

PROGRAMMING MANUAL

for

MAZATROL FUSION 640M

(for ANGULAX)

Programming EIA/ISO

MANUAL No. : H740PB0010E

Serial No. :

Before using this machine and equipment, fully understand the contents of this manual to ensure proper operation. Should any questions arise, please ask the nearest Technical Center or Technology Center.

IMPORTANT NOTICE

1. Be sure to observe the safety precautions described in this manual and the contents of the safety plates on the machine and equipment. Failure may cause serious personal injury or material damage. Please replace any missing safety plates as soon as possible.
2. No modifications are to be performed that will affect operation safety. If such modifications are required, please contact the nearest Technical Center or Technology Center.
3. For the purpose of explaining the operation of the machine and equipment, some illustrations may not include safety features such as covers, doors, etc. Before operation, make sure all such items are in place.
4. This manual was considered complete and accurate at the time of publication, however, due to our desire to constantly improve the quality and specification of all our products, it is subject to change or modification. If you have any questions, please contact the nearest Technical Center or Technology Center.
5. Always keep this manual near the machinery for immediate use.
6. If a new manual is required, please order from the nearest Technical Center or Technology Center with the manual No. or the machine name, serial No. and manual name.

Issued by *Manual Publication Section, Yamazaki Mazak Corporation, Japan*

SAFETY PRECAUTIONS

Preface

Safety precautions relating to the CNC unit (in the remainder of this manual, referred to simply as the NC unit) that is provided in this machine are explained below. Not only the persons who create programs, but also those who operate the machine must thoroughly understand the contents of this manual to ensure safe operation of the machine.

Read all these safety precautions, even if your NC model does not have the corresponding functions or optional units and a part of the precautions do not apply.

Rule

1. This section contains the precautions to be observed as to the working methods and states usually expected. Of course, however, unexpected operations and/or unexpected working states may take place at the user site.
During daily operation of the machine, therefore, the user must pay extra careful attention to its own working safety as well as to observe the precautions described below.
2. Although this manual contains as great an amount of information as it can, since it is not rare for the user to perform the operations that overstep the manufacturer-assumed ones, not all of “what the user cannot perform” or “what the user must not perform” can be fully covered in this manual with all such operations taken into consideration beforehand.
It is to be understood, therefore, that functions not clearly written as “executable” are “inexecutable” functions.
3. The meanings of our safety precautions to DANGER, WARNING, and CAUTION are as follows:



DANGER

: Failure to follow these instructions could result in loss of life.



WARNING

: Failure to observe these instructions could result in serious harm to a human life or body.



CAUTION

: Failure to observe these instructions could result in minor injuries or serious machine damage.

Basics



- After turning power on, keep hands away from the keys, buttons, or switches of the operating panel until an initial display has been made.
- Before proceeding to the next operations, fully check that correct data has been entered and/or set. If the operator performs operations without being aware of data errors, unexpected operation of the machine will result.
- Before machining workpieces, perform operational tests and make sure that the machine operates correctly. No workpieces must be machined without confirmation of normal operation. Closely check the accuracy of programs by executing override, single-block, and other functions or by operating the machine at no load. Also, fully utilize tool path check, solid check, and other functions, if provided.
- Make sure that the appropriate feed rate and rotational speed are designated for the particular machining requirements. Always understand that since the maximum usable feed rate and rotational speed are determined by the specifications of the tool to be used, those of the workpiece to be machined, and various other factors, actual capabilities differ from the machine specifications listed in this manual. If an inappropriate feed rate or rotational speed is designated, the workpiece or the tool may abruptly move out from the machine.
- Before executing correction functions, fully check that the direction and amount of correction are correct. Unexpected operation of the machine will result if a correction function is executed without its thorough understanding.
- Parameters are set to the optimum standard machining conditions prior to shipping of the machine from the factory. In principle, these settings should not be modified. If it becomes absolutely necessary to modify the settings, perform modifications only after thoroughly understanding the functions of the corresponding parameters. Modifications usually affect any program. Unexpected operation of the machine will result if the settings are modified without a thorough understanding.

Remarks on the cutting conditions recommended by the NC



- Before using the following cutting conditions:
 - Cutting conditions that are the result of the MAZATROL Automatic Cutting Conditions Determination Function
 - Cutting conditions suggested by the Machining Navigation Function
 - Cutting conditions for tools that are suggested to be used by the Machining Navigation Function
- Confirm that every necessary precaution in regards to safe machine setup has been taken – especially for workpiece fixturing/clamping and tool setup.
- Confirm that the machine door is securely closed before starting machining.
Failure to confirm safe machine setup may result in serious injury or death.

Programming



- Fully check that the settings of the coordinate systems are correct. Even if the designated program data is correct, errors in the system settings may cause the machine to operate in unexpected places and the workpiece to abruptly move out from the machine in the event of contact with the tool.
- During surface velocity hold control, as the current workpiece coordinates of the surface velocity hold control axes approach zeroes, the spindle speed increases significantly. For the lathe, the workpiece may even come off if the chucking force decreases. Safety speed limits must therefore be observed when designating spindle speeds.
- Even after inch/metric system selection, the units of the programs, tool information, or parameters that have been registered until that time are not converted. Fully check these data units before operating the machine. If the machine is operated without checks being performed, even existing correct programs may cause the machine to operate differently from the way it did before.
- If a program is executed that includes the absolute data commands and relative data commands taken in the reverse of their original meaning, totally unexpected operation of the machine will result. Recheck the command scheme before executing programs.
- If an incorrect plane selection command is issued for a machine action such as arc interpolation or fixed-cycle machining, the tool may collide with the workpiece or part of the machine since the motions of the control axes assumed and those of actual ones will be interchanged. (This precaution applies only to NC units provided with EIA functions.)
- The mirror image, if made valid, changes subsequent machine actions significantly. Use the mirror image function only after thoroughly understanding the above. (This precaution applies only to NC units provided with EIA functions.)
- If machine coordinate system commands or reference position returning commands are issued with a correction function remaining made valid, correction may become invalid temporarily. If this is not thoroughly understood, the machine may appear as if it would operate against the expectations of the operator. Execute the above commands only after making the corresponding correction function invalid. (This precaution applies only to NC units provided with EIA functions.)
- The barrier function performs interference checks based on designated tool data. Enter the tool information that matches the tools to be actually used. Otherwise, the barrier function will not work correctly.
- The system of G-code and M-code commands differs, especially for turning, between the machines of INTEGREX e-Series and the other turning machines.
Issuance of the wrong G-code or M-code command results in totally non-intended machine operation. Thoroughly understand the system of G-code and M-code commands before using this system.

Sample program	Machines of INTEGREX e-Series	Turning machines
S1000M3	The milling spindle rotates at 1000 min ⁻¹ .	The turning spindle rotates at 1000 min ⁻¹ .
S1000M203	The turning spindle rotates at 1000 min ⁻¹ .	The milling spindle rotates at 1000 min ⁻¹ .

- For the machines of INTEGREX e-Series, programmed coordinates can be rotated using an index unit of the MAZATROL program and a G68 command (coordinate rotate command) of the EIA program. However, for example, when the B-axis is rotated through 180 degrees around the Y-axis to implement machining with the turning spindle No. 2, the plus side of the X-axis in the programmed coordinate system faces downward and if the program is created ignoring this fact, the resulting movement of the tool to unexpected positions may incite collisions.

To create the program with the plus side of the X-axis oriented in an upward direction, use the mirror function of the WPC shift unit or the mirror imaging function of G-code command (G50.1, G51.1).

- After modifying the tool data specified in the program, be sure to perform the tool path check function, the solid check function, and other functions, and confirm that the program operates properly. The modification of tool data may cause even a field-proven machining program to change in operational status.

If the user operates the machine without being aware of any changes in program status, interference with the workpiece could arise from unexpected operation.

For example, if the cutting edge of the tool during the start of automatic operation is present inside the clearance-including blank (unmachined workpiece) specified in the common unit of the MAZATROL program, care is required since the tool will directly move from that position to the approach point because of no obstructions being judged to be present on this path.

For this reason, before starting automatic operation, make sure that the cutting edge of the tool during the start of automatic operation is present outside the clearance-including workpiece specified in the common unit of the MAZATROL program.



- If axis-by-axis independent positioning is selected and simultaneously rapid feed selected for each axis, movements to the ending point will not usually become linear. Before using these functions, therefore, make sure that no obstructions are present on the path.

Operations



- Single-block, feed hold, and override functions can be made invalid using system variables #3003 and #3004. Execution of this means the important modification that makes the corresponding operations invalid. Before using these variables, therefore, give thorough notification to related persons. Also, the operator must check the settings of the system variables before starting the above operations.
- If manual intervention during automatic operation, machine locking, the mirror image function, or other functions are executed, the workpiece coordinate systems will usually be shifted. When making machine restart after manual intervention, machine locking, the mirror image function, or other functions, consider the resulting amounts of shift and take the appropriate measures. If operation is restarted without any appropriate measures being taken, collision with the tool or workpiece may occur.
- Use the dry run function to check the machine for normal operation at no load. Since the feed rate at this time becomes a dry run rate different from the program-designated feed rate, the axes may move at a feed rate higher than the programmed value.
- After operation has been stopped temporarily and insertion, deletion, updating, or other commands executed for the active program, unexpected operation of the machine may result if that program is restarted. No such commands should, in principle, be issued for the active program.



- During manual operation, fully check the directions and speeds of axial movement.
- For a machine that requires manual homing, perform manual homing operations after turning power on. Since the software-controlled stroke limits will remain ineffective until manual homing is completed, the machine will not stop even if it oversteps the limit area. As a result, serious machine damage will result.
- Do not designate an incorrect pulse multiplier when performing manual pulse handle feed operations. If the multiplier is set to 1000 times and the handle operated inadvertently, axial movement will become faster than that expected.

OPERATIONAL WARRANTY FOR THE NC UNIT

The warranty of the manufacturer does not cover any trouble arising if the NC unit is used for its non-intended purpose. Take notice of this when operating the unit.

Examples of the trouble arising if the NC unit is used for its non-intended purpose are listed below.

1. Trouble associated with and caused by the use of any commercially available software products (including user-created ones)
2. Trouble associated with and caused by the use of any Windows operating systems
3. Trouble associated with and caused by the use of any commercially available computer equipment

Operating Environment

1. Ambient temperature

During machine operation: 0° to 50°C (0° to 122°F)

2. Relative humidity

During machine operation: 10 to 75% (without bedewing)

Note: As humidity increases, insulation deteriorates causing electrical component parts to deteriorate quickly.

CONTENTS

Page

1	CONTROL AXES	1-1
1-1	Coordinate Words and Control Axes	1-1
2	UNITS OF PROGRAM DATA INPUT	2-1
2-1	Units of Program Data Input	2-1
2-2	Units of Data Setting.....	2-1
2-3	Ten-Fold Program Data.....	2-1
3	DATA FORMATS	3-1
3-1	Tape Codes.....	3-1
3-2	Program Formats	3-5
3-3	Tape Data Storage Format.....	3-6
3-4	Optional Block Skip	3-6
3-5	Program Number, Sequence Number and Block Number: O, N	3-7
3-6	Parity-H/V	3-8
3-7	List of G-Codes	3-10
3-8	List of M-Codes	3-11
4	BUFFER REGISTERS	4-1
4-1	Input Buffer.....	4-1
4-2	Pre-Read Buffer	4-2
5	POSITION PROGRAMMING.....	5-1
5-1	Dimensional Data Input Method: G90, G91	5-1

5-2	Inch/Metric Selection: G20/G21	5-3
5-3	Decimal Point Input	5-4
6	INTERPOLATION FUNCTIONS	6-1
6-1	Positioning (Rapid Feed): G00	6-1
6-2	One-Way Positioning: G60	6-3
6-3	Linear Interpolation: G01	6-4
6-4	Plane Selection: G17, G18, G19	6-5
6-5	Circular Interpolation: G02, G03	6-7
6-6	Radius-Designated Circular Interpolation: G02, G03	6-10
6-7	Helical Interpolation: G17, G18, G19 and G02, G03	6-11
6-8	Spiral Interpolation: G2.1, G3.1 (Option)	6-14
6-9	Virtual-Axis Interpolation: G07	6-22
6-10	Spline Interpolation: G06.1 (Option)	6-23
6-11	NURBS Interpolation: G06.2 (Option)	6-34
6-12	Cylindrical Interpolation (Option)	6-41
7	FEED FUNCTIONS	7-1
7-1	Rapid Feed Rates	7-1
7-2	Cutting Feed Rates	7-1
7-3	Synchronous/Asynchronous Feed: G95/G94	7-2
7-4	Selecting a Feed Rate, and Effects on Each Control Axis	7-3
7-5	Exact-Stop Check: G09	7-6
7-6	Exact-Stop Check Mode: G61	7-8

7-7	Automatic Corner Override: G62	7-8
7-8	Tapping Mode: G63.....	7-12
7-9	Cutting Mode: G64	7-13
7-10	Inverse Time Feed: G93 (Option)	7-13
8	DWELL FUNCTIONS	8-1
8-1	Dwell Command in Time: (G94) G04.....	8-1
8-2	Dwell Command in Number of Revolutions: (G95) G04	8-2
9	MISCELLANEOUS FUNCTIONS	9-1
9-1	Miscellaneous Functions (M3-Digit).....	9-1
9-2	No. 2 Miscellaneous Functions (B3-Digit).....	9-2
10	SPINDLE FUNCTION.....	10-1
11	TOOL FUNCTIONS.....	11-1
11-1	Tool Function (T3-Digit).....	11-1
11-2	Tool Function (T8-Digit).....	11-1
12	TOOL OFFSET FUNCTIONS.....	12-1
12-1	Tool Offset.....	12-1
12-2	Tool Length Offset/Cancellation: G43, G44/G49	12-5
12-3	Tool Position Offset: G45 to G48.....	12-8
12-4	Tool Diameter Offset Function: G40, G41, G42	12-14
12-4-1	Overview	12-14
12-4-2	Tool diameter offsetting	12-14
12-4-3	Tool diameter offsetting operation using other commands	12-23

12-4-4	Corner movement	12-30
12-4-5	Interruptions during tool diameter offsetting.....	12-30
12-4-6	General precautions on tool diameter offsetting	12-32
12-4-7	Offset number updating during the offset mode.....	12-33
12-4-8	Excessive cutting due to tool diameter offsetting	12-35
12-4-9	Interference check.....	12-37
12-5	Three-Dimensional Tool Diameter Offsetting (Option).....	12-44
12-5-1	Function description	12-44
12-5-2	Programming methods	12-45
12-5-3	Correlationships to other functions	12-49
12-5-4	Miscellaneous notes on three-dimensional tool diameter offsetting	12-49
12-6	Programmed Input of Offset Data: G10	12-50
12-7	Tool Offsetting Based on MAZATROL Tool Data	12-58
12-7-1	Selecting parameters	12-58
12-7-2	Tool length offsetting	12-58
12-7-3	Tool diameter offsetting	12-59
12-7-4	Tool data update (during automatic operation)	12-60
12-8	Shaping Function (Option).....	12-61
12-8-1	Overview	12-61
12-8-2	Programming format	12-62
12-8-3	Detailed description.....	12-62
12-8-4	Remarks	12-69
12-8-5	Compatibility with the other functions	12-70
12-8-6	Sample program.....	12-71

13	AUXILIARY FUNCTIONS FOR PROGRAMMING	13-1
13-1	Fixed-Cycle Functions	13-1
13-1-1	Function description	13-1
13-1-2	List of fixed-cycle functions	13-1
13-1-3	Fixed-cycle machining data format	13-2
13-1-4	G71.1 (Chamfering cutter CW)	13-5
13-1-5	G72.1 (Chamfering cutter CCW)	13-6
13-1-6	G73 (High-speed deep-hole drilling)	13-7
13-1-7	G74 (Reverse tapping)	13-8
13-1-8	G75 (Boring)	13-9
13-1-9	G76 (Boring)	13-10
13-1-10	G77 (Back spot facing)	13-11
13-1-11	G78 (Boring)	13-12
13-1-12	G79 (Boring)	13-13
13-1-13	G81 (Spot drilling)	13-13
13-1-14	G82 (Drilling)	13-14
13-1-15	G83 (Deep-hole drilling)	13-15
13-1-16	G84 (Tapping)	13-16
13-1-17	G85 (Reaming)	13-17
13-1-18	G86 (Boring)	13-17
13-1-19	G87 (Back boring)	13-18
13-1-20	G88 (Boring)	13-19
13-1-21	G89 (Boring)	13-19
13-1-22	Synchronous tapping (Option)	13-20
13-1-23	Tornado cycle (Option)	13-24

13-2	Suppression of Single-Block Stop for Fixed Cycles.....	13-26
13-2-1	Function description	13-26
13-2-2	Examples of operation.....	13-26
13-3	Initial Point and R-Point Level Return: G98, G99	13-27
13-4	Workpiece Coordinate Setting during the Fixed-Cycle Mode	13-28
13-5	Subprogram Control Commands: M98, M99	13-28
13-6	Mutual Subprogram Call between EIA/ISO and MAZATROL (Option)	13-33
13-7	Variables Commands	13-36
13-8	Figure Rotation: M98 (Option)	13-38
13-9	Programmed Coordinate Rotation	13-43
13-10	User Macros (Option)	13-45
13-10-1	User macros.....	13-45
13-10-2	Macro call instructions.....	13-46
13-10-3	Variables	13-55
13-10-4	Types of variables	13-57
13-10-5	Arithmetic operation commands	13-74
13-10-6	Control commands	13-79
13-10-7	External output commands (Output via RS-232C).....	13-82
13-10-8	External output command (Output onto the hard disk).....	13-84
13-10-9	Precautions	13-86
13-10-10	Specific examples of programming using user macros	13-88
13-11	Scaling: G50, G51	13-92
13-12	Mirror Image On/Off G-Codes: G50.1, G51.1	13-105
13-13	Linear Angle Commands	13-106

13-14 Geometric Commads.....	13-107
13-15 Corner Chamfering and Corner Rounding Commands.....	13-108
13-15-1 Corner chamfering (, C_).....	13-108
13-15-2 Corner rounding (,R_)	13-110
14 COORDINATE SYSTEM SETTING FUNCTIONS	14-1
14-1 Fundamental Machine Coordinate System, Workpiece Coordinate Systems, and Local Coordinate Systems	14-1
14-2 Machine Zero Point and Second, Third, and Fourth Reference Points	14-2
14-3 Fundamental Machine Coordinate System Selection: G53	14-3
14-4 Coordinate System Setting: G92	14-4
14-5 Automatic Coordinate System Setting	14-5
14-6 Reference Point Return: G28, G29	14-6
14-7 Second, Third, or Fourth Reference Point Return: G30.....	14-8
14-8 Reference Point Check Command: G27	14-10
14-9 Workpiece Coordinate System Setting and Selection: (G92) G54 to G59.....	14-11
14-10 Additional Workpiece Coordinate System Setting and Selection: G54.1 (Option)	14-16
14-11 Local Coordinate System Setting : G52.....	14-22
14-12 Reading/Writing of MAZATROL Program Basic Coordinates.....	14-27
14-12-1 Calling a macroprogram (for data writing).....	14-27
14-12-2 Data reading	14-27
14-12-3 Rewriting	14-28
14-13 Workpiece Coordinate System Rotation.....	14-29

15	PROTECTION FUNCTIONS	15-1
15-1	Before-Movement Stroke Limit Check: G22, G23	15-1
16	SKIP FUNCTION: G31	16-1
16-1	Skip Function.....	16-1
16-2	Skip Coordinate Reading.....	16-2
16-3	Amount of Coasting.....	16-3
16-4	Skip Coordinate Reading Error.....	16-4
16-5	Multi-Step Skip: G31.1, G31.2, G31.3, G04	16-5
17	THREADING: G33 (Option).....	17-1
17-1	Equal-Lead Threading	17-1
17-2	Continuous Threading	17-4
17-3	Inch Threading	17-4
18	AUTOMATIC TOOL LENGTH MEASUREMENT: G37 (Option)	18-1
19	DYNAMIC OFFSETTING: M173, M174 (Option)	19-1
20	HIGH-SPEED MACHINING MODE FEATURE (OPTION).....	20-1
21	FIVE-SURFACE MACHINING FUNCTION (OPTION).....	21-1
21-1	Coordinate Systems for Five-Surface Machining	21-1
21-2	Surface Selection Command.....	21-2
21-3	Head Offsetting	21-3
21-3-1	Specification of head offsetting.....	21-3
21-3-2	Cancellation of head offsetting	21-3

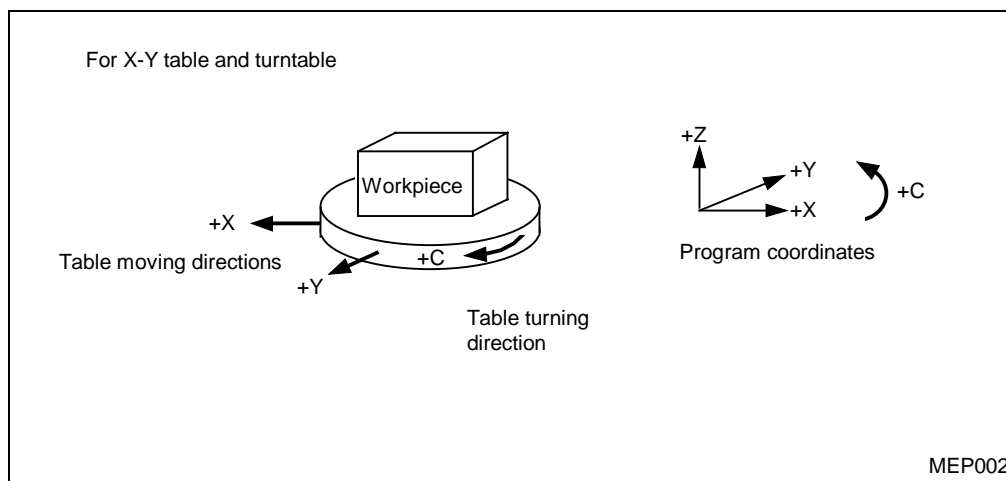
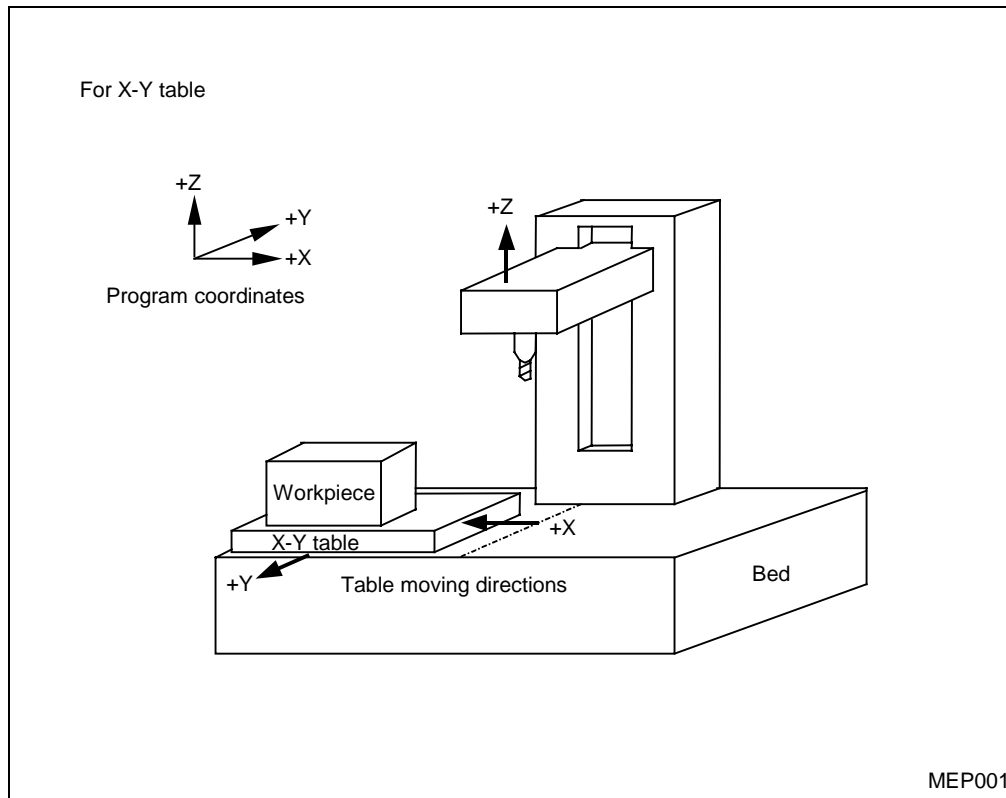
21-4	Relationship to Other Functions	21-4
22	ARBITRARY-SURFACE MACHINING FUNCTION (OPTION)	22-1
22-1	Three-Dimensional Coordinate Conversion: G68.....	22-1
22-2	Simultaneous-Operation M-Codes	22-9
22-3	Table Rotational Machining (Option)	22-10
22-4	Workpiece Coordinate Conversion	22-12
22-5	Inclined-Surface Synchronous Tapping.....	22-14
22-6	Inclined-Surface Boring	22-15
22-7	Sample Program	22-16
23	PROGRAMS EXAMPLES	23-1
24	EIA/ISO PROGRAM DISPLAY	24-1
24-1	Procedures for Constructing an EIA/ISO Program	24-1
24-2	Editing Function of EIA/ISO PROGRAM Display.....	24-2
24-2-1	General	24-2
24-2-2	Operation procedure	24-2
24-3	Macro-Instruction Input.....	24-8

- NOTE -

1 CONTROL AXES

1-1 Coordinate Words and Control Axes

Under standard specifications, there are three-dimensional axes of control. With an added feature and a special option, the machine can control up to a maximum of six axes, including the three fundamental axes. The direction of machining can be designated using a predetermined coordinate word consisting of an alphabetic character.



- NOTE -

2 UNITS OF PROGRAM DATA INPUT

2-1 Units of Program Data Input

The movements on coordinate axes are to be commanded in the MDI mode or machining program. The movement data are expressed in millimeters, inches or degrees.

2-2 Units of Data Setting

Various data commonly used for control axes, such as offsetting data, must be set for the machine to perform an operation as desired.

The units of data setting and those of program data input are listed below.

	Linear axis		Rotational axis
	Metric system	Inch system	
Units of program data input	0.0001 mm	0.00001 in.	0.0001 deg
Units of data setting	0.0001 mm	0.00001 in.	0.0001 deg

Note 1: Inch/metric selection can be freely made using either bit 4 of parameter **F91** ("0" for metric, "1" for inches; validated through power-off and -on) or G-code commands (G20, G21).

Selection using the G-code commands is valid only for program data input.

Variables and offsetting data (such as tool offsetting data) should therefore be set beforehand using the appropriate unit (inch or metric) for the particular machining requirements.

Note 2: Metric data and inch data cannot be used at the same time.

2-3 Ten-Fold Program Data

Using a predetermined parameter, machining program data can be processed as set in units of one micron. There may be cases that a machining program which has been set in units of one micron is to be used with a numerical control unit based on 0.1 micron increments. In such cases, use of this parameter allows the machine to perform the required machining operations without rewriting the program.

Use bit 0 of user parameter **F91** for this purpose.

All types of coordinate data (axis movement data) not provided with the decimal point will be multiplied by a factor of 10. This does not apply, indeed, to preset tool-offsetting data designated with addresses H and D.

Control axis	Program command	Moving distance when program commands are executed			Program applicability (A) → (B)
		NC (A) for which the program was prepared	MAZATROL (B)		
			Bit 0 of F91 = 0	Bit 0 of F91 = 1	
Linear axis	X1 (Y1 / Z1)	1 micron	0.1 micron	1 micron	Applicable
Rotational axis	B1	0.001°	0.0001°	0.001°	Applicable

- NOTE -

3 DATA FORMATS

3-1 Tape Codes

This numerical control unit (in the remainder of this manual, referred to as the NC unit) uses command information that consists of letters of the alphabet (A, B, C ... Z), numerics (0, 1, 2 ... 9), and signs (+, -, /, and so on). These alphanumerics and signs are referred to collectively as characters. On paper tape, these characters are represented as a combination of a maximum of eight punched holes.

Such a representation is referred to as a code.

The NC unit uses either the EIA codes (RS-244-A) or the ISO codes (R-840).

Note 1: Codes not included in the tape codes shown in Fig.3-1 will result in an error when they are read.

Note 2: Of all characters specified as the ISO codes but not specified as the EIA codes, only the following characters can be designated using the parameters **TAP9** to **TAP16**:

- [Bracket Open
-] Bracket Close
- (Parenthesis Open
-) Parenthesis Close
- # Sharp
- * Asterisk
- = Equal sign
- : Colon

However, you cannot designate characters that overlap existing ones or that result in parity error.

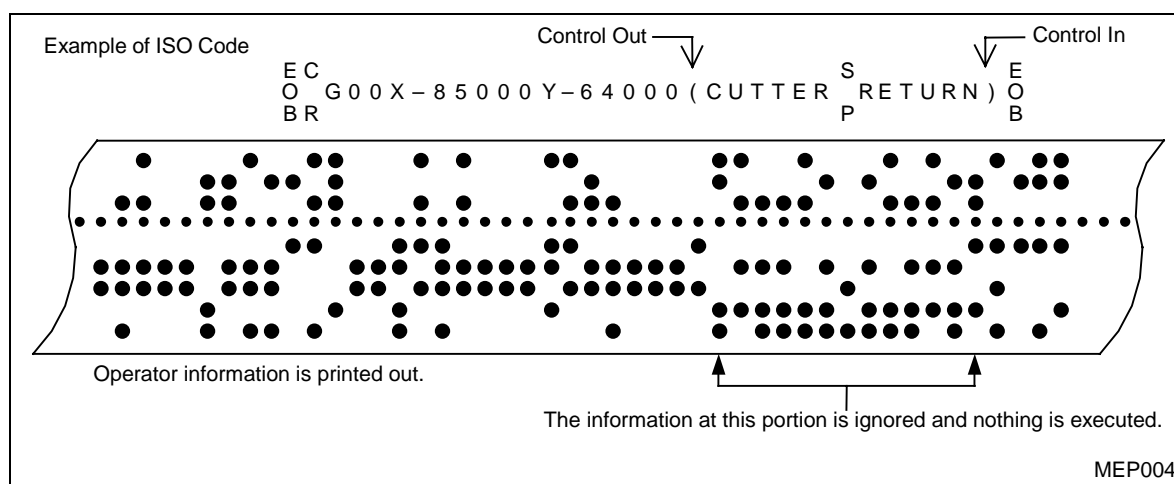
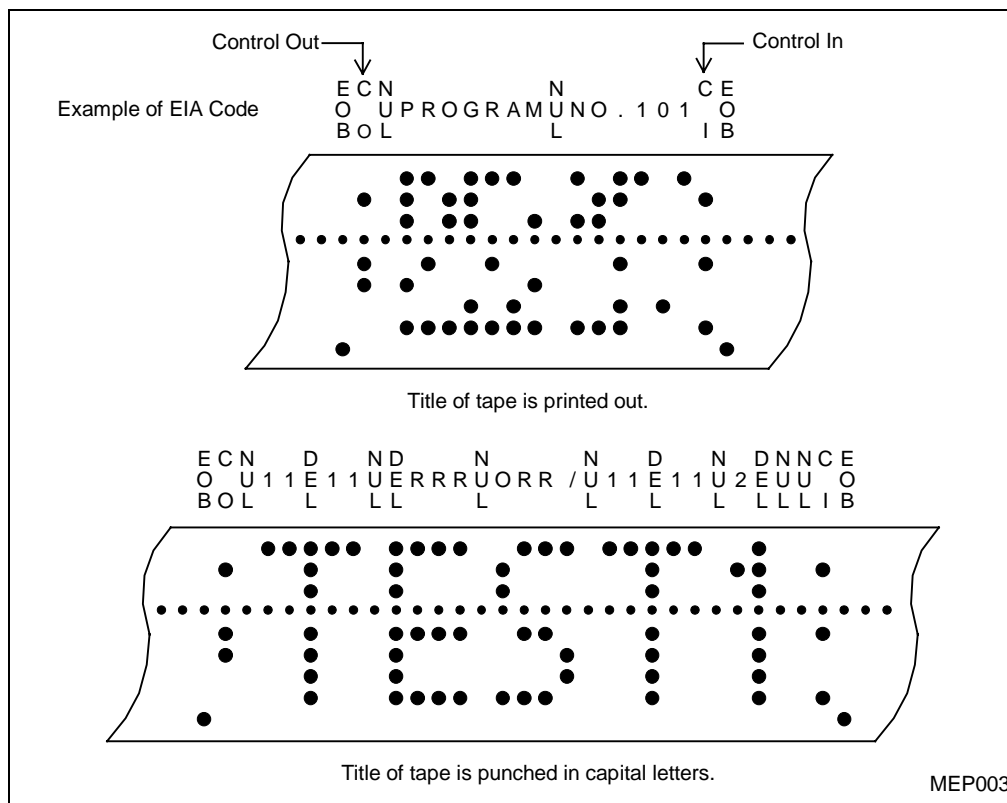
Note 3: EIA/ISO code identification is made automatically according to the first EOB/LF code appearing after the NC unit has been reset.

1. Significant information area (LABEL SKIP function)

During tape-based automatic operation, data storage into the memory, or data searching, the NC unit will ignore the entire part up to the first EOB code in the tape when the unit is turned on or reset. That is, significant information in a tape refers to the information contained in the interval from the time a character or numeric code appears, following the first EOB code after the NC unit has been reset, until a reset command is given.

2. Control Out, Control In

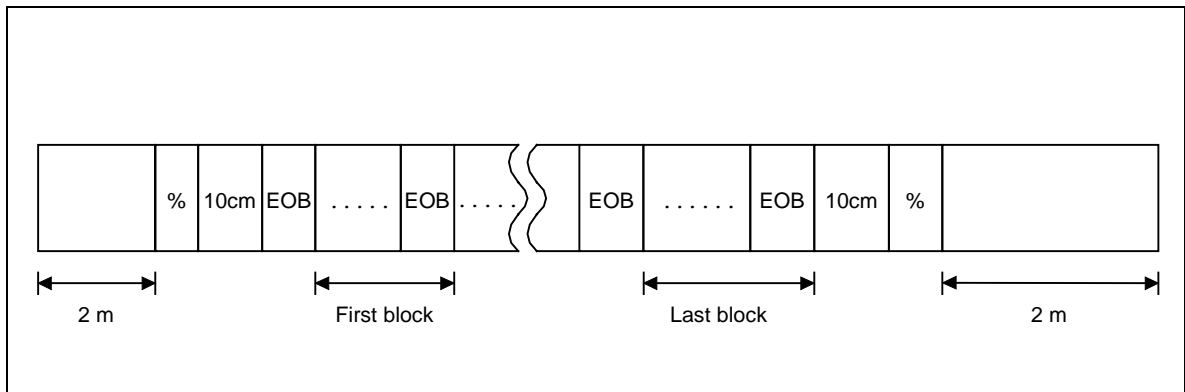
The entire information in the area from Control Out (“) to Control In (“)”) will be ignored in regard to machine control, while they will surely be displayed on the data display unit. Thus, this area can be used to store information, such as the name and number of the command tape, that is not directly related to control. During tape data storage, the information in this area will also be stored. The NC unit will enter the Control In status when power is turned on.



3. EOR code (%)

In general, the EOR (End Of Record) code is punched at both ends of a tape and has the following functions:

- To stop rewinding (only when a rewinding device is provided)
- To start rewinding during tape data search (only when a rewinding device is provided)
- To terminate the storage of tape data.

4. Tape creation method for tape operation (Only when a rewinding device is used)

The two meters of dummy at both ends and the EOR (%) at the head are not required when a rewinding device is not used.

EIA/ISO identification is made automatically by detecting whether EOB or LF initially appears after the NC unit has been reset.

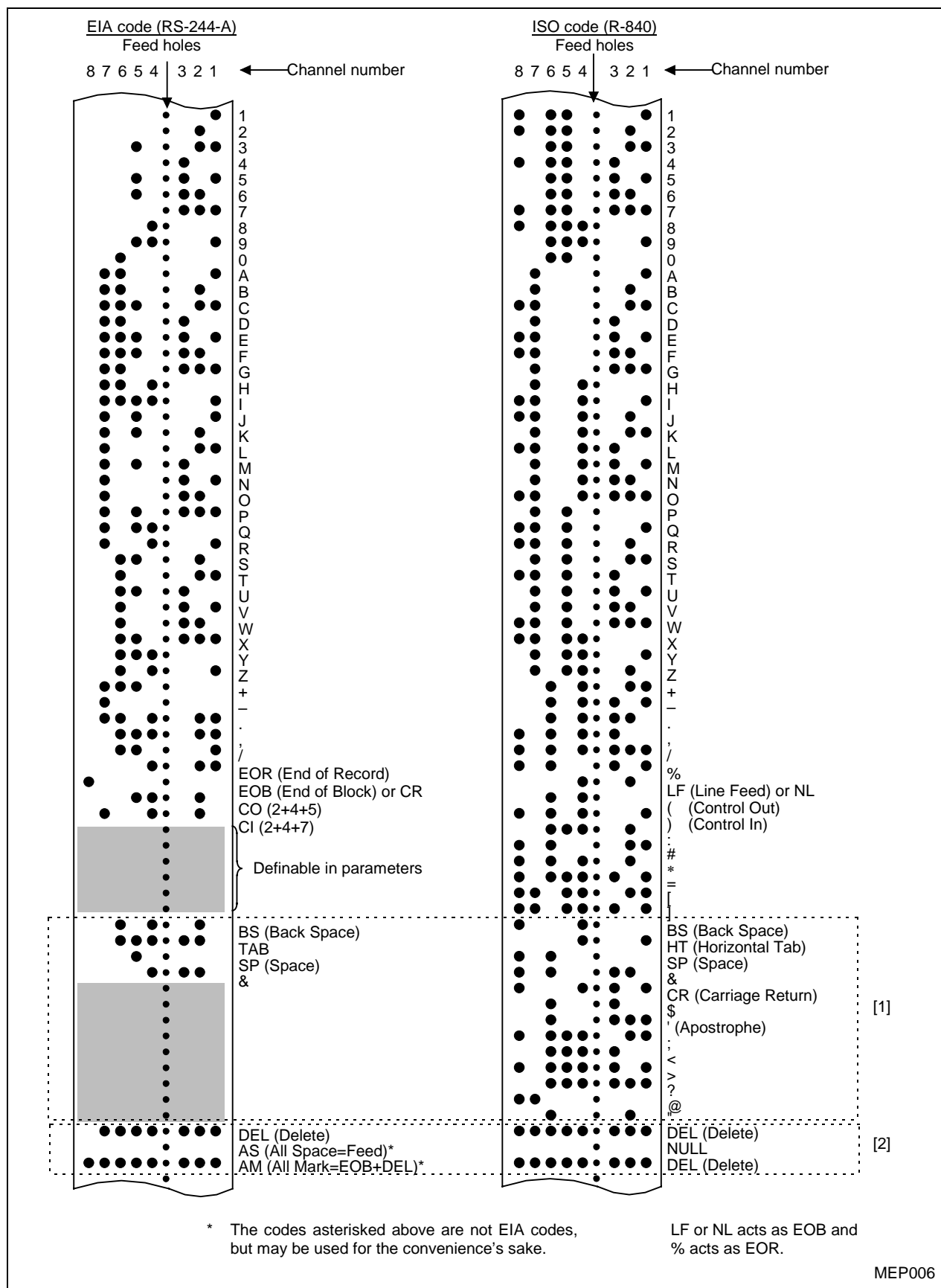


Fig. 3-1 List of tape code

Codes in square [1] will only be stored as tape data when they are present in a comment section, and ignored elsewhere in the significant information area.

Codes in square [2] are non-operative and will always be ignored (but undergo the parity-V check).

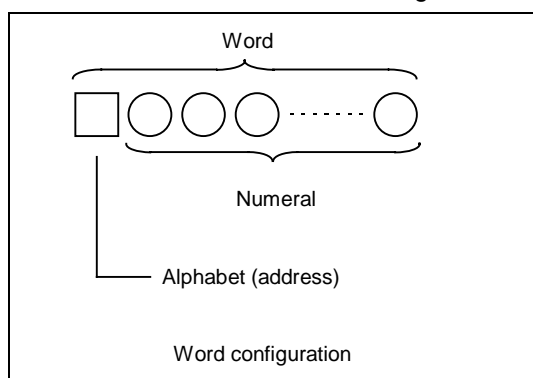
A dotted area indicates that the EIA Standard provides no corresponding codes.

3-2 Program Formats

A format predetermined for assigning control information to the NC unit is referred to as a program format. The program format used for our NC unit is word address format.

1. Words and addresses

A word is a set of characters arranged as shown below, and information is processed in words.



The alphabetic character leading a word is referred to as an address, which defines the meaning of the succeeding numeric information.

Table 3-1 Type and format of words

Item		Metric command	Inch command
Program No.		O8 or O4	
Sequence No.		N5	
Preparatory function		G21	
Moving axis	Input unit : 0.0001 mm (deg.), 0.00001 inch	X+54 Y+54 Z+54 α+54	X+45 Y+45 Z+45 α+45
Auxiliary axis	Input unit : 0.0001 mm (deg.), 0.00001 inch	I+54 J+54 K+54	I+45 J+45 K+45
Dwell	Input unit : 0.001 mm (deg.), 0.0001 inch	X53 or P8	
Feed	Input unit : 0.0001 mm (deg.), 0.00001 inch	F54	F45
Fixed cycle	Input unit : 0.0001 mm (deg.), 0.00001 inch	R+54 Q54 P23 L4	R+45 Q45 P23 L4
Tool offset		H3 or D3	
Miscellaneous function		M4	
Spindle function		S5	
Tool function		T8	
No. 2 miscellaneous function		B8, A8 or C8	
Subprogram		P8 H5 L4	
Variables number		#5	

- A. Code O8 here indicates that program number can be set as an unsigned integer of eight digits following O, and for X + 54, "+" indicates that the value can be signed (negative) and the two-digit number (54) indicates that the decimal point can be used and that five digits before and four after the decimal point are effective (5 + 4 = 9 digits are effective for a designation without decimal point).
- B. The alpha sign (α) denotes additional axis address U, V, W, A, B, or C.
- C. The format shown above applies to all types of data input, irrespective of whether a tape, the NC memory, MDI, or the operation panel is used for input of data.
- D. All numerics can have their leading zeros omitted.
- E. A program number must be set in a single-command block and in the starting block of each program.

2. Blocks

A block, unit of instruction, contains a number of words which constitute information necessary for the NC machine to perform an operation. The end of each block must be indicated by an EOB (End Of Block) code.

3. Programs

A number of blocks form one program.

3-3 Tape Data Storage Format

As with tape operation, tape data to be stored into the memory can be either of ISO or EIA code. The first EOB code read in after resetting is used by the NC unit for automatic identification of the code system ISO or EIA.

The area of tape data to be stored into the memory is, if the NC unit has been reset, from the character immediately succeeding the first EOB code to the EOR code, and in all other cases, from the current tape position to the EOR code. Usually, therefore, start tape data storage operation after resetting the NC unit.

3-4 Optional Block Skip

1. Function and purpose

Optional block skip is a function that selectively ignores that specific block within a machining program which begins with the slash code "/".

Any block beginning with "/" will be ignored if the **[BLOCK SKIP]** menu function is set to ON, or will be executed if the menu function is set to OFF.

For example, if all blocks are to be executed for a type of parts but specific blocks are not to be executed for another type, then different parts can be machined using one and the same program that contains the "/" code at the beginning of the specific blocks.

2. Operating notes

1. Blocks that have already been read into the pre-read buffer cannot be skipped.
2. This function is valid even during sequence number search.
3. During tape data storage (input) or output, all blocks, including those having a "/" code, are in- or outputted, irrespective of the status of the **[BLOCK SKIP]** menu function.

3-5 Program Number, Sequence Number and Block Number: O, N

Program numbers, sequence numbers and block numbers are used to monitor the execution status of a machining program or to call a machining program or a specific process within a machining program.

Program numbers identify programs on a workpiece-by-workpiece or subprogram-by-subprogram basis. A program number must be set using the letter O (address) and a numeric of a maximum of eight digits (or four digits when **TAP26** bit 3 = 1).

Sequence numbers are assigned to command blocks as required. A sequence number must be set using the letter N (address) and a numeric of a maximum of five digits.

Block numbers are counted automatically within the NC unit, and preset to 0 each time a program number or a sequence number is read. The number will be counted up by one if the block to be read does not have an assigned program number or sequence number.

All blocks of a machining program, therefore, can be uniquely defined by combining program number, sequence number and block number as shown in the table below.

NC input machining program	NC monitor display		
	Program No.	Sequence No.	Block No.
O1234 (DEMO. PROG)	1234	0	0
G92X0Y0	1234	0	1
G90G51X-150. P0.75	1234	0	2
N100G00X-50. Y-25.	1234	100	0
N110G01X250. F300	1234	110	0
Y-225.	1234	110	1
X-50.	1234	110	2
Y-25.	1234	110	3
N120G51Y-125. P0.5	1234	120	0
N130G00X-100. Y-75.	1234	130	0
N140G01X-200.	1234	140	0
Y-175.	1234	140	1
X-100.	1234	140	2
Y-75.	1234	140	3
N150G00G50X0Y0	1234	150	0
N160M02	1234	160	0
%			

3-6 Parity-H/V

One method of checking if the tape is correctly created is by parity checks. Parity checks are performed to check a tape for errors in punched codes, that is, for punching errors. There are two types of parity checks: parity-H and parity-V.

1. Parity-H check

Parity-H checks are intended to check the quantity of punched holes which form one character, and performed during tape operation, tape loading, and sequence-number searching.

A parity-H error occurs in the following cases:

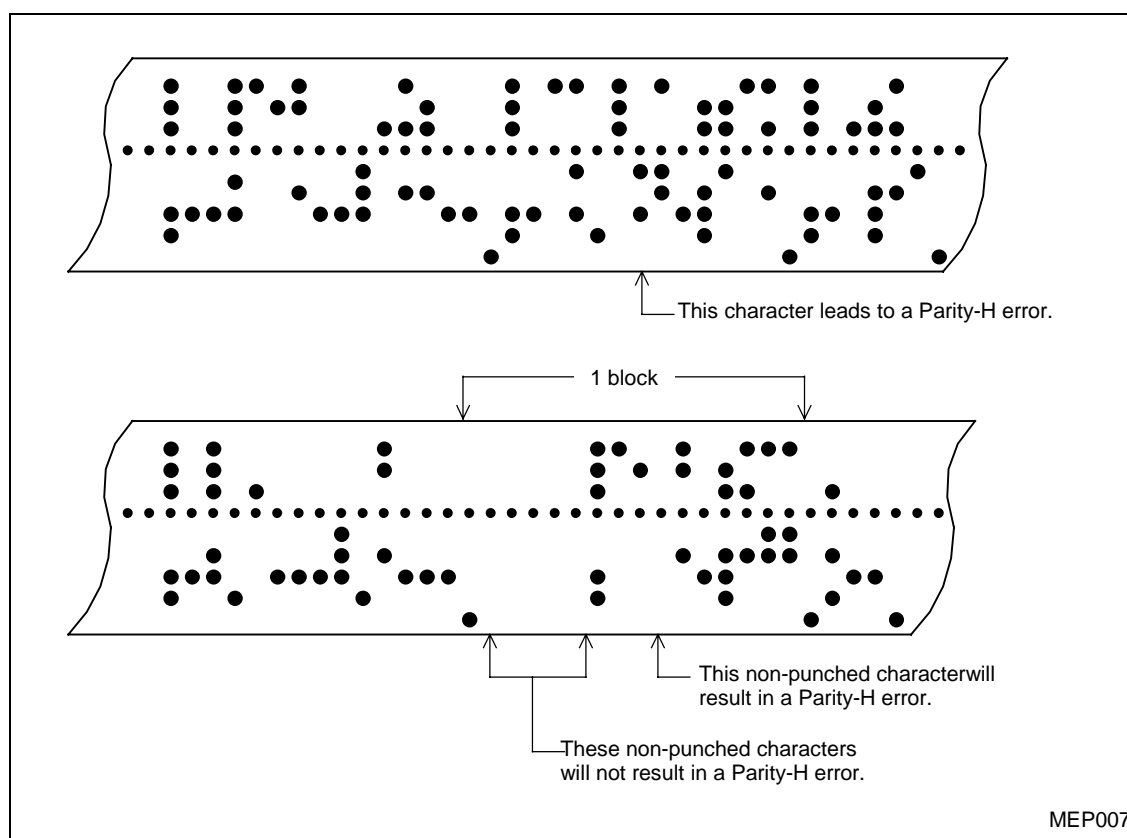
- ISO Codes

If a code with an odd number of punched holes is present in the significant information area

- EIA Codes

If a code with an even number of punched holes is present in the significant information area or if non-punched holes (sprockets only) are present after a significant code in one block

Example 1: Parity-H error (for EIA codes)



If a parity-H error occurs, the tape will stop at the position next to the error code.

2. Parity-V check

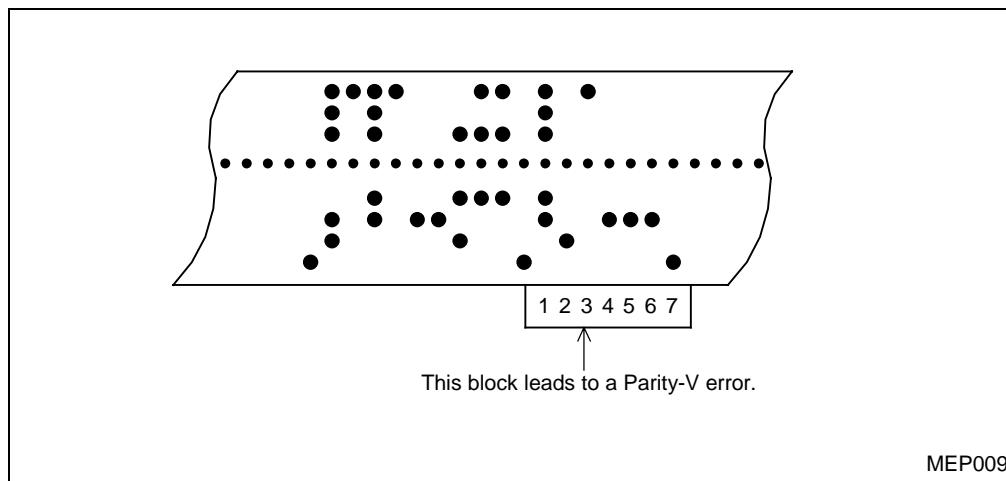
Parity-V checks will be performed during tape operation, tape loading, or sequence-number searching, if Parity V check item on the **PARAMETER** display is set to ON. Parity V will not be checked, however, during memory operation.

A parity-V error occurs in the following case:

If an odd number of codes are present in the significant information area from the first significant code in the vertical direction to the EOB code, that is, if an odd number of characters are present in one block.

In the event of a parity-V error, the tape stops at the code next to the EOB.

Example 3: Parity-V error



Note 1: During parity-V check, some types of code in a tape are not counted as characters. Refer to Fig. 3-1 "Tape codes" for further details.

Note 2: Space codes in the area from the first EOB code to the first address code or slash code "/" are not subjected to counting for parity-V check.

3-7 List of G-Codes

G-code	Group	Function	G-code	Group	Function
*00	01	Positioning	*50	11	Scaling cancel
*01	01	Linear interpolation	51	11	Scaling
02	01	Circular interpolation CW	*50.1	19	Mirror image cancel
03	01	Circular interpolation CCW	51.1	19	Mirror image
02.1	00	Spiral interpolation CW	52	00	Local coordinate system setting
03.1	00	Spiral interpolation CCW	53	00	Machine coordinate system selection
04		Dwell	*54	12	Workpiece coordinate system 1 selection
05		High-speed machining mode	55	12	Workpiece coordinate system 2 selection
06			56	12	Workpiece coordinate system 3 selection
06.1	01	Spline interpolation (fine spline function)	57	12	Workpiece coordinate system 4 selection
06.2	01	NURBS interpolation	58	12	Workpiece coordinate system 5 selection
07		Virtual-axis interpolation	59	12	Workpiece coordinate system 6 selection
08			60	00	One-way positioning
09	00	Exact-stop check	61	13	Exact-stop check mode
10	00	Programmed parameter input	61.1	13	Shape correction mode
11			61.2	13	Shape correction (Modal spline interpolation)
12			62	13	Automatic corner override
13			63	13	Tapping mode
14			*64	13	Cutting mode
15			65	00	User macro simple call
16			66	14	User macro modal call A
*17	02	Plane selection X-Y	66.1	14	User macro modal call B
18	02	Plane selection Z-X	*67	14	User macro modal call cancel
19	02	Plane selection Y-Z	68	16	Programmed coordinates rotation
*20	06	Inch command	69	16	Programmed coordinates rotation cancel
*21	06	Metric command	70		
22	04	Pre-move stroke check	71.1		Fixed cycle (chamfering cutter 1)
*23	04	Pre-move stroke check cancel	72.1		Fixed cycle (chamfering cutter 2)
24			73	09	Fixed cycle (high-speed deep-hole drilling)
25			74	09	Fixed cycle (reverse tapping)
26			75	09	Fixed cycle (boring)
27	00	Reference point check	76	09	Fixed cycle (boring)
28	00	Reference point return	77	09	Fixed cycle (back spot facing)
29	00	Starting point return	78	09	Fixed cycle (boring)
30	00	No. 2 through 4 reference point return	79	09	Fixed cycle (boring)
31	00	Skip	*80	09	Fixed cycle cancel
31.1	00	Multi-step skip 1	81	09	Fixed cycle (spot drilling)
31.2	00	Multi-step skip 2	82	09	Fixed cycle (counter boring)
31.3	00	Multi-step skip 3	83	09	Fixed cycle (deep-hole drilling)
32			84	09	Fixed cycle (tapping)
33	01	Threading	84.2	09	Fixed cycle (synchr. tapping)
34			84.3	09	Fixed cycle (synchr. reverse tapping)
35			85	09	Fixed cycle (reaming)
36			86	09	Fixed cycle (boring)
37	00	Automatic tool length measurement	87	09	Fixed cycle (back boring)
38	00	Tool diameter offset vector selection	88	09	Fixed cycle (boring)
39	00	Tool diameter offset corner arc	89	09	Fixed cycle (boring)
*40	07	Tool diameter offset cancel	*90	03	Absolute data input
40.1	15	Shaping cancel	*91	03	Incremental data input
41	07	Tool diameter offset to the left	92	00	Coordinate system setting
41.1	15	Shaping to the left	92.5		Workpiece coordinate system rotation
42	07	Tool diameter offset to the right	93	05	Inverse time feed
42.1	15	Shaping to the right	*94	05	Asynchronous feed (feed per minute)
43	08	Tool length offset (+)	95	05	Synchronous feed (feed per revolution)
44	08	Tool length offset (-)	96		
45	00	Tool position offset, extension	97		
46	00	Tool position offset, reduction	*98	10	Initial point level return in fixed cycle
47	00	Tool position offset, double extension	99	10	R-point level return in fixed cycle
48	00	Tool position offset, double reduction	01 - 255		User macro G-code call (max. 10 codes)
*49	08	Tool length offset cancel			

- The asterisk (*) indicates that the code can be or is automatically selected as an initial status.

3-8 List of M-Codes

The following list shows the general M-codes used in the machining centers. It must be noted, however, that certain codes can not be used in certain machines and that other codes than those listed here can be ordered. For details, refer to the Operating Manual of the machine.

M-code	Function	M-code	Function
0	Programmed stop	48	Spindle speed and feed rate correction valid
1	Optional stop	49	Spindle speed and feed rate correction invalid
2	End of program (EIA/ISO)	50	Air blast
3	Revolution of spindle (normal)	51	Spindle-through coolant
4	Revolution of spindle (reverse)	52	Tapping coolant
5	Stop of spindle		
6	Changing of tool (EIA/ISO)	58	Check of tool life for spare-tool management
7	Mist coolant		
8	Liquid coolant	64	Pallet door closed
9	Stop of all coolant and compressed air	65	Pallet door opened
10	Clamping of tool on the spindle		
11	Unclamping of tool on the spindle	68	Clamping of pallet
		69	Unclamping of pallet
15	Magazine cover closed		
16	Magazine cover opened	71	Selection of pallet No. 1
		72	Selection of pallet No. 2
19	Orientation of spindle	73	Selection of pallet No. 3
		74	Selection of pallet No. 4
23	Error detection valid	75	Selection of pallet No. 5
24	Error detection invalid	76	Selection of pallet No. 6
30	End of program and rewinding of tape (EIA/ISO)	90	Cancellation of mirror image (MAZATROL)
		91	Mirror image WPC-X valid (MAZATROL)
33	Tool length measurement unit advanced	92	Mirror image WPC-Y valid (MAZATROL)
34	Tool length measurement unit retracted	93	Mirror image WPC-4 valid (MAZATROL)
35	Detection of tool breakage		
36	Selection of spindle speed range (Low)	98	Call-up of subprogram (EIA/ISO)
37	Selection of spindle speed range (Low/Medium low)	99	End of subprogram (EIA/ISO)
38	Selection of spindle speed range (Low/Medium/Medium high)	100	External M-code 1
39	Selection of spindle speed range (High)	101	External M-code 2
40	Selection of spindle speed range (Neutral)		
		122	Gap eliminator valid
42	Reverse rotation of indexing table	123	Gap eliminator invalid
43	External M-code 3/C-axis unclamping		
44	External M-code 4/C-axis clamping	130	Niagara coolant
45	External M-code 5		
46	A-axis unclamping	132	Spindle-through air blast
47	A-axis clamping		

- NOTE -

4 BUFFER REGISTERS

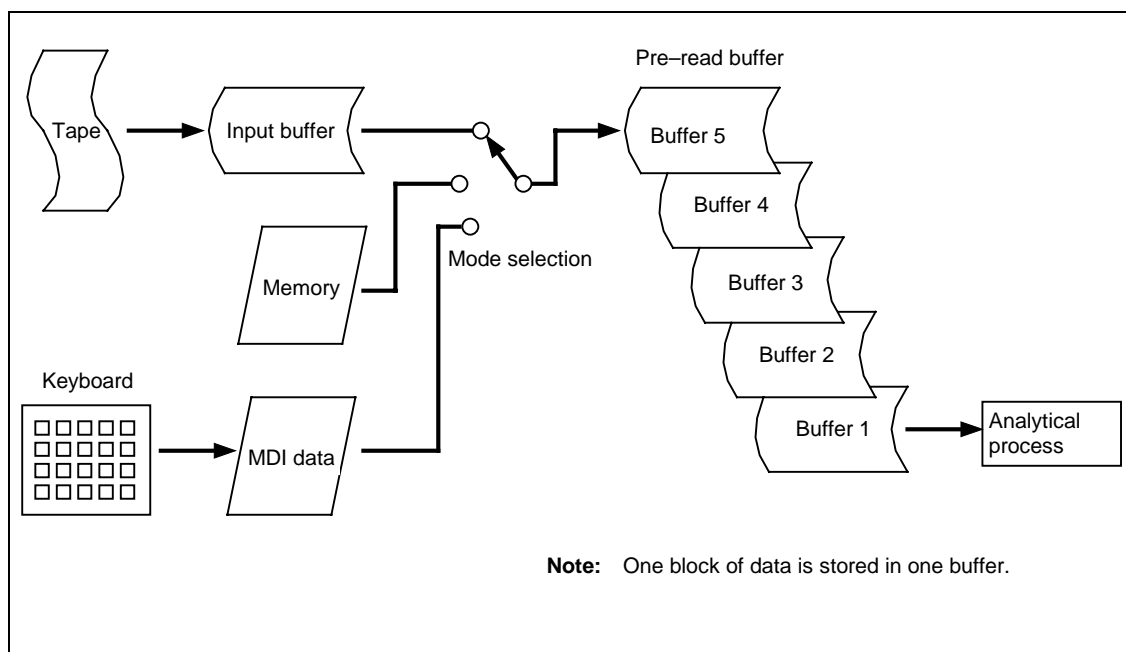
4-1 Input Buffer

1. Outline

During tape operation, when the pre-read buffer becomes empty, the contents of the input buffer will be immediately shifted into the pre-read buffer and next data (up to 248 characters) will be pre-read and then stored into the input buffer.

The input buffer makes block-to-block connections smooth by eliminating any operational delays due to the tape-reading time of the tape reader.

The effectiveness of pre-reading, however, may be lost if the status persists that the execution time of the block is shorter than the tape-reading time of the next block continues.



2. Remarks

- The codes which are read into the buffer are the significant codes in the significant information.
- The contents of the buffer are cleared by resetting.

4-2 Pre-Read Buffer

1. Outline

During automatic operation, one block of data is usually pre-read to ensure smooth analysis of the program. During tool diameter offsetting, however, two to five blocks of data (only during interference check) are pre-read for crossing-point calculation.

2. Remarks

- The memory capacity of the total buffer is 248 characters.
- One block of data is stored into one buffer.
- Only the significant codes in the significant information area are read into the buffer.
- Codes existing between Control Out and Control In are not read into the pre-read buffer. If Optional Block Skip is valid, codes from / to EOB will not be read into the pre-read buffer.
- The contents of the buffer are cleared by resetting.
- If SINGLE BLOCK is turned ON during continuous operation, processing will stop after pre-reading the next block data.

5 POSITION PROGRAMMING

5-1 Dimensional Data Input Method: G90, G91

1. Function and purpose

Setting of G90 or G91 allows succeeding dimensional data to be processed as absolute data or incremental data.

Setting of arc radius (with address R) or arc center position (with addresses I, J, K) for circular interpolation, however, must always refer to incremental data input, irrespective of preceding G90 command.

2. Programming format

G90 (or G91) Xx₁ Yy₁ Zz₁ αα₁ (α: Additional axis)

where G90: Absolute data input

G91: Incremental data input

3. Detailed description

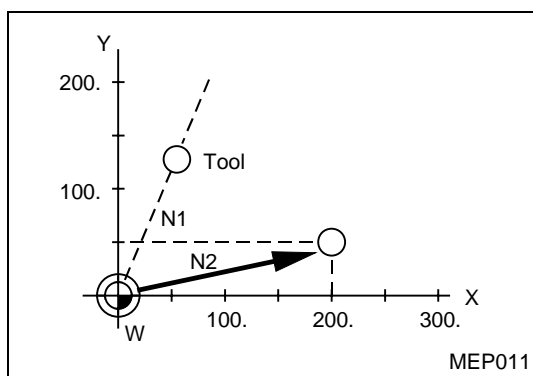
1. In the absolute data mode, axis movement will be performed to the program-designated position within the workpiece coordinate system, irrespective of the current position.

```
N1 G90G00X0 Y0
```

In the incremental data mode, axis movement will be performed through the program-designated distance as relative data with respect to the current position.

```
N2 G91G01X200. Y50. F100
```

```
N2 G90G01X200. Y50. F100
```



Commands for a movement from the origin of the workpiece coordinate system are given with the same values, irrespective of whether the absolute data mode or the incremental data mode is used.

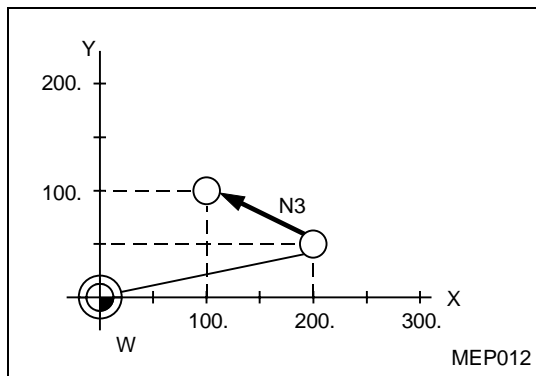
2. The last G90 or G91 command works as a modal one for the following blocks.

(G90) N3 X100. Y100.

This block will perform a movement to the position of X = 100 and Y = 100 in the workpiece coordinate system.

(G91) N3 X-100. Y50.

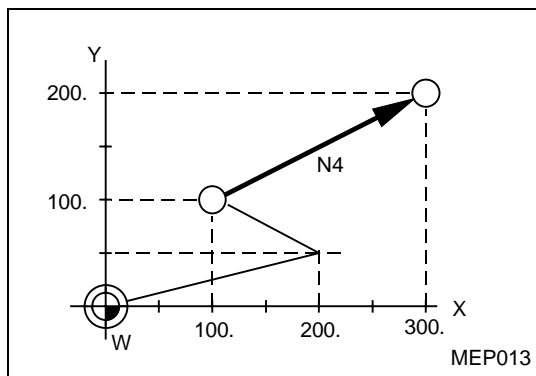
This block will perform a movement of -100 on the X-axis and +50 on the Y-axis, and thus result in a movement to the position of X = 100 and Y = 100.



3. Multiple G90 or G91 commands can be set in one block, and thus only a specific address can be set as absolute data or incremental data.

N4 G90X300. G91Y100.

In this example, dimensional data X300 preceded by G90 will be processed as an absolute data input, and Y100 preceded by G91 as an incremental data input. Therefore, this block will result in a movement to the position of X = 300 and Y = 200 (100 + 100) in the workpiece coordinate system.



Moreover, G91 (incremental data input mode) will work for the succeeding blocks.

4. Either the absolute data mode or the incremental data mode can be freely selected as initial mode by setting the bit 2 of user parameter **F93**.
5. Even in the MDI (Manual Data Input) mode, G90 and G91 will also be handled as modal commands.

5-3 Decimal Point Input

1. Function and purpose

The decimal point can be used to determine the units digit (mm or inch) of dimensional data or feed rate. Moreover, a parameter is provided to specify whether the least significant digit of data without decimal point refers to the minimum data input unit (type I) or to the units digit (type II).

2. Programming format

○○○○○.○○○○ Metric system
○○○○.○○○○○ Inch system

3. Detailed description

- Decimal-point commands are valid only for the distance, angle, time, speed, and scaling factor (only after G51) that have been set in the machining program.
- As listed in the table below, the meaning of command data without the decimal point differs between decimal-point input types I and II according to the type of command unit system.

Command	Command unit × 10	Type I	Type II
X1	OFF	0.0001 (mm, inches, deg)	1.000 (mm, inches, deg)
	ON	0.001 (mm, inches, deg)	1.00 (mm, inches, deg)

- Decimal-point commands are only valid for addresses X, Y, Z, U, V, W, A, B, C, I, J, K, E, F, P, Q and R, where address P only refers to a scaling factor.
- The number of effective digits for each type of decimal-point command is as follows:

	Move command (Linear)		Move command (Rotational)		Feed rate		Dwell	
	Integral part	Decimal part	Integral part	Decimal part	Integral part	Decimal part	Integral part	Decimal part
mm	0. - 99999.	.0000 - .9999	0. - 99999.	.0000 - .9999	0. - 200000.	.0000 - .9999	0. - 99999.	.000 - .999
inch	0. - 9999.	.00000 - .99999	0. - 99999. (359.)	.0000 - .9999	0. - 20000.	.00000 - .99999	0. - 99999.	.000 - .999

- Decimal-point commands are also valid for definition of variables data used in subprograms.
- For data which can be, but is not specified with the decimal point, either the minimum program data input unit or mm (or in.) unit can be selected using bit 5 of parameter **F91**.
- A decimal-point command issued for an address which does not accept the decimal point will be processed as data that consists of an integral part only. That is, all decimal digits will be ignored. Addresses that do not accept the decimal point are D, H, L, M, N, O, S and T. All types of variables command data are handled as the data having the decimal point.

4. Sample programs

A. Sample programs for addresses accepting the decimal point

Command category Program example	For 1 = 1 μ	For 1 = 10 μ	1 = 1 mm
G0X123.45 (With the decimal point always given as the millimeter point)	X123.450 mm	X123.450 mm	X123.450 mm
G0X12345	X12.345 mm*	X123.450 mm**	X12345.000 mm***
#111=123 #112=5.55 X#111 Y#112	X123.000 mm Y5.550 mm		
#113=#111+#112 (ADD)	#113 = 128.550		
#114=#111-#112 (SUBTRACT)	#114 = 117.450		
#115=#111.#112 (MULTIPLY)	#115 = 682.650		
#116=#111/#112 #117=#112/#111 (DIVIDE)	#116 = 22.162 #117 = 0.045		

* The least significant digit is given in 1 micron.

** The least significant digit is given in 10 microns.

*** The least significant digit is given in 1 mm.

B. Validity of decimal point for each address

Address	Decimal point command	Application	Remarks	Address	Decimal point command	Application	Remarks
A	Valid	Coordinate position data		P	Invalid	Dwell time	
	Invalid	Rotary table Miscellaneous function code			Valid	Subprogram call number	
	Valid	Linear angle data			Invalid	Number of helical pitches	
B	Valid	Coordinate position data			Invalid	Offset amount (in G10)	
	Invalid	Rotary table Miscellaneous function code			Valid	Scaling factor	
C	Valid	Coordinate position data			Invalid	Rank for NURBS curve	
	Invalid	Rotary table Miscellaneous function code		Q	Valid	Cutting depth for deep-hole drilling cycle	
	Valid	Corner chamfering amount			Valid	Shift amount for back boring	
D	Invalid	Offset number (tool position, tool length and tool diameter)			Valid	Shift amount for fine boring	
E	Valid			R	Valid	R point in fixed cycle	
F	Valid	Feed rate			Valid	Radius of an arc with R selected	
G	Valid	Preparatory function code			Valid	Radius of an arc for corner rounding	
					Valid	Offset amount (in G10)	
H	Invalid	Offset number (tool position, tool length and tool diameter)			Valid	Weight for NURBS curve	
	Invalid	Intra-subprogram sequence number		S	Invalid	Spindle function code	
I	Valid	Coordinate of arc center		T	Invalid	Tool function code	
	Valid	Vector component for tool diameter offset		U	Valid	Coordinate position data	
J	Valid	Coordinate of arc center		V	Valid	Coordinate position data	
	Valid	Vector component for tool diameter offset		W	Valid	Coordinate position data	
K	Valid	Coordinate of arc center		X	Valid	Coordinate position data	
	Valid	Vector component for tool diameter offset			Valid	Dwell time	
	Valid	Knot for NURBS curve		Y	Valid	Coordinate position data	
L	Invalid	Fixed cycle/subprogram repetition		Z	Valid	Coordinate position data	
M	Invalid	Miscellaneous function code					
N	Invalid	Sequence number					
O	Invalid	Program number					

Note: The decimal point is valid in all the arguments for a user macro program.

6 INTERPOLATION FUNCTIONS

6-1 Positioning (Rapid Feed): G00

1. Function and purpose

Positioning command G00 performs a linear positioning to the ending point specified by a coordinate word.

2. Programming format

G00 Xx Yy Zz αα (α: Additional axis)

where x, y, z, and α each denote a coordinate. The absolute or the incremental data input is used according to the status of G90/G91 existing at the particular time.

3. Detailed description

1. Once this command has been given, the G00 mode will be retained until any other G-code command that overrides this mode, that is, G01, G02, or G03 of group 01 is given. Thus, it is merely required to input coordinate words for G00 movements in the succeeding blocks. This function is referred to as the modal function of command.
2. In the G00 mode, acceleration/deceleration always takes place at the starting/ending point of a block and, after an in-position status check, the program proceeds to the next block. The width of in-position can be changed using machine parameter **S13**.
3. The preparatory functions (G72 to G89) of group 09 are cancelled by the G00 command.
4. The tool path can be selected using the bit 6 of user parameter **F91**, but the positioning time remains unchanged.

Linear path (**F91**bit6 = 0):

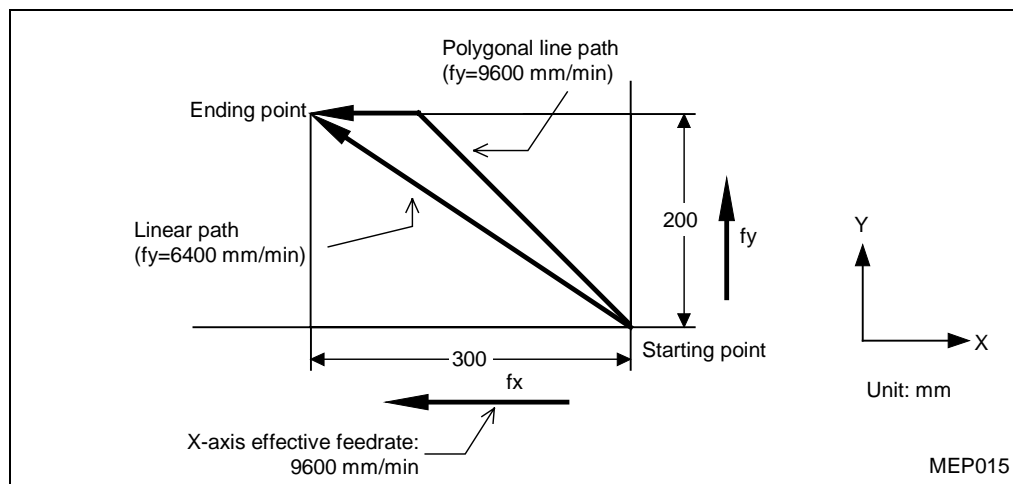
As with linear interpolation (G01), the tool speed is limited according to the rapid feed rates of related axes.

Polygonal line path (**F91**bit6 = 1): The tool is positioned at the rapid feed rate independently on each axis.

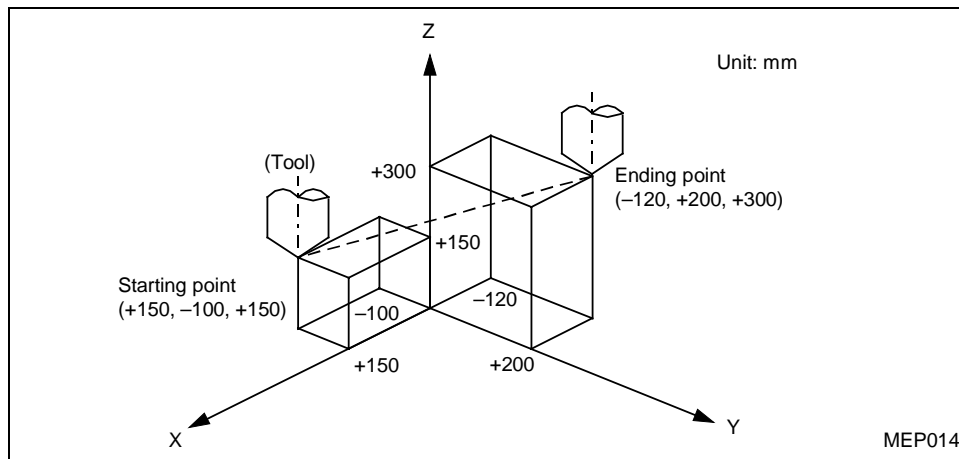
For example, if a rapid feed rate of 9600 mm/min is preset for both X- and Y-axes, then the command

```
G91 G00 X-300.000 Y200.000
```

will move the tool as shown in the figure below.



4. Sample programs



```
G91 G00 X-270.000 Y300.000 Z150.000
```

5. Rapid feed (G00) deceleration check

When processing of rapid feed (G00) is completed, checks will be performed to ensure that the remaining distance of movement on each axis has decreased below the predetermined value. Only after completion of these checks, will the next block be executed. See Figure 6-1 below.

The remaining distance is checked, based on the rapid feed in-position width, L_R . (L_R : Data set in parameter **S13**.)

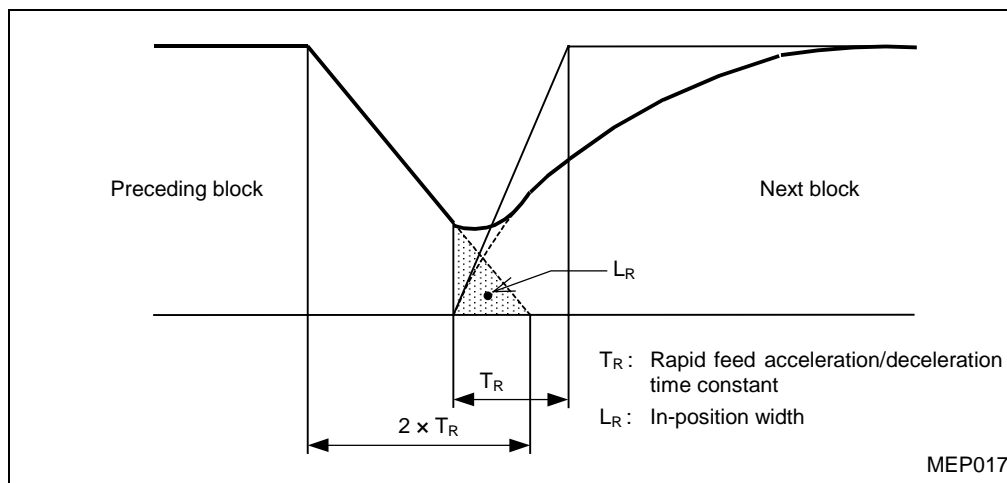


Fig. 6-1 G00 Deceleration pattern

The in-position width L_R which is equivalent to the hatched area in the figure above, relates to the remaining distance of the preceding block at the start of processing the next block.

Rapid feed deceleration checking is intended to reduce the positioning time. Increasing the setting of parameter **S13** reduces the positioning time correspondingly, while at the same time it increases the remaining distance of the preceding block at the start of the next block. As a result, trouble may occur during machining.

The checks of the remaining distance are performed at fixed time intervals. Because of this, the positioning time may not be reduced as much as expected from the setting of parameter **S13**.

6-2 One-Way Positioning: G60

1. Function and purpose

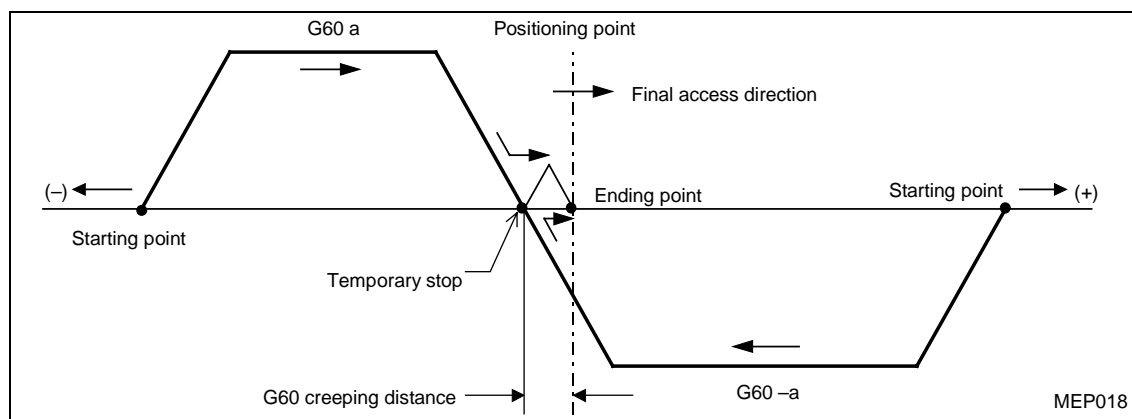
Highly accurate positioning free from any backlash error can be performed when the axis movement is controlled by the G60 command so that the final access always takes place in one determined direction.

2. Programming format

G60 Xx Yy Zz $\alpha\alpha$ (α : Additional axis)

3. Detailed description

1. The direction of final access and its creeping distance must be set in parameter I1.
2. After rapid approach to a position away from the ending point by the creeping distance, the final access is performed in the predetermined direction at a speed corresponding with the rapid feed.



3. The positioning pattern described above also applies during machine locking or for a Z-axis command with the Z-axis cancellation activated.
4. The mirror-image function does not affect the final access direction: the reverse approach in direction of the mirrored ending point will occur to such a temporary stop position that the final access can take place in the predetermined direction.
5. In the dry run mode (G00 mode), the whole positioning is carried out at the dry-running speed.
6. The creeping to the ending point can be halted with Reset, Emergency stop, Interlock, or Feed hold, or by setting the rapid feed override to 0 (zero).
The creeping is performed according to the setting of the rapid feed, and the rapid feed override function is also effective for the creeping.
7. One-way positioning is automatically invalidated for the hole-drilling axis in hole-drilling fixed-cycle operations.
8. One-way positioning is automatically invalidated for shifting in fine-boring or back-boring fixed-cycle operations.
9. Usual positioning is performed for an axis not having a parameter-set creeping distance.
10. One-way positioning is always of non-interpolation type.
11. An axis movement command for the same position as the ending point of the preceding block (movement distance = 0) will cause reciprocation through the creeping distance so that the final access can be performed in the predetermined direction for an accurate positioning to the desired point.

6-3 Linear Interpolation: G01

1. Function and purpose

This command moves (interpolates) a tool from the current position to the ending point specified by a coordinate word, at the feed rate specified with address F. The specified feed rate acts here as the linear velocity relative to the direction of movement of the tool center.

2. Programming format

G01 Xx Yy Zz αα Ff (α: Additional axis)

where x, y, z, and α each denote a coordinate. The absolute or the incremental data input is used according to the status of G90/G91 existing at the particular time.

3. Detailed description

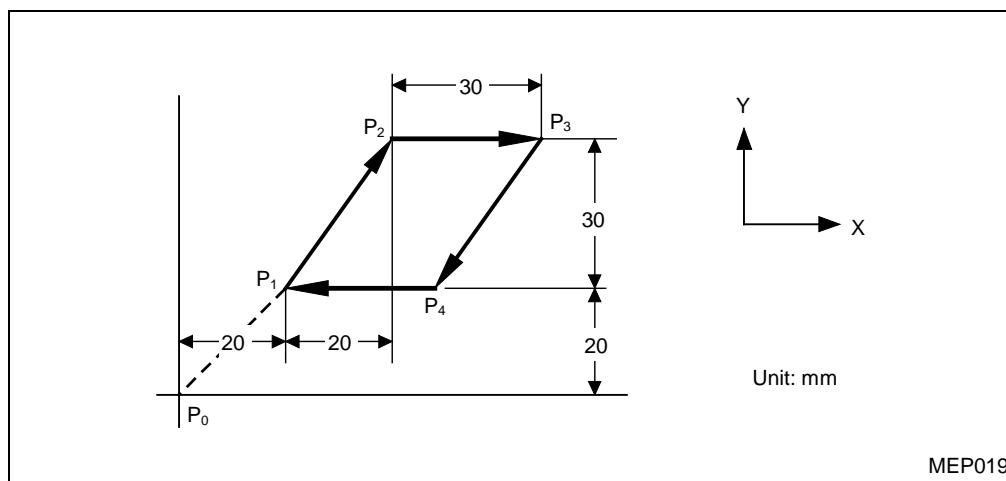
Once this command has been given, the G01 mode will be retained until any other G-code command that overrides this mode, that is, either G00, G02, or G03 of command group 01 is given. Thus, it is merely required to input coordinate words for linear interpolations in the succeeding blocks unless the feed rate must be changed.

A programming error will result if no F-code command has been given to the first G01 command. The feed rates for rotational axes must be set in deg/min. (Example: F300. = 300 deg/min)

The preparatory functions (G72 to G89) of group 09 are cancelled by G01.

4. Sample program

The following shows a program for moving the tool at a cutting feed rate of 300 mm/min on the route of $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$ (where the section from $P_0 \rightarrow P_1$ forms a positioning route for the tool):



G91	G00	X20.000	Y20.000	$P_0 \rightarrow P_1$
	G01	X20.000	Y30.000 F300	$P_1 \rightarrow P_2$
		X30.000		$P_2 \rightarrow P_3$
		X-20.000	Y-30.000	$P_3 \rightarrow P_4$
		X-30.000		$P_4 \rightarrow P_1$

6-4 Plane Selection: G17, G18, G19

1. Function and purpose

Commands G17, G18 and G19 are used to select a plane on which circular interpolation (helical cutting included) and tool diameter offsetting are to be done.

These commands are also used to select a plane where figures or program coordinates are to be rotated.

2. Programming format

G17 (X-Y plane selection)	} X, Y and Z denote respective coordinate axes or their corresponding parallel axes.
G18 (Z-Y plane selection)	
G19 (Y-Z plane selection)	

3. Detailed description

Registering both the three fundamental axes and their corresponding parallel axes as parameters allows you to select a plane defined by any two non-parallel axes. In addition, a plane compassing rotational axes can also be selected by registering those rotational axes beforehand as parallel ones.

For standard plane selection, there is a fixed relationship between the three fundamental axes (X, Y, Z) and their corresponding parallel axes (U, V, W); a plane compassing rotational axes A, B or C cannot be selected.

The available planes are the following five types:

- Plane for circular interpolation (helical cutting included)
- Plane for tool diameter offsetting
- Plane for figure rotation
- Plane for programmed coordinate rotation
- Plane for tool positioning in fixed-cycle machining (Only for fixed-cycle machining with a positioning plane selection)

4. Parameter registration

Table 6-1 Plane selecting parameters

	Basic axis	Parallel axis 1	Parallel axis 2
I	X	U	A
J	Y	V	B
K	Z	W	C

5. Plane selection methods

1. In Table 6-1 shown above:

“I” signifies the horizontal axis of a G17 plane or the vertical axis of a G18 plane.

“J” signifies the vertical axis of a G17 plane or the horizontal axis of a G19 plane.

“K” signifies the horizontal axis of a G18 plane or the vertical axis of a G19 plane.

That is, G17: I-J plane, G18: K-I plane, G19: J-K plane.

2. Which of the fundamental axes or their parallel axes are to form the respective plane is determined by the plane selection command (G17, G18 or G19) and the axis addresses specified in the same block.

G17X_Y_	X-Y plane
G17U_Y_	U-Y plane
G18X_Z_	Z-X plane
G18U_W_	W-U plane
G19Y_Z_	Y-Z plane
G19Y_W_	Y-W plane

3. Plane selection does not occur for blocks that do not have an issued plane-selection command (G17, G18 or G19).

G17X_Y_	X-Y plane
Y_Z_	X-Y plane (No plane change)

4. An omitted axis address in the block of plane-selection command (G17, G18 or G19) will be regarded as the corresponding fundamental axis.

G17	X-Y plane
G17U_	U-Y plane
G18U_	Z-U plane
G18W_	W-X plane
G19Y_	Y-Z plane
G19W_	Y-W plane

5. Commands for an axis other than those of the plane selected by G17, G18 or G19 will be independently processed:

G17U_Z_

By the command above, the U-Y plane will be selected and Z-axis movement will occur independently of the selected plane.

6. If fundamental axes and their parallel axes are set in overlapping form in a block containing G17, G18 or G19, the plane is selected according to the axis priority.

Order of priority : Fundamental axis > First parallel axis > Second parallel axis.

Provided that the parameter registration is as shown in Table 6-1 above, if you set

G17U_Y_A_

then, U-Y plane will be selected and A-axis movement will occur independently of the selected plane.

Remark: Use bits 0 and 1 of parameter **F92** to set the initial plane which is to be selected upon power-on or resetting.

6-5 Circular Interpolation: G02, G03

1. Function and purpose

Commands G02 and G03 feed the tool along an arc.

2. Programming format

G02 (G03) Xx Yy Ii Jj Ff

where G02/G03 : Circular movement direction CW/CCW

x, y : Coordinates of ending point

i, j : Coordinates of arc center

f : Feed rate

Use addresses X, Y and Z (or their parallel axes) to specify the coordinates of the ending point of arc, and addresses I, J and K for the coordinates of arc center.

Combined use of absolute and incremental data input is available for setting the coordinates of the ending point of arc. For the coordinates of the arc center, however, incremental data relative to the starting point must always be set.

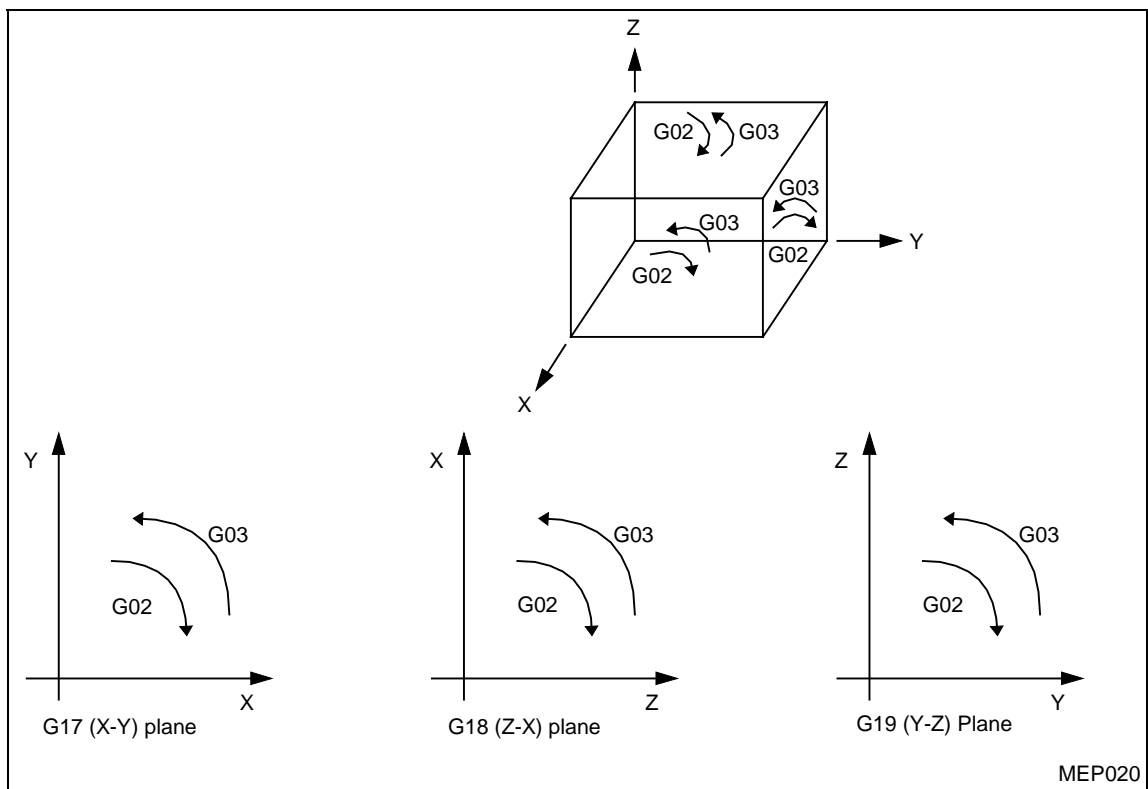
3. Detailed description

Once the G02 (or G03) command has been given, this command mode will be retained until any other G-code command used to override the G02 (or G03) command mode, that is, G00 or G01 of group 01 is given.

The direction of circular movement is determined by G02/G03.

G02: CW (Clockwise)

G03: CCW (Counterclockwise)



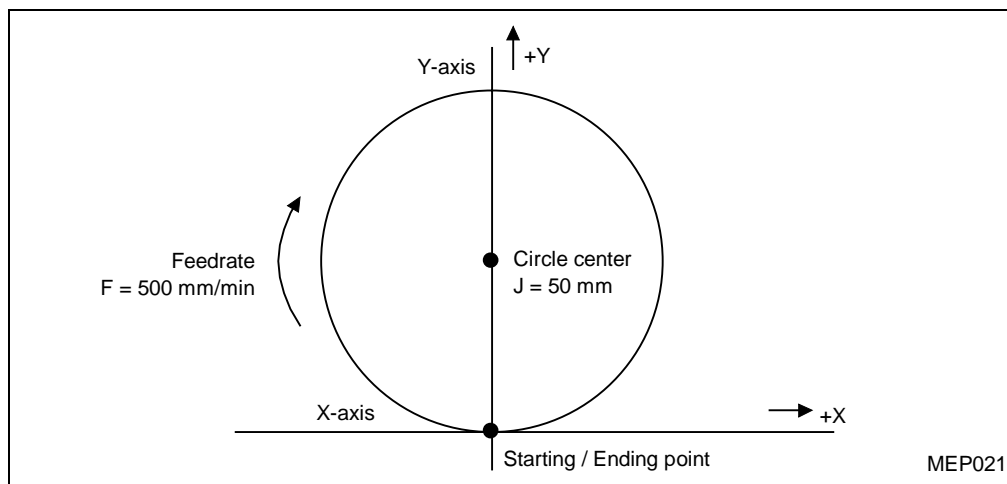
Interpolation of an arc that spans multiple quadrants can be defined with one block.

