut cycle
G88	Continuous thread cutting cycle

19-3-2. 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

* Not available with the A-specification.

19-3-3. Parameters

Parameter	Description
D	Depth of cut in rough cut cycle 1) Alarm 2) $D > \emptyset$
DA	Depth of cut after rough cut conditions change point A 1) $DA = D$ 2) $DA > \emptyset$
DB	Depth of cut after rough cut conditions change point B 1) $DB = DA$ 2) $DB > \emptyset$
FA	Feedrate after rough cut conditions change point A 1) $FA = F$ 2) $FA > \emptyset$

Parameter	Description
FB	Feedrate after rough cut conditions change point B 1) $FB = FA$ 2) $FB > \emptyset$
E	Feedrate in rough cut cycle along finish contour 1) F active at entry of LAP mode 2) $E > \emptyset$
XA	X coordinate of rough cut condition change point A 1) No change of cutting conditions 2) $ XA \leq 99999.999$
XB	X coordinate of rough cut condition change point B 1) No change of cutting conditions at point B 2) $ XB \leq 99999.999$
ZA	Z coordinate of rough cut condition change point A 1) No change of cutting conditions 2) $ ZA \leq 99999.999$
ZB	Z coordinate of rough cut condition change point B 1) No change of cutting conditions 2) $ ZB \leq 99999.999$
U	Stock removal amount in X-axis direction for finish cut cycle 1) $U = \emptyset$ 2) $U \geq \emptyset$
W	Stock removal amount in Z-axis direction for finish cut cycle 1) $W = \emptyset$ 2) $W \geq \emptyset$
H	Thread height in G88 thread cutting cycle 1) Alarm 2) $H > \emptyset$
B	Tip point angle of thread cutting tool in G88 1) $B = \emptyset$ 2) $\emptyset \leq B < 180^\circ$ 1) Default 2) Restriction on numeral data specification.

Note 1: The following words should be specified in incremental values.

D, DA, DB, U, W and H

Note 2: D, DA, DB, XA, XB, U and H words should be commanded in diameter.

Note 3: In thread cutting cycle using the M73 pattern, "H - U" must be greater than or equal to D:

$$H - U \geq D$$

In the M74 and M75 patterns, it must be positive:

$$H - U \geq \emptyset$$

Note 4: When more than one alphabetic characters are 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=

19-4. EXECUTION MODE OF LAP

19-4-1. G85: Bar Turning Cycle

Format

N0103 G85 NLAP1 D F U W G84

- N0103: Sequence number
- G85 : G code calling out bar turning cycle
To be provided right after sequence number (name)
- NLAP1: Sequence name in the first block of contour defining blocks
- Blank: Enter either tab or space code
- D : Depth of cut in rough cut cycle
- F : Feedrate in rough cut cycle
- U : Stock removal in finish cut cycle, X component
- W : Stock removal in finish cut cycle, Z component
- G84 : Change of rough cut conditions

With the commands above, the control starts searching of contour definition program beginning with the sequence name NLAP1. After assigning parameter data of D, F, U, W and G84 for NLAP1, the control starts bar turning cycle.

Note 1: No S, T or M code may be provided in the G85 block.

Note 2: D word is used to specify depth of cut in rough cut cycle. When the command indicating change of cutting conditions is provided, the D word is effective up to such point, XA and ZA.

D word which has a positive value must be provided in the G85 block without fail, if not, i.e., if the numeral data of the D word is not positive, or if it is omitted, an alarm results.

Note 3: F word is used to specify the feedrate in rough cut 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 effective before the execution of the G85 block is effective.

F word must be positive. If not, an alarm results.

Note 4: When U and/or W word is not provided, U and/or W is assumed "0". U and W words must be positive or zero. If not, an alarm results.

19-4-2. G84: Change of Cutting Conditions in Bar Turning Cycle

Format

N	G85	N			
\$	G84	XA=(ZA=)	DA=	FA=	
\$		<u>XB=(ZB=)</u>	<u>DB=</u>	<u>FB=</u>	
(1)		(2)	(3)	(4)	

- (1): Indicates that the commands are continuous.
 (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 should 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 one line with the commands directly associated with G85, they are provided in different lines preceded by "\$" character 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 cut cycle from the desired point(s). If change of cutting conditions is not necessary, omit them.

- Note 1: G84 and following commands must be provided after "N G85 N", if any.
- Note 2: For OD turning, coordinate values of "LAP starting point", "rough cut condition change point A" and "rough cut condition change point B" must be provided so that they become smaller in this order. For ID turning, they should be provided so that they become larger in that order.
- Note 3: If rough cut condition change point A and B both exist within a range where cut is made present position, the rough cut cycle after the infeed of D is carried out under the conditions of FB and DB.
- Note 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 provided the present position is outside XA, and with DA when the present position is on XA.
- Note 5: In longitudinal cutting, ZA= and ZB= commands may not be provided. In transverse cutting, XA= and XB= commands may not be provided, either.

19-4-3. G86: Copy Turning

Format

N0123 G86 NLAP2 D F U W

N0123: Sequence number

G86 : G code calling out copy turning cycle
To be provided right after sequence number (name)

NLAP2: 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 cut cycle, X component

W : Stock removal in finish cut cycle, Z component

With the commands above, the control starts searching of contour definition program beginning with the sequence name NLAP2. After assigning parameter data of D, F, U and W of NLAP2, the control starts copy turning cycle.

Note 1: No S, T or M code may be provided in the G86 block.

Note 2: D word is used to specify depth of cut in each cycle and must be provided in the G86 block without fail.

D word value must be positive. If not, an alarm results.

Note 3: F word specifies the feedrate for the blocks until an E word is provided in the contour definition program.

If no F word is provided in the G86 block, the feedrate effective before the execution of the G86 block is effective.

Note 4: When U and/or W word is not provided, U and/or W is assumed "0".

U and W words must be positive or zero. If not, an alarm results.

19-4-4. G87: Finish Cut Cycle

Format

N0203 G87 NLAP1 U W

N0203: Sequence number

G87 : G code calling out finish cut cycle
To be provided right after sequence number (name)

NLAP1: Sequence name in the first block of contour defining blocks

Blank: Enter either tab or space code

U : Stock removal in finish cut cycle, X component

W : Stock removal in finish cut cycle, Z component

With the commands above, the control starts searching of contour definition program beginning with the sequence name NLAP1. After assigning parameter data of U and W of NLAP1, the control starts the finish cut cycle.

Note 1: No S, T or M code may be provided in the G87 block.

Note 2: As a feedrate, the one provided in the contour definition program is effective.

If no F word is provided in the contour definition program, the feedrate effective before this block becomes effective.

NOTE THAT THE G87 BLOCK CANNOT CONTAIN AN F WORD.

Note 3: When U and/or W word is not provided, U and/or W is assumed "0".

U and W words must be positive or zero. If not, an alarm results.

19-4-5. G88: Continuous Thread Cutting Cycle

Format

NØ143 G88 NLAP3 D H B U W M32 (M33, M34) M73 (M74, M75)

NØ143 : Sequence number

G88 : G code calling for continuous thread cutting cycle
To be provided right after sequence number (name)

NLAP3 : 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 finish cut cycle, X component

W : Stock removal in finish cut 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 NLAP3. After assigning parameter data of D, H, B, U, W, M32 (M33, M34) and M73 (M74, M75) for NLAP3, the control starts the thread cycle.

Note 1: No S, T or M code may be provided in the G88 block.

Note 2: D word is used to specify the depth of cut in the first thread cutting cycle. After that the depth of cut in each thread cutting cycle varies according to the selected infeed pattern.

D word which has a positive value must be provided in the G88 block without fail. If not, i.e., if the numeral data of the D word is not positive, or if it is omitted, an alarm results.

Note 3: H word must have a positive value and must be specified in the G88 block without fail. If not, i.e., if the numeral data of the D word is not positive, or if it is omitted, an alarm results.

H value must be greater than U and/or W value. If not, an alarm results.

Note 4: B word specifying the tip point angle of thread cutting tool must have the value within a range of \emptyset deg. to $18\emptyset$ deg.:

$$\emptyset^\circ \leq B < 18\emptyset^\circ$$

When no B word is provided, it is assumed " \emptyset ".

Note 5: M32, M33 and M34 are used to select cutting mode:

M32 straight infeed along thread face (on left face)

M33 zigzag infeed in G88

M34 straight infeed along thread (on right side)

When neither M32, M33 nor M34 is provided, the control selects the M32 infeed mode.

Note 6: M73, M74 and M75 are used to select infeed pattern. When no such M code is presented, the M73 pattern is automatically selected. In the M73 pattern, "H - U" must be greater or equal to "D".

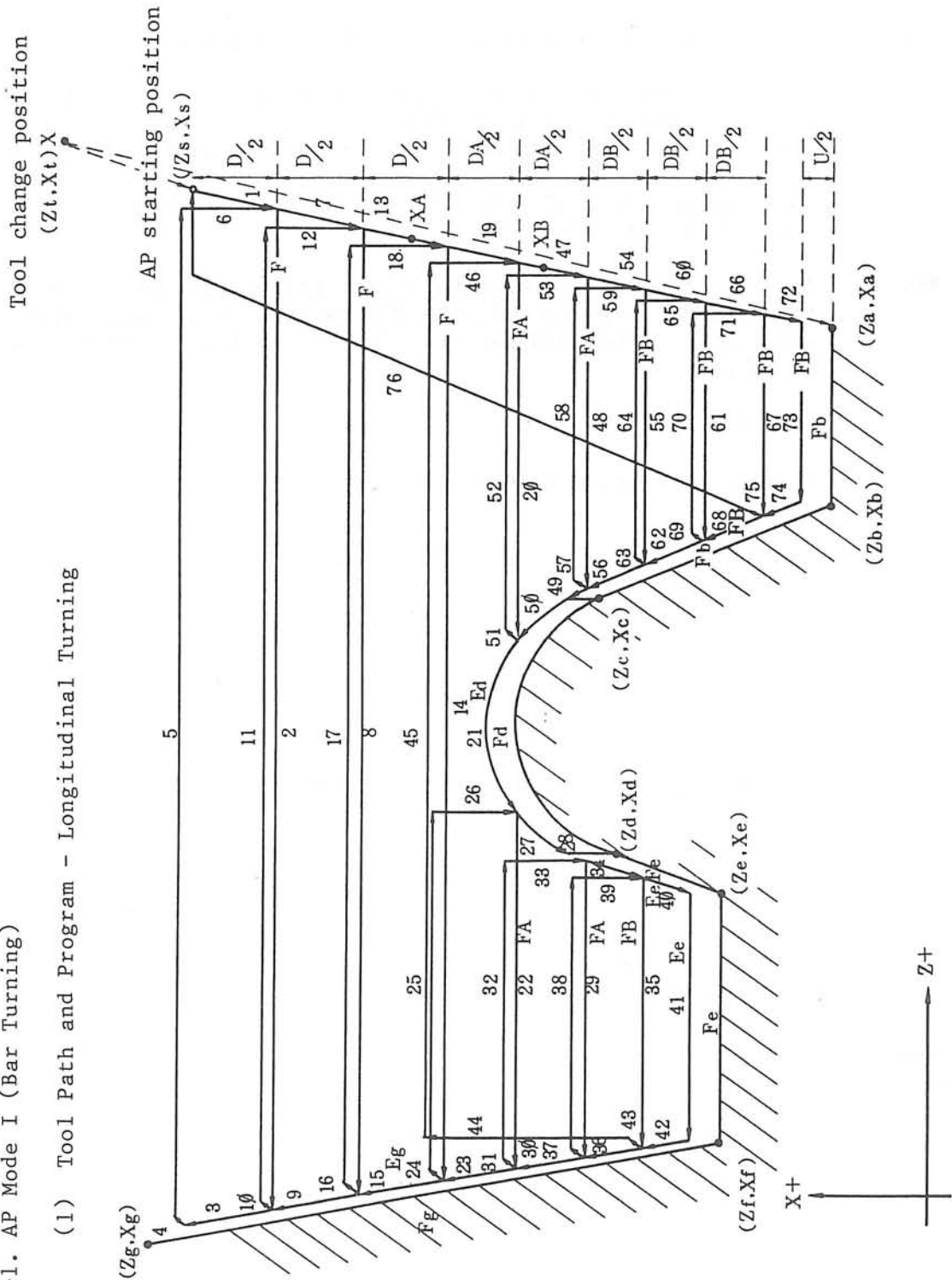
$$H - U \geq D$$

If not, an alarm results.

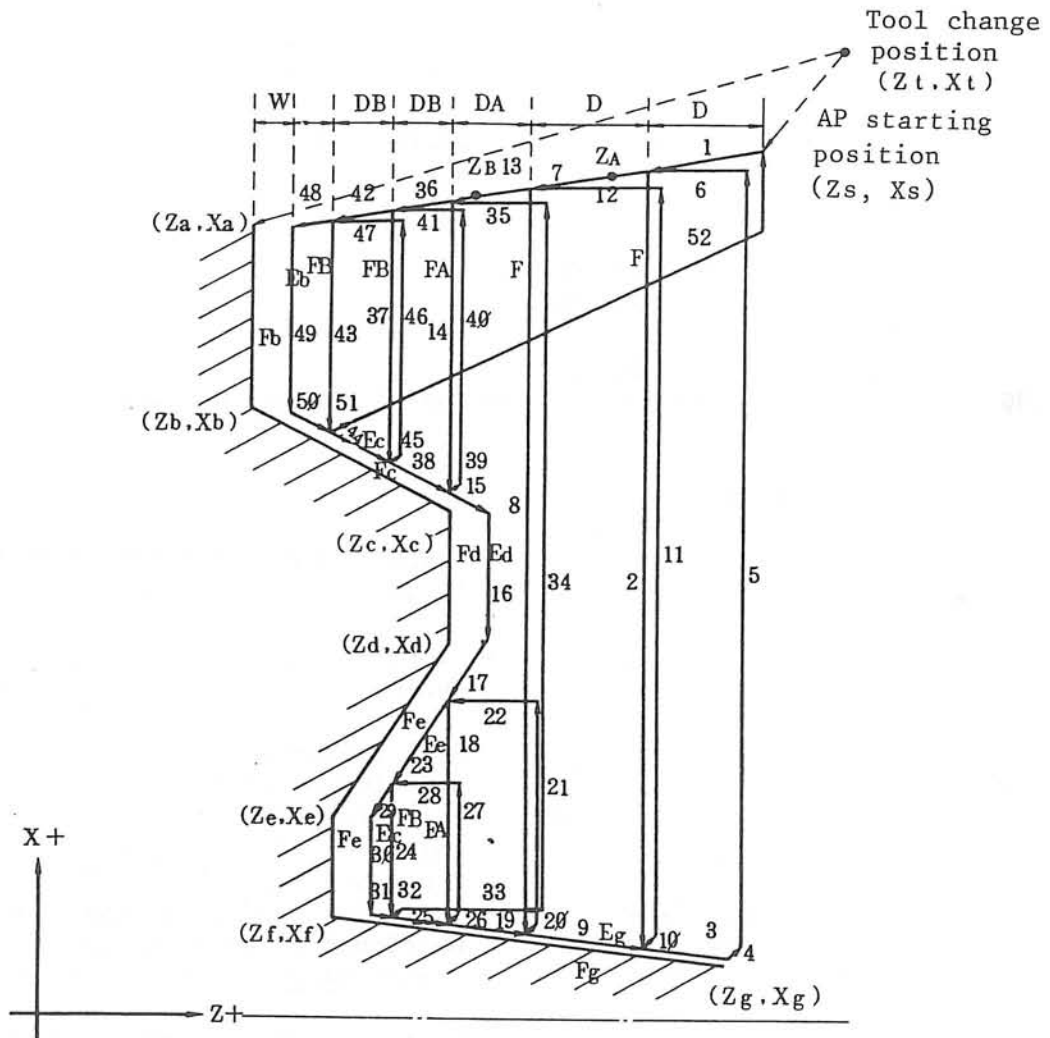
19-5. EXPLANATION OF LAP FUNCTIONS AND PROGRAM

19-5-1. AP Mode I (Bar Turning)

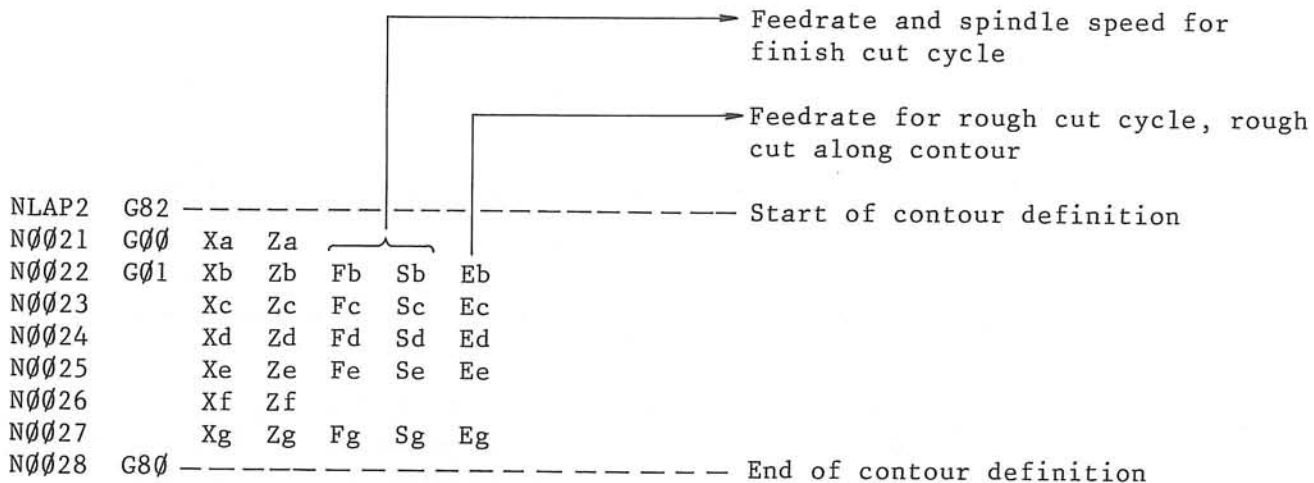
(1) Tool Path and Program - Longitudinal Turning



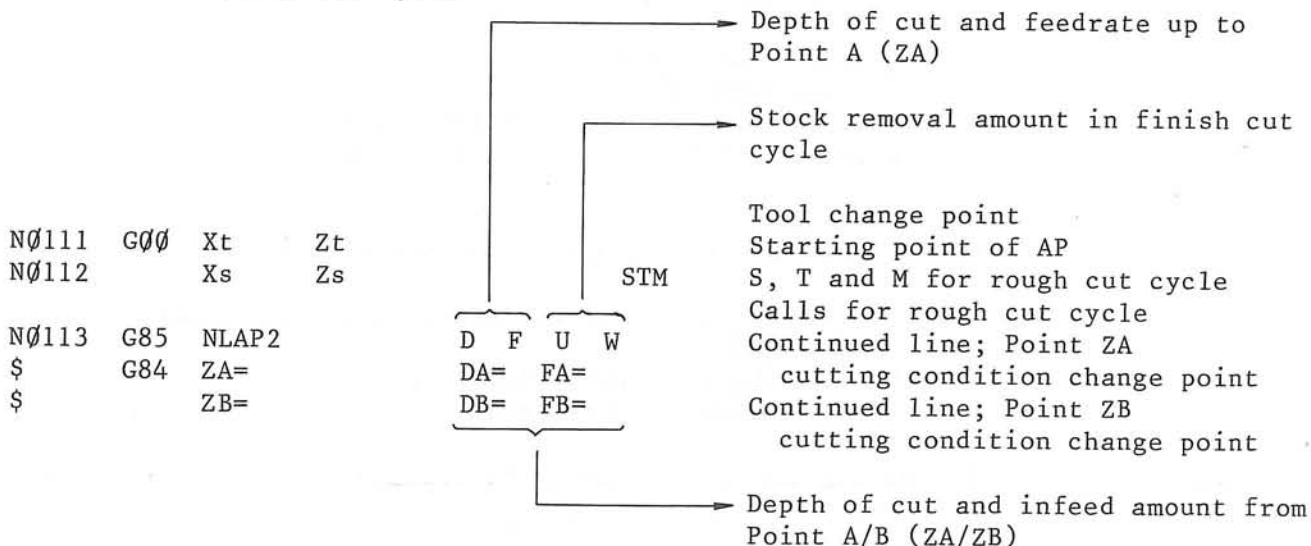
(2) Tool Path and Program - Transverse Cutting



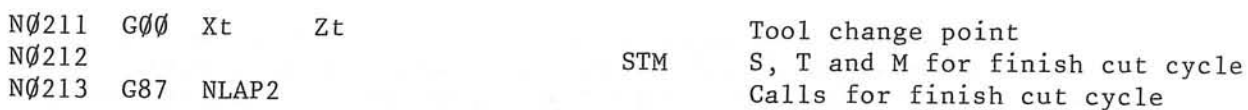
Contour Definition



Rough Cut Cycle



Finish Cut Cycle



(6) Outline of Bar Turning Cycle

a) Rough cut cycle in longitudinal cutting

- 1) With the commands in block NØ1Ø1, positioning at the tool change point is performed.
- 2) With the commands in block NØ1Ø2, 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 NAP1 command in block NØ1Ø3, the control searches the program assigned with the program name NLAP1. Rough cut cycle in bar turning mode is performed on this program.

In the same block, cutting conditions for rough cut cycle are also specified.

D depth of cut
 F feedrate
 U X component of finish allowance
 W Z component of finish allowance

When cutting conditions are to be changed during rough cut cycle, provide 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

If cutting conditions are to be changed again, further provide 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 associated with such point(s) are provided 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 provided in this block, the feedrate commanded last becomes effective.

Point data of cutting conditions change point(s) must become smaller in the order of the AP starting point, X_A and X_B 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 $N\text{O}001$, the control calculates the intersection point of the following two straight lines. The line parallel to Z-axis running at " $X_s - D/2$ " and the one passing the two points (X_s, Z_s) and $(X_a + U, Z_a + W)$. Positioning at the calculated point A (X_p, Z_p) is then performed.

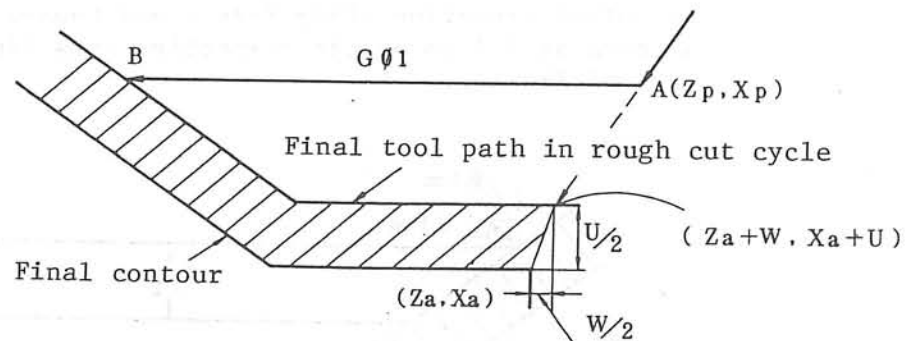
Select the AP starting point (X_s, Z_s) with respect to the coordinate point (X_a, Z_a) to meet the following requirements:

$X_s < X_a$ for ID cutting

$X_s > X_a$ for OD cutting

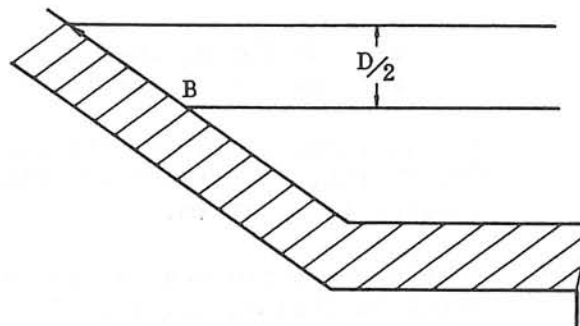
If too large a finish allowance U is taken to the degree " X_a+U " falls outside " X_s " with respect to the workpiece, it results in an alarm.

- 5) Cutting is performed in the $G01$ mode up to point B where the straight line parallel to Z-axis and passing point A intersects final contour of rough cut cycle. The feedrate in this cutting cycle is as selected by the F word when rough cut cycle is called out.

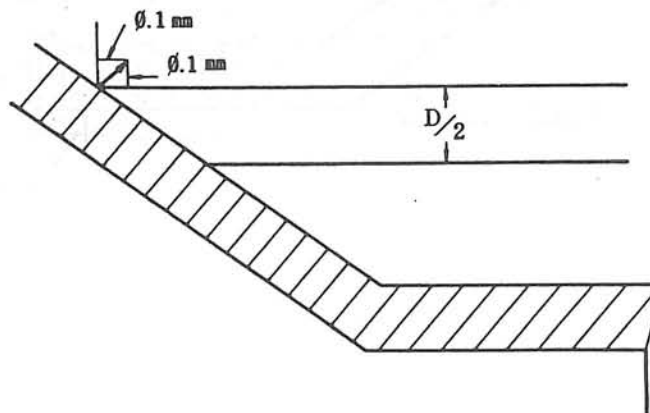


- 6) After point B is reached, cutting is then performed along the final contour of rough cut cycle up to the point whose X coordinate is $X_B + D$. If G8Ø indicating end of contour definition is existent before such point is reached, cut along the final rough cut contour is performed up to the point specified in the block preceding the G8Ø block.

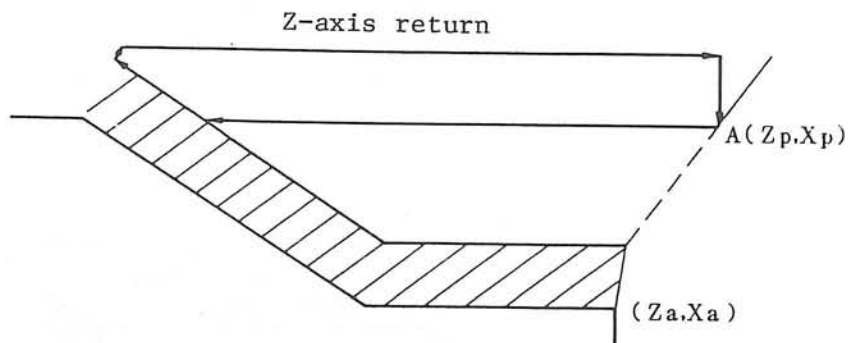
Feedrate in this cut is as specified by E which is provided in a contour definition program. If no E word is provided in the corresponding contour definition program, the one specified last becomes effective. When an E word has not been specified, the feedrate specified when calling out rough cut cycle becomes active.



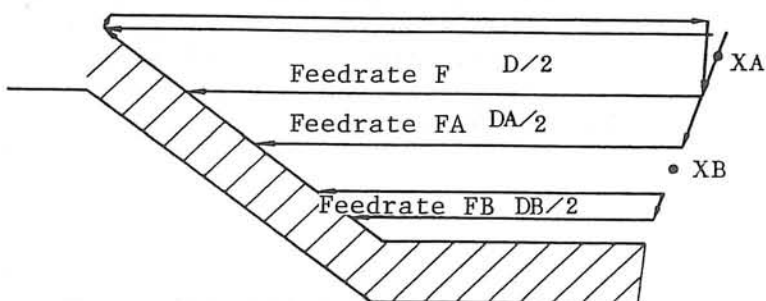
- 7) After the completion of cutting explained in 6), the cutting tool relieves from the workpiece in the direction opposite to infeed direction along X-axis and toward Z_s along Z-axis as much as $\phi.1$ mm on the respective axes (in diameter in the case of X-axis).



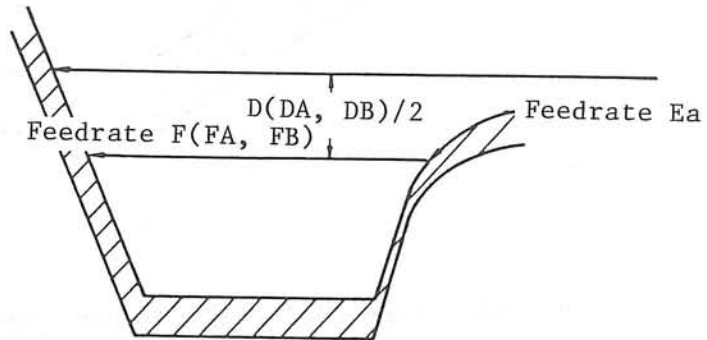
- 8) This completes the first rough cut cycle. Z-axis returns to Z_p determined in step 4) at a rapid traverse rate and then X-axis to X_p .



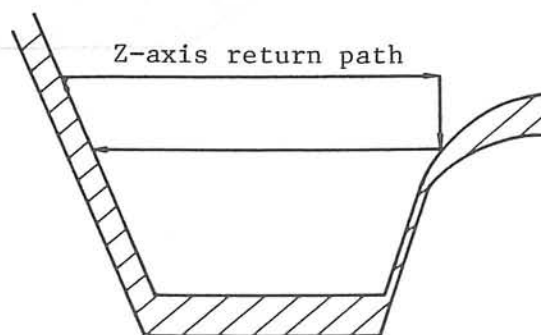
- 9) The steps 4) through 8) are repeated up to the cutting condition change point. After that point, the same cycle is repeated with the depth of cut (D) and feedrate (F) changed.



- 10) If cutting along a descending slope is performed in step 6), and when contour to be cut exists below the point of cutting (X_p), cutting is first performed along the contour until the programmed depth of cut is reached and then it is performed in parallel with Z-axis up to the point where the straight line passing that point intersects the final rough cut contour. Cutting along the parallel line is performed at the feedrate specified by an F word (F_A/F_B).

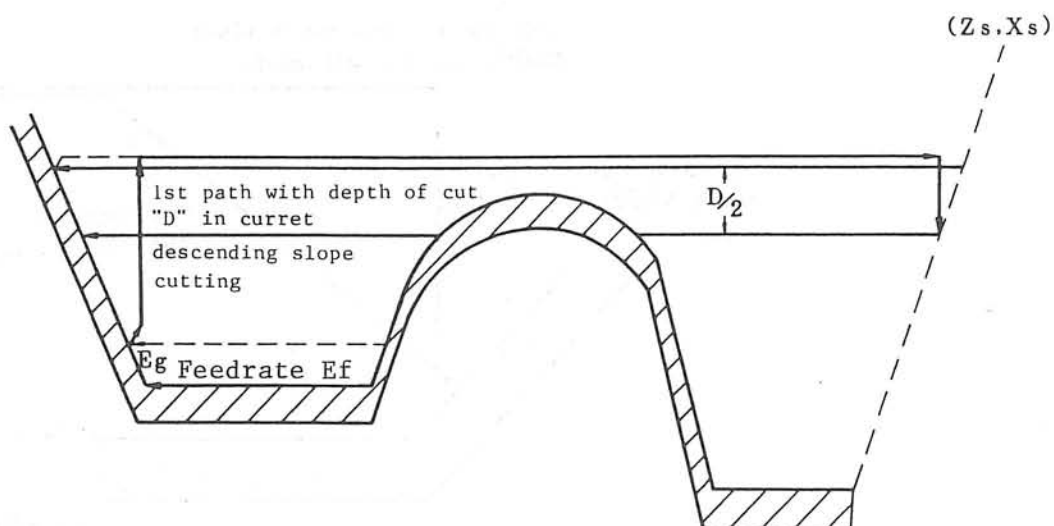


- 11) Steps 6) and 7) are repeated after that. Z-axis then returns to the point where cutting along Z-axis is started in step 10). After the completion of positioning of Z-axis, X-axis is positioned at the point, where previous cutting cycle has been started.



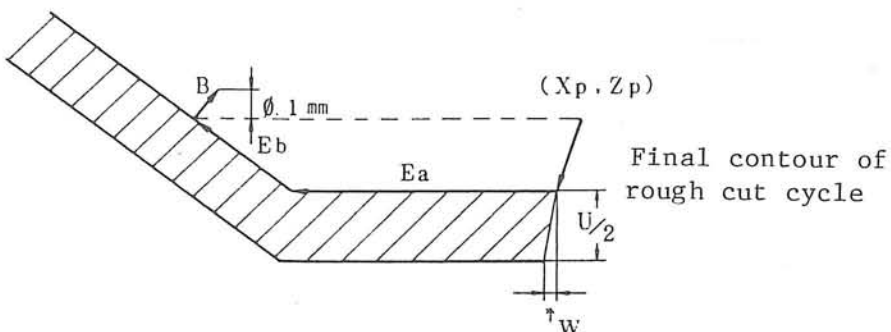
- 12) Steps 10) and 11) are repeated until the most recessed section along X-axis is cut. After the final cutting on such section, both X- and Z-axis relieve as indicated before, and then positioning of X-axis is made at the point having coordinate value "the first cutting level along the descending slope + $(D + 0.1)$ " mm. 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 X-axis at that point is then executed.

After completing the cutting along the descending slope, cutting before starting such cutting is resumed and steps after 4) are repeated.



- 13) The steps indicated before are repeated until X-axis reaches the level where tool path is generated below " $X_a + U$ " if ordinary infeed along X-axis is taken. When such level is reached, the final rough cutting is carried out along the contour up to point B leaving finish cut allowance.

The feedrate in cutting along the final rough cut contour is the one specified by the E word.

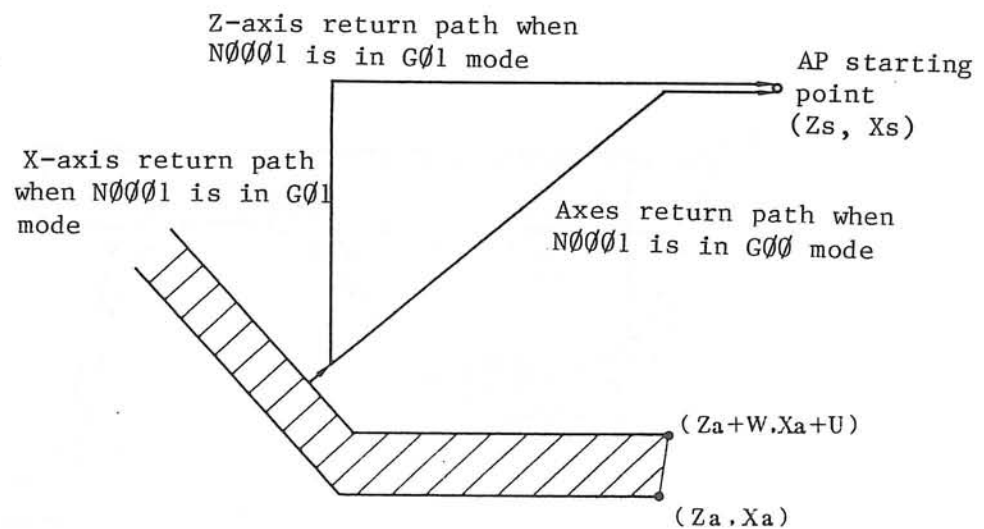


- 14) At the completion of step 13), the axes return to the AP starting point (X_s, Z_s).

There are two patterns of axes return motion as:

Simultaneous two axes return when the first block of the contour definition program (the block following the one containing either G81 or G82) is controlled in the G00 mode, or

Positioning along X-axis is made first and then Z-axis returns to the AP starting point when the above indicated block is in the G01 mode.



If the N0001 block has neither G00 nor G01, the return operation is carried out assuming G00.

This completes a rough cut cycle.

b) Finish cut cycle in bar turning

- 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 cut cycle are selected.
- 3) With the NLAP1 command in block N0203, the control searches the program assigned with the program name NLAP1. Finish cut cycle in bar turning mode is performed on this program.
- 4) The finish cut cycle is performed following the dimension data provided in the contour definition program in the specified cutting conditions for the finish cut cycle.
- 5) After the finish cut cycle is completed, the commands in the block following N0203 are executed.

(4) Remark on bar turning

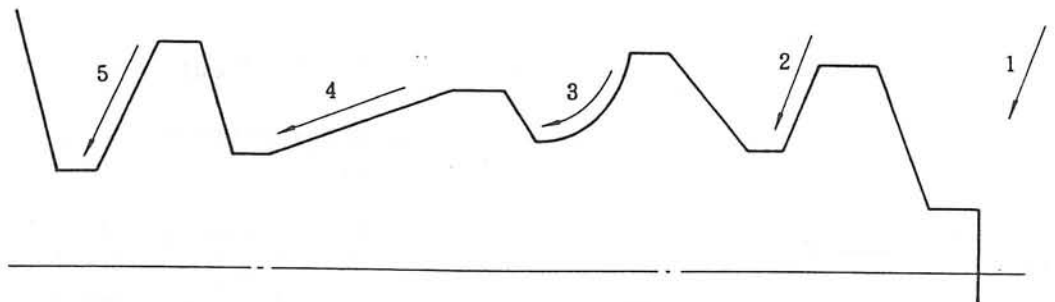
- a) If both U and W words are specified for descending slope cutting, overcutting will occur on slopes. Therefore, never program those commands at the same time when cutting descending slopes.

For longitudinal cutting, use only a U word and only a W word for transverse cutting.

With U or W word presented, X- or Z-axis is constantly offset depending on the specified word.

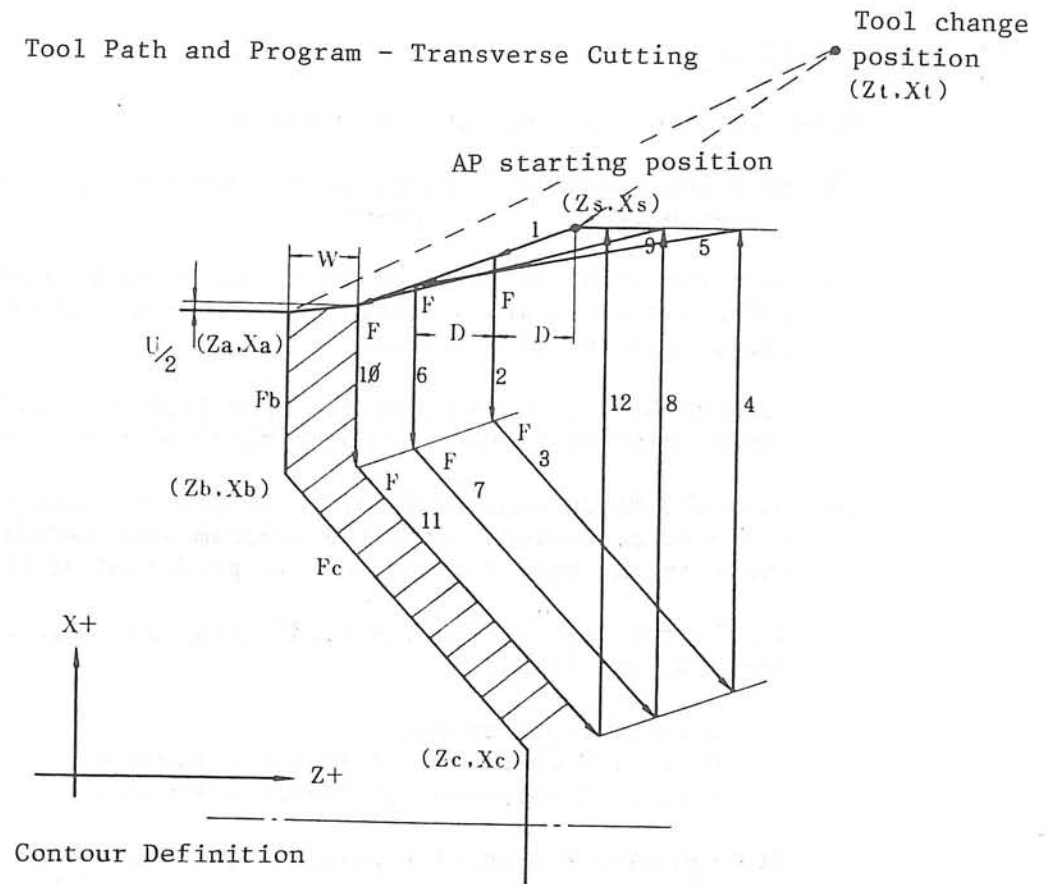
- b) Maximum allowable number of descending slopes in the AP Mode I

The maximum programmable number of descending slopes 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.

(2) Tool Path and Program - Transverse Cutting



Contour Definition

NAP30	G82								Feedrate and spindle speed for finish cut cycle
N0301	G00	Xa	Za						End face of contour definition
N0302	G01	Xb	Zb	Fb	Sb				
N0303		Xc	Zc	Fc	Sc				
N0304	G80								End of contour definition

Rough Cut Cycle

N0131	G00	Xt	Zt						Depth of cut and feedrate
N0132		Xs	Zs						Stock removal amount in finish cut cycle
N0133	G86	NAP30		D F	U W				Tool change point Starting point of AP S, T and M for rough cut cycle Calls for rough cut cycle

Finish Cut Cycle

N0221	G00	Xt	Zt						Tool change point
N0222									S, T and M for finish cut cycle
N0223	G87	NAP30							Calls for finish cut cycle

(3) Outline of Copy Turning Cycle

a) Rough cut cycle in longitudinal cutting

- 1) With the commands in block NØ121, positioning at the tool change position is performed.
- 2) With the commands in block NØ122, S, T and M commands for a rough cut cycle are selected and then positioning at the AP starting point is performed.

When no S, T or M commands are specified in this block, those selected in the preceding block(s) become effective.

- 3) With the NAP2Ø command in block NØ123, the control searches the program assigned with the program name NAP2Ø. Rough cut cycle in the copy turning mode is performed on this program.

In the same block, cutting conditions for rough cut cycle are also specified:

D depth of cut
 U X component of finish allowance
 W Z component of finish allowance

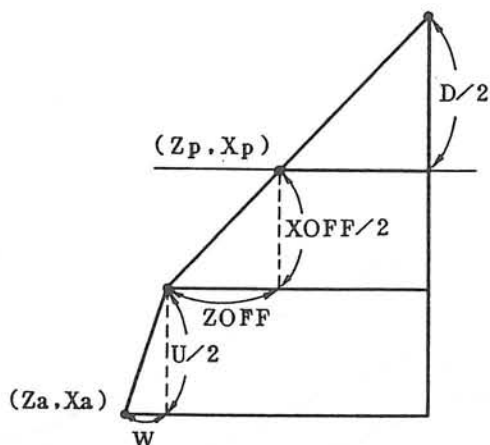
Program also F word if required. When no F word is presented in the contour definition program, either, the feedrate commanded last becomes effective.