To perform circular interpolation, the following conditions are required:

- Plane selectionX-Y, Z-X, or Y-Z plane
- Rotational direction.....CW (G02) or CCW (G03)
- Arc ending point coordinates.....Given with addresses X, Y and Z.
- Arc center coordinates.....Given with addresses I, J and K (Incremental data input)
- Feed rateGiven with address F.

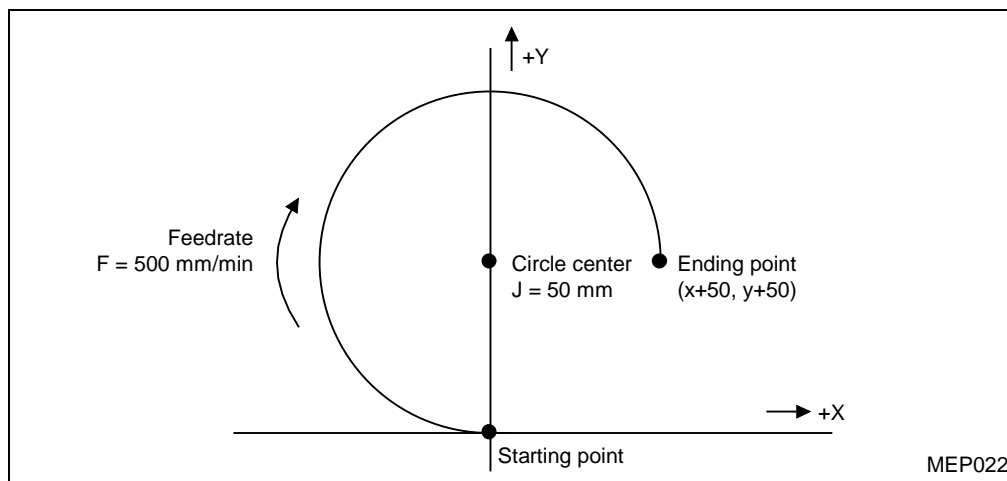
4. Sample programs

Example 1: Complete-circle command



G02 J50.000 F500

Example 2: Three-quarter circle command

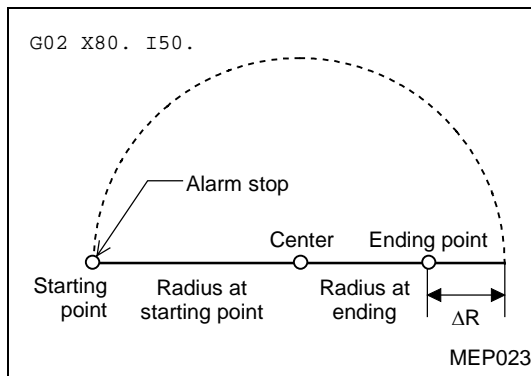


G91 G02 X50.000 Y50.000 J50.000 F500

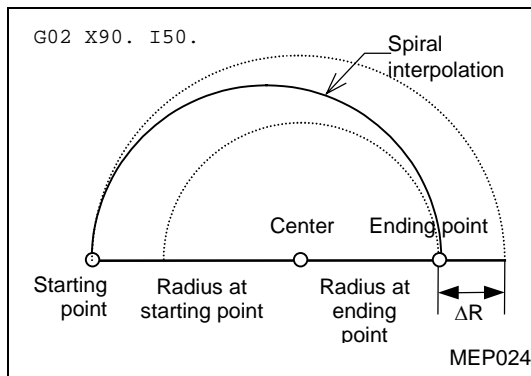
Remark: If the coordinates of the ending point are all omitted or set to the same position as the starting point, designating the center using addresses I, J and K will result in an arc of 360 degrees (complete circle).

Note: Either below will result if the starting-point radius and the ending-point radius are not the same.

- If error ΔR is larger than the parameter data, a program error will occur at the starting point of the arc (Alarm **817 INCORRECT ARC DATA**).



- If error ΔR is equal to or smaller than the parameter data, interpolation will take a spiral form heading for the programmed ending point of the arc.



Error ΔR can be set in user parameter **F19**. The examples shown above assume that excessively large parameter data is given to facilitate your understanding.

6-6 Radius-Designated Circular Interpolation: G02, G03

1. Function and purpos

Circular interpolation can be performed by designating directly the arc radius R as well as specifying conventional arc center coordinates (I, J, K).

2. Programming format

G02 (G03) Xx Yy Rr Ff

where x : X-axis coordinate of the ending point

y : Y-axis coordinate of the ending point

r : Radius of the arc

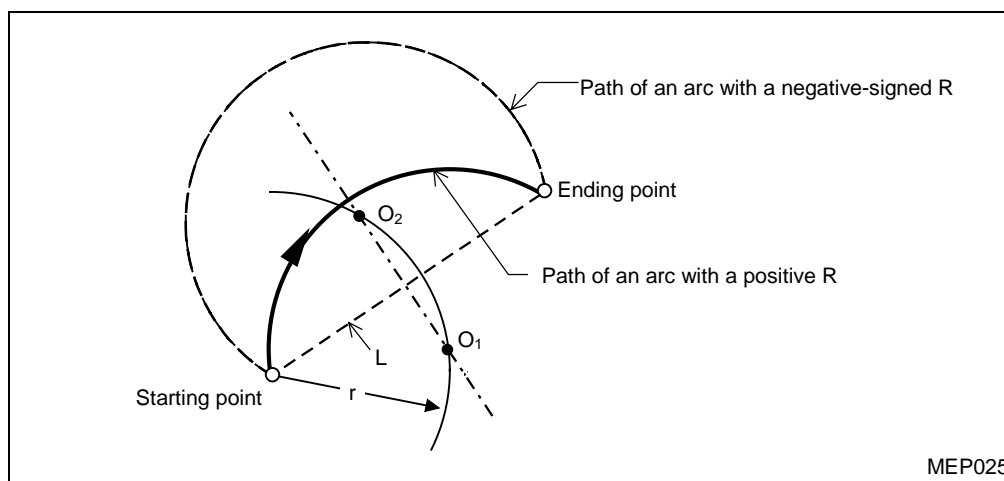
f : Feed rate

3. Detailed description

The arc center is present on the bisector perpendicular to the segment which connects the starting point and the ending point. The crossing point of the bisector and that circle of the designated radius r that has the center set at the starting point gives the center coordinates of the designated arc.

A semi-circle or smaller will be generated if R is a positive value.

An arc larger than a semi-circle will be generated if R is a negative value.



To use the radius-designated circular interpolation commands, the following requirement must be met:

$$\frac{L}{2 \cdot r} \leq 1$$

where L denotes the length of the line from the starting point to the ending point.

Alarm **818 MISSING CENTER (NO DATA)** will result if the requirement is not met.

If radius data and arc center data (I, J, K) are both set in the same block, then circular interpolation by radius designation will have priority in general.

For complete-circle interpolation (the ending point = the starting point), however, use center-designation method with addresses I, J and K, since the radius-specification command in this case will immediately be completed without any machine operation.

Select a plane for the radius-designated circular interpolation in the same manner as for center-specification method.

4. Sample programs

- G02 $XX_1 YY_1 Rr_1 Ff_1$ X-Y plane, radius-designated arc
- G03 $ZZ_1 XX_1 Rr_1 Ff_1$ Z-X plane, radius-designated arc
- G02 $XX_1 YY_1 Jj_1 Rr_1 Ff_1$ X-Y plane, radius-designated arc
(If radius data and center data (I, J, K) are set in the same block, circular interpolation by radius designation will have priority.)
- G17 G02 $Ii_1 Jj_1 Rr_1 Ff_1$ X-Y plane, center-designated arc
(Radius-specification is invalid for complete circle)

6-7 Helical Interpolation: G17, G18, G19 and G02, G03

1. Function and purpose

Command G02 or G03 with a designation for the third axis allows synchronous circular interpolation on the plane specified by plane-selection command G17, G18 or G19 with the linear interpolation on the third axis.

2. Programming format

G17 G02 $XX_1 YY_1 ZZ_1 Ii_1 Jj_1 Pp_1 Ff_1$
(G03)

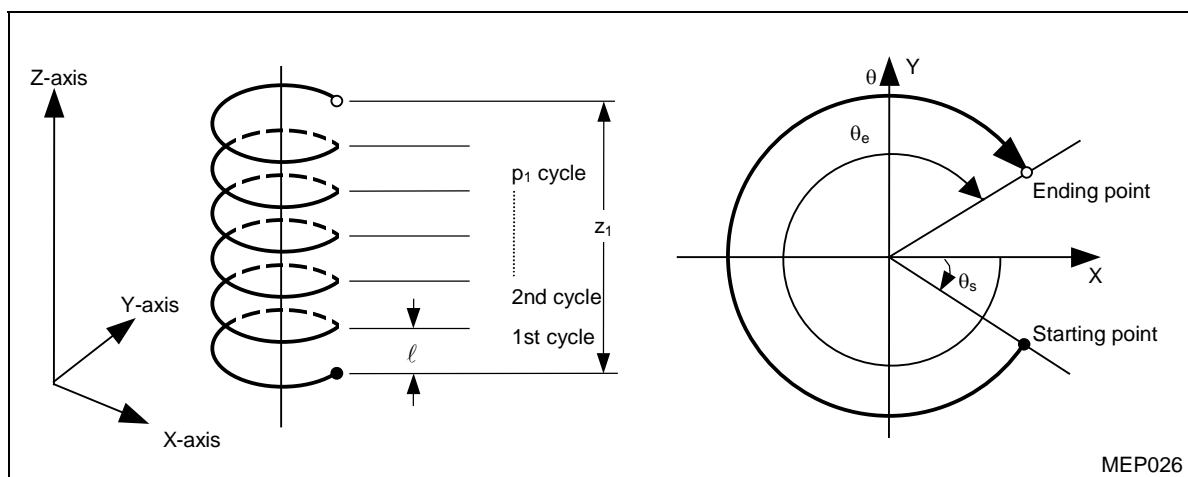
Feed rate
Number of pitches
Arc center coordinates
Linear axis ending point coordinate
Arc ending point coordinates

or

G17 G02 $XX_2 YY_2 ZZ_2 Rr_2 Pp_2 Ff_2$
(G03)

Feed rate
Number of pitches
Arc radius
Linear axis ending point
Arc ending point coordinates

3. Detailed description



- For helical interpolation, movement designation is additionally required for one to two linear axes not forming the plane for circular interpolation.
- The velocity in the resultant direction of X-, Y-, and Z-axial components must be designated as the feed rate F.

3. The pitch ℓ is calculated as follows:

$$\ell = \frac{z_1}{(2\pi \cdot p_1 + \theta) / 2\pi}$$

$$\theta = \theta_e - \theta_s = \tan^{-1} \frac{y_e}{x_e} - \tan^{-1} \frac{y_s}{x_s} \quad (0 \leq \theta < 2\pi)$$

where (xs, ys): relative coordinates of starting point with respect to the arc center

(xe, ye): relative coordinates of ending point with respect to the arc center

4. Address P can be omitted if the number of pitches is 1.

5. Plane selection

As with circular interpolation, the circular-interpolation plane for helical interpolation is determined by the plane-selection code and axis addresses. The basic programming procedure for helical interpolation is: selecting a circular-interpolation plane using a plane-selection command (G17, G18 or G19), and then designating the two axis addresses for circular interpolation and the address of one axis (perpendicular to the circular-interpolation plane) for linear interpolation.

- X-Y plane circular, Z-axis linear

After setting G02 (or G03) and G17 (plane-selection command), set the axis addresses X, Y and Z.

- Z-X plane circular, Y-axis linear

After setting G02 (or G03) and G18 (plane-selection command), set the axis addresses Z, X and Y.

- Y-Z plane circular, X-axis linear

After setting G02 (or G03) and G19 (plane-selection command), set the axis addresses Y, Z and X.

Plane selection with an additional axis is also possible as with circular interpolation.

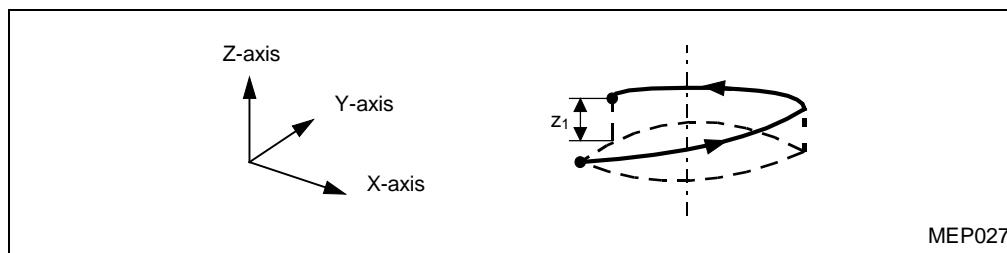
- U-Y plane circular, Z-axis linear

After setting G02 (or G03) and G17 (plane-selection command), set the axis addresses U, Y and Z.

In addition to the basic programming method described above, you can also use the special programming methods shown in the description of the sample programs below. See Section 6-4 "Plane Selection" for details on plane selection.

4. Sample programs

Example 1:



G17

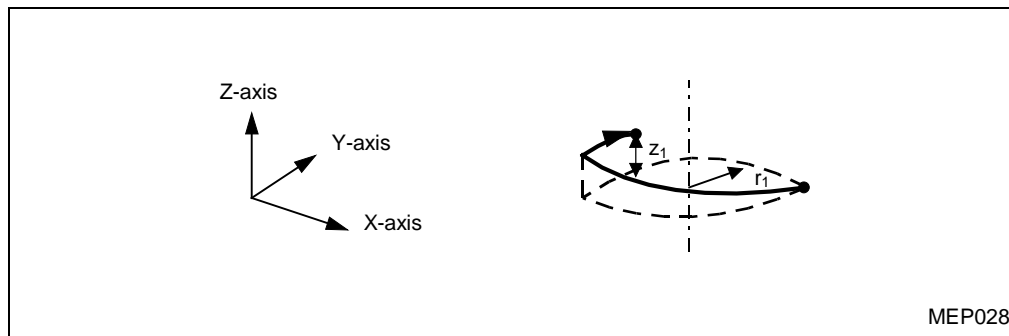
X-Y plane

...

G03 XX₁ YY₁ ZZ₁ II₁ JJ₁ P1 FF₁

X-Y plane circular, Z-axis linear

* Address P can be omitted if the number of pitches is 1.

Example 2:

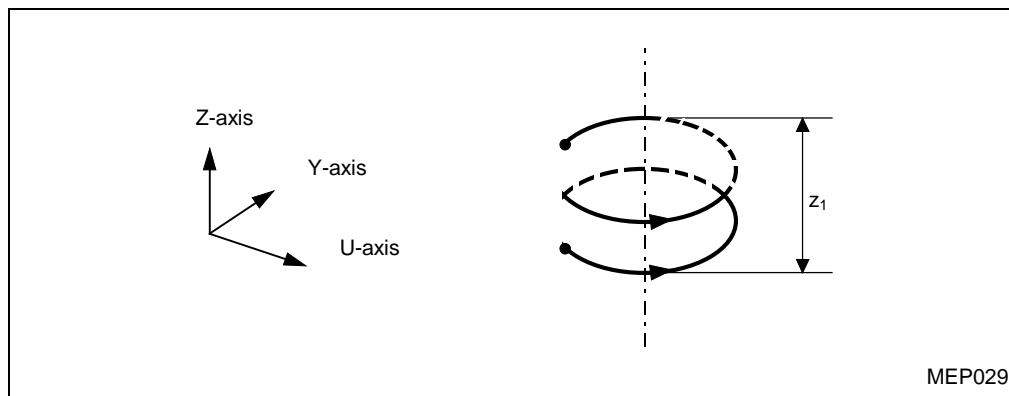
G17

X-Y plane

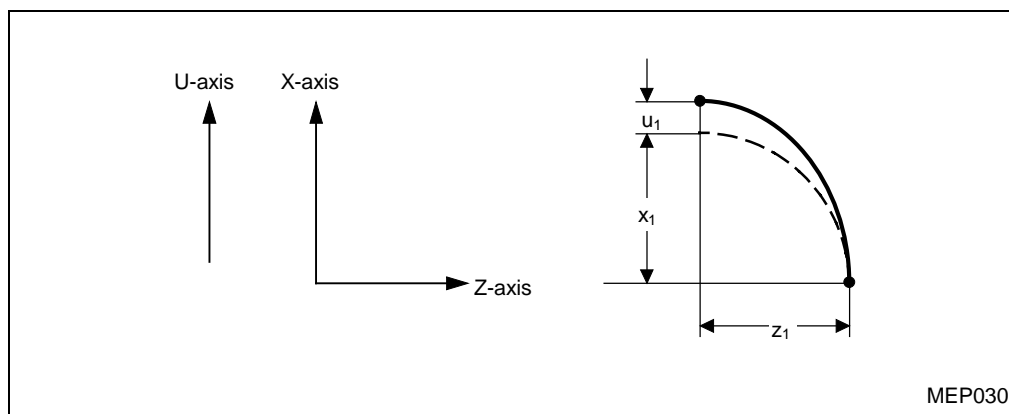
...

G02 xx_1 yy_1 zz_1 Rr_1 Ff_1

X-Y plane circular, Z-axis linear

Example 3:G17 G03 Uu_1 Yy_1 ZZ_1 Ii_1 Jj_1 $P2$ Ff_1

U-Y plane circular, Z-axis linear

Example 4:G18 G03 xx_1 Uu_1 ZZ_1 Ii_1 Jj_1 Kk_1 Ff_1

Z-X plane circular, U-axis linear

* If multiple axes of the same parallel-axes group are specified, circular interpolation will be performed with the data of the standard axis, and the data of the added axis will be used for linear interpolation.

Example 5: G18 G02 xx_1 Uu_1 Yy_1 ZZ_1 Ii_1 Jj_1 Kk_1 Ff_1

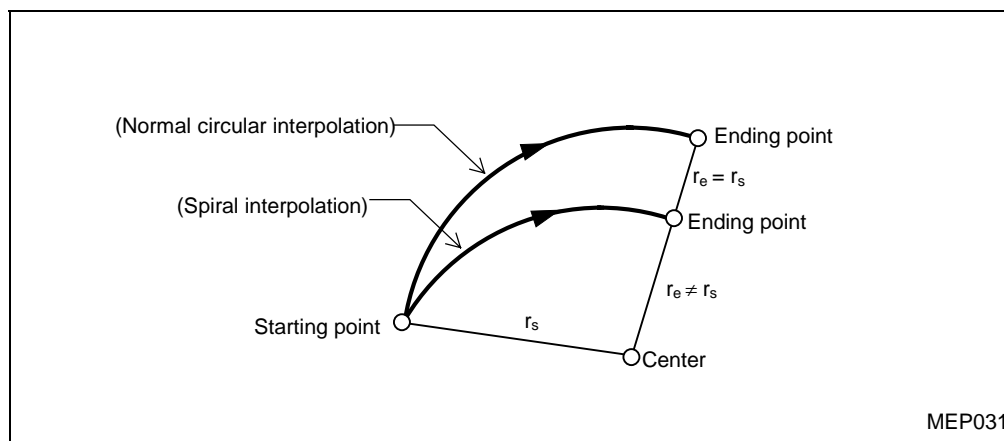
Z-X plane circular, U and Y-axes linear (J-code command ignored.)

* Two axes can be designated for linear interpolation.

6-8 Spiral Interpolation: G2.1, G3.1 (Option)

1. Function and purpose

Commands G2.1 and G3.1 provide such an interpolation that the starting and ending points are connected smoothly for an arc command where the radii of the both points differ from each other.



2. Programming format

G17 G2.1 (or G3.1) $\underbrace{X_p \ Y_p}_{\text{Arc center coordinates}} \ \underbrace{I \ J}_{\text{Arc ending point coordinates}} (\alpha) F \ P$

G18 G2.1 (or G3.1) $Z_p \ X_p \ K \ I (\alpha) F \ P$

G19 G2.1 (or G3.1) $Y_p \ Z_p \ J \ K (\alpha) F \ P$

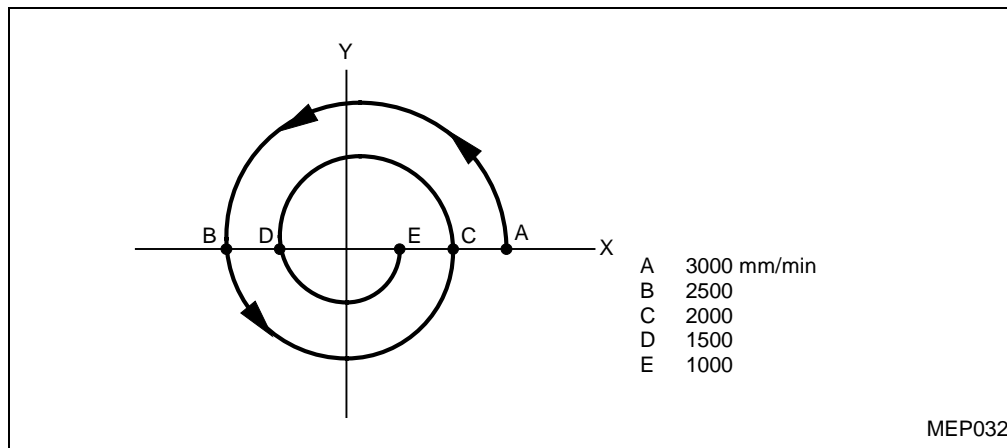
P : Number of pitches (revolutions) (P can be omitted if equal to 0.)
 α : Any axis other than circular interpolation axes (For helical cutting only)
F : Rate of feed along the tool path

3. Detailed description

1. Circular movement directions of G2.1 and G3.1 correspond with those of G02 and G03, respectively.
2. Radius designation is not available for spiral interpolation. (The starting and ending points must lie on the same arc for a radius designation.)
Note: When a radius is designated, this command will be regarded as a radius-designated circular interpolation.
3. Conical cutting or tapered threading can be done by changing the radii of the arc at its starting and ending points and designating a linear-interpolation axis at the same time.
4. Even for normal circular command G2 or G3, spiral interpolation will be performed if the difference between the radii of the starting point and the ending point is smaller than the setting of parameter **F19**.

Example: When the following program is executed, the feed rates for each of the points will be as shown in the diagram below.

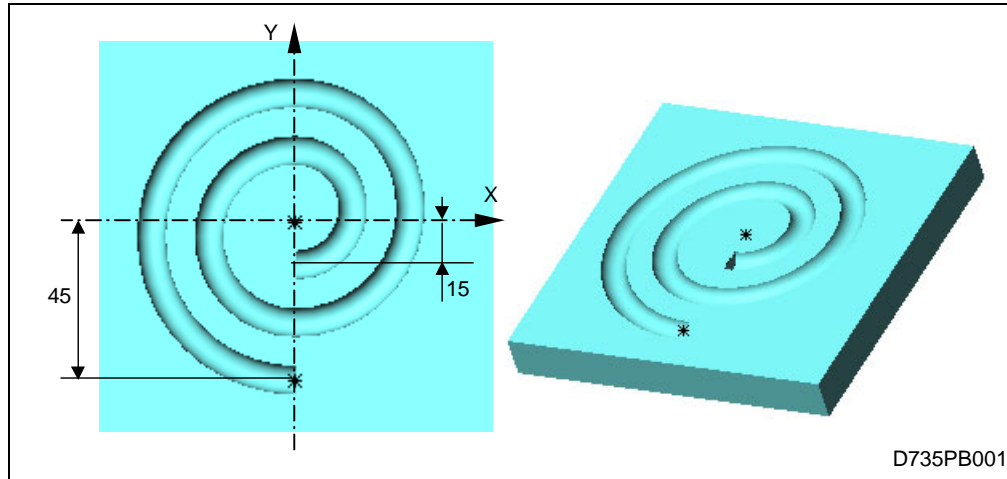
```
G91 G28 X0 Y0  
G00 Y-200.  
G17 G3.1 X-100. Y0 I-150. J0 F3000 P2  
M30
```



4. Sample programs

Example 1: Spiral cutting

Shown below is an example of programming for spiral contouring with incremental data input of the arc center (X = 0, Y = 45.0) and absolute data input of the arc ending point (X = 0, Y = -15.0).



G91 G28 Z0	Zero point return on the Z-axis
G80 G40	Fixed-cycle cancellation
T15 T00 M06	Tool change
G54.1 P40	Coordinate system setting
G90 G94 G00 X0 Y-45.0	Approach in the XY-plane to the starting point (0, -45.0)
G43 Z30.0 H01	Positioning on the Z-axis to the initial point
Z3.0	
S1500 M03	Normal rotation of the spindle
M50	Air blast ON
G01 Z-1.0 F150	Infeed on the Z-axis
G2.1 X0 Y-15.0 I0 J45.0 F450 P2	Command for spiral interpolation with arc ending point = (0, -15.0), arc center = (0, 0)*, and pitch = 2. * I- and J-values refer to increments to the starting point.
G00 Z3.0	Return on the Z-axis
M05 M09	Spindle stop and Air blast OFF
Z30.0	
M30	End of machining

The rate of feed at the starting point is 450 mm/min, as specified in the block of G2.1, and the rate of feed at the ending point can be calculated as follows:

(Ending point's radius/Starting point's radius) × Command value of the rate of feed.

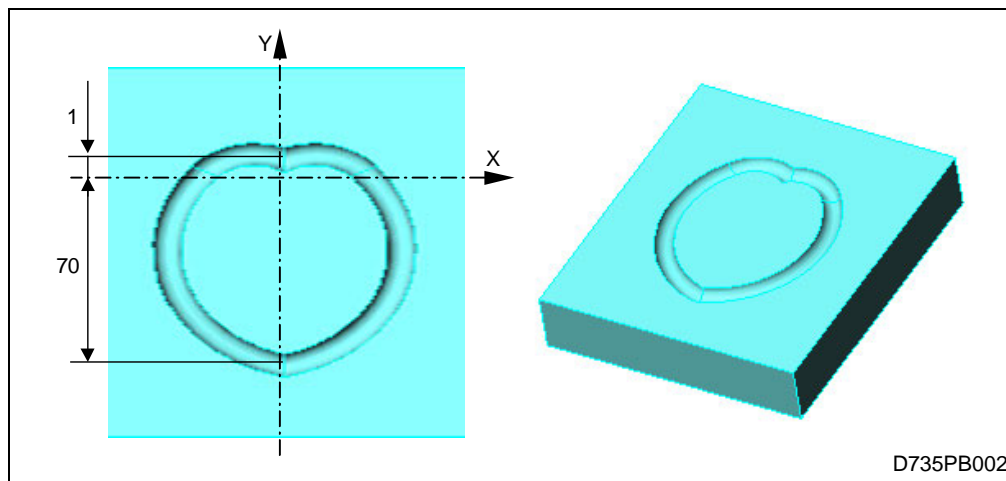
As the radius of the starting point = 45.0, that of the ending point = 15.0, and the command rate of feed (F) = 450, the rate of feed results in

$$(15.0/45.0) \times 450 = 150 \text{ mm/min}$$

at the ending point.

Note 1: Take care not to use radius designation (argument R) for spiral interpolation; otherwise a normal circular interpolation (by G02 or G03) will be executed.

Note 2: It is not possible to give the command for a spiral interpolation the starting and ending points of which should have different centers specified.

Example 2: Heart-shaped cam (by absolute data input)

```

G91 G28 Z0
G80 G40
T15 T00 M06
G54.1 P40
G90 G94 G00 X0 Y-70.0
G43 Z30.0 H01
S1500 M03
Z3.0
M50
G01 Z-1.0 F150
G2.1 X0 Y1.0 I0 J70.0 F450
X0 Y-70.0 I 0 J-1.0
G00 Z3.0
M05 M09
Z30.0
M30

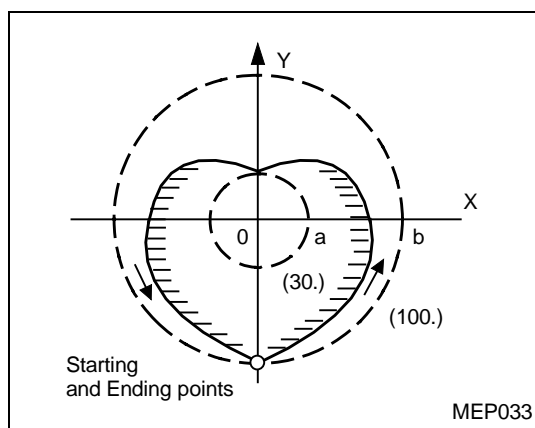
```

Zero point return on the Z-axis
 Fixed-cycle cancellation
 Tool change
 Coordinate system setting
 Approach in the XY-plane to the starting point (0, -70.0)
 Positioning on the Z-axis to the initial point
 Normal rotation of the spindle

 Air blast ON
 Infeed on the Z-axis
 Command for the left-hand half curve
 Command for the right-hand half curve
 Return on the Z-axis
 Spindle stop and Air blast OFF

 End of machining

Example 3: Heart-shaped cam (by incremental data input)



The difference $(b-a)$ between the radii of the starting point and ending point denotes a displacement for heart shape.

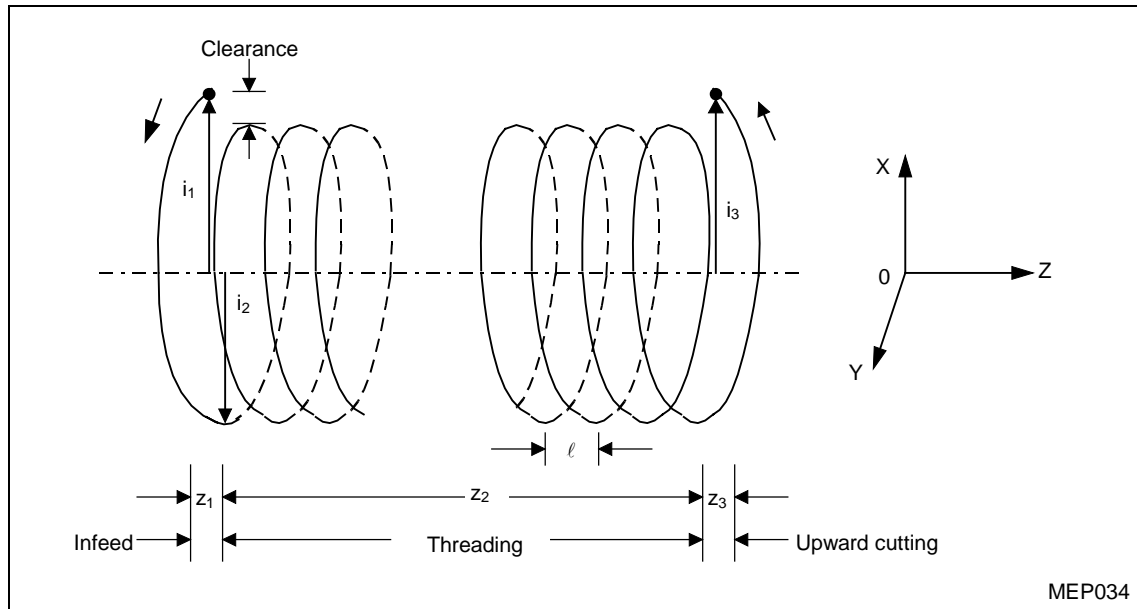
Use two blocks for programming separately the right-half and the left-half shape.

A sample program in incremental data input:

```
G3.1 Y130. J100. F1000..... (Right half)
      a+b    b
G3.1 Y-130. J-30 ..... (Left half)
      -a-b   -a
      a = 30. (Minimum arc radius)
      b = 100. (Maximum arc radius)
      a + b = 130. (Ending-point coordinate of the right half-circle)
      -a - b = -130. (Ending-point coordinate of the left half-circle)
```

Example 4: Large-size threading

To perform large-size threading, use three helical-interpolation blocks for programming separately infeed section, threading section and upward-cutting section. Spiral interpolation is required to designate the amounts of diameter clearance for both the infeed block and the upward-cutting block. (The starting and ending points are shifted through the designated clearance amounts from the circumference of threading section.)



G3.1 X- i_1 - i_2 Y0 ZZ₁ I- i_1 J0 Ff₁ (Infeed block, half-circle)

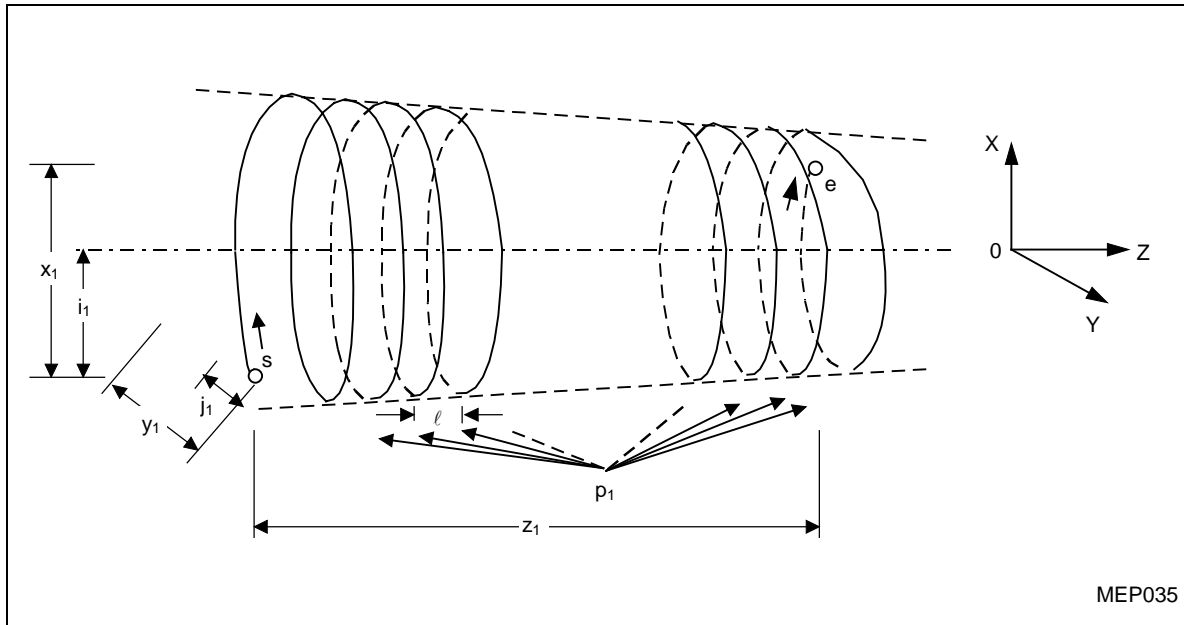
G03 X0 Y0 ZZ₂ I i_2 J0 Pp₂ (Threading block, complete circle)

G3.1 X i_2 + i_3 Y0 ZZ₃ I i_2 J0 (Upward-cutting block, half-circle)

- * The number of pitches, p_2 , in the threading block is given by dividing the stroke z_2 by the pitch ℓ . Note that the value p_2 must be an integer.

Example 5: Tapered threading

As shown in the figure below, tapered helical cutting that begins at any angle can be performed.



Data with addresses X, Y and Z must be the increments x_1 , y_1 and z_1 respectively, from the starting point s to the ending point e; data of I and J must be the increments i_1 and j_1 respectively, from the starting point s to the circular center, and data of P must be equal to the number of pitches p_1 .

G3.1 XX₁ YY₁ ZZ₁ II₁ JJ₁ PP₁ FF₁

The amount of taper t and the pitch ℓ are calculated as follows:

$$t = \frac{2(re - rs)}{x_1}$$

where $rs = \sqrt{i_1^2 + j_1^2}$, $re = \sqrt{(x_1 - i_1)^2 + (y_1 - j_1)^2}$;

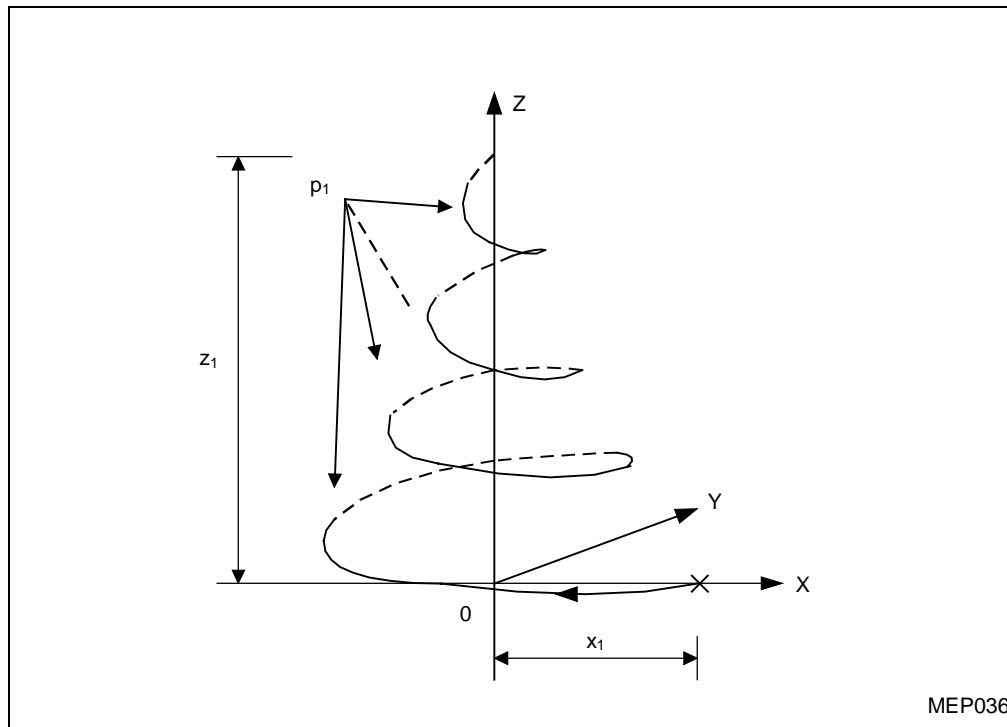
$$\ell = \frac{z_1}{(2\pi \cdot p_1 + \theta) / 2\pi}$$

$$\text{where } \theta = \theta_e - \theta_s = \tan^{-1} \frac{j_1 - y_1}{i_1 - x_1} - \tan^{-1} \frac{-j_1}{-i_1}$$

where rs and re denote the radii at the starting point and the ending point respectively, and qs and qe denote the angles at the starting point and the ending point respectively.

Example 6: Conical cutting

Conical cutting is an application of tapered threading, and have its starting or ending point on the center line. Tapering results from gradually increasing or decreasing the arc diameter. The pitch is determined by z_1/p_1 .



MEP036

G2.1 X-x₁ Y0 ZZ₁ I-x₁ Pp₁ Ff₁

x₁ : Radius of the base

z₁ : Height

p₁ : Number of pitches

f₁ : Feed rate

Note: Use the **TRACE** display to check the tool path during spiral interpolation.

6-9 Virtual-Axis Interpolation: G07

1. Function and purpose

Specify with G07 code one of the two circular-interpolation axes for helical or spiral interpolation with synchronous linear interpolation as a virtual axis (a pulse-distributed axis without actual movement), and an interpolation on the plane defined by the remaining circular axis and the linear axis can be obtained along the sine curve which corresponds with the side view of the circular interpolation with synchronous linear interpolation.

2. Programming format

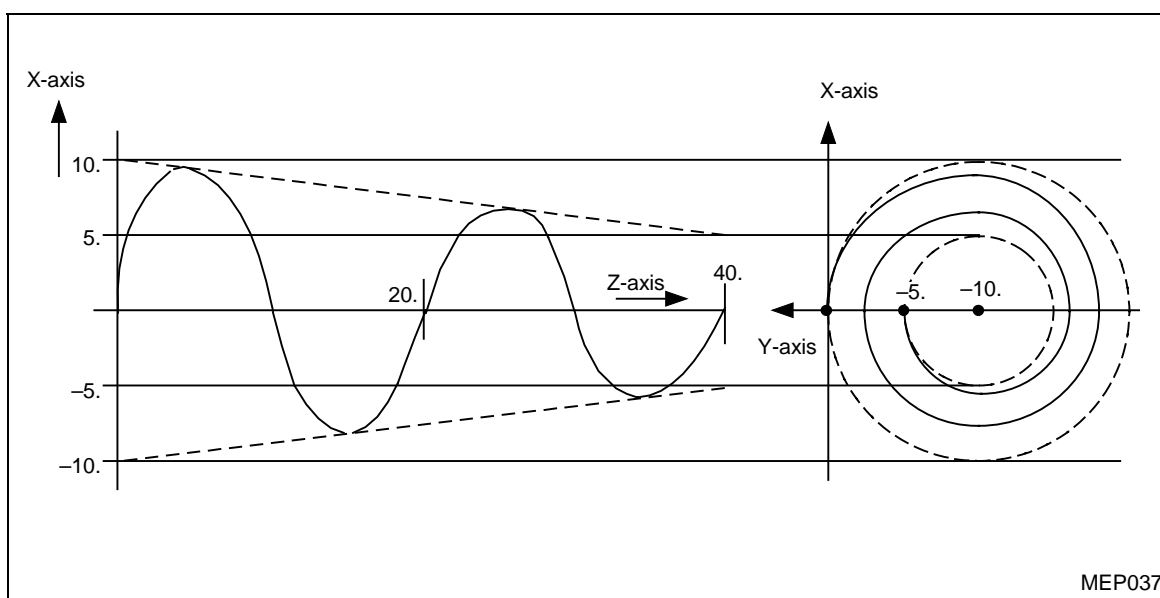
G07 α0	To set a virtual axis
⋮	} To interpolate with the virtual axis
G07 α1	

3. Detailed description

- Only helical or spiral interpolation can be used for the virtual-axis interpolation.
- In the program section from G07α0 to G07α1, the “alpha” axis is processed as a virtual axis. If, therefore, the alpha axis is included independently in this section, the machine will remain in dwell status until pulse distribution to the virtual axis is completed.
- The virtual axis is valid only for automatic operation; it is invalid for manual operation.
- Protective functions, such as interlock, stored stroke limit, etc., are valid even for the virtual axis.
- Handle interruption is also valid for the virtual axis. That is, the virtual axis can be shifted through the amount of handle interruption.

4. Sample program

G07 Y0	Sets the Y-axis as a virtual axis.
G17G2.1X0Y-5.I0J-10.Z40.P2F50	Sine interpolation on X-Z plane
G07 Y1	Resets the Y-axis to an actual axis.



6-10 Spline Interpolation: G06.1 (Option)

1. Function and purpose

The spline interpolation automatically creates a curve that smoothly traces specified points, and thus enables a high-speed and high-accuracy machining for free shapes along smoothly curved tool path.

2. Programming format

G06.1 Xx₁ Yy₁

3. Detailed description

A. Setting and cancellation of spline interpolation mode

The spline interpolation mode is set by the preparatory function G06.1, and cancelled by another Group 01 command (G00, G01, G02 or G03).

Example 1:

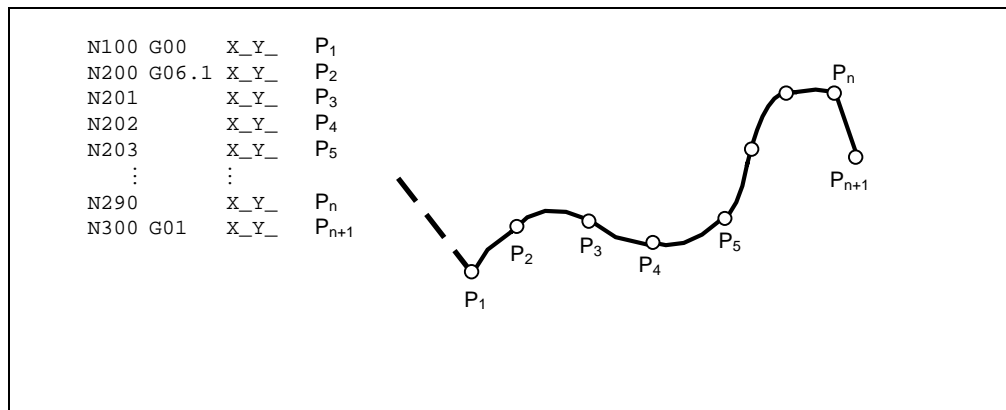


Fig. 6-2 Interpolated line by spline interpolation

In the above example, the spline interpolation is activated at N200 (block for movement from P₁ to P₂) and it is cancelled at N300. Therefore, a spline curve is created for a group of ending points from P₁ to P_n, and interpolation is applied along the created curve.

For creating a spline interpolation curve, it is generally required to specify two or more blocks (at least three points to be traced) in the mode. If the spline interpolation mode is set just for one block, the path to the ending point of the block is interpolated in a straight line.

Example 2:

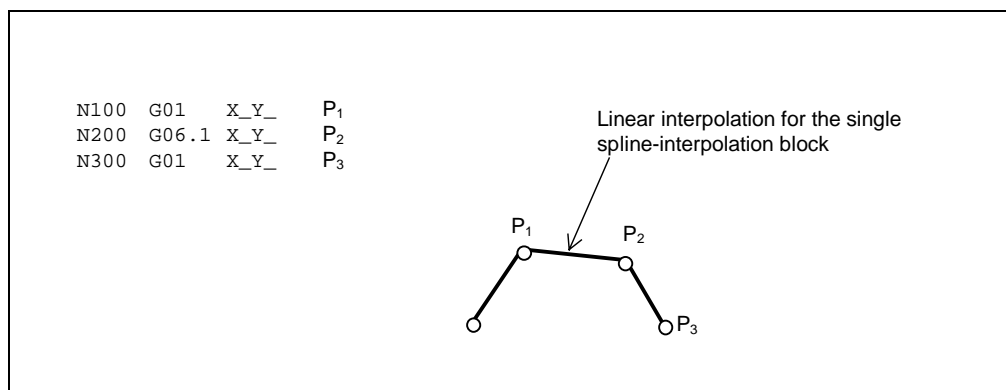


Fig. 6-3 Spline interpolation applied to a single block

B. Division of spline curve in spline-interpolation mode

The spline interpolation mode generally creates a continuous curve that smoothly connects all specified points from the beginning of the mode to the end of it. However, the spline curve is divided into two discontinuous curves as often as one of the following conditions is satisfied:

- When the angle between linear movement lines of two neighboring blocks is beyond the spline-cancel angle,
- When the movement distance of a block exceeds the spline-cancel distance, or
- When there is a block without any movement command in the spline-interpolation mode.

1. When the relative angle of two neighboring blocks is beyond the spline-cancel angle

Spline-cancel angle Parameter **F101**

As to the sequence of points $P_1, P_2, P_3, \dots, P_n$ in a spline interpolation mode, when the angle θ_i made by two continuous vectors $\overrightarrow{P_{i-1}P_i}$ and $\overrightarrow{P_iP_{i+1}}$ is larger than **F101**, the point P_i is regarded as a corner. In that event, the point group is divided into two sections of P_1 to P_i and P_i to P_n at P_i , and spline curve is individually created for each section.

When the spline-cancel angle is not set (**F101** = 0), this dividing function is not available.

Example 1: **F101** = 80 deg

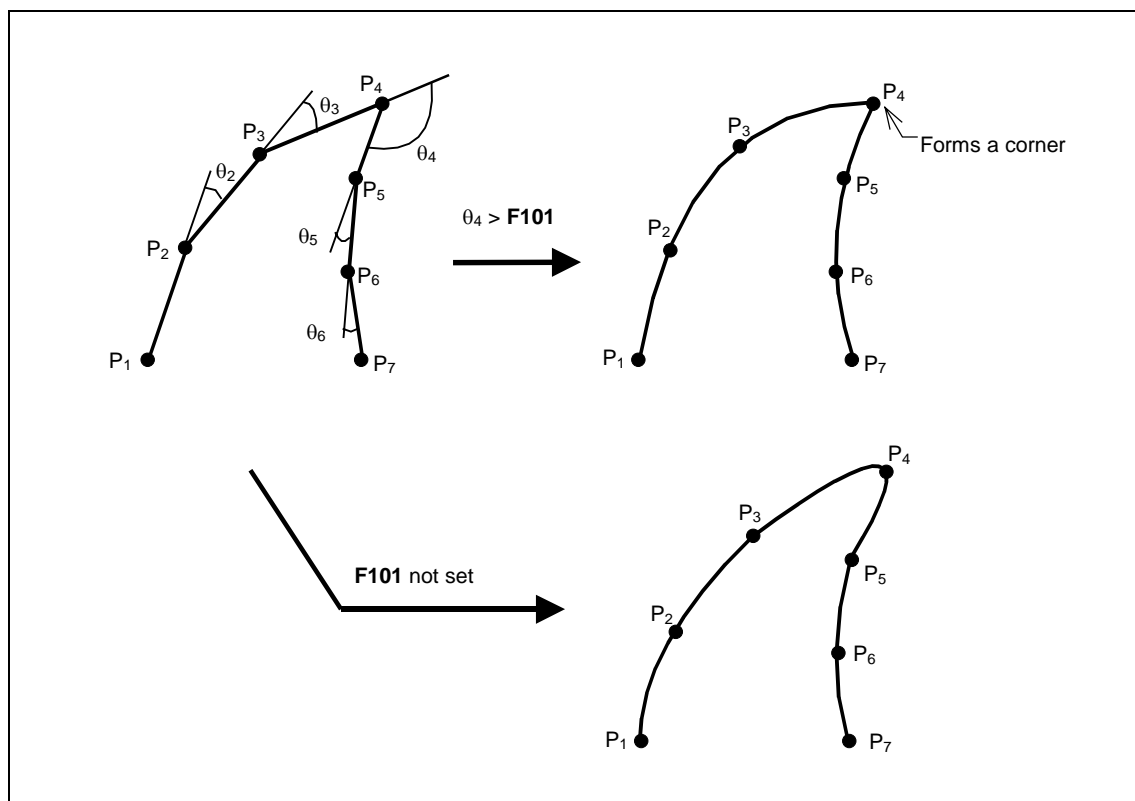


Fig. 6-4 Spline cancel depending on angle

When there are more than one point where $\theta_i > \mathbf{F101}$, such points are treated as corners to divide the point group and multiple spline curves are created for respective sections.

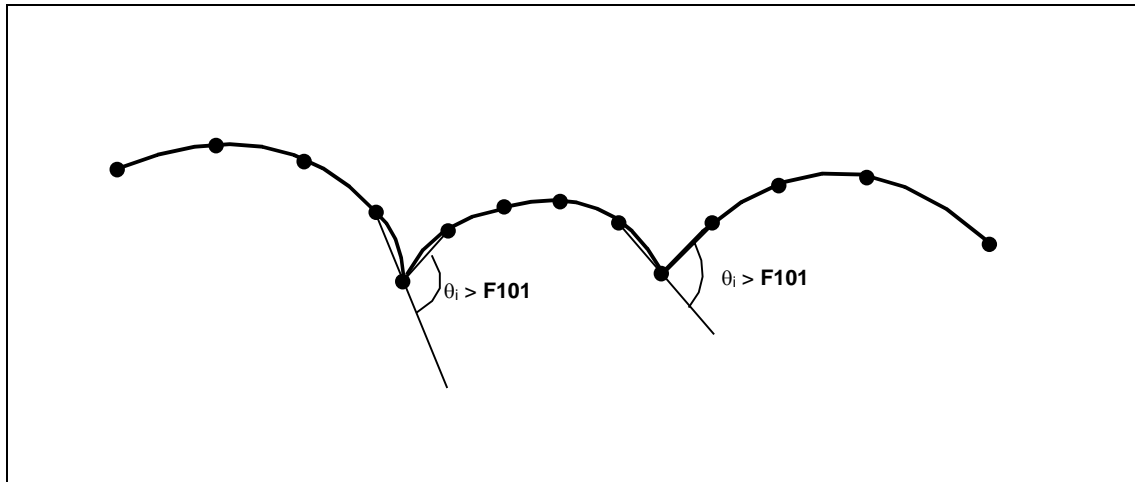


Fig. 6-5 Multiple-cornered spline curve depending on angle

When any two corner points (where $\theta_i > \mathbf{F101}$) successively exist, the block for the second point is automatically set under control of linear interpolation. Therefore, it can be omitted to specify G01 code in each intermediate block of pick feed, for example, during 2.5-dimensional machining, which considerably simplifies the programming.

Example 2: $\mathbf{F101} < 90$ (deg)

In the following program, the angle of the Y-directional pick feed to the X-Z plane (of spline interpolation) is always 90° . If $\mathbf{F101}$ is set slightly smaller than 90° , spline interpolation is automatically cancelled in the pick-feed blocks (N310, N410, ...), which are then linearly interpolated each time. If no value is set for $\mathbf{F101}$, it is required to specify G-codes parenthesized in the program below to change the mode of interpolation.

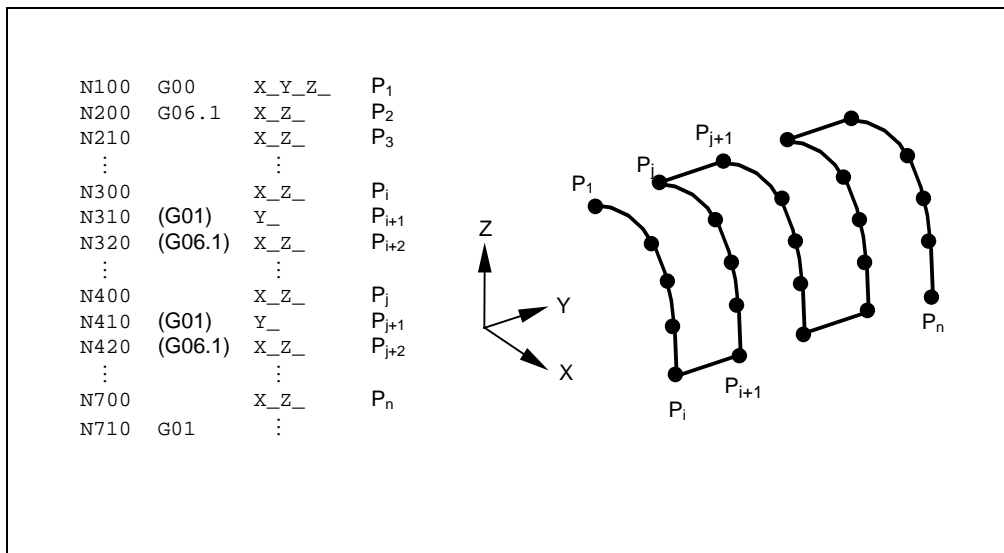


Fig. 6-6 Linear interpolation for pick feed in spline-interpolation mode

2. When the movement distance of a block exceeds the spline-cancel distance

Spline-cancel distance Parameter **F100**

As to the sequence of points $P_1, P_2, P_3, \dots, P_n$ in a spline interpolation mode, when the length $\overline{P_i P_{i+1}}$ of the vector $\overrightarrow{P_i P_{i+1}}$ is longer than **F100**, the block for point P_{i+1} is automatically set under control of linear interpolation, while the preceding and succeeding sections P_1 to P_i and P_{i+1} to P_n are individually interpolated in spline curves.

In this case, the inclination of the tangent vector at P_i (at the end of spline P_1 to P_i) and the inclination of the tangent vector at P_{i+1} (at the beginning of spline P_{i+1} to P_n) do not correspond to that of the line segment $\overline{P_i P_{i+1}}$ in general.

When the spline-cancel distance is not set (**F100** = 0), this dividing function is not available.

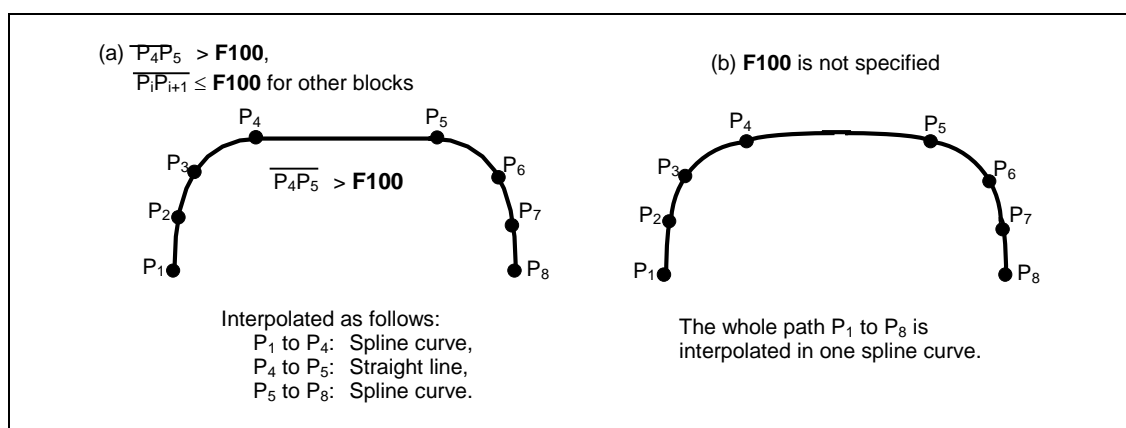


Fig. 6-7 Spline cancel depending on movement distance of a block

When there are more than one block where $\overline{P_i P_{i+1}} > \mathbf{F100}$, all those blocks will individually undergo the linear interpolation.

3. When there is a block without any movement command in the spline-interpolation mode
- Any block without movement command temporarily cancels the spline interpolation, and the sections before and after such a block will independently be spline-interpolated.

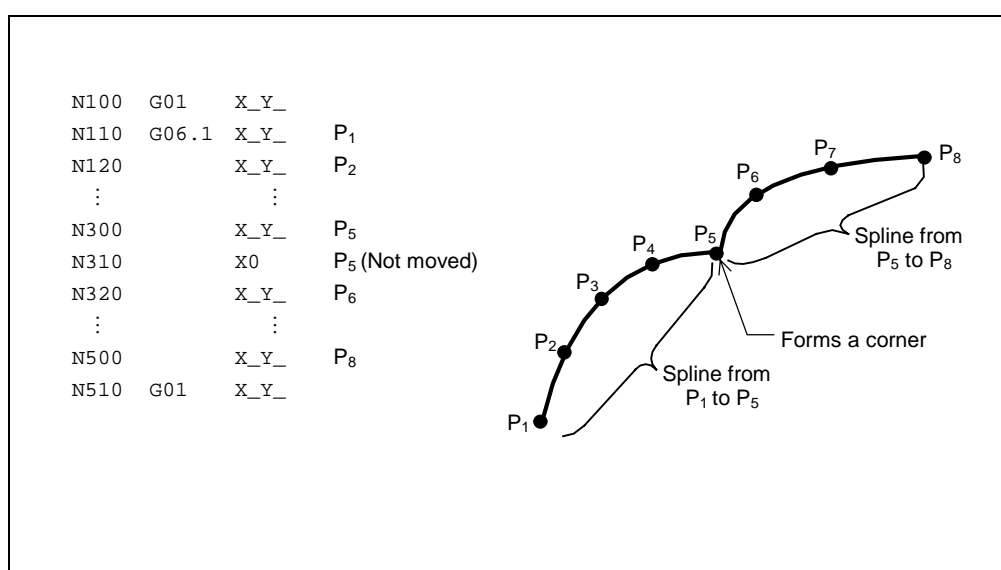


Fig. 6-8 Spline cancel by a block without movement command

C. Fine spline function (curved shape correction)

The fine spline function works with spline interpolation and automatically corrects the shape of a spline curve, as required, to make the path of the curve smoother.

More specifically, the fine spline function works in the following two cases:

- The case that the curve errors in blocks are significant
- The case that an unusually short block exists (automatic correction in this case is referred to as fairing.)

Automatic correction in the above cases is explained below.

1. Automatic correction for significant curve errors in blocks

When the curve data in CAD undergoes micro-segmentation with CAM, approximation using a polygonal line is usually executed with a curve tolerance (chord error) of about 10 microns. At this time, if any inflection points are included in the curve, the micro-segment block including the inflection points may increase in length (see $\overline{P_3 P_4}$ in the figure below)

Also, if the length of this block becomes unbalanced against those of the immediately preceding and succeeding blocks, the spline curve in this zone may have a significant error with respect to the original curve.

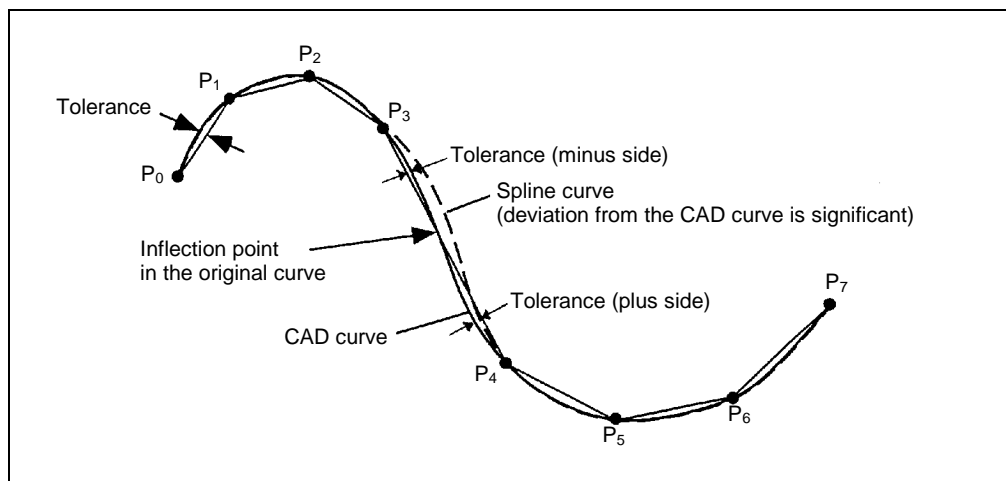


Fig. 6-9 Spline curve having a significant chord error (inflection points present)

This function detects the sections whose chord errors in the curve due to the presence of inflection points become significant, and corrects the shape of the spline curve in that zone automatically so that the chord errors in the curve fall within the data range of the specified parameter.

Curve error 1 Parameter **F102**

If a block in the spline interpolation mode is judged to have inflection points in the spline curve and the maximum chord error of the spline curve from the segment is greater than the value of **F102**, the shape of that spline curve will be corrected for a maximum chord error not exceeding the value of **F102**.

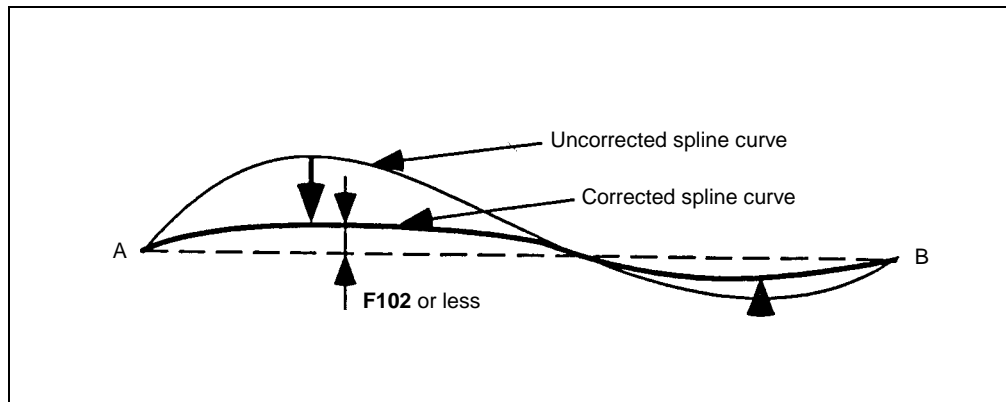


Fig. 6-10 Shape correction 1 for spline curve

The shape of a curve can also be corrected if the chord error in the spline curve increases due to an imbalance in the lengths of adjoining blocks occurs for any reasons other than the presence of inflection points or for other reasons.

Curve error 2 Parameter **F104**

If a blocks in the spline interpolation mode is judged to have no inflection points in the spline curve and the maximum chord error in the spline curve and block is greater than the value of **F104**, the shape of that spline curve will be corrected for a maximum chord error not exceeding the value of **F104**.

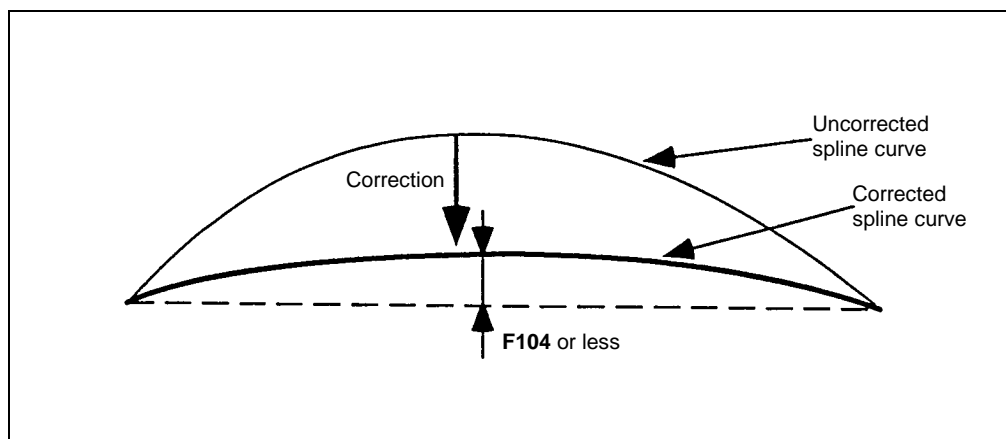


Fig. 6-11 Spline curve having a significant chord error (no inflection points)

- Remark 1:** In all types of spline curve correction, the curve correction function works only for the corresponding block. Therefore, the tangential vectors at the boundaries with the immediately preceding and succeeding blocks become discontinuous.
- Remark 2:** If parameter **F102** is set to 0, all blocks regarded as including inflection points will become linear. If parameter **F104** is set to 0, all blocks regarded as including no inflection points will become linear.
- Remark 3:** Curved-shape correction based on parameter **F102** or **F104** usually becomes necessary when adjoining blocks are unbalanced in length. If the ratio of the adjoining block lengths is very large, however, spline interpolation may be temporarily cancelled between the blocks prior to evaluation of the chord error.

2. Automatic correction of the spline curve in an unusually short block (Fairing)

When CAD data is developed into micro-segments by CAM, a very small block may be created in the middle of the program because of internal calculation errors. Such a block is often created during creation of a tool diameter offset program which requires convergence calculation, in particular. Since this unusually small block usually occurs at almost right angles to the direction of the spline curve, this curve tends not to become smooth.

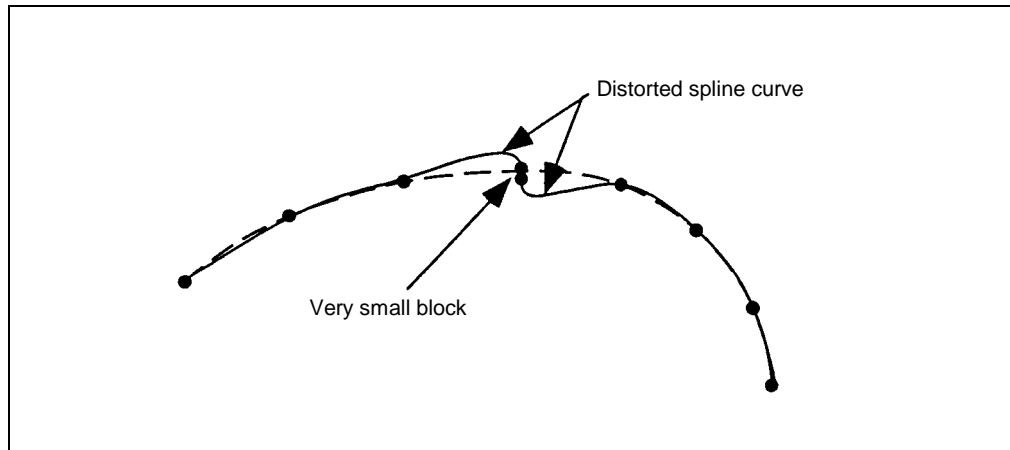


Fig. 6-12 Distortion of a spline curve due to the effects of a very small block

If it detects such an extremely small block during spline interpolation, the shape correction function will remove that block and then connect the preceding and succeeding blocks directly (this is referred to as fairing) to create a smooth spline curve free from distortion.

Block fairing length Parameter **F103**

Assume that the length of the i -th block in spline interpolation mode is taken as l_i and that the following expressions hold:

$$l_{i-1} > \mathbf{F103} \times 2$$

$$l_i \leq \mathbf{F103}$$

$$l_{i+1} > \mathbf{F103} \times 2$$

In the above case, the ending point of the $(i-1)$ -th block and the starting point of the $i+1$ block are moved to the mid-point of the i th block and as a result, the i th block is deleted. Spline interpolation is executed for the sequence of points that has thus been corrected.

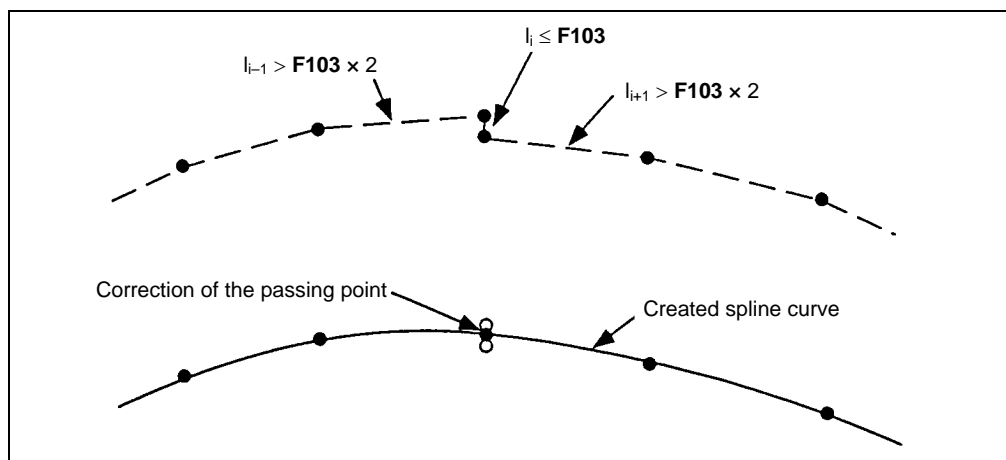


Fig. 6-13 Correction of spline curve passing points by fairing

Assume that the first block in spline interpolation mode is very small and that the following expressions hold:

$$l_1 \leq F103$$

$$l_2 > F103 \times 2$$

In the above case, the starting point of the second block is changed to that of the first block and as a result, the first block is deleted.

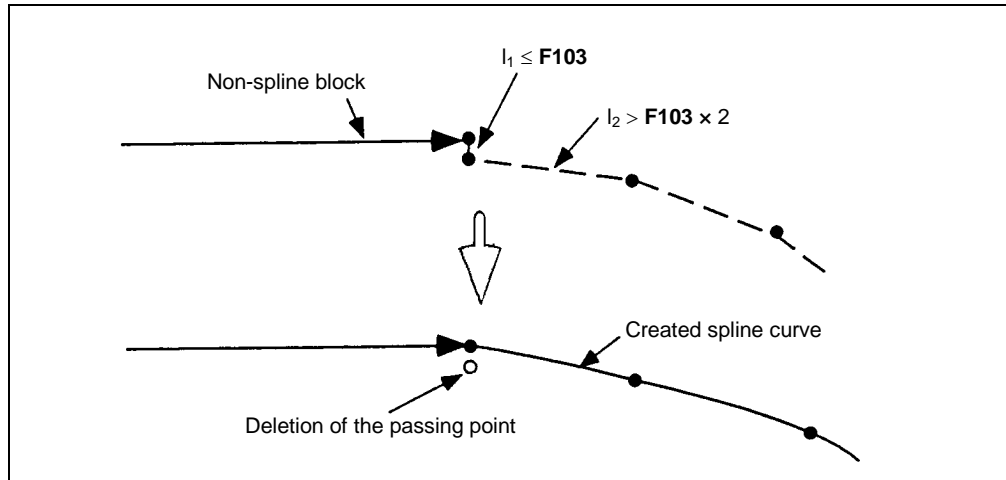


Fig. 6-14 Fairing at the starting point of a spline curve

Assume that the last block in spline interpolation mode is very small and that the following expressions hold:

$$l_{n-1} > F103 \times 2$$

$$l_n \leq F103$$

In the above case, the ending point of the (n-1)-th block is changed to that of the nth block and as a result, the nth block is deleted.

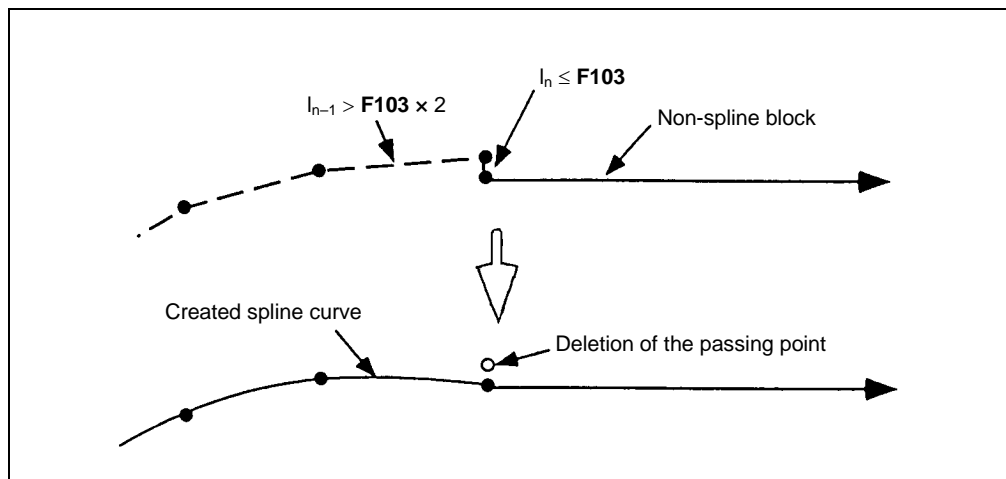


Fig. 6-15 Fairing at the ending point of a spline curve

This function is executed preferentially over the curve slitting function based on the angle of spline cancellation.

D. Feed-rate limitation in spline-interpolation mode

The modal cutting feed rate F remains valid in general for the spline interpolation; however, if the feed rate should be kept constant, it would yield excessively high acceleration at portions where the curvature is big (the curvature radius is small) as shown in Fig. 6-16.

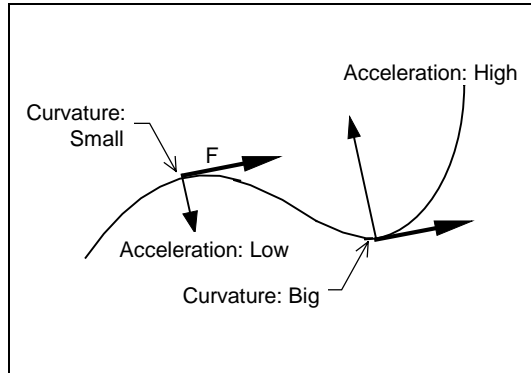


Fig. 6-16 Change of acceleration depending on curvature

In the spline-interpolation mode of our NC, the feed rate can be controlled so that it does not exceed the allowable limit, calculated from the related parameters, for pre-interpolation acceleration.

To obtain an appropriate feed rate for each block of spline interpolation, the limit feed rate F' is calculated by the equation [1] shown below where the smaller between two radii R_s (curvature radius at the starting point of the block) and R_e (curvature radius at its ending point) will be regarded as the reference radius R for the block. The modal feed rate F will then be temporarily overridden by F' for the respective block if $F > F'$, so that the whole spline curve can be interpolated block-by-block at the appropriate feed rate according to the curvature radius.

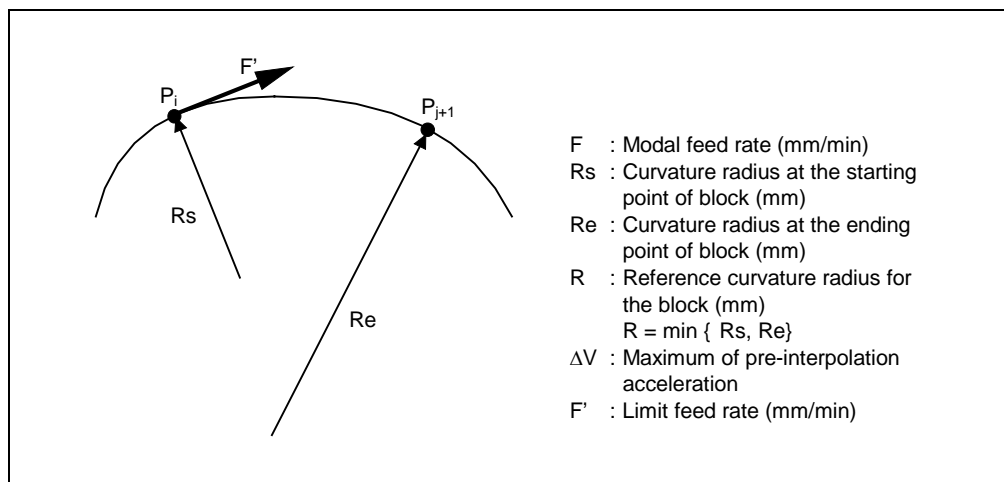


Fig. 6-17 Feed-rate limitation for spline interpolation

$$F' = \sqrt{R \times \Delta V \times 60 \times 1000} \quad \dots\dots [1]$$

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

E. Spline interpolation during tool-diameter offset

The spline interpolation can be performed during tool-diameter offset as follows.

1. Tool-diameter offset (2-dimensional)

Shown in Fig. 6-18 is an example that the command route is straight in the section P_0P_1 , polygonal line in the section $P_1P_2 \dots P_n$ that is the object of spline interpolation, and straight in the section P_nP_{n+1} . The interpolation route with tool-diameter offset is created by the following procedure.

- 1) In the first step is created a polygonal line $P_0'P_1'P_2' \dots P_n'P_{n+1}'$ that is offset by the tool-diameter offset value r compared with the original polygonal line $P_0P_1P_2 \dots P_nP_{n+1}$.
- 2) Next, a point P_i'' where $\overline{P_iP_i''} = r$ on the vector $\overrightarrow{P_iP_i'}$ is determined for all the pass points P_i ($i = 2, 3, \dots, n-1$) other than the starting point P_1 and the ending point P_n of the spline curve.
- 3) Spline interpolation is now conducted for the polygonal line $P_1'P_2''P_3'' \dots P_{n-1}''P_n'$ and the curve thus created will act an offset path of tool center for the commanded spline curve.

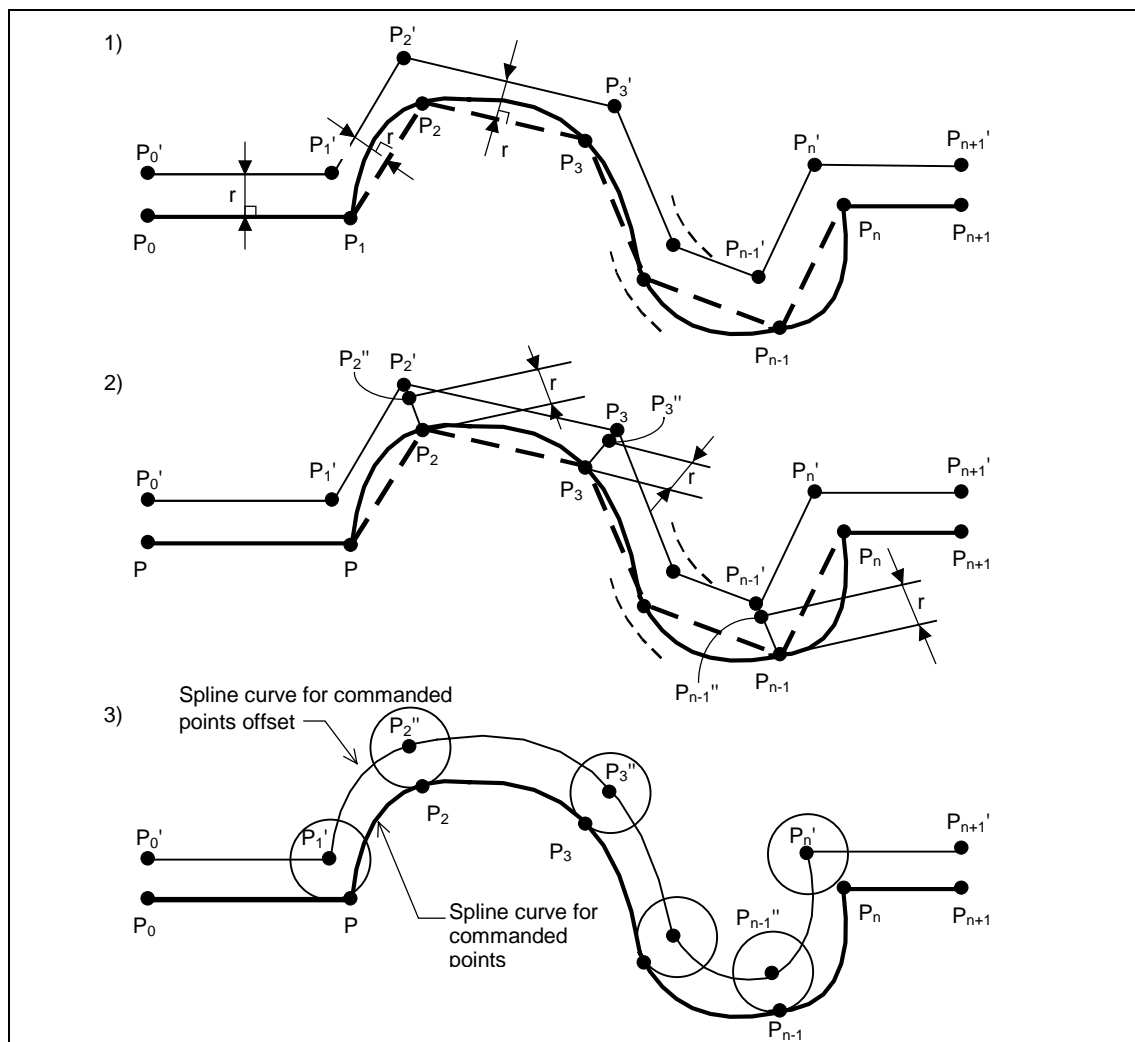


Fig. 6-18 Spline interpolation during tool-diameter offset

The spline curve created in the above-mentioned procedure is not the strict offset, indeed, of the commanded spline curve, but an approximation of it.

2. 3-dimensional tool-diameter offset

In the 3-dimensional tool-diameter offset, each point defined with programmed coordinates is first offset through the tool radius “r” in the direction of the specified normal vector (i, j, k) and then, the serial points thus offset in the spline-interpolation section are connected in a smooth curve, which will act as the path of tool-radius center for the 3-dimensional spline interpolation.

F. Others

1. The spline interpolation targets the basic coordinate axes of X, Y and Z; however, it is not always required to specify objective axes on commanding the spline interpolation. Moreover, the spline-interpolation command code (G06.1) can be given in a block without any movement command.

Example:

N100	G06.1	X_Y_Z0	→	N100	G06.1	X_Y_
N200		X_Y_Z_		N200		X_Y_Z_
N300		X_Y_Z_		N300		X_Y_Z_
⋮		⋮		⋮		⋮
N100	G06.1	F_ (← No movement commands)				
N200		X_Y_Z_				
N300		X_Y_Z_				
⋮		⋮				

2. The spline-interpolation command (G06.1) falls under the G-code group 01.
3. In the single-block operation mode, the spline interpolation is cancelled and all the respective blocks will individually undergo the linear interpolation.
4. In tool-path check, the blocks of spline interpolation are not actually displayed in a spline curve but in a polygonal line that connects linearly the respective points, which, in case of tool-diameter offset, will have been offset in the same manner as described in the foregoing article E.
5. During spline interpolation, when feed hold is executed, the block for which the feed hold function has been executed will be interpolated, at the beginning of the restart operation along the spline curve existing before the feed hold function was executed, and then the spline curve in the next block onward will be re-created and interpolation executed.
6. Although spline interpolation can also be executed in the high-speed machining mode (G05P2 mode), curve shape correction by fairing becomes invalid in the G05P2 mode.

6-11 NURBS Interpolation: G06.2 (Option)

1. Function

The NURBS interpolation function provides interpolation by performing NURBS-defined CNC-internal computations on the command issued from the CAD/CAM system in the NURBS format. With this optional function, a very smooth interpolation path can be obtained since the interpolation process is performed directly without dividing a NURBS-formatted free-form curve into minute line segments.

2. Definition of the NURBS curve

NURBS, short for Non-Uniform Rational B-Spline, provides rationalization of the B-spline function.

The NURBS curve is defined as follows:

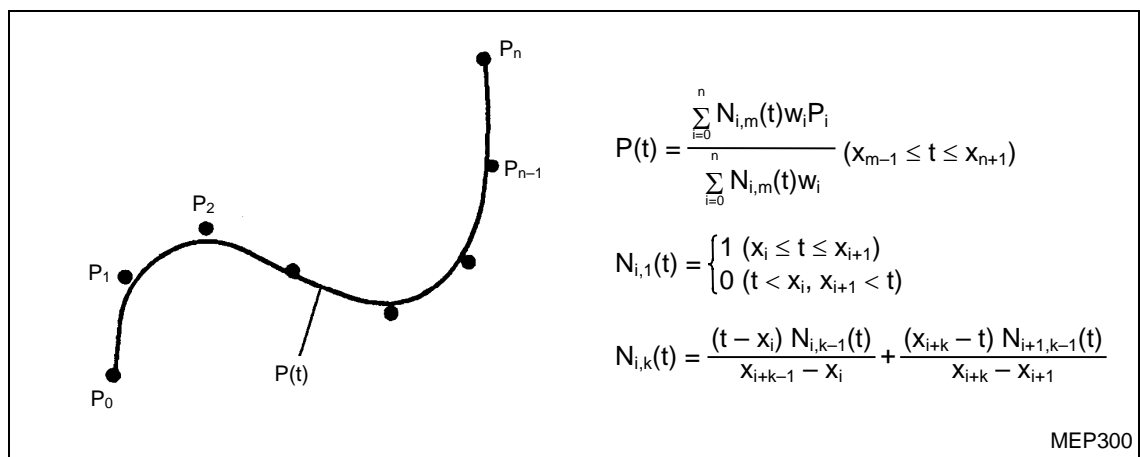


Fig. 6-19 NURBS curve

- “ P_i ” and “ w_i ” denote respectively a control point and the weight on the control point.
- “ m ” denotes the rank, and the NURBS curve of rank “ m ” is a curve of the $(m-1)$ -th order.
- “ x_i ” denotes a knot ($x_i \leq x_{i+1}$), and an array of knots $[x_0 \ x_1 \ x_2 \ \dots \ x_{n+m}]$ is referred to as the knot vector.
- A variation in parameter “ t ” from x_{m-1} to x_{n+1} produces NURBS curve $P(t)$.
- $N_{i,k}(t)$ is the B-spline basis function expressed by the above recurrence equation.

Thus the NURBS curve is uniquely defined from the weighted control points and the knot vector.

3. Programming format

G6.2[P] K_X_Y_Z_[R_] [F_] ← NURBS interpolation ON

K_X_Y_Z_[R_]

K_X_Y_Z_[R_]

K_X_Y_Z_[R_]

K_X_Y_Z_[R_]

⋮

K_X_Y_Z_[R_]

K_

K_

K_

K_

← NURBS interpolation OFF

P : Rank (omissible)

X, Y, Z : Coordinates of the control point

R : Weight on the control point (omissible)

K : Knot

F : Speed of interpolation (omissible)

4. Detailed description

Set the G6.2 code to select the NURBS interpolation mode. Subsequently, designate the rank, the coordinates and weights of the control points, and the knots to determine the shape of the NURBS curve.

The modal code G6.2, which belongs to group 1 of G-codes, is of temporary validity and the modal function relieved by a G6.2 code will automatically be retrieved upon cancellation (termination) of the NURBS interpolation. The G6.2 code can only be omitted for an immediately subsequent setting of the next NURBS curve.

Address P is used to set the rank, and the NURBS curve of rank “m” is of the (m–1)-th order, that is, set as the rank

- P2 for a straight line (curve of the first order),
- P3 for a quadratic curve (of the second order) or
- P4 for a cubic curve (of the third order).

Setting another value than 2, 3 and 4 will cause an alarm, and P4 will be used in default of argument P. The rank, moreover, should be specified in the first block (containing the G6.2 code).

Designate the control points in as many sequential blocks as required by specifying their respective coordinates and weights at addresses X, Y, Z and R. Argument R denotes the weight proper to each control point (R1.0 will be used in default), and the more the weight is applied, the closer will be drawn the NURBS curve to the control point.

Address K is assigned to knots, and the NURBS curve of rank “m” for an “n” number of control points requires an (n+m) number of knots. The required array of knots, referred to as knot vector, is to be designated in sequential blocks, namely: the first knot in the same block as the first control point, the second knot in the same block as the second control point, and so forth. Following the “n” blocks entered thus, designate the remaining “m” knots in single-command blocks. The leading single-command block of argument K also notifies the NC of the completion of entering the control points, and the NURBS interpolation function itself will be terminated with the last block for the “m” knots.

5. Remarks

1. Only the fundamental axes X, Y and Z can undergo the NURBS interpolation.
2. Do not fail to explicitly designate all the required axes X, Y and/or Z in the first block (containing G6.2). Designating a new axis in the second block onward will cause an alarm.
3. Since the first control point serves as the starting point of the NURBS curve, set in the first block (with G6.2) the same coordinates as the final point of the previous block. Otherwise, an alarm will be caused.
4. The setting range for the weight (R) is from 0.0001 to 99.9999. For a setting without decimal point, the least significant digit will be treated as units digit (for example, 1 = 1.0).
5. The knot (K) must be designated for each block. Omission results in an alarm.
6. Knots, as with the weight, can be set down to four decimal digits, and the least significant digit of a setting without decimal point will be regarded as units digit.
7. Knots must be monotonic increasing. Setting a knot smaller than that of the previous block will result in an alarm.
8. The order of addresses in a block can be arbitrary.

9. The shape of the NURBS curve can theoretically be modified very flexibly by changing the rank, the positions and weights of the control points, and the knot vector (the relative intervals of knots).

In practice, however, manual editing is almost impossible, and a special CAD/CAM system should be used to edit the NURBS curve and create the program for the interpolation. Generally speaking, do not edit manually the program created by a CAD/CAM system for the NURBS interpolation.

6. Variation of curve according to knot vector

The NURBS curve, which in general passes by the control points, can be made to pass through a specific control point by setting a certain number of knots in succession with the same value. In particular, setting as many leading and trailing knots as the rank (value of P) with the respective identical values will cause the NURBS curve to start from the first control point (P_0) and to end in the last one (P_5).

The examples given below exhibit a variation of the NURBS curve according to the knot vector with the control points remaining identical.

Example 1: Rank : 4
 Number of control points : 6
 Knot vector : [0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0]

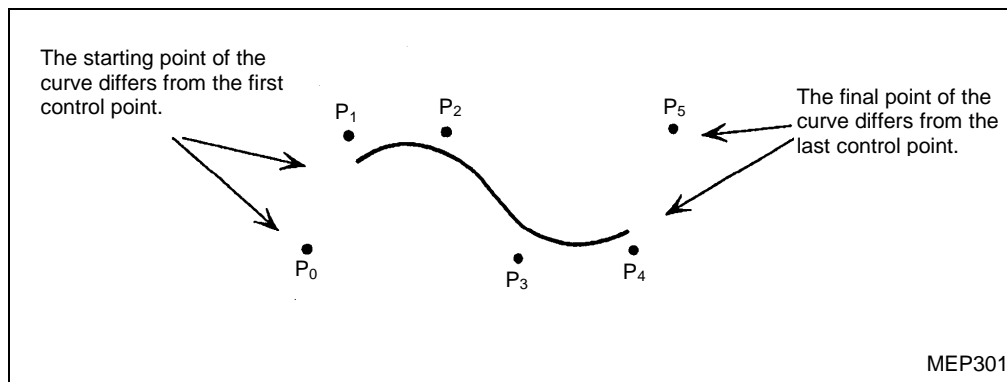


Fig. 6-20 NURBS curve for continuously increasing knots

Example 2: Rank : 4
 Number of control points : 6
 Knot vector : [0.0 0.0 0.0 0.0 1.0 2.0 3.0 3.0 3.0 3.0]

[1] [2]

Point [1]: The first four (=rank) knots have the same value assigned.

Point [2]: The last four (=rank) knots have the same value assigned.

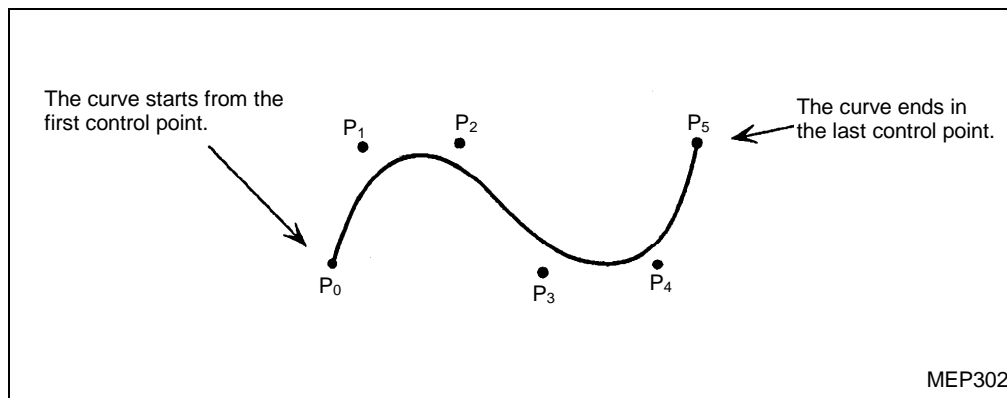


Fig. 6-21 NURBS curve for some identical knots

Note 1: The NURBS interpolation can be performed only for the NURBS curve that starts and ends from the first and in the last control point. Do not fail, therefore, to set as many leading and trailing knots as the rank with the respective identical values.

Note 2: The NURBS interpolation is executed at the designated feed rate (F-code). During the shape correction mode, however, the interpolation speed is controlled in order that the maximum available acceleration may not be exceeded in the section of a considerable curvature.

7. Compatibility with the other functions

The tables in this section specify the compatibility of the NURBS interpolation with the other functions. Pay attention to the incompatible functions, especially G-codes.

A. Preparatory, feed and auxiliary functions

The table below enumerates the G-codes, F-, M-, S-, T- and B-codes with regard to their admissibility before, with and after G6.2.

O: admissible x: not admissible

Function	Code	before G6.2	with G6.2	after G6.2
G-codes of group 00	all	O	x	x
G-codes of group 01	all	O	O (Note)	x
G-codes of group 02	G17	O	O	x
	G18			
	G19			
G-codes of group 03	G90	O	O	x
	G91			
G-codes of group 04	G22	x	x	x
	G23	O	x	x
G-codes of group 05	G93	O	O	x
	G94			
	G95			
G-codes of group 06	G20	O	O	x
	G21			
G-codes of group 07	G40	O	x	x
	G41	x	x	x
	G42	x	x	x
G-codes of group 08	G43	O	O	x
	G44			
	G49			
G-codes of group 09	G80	O	x	x
	the others	x	x	x
G-codes of group 10	G98	O	O	x
	G99			
G-codes of group 11	G50	O	x	x
	G51	x	x	x
G-codes of group 12	G54 - G59	O	O	x
G-codes of group 13	G61.1	O	x	x
	G61.2	O	x	x
	G61	x	x	x
	G62	x	x	x
	G63	x	x	x
	G64	O	x	x
G-codes of group 14	G66	x	x	x
	G66.1	x	x	x
	G66.2	x	x	x
	G67	O	x	x

Function	Code	before G6.2	with G6.2	after G6.2
G-codes of group 15	G40.1	○	×	×
	G41.1	×	×	×
	G42.1	×	×	×
G-codes of group 16	G68	×	×	×
	G69	○	×	×
G-codes of group 19	G50.1	○	×	×
	G51.1	×	×	×
High-speed machining mode	G5P0	○	×	×
	G5P2	×	×	×
Feed function	F	○	○	×
Auxiliary function	MSTB	○	×	×

Note: The G-code given last in the block takes priority in group 01.

B. Skip instructions

The table below enumerates the skip instructions with regard to their admissibility before, with and after G6.2.

○: admissible ×: not admissible

Instruction	before G6.2	with G6.2	after G6.2
Optional block skip	○	○	×
Control Out/In	○	○	×

Note: Designating another address than X, Y, Z, R and K in the mode of (i. e. after) G6.2 will cause an alarm.

C. Interruption and restart

The table below enumerates the functions for interrupting and restarting the program flow with regard to their admissibility before, with and after G6.2.

○: admissible ×: not admissible

Function	before G6.2	with G6.2	after G6.2
Single-block operation	○	×	○ (Note)
Feed hold	○	×	○
Reset	○	○	○
Program stop	○	×	×
Optional stop	○	×	×
Manual interruption (Pulse feed and MDI)	○	×	×
Restart	○	×	×
Comparison stop	○	×	×

Note: The single-block stop only occurs between blocks with different knots.

D. Tool path check

The tool path in a section of the NURBS interpolation can only be displayed as if the control points were linearly interpolated (in the mode of G01).

8. Sample program

The program section below refers to a NURBS interpolation of rank 4 (cubic curve) for seven control points.

```
Control points: P0 P1 P2 P3 P4 P5 P6
Knot vector:   [ 0.0  0.0  0.0  0.0  1.0  2.0  3.0  4.0  4.0  4.0  4.0 ]
               ⋮
               ⋮
G90 G01 X0 Y120.F3000
Y100. .... P0
G6.2 P4 X0 Y100.R1.K0... P0
X10.Y100.R1.K0..... P1
X10.Y60.R1.K0..... P2
X60.Y50.R1.K0..... P3
X80.Y60.R1.K1..... P4
X100.Y40.R1.K2..... P5
X100.Y0 R1.K3..... P6
K4.
K4.
K4.
K4.
G01 X120..... P7
               ⋮
               ⋮
```

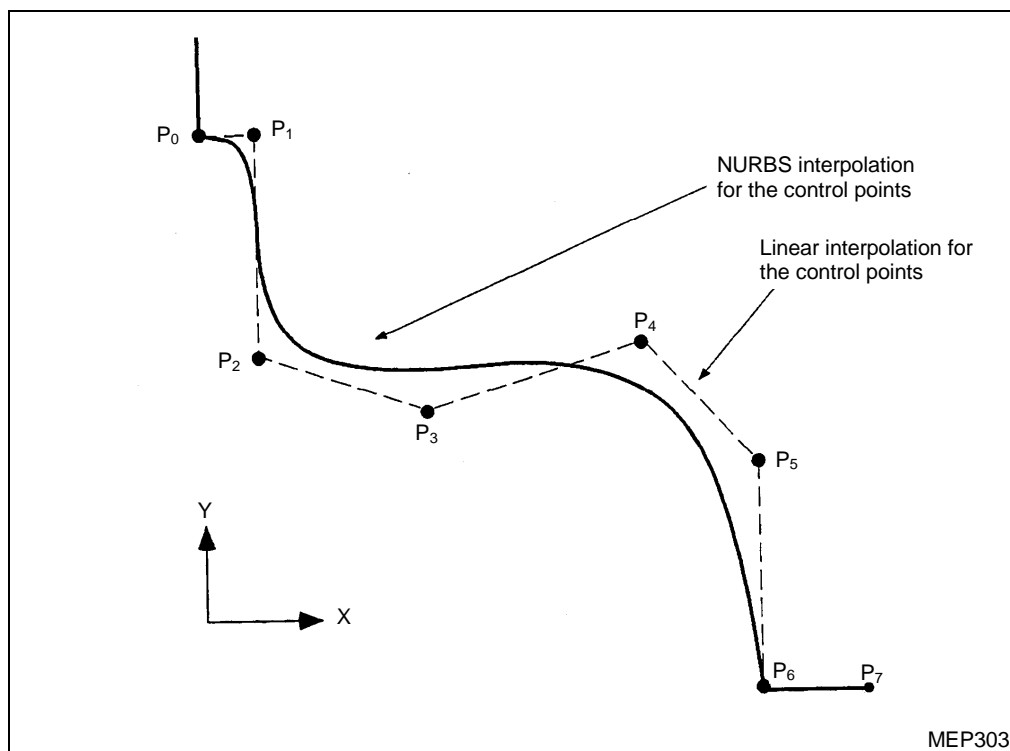


Fig. 6-22 NURBS interpolation and linear interpolation

9. Related alarms

The table below enumerates the alarms related to the NURBS interpolation.

Alarm list

Alarm No.	Alarm message	Cause	Remedy
806	ILLEGAL ADDRESS	Another address than those for the nominated axes (X, Y and/or Z), the weight (R) and the knot (K) is set in the G6.2 mode.	Clear the inadequate address.
807	ILLEGAL FORMAT	1. The modal condition is not appropriate to set G6.2.	1. Satisfy the modal condition with reference to item 7-A.
		2. A block in the G6.2 mode is set without knot (K).	2. Do not fail to set a knot in each block in the G6.2 mode.
		3. The number of blocks with the same knot in succession does not reach the rank.	3. Set an appropriate knot vector with reference to example 2 given in item 6.
809	ILLEGAL NUMBER INPUT	1. The number of digits exceeds the specification of axis commands (X, Y or Z).	1. Specify the axis command within eight digits.
		2. The rank (P) is not admissible.	2. Set 2, 3 or 4 at address P.
		3. The value of a knot is not admissible.	3. Set a value in a range of 0.0001 to 99.9999.
		4. The knot vector is not monotonic increasing.	4. Check the blocks for a decreasing knot.
816	FEEDRATE ZERO	The feed rate (F-code) has not yet been designated.	Set an F-code before or in the same block as the G6.2 code.
936	OPTION NOT FOUND	The system is not equipped with the optional function of the NURBS interpolation.	Purchase and install the optional function.
955	START AND END POINT NOT AGREE	The axis coordinates designated in the block of G6.2 do not correspond to the final point of the previous block.	Designate in the first block of the NURBS interpolation the same position as the final point of the previous block.
956	RESTART OPERATION NOT ALLOWED	The designated restart block falls within the mode of G6.2.	Restart operation is not allowed from the midst of the NURBS interpolation.
957	MANUAL INTERRUPT NOT ALLOWED	An interruption by pulse handle or MDI operation is commanded in the midst of the G6.2 mode.	Manual interruption is not allowed in the midst of the NURBS interpolation.

6-12 Cylindrical Interpolation (Option)

1. Function

Cylindrical interpolation refers to a function by which the cylindrical surface of a workpiece can be machined according to a program prepared on its development plane. This function is among others very efficient in the creation of a cam grooving program.

2. Programming format

A. Selection and cancellation of the cylindrical interpolation mode

- When the A-axis functions as the rotational axis:

G07.1 Ar; Cylindrical interpolation mode ON (r = radius of cam groove bottom)

G07.1 A0; Cylindrical interpolation mode OFF

- When the B-axis functions as the rotational axis:

G07.1 Br; Cylindrical interpolation mode ON (r = radius of cam groove bottom)

G07.1 B0; Cylindrical interpolation mode OFF

Note 1: The above preparatory function (G-code) must be given in a single-command block.

Note 2: Enter a precise value for the radius of the cam groove bottom (r), which is used for the internal calculation of the dimensions and the rate of feed in the developed plane.

Note 3: Enter a positive value for the radius of the cam groove bottom (r).

Note 4: In the mode of cylindrical interpolation the radius of the cam groove bottom (r) cannot be otherwise modified than to zero. That is, the modification must be done after canceling the mode temporarily.

Note 5: Cylindrical interpolation is not available for machines with a linear type rotational axis (F85 bit 2 = 1; HV-machines or machines with a tilting table).

Note 6: The figure below refers to a vertical machining center. To perform machining on the surface of the cylinder, position the tool on the Y-axis to the axis of the cylinder first, perform an infeed on the Z-axis even to the groove bottom and then select the mode of a cylindrical interpolation with simultaneous control of the X- and A-axis.

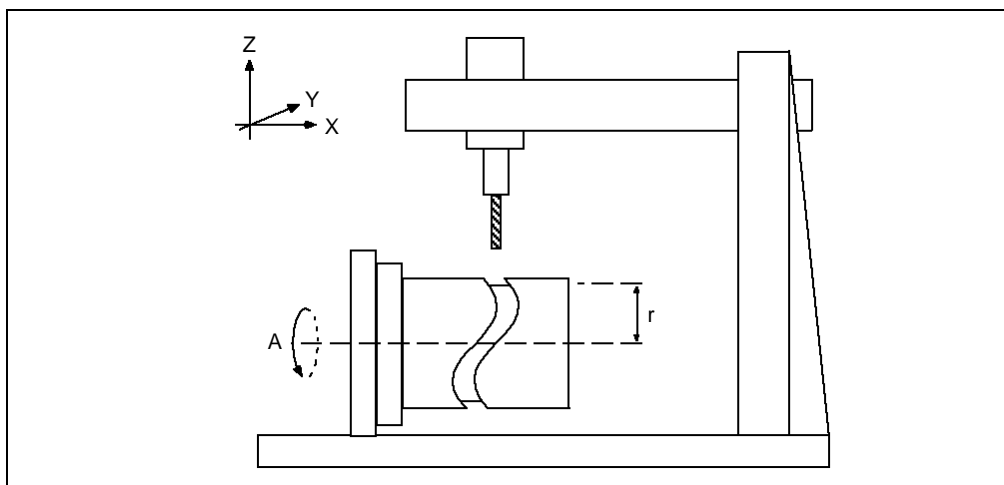


Fig. 6-23 Diagrammatic view of cylindrical interpolation (1/2)

Note 7: The next figure shows a machining center with the fixture mounted perpendicular to the Y-axis. To use the cylindrical interpolation on such a machine, position the tool on the X-axis to the axis of the cylinder first, perform an infeed on the Z-axis even to the groove bottom and then select the mode of a cylindrical interpolation with simultaneous control of the Y- and B-axis.

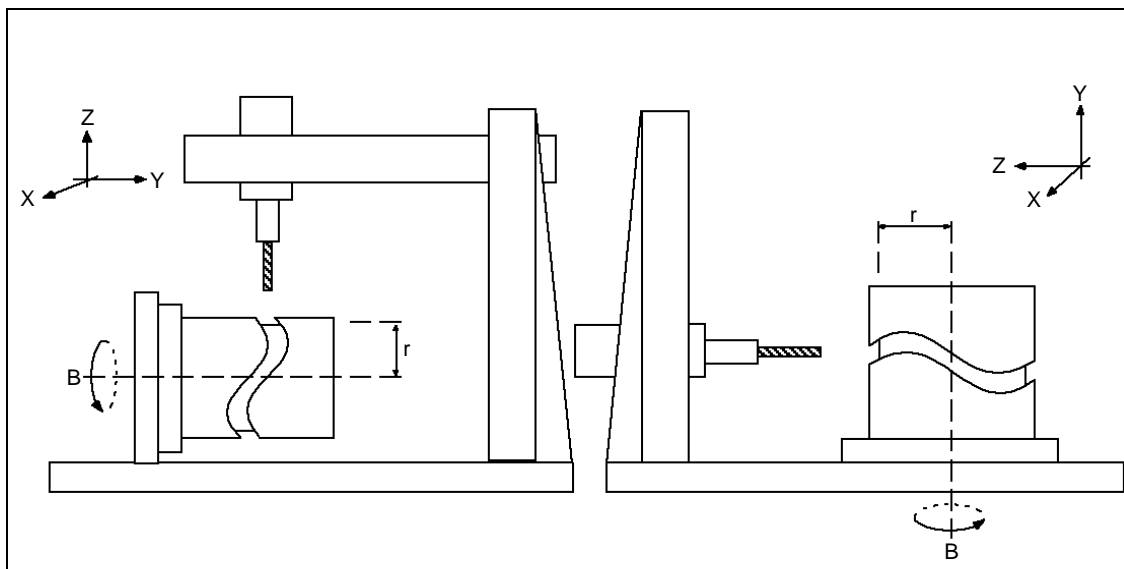


Fig. 6-24 Diagrammatic view of cylindrical interpolation (2/2)

B. Commands in the cylindrical interpolation mode

GgXxAaRrIiJjDdPpFf ;

GgYyBbRrIiJjDdPpFf ;

Gg: Refer to Table 6-3 (in paragraph 1 under 3-B) for the available G-codes.

Xx/Aa: } Cylindrical interpolation occurs in the X-A plane or Y-B plane.

Yy/Bb: } The values of X (or Y) and A (or B) are of linear and angular dimensions, respectively.

Rr: Radius for a circular interpolation

Ii: Abscissa (incremental) of the center for a circular interpolation

Jj: Ordinate (incremental) of the center for a circular interpolation

Dd: Offset number for tool diameter offset

Pp/Xx: Dwell time

Ff: Rate of feed (Refer to the detailed description in paragraph 2 under 3-B.)

3. Detailed description

A. Conditions necessary for the selection of the cylindrical interpolation mode

The G-code modal states required for the cylindrical interpolation mode selection are as follows:

Table 6-2 Conditions necessary for the mode selection

G-code group	Condition
G-code group 1	G0/G1 (Positioning or Linear interpolation only)
G-code group 2	G17 (X-Y plane only)
G-code group 3	G90/G91 (Absolute/Incremental programming) [unconditional]
G-code group 4	G22/G23 (Stroke check ON/OFF) [unconditional]
G-code group 5	G94/G95 (Asynchronous/Synchronous feed) [unconditional]
G-code group 6	G20/G21 (Inch/Metric data input) [unconditional]
G-code group 7	G40 (Tool diameter offset OFF) (Note 1)
G-code group 8	G43/G49 (Tool length offset ON/OFF) [unconditional]
G-code group 9	G80 (Fixed cycle OFF)
G-code group 10	Invalid (Return level selection [between initial and R-point] is only valid for fixed cycles.)
G-code group 11	G50 (Scaling OFF)
G-code group 12	G54 to G59, G54.1 (Standard/Additional workpiece coordinate system) [unconditional]
G-code group 13	G64 (Cutting mode)
G-code group 14	G67 (User macro modal call OFF)
G-code group 15	G40.1 (Shaping OFF)
G-code group 16	G69 (Programmed coordinates rotation OFF) (Note 2)
G-code group 19	G50.1/G51.1 (Mirror image ON/OFF) [unconditional] (Note 3)
G-code group 19	No selection of a plane for five-surface machining
Others	G5P0 (High-speed machining OFF)
Others	G7.1B0 (Cylindrical interpolation OFF)

Otherwise the selection will only lead to an alarm.

Note 1: Select and cancel the tool diameter offset as required in the mode of cylindrical interpolation. An alarm will be caused if the cylindrical interpolation is selected in the mode of tool diameter offset.

Note 2: The cylindrical interpolation cannot be selected in the mode of G68 (3-dimensional coordinate conversion). On an HV machining center, therefore, the cylindrical interpolation function is not available to an inclined or top surface.

Note 3: To use the cylindrical interpolation with the mirror image function being selected, take the following precautions in order to prevent errors from occurring in the development of the cylindrical surface:

- [1] Set the mirroring center to 0° for the rotational axis of the cylindrical interpolation.
- [2] Select the cylindrical interpolation with the rotational axis being positioned at its workpiece origin (0°).
- [3] Also cancel the cylindrical interpolation with the rotational axis being positioned at its workpiece origin (0°).

An example of programming is given later in the sample program under 5-B.

B. Commands in the cylindrical interpolation mode

- The table below enumerates the G-codes available in the cylindrical interpolation mode. Any other G-code will cause an alarm.

Table 6-3 Available G-codes

G-code	Function
G0	Rapid positioning
G1, G2, G3	Linear and circular interpolation
G4	Dwell
G9	Exact-stop check
G17	Plane selection (Note 1)
G40, G41, G42	Tool diameter offset (Note 2)

Note 1: After the mode selection by a block of G7.1, do not fail to give a plane selection command of the following format in order to specify the cylindrical interpolation plane determined by the corresponding linear and rotational axes:

G17X__A__ when the A-axis functions as the rotational axis.

G17Y__B__ when the B-axis functions as the rotational axis.

Note 2: Select and cancel the tool diameter offset as required in the mode of cylindrical interpolation. An alarm will be caused if the cylindrical interpolation is selected in the mode of tool diameter offset.

- In the mode of the cylindrical interpolation the rate of feed refers to a resultant speed (of Fx and Fa, or Fy and Fb) in the plane onto which the surface of the cylinder is developed.

The speed on each component axis is calculated for a block of G1XxAaFf; as follows:

$$F_x = \frac{x}{\sqrt{x^2 + \left(\frac{a}{360} 2\pi r\right)^2}} \cdot f$$

$$F_a = \frac{\frac{a}{360} 2\pi r}{\sqrt{x^2 + \left(\frac{a}{360} 2\pi r\right)^2}} \cdot f$$

x: metric (0.001 mm)
a: degree (0.001°)
r: radius of cam groove bottom
f: command value of speed

The speed on each component axis is calculated for a block of G1YyBbFf; as follows:

$$F_y = \frac{y}{\sqrt{y^2 + \left(\frac{b}{360} 2\pi r\right)^2}} \cdot f$$

$$F_b = \frac{\frac{b}{360} 2\pi r}{\sqrt{y^2 + \left(\frac{b}{360} 2\pi r\right)^2}} \cdot f$$

y: metric (0.001 mm)
b: degree (0.001°)
r: radius of cam groove bottom
f: command value of speed

The speed of rapid traverse and the upper limit of cutting feed, both specified in a parameter, are expressed in an angular velocity (°/min) for a rotational axis. The actual linear speed on the rotational axis of the cylindrical interpolation is therefore allowed to increase in the developed plane just in proportion to the radius of the groove bottom.

4. Remarks

A. Positioning accuracy (on the rotational axis)

Each angular dimension entered is internally converted into linear one on the circumference, which is to be used in the calculation of the interpolation with the other linear axis. The actual angular motion is then determined by the results of that calculation. As a result, depending on the cylinder radius, positioning errors on the rotational axis may occur in the level of the least significant digit, but they are not cumulative.

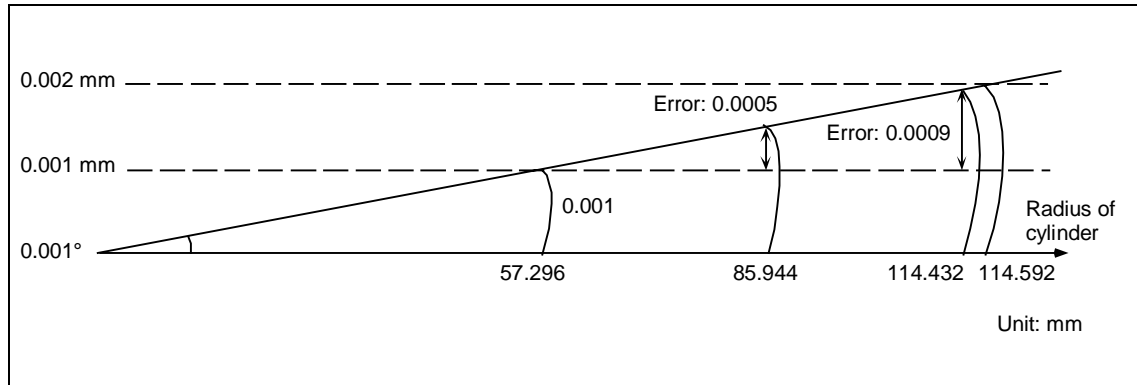


Fig. 6-25 Positioning error according to the angle and radius

B. Changing the radius of the groove bottom

In the mode of cylindrical interpolation, as mentioned before, the radius of the cam groove bottom cannot be otherwise modified than to zero. That is, the modification must be done after canceling the mode temporarily.

C. Rotational axis of the cylindrical interpolation

Only one rotational axis can be used for the cylindrical interpolation. It is not possible to designate multiple rotational axes in a command of G7.1.

D. Manual interruption

1. With "manual absolute" ON

The "manual absolute" function is suspended during cylindrical interpolation, and the first motion block after the cancellation of cylindrical interpolation is executed for the very target position as programmed by canceling the amount of manual interruption.

Refer to the example given later under 5-D.

2. With "manual absolute" OFF

The amount of manual interruption remains intact, irrespective of the selection and cancellation of cylindrical interpolation. See the example given under 5-D.

E. Restart operation

For restarting in the middle of the cylindrical interpolation, follow the normal restart procedure to ensure normal operation by retrieving the necessary modal information (on the radius of groove bottom, etc.). Never use the **[RESTART 2]** menu function that skips the preceding blocks.

F. Resetting

Resetting (by the RESET key on the operating panel) cancels the cylindrical interpolation mode.

5. Sample programs

Note: The examples in this section are all given for a cylindrical interpolation in the Y-B plane (with respect to a machining center as shown in Fig. 6-24).

Replace the axis names X, Y and B with Y, X and A, respectively, for a cylindrical interpolation in the X-A plane (on a vertical machining center as shown in Fig. 6-23).

A. Cam grooving program

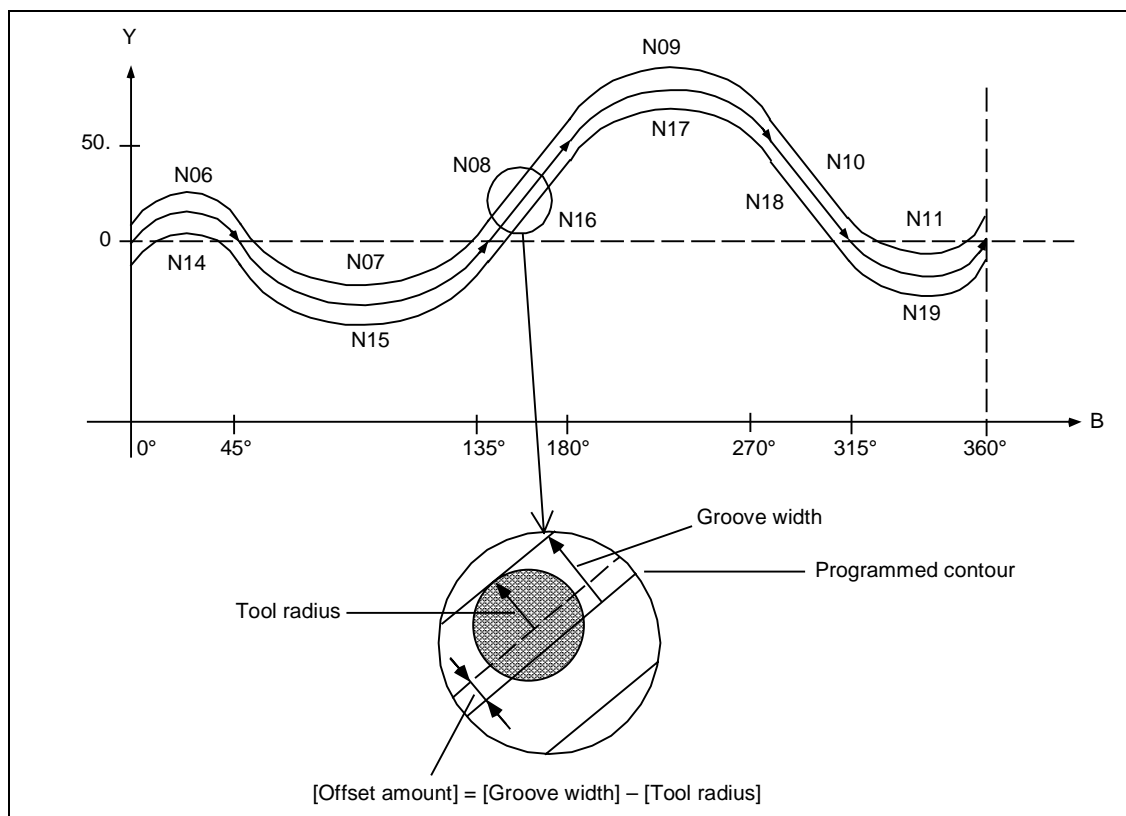


Fig. 6-26 Cam grooving program

```

N01 G54G0G90X0Y0B0; .. Positioning to the cylinder axis
N02 Z0S800M3;
N03 G1Z-5.F2000; ..... Infeed to the groove bottom
N04 G7.1B63.662; ..... Cylindrical interpolation ON
N05 G17G1G41Y0B0D1;
N06 G2B45.R30.;
N07 G3B135.R60.;
N08 G1Y50.B180.;
N09 G2B270.R60.;
N10 G1Y0B315.;
N11 G3B360.R30.;
N12 G1G40;
N13 G1G42Y0B360.;
N14 G2B45.R30.;
N15 G3B135.R60.;
N16 G1Y50.B180.;
N17 G2B270.R60.;
N18 G1Y0B315.;
N19 G3B360.R30.;
N20 G1G40;
N21 G7.1B0; ..... Cylindrical interpolation OFF

```

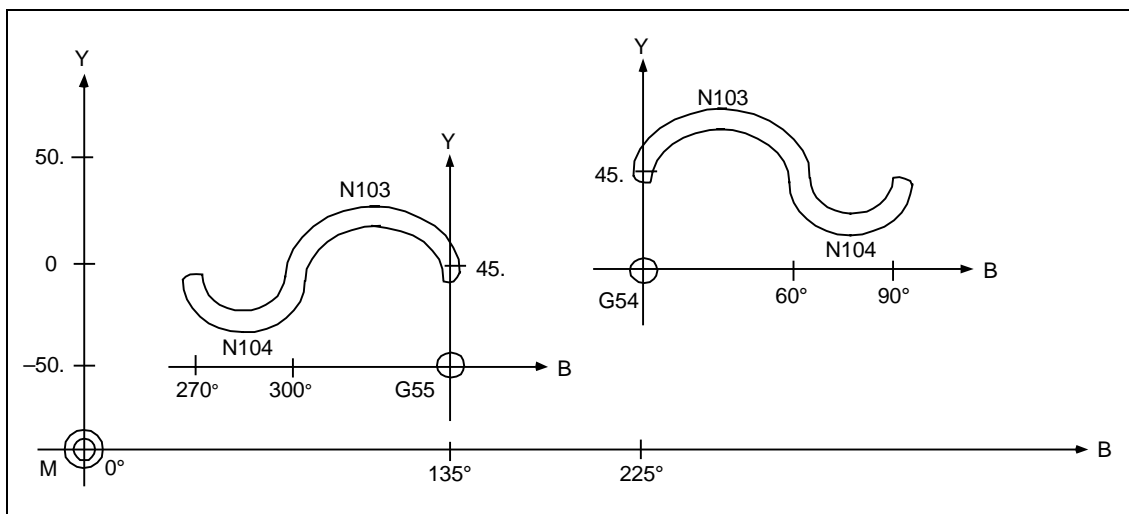

B. Use of mirror image function

Fig. 6-27 Use of mirror image function

Main program

```

N01 G54G0G17G90X0Y0B0;
N02 Z0S800M3;
N03 M98P2000;
N04 G55G0G17G90X0Y0B0;
N05 Z0S800M3;
N06 G51.1B0; .....Mirror image ON
N07 M98P2000;
N08 G50.1B0; .....Mirror image OFF

```

Subprogram

```

N100 G7.1 B47.746 ;
N101 G17 G0 Y45.B0 ;
N102 G1 Z-5.F1000 ;
N103 G2 B60.R25. ;
N104 G3 B90.R12.5 ;
N105 G0 Z0 ;
N106 G0 B0 ;
N107 G7.1 B0 ;
N108 M99 ;

```

Table 6-4 Workpiece origin data

	G54	G55
X	0.	0.
Y	0.	-50.
Z	0.	0.
B	225.	135.

C. Use of circular interpolation commands

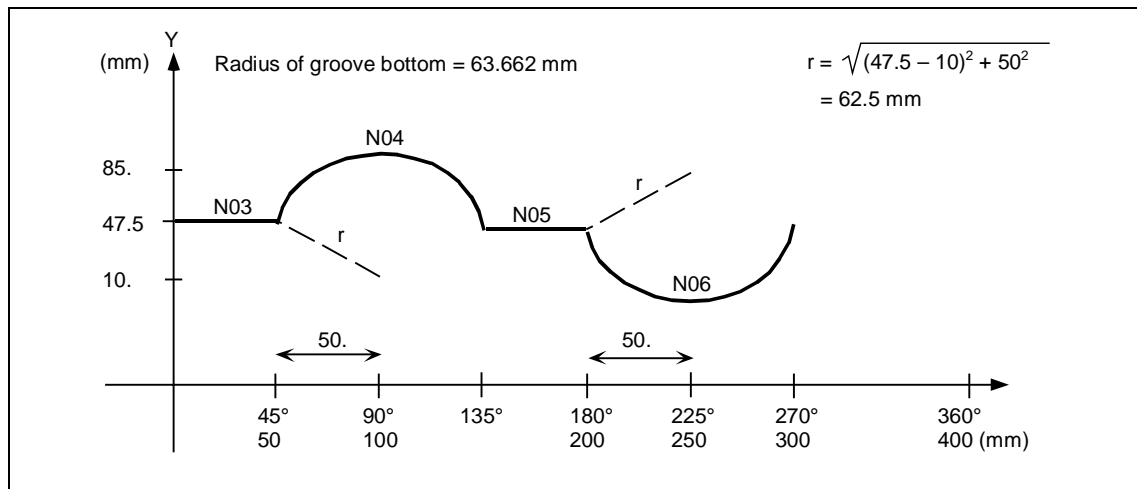


Fig. 6-28 Use of circular interpolation commands

```

N01 G90 G1 Y47.5 F1000 ;
N02 G7.1 B63.662 ;
N03 G17 G1 Y47.5 B45. ;
N04 G2 B90.R62.5 ;
N05 G1 B180. ;
N06 G3 B270.I50.J37.5 ;

```

D. Operation examples with manual interruption

1. With "manual absolute" ON

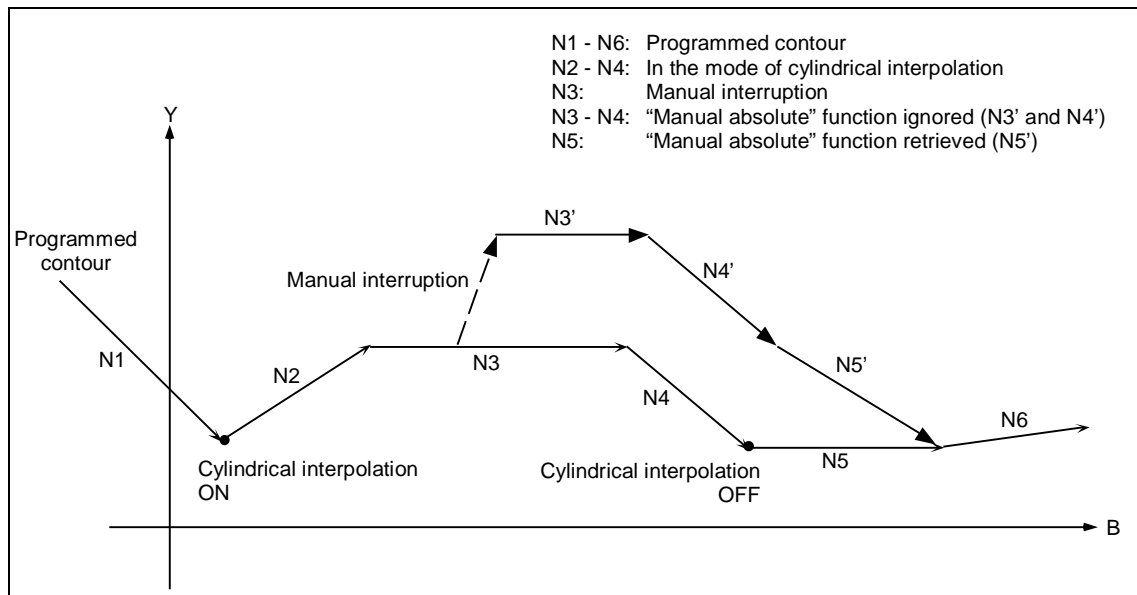


Fig. 6-29 With "manual absolute" ON

2. With “manual absolute” OFF

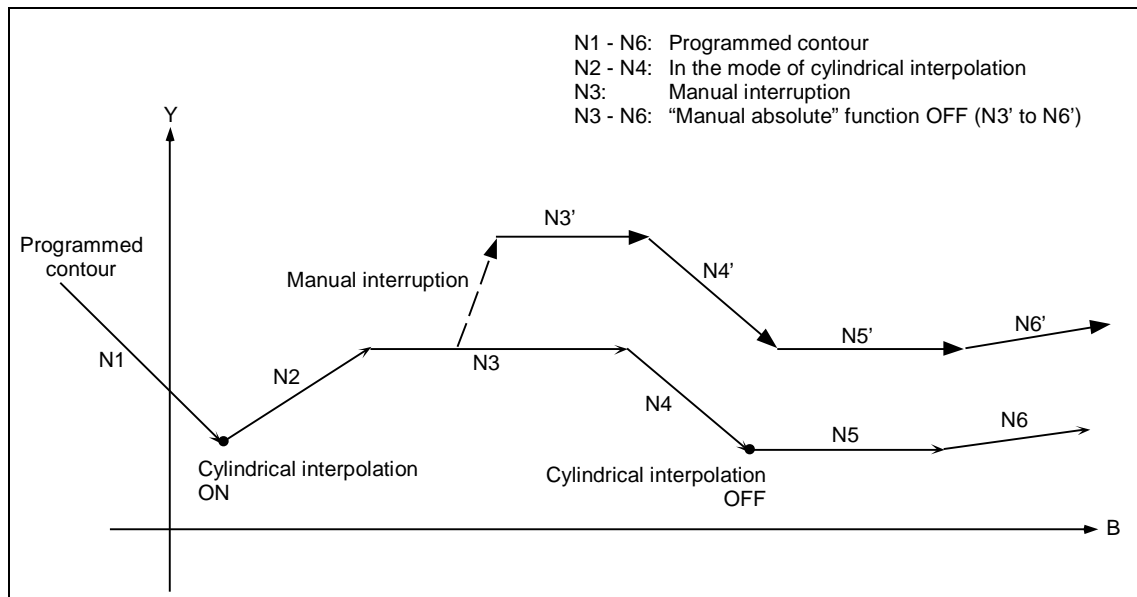


Fig. 6-30 With “manual absolute” OFF

6. Related parameters

- Set the type of the rotational axis (A or B) as “rotational” (with short-cut approach) in order that the initial angular movement should not occur on the roundabout route for a command of tool diameter offset given in the vicinity of the 0° position.

F85 bit 2 = 0: Setting the type of the rotational axis as rotational

bit 3 = 1: Short-cut approach valid

- Set the B- and A-axis as the primary parallel ones to the axis of abscissa and that of the ordinate, respectively.

F31 = 66: Setting the B-axis as parallel axis 1 for the axis of abscissa

F34 = 65: Setting the A-axis as parallel axis 1 for the axis of the ordinate

- The following parameter setting is required for a cylindrical interpolation in the X-A plane on vertical machining centers:

F85 bit 7 = 1

- The following parameter setting is required for inch data input in order that the speed of the angular movement may be calculated appropriately:

F85 bit 4 = 1

7. Related alarms

Table 6-5 Related alarms

Alarm No.	Alarm message	Description	Remedy
806	ILLEGAL ADDRESS	The address of argument in the block of selecting the cylindrical interpolation does not refer to any rotational axis.	Use the correct address.
807	ILLEGAL FORMAT	The necessary conditions for the selection of cylindrical interpolation are not yet all satisfied.	Refer to Table 6-2.
808	MIS-SET G CODE	An unavailable G-code is given in the mode of cylindrical interpolation.	Refer to Table 6-3.
936	OPTION NOT FOUND (7, 0, 0)	The optional function for cylindrical interpolation is not provided.	Equip the machine with the option as required.

- NOTE -

7 FEED FUNCTIONS

7-1 Rapid Feed Rates

A separate rapid feed rate can be set for each axis. The range of rapid feed rates which can be set is from 1 mm/min to 240000 mm/min. The maximum feed rate, however, is limited according to the particular machine specifications.

Rapid feed rates apply for commands G00, G27, G28, G29, G30 and G60.

7-2 Cutting Feed Rates

A cutting feed rate must be designated using address F and an eight-digit number (F8-digit direct designation).

The F8 digits can consist of five integral digits and three decimal digits, with the decimal point. Cutting feed rates apply for commands G01, G02, G03, G2.1 and G3.1.

Example:

	Feed rate
G01 X100. Y100. F200*	200.0 mm/min
G01 X100. Y100. F123.4	123.4 mm/min
G01 X100. Y100. F56.789	56.789 mm/min

* It means the same if F200. or F200.000 is set in stead of F200.

Setting range for cutting feed rate

Command mode	F command range	Feed rate range	Remarks
mm/min	0.0001 to 200000.0000	0.0001 to 200000.0000 mm/min	
inches/min	0.00001 to 20000.00000	0.00001 to 20000.00000 inches/min	
deg/min	0.0001 to 200000.0000	0.0001 to 200000.0000 deg/min	

Note 1: A program error will result if no F-code command has been set for the first cutting command (G01, G02, G03, G2.1 or G3.1) after switching-on.

Note 2: If you set data in inches for machines of the metric system, the maximum available feed rate is 9999 inches/min for both rapid feed and cutting feed.

7-3 Synchronous/Asynchronous Feed: G95/G94

1. Function and purpose

Command G95 allows a feed rate per spindle revolution to be set using an F code.

2. Programming format

G94: Feed per minute (mm/min) [Asynchronous feed]

G95: Feed per revolution (mm/rev) [Synchronous feed]

Since command G95 is modal, it will remain valid until command G94 is issued.

3. Detailed description

- Feed rates that can be set using F codes are listed in the table below.

The table below also lists synchronous feed rates, which are to be set in millimeters (or inches) per spindle revolution using F codes.

Feed mode	Metric input		Inch input	
	Range of command data	Range of feed rates	Range of command data	Range of feed rates
Feed per revolution	F0.0001 to F200000.0000	0.0001 to 200000.0000 mm/rev	F0.00001 to F20000.00000	0.00001 to 20000.00000 in./rev
Feed per minute	F0.0001 to F200000.0000	0.0001 to 200000.0000 mm/min	F0.00001 to F20000.00000	0.00001 to 20000.00000 in./min

- The effective feed rate for synchronous feed, that is, the actual moving speed of the machine, can be calculated as follows:

$$FC = F \times N \times OVR \text{ (Expression 1)}$$

where FC : Effective feed rate (mm/min or inches/min)
 F : Designated feed rate (mm/rev or inches/rev)
 N : Spindle speed (rpm)
 OVR : Cutting feed override

For a simultaneous movement on multiple axes, effective feed rate FC given by expression 1 above refers to the direction of commanded movement.

4. Supplement

- On the **POSITION** display, FC denotes an effective feed rate that is expressed in a feed rate per minute (mm/min or inches/min), based on the selected feed rate, the spindle speed, and the cutting feed override.
- If the effective feed rate should become larger than the cutting feed limit speed, that limit speed will govern.
- An alarm will occur if the spindle speed is set to zero during synchronous feed.
- In the dry run mode, feed will become asynchronous and the machine will operate at an externally preset feed rate (mm/min, inches/min or deg/min).
- Fixed-cycle machining commands G84 (tapping cycle) and G74 (reverse tapping cycle) are executed according to the preselected feed mode.

7-4 Selecting a Feed Rate, and Effects on Each Control Axis

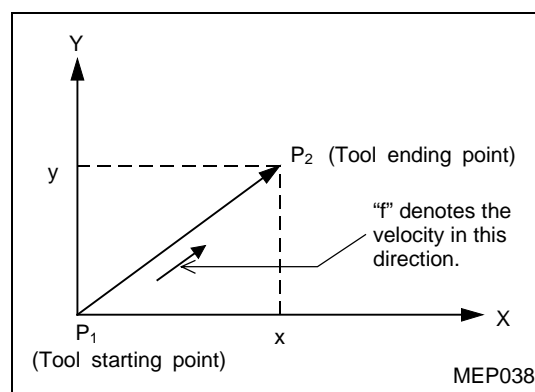
As mentioned earlier, the machine has various control axes. These control axes can be broadly divided into linear axes, which control linear motions, and rotational axes, which control rotational motions. Feed rates for control axes have different effects on the tool speed, which is of great importance for machining quality, according to the particular type of axis controlled.

The amount of displacement must be designated for each axis, whereas the feed rate is to be designated as a single value for the intended tool movement. Before letting the machine control two or more axes at the same time, therefore, you must understand how the feed rate designated will act on each axis. In terms of this, selection of a feed rate is described below.

1. Controlling linear axes

The feed rate that has been selected using an F code acts as a linear velocity in the moving direction of the tool, irrespective of whether only one axis is to be controlled or multiple axes simultaneously.

Example: If linear axes (X- and Y-axes) are to be controlled using a feed rate of f :



When only linear axes are to be controlled, setting of a cutting feed rate itself is only required. The feed rate for each axis refers to that component of the specified feed rate which corresponds with the ratio of movement stroke on the respective axis to the actual movement distance.

In the example shown above:

$$\text{X-axis feed rate} = f \times \frac{x}{\sqrt{x^2 + y^2}}$$

$$\text{Y-axis feed rate} = f \times \frac{y}{\sqrt{x^2 + y^2}}$$

2. Controlling a rotational axis

When a rotational axis is to be controlled, the selected feed rate acts as the rotating speed of the rotational axis, that is, as an angular velocity.

Thus, the cutting speed in the moving direction of the tool, that is, a linear velocity varies according to the distance from the rotational center to the tool. This distance must be considered when setting a feed rate in the program.

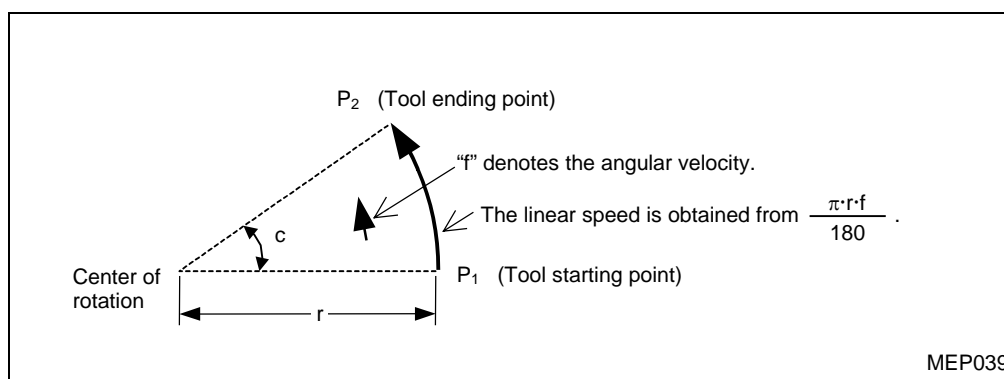
Example 1: If a rotational axis (C axis) is to be controlled using a feed rate of f (deg/min):

In this case, the cutting speed in the moving direction of the tool (linear velocity) is calculated as follows:

$$f_c = f \times \frac{\pi \cdot r}{180}$$

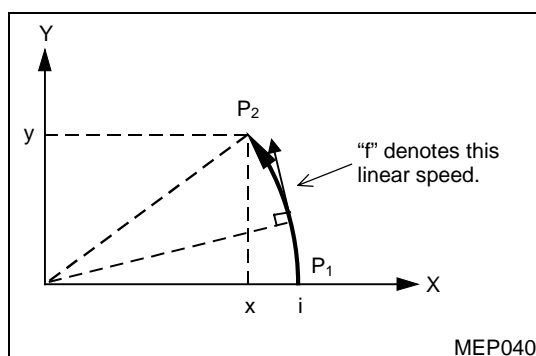
Hence, the feed rate to be programmed for the required value f_c is:

$$f = f_c \times \frac{180}{\pi \cdot r}$$



Note: If the tool is to be moved by controlling linear axes along the circumference using the circular interpolation function, the feed rate to be programmed is the velocity acting in the moving direction of the tool, that is, in the tangential direction.

Example 2: If linear axes (X- and Y-axes) are to be controlled at a feed rate of f using the circular interpolation function: X- and Y-axis feed rates will change with the movement of the tool. The resultant velocity, however, will be kept at the constant value, f .

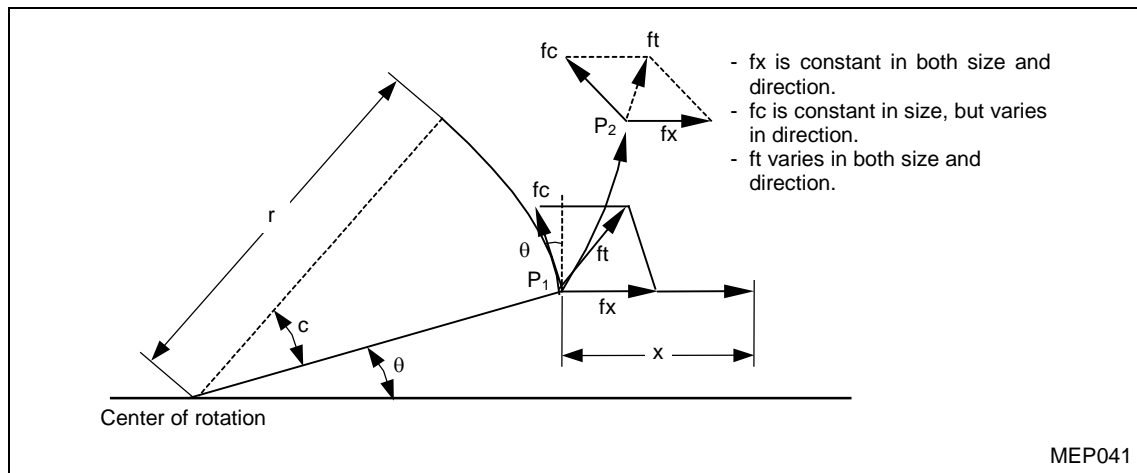


3. Controlling a linear axis and a rotational axis at the same time

The NC unit controls linear axes and rotational axes in exactly the same manner.

For control of rotational axes, data given in a coordinate word (with A, B or C) is handled as an angle, and data given as a feed rate (F) is handled as a linear velocity. In other words, an angle of one degree for a rotational axis is handled as equivalent to a moving distance of 1 mm for a linear axis. Thus, for simultaneous control of a linear axis and a rotational axis, the magnitudes of the individual axis components of the data that has been given by F are the same as those existing during linear axis control described previously in Subparagraph 1. above. In this case, however, the velocity components during linear axis control remain constant in both magnitude and direction, whereas those of rotational axis control change in direction according to the movement of the tool. Therefore, the resulting feed rate in the moving direction of the tool changes as the tool moves.

Example: If a linear axis (X-axis) and a rotational axis (C-axis) are to be controlled at the same time at a feed rate of f :



X-axis incremental command data is expressed here as x , and that of C-axis as c . The X-axis feed rate (linear velocity), f_x , and the C-axis feed rate (angular velocity), ω , can be calculated as follows:

$$f_x = f \times \frac{x}{\sqrt{x^2 + c^2}} \dots\dots\dots [1] \quad \omega = f \times \frac{c}{\sqrt{x^2 + c^2}} \dots\dots\dots [2]$$

The linear velocity f_c that relates to C-axis control is expressed as:

$$f_c = \omega \cdot \frac{\pi \cdot r}{180} \dots\dots\dots [3]$$

If the velocity in the moving direction of the tool at starting point P_1 is taken as f_t , and its X- and Y-axis components as f_{tx} and f_{ty} respectively, then one can express f_{tx} and f_{ty} as follows:

$$f_{tx} = -r \sin \left(\frac{\pi}{180} \theta \right) \times \frac{\pi}{180} \omega + f_x \dots\dots\dots [4]$$

$$f_{ty} = -r \cos \left(\frac{\pi}{180} \theta \right) \times \frac{\pi}{180} \omega \dots\dots\dots [5]$$

where r denotes the distance (in millimeters) from the rotational center to the tool, and θ denotes the angle (in degrees) of starting point P_1 to the X-axis at the rotational center.

From expressions [1] through [5] above, the resultant velocity f_t is:

$$f_t = \sqrt{f_{tx}^2 + f_{ty}^2} \\ = f \times \frac{\sqrt{x^2 - x \cdot c \cdot r \sin \left(\frac{\pi}{180} \theta \right) \frac{\pi}{90} + \left(\frac{\pi \cdot r \cdot c}{180} \right)^2}}{\sqrt{x^2 + c^2}} \dots\dots [6]$$

The feed rate f that is to be set in the program must be therefore:

$$= f_t \times \frac{\sqrt{x^2 + c^2}}{\sqrt{x^2 - x \cdot c \cdot r \sin \left(\frac{\pi}{180} \theta \right) \frac{\pi}{90} + \left(\frac{\pi \cdot r \cdot c}{180} \right)^2}} \dots\dots [7]$$

In expression [6], f_t is the velocity at starting point P_1 and thus the value of f_t changes with that of θ which changes according to the rotational angle of the C axis. To keep cutting speed f_t as constant as possible, the rotational angle of the C axis in one block must be minimized to ensure a minimum rate of change of θ .

7-5 Exact-Stop Check: G09

1. Function and purpose

Only after the in-position status has been checked following machine stop, may you want to start the next block command in order to reduce possible machine shocks due to abrupt changes in tool feed rate and to minimize any rounding of workpieces during corner cutting. An exact-stop check function is provided for these purposes.

The in-position width can be set in parameter **S14**.

2. Programming format

G09

Exact-stop check command G09 is valid only for the cutting command code (G01, G02 or G03) given in the same block.

3. Sample program

```
N001 G09 G01 X100.000 F150
```

The next block is executed after an in-position status check following machine stop.

```
N002 Y100.000
```

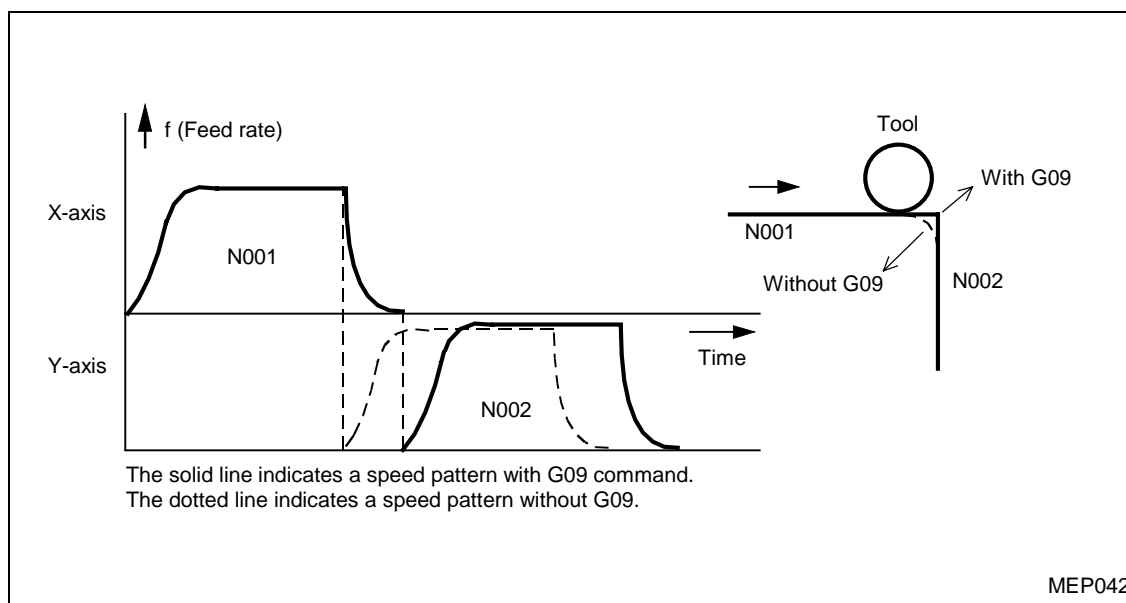


Fig. 7-1 Effect of exact-stop check

4. Detailed description

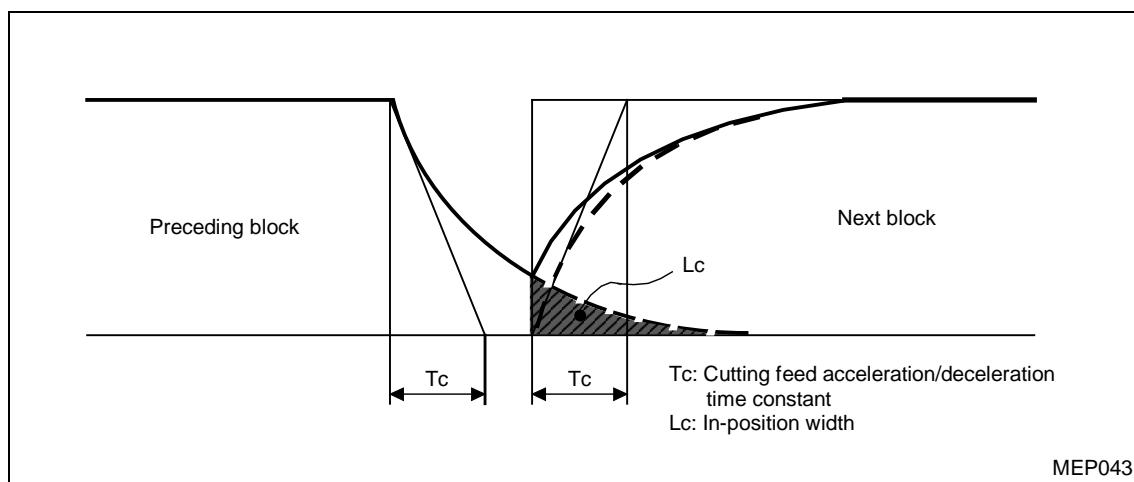


Fig. 7-2 Block-to-block connections in cutting feed deceleration check mode

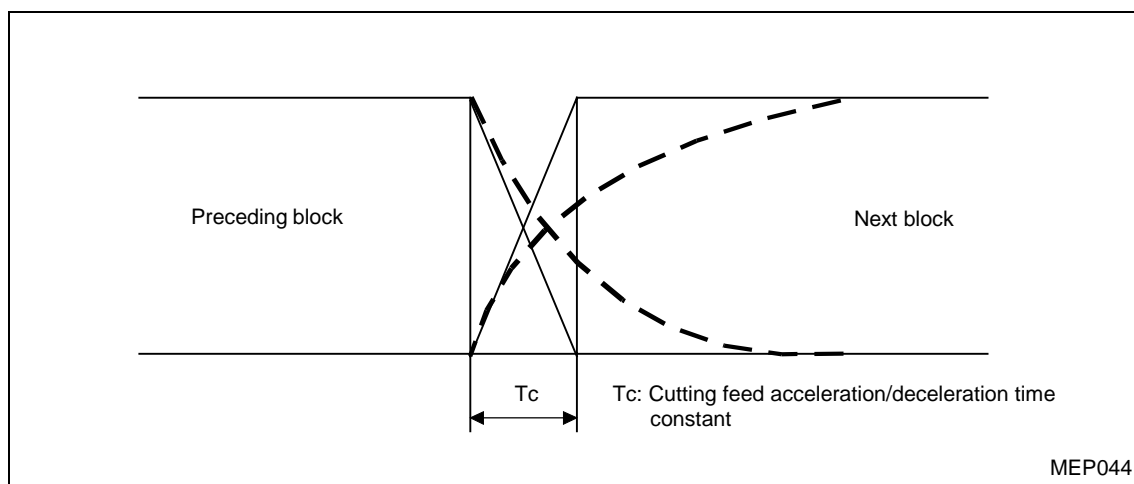
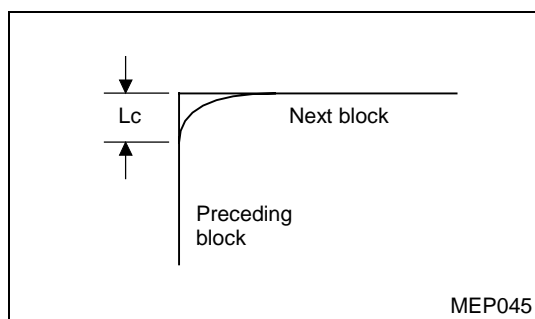


Fig. 7-3 Continuous cutting feed command

As shown in Fig. 7-2, in-position width L_c is the remaining distance of movement command of a block (equivalent to the hatched area) at the start of execution of the next block.

The in-position width can be set in parameter **S14**, with an integer in 0.001 mm or 0.0001 inches. The in-position width is effective to keep any rounding of workpieces during corner cutting under the fixed level.



Include G09 in a fixed-cycle machining subprogram if exact-stop check is to be performed for a cutting block of the fixed-cycle.

If rounding of workpieces at corners is to be completely suppressed, reduce the value of parameter **S14** or include a dwell command (G04) between cutting blocks.

7-6 Exact-Stop Check Mode: G61

1. Function and purpose

Unlike exact-stop check command G09 which performs an in-position status check on the related block only, command G61 functions as a modal command. That is, this command acts on all its succeeding cutting commands (G01, G02 and G03) so that deceleration occurs at the end of each block, followed by an in-position status check. This command is cleared by automatic corner override command G62, tapping mode command G63 or cutting mode command G64.

2. Programming format

G61

7-7 Automatic Corner Override: G62

1. Function and purpose

Command G62 automatically overrides in the tool-diameter offset mode the selected feed rate to reduce the tool load during inner-corner cutting.

Once this command has been issued, the automatic corner override function will remain valid until it is cancelled by tool-diameter offsetting cancellation command G40, exact-stop check mode command G61, tapping mode command G63 or cutting mode command G64.

2. Programming format

G62

3. Detailed description

When an inner corner is cut as shown in the figure below, the load on the tool increases because of larger amount of cutting. Using G62 in such a case allows the cutting feed rate to be automatically overridden within the preset zone, and thus the tool load to be reduced to accomplish appropriate cutting.

This function, however, is valid only for programming the as-finished shape of a workpiece.

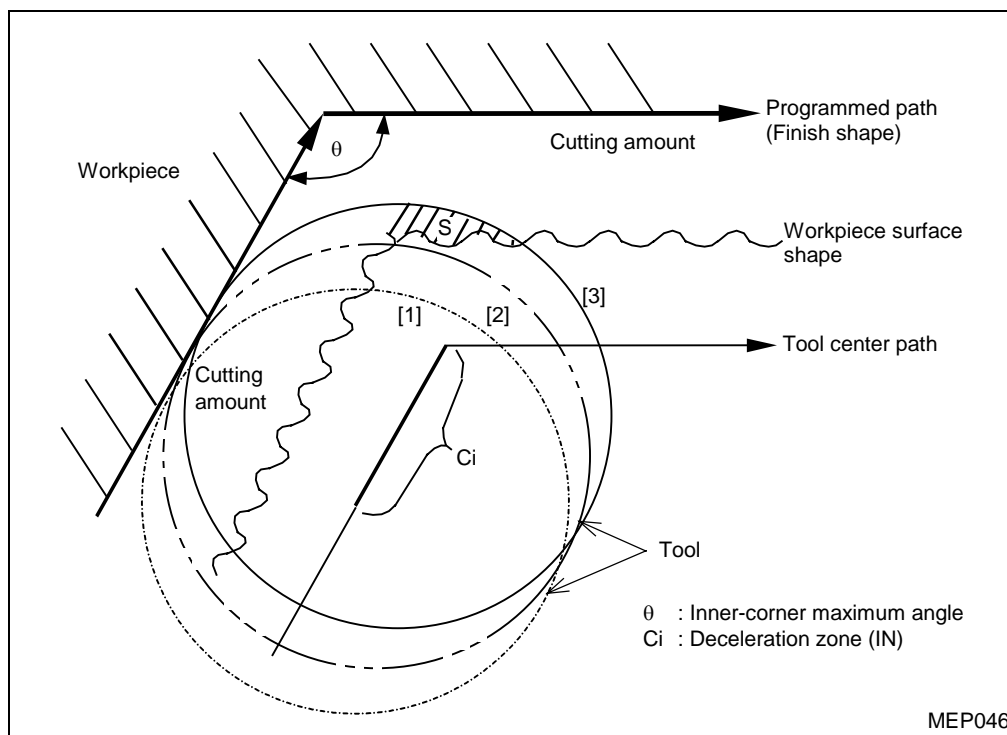


Fig. 7-4 Inner-corner cutting

<Machine operation>

- When the automatic corner override function is not used:

As shown in the figure above, as the tool is moving in order of positions [1]→[2]→[3], the load on the tool increases because the cutting amount at position [3] is larger than that of position [2] by the area of hatched section S.

- When the automatic corner override function is used:

If maximum angle θ of the inner corners is smaller than that preset in the appropriate parameter, the feed rate is automatically overridden with the preset value for movement through the deceleration zone C_i .

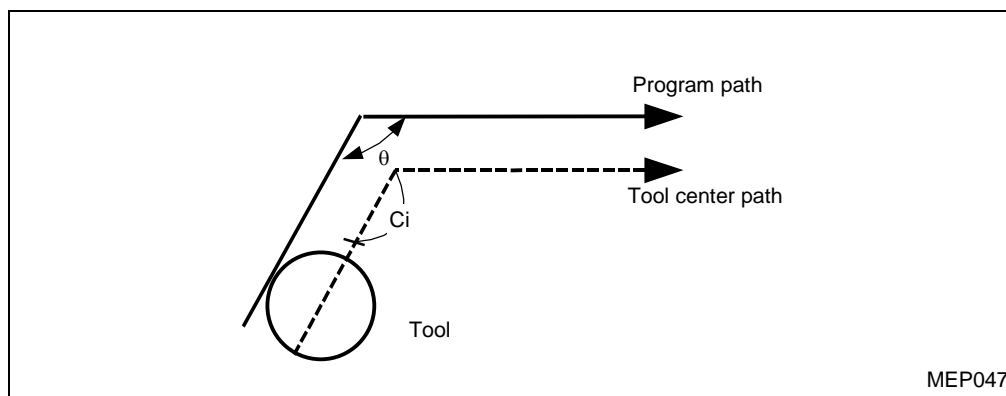
4. Setting parameters

Set the following parameters as user parameters:

- **F29:** Override 0 to 100 (%)
- **F21:** Inner-corner maximum angle θ 0 to 179 (deg)
- **F22:** Deceleration zone C_i data 0 to 99999.999 (mm) or 0 to 3937.000 (inches)

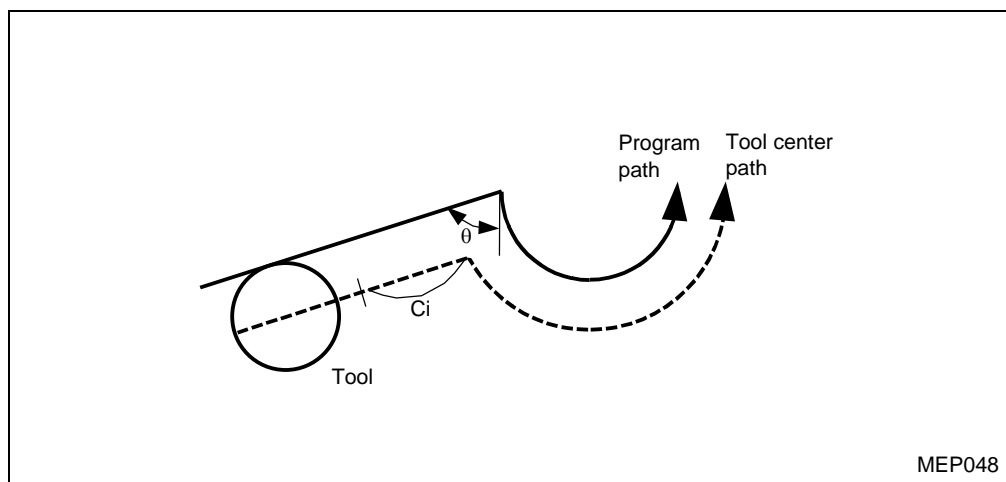
5. Examples

- Line-to-line corner



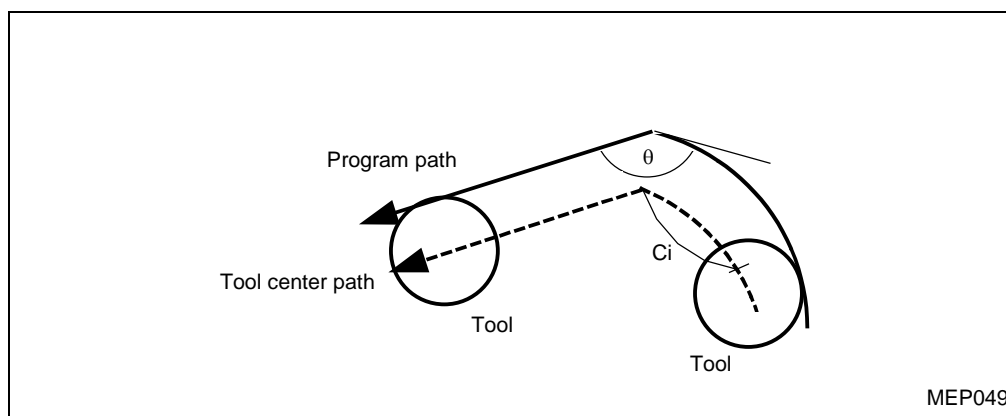
The feed rate is automatically overridden with the preset value of parameter **F29** through the deceleration zone C_i .

- Line-to-arc (outside offsetting) corner



The feed rate is automatically overridden with the preset value of parameter **F29** through the deceleration zone C_i .

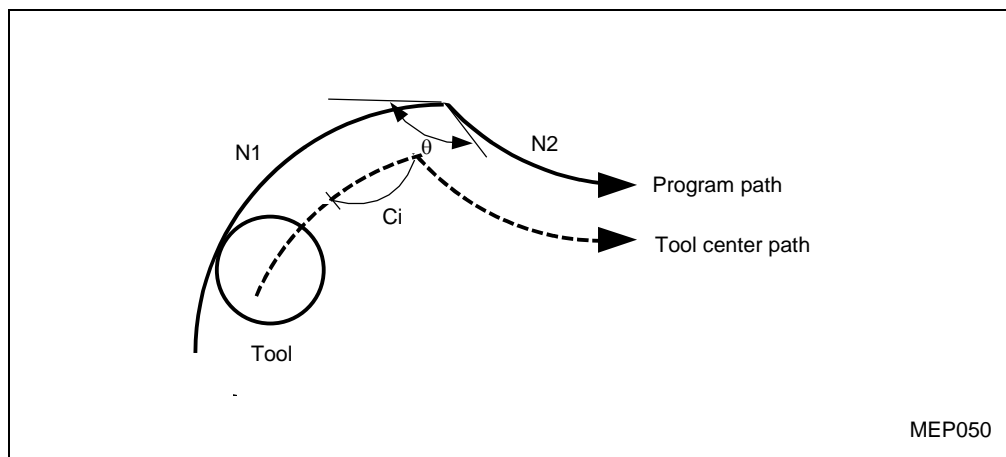
- Arc (inside offsetting)-to-line corner



The feed rate is automatically overridden with the preset value of parameter **F29** through the deceleration zone C_i .

Note: Data of deceleration zone C_i refers to the length of arc in the circular interpolation.

- Arc (inside offsetting)-to-arc (outside offsetting) corner



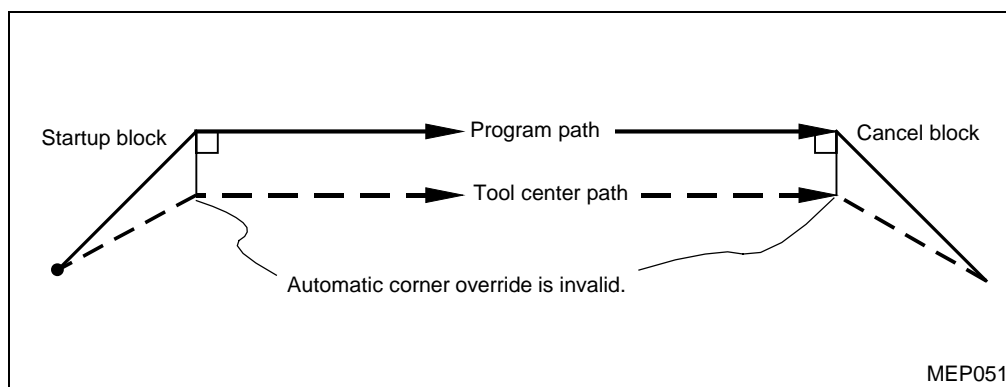
The feed rate is automatically overridden with the preset value of parameter **F29** through the deceleration zone Ci.

6. Correlations to other command functions

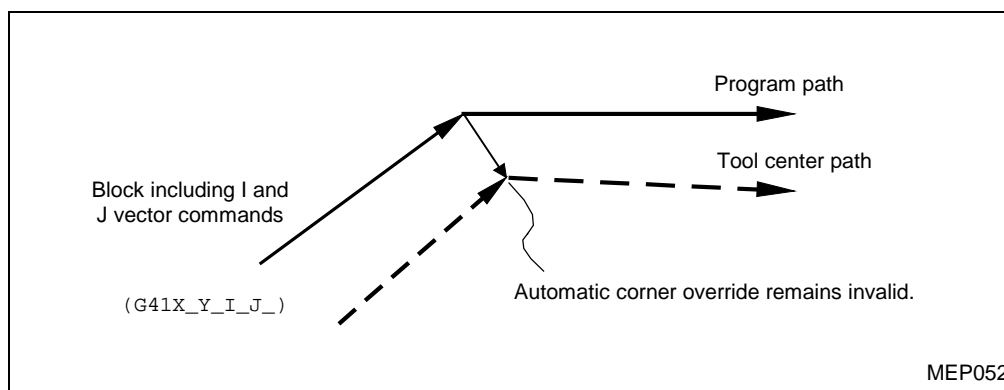
Function	Override at corners
Cutting feed override	Automatic corner override is applied after cutting feed override.
Cancellation of override	Automatic corner override is not cancelled by override cancel.
Limitation of feed rate	Valid (for the feed rate after automatic corner override)
Dry run	Automatic corner override is invalid.
Synchronous feed	A synchronous feed rate is automatically corner-overridden.
Skip (G31)	During tool-diameter offset, G31 will result in a program error.
Machine lock	Valid
G00	Invalid
G01	Valid
G02, G03	Valid

7. Precautions

- Automatic corner override is valid only during the G01, G02 or G03 modes; it is invalid during the G00 mode. Also, when the command mode is changed over from G00 to G01, G02 or G03 (or vice versa) at a corner, automatic corner override is not performed on the G00-containing block at that corner.
- Even in the automatic corner override mode, automatic corner override is not performed until the tool-diameter offset mode has been set.
- Automatic corner override does not occur at corners where tool diameter offsetting is to start or to be cancelled.



4. Automatic corner override does not occur at corners where I, J and K vector commands for tool-diameter offset are to be executed.



5. Automatic corner override occurs only when crossing point is calculated. Crossing points can not be calculated in the following case:
 - Four or more blocks that do not include move command are set in succession.
6. In case of circular interpolation, the deceleration zone refers to the length of arc.
7. The parameter-set angle of inner corner is applied to the angle existing on the program path.
8. If the maximum angle is set to 0 in the angle parameter, then no automatic corner override will be performed.
9. If the override is set to 0 or 100 in the override parameter, then no automatic corner override can be performed.

7-8 Tapping Mode: G63

1. Function and purpose

Command G63 enters the NC unit into a control mode suitable for tapping. This mode has the following features:

- The cutting feed override is fixed at 100%.
- Commands for deceleration at block-to-block connections are invalidated.
- The feed hold function is invalidated.
- The single-block function is invalidated.
- Tapping-mode signal is output.

The G63 command mode is cleared by exact-stop check mode command G61, automatic corner override command G62 or cutting mode command G64.

2. Programming format

G63

7-9 Cutting Mode: G64

1. Function and purpose

Command G64 enters the NC unit into a control mode proper to obtain smoothly cut surfaces. Unlike the exact-stop check mode (G61 command mode), the cutting mode allows the next block to be executed without decelerating/stopping the machine between cutting feed blocks.

G64 command mode (initial mode of the NC) is cleared by exact-stop check mode command G61, automatic corner override command G62 or tapping mode command G63.

2. Programming format

G64

7-10 Inverse Time Feed: G93 (Option)

1. Function and purpose

When tool-diameter offsetting is performed for a smooth linear or circular small-line-segment command, differences will occur between the shape defined in the program and that existing after tool-diameter offsetting. The feed commands with G94 and G95 only apply for the tool path existing after offsetting, and the tool speed at the point of cutting (that is, along the shape defined in the program), therefore, will not be kept constant so that the resulting speed fluctuations will cause seams on the surface machined.

Setting of an Inverse Time Feed command code makes constant the processing time for the corresponding block of the machining program, and thus provides control to ensure a constant machining feed rate at the point of cutting (along the programmed shape).

Setting of command code G93 specifies the inverse time assignment mode.

In G93 mode, the reciprocal of the machining time for the block of cutting command code G01, G02 or G03 is to be assigned using an F-code. Data that can be assigned with address F is from 0.001 to 99999.999.

The feed rate for the corresponding block is calculated (by NC) from the commanded length of the program block and the value of the F-code.

- For linear interpolation (G01)

$$\text{F-code value} = \frac{[\text{Speed}]}{[\text{Distance}]}$$

[Speed] : mm/min (for metric system) or inches/min (for inch system)

[Distance] : mm (for metric system) or inches (for inch system)

- For circular interpolation (G02 or G03)

$$\text{F-code value} = \frac{[\text{Speed}]}{[\text{Arc radius}]}$$

[Speed] : mm/min (for metric system) or inches/min (for inch system)

[Arc radius] : mm (for metric system) or inches (for inch system)

2. Programming formats

- Linear interpolation: G93 G01 Xx₁ Yy₁ Ff₁

- Circular interpolation: G93 G02 Xx₁ Yy₁ Rr₁ Ff₁

(Code G03 can be used, instead of G02, and code I, J and/or K instead of R.)

3. Precautions

- Code G93, which belongs to the same G-code group as that of G94 (feed per minute) and G95 (feed per revolution), is a modal G-code.
- In G93 mode, since F codes are not handled as modal codes, they must be set for each block. The absence of F-code results in alarm **816 FEEDRATE ZERO**.
- Setting of F0 during G93 mode results in alarm **816 FEEDRATE ZERO**.
- For a corner insertion block during tool-diameter offsetting, the F-code value in the previous block is regarded as the inverse time command value.
- A modal F-code must be set if the G93 mode is changed over to G94 or G95.

4. Description of alarms

No.	Message	Description
940	NO INVERSE TIME OPTION	The Inverse Time Feed option is not present.
941	G93 MODE	An illegal G-code* has been set during G93 mode.

* Illegal G-codes are:

G31	Skip
G33	Threading
G7□, G8□	Fixed Cycle

5. Sample program

```

N01 G90 G00 X-80. Y80.
N02 G01 G41 X0 Y0 D11 F500
N03      X200.
N04 G93 G02      Y-200.R100. F5
N05      G03      Y-400.R100. F5
N06      G02      Y-600.R100. F5
N07 G94 G01 X0      F500
N08      Y0
N09 G40      X-80. Y80.
N10 M02

```

----- D11 = 10mm
- - - - - D11 = 20mm

MEP053

In this example, set data as follows if the machining speed in the circular-interpolation blocks is to be made equal to 500 mm/min that is specified for the linear-interpolation block of G01:

$$\text{F-code value} = \frac{[\text{Speed}]}{[\text{Arc radius}]} = \frac{500}{100}$$

8 DWELL FUNCTIONS

The start of execution of the next block can be delayed using a G04 command. Depending on the respective feed mode (G94: feed-per-minute, G95: feed-per-revolution), the type of G04 command value (specification of time or number of revolutions) can automatically be identified. In addition, the remaining time of dwell can be nullified by a multi-step skip function.

8-1 Dwell Command in Time: (G94) G04

1. Function and purpose

Setting command G04 in the feed-per-minute mode (command G94) delays the start of execution of the next block for the specified time.

2. Programming format

G94 G04 X_

or

G94 G04 P_

Data can be set in 0.001 seconds.

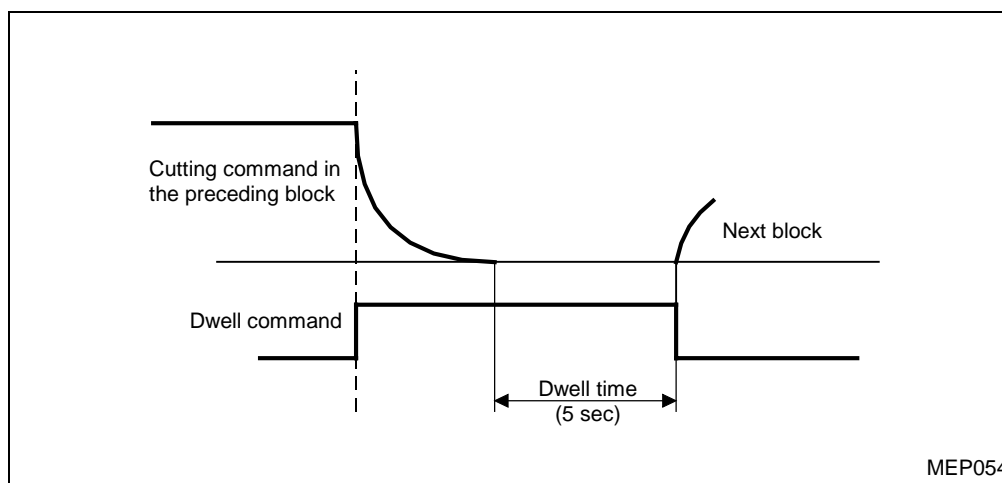
For address P, the decimal point is not available. The part after the decimal point will be ignored.

3. Detailed description

- The setting range for dwell time is as follows:

Unit of data setting	Range for address X	Range for address P
0.01/0.001 mm	0.001 to 99999.999 (sec)	1 to 99999999 (× 0.001 sec)
0.0001 inches	0.001 to 99999.999 (sec)	1 to 99999999 (× 0.001 sec)

- The count for the dwell command which is preceded by a block with cutting-feed command is not started until the movement of the preceding block has been brought to a complete stop.



If the dwell command is given in one block together with an M-, S- T- or B-code, the dwell count and the execution of the respective code will be started at the same time.

- The dwell function is valid during the interlock mode.
- The dwell function is also valid during machine lock. It can be, when required, immediately terminated using bit 4 of user parameter **F93**.

4. Sample programs

- When data is to be set in 0.01 mm, 0.001 mm or 0.0001 inches:

G04 X500	Dwell time = 0.5 sec
G04 X5000	Dwell time = 5.0 sec
G04 X5.	Dwell time = 5.0 sec
G04 P5000	Dwell time = 5.0 sec
G04 P12.345	Dwell time = 0.012 sec

- When data is to be set in 0.0001 inches and included before G04:

X5. G04	Dwell time = 50 sec (Equivalent to X50000G04.)
---------	------------------------------------------------

8-2 Dwell Command in Number of Revolutions: (G95) G04

1. Function and purpose

Setting command G04 in the feed-per-revolution mode (command G95) suspends the start of execution of the next block until the spindle has rotated the specified number of revolutions.

2. Programming format

G95 G04 X_

or

G95 G04 P_

Data can be set in 0.001 revolutions.

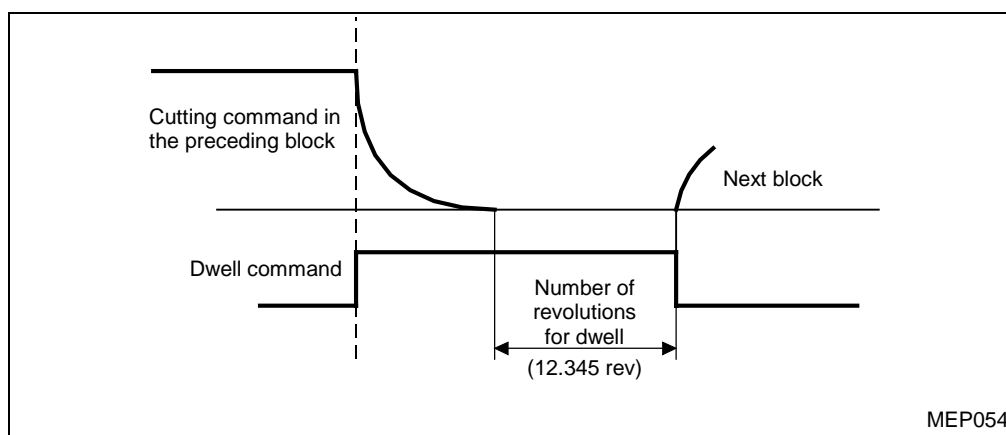
For address P, the decimal point is not available. The part after the decimal point will be ignored.

3. Detailed description

- The setting range for number of dwell revolutions is as follows:

Unit of data setting	Range for address X	Range for address P
0.01/0.001 mm	0.001 to 99999.999 (rev)	1 to 99999999 (× 0.001 rev)
0.0001 inches	0.001 to 99999.999 (rev)	1 to 99999999 (× 0.001 rev)

- The count for the dwell command which is preceded by a block with cutting-feed command is not started until the movement of the preceding block has been brought to a complete stop.



If the dwell command is given in one block together with an M-, S- T- or B-code, the dwell count and the execution of the respective code will be started at the same time.

-
3. The dwell function is valid during the interlock mode. That is, dwell is executed even during the interlock mode.
 4. The dwell function is also valid during machine lock.
 5. During rest of the spindle, dwell count is also halted. When the spindle restarts rotating, dwell count will also restart.
 6. If the bit 2 of parameter **F92** is set to 1, dwell command value is always processed in time specification.

- NOTE -

9 MISCELLANEOUS FUNCTIONS

9-1 Miscellaneous Functions (M3-Digit)

Miscellaneous functions, which are also referred to as M-code functions, give spindle forward/backward rotation and stop commands, coolant on/off commands, and other auxiliary commands for the NC machine.

For the NC unit, these functions must be selected using M3-digit data (three-digit data preceded by address M). Up to four sets of M3-digit data can be included in one block.

Example: G00 Xx₁ Mm₁

If five or more sets of M3-digit data are set, only the last four sets will become valid.

Setting of an M code not listed in the machine specifications will result in an alarm **228 ILLEGAL M CODE**. Refer to the machine specification for description of each M-code.

For M-codes M00, M01, M02 and M30, the next block of data is not read into the input buffer since pre-reading is disabled automatically.

The M-codes can be included in any block that contains other command codes. If, however, the M-codes are included in a block that contains movement commands, then the execution priority will be either

- the M-code functions are executed after completion of movement, or
- the M-code functions are executed together with movement.

It depends on the machine specifications whether which is applied.

The following lists six types of special M-code functions:

1. Program Stop: M00

When this M-code is read, the tape reader will stop reading the subsequent block. On the NC side, spindle rotation, coolant on/off control, and other auxiliary machine functions are deactivated at the same time.

The machine operation can be restarted by pressing the CYCLE START button on the operation panel.

2. Optional Stop: M01

When the M01 code is read with the **[OPTIONAL STOP]** menu function set to ON, the tape reader will stop operating to perform the same function as M00.

The M01 command will be ignored if the **[OPTIONAL STOP]** menu function is set OFF.

Example:

```

:
N10 G00 X1000
N11 M01
N12 G01 X2000 Z3000 F600
:

```

<**[OPTIONAL STOP]** menu function status and operation>

If the menu function is on, operation stops at N11.

If the menu function is off, operation does not stop at N11 and N12 is executed.

3. Program End: M02 or M30

Usually, the program end command is given in the final block of machining program. Use this command mainly for reading data back to the head of the program during memory operation, or rewinding the tape. The NC unit is automatically reset after tape rewinding and execution of other command codes included in that block.

Automatic resetting by this command cancels both modal commands and offsetting data, but the designated-position display counter is not cleared to zero.

The NC unit will stop operating when tape rewinding is completed (the automatic run mode lamp goes out). To restart the NC unit, the CYCLE START button must be pressed.

Beware that if, during the restart of the NC unit following completion of M02 or M30 execution, the first movement command has been set in a coordinate word only, the valid mode will be the interpolation mode existing when the program ended. It is recommended, therefore, that the first movement command be given with an appropriate G-code.

4. Subprogram Call/End: M98, M99

Use M98 or M99 to branch the control into a subprogram or to recall it back to the calling program.

<Internal processing by the NC unit when M00, M01, M02 or M30 is used>

After M00, M01, M02 or M30 has been read, data pre-reading is automatically aborted.

Note 1: M00, M01, M02 and M30 output independent signals, which will be cancelled by pressing the RESET key.

Note 2: Manual data input (MDI) can be used to designate M02 or M30.

As with tape operation, MDI allows simultaneous designation of M02 or M30 with other commands.

Note 3: Tape rewinding is performed only when the tape reader has a rewinding function.

Note 4: For memory operation, M30 is issued automatically, even if M02 or M30 is not present at the end of the program.

9-2 No. 2 Miscellaneous Functions (B3-Digit)

The No. 2 miscellaneous functions are used for positioning the index table. For our NC unit, these functions must be designated using a three-digit value (from 0 to 359) preceded by address B.

The B-code functions can be designated together with any other commands. If included in a block that contains movement commands, however, the B-code is always executed only after execution of the movement commands.

No. 4 axis name and No. 2 miscellaneous function are not permitted to use an identical address.

10 SPINDLE FUNCTION

This function allows a spindle speed to be set using the numerical command of five digits that follows address S. During selection of gears, however, care must be taken since the maximum spindle speed is limited according to the type of gear selected using an M-code.

During gear selection for tapping, due consideration must also be given to ensure the minimum spindle acceleration/deceleration time. The maximum speed of each gear and the recommended maximum speed for tapping depend on the particular machine specifications. Refer to the machine-operating manual for further details.

- NOTE -

11 TOOL FUNCTIONS

11-1 Tool Function (T3-Digit)

Tool function is also referred to as T-code function. This function is used to select tool numbers. For our NC unit, T function allows you to select a maximum of 1,000 numbers (0 to 999) using three-digit command data preceded by address T. The maximum number of tool numbers you can select for your machine, however, differs according to the machine specifications. Refer to the relevant machine specification for further details.

Selection of an illegal T-code results in an alarm **294 NO TOOL SELECT (INCORRECT TNo.)**.

The T-code function can be used with any other commands. If, however, the T-code is included in a block that contains a movement command, the execution priority will be either one of the following two types, depending on the machine specifications:

- The T-code function is executed after the movement command has been executed.
- The T-code function is executed together with the movement command.

11-2 Tool Function (T8-Digit)

This function allows you to select a maximum of 100,000,000 numbers (0 to 99999999) using eight-digit command data preceded by address T. Only one number can be included in the same block.

Set bit 4 of parameter **F94** to 0 to select the group-number designation for T-code command, or set this bit to 1 to select the tool-number designation.