- 4) Upon reading the commands in block NØ2Ø1 in the contour definition program, the control calculates the point of intersection of the following two straight lines. The line parallel to 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).

$$\begin{aligned} X_p &= X_s - D \\ Z_p &= Z_a + W + (Z_s - Z_a - W) (1 - D / (X_s - X_a - U)) \\ X_{OFF} &= X_p - (X_a + U) \\ Z_{OFF} &= Z_p - (Z_a + W) \end{aligned}$$

See explanatory illustration below:

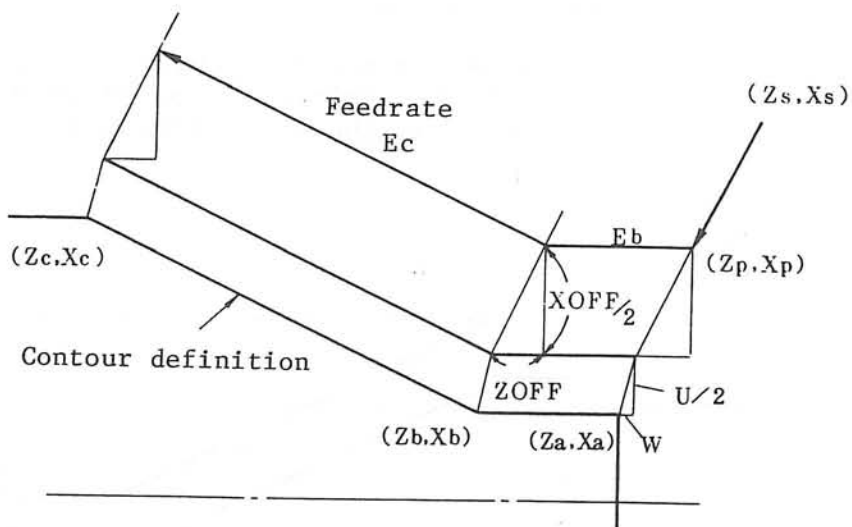


If too large a finish allowance U or W is commanded, causing the infeed direction to be reversed and an alarm occurs.

- 5) Cutting is started from (X_p, Z_p) to the target point (*1) calculated by the OSP500L-G/OSP500L-G.

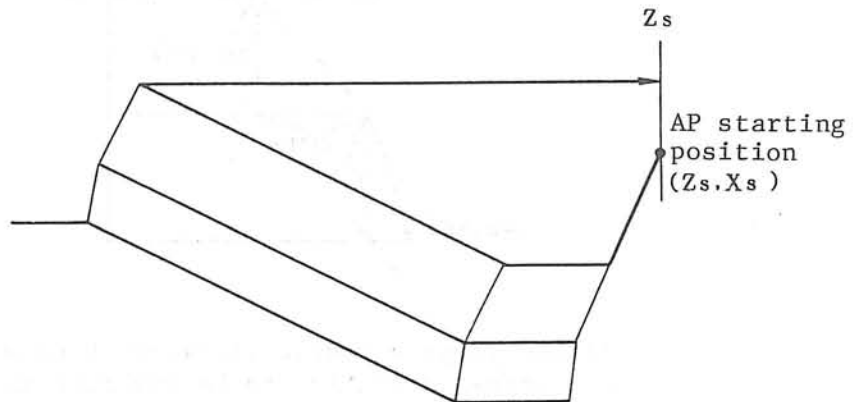
Target point (*1): The point obtained by offsetting the points commanded in the contour definition program in parallel with respective axes as much as $(XOFF + U, ZOFF + W)$.

Cutting is performed at the feedrate specified by an E word in each block of the contour defining blocks.



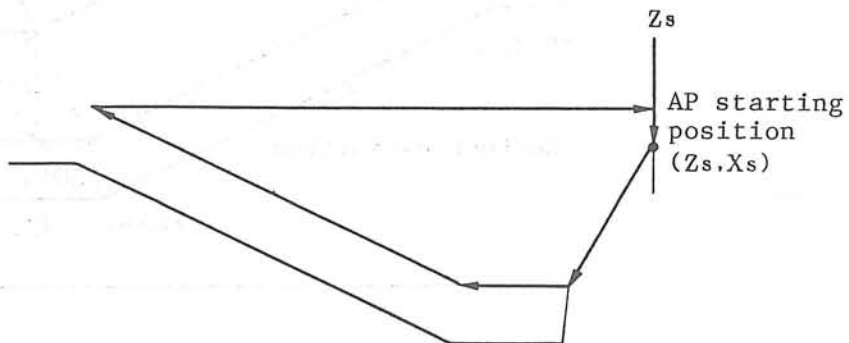
- 6) Step 5) is repeated up to the contour definition ends (G80 active).

Z-axis then returns to the Z coordinate of AP starting point, Z_s .



- 7) This completes one cycle of rough cutting. New ZOFF and ZOFF are calculated and steps 4) through 6) are repeated.
- 8) The above indicated steps are repeated until the infeed point reaches or exceeds " $X_a + U$ ". At such point, the control takes (\emptyset, \emptyset) for $(XOFF, ZOFF)$ and performs cutting along the path offset from the specified contour by the amount (U, W) .

At the end of the defined contour, Z-axis moves to the same Z coordinate position as the AP starting point and then X-axis moves to the AP starting point.

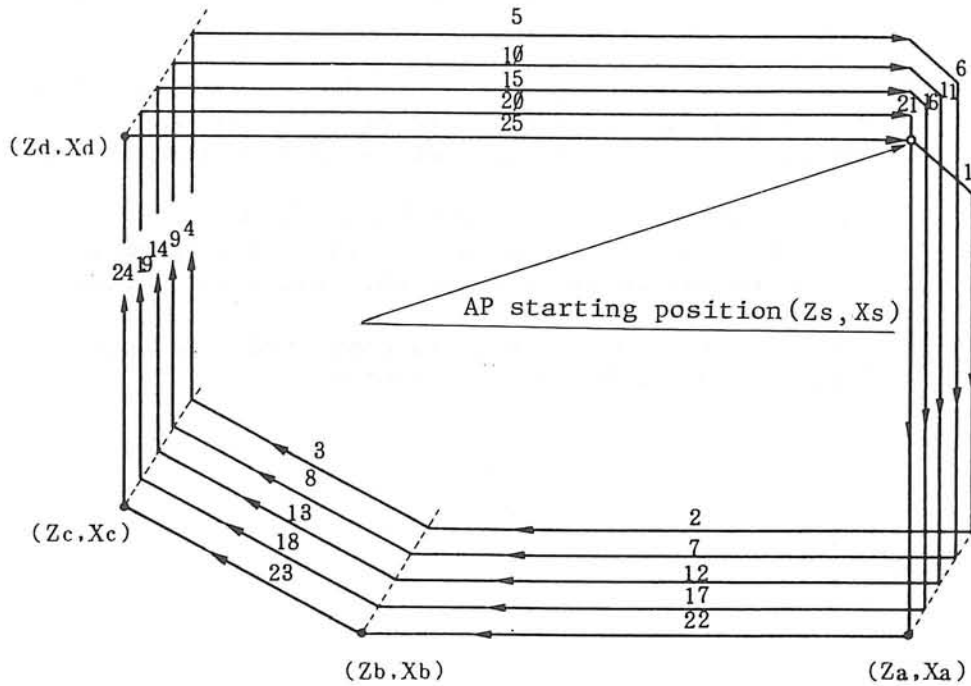


- 9) This completes the rough cut cycle and the block of commands following N0123 is executed.

- 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 cut cycle are selected.
 - 3) With the NAP20 command in block N0223, the control searches the program assigned with the program name NAP20. Finish cut cycle in bar turning mode is performed on this program.
 - 4) The finish cut cycle is performed following the dimension data provided in the contour definition program in the specified cutting conditions for the finish cut cycle.
 - 5) After the finish cut cycle is completed, the commands in the block following N0223 are executed.

19-5-3. AP Mode III (Continuous Thread Cutting Cycle)

(1) Tool Path and Program - G34 and G35 Mode



Contour Definition

```

NAP4Ø G81
NØ4Ø1 GØØ Xa Za
NØ4Ø2 G34 Xb Zb E F J
NØ4Ø3 Xc Zc
NØ4Ø4 GØ1 Xd Zd
NØ4Ø5 G8Ø
    
```

Programming Calling for Thread Cutting Cycle

```

NØ141 GØØ STM
NØ142 Xs Zs
NØ143 G88 NAP4Ø M32(M33, M34) M73(M74, M75) B H D U
    
```

(2) Outline of continuous Thread Cutting Cycle

- a) With the commands in 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 amount for finish cut

Two types of M codes are used to select the mode of thread cutting and tool infeed pattern.

G88 NAP40 calls out contour definition program and executes the required thread cutting cycle (AP Mode III).

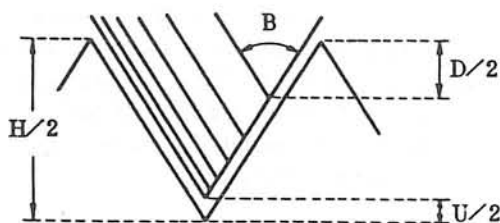
For details of thread cutting cycle, refer to (3).

Note 1: For thread cutting on end face, use G80 - G82 for defining the thread contour as in AP Modes I and II. Program M27 which selects reference axis of thread lead on X-axis in G34/G35/G112/G113 block. Finish allowance is specified by a W word instead of U word.

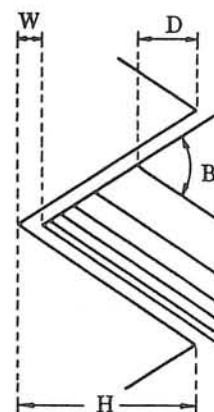
(3) Detail of Thread Cutting Cycle

- a) The data necessary for thread cutting cycle are as follows:

Thread Cutting in
Longitudinal Direction



Thread Cutting in
Transverse Direction
(on End Face)



B : Tip point
 H : Thread height
 D : Depth of cut in the 1st cycle
 U(W): Finish allowance

b) M Code Specifying Thread Cutting Mode and Infeed Pattern

In the block calling for continuous thread cutting cycle, M codes specifying thread cutting mode and infeed pattern should be specified.

1) M code specifying 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.

2) M code specifying infeed pattern:

M73 Infeed pattern 1
 Infeed is made by D (in diameter) in each thread cutting cycle until 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 finishing allowance 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 pattern 3
 The depth of cut in the first thread cutting cycle is "D", and that in the second thread cutting is $(\sqrt{2}-1) \times D$. [The depth of cut in the nth thread cutting cycle is $(\sqrt{n}-\sqrt{n-1}) \times D$.] The thread cutting cycle is repeated until "H - U (W)" position is reached. When $(\sqrt{n}-\sqrt{n-1}) \times D$ becomes smaller than U (W) during the cycle, infeed amount to be taken after that becomes U (W). Finally, the finishing cycle is performed with infeed amount of U (W). If no U (W) word is specified, finishing cycle is not performed.

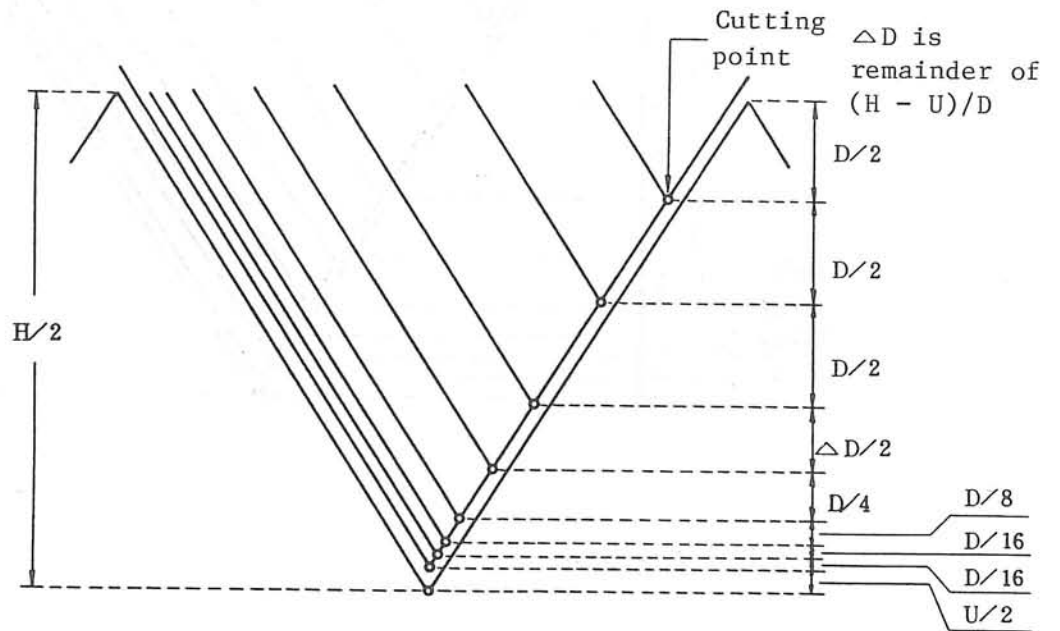
Note: Since X commands are specified in diameter, actual infeed amount is "D/2"

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

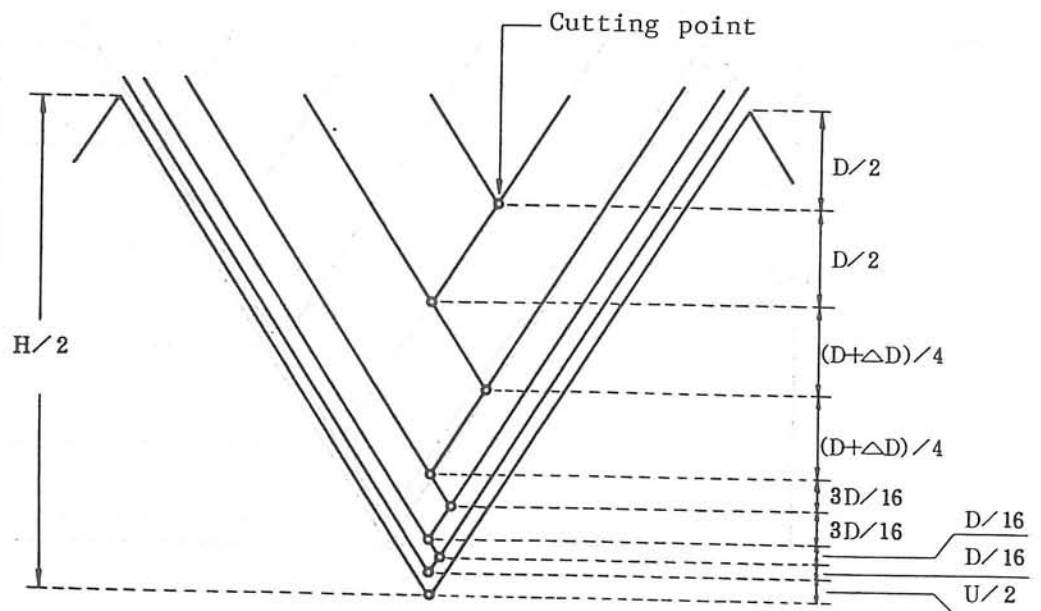
With combining the M codes specifying cutting mode and infeed pattern, six types of thread cutting cycles are available for longitudinal thread cutting cycle and transverse thread cutting cycle, respectively.