- NOTE -

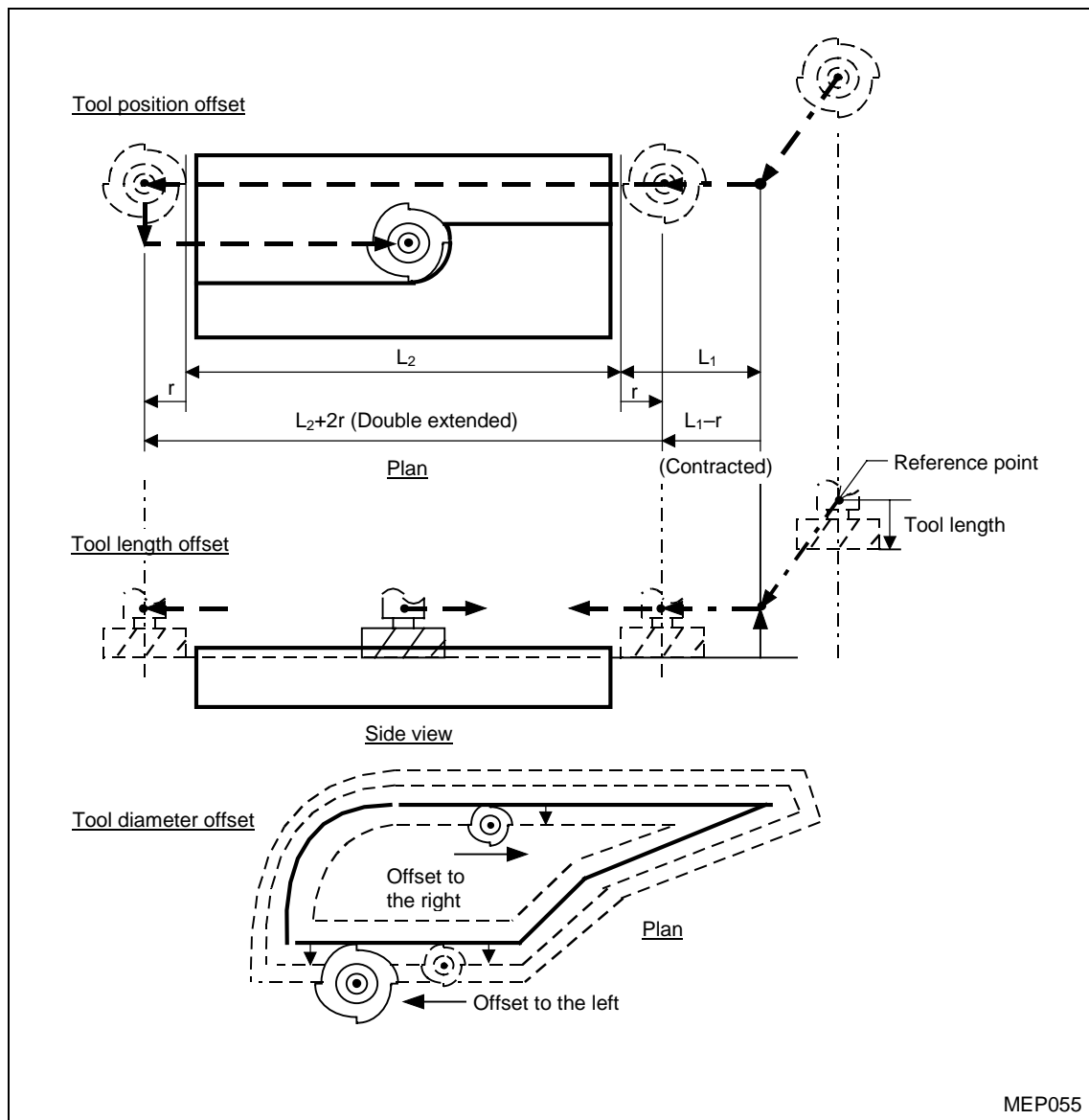
12 TOOL OFFSET FUNCTIONS

12-1 Tool Offset

1. Overview

As shown in the diagram below, three types of basic tool offset functions are available: tool position offset, tool length offset, and tool diameter offset.

These three types of offset functions use offset numbers for designation of offset amount. Set the amount of offset directly using the operation panel or by applying the function of programmed parameter input. MAZATROL tool data can also be used for tool length offset or tool diameter offset operations according to the parameter setting.



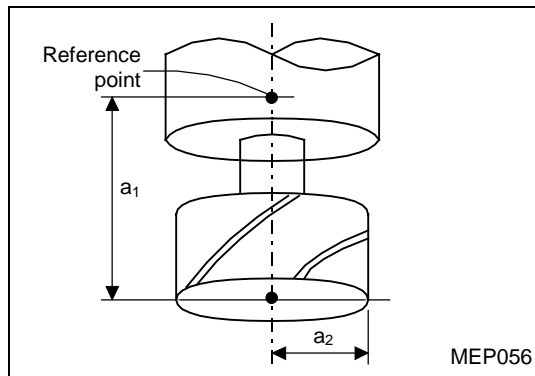
2. Selecting the amounts of tool offset

The amounts of tool offset corresponding to the offset numbers must be prestored into the tool offset memories of the NC unit using the operation panel or programmed parameter input function.

The mounts of tool offset can be selected using one of the following two types:

A. Type A (Standard)

The same amount of offset will be set if identical offset numbers are selected using commands D and H.

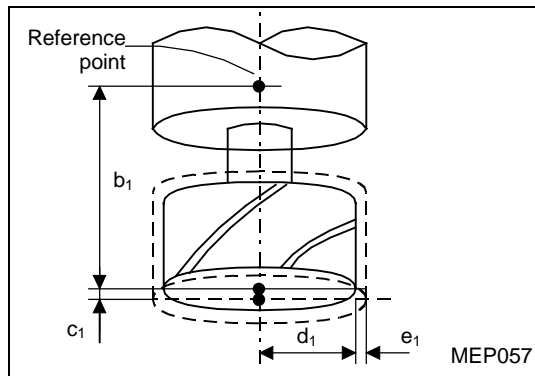


$$(Dn) = a_n$$

$$(Hn) = a_n$$

B. Type B (Optional)

For tool length offset, the total sum of the shape offset amount and the wear offset amount can be set using command H. For tool diameter offset, the total sum of the shape offset amount and the wear offset amount can be set using command D.



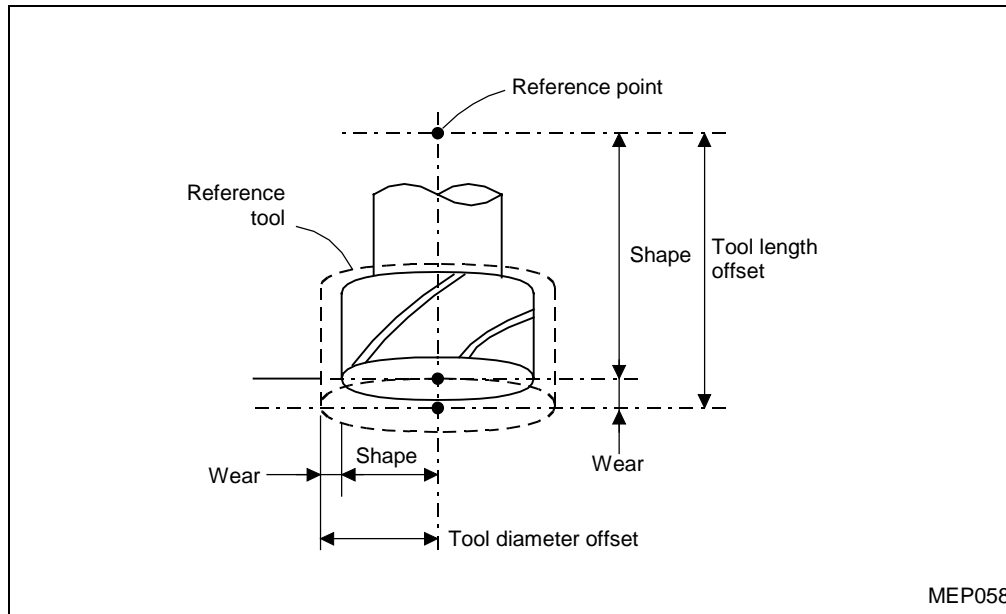
$$(Hn) = b_n + c_n$$

$$(Dn) = d_n + e_n$$

3. Tool offset memories

For storage of tool offset data to be set/selected, two types of tool offset memories are provided: Type A and Type B.

Type of tool offset memory	Discrimination between length and diameter offsets	Discrimination between shape and wear offsets
Type A	Unavailable	Unavailable
Type B	Available	Available



A. Type A (Standard)

As listed in the table below, one offset data is given for one offset number. No distinction is drawn between length, diameter, shape and wear offset amounts. That is, one set of offset data comprises all these four factors.

$$\begin{aligned}
 (D1) &= a_1, & (H1) &= a_1 \\
 (D2) &= a_2, & (H2) &= a_2 \\
 \vdots & & \vdots & \\
 (Dn) &= a_n, & (Hn) &= a_n
 \end{aligned}$$

Offset No.	Offset amount
1	a_1
2	a_2
3	a_3
\vdots	\vdots
\vdots	\vdots
n	a_n

B. Type B (Optional)

As listed in the table below, two types of offset data can be set for one offset number. That is, different amounts of shape offset and wear offset can be set for each of the selected tool length and the selected tool diameter.

Use command H to select offset data concerning the tool length, and use command D to select offset data concerning the tool diameter.

$$(H1) = b_1 + c_1, \quad (D1) = d_1 + e_1$$

$$(H2) = b_2 + c_2, \quad (D2) = d_2 + e_2$$

$$\vdots \quad \vdots$$

$$(Hn) = b_n + c_n, \quad (Dn) = d_n + e_n$$

Offset No.	Tool length (H)		Tool diameter (D) / (Position offset)	
	Shape offset	Wear offset	Shape offset	Wear offset
1	b_1	c_1	d_1	e_1
2	b_2	c_2	d_2	e_2
3	b_3	c_3	d_3	e_3
\vdots	\vdots	\vdots	\vdots	\vdots
\vdots	\vdots	\vdots	\vdots	\vdots
n	b_n	c_n	d_n	e_n

4. Tool offset numbers (H/D)

Tool offset numbers can be selected using address H or D.

- Use address H to offset the selected tool length. Use address D to offset the selected tool position or the selected tool diameter.
- Once a tool offset number has been selected, it will remain unchanged until a new H or D is used.
- Offset numbers can be set only once for one block. If offset numbers are set more than once for one block, only the last offset number will be used.
- The maximum available number of sets of offset numbers is as follows:
Standard: 128 sets: H01 to H128 (D01 to D128)
Optional: 512 sets: H01 to H512 (D01 to D512)
- An alarm **839 ILLEGAL OFFSET No.** will result if these limits are exceeded.
- The offset data range is as listed in the table below.
Offset data corresponding to each offset number must be set on the **TOOL OFFSET** display beforehand.

Shape offset stroke		Wear offset	
Metric system	Inch system	Metric system	Inch system
± 1999.9999 mm	± 84.50000 in.	± 99.9999 mm	± 9.99999 in.

Note: Tool offset numbers can be set only during the offsetting mode. They will become invalid if set during other modes.

5. Number of sets of tool offset numbers

The maximum available number of sets of tool offset numbers depends on the particular machine specifications.

	Number of tool offset combinations (max.)
Standard specifications	128
Optional specifications	512

Note: The maximum available number of sets of tool offset numbers under optional machine specifications refers to the total number of sets of tool offset numbers including those available under the standard machine specifications.

12-2 Tool Length Offset/Cancellation: G43, G44/G49

1. Function and purpose

Commands G43 and G44 allow the ending point of execution of move commands to be moved through the previously set offset amount for each axis. Any deviations between programmed tool lengths/diameters and actual lengths/diameters can be set as offset data using these commands to make the program more flexible.

2. Programming format

G43 Zz Hh Tool length offset +
 G44 Zz Hh Tool length offset –
 G49 Zz Tool length offset cancellation

3. Detailed description

The maximum available number of sets of offset numbers is as follows:

Standard: 128 sets : H1 to H128

Optional: 512 sets : H1 to H512

where the maximum available number of sets of offset numbers refers to the total number of sets of offset numbers including those concerning the tool length, the tool position, and the tool diameter.

The following represents the relationship between the programming format and the stroke of movement after offsetting:

Z-axial moving stroke

G43 Z ± z Hh $\pm z + \{ \pm h_1 - (\pm \ell h_0) \}$ To offset by tool offset stroke in the positive direction

G44 Z ± z Hh $\pm z + \{ \pm h_1 - (\pm \ell h_0) \}$ To offset by tool offset stroke in the negative direction

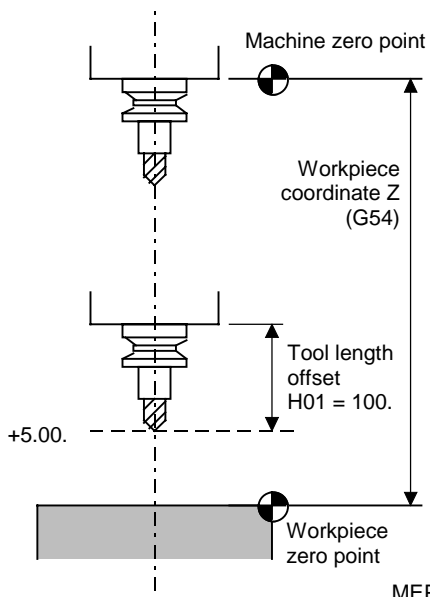
G49 Z ± z $\pm z - (\pm \ell h_1)$ Z-axial moving stroke

where, ℓh_1 : Offset stroke for offset No. h_1

ℓh_0 : Offset stroke prior to G43 and G44 blocks

Irrespective of whether absolute data commands or incremental data commands are used, the actual ending point coordinates are calculated by offsetting the programmed end point coordinate by the offset amount. The G49 command mode (tool length offset cancellation) will be set when power is turned on or after code M02 has been executed.

4. Sample programs



1. For absolute data input; H01 = 100.

```

N001 G90 G94 G00 G40 G80
N002 G91 G28 Z0
N003 T01 T00 M06
N004 G90 G54 X-100. Y0
N005 G43 Z5. H01
N006 G01 Z-50. F100

```

2. For incremental data input; H01 = 100.

```

N001 G90 G94 G00 G40 G80
N002 G91 G28 Z0
N003 T01 T00 M06
N004 G90 G54 X-100. Y0
N005 G91 G43 Z5. H01
N006 G01 Z-55. F100

```

MEP059

5. Supplement

- Tool length offset data can be set for the X-axis, the Y-axis, and additional axes, as well as the Z-axis. Whether the offset data is to be used for the Z-axis only or for the axes that correspond to commands G43 or G44 can be selected using bit 3 of parameter **F92**.
- Even if multiple axis addresses are programmed for one block, offsetting will be performed on only one of those axes and the priority in this case will be as follows:
 $\alpha > Z > Y > X$ (α : Additional axis)

Example:

G43 XX_1 Hh ₁	}	To positively offset the X-axis and to cancel the offset mode
⋮		
G49 XX_2		
G44 YY_3 Hh ₃	}	To negatively offset the Y-axis and to cancel the offset mode
⋮		
G49 YY_4		
G43 $\alpha\alpha_5$ Hh ₅	}	To offset an additional axis and to cancel the offset mode
⋮		
G49 $\alpha\alpha_6$		
G43 XX_7 YY_7 ZZ_7 Hh ₇		To positively offset the Z-axis.

- Offsetting will always be performed on the Z-axis if no axis addresses are programmed for the particular block.

Example:

G43 Hh ₁	}	To offset the Z-axis and to cancel the offset mode
⋮		
G49		

4. If reference point (home position) return is performed during offsetting, the offsetting mode will be cancelled after completion of the returning operation.

Example:

G43 Hh ₁	}	Upon completion of return to the reference point (zero point), the offset stroke is cleared. The offset mode is also changed over to G49.
⋮		
G28 ZZ ₂		
G43 Hh ₁	}	After cancelling the offset on the Z-axis, the machine makes a return to the reference point (zero point).
G49 G28 ZZ ₂		

5. If command G49 or H00 is executed, offsetting will be immediately cancelled (the corresponding axis will move to clear the offset data to zero).
When using MAZATROL tool data, do not use G49 as a cancellation command code. If G49 is included in the program, interference with the workpiece may result since automatic cancellation will move the Z-axis in its minus direction through the distance equivalent to the tool length.
Use an H00 command, rather than a G49 command, if G43/G44 mode is to be cancelled temporarily.
6. An alarm **839 ILLEGAL OFFSET No.** will occur if an offset number exceeding the machine specifications is set.

12-3 Tool Position Offset: G45 to G48

1. Function and purpose

Command G45 or G46 allows the axis movement distance set previously in that block to be increased or decreased, respectively, according to the offset data. Likewise, command G47 or G48 extends or contracts the previously set distance by twice the offset stroke, respectively.

The maximum available number of sets of offset numbers is as follows:

Standard: 128 sets: D1 to D128

Optional: 512 sets: D1 to D512

where the maximum available number of sets of offset numbers refers to the total number of sets of offset numbers including those concerning the tool length, the tool position, and the tool diameter.

G45 command	G46 command
Extended thru offset stroke only	Contracted thru offset stroke only
<p>Internal calculation</p> <p>Moving stroke</p> <p>Starting point</p> <p>Ending point</p>	<p>Internal calculation</p> <p>Moving stroke</p> <p>Starting point</p> <p>Ending point</p>
G47 command	G48 command
Extended thru twice the offset stroke	Contracted thru twice the offset stroke
<p>Internal calculation</p> <p>Moving stroke</p> <p>Starting point</p> <p>Ending point</p>	<p>Internal calculation</p> <p>Moving stroke</p> <p>Starting point</p> <p>Ending point</p>

$$\begin{array}{c}
 \text{Hatched arrow} \\
 \text{(Program command value)}
 \end{array}
 \pm
 \begin{array}{c}
 \text{White arrow} \\
 \text{(Offset stroke)}
 \end{array}
 =
 \begin{array}{c}
 \text{Grey arrow} \\
 \text{(Moving stroke after offset)}
 \end{array}$$

2. Programming format

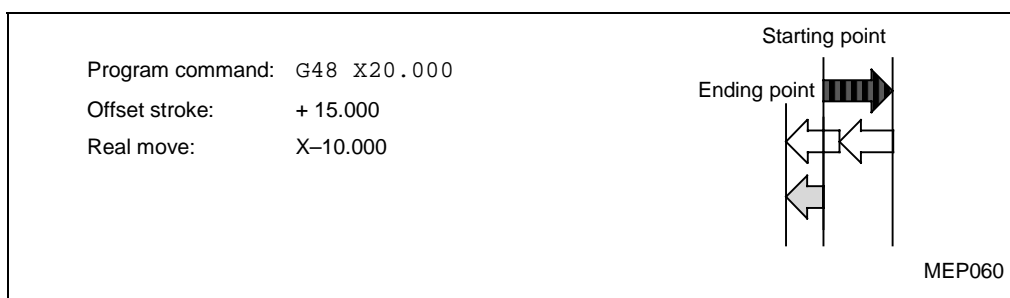
Command format	Function
G45 Xx Dd	To extend a moving stroke by the offset stroke which has been set in the offset memory.
G46 Xx Dd	To contract a moving stroke by the offset stroke which has been set in the offset memory.
G47 Xx Dd	To extend a moving stroke by twice the offset stroke which has been set in the offset memory.
G48 Xx Dd	To contract a moving stroke by twice the offset stroke which has been set in the offset memory.

3. Detailed description

- Programming based on incremental data is shown below.

Tape command	Stroke of movement by equivalent tape command (selected offset stroke = ℓ)	Example (with $x = 1000$)
G45 Xx Dd	$X \{ x + \ell \}$	$\ell = 10 \quad X = 1010$ $\ell = -10 \quad X = 990$
G45 X-x Dd	$X - \{ x + \ell \}$	$\ell = 10 \quad X = -1010$ $\ell = -10 \quad X = -990$
G46 Xx Dd	$X \{ x - \ell \}$	$\ell = 10 \quad X = 990$ $\ell = -10 \quad X = 1010$
G46 X-x Dd	$X - \{ x - \ell \}$	$\ell = 10 \quad X = -990$ $\ell = -10 \quad X = -1010$
G47 Xx Dd	$X \{ x + 2\ell \}$	$\ell = 10 \quad X = 1020$ $\ell = -10 \quad X = 980$
G47 X-x Dd	$X - \{ x + 2\ell \}$	$\ell = 10 \quad X = -1020$ $\ell = -10 \quad X = -980$
G48 Xx Dd	$X \{ x - 2\ell \}$	$\ell = 10 \quad X = 980$ $\ell = -10 \quad X = 1020$
G48 X-x Dd	$X - \{ x - 2\ell \}$	$\ell = 10 \quad X = -980$ $\ell = -10 \quad X = -1020$

- Even if no offset numbers are set in the same block as that which contains commands G45 to G48, offsetting will be performed, based on previously stored tool position offset numbers.
- An alarm **839 ILLEGAL OFFSET No.** will occur if the designated offset number is an unavailable one.
- These G-code commands are not modal ones, and thus they are valid only for the designated block.
- These commands must be used in modes other than the fixed-cycle mode. They will be ignored if used in the fixed-cycle mode.
- The axis will move in reverse if internal calculation for changing the movement distance results in inversion of the direction of movement.



- The following lists how the machine operates if a movement distance of 0 using the incremental data command mode (G91) is programmed:

NC command	G45 X0 D01	G45 X-0 D01	G46 X0 D01	G46 X-0 D01
Equivalent command	X1234	X-1234	X-1234	X1234

D01: Offset number
1234: Offset amount for D01

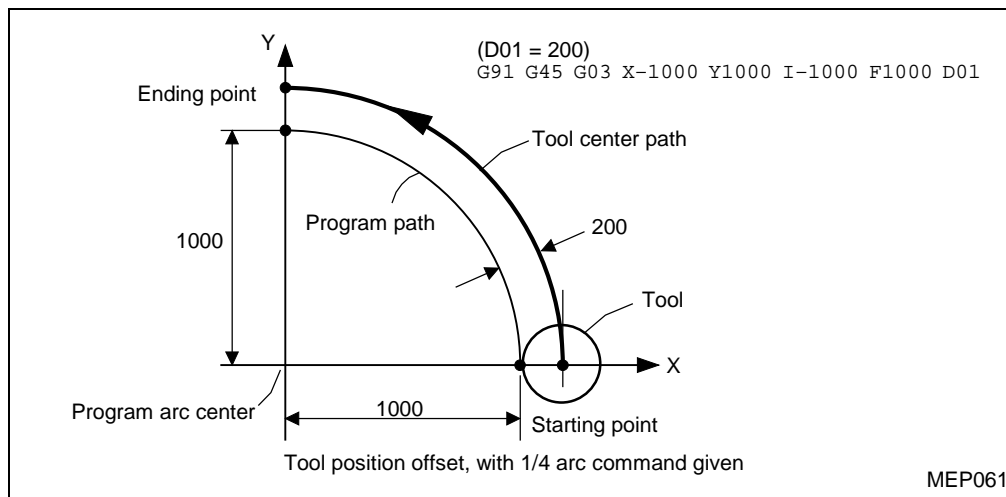
For absolute data commands, if the movement distance is set equal to 0, the block will be immediately completed and no movement through the offset distance will occur.

- When absolute data commands are used, each axis will also move from the ending point preset in the preceding block to the position set in the block that contains commands G45 through G48.

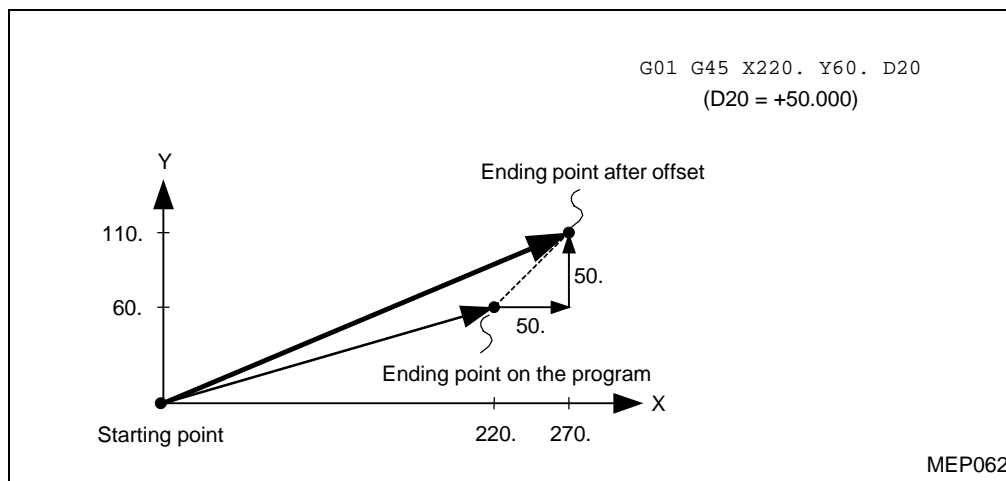
That is, when absolute data commands are used, offsetting will be performed according to the movement distance (increments in distance) set in that block.

4. Sample programs

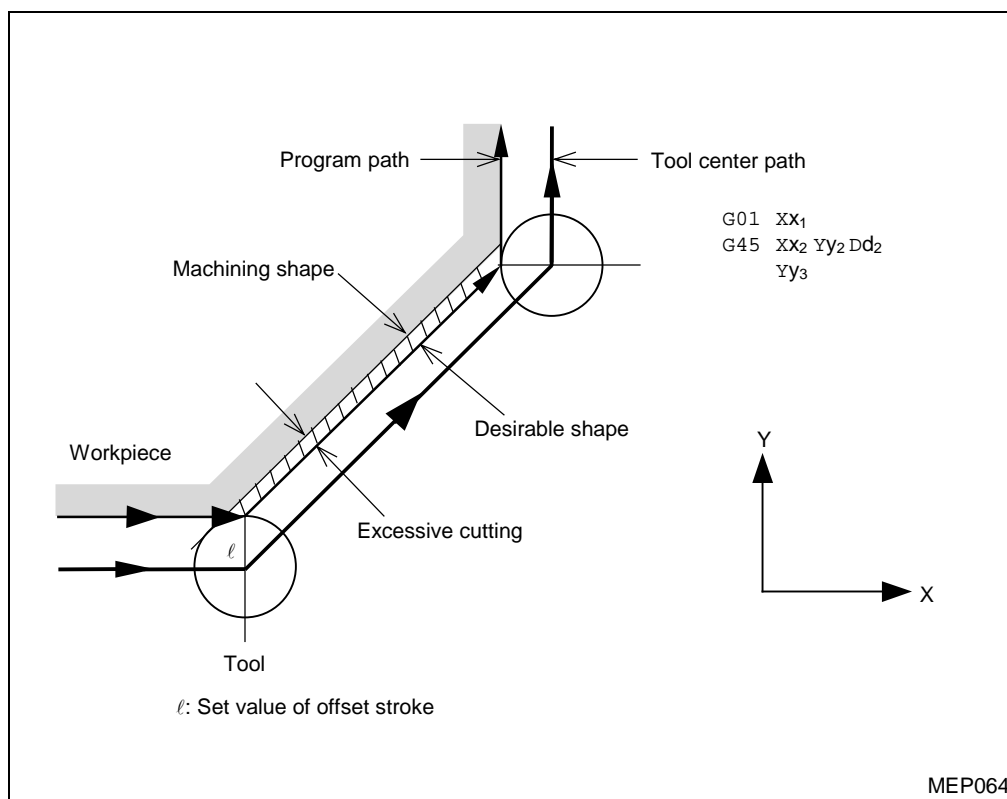
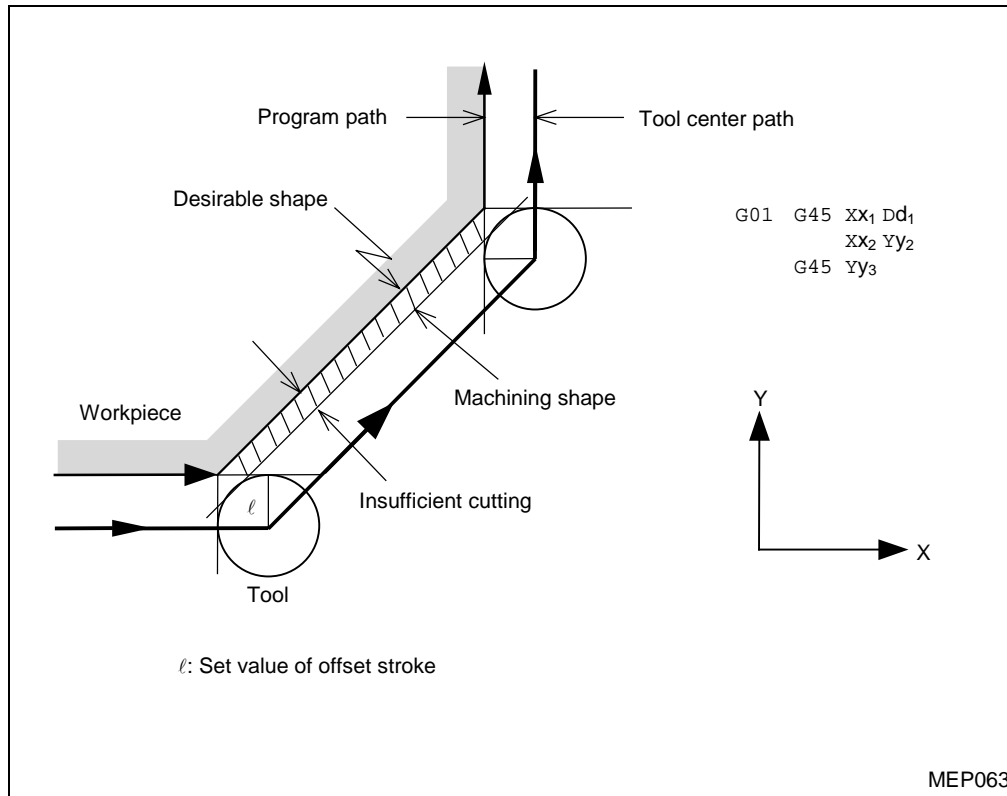
1. During arc interpolation, tool diameter offsetting using commands G45 to G48 can be done only for a 1/4, 1/2, or 3/4 circle whose starting and ending points are present on a coordinate axis passing through the arc center.



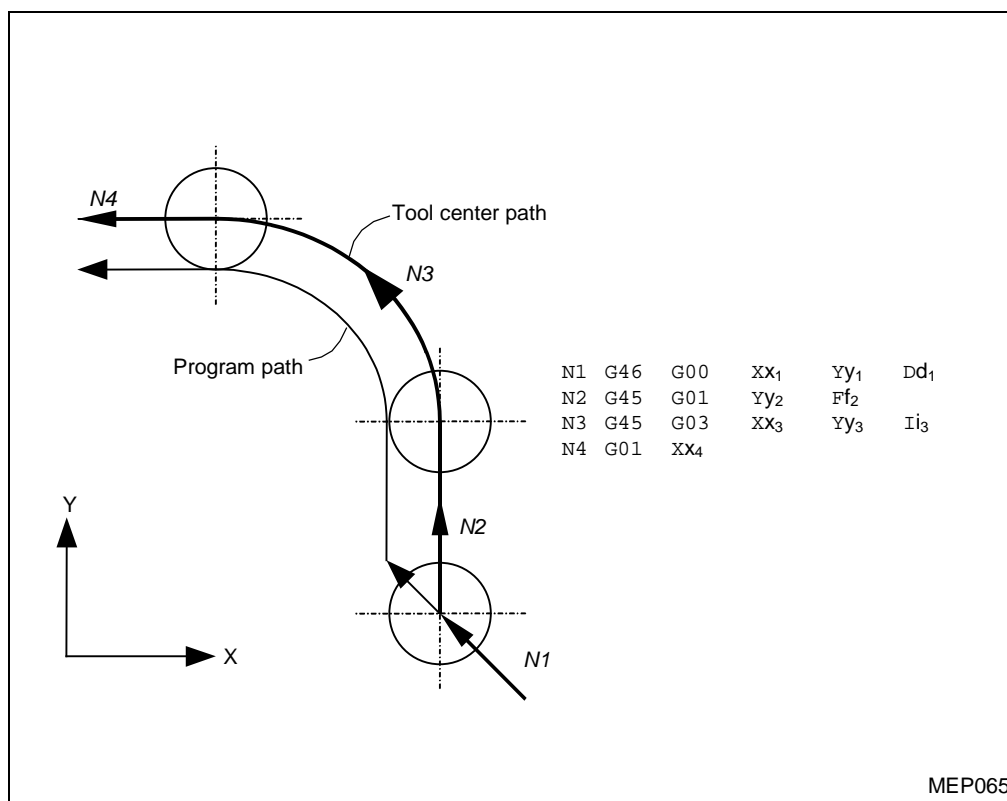
2. If an "n" number of axes are designated at the same time, the same amount of offsetting will be performed on all designated axes. This also applies to additional axes, but within the limits of the simultaneously controllable axis quantity.



Note: Use tool diameter offset commands G40, G41, or G42 if simultaneous offsetting of two axes is likely to result in excessive or insufficient cutting as shown below.



3. Cornering in a 1/4 circle



- [illegible]

N100	G91	G46	G00	X40.	Y40.	D01
N101	G45	G01	X100.	F200		
N102	G45	G03	X10.	Y10.	J10.	
N103	G45	G01	Y40.			
N104	G46	X0				
N105	G46	G02	X-20.	Y20.	J20.	
N106	G45	G01	Y0			
N107	G47	X-30.				
N108		Y-30.				
N109	G48	X-30.				
N110		Y30.				
N111	G45	X-30.				
N112	G45	G03	X-10.	Y-10.	J-10.	
N113	G45	G01	Y-20.			
N114		X10.				
N115		Y-40.				
N116	G46	X-40.	Y-40.			
N117	M02					

12-4 Tool Diameter Offset Function: G40, G41, G42

12-4-1 Overview

1. Function and purpose

Offsetting in any vectorial direction can be done according to the tool radius preselected using G-code commands G38 to G42 and D-code commands. This function is referred to as tool diameter offsetting.

2. Programming format

Command format	Function	Remarks
G40X_Y_	To cancel a tool diameter offset	
G41X_Y_	To offset a tool diameter (Left)	
G42X_Y_	To offset a tool diameter (Right)	
G38 I_J_	To change and hold an offset vector	These commands can be given during the diameter offset mode.
G39	To interpolate a corner arc	

3. Detailed description

The maximum available number of sets of offset numbers is as follows:

Standard: 128 sets: D1 to D128

Optional: 512 sets: D1 to D512

where the maximum available number of sets of offset numbers refers to the total numbers including those concerning the tool length, the tool position, and the tool diameter.

For tool diameter offsetting, all H-code commands are ignored and only D-code commands become valid.

Also, tool diameter offsetting is performed for the plane that is specified by either the plane selection G-code command or two-axis address code command appropriate for tool diameter offsetting. No such offsetting is performed for axes other than those corresponding or parallel to the selected plane. See Section 6-4 "Plane Selection" to select a plane using a G-code command.

12-4-2 Tool diameter offsetting

1. Tool diameter offsetting cancellation

Tool diameter offsetting is automatically cancelled in the following cases:

- After power has been turned on
- After the reset key on the NC operation panel has been pressed
- After M02 or M30 has been executed (these two codes have a reset function)
- After G40 (offsetting cancellation command) has been executed

In the offsetting cancellation mode, the offset vector becomes zero and the tool center path agrees with the programmed path.

Programs containing the tool diameter offset function must be terminated during the offsetting cancellation mode.

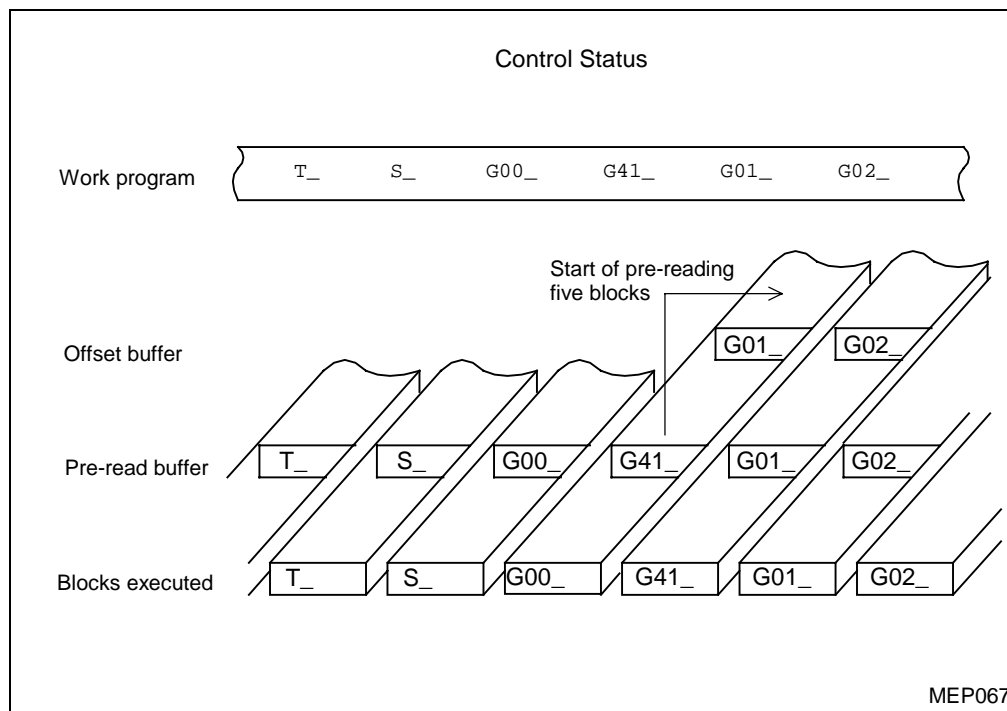
2. Startup of tool diameter offsetting

Tool diameter offsetting will begin during the offset mode when all the following three conditions are met:

- Command G41 or G42 has been executed.
- The offset number for tool diameter offsetting is larger than zero, but equal to or smaller than the maximum available offset number.
- The command used with the offsetting command is a move command other than those used for arc interpolation.

Offsetting will be performed only when reading of five blocks in succession is completed, irrespective of whether the single-block operation mode is used.

During offsetting, five blocks are pre-read and then calculation for offsetting is performed.



There are two types of offsetting startup operation: Type A and Type B.

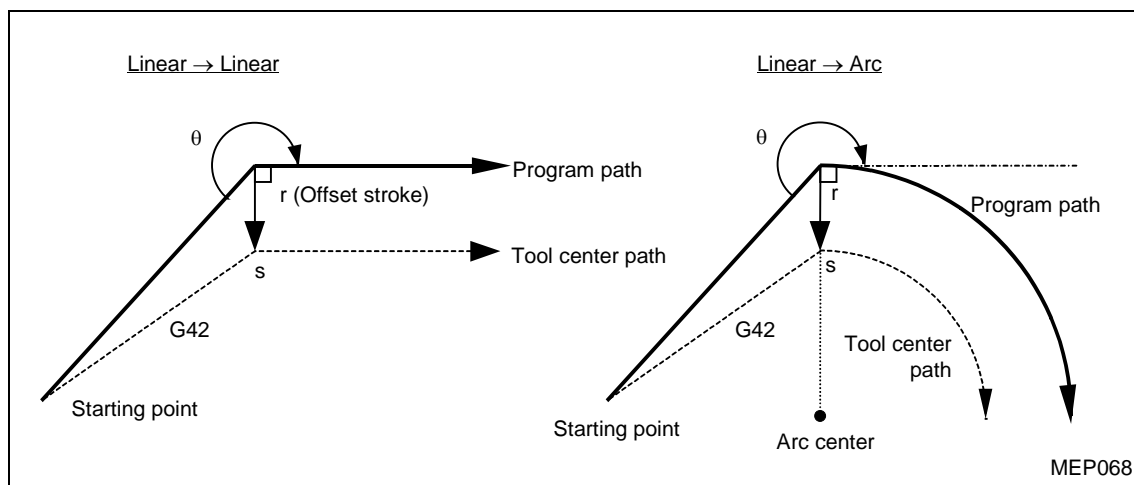
It depends on the setting of bit 4 of parameter **F92** whether Type A or Type B is automatically selected.

These two types of startup operation are similar to those of offsetting cancellation.

In the descriptive diagrams below, "s" signifies the ending point of single-block operation.

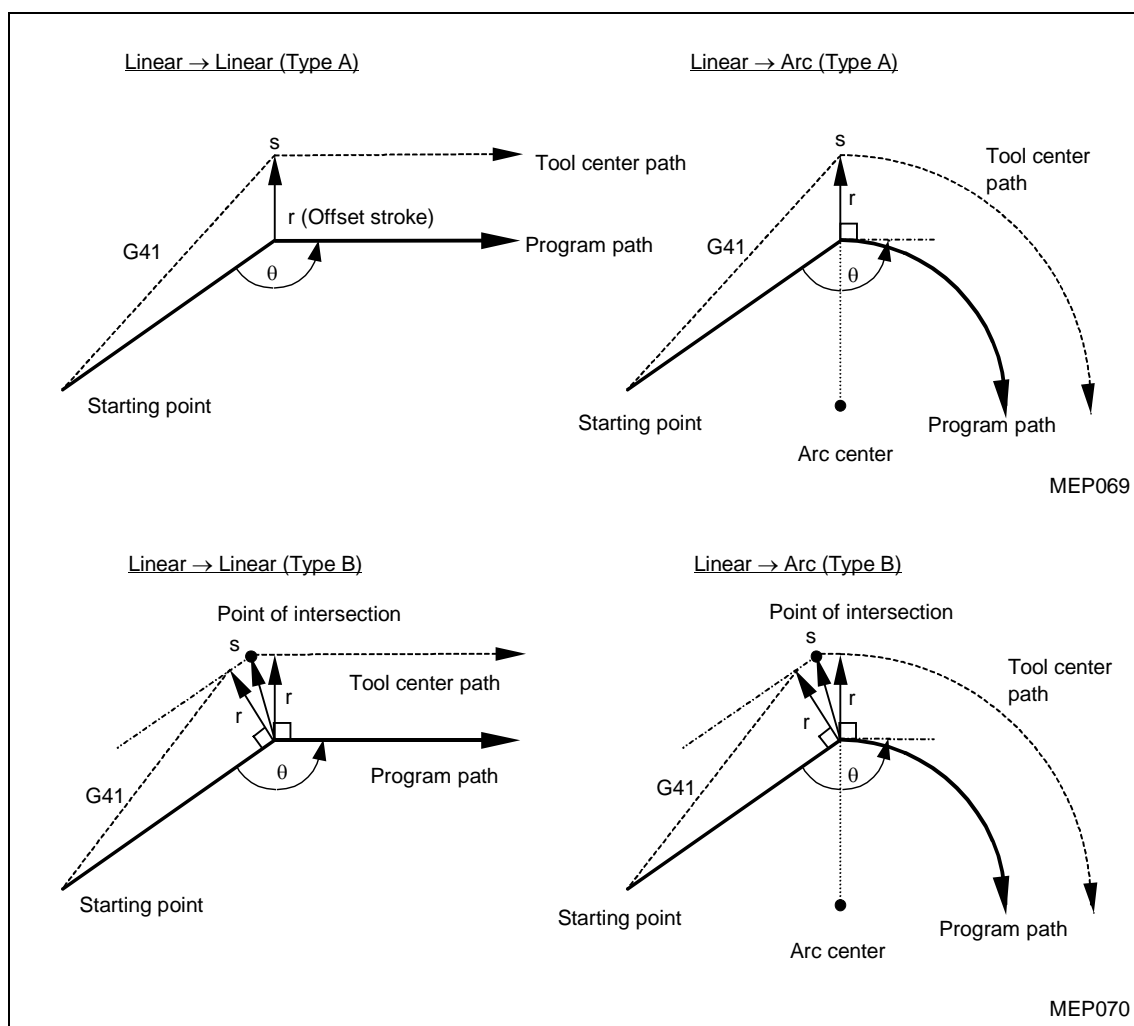
3. Tool diameter offsetting startup operation

A. For the corner interior



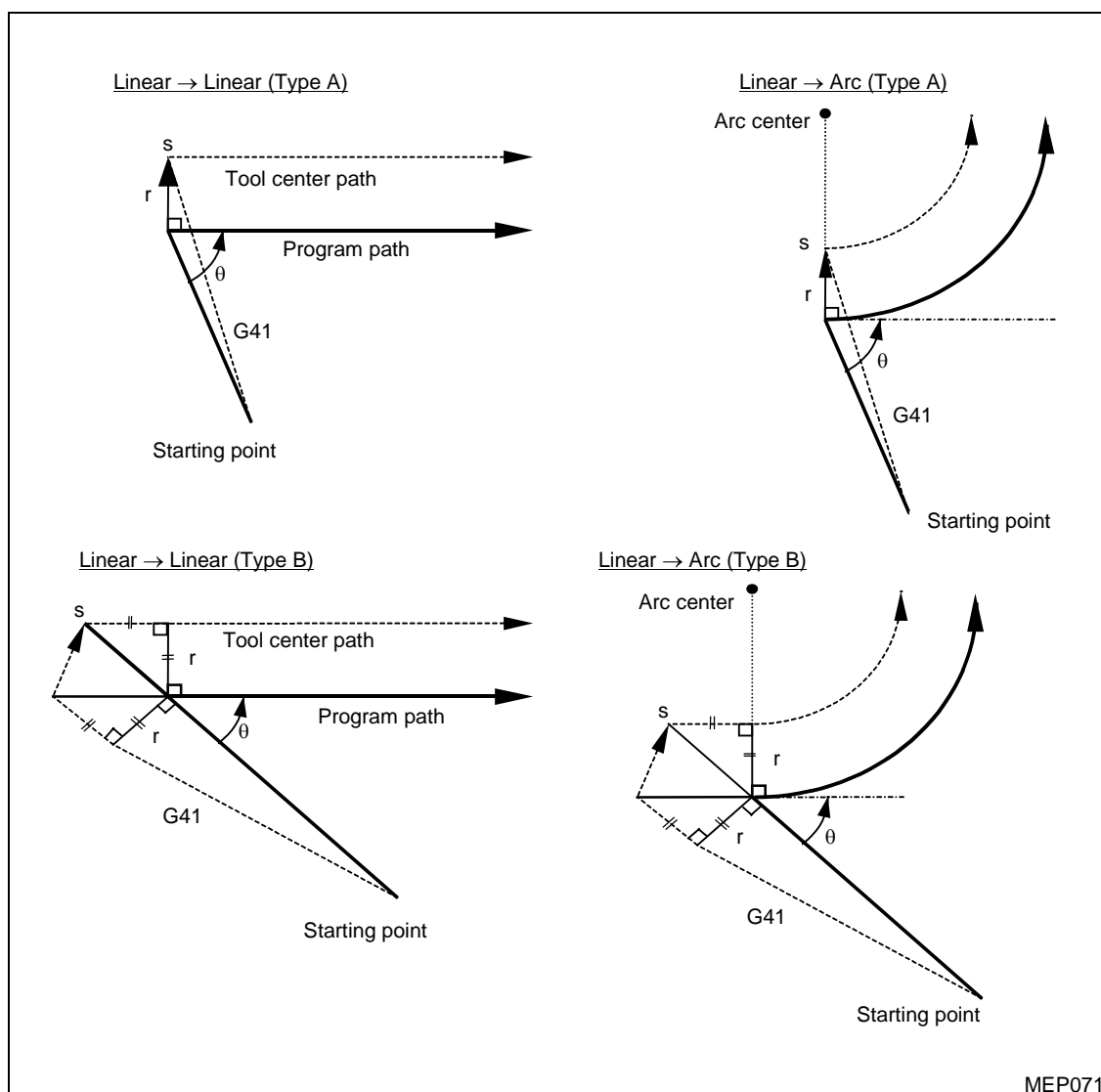
B. For the corner exterior (obtuse angle) [$90^\circ \leq \theta < 180^\circ$]

(Type A/B selection is possible with a predetermined parameter.)



C. For the corner exterior (sharp angle) [$\theta < 90^\circ$]

(Type A/B selection is possible with a predetermined parameter.)



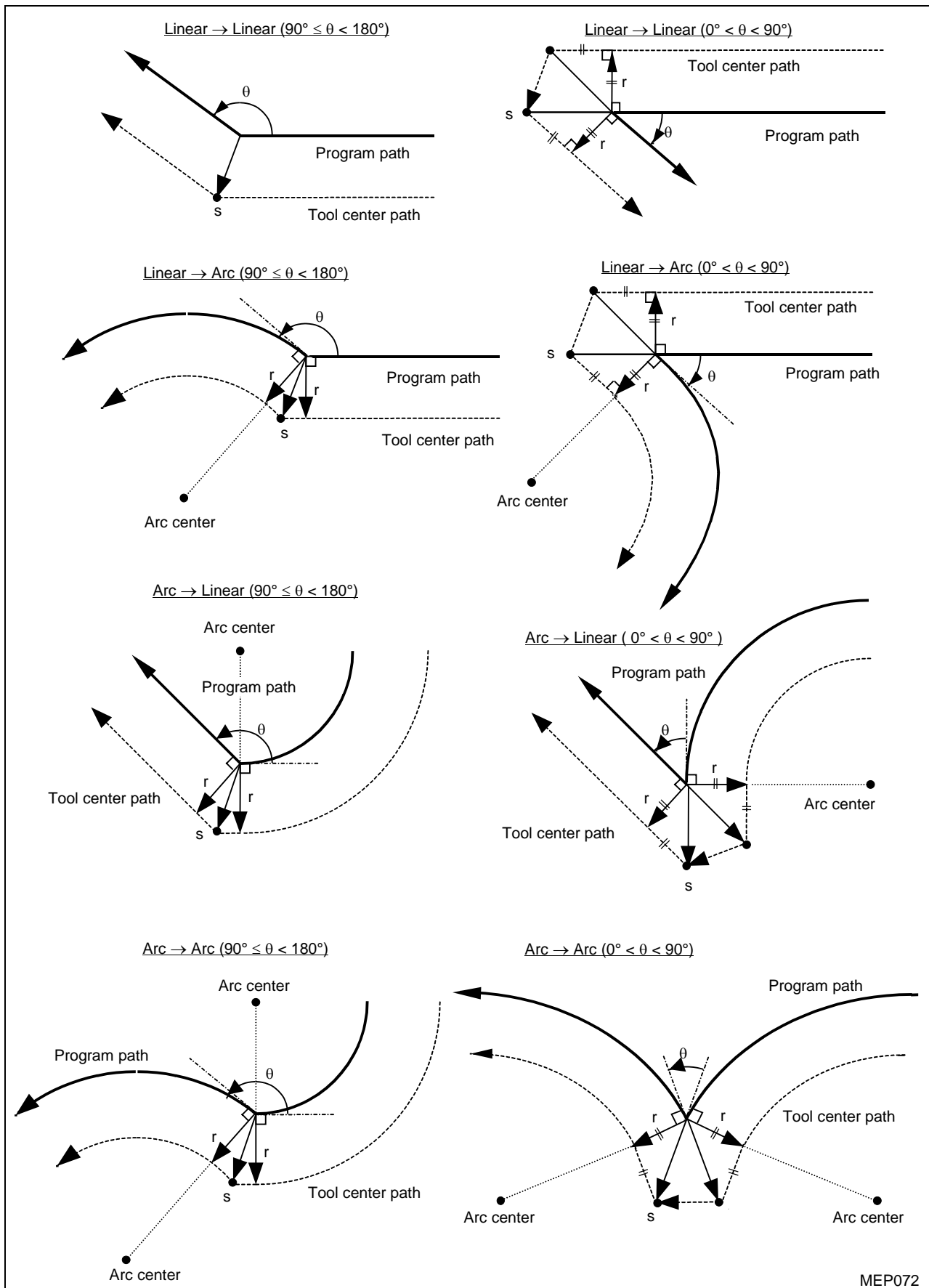
MEP071

4. Operation during the offset mode

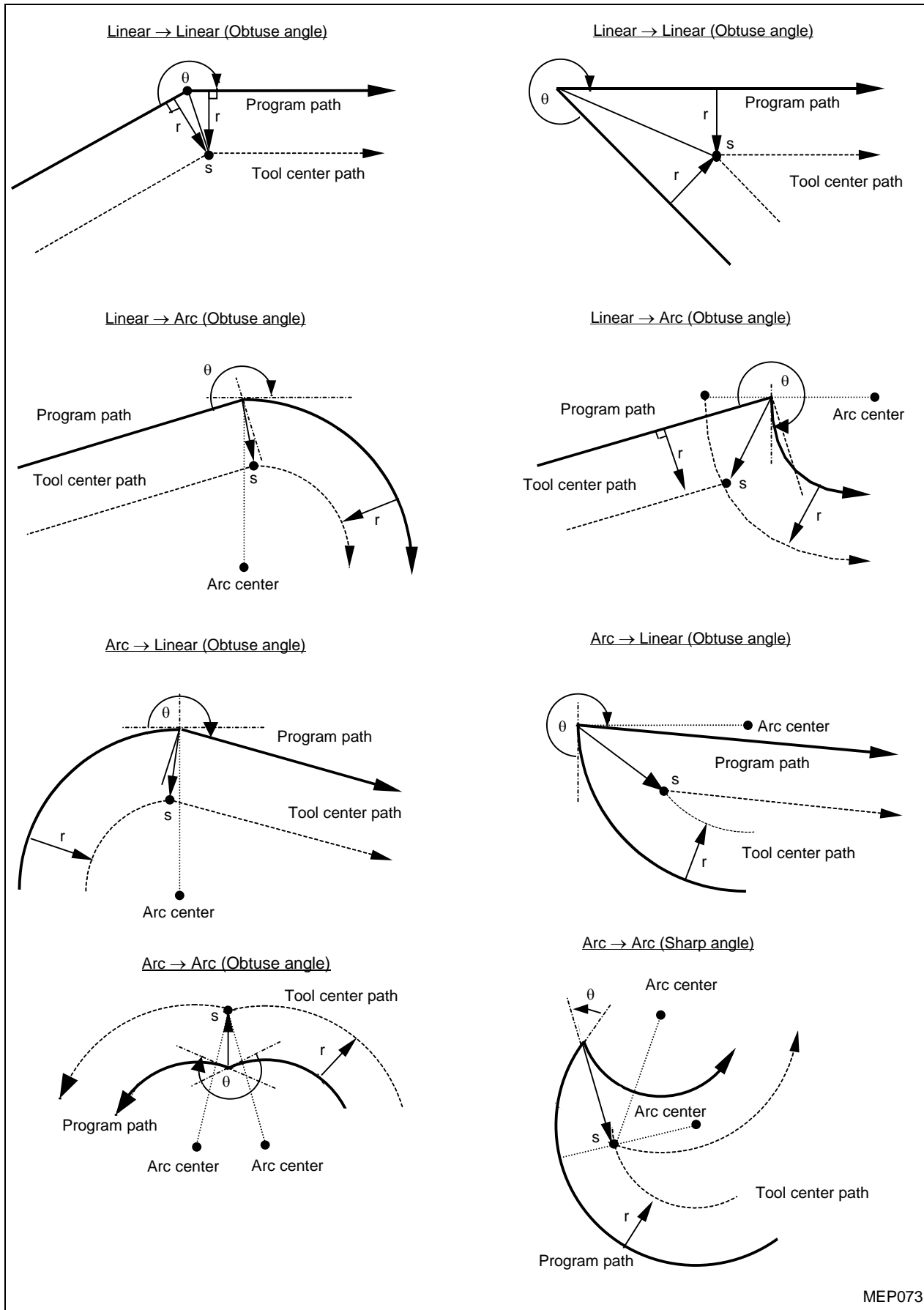
Offsetting is performed for linear or arc interpolation commands and positioning commands. Identical offset commands G41 or G42 will be ignored if they are used during the offset mode.

Successive setting of four or more blocks that do not involve movement of axes during the offset mode will result in excessive or insufficient cutting.

A. For the corner exterior



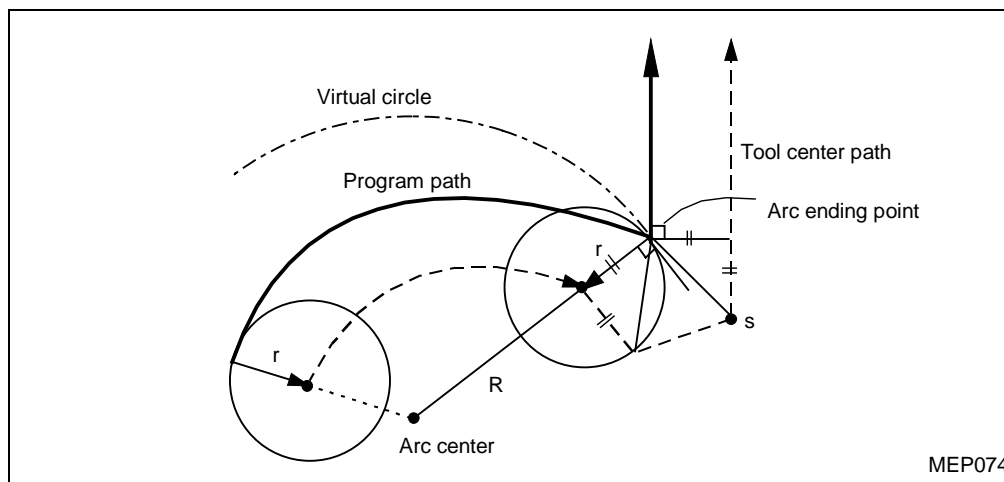
MEP072

B. For the corner interior

MEP073

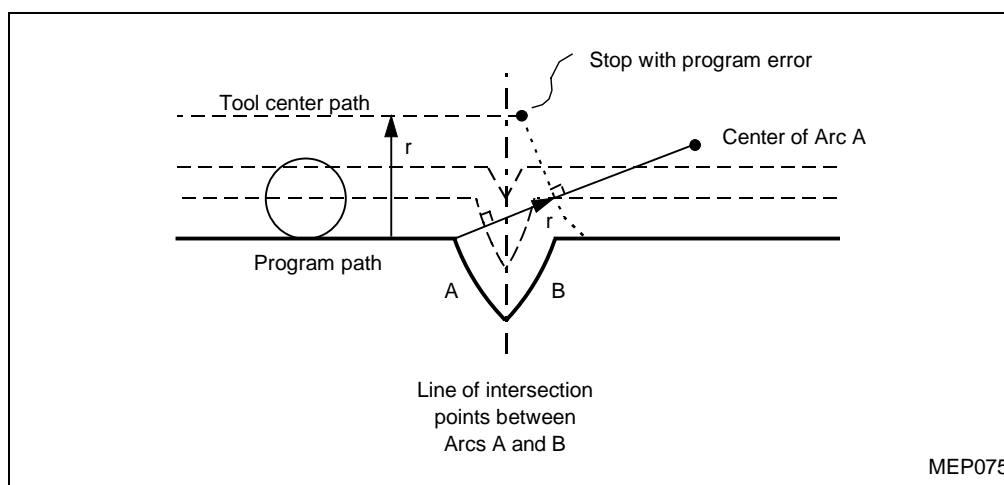
C. For an arc that does not have the ending point on it

The area from the starting point of the arc to the ending point is interpolated as a spiral arc.



D. For arcs that do not have their inner crossing point

In cases such as those shown in the diagram below, there may or may not be a crossing point of arcs A and B, depending on the particular offset data. In the latter case, the program terminates at the ending point of the preceding block after an alarm **836 NO INTERSECTION** has been displayed.



5. Tool diameter offsetting cancellation

During the tool diameter offset mode, tool diameter offsetting will be cancelled in any of the two cases listed below.

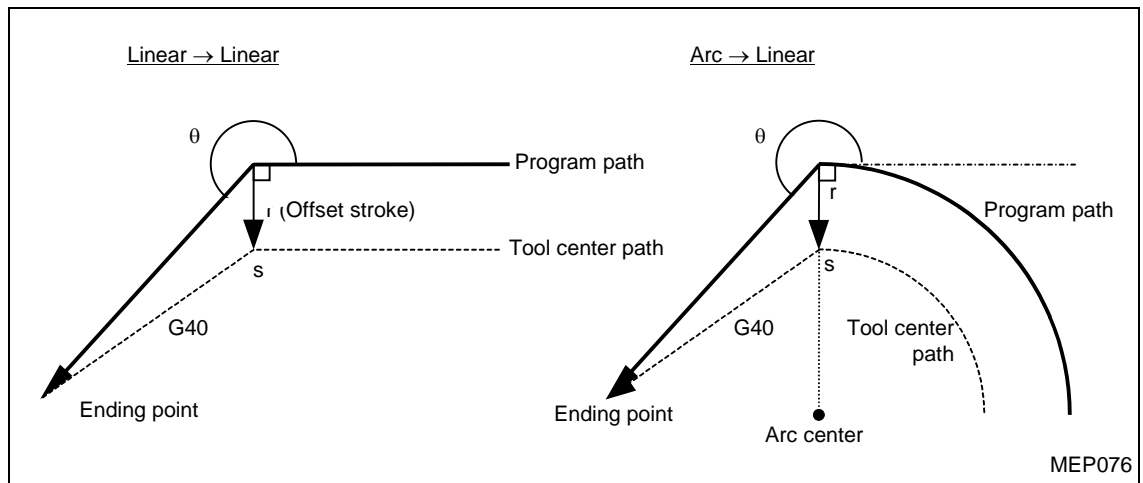
- Command G40 has been executed.
- Offset number code D00 has been executed.

At this time, however, the move command executed must be one other than those used for arc interpolation. An alarm **835 G41, G42 FORMAT ERROR** will occur if an attempt is made to cancel offsetting using an arc command.

After the offsetting cancellation command has been read into the offset buffer, the cancellation mode is set automatically and subsequent blocks of data are read into the pre-read buffer, not the offset buffer.

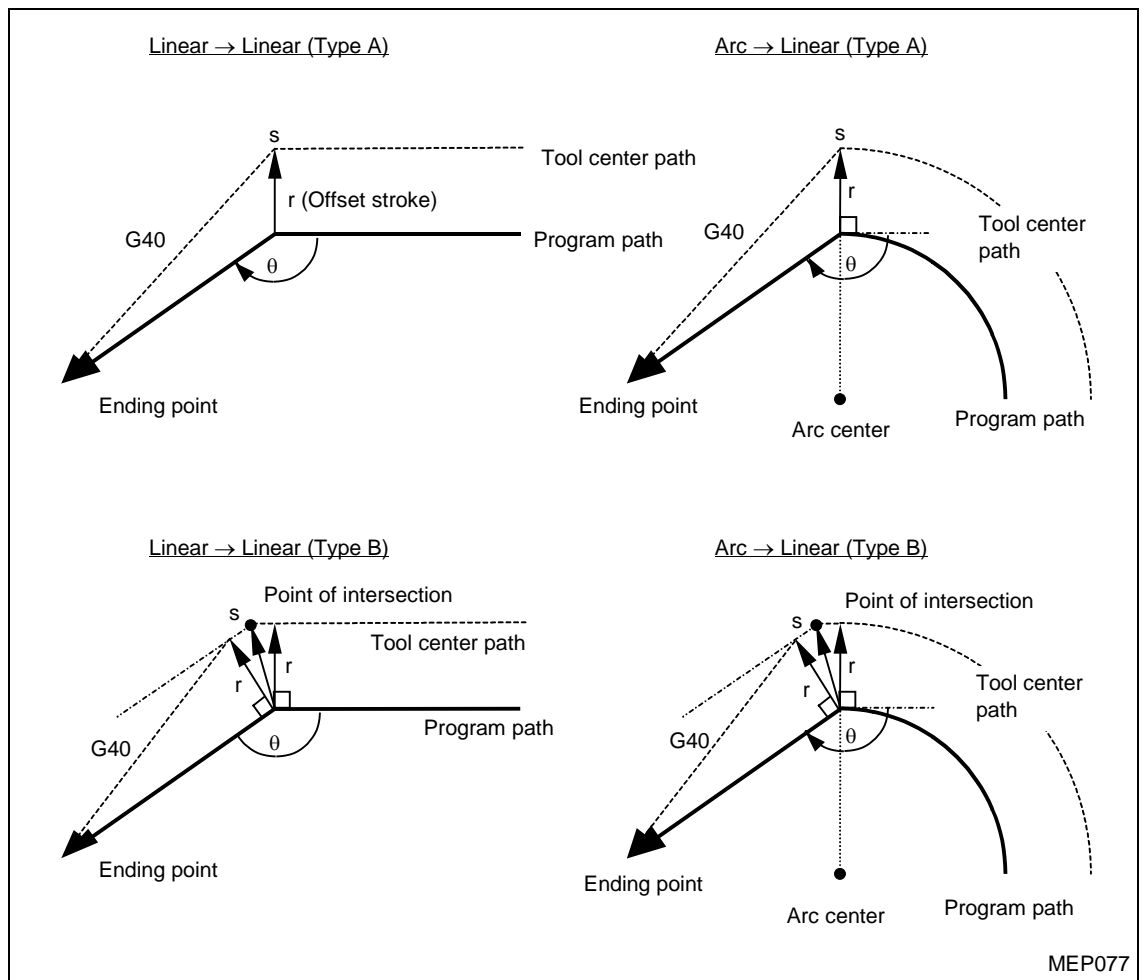
6. Tool diameter offsetting cancellation operation

A. For the corner interior



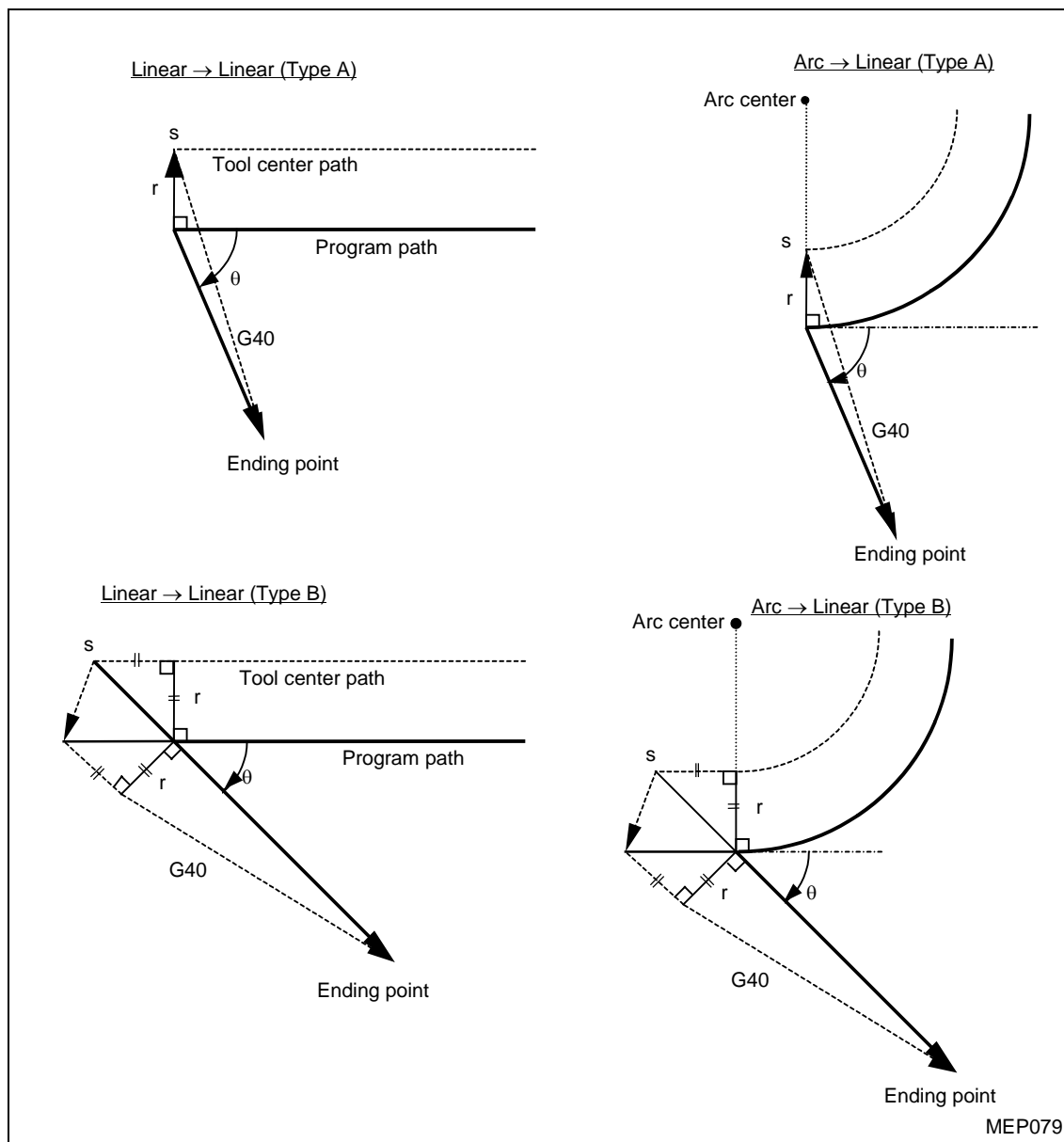
B. For the corner exterior (obtuse angle)

(Type A/B selection is possible with a predetermined parameter)



C. For the corner exterior (sharp angle)

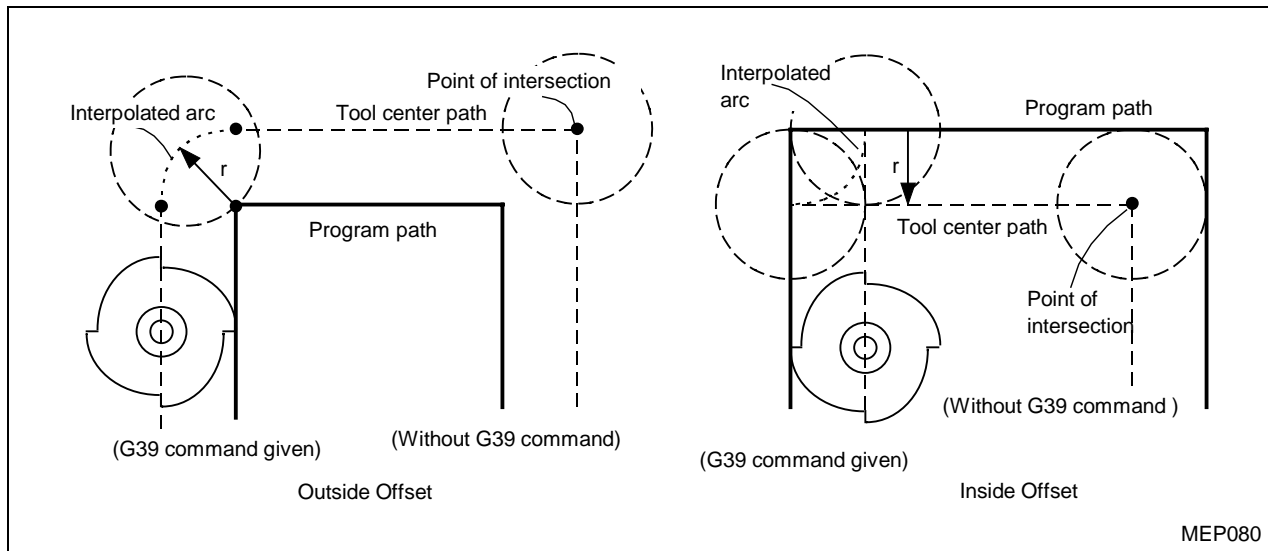
(Type A/B selection is possible with a predetermined parameter)



12-4-3 Tool diameter offsetting operation using other commands

1. Interpolation of the corner arc

When command G39 (corner-arc interpolation) is used, the coordinates of the crossing points at workpiece corners will not be calculated and an arc with offset data as its radius will be interpolated.



2. Changing/retaining offset vectors

Using command G38, you can change or retain offset vectors during tool diameter offsetting.

- Retaining vectors

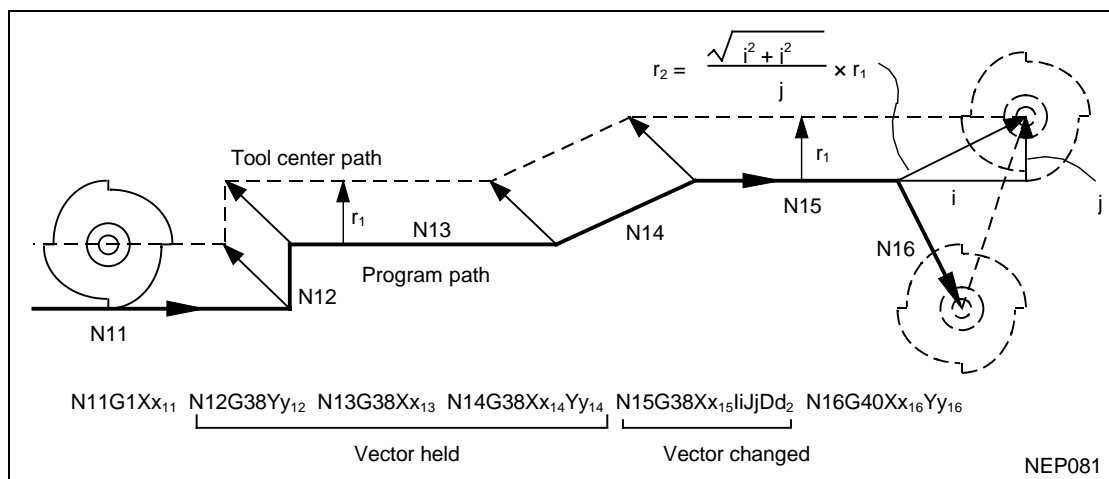
Setting G38 in block that contains move commands allows crossing-point calculation at the ending point of that block to be cancelled and the vectors in the preceding block to be retained. This can be used for pick and feed operations.

G38 Xx Yy

- Changing vectors

The directions of new offset vectors can be designated using I, J, and K (I, J, and K depend on the selected type of plane), and offset data can be designated using D. (These commands can be included in the same block as that which contains move commands.)

G38 Ii Jj Dd

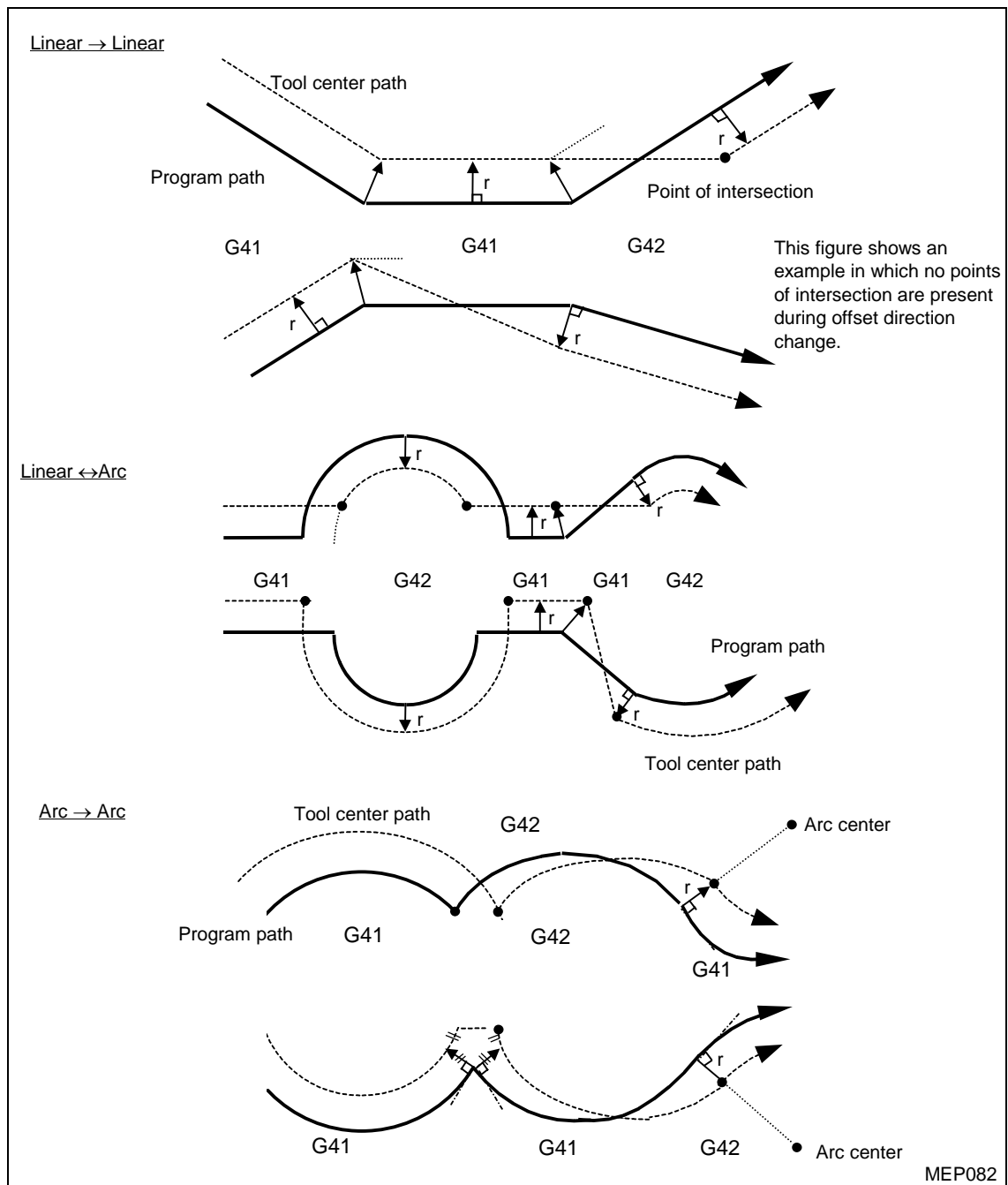


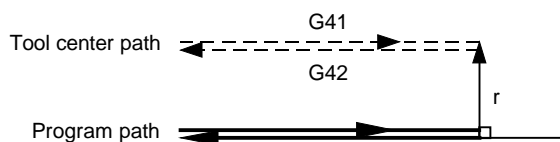
3. Changing the offset direction during tool diameter offsetting

The offset direction is determined by the type of tool diameter offset command (G41 or G42) and the sign (plus or minus) of the offset data.

Offset stroke sign G-code	+	-
G41	Left side offset	Right side offset
G42	Right side offset	Left side offset

The offset direction can be changed by updating the offset command without selecting the offsetting cancellation function during the offset mode. This can, however, be done only for blocks other than the offset startup block and the next block. See subsection 12-4-6, General precautions on tool diameter offsetting, for NC operation that will occur if the sign is changed.

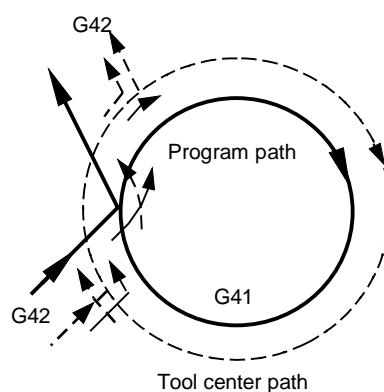


Linear turnaround

MEP083

The arc of more than 360 degrees may result in the following cases:

- The offset direction has been changed by G41/G42 selection.
- Commands I, J, and K have been set for G40.

Arc of 360° or more (depends on the offsetting method used)

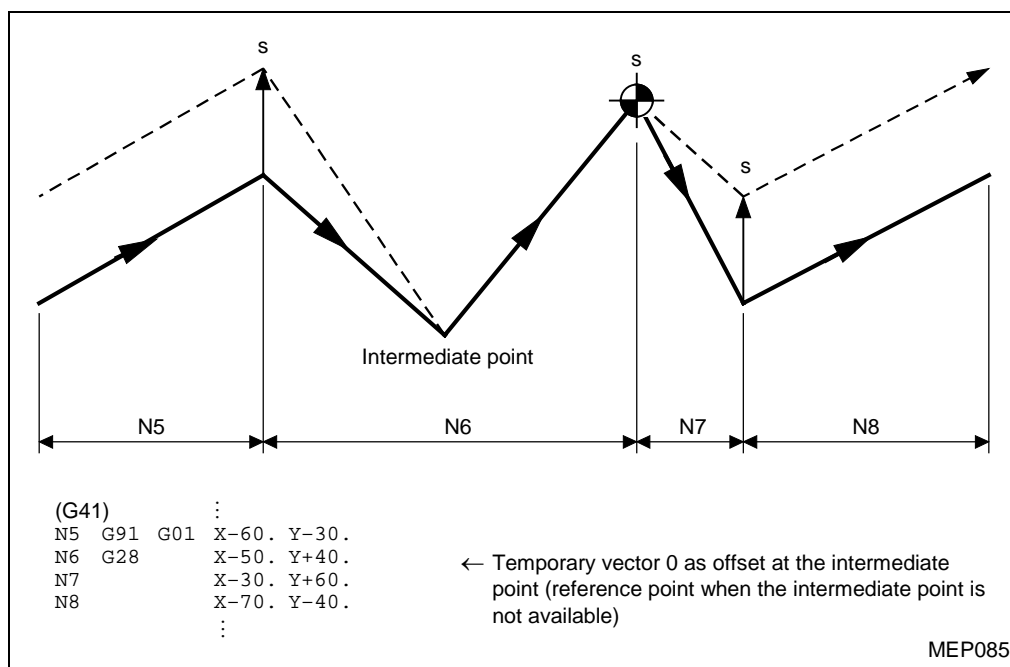
MEP084

4. Cases where the offset vectors are temporarily lost

If the command listed below is used during the offset mode, the current offset vectors will be lost temporarily and then the NC unit will re-enter the offset mode.

In that case, movements for offsetting cancellation will not occur and program control will be transferred from one crossing-point vector directly to the vector-less point, that is, to the programmed point. Control will also be transferred directly to the next crossing point when the offset mode is re-entered.

A. Reference-point return command



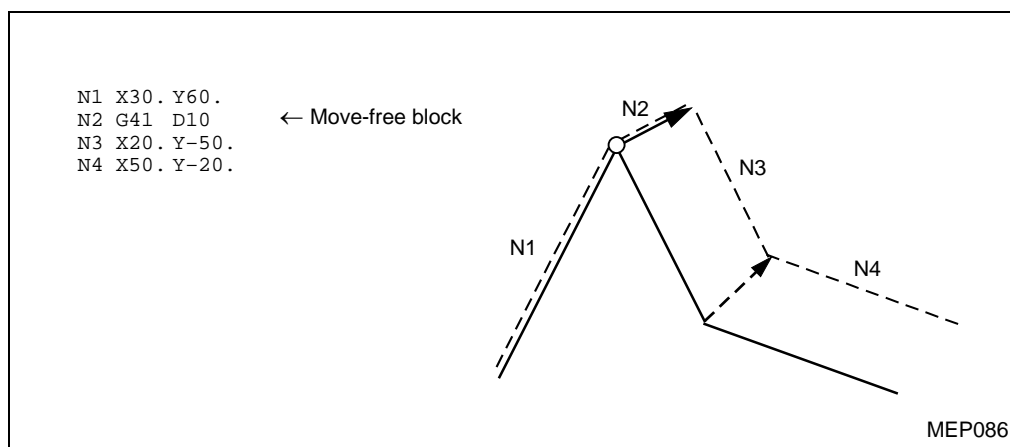
5. Blocks that do not include movement

The blocks listed below are referred to as those which do not include movement:

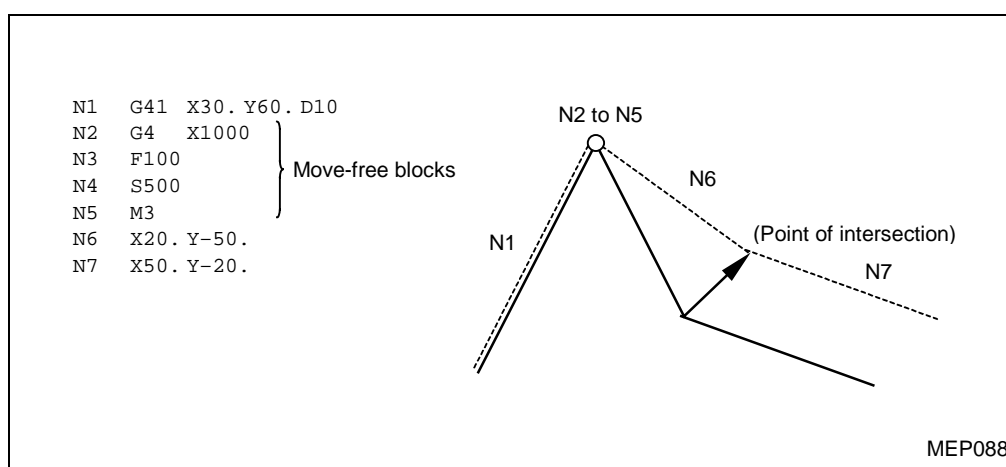
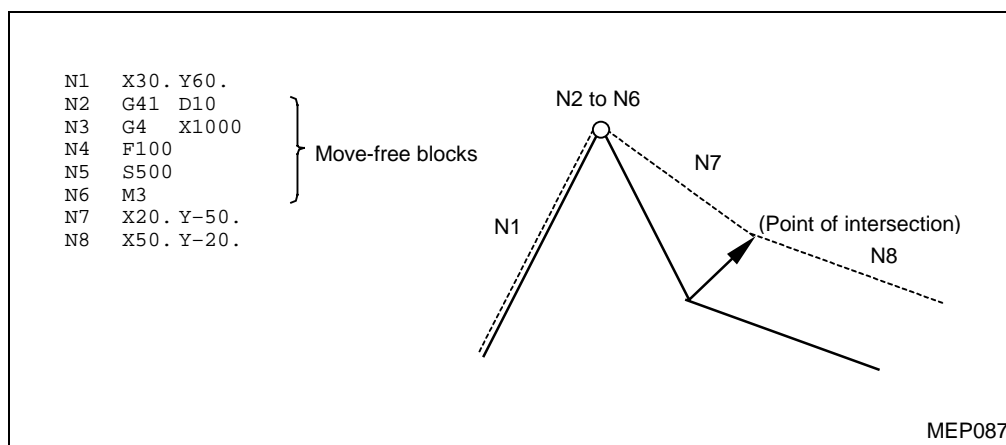
M03.....	M command	} Move-free
S12.....	S command	
T45.....	T command	
G04 X500.....	Dwell	
G22 X200. Y150. Z100.....	To set a machining-prohibited area	
G10 P01 R50.....	To set an offset stroke	
G92 X600. Y400. Z500.....	To set a coordinate system	
(G17) Z40.	To move outside the offsetting plane	
G90.....	G code only	} Moving stroke is 0.
G91 X0.....	Moving stroke 0	

A. When a block that does not include movement is set during the start of offsetting

Vertical offsetting will be performed on the next move block.

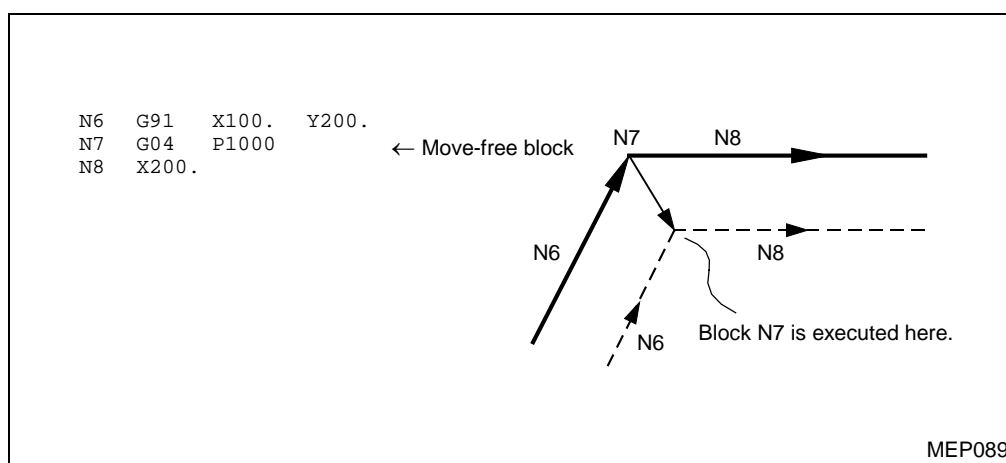


Offset vectors, however, will not be generated if four or more blocks that do not include move commands appear in succession.

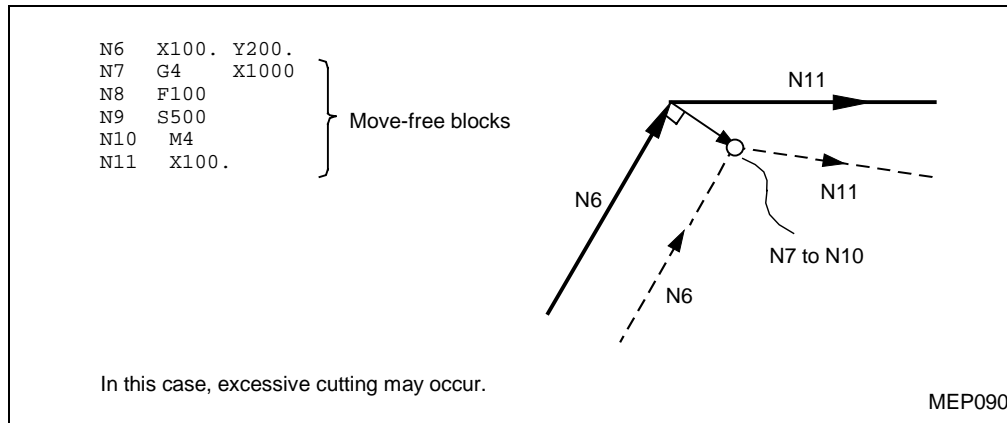


B. When a block that does not include movement is set during the offset mode

Usual crossing-point vectors will be generated unless four or more blocks that do not include movement appear in succession.

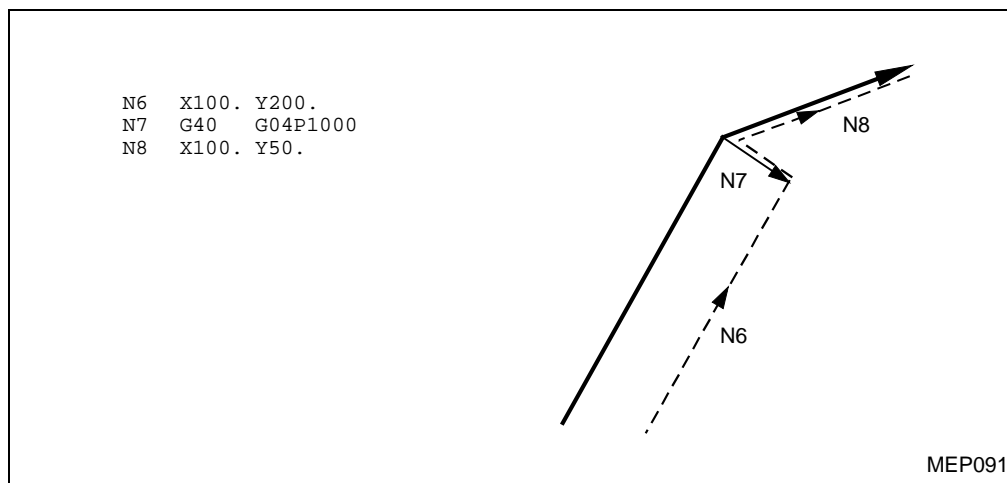


Vertical offset vectors will be generated at the end point of preceding block if four or more blocks that do not include movement appear in succession.



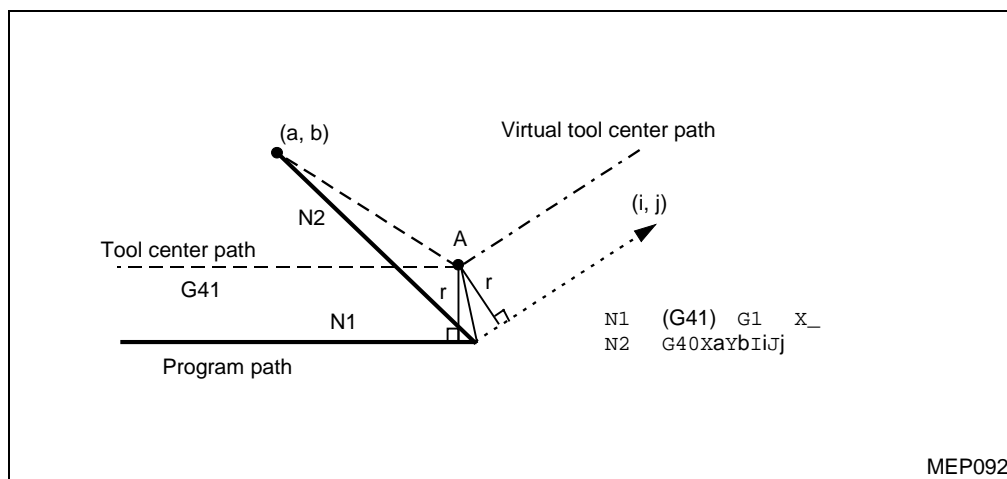
C. When a block that does not include movement is set together with offsetting cancellation

Only offset vectors will be cancelled if the block that does not include movement contains G40.

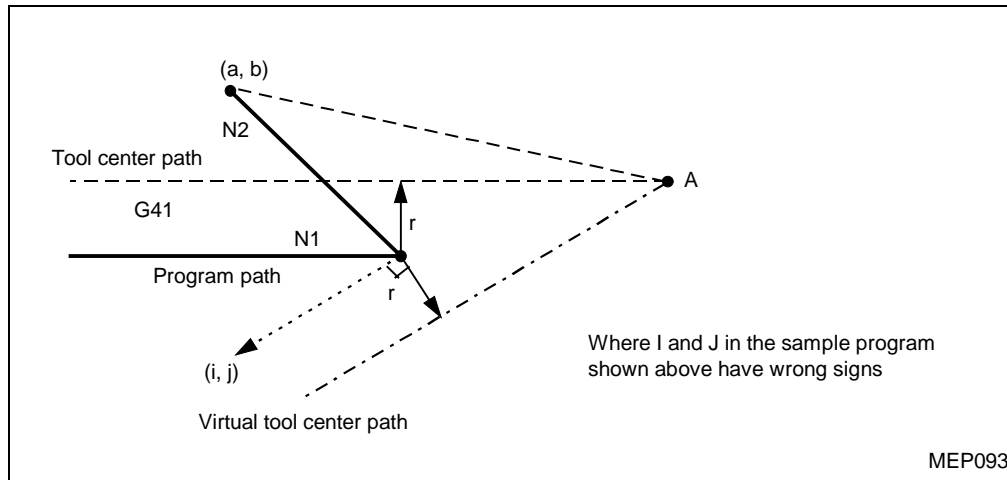


6. If I, J, and K are set with G40

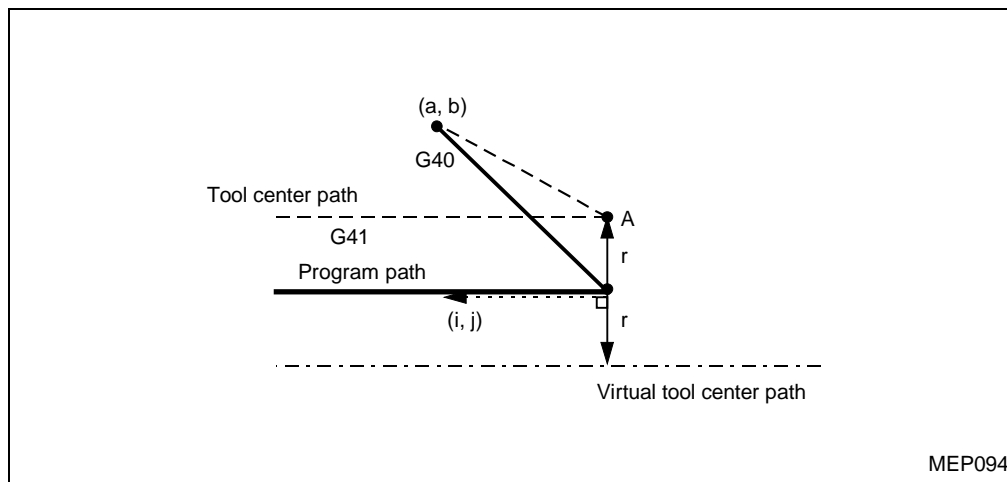
When the last of the four move command blocks which immediately precede the G40 command block contains G41 or G42, movement will be handled as if it had been programmed to occur in the vectorial direction of I, J, and K from the ending point of that last move command. That is, the area up to the crossing point with the virtual tool center path will be interpolated and then offsetting will be cancelled. The offset direction will remain unchanged.



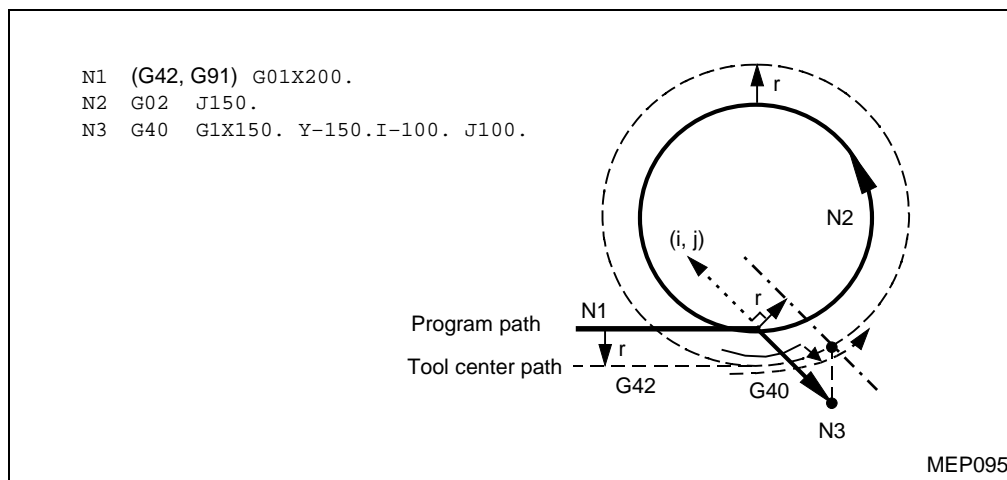
In this case, beware that irrespective of the offset direction, the coordinates of the crossing point will be calculated even if wrong vectors are set as shown in the diagram below.



Also, beware that a vertical vector will be generated on the block before that of G40 if crossing-point calculation results in the offset vector becoming too large.



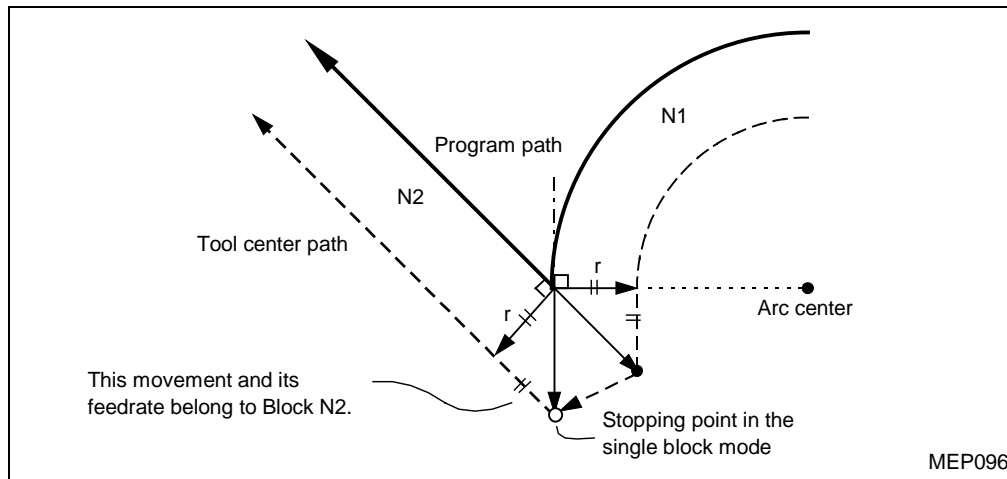
Note: Part of the workpiece will be cut twice if the I/J/K command data in G40 preceded by an arc command generates an arc of more than 360 degrees.



12-4-4 Corner movement

If multiple offset vectors are generated at connections between move command blocks, the tool will move linearly between those vectors. This action is referred to as corner movement.

If the multiple vectors do not agree, the tool will move around the corresponding corners (but this movement belongs to the next block). During single-block operation, the section of (Preceding block + Corner movement) is executed as one block and the remaining section of (Connections movement + Next block) is executed during next movement as another block.



12-4-5 Interruptions during tool diameter offsetting

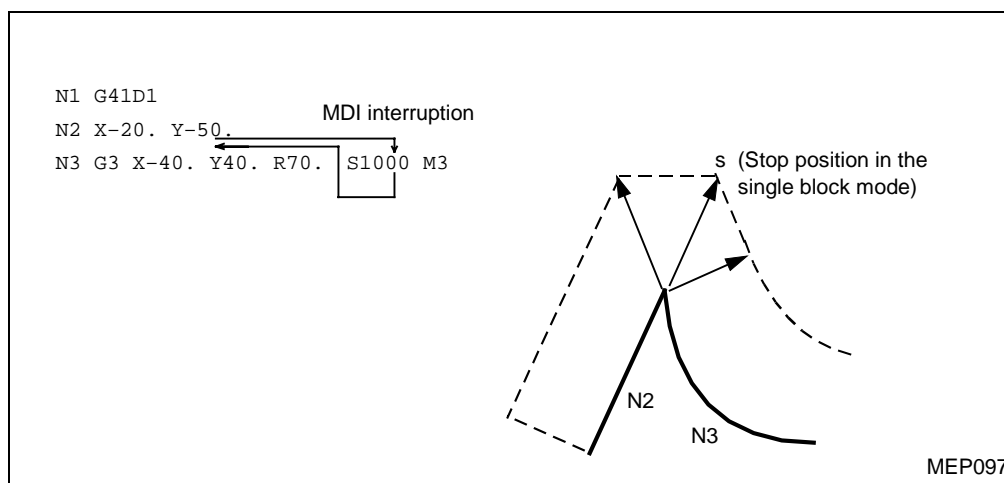
1. Interruption by MDI

Tool diameter offsetting is valid during automatic operation, whether it is based on the tape, memory, or MDI operation mode.

The following diagrams show what will occur if tape or memory operation is interrupted using the MDI function following termination of the program at a block:

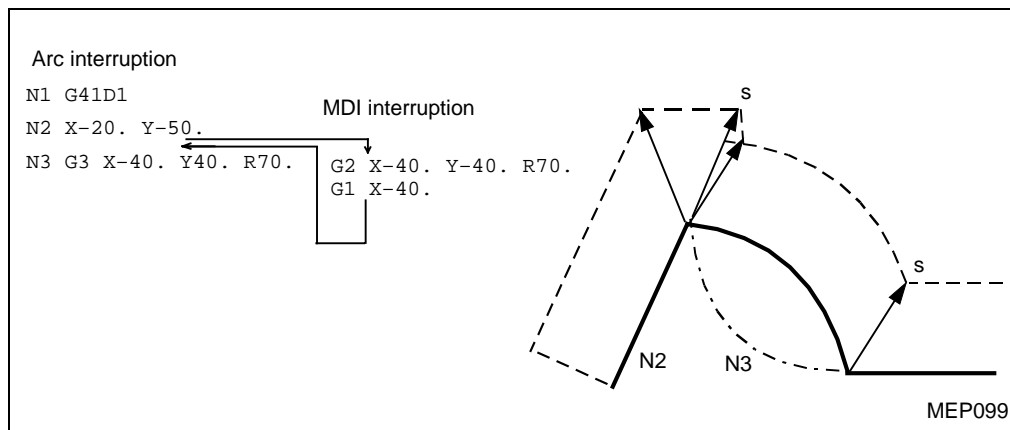
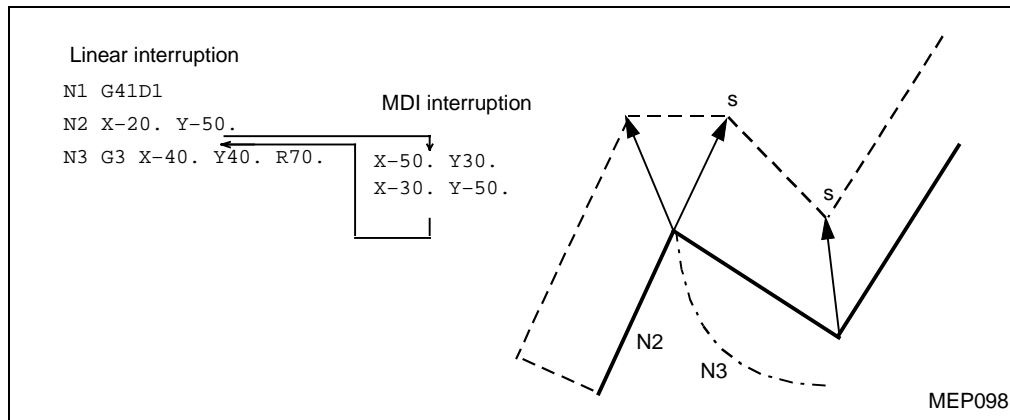
A. Interruption without movement

No change in tool path



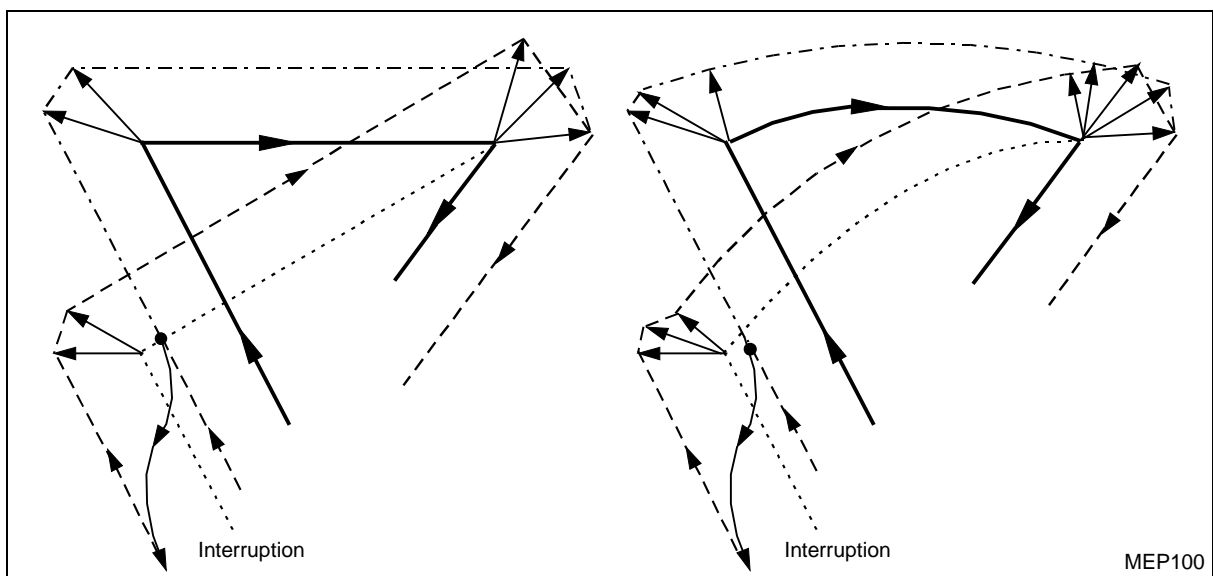
B. Interruption with movement

The offset vectors are recalculated automatically at the first effective move block after interruption.



2. Manual interruption

- For the incremental data command mode, the tool path shifts through the interruption amount.
- For the absolute data command mode, the intended tool path is restored at the ending point of the block immediately following that at which interruption has been performed. This state is shown in the diagram below.



12-4-6 General precautions on tool diameter offsetting

1. Selecting the amounts of offset

The amounts of offset are selected by specifying an offset number using a D code. Once a D code has been used, it will remain valid until a second D code is used. No H codes can be used to make these selections.

D codes are also used to select tool position offset data.

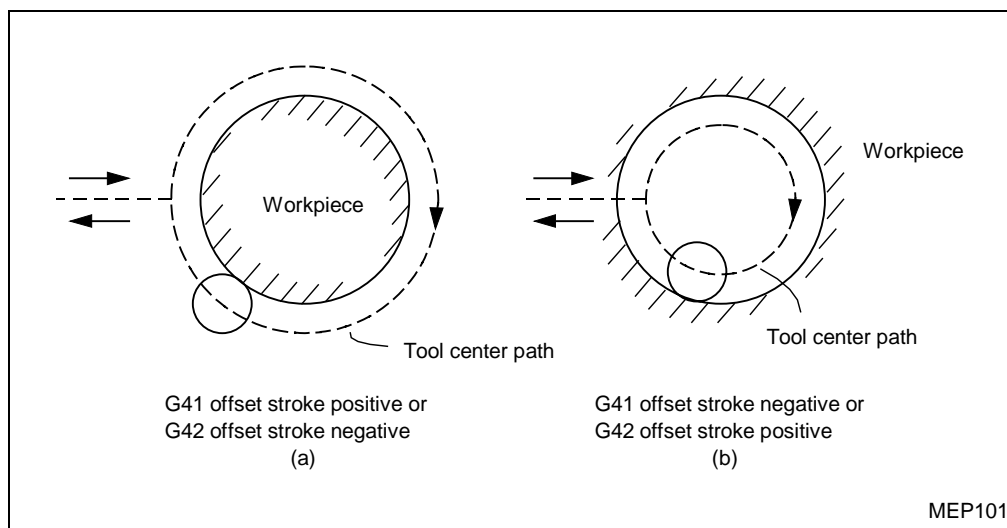
2. Updating the selected amounts of offset

Updating of the selected amounts of offset is usually to be done after a different tool has been selected during the diameter offsetting cancellation mode. If such updating is done during the offset mode, vectors at the ending point of a block will be calculated using the offset data selected for that block.

3. The sign of offset data and the tool center path

Minus-signed (–) offset data generates the same figure as that obtained when G41 and G42 are exchanged each other. Therefore, the tool center will move around the inside of the workpiece if it has been moving around the outside. Conversely, the tool center will move around the outside of the workpiece if it has been moving around the inside.

Sample programs are shown below. Usually, offset data is to be programmed as plus (+) data. If the tool center has been programmed to move as shown in diagram (a) below, the movement can be changed as shown in diagram (b) below by changing the sign of the offset data to minus (–). Conversely, if the tool center has been programmed to move as shown in diagram (b) below, the movement can be changed as shown in diagram (a) below by changing the sign of the offset data to plus (+). One tape for machining of both inside and outside shapes can be created in this way. Also, a dimensional tolerance between both shapes can be freely set by selecting appropriate offset data (however, Type A is to be used during the start of offsetting or during its cancellation).



12-4-7 Offset number updating during the offset mode

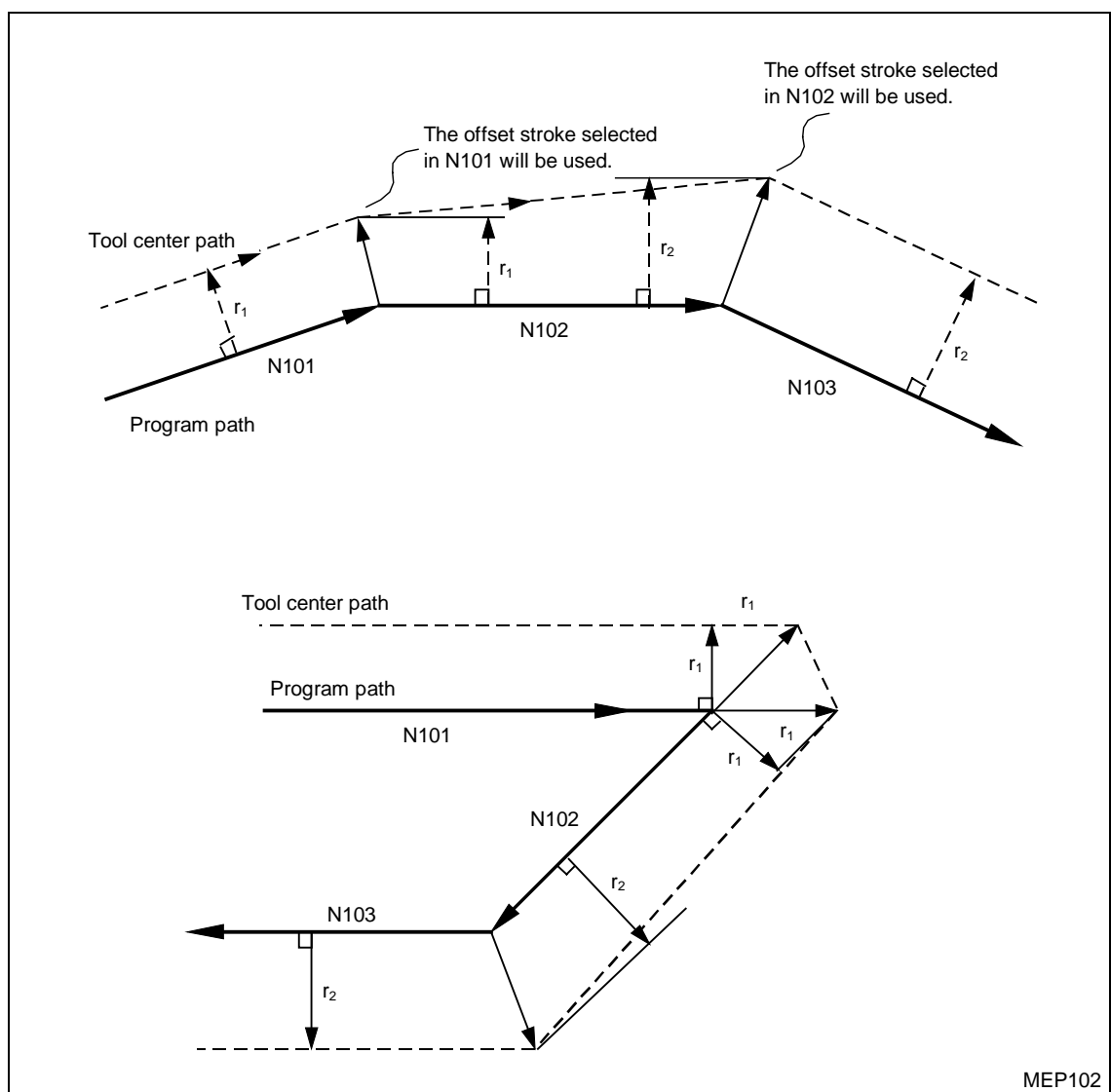
In principle, offset numbers should not be updated during the offset mode. If updating is done, the tool center will move as shown below.

If an offset number (offset data) is updated

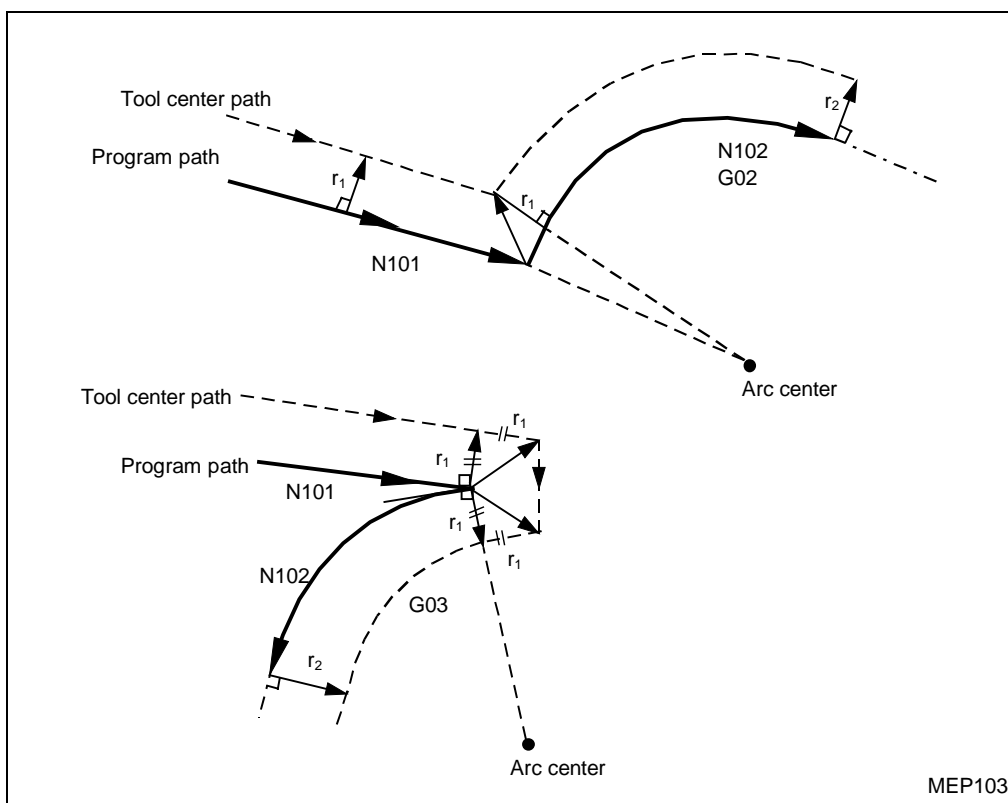
```

G41 G01      Dr1
:
:
:
α = 0, 1, 2, 3
N101 G0α Xx1 Yy1
N102 G0α Xx2 Yy2 Dr2 To change an offset number
N103      Xx3 Yy3
  
```

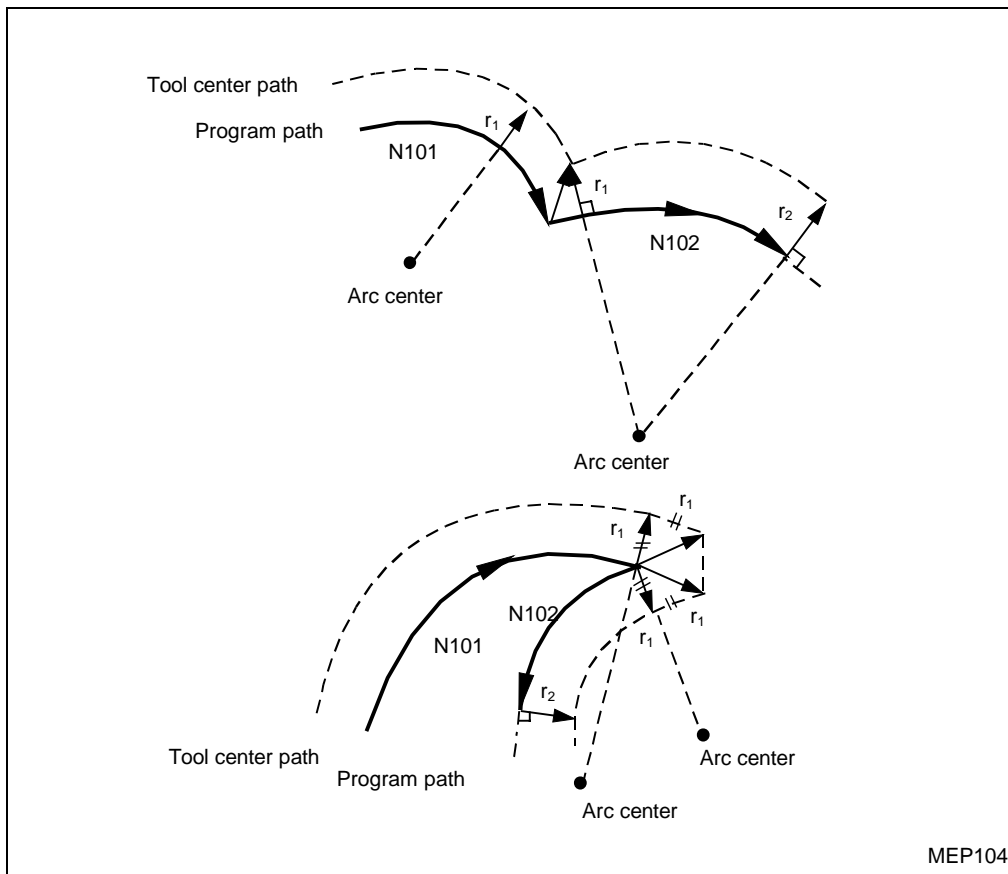
1. Line-to-line movement



2. Line-to-arc movement



3. Arc-to-arc movement

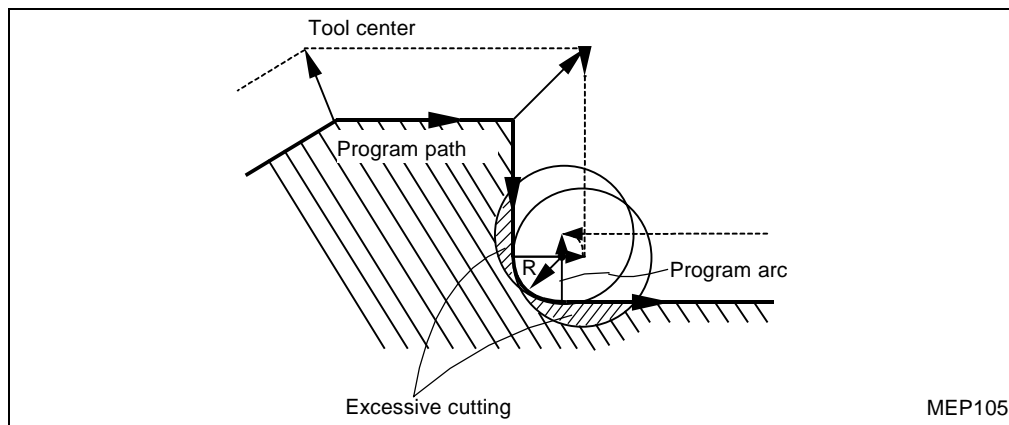


12-4-8 Excessive cutting due to tool diameter offsetting

If an interference check function is not provided, excessive cutting may result in the following three cases:

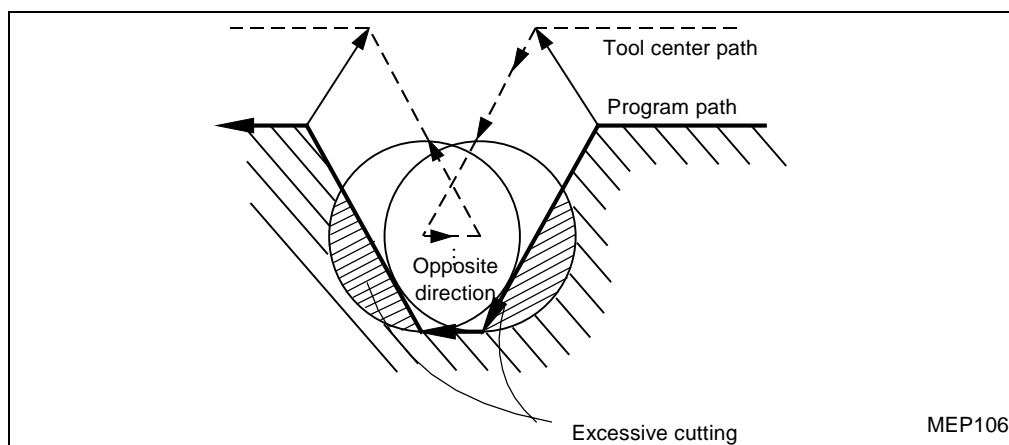
1. Machining of the inside of an arc smaller than the tool radius

If the radius of the programmed arc is smaller than that of the tool, excessive cutting may result from offsetting of the inside of the arc.

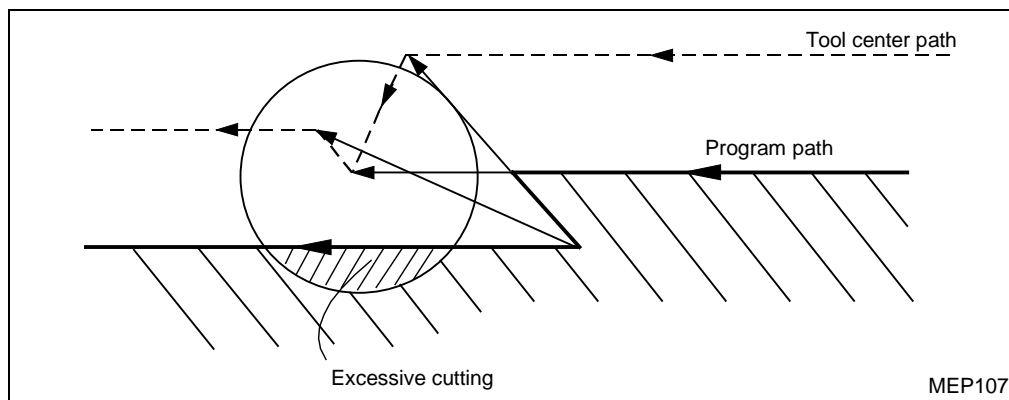


2. Machining of a groove smaller than the tool radius

Excessive cutting may result if tool diameter offsetting makes the moving direction of the tool center opposite to that of the program.



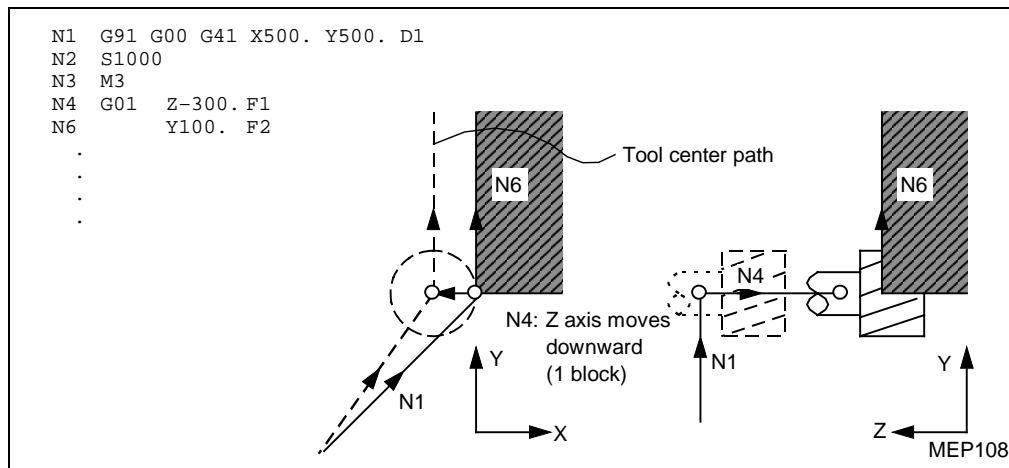
3. Machining of a stepped section smaller than the tool radius



4. Relationship between the start of tool diameter offsetting and the cutting operation in the Z-axis direction

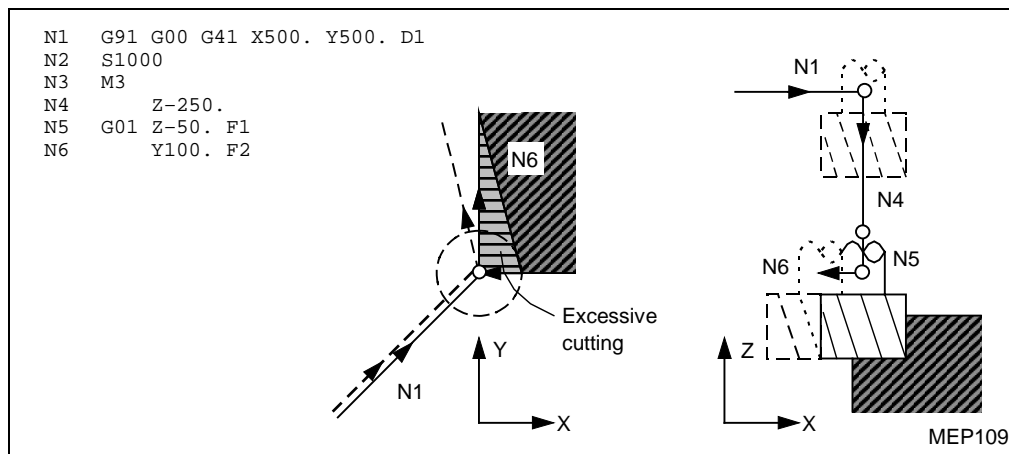
It is generally done that diameter offsetting (usually, on the X-Y plane) is done at a suitable distance from the workpiece during the start of cutting and then the workpiece is cut along the Z-axis. At this time, incorporate the following programming considerations if you want to split the Z-axis action into rapid feed and cutting feed which is to follow only after the Z-axis has moved close to the workpiece:

If you make a program such as that shown below:



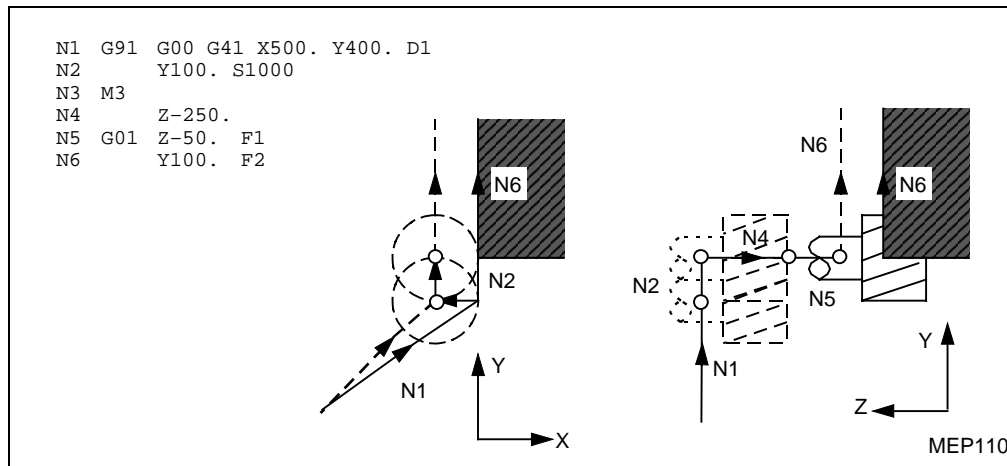
With this program, all blocks up to N6 can be read during the start of offsetting based on N1. Thus, the NC unit will judge the relationship between N1 and N6 and correctly perform the offset operation as shown in the diagram above.

A sample program in which the N4 block in the program shown above has been split into two parts is shown below.



In this case, the N2 through N5 blocks do not have any command corresponding to the X-Y plane and the relevant block N6 cannot be read during the start of offsetting based on N1. As a result, offsetting will be based only on the information contained in the N1 block and thus the NC unit will not be able to create offset vectors during the start of offsetting. This will cause excessive cutting as shown in the diagram above.

Even in such a case, however, excessive cutting can be prevented if a command code that moves the tool in exactly the same direction as that existing after the Z-axis has moved downward is included immediately before the Z-direction cutting block.



For the sample program shown above, correct offsetting is ensured since the moving direction of the tool center at N2 is the same as at N6.

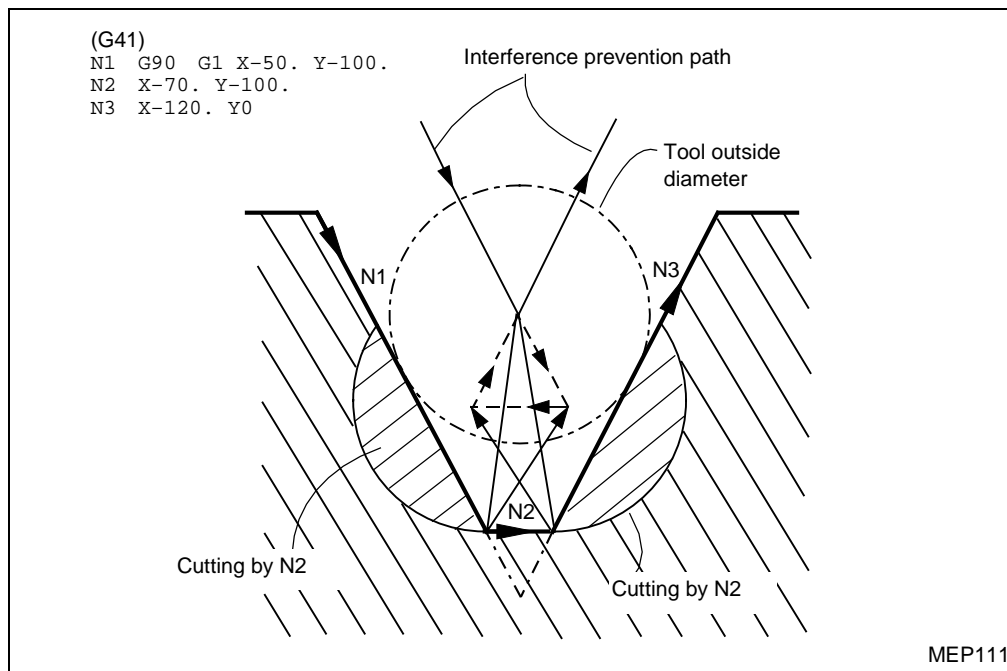
12-4-9 Interference check

1. Overview

Even a tool whose diameter has been offset by usual tool-diameter offsetting based on two-block pre-reading may move into the workpiece to cut it. This status is referred to as interference, and a function for the prevention of such interference is referred to as interference check.

The following two types of interference check are provided and their selection is to be made using bit 5 of parameter **F92**.

Function	Parameter	Operation
Interference check and alarm	Interference check and prevention off	The system will stop, with a program error resulting before executing the cutting block.
Interference check and prevention	Interference check and prevention on	The path is changed to prevent cutting from taking place.

Example:

- For the alarm function

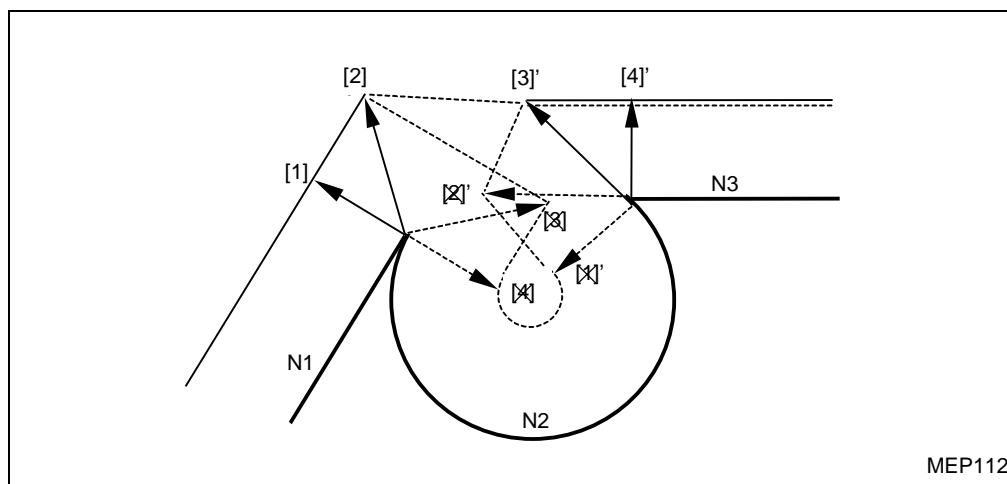
An alarm occurs before N1 is executed. Machining can therefore be proceeded with by updating the program into, for example,

```
N1 G90 G1 X-20. Y-40.
```

using the buffer correction function.

- For the prevention function

Interference prevention vectors are generated by N1 and N3 crossing-point calculation.



Vector [1] [4]' check → No interference



Vector [2] [3]' check → No interference



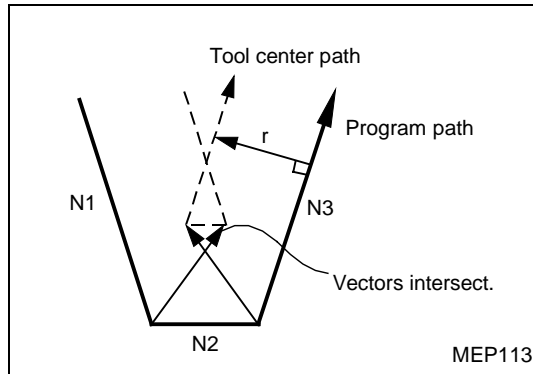
Vector [3] [2]' check → Interference → Vector [3] [2]' deletion → Vector [4] [1]' deletion

The above process is performed to leave vectors [1] [2] [3]' and [4]' as effective ones. Resultantly, the route that connects vectors [1] [2] [3]' and [4]' is taken as a bypass for the prevention of interference.

2. Detailed description

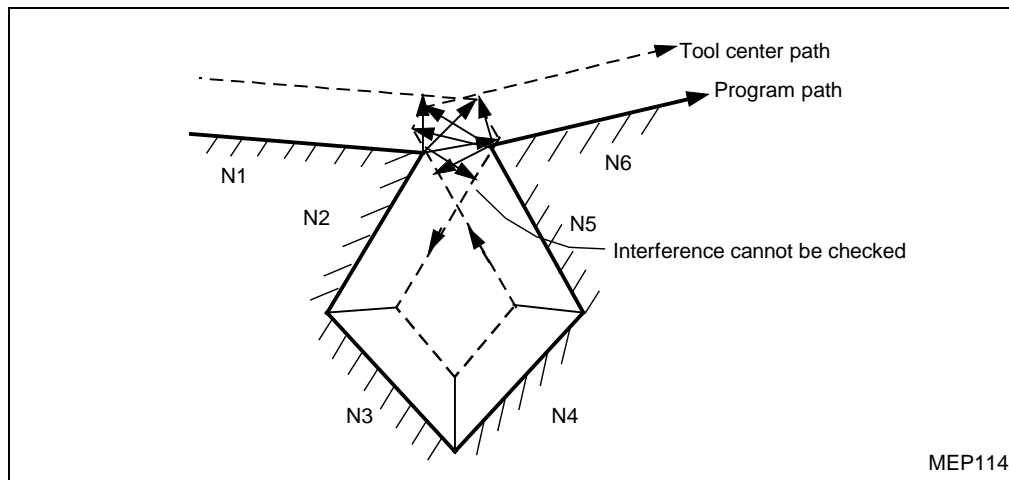
A. The case where interference is regarded as occurring

When move commands are present in three of the five command blocks to be pre-read, interference will be regarded as occurring, if the offset calculation vectors at the block connections of the individual move commands intersect.



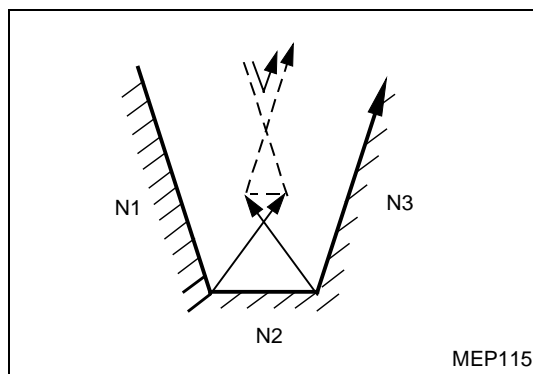
B. Cases where interference check cannot be performed

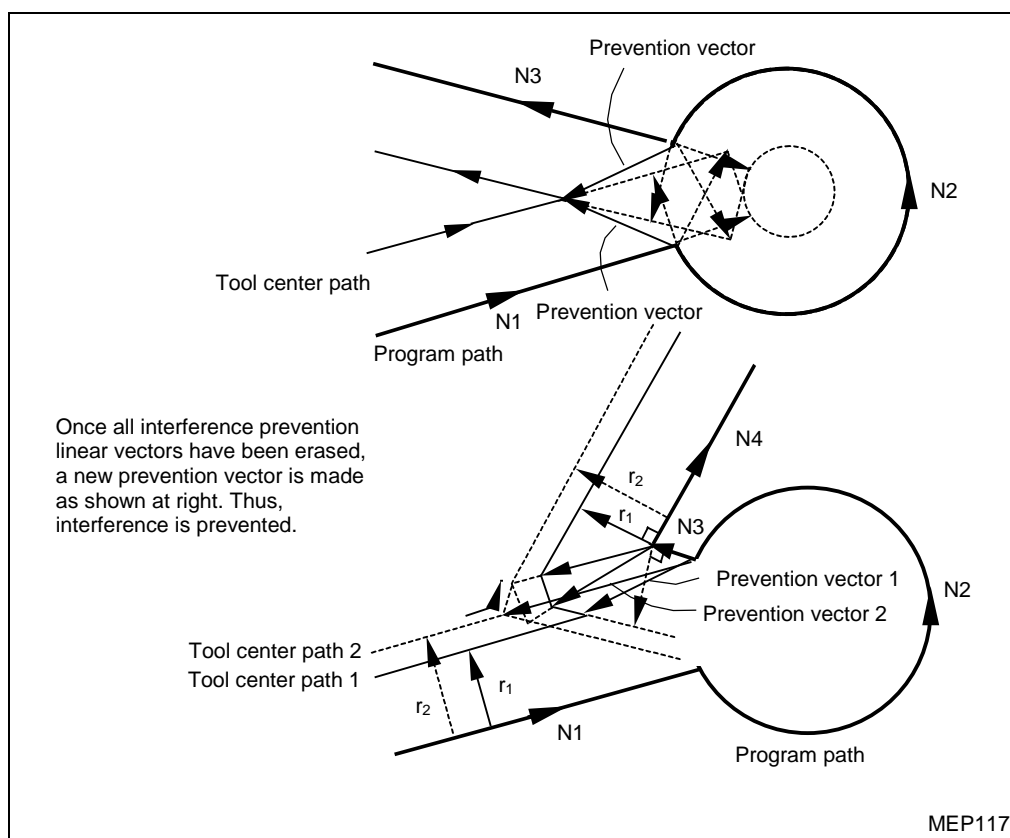
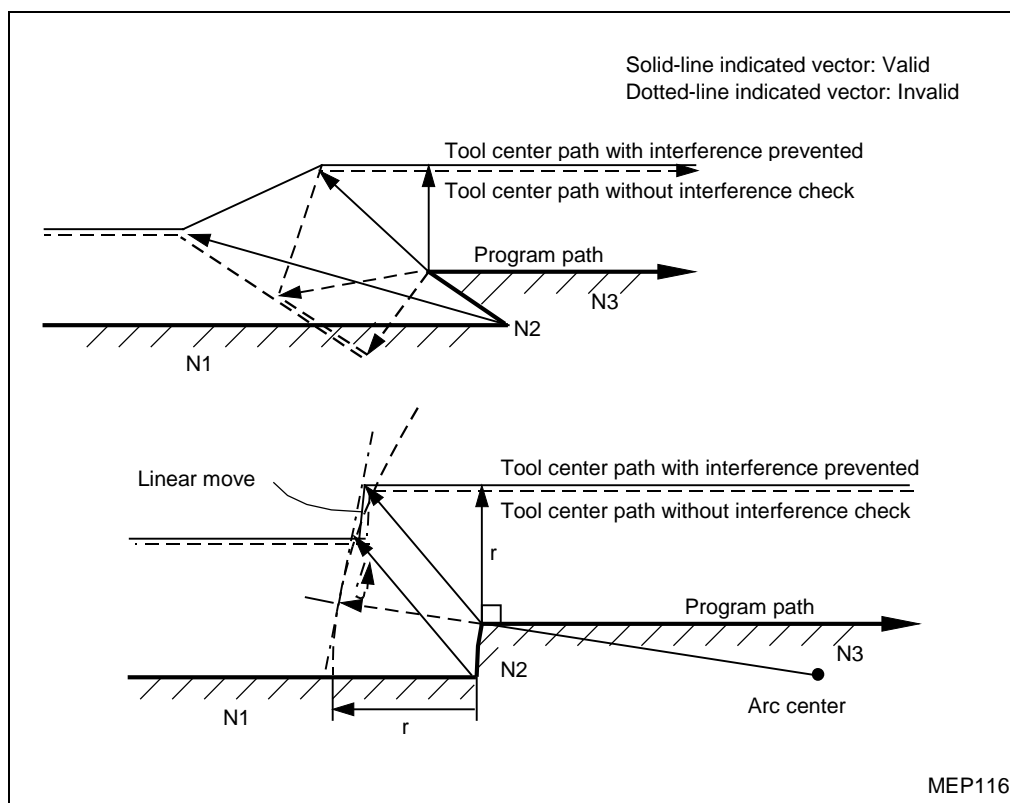
- When pre-reading of three move command blocks of the five to be pre-read is not possible (since the three blocks do not contain move commands).
- When the fourth and subsequent move command blocks themselves interfere.



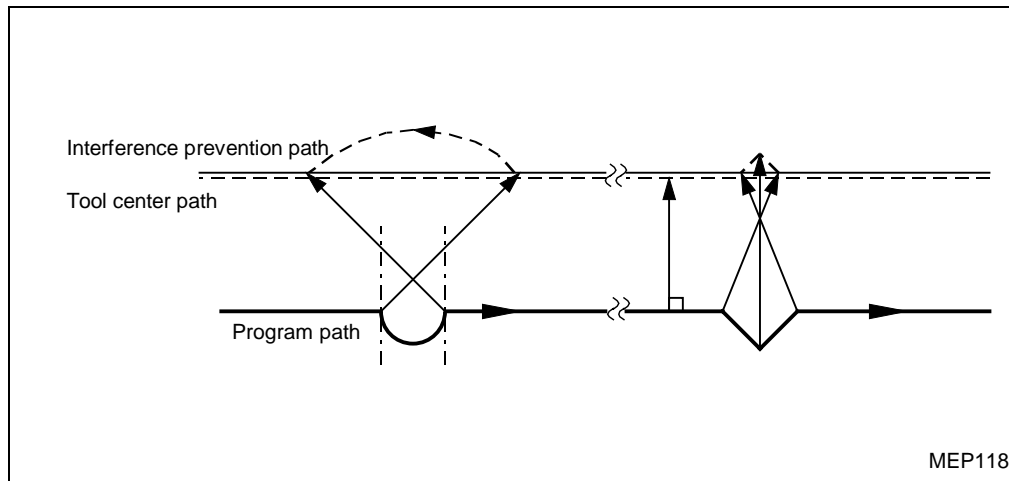
C. Movements during the prevention of interference

The following shows the movements occurring when interference prevention is provided:





In the diagram shown below, part of the groove is left uncut:

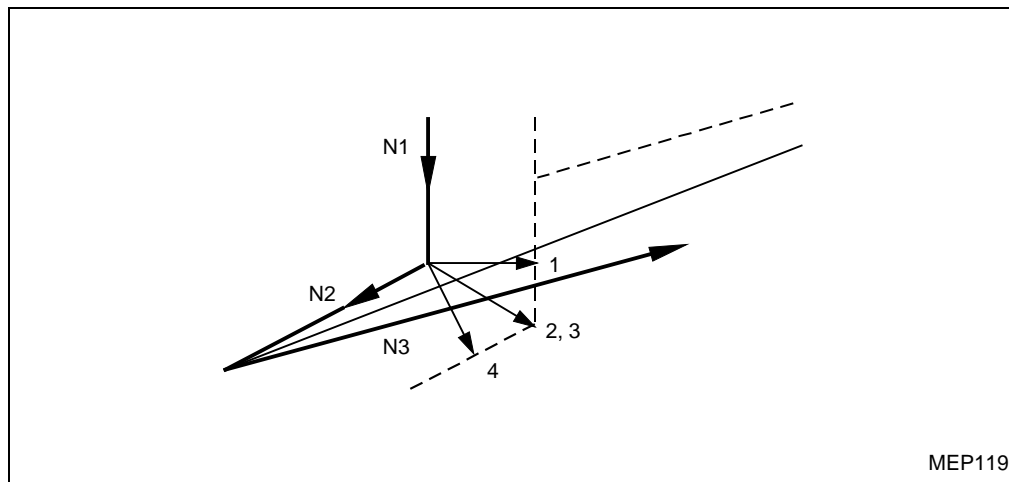


3. Interference alarm

Cases that an interference alarm **837 TOOL OFFSET INTERFERENCE ERROR** occurs are listed below.

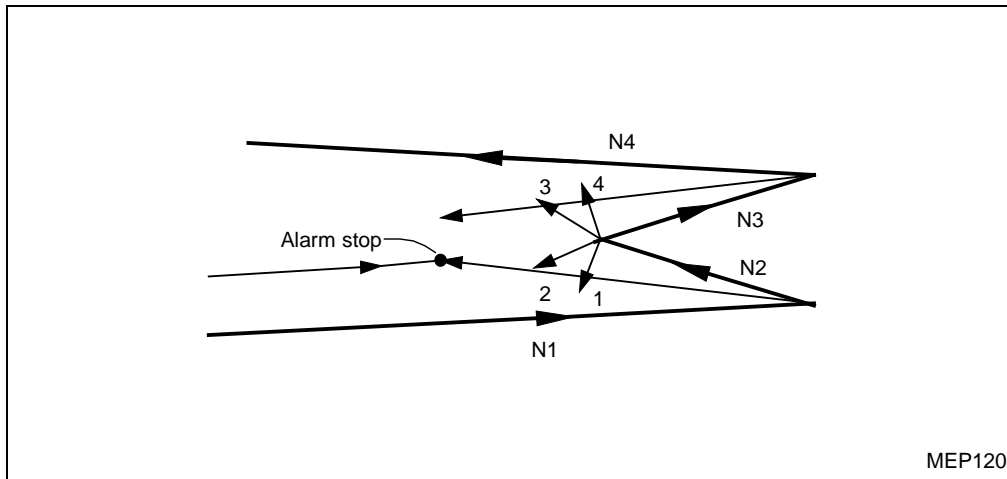
When interference check and alarm is selected

- 1) If all vectors at the ending point of the current block are erased:
Prior to execution of N1, a program error will result if vectors 1 through 4 at the ending point of the N1 block are all erased as shown in the diagram below.

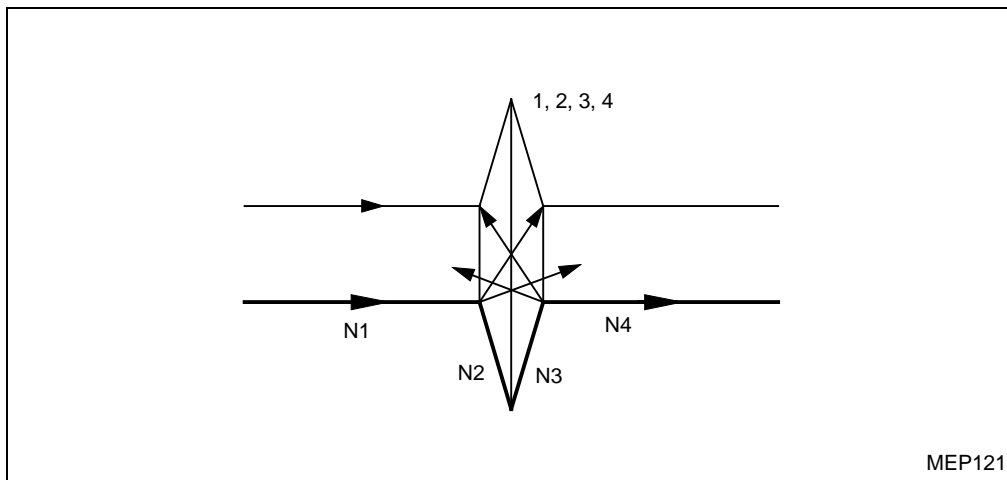


When interference check and prevention is selected

- 2) If all vectors at the ending point of the current block are erased but an effective vector(s) remains at the ending point of the next block:
- For the diagram shown below, interference checking at N2 will erase all vectors existing at the ending point of N2, but leave the vectors at the ending point of N3 effective. At this time, a program error will occur at the ending point of N1.



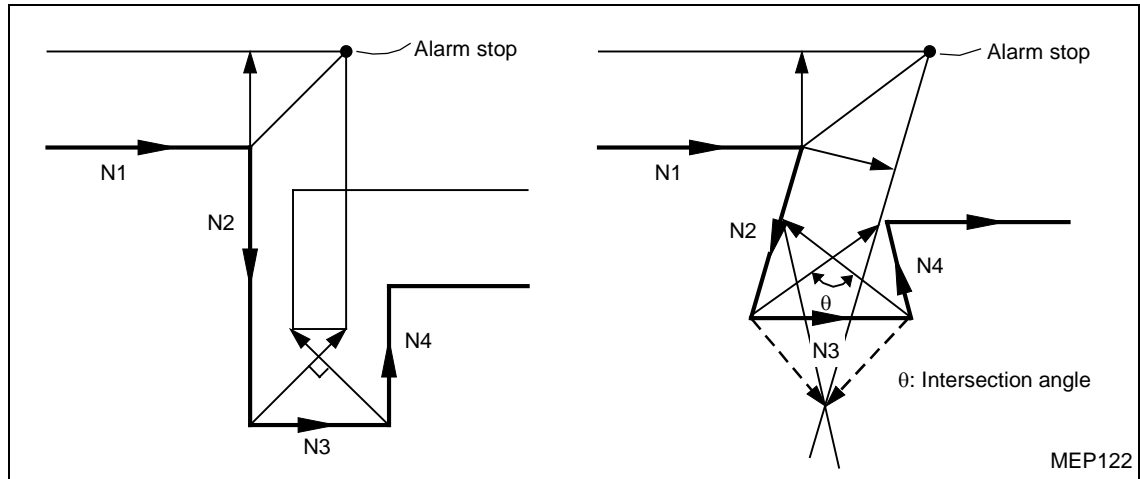
- For the diagram shown below, the direction of movement becomes opposite at N2. At this time, a program error will occur before execution of N1.



3) When prevention vectors cannot be generated:

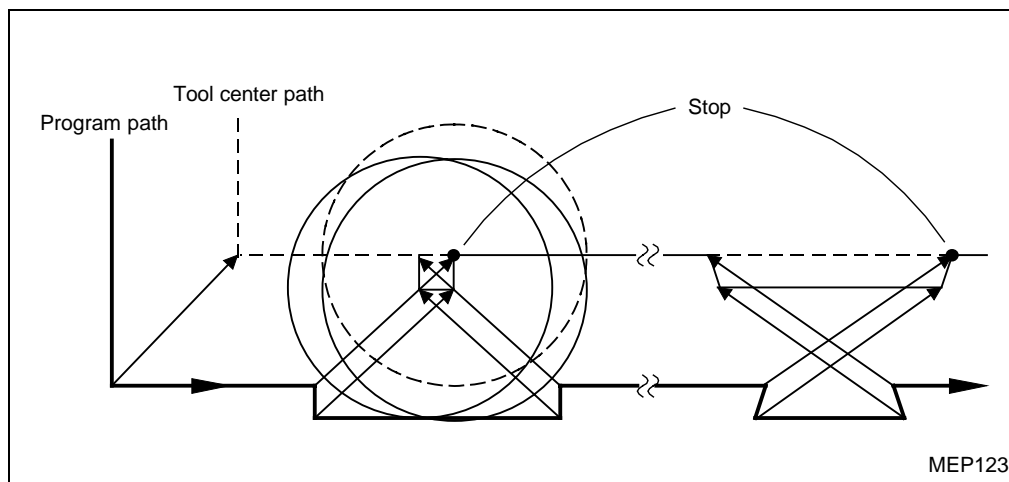
Prevention vectors may not be generated even when the conditions for generating them are satisfied. Or even after generation, the prevention vectors may interfere with N3.

A program error will therefore occur at the ending point of N1 if those vectors cross at angles of 90 degrees or more.



4) When the after-offsetting moving direction of the tool is opposite to that of the program:

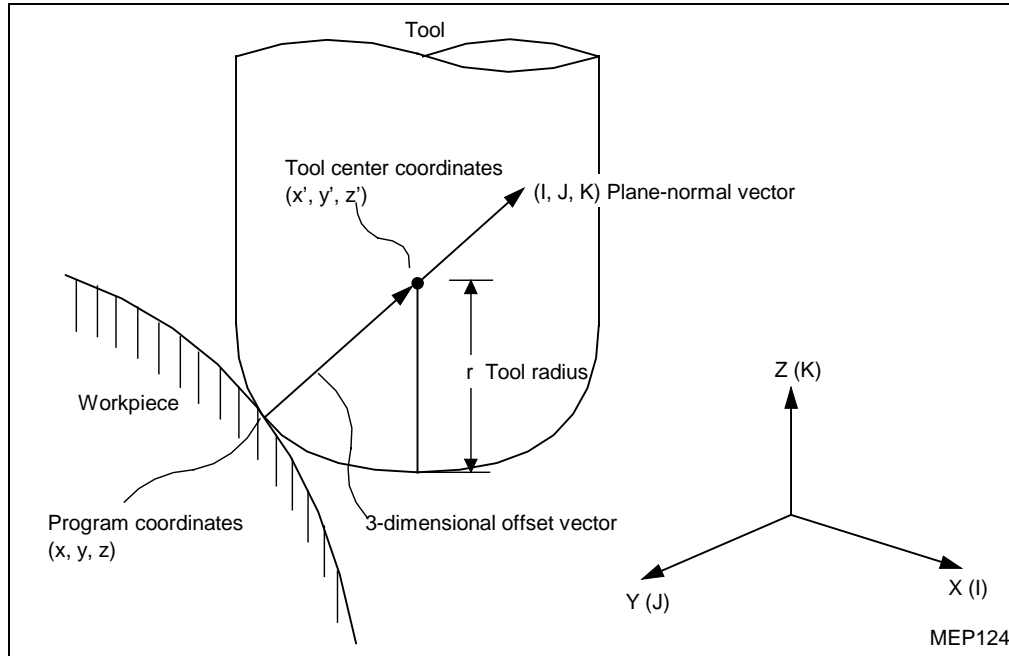
For a program for the machining of parallel or downwardly extending grooves narrower than the tool diameter, interference may be regarded as occurring even if it is not actually occurring.



12-5 Three-Dimensional Tool Diameter Offsetting (Option)

Three-dimensional tool diameter offsetting is performed to offset a tool in three-dimensional space according to the previously designated three-dimensional vectors.

12-5-1 Function description



As shown in the diagram above, the tool is moved through the tool radius r in the plane-normal vectorial direction of (I, J, K) from the program coordinates (x, y, z) to the offset tool center coordinates (x', y', z') . Also, unlike two-dimensional tool diameter offsetting, which generates vectors perpendicular to the direction of (I, J, K) , three-dimensional tool diameter offsetting generates vectors in the direction of (I, J, K) . (The vectors are generated at the ending point of that block.) The axis components of three-dimensional offset vectors become:

$$H_x = \frac{I}{\sqrt{I^2 + J^2 + K^2}} \cdot r$$

$$H_y = \frac{J}{\sqrt{I^2 + J^2 + K^2}} \cdot r$$

$$H_z = \frac{K}{\sqrt{I^2 + J^2 + K^2}} \cdot r$$

Hence, the tool center coordinates (x', y', z') are expressed as

$$x' = x + H_x$$

$$y' = y + H_y$$

$$z' = z + H_z$$

where (x, y, z) denote the program coordinates.

Note 1: The three-dimensional vectors (H_x, H_y, H_z) refer to plane-normal vectors that are identical to the plane-normal vectors (I, J, K) in direction and have a magnitude of r (tool radius).

Note 2: If parameter **F11** is set to a value other than 0, the value of **F11** will be used as $\sqrt{I^2 + J^2 + K^2}$.

12-5-2 Programming methods

1. G-codes and their functions

G-code	Parameter and feature		
	Offset stroke positive	Offset stroke negative	Offset No. D00
G40	To cancel the 3-dimensional tool diameter offset	To cancel	To cancel
G41	To offset in (I, J, K) direction	To offset in the direction opposite to (I, J, K)	To cancel
G42	To offset in the direction opposite to (I, J, K)	To offset in (I, J, K) direction	To cancel

2. Offset data

For the tool radius r that is to be offset, the offset number under which that offset amount has been registered must be selected using D.

The maximum available number of sets of offset numbers is as follows:

Standard: 128 sets: D1 to D128

Optional: 512 sets: D1 to D512 (max.)

3. Space in which offsetting is to be performed

The space in which offsetting is to be performed is determined by the axis address commands (X, Y, Z, U, V, W) that are contained in the starting block of three-dimensional tool diameter offsetting. When the U-, V-, and W-axes are taken as additions to the X-, Y-, and Z-axes, respectively, priority will be given to the X-, Y-, or Z axis if the X axis and the U axis (or Y and V, or Z and W) are selected at the same time. Coordinate axes that have not been addressed will be interpreted as the X axis, the Y axis, and the Z axis, respectively.

Example:

G41	Xx ₁ Yy ₁ Zz ₁ Ii ₁ Jj ₁ Kk ₁	XYZ space
G41	Yy ₂ Ii ₂ Jj ₂ Kk ₂	XYZ space
G41	Xx ₃ Vv ₃ Zz ₃ Ii ₃ Kk ₃	XVZ space
G41	Ww ₄ Ii ₄ Jj ₄ Kk ₄	XYW space

4. Starting a three-dimensional tool diameter offset operation

Offset number D and the plane-normal vectors (I, J, K) must be set in the same block as that which contains three-dimensional tool diameter offset command code G41 (or G42). In that case, (I, J, K) must be set for each of the X-, Y-, and Z-axes. If this vector setting is not complete (setting of zero for I, J or K is effective), the usual tool diameter offset mode will be set. If, however, the machine does not have the three-dimensional tool diameter offset function, an alarm **838 3-D OFFSET OPTION NOT FOUND** will result.

G41 (G42) Xx₁ Yy₁ Zz₁ Ii₁ Jj₁ Kk₁ Dd₁

G41 (G42) : 3-dimensional tool diameter offset command

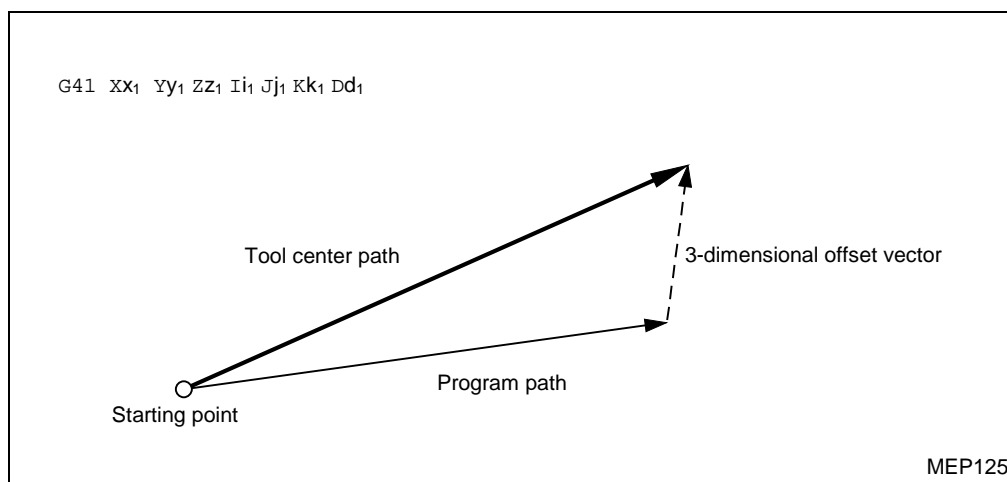
X, Y, Z : Command to move each axis and to determine an offsetting space

I, J, K : To indicate the offsetting direction in plane-normal vectors

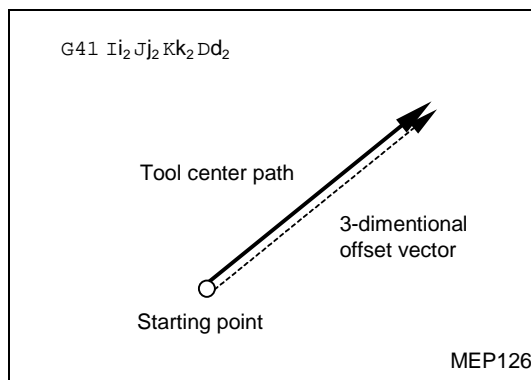
D : Offset number

Use the G00 or G01 mode to start the three-dimensional tool diameter offset operation. Use of the G02 or G03 mode results in an alarm **835 G41, G42 FORMAT ERROR**.

Example 1: If move commands are present:



Example 2: If move commands are not present:

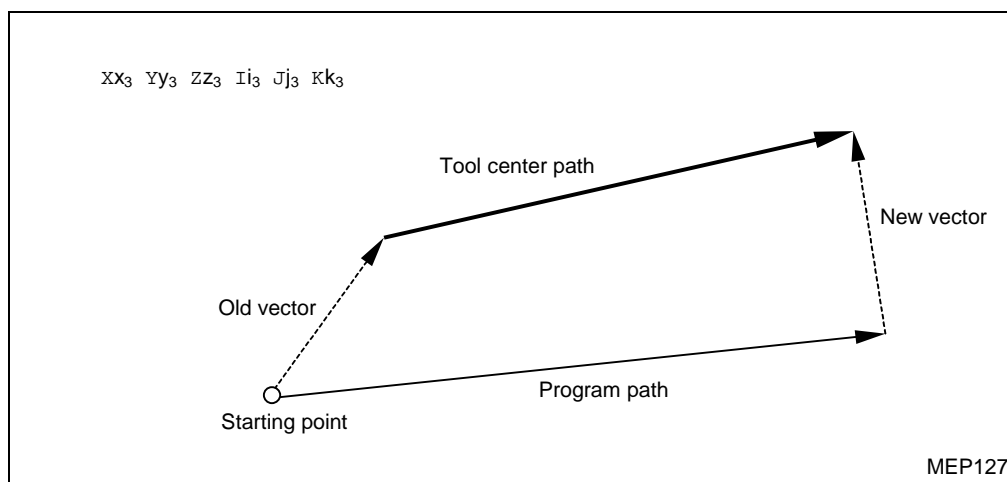


5. During three-dimensional tool diameter offsetting

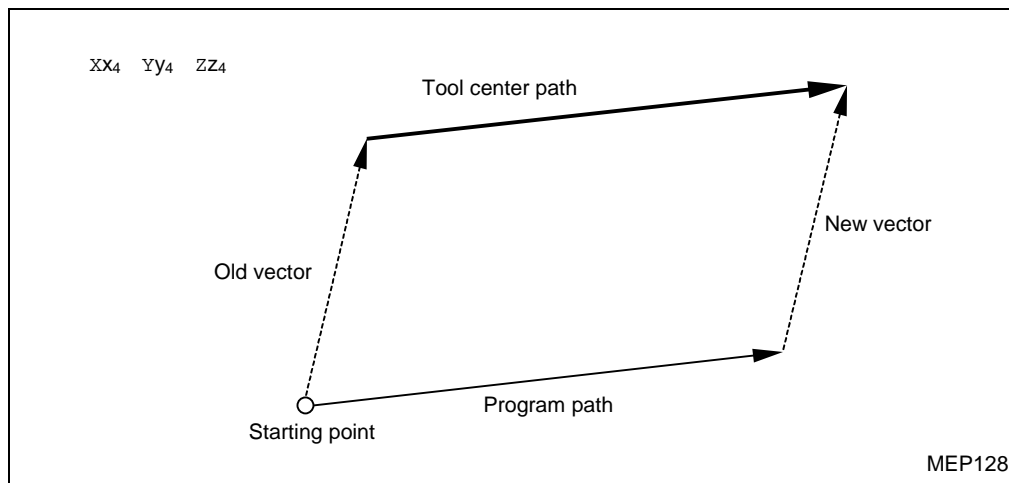
Set move commands and new plane-normal vector commands as follows:

XX_3 YY_3 ZZ_3 Ii_3 Jj_3 Kk_3

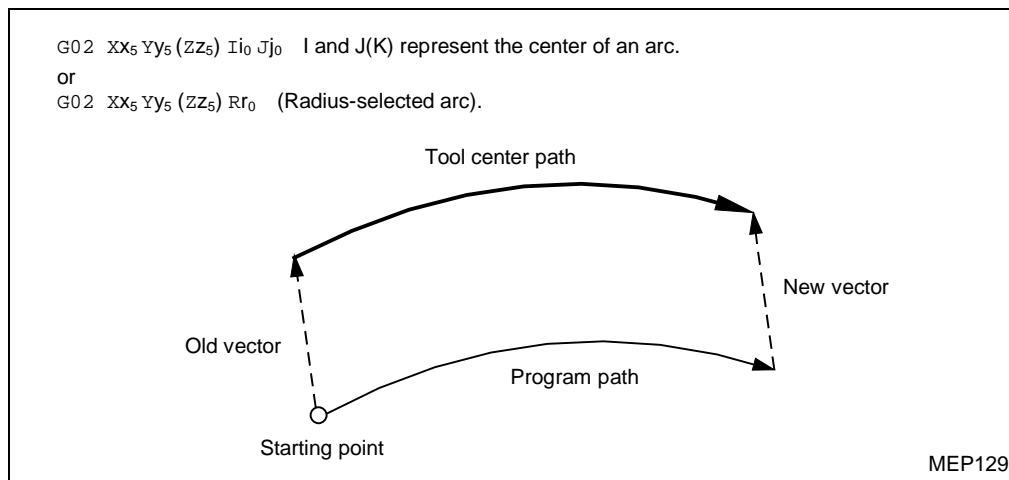
Example 1: If move commands and plane-normal vector commands are present:



Example 2: If plane-normal vector commands are not present:
The new vector is the same as the old one.

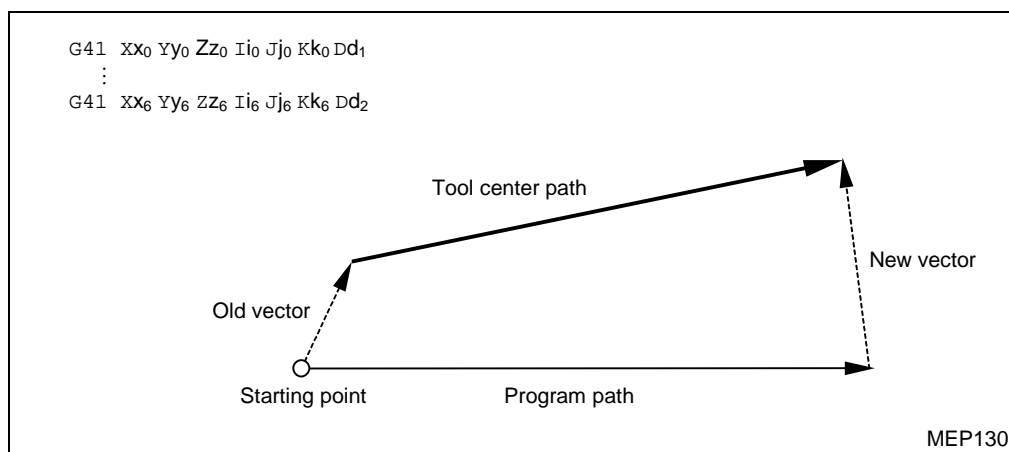


Example 3: For arc or helical cutting:
The new vector is the same as the old one.

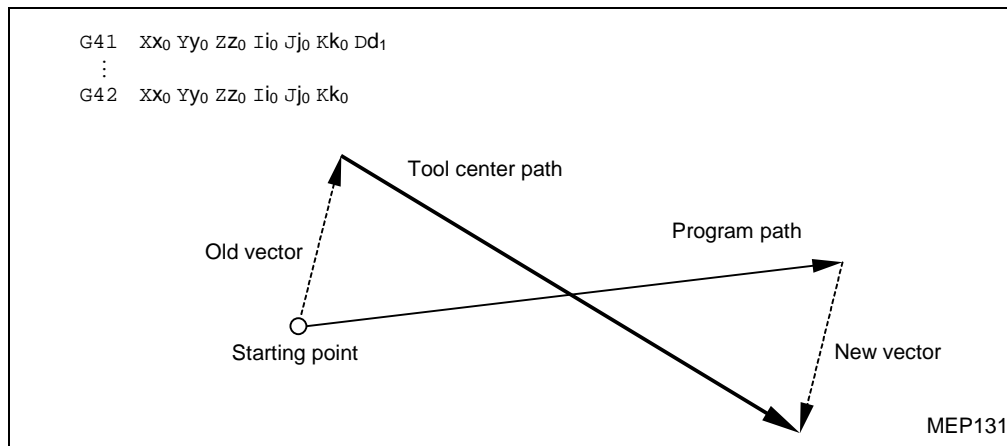


Note: The arc shifts through the amount of vector.

Example 4: For changing the offset data:
Set offset number D in the same block as that of three-dimensional tool diameter offset command G41 or G42. Use the G00 or G01 mode to change the offset data.
Use of the arc mode results in **835 G41, G42 FORMAT ERROR**.



Example 5: For changing the offset direction:



Use the G00 or G01 mode to change the offset direction. Use of the arc mode results in an alarm **835 G41, G42 FORMAT ERROR**.

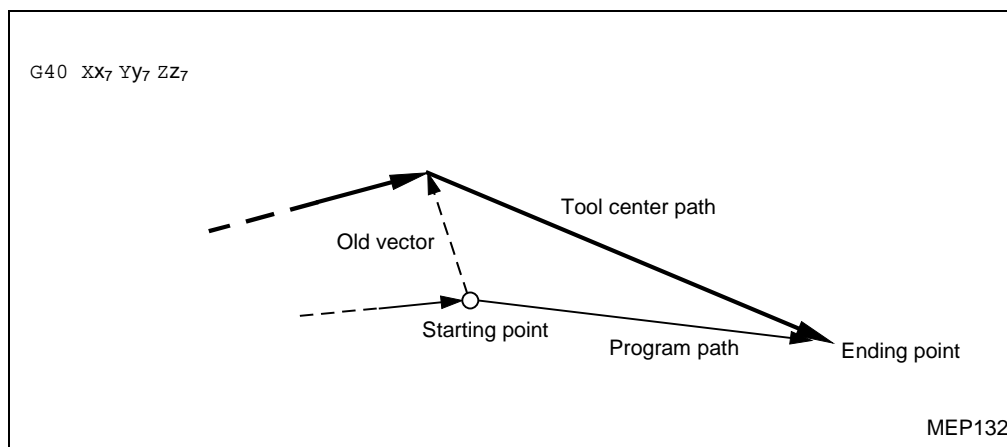
6. Cancelling the three-dimensional tool diameter offset operation

Make the program as follows:

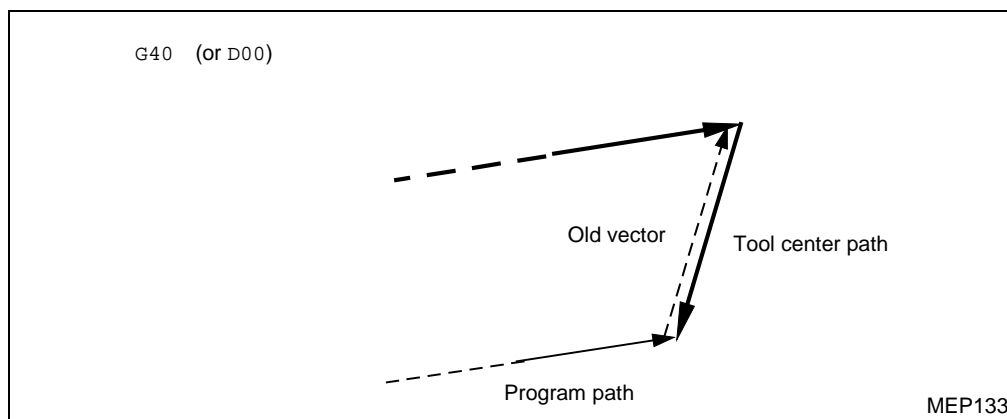
```
G40  Xx7 Yy7 Zz7
```

Use the G00 or G01 mode to cancel three-dimensional tool diameter offsetting. Use of the G02 or G03 mode results in an alarm **835 G41, G42 FORMAT ERROR**.

Example 1: If move commands are present:



Example 2: If move commands are not present:



12-5-3 Correlationships to other functions

1. Tool diameter offset
The usual tool-diameter offset mode will be selected if setting of plane-normal vectors (I, J, K) in the starting block of three-dimensional tool diameter offsetting is not done for each of the X-, Y-, and Z-axes.
2. Tool length offset
Tool length offsetting is performed according to the coordinates existing after execution of three-dimensional tool diameter offsetting.
3. Tool position offset
Tool position offsetting is performed according to the coordinates existing after execution of three-dimensional tool diameter offsetting.
4. Selection of fixed-cycle operation results in an alarm **901 INCORRECT FIXED CYCLE COMMAND**.
5. Scaling
Three-dimensional tool diameter offsetting is performed according to the coordinates existing after execution of scaling.
6. Home position check (G27)
The current offset data is not cancelled.

12-5-4 Miscellaneous notes on three-dimensional tool diameter offsetting

1. Although they can be used to select offset numbers, D-code commands are valid only after command G41 or G42 has been set. If a D-code command is not present, the previous D-code command becomes valid.
2. Use the G00 or G01 mode to change the offset mode, the offset direction or the offset data. An alarm **835 G41, G42 FORMAT ERROR** will occur if an attempt is made to perform these changes in an arc mode.
3. During the three-dimensional tool diameter offset mode using a space, three-dimensional tool diameter offsetting cannot be done using any other space. The cancel command code (G40 or D00) must be executed to select some other offset space.

Example:

G41 X_ Y_ Z_ I_ J_ K_ To start offsetting in X, Y and Z space
:

G41 U_ Y_ Z_ I_ J_ K_ To offset in X, Y and Z space while the U axis
moves by the command value

4. Selection of an offset number falling outside the range from 1 to 128 (for standard machine specifications) or from 1 to 512 (for optional machine specifications) results in an alarm **839 ILLEGAL OFFSET No.**
5. Only the G40 or D00 command code can be used to cancel three-dimensional tool diameter offsetting. Cancellation is not possible with the NC reset key or external reset functions.
6. A program error will result if the vectorial magnitude specified by (I, J, K), that is $\sqrt{I^2 + J^2 + K^2}$, overflows.

12-6 Programmed Input of Offset Data: G10

1. Function and purpose

The G10 command allows tool offset data, work offset data and parameter data to be set or modified in the flow of program.

2. Programming formats

A. Programming workpiece offsets

- Programming format for the workpiece origin data

G10 L2 P_ X_ Y_ Z_ α _ (α : Additional axis)

P: 0.....Coordinate shift (Added feature)

1.....G54

2.....G55

3.....G56

4.....G57

5.....G58

6.....G59

Data of P-commands other than those listed above are handled as P = 1.

If P-command setting is omitted, the workpiece offsets will be handled as currently effective ones.

- Programming format for the additional workpiece origin data (option)

G10 L20 P_ X_ Y_ Z_ α _ (α : Additional axis)

P1: G54.1 P1

P2: G54.1 P2

⋮

P47: G54.1 P47

P48: G54.1 P48

The setting ranges of the data at axial addresses are as follows:

Table 12-1

	Metric (F91 bit 4 = 0)	Inch (F91 bit 4 = 1)
Linear axis	–99999.9999 to 99999.9999	–9999.99999 to 9999.99999
Rotational axis	–99999.9999 to 99999.9999	–99999.9999 to 99999.9999

B. Programming tool offsets

- Programming format for the tool offset memory of Type A

G10 L10 P_ R_

P: Offset number

R: Offset amount

- Programming format for the tool offset memory of Type B

G10 L10 P_ R_ Shape offset concerning the length

G10 L11 P_ R_ Wear offset concerning the length

G10 L12 P_ R_ Shape offset concerning the diameter

G10 L13 P_ R_ Wear offset concerning the diameter

The setting ranges for programming tool offset data are as follows:

Offset number (P):

1 to 128 or 512 (according to the number of available data sets)

Offset amount (R):

Table 12-2

	Metric (F91 bit 4 = 0)	Inch (F91 bit 4 = 1)
Shape offset (Tool offset)	± 1999.9999 mm	± 84.50000 in.
Wear offset	± 99.999 mm	± 9.99999 in.

C. Programming parameter data

G10 L50..... Parameter input mode ON

N_P_R_

N_R_

G11 Parameter input mode OFF

N: Parameter number

P: Axis number (for axis type parameter)

R: Data of parameter

Specify the parameters with address N as indicated below:

Table 12-3

Parameter		N: Number	P: Axis No.
A	1 to 108	1001 to 1108	—
B	1 to 108	2001 to 2108	—
C	1 to 108	3001 to 3108	—
D	1 to 144	4001 to 4144	—
E	1 to 144	5001 to 5144	—
F	1 to 168 (Setting prohibited for 47 to 66)	6001 to 6168	—
I	1 to 24	9001 to 9024	1 to 14
J	1 to 144	10001 to 10144	—
K	1 to 144	11001 to 11144	—
L	1 to 144	12001 to 12144	—
M	1 to 48	13001 to 13048	1 to 14
N	1 to 48	14001 to 14048	1 to 14
P	1 to 5	150001 to 150005	1 to 14
#	0 to 4095	150100 to 154195	1 to 14
S	1 to 48	16001 to 16048	1 to 14
SV	1 to 96	17001 to 17096	1 to 14
SP	1 to 384	18001 to 18384	1 to 4
SA	1 to 144	19001 to 19144	1 to 4
BA	1 to 132	20001 to 20132	—

Note: As for the setting ranges of parameter data, refer to the Parameter List.

3. Detailed description

A. Workpiece origin data input

1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59, G90 and G91.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.
3. Irrespective of workpiece offset type (G54 - G59 and G54.1), the data to the axial addresses have to refer to the origin of the fundamental machine coordinate system.
4. Depending upon the data input mode — absolute (G90) or incremental (G91) — the designated data will overwrite, or will be added to, the existing data.
5. L-code and P-code commands can be omitted, indeed, but take notice of the following when omitting them:
 - 1) Omit both L-code and P-code commands only when
The axial data should refer to the coordinate system that was last selected.
 - 2) The L-code command only may be omitted when the intended axial data refer to a coordinate system of the same type (in terms of L-code: L2 or L20) as the last selected one; give a P-command in such a case as follows:
 - Set an integer from 0 to 6 with address P to specify the coordinate shift data or one of the coordinate systems from G54 to G59.
 - Set an integer from 1 to 48 with address P to specify one of the additional workpiece coordinate systems of G54.1.
 - 3) If the P-code command only is omitted:
An alarm will result if the value of L mismatches the coordinate system last selected.
6. The origin data updated by a G10 command are not indicated as they are on the **WORK OFFSET** display until that display has been selected anew.
7. Setting an illegal L-code value causes an alarm.
8. Setting an illegal P-code value causes an alarm.
9. Setting an illegal axial value causes an alarm.
10. The G10 command is invalid (or skipped) during tool path check.

B. Tool offset data input

1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59, G90 and G91.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.
3. Depending upon the data input mode — absolute (G90) or incremental (G91) — the designated data will overwrite, or will be added to, the existing data.
4. The offset data updated by a G10 command are not indicated as they are on the **TOOL OFFSET** display until that display has been selected anew.
5. Setting an illegal L-code value causes an alarm.
6. A command of "G10 P_ R_" without an L-code is also available for tool offset data input.
7. Setting an illegal P-code value causes an alarm.

8. Setting an illegal offset value (R) causes an alarm.
9. The G10 command is invalid (or skipped) during tool path check.

C. Parameter data input

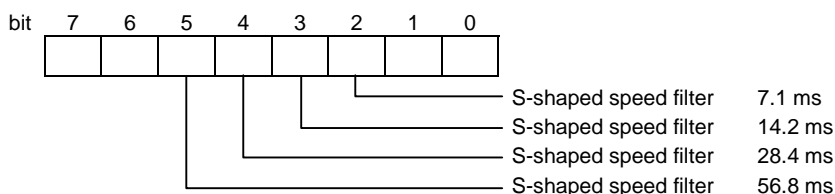
1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59, G90 and G91.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.
3. Other NC statements must not be given in the parameter input mode.
4. No sequence number must be designated with address N in the parameter input mode.
5. Irrespective of the data input mode — absolute (G90) or incremental (G91) — the designated data will overwrite the existing parameter. Moreover, describe all the data in decimal numbers (hexadecimal and bit type data, therefore, must be converted).

Example: For changing a bit type data of 00110110 to 00110111:

Since $(00110111)_2 = (55)_{10}$ [a binary number of 00110111 corresponds to “55” in decimal notation], set 55 with address R.