1) Longitudinal Thread Cutting Cycles

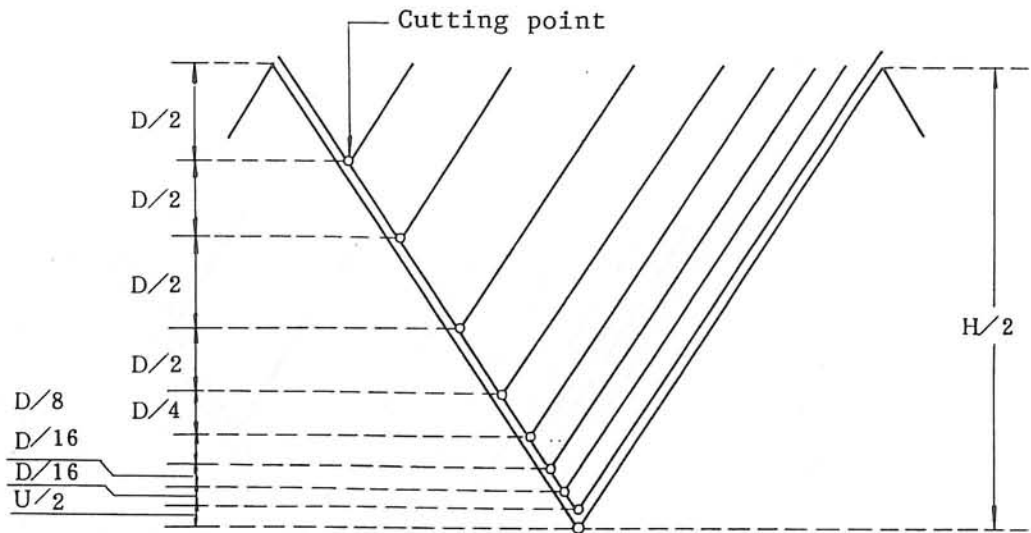
M32 + M73 Mode



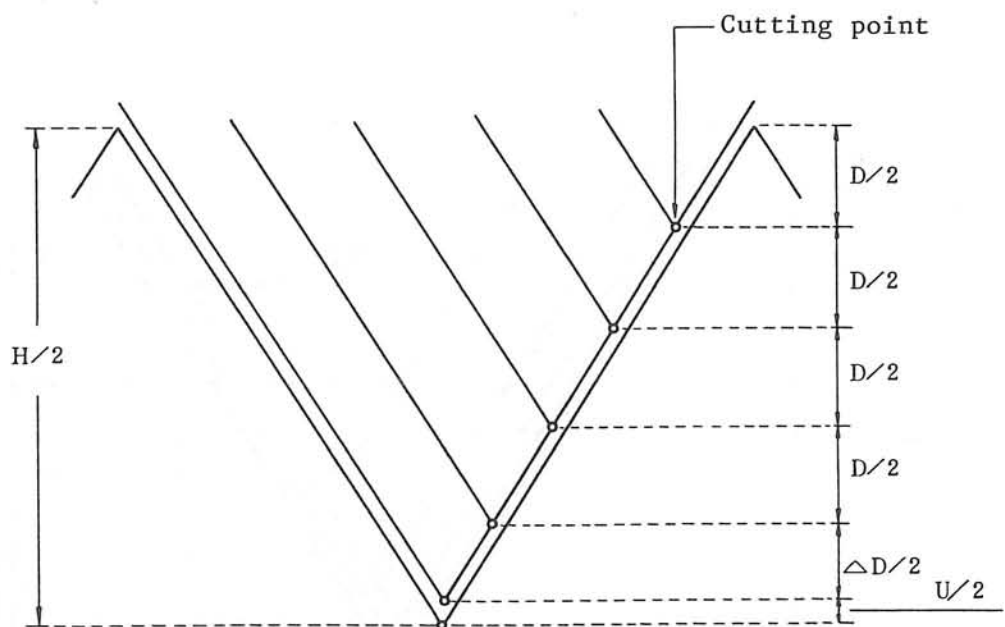
M33 + M73 Mode



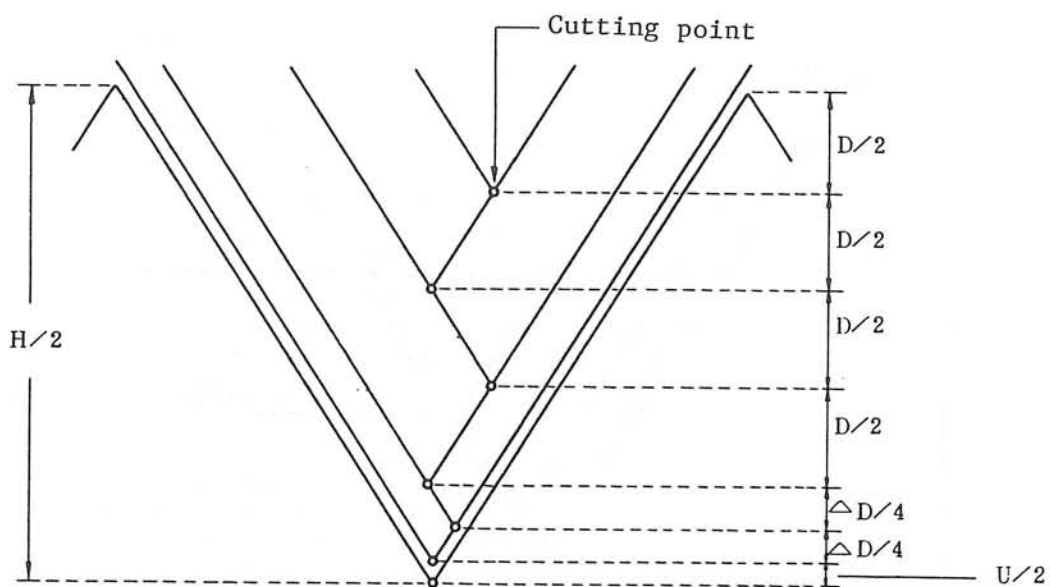
M34 + M73 Mode



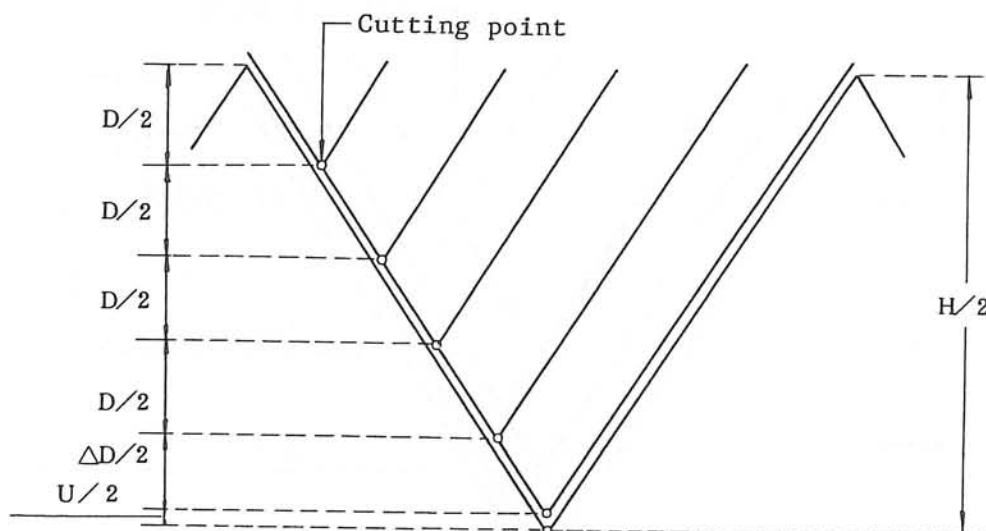
M32 + M74 Mode



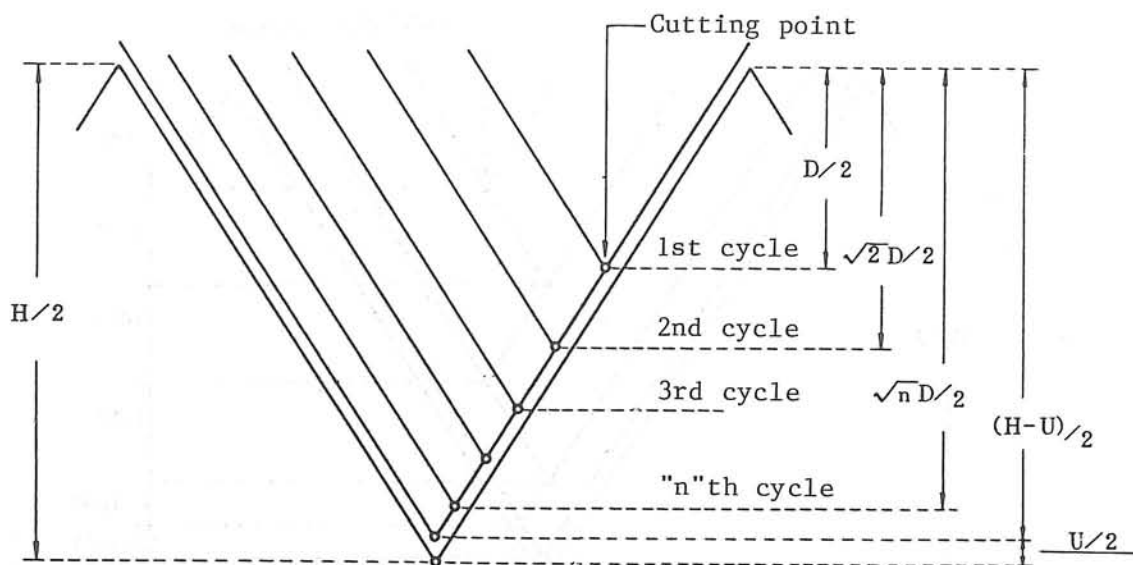
M33 + M74 Mode



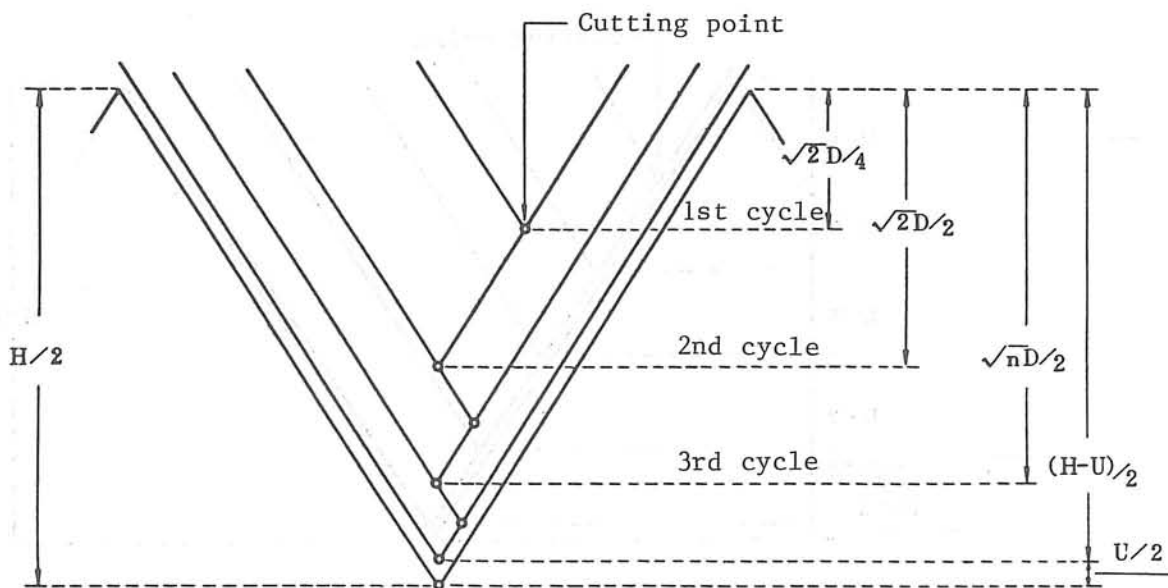
M34 + M74 Mode



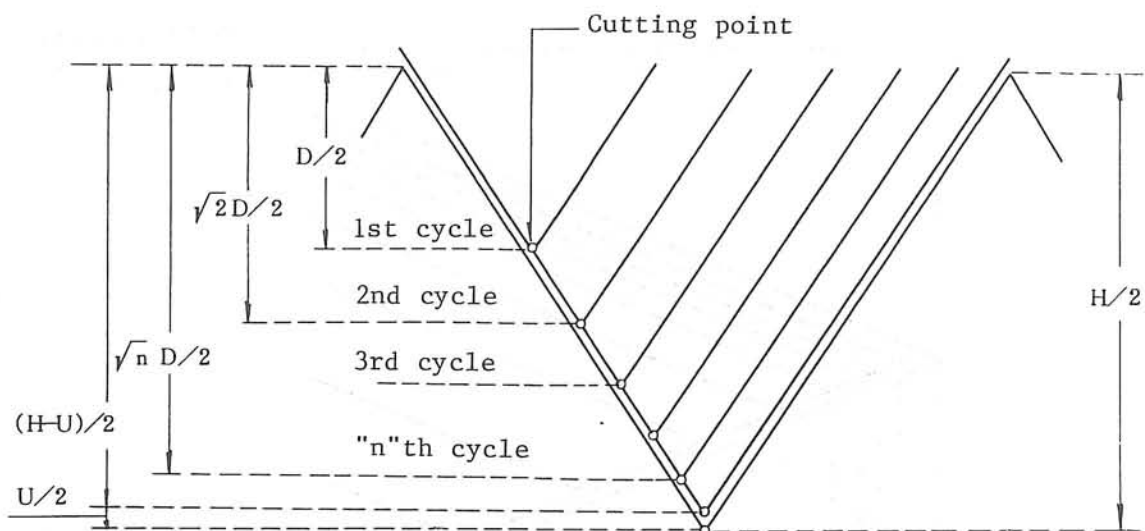
M32 + M75 Mode



M33 + M75 Mode

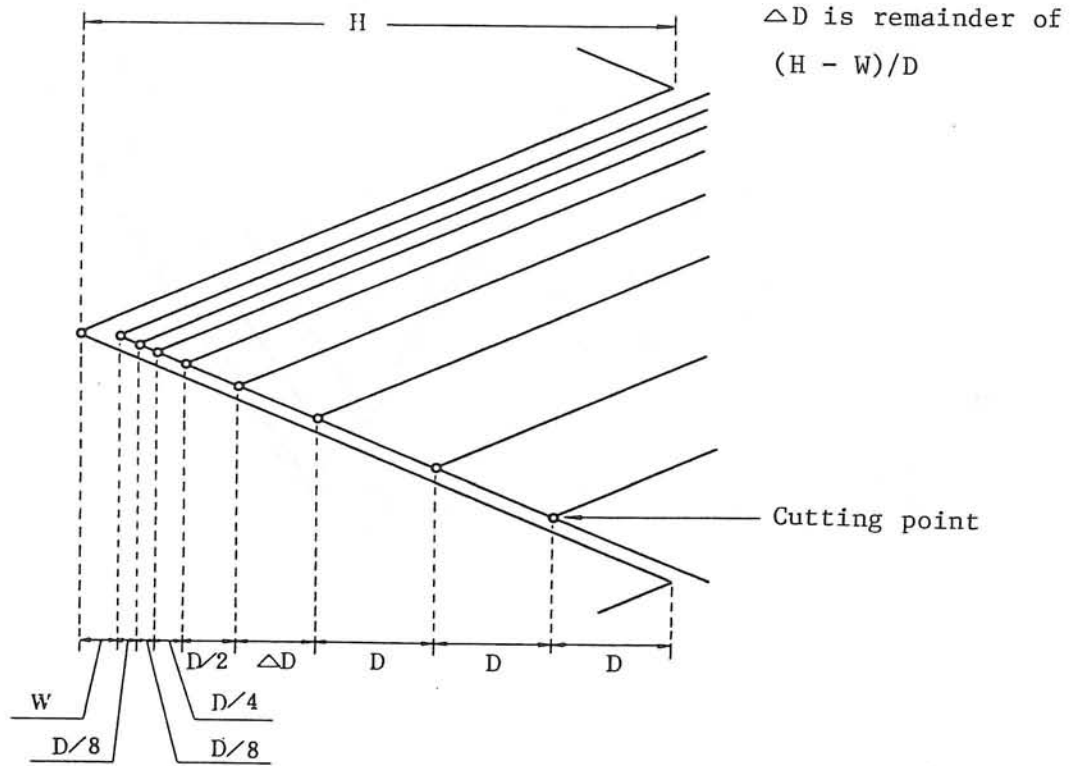


M34 + M75 Mode

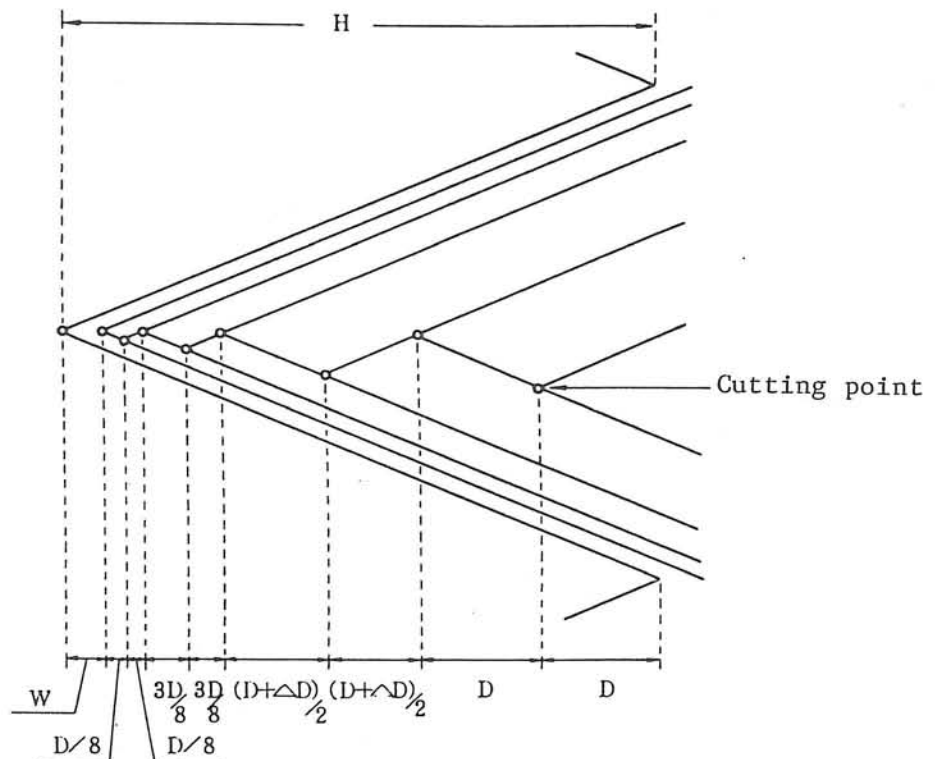


2) Transverse Thread Cutting Cycles

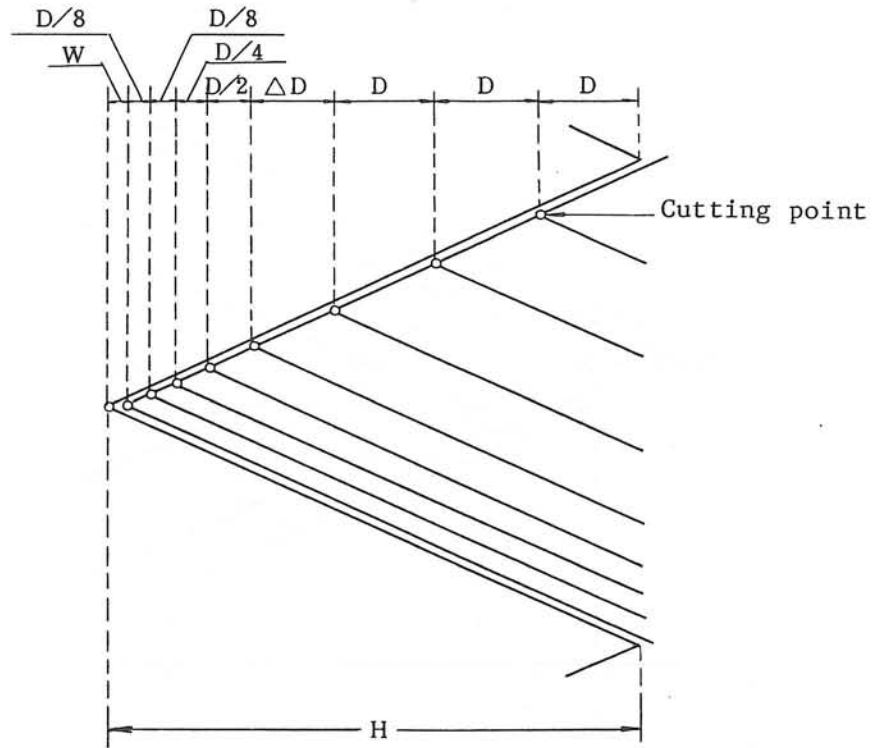
M32 + M73 Mode



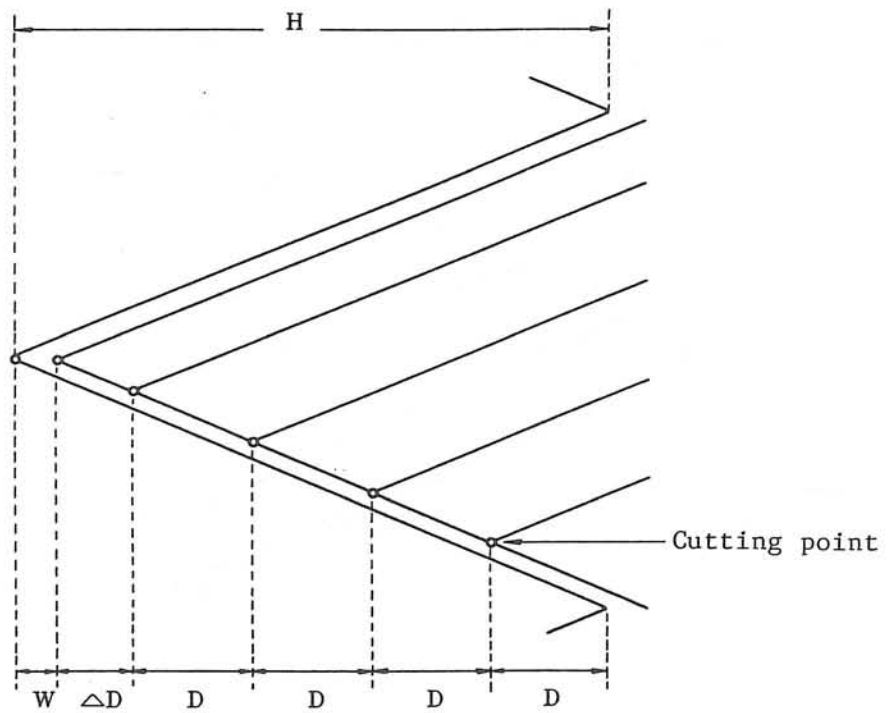
M33 + M73 Mode



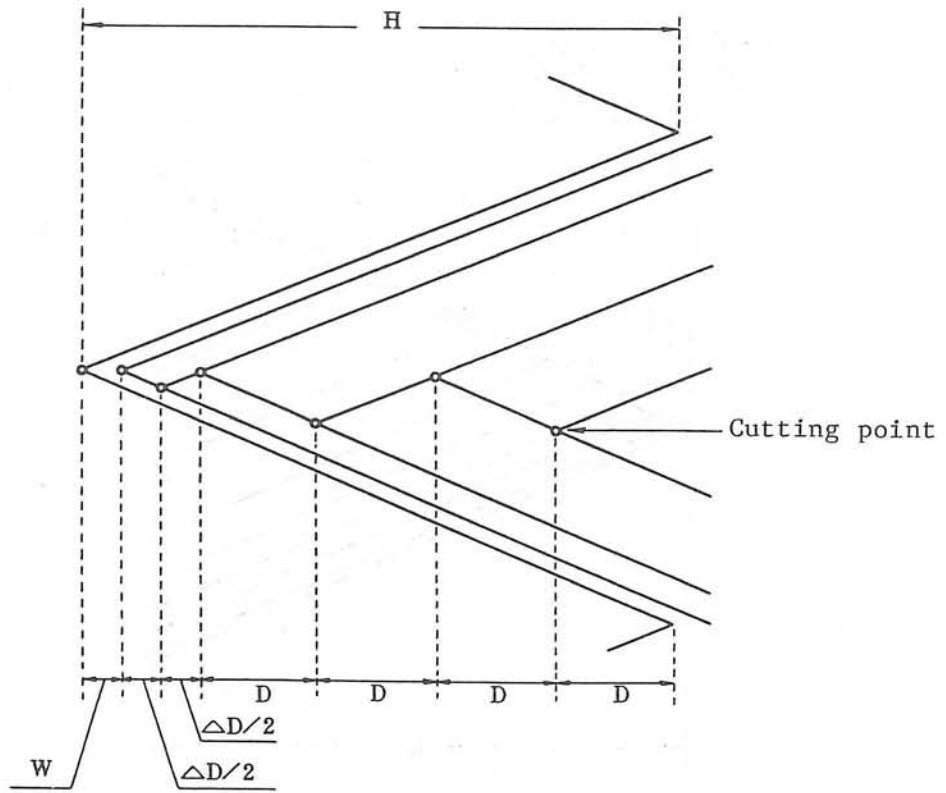
M34 + M73 Mode



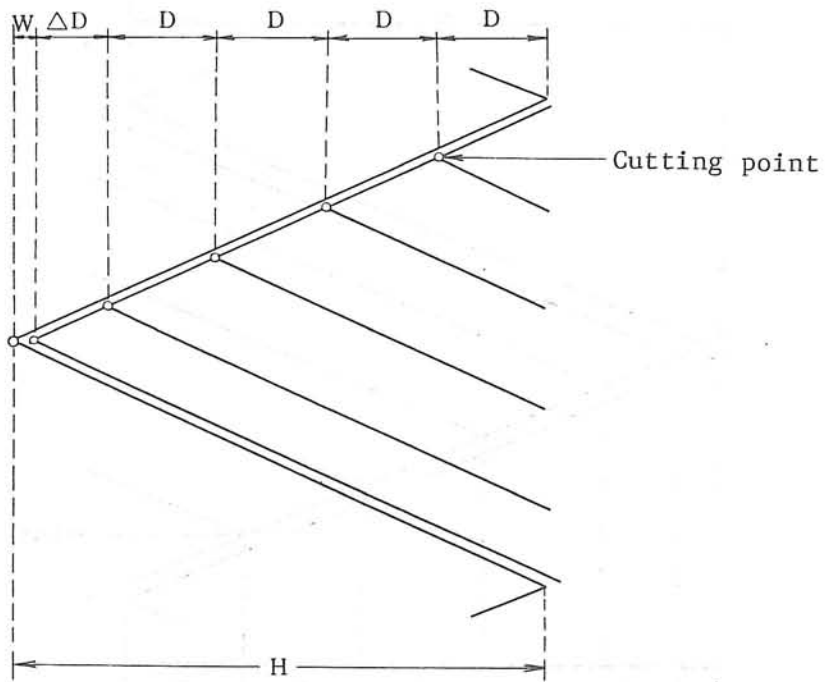
M32 + M74 Mode



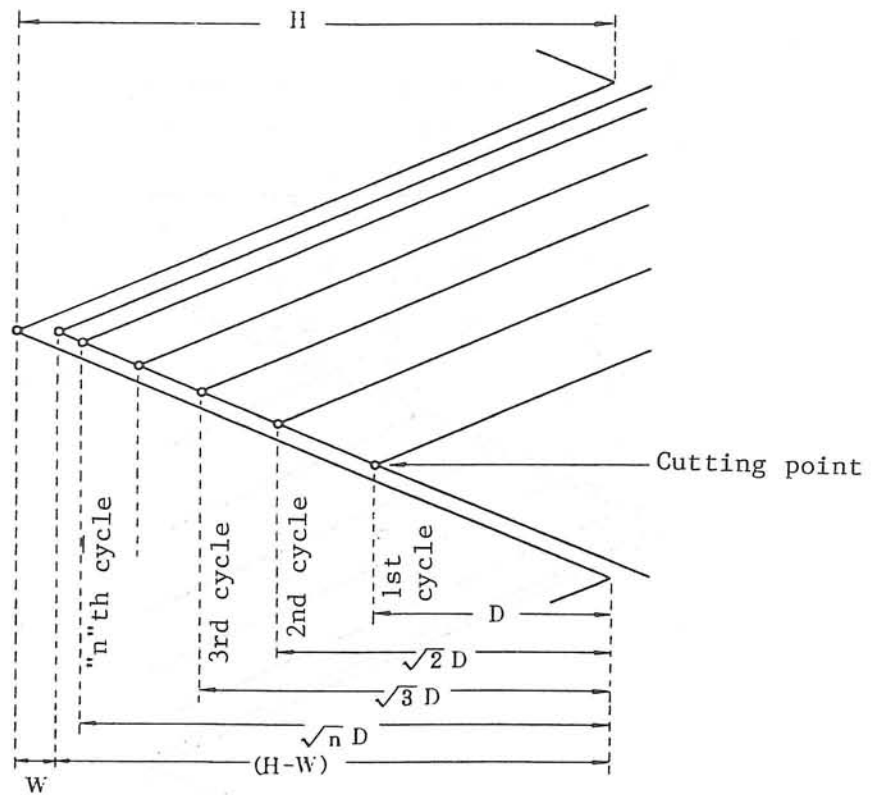
M33 + M74 Mode



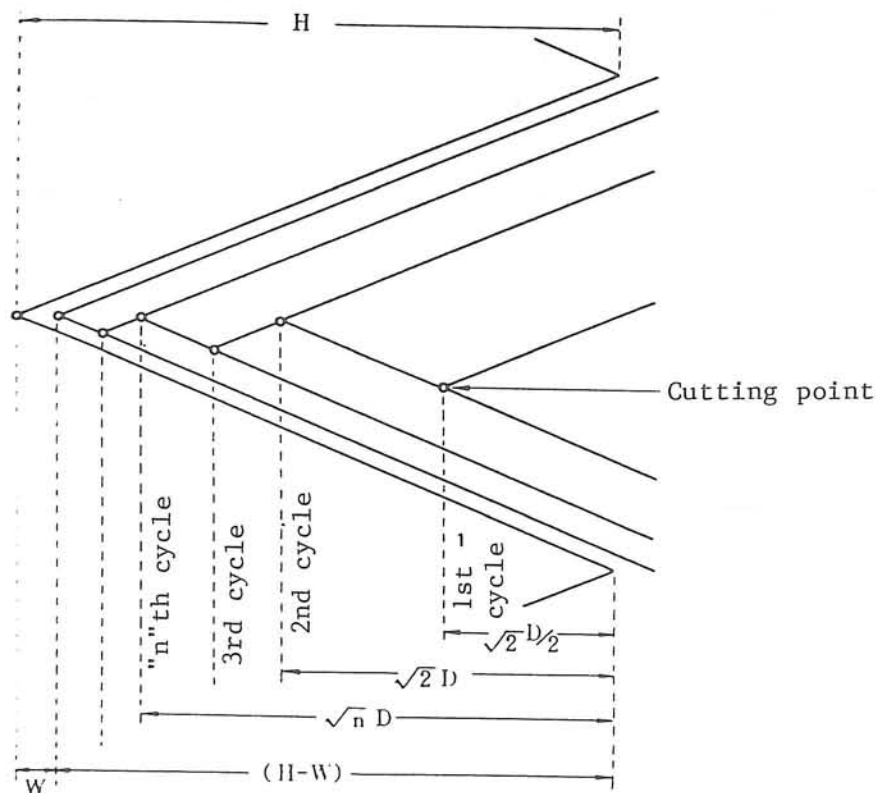
M34 + M74 Mode



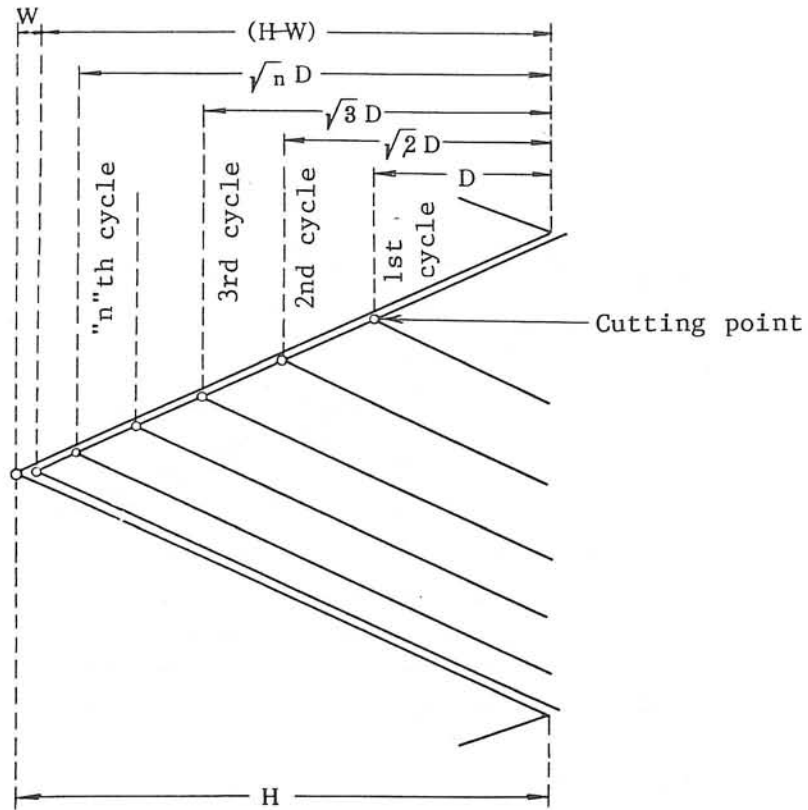
M32 + M75 Mode



M33 + M75 Mode



M34 + M75 Mode



19-6. 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 G81 and G82 codes used to indicate the start of contour definition must be assigned with a proper sequence name.
- (3) For absolute or incremental programming, G90 and 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 contour definition program.
- (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, G90, G91, G94, G95, G96, G97, G112 and G113, those effective when G85, G86 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 provided 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.

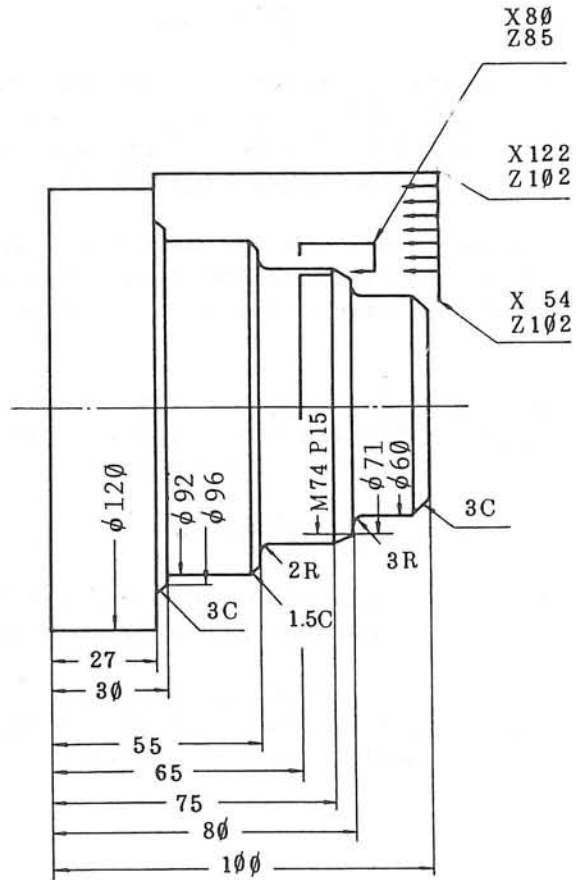
```

NLAP1 G81
N0001 G00           Xa Za
N0002 G41(G42)G00   Xb Zb Fb
      :
      :
      :
N0010 G40           X Z
      :
N0011 G80
  
```

Be sure to activate and cancel the LAP function between G81 (G82) block and G80 block.

19-7. APPLICATION OF LAP FUNCTION

In this section, an example using the LAP Mode I is provided.



Example of Program

O0001										
NAP1	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				
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							(*1)
N101						S900	T0101	M43	M03	(*2)
N102		X122								(*3)
N103	G85	NAP1	D8	U0.2		F0.45				(*4)
N104	G00	X800	Z102							
N105						S1000	T0303			(*5)
N106	G87	NAP1								(*6)
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 definition

- *1: Tool change position
- *2: S, T and M for rough cut cycle
- *3: Rough cut starting point
- *4: Calling for bar turning rough cut cycle
- *5: S, T and M for finish cut cycle
- *6: Calling for finish cut cycle

Note: Contour defining program beginning from G81 and ending with G80 may be entered any part within the above indicated program.

19-8. SYNCHRONIZATION CODES FOR LAP

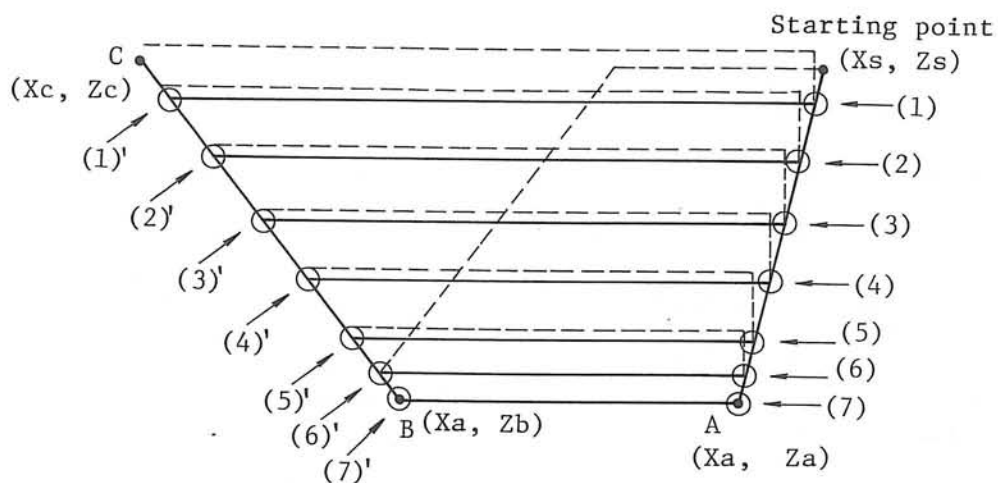
If a P code is designated within the G85, G86, G87 and G88 sequences, the execution of the programmed LAP function by the turrets A and B is synchronized at the start point of the LAP. If a P code is designated within the contour definition sequences between the G81 (or G82) code and the G80 code, execution of the LAP by these turrets A and B is synchronized at each cycle start position of the LAP cycles.

Note 1: P codes in the contour defining sequence (sequences beginning with the G81 or G82 code and ending with the G80 code) are independent of P codes designated in other parts of the program. Therefore, P codes in the contour definition sequences may be designated in the order within the sequences.

Note 2: For assuring the synchronization of turrets A and B at each cycle start point of the LAP functions, P codes must be programmed in the G85, G86, G87 and G88 sequences.

Program Example

	OAPT1								
	N100	G13							
	N101	G00	Xt	Zt					
	N102		Xs	Zs	S	T	M		P10
	N103	G85	NLAP1		D	F	U	W	P20
	N104	G00	Xt	Zt					
	N105		Xs	Zs	S	T	M		P30
	N106	G87	NLAP1						P40
	N107	G00	Xt	Zt			M		P50
	NLAP1	G81							
Contour Definition	N001	G00	Xa	Za					
	N002	G01		Zb	F				P10
	N003		Xc	Zc					
	N004	G80							
	N200	G14							
	N201	G00	Xt	Zt					
	N202		Xs	Zs	S	T	M		P10
	N203	G85	NLAP1		D	F	U	W	P20
	N204	G00	Xt	Zt					
	N205		Xs	Zs	S	T	M		P30
	N206	G87	NLAP1						P40
	N207	G00	Xt	Zt					P50
	NLAP1	G81					M		
	N001	G00	Xa	Za					P10
	N002	G01		Zb	F				
	N003		Xc	Zc					
	N004	G80							
	:								
	:								



P2Ø in N1Ø3 and P1Ø in NØØ2, or P2Ø in N2Ø3 and P1Ø in NØØ2 allows the synchronization of the start of turrets A and B at each cutting cycle start point (1) through (7) above.

P4Ø in N1Ø6 and P1Ø in NØØ2, or P4Ø in N2Ø6 and P1Ø in NØØ2 allows the synchronization of the start of turrets A and B at the starting point of the finishing cycle (point (7)).

Programming P2Ø in NØØ3 will allow the synchronization of the start of the cutting at points (1)' through (7)'.

SECTION 2Ø TOOL NOSE RADIUS COMPENSATION FUNCTION

2Ø-1. OUTLINE

2Ø-1-1. General Description

The data processing performance of the OSP5ØØØL-G/OSP5ØØØØL-G series is all the more enhanced by one distinctive feature: The Tool Radius Compensation Function.

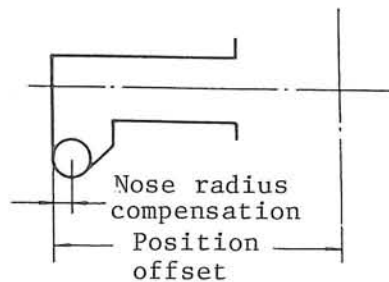
The tool tip point radius of most cutting tools used in turning operation is the cause of inconsistencies between the designated tool paths and the actually finished workpiece contour. With the tool radius compensation function, such geometric error is automatically compensated by simple programming.

In die sinking or milling operation, cutting is performed by a large diameter milling cutter. There, control systems are equipped with the Cutter Offset function where the actual tool path generated is offset from the designated path by the amount equivalent to the radius of the cutter used. This is a standard function with OSP5ØØØØ series controllers. However, in comparison with the cutter offset function, the tool nose compensation function offers special features that are requisites for lathes.

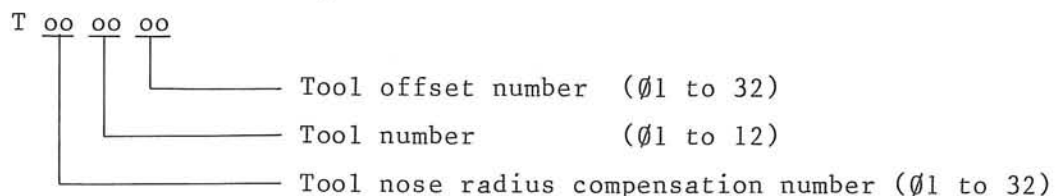
2Ø-2. TOOL NOSE RADIUS COMPENSATION FOR TURNING OPERATION

2Ø-2-1. 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.



The tool nose radius function is activated by a six-digit T command:



20-2-2. Tool Nose Radius Compensation at Discontinuous Point

One of the characteristics of turning operations is that there are many discontinuous points along a tool path. What more, many of those point have angles smaller than 180° deg.

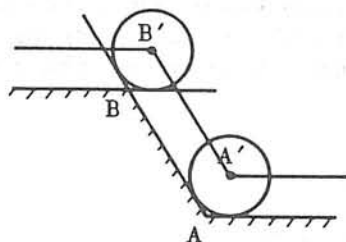


Fig. 20-1 Tool Path in Path 500L-G/5000L-G

Point A in Fig. 20-1 constitutes a discontinuous point with an angle less than 180° deg. In this case as the OSP500L-G/5000L-G features the tool nose radius compensation function, it can generate a tool path by simply entering the coordinates of points A and B.

2Ø-3. COMPENSATION OPERATION

2Ø-3-1. Geometrical Cutting Error Due to Tool Nose Radius

If cutting along paths A-B-C-D-E in Fig. 2Ø-3 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 in Fig. 2Ø-2 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.

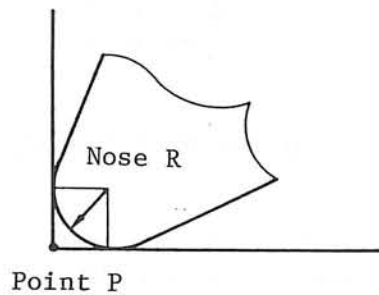


Fig. 2Ø-2 Tool Setting Point

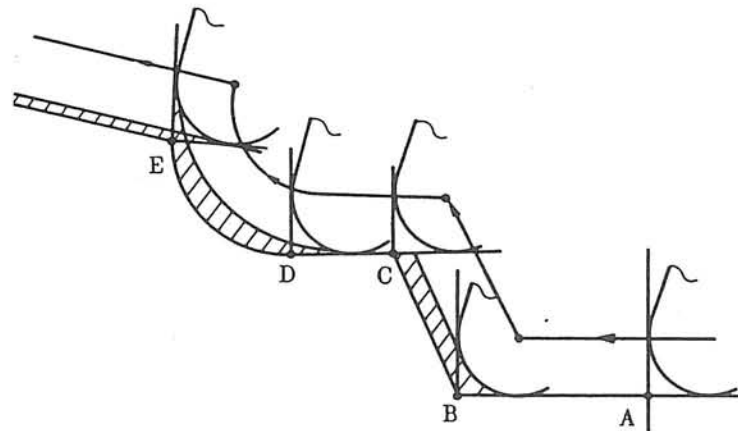


Fig. 2Ø-3 Tool Path and Resulting Error
- Without Tool Nose Radius Compensation -

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

2Ø-3-2. Compensation Movement

With the tool nose radius compensation function activated, the tool path is compensated as illustrated in Fig. 2Ø-4 to eliminate the portions left uncut, shown in Fig. 2Ø-2. This assures accurate finish as programmed.

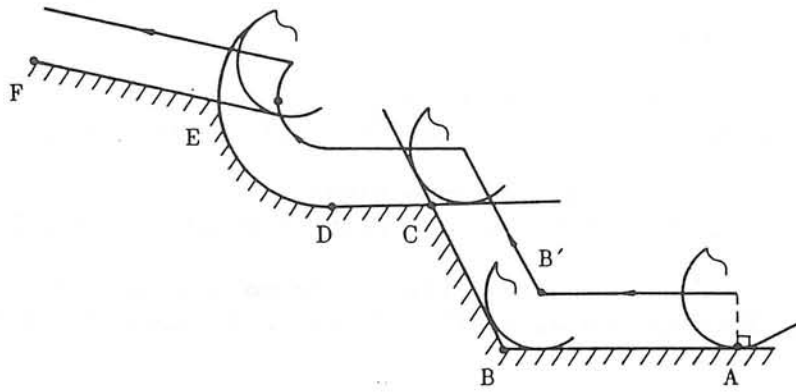


Fig. 2Ø-4 Tool Path with Tool Nose Radius Compensation

2Ø-4. PROGRAMMING

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

2Ø-4-1. G Codes

- G4Ø: 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.

Note 1: The term indicating the side of the workpiece, right or left, is determined according to the direction in which the tool is advancing.

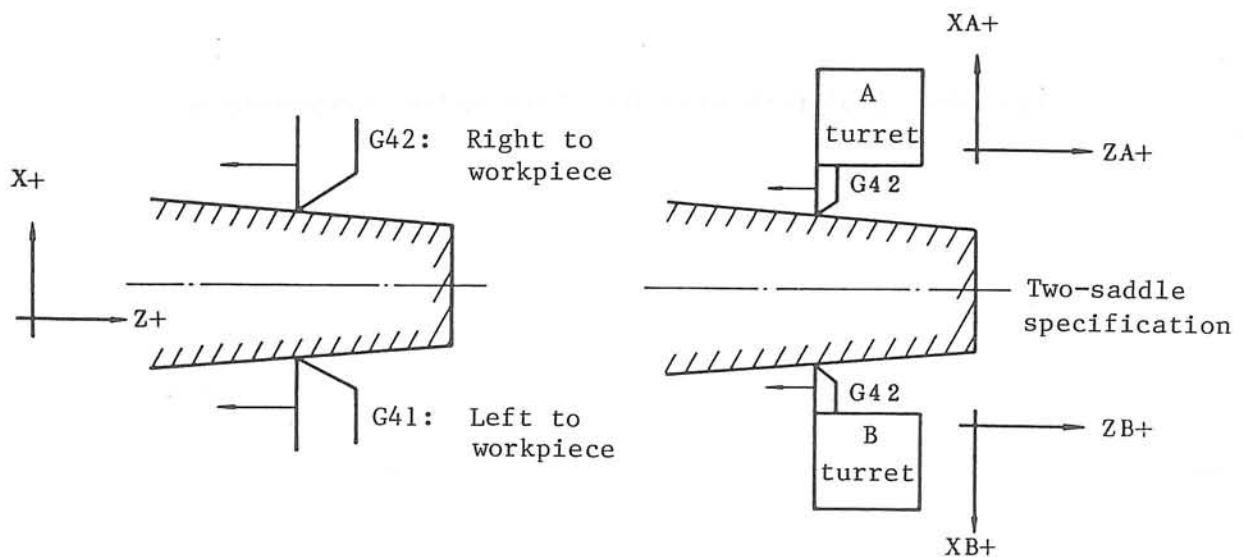
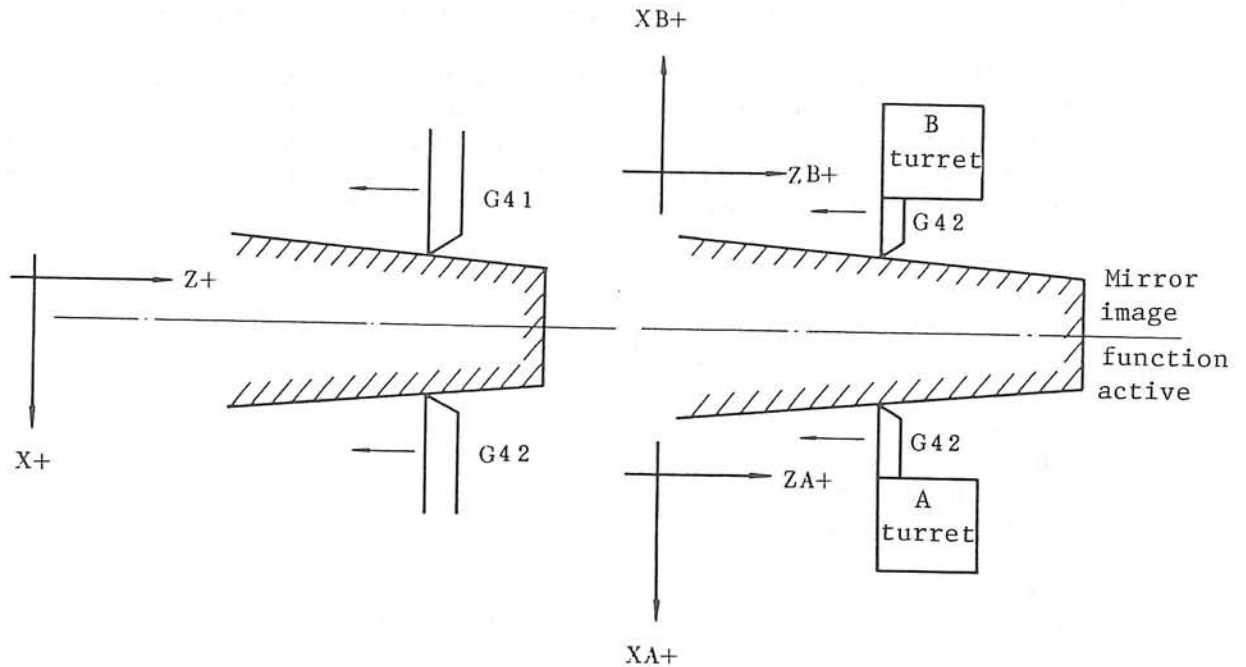


Fig. 2Ø-5 Designation of G41 & G42

Since G41 and G42 codes are selected to meet the coordinate 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. LH and LS series, for instance.

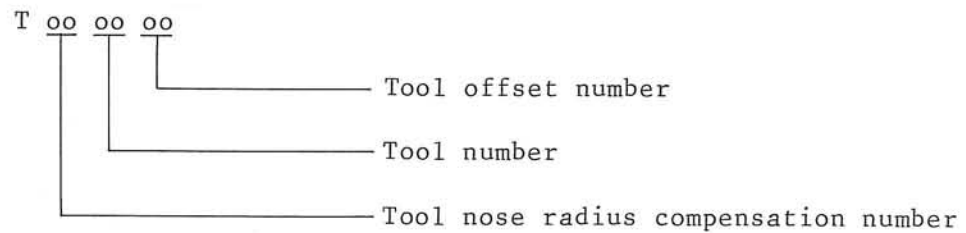
Designation of those G codes for lathes equipped with a rear turret (with the mirror image function) can be made in the same manner as for the two-saddle model. However, turret designation, A and B is inverted.



2Ø-4-2. T Codes

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

To set offset data through the keyboard, 32 pairs of compensation data for the tool nose radius and tool offset function can be entered (Ø1 through 32). For the tool number, Ø1 through 12 are available.



2Ø-4-3. Program Example

Provided below is the program to cut the contour shown in Fig. 2Ø-4.

Dimensioning commands, miscellaneous function commands, and other commands used for cutting that contour without tool nose radius compensation feature are all designated in the same manner; only G codes calling for tool nose radius compensation mode and T command in which the nose radius compensation number is added are newly entered in the program.

```

Nooo ... ..
Nooo G42 GØØ Xa Za TØ1Ø2Ø3
Nooo GØ1 Zb F
Nooo Xc Zc
Nooo Zd
Nooo GØ2 Xe Ze I K
Nooo G4Ø GØØ Xf Zf
Nooo ... ..

```

CAUTION

If no point of intersection is obtained from the commanded dimension words, a program alarm occurs.

2Ø-5. TOOL NOSE RADIUS COMPENSATION DATA

As seen in the previous section, programming procedure to activate the tool nose radius compensation function has been simplified. However, to finish workpieces accurately using this function, the tool nose radius of the tool to be used must be measured precisely and the measured value entered correctly in the NC memory.

2Ø-5-1. Measuring Nose Radius

Measure the center of the tool nose circle with respect to the tool tip reference point which is taken as the imaginary tool tip point for tool presetting. See Fig. 2Ø-6 below.

The imaginary tool tip point indicates the ideal tool tip point which can be expressed only by the tool offset amounts without tool nose R. Actually, such tools are not present.

When the control has no tool nose radius compensation function, it controls the coordinated axes motion so that the tool tip reference point follows the programmed path.

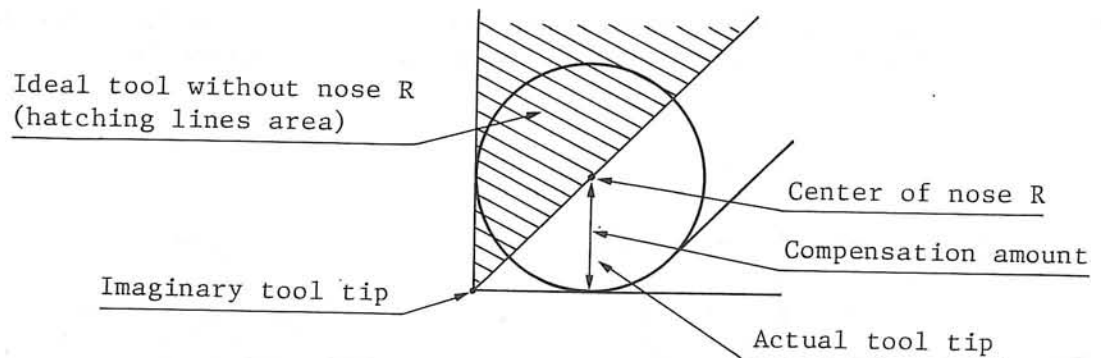


Fig. 2Ø-6

In the measurement of nose R compensation data, both the nose R compensation amount and the direction of nose R center in reference to the imaginary tool tip.

Direction of nose R center in reference to the imaginary tool tip is expressed in the following two ways:

- a) By positive and negative signs of X, Z compensation amounts
- b) By a P number

If method a) is used, positive and negative signs are determined from the position of the nose R center in reference to the origin as in Fig. 2Ø-7.

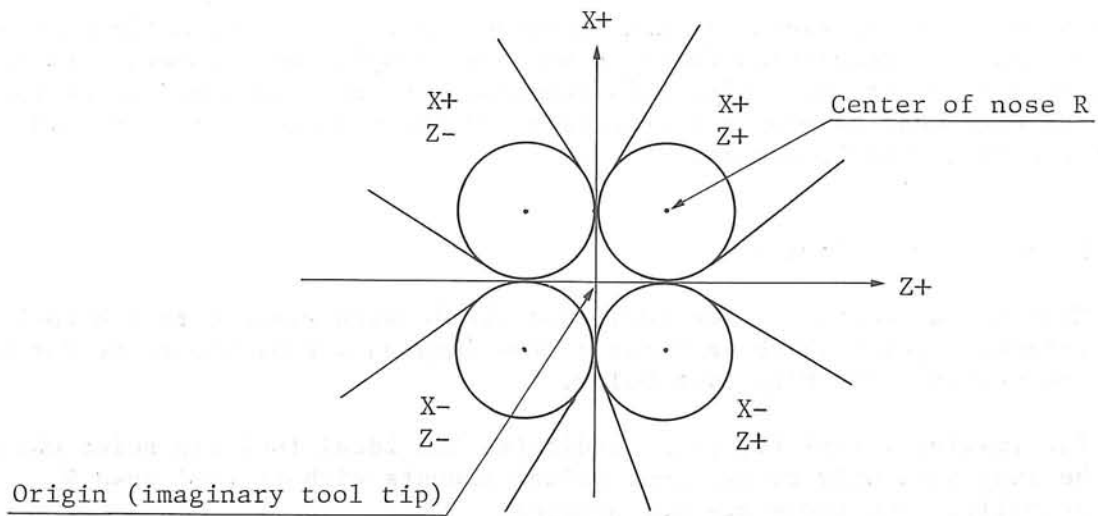
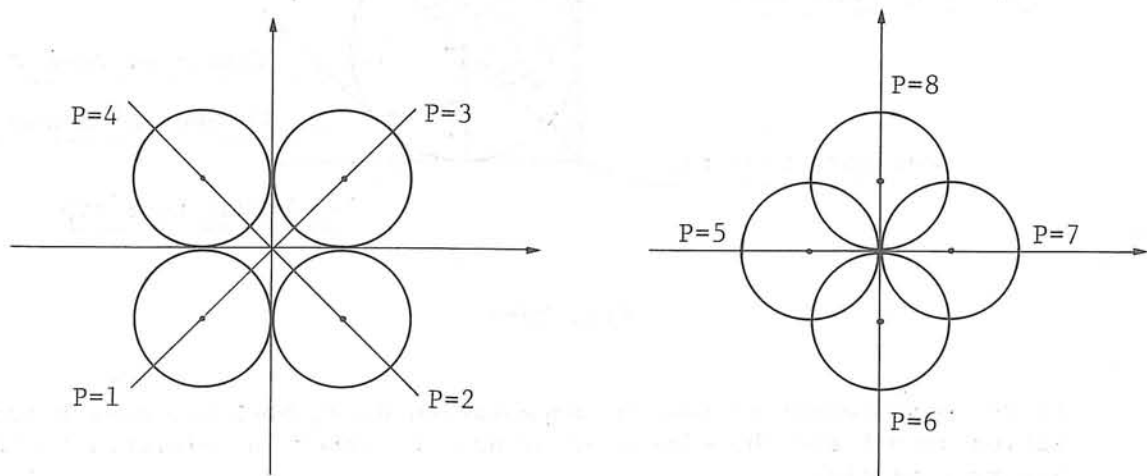


Fig. 20-7 Signs of Compensation Amount by Nose R Center Position

With method b), coded numbers are assigned in advance for individual nose R center position orientation to distinguish the directions.



Compensation amount = \emptyset when $P = 9$

Fig. 20-8 P Numbers for Nose R Center Positions

20-5-2. Setting Compensation Amounts

Set the tool compensation amount at the NOSE R COMP columns at the TOOL DATA SET screen. The compensation amounts can be set in the same manner as setting tool offset amounts.

Orientation of nose R center in reference to the imaginary tool tip may be set either by positive or negative sign preceding the compensation amount or P number. If P numbers are used, set the number at P column.

Tool Offset and Nose R Compensation Data Setting Screen

a) Monochrome specification

TOOL DATA SET						N	0
Page 1	A turret			UNIT 1mm			
BC=19	* TOOL OFFSET *			* NOSE-R COMP *			
NO.	T	X A	Z A	X A	Z A	P	
1	A	0.000	0.000	0.000	0.000	0	
2	A	0.000	0.000	0.000	0.000	0	
3	A	0.000	0.000	0.000	0.000	0	
4	A	0.000	0.000	0.000	0.000	0	
5	A	0.000	0.000	0.000	0.000	0	
6	A	0.000	0.000	0.000	0.000	0	
7	A	0.000	0.000	0.000	0.000	0	
8	A	0.000	0.000	0.000	0.000	0	
9	A	0.000	0.000	0.000	0.000	0	
10	A	0.000	0.000	0.000	0.000	0	
11	A	0.000	0.000	0.000	0.000	0	
12	A	0.000	0.000	0.000	0.000	0	
LAST DATA		* 0.000	X= 200.000	Z= 200.000	TOOL	T 0	

ADD	CONST. ADD	CONST. SUB	ITEM1	ITEM2	ENDTEND
-----	------------	------------	-------	-------	---------

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
-----	-----	-----	-----	-----	-----	-----	-----

- 4) After pressing the function key [F1] (SET), key in [Ø][.][3][5] and press the WRITE key. Then, move the cursor to the ZA column and key in [Ø][.][3][5] after pressing the function key [F1] (SET), then press the WRITE key.

Make sure that the compensation amounts have been correctly input.

The maximum setting value of the nose R compensation amount is +999.999 mm.

- 5) Set the imaginary tool tip orientation by a coded number. First locate the cursor at the P column of the tool offset number 1. After pressing the function key [F1] (SET), key in the imaginary tool tip orientation number and WRITE key.

For the details of the tool tip orientation number, refer to 2Ø-5-3.

20-5-3. Relationship between Imaginary Tool Tip Orientations and Coded Numbers

The relationships between the imaginary tool tip point orientations and the coded numbers are shown in Fig. 20-8.

When the data set as the tool nose radius compensation amount is required to be used as set, set the imaginary tool tip orientation at "0".

When the tool tip orientation number "2" is set, the calculation below is carried out. This means that the signs used for setting the nose radius compensation data are ignored.

Nose R on X-axis = - (Set value for nose R on X-axis)
 Nose R on Z-axis = + (Set value for nose R on Z-axis)

Graphic Specification:

TOOL DATA SET N 0

Page 1 UNIT 1mm
 BC=1A * TOOL OFFSET * * NOSE-R COMP *

NO.	T	X A	Z A	X A	Z A	P
1	A	0.000	0.000	0.000	0.000	0
2	A			0.000	0.000	0
3	A			0.000	0.000	0
4	A			0.000	0.000	0
5	A			0.000	0.000	0
6	A			0.000	0.000	0
7	A			0.000	0.000	0
8	A			0.000	0.000	0
9	A			0.000	0.000	0
10	A			0.000	0.000	0
11	A			0.000	0.000	0
12	A	0.000	0.000	0.000	0.000	0

LAST DATA * 0.000 X= 200.000 Z= 200.000 TOOL T 0

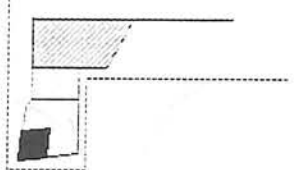
	ADD	CONST. ADD	CONST. SUB	GUIDE ON/OFF	ITEM1	ITEM4	[EXTEND]
--	-----	------------	------------	--------------	-------	-------	----------

F 1
F 2
F 3
F 4
F 5
F 6
F 7
F 8

When the guide display below is indicated, set "3". Note that the orientation number is determined from the tool shape selected for each tool in the tool shape selection step and the tool offset number. Therefore, if the setting of these parameters is not correct, wrong numbers will be displayed.

Graphic Specification:

TOOL DATA SET				R	Q
Page 1					
BC=BC				UNIT 1mm	
TOOL NO. 1	* TOOL FORM SELECT *				
	TOOL CODE NO. 1	ROUGH OD-			
	FORM CODE NO. 1				
TOOL EDGE DATA					
TOOL ANGLE	A1=	80.000			
EDGE ANGLE	A2=	3.000			
STICKING OUT	L =	50.000			
OFFSET NO.					
	ON1=	1			
	ON2=	0			
	ON3=	0			
=IB					
=					
SET	ADD	TOOL KIND		ITEM1	ITEM2
F 1	F 2	F 3	F 4	F 5	F 6
					F 7
					F 8



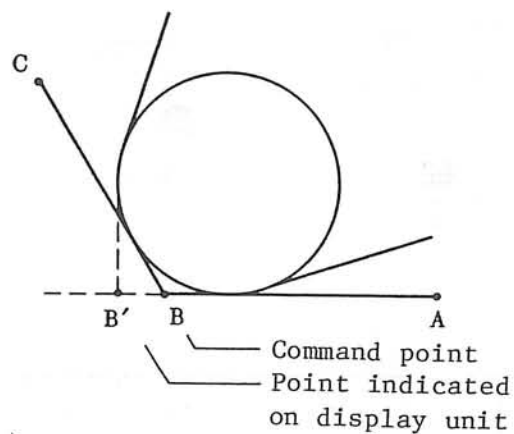
The diagram illustrates a tool profile with a shaded area labeled "INTERFER AREA". The tool has a top edge that is slightly curved, and a bottom edge that is also slightly curved. The shaded area is located between the top and bottom edges, indicating a region of interference or overlap.

20-6. DISPLAY

The CRT displays various data (see Operation Manual for the OSP500L-G/OSP500L-G). The data displayed in the tool nose radius compensation mode are as follows.

20-6-1. Actual Position

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



20-6-2. 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 CRT displays the associated alarm message indicating its contents.

20-7. OTHER REMARKS

20-7-1. Conditions of Nose Radius Compensation Function

- (1) The OSP500L-G/OSP5000L-G usually operates in the 3-buffer mode. See Fig. 20-9. 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.

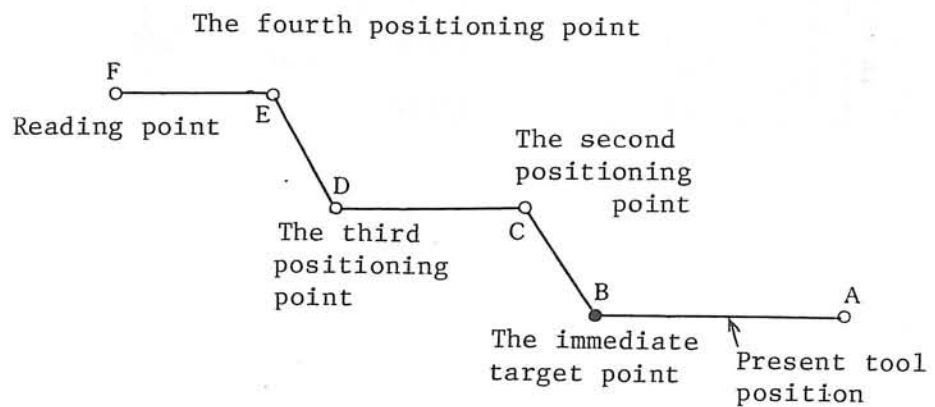


Fig. 20-9 Data in Buffer

- (2) Availability of Nose Compensation on Model with Rear Turret

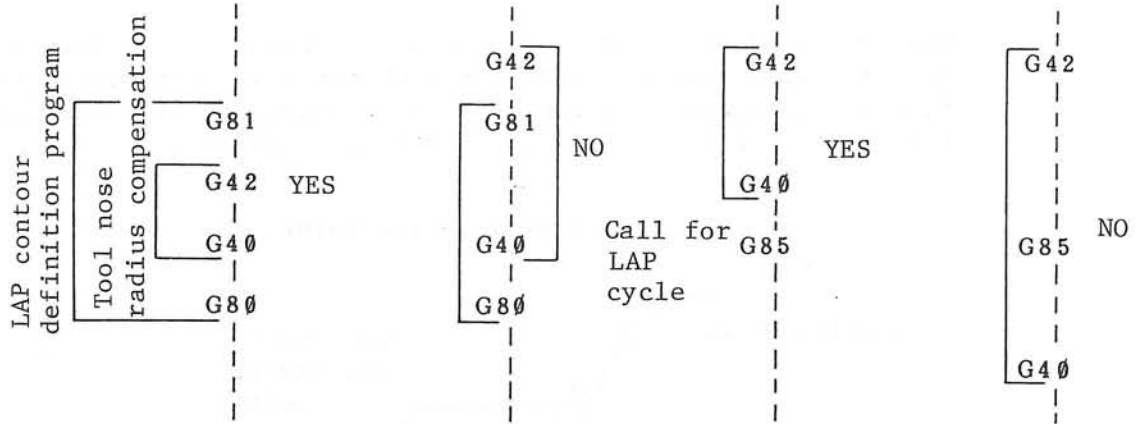
For machines equipped with a rear turret, the tool nose radius compensation function can be activated. However, G code selecting the turret, G13 and G14, cannot be programmed while such a mode is active. If the turret selection is required during the tool nose radius compensation mode, cancel mode one by providing G40 before commanding G13 or G14.

- (3) Availability of Nose Radius Compensation during LAP Mode

The tool nose radius compensation function can be activated during the LAP mode of operation.

(4) Availability of Nose Radius Compensation during Simultaneous 4-Axis Cuts Mode

For two-saddle models, the tool nose radius compensation function can be activated. In this case, programs for respective turrets must contain nose radius compensation program independently.



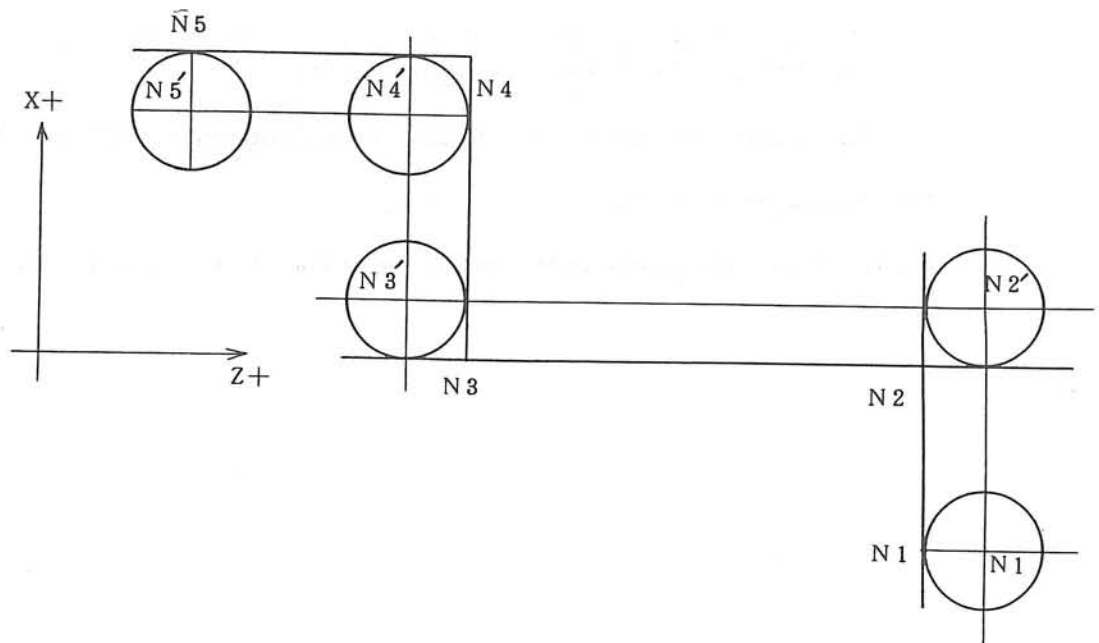
20-7-2. Path of Tool Nose "R" Center in Tool Nose Radius Compensation Mode

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

```



- (1) 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'.

(2) 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'.

(3) To obtain point N4':

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

SECTION 21 APPLICATION OF TOOL NOSE RADIUS COMPENSATION PROGRAMMING

21-1. 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.

CAUTION

The terms "left" and "right" used in tool nose radius compensation mode are defined with respect to the established coordinate system. Therefore, for models LH and LS, the G41 code is used when the cutting tool advances on the left side of the workpiece, viewing the turret from the lower side.

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.

CAUTION

The terms "left" and "right" used in tool nose radius compensation mode are defined with respect to the established coordinate system. Therefore, for models LH and LS, the G42 code is used when the cutting tool advances on the right side of the workpiece, viewing the turret from the lower side.

21-2. BEHAVIOR ON TOOL NOSE RADIUS COMPENSATION MODE ENTRY

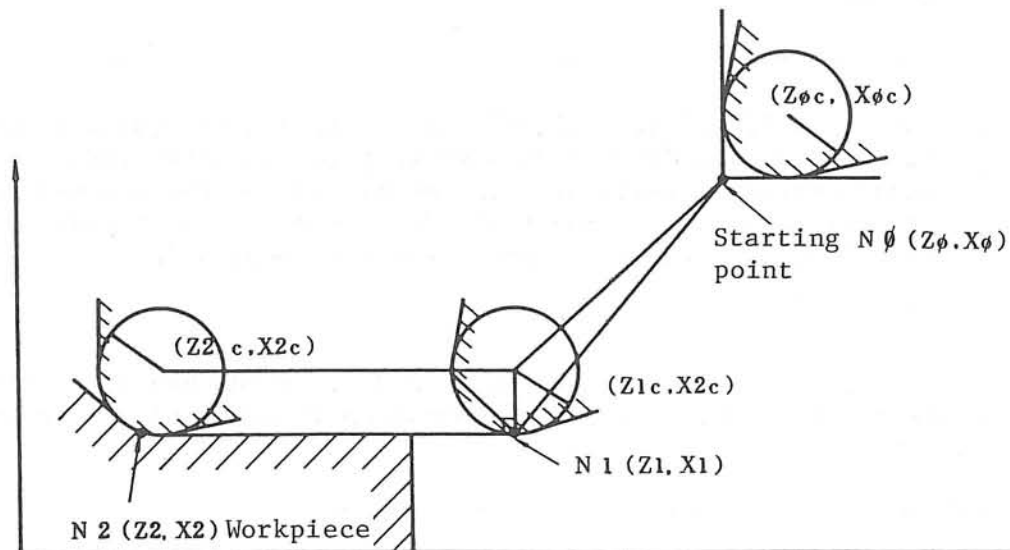
21-2-1. General Description

```

N0 G00 X0 Z0
N1 G42 X1 Z1 T000000
N2 G01 X2 Z2

```

The following case uses the above program to perform OD cuts with an OD turning tool having a positive nose radius compensation amount for each axis $Rx+$, $Rz+$.



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.)

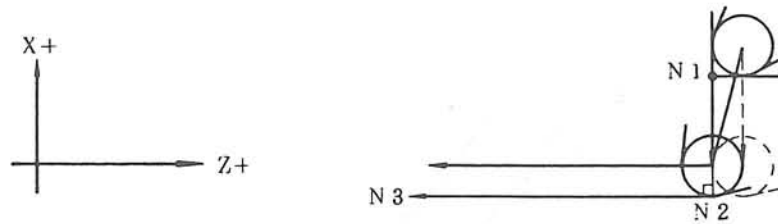
21-2-2. Program Examples

(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:

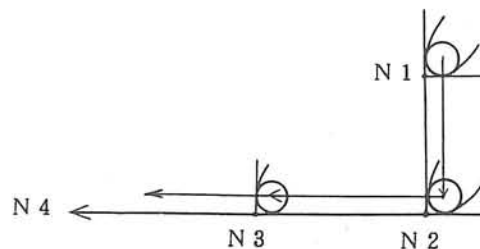
```

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

```

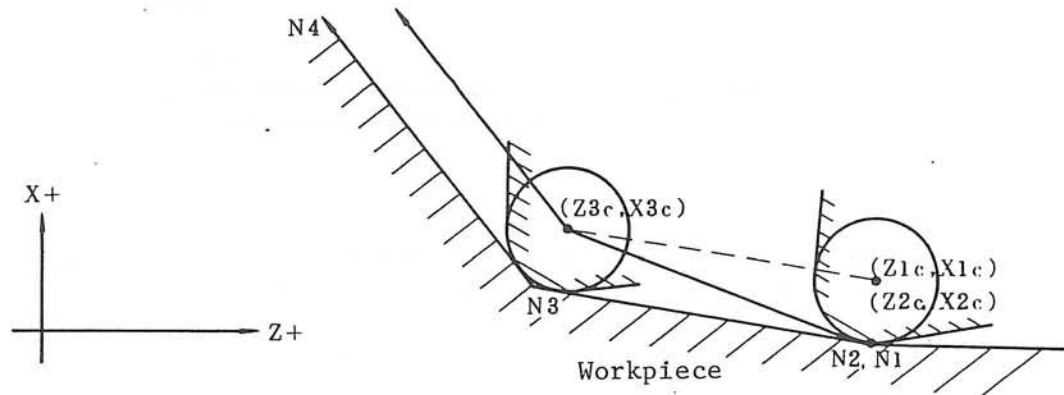
In this program, the G42 block contains only a Z word, and points N2, N3 and N4 are all positioned on the same straight line.

Note: 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 the 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.

Note: 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 G42
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.

21-2-3. I and K Command with G41 and G42

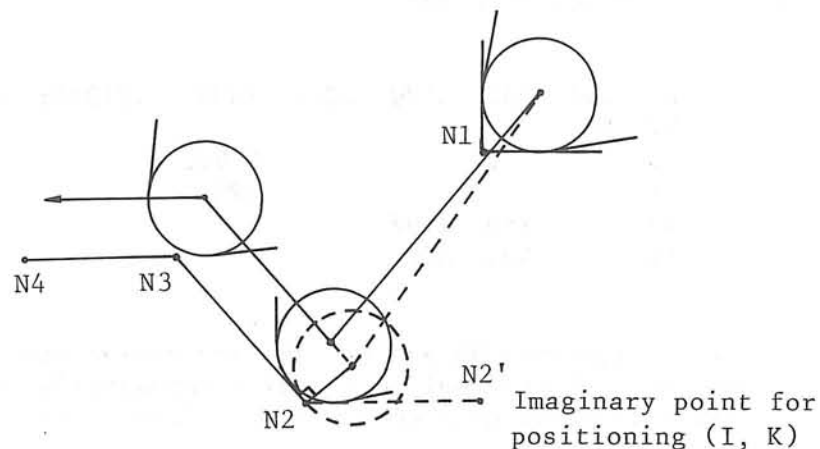
In the block containing G41 and G42, by entering I and K words that specify the imaginary point, along with X and Z words that specify the nose radius compensation start-up, unnecessary axis motion required in conventional start-up program is eliminated.

Example:

```

N1 G00 X100 Z100 F0.2 S1000 T010101 M03
N2 G42 X 60 Z 80 K20
N3 G01 X 80 Z 65
N4      Z 50
:      :

```



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.

Note: 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, K0 in the above program can be omitted.

21-3. BEHAVIOR IN TOOL NOSE RADIUS COMPENSATION MODE

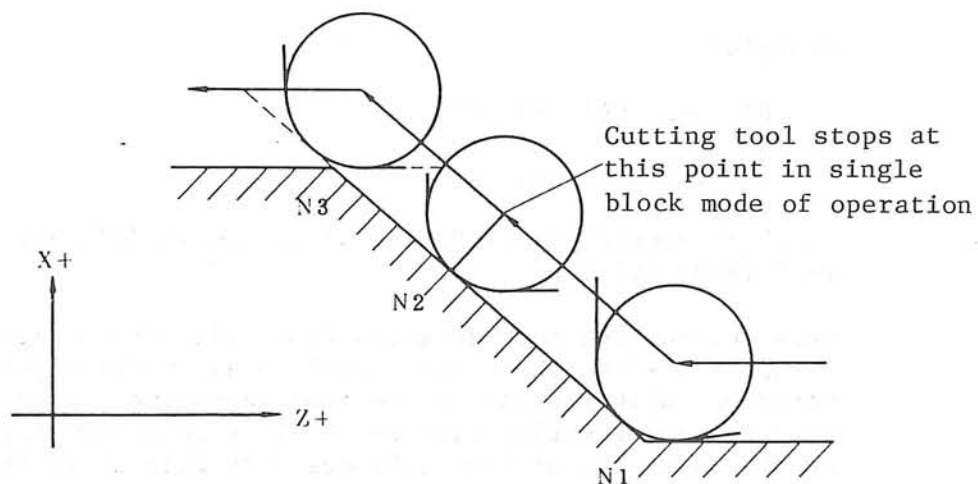
The tool nose radius compensation function provides means to automatically compensate for the tool nose radius in continuous cutting.

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

21-3-1. Straight Line to Straight Line Cutting

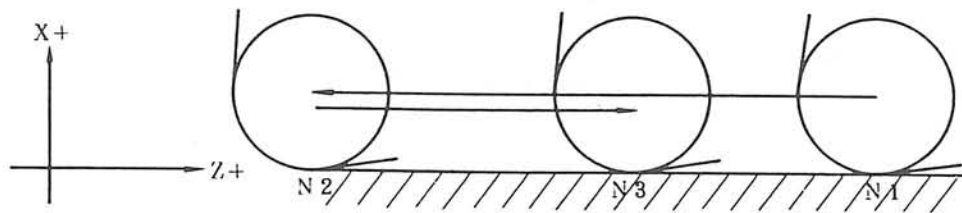
(1) Midpoint on a Straight Line

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:

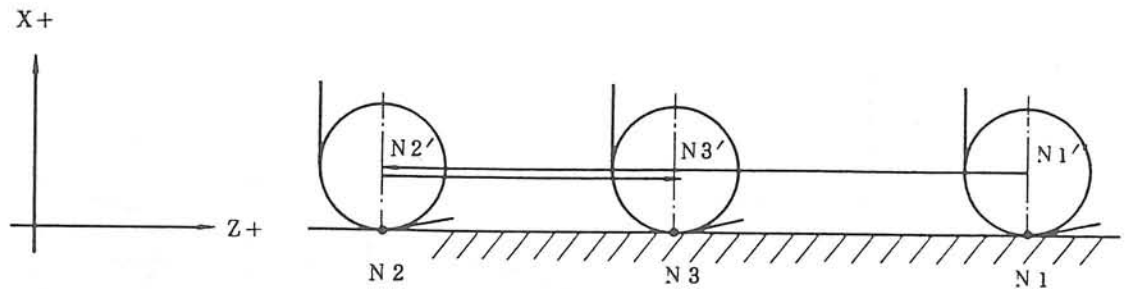
```

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.

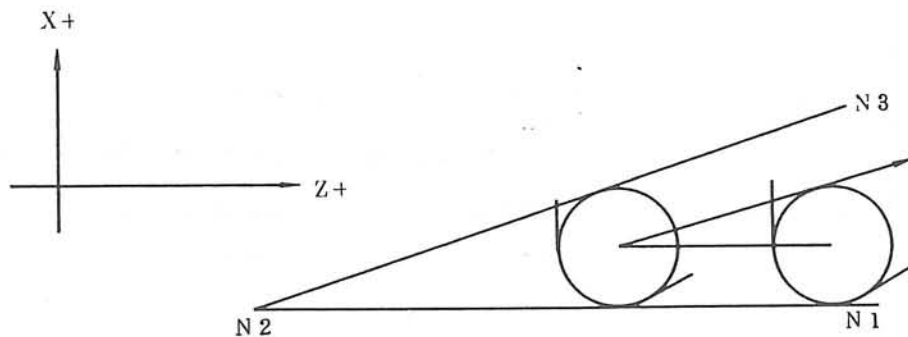
When the program above is executed, the actual path taken by the center of the tool nose R is obtained as follows.

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

Note: 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.

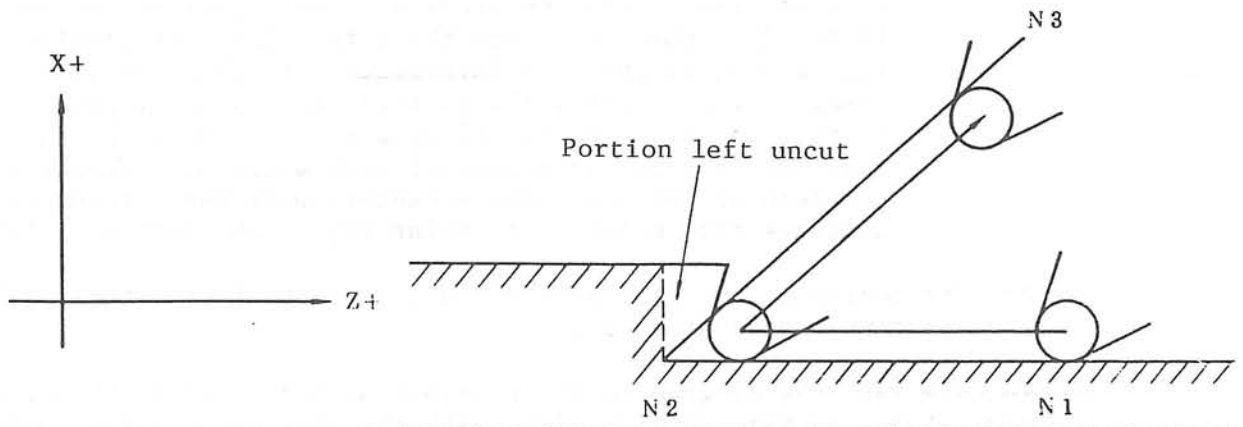
The example provided in the previous page illustrates a case where programmers are apt to be confused. Another example is provided below:

a) 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 intension, however, the cutting tool leaves a considerable amount of uncut section since it stops before reaching the desired point.

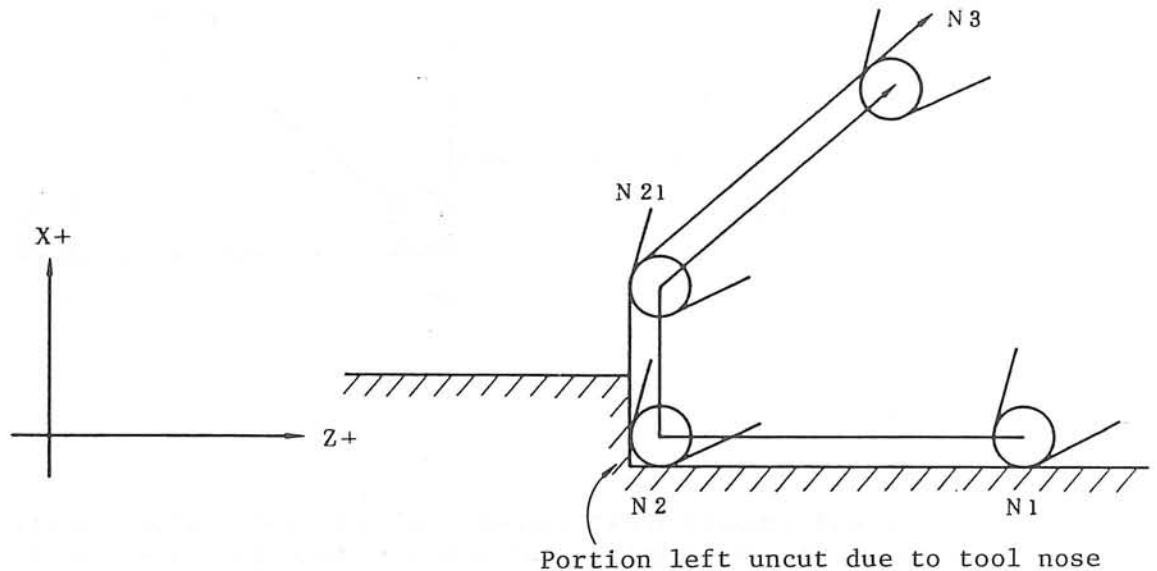
To improve such a program, enter one more point in the program as shown below:

b) Example of improved program 1:

```

N1 G42 G01 X100 Z100 F0.2 S1000 T010101 M03
N2                                     Z50
N21                                    X104 [> 100 + 4 x (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.

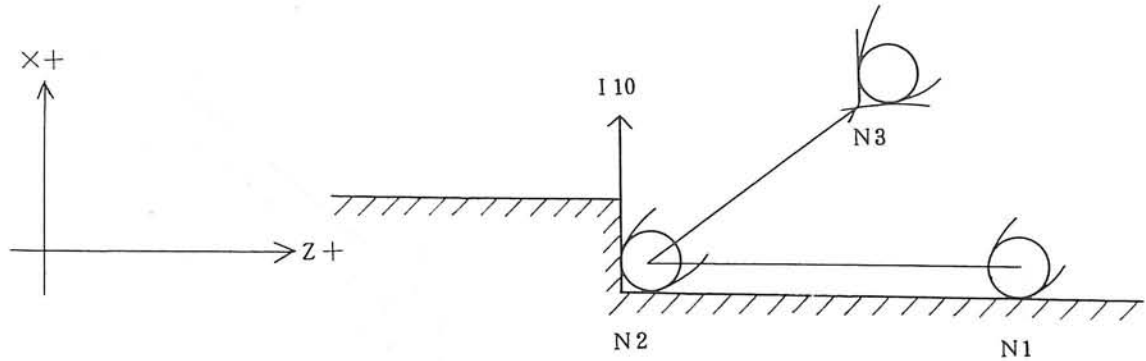
If a value smaller than the required amount is used, it might cause the cutting tool to move in the opposite direction toward point N21 and cut into N1-N2 surface.

c) Example of improved program 2:

```

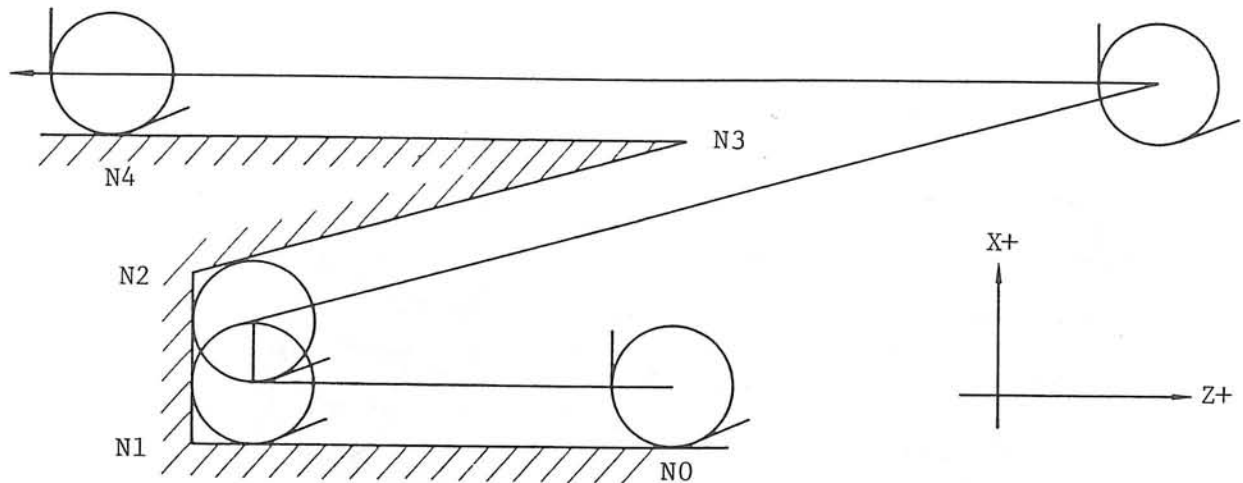
N1 G42 G01 X100 Z100 F0.2 S1000 T010101 M03
N2                               Z50
N3 G40 G00 X300 Z300 I10 M50

```



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.

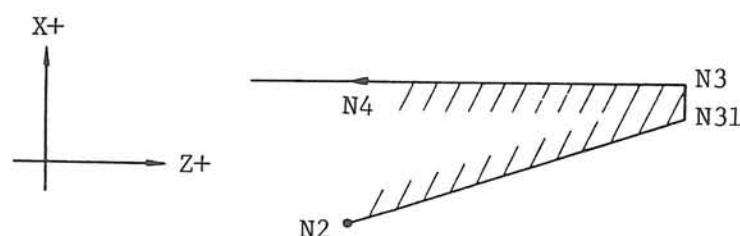
When preparing a program in which cutting similar to that contour is required, it is necessary to check the safety of the tool motion so that it does not strike against obstacles when moving to such a distant point.

Example:

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

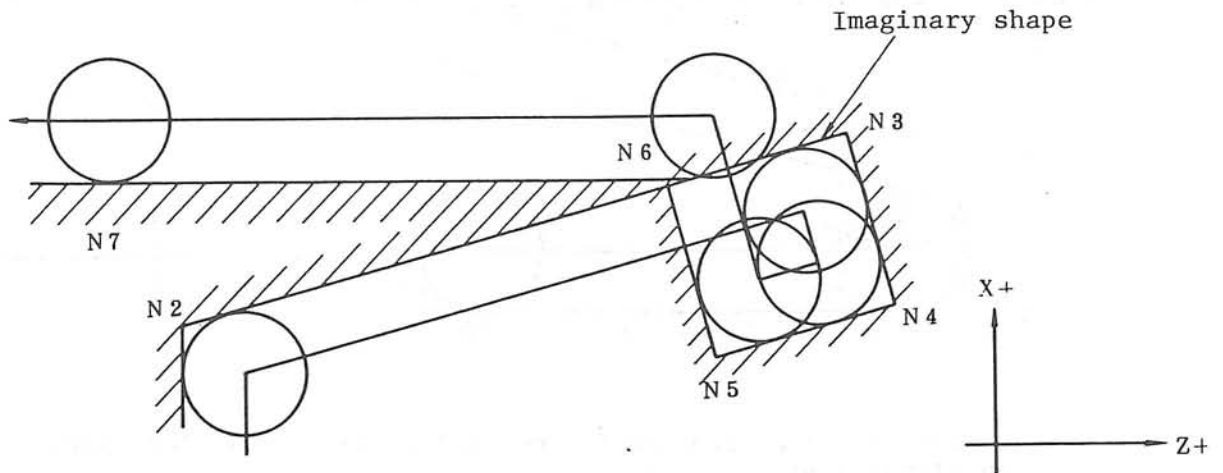
It is advantageous to improve the program and eliminate a positioning sequence to a distant point through commands in N3 block.

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.

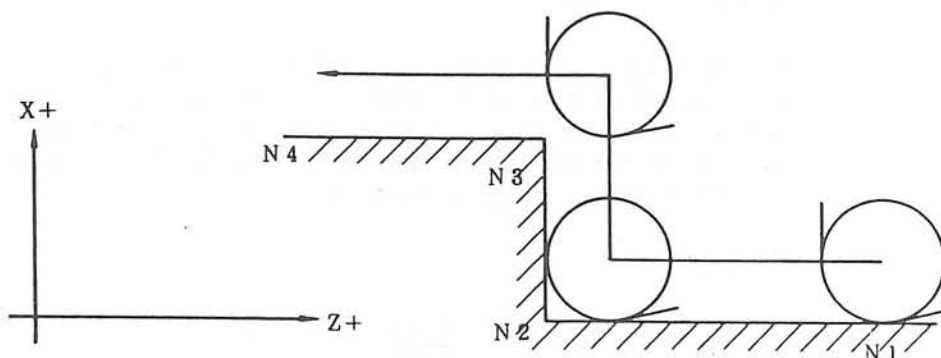
Example of Improved Program:



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.

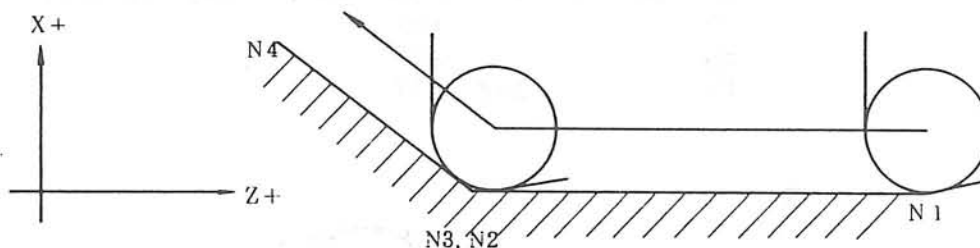
(6) Command of Identical Point

- a) 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

```



- b) 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.

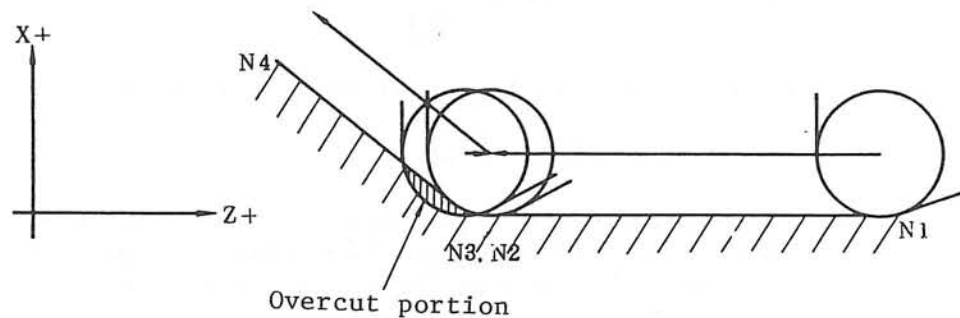
Program 1:

```

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:



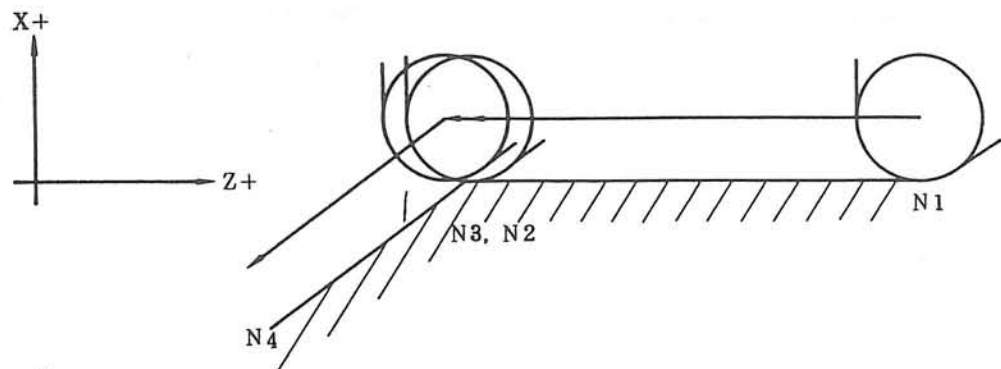
Depending on the contour to be cut, such unexpected motion does not result in overcut as seen in program 2.

Program 2:

```

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

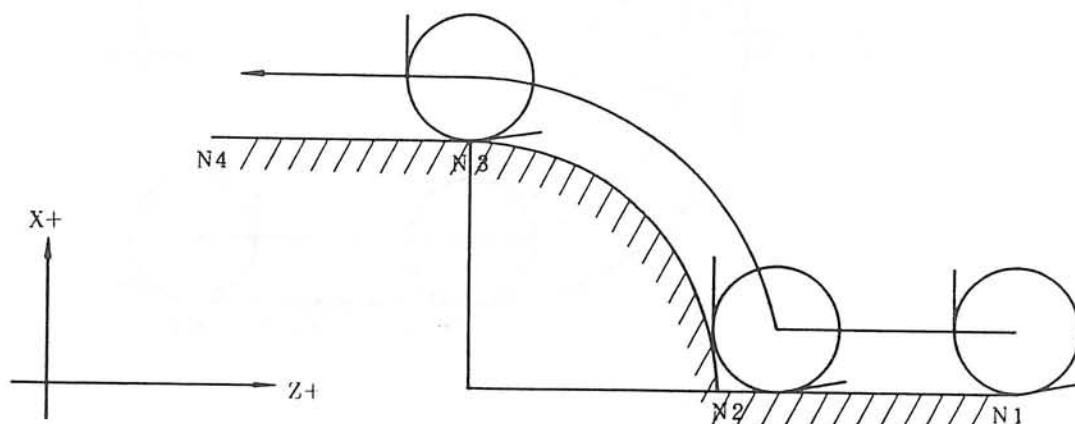
```



21-3-2. 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.



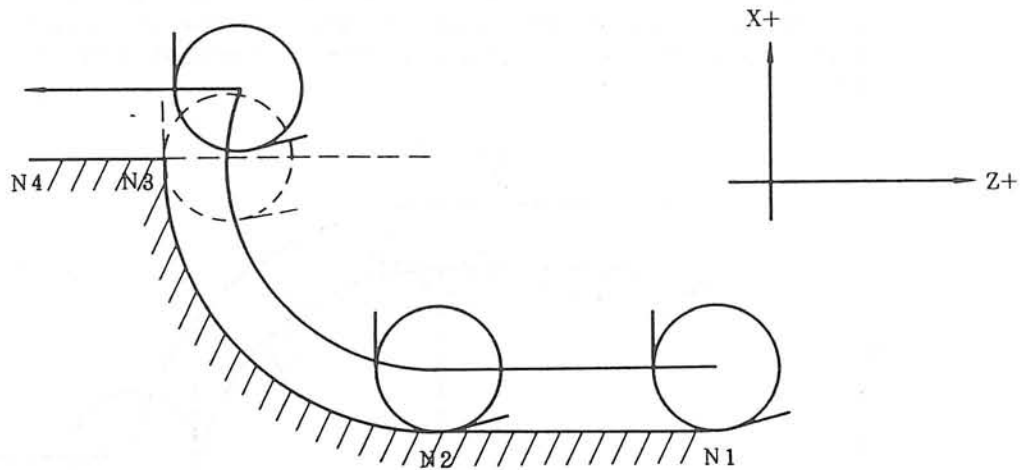
N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3		G03	X140	Z60	K-20			
N4		G01		Z40				

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

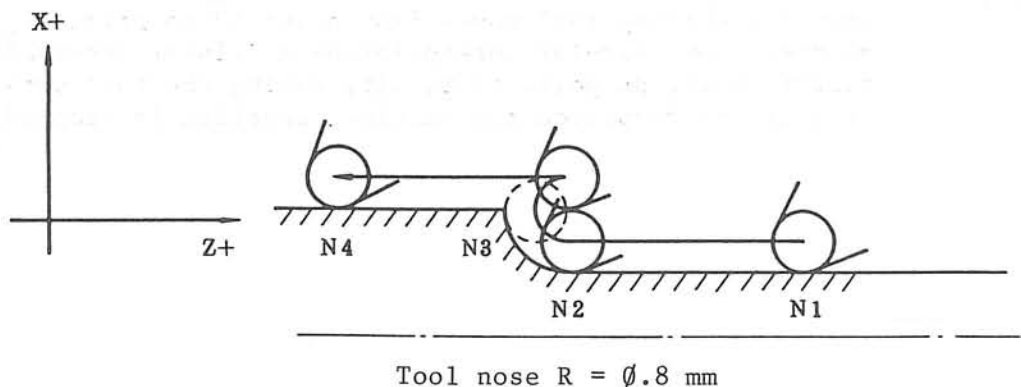
a) Case where the arc radius is greater than "2 x nose R":



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

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.

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



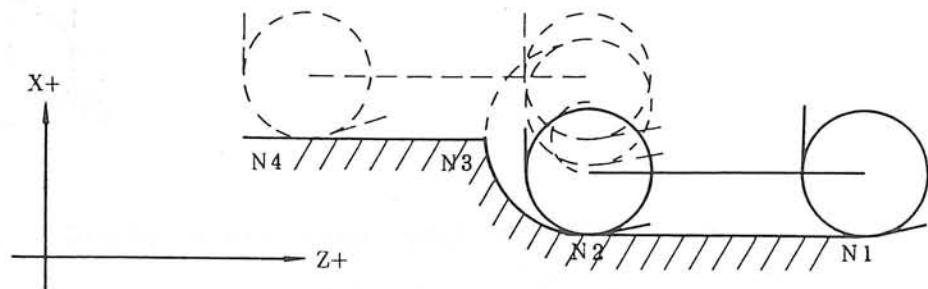
```

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.

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



```

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

```

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.

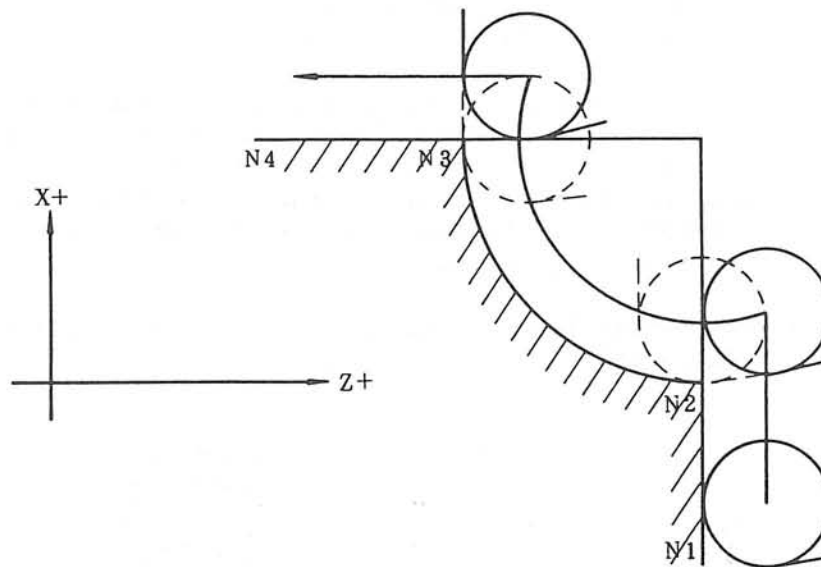
CAUTION

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

$$R \geq 2 \times RN$$

R = arc radius
RN = 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

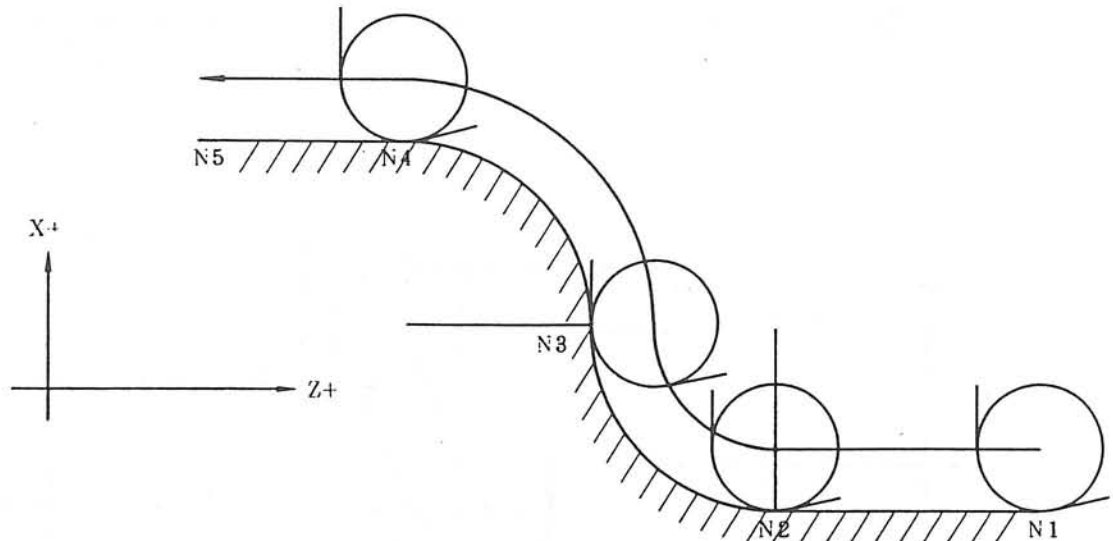
```

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).

21-3-3. 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 CRT.

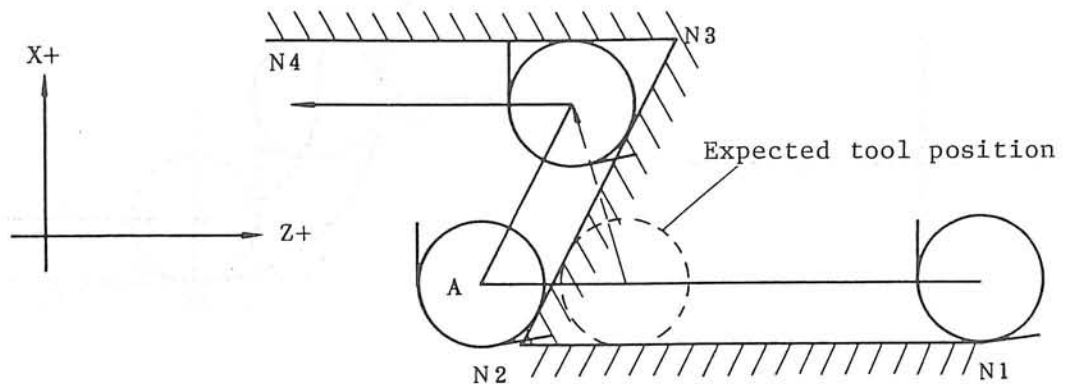
Other motions of the cutting tool are as explained in 21-3-2, "Straight Line to Arc Cutting."

21-3-4. 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	X1	Z1	T
N2		G01	X2	Z2	F
N3	G41	G00	X3	Z3	
N4			X4	Z4	

The motion of the cutting tool generated by the above program is as follows:

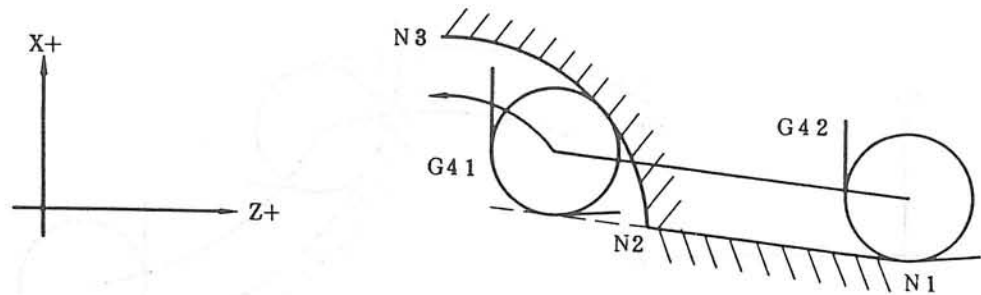
Commands in blocks, N1 and N2 are governed by G42 and those in blocks N3 and later are governed by G41. To position the cutting tool at point N2, the tool nose R center lies to the right side of straight line N1-N2 since block N2 is in the G42 mode. As for block N3, the tool nose R center lies to the 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.

Note: 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).

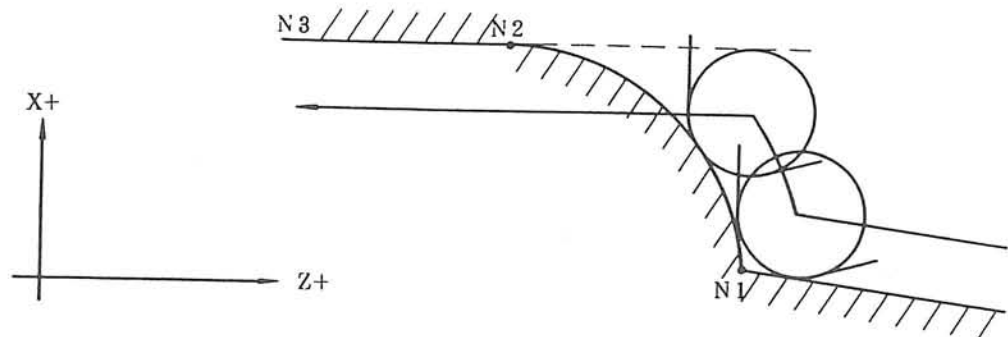
Example: N1 G42 GØ1 X1 Z1 F1 T
 N2 X2 Z2
 N3 G41 GØ3 X3 Z3 I3 K3



(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 GØ1 X1 Z1 F1 T
 N2 GØ3 X2 Z2 I2 K2
 N3 G41 GØ1 X3 Z3

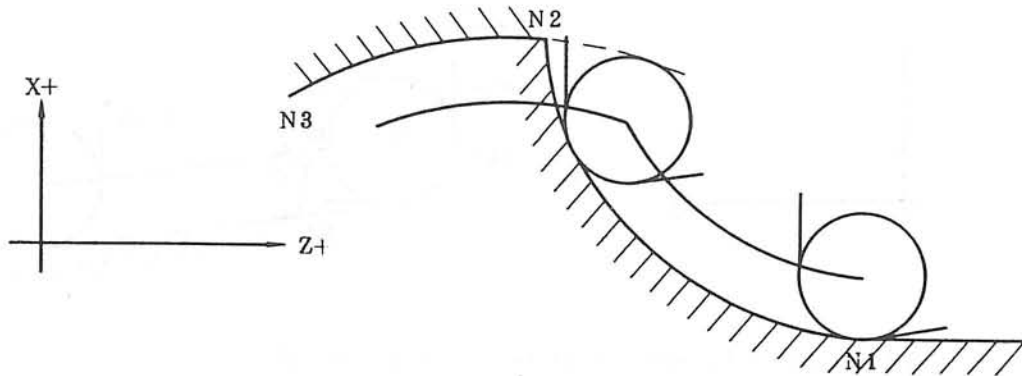


(4) Switch-Over in Arc to Arc Cutting

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

Example:

N1	G42	GØ1	X1	Z1	F1	T
N2	GØ2		X2	Z2	I2	Z2
N3	G41		X3	Z3	I3	Z3

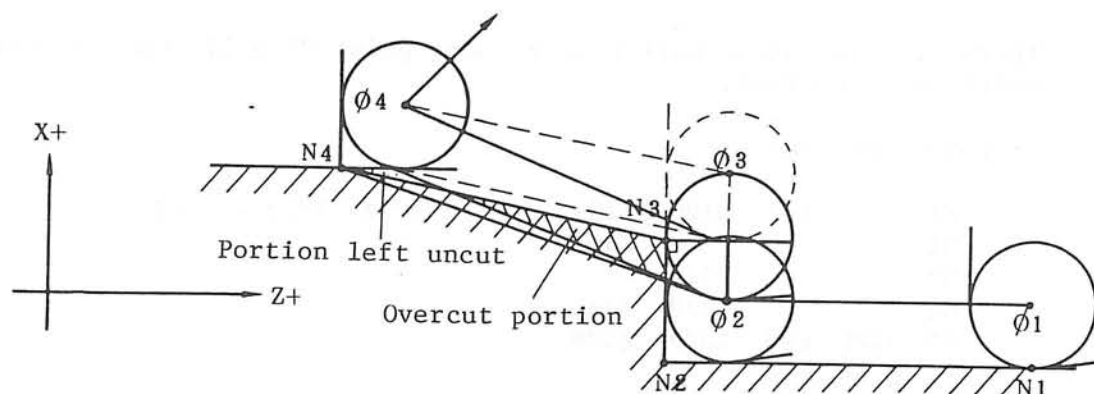


21-4. BEHAVIOR WITH TOOL NOSE RADIUS COMPENSATION MODE CANCEL

21-4-1. 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 cancelling 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 T0101 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.

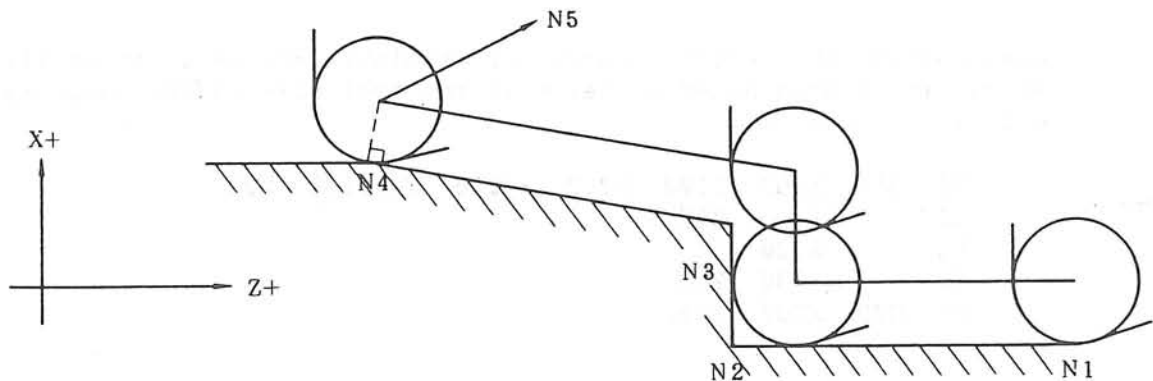
Therefore, the uncut part will be near point N4 while the section near point N3 is overcut.

Improved program:

```

N1 G42 G01 X100 Z100 F0.2 S1000 T010101 M03
N2                               Z60
N3                               X120
N4 G40 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.