6. All decimal places, even if inputted, are ignored.
7. Some specific bit-type parameters require selection of one of multiple bits. For the parameter shown as an example below, set data that turns on only one of bits 2 to 5.

Example: Parameter **K107**



Setting “1” for bits 2 and 3, for example, could not make valid a speed filter of 21.3 msec (= 7.1 + 14.2).

8. The parameter data updated by a G10 L50 command are not made valid till the execution of a G11 command.
9. The parameter data updated by a G10 L50 command are not indicated as they are on the **PARAMETER** display until that display has been selected anew.
10. Setting an illegal L-code value causes an alarm.
11. Setting an illegal N-code value (parameter No.) causes an alarm.
12. Omission of P-code for an axis type parameter causes an alarm.
13. Setting an illegal parameter value with address R causes an alarm.
14. The G10 command is invalid (or skipped) during tool path check.

4. Sample programs

A. Entering tool offset data from tape

```

... G10L10P10R-12345 G10L10P05R98765 G10L10P40R2468 ...
H10 = -12345    H05 = 98765    H40 = 2468
  
```

B. Updating tool offset data

Example 1: Assumes that H10 has already been set equal to -1000.

```

N1 G01 G90 G43 Z-100000 H10 (Z = -101000)
N2 G28 Z0
N3 G91 G10 L10 P10 R-500 (-500 is added in the G91 mode.)
N4 G01 G90 G43 Z-100000 H10 (Z = -101500)

```

Example 2: Assumes that H10 has already been set equal to -1000.

Main program

```

N1 G00 X100000 ..... a
N2 #1=-1000
N3 M98 P1111L4 ..... b1, b2, b3, b4

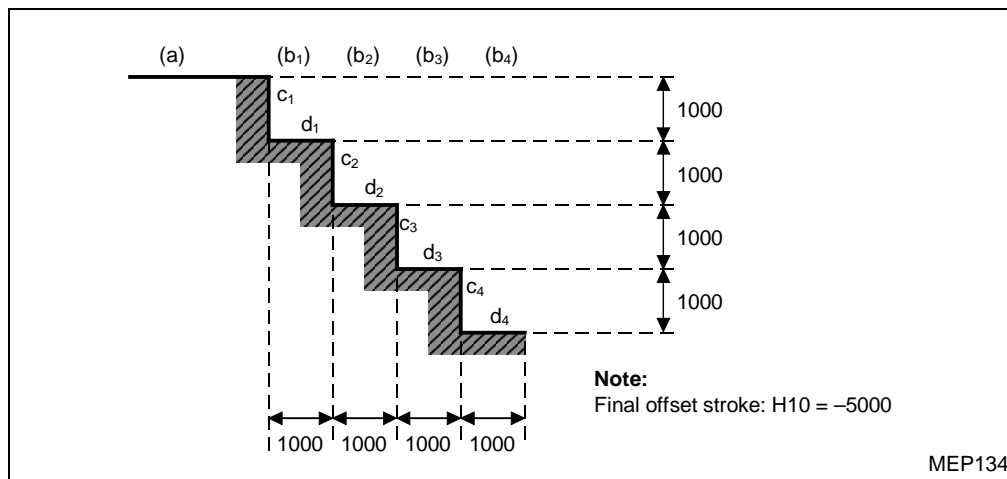
```

Subprogram O1111

```

N1 G01 G91 G43 Z0 H10 F100 ..... c1, c2, c3, c4
N2 G01 X1000 ..... d1, d2, d3, d4
N3 #1=#1-1000
N4 G90 G10 L10 P10 R#1
N5 M99

```



Example 3: The programs in Example 2 above can be rewritten as follows:

Main program

```

N1 G00 X100000
N2 M98 P1111 L4

```

Subprogram O1111

```

N1 G01 G91 G43 Z0 H10 F100
N2 G01 X1000
N3 G10 L10 P10 R-1000
N4 M99

```

Note: Even when the command code is displayed on <Next Command>, the current offset number and variables will remain unupdated until that command is executed.

```

N1 G10 L10 P10 R-100
N2 G43 Z-10000 H10
N3 G0 X-10000 Y-10000
N4 G10 L10 P10 R-200

```

Executing block N4 will cause an offset stroke in H10 to be updated.

C. Updating the workpiece coordinate system offset data

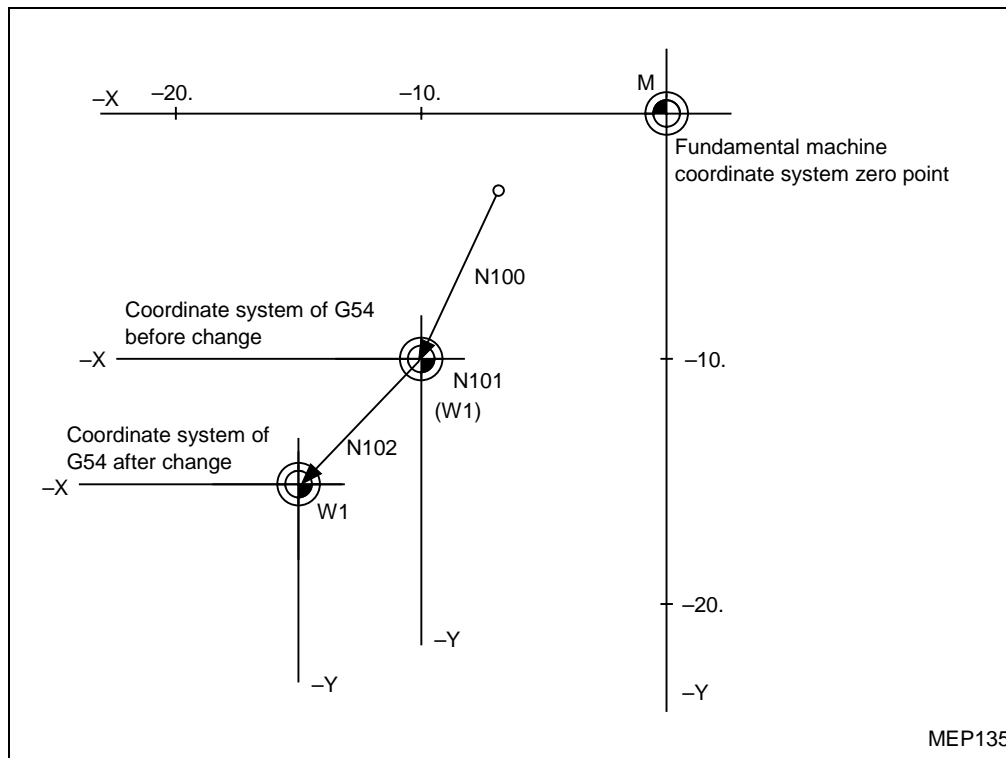
Assume that the previous workpiece coordinate system offset data is as follows:

$X = -10.000$ $Y = -10.000$

```

:
N100 G00 G90 G54 X0 Y0
N101 G10 L2 P1 X-15.000 Y-15.000
N102 X0 Y0
:
M02

```



Note 1: Changes in the display of the workpiece position at N101

At N101, the display of tool position in the G54 coordinate system changes before and after workpiece coordinate system updating with G10.

$X = 0$ \Rightarrow $X = +5.000$
 $Y = 0$ $Y = +5.000$

Note 2: Prepare the following program to set workpiece coordinate system offset data in G54 to G59:

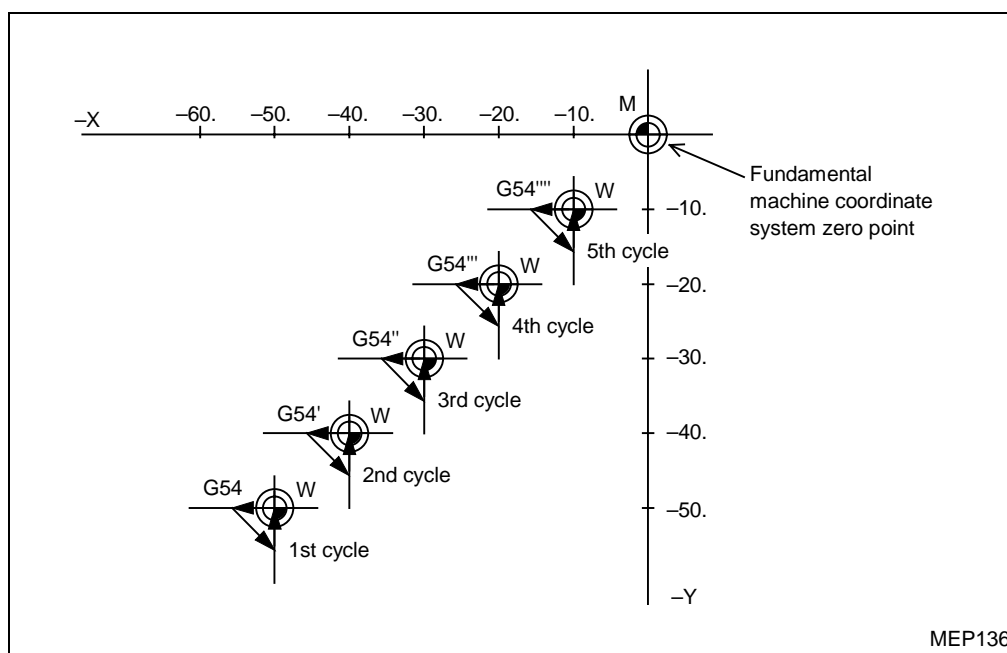
```

G10L2P1X-10.000 Y-10.000
G10L2P2X-20.000 Y-20.000
G10L2P3X-30.000 Y-30.000
G10L2P4X-40.000 Y-40.000
G10L2P5X-50.000 Y-50.000
G10L2P6X-60.000 Y-60.000

```

D. Programming for using one workpiece coordinate system as multiple workpiece coordinate systems

Main program	{	⋮
		#1=-50. #2=10.
		M98 P200 L5
		⋮
		M02
Subprogram (O200)	{	%
		N1 G90 G54 G10 L2 P1 X#1 Y#1
		N2 G00 X0 Y0
		N3 X-5. F100
		N4 X0 Y-5.
		N5 Y0
		N6 #1=#1+#2
		N7 M99
		%



E. Programming for parameter data input

G10L50	Parameter input mode ON
N4017R10	D17 is set to "10".
N6088R96	F88 is set to "01100000". [$(01100000)_2 = (96)_{10}$]
N12067R-1000	L67 is set to "-1000".
N12072R67	L72 is set to "0x43". [$(43)_{16} = (67)_{10}$]
N15004P1R50	P4 data for the 1st axis (X-axis) is set to "50".
G11	Parameter input mode OFF

5. Related alarms

Alarm No.	Alarm message	Cause	Remedy
807	ILLEGAL FORMAT	<p>Work offset input: P-command is omitted in a block of G10 L20 (or L2) although the last selected coordinate system is one of the systems from G54 to G59 (or of the G54.1 systems).</p> <p>Parameter input: An illegal parameter number is set.</p>	Review the program data.
809	ILLEGAL NUMBER INPUT	<p>Work offset input: The setting range of the coordinate system number or the offset data is overstepped.</p> <p>Tool offset input: The setting range of the offset data is overstepped.</p> <p>Parameter input: The axis number is not specified for an axis type parameter. The setting range of the axis number or the parameter data is overstepped.</p>	Review the program data.
839	ILLEGAL OFFSET No.	<p>Tool offset input: The specified offset number is greater than the number of available data sets.</p>	Correct the offset number according to the number of available data sets.
903	ILLEGAL G10 L NUMBER	<p>Work offset input: A command of G10 L20 is set although the corresponding function for the G54.1 coordinate systems is not provided.</p>	Give an available L-code command.

12-7 Tool Offsetting Based on MAZATROL Tool Data

Parameter selection allows you to offset both the tool length and the tool diameter using MAZATROL tool data (tool diameter and tool length data).

12-7-1 Selecting parameters

Using the following parameters, select whether or not MAZATROL tool data is to be used:
User parameters

F92 bit 7 = 1: Tool diameter offsetting uses the MAZATROL tool data **ACT-φ** (tool diameter data).

F93 bit 3 = 1: Tool length offsetting uses the MAZATROL tool data **LENGTH** (tool length data).

F94 bit 2 = 1: Tool length offsetting using the MAZATROL tool data is prevented from being cancelled by a reference-point return command.

P94 bit 7 = 1: Tool offsetting uses the MAZATROL tool data **ACT-φ CORR.** (or **No.**) and **LENG CORR.** (or **No.**).
(Set **P94** bit 7 to 0 to use the data stored on the **TOOL OFFSET** display.)

12-7-2 Tool length offsetting

1. Function and purpose

Even when offset data is not programmed, tool length offsetting will be performed according to the MAZATROL tool data **LENGTH** that corresponds to the designated tool number.

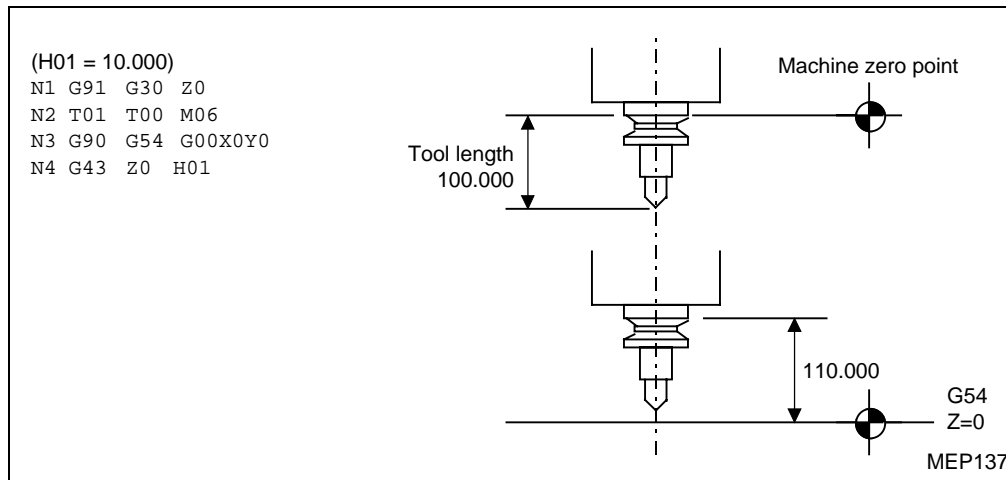
2. Parameter setting

Set both bit 3 of parameter **F93** and bit 2 of parameter **F94** to 1.

3. Detailed description

1. Tool length offsetting is performed automatically, but its timing and method differ as follows:
 - After a tool change command has been issued, offsetting is performed according to the **LENGTH** data of the tool mounted in the spindle. (A tool change command code must be set in the program before tool length offsetting can be done.)
 - After command G43 has been set, offsetting is performed according to the **LENGTH** data of the tool mounted in the spindle.
2. Tool length offsetting is cancelled in the following cases:
 - When a command for tool change with some other tool is executed
 - When M02 or M30 is executed
 - When the reset key is pressed
 - When command G49 is issued
 - When a reference-point return command is executed with bit 2 of parameter **F94** set to 0
3. Tool length offsetting becomes valid for the block onward that first involves Z-axis movement after tool change.

4. If this offset function is used with a G43 H-command, offsetting will use as its offset data the sum total of the MAZATROL tool data **LENGTH** and the offset amount specified by the G43 H (or G44 H) command.



- Note 1:** Set G43 H0 if tool length offsetting is to be done using a G43 H-command and only the offset amount specified by H is to be cancelled.
- Note 2:** With a G44 command, tool length offsetting based on MAZATROL tool data is not performed.
- Note 3:** The restart operation must begin from a position before a G43 command code or a tool change command code. Even when the spindle has a mounted tool, G43 or the tool change command must be executed before offsetting based on MAZATROL tool data can take place.
- Note 4:** Offsetting will fail if registered MAZATROL tool data **LENGTH** is not present.
- Note 5:** For an EIA/ISO program, to carry out tool length offset operations using the tool length data included in MAZATROL tool data, it becomes necessary to set data in the validation parameter for the tool length data of the MAZATROL tool data and to insert a tool change T- and M-code command block. It is to be noted that the tool change command block may not be missed particularly in the following cases:
- During automatic operation, if the first tool to be used has already been mounted in the spindle.
 - During call of an EIA/ISO program as a subprogram from the MAZATROL main program, if the tool to be used immediately prior to call of the subprogram is the same as that which is to be designated in that subprogram as the first tool to be used.

12-7-3 Tool diameter offsetting

1. Function and purpose

Tool diameter offsetting by a G41 or G42 command uses MAZATROL tool data **ACT-φ** as the offset amounts.

2. Parameter setting

Set bit 7 of parameter **F92** to 1.

3. Detailed description

- Tool diameter offsetting uses as its offset amounts the diameter data of the tool which is mounted in the spindle at the issuance of G41/G42.
- Tool diameter offsetting is cancelled by G40.
- If the tool diameter offset function is used with a D-command, the sum total of the data indicated by the corresponding offset number (D) and the radius of the tool will be used as the offset data.

Note 1: The tool used must be mounted in the spindle before restarting the program.

Note 2: Offsetting based on tool diameter data will not occur if registered MAZATROL tool diameter data is not present or if a tool for which tool diameter data cannot be entered is to be used.

Note 3: To carry out for an EIA/ISO program the tool diameter offset operations using the tool diameter data included in MAZATROL tool data, it is necessary to insert tool change command blocks, as it is the case for tool length offsetting (refer to **Note 5** in Subsection 12-7-2).

12-7-4 Tool data update (during automatic operation)

1. Function and purpose

Tool Data Update allows MAZATROL tool data to be updated during automatic operation based on an EIA/ISO program.

2. Parameter setting

Set parameter **L57** to 1.

3. Detailed description

This function allows the entire tool data, except for spindle tools, to be updated during automatic operation based on an EIA/ISO program.

Parameter	TOOL	NOM-φ	ACT-φ	LENGTH	COMP.	THR/HP	LIFE	TIME	MAT.	REV.
L57 = 0	No	No	No	No	No	No	Yes	Yes	No	Yes
L57 = 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note 1: In the table given above, “Yes” indicates that you can update the data, and “No” indicates that you cannot update the data.

Identification between MAZATROL programs and EIA/ISO programs is automatically made by whether the program currently being executed, is MAZATROL or EIA/ISO, irrespective of whether it is a main program or subprogram.

If, however, the main program is MAZATROL and its subprograms are EIA/ISO, then the currently active set of programs is regarded as a MAZATROL program.

Note 2: An alarm **428 MEMORY PROTECT (AUTO OPERATION)** will occur if the spindle tool data is modified during automatic operation based on an EIA/ISO program.

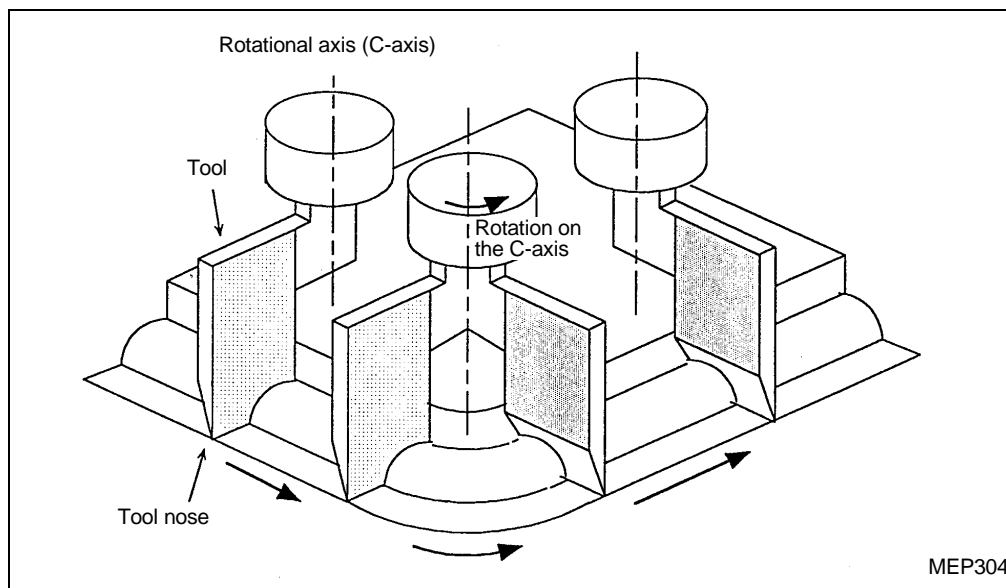
12-8 Shaping Function (Option)

12-8-1 Overview

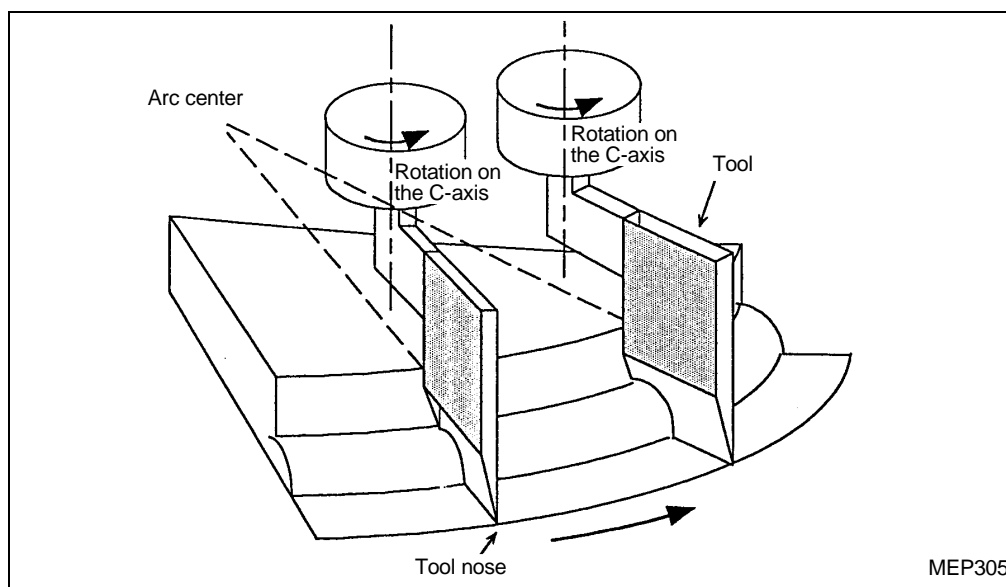
The shaping function is provided to control the rotational axis (C-axis) in order to keep the tool set normal (perpendicular) to the direction of the movement in the XY-plane.

This optional function permits free-form shapes such as the rubber oil-seal surface to be cut out for a better surface finish than with an end-milling tool.

- The C-axis control is automatically performed at block connections to keep the tool normally oriented.



- During circular interpolation, the C-axis is continuously controlled in synchronization with the tool movement.



12-8-2 Programming format

```
G40.1 }
G41.1 } Xx Yy Ff
G42.1 }
```

G40.1: Cancellation of shaping

G41.1: Selection of shaping to the left (normal orientation on the left side)

G42.1: Selection of shaping to the right (normal orientation on the right side)

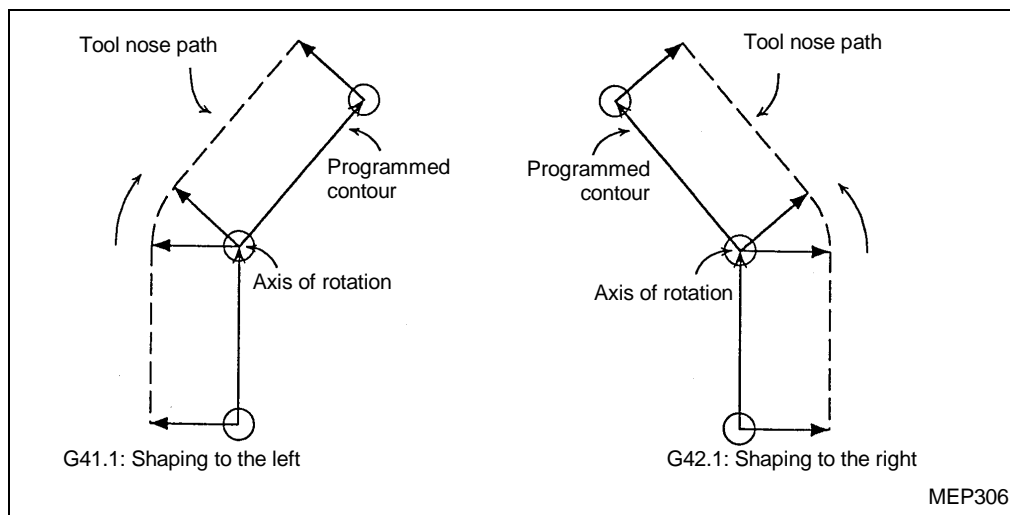
x: X-axial position of ending point

y: Y-axial position of ending point

f: Feed rate

Note 1: The codes G40.1, G41.1 and G42.1 belong to group 15 of G-codes.

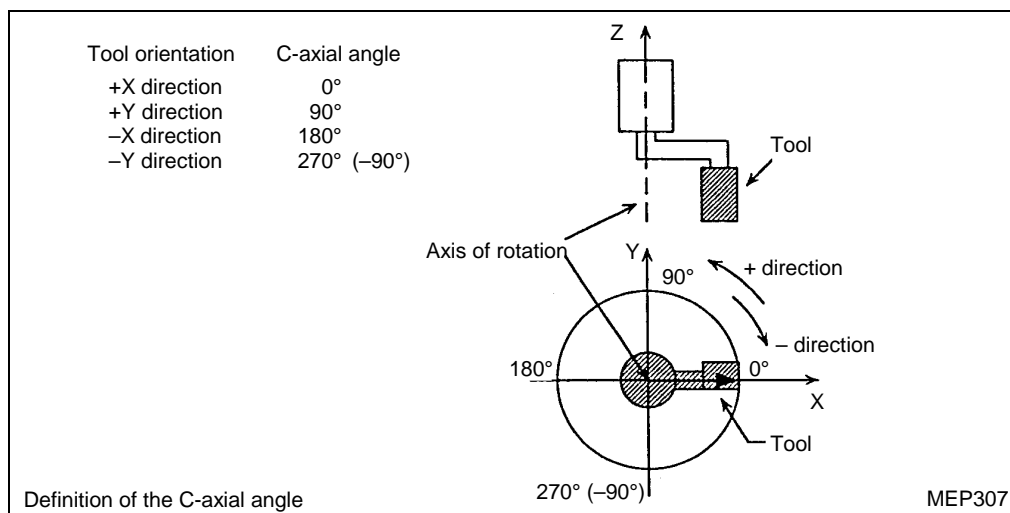
Note 2: The shaping control (orientation of the tool) can only be performed in the XY-plane, regardless of the plane currently selected.

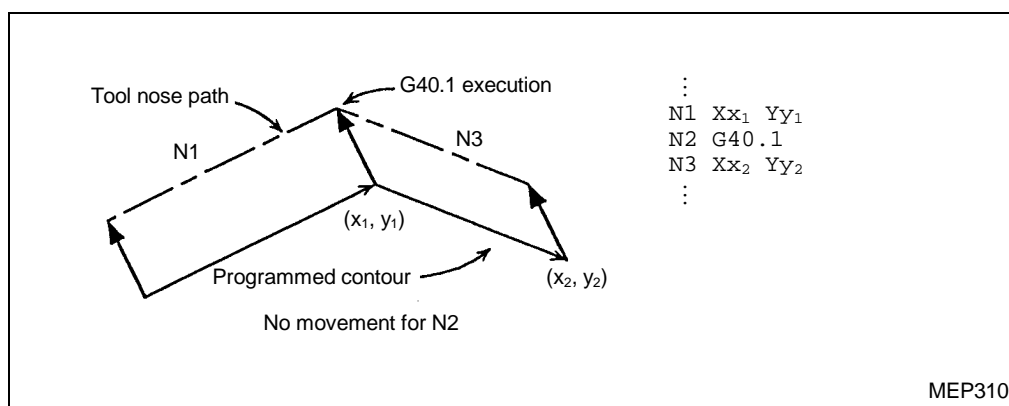


12-8-3 Detailed description

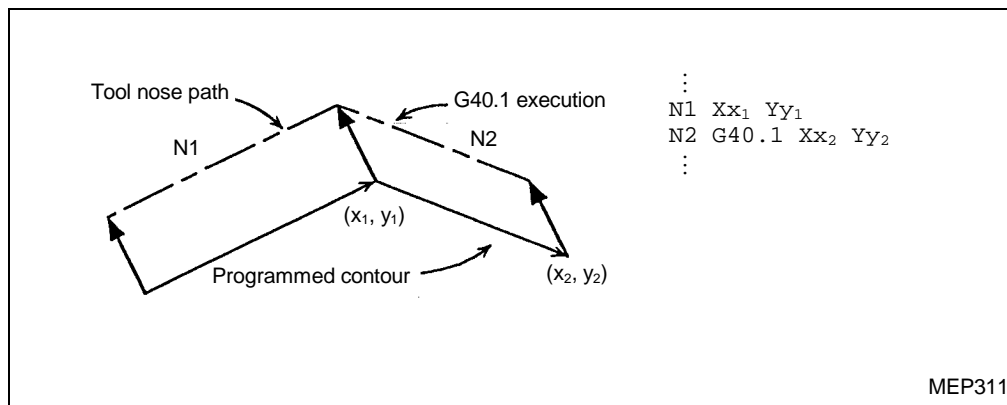
1. Definition of the C-axial angle

The C-axial angle with the tool oriented in the +X direction is defined as 0°, and counterclockwise rotation is defined as positive (+).





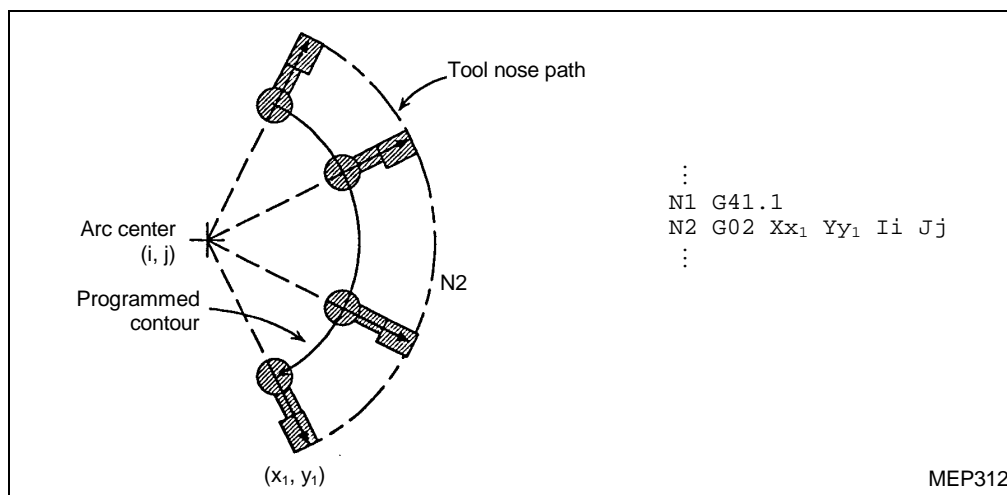
- Cancellation in a block containing motion command



C. Movement in the shaping mode

Execution of a block

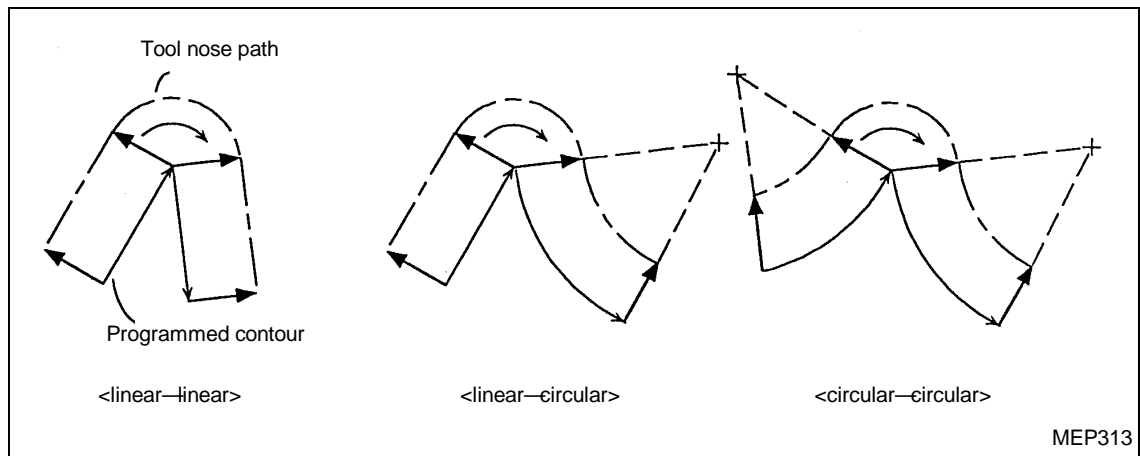
- Block of linear interpolation
The tool moves linearly without C-axis rotation.
- Block of circular interpolation
The angular position on the C-axis is continuously controlled in synchronization with the circular movement of the tool.



Connection between blocks

Without tool diameter offsetting

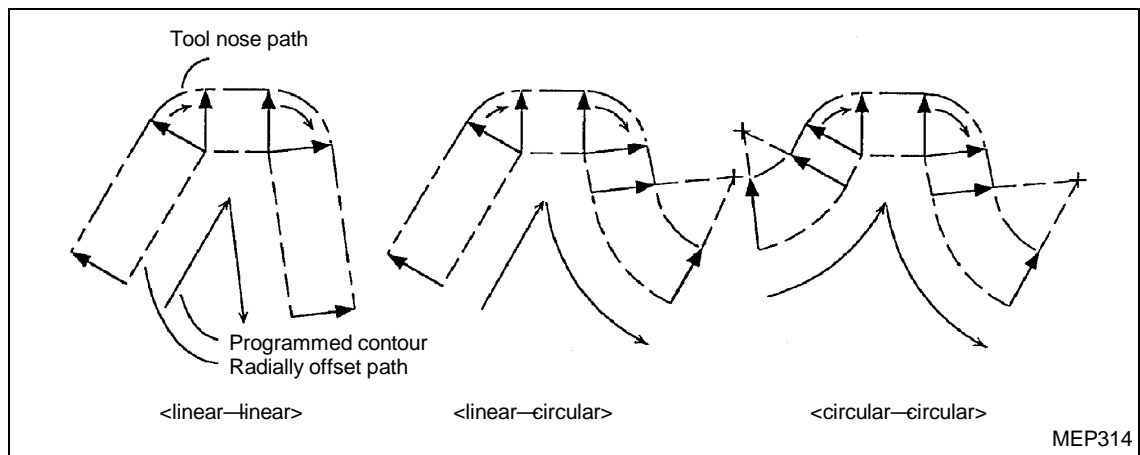
An independent C-axis rotation is performed at the end of the preceding block to orient the tool in the normal direction with respect to the starting motion of the next block.



- With tool diameter offsetting

The tool diameter offsetting automatically inserts linear segments for connection between blocks whose paths cross each other at a sharp angle.

The shaping function controls the C-axis so as to orient the tool in accordance with the offset tool path.



Direction of C-axis rotation at block connections

The rotation on the C-axis occurs in the negative direction (clockwise) in the mode of G41.1, or in the positive direction (counterclockwise) in the mode of G42.1, at block connections.

Parameter **K2** (ε : minimum allowable angle of C-axis rotation) is provided to suppress the rotation as described below.

- Direction of C-axis rotation at block connections

For G41.1: negative (CW)

For G42.1: positive (CCW)

- Suppression or prohibition of C-axis rotation at block connections

θ : Rotational angle required

ε : Parameter **K2** (minimum allowable angle of C-axis rotation)

$$|\theta| < \varepsilon$$

The C-axis rotation is suppressed.

In the mode of G41.1:

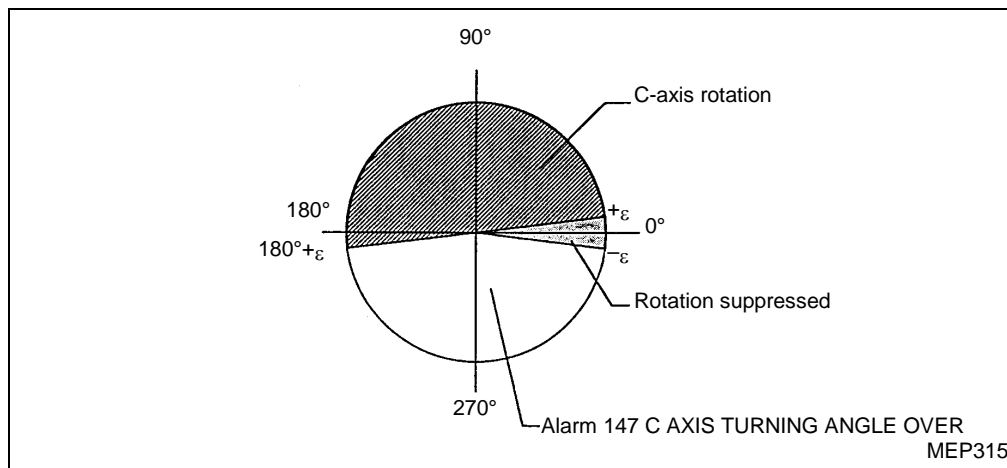
$$\varepsilon \leq \theta < 180^\circ - \varepsilon$$

Alarm No. **147 C AXIS TURNING ANGLE OVER** will be caused.

In the mode of G42.1:

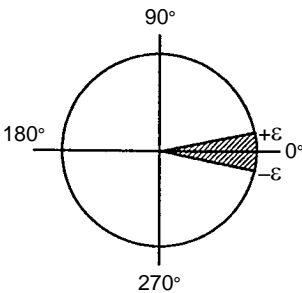
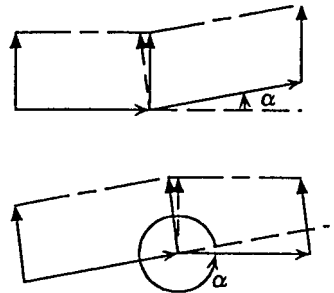
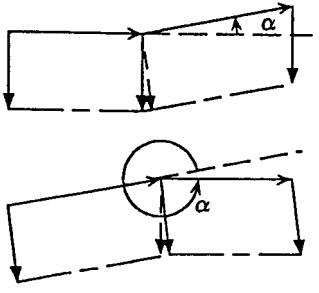
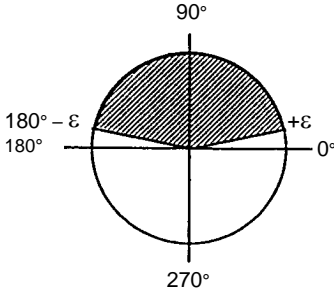
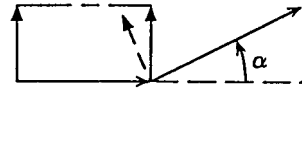
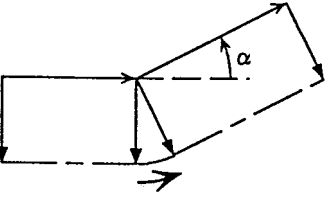
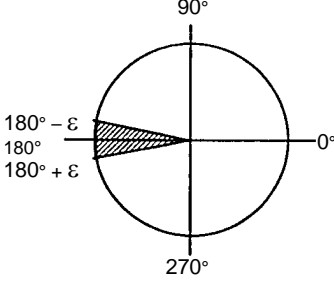
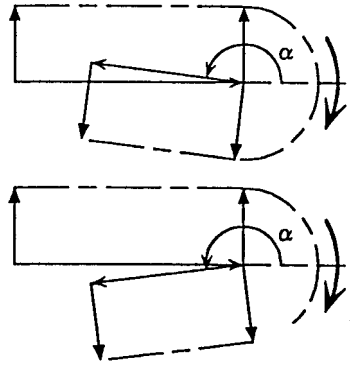
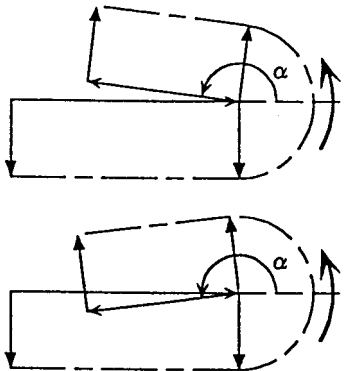
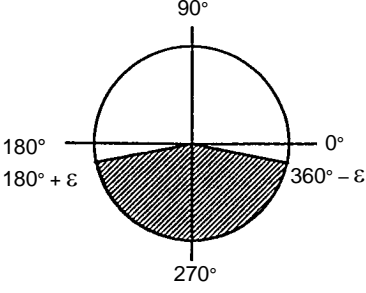
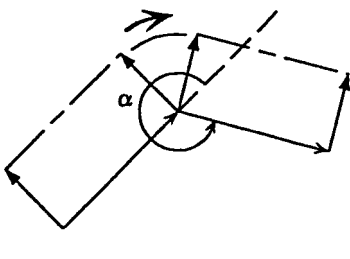
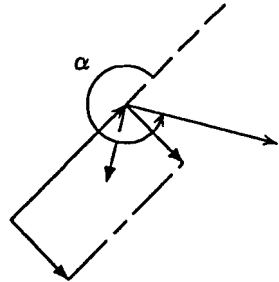
$$180^\circ + \varepsilon \leq \theta < 360^\circ - \varepsilon$$

Alarm No. **147 C AXIS TURNING ANGLE OVER** will be caused.



Note: The C-axis rotation is suppressed if the angle of rotation required is smaller than parameter **K2** ($|\theta| < \varepsilon$).

The rotational angle thus ignored will surely be added to the angle of the next rotation required, which will then be actually performed or further suppressed according to the result of the accumulation.

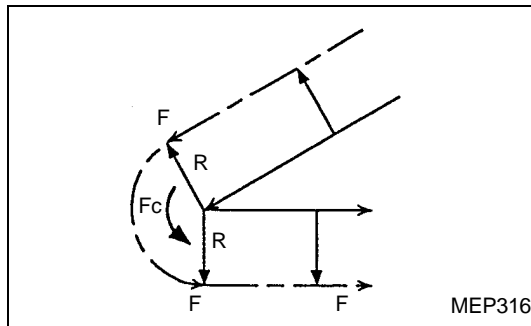
Angle at a block connection: α	G41.1	G42.1
1. $-\varepsilon < \alpha < +\varepsilon$ 	 C-axis rotation suppressed	 C-axis rotation suppressed
2. $+\varepsilon < \alpha < (180^\circ - \varepsilon)$ 	 Alarm 147 C AXIS TURNING ANGLE OVER	
3. $(180^\circ - \varepsilon) \leq \alpha \leq (180^\circ + \varepsilon)$ 		
4. $(180^\circ + \varepsilon) \leq \alpha \leq (360^\circ - \varepsilon)$ 		 Alarm 147 C AXIS TURNING ANGLE OVER

3. Speed of C-axis rotation for shaping

- At block connection

The C-axis rotation is performed at such a speed that the tool nose will move at the speed specified by the F-code.

The C-axis rotational speed F_c is calculated as follows:



If parameter **K1** (radius of C-axis rotation) $\neq 0$

$$F_c = \frac{F}{R} \times \frac{180}{\pi} \text{ (deg/min)}$$

If parameter **K1** (radius of C-axis rotation) $= 0$

$$F_c = F \times \frac{180}{\pi} \text{ (deg/min)}$$

F: Feed rate (mm/min)

R: Parameter **K1** (mm) [radius of C-axis rotation (distance between C-axis and tool nose)]

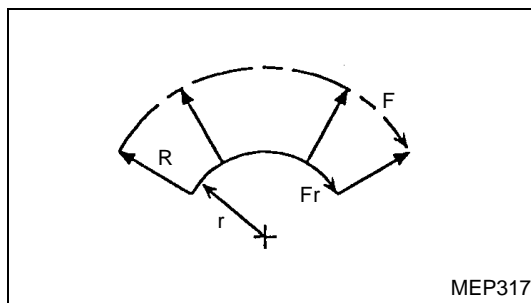
The C-axis rotation, however, is controlled in order that the preset maximum allowable cutting speed of the C-axis should not be exceeded, regardless of the result F_c of the above calculation.

Similar formulae apply to the rapid traverse.

- During circular interpolation

The circular interpolation is performed at such a speed that the tool nose will move at the speed specified by the F-code.

The cutting feed rate of the circular interpolation (F_r) is calculated as follows:



$$F_r = F \times \frac{r}{R + r} \text{ (mm/min)}$$

F: Feed rate (mm/min)

r : Radius of circular interpolation (mm)

R: Parameter **K1** (mm) [radius of C-axis rotation (distance between C-axis and tool nose)]

The speed of the circular interpolation (F), however, is automatically controlled in order that the preset maximum allowable cutting speed of the C-axis should not be exceeded.

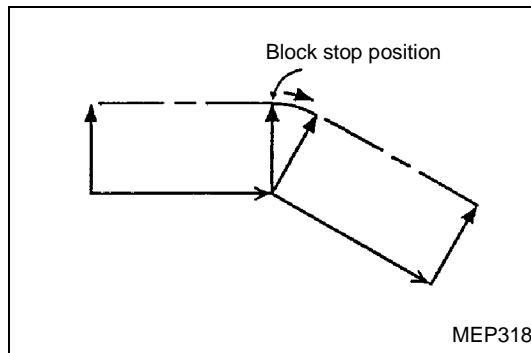
12-8-4 Remarks

1. If the axis of the work spindle is to be used for shaping control, the spindle axis must be changed over to a servoaxis (C-axis). The following M-codes are provided to select the control mode of the work spindle.

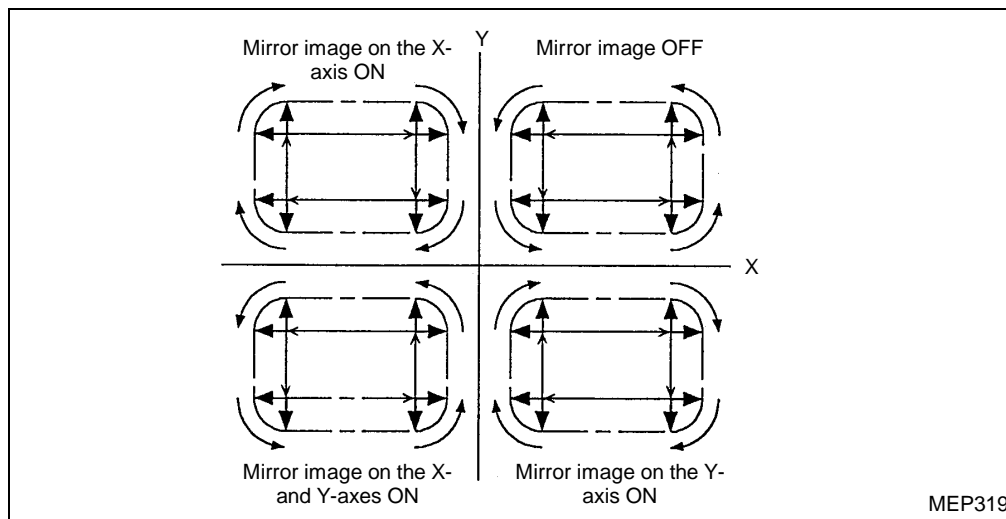
M193: Selection of spindle as the C-axis (Servo On)

M194: Selection of spindle as the milling spindle (Servo Off)

2. In the mode of single-block operation, interlocking at the start of cutting block or each block, the operation will be stopped before the preparatory rotation on the C-axis.



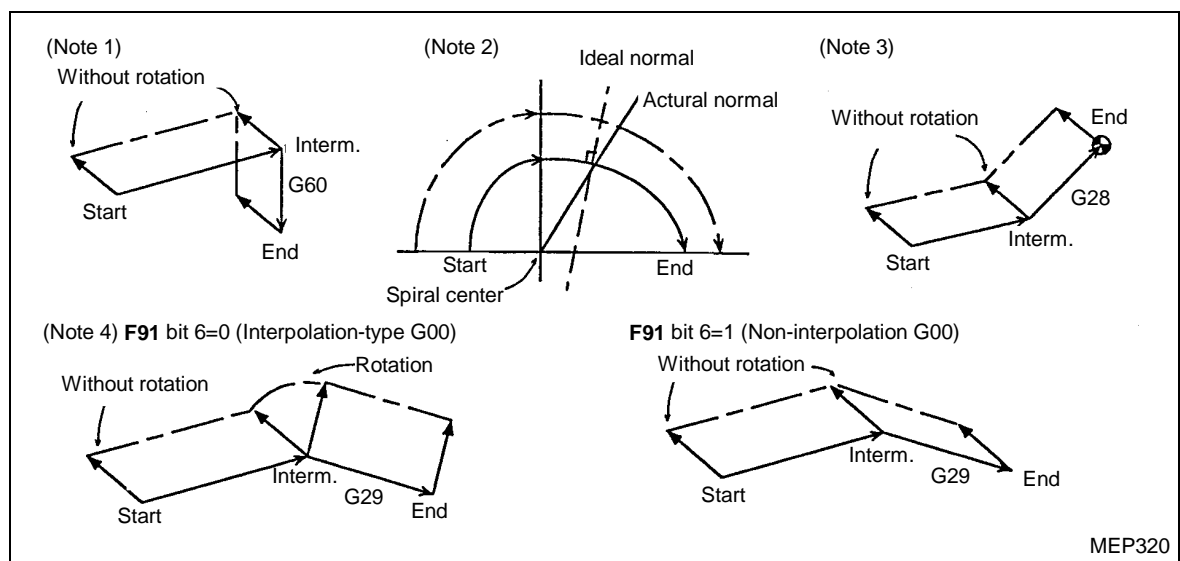
3. The C-axis motion command is ignored in the mode of shaping.
4. The workpiece origin offsetting for the C-axis (G92 Cc) cannot be set in the mode of shaping (G41.1 or G42.1). Setting such a command will only result in alarm **807 ILLEGAL FORMAT**.
5. With the mirror image selected for the X- or Y-axis, the direction of the C-axis rotation is reversed.



6. The indication for the C-axis under **BUFFER** on the **POSITION** display refers to an absolute value.
7. For the connection between blocks, the **BUFFER** area on the **POSITION** display indicates the angle of the C-axis rotation in addition to the distance of the X- and Y-axial movement.
8. The setting in bit 4 of parameter **F85** (rotational axis feed rate $\times 1/10$) is ignored in the mode of the shaping for inch system.

12-8-5 Compatibility with the other functions

Function	Description
One-way positioning	Shaping control is suppressed. (Note 1)
Helical interpolation	Shaping is realized adequately.
Spiral interpolation	Shaping cannot be realized correctly since the starting and ending point do not lie on one and the same circumference. (Note 2)
Synchronous feed	The designated feed rate cannot be obtained since the work spindle is controlled as the C-axis.
Shape correction	Shaping cannot be realized correctly since the control for constant acceleration and deceleration is not applicable to the rotation on the C-axis.
High-speed machining	Alarm 807 ILLEGAL FORMAT will be caused.
Exact-stop check	Deceleration and stop do not occur for the rotation on the C-axis.
Error detection	Deceleration and stop do not occur for the rotation on the C-axis.
Overriding	Overriding is applied adequately to the rotation on the C-axis.
Figure rotation	Shaping control is performed for the rotated figure.
Coordinates system rotation	Shaping control is performed for the rotated figure.
Scaling	Shaping control is performed for the scaled figure.
Mirror image	Shaping control is performed for the mirrored figure.
Linear angle command	Shaping control is performed for the calculated connection between linear segments.
Return to reference point	Shaping control is suppressed. (Note 3)
Return to starting point	Shaping control is suppressed for the movement to the intermediate point, indeed, but it is performed for the movement from the intermediate point to the programmed position if the interpolation-type rapid traverse (G00) is selected [F91 bit 6 = 0]. (Note 4)
Workpiece coordinate system setting	The rotation on the C-axis is performed with reference to the coordinate system established in the shaping mode.
Local coordinate system setting	The rotation on the C-axis is performed with reference to the coordinate system established in the shaping mode.
Dry run	The speed of C-axis rotation is also modified by the external signal.
Modal restart	Restart from a block in the shaping mode can be performed with adequate control of the C-axis.
Non-modal restart	Restart from the midst of the shaping mode is only performed without C-axis control since the modal information before the restart block is ignored.
Tool path check (plane)	The rotation on the C-axis cannot be displayed.
Tool path check (solid)	The rotation on the C-axis cannot be displayed.



12-8-6 Sample program

Main program
WNo. 1000

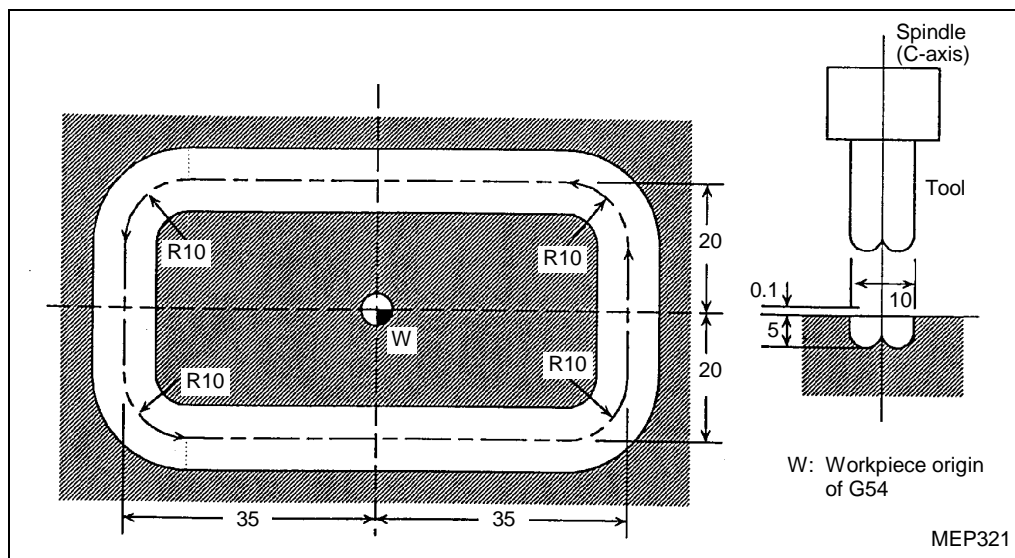
```
O1000
G91G28 X0 Y0 Z0
M193
G28 C0
G90 G92 G53 X0 Y0 Z0
G00 G54 G43 X35.Y0.Z100.H1
G00 Z3.
G01 Z0.1 F3000
G42.1
M98 P1001 L510
M98 P1002 L2
G91 G01 Y10.Z0.05
G40.1
G90 G00 Z100.
G28 X0 Y0 Z0
G00 C0
M194
M30
%
```

Subprogram
WNo. 1001

```
O1001
G17 G91 G01 Y20.,R10.Z-0.01
X-70.,R10.
Y-40.,R10.
X70.,R10.
Y20.
M99
%
```

WNo. 1002

```
O1002
G17 G91 G01 Y20.,R10.
X-70.,R10.
Y-40.,R10.
X70.,R10.
Y20.
M99
%
```



- NOTE -

13 AUXILIARY FUNCTIONS FOR PROGRAMMING

13-1 Fixed-Cycle Functions

13-1-1 Function description

The fixed-cycle functions allow positioning, hole-drilling, boring, tapping, or other machining programs to be executed according to the predetermined job sequence by the commands of a single block. The available job sequences for machining are listed below.

The fixed-cycle function mode is cancelled on reception of G80 or a G-command (G00, G01, G02, G03, G2.1, or G3.1) of command group G01. All related types of data are also cleared to zero at the same time.

13-1-2 List of fixed-cycle functions

Table 13-1 List of fixed-cycle functions

G-Code	Description	Arguments	Notes
G71.1	Chamfering cutter (CW)	[X, Y] Z, Q, R, F [P, D]	
G72.1	Chamfering cutter (CCW)	[X, Y] Z, Q, R, F [P, D]	
G73	High-speed deep-hole drilling	[X, Y] Z, Q, R, F [P, D, K, I, J(B)]	
G74	Reverse tapping	[X, Y] Z, R, F [P, D, J(B), H]	Dwell in seconds
G75	Boring	[X, Y] Z, R, F [Q, P, D, K, I, J(B)]	
G76	Boring	[X, Y] Z, R, F [Q, P, D, J(B)]	
G77	Back spot facing	[X, Y] Z, R, F [Q, P, E, J(B)]	Return mode is for return to initial point only.
G78	Boring	[X, Y] Z, R, F [Q, P, D, K]	
G79	Boring	[X, Y] Z, R, F [Q, P, D, K, E]	
G81	Spot drilling	[X, Y] Z, R, F	
G82	Drilling	[X, Y] Z, R, F [P, D, I, J(B)]	
G83	Deep-hole drilling	[X, Y] Z, Q, R, F [P, D, K, I, J(B)]	
G84	Tapping	[X, Y] Z, R, F [P, D, J(B), H]	Dwell in seconds
G85	Reaming	[X, Y] Z, R, F [P, D, E]	
G86	Boring	[X, Y] Z, R, F [P]	
G87	Back boring	[X, Y] Z, R, F [Q, P, D, J(B)]	Return mode is for return to initial point only.
G88	Boring	[X, Y] Z, R, F [P]	
G89	Boring	[X, Y] Z, R, F [P]	

Note 1: Omission of the arguments enclosed in brackets ([]) is possible.

Note 2: Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

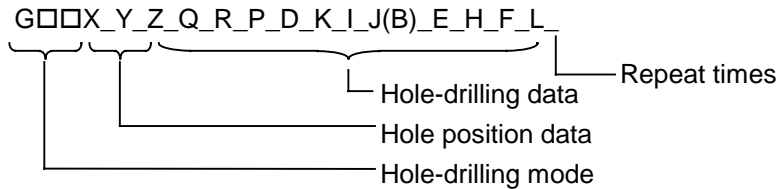
Parameter **F84** bit 1 = 1: Argument J-command

= 0: Argument B-command

13-1-3 Fixed-cycle machining data format

1. Setting fixed-cycle machining data

Set fixed-cycle machining data as follows:



- Hole-drilling mode (G-code)

See the list of fixed cycle functions.

- Hole position data (X, Y)

Set hole positions using incremental or absolute data.

- Hole-drilling data

Z..... Set the distance from point R to the hole bottom using incremental data, or set the position of the hole bottom using absolute data.

Q..... Set this address code using incremental data. (This address code has different uses according to the type of hole-drilling mode selected.)

R..... Set the distance from the initial point of machining to point R using incremental data, or set the position of point R using absolute data.

P..... Set the desired time or the number of spindle revolutions, for dwell at the hole bottom. (Set the overlapping length for the chamfering cutter cycles G71.1 and G72.1.)

D..... Set this address code using incremental data. (This address code has different uses according to the type of hole-drilling mode selected.)

K..... Set this address code using incremental data. (This address code has different uses according to the type of hole-drilling mode selected.)

I..... Set the feed override distance for the tool to be decelerated during the last cutting operation of drilling with a G73, G82, or G83 command code.

J(B)..... For G74 or G84, set the timing of dwell data output; for G75, G76, or G87, set the timing of M3 and M4 output, or; for G73, G82, or G83, set the feed override ratio for deceleration during the last cutting operation.

E..... Set a cutting feed rate (for G77, G79 and G85).

H..... Select synchronous/asynchronous tapping cycle and set the return speed override during a synchronous tapping cycle.

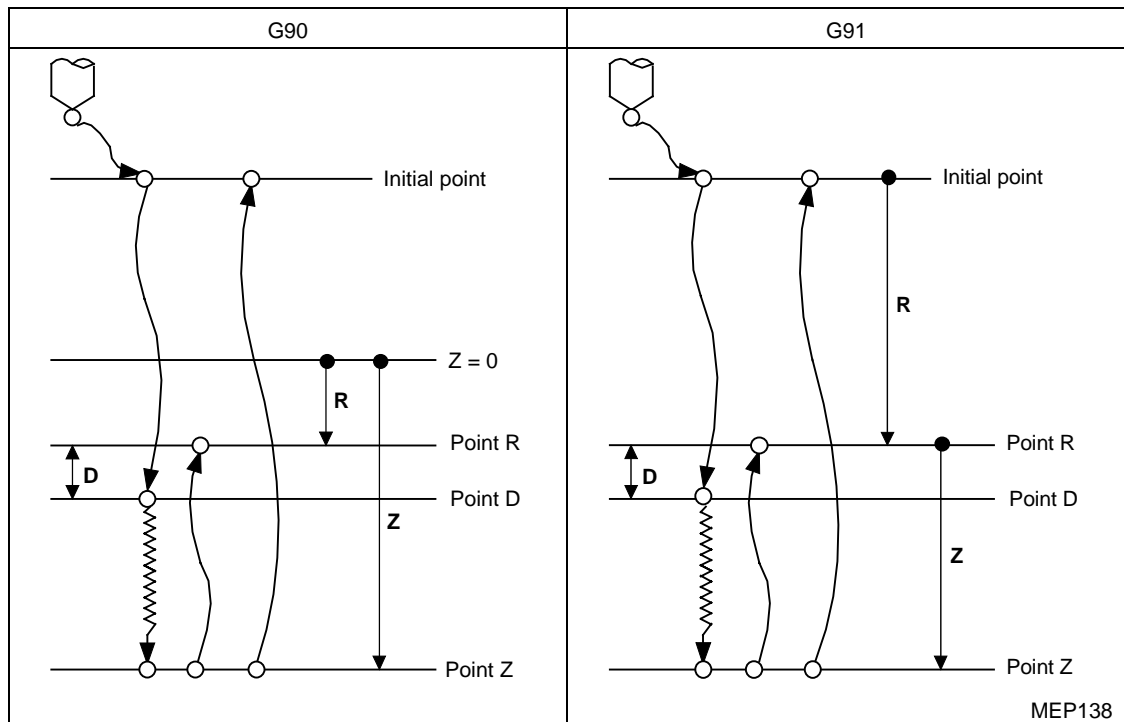
F..... Set a cutting feed rate.

- Repeat times (L)

If no data is set for L, it will be regarded as equal to 1.

If L is set equal to 0, hole-drilling will not occur; hole-drilling data will only be stored into the memory.

- The differences between the G90 mode data setting method and the G91 mode data setting method are shown in the diagram below.



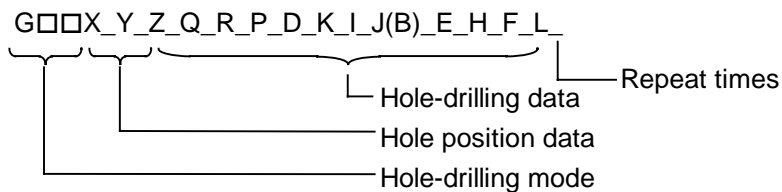
- → : Signifies signed distance data that begins at ●.
- ↔ : Signifies unsigned distance data.

Note 1: The initial point refers to the Z-axis position existing at the moment of the fixed-cycle mode selection.

Note 2: Point D is that at which positioning from point R can be done further at a rapid feed rate.

2. Programming format

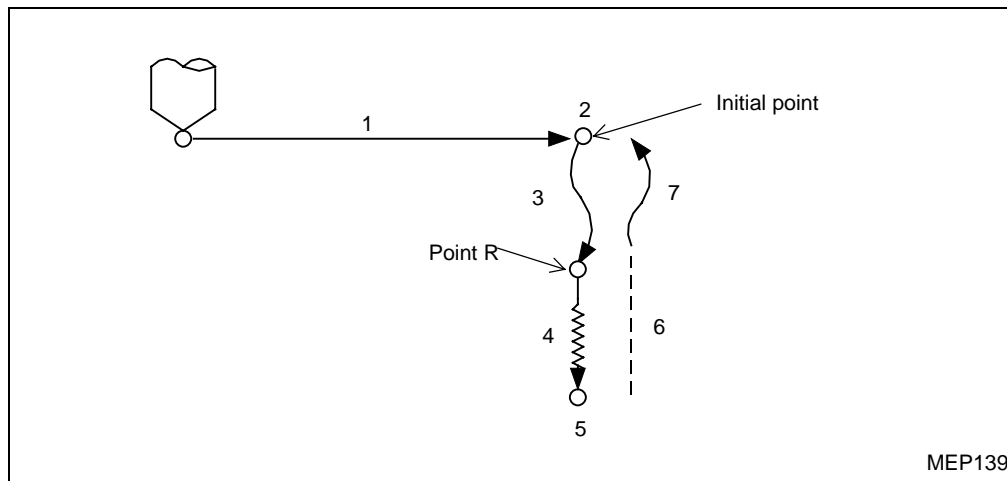
As shown below, the fixed-cycle command consists of a hole-drilling mode section, a hole position data section, a hole-drilling data section, and a repeat instruction section.



3. Detailed description

1. The hole-drilling mode refers to a fixed-cycle mode used for drilling, counterboring, tapping, boring, or other machining operations. Hole position data denotes X-axis and Y-axis positioning data. Hole-drilling data denotes actual machining data. Hole position data and repeat times are unmodal, whereas hole-drilling data is modal.
2. If M00 or M01 is set either in the same block as a fixed-cycle command or during the fixed-cycle mode, then the fixed-cycle command will be ignored and then after positioning, M00 or M01 will be output. The fixed-cycle command will be executed if either X, Y, Z, or R is set.

3. During fixed-cycle operation, the machine acts in one of the following seven types of manner:
- Action 1 For positioning of the X-axis and the Y-axis, the machine acts according to the current G code of command group 01 (if the code is G02 or G03, it will be regarded as G01).
 - Action 2 Command M19 will be sent from the NC unit to the machine at the positioning complete point (initial point) if the command is G87. After execution of this M-command, the next action will begin. In the single-block operation mode, positioning will be followed by block stop.



- Action 3 Positioning up to point R occurs at a rapid feed rate.
- Action 4 Hole-drilling using cutting feed is performed.
- Action 5 Depending on the selected type of fixed-cycle mode, either spindle stop (M05), spindle backward rotation (M04), spindle forward rotation (M03), dwell, or tool shift is performed at the hole bottom.
- Action 6 Tool relief to point R is performed at a cutting feed rate or a rapid feed rate, depending on the selected type of fixed-cycle mode.
- Action 7 Return to the initial point is performed at a rapid feed rate.

Whether fixed-cycle mode operation is to be terminated at machine action 6 or machine action 7 can be selected using the following G-commands:

G98: Return to the initial point level

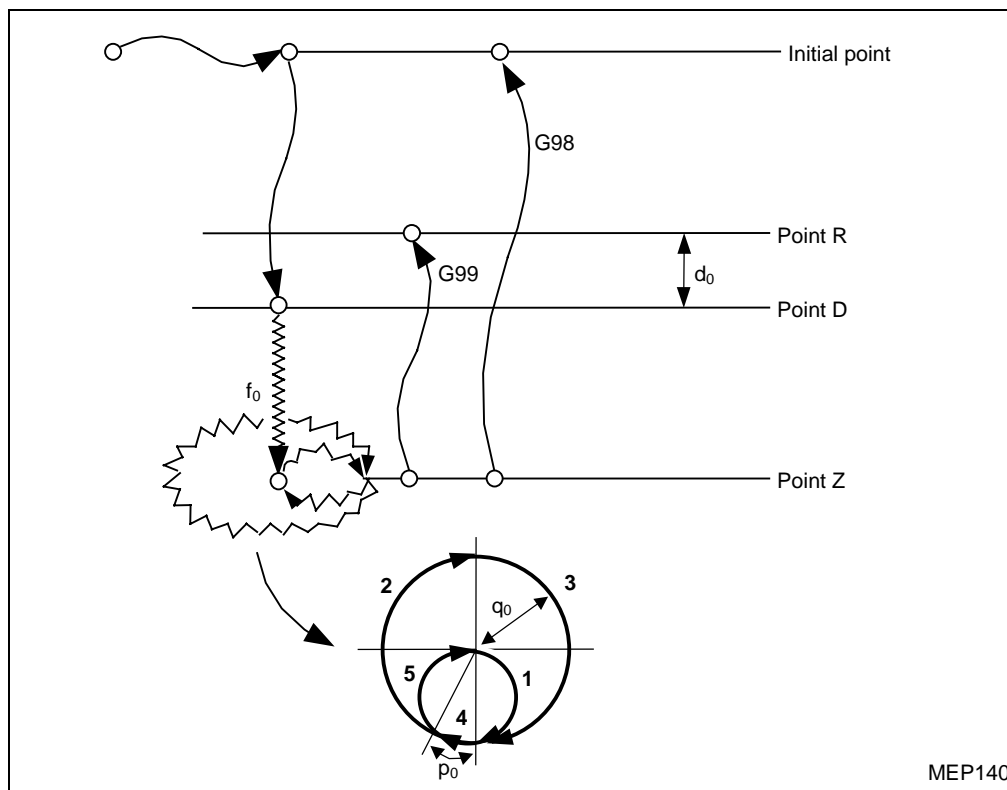
G99: Return to the R-point level

Both two commands are modal. Once G98, for example, has been selected, the G98 mode will therefore remain valid until G99 is selected. The G98 mode will be set when the NC unit is initialized (NC operation ready status).

For a command block without positioning data, fixed cycle operation will not be performed and the hole-drilling data will only be stored into the memory.

13-1-4 G71.1 (Chamfering cutter CW)

G71.1 [Xx Yy] Rr Zz Qq₀ [Pp₀ Dd₀] Ff₀



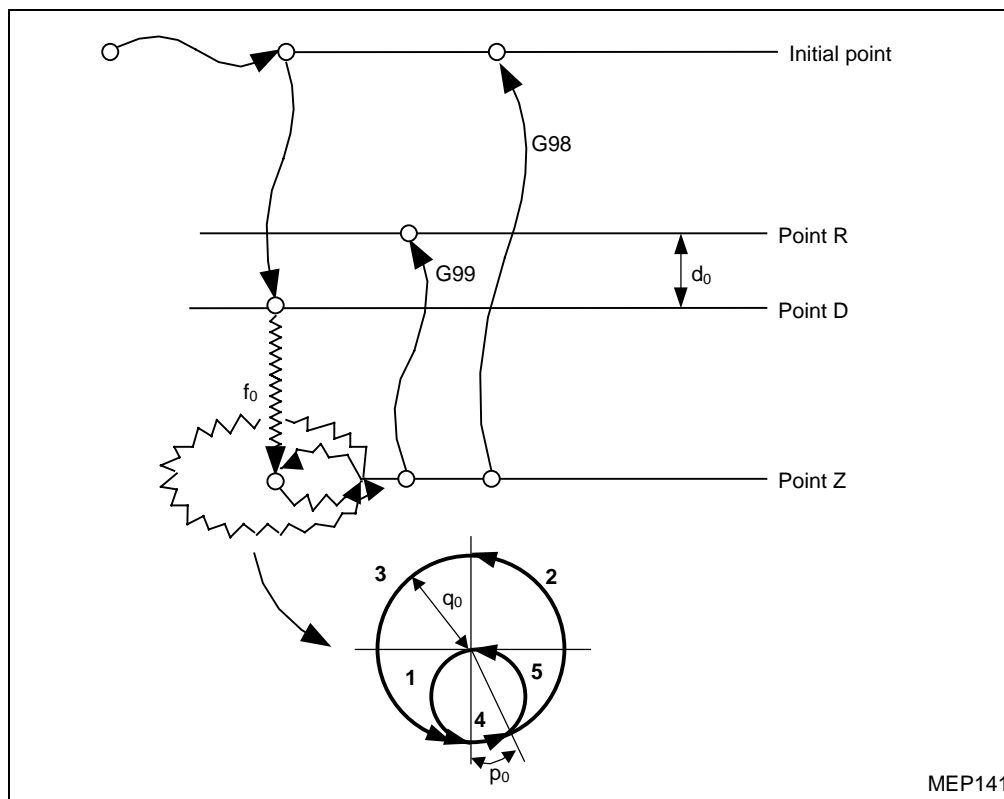
q_0 : Radius

d_0 : Distance from point R

p_0 : Overlapping length (in arc)

f_0 : Feed rate

- Omission of X, Y, P, and/or D is possible.
- Omission of Q or setting of Q equal to 0 results in a program error.

13-1-5 G72.1 (Chamfering cutter CCW)G72.1 [Xx Yy] Rr Zz Qq₀ [Pp₀ Dd₀] Ff₀

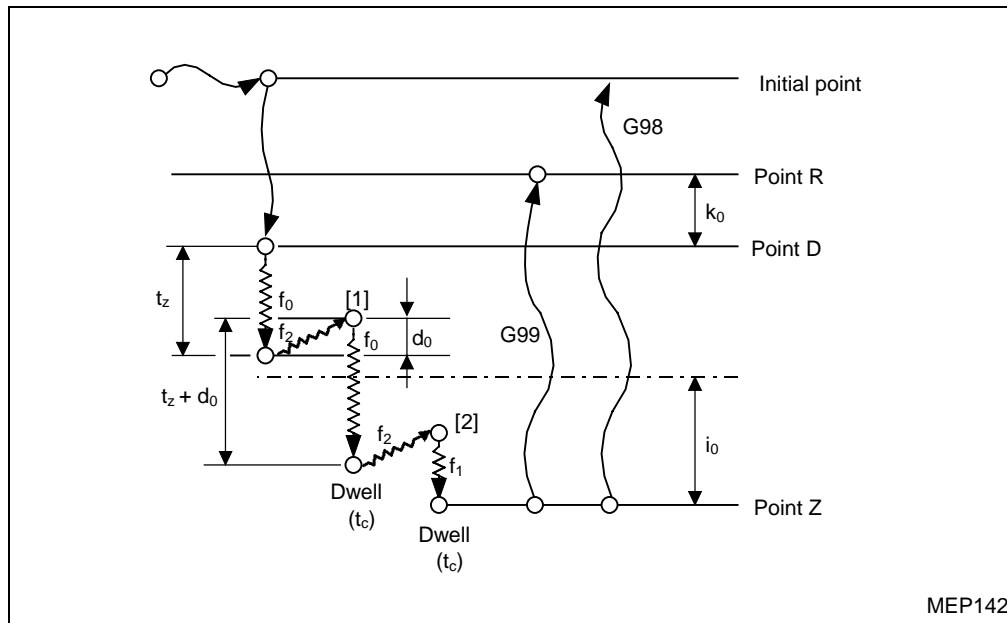
MEP141

 q_0 : Radius d_0 : Distance from point R p_0 : Overlapping length (in arc) f_0 : Feed rate

- Omission of X, Y, P, and/or D is possible.
- Omission of Q or setting of Q equal to 0 results in a program error.

13-1-6 G73 (High-speed deep-hole drilling)

G73 [Xx Yy] Rr Zz Qt_z [Pt_c] Ff₀ [Dd₀ Kk₀ Ii₀ Jj₀(Bb₀)]



t_z : Depth of cut per pass

t_c : Dwell time or number of spindle revolutions

d_0 : Return distance

k_0 : Distance from point R to the starting point of cutting feed

i_0 : Feed override distance

$j_0(b_0)$: Feed override ratio(%)

f_0 : Feed rate

f_1 : Override feed rate ($f_1 = f_0 \times j_0(b_0)/100$)

f_2 : Return speed (fixed)

9999 mm/min (for mm-spec.)

999.9 in./min (for in.-spec.)

- The feed rate will remain unchanged if either I or J(B) is omitted.
- Omission of X, Y, P, D, K, I, and/or J(B) is possible.
If D is omitted or set to 0, the machine will operate according to the value of parameter **F12**.
- An alarm **809 ILLEGAL NUMBER INPUT** will occur if Q is set to 0.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

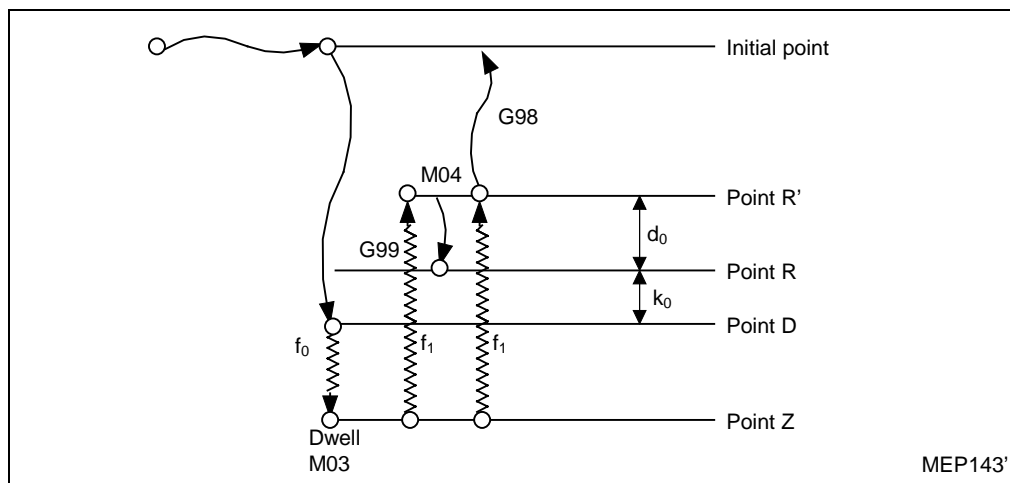
Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- The feed rate is f_1 only if the starting point of a cutting pass is within the range of i_0 .

Example: In the diagram shown above, during the second cutting operation, since pecking return point [1] falls outside the range of feed override distance i_0 , feeding does not decelerate and cutting is performed at feed rate f_0 ; during the third cutting operation, since pecking return point [2] falls within the range of i_0 , feeding decelerates and cutting is performed at feed rate f_1 .

13-1-7 G74 (Reverse tapping)

G74 [Xx Yy] Rr Zz [Pt_c] Ff₀ [Jj₀(Bb₀) Dd₀ Hh₀ Kk₀]



t_c : Dwell time only (in seconds)

f_0 : Feed rate

$j_0(b_0)$: 1: dwell at the bottom of hole before M03 output

2: dwell at the bottom of hole after M03 output

4: dwell at point R before M04 output

d_0 : Distance from point R (Tap lifting distance)

h_0 : Synchronous/asynchronous tapping selection flag and the return speed override (%) during a synchronous tapping cycle

$h_0 = 0$: Asynchronous tapping cycle

$h_0 > 0$: Synchronous tapping cycle

k_0 : Distance from point R

- Omission of X, Y, P, J(B), D, H and/or K is possible.

If, however, J(B) is omitted or set to 0, the setting of J(B) will be regarded as 2.

If H is omitted, the selection of synchronous/asynchronous tapping cycle is performed by the bit 6 of parameter **F94**.

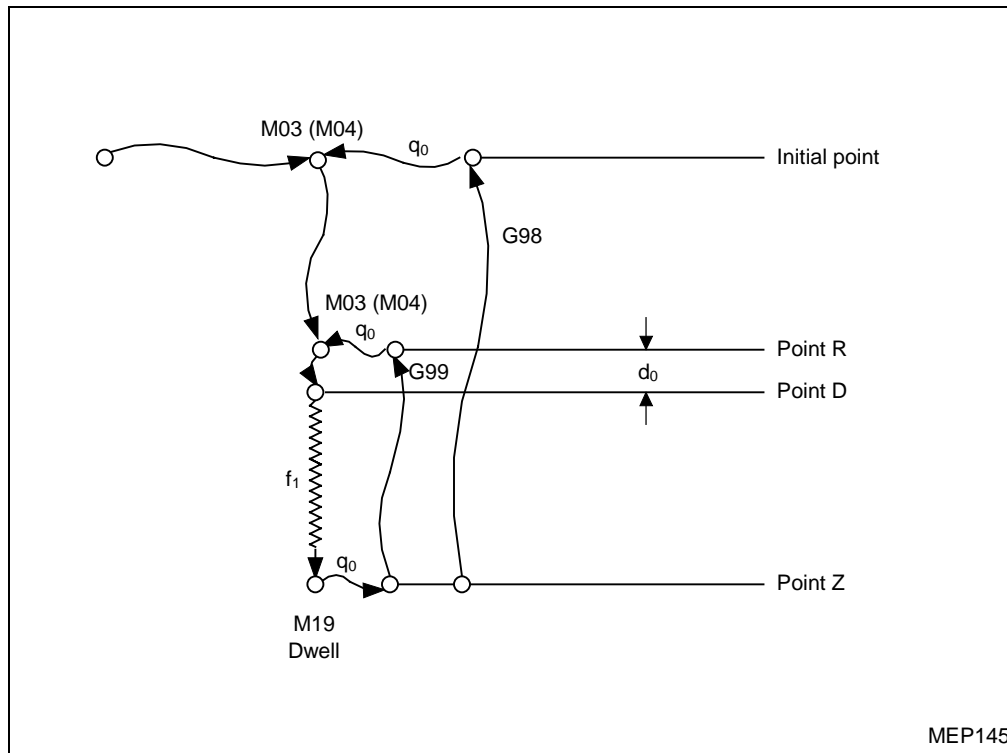
- For synchronous tapping, see section 13-1-22.

- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

13-1-9 G76 (Boring)

G76 [Xx Yy] Rr Zz [Pt_c Qq₀] Ff₁ [Dd₀ Jj₀(Bb₀)]

t_c : Dwell time or number of spindle revolutions

q_0 : The amount of relief on the X-Y plane
(The direction of relief is determined by the settings of bits 3 and 4 of parameter **I14**)

f_1 : Feed rate

d_0 : Distance from point R

$j_0(b_0)$: If 0 or not set, then M03 output after the end of machining.
If other than 0, then M04 output after the end of machining.

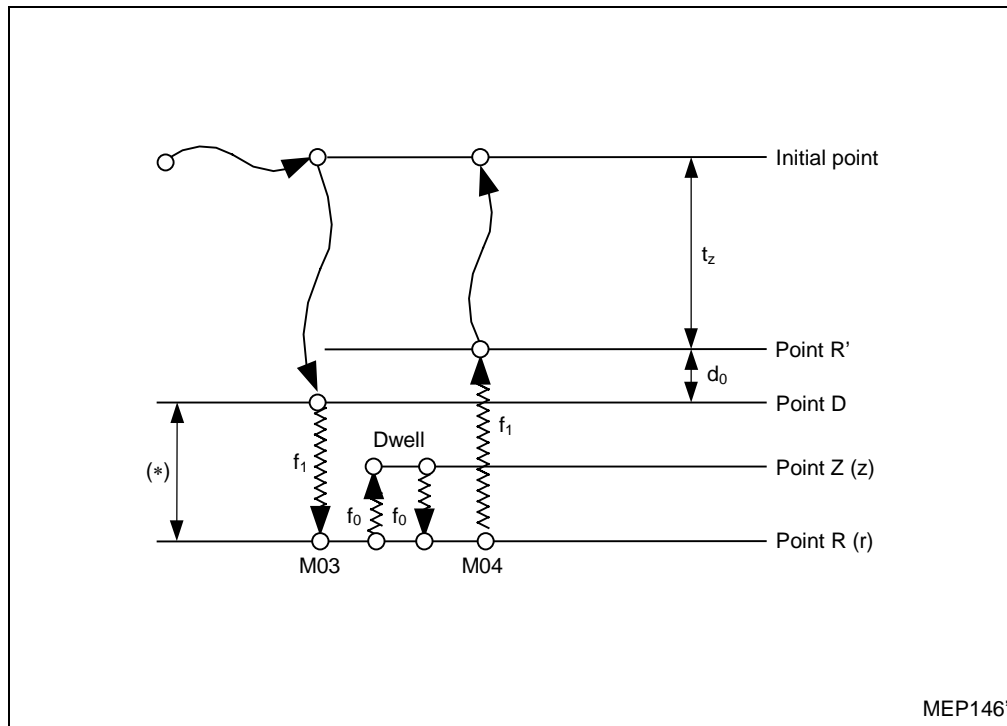
- Omission of X, Y, P, Q, D, and/or J(B) is possible.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

13-1-10 G77 (Back spot facing)

G77 [Xx Yy] Rr Zz [Pt_c Qt_z] Ff₀ [Ef₁ Jj₀(Bb₀) Dd₀]



t_c : Dwell time or number of spindle revolutions

t_z : Distance from the initial point

f_0 : Feed rate 0

f_1 : Feed rate 1

$j_0(b_0)$: If 0, then at the hole bottom, M03 and M04 are output in that order for spindle forward rotation.
If 1, then at the hole bottom, M04 and M03 are output in that order for spindle backward rotation.

d_0 : Distance from point R'

- Normally, asynchronous feed (G94) is used for the pass marked with (*). If $f_1 = 0$, or if f_1 is omitted, however, synchronous feed (G95) is used (feed rate = 0.5 mm/rev).
- Omission of X, Y, P, Q, E, J (B), and/or D is possible.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

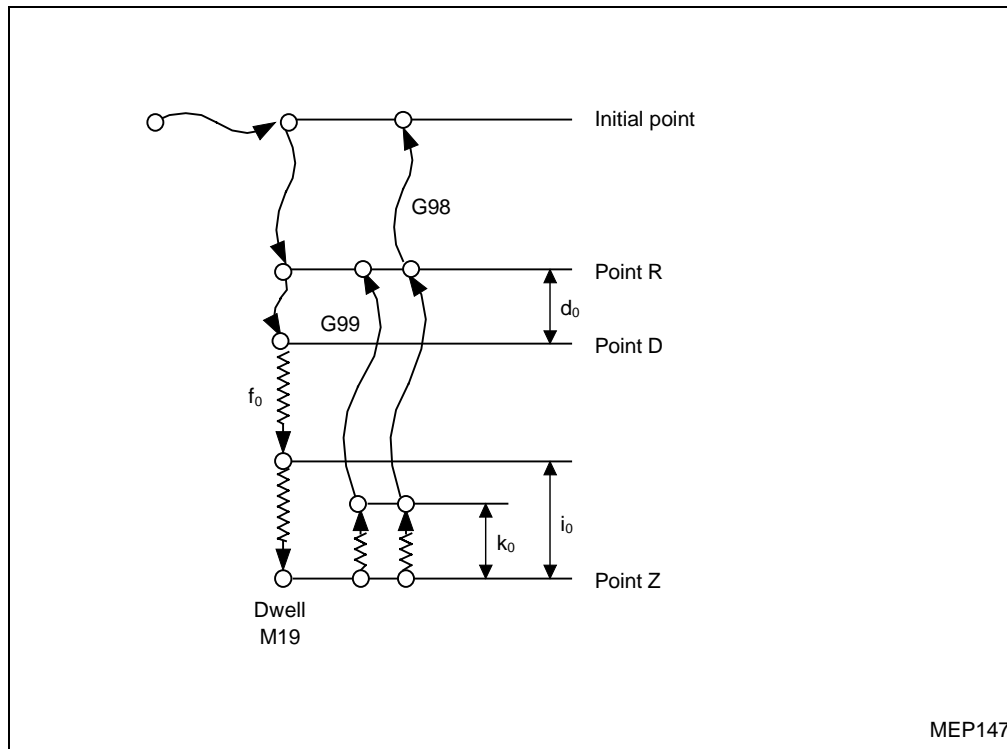
Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- In G91 (incremental data input) mode, the direction of hole machining is automatically determined according to the sign of Z data (the sign of data at address R will be ignored).

13-1-11 G78 (Boring)

G78 [Xx Yy] Rr Zz [Pt_c] Ff₀ [Dd₀ Kk₀ Qi₀]



t_c : Dwell time or number of spindle revolutions

d_0 : Distance from point R

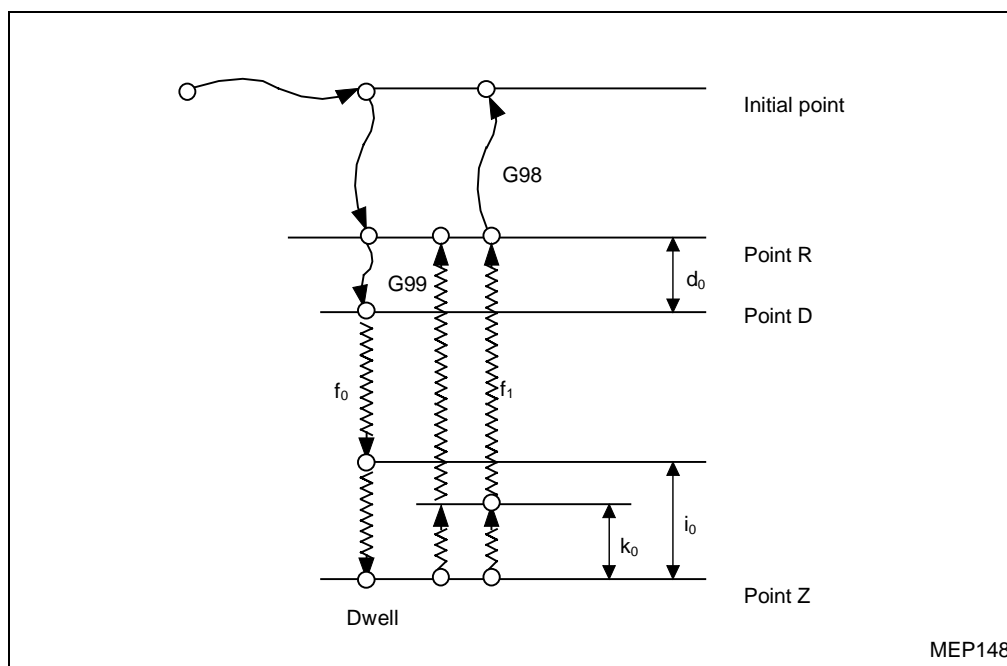
k_0 : Distance from point Z

i_0 : Distance from point Z

- Omission of X, Y, P, D, K, and/or Q is possible.

13-1-12 G79 (Boring)

G79 [Xx Yy] Rr Zz [Pt_c] Ff₀ [Dd₀ Kk₀ Qi₀ Ef₁]



t_c : Dwell time or number of spindle revolutions

f_0 : Feed rate 0

d_0 : Distance from point R

k_0 : Distance from point Z

i_0 : Distance from point Z

f_1 : Feed rate 1

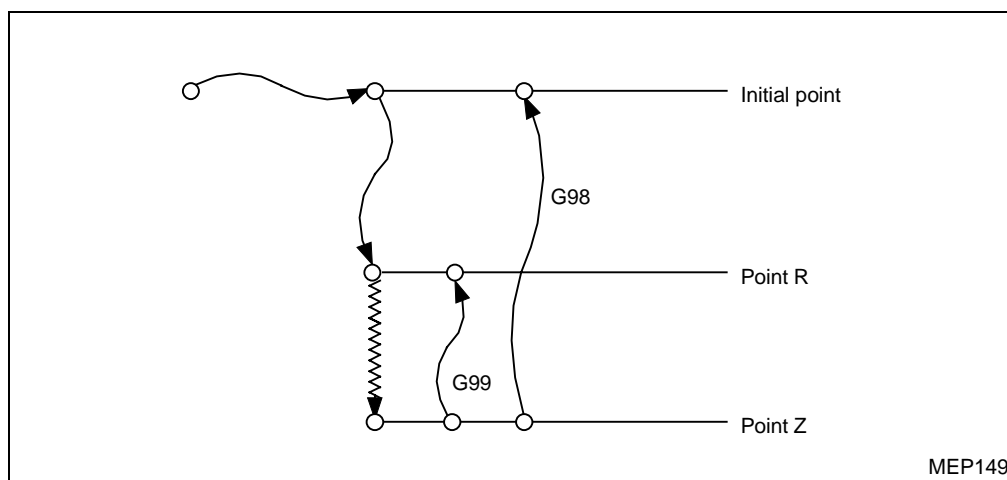
- Asynchronous feed will be used for f_1 .

If, however, f_1 is set equal to 0 or is not set, then feed will use the setting of f_0 .

- Omission of X, Y, P, D, K, Q, and/or E is possible.

13-1-13 G81 (Spot drilling)

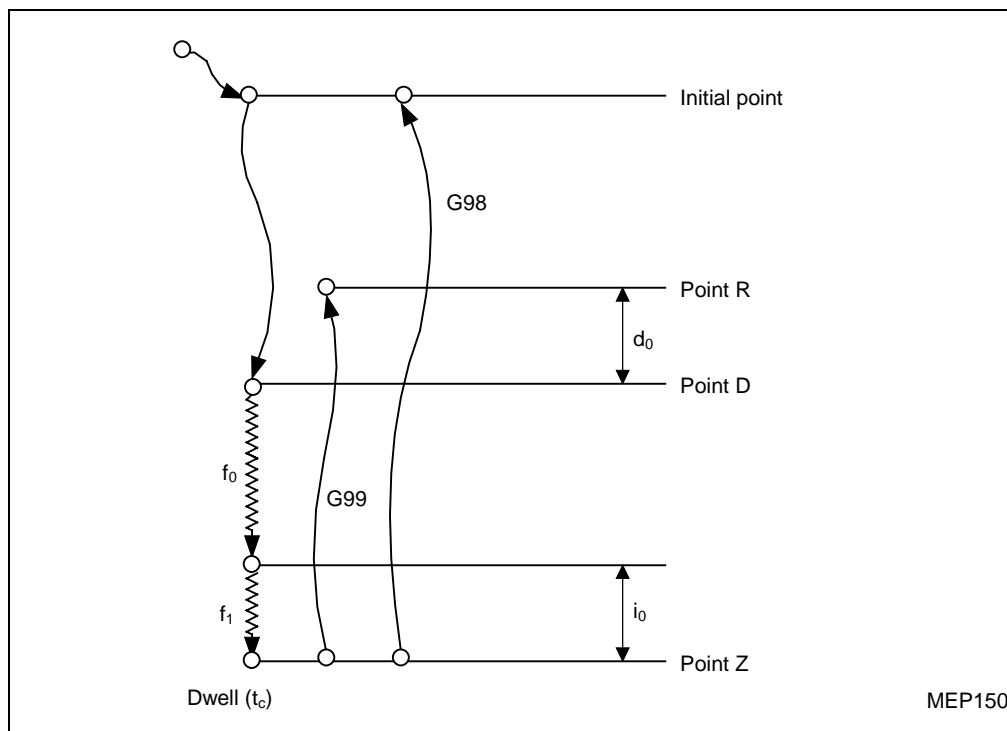
G81 [Xx Yy] Rr Zz



- Omission of X and/or Y is possible.

13-1-14 G82 (Drilling)

G82 [Xx Yy] Rr Zz [Pt_c] Ff₀ [Dd₀ Ii₀ Jj₀(Bb₀)]



t_c : Dwell time or number of spindle revolutions

d_0 : Distance from point R to the starting point of cutting feed

i_0 : Feed override distance

$j_0(b_0)$: Feed override ratio (%)

f_0 : Feed rate

f_1 : Override feed rate ($f_1 = f_0 \times j_0(b_0)/100$)

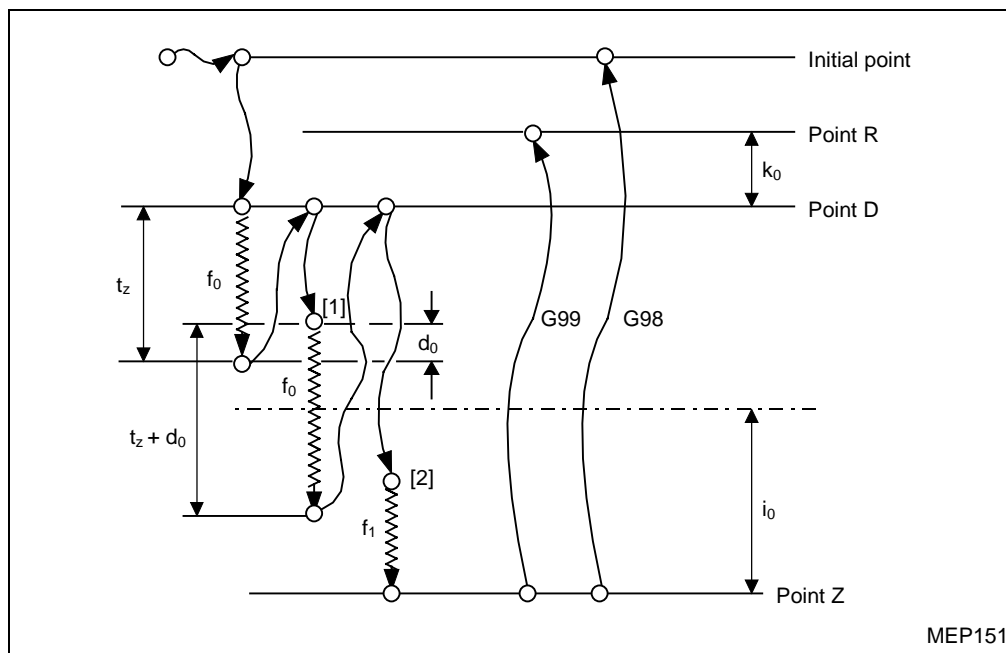
- The feed rate will remain unchanged if either I or J(B) is omitted.
- Omission of X, Y, P, D, I, and/or J(B) is possible.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

Parameter **F84** bit 1 = 1 : Argument J-command
= 0 : Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

13-1-15 G83 (Deep-hole drilling)

G83 [Xx Yy] Rr Zz Qt_z Ff₀ [Dd₀ Kk₀ Ii₀ Jj₀(Bb₀)]



t_z : Depth of cut per pass

d_0 : Rapid feed stop allowance

k_0 : Distance from point R to the starting point of cutting feed

i_0 : Feed override distance

$j_0(b_0)$: Feed override ratio (%)

f_0 : Feed rate

f_1 : Override feed rate ($f_1 = f_0 \times j_0(b_0)/100$)

- The feed rate will remain unchanged if either I or J(B) is omitted.
- Omission of X, Y, D, K, I, and/or J(B) is possible.
If D is omitted or set to 0, the machine will operate according to the value of parameter **F13**.
- An alarm **809 ILLEGAL NUMBER INPUT** will occur if Q is set to 0.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

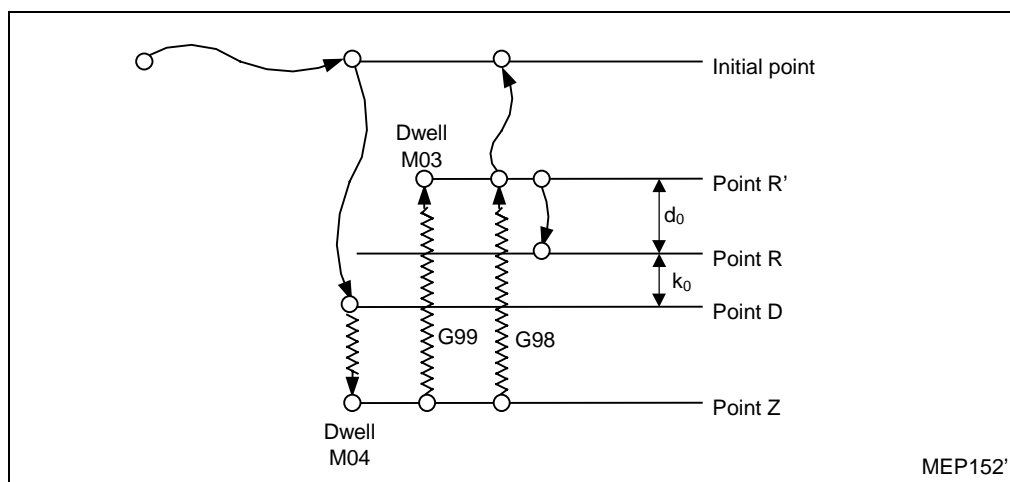
Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- The feed rate is f_1 only if the starting point of a cutting pass is within the range of i_0 .

Example: In the diagram shown above, during the second cutting operation, since rapid feed positioning point [1] falls outside the range of feed override distance i_0 , feeding does not decelerate and cutting is performed at feed rate f_0 ; during the third cutting operation, since rapid feed positioning point [2] falls within the range of i_0 , feeding decelerates and cutting is performed at feed rate f_1 .

13-1-16 G84 (Tapping)

G84 [Xx Yy] Rr Zz [Pt_c] Ff₀ [Jj₀(Bb₀) Dd₀ Hh₀ Kk₀]



t_c : Dwell time only (in seconds)

 f_0 : Feed rate

$j_0(b_0)$:1: dwell at the bottom of hole
before M04 output

2: dwell at the bottom of hole
after M04 output

4: dwell at point R before M03
output

d_0 : Distance from point R (Tap lifting distance)

h_0 : Synchronous/asynchronous tapping selection flag and the return speed override (%) during a synchronous tapping cycle

$h_0 = 0$: Asynchronous tapping cycle

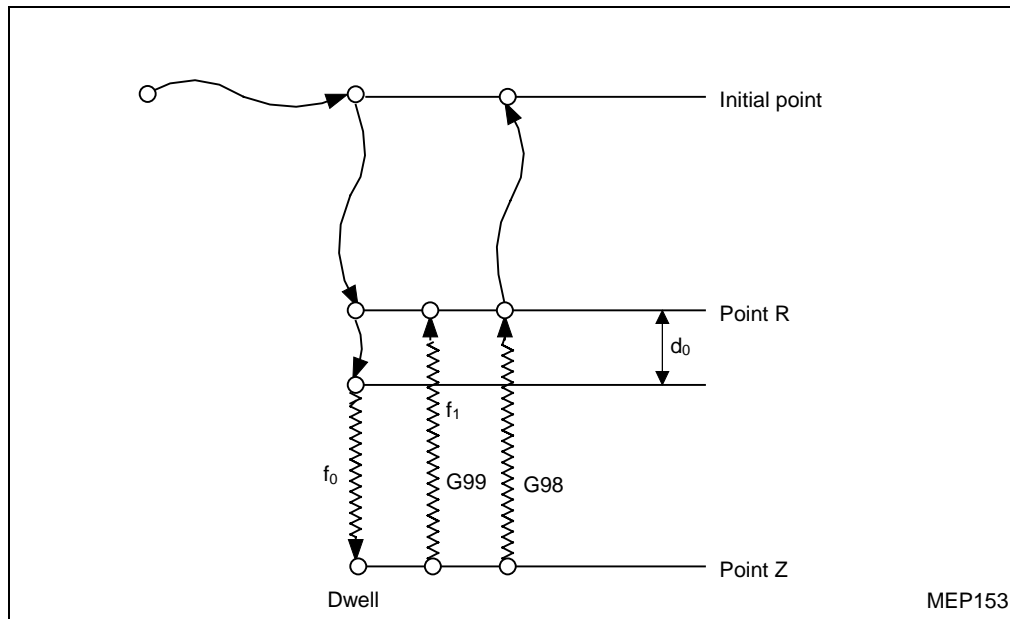
$h_0 > 0$: Synchronous tapping cycle

k_0 : Distance from point R

- Omission of X, Y, P, J(B), D, H and/or K is possible.
If, however, J(B) is omitted or set to 0, the setting of J(B) will be regarded as 2.
If H is omitted, the selection of synchronous/asynchronous tapping cycle is performed by the bit 6 of parameter **F94**.
- For synchronous tapping, see section 13-1-22.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

Parameter **F84** bit 1 = 1 : Argument J-command
= 0 : Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

13-1-17 G85 (Reaming)G85 [Xx Yy] Rr Zz [Pt_z] Ff₀ [Ef₁ Dd₀]

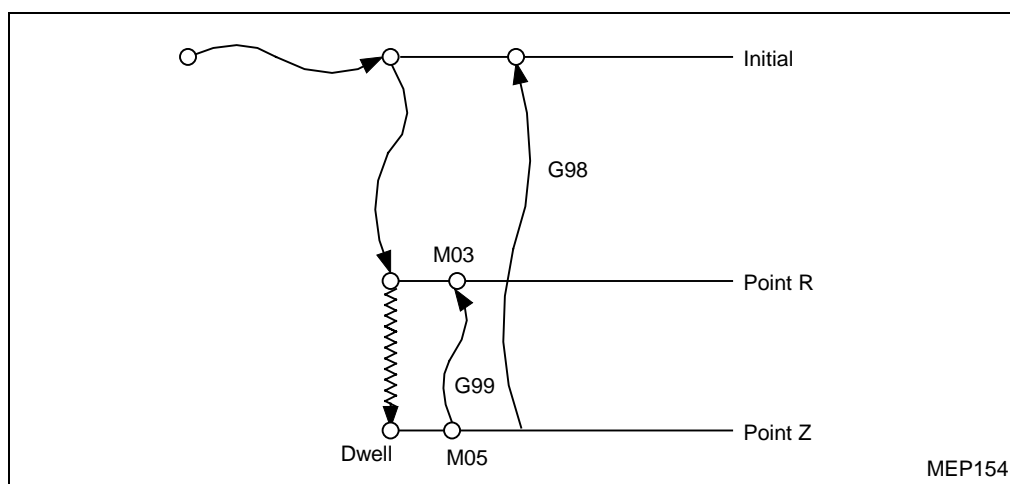
t_z : Dwell time or number of spindle revolutions

f_1 : Feed rate 1

f_0 : Feed rate 0

d_0 : Distance from point R

- Asynchronous feed will be used for f_1 .
If, however, f_1 is set equal to 0 or is not set, then feed will use the setting of f_0 .
- Omission of X, Y, P, E, and/or D is possible.

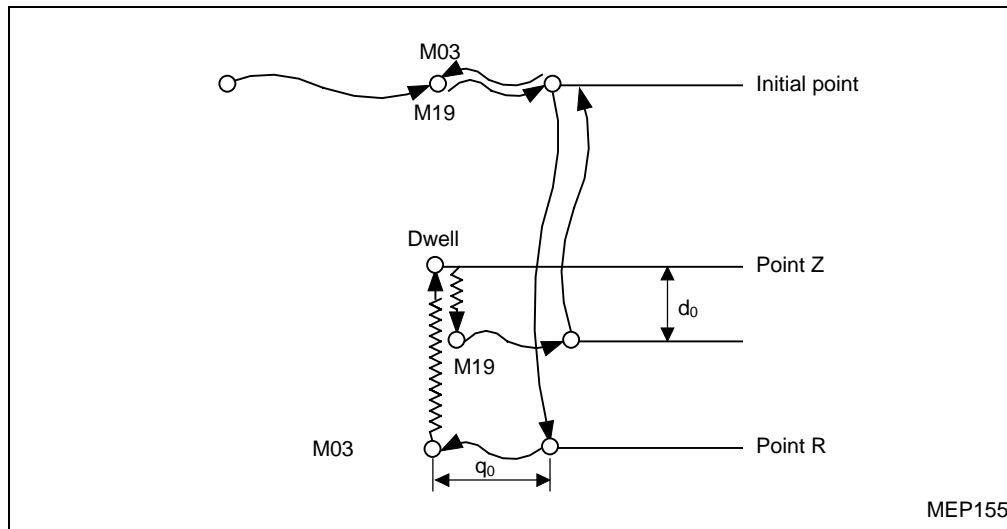
13-1-18 G86 (Boring)G86 [Xx Yy] Rr Zz [Pt_c]

t_c : Dwell time or number of spindle revolutions

- Omission of X, Y, and/or P is possible.

13-1-19 G87 (Back boring)

G87 [Xx Yy] Rr Zz [Pt_c Qq₀] Ff₀ [Dd₀ Jj₀/(Bb₀)]



t_c : Dwell time or number of spindle revolutions

q_0 : The amount of relief on the X-Y plane
(The direction of relief is determined by the settings of bits 3 and 4 of parameter **I14**)

f_0 : Feed rate

d_0 : Distance from point Z

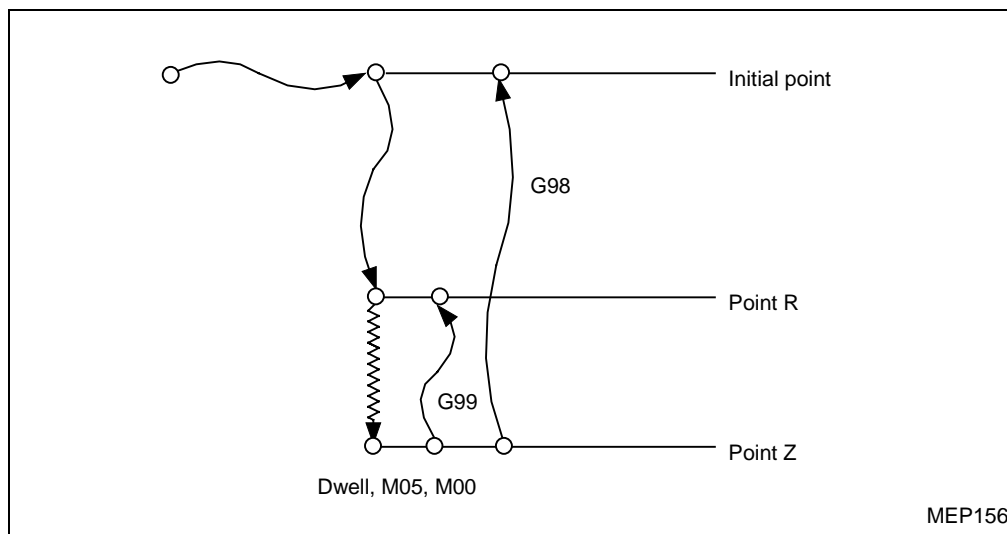
$j_0(b_0)$: If 0 or not set, then M03 output at point R.
If other than 0, then M04 output at point R.

- Omission of X, Y, P, Q, D, and/or J(B) is possible.
- For G87, initial-point return is used, irrespective of the type of return mode (G98 or G99).
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

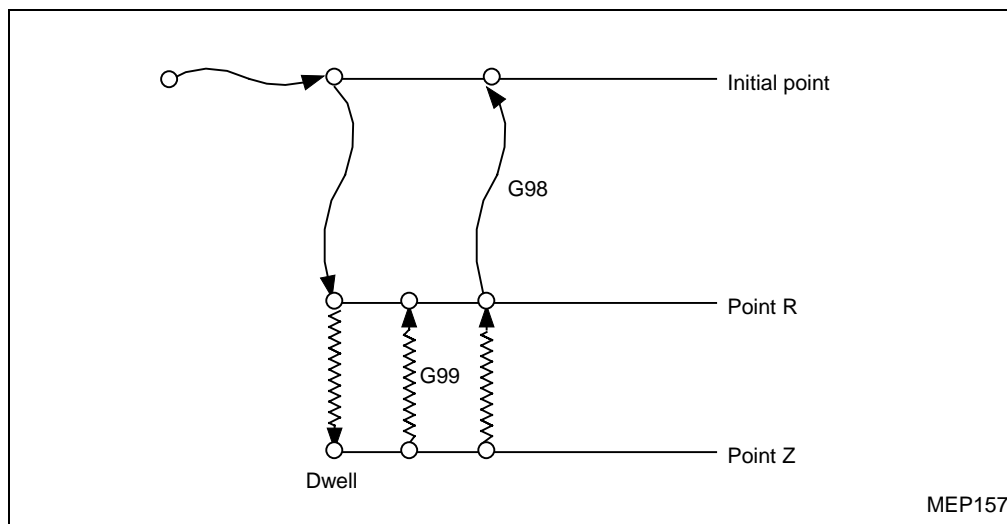
Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- In G91 (incremental data input) mode, the direction of hole machining is automatically determined according to the sign of Z data (the sign of data at address R will be ignored).

13-1-20 G88 (Boring)G88 [Xx Yy] Rr Zz [Pt_c] t_c : Dwell time or number of spindle revolutions

- Omission of X, Y, and/or P is possible.
- At the hole bottom, M00 and M05 are output.

13-1-21 G89 (Boring)G89 [Xx Yy] Rr Zz [Pt_c] t_c : Dwell time or number of spindle revolutions

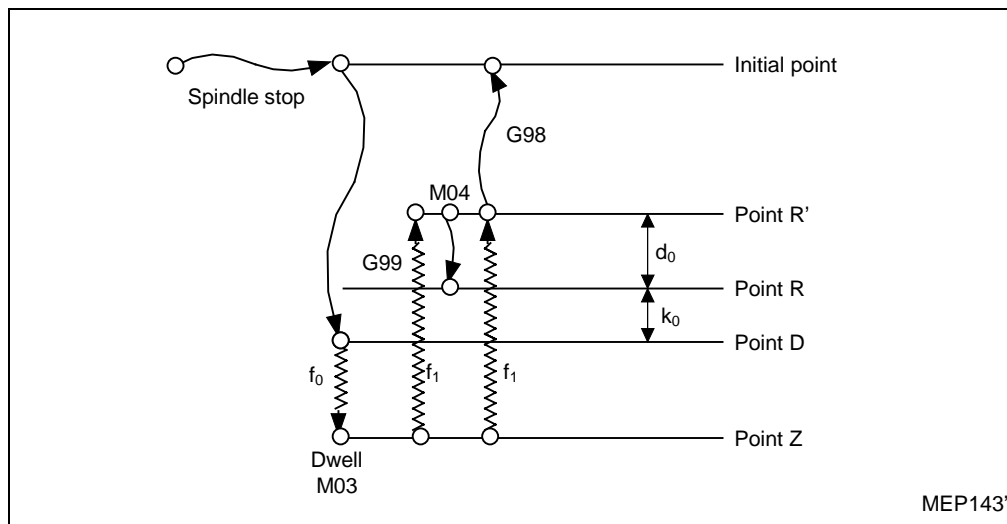
- Omission of X, Y, and/or P is possible.

13-1-22 Synchronous tapping (Option)

In an EIA/ISO program, synchronous tapping can be selected by additionally setting data at the address H in the tapping cycle block of G74 or G84. Address H is used to select a synchronous/asynchronous tapping and to designate the override of return speed. Special preparatory functions G84.2 and G84.3 are also provided for both types of synchronous tapping.

1. G74 (Reverse tapping)

G74 [Xx Yy] Rr Zz [Pt_c] Ff₀ [Jj₀(Bb₀) Dd₀ Hh₀ Kk₀]



t_c : Dwell time only (in seconds)
 f_0 : Feed rate (For synchronous tapping cycle operation, however, set the pitch)
 $j_0(b_0)$: 1:dwell at the bottom of hole before M03 output
 2:dwell at the bottom of hole after M03 output
 4:dwell at point R before M04 output

d_0 : Distance from point R (Tap lifting distance)
 h_0 : Return speed override (%)
 $h_0 = 0$: Asynchronous tapping cycle
 $h_0 \geq 1$: Synchronous tapping cycle
 k_0 : Distance from point R

- Omission of X, Y, P, J(B), D, H, and/or K is possible.

If, however, J(B) is omitted or set to 0, the setting of J(B) will be regarded as 2.

If H is omitted, the selection of synchronous/asynchronous tapping cycle is performed by the bit 6 of parameter **F94**.

- H is used to select whether synchronous tapping cycle operation or asynchronous tapping cycle operation is to be performed using a machine capable of synchronous tapping. This code is also used to override the return speed for synchronous tapping cycle operation. H becomes invalid for a machine not capable of synchronous tapping, or if your machine has synchronous tapping function but bit 6 of parameter **F94** is not set to 1.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

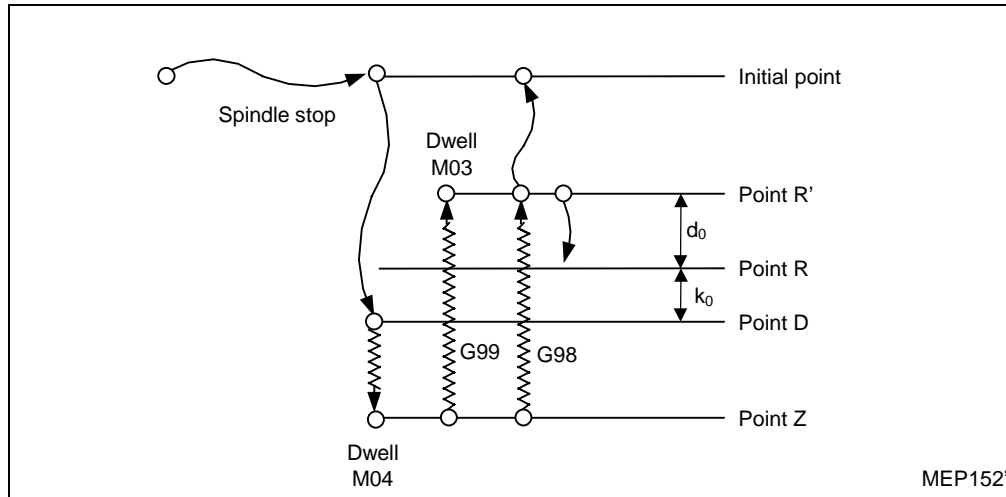
Parameter **F84** bit 1 = 1: Argument J-command
 = 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- During gear selection for tapping, due consideration must be given to ensure the minimum spindle acceleration/deceleration time. Refer to the machine-operating manual.

2. G84 (Normal tapping)

G84 [Xx Yy] Rr Zz [Pt_c] Ff₀ [Jj₀(Bb₀) Dd₀ Hh₀ Kk₀]



t_c :	Dwell time only (in seconds)	d_0 :	Distance from point R (Tap lifting distance)
f_0 :	Feed rate (For synchronous tapping cycle operation, however, set the pitch)	h_0 :	Return speed override (%)
$j_0(b_0)$:	1: dwell at the bottom of hole before M04 output	$h_0 = 0$:	Asynchronous tapping cycle
	2: dwell at the bottom of hole after M04 output	$h_0 \geq 1$:	Synchronous tapping cycle
	4: dwell at point R before M03 output	k_0 :	Distance from point R

- Omission of X, Y, P, J(B), D, H, and/or K is possible.
If, however, J(B) is omitted or set to 0, the setting of J(B) will be regarded as 2.
If H is omitted, the selection of synchronous/asynchronous tapping cycle is performed by the bit 6 of parameter **F94**.
- H is used to select whether synchronous tapping cycle operation or asynchronous tapping cycle operation is to be performed using a machine capable of synchronous tapping. This code is also used to override the return speed for synchronous tapping cycle operation. H becomes invalid for a machine not capable of synchronous tapping, or if your machine has synchronous tapping function but bit 6 of parameter **F94** is not set to 1.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

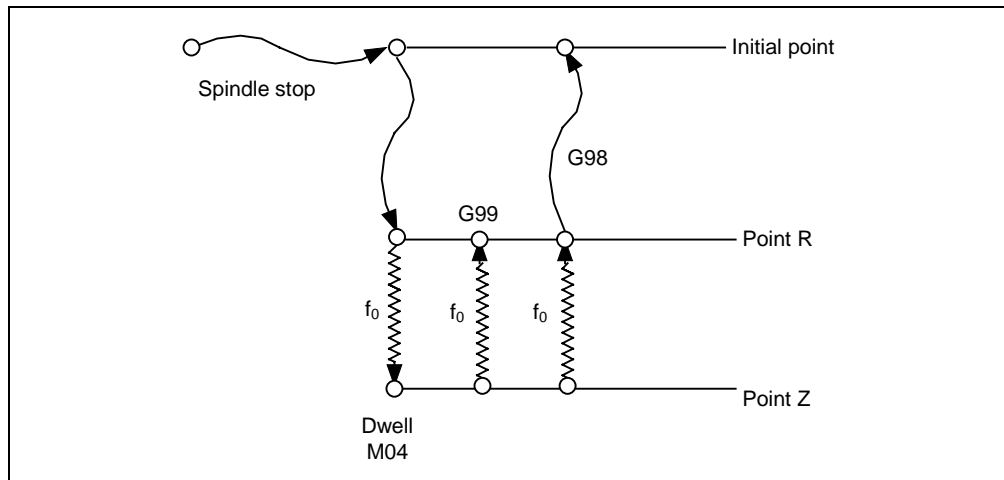
Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- During gear selection for tapping, due consideration must be given to ensure the minimum spindle acceleration/deceleration time. Refer to the machine-operating manual.

3. G84.2 (Normal tapping)

G84.2 [Xx Yy] Rr Zz [Pt_c] Ff₀



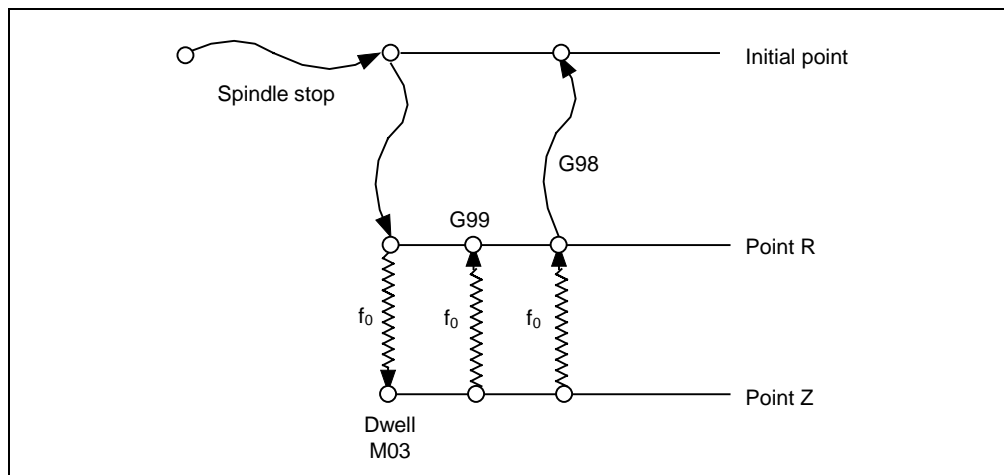
t_c : Dwell time (in seconds) at point Z and upon return to point R

f_0 : Feed rate (in pitch)

- Omission of X, Y, and/or P is possible.
- The codes G84.2 and G84.3 always perform a synchronous tapping, irrespective of the setting in bit 6 of parameter **F94**.
- Designation of G84.2 or G84.3 without the corresponding option would cause the alarm **No. 952 NO SYNCHRONIZED TAP OPTION**.
- During gear selection for tapping, due consideration must be given to ensure the minimum spindle acceleration/deceleration time. Refer to the machine-operating manual.
- The value of parameter **K90** is always referred to as the return speed override (%).

4. G84.3 (Reverse tapping)

G84.3 [Xx Yy] Rr Zz [Pt_c] Ff₀



t_c : Dwell time (in seconds) at point Z and upon return to point R

f_0 : Feed rate (in pitch)

- Omission of X, Y, and/or P is possible.
- The codes G84.2 and G84.3 always perform a synchronous tapping, irrespective of the setting in bit 6 of parameter **F94**.
- Designation of G84.2 or G84.3 without the corresponding option would cause the alarm **No. 952 NO SYNCHRONIZED TAP OPTION**.
- During gear selection for tapping, due consideration must be given to ensure the minimum spindle acceleration/deceleration time. Refer to the machine-operating manual.
- The value of parameter **K90** is always referred to as the return speed override (%).

13-1-23 Tornado cycle (Option)

1. Overview

The optional Tornado cycle is provided for two types of tool: Use an end-mill for boring (Tornado boring cycle) and a special tool for tapping (Tornado tapping cycle).

Both Tornado cycles can be executed with a single tool, while usual boring and tapping cycles require multiple tools.

Note: The Tornado cycle program is to be called up by a “macro-call” G-code in accordance with the following parameters:

J37: ID-No. of the Tornado cycle program 100009401 (fixed)

J38: No. of the calling G-code 130 (fixed)

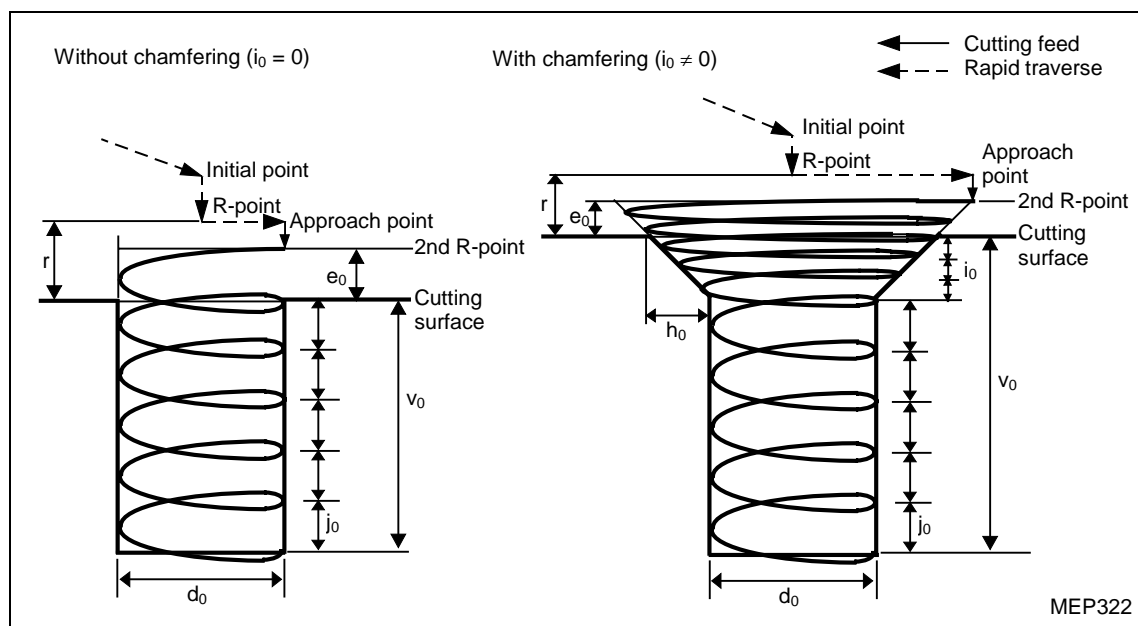
J39: Type of the call 2 (fixed)

2. Programming format

G13₀ Rr Zz Dd₀ Tt₀ Vv₀ Ff₀ Hh₀ Ii₀ Jj₀ Kk₀ Qq₀ Ee₀

Xx Yy

G67



d₀ : Hole diameter

t₀ : Tool diameter

v₀ : Hole depth

f₀ : Feed rate

h₀ : Chamfering amount

i₀ : Pitch 1

j₀ : Pitch 2

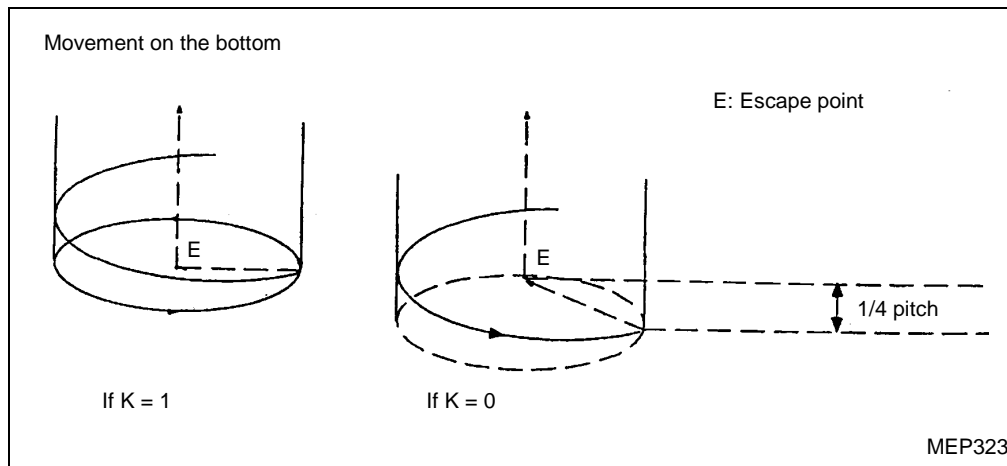
k₀ : Bottom finishing (0: No, 1: Yes, Others: Yes)

q₀ : Machining direction (0: CW, 1: CCW)

e₀ : Position of 2nd R-point

- The chamfering angle is fixed at 45°.
- Set the hole position (Xx, Yy) separately from the macro-call G-code (G130).
- As is the case with usual fixed cycles, actual machining with the Z-axial movement can only be executed for a block containing the hole position data.
- Do not fail to set the code G67 as required to cancel the modal call.

- Use address K to select whether finishing is to be or not to be executed on the bottom of the hole.



K = 1 (With finishing on the hole bottom)

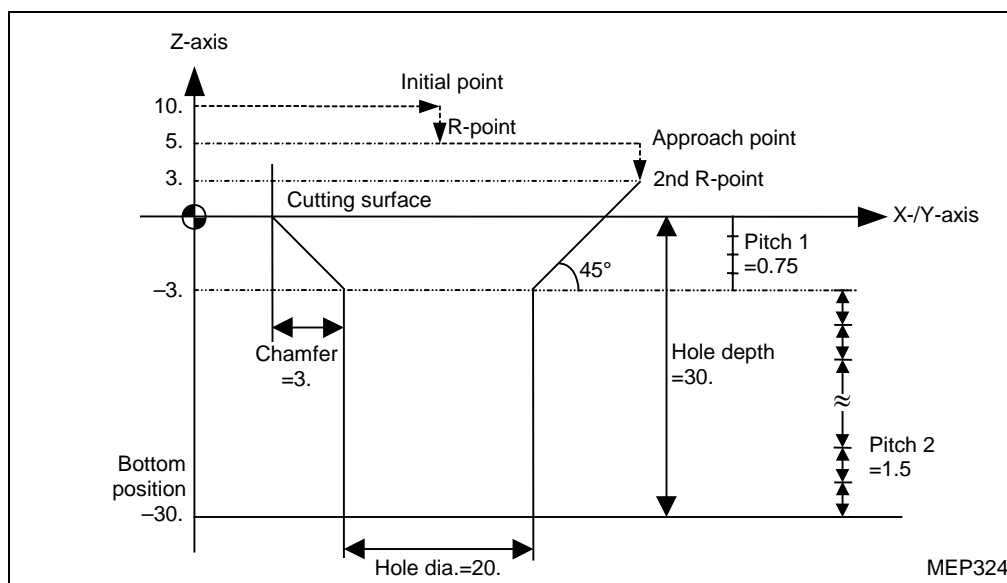
After cutting down to the bottom of the hole by helical interpolation, the tool performs a circular interpolation for full circle, and then escapes to the axis of the hole before returning to the initial point or R-point on the Z-axis at the rapid traverse.

K = 0 (Without finishing on the hole bottom)

After cutting down to the bottom of the hole by helical interpolation, the tool escapes to the axis of the hole while returning through quarter the pitch, and then returns to the initial point or R-point on the Z-axis at the rapid traverse.

3. Sample program

```
G28 X Y Z
S3000 M3
G90 G95 G0 G43 Z10. H2
G130 R5.Z-30.D20.T15.V30.F0.5 H3. I0.75 J1.5 K1 Q1 E3.
X10. Y10.
G67
M30
```



13-2 Suppression of Single-Block Stop for Fixed Cycles

13-2-1 Function description

This function allows point-machining (i.e. hole-machining) operation to be checked efficiently in the mode of single-block operation by reducing the number of CYCLE START button operations (in other words, the frequency of operation stop).

When this suppression function is valid, the single-block stop is reduced to the following three positions:

1. Completion position of positioning,
2. Completion position of approaching to point R,
3. Completion position of successive hole-machining operations and return.

Remark 1: Use the following parameter to make this function valid or invalid:

Suppression of single-block stop for fixed cycles:

F87 bit 4 = 1 (Valid) / 0 (Invalid)

Remark 2: This function is available for both EIA/ISO and MAZATROL program types.

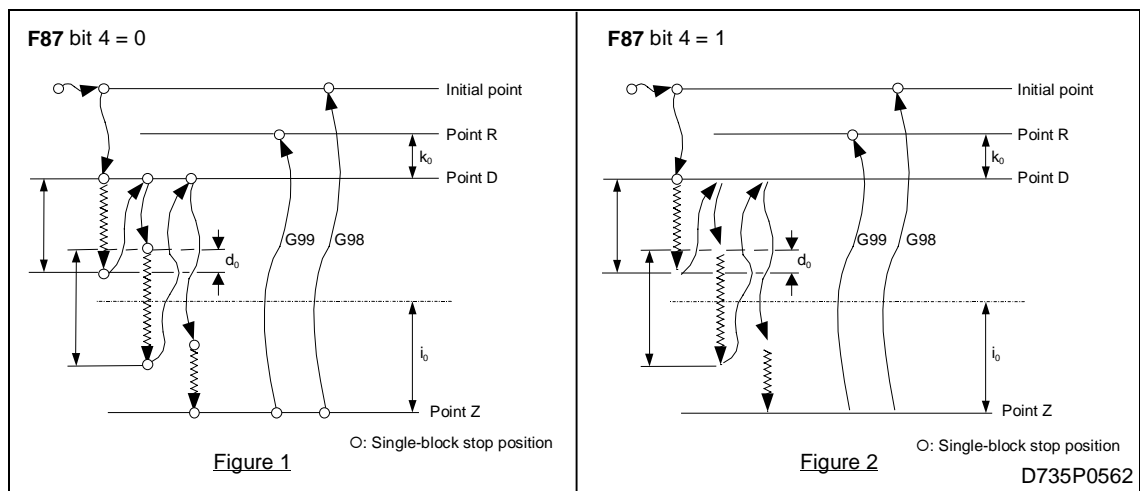
13-2-2 Examples of operation

The changes made by this function to the single-block operation are described below taking the deep-hole drilling cycle (G83) as an example.

G83 [Xx Yy] Rr Zz Ff₀ Qq₀ [Dd₀ Kk₀ Pt_c li₀ Hh₀]

In the deep-hole drilling cycle, the tool is to return repeatedly after drilling through a certain depth in order to avoid an increase in load due to clogging with chips. As shown in Figure 1, there are a number of positions at which the operation stops in the machining cycle.

The number of stopping positions can be reduced as in Figure 2 by the single-block stop suppression.



13-3 Initial Point and R-Point Level Return: G98, G99

1. Function and purpose

Commands G98 or G99 can be used to select whether the return level of the final sequence during fixed-cycle operation is to be set at point R or at the initial point of machining.

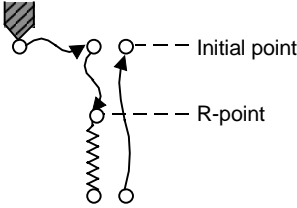
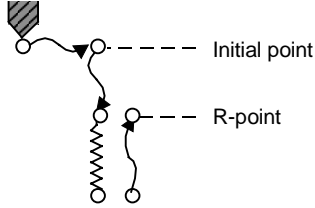
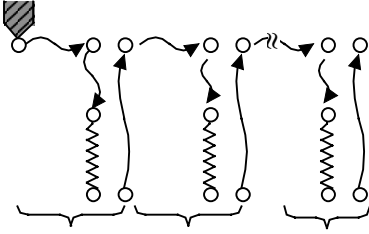
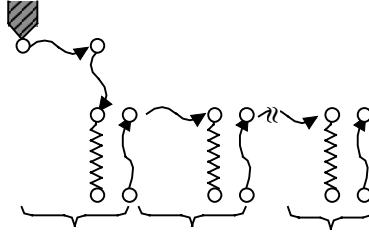
2. Programming format

G98: Initial point level return

G99: R-point level return

3. Detailed description

The following represents the relationship between the G98/G99 mode and repeat times:

Drilling time	Sample program	G98 (At power-on or after cancellation using M02, M30, or RESET key)	G99
Only once	G81 X100. Y100. Z-50. R25. F1000	 <p>Initial point R-point</p> <p>Return to initial point level.</p>	 <p>Initial point R-point</p> <p>Return to R-point level.</p>
Twice or more	G81 X100. Y100. Z-50. R25. L5 F1000	 <p>No. 1 time No. 2 time Final time</p> <p>Always return to initial point.</p>	 <p>No. 1 time No. 2 time Final time</p> <p>MEP158</p>

13-4 Workpiece Coordinate Setting during the Fixed-Cycle Mode

The specified axis moves in the workpiece coordinate system currently valid.

The new coordinate system does not become valid for Z-axis till the R-point positioning or another Z-axis movement after completion of positioning on X-Y plane.

Note: For axis addresses and R, reprogramming must be done during workpiece coordinate updating even if their settings are the same as respective previous ones.

Example: G54 XX₁ YY₁ ZZ₁

G81 XX₂ YY₂ ZZ₂ Rr₂

⋮

G55 XX₃ YY₃ ZZ₂ Rr₂

XX₄ YY₄

XX₅ YY₅

⋮

Reprogramming is required even if Z and R are of the same values as for the last time.

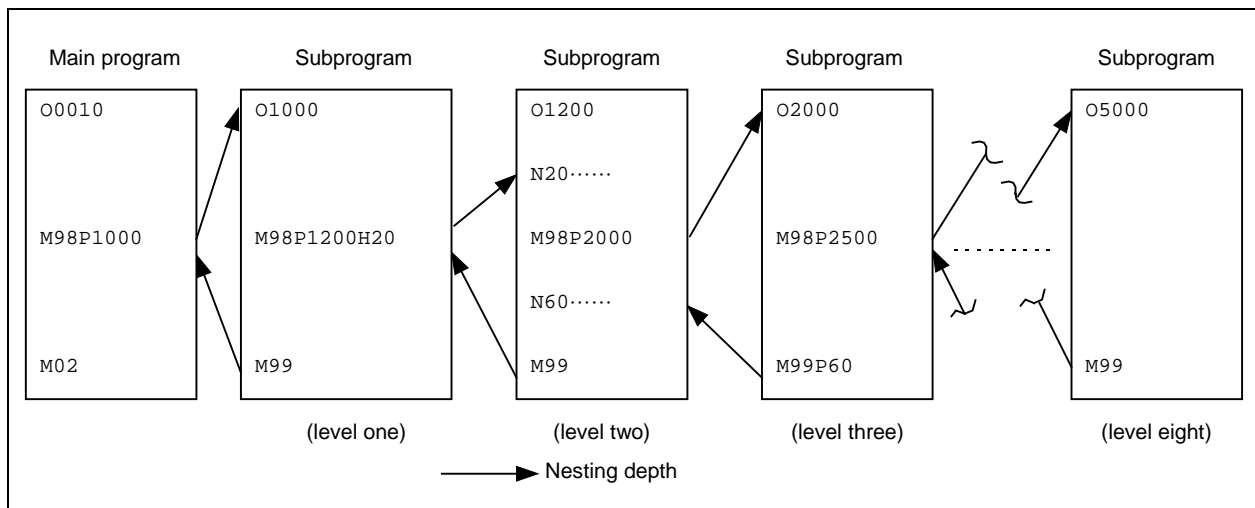
13-5 Subprogram Control Commands: M98, M99

1. Function and purpose

Prestoring a fixed sequence or pattern to be repeatedly used into the memory as a subprogram allows the subprogram to be called from the main program as required.

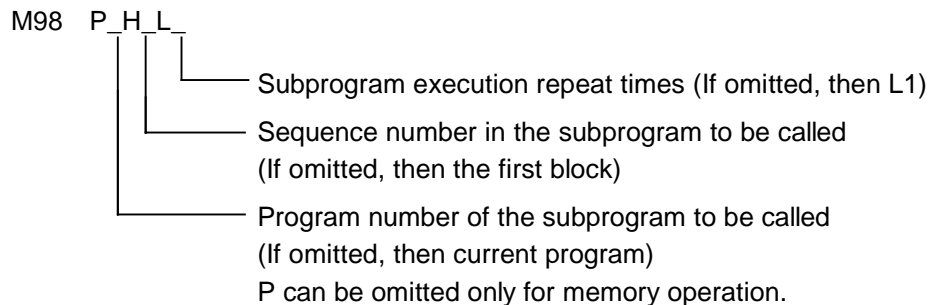
Use command M98 to call a subprogram, and use command M99 to return from the subprogram.

You can also call another subprogram from the current one. Its maximum available depth of nesting (call of one subprogram from another) is of eight levels.

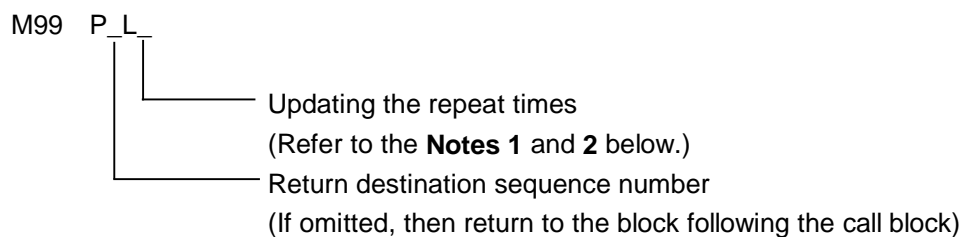


2. Programming format

[Subprogram call]



[Return from the subprogram]

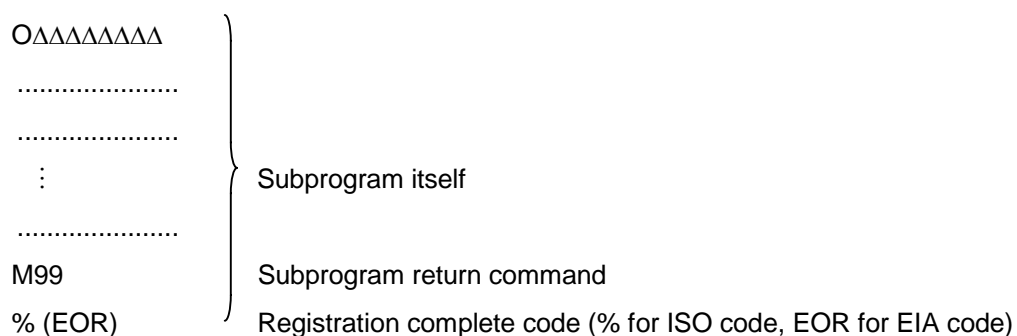


Note 1: If 0 is set, the program flow will return to the return destination sequence, irrespective of the setting of L (repeat times) specified with M98, since the residual value of L will be regarded as 0. If selection of the return destination sequence number has been omitted, the program flow will return to the block following the call block.

Note 2: If a value equal to or larger than 1 is set, the current subprogram will endlessly loop.

3. Creating and registering subprograms

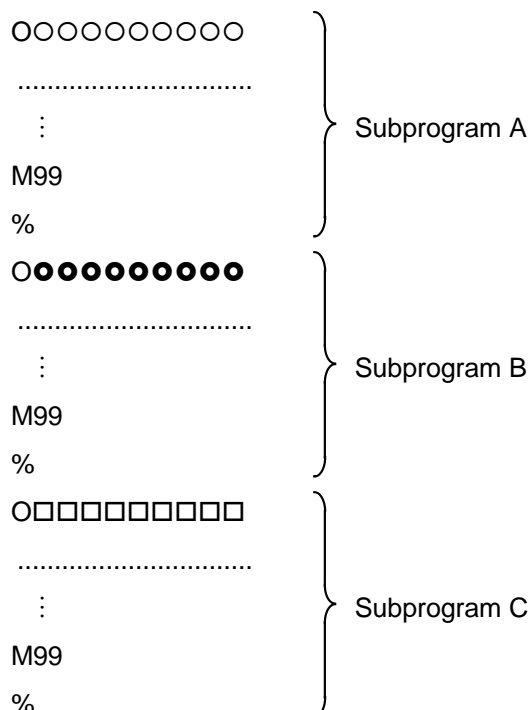
The subprogramming format is similar to the usual machining program creation format for memory operation, except that the subprogram termination instruction code M99 (P_ L_) must be included in the final block as a single-command block.



Any number of subprogram identification from 1 to 99999999 can be used, but within optional machine specifications. Even a program stored on a tape without identification number can be registered under any number specified for program loading. The maximum of call of a program from its subprogram is of eight levels. An alarm **842 SUB PROGRAM NESTING EXCEEDED** will occur if this limit is exceeded.

Programs are stored into the memory without distinction between main and sub under the specified number. The identification numbers of the main programs and subprograms stored, therefore, must not overlap. Otherwise, an alarm **576 SAME PROGRAM No. DESIGNATED** will occur during registration.

Example of registration



Note 1: Main programs can be used for memory operation or tape operation, whereas subprograms must always reside in the memory.

Note 2: In addition to M98, the following commands are subjected to subprogram nesting:

- G65 (Macro single call)
- G66 (Macro modal call)
- G66.1 (Macro modal call)
- G-code macro call
- Auxiliary function (M, S, T, etc.) macro call
- MDI interruption
- Automatic tool length measurement
- Multi-step skip function

Note 3: The following commands are not subjected to subprogram nesting and can thus be used for call at more than eight levels of nesting:

- Fixed-cycle
- Macro interruption

4. Executing a subprogram

M98: Subprogram call command

M99: Subprogram return command

Programming format

M98 P_{p₁} Hh₁ Ll₁

p₁: Number of the subprogram to be called (eight digits, maximum)

h₁: Any sequence number in the subprogram to be called (five digits, maximum)

l₁: Repeat times (four digits, maximum, from 1 to 9999) Execution will occur only once if L is omitted, or will not occur if L is set equal to 0.

For example,

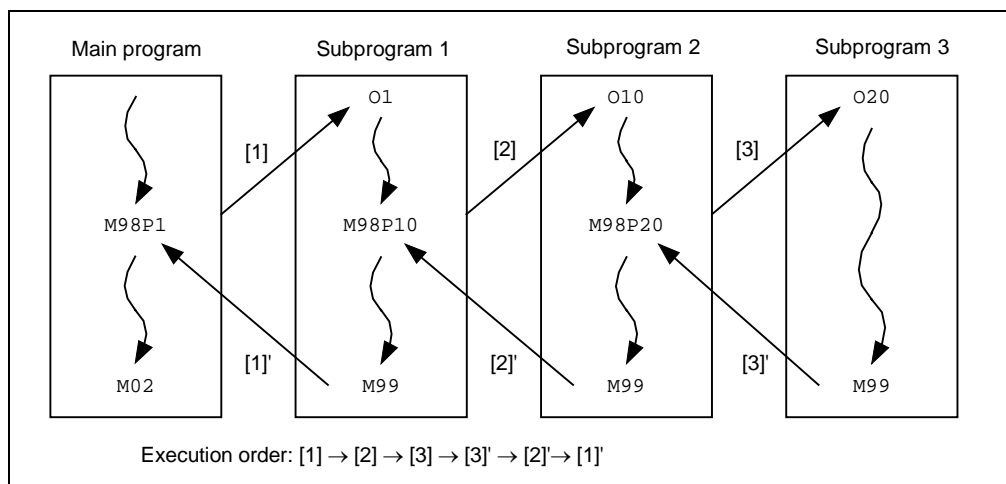
M98 P1 L3 is equivalent to:

M98 P1

M98 P1

M98 P1

Example 1: For three subprogram calls (level three nesting):

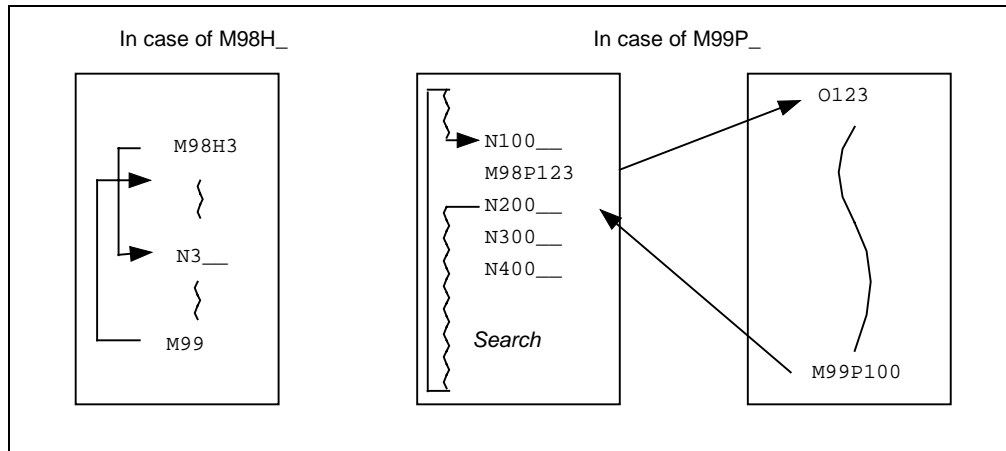
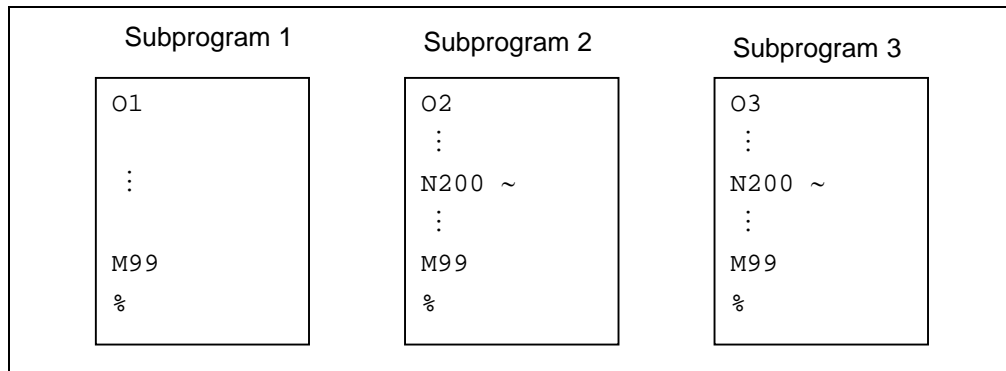


For nesting, the number of codes M98 and that of codes M99 must be the same (for example [1] for [1]', [2] for [2]', and so on up to the final M98 or M99).

Modal information is automatically updated in the order of actual execution, irrespective of whether the program is a main program or a subprogram. After subprogram call, therefore, extra careful attention must be paid to the status of the modal data.

Example 2: M98H_ M99P_

In this example, sequence number search will be made within the program that contains the call instruction.

**Example 3:** M98 P2 given in the main program:

- If memory search is executed to search for the block of O2 N200, modal data will be updated according to the data in the area from O2 to N200.
- The same sequence number can be used in different subprograms.
- A subprogram can be made to loop an ℓ_1 number of times by programming M98 P₁ L ℓ_1 .

5. Miscellaneous notes

1. An alarm **844 PROGRAM No. NOT FOUND** will result if the designated P (program number) is not found.
2. The block of M98P_ or M99 can cause a single-block stop only when addresses other than O, N, P, L, and H are present. (If the block is programmed as X100. M98P100, then the program will branch into O100 after execution of X100.)
3. Setting an M99 P_ code in the main program returns the program flow to the starting position.
4. Branching from tape operation into a subprogram is possible with M98 P_, but the return destination sequence number cannot be designated with M99 P_.
5. It takes much time for the program to search for a sequence number if it has been designated with M99 P_.
6. MAZATROL program cannot be called up from EIA/ISO program unless the related option is provided.

13-6 Mutual Subprogram Call between EIA/ISO and MAZATROL (Option)

1. Function and purpose

This optional function is provided to allow a MAZATROL program to be called as subprogram from EIA/ISO program as well as vice versa. Refer to the section related to “subprogram unit” in the Programming Manual (MAZATROL Programming) for details on the procedure for calling an EIA/ISO program from MAZATROL program.

Note 1: The option is only required for calling a MAZATROL program from EIA/ISO program. (As a standard is provided the function for calling an EIA/ISO program from MAZATROL program.)

Note 2: Necessary modal G-codes must all be contained in a subprogram, in both cases of mutual call between EIA/ISO and MAZATROL, since modal information of G-codes may be changed in transition to subprogram.

Note 3: The amount of tool-length offset is not cancelled through transition from MAZATROL to EIA/ISO nor through return from EIA/ISO back to MAZATROL.

2. Programming format

Subprogram call from an EIA/ISO program

M98 P H L

- Subprogram execution repeat times (Default: L1)
- Designation of a sequence No. in the EIA/ISO subprogram to be called (Default: Head block)
- H-code is not available for a MAZATROL subprogram
- Program No. (Work No.) of the subprogram to be called (Default: Reflexive [same No. as the calling program])
- P-code omission for reflexive call, however, is only available for memory operation mode.

Subprogram call from a MAZATROL program

UNo.	UNIT	WORK No.	\$	REPEAT		
	SUB PRO	[1]		[2]		
SNo.	ARGM 1	ARGM 2	ARGM 3	ARGM 4	ARGM 5	ARGM 6
1	[3]					

[1] Program No. (Work No.) of the required subprogram

[2] Subprogram execution repeat times (Default: 1)

[3] Argument data for the subprogram (as required)

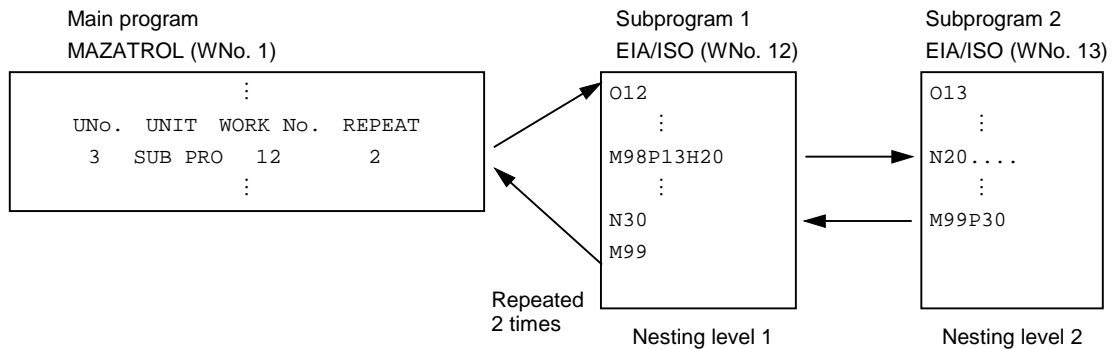
Return from an EIA/ISO subprogram

M99 P L

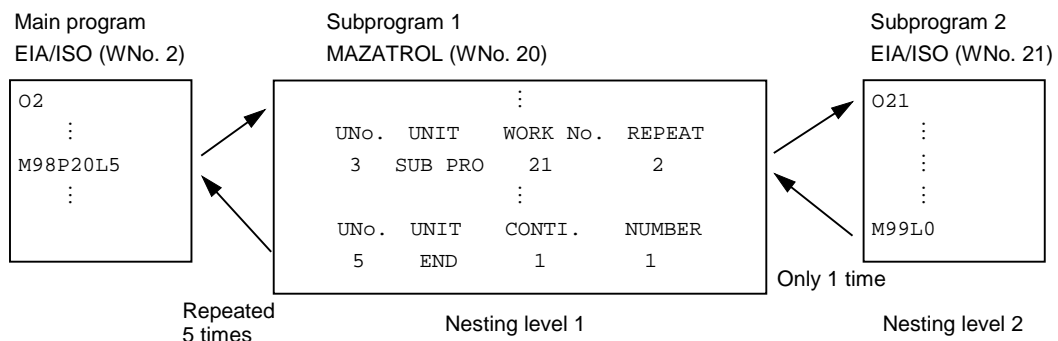
- Correction for repeat times (Setting “L0” here returns the control to the calling program, irrespective of the L data in calling block of M98. Note that the subprogram will endlessly loop if a value equal to or larger than 1 is set here at address L.)
- Return-destination sequence No. (Default: Next block of the calling block)
- P-code is not available if the return destination is a MAZATROL program.

3. Example of subprogram call

1. EIA/ISO program called from a MAZATROL program

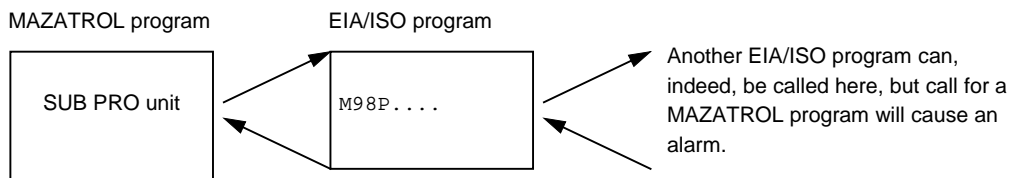


2. MAZATROL program called from an EIA/ISO program



4. Restrictions

1. An EIA/ISO program which is to be called from MAZATROL program should not contain any subprogram-calling block with destination of a MAZATROL program. Otherwise, an alarm will be caused.



2. Maximum available nesting level is 9 (or 8 when the main program is an EIA/ISO program).
3. Set "1" under **CONTI.** item in the **END** unit of a MAZATROL program to be called from EIA/ISO program. Otherwise (i. e. **CONTI.** = 0), the flow of program execution will be terminated at the end of that MAZATROL subprogram.
4. Unit number search (for subprogram execution from the midst) is just as little available in calling a MAZATROL program from EIA/ISO program, as sequence number search in calling an EIA/ISO program from MAZATROL program. In both cases, the subprogram can be executed only from the beginning.
5. If the MAZATROL program which is called as a subprogram has an M99 command designated in its **MANU PRO** or **M-CODE** unit, then the execution of the subprogram will be terminated upon completion of that unit to return to the main program.

6. Remarks on coordinate system

For an EIA/ISO program called from MAZATROL program

- The data of basic coordinate system (in **WPC** unit) are all valid for the subprogram with the exception of **th**.
- The data of auxiliary coordinate system (in **OFFSET** unit) are all invalid for the subprogram.
- The last coordinate system established within the EIA/ISO subprogram will be cancelled upon return to the main program and the MAZATROL basic coordinate system will be restored.

For a MAZATROL program called from EIA/ISO program

- The last coordinate system established in the EIA/ISO main program will be cancelled upon branch into the subprogram to validate the basic coordinate system for execution of the MAZATROL subprogram.
- Do not fail to set an appropriate coordinate system anew in the EIA/ISO program for the succeeding part upon return from the MAZATROL subprogram.
- When theta (**th**) is included in the basic coordinate system for a MAZATROL subprogram, set "G92.5X0Y0R0" (cancellation of the **WPC** unit's theta) to cancel the theta data.

7. Remarks on modal data

The following table lists the modal (*1) preparatory functions which are automatically set for an EIA/ISO program called from or resumed after MAZATROL program.

G-code	Function	G-code	Function
00	Positioning	50.1	Cancellation of G-command mirror image
15	Cancellation of polar coordinate command	64	Cutting mode
17	X-Y plane selection	67	Cancellation of modal call for user macro
20	Inch command (*2)	68	Coordinate rotation
21	Metric command (*2)	80	Cancellation of fixed-cycle call
23	Cancellation of pre-move stroke check	90	Absolute data input
40	Cancellation of tool-diameter offset	95	Synchronous feed (per revolution)
50	Cancellation of scaling	98	Return to initial-point level during fixed cycle

*1: The adjective "modal" refers to a function code or an address value which remains valid until it is cancelled by an overriding command.

*2: It depends on the preset specification whether the inch or metric command mode is validated.

8. The MAZATROL tool data (length and diameter) can be referred to in an EIA/ISO program which is called from MAZATROL program. After the MAZATROL program has been called up from the EIA/ISO program, tool length data is not cancelled when control is returned to the EIA/ISO program.

9. Remarks on the M-code mirror image

To retain the M-code mirror image function (for additional machining of a symmetrical shape with respect to an axis or a point), the related M-code command is to be cancelled (by M90) temporarily before subprogram call and then given anew at the head of the subprogram in case of the call for an EIA/ISO program from MAZATROL program or vice versa.

13-7 Variables Commands

1. Function and purpose

A program can be made more flexible and versatile by designating variables, in stead of assigning data directly to certain addresses within the program, and then assigning the data of those variables each time that program is to be executed.

2. Programming format

#●●● = ○○○○○○○○○

or

#●●● = [Expression]

3. Detailed description

A. Expressing a variable

	Example
#m where m must be a numerical value.	#100
#[f] where f must be either one of the following:	
Value m	#[123]
Variable	#[#543]
Expression Operator Expression	#[#110+#119]
– (minus) expression	#[-#120]
[Expression]	#[#119]
Function [Expression]	#[SIN[#110]]

<Remarks>

- The available standard operators are +, –, * and /.
- Functions can be used only under user macro specifications.
- A negative variables number causes a program error.
- Examples of wrong variables expression are shown below for reference only.

Wrong		Correct
#6 / 2	→	#[6 / 2]
#--5	→	#[-[-5]]
#-[#1]	→	#[-#1]

B. Available types of variables

The following table lists available types of variables:

Type	Number	Description	Remarks
Common variables	100 - 149, 500 - 549	Commonly used for main-, sub-, or macroprograms.	Type A: 100 sets
	100 - 199, 500 - 599		Type B: 200 sets
	100 - 199, 500 - 699		Type C: 300 sets
	100 - 199, 500 - 999		Type D: 600 sets
Local variables	1 to 32	Locally used in macroprograms.	
System variables	from 1000 on	Uses depend on the system.	

Note 1: All common variables do not get lost by switching-off.

Note 2: The type D (600 sets) is optional.

C. Using variables

Variables can be used for all addresses, except O, N, and / (slash).

- 1) Using the data of a variable directly
 $X\#1$ The value of #1 is used as that of X.
- 2) Using the complement of a variables value
 $X-\#2$ The sign-changed value of #2 is used as the value of X.
- 3) Defining variables
 $\#3=\#5$ The value of variable #5 is used as the value of #3.
 $\#1=1000$ The value of 1000 (regarded as 1000.) is used as the value of #1.
- 4) Defining variables calculation expressions
 $\#1=\#3+\#2-100$ The result of calculation of $\#3 + \#2 - 100$. is used as the value of #1.
 $X[\#1+\#3+1000]$ The result of calculation of $\#1 + \#3 + 1000$. is used as the value of X.

<Remarks>

- The variable cannot be defined in the same block as the corresponding address. The definition must be given in the preceding block as follows:

Wrong		Correct
$X\#1=\#3+100$	→	$\#1=\#3+100$ $X\#1$

- Up to five sets of brackets [] can be used.

$\#543=-[[[[[\#120] / 2 + 15 .] * 3 - \#100] / \#520 + \#125 + \#128] * \#130 + \#132]$

- For definition of variables, there are no limitations on the number of variables or the number of characters per variable.
- The value of a variable must be one from 0 to ± 99999999 . Incorrect calculations may result if this data range is overstepped.
- Definition of variables becomes valid for the next block and its successors.

$\#1=100$ #1 = 100 becomes valid from the next block onwards.

$\#1=200 \quad \#2=\#1+200$ #1 = 200 and #2 = 300 become valid from the next block onwards.

$\#3=\#1+300$ #3 = 500 becomes valid from the next block onwards.

- All variables are regarded as followed by the decimal point.
 For $\#100 = 10$, for example, $X\#100$ is regarded as "X10."

13-8 Figure Rotation: M98 (Option)

1. Function and purpose

The figures commanded by subprograms can be executed after rotation by using subprogram call-out, center commands I, J, K, and word L.

2. Programming format

M98 P_H_I_J_ (K_) L_

(G17 mode: I, J G18 mode: K, I G19 mode: J, K)

M98 : Subprogram call-out M code

P : Program number of subprogram to be called out

H : Sequence number of subprogram to be called out

I, J, K : Incremental value of figure rotation-center coordinate (incremental value from the start point)

L : Subprogram repeat times (when L is 1 and under, it is not regarded as the figure rotation)

3. Detailed description

1. Subprograms are executed by the above format commands and they are completed one time by M99 subprogram return, then the subprogram commands are rotated on the rotation information which consists of the start point, the center, and the end point. The rotation angle can be added up every one-time completion by specifying repeat times as two and over, so the figures commanded by subprograms can be arranged by the times specified after rotation with the center coordinates as their standard.
2. The first time of subprograms by subprogram called-out is executed on the basis of 0 rotation angel to trace the locus specified. All blocks in the subprogram are rotated.
3. When the start point and the end point of the subprogram are not on the same circle with the figure rotation center as the center, the interpolation based on the command, in which the subprogram end point is its start point and the end point of the first move command block of the subprogram rotated is its end point, is carried out.
4. Cooperative use of absolute value and incremental value commands
In figure rotation subprograms, the cooperative use of absolute values and incremental values is possible. For the absolute value mode the rotation of two times and over can be executed by the same command if the standard figure is programmed by absolute values.
5. Subprogram controlling
The nesting of subprograms is possible even during figure rotation. However, one-time completion of figure rotation is no other than M99 of the nesting level called-out by figure rotation subprogram.

Note 1: Figure rotation is carried out on the workpiece coordinate system, so it can be shifted by the commands G92, G52, G54 to G59 (workpiece coordinate system shift).

Note 2: Figure rotation is carried out on the workpiece coordinate system, so the functions on the machine coordinate system (return to the zero point, one-direction positioning, etc.) are not rotated.

Note 3: The figure rotation command during diagram rotation will cause an alarm **849 FIGURE ROTATE NESTING EXCEEDED**.

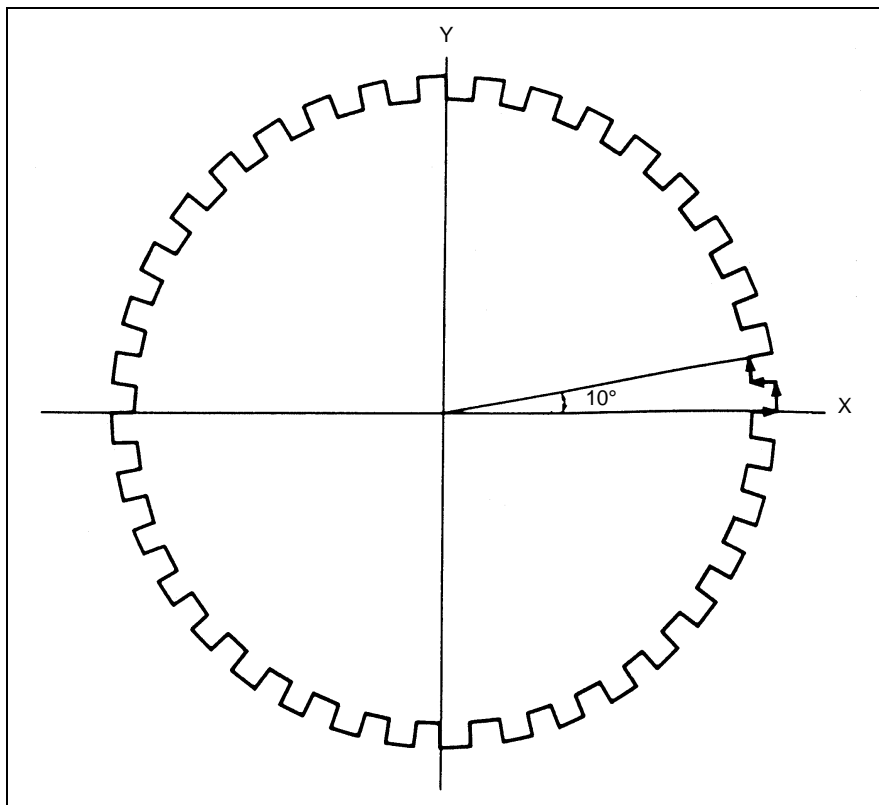
Note 4: Figure rotation and program coordinate rotation cannot be commanded simultaneously. If so commanded, it will cause an alarm **850 G68 AND M98 COMMANDS SAME BLOCK.**

4. Sample programs

Example 1: Cutting into a gear shape

Prepare the program for machining of one tooth using a subprogram first and then designate the teeth quantity during program call:

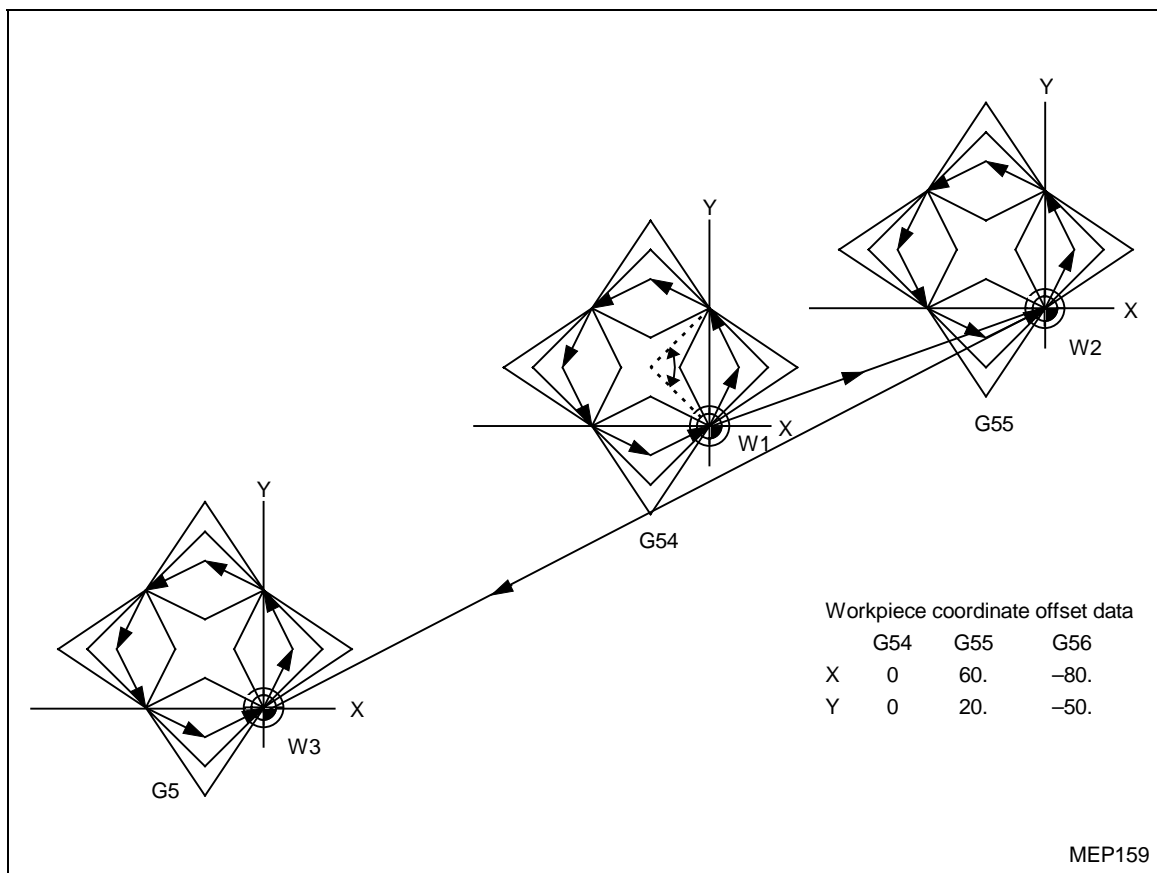
G92X0Y0	}	Main program		
G90G00X50.		}	Startpoint positioning and call-out of	
M98P7L36I-50.			}	diagram rotation of gear cut
G00X0Y0				
M02				
O7	}	Subprogram (O7)		
G03X54.358Y.190J50.F100		}	Data of gear basic form	
X54.135Y4.927I-54.354J-.190				
X49.810Y4.358J-50.				
X49.240Y8.682I-49.810J-4.358				
M99				
%				



Example 2: Cooperative use with workpiece offset function

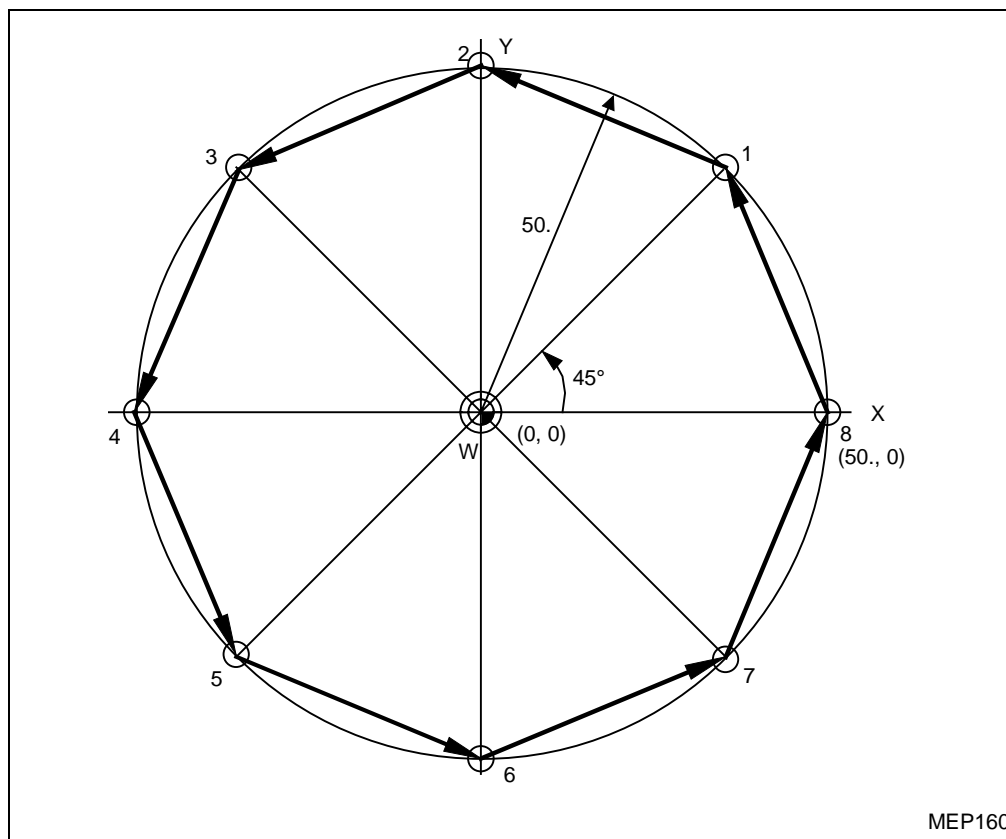
Diagram rotation can be carried out on the workpiece coordinate system.

<pre> G54(G55, G56)G90XY G90X0Y0 M98P10H1I-10.J10.L4F100 M98P10H2I-10.J10.L4 M98P10H3I-10.J10.L4 M98P10H4I-10.J10.L4 M02 % O10 N1G01X-5.Y10. X0Y20. M99 N2G01X5.Y10. X0Y20. M99 N3G01X10.Y10. X0Y20. M99 N4G01X15.Y10. X0Y20. M99 %</pre>	<p>Main program</p> <p>Subprogram (O10)</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------



Example 3: Using the fixed-cycle machining function

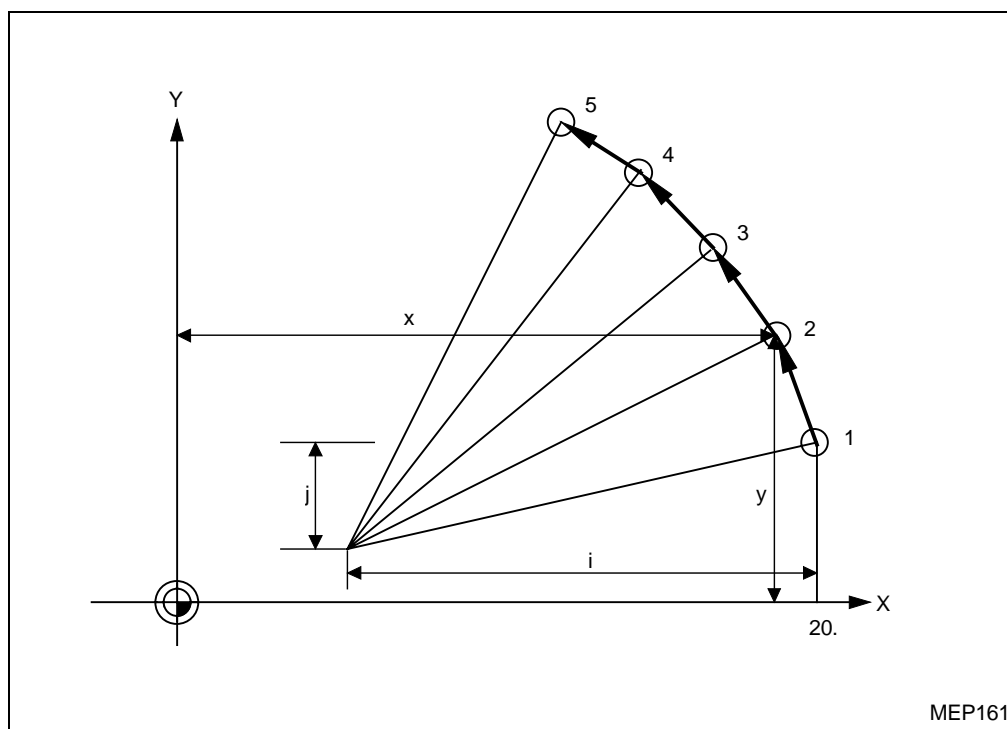
Make a subprogram which contains only the positioning data for fixed-cycle operation first and then store fixed-cycle hole-drilling data into the memory during subprogram call (executing such a program allows rotation of the setting position in the subprogram, and hence, bolt-hole circle machining).



```
G92X0Y0Z0
G91X50.
G90G81Z-10.R-50.F100
M98P101I-50.L8
G00X0Y0
M02
%
O101
X35.355Y35.355
M99
%
```

Main program
Hole-drilling data storage and
subprogram call

Subprogram (O101)
Positioning data (absolute data)

Example 4: Application to fixed-cycle operation (Application to arcs)

```
G90G82X20.Z-20.R-5.P100F200
```

```
M98P102I-iJ-jL5
```

```
M02
```

```
%
```

```
O102
```

```
XxYy
```

```
M99
```

```
%
```

Main program

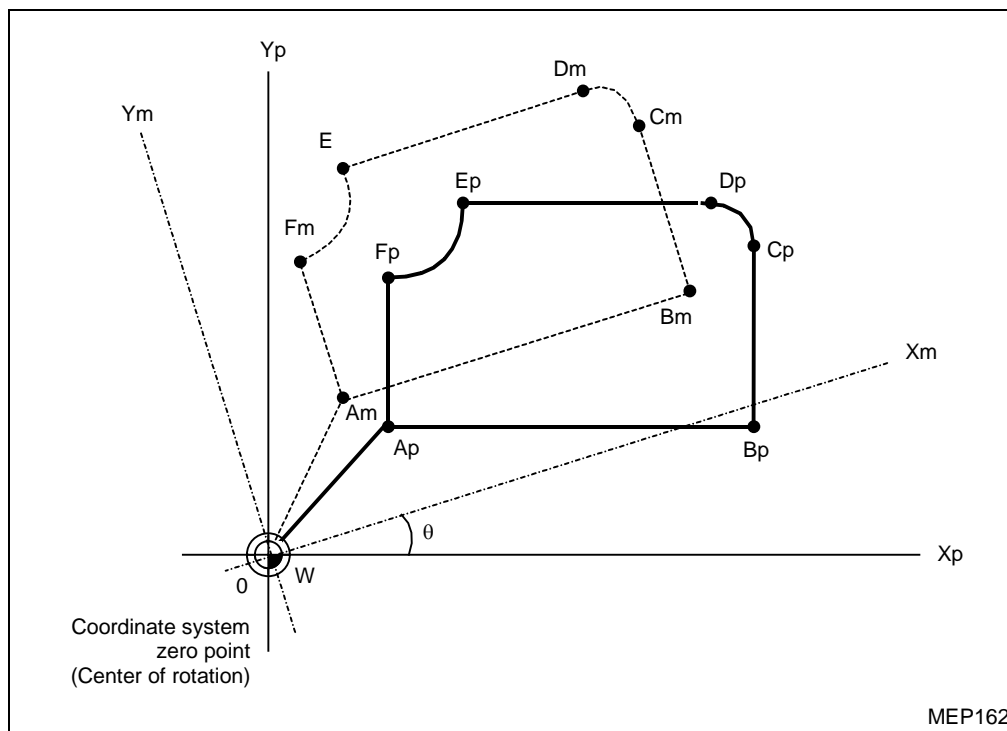
Start point positioning, hole-drilling, figure rotation call

Subprogram (O102)

Positioning data

13-9 Programmed Coordinate Rotation

Programmed coordinate rotation is performed to rotate the machining pattern itself on the workpiece by rotating the workpiece coordinate system.



The program coordinate path becomes:

$A_p \rightarrow B_p \rightarrow C_p \rightarrow D_p \rightarrow E_p \rightarrow F_p \rightarrow A_p$

The machine coordinate path after coordinate rotation becomes:

$A_m \rightarrow B_m \rightarrow C_m \rightarrow D_m \rightarrow E_m \rightarrow F_m \rightarrow A_m$

1. Correlationships to other functions, and precautions

1. Coordinate rotation is valid only for the automatic operation modes (tape operation, memory operation, and MDI); it is invalid for manual jog or rapid feed, handle feed, or reference point (zero point) return, whether it is automatic or manual.

Note 1: Even in the automatic operation modes, rotation does not occur during creeping feed for one-way positioning.

Note 2: If manual interruption using one of the manual modes mentioned above is made during an automatic operation mode, automatic operation must not be executed for any subsequent absolute data commands.

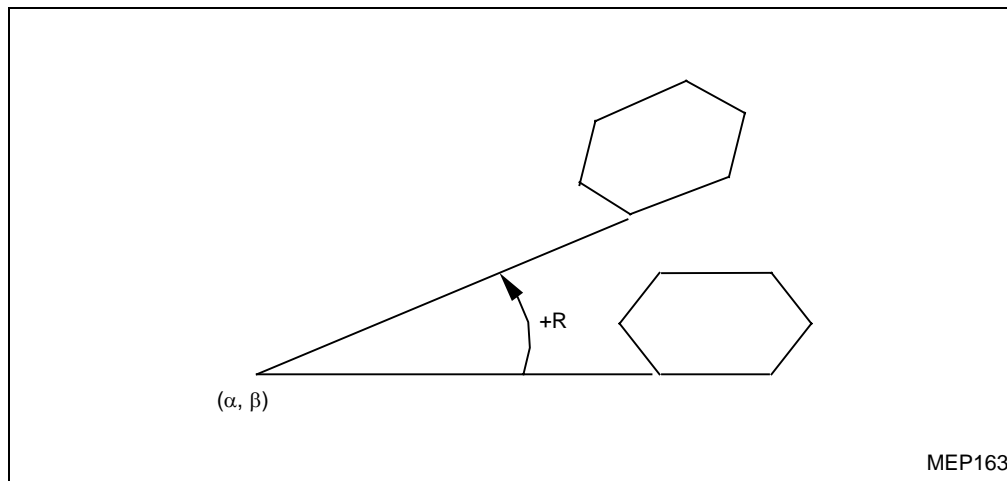
2. Calculation for offset processing is performed lastly. That is, coordinate rotate processing using the designated program is performed and tool diameter offsetting, tool length offsetting, tool position offsetting, etc. follow.
3. Higher priority is given to coordinate rotation than to the mirror image function. If these two functions are selected, therefore, coordinate rotation will precede mirror image processing.
4. The display of the current position represents the movements existing after coordinate rotation. If coordinate rotate processing is carried out for a single-axis command, therefore, simultaneous dual-axis movements will be displayed in general.

2. Programming format

Gn G68 α _ β _R_

where

- n : Plane selection code 17, 18, 19
- α, β : Coordinates of the center of rotation [Of the three axes (X, Y, Z), specify any two that correspond to the selected plane.]
- R : Angle of rotation (for counterclockwise rotation, use the plus sign)
Set an angle from -360.000 to $+360.000$ degrees in 0.001 degree steps.

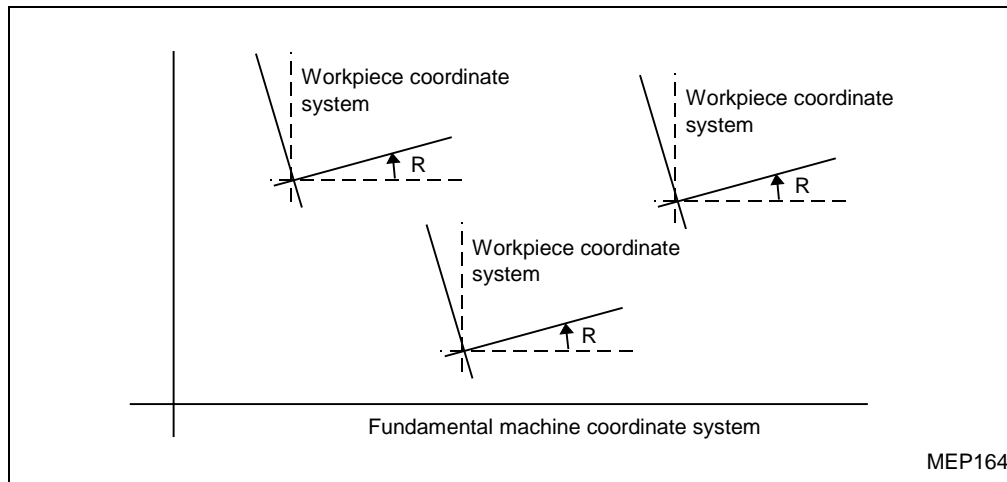


G69 (Coordinate rotate cancellation) This command code can be included in any block, whether independent from or together with other command codes.