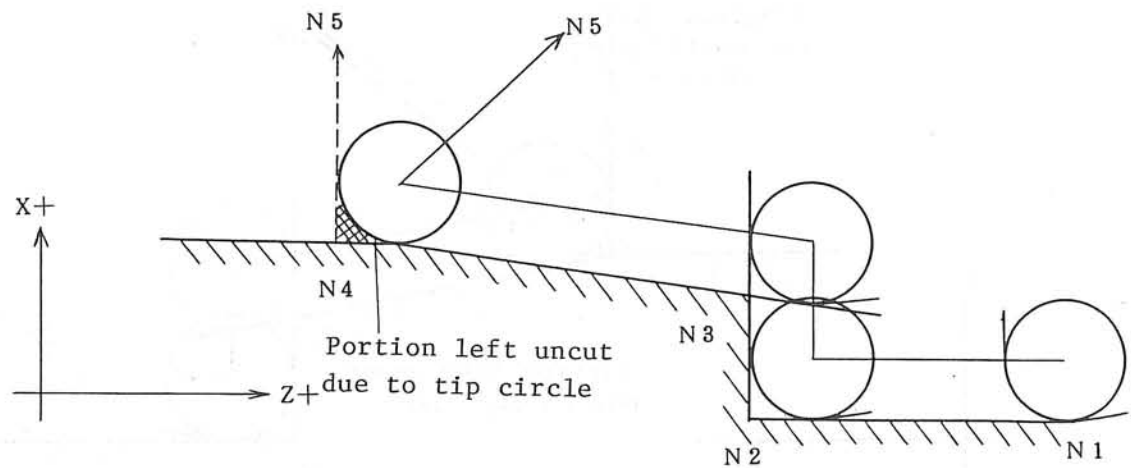
To eliminate such possible overcutting along Z-axis, see the program below.

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.

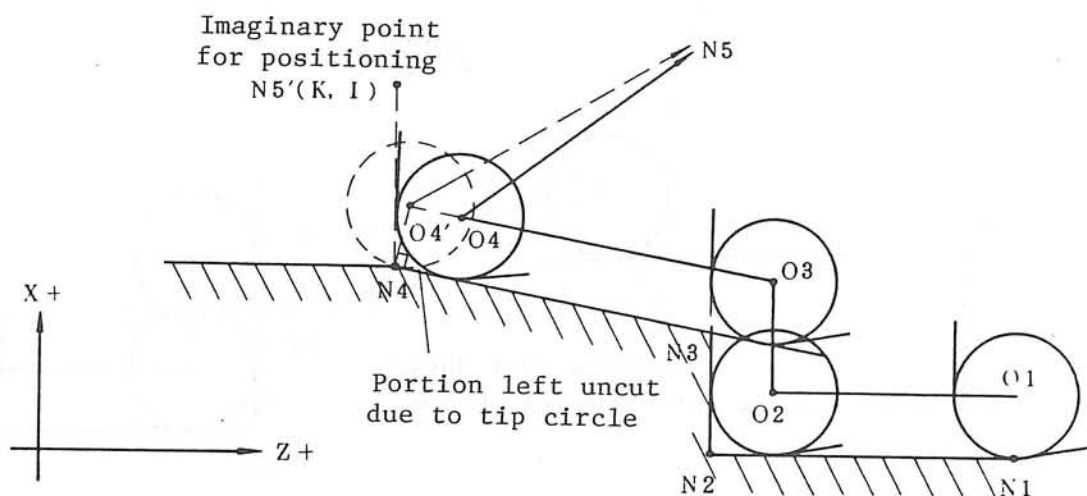
21-4-2. I and K Command with G40

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 cancelled, unnecessary axis motion required in conventional cancelling program is eliminated.

```

N1 G42 G01 X100 Z100 F0.2 S1000 T010101 M03
N2
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.

Note: 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 interpretes such word to have "0" value.

Therefore, K0 in the above program can be omitted.

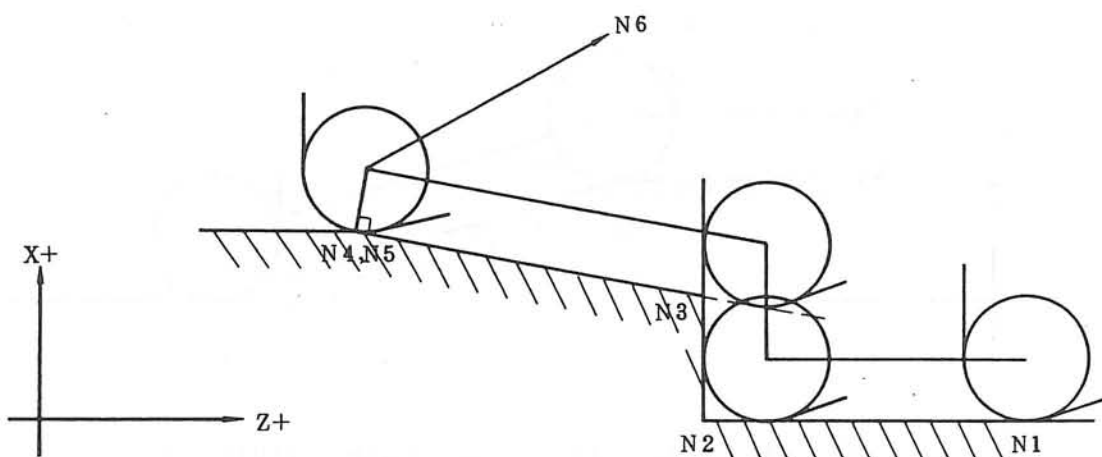
21-4-3. Independent G40

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 T0101 M03
N2
N3 X120
N4 X130 Z20
N5 G40
N6 G00 X300 Z300

```



CAUTION

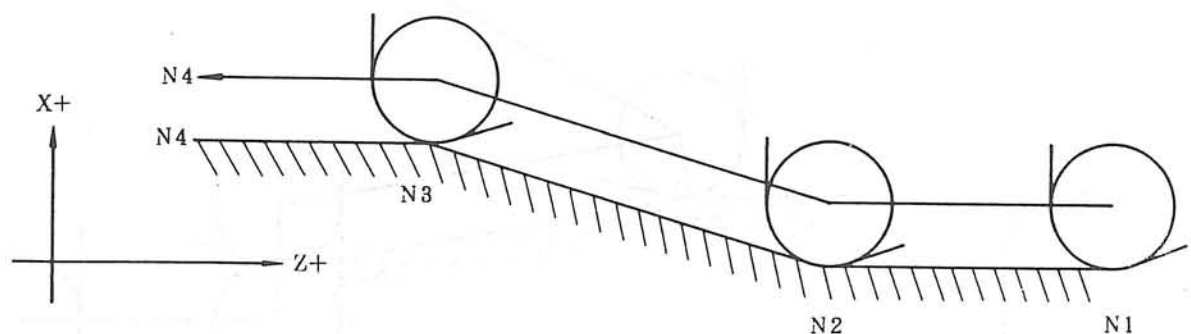
While cancelling the tool nose radius compensation mode (G40), the mode of operation must be either G00 or G01. If not, an alarm occurs.

21-5. 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 which 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.

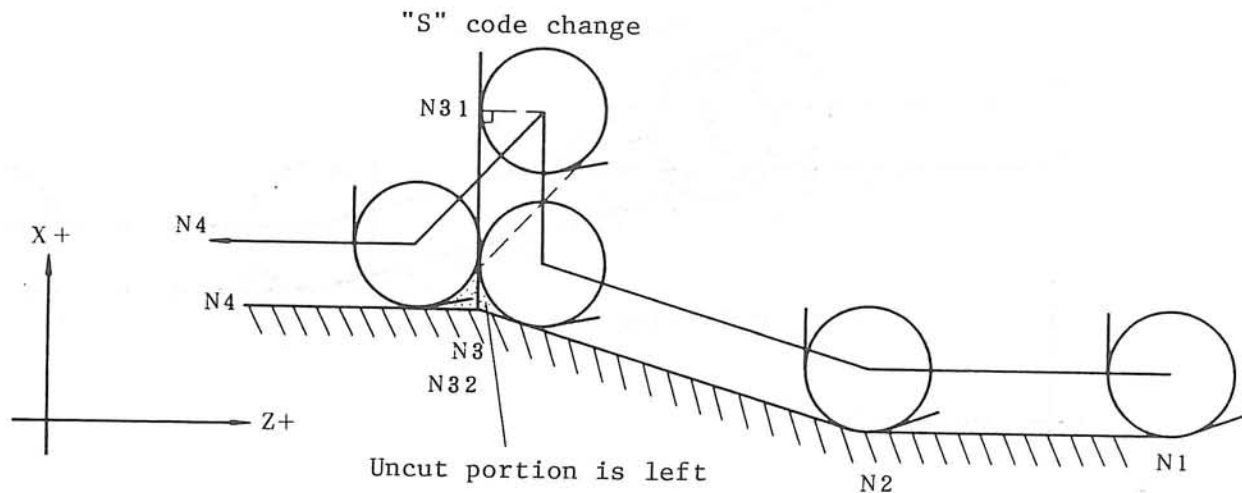
Based on this program, the program to relieve the cutting tool at point N3 in the +X direction in order to change the spindle speed is made. See the programs provided hereafter.

21-5-1. 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.

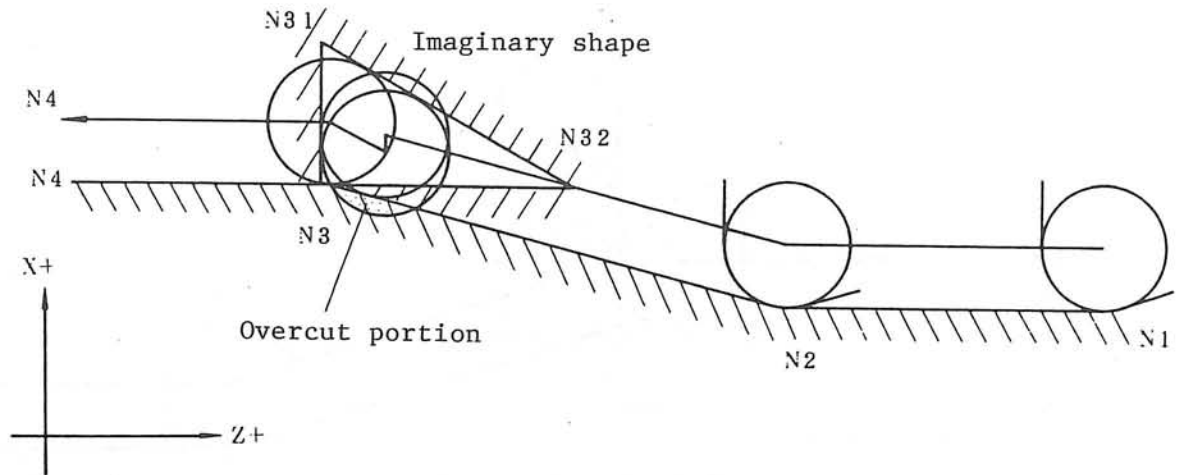
Such cutting tool movements leave an uncut portion as shown above.

21-5-2. Program 3

```

N1 G42 G01 X100 Z100 F0.2 S1500 T010101 M03
N2
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.

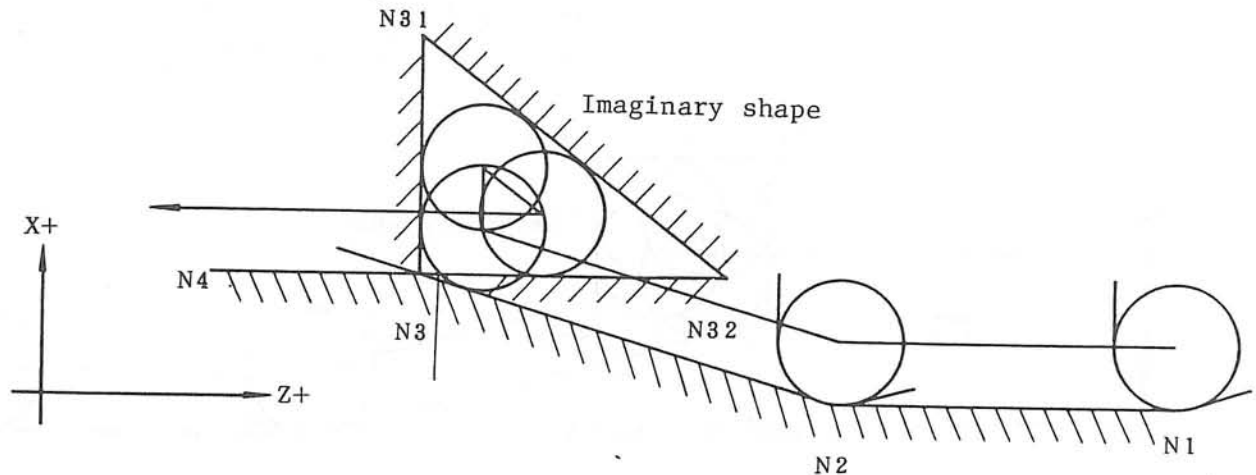
Similarly, positioning of the cutting tool in block N32 is carried out at the point where the tool nose R comes into contact with both straight lines N31-N32 and N32-N4. This also causes the cutting tool to move in the direction opposite to the programmed direction.

21-5-3. Program 4

```

N1 G42 G01 X100 Z100 F0.2 S1500 T010101 M03
N2                               Z80
N3                               X120 Z40
N31 G00 X126
N32 X120 Z43 S10000
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.

21-5-4. Program 5

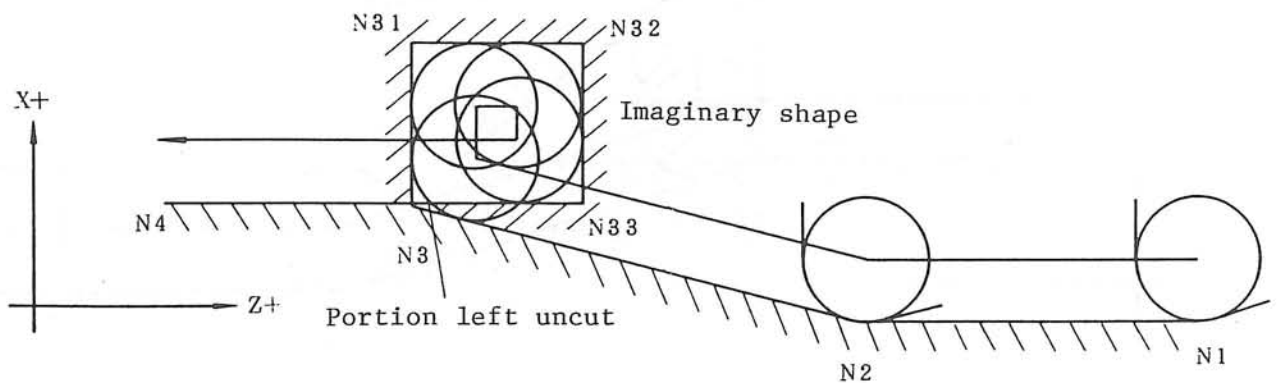
```

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: > $120 + 4 \times (\text{nose R})$

*b: > $40 + 2 \times (\text{nose R})$



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.

21-5-5. Program 6

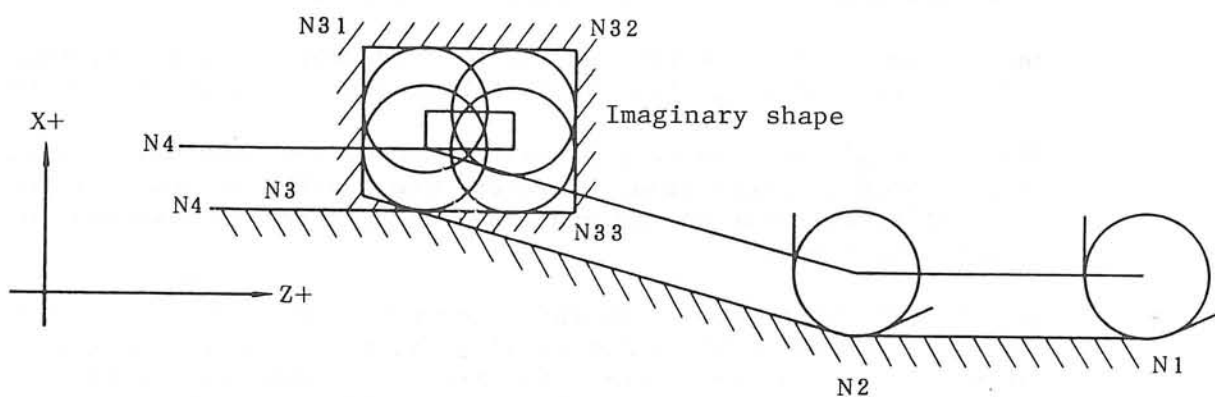
```

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: $> 120.5 + 4 \times (\text{nose R})$

*b: $> 39 + 2 \times (\text{nose R})$



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.

21-6. PRECAUTIONS

- (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 reads 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.

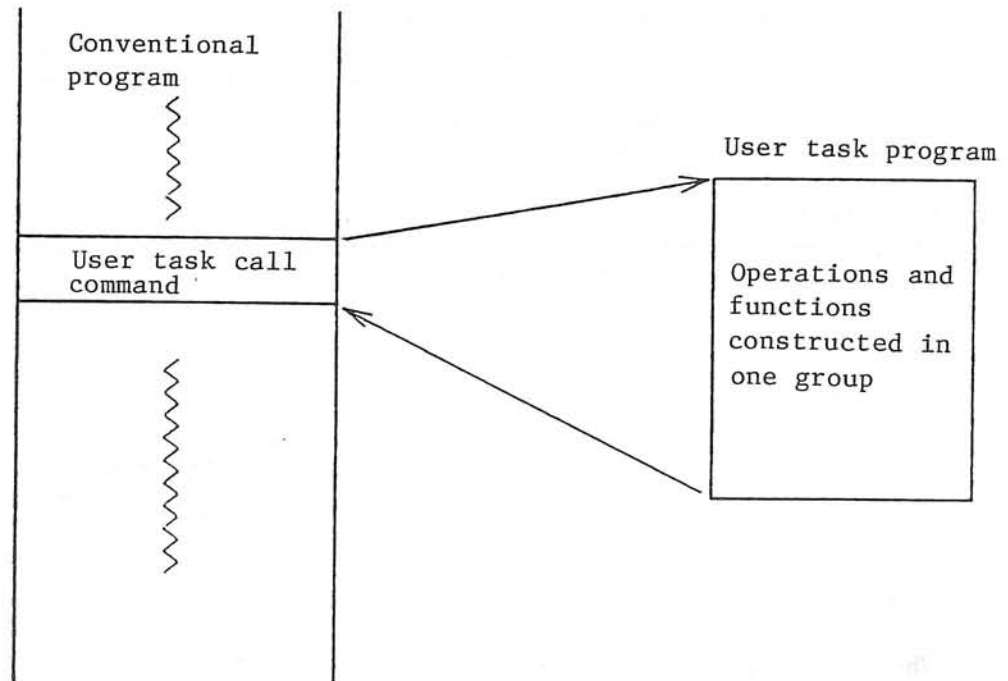
SECTION 22 USER TASK 2

22-1. OUTLINE OF USER TASK FUNCTION

After years of work in data processing, we at Okuma are now able to provide our customer with the most efficient use of high data processing capacity in OSP5000L-G/OSP50000L-G series. We have accomplished this in the development of User Task Function.

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 all available from Okuma. However, it is also 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.

22-2. TYPES OF USER TASK FUNCTION

22-2-1. Relation between Types of Program Files and User Task Functions

Types of program files are discussed in the Operation Manual of OSP5000L-G, Section 4, "Application" (Publication No. 2446-E). Shown below summarizes the relation between the types of program files and user task functions.

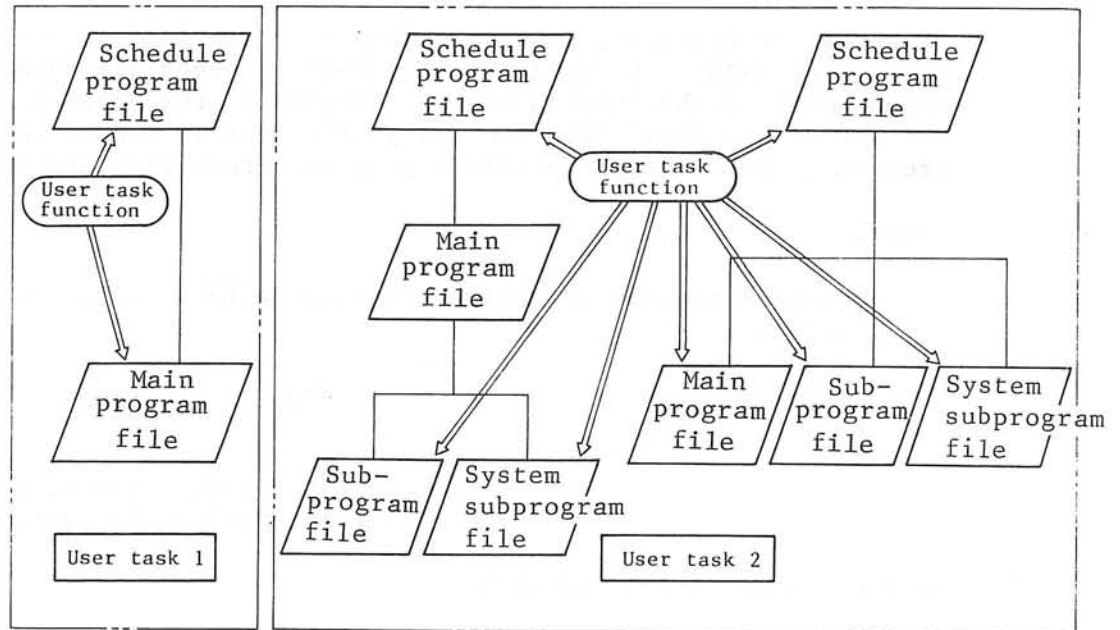


Fig. 22-1

Fig. 22-2

Programs in Fig. 22-1 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". User task 1 is the standard function.

The case shown in Fig. 22-2 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".

This manual covers "User Task 2" and functions available with "User Task 1" are all contained in "User Task 2".

22-3. 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 X, Y, and 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:

X 1 3 5

When a variable is used:

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:

X 1 3 5

When arithmetic operation
function is used:

X = 100 + XP2 XP2 = 35 or
XP2 = 35 X = 100 + XP2

Provided in later pages are the detailed descriptions of respective fundamental user task functions.

22-3-1. Control Statement

User Task 1 can use following six control statements:

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. 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
      |
      |
NLAP1L 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.

Note 1: Sequence Name

A sequence name is the code to identify respective blocks in a program, and it consists of four alphanumeric characters following address character N.

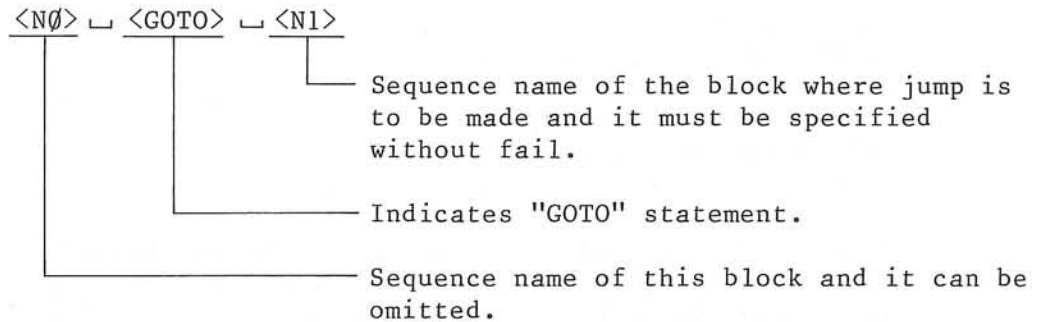
There are two types of sequence names:

<N> <four alphanumeric>, and
 <N> <letter> <three alphanumeric>

"Sequence name" handled in this manual refers to both types of sequence names.

(1) GOTO Statement - Unconditional Jump

a) Program Format



Note: Sequence name $\langle N1 \rangle$ for jump must be the one in the program containing such control statement.

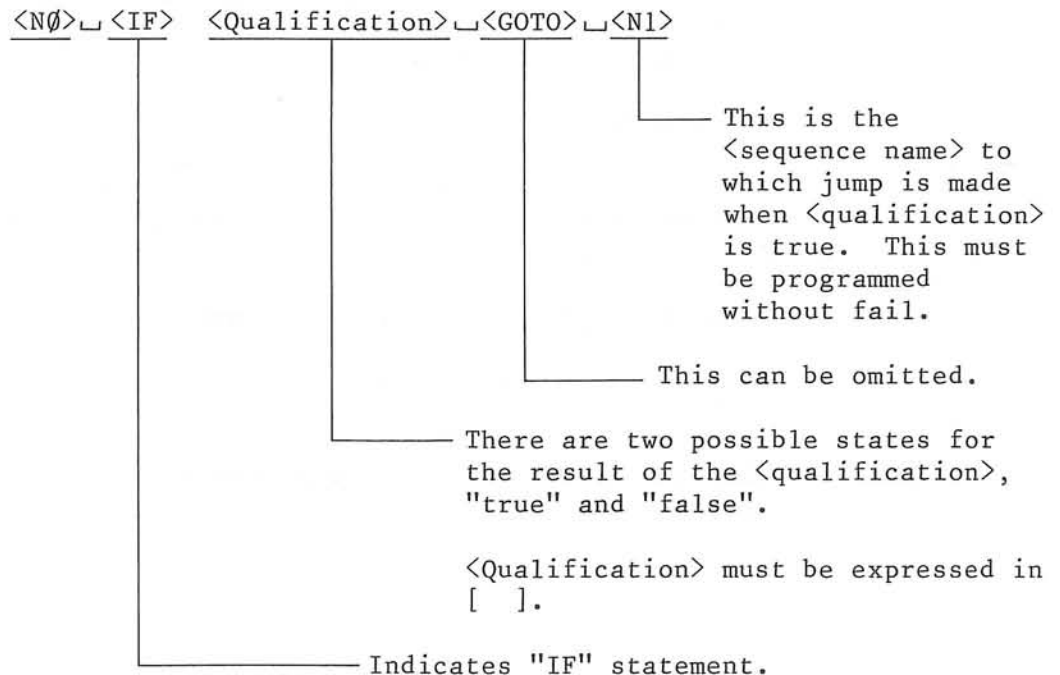
b) Function

When the control statement is executed, jump to the programmed block, $\langle N1 \rangle$ is made unconditionally.

Note: On two-saddle models, jump from the program for A saddle to the program for B saddle or vice versa is not allowed.

(2) IF Statement - Conditional Jump

a) Program Format



b) Function

- 1) When the <qualification> is true, execution of the sequence jumps to the sequence <N1>.
- 2) When the <qualification> is false, the following sequence is executed.

Example:

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.

c) Jump by Judgement of Definition of Local Variable using IF statement

<N2> <IF> <Local variables name> <GOTO> <N3>

This is the <sequence name> to which jump is made when the designated variable has been defined.

This can be omitted.

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.

Indicates IF statement.

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.

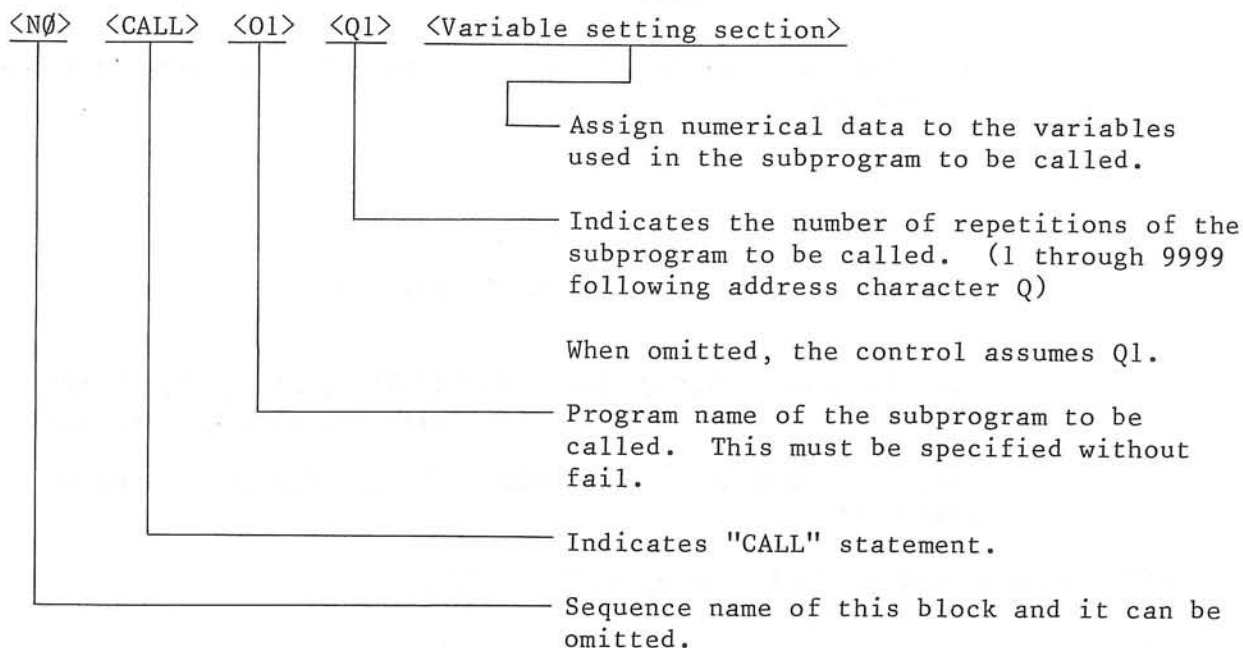
Example:

N1000 IF ABC N2000

If local variable AC has been defined, jump to N2000 sequence is made. If it has not been defined, the next block is executed.

(3) Call Statement - Calling Program

a) Program format



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.

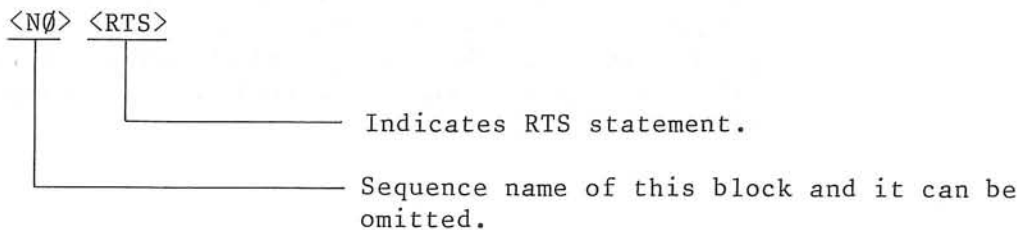
Example:

N1ØØØ CALL O1234 XP1=15Ø ZP1=1ØØ

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



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.

Example:

Main Program

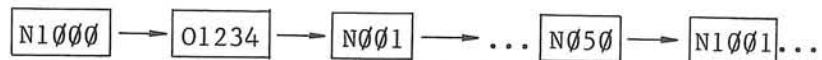
```
N1000 CALL 01234 XP1 = 150 ZP1 = 10
:
:
```

Subprogram

```
01234
N001 G00 X      Z
:
:
:
N050 RTS
```

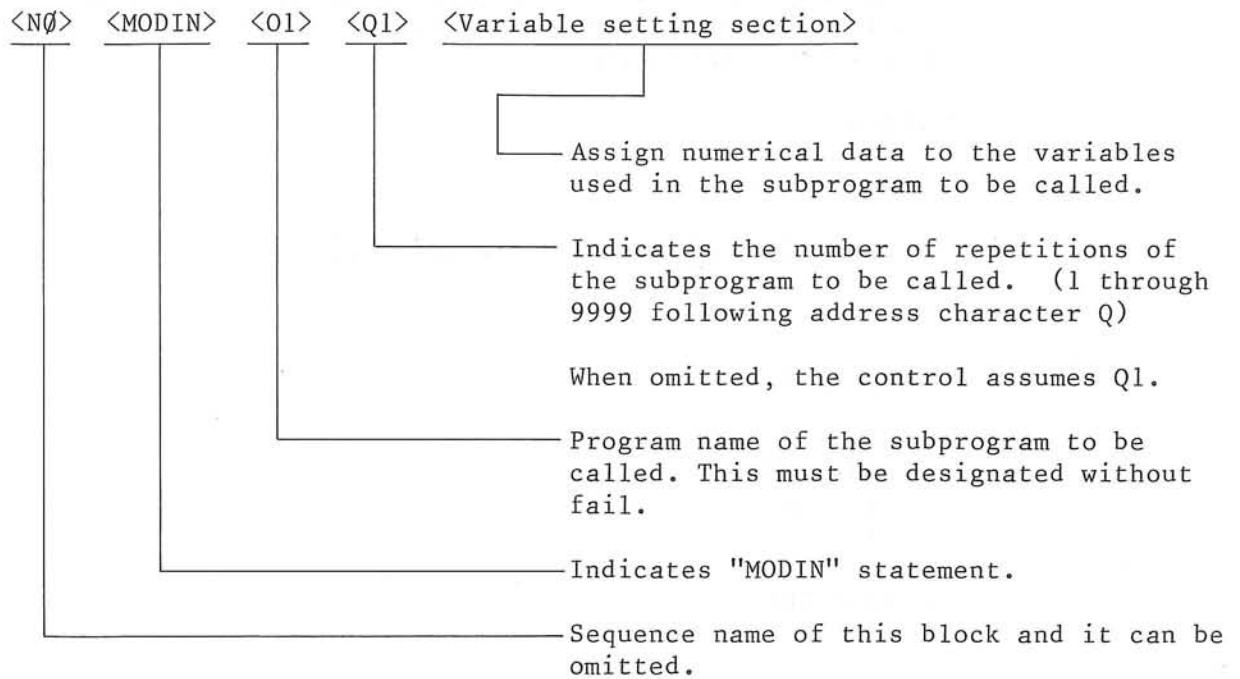
When block N1000 in the main program is executed, jump to subprogram 01234 is made. The subprogram is executed from N001 and when the control reads the RTS statement in N050, jump is then made to N1001 of the main program and the commands in that block and those after that are executed. At the instance the jump from the subprogram to the main program is made, variables XP1 and ZP1 are cleared.

Order of program execution:



(5) MODIN Statement

a) Program Format



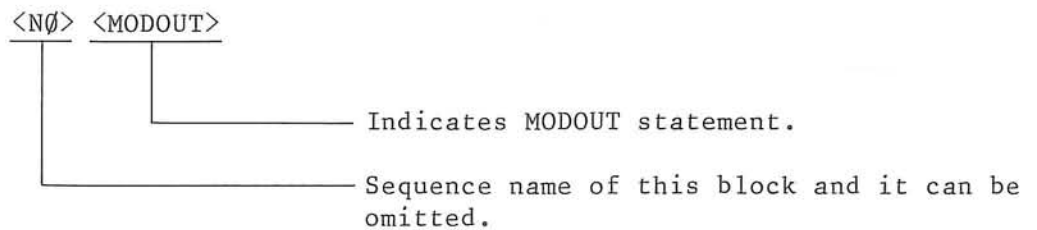
b) Function

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 GØØ, GØ1, GØ3 and G31 through G35 modes which contain either or both of X and Z words.

(6) MODOUT Statement

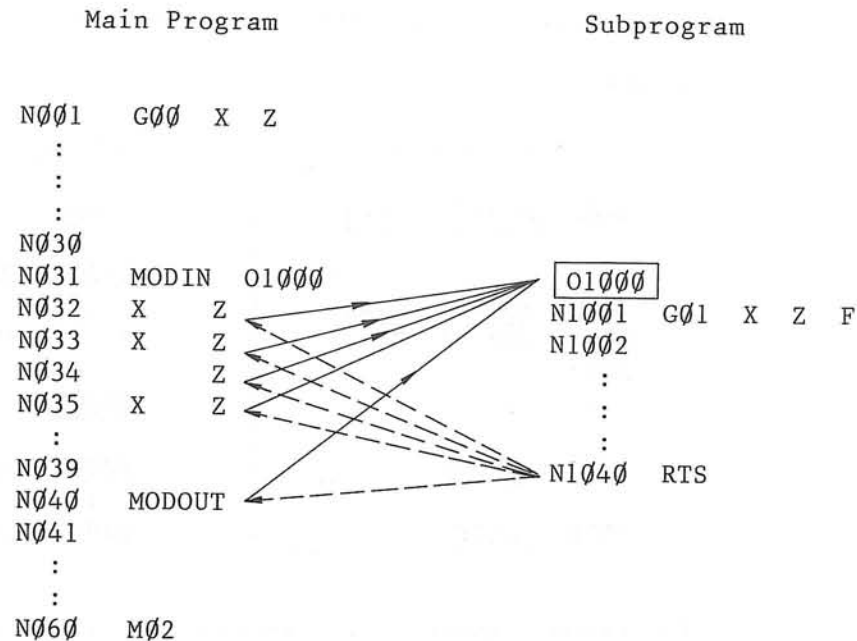
a) Program Format



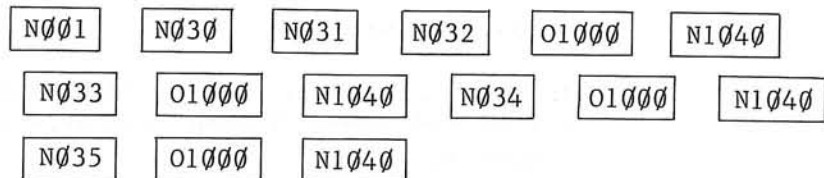
b) Function

This is the statement to cancel the MODIN mode.

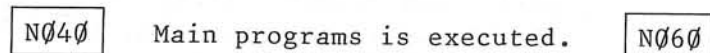
c) Program Example of MODIN and MODOUT Statements



Order of program execution:



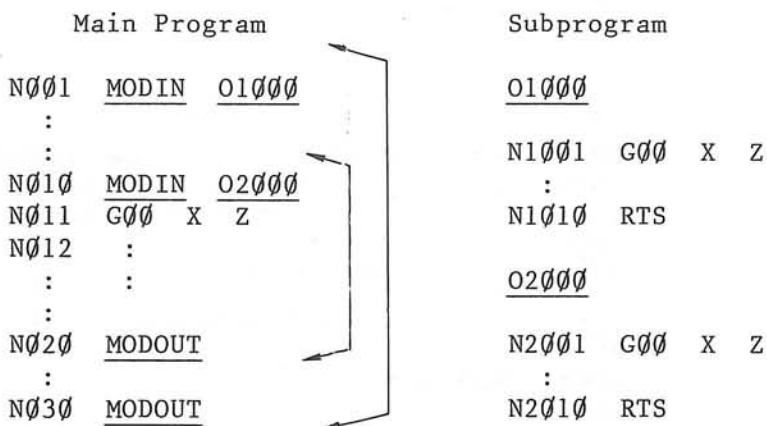
Main and subprograms are executed repeatedly.



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.

Note 1: Nesting and effective range of MODIN/MODOUT mode
Permissible nesting level in MODIN mode is eight.

Example



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.

Note 2: The MODIN mode must be cancelled by the MODOUT statement designated in the same program. That is, the MODIN mode activated in the main program cannot be cancelled by the MODOUT statement in the subprogram, or MODIN mode active in the subprogram cannot be cancelled by the MODOUT statement in the main program.

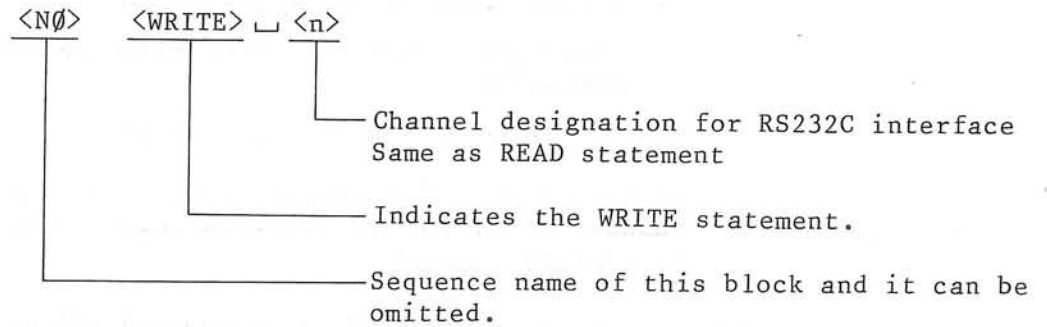
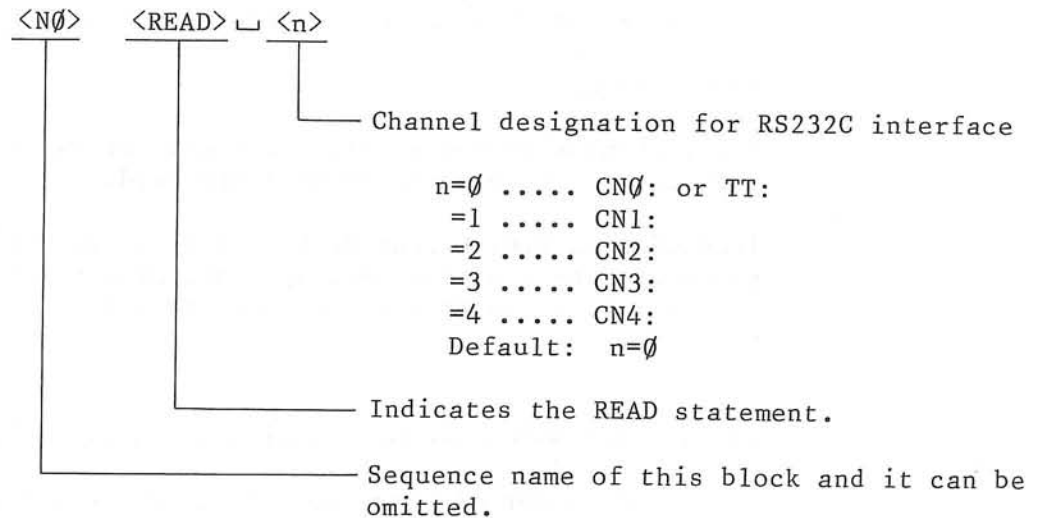
Note 3: Maximum number of subprograms usable in a single program is 125.

(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



b) Function

READ Receives the data from the external device connected to the channel designated by $\langle n \rangle$ and stores it in read area (max. of 16∅ characters) in the JIS8 code.

WRITE Sends the data stored in the write area (max. of 16∅ characters) in the JIS8 code to the external device connected to the channel designated by $\langle n \rangle$.

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.

For setting, No. 39 through No. 43 of the optional parameters (baud rate CN0: to CN4:) and No. 12 through No. 16 of the optional parameter bits (signal construction CN0: to CN4:) are used. For details, refer to the operation explained in 4-3, "Parameter Setting" of the Operation Manual.

d) Transmission code

For transmission codes, JIS 8 bit code or JIS 7 bit code (in this case, use even parity bit) are used.

Transmission finish code NULL or % is used. This selection is referenced to 3 of the optional parameter bit No. 1. (% 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) READ/WRITE areas are common for A and B saddles. 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 should 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

<NØ> <GET> └ <s>, <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 1Ø digits)

If "s" equals "Ø", this indicates the number of characters to be skipped.
(Max. of 16Ø characters)

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)

Indicates GET statement.

Sequence name of this block and it can be omitted.

<NØ> <PUT> └ <s>, <l>

This indicates the number of digits to be written to the write area using the WRITE statement. (Max. of 1Ø digits)

If "s" equals "Ø", this indicates the number of space codes.
(Max. of 16Ø space codes)

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", \$ØA)

Indicates PUT statement.

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.

As mentioned before, since the READ area is common for both A and B saddles, when it is necessary to read the mixed data of A and B saddles, synchronization by M100 is required.

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 '

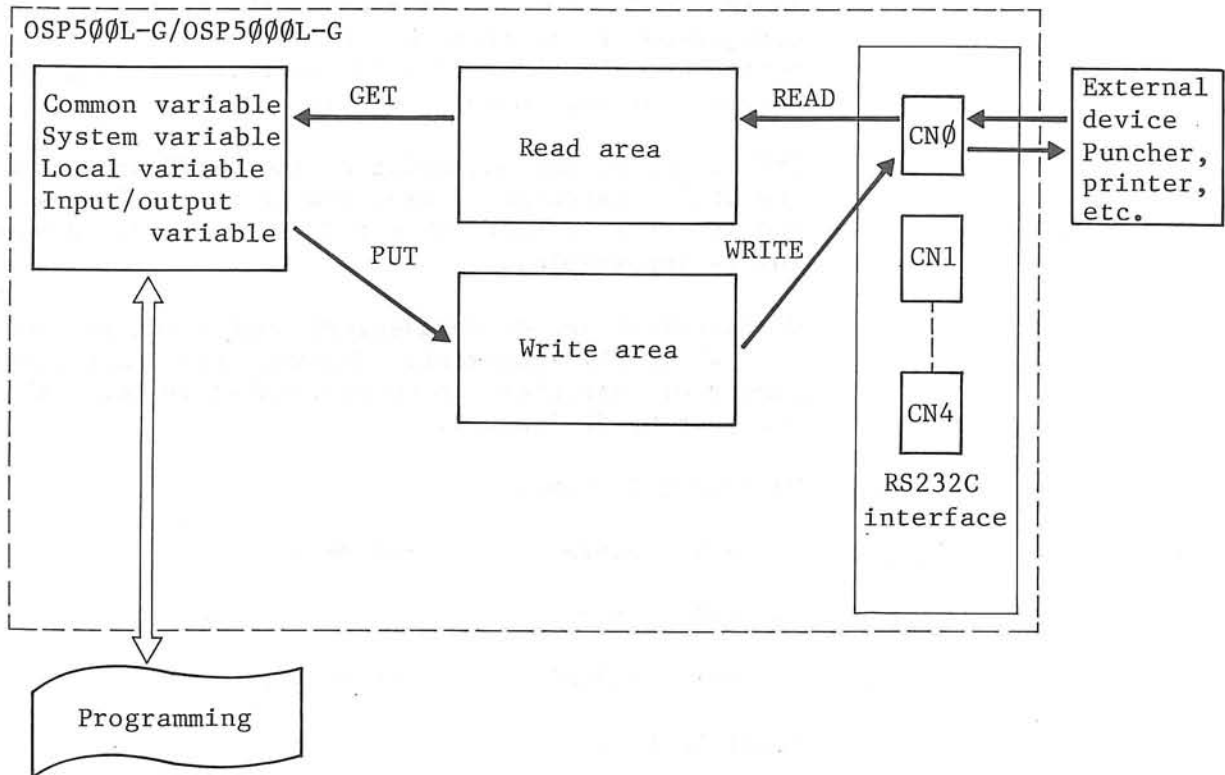
Since the WRITE area is common for both A and B saddles, when it is necessary to transmit the mixed data of A and B saddles, synchronization by M100 is required.

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. = <input type="checkbox"/> 3 <input type="checkbox"/> OX = <input type="checkbox"/> 0.02 <input type="checkbox"/> 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'
N103  PUT  '  **'
N104  PUT  $D0A                      ..... 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  VTOFZ[V1],8
N111  PUT  $D0A                      ..... 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
    
```

22-3-2. 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 are 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 variable, the result of calculation in one program, can be referred to in other programs.

a) Program format

<V> <one or two digits> = numerical data or expression

Designation of a common variable is made with up to two digits following "V". Usable common variable is V1 through V32.

In a program, a common variable is used as:

$$N1\emptyset LV5 = 1\emptyset, \text{ or } N1\emptyset 1 LV5 = V5 + 1$$

Note 1: Common variables are used in common to A-turret and B-turret.

Note 2: Common variables are effective both in main and subprogram.

Note 3: 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 program tape is loaded.

Note 4: 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 Operation Manual for OSP5000L-G/OSP5000L-G, 4-3, "Operation in Parameter Mode" (Publication No. 2446-E).

(2) Local Variables

As may be known from 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 for A-turret and B-turret, respectively.

a) Program Format

<Letter><Letter><two alphanumerics> = Numerical data or expression

↑
"0", "N" and "V" cannot be used.

Example: 'DIA1' 'ITH5'

The name identical to function names (*2), relative operators (*2), logical operators and extended address characters (*2) cannot be used.

*1: Extended address characters are used for LAP, pattern processing, user fixed cycle and other extended functions which cannot be covered with conventional address characters A through Z. Extended address characters are:

<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>, <AB>, <BC>,

*2: Explained in detail in 22-3-3.

b) Characteristics of Local Variables

- 1) Local variables can be used independently on A-turret and B-turret, and 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.

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

```

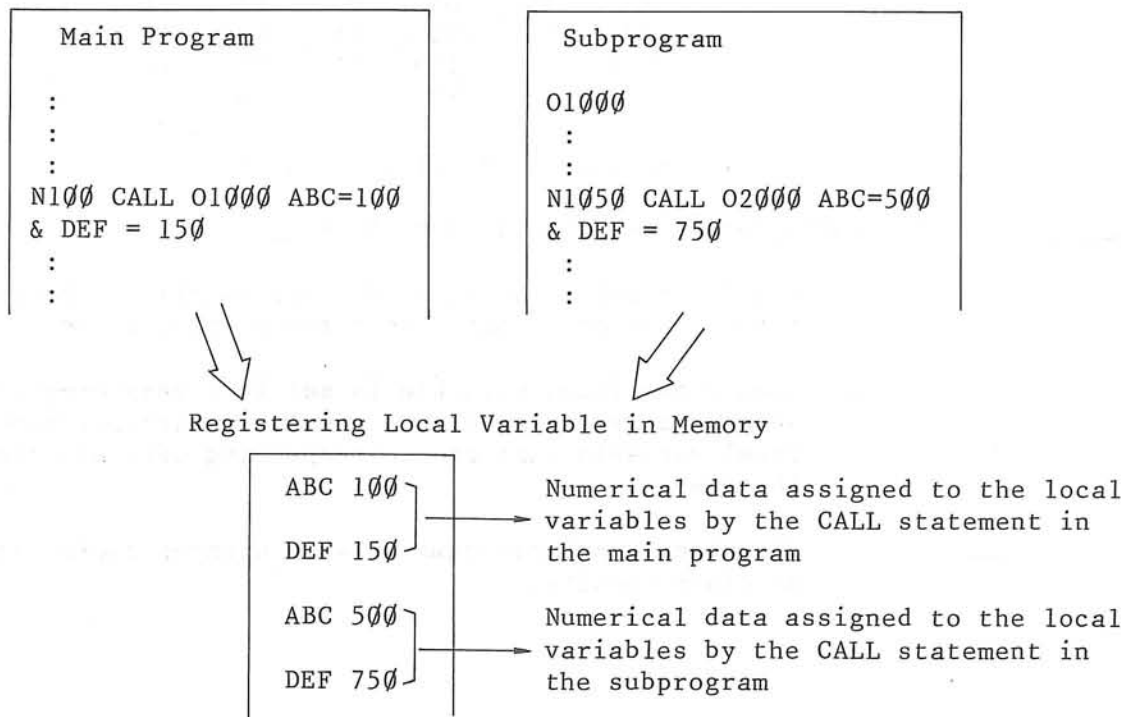
:
:
N0010 DIA1 = 160
:
N0049
N0050 DIA1 = 200

```

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 on A-turret and B-turret, respectively.
- 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.

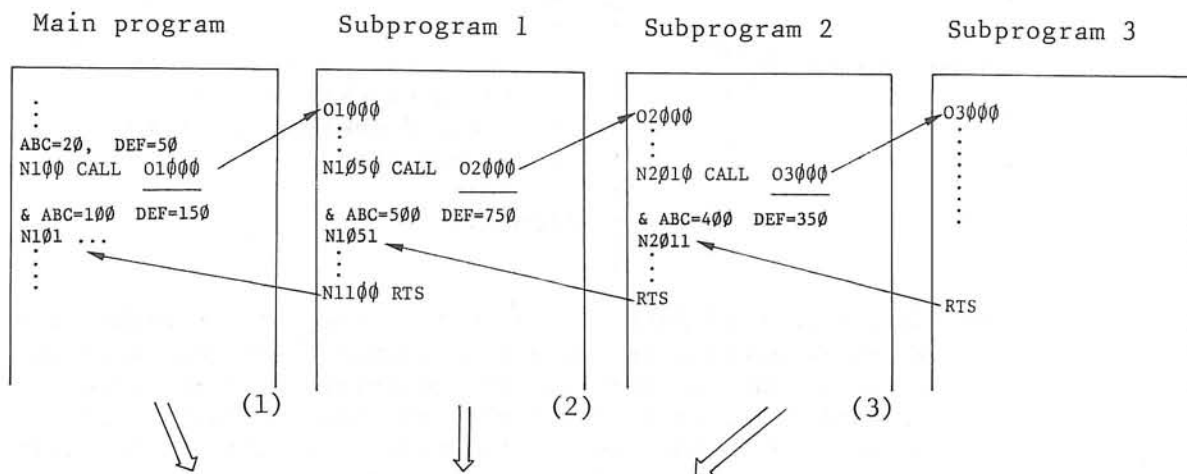
Even when the local variable having the same name as the one already registered before the call statement is programmed is designated, it is newly registered as a new variable.



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 Cleared in (1)
	DEF	150	
Subprogram 2	ABC	500 Cleared in (2)
	DEF	750	
Subprogram 3	ABC	400 Cleared in (3)
	DEF	350	

In the example above, execution of N2010 in the subprogram 2 registers 4 kinds of local variables ABCs and DEFs and the subprogram 03000 is executed. If the subprogram 03000 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 03000, that is, when the RTS statement in 03000 is executed, the local variables registered by the subprogram CALL 03000, ABC=400 and DEF=350 are cleared from the memory.

(3) System Variables

System variable means the variable specific to respective systems and its name is fixed.

The system variables are set for respective turrets independently and 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 the Operation Manual for OSP500L-G/OSP500L-G.

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
 <VZOFW> Zero OFFset of W-axis
 (only for programmable tailstock specification)
 <VZOFC> Zero OFFset of C-axis
 (only 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
 (only 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

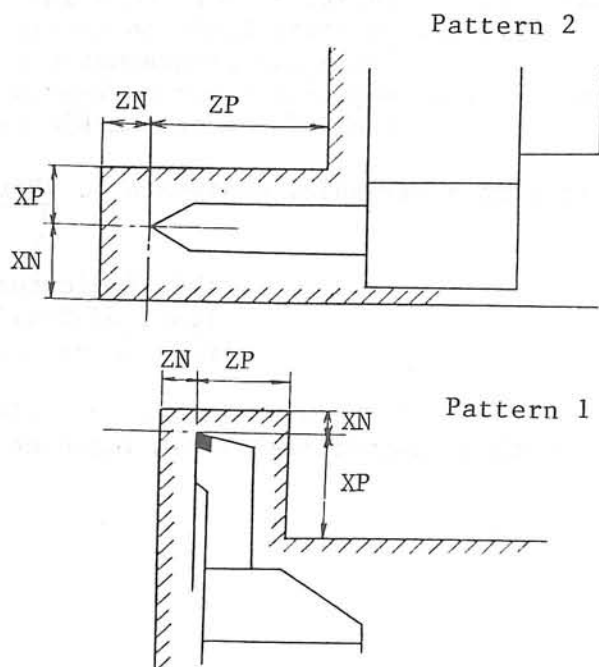
<VNSRX> [Nose R Compensation No.]

NoSe Radius compensation of X-axis

When setting a variable, designate as VNSRZ [4] = \emptyset .8

This indicates that nose radius (in Z-axis) of the tool assigned with nose radius compensation no. 4 is set to \emptyset .8 mm.

e) Tool interference data



Variable name

<VTIZN> [Tool No.] Tool Interference on Z-axis at Negative
 end; ZN
 <VTIZP> [Tool No.] Tool Interference on Z-axis at Positive
 end; ZP
 <VTIXN> [Tool No.] Tool Interference on X-axis at Negative
 end; XN
 <VTIXP> [Tool No.] Tool Interference on X-axis at Positive
 end; XP
 <VTIPN> [Tool No.] Pattern number


Note: In the graphic specification, tool interference data is calculated in the NC device by the protrusion amount designated on the tool shape selection page and tool diameter and tool interference variable for automatic determination. Therefore, if the interference data is rewritten by using tool interference data variables, it is not used.

f) Variable soft-limit variable

Variable name

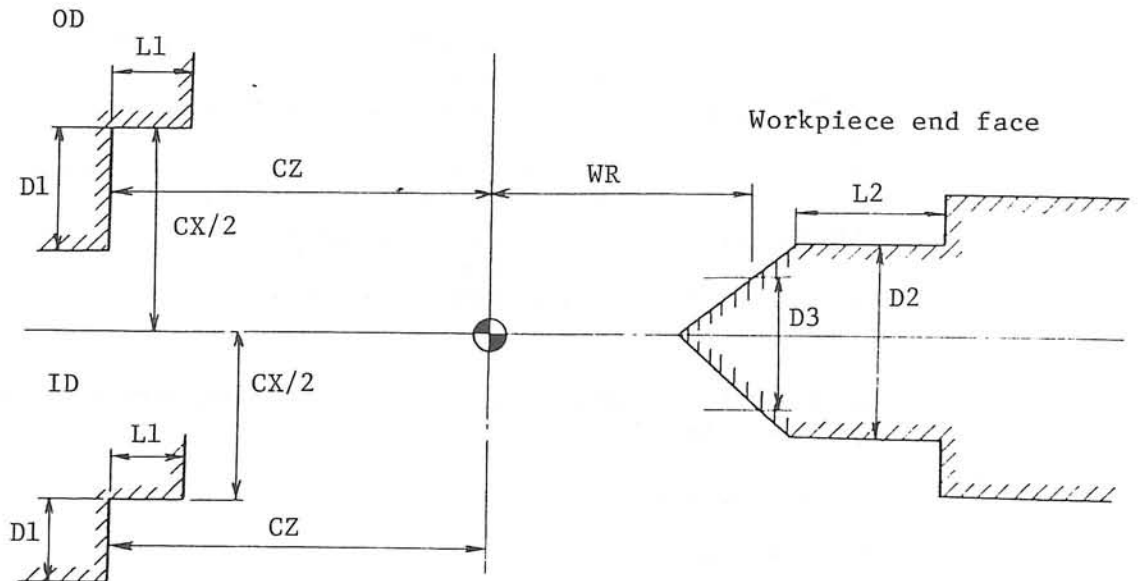
<VPVLZ> Positive Limit on Z-axis
 <VPVLX> Positive Limit on X-axis
 <VNVLZ> Negative Limit on Z-axis
 <VNVLX> Negative Limit on X-axis
 <VPVLW> Positive Limit on W-axis
 (only for programmable tailstock specification)
 <VNVLW> Negative Limit on W-axis
 (only for programmable tailstock specification)

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).

g) 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
 <VTSDA> Center position D3
 <VWKR> Workpiece end face

Numerical data of these variables are referenced to the origin of the program coordinate system (programming zero).

h) 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

i) 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.

j) Restart variable

Variable name

<VRSTT> ReSTarT

This indicates the restart state in the sequence restart operation.

$$\left\{ \begin{array}{l} \#000 = \text{Not in restart} \\ \#800 = \text{In restart} \end{array} \right.$$

Example:

N100 IF [VRSTT NE 0] N200

k) 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 3-1-8.

Note: As hexadecimal code, setting is possible up to four characters.

Output of a comment to the CRT is designated with output variable VDOUT [*]=code number. See below.

```
*=991: 909 Alarm C
*=992: 462 Alarm B
*=993: 147 Alarm A
```

As a code number, numbers 1 - 9999 can be used.

Program example 1:

```
N202 VUACM[1] = 'ABC=100 L'
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.

CRT display:

```
909 Alarm C User dedicated 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.

CRT display:

```
462 Alarm B User dedicated code 1000 PART=GEAR
```

Program example 4:

```
N301 VUACM[1] = $41424344
N302 VDOUT[992] = 2000
:
:
```

CRT display:

462 Alarm B User dedicated code 2000 ABCD

(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.	Panel input
10	5 bits data of hexadecimal number, S0 through S1F	Panel input

Input Variable No.	Contents of Data	Input Equipment
11 - 18 19	Bit data: 0 (OFF), 1 (ON) 1 byte data in which data of variables #11 through #18 are corresponded to bit 0 through bit 7.	External input (EC) External input (EC)
21 - 22	Bit data: 0 (OFF), 1 (ON)	Panel input
23 - 24	Bit data: 0 (OFF), 1 (ON)	External input (EC)
31 - 38 39	Bit data: 0 (OFF), 1 (ON) 1 byte data in which data of variables #31 through #38 are corresponded to bit 0 through bit 7.	Panel output
81 - 88 89	Bit data: 0 (OFF), 1 (ON) 1 byte data in which data of variables #81 through #88 are corresponded to bit 0 through bit 7.	Robot input
801 - 812	1 byte data	Post-processing input
821 - 832	1 word data	Post-processing input
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)	
1100 - 1123	Condition of external inputs: #1 through #24	External input (EC)
1124 - 1149	Condition of panel inputs: #1 through #26	Panel input
1150 - 1165	Condition of external outputs: #1 through #16	External output (EC)
1166 - 1194	Condition of panel outputs: #1 through #29	Panel output
1195 - 1218	Condition of extended external inputs: #1 through #24	EC extended input
1219 - 1234	Condition of extended external inputs: #1 through #16	EC extended output
1235 - 1250	Condition of specification codes: #1 through #16	

Note: Details of specifications must be discussed with a customer.

Output Variable No.	Contents of Data	Output Equipment
991	Alarm C User code	Alarm output
992	Alarm B User code	Alarm output
993	Alarm A User code	Alarm output

Note: Details of specifications must be discussed with a customer.

22-3-3. 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	(<u>L</u> ess <u>T</u> han, <)	IF [V1 LT 5] N100	Jump to N100 when V1 is less than 5.	Provide a space on either side of the operator.
LE	(<u>L</u> ess than or <u>E</u> qual to, ≤)	IF [V1 LE 5] N100	Jump to N100 when V1 is less than or equal to 5.	
EQ	(<u>E</u> qual to, =)	IF [V1 EQ 5] N100	Jump to N100 when V1 is equal to 5.	
NE	(<u>N</u> ot <u>E</u> qual to, ≠)	IF [V1 NE 5] N100	Jump to N100 when V1 is not equal to 5.	
GT	(<u>G</u> reater <u>T</u> han, >)	IF [V1 GT 5] N100	Jump to N100 when V1 is greater than 5.	
GE	(<u>G</u> reater than or <u>E</u> qual, ≥)	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.
COS	Cosine	Z = 15*COS [22.5]	
TAN	Tangent	Z = 15*TAN [12.5]	
ATAN	Arctangent (1) Value range: -90° to 90°	X = 15*ATAN [22.5]	
ATAN2	Arctangent (2)*	ATAN2 [10.15]	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 μ m unit system, 0.0001 deg. for 0.0001 inch unit system.
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 below 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 system) or below the fourth place of decimals (inch system) as a whole number	DFUP [V21]	
MOD	Remainder (a-fix[a/b]*b)		

*: The value of ATAN2 [b,a] represents the angle of the point defined by the coordinate values (a, b). Its range is from -180° to 180° .

22-3-4. Combination of Operations

- (1) Operations and functions explained in the previous page can be combined as needed.

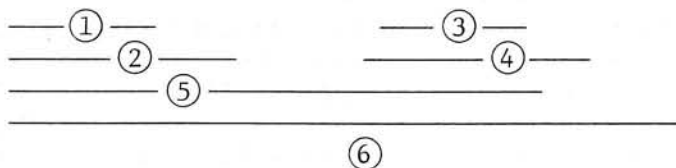
$$X = V1 + V2 - V3 + V4 * \text{COS} [30]$$

- (2) Designating operator precedence by []

Operator precedence can be determined with the use of [].

Example:

$$V1 = [V2 + V3] * V4 + \text{SIN} [[V5 - V6]/V7] + V8$$



22-4. 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]

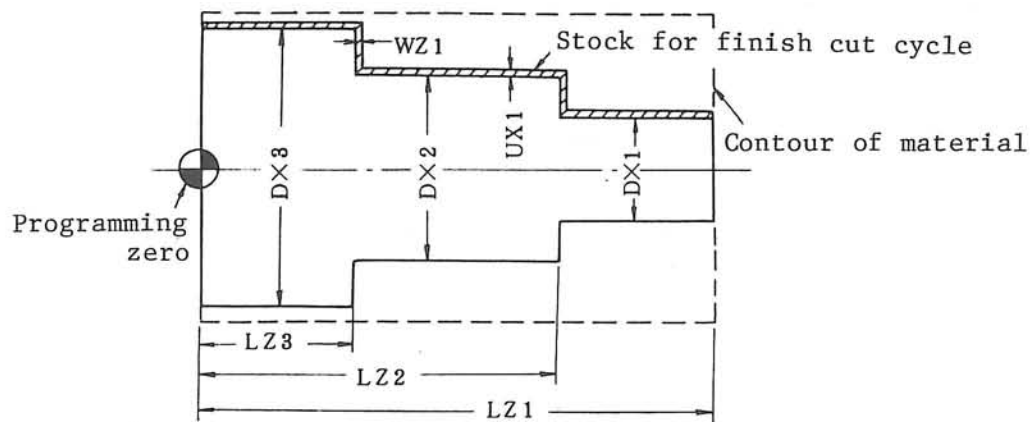


Fig. 22-3

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
LZ1 = Longitudinal dimension LZ1	WZ1
LZ2 = Longitudinal dimension LZ2	UDX1 = Finish allowance on diameter UX1
LZ3 = Longitudinal dimension LZ3	Xs = X coordinate of LAP starting point
	Zs = Z coordinate of LAP starting point

- (4) For cutting a workpiece, LAP mode is used.

- (5) Described above are summarized in Table 22-1.

Table 22-1

		Work A	Work B	Work C
		SHAFT-A	SHAFT-B	SHAFT-C
Roughing tool	V1	Ø1Ø1	Ø3Ø3	Ø5Ø5
Finishing tool	V2	Ø2Ø2	Ø4Ø4	Ø6Ø6
Cutting speed in roughing cycle	V3	1ØØ	11Ø	9Ø
Cutting speed in finishing cycle	V4	12Ø	13Ø	15Ø
LZ1	LZ1	2ØØ	25Ø	3ØØ
LZ2	LZ2	15Ø	17Ø	2ØØ
LZ3	LZ3	8Ø	1ØØ	12Ø
DX1	DX1	3Ø	4Ø	5Ø
DX2	DX2	5Ø	7Ø	9Ø
DX3	DX3	8Ø	12Ø	15Ø
WZ1	WLZ1	Ø.1	Ø.15	Ø.2
UX1	UDX1	Ø.2	Ø.25	Ø.3
Xs	Xs	1ØØ	14Ø	17Ø
Zs	Zs	21Ø	26Ø	3ØØ

The subprogram defining the contour, prepared using local and common variables, can be programmed as below according to Table 22-1.

\$ SHAFT-ABC.SUB						
%						
O1ØØØ						
NLAP1 G81						
N1ØØ1 GØØ X=DX1 Z=LZ1+2						
N1ØØ2 GØ1 Z=LZ2 FØ.2						
N1ØØ3 X=DX2						
N1ØØ4 Z=LZ3						
N1ØØ5 X=DX3						
N1ØØ6 Z=Ø						
N1ØØ7 G8Ø						
N1Ø1Ø GØØ X8ØØ Z=4ØØ						
N1Ø11 G96 X=Xs Z=Zs S=V3 T=V1 MØ3 MØ8						
N1Ø12 G85 NLAP1 D4 FØ.35 U=UDX1 W=WLZ1						
N1Ø2Ø GØØ X8ØØ Z4ØØ						
N1Ø21						
N1Ø22 G87 NLAP1						

The main program for each workpiece is programmed as below using the subprogram.

Workpiece A							
\$	SHAFT-A.MIN					 a)
%						 b)
O100						 c)
N101	G00	X800	Z400				
N102	CALL	O1000	V1=0101	V2=0202	V3=100	V4=120 d)
	LZ1=200	LZ2=150	LZ3=80				
\$	DX1=30	DX2=50	DX3=80	WLZ1=0.1	UDX1=0.2	XS=100	ZS=210
N103	M02						

- a) File name of the main program ... Program \$ proceeding the file name.
- b) Specify % code at the beginning of a program.
- 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						
%							
O101							
N101	G00	X800	Z400				
N102	CALL	O1000	V1=0303	V2=0404	V3=110	V4=130	LZ1=250
	LZ2=170	LZ3=100					
\$	DX1=40	DX2=70	DX3=120	WLZ1=0.15	UDX1=0.25	XS=140	ZS=260
N103	M02						

Workpiece C							
\$	SHAFT-C.MIN						
%							
O102							
N101	G00	X800	Z400				
N102	CALL	O1000	V1=0505	V2=0606	V3=90	V4=150	LZ1=300
	LZ2=200	LZ3=120					
\$	DX1=50	DX2=90	DX3=150	WLZ1=0.2	UDX1=0.3	XS=170	ZS=360
N103	M02						

<Program Example 2> (Cutting contour requiring calculation for defining 1)

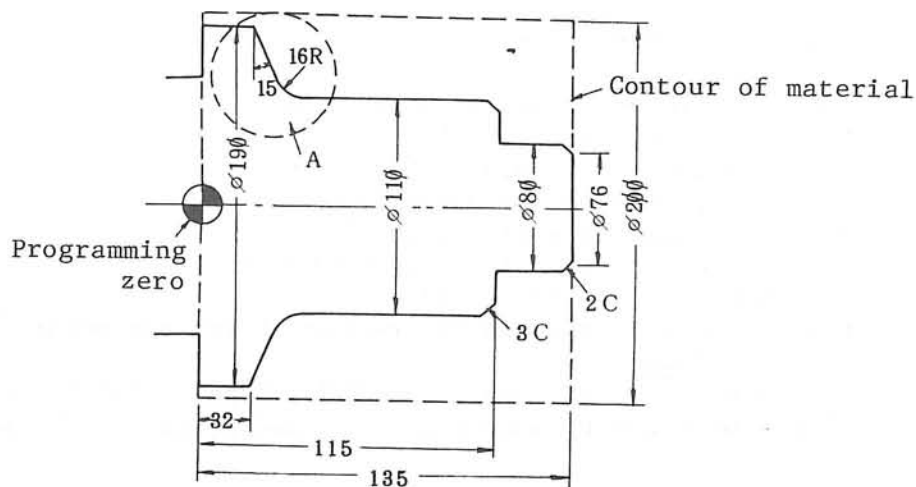


Fig. 22-4

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.

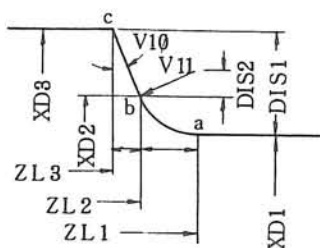


Fig. 22-5

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

V1Ø = Taper angle (15°)
 V11 = Arc radius (16 mm)

Set using local variables

XD1 = Diameter of point "a" (11Ø mm)
 XD2 = Diameter of point "b"
 XD3 = Diameter of point "c" (19Ø 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 [V1Ø]
 DIS3 = V11*COS [V1Ø]
 DIS4 = [DIS1 + DIS2 - V11]*TAN [V1Ø]
 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 "RADIUS-TAPER.SUB" (file name). 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				
%				
O1				
N1000	XD2=XD1+2*[V11-DIS2]	ZL2=ZL3+DIS4		
\$	ZL1=ZL2+DIS3			
N1001	G01	Z=ZL1		
N1002	G02	X=XD2	Z=ZL2	L=V11
N1003	G01	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=190	ZL3=32		
N102	G00	X800	Z300				
NLAP1 G81							
N103	G00	X76	Z137				
N104	G42	G01	Z135	F0.2			
N105	G75	X80	L-2				
N106			Z115				
N107	G75	X=XD1	L-3				
N108	CALL	ORT01	DIS1=[XD3-XD1]/2	DIS2=V11*SIN[V10]			
	\$	DIS3=V11*COS[V10]	DIS4=[DIS1+DIS2-V11]*TAN[V10]				
N109			Z-2				
N110	G40	G00	X200	I10			
N111	G80						
N120	G00	X800	Z300				
N121	G96	X202	Z137	S110	T010101	M03	M08
N122	G85	NLAP1	D6	F0.35	U0.2	W0.1	
N123	G00	X800	Z300				
N124			S130	T020202			
N125	G87	NLAP2					
N126	G00	X800	Z300				
				M05	M09		
N128	M02						

Note: The subprogram ORT01 is called for by the command in block N108 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.

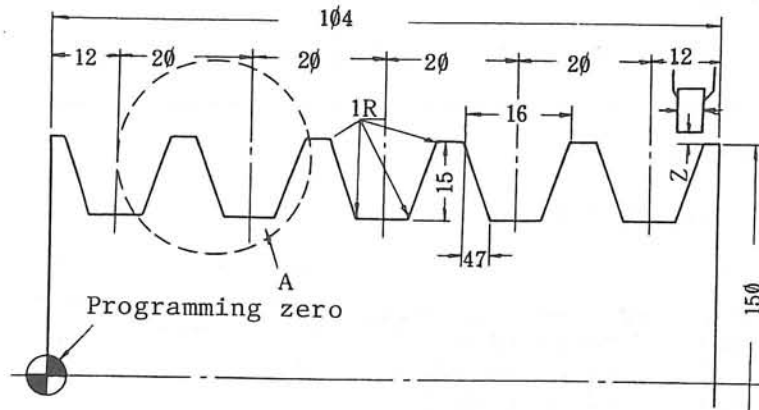


Fig. 22-6

- (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.

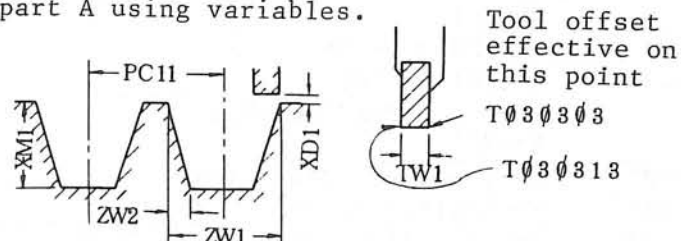


Fig. 22-7

Variable Name	Contents	Numerical Value for This Example (Fig. 22-6)
PC11	Pitch between the pulley grooves	20 mm
XH1	Height of pulley groove	15
XD1	Starting point of cutting	2
ZW1	Width of groove	16
ZW2	Taper amount of groove in Z-axis direction	4.7
TW1	Width of cutting off tool	6
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			
%			
OPP1			
N1	G91	G00	Z=-(ZW1/2)+(TW1/2)
N2	G42		X=-XD1+0.2
N3	G01		X=-(XH1*2)-0.2 F0.05
N4	G00		X=(XH1*2)+XD1
N5			Z=ZW2
N6			X=-XD1
N7	G01		X=-(XH1*2) Z=-ZW2
N8	G00		X=(XH1*2)+XD1
N9	G41		Z=-ZW2 T030313
N10			X=-XD1
N11	G01		X=-(XH1*2) Z=ZW2
N12	G00		X=(XH1*2)+XD1
N13			Z=-ZW2-DK-0.3
N14			X=-XD1
N15	G02		X=-(1-D1)*2 Z=DK I=-1
N16	G01		X=-(XH1*2)+(1-D1)*4 Z=ZW2-DK
N17	G03		X=-(1-D1)*2 Z=DK I=DI K=DK
N18	G00		X=(XH1*2)+XD1
N19			Z=ZW2+DK+(0.3*2) T030303
N20	G42		X=-XD1
N21	G03		X=-(1-D1)*2 Z=-DK I=-1
N22	G01		X=-(XH1*2)+(1-D1)*4 Z=-ZW2+DK
N23	G02		X=-(1-D1)*2 Z=-DK I=DI K=-DK
N24	G00		X=-(XH1*2)+XD1
N25			Z=ZW2
N26	G90		
N27	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. 22-6 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	G96		S70	T030303	M03
N004		PHC=20	XD1=2		
N005			Z100		
N006	MODIN	OPPI	XH1=15	ZW1=16	ZW2=4.7 TW1=6
			D1=0.299	DK=0.954	
N007	G00	X=150+XD1	Z100		
N008	G00		Z=100-PCH		
N009	G00		Z=100-[2*PCH]		
N010	G00		Z=100-[3*PCH]		
N011	G00		Z=100-[4*PCH]		
N012	MODOUT				
N013	G40	G00	X800	Z300	M05
N014	M02				

The MODIN statement in block N004 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 OPPI are also set.

In blocks N005 through N009, the subprogram OPPI 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.

22-5. SUPPLEMENTS

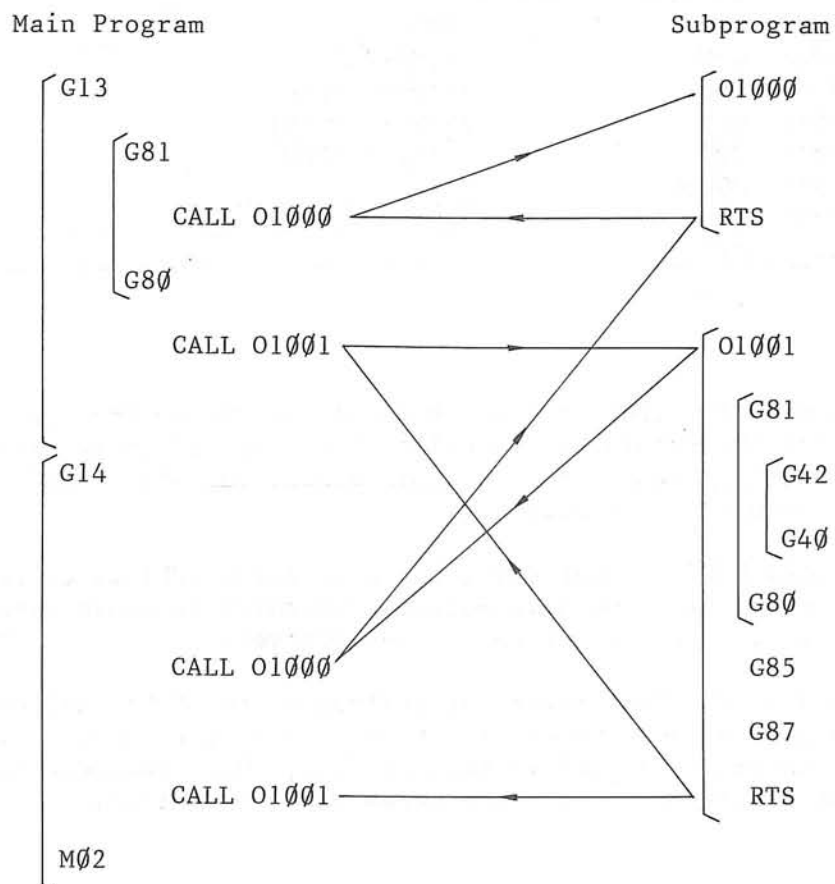
22-5-1. Sequence Return in Program Using User Task

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.

22-5-2. Simultaneous 4-Axis Control Mode

Selection of the turret can be made only in the main program. The same subprogram can be called from G13 and G14 turrets. Shown below is the example of combination of simultaneous 4-axis control, LAP, nose radius compensation, and subprogram.



Note: Up to 125 subprograms can be used in a main program and there is no limit in the number of subprogram files which can be called from a main program.

22-5-3. 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 (1) 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.

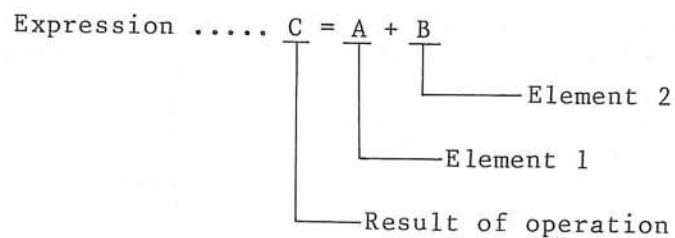
Note: Abbreviation symbol

[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 1/100000 inch	[I]	Not changed
		[R]	[R] → [I] (inch system value is converted into metric system value) (rounding off fractions)
	mm inch	[I]	[I] * 10000 (metric system) [I] * 100000 (inch system)
		[R]	[R] * 10000 → (metric system) (rounding off fractions) [R] * 100000 → (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:



Note: 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 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]
			[R]	[R]	
Comparison expression	LT	Less than <			[b]
	LE	Less than or equal to ≤	[I]	[I]	
	EQ	Equal to =	[I] [R]	[R]	

Type of Expression	Operator	Meaning	Type of Element 1 "A"	Type of Element 2 "B"	Type of Result of Operation "C"
Comparison expression	NE	Not equal to \neq	[R]	[I] \longrightarrow [R]	[b]
	GT	Greater than $>$	[R]	[R]	
	GE	Greater than or equal to \geq			
Logical expression	EOR	Exclusive OR	[I]	[I]	[I]
	OR	Logical sum	[R] \longrightarrow [I]	[R] \longrightarrow [I]	
	AND	Logical product	(Cutting off fractions)	(Cutting off fractions)	
	NOT	Negation	[I] or [R] \longrightarrow [I] (Cutting off fractions)		

(5) Operation rule of functions and evaluation of value

Note: 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	Sine	1 mm	[I] → [R]	[R]/1000 deg. (metric system)	[R]
COS	Cosine	1/100000 inch	[R]	[R]/1000 deg. (inch system)	[R]
TAN	Tangent	mm inch	[I] → [R] [R]	(degree)	[R] [R]
ATAN ATAN2	Arctangent	1 mm 1/100000 inch	[I] → [R] [R]		[R]*1000 (1/1000 degree) (metric system) [R]*100000 (1/100000 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]

SECTION 23 INCH/METRIC SWITCHABLE

The OSP500L-G/OSP500L-G allows the system to operate under the desired dimensioning unit system by setting a proper parameter.

(1) Parameter setting

PARAMETER SET				H	G
Page 1		A turret			
* OPTIONAL PARAMETER BIT *					
No.	No.	No.	No.	No.	No.
1	01000011	9	00000000	17	00000000
2	00000000	10	00000000	18	00000000
3	00000010	11	00000000	19	00000000
4	00000000	12	00000000	20	00000000
5	00000000	13	00000000	21	00000000
6	00000000	14	00000000	22	00000000
7	00000000	15	00000000	23	00000000
8	00000000	16	00000000	24	00000000
				25	00000000
				26	00000000
				27	00000000
				28	00000000
				29	00000000
				30	00000000
				31	00000000
				32	00000000

SET	ADD	CAL			ITEM1	ITEM4	
-----	-----	-----	--	--	-------	-------	--

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
-----	-----	-----	-----	-----	-----	-----	-----

Setting of the dimensioning unit system is possible by the setting of parameter data at bit 0, 1 and 2 of optional parameter (bit) No. 3.

Bit 0 inch/metric selection
 Bit 1 mm unit (inch unit)
 Bit 2 10 μ m unit

	Bit 2	Bit 1	Bit 0	Operation Condition
A	0	1	0	1 mm unit system
B	1	0	0	10 μ m unit system
C	0	0	0	1 μ m unit system
D	0	1	1	1 inch unit system
E	0	0	1	1/10000 inch unit system

The standard setting is A.

(2) Operations

- a) Inside the system, dimension data is all processed in units of $1 \mu\text{m}$ (metric system). The units for function generation and position detection are also $1 \mu\text{m}$.
- b) For the entry of tool offset data, zero set data, parameter data and others, the data is entered in units of inches. But they are stored in units of metric system selected.

For the purpose of display, the data is converted into the unit in inch system selected and displayed on the CRT.

- c) Both program data and MDI data are input in the unit in inch system. However, they are immediately converted into the data in units of $1 \mu\text{m}$ for positioning operations.
- d) For instance, when an axis is to be fed in units of one pulse (0.0001 inches), the axis is fed in accordance with the values converted into those of $1 \mu\text{m}$ units.

<u>Commands in Inch System</u>	<u>Theoretical Metric Values</u>	<u>Metric Values for Positioning</u>
0.00001 inch	2.54μ	3.0
2	5.08	5.0
3	7.62	8.0
4	10.16	10.0
:	:	:
:	:	:

The command in inch unit is multiplied by 2.54 to be converted into metric values. Fractions smaller than $1 \mu\text{m}$ are rounded.

(3) Unit table (units when "1" is designated)

	Metric System			Inch System	
	1 μm	1 \emptyset μm	1 mm	1/1000000 inch	1 inch
Length: X,Z,I, K,D,H,L,U,W	$\emptyset.\emptyset\emptyset 1$ (mm)	$\emptyset.\emptyset 1$ (mm)	1 (mm)	$\emptyset.\emptyset\emptyset\emptyset 1$ (inch)	1 (inch)
Feed (/rev): F,E	$\emptyset.\emptyset\emptyset 1$ (mm/rev)	$\emptyset.\emptyset 1$ (mm/rev)	1 (mm/rev)	$\emptyset.\emptyset\emptyset\emptyset 1$ (inch/rev)	1 (inch/rev)
Feed (/min): F,E	$\emptyset.1$ (mm/min)	1 (mm/min)	1 (mm/min)	$\emptyset.\emptyset 1$ (inch/min)	1 (inch/min)
Angle: A,B,C	$\emptyset.\emptyset\emptyset 1$ (deg.)	$\emptyset.\emptyset 1$ (deg.)	1 (deg.)	$\emptyset.\emptyset\emptyset\emptyset 1$ (deg.)	1 (deg.)
Time: F,E	$\emptyset.\emptyset 1$ (sec)	$\emptyset.1$ (sec)	1 (sec)	$\emptyset.\emptyset 1$ (sec)	1 (sec)
Spindle RPM: S	1 (rpm)	1 (rpm)	1 (rpm)	1 (rpm)	1 (rpm)
Surface speed: S	1 (m/min)	1 (m/min)	1 (m/min)	1 (feet/min)	1 (feet/min)

SECTION 24 GRAPHIC CRT DISPLAY

24-1. OUTLINE OF GRAPHIC CRT DISPLAY

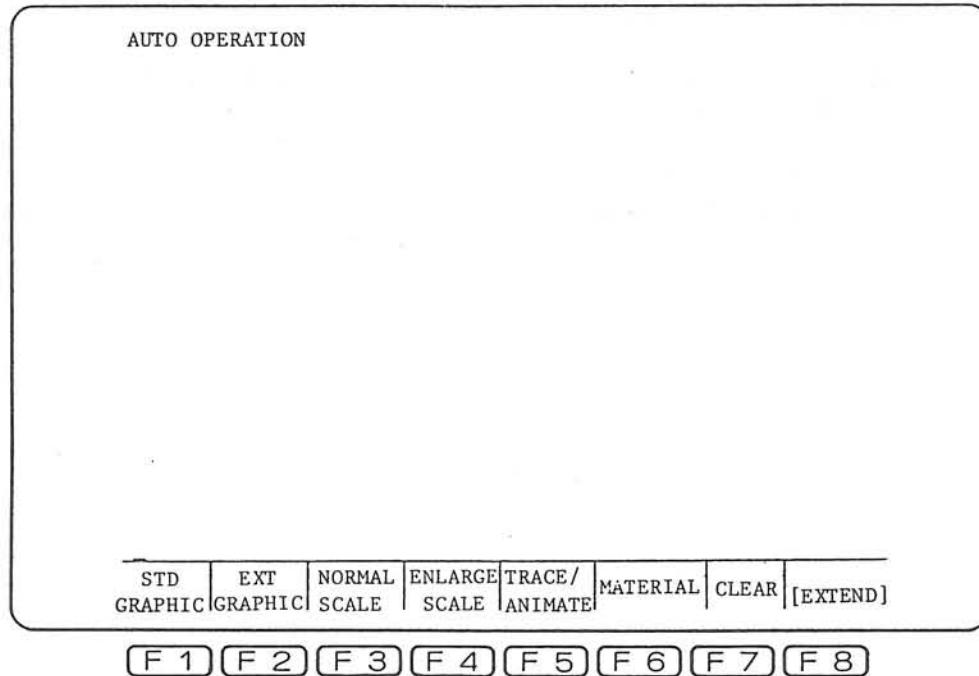
This function displays the progress of the program in animation in addition to the programmed tool paths on CRT.

The OSP5000L-G has the color graphic CRT on which workpieces, cutting tools, etc. are differentiated by colors.

The OSP500L-G has the monochrome graphic CRT on which workpieces, cutting tools, etc. are differentiated by patterns, line types, and brightness.

24-2. FUNCTION KEYS USED FOR GRAPHIC DISPLAY OPERATION

The graphic display is possible in the auto, MDI and manual operation modes.



(1) F1 (STD GRAPHIC)

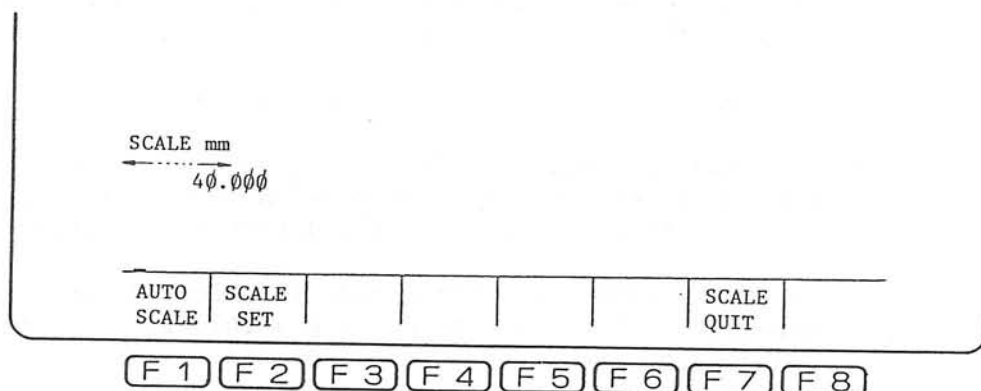
Graphic display mode in the scale set from the NORMAL SCALE function is selected.

(2) F2 (EXT GRAPHIC)

Graphic display mode in the scale set from the ENLARGE SCALE function is selected.

Note: Switching between the standard and enlarged graphic display modes during the execution of a program is impossible.

(3) F3 (NORMAL SCALE)



This function selects the unit length of an axis on the graphic display. On the graphic display, a dotted line with arrow marks at both ends is displayed with SCALE indication and scale value. This represents the scale length.

The standard scale is set in two different methods such as

- a) Automatic determination
- b) Arbitrary setting

The procedure for setting the standard scale is explained below. Note that the standard graphic page must be displayed for setting the standard scale.

- a) Automatic determination

The graphic scale is automatically determined by pressing the function key [F1] (AUTO SCALE). When this key is pressed, the program is read up to the end of the program (M02) and the scale is determined automatically so that the machining area can be displayed on the CRT.

Remarks:

- 1) Before executing the automatic determination, select the desired program.
- 2) If following commands are used in a part program, they are executed when the program is read by the pressing of the function key [F1] (AUTO SCALE).

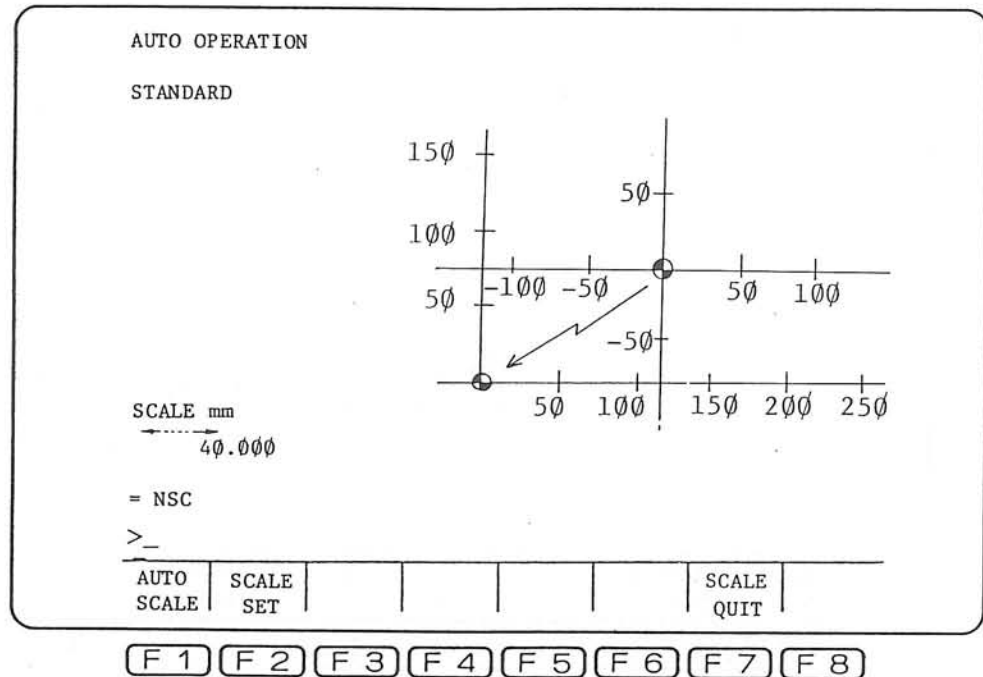
READ, WRITE, GET, PUT, DELETE, SAVE and DEF

- 3) If output variables, system variables and common variables are used in a left member of expression in a part program, they are rewritten by the automatic determination operation.
- 4) If the scheduled operation mode is selected, automatic determination of the scale value is possible only for the program selected by the first PSELECT command.
- 5) If an alarm occurs during the reading of a program for automatic determination, or if the program reading is looped due to the statements GOTO, IF and other program execution order designating commands and automatic determination of the scale is not completed, then escape such state by resetting the control.
- 6) The drawing area automatically determined is calculated from the following formula.

$$\left(\begin{array}{l} \text{Operating area in the} \\ \text{cutting G code mode} \end{array} \right) \times (100 + n)\%$$

Here n is a factor to allow margin in the tool path drawing operation and set by optional parameter (word) No. 51.
 $(0 \leq n \leq 100)$

b) Arbitrary determination

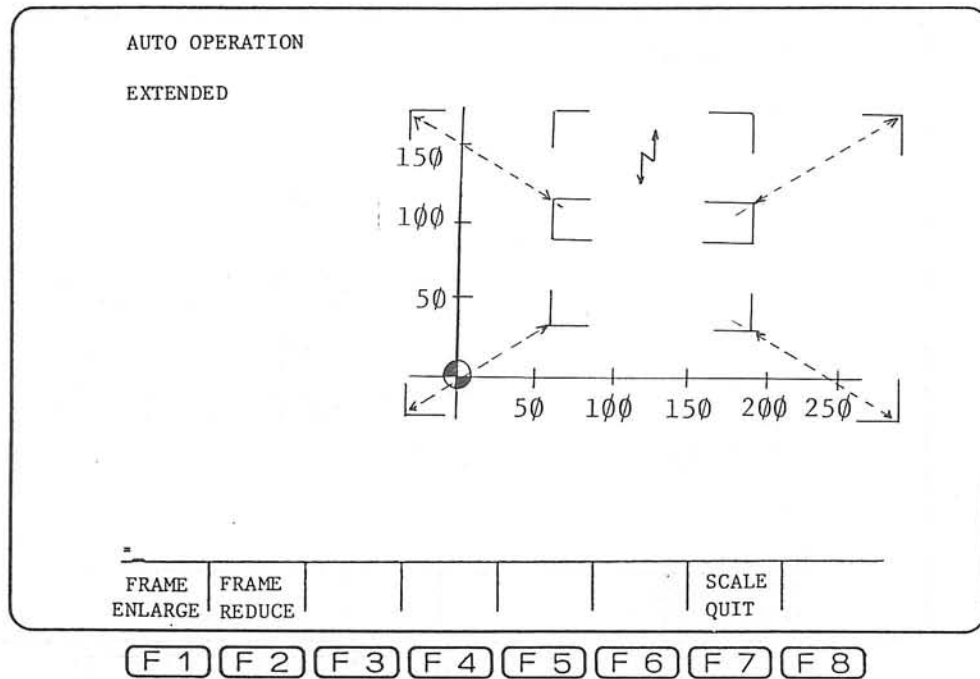


The scale value can be directly entered through the keyboard after pressing the function key [F2] (SCALE SET). In this scale setting, setting range is from 12.5 mm to 1250 mm.

The position of the coordinate axes can be set at a required position using the cursor after setting the scale value.

The standard scale setting operation ends by pressing the function key [F7] (SCALE QUIT).

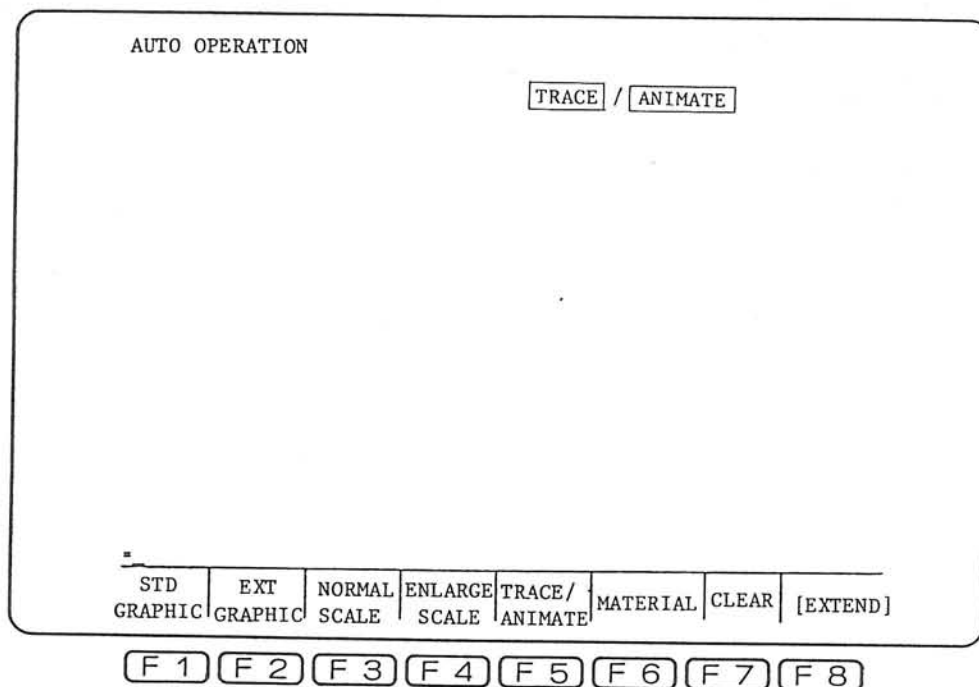
(4) F4 (ENLARGE SCALE)



The display area to be zoomed up from the standard graphic display or enlarged graphic display can be set as needed.

Pressing the function key [F1] (FRAME ENLARGE) reduces the frame indicating the display area. Pressing the function key [F2] (FRAME REDUCE) enlarges the display area indicating frame. Setting the position of the display area frame is made using the cursor key. The enlarge scale setting function terminates when the function key [F7] (SCALE QUIT) is pressed.

(5) F5 (TRACE/ANIMATE)



The graphic display mode is selectable from the three modes indicated below by pressing the function key [F5] (TRACE/ANIMATE). Note that selection of the graphic display mode must be made before starting the operation. The selected mode can not be changed during the operation.

a) TRACE/ANIMATE mode

In this mode, tool shape, chuck shape, blank shape and tailstock spindle shape are displayed in addition to programmed tool paths. The blank shape is actually removed in accordance with the progress of the part program.

b) TRACE mode

In this mode, only the programmed tool paths are drawn.

c) ANIMATE Mode

In this mode, tool shape, chuck shape, blank shape and tailstock spindle shape are displayed and blank shape is actually removed in accordance with the progress of the part program.

Display of the tool paths is not available.

(6) F6 (MATERIAL)

The blank shape, chuck shape and tailstock spindle shape registered are displayed.

(7) F7 (CLEAR)

Tool paths, blank shape, chuck shape and tailstock spindle shape displayed on the CRT are all cleared.

24-3. GENERAL PRECAUTIONS FOR ANIMATION MODE DISPLAY

(1) Scale value setting range

Setting ranges of the scale value are indicated below:

Normal scale: 12.5 mm to 125 \emptyset mm
 Enlarge scale: \emptyset .4 mm to 125 \emptyset mm

(2) Relationship between animation display and scale value

Available animation display differs depending on the scale value setting and the relationship between the set scale value and animation display available is summarized in the table below.

Function Scale Value	Tool Shape Display	Blank Removal Display	Tool Path Display
125 \emptyset mm - 5 mm	o	o	o
5 mm - 2.5 mm	x	o	o
2.5 mm - \emptyset .4 mm	x	x	o

(3) Animation drawing speed

The animation drawing function has a limit in the drawing speed (feedrate of 1 meter per min. will be the maximum drawing speed at a scale value setting of 25 mm).

This means simulation with the animation drawing in the machine lock mode automatically overrides the specified feedrate (animation speed override). In actual machining operation, however, animation speed override is impossible and if program is executed at a feedrate for which animation drawing is impossible, the control displays the message "ANIM SPEED OVER" and erases animation drawing. In this case, only programmed paths are displayed on the CRT.

Note: To clear the message "ANIM SPEED OVER", press the function key [F7] (CLEAR) after the part program has been completed.

- (4) Parameter data for selecting the elements of animation and tool paths drawing

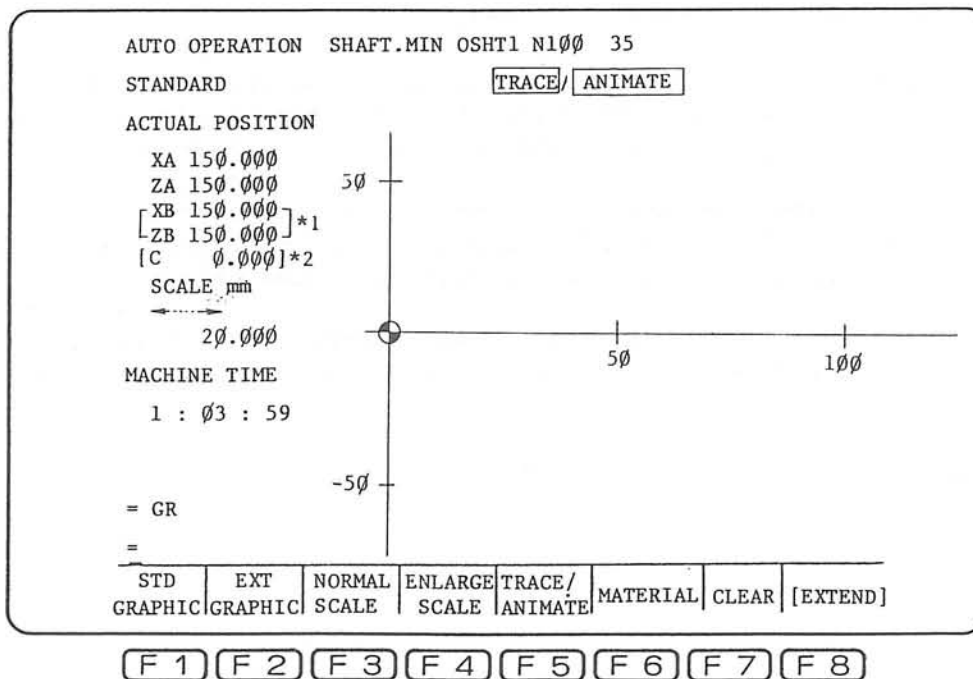
For the display or drawing of animation and tool path, display elements can be selected by a parameter data - optional parameter (bit) No. 10.

<u>Bit</u>	<u>ON/OFF</u>
Bit 0	Jog feed path drawing
Bit 1	Rapid traverse path drawing
Bit 2	Cutting feed path drawing
Bit 3	Blank shape drawing
Bit 4	Chuck shape drawing
Bit 5	Tailstock shape drawing
Bit 6	Tool shape drawing
Bit 7	Blank removal drawing

24-4. HIGH SPEED DRAWING

Programmed tool paths can be drawn at a high speed in the machine lock and dry run on mode. Setting of the single block function is effective. In this programmed path drawing operation, since the tool path in the G00 rapid traverse mode is not the same as the actual movements, this must be taken into consideration.

24-5. DISPLAY PAGE



*1 Data in [] is available for two-saddle models.

*2 Data in [] is available for multi-machining specs.

24-6. MACHINE TIME DISPLAY

Actual program execution time in the automatic and MDI mode operation is counted and displayed. While the control is in the slide hole mode or the program stop mode, such time is not counted.

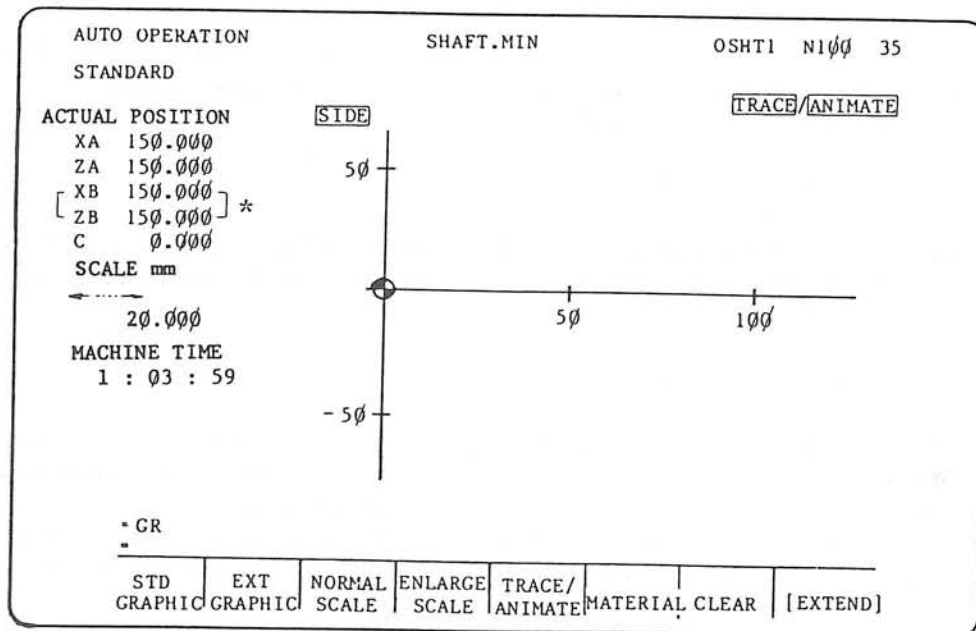
In the high speed drawing mode (machine lock and dry run on), machine time display is not available. If machine time calculation is required through the execution of the high speed drawing operation, optional function for this operation is necessary.

24-7. ADDITIONAL FUNCTIONS FOR MULTI-MACHINING MODELS

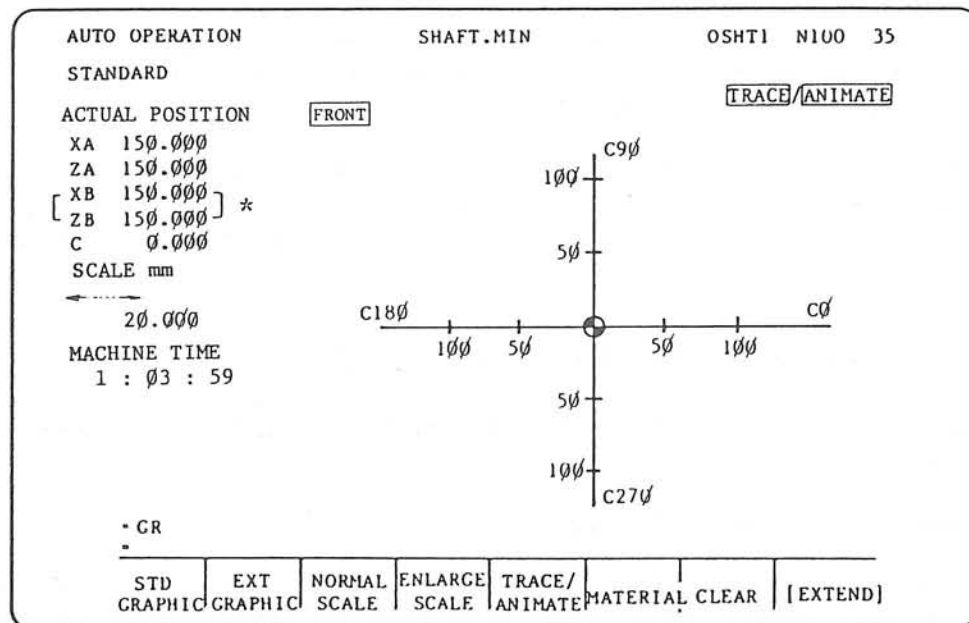
The function keys available are basically the same as used for standard models.

(1) Standard graphic display

For the blank display, the page key is used for switching the view angle from the side to the front and from the front to the side. This display change is possible even during the progress of cutting operations. Note that the page key is not effective for other than the multi-machining models.



* Data in [] is available for 2-saddle models.



* Data in [] is available for 2-saddle models.

The front view is displayed on the coordinate system on which C-axis angle is fixed as shown above. The indications on the scale represent X-axis values (in radius).

(2) Enlarged graphic display

In the enlarged graphic display mode, switching the view angle between the front and side views is possible as in the standard graphic display mode.

(3) Standard scale setting

Independent scale setting on the side and front views is possible. For setting the scale for these two views, first display the required view. When the same scale setting is used for both of the displays in common, then the setting is necessary only from either of the displays. In this case, specific scale value is set both for the front and side views.

Setting procedure is the same as used with standard models.

(4) Enlarged scale setting

In the enlarged graphic display mode, independent scale setting for the different two displays - front and side. As in the scale setting from the standard display mode, display the required view and setting should be done from that display.

(5) Switching between trace and animation displays

As with the standard models, the function key [F5] (TRACE/ANIMATE) is used for selecting the three different display modes.

a) Trace/Animate

Side View:

Tool shape, chuck shape, blank shape and tailstock spindle shape are displayed. Tool paths are also displayed in cyan and the blank is removed as the progress of cutting.

While the C-axis is connected (only for A-turret tools for two-saddle model), the tool paths are displayed in magenta and removal of blank will not occur.

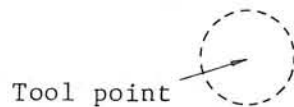
Front View:

The front view of the blank shape is displayed and the tool tip point is indicated by red marker. During the cutting, the tool outline is displayed in the specified interval*¹. Removal of blank does not occur.

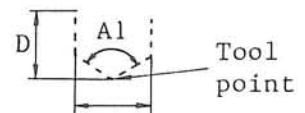
The marker and tool outline are displayed only while the C-axis is connected.

*1: The tool outline is drawn in dots in accordance with the tool mounting direction, tool diameter (D) and cutting angle (A1) when the revolving tool is selected (tool classification code: 27-38) as the tool shape.

Tool Front Display



Tool Side Display



The interval of the tool outline drawing is the distance the tool point has moved from the previous tool outline display position as indicated below:

$$\text{Drill diameter} + \frac{n}{100}$$

"n" represents the factor to designate the tool outline drawing interval and is set at optional parameter (word) No. 55 within a range of 10 to 100.

The tool outline is drawn on the display, in addition to the conditions indicated above, when the feedrate is changed from the rapid traverse to cutting feedrate or the commanded point is reached (with C-axis connected).

b) Trace

Side View:

The display is only tool paths, chuck shape and tailstock spindle shape. As in case a) above, tool path is drawn in magenta while the C-axis is in connection (only for A-turret for two-saddle model) and in cyanic in other cases.

Front View:

The tool point position is identified by red marker and tool outline is drawn in magenta in the specified interval for cutting feed (only during C-axis connection). Blank removal will not occur. Tool outline drawing conditions are the same as in a) above.

c) Animate

Side View:

Tool shape, chuck shape, blank shape and tailstock spindle shape are displayed. Tool path display is not available. Removal of blank occurs in accordance with the progress of cutting. Note that when the C-axis is in connection (only for A-turret for two-saddle model), removal of the blank does not occur.

Front View:

Same as in case a) above.

(6) Blank drawing function key

Side view:

Blank shape, chuck shape and tailstock spindle shape registered are displayed.

Front View:

Blank shape registered is displayed.

This function key is effective only for the page currently displayed. That is, pressing this key while the side view is being displayed displays only the side view of the blank and front view of the blank cannot be display. This is the same for the front view display page.

(7) Delete function key

Side View:

Tool paths, blank shape, chuck shape, the tailstock spindle shape and tool shape displayed on the side view page are all deleted.

Front View:

The tool outline drawing and the blank shape displayed on the side view page are deleted.

This function key is also effective only for the page currently displayed.

(8) Others

Parameter for canceling front view:

Setting "1" at bit 8 of optional parameter (bit) No. 19 cancels the display of the front view as with the standard models. With this parameter setting, functions (1) through (7) above are all ineffective. Initial setting of this parameter is "0" (front view display active).

Differences in side view display depending on whether the front view display is available or not are indicated below:

When the front view display is canceled, the blank shape obtained in the process of cutting is stored and can be displayed again even after the display mode is switched from the standard (enlarge) to the enlarge (standard) and then back to the standard (enlarge). When the front view display is selected effective, then the switching of the display mode cancels once. Here, the switching of the display between the standard and the enlarge indicates the switching operation while the control is not in operation.

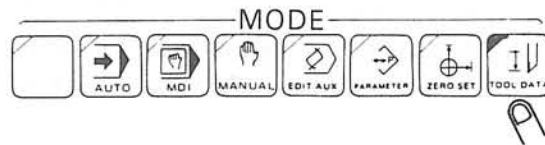
The parameter setting requires turning off of power three minutes after the new parameter data has been set.

24-8. TOOL FORM SELECTION



Procedure to set the tool form used in animation display is explained below.

The operation to set the tool form means that the data necessary to select the form pattern appropriate to the tool to be used from the tool form patterns already defined.

- 1) Select the TOOL DATA SET mode.



- 2) Press the function key [F7] (ITEM↑) to display the TOOL FORM SELECT page.

- 3) Using the PAGE keys   , display the screen for the tool number to be set.

TOOL DATA SET N 0

Page 1 UNIT 1mm

BC=1C * TOOL FORM SELECT *

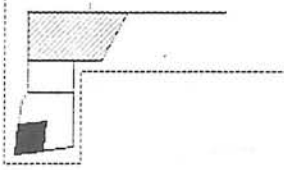
TOOL NO. 1 [TOOL CODE NO. 1 (ROUGH OD-
FORM CODE NO. 1

TOOL EDGE DATA

TOOL ANGLE	A1=	80.000
EDGE ANGLE	A2=	3.000
STICKING OUT	L =	50.000

OFFSET NO.

ON1=	1
ON2=	0
ON3=	0



INTERFER AREA

=IE

SET	ADD	TOOL KIND			ITEM1	ITEM1
-----	-----	--------------	--	--	-------	-------

- 4) Press the function key [F3] (TOOL KIND) to display the page from which the tool code number can be set.

TOOL DATA SET N 0

Page 1 UNIT 1mm

* TOOL FORM SELECT *

TOOL NO. 1 [TOOL CODE NO. 1(ROUGH OD-
FORM CODE NO. 1

- TOOL CODE TABLE -

NO.	NO.	NO.
1 ROUGH OD-	10 FINISH OD-	19 GROOVE OD1
2 ROUGH ID-	11 FINISH ID-	20 GROOVE ID1
3 ROUGH FACE1	12 FINISH FACE1	21 GROOVE FACE-
4 ROUGH OD-	13 THREAD OD-	22 DRILL, HSS
5 ROUGH ID-	14 THREAD ID-	23 DRILL, CARBIDE
6 ROUGH FACE1	15 THREAD FACE1	24 DRILL, CENTER
7 FINISH OD-	16 THREAD OD-	25 RECESS OD1
8 FINISH ID-	17 THREAD ID-	26 RECESS ID1
9 FINISH FACE1	18 THREAD FACE1	

=TK
tool item code NO. 13

SET	ADD	TOOL KIND			ITEM1	ITEM4
-----	-----	--------------	--	--	-------	-------

[F 1] [F 2] [F 3] [F 4] [F 5] [F 6] [F 7] [F 8]

- 5) Input the tool code number.

If keys [3] and [WRITE] are pressed, the display indicated below is displayed.

TOOL DATA SET N 0

Page 1 UNIT 1mm

* TOOL FORM SELECT *

TOOL NO. 1 [TOOL CODE NO. 1(ROUGH OD-
FORM CODE NO. 1

- FORM PATTERN FIGURE -

=TK
tool item code NO. 13
tool form code NO. 13

SET	ADD	TOOL KIND			ITEM1	ITEM4
-----	-----	--------------	--	--	-------	-------

[F 1] [F 2] [F 3] [F 4] [F 5] [F 6] [F 7] [F 8]

24-9. DRAWING BLANK SHAPE

The user graphic commands (UGC) are provided so that arbitrary blank shapes may be drawn on the color graphic CRT.

(1) Definition Format

The format used for blank shape drawing is indicated below.

DEF WORK Declaration of blank definition

A group of user graphic commands used for drawing blank shape

END End of blank shape definition

DRAW Blank shape defined is drawn

(2) User Graphic Commands (UGC)

The user graphic commands are largely classified into four types such as drawing point setting, line drawing, circle drawing and painting.

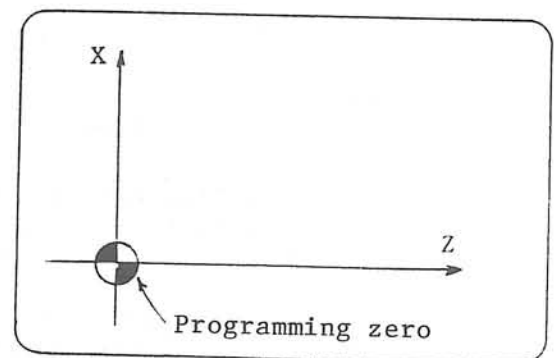
The explanation below provides the format of the representative commands.

1) Coordinate System for Drawing and Setting Unit

The coordinate system used for drawing the blank shape is the Z-X coordinate system having the programming zero as the origin.

The unit system used is:

1 mm for metric system, and
 \emptyset .1 inches for inch system



2) Setting Drawing Point

```
POINT [ZØ, XØ]
```

Note: The command underlined may be used instead of fully giving command characters. POINT, for instance, can be commanded by the underlined characters PO.

The POINT command simply sets the starting point (ZØ, XØ) for the drawing and no actual drawing operation occurs.

This sets the last reference point (LP) for the drawing.

3) Drawing Straight Line

```
LINE [Z1, X1]<,line-code>
```

A straight line is drawn from the last reference point (LP) of the drawing up to the commanded end point (Z1, X1). The end point is referenced to the LP. That is, the command must be given in a relative value. After the execution of the command, the LP is established at the end point of this command.

Line code: Ø = Solid line _____
 1 = Dotted line -----
 2 = Short dashes line - - - - -
 3 = Long dashes line _____
 4 = Alternate short and long dashes line - - - - -
 5 = Alternate long and two short dashes line - - - - -
 6 = No axis display
 7 = Erasing

Default is "Ø (solid line)".

4) Circle Drawing

```
CIRCLE [Z1, X1], [Zc, Xc] <,rotation-direction><,line-code>
```

An arc is drawn in the following conditions:

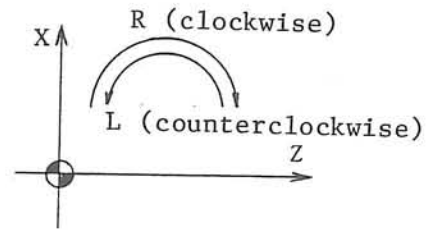
Starting point: LP
 End point : (Z1, X1)
 Center : (Zc, Xc)

The coordinates of the end point and the center are referenced to the LP. After the execution of the command, the LP is established at the end point of this command.

Rotation Direction:

R: Clockwise
L: Counterclockwise

Default is "R (clockwise)".



Line code: Eight codes as used for drawing a straight line.

5) Painting

PAINT (or PF) <number-of-vertexes><,vertex-coordinates>
<,tile-pattern>

The PAINT command is used for painting triangles, rectangles and circles defined on the absolute coordinate system.

Number of Vertexes:

∅: Circles or sectors
3: Triangles
4: Rectangles

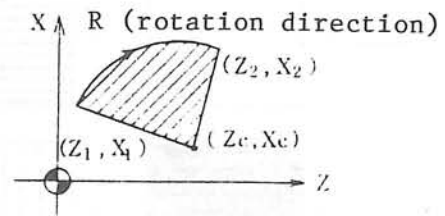
Vertex Coordinates:

The designation of coordinates depends on the number of vertexes specified.

Number of vertexes = ∅

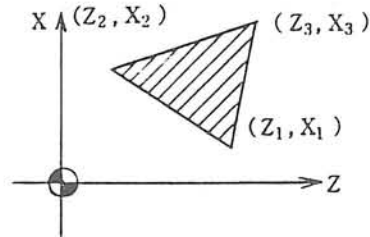
[Z1, X1], [Z2, X2] [Zc, Xc] <,rotation-direction>

Z1, X1: Start point
Z2, X2: End point
Zc, Xc: Center



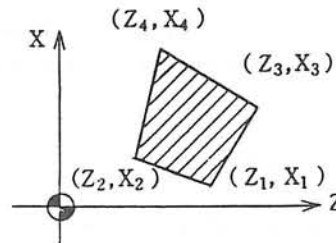
Number of vertexes = 3

$[Z_1, X_1], [Z_2, X_2], [Z_3, X_3]$
 $(X_1 \leq X_2 \leq X_3)$



Number of vertexes = 4

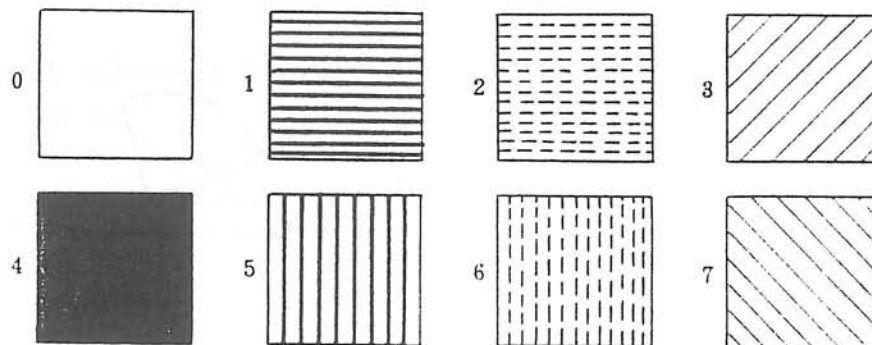
$[Z_1, X_1], [Z_2, X_2], [Z_3, X_3], [Z_4, X_4]$
 $(X_1 \leq X_2 \leq X_3 \leq X_4)$



Note that the coordinates of vertexes must be given in absolute values.

Tile Pattern:

- ∅: No drawing (all dots off)
- 1: All dots painted; on every other horizontal line
- 2: Dots alternately on and off; on every other horizontal line
- 3: Oblique line; from upper right to lower left
- 4: All dots painted
- 5: All dots painted; on every other vertical line
- 6: Dots alternately on and off; on every other vertical line
- 7: Oblique line; from upper left to lower right



Default is "4".

$\underline{\text{PAINTI}}$ <number-of-vertexes><,reference-point-coordinates> <,vertex-coordinates><,tile-pattern>
--

The PAINTI command is used for painting triangle, rectangles and circles defined on the incremental coordinate system in reference to the reference point specified.

Number of Vertexes:

\emptyset : Circles or sectors
 3: Triangles
 4: Rectangles

Reference Point Coordinates:

The coordinates of the reference point for defining a shape;

$[Z\emptyset, X\emptyset]$

The values must be given in absolute values.

Vertex Coordinates:

The designation of coordinates depends on the number of vertexes specified.

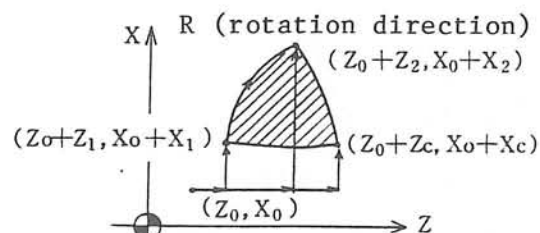
Number of vertexes = \emptyset

$[Z2, X2], [Z2, X2] [Zc, Xc] <,rotation-direction>$

$Z1, X1$: Start point

$Z2, X2$: End point

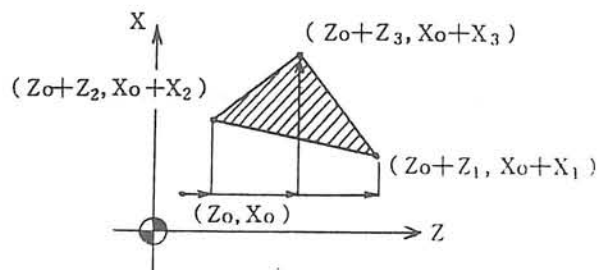
Zc, Xc : Center



Number of vertexes = 3

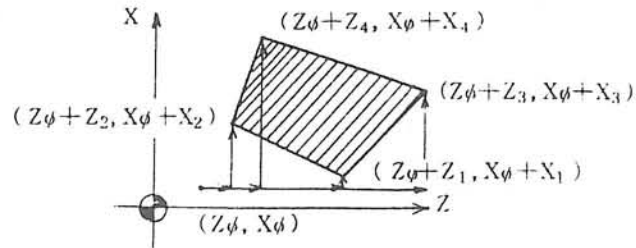
$[Z1, X1], [Z2, X2], [Z3, X3]$

$(X1 \leq X2 \leq X3)$



Number of vertexes = 4

$[Z_1, X_1], [Z_2, X_2], [Z_3, X_3], [Z_4, X_4]$
 $(X_1 \leq X_2 \leq X_3 \leq X_4)$



Tile Pattern: Eight patterns, \emptyset through 7.

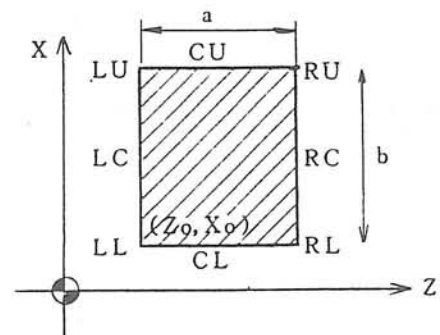
PAINTS <reference-point-position>, $[Z\emptyset, X\emptyset]$, $[a, b]$
 <,tile-pattern>

The PAINTS command is used for painting a rectangle having lengths of a and b.

Reference Point Position:

This specifies the position of the reference point on the rectangle being painted.

LL: Left lower
 LC: Left center
 LU: Left upper
 CU: Center upper
 RU: Right upper
 RC: Right center
 RL: Right lower
 CL: Center lower



Reference Position $[Z\emptyset, X\emptyset]$:

The coordinates must be specified in absolute values.

Tile Pattern: Eight patterns, \emptyset through 7.

$\underline{\text{PAINTP}}$ <reference-point-position>,[Z \emptyset , X \emptyset],[a, b] <,angle-code><,tile-pattern>
--

The PAINTP command is used for painting rectangles and triangles.

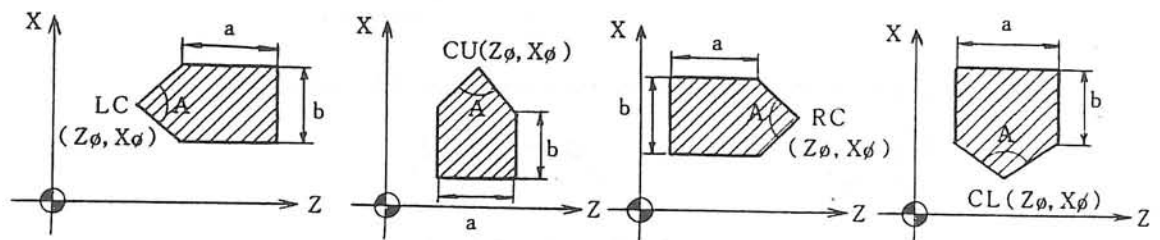
Reference Point Position:

This specifies the position of the reference point on the rectangle and triangle being painted.

LC: Left center
 CU: Center upper
 RC: Right center
 CL: Center lower

Reference Position [Z \emptyset , X \emptyset]:

The coordinates must be specified in absolute values.



Angle Code:

\emptyset : Angle A of triangle is $3\emptyset^\circ$
 1: Angle A of triangle is $6\emptyset^\circ$
 2: Angle A of triangle is $12\emptyset^\circ$

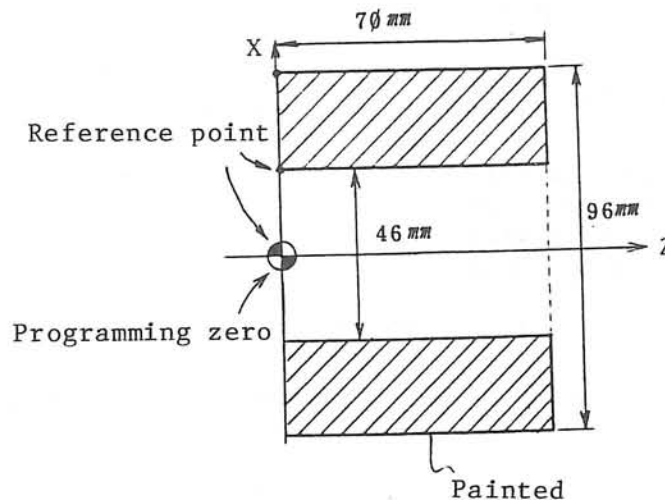
Default is " $\emptyset = 3\emptyset^\circ$ ".

Tile Pattern: Eight patterns, \emptyset through 7.

(3) Example of Painting

a) Blank material viewed from side

This item explains the procedure to paint the blank shape indicated below.



This painting operation uses the PAINTS command.

Reference point $[Z\emptyset, X\emptyset] = [\emptyset, \emptyset]$
 Dimensions $[a, b] = [7\emptyset, 96]$

PAINTS LC, $[\emptyset, \emptyset]$, $[7\emptyset, 96]$,4

The opening in the blank should be defined using the tile pattern "Ø" which represents all dots off.

Reference point $[Z\emptyset, X\emptyset] = [\emptyset, \emptyset]$
 Dimensions $[a, b] = [7\emptyset, 46]$

PAINTS LC, $[\emptyset, \emptyset]$, $[7\emptyset, 46]$,Ø

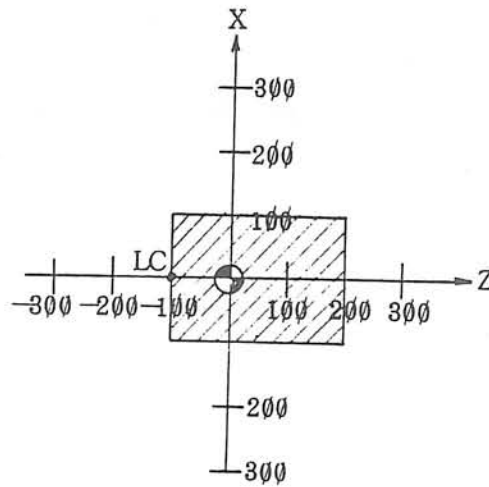
Therefore, the painting is programmed as indicated below:

```
DEF WORK
PAINTS LC, $[\emptyset, \emptyset]$ , $[7\emptyset, 96]$ ,4
PAINTS LC, $[\emptyset, \emptyset]$ , $[7\emptyset, 46]$ ,Ø
END
DRAW
```

Using simple commands, the same painting is programmed as indicated below:

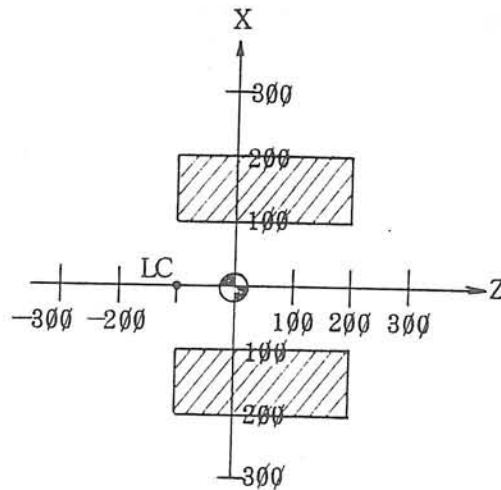
```
DEF WORK
PS LC, $[\emptyset, \emptyset]$ , $[7\emptyset, 96]$ 
PS LC, $[\emptyset, \emptyset]$ , $[7\emptyset, 46]$ ,Ø
END
DRAW
```

Example Program 1



```
Reference Point ..... LC
DEF WORK
OS KCM [-100, 0], [300, 200]
END
```

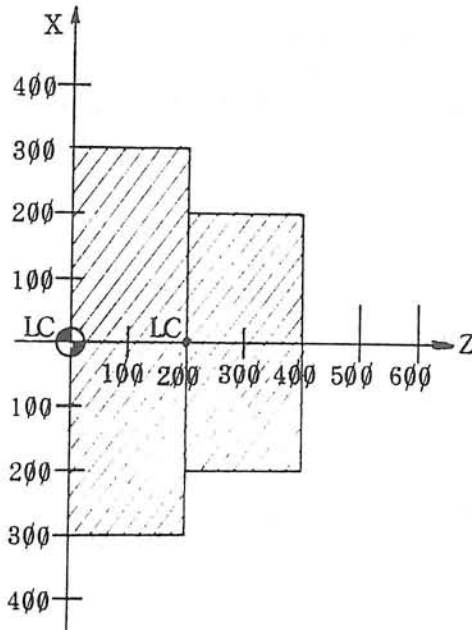
Example Program 2



```
Reference Point ..... LC
DEF WORK
PS LC, [-100, 0], [300, 400]
PS LC, [-100, 0], [300, 200],
    0, (*1)
END
```

Note: 0 (*1) indicates drawing not to be done.

Example Program 3



```

Reference Point ..... LC
DEF WORK
PS LC, [ 0, 0], [200, 600]
PS LC, [200, 0], [200, 400]
END

```

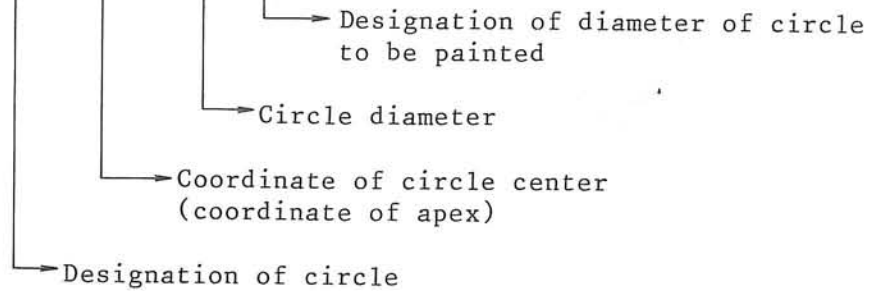
Tile patterns in the examples 1, 2 and 3 are painting of all dots.

b) Blank material viewed from front (only for multiple machining specification)

Description of Commands:

DEF WORK Definition statement - Front view of blank

PF \emptyset . [\emptyset , \emptyset], $1\emptyset\emptyset$, D Command to draw blank



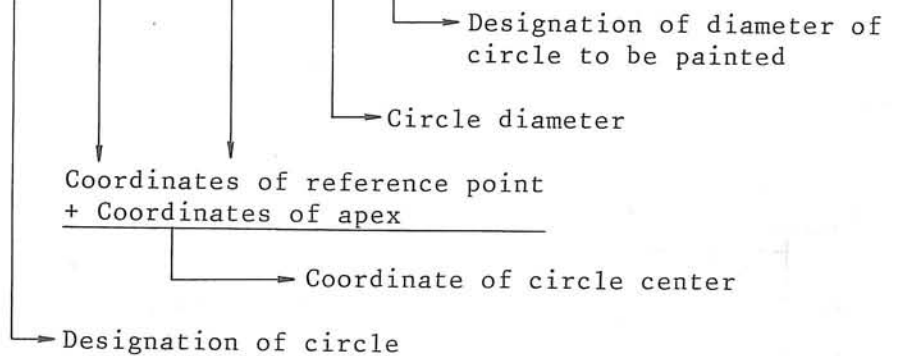
END End statement



or

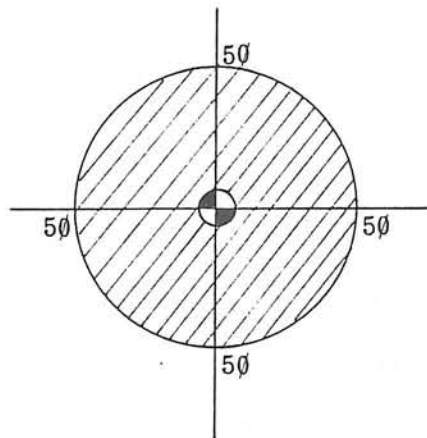
DEF WORKF

PI \emptyset , [\emptyset , \emptyset], [\emptyset , \emptyset], $1\emptyset\emptyset$, D

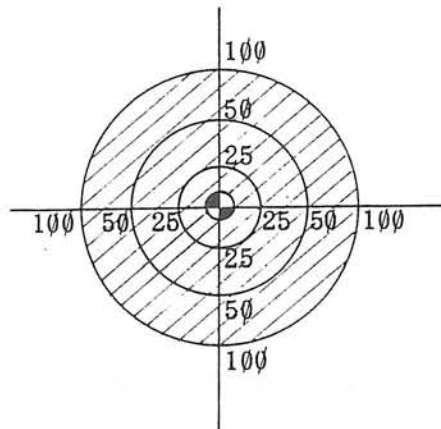


END

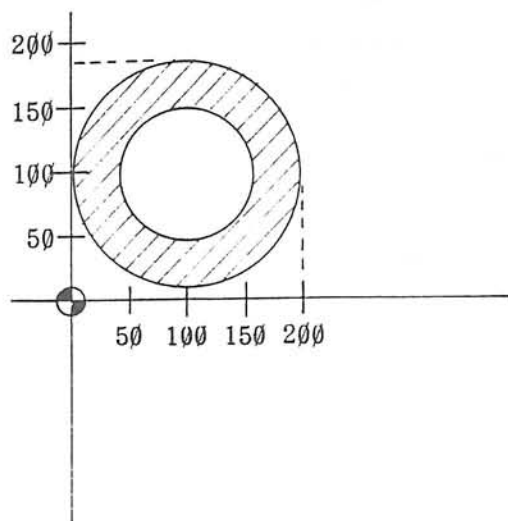
Example Program



```
DEF WORKF
PF Ø, [Ø, Ø], 1ØØ, D
END
```



```
DEF WORKF
PF Ø, [Ø, Ø], 2ØØ, D
PF Ø, [Ø, Ø], 1ØØ, D
PF Ø, [Ø, Ø], 5Ø, D
END
```



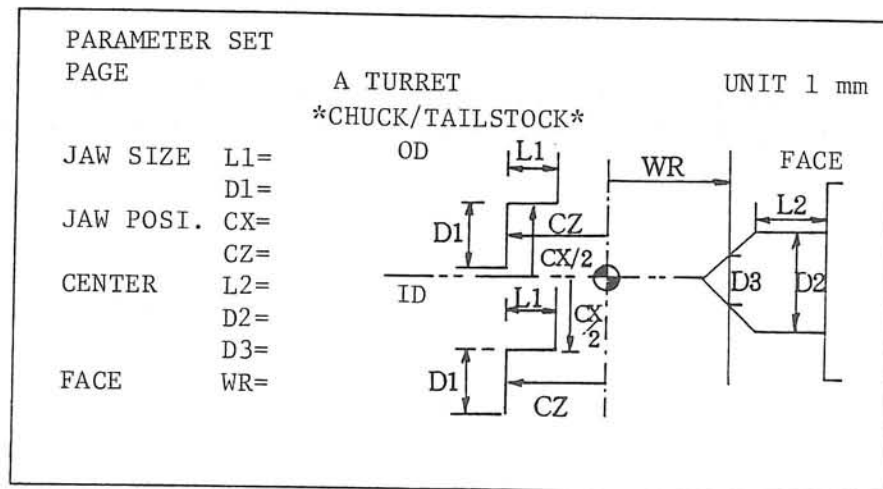
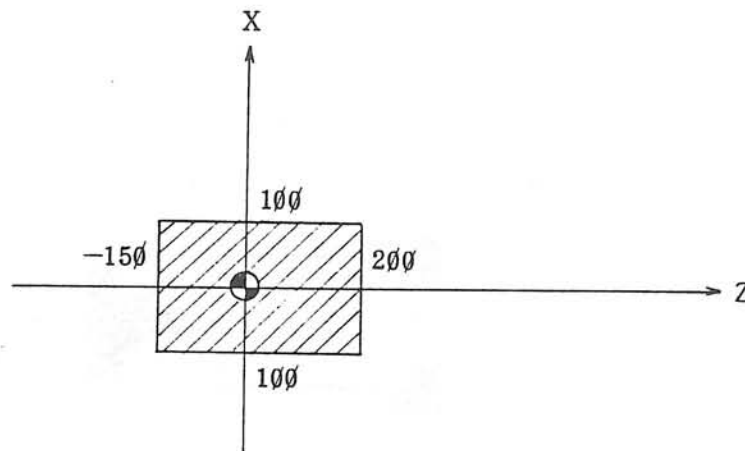
```
DEF WORKF
PI Ø, [1ØØ, 1ØØ], [Ø, Ø], 2ØØ, D
[1ØØ, 1ØØ], [Ø, Ø], 1ØØ, D, Ø
(*1)
END
```

Note: Ø (*1) indicates drawing not to be done.

24-1Ø. HOW TO SET DRAWING POSITIONS OF CHUCK AND TAILSTOCK

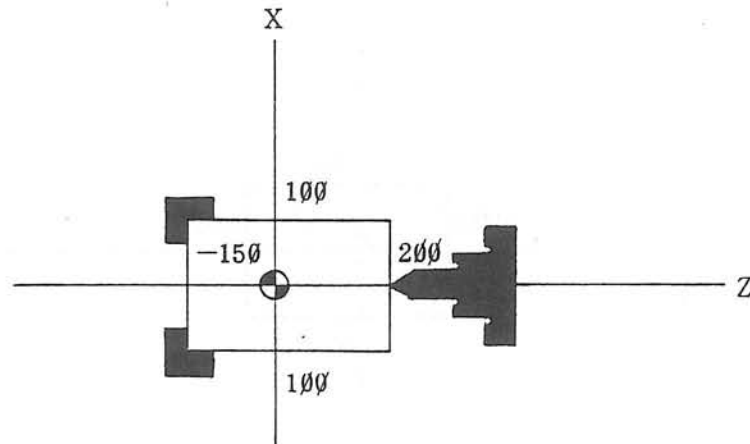
To set the chuck and tailstock positions on the CRT screen with respect to the workpiece material, input proper data at JAW SIZE (L1, D1), JAW POSI. (CX, CZ), CENTER (L2, D2), CENTER (D3) and FACE (WR) on the *CHUCK/TAILSTOCK BARRIER* page in the PARAMETER SET mode.

Example:



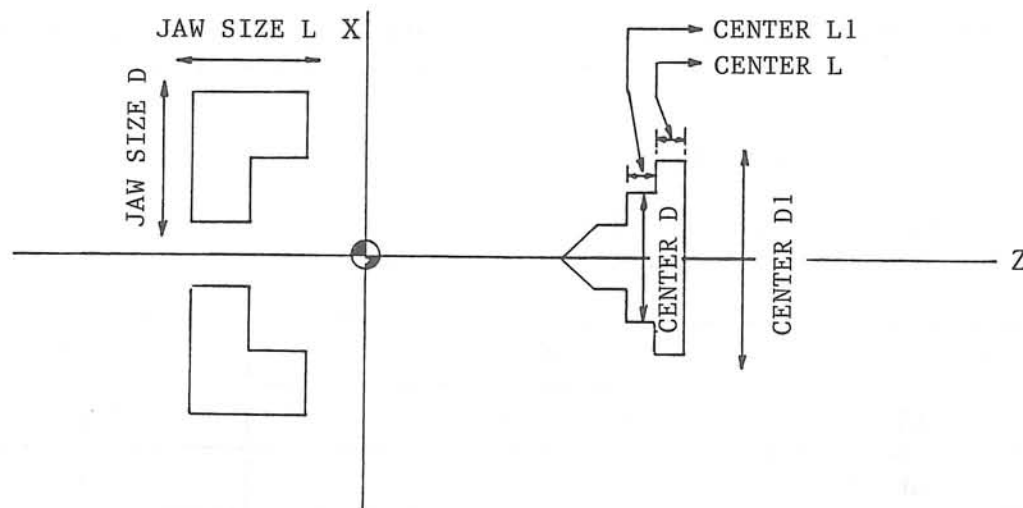
When the following data is input at the *CHUCK/TAILSTOCK BARRIER* page in reference to the blank material illustrated above, the chuck and tailstock are drawn as below.

L1 = 50
D1 = 50
CX = 200
CZ = -150
L2 = 50
D2 = 50
D3 = 20
WR = 200



Note: If the blank material shape is not symmetric in reference to Z-axis, the chuck and the tailstock cannot hold the material correctly.

Changing Chuck and Tailstock Dimensions



Chuck and tailstock dimensions can be changed by setting required values at parameters No. 18 through No. 23 of optional parameter (long word) which can be displayed in the parameter set mode.

- No. 18 JAW SIZE L
- No. 19 JAW SIZE D
- No. 20 CENTER L
- No. 21 CENTER D
- No. 22 CENTER L1
- No. 23 CENTER D1

24-11. SYSTEM VARIABLES FOR ANIMATION DISPLAY

The tool shape data, chuck barrier data and tailstock barrier data can be commanded in the same manner as conventional part program using the system variables.

(1) Tool Shape Definition

The seven system variables indicated below are used for defining the tool shape.

Variable Name	Contents	Data Size
VTLIN[I]	Tool classification code	1 to 26
VTLFN[I]	Tool shape code	1 to 4
VTLA1[I]	Tool angle	0 to 360.000
VTLA2[I]	Edge angle	-360.000 to 360.000
VTLL[I]	Holder length/Sticking out length/Drill length	0 to 9999.999
VTLD[I]	Holder diameter/Drill diameter	0 to 9999.999
VTLW[I]	Tool width	0 to 9999.999

The index [I] of the system variable is designated by the tool number.

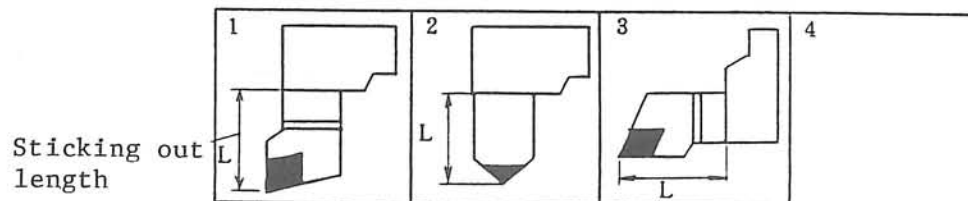
The tool number indicates the station number of the turret.

For the tool life management specification, the system variable name is assigned in the format VGR**[I]. The index [I] is the tool group number.

Tool Classification Code:

1	=	ROUGH OD ←
2	=	ROUGH ID ←
3	=	ROUGH FACE ↓
4	=	ROUGH OD →
5	=	ROUGH ID →
6	=	ROUGH FACE ↑
7	=	FINISH OD ←
8	=	FINISH ID ←
9	=	FINISH FACE ↓
10	=	FINISH OD →
11	=	FINISH ID →
12	=	FINISH FACE ↑
13	=	THREAD OD ←
14	=	THREAD ID ←
15	=	THREAD FACE ↓
16	=	THREAD OD →
17	=	THREAD ID →
18	=	THREAD FACE ↑
19	=	GROOVE OD ↓
20	=	GROOVE ID ↑
21	=	GROOVE FACE ←
22	=	DRILL HSS
23	=	DRILL CARBIDE
24	=	DRILL CENTER
25	=	RECESS OD ↙
26	=	RECESS ID ↘

Tool Shape Code: 1 Tool shape number which corresponds to the
2 tool shape usable for the tool selected by
3 the tool classification code.
4



(ROUGH OD ←, FINISH OD ←)

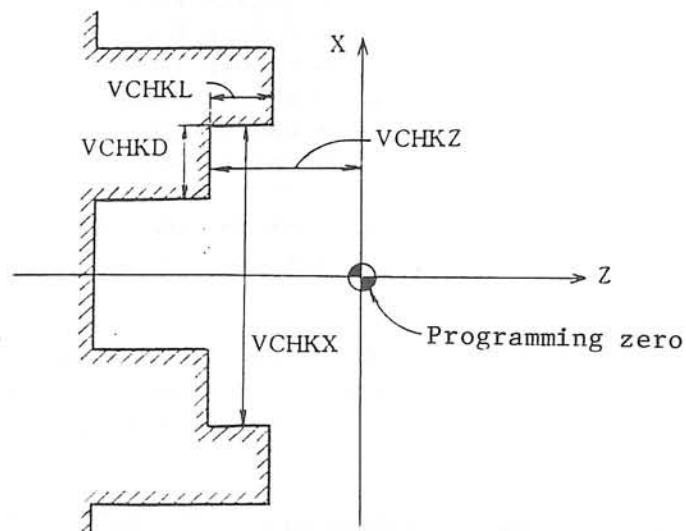
Example: ROUGH OD
TOOL NO. TN=2
TOOL ANGLE A1=80°
EDGE ANGLE A2= 3°
STICKING OUT L=40 mm

VTLIN[2] = 1
VTLFN[2] = 1
VTLA1[2] = 80.000
VTLA2[2] = 3.000
VTLL[2] = 40.000

(2) Chuck Barrier Definition

The four system variables indicated below are used for defining the chuck barrier.

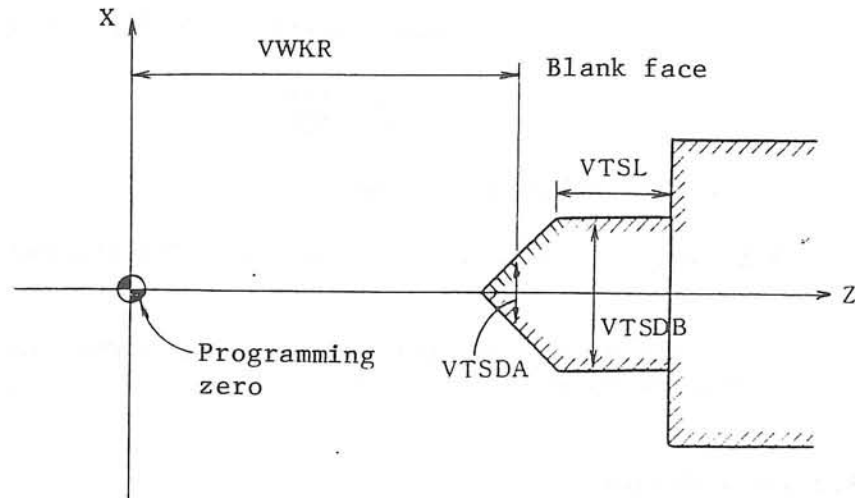
Variable Name	Contents	Data Size
VCHKL	Chuck jaw length	\emptyset to 9999.999
VCHKD	Step on chuck jaw	\emptyset to 9999.999
VCHKX	Gripping diameter	-9999.999 to 9999.999
VCHKZ	Distance from programming zero	-9999.999 to 9999.999



(3) Tailstock Barrier Definition

The four system variables indicated below are used for defining the tailstock barrier.

Variable Name	Contents	Data Size
VWKR	Blank face position	-9999.999 to 9999.999
VTSL	Sticking out length of tailstock spindle	\emptyset to 9999.999
VTSDA	Center hole diameter	\emptyset to 9999.999
VTSDB	Tailstock spindle diameter	\emptyset to 9999.999



24-12. BLANK MATERIAL DRAWING METHOD

To draw blank material, designate the blank material drawing commands and display the shape (square, rectangle) of the material by pressing the material shape key.

(1) Blank Material Viewed from Side

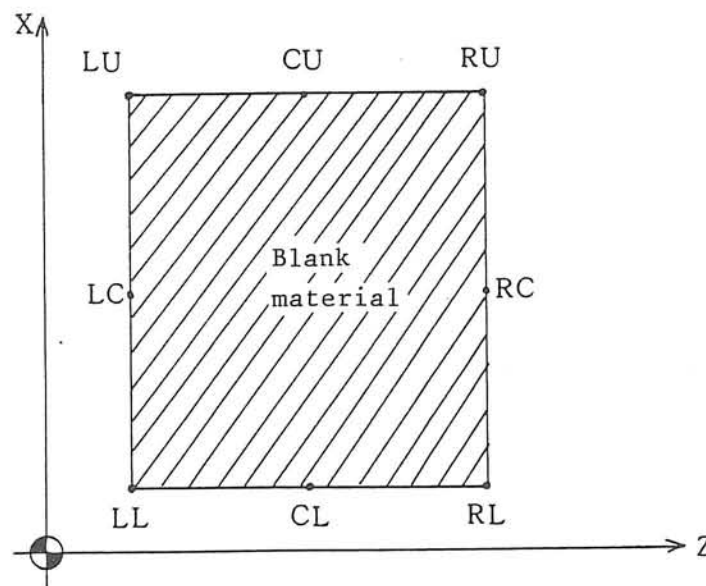
Description of Commands:

```

DEF WORK ..... Definition statement
PS LU, [100, -100], [200, 300] ..... Command to draw blank
  |
  |-----Blank material size
  |           Width  200
  |           Height 300
  |
  |-----Coordinates of reference point
  |           Z  100
  |           X -100
  |
  |-----Reference point
END ..... End statement
  
```

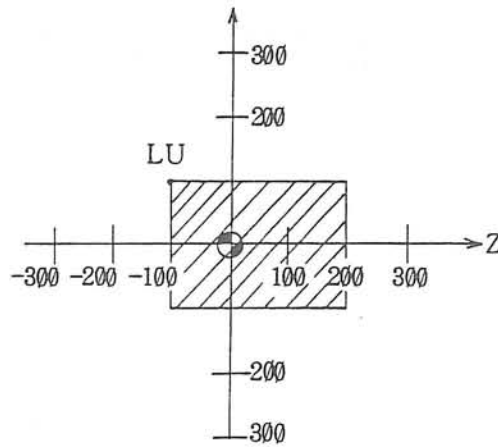
Note: To draw blank material, at least the above indicated three statements must be written.

Reference Points:



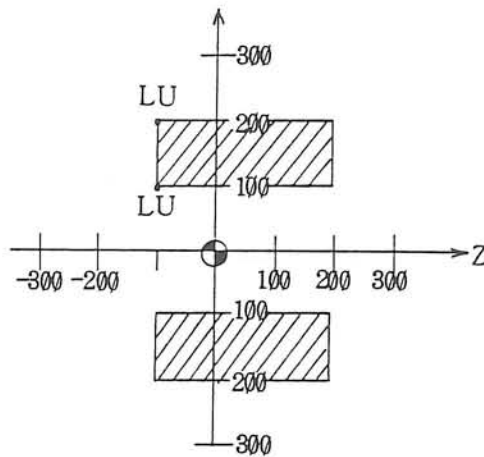
Minimum Setting Unit: 1 mm (0.1 in.)

Example Program 1



```
Reference Point ..... LU
DEF WORK
PS LU, [-100, 100], [300, 200]
END
```

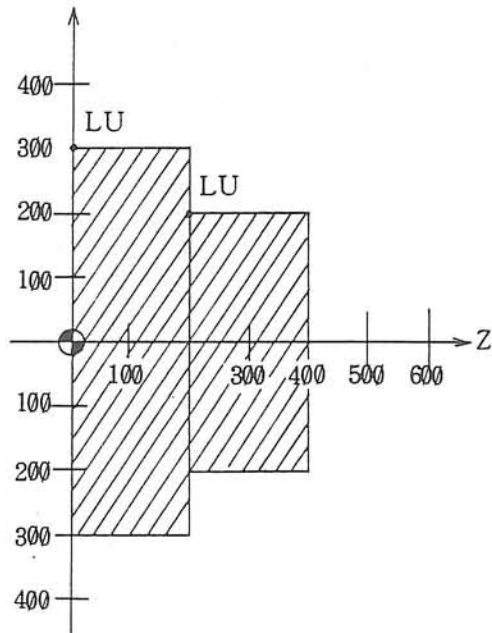
Example Program 2



```
Reference Point ..... LU
DEF WORK
PS LU, [-100, 200], [300, 400]
PS LU, [-100, 100], [300, 200],
      0, (*1)
END
```

Note: 0 (*1) indicates drawing not to be done.

Example Program 3



```

Reference Point ..... LU
DEF WORK
PS LU, [ 0, 300], [200, 600]
PS LU, [200, 200], [200, 400]
END

```

(2) Blank material viewed from front (only for multiple machining specification)

Description of Commands:

DEF WORK Definition statement - Front view of blank

PF \emptyset , [\emptyset , \emptyset], $1\emptyset\emptyset$, D Command to draw blank

Designation of diameter of circle to be painted

Circle diameter

Coordinate of circle center (coordinate of apex)

Designation of circle

END End statement



or

DEF WORKF

PI \emptyset , [\emptyset , \emptyset], [\emptyset , \emptyset], $1\emptyset\emptyset$, D

Designation of diameter of circle to be painted

Circle diameter

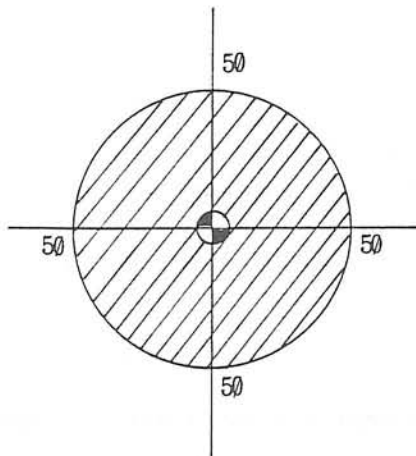
Coordinates of reference point + Coordinates of apex

Coordinate of circle center

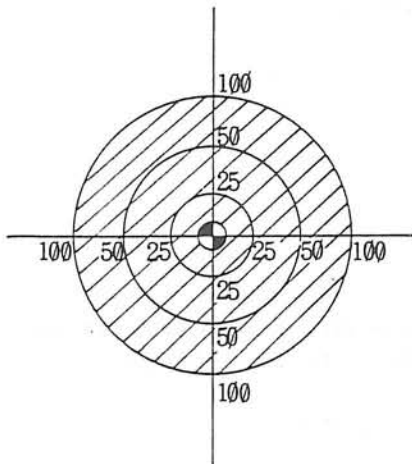
Designation of circle

END

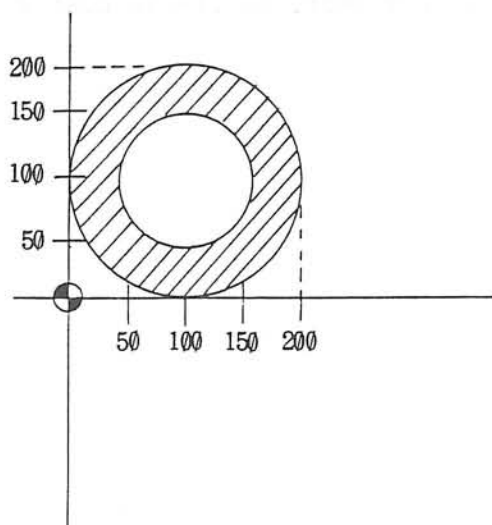
Example Program



```
DEF WORKF
PF Ø, [Ø, Ø], 1ØØ, D
END
```



```
DEF WORKF
PF Ø, [Ø, Ø], 2ØØ, D
PF Ø, [Ø, Ø], 1ØØ, D
PF Ø, [Ø, Ø], 5Ø, D
END
```



```
DEF WORKF
PI Ø, [1ØØ, 1ØØ], [Ø, Ø], 2ØØ, D
PI Ø, [1ØØ, 1ØØ], [Ø, Ø], 1ØØ, D, Ø
(*1)
```

```
END
```

Note: Ø (*1) indicates drawing not to be done.

118 ALARM-A Chuck condition illegal

Chuck open or close command (M83 or M84) is designated while the spindle or the milling tool spindle is rotating. Or while the external interlock signal is active.

This alarm does not take place for SMW chuck, or while the internal interlock is deactivated.

Index	:	TURRET
Character-string:	:	None
Code	:	1 No spindle zero speed signal (Bit 3 of EC input #9 is \emptyset .)
		2 The spindle is not at a still. (5 rpm or higher)
		3 Spindle rotation command is ON.
		4 External interlock signal is active. (Bit 7 of EC input #9 is \emptyset .)
		5 No milling tool spindle zero speed signal (Bit 4 of EC input #22 is \emptyset)
		6 Milling tool spindle is not at a still.
		7 Milling tool spindle rotation command is ON.

119 ALARM-A Chuck open

A spindle rotation command or a milling tool spindle rotation command is designated while the chuck is not closed, or the chuck close signal is turned off during the rotation of the spindle or the milling tool spindle.

This alarm does not take place while the internal interlock is deactivated.

Index	:	TURRET or None
Character-string:	:	None
Code	:	1 Spindle rotation command is designated.
		2 Chuck close signal turned off during spindle rotation.
		3 Milling tool spindle rotation command is designated.
		4 Chuck close signal turned off during Milling tool spindle rotation.

SECTION 25 CHUCK AUTO OPEN/CLOSE BY M CODES

25-1. M CODES USED FOR OPENING/CLOSING CHUCK

M83 Chuck close
 M84 Chuck open

Selected mode is not influenced by turning on the control or by resetting it.

For the two-saddle models, the M codes above may be designated at either G13 or G14 mode.

25-2. ALARM MESSAGES

117 ALARM-A Chuck clamp

During a chuck open/close operation, ID/OD gripping was changed. Or low chucking pressure state has continued for more than two seconds.

Index	:	None
Character-string:	:	None
Code	:	100 Low chucking pressure state has continued for more than two seconds (Bit 6 of EC input #1 is 0.)
	:	None Both panel OD gripping signal (Bit 0 of panel input #6) and panel ID gripping signal (Bit 1 of panel input #6) are OFF.
	:	3 Both panel OD gripping signal and panel ID gripping signal are ON.