3. Detailed description

1. The plane selection command code (G17, G18 or G19) does not need be included in the block of G68; it will become valid even if included before the G68 block.
2. The position where the G68 command code has been set becomes the rotational center if the rotations center coordinates (α, β) are omitted.
3. The absolute data command mode must always be used for rotational center coordinates (α, β) . For the angle of rotation (R), either the absolute data command mode or the incremental data command mode is to be used, depending on whether a G90 command or a G91 command has been issued.
4. If coordinate rotation is commanded during coordinate rotation, that command will be processed as center coordinate and rotational angle override command data.

5. Since program coordinate rotation is a function related to the workpiece coordinate system, the relationship between the fundamental machine coordinate system and the as-rotated workpiece coordinate system is as shown in the diagram below.

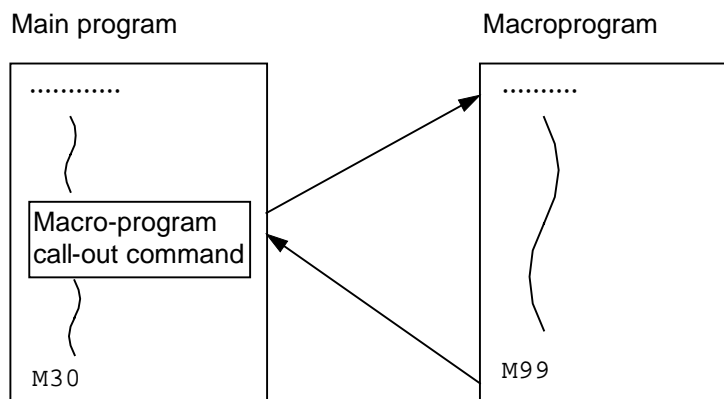


6. Program coordinate rotation cannot be executed together with figure rotation.

13-10 User Macros (Option)

13-10-1 User macros

Macroprogram call, data calculation, data input to/output from a personal computer, data control, judgment, branching, and various other instructions can be used with variables commands to perform measurements and other operations.



A macroprogram is a subprogram which is created using variables, calculation instructions, control instructions, etc. to have special control features.

These special control features (macroprograms) can be used by calling them from the main program as required. These calls use macro call instructions.

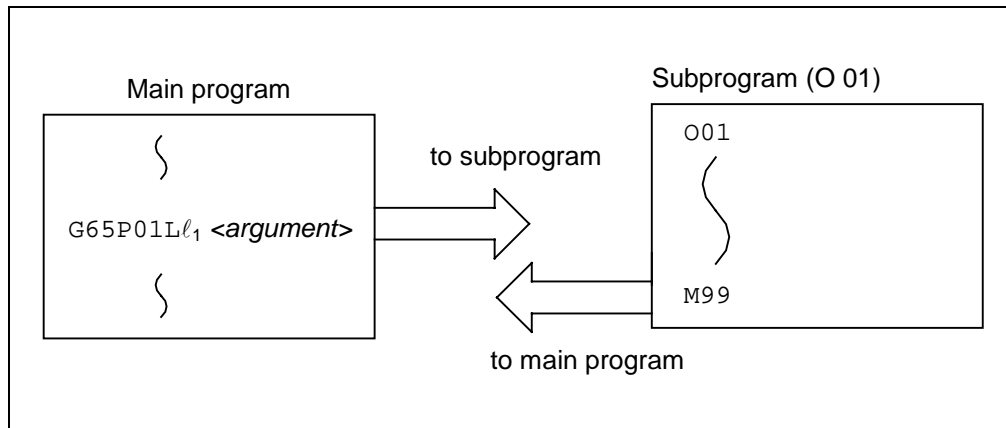
Detailed description

- When command G66 is entered, the designated user macro subprogram will be called every time after execution of the move commands within a block until G67 (cancellation) is entered.
- Command codes G66 and G67 must reside in the same program in pairs.

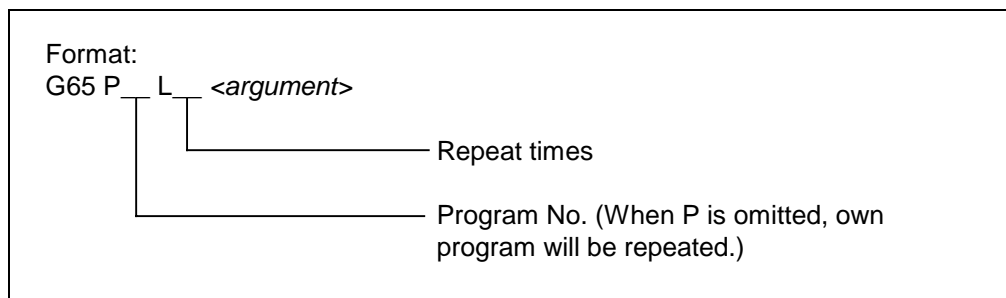
13-10-2 Macro call instructions

Two types of macro call instructions are provided: single-call instructions used to call only at the designated block, and modal call instructions used to call at each block within a macro call mode. Modal call instructions are further divided into type A and type B.

1. Single call



The designated user macro subprogram ends with M99.
Instruction G65 calls the designated user macro subprogram only once.



<Argument>

When argument is to be delivered to the user macro subprogram as a local variable, designate the required data with the respective addresses. (Argument designation is not available for a user macro subprogram written in MAZATROL language.)

In such a case, the argument can have a sign and a decimal point, irrespective of the address. Arguments can be specified using method I or II, as shown below.

A. Argument specification I

Format: A_B_C_ X_Y_Z_

Detailed description

- An argument can be specified using all addresses, except G, L, N, O, and P.
- Except for I, J, and K, addresses does not need be specified in an alphabetical order.
I_J_K_ ... Correct
J_I_K_ ... Wrong
- Addresses whose specification is not required can be omitted.
- The relationship between addresses that can be specified using argument specification I, and variables numbers in a user macro unit, is shown in the following table:

Relationship between address and variables number		Call commands and usable addresses	
Address specified using method I	Variable in macro-program	G65, G66	G66.1
A	#1	○	○
B	#2	○	○
C	#3	○	○
D	#7	○	○
E	#8	○	○
F	#9	○	○
G	#10	×	×
H	#11	○	○
I	#4	○	○
J	#5	○	○
K	#6	○	○
L	#12	×	×
M	#13	○	○
N	#14	×	×
O	#15	×	×
P	#16	×	×
Q	#17	○	○
R	#18	○	○
S	#19	○	○
T	#20	○	○
U	#21	○	○
V	#22	○	○
W	#23	○	○
X	#24	○	○
Y	#25	○	○
Z	#26	○	○

○: Usable ×: Unusable *: Usable in G66.1 modal

B. Argument specification II

Format: A_B_C_I_J_K_I_J_K_.....

Detailed description

- Up to a maximum of 10 sets of arguments that each consist of addresses I, J, and K, as well as A, B, and C, can be specified.
- If identical addresses overlap, specify them in the required order.
- Addresses whose specification is not required can be omitted.
- The relationship between addresses that can be specified using argument specification II, and variables numbers in a user macro unit, is shown in the following table:

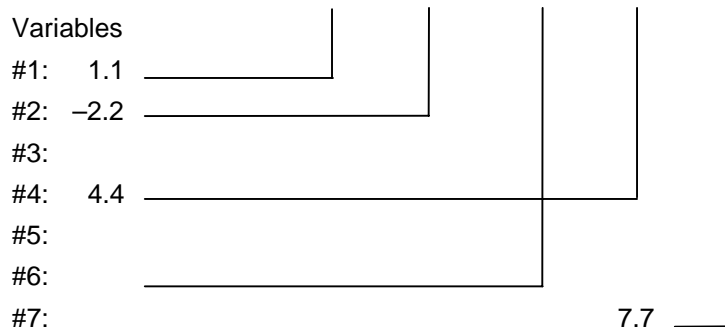
Argument specification II addresses	Variables in macro-programs	Argument specification II addresses	Variables in macro-programs
A	#1	K5	#18
B	#2	I6	#19
C	#3	J6	#20
I1	#4	K6	#21
J1	#5	I7	#22
K1	#6	J7	#23
I2	#7	K7	#24
J2	#8	I8	#25
K2	#9	J8	#26
I3	#10	K8	#27
J3	#11	I9	#28
K3	#12	J9	#29
I4	#13	K9	#30
J4	#14	I10	#31
K4	#15	J10	#32
I5	#16	K10	#33
J5	#17		

Note: In the table above, the numerals 1 through 10 have been added to addresses I, J, and K just to denote the order of arrangement of the designated sets of arguments: these numerals are not included in actual instructions.

C. Combined use of argument specification I and II

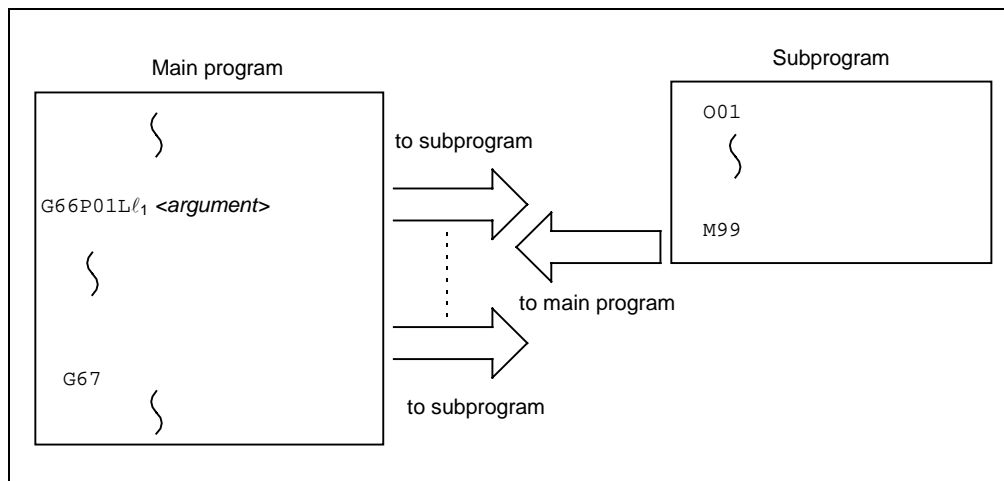
When both method I and method II are used to specify arguments, only the latter of two arguments which have an address corresponding to the same variable will become valid.

Example: Call command G65 A1.1 B-2.2 D3.3 I4.4 I7.7



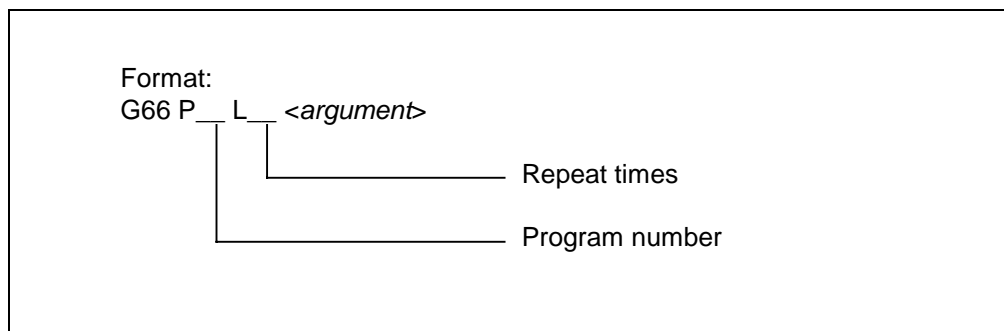
If two arguments (D3.3 and I7.7) are designated for the variable of #7, only the latter argument (I7.7) will be used.

2. Modal call, type A (Move command call)



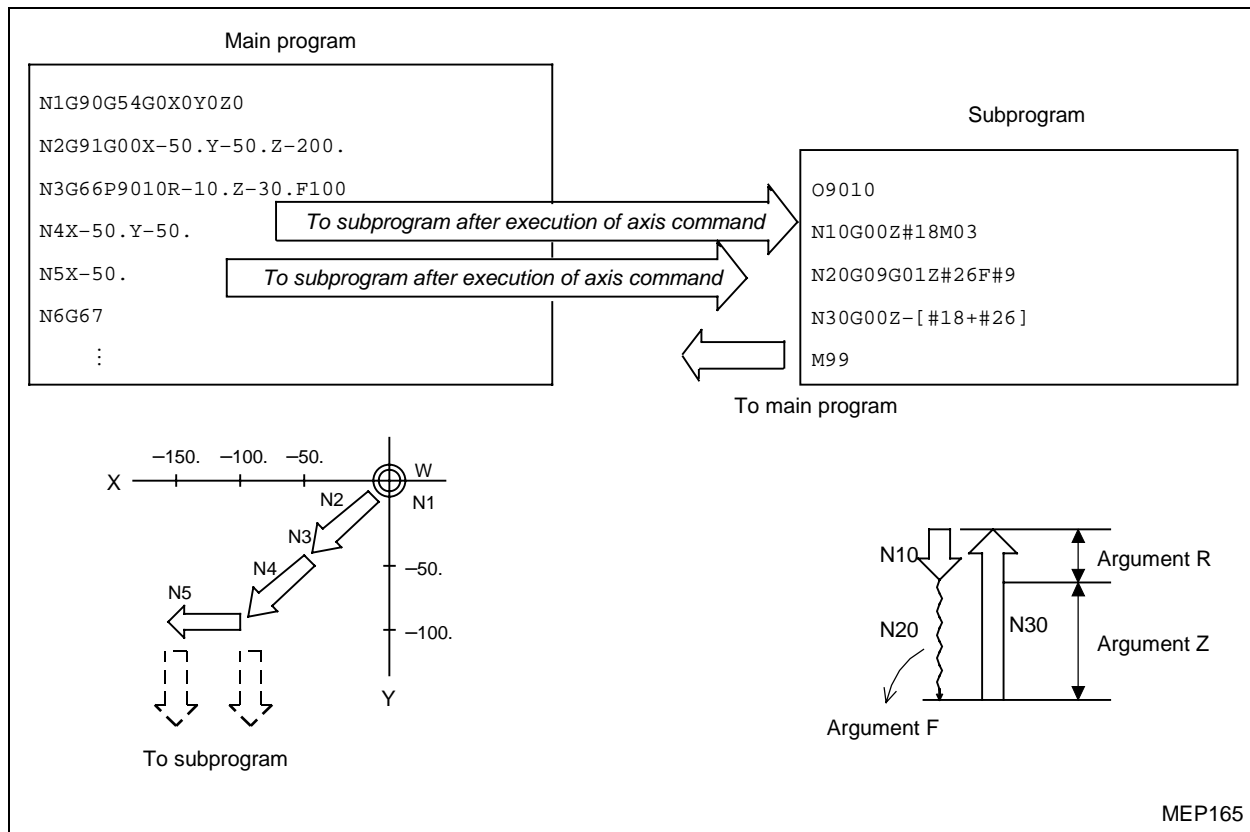
For a block that has a move command code between G66 and G67, the designated user macro subprogram is executed after that move command has been executed. The subprogram is executed an ℓ_1 number of times for the first call, or once for subsequent calls.

For modal call of type A, the methods of specifying <argument> are the same as used for single call.



Detailed description

- When command G66 is entered, the designated user macro subprogram will be called every time after execution of the move commands within a block until command G67 (cancellation) is entered.
- Command codes G66 and G67 must reside in the same program in pairs.
Entry of a G67 command without a G66 command results in an alarm **857 INCORRECT USER MACRO G67 PROG.**

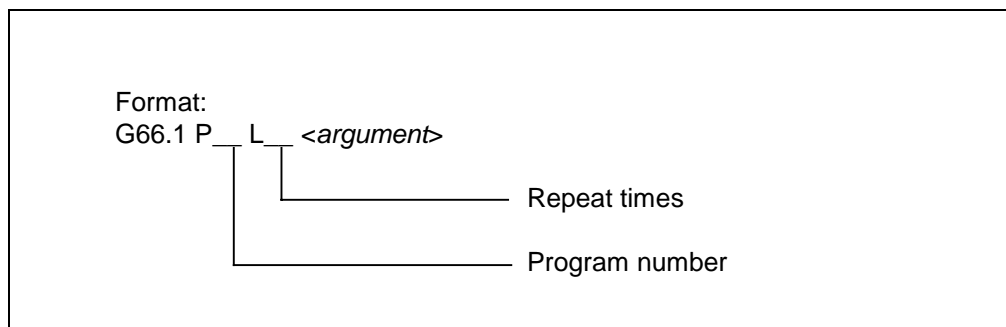
Example: Drilling cycle

Note 1: The designated subprogram is executed after the axis commands in the main program have been executed.

Note 2: No subprograms are executed for the G67 block and its successors.

3. Modal call, type B (Block-to-block call)

The designated user macro subprogram is called unconditionally for each of the command blocks present between G66.1 and G67. Execution of the macro program is repeated as specified with L for the first call, and only once for each of subsequent calls.



Detailed description

- During the G66.1 mode, only the codes O, N, and G in each of the read command blocks are executed. No other codes in those blocks are executed; codes other than O, N, and G are handled as arguments. However, only the last G-code and the N-codes following a code other than O or N become arguments.

- All significant blocks in the G66.1 mode are regarded as preceded by the command G65P_.

For example, the block of

```
N100G01G90X100. Y200. F400R1000
```

in the G66.1P1000 mode is handled as equivalent to

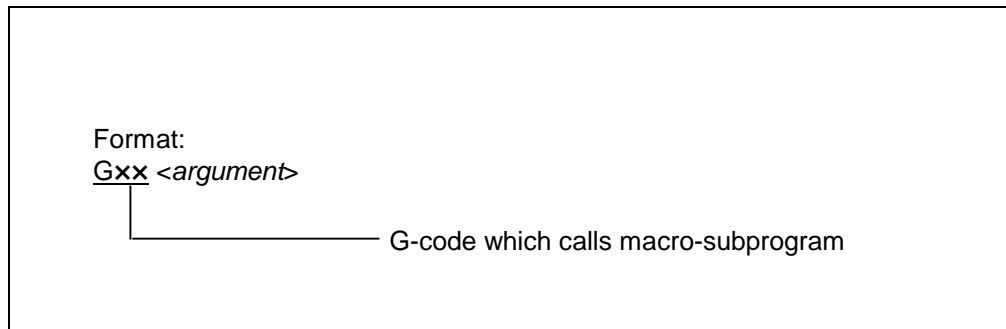
```
N100G65P1000G01G90X100. Y200. F400R1000.
```

Note: Call is executed even for the G66.1 command block of the G66.1 mode, with the relationship between the addresses of the arguments and the variables numbers being the same as for G65 (single call).

- The data range of the G, L, P, and N commands that you can set as new variables using the G66.1 mode is the same as the data range of usual NC commands.
- Sequence number N, modal G-codes, and O are all updated as modal information.

4. G-code macro call

The user macro subprograms of the required program number can be called just by setting G-codes.



Detailed description

- The instruction shown above performs the same function as those of the instructions listed below. Which of these listed instructions will apply is determined by the parameter data to be set for each G-code.

```
M98PΔΔΔΔ
```

```
G65PΔΔΔΔ <argument>
```

```
G66PΔΔΔΔ <argument>
```

```
G66.1PΔΔΔΔ <argument>
```

- Use parameters to set the relationship between Gxx (macro call G-code) and PΔΔΔΔ (program number of the macro to be called).
- Of G00 through G255, up to a maximum of 10 command codes can be used with this instruction unless the uses of these codes are clearly predefined by EIA Standards, such as G00, G01, G02, etc.
- The command code cannot be included in user macro subprograms that have been called using G-codes.

5. Auxiliary command macro call (M-, S-, T-, or B-code macro call)

The user macro subprograms of the required program number can be called just by setting M-, S-, T-, or B-codes.

Format:

Mm (or Ss, Tt and Bb)

————— M (or S, T and B) code which calls macro-subprogram

Detailed description (The following description also applies to S-, T-, and B-codes.)

- The instruction shown above performs the same function as those of the instructions listed below. Which of these listed instructions will apply is determined by the parameter data to be set for each M-code.

M98P $\Delta\Delta\Delta\Delta$

G65P $\Delta\Delta\Delta\Delta$ Mm

G66P $\Delta\Delta\Delta\Delta$ Mm

G66.1P $\Delta\Delta\Delta\Delta$ Mm

- Use parameter to set the relationship between Mm (macro call M-code) and P $\Delta\Delta\Delta\Delta$ (program number of the macro to be called).
Up to a maximum of 10 M-codes, ranging from M00 to M95, can be registered. Do not register the M-codes that are fundamentally required for your machine, nor M0, M1, M2, M30, and M96 through M99.
- If registered auxiliary command codes are set in the user macro subprograms that have been called using M-codes, macro calls will not occur since those special auxiliary command codes will be handled as usual ones (M-, S-, T-, or B-codes).

6. Differences in usage between commands M98, G65, etc.

- Arguments can be designated for G65, but cannot be designated for M98.
- Sequence numbers can be designated for M98, but cannot be designated for G65, G66, or G66.1.
- Command M98 executes a subprogram after M98 block commands other than M, P, H, and L have been executed, whereas G65 just branches the program into a subprogram without doing anything.
- Single-block stop will occur if the block of command M98 has addresses other than O, N, P, H, and L. For G65, however, single-block stop will not occur.
- The level of local variables is fixed for M98, but for G65 does change according to the depth of nesting. (For example, #1s, if present before and after M98, always mean the same, but if present before and after G65, they have different meanings.)
- Command M98 can have up to a maximum of eight levels of call multiplexity when combined with G65, G66, or G66.1, whereas the maximum available number of levels for command G65 is four when it is combined with G66 or G66.1.

7. Multiplexity of macro call commands

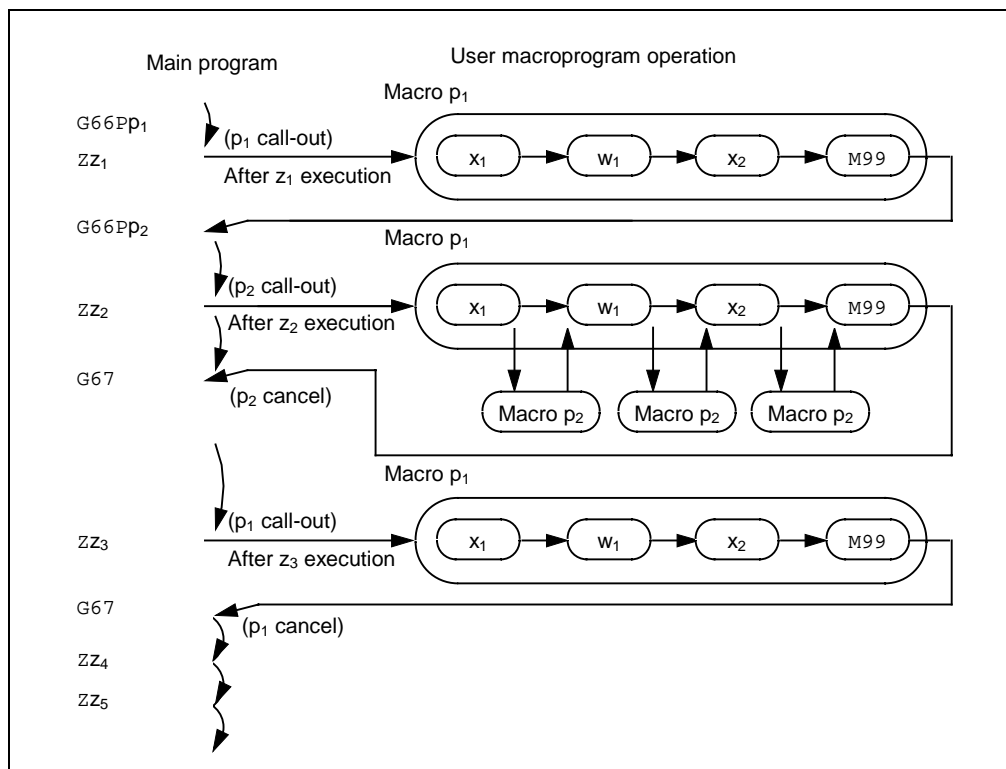
The maximum available number of levels of macro subprogram call is four, whether it is single or modal. Arguments in macro call instructions become valid only within the level of the called macro. Since the multiplexity of macro call is of up to a maximum of four levels, arguments can be included in a program as local variables each time a macro call is made.

Note 1: When a G65, G66, or G66.1 macro call or an auxiliary command macro call is made, nesting will be regarded as single-level and thus the level of local variables will also increase by 1.

Note 2: For modal call of type A, the designated user macro subprogram is called each time a move command is executed. If, however, multiple G66s are present, the next user macro subprogram will be called even for the move commands in the macro each time axis movement is done.

Note 3: User macro subprograms are cancelled in a reverse order to that in which they have been arranged.

Example:



8. User macro call based on interruption

Outline

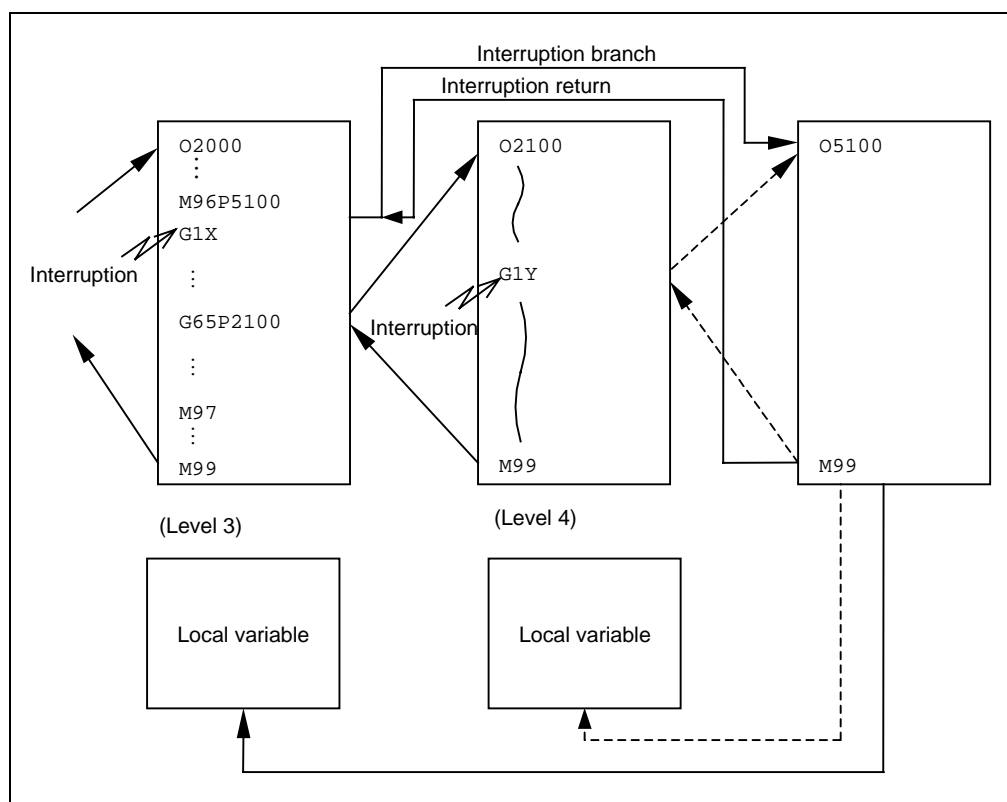
Prior creation of special user macros for interrupt processing allows the user macros to be executed during automatic operation when a user macro interrupt signal is input. After the user macro has been executed, the program can also be returned to the interrupted program block and then started from this block.

Detailed description

- Format for selecting the user macro branching destination

<pre> : M96P_L_ : : M97 (Branching mode off) : </pre>	}	<p>(Branching mode on)</p> <p>When user macroprogram interrupt signal is input during this space, the branch into the specified user macroprogram will be applied.</p>
-------------------------------------------------------	---	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- User macro interrupts can be processed even when the number of levels of macro call multiplexity during the occurrence of an interrupt is four. The local variables' level of the user macros used for interruption is the same as the level of the user macros existing during the occurrence of an interrupt.



13-10-3 Variables

Of all types of variables available for the NC unit, only local variables, common variables, and part of system variables are retained even after power-off.

1. Multiplexing of variables

Under user macro specifications, variables can have their identifiers (identification numbers) either transformed into variables, which is referred to as multiplexing, or replaced with <expression>.

For <expression>, only one arithmetic expression (for either multiplication, division, addition, or subtraction) can be used.

Example 1: Multiplexing variables

#1=10 #10=20 #20=30	From #1 = 10, #[[#1]] = #[#10] will result.
#5=#[[#1]]	From #10 = 20, #[#10] = #20 will result. Therefore #5 = #20, i.e. #5 = 30 will result.
#1=10 #10=20 #20=30	From #1 = 10, #[[#1]] = #[#10] will result.
#5=1000	From #10 = 20, #[#10] = #20 will result. Therefore #20 = #5, i.e.
#[[#1]]=#5	#20 = 1000 will result.

Example 2: Replacing variables identifiers with <expression>

#10=5	
#[#10+1]=1000	#6 = 1000 will result.
#[#10-1]=-1000	#4 = -1000 will result.
#[#10*3]=100	#15 = 100 will result.
#[#10/2]=100	#2 = -100 will result.

2. Undefined variables

Under user macro specifications, variables remaining unused after power-on or local variables that are not argument-specified by G65, G66, or G66.1 can be used as <empty>. Also, variables can be forcibly made into <empty>.

Variable #0 is always used as <empty> one, and this variable cannot be defined on the left side of the expression.

A. Arithmetic expression

```
#1=#0.....#1 = <empty>
#2=#0+1 .....#2 = 1
#3=1+#0 .....#3 = 1
#4=#0*10 .....#4 = 0
#5=#0+#0 .....#5 = 0
```

Note: Be careful that <empty> is handled the same as 0 during processing of expressions.

<empty> + <empty> = 0

<empty> + <constant> = constant

<constant> + <empty> = constant

B. Applying variables

Application of an undefined variable alone causes even the address to be ignored.

If #1 = <empty>

G0X#1Y1000 is equivalent to G0Y1000, and

G0X[#1+10]Y1000 is equivalent to G0X10Y1000.

C. Conditional expression

Only for EQ and NE, does <empty> differ from 0 in meaning.

If #101 = <empty>		If #101 = 0	
#101EQ#0	<empty> = <empty> holds.	#101EQ#0	0 = <empty> does not hold.
#101NE0	<empty> ≠ 0 holds.	#101NE0	0 ≠ 0 does not hold.
#101GE#0	<empty> ≥ <empty> holds.	#101GE#0	0 ≥ <empty> holds.
#101GT0	<empty> > 0 does not hold.	#101GT0	0 > 0 does not hold.

Hold-conditions and not-hold-conditions list

(For conditional expressions including undefined variables)

Right side Left side	EQ		NE		GT		LT		GE		LE	
	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant
Empty	H			H				H	H		H	
Constant			H		H				H			

H: Holds (The conditional expression holds.)

Blank: The conditional expression does not hold.

13-10-4 Types of variables

1. Common variables (#100 to #199, and #500 to #999)

Common variables refer to the variables to be used in common at any position. The identifiers of common variables which can be used are from #100 to #199, or from #500 to #999.

2. Local variables (#1 to #33)

Local variables refer to variables that can be defined as <argument> when calling a macro subprogram, or those which can be used locally within the main program or a subprogram. There is no relationship between macros. Thus, these variables can be overlapped on each other, but up to a maximum of four levels of overlapping.

$G65P_{p_1}L_{l_1} <argument>$

where p_1 : Program number

l_1 : Number of repeat times

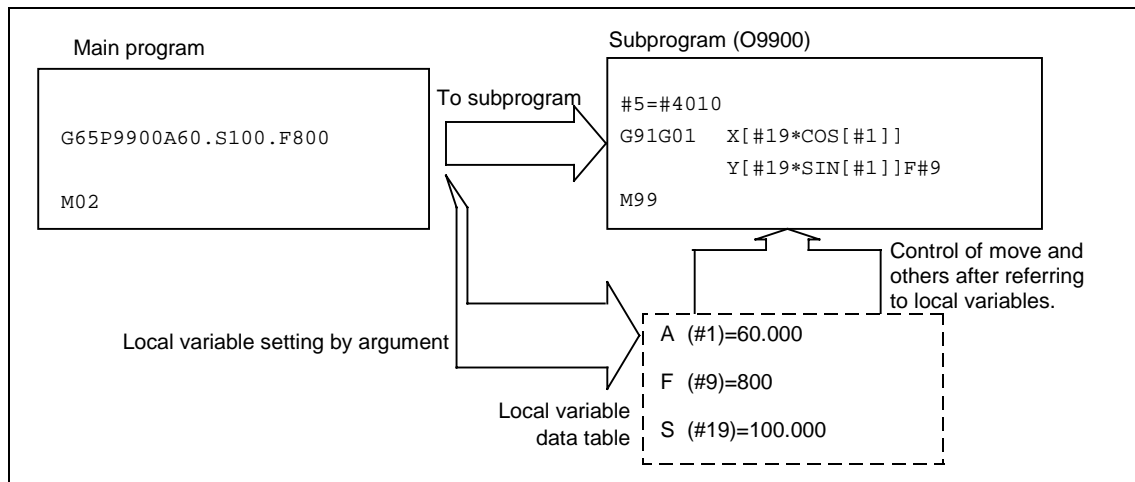
<Argument> must be: $Aa_1 Bb_1 Cc_1 \cdots Zz_1$.

The following represents the relationship between the address specified by <argument> and the local variables number used in the user macro unit:

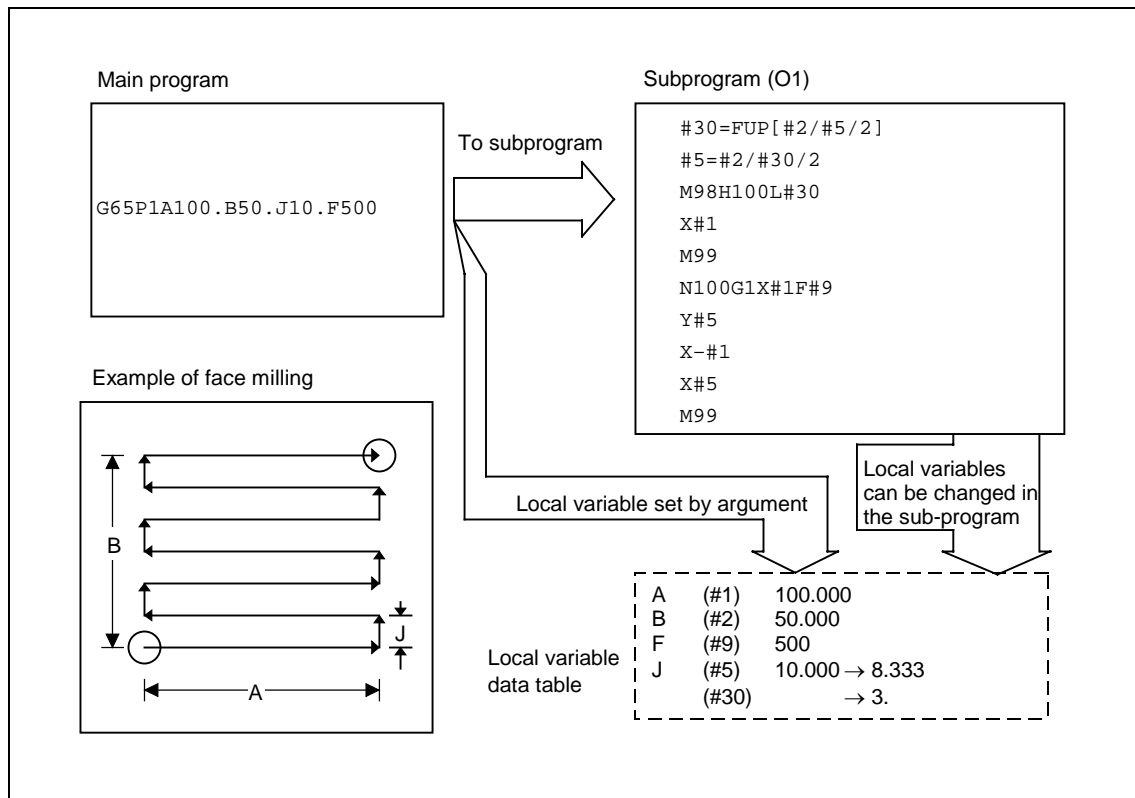
Call commands		Argument address	Local variable	Call commands		Argument address	Local variable
G65	G66.1			G65	G66.1		
○	○	A	#1	○	○	R	#18
○	○	B	#2	○	○	S	#19
○	○	C	#3	○	○	T	#20
○	○	D	#7	○	○	U	#21
○	○	E	#8	○	○	V	#22
○	○	F	#9	○	○	W	#23
x	x*	G	#10	○	○	X	#24
○	○	H	#11	○	○	Y	#25
○	○	I	#4	○	○	Z	#26
○	○	J	#5			–	#27
○	○	K	#6			–	#28
x	x*	L	#12			–	#29
○	○	M	#13			–	#30
x	x*	N	#14			–	#31
x	x	O	#15			–	#32
x	x*	P	#16			–	#33
○	○	Q	#17				

Argument addresses marked as x in the table above cannot be used. Only during the G66.1 mode, however, can argument addresses marked with an asterisk (*) in this table be additionally used. Also, the dash sign (–) indicates that no address is crosskeyed to the local variables number.

1. Local variables for a subprogram can be defined by specifying <argument> when calling a macro.



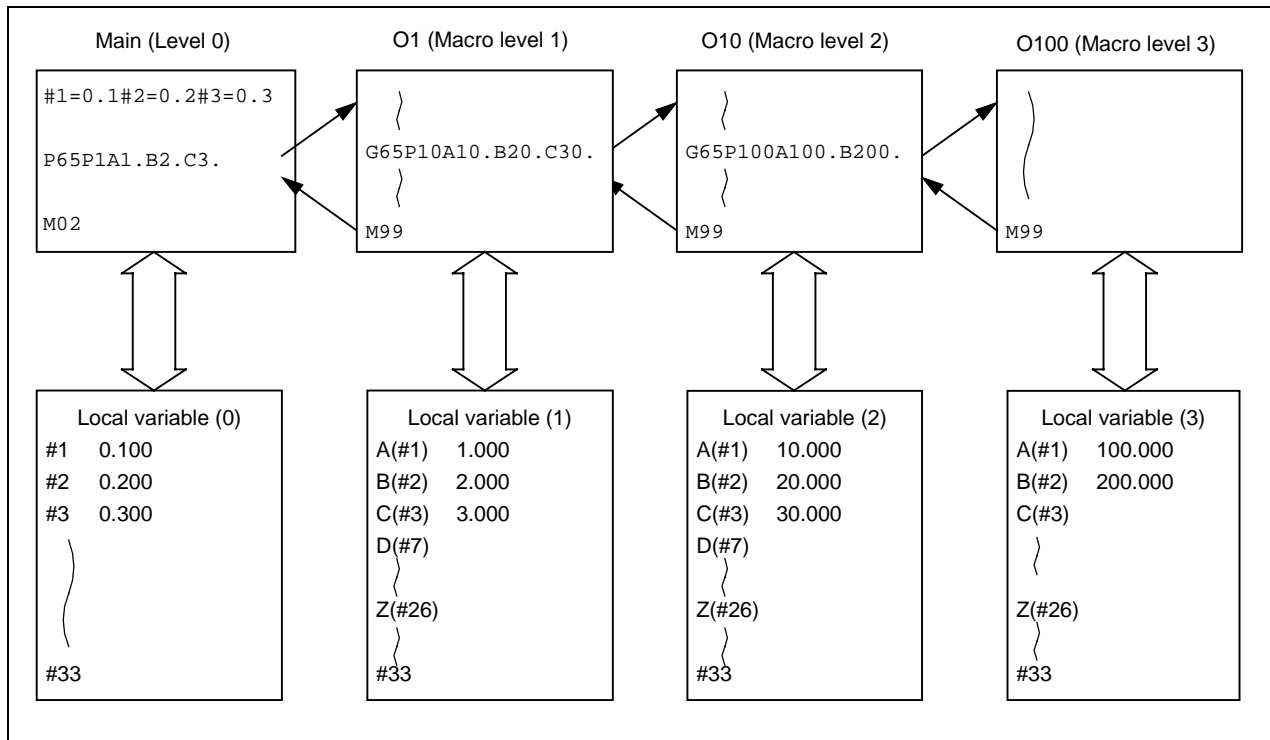
2. Within a subprogram, local variables can be freely used.



In the sample program for face-milling that is shown above, although the argument J has initially been programmed as a machining pitch of 10 mm, it has been changed into 8.333 mm to ensure equal-pitched machining.

Also, local variable #30 contains the calculated data about the number of times of reciprocal machining.

3. Local variables can be used for each of the four levels of macro call separately. For the main program (macro level 0), separate local variables are also provided. The local variables of level 0, however, cannot be designated with arguments.



How the local variables are currently being used is displayed on the screen.

For further details, refer to the Operating Manual.

3. Macro interface input system variables (#1000 to #1035)

You can check the status of an interface input signal by reading the value of the appropriate variables number (#1000 to #1035).

The read value of the variables number is either 1 (contact closed) or 0 (contact open). You can also check the status of all input signals of the variables from #1000 to #1031 by reading the value of variables number 1032. Variables from #1000 to #1035 can only be read; they cannot be placed on the left side of an arithmetic expression.

System variable	Points	Interface input signal	System variable	Points	Interface input signal
#1000	1	Register R72, bit 0	#1016	1	Register R73, bit 0
#1001	1	Register R72, bit 1	#1017	1	Register R73, bit 1
#1002	1	Register R72, bit 2	#1018	1	Register R73, bit 2
#1003	1	Register R72, bit 3	#1019	1	Register R73, bit 3
#1004	1	Register R72, bit 4	#1020	1	Register R73, bit 4
#1005	1	Register R72, bit 5	#1021	1	Register R73, bit 5
#1006	1	Register R72, bit 6	#1022	1	Register R73, bit 6
#1007	1	Register R72, bit 7	#1023	1	Register R73, bit 7
#1008	1	Register R72, bit 8	#1024	1	Register R73, bit 8
#1009	1	Register R72, bit 9	#1025	1	Register R73, bit 9
#1010	1	Register R72, bit 10	#1026	1	Register R73, bit 10
#1011	1	Register R72, bit 11	#1027	1	Register R73, bit 11
#1012	1	Register R72, bit 12	#1028	1	Register R73, bit 12
#1013	1	Register R72, bit 13	#1029	1	Register R73, bit 13
#1014	1	Register R72, bit 14	#1030	1	Register R73, bit 14
#1015	1	Register R72, bit 15	#1031	1	Register R73, bit 15

System variable	Points	Interface input signal
#1032	32	Register R72 and R73
#1033	32	Register R74 and R75
#1034	32	Register R76 and R77
#1035	32	Register R78 and R79

Note: The following interface input signals are used exclusively in the NC system operation (cannot be used for other purposes).

Interface input signal	Description
Register R72, bit 0	Touch sensor mounted in the spindle
Register R72, bit 4	X- and Y-axis machine lock ON
Register R72, bit 5	M-, S-, T-code lock ON
Register R72, bit 6	Z-axis machine lock ON

4. Macro interface output system variables (#1100 to #1135)

You can send an interface output signal by assigning a value to the appropriate variables number (#1100 to #1135).

All output signals can take either 0 or 1.

You can also send all output signals of the variables from #1100 to #1131 at the same time by assigning a value to variables number 1132. In addition to the data writing for offsetting the #1100 to #1135 output signals, the reading of the output signal status can be done.

System variable	Points	Interface output signal	System variable	Points	Interface output signal
#1100	1	Register R172, bit 0	#1116	1	Register R173, bit 0
#1101	1	Register R172, bit 1	#1117	1	Register R173, bit 1
#1102	1	Register R172, bit 2	#1118	1	Register R173, bit 2
#1103	1	Register R172, bit 3	#1119	1	Register R173, bit 3
#1104	1	Register R172, bit 4	#1120	1	Register R173, bit 4
#1105	1	Register R172, bit 5	#1121	1	Register R173, bit 5
#1106	1	Register R172, bit 6	#1122	1	Register R173, bit 6
#1107	1	Register R172, bit 7	#1123	1	Register R173, bit 7
#1108	1	Register R172, bit 8	#1124	1	Register R173, bit 8
#1109	1	Register R172, bit 9	#1125	1	Register R173, bit 9
#1110	1	Register R172, bit 10	#1126	1	Register R173, bit 10
#1111	1	Register R172, bit 11	#1127	1	Register R173, bit 11
#1112	1	Register R172, bit 12	#1128	1	Register R173, bit 12
#1113	1	Register R172, bit 13	#1129	1	Register R173, bit 13
#1114	1	Register R172, bit 14	#1130	1	Register R173, bit 14
#1115	1	Register R172, bit 15	#1131	1	Register R173, bit 15

System variable	Points	Interface output signal
#1132	32	Register R172 and R173
#1133	32	Register R174 and R175
#1134	32	Register R176 and R177
#1135	32	Register R178 and R179

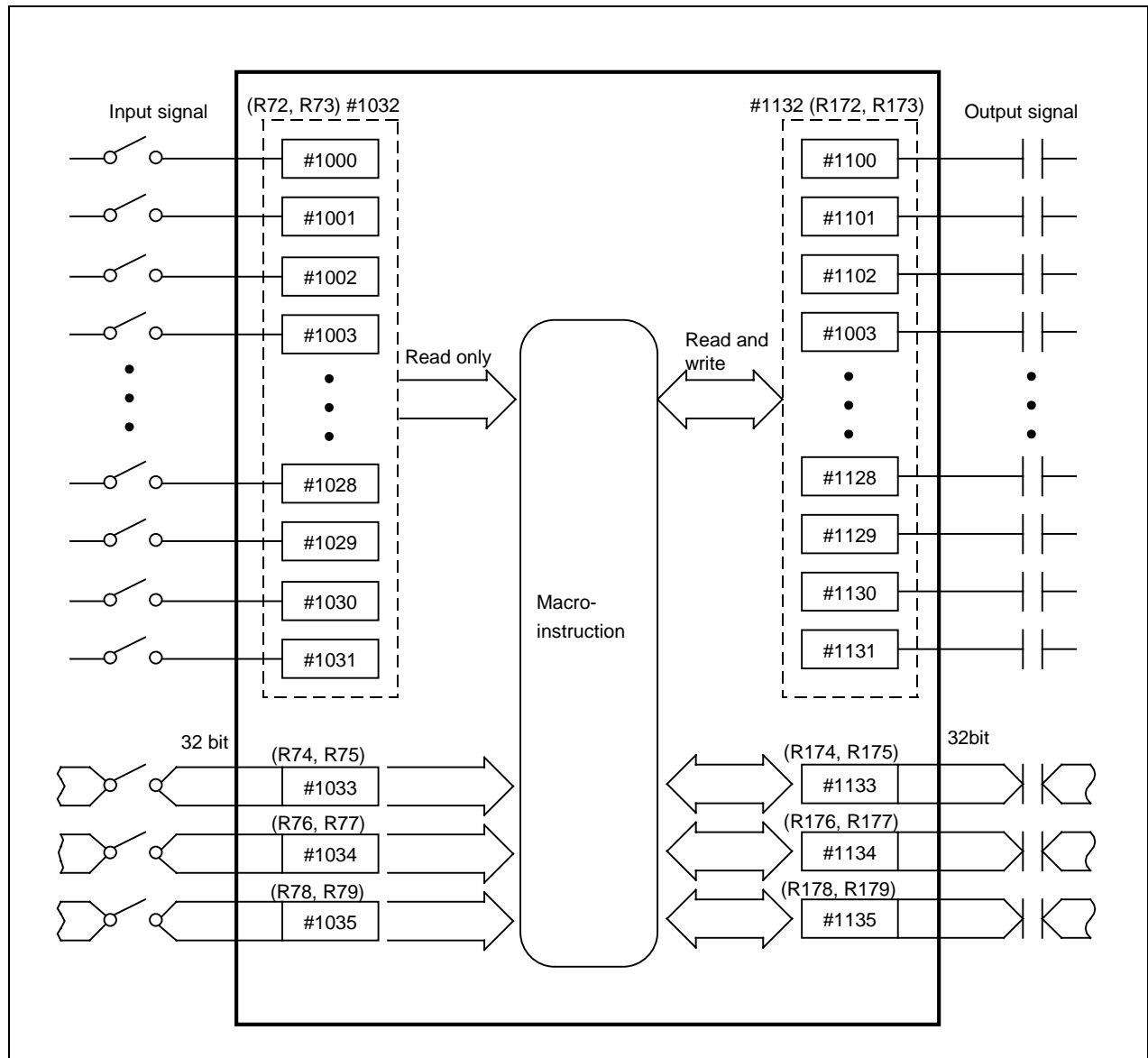
Note 1: Data of the system variables from #1100 to #1135 is saved according to the logical level (1 or 0) of the signal that has been lastly sent. The saved data is cleared by power-on/off automatically.

Note 2: The following applies if a data other than 1 or 0 is assigned to the variables from #1100 to #1131:

<empty> is regarded as equal to 0.

Data other than 0 and <empty> is regarded as equal to 1.

Data less than 0.00000001, however, is regarded as undefined.



5. Tool offset

Standard 128 sets
Optional 512 sets

Range of variable Nos.		Type A	Type B
#10001 to #10000+n	#2001 to #2000+n	○	○ Tool length geometric compensation.
#11001 to #11000+n	#2201 to #2200+n	×	○ Tool length wear compensation.
#16001 to #16000+n * (#12001 to #12000+n)	#2401 to #2400+n	×	○ Tool radius geometric compensation.
#17001 to #17000+n * (#13001 to #13000+n)	#2601 to #2600+n	×	○ Tool radius wear compensation.

*: The numbers of variables used for tool offset depend upon a parameter:

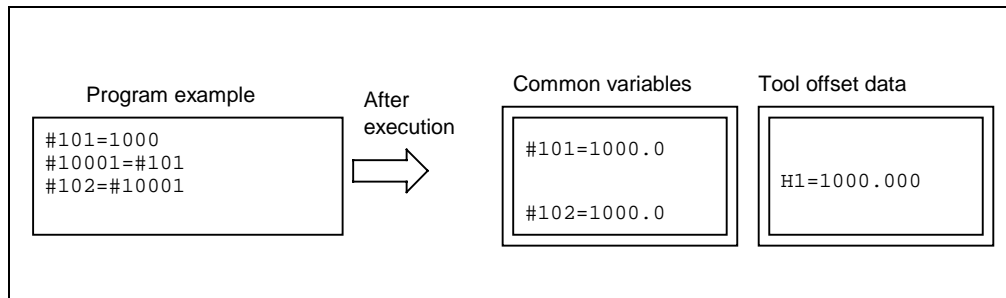
F96 bit 0 = 0: #16001 to #16000+n, and #17001 to #17000+n
= 1: #12001 to #12000+n, and #13001 to #13000+n.

Using variables numbers, you can read tool data or assign data.

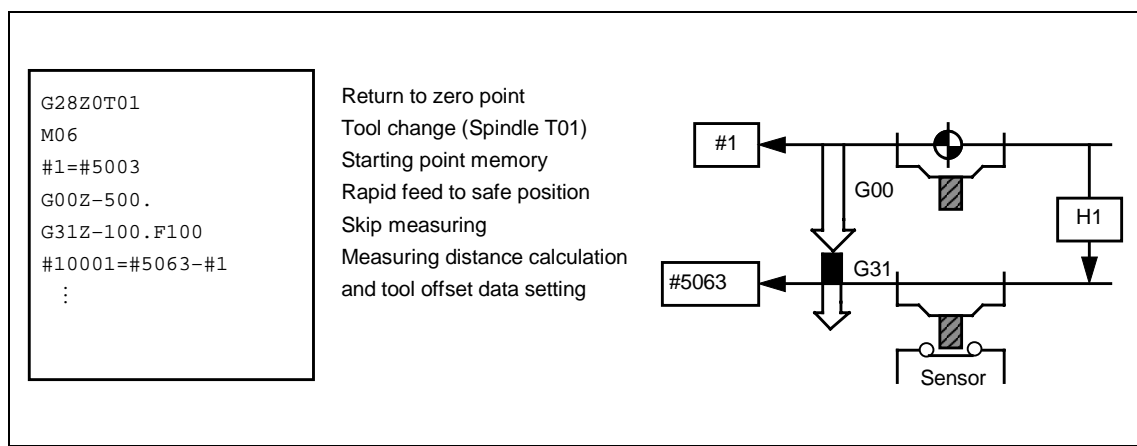
Usable variables numbers are of the order of either #10000 or #2000. For the order of #2000, however, only up to 200 sets of tool offsets can be used.

The last three digits of a variables number denote a tool offset number.

As with other variables, tool offset data is to contain the decimal point. The decimal point must therefore be included if you want to set data that has decimal digits.



Example: Tool offset data measuring



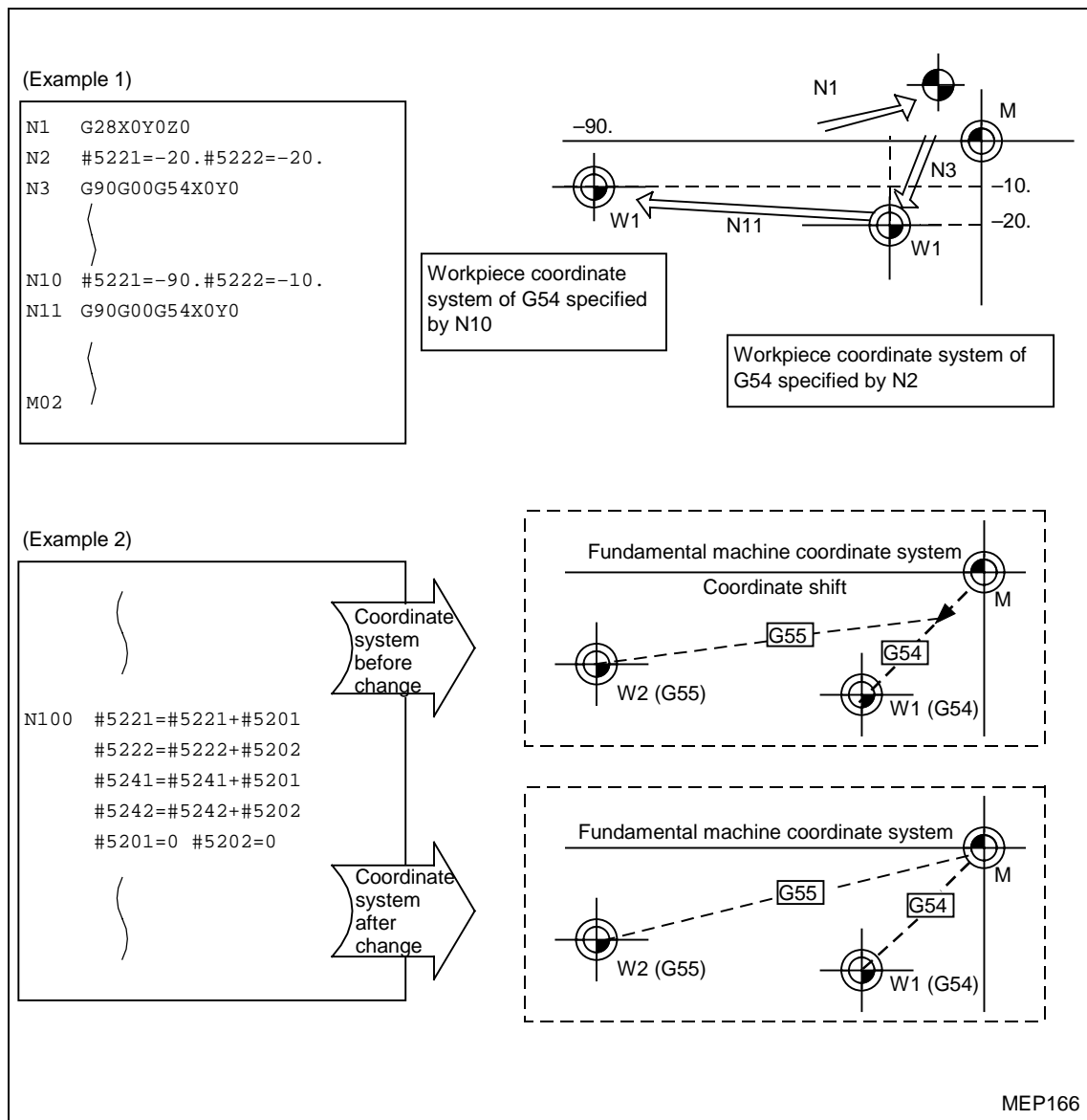
Note: The example shown above does not allow for any skip sensor signal delay. Also, #5003 denotes the position of the starting point of the Z-axis, and #5063 denotes the skip coordinate of the Z-axis, that is, the position at which a skip signal was input during execution of G31.

6. Workpiece coordinate system offset

Using variables numbers from 5201 to 5326, you can read workpiece coordinate system offset data or assign data.

Note: The number of controllable axes depends on the machine specifications.

Axis No. Coordinate name	1st axis	2nd axis	3rd axis	6th axis	Remarks
Coordinates shift (SHIFT)	#5201	#5202	#5203	#5206	An external data input/output optional spec. is required.
G54	#5221	#5222	#5223	#5226	A workpiece coordinate system offset feature is required.
G55	#5241	#5242	#5243	#5246	
G56	#5261	#5262	#5263	#5266	
G57	#5281	#5282	#5283	#5286	
G58	#5301	#5302	#5303	#5306	
G59	#5321	#5322	#5323	#5326	



The example 2 shown above applies only when coordinate shift data is to be added to the offset data of a workpiece coordinate system (G54 or G55) without changing the position of the workpiece coordinate system.

[Additional workpiece coordinate system offset]

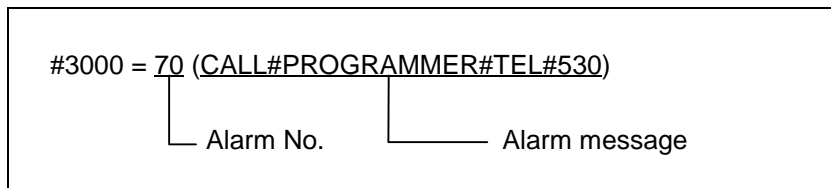
Variables numbered 7001 to 7946 can be used to read or assign additional workpiece coordinate system offsetting dimensions.

Note: The total number of controllable axes depends on the machine specifications.

Axis No. Coordinate name	1st axis	2nd axis	3rd axis	4th axis	5th axis	6th axis	Remarks
G54.1 P1	#7001	#7002	#7003	#7004	#7005	#7006	Only available with the optional function for additional coordinate system offset.
G54.1 P2	#7021	#7022	#7023	#7024	#7025	#7026	
G54.1 P3	#7041	#7042	#7043	#7044	#7045	#7046	
G54.1 P4	#7061	#7062	#7063	#7064	#7065	#7066	
G54.1 P48	#7941	#7942	#7943	#7944	#7945	#7946	

7. NC alarm (#3000)

The NC unit can be forced into an alarm status using variables number 3000.



The setting range for the alarm No. is from 1 to 6999.

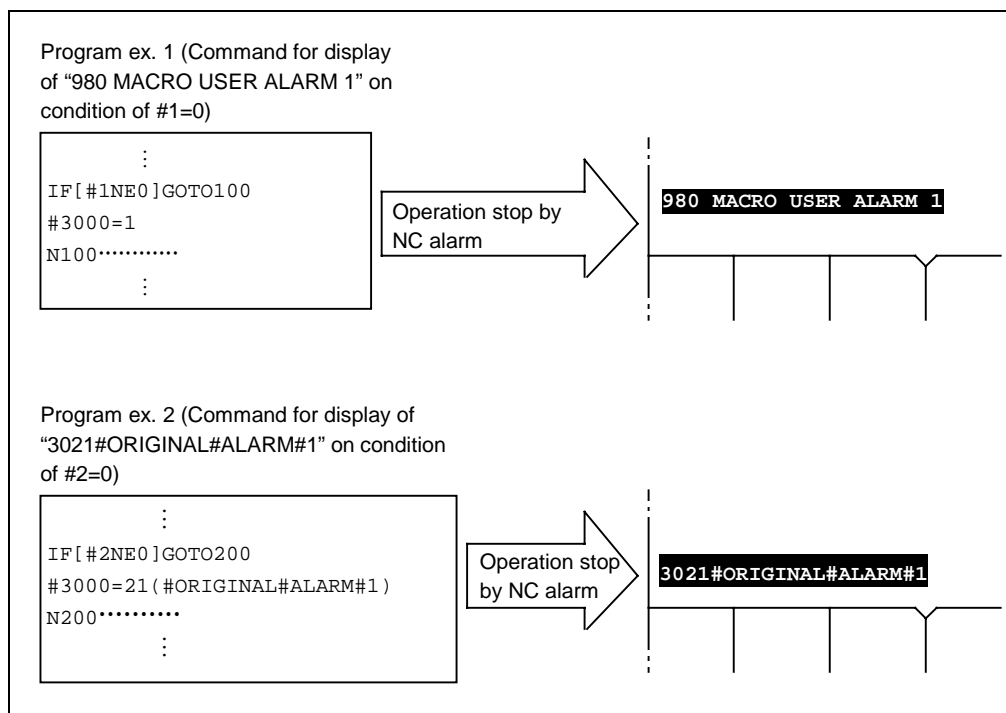
The maximum available length of the alarm message is 31 characters.

Note: The type of alarm message displayed on the screen depends on the designated alarm number, as indicated in the following table.

Designated alarm No.	Displayed alarm No.	Displayed alarm message
1 to 20	[Designated alarm No.] + 979	Message preset for the displayed alarm No. *1
21 to 6999	[Designated alarm No.] + 3000	Designated alarm message as it is *2

*1 Refers to alarm Nos. 980 to 999 whose messages are preset as indicated in Alarm List.

*2 Display of a message as it is set in the macro statement.

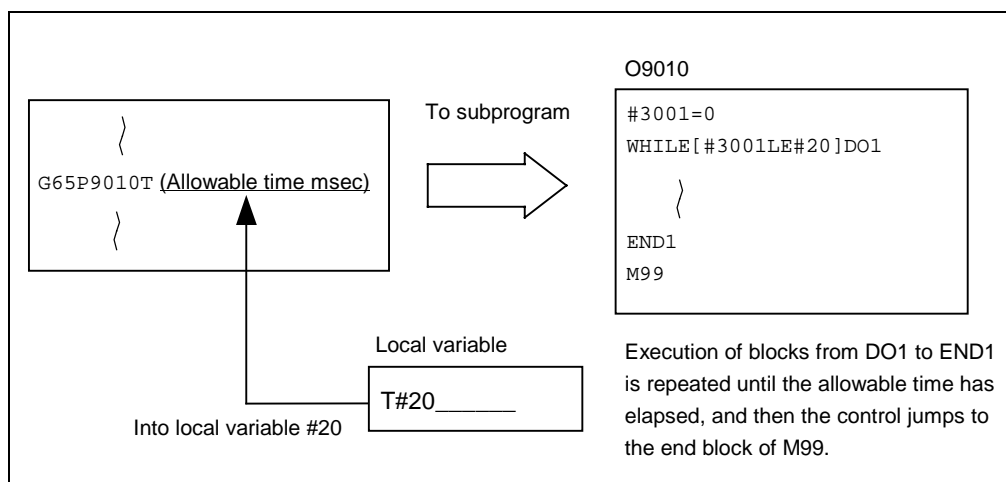


8. Integrated time (#3001, #3002)

Using variables #3001 and #3002, you can read the integrated time existing during automatic operation or assign data.

Type	Variable No.	Unit	Data at power-on	Initialization	Counting
Integrated time 1	3001	msec	Same as at power-off	Data is assigned in variables.	Always during power-on
Integrated time 2	3002				During auto-starting

The integrated time is cleared to 0 after having reached about 2.44×10^{11} msec (about 7.7 years).



9. Validation/invalidation of single-block stop or auxiliary-function finish signal wait (#3003)

Assigning one of the values listed in the table below to variables number 3003 allows single-block stop to be made invalid at subsequent blocks or the program to be advanced to the next block without ever having to wait for the arrival of an auxiliary-function code (M, S, T, or B) execution finish signal (FIN).

#3003	Single block stop	Auxiliary-function completion signal
0	Effective	Wait
1	Ineffective	Wait
2	Effective	No wait
3	Ineffective	No wait

Note: Variable #3003 is cleared to 0 by resetting.

10. Validation/invalidation of feed hold, feed rate override, or G09 (#3004)

Feed hold, feed rate override, or G09 can be made valid or invalid for subsequent blocks by assigning one of the values listed in the table below to variables number 3004.

#3004	Bit 0	Bit 1	Bit 2
Contents (Value)	Feed hold	Feed rate override	G09 check
0	Effective	Effective	Effective
1	Ineffective	Effective	Effective
2	Effective	Ineffective	Effective
3	Ineffective	Ineffective	Effective
4	Effective	Effective	Ineffective
5	Ineffective	Effective	Ineffective
6	Effective	Ineffective	Ineffective
7	Ineffective	Ineffective	Ineffective

Note 1: Variable #3004 is cleared to 0 by resetting.

Note 2: Each of the listed bits makes the function valid if 0, or invalid if 1.

11. Program stop (#3006)

Use of variables number 3006 allows the program to be stopped after execution of the immediately preceding block.

Format:

#3006 = 1 (CHECK OPERAT)

Character string to be displayed

Additional setting of a character string (in 29 characters at maximum) in parentheses allows the required stop message to be displayed on the monitor.

12. Mirror image (#3007)

The mirror image status of each axis at one particular time can be checked by reading variables number 3007.

Variable #3007 has its each bit crosskeyed to an axis, and these bits indicate that:

If 0, the mirror image is invalid.

If 1, the mirror image is valid.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Axis no.											6	5	4	3	2	1

13. G-command modal status

The G-command modal status in a pre-read block can be checked using variables numbers from 4001 to 4021. For variables numbers from #4201 to #4221, the modal status of the block being executed can be checked in a similar manner to that described above.

Variable Nos.		Function	
Block pre-read	Block executed		
#4001	#4201	Interpolation mode	G00-G03:0-3, G2.1:2.1, G3.1:3.1, G33:33
#4002	#4202	Plane selection	G17:17, G18:18, G19:19
#4003	#4203	Absolute/incremental	G90:90, G91:91
#4004	#4204	Programmed software limit	G22:22, G23:23
#4005	#4205	Feed specification	G94:94, G95:95
#4006	#4206	Inch/metric	G20:20, G21:21
#4007	#4207	Tool diameter offset	G40:40, G41:41, G42:42
#4008	#4208	Tool length offset	G43:43, G44:44, G49:49
#4009	#4209	Fixed cycle	G80:80, G73/74:73/74, G76:76, G81-G89:81-89
#4010	#4210	Return level	G98:98, G99:99
#4011	#4211	Scaling	G50:50, G51:51
#4012	#4212	Workpiece coordinate system	G54-G59:54-59, G54.1:54.1
#4013	#4213	Acceleration/Deceleration	G61-G64:61-64
#4014	#4214	Macro modal call	G66:66, G66.1: 66.1, G67:67
#4015	#4215	Shaping	G40.1:40.1, G41.1:41.1, G42.1:42.1
#4016	#4216	Coordinate rotation	G68:68, G69:69
#4019	#4219	Mirror image	G50.1:50.1, G51.1:51.1
#4020	#4220		
#4021	#4221		

14. Other modal information

Modal information about factors other than the G-command modal status in a pre-read block can be checked using variables numbers from 4101 to 4130. For variables numbers from #4301 to #4330, the modal information of the block being executed can be checked in a similar manner to that described above.

Variable Nos.		Modal information	Variable Nos.		Modal information
Preread	Execution		Preread	Execution	
#4101	#4301		#4112	#4312	
#4102	#4302	No. 2 miscellaneous function...B	#4113	#4313	Miscellaneous function...M
#4103	#4303		#4314	#4114	Sequence No...N
#4104	#4304		#4115	#4315	Program No...O
#4105	#4305		#4116	#4316	
#4106	#4306		#4117	#4317	
#4107	#4307	Tool diameter offset No...D	#4118	#4318	
#4108	#4308		#4119	#4319	Spindle function...S
#4109	#4309	Feed rate...F	#4120	#4320	Tool function...T
#4110	#4310		#4130	#4330	Addt. Workpiece coordinate system G54-G59:0, G54.1P1-P48:1-48
#4111	#4311	Tool length offset No...H			

15. Position information

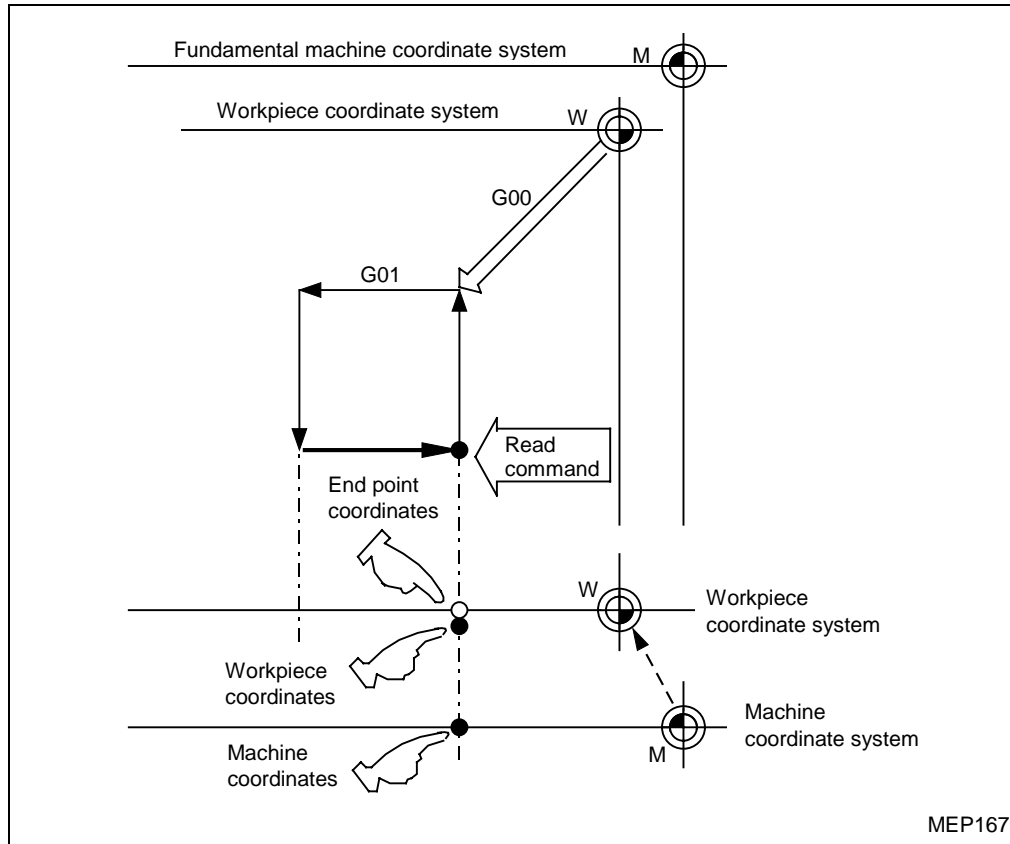
Using variables numbers from #5001 to #5106, you can check the ending-point coordinates of the previous block, machine coordinates, workpiece coordinates, skip coordinates, tool position offset coordinates, and servo deviations.

Position information Axis. No.	End point coordinates of precedent	Machine coordinate	Workpiece coordinate	Skip coordinate	Tool position offset coordinates	Servo deviation amount
1	#5001	#5021	#5041	#5061	#5081	#5101
2	#5002	#5022	#5042	#5062	#5082	#5102
3	#5003	#5023	#5043	#5063	#5083	#5103
6	#5006	#5026	#5046	#5066	#5086	#5106
Remarks (Reading during move)	Possible	Impossible	Impossible	Possible	Impossible	Possible

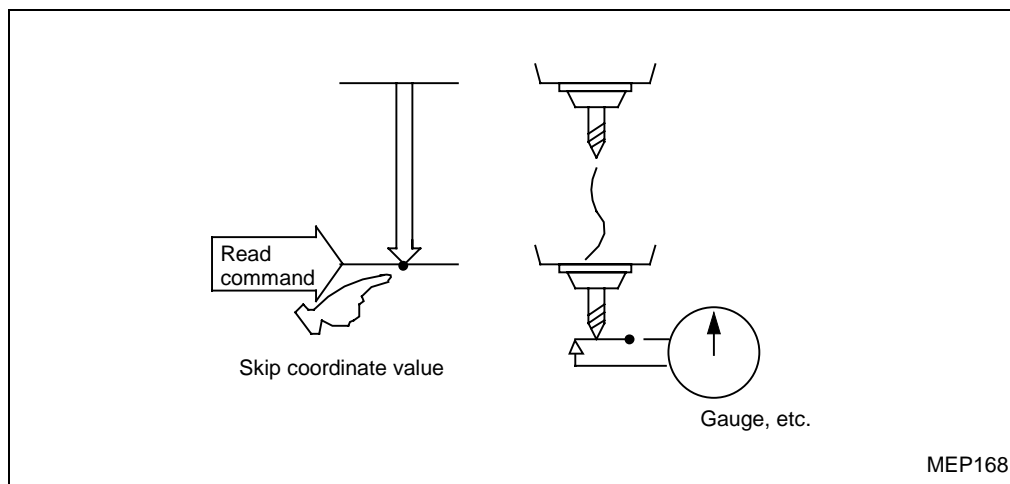
Note: The number of controllable axes depends on the machine specifications.

1. The ending-point coordinates and skip coordinates read will be those related to the workpiece coordinate system.

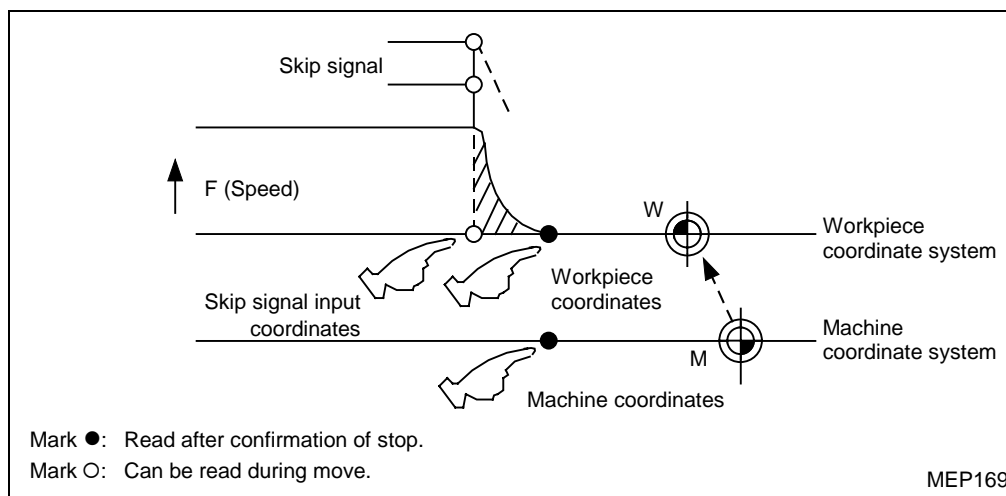
2. Ending-point coordinates, skip coordinates, and servo deviations can be checked even during movement. Machine coordinates, workpiece coordinates, and tool position offset coordinates must be checked only after movement has stopped.



3. Skip coordinates denote the position at which a skip signal has turned on at the block of G31. If the skip signal has not turned on, skip coordinates will denote the corresponding ending-point position.



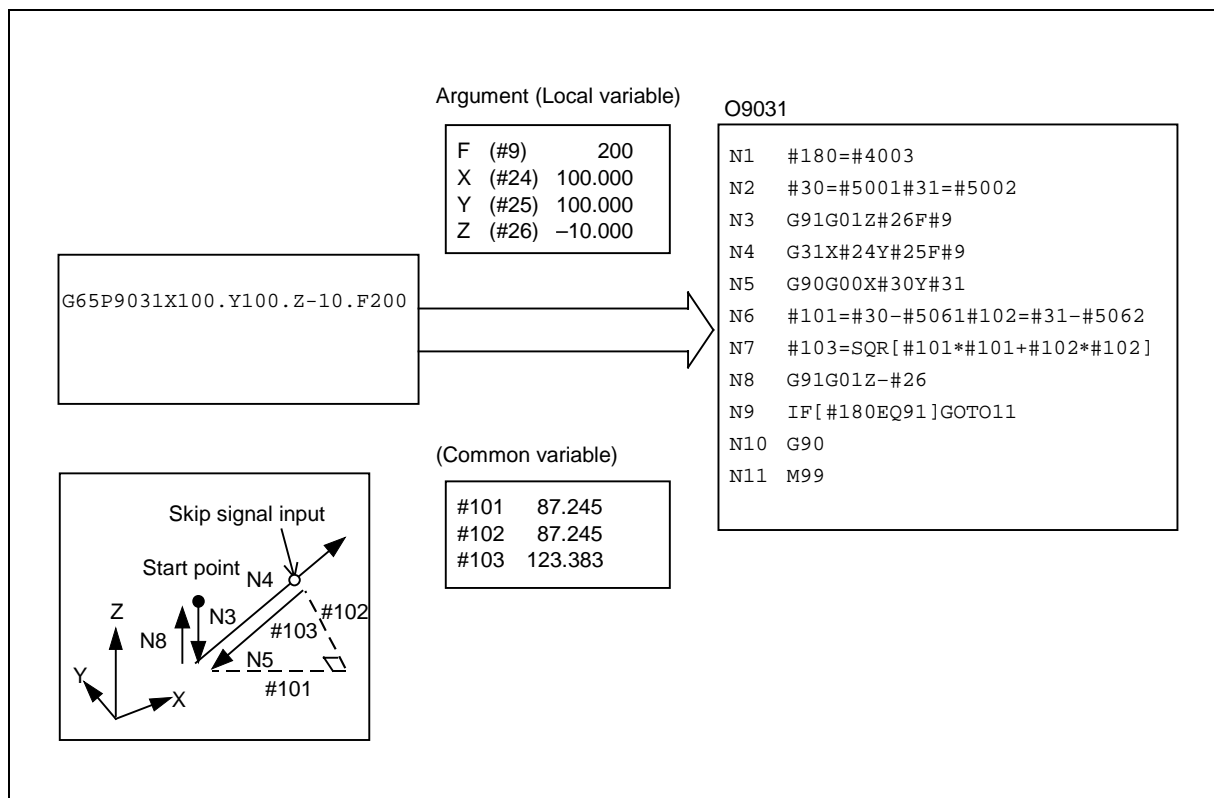
4. The ending-point position denotes the tool tip position which does not allow for any tool offsets, whereas machine coordinates, workpiece coordinates, and skip coordinates denote the tool reference-point position which allows for tool offsets.



The input coordinates of a skip signal denote the position within the workpiece coordinate system. The coordinates stored in variables from #5061 to #5066 are those existing when skip signals were input during movement of the machine. These coordinates can therefore be read at any time after that. See the section (Chapter 16) on skip functions for further details.

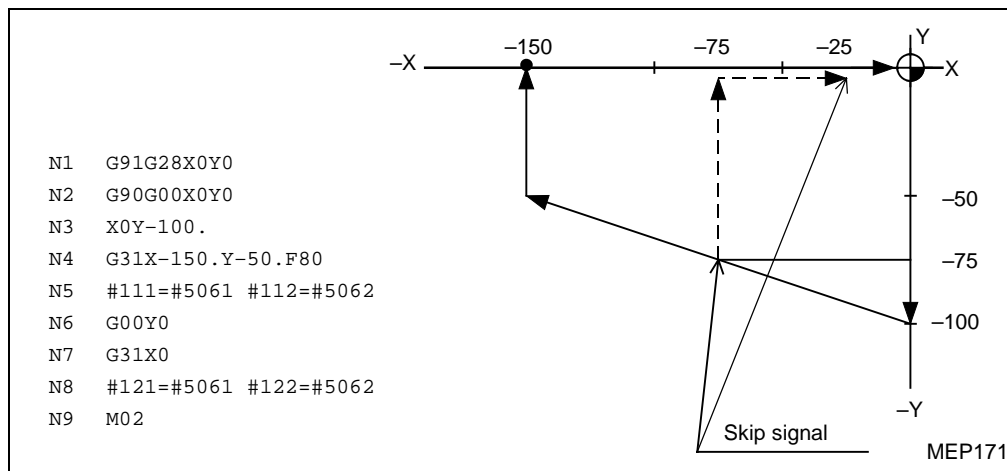
Example 1: Workpiece position measurement:

The following shows an example of measuring the distance from a reference measurement point to the workpiece end:



#101	X-axis measuring amount	N1	Modal data storage of G90/G91
#102	Y-axis measuring amount	N2	X, Y starting point data storage
#103	Measuring line linear amount	N3	Z-axis entry
#5001	X-axis measuring start point	N4	X, Y measuring (Stop at skip input)
#5002	Y-axis measuring start point	N5	Return to X, Y starting point
#5061	X-axis skip input point	N6	X, Y measuring incremental data calculation
#5062	Y-axis skip input point	N7	Measuring line linear amount calculation
		N8	Z-axis escape
		N9, N10	Modal return of G90/G91
		N11	Return from subprogram

Example 2: Skip input coordinates reading:



$$\begin{aligned} \#111 &= -75. + \varepsilon & \#112 &= -75. + \varepsilon \\ \#121 &= -25. + \varepsilon & \#122 &= -75. + \varepsilon \end{aligned}$$

where ε denotes an error due to response delay. (See Chapter 16 on skip functions for further details.)

Variable #122 denotes the skip signal input coordinate of N4 since N7 does not have a Y-command code.

16. Spindle tool (#51999)

Using variables number 51999, you can check the tool number of the tool mounted in the spindle.

System variable	Description
#51999	Spindle tool number

Note: This system variable is a read-only variable.

17. MAZATROL program basic coordinate systems

- Using variables numbers from #5341 to #5347, you can check the basic coordinate data valid during execution of a MAZATROL program or assign data to those variables.

System variable	Coordinate axis name	System variable	Coordinate axis name
#5341	WPC-X	#5344	WPC-4
#5342	WPC-Y	#5345	WPC-5
#5343	WPC-Z	#5347	WPC-th

Note 1: Data of the basic coordinate unit of a MAZATROL program will not be updated even when data is assigned to the system variables listed above. The original coordinate system will therefore become valid for the succeeding units. Refer to Section 14-13 for details of updating procedures.

Note 2: The data written into the system variables will not become valid until the program operation is stopped (in single-block operation mode, at M00, etc.).

Note 3: In continuous operation mode, the execution conditions may not have been punctually prepared for a movement-command block which immediately succeeds one for data writing into the system variables. Do not fail, therefore, to set a stop command after the data writing block as follows to ensure optimal conditions:

```
#5341=#5341-20.
```

```
M00
```

Note 4: Use variables numbers from #5351 to #5353, as indicated below, to read the basic coordinates data specified in the plane definition sequence for a plane angle (ANGLE) other than 0°.

- Variables numbers from #5351 to #5353 are to be used to read the basic coordinates data specific to a plane definition unit.

System variable	Description
#5351	X (origin defined in the DEF. FACE sequence)
#5352	Y (origin defined in the DEF. FACE sequence)
#5353	Z (origin defined in the DEF. FACE sequence)

18. MAZATROL tool data

MAZATROL tool data can be checked (or assigned) using the following variables numbers:

Usable variables numbers	MAZATROL tool data
#60001 to #60000 + n	Tool length
#61001 to #61000 + n	Tool diameter
#62001 to #62000 + n	Tool life flag
#63001 to #63000 + n	Tool damage flag

Tool quantity (n): 960 (maximum)

$1 \leq n \leq 960$ (The maximum applicable tool quantity depends on the machine specifications.)

Note 1: During tool path check, tool data can be checked but cannot be assigned.

Note 2: Tool life flags (variables numbers of the order of #62000) and tool damage flags (likewise, the order of #63000) can take either 1 or 0 as their logical states (1 for ON, 0 for OFF).

19. EIA/ISO tool data

Using variables numbers tabulated below, EIA/ISO tool data (tool life management data) can be read or updated, as required.

Tool quantity (n): 960 (maximum)

$1 \leq n \leq 960$

System variables	Corresponding data
#40001 to #40000 + n	Tool length offset numbers or tool length offset amounts
#41001 to #41000 + n	Tool diameter offset numbers or tool diameter offset amounts
#42001 to #42000 + n	Tool life flags
#43001 to #43000 + n	Tool damage flags
#44001 to #44000 + n	Tool data flags
#45001 to #45000 + n	Tool operation time (sec)
#46001 to #46000 + n	Tool life time (sec)

Note 1: During tool path check, tool data can be checked but cannot be assigned.

Note 2: Tool life flags (variables numbers of the order of #42000) and tool damage flags (likewise, the order of #43000) can take either 1 or 0 as their logical states (1 for ON, 0 for OFF)

Note 3: The identification between number and amount of tool length or diameter offset is made by referring to the tool data flag.

Tool data flag	bit 0	bit 1	bit 2	bit 3
Length offset No.	0	0	—	—
Length offset amount	0	1	—	—
Diam. offset No.	—	—	0	0
Diam. offset amount	—	—	0	1

20. Date and time (Year-month-day and hour-minute-second)

Variables numbered 3011 and 3012 can be used to read date and time data.

Variable Nos.	Description
#3011	Date (Year-month-day)
#3012	Time (Hour-minute-second)

Example: If the date is December 15, 1995 and the time is 16:45:10, data is set as follows in the corresponding system variables:

#3011 = 951215

#3012 = 164510.

21. Total number of machined parts and the number of parts required

Variables numbered 3901 and 3902 can be used to read or assign the total number of machined parts and the number of parts required.

Variable Nos.	Description
#3901	Total number of machined parts
#3902	Number of parts required

Note 1: These variables must be integers from 0 to 9999.

Note 2: Data reading and writing by these variables is surely suppressed during tool path checking.

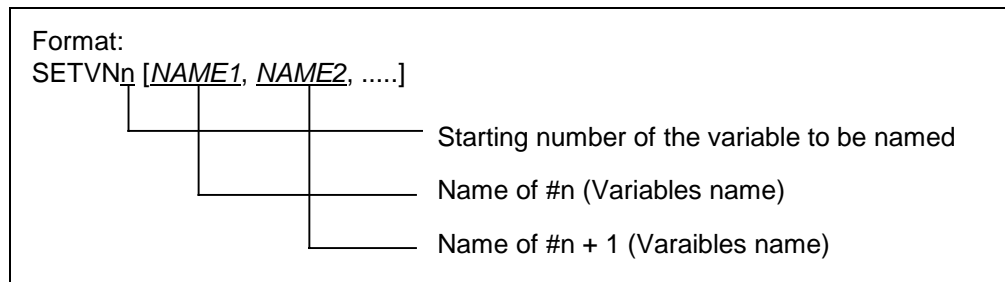
22. Parameter for shaping (#1900)

The value of parameter **K1** (distance from the C-axis to the nose of the tool) can be read or updated, as required, using variables number 1900.

Variable No.	Description	Setting range
#1900	Parameter K1 (Rotational radius of the C-axis)	0 to 9999.999 mm 0 to 999.99999 in.

23. Setting and using variables names

Any variables name can be assigned to each of common variables #500 through #519. The variables name, however, must be of seven alphanumeric or less that begin with a letter of the alphabet.



Each variables name must be separated using the comma (,).

Detailed description

- Once a variables name has been set, it remains valid even after power-off.
- Variables in a program can be called using the variables names. The variable to be called must, however, be enclosed in brackets ([]).

Example: G01X[#POINT1]
[#TIMES]=25

- Variables names can be checked on the **USER PARAMETER - EIA/ISO** display. The names assigned to variables #500 to #519 are displayed at F47 to F66.

Example: Program SETVN500[ABC,EFG]

On the display

```

F46  0
F47  ABC  ←  Variables name assigned to #500
F48  EFG  ←  Variables name assigned to #501
F49           ←  Variables name assigned to #502
F50

```

13-10-5 Arithmetic operation commands

Various operations can be carried out between variables using the following format.

#i = <expression>

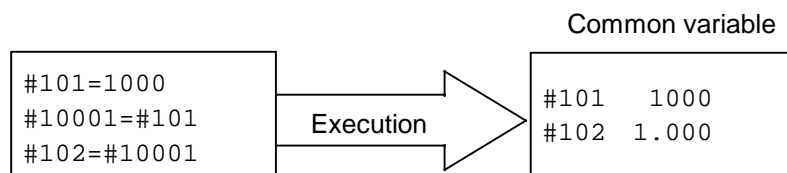
where <expression> must consist of a constant(s), a variable(s), a function(s), or an operator(s). In the table given below, constants can be used instead of #j and/or #k.

[1] Definition/replacement of variables	#i=#j	Definition/replacement
[2] Additional-type operations	#i=#j+#k	Addition
	#i=#j-#k	Subtraction
	#i=#jOR#k	Logical addition (For each of 32 bits)
	#i=#jXOR#k	Exclusive OR (For each of 32 bits)
[3] Multiplicative-type operations	#i=#j*#k	Multiplication
	#i=#j/#k	Division
	#i=#jMOD#k	Surplus
	#i=#jAND#k	Logical product (For each of 32 bits)
[4] Functions	#i=SIN[#k]	Sine
	#i=COS[#k]	Cosine
	#i=TAN[#k]	Tangent (tanq is used as sinq/cosq.)
	#i=ATAN[#j]	Arc-tangent (Either ATAN or ATN can be used.)
	#i=ACOS[#j]	Arc-cosine
	#i=SQRT[#k]	Square root (Either SQRT or SQR is available.)
	#i=ABS[#k]	Absolute value
	#i=BIN[#k]	BINARY conversion from BCD
	#i=BCD[#k]	BCD conversion from BINARY
	#i=ROUND[#k]	Rounding to the nearest whole number (Either ROUND or RND is available.)
	#i=FIX[#k]	Cutting away any decimal digits
	#i=FUP[#k]	Counting any decimal digits as 1s
	#i=LN[#k]	Natural logarithm
	#i=EXP[#k]	Exponent with the base of e (= 2.718 ...)

Note 1: In principle, data without a decimal point is handled as data that has a decimal point.
(Example: 1 = 1.000)

Note 2: Offsets from variable #10001, workpiece coordinate system offsets from variable #5201, and other data become data that has a decimal point. If data without a decimal point is defined using these variables numbers, therefore, a decimal point will also be assigned to the data.

Example:



Note 3: The <expression> after a function must be enclosed in brackets ([]).

1. Operation priority

Higher priority is given to functions, multiplicative operations, and additive operations, in that order.

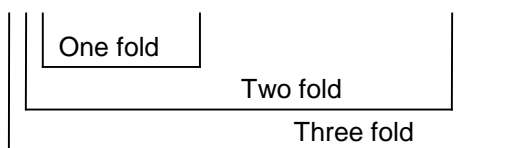
```
#101=#111+#112*SIN[#113]
```



2. Specifying an operational priority level

The part to which the first level of operation priority is to be given can be enclosed in brackets ([]). Up to five sets of brackets, including those of functions, can be used for one expression.

#101=SQRT[[[#111-#112]*SIN[#113]+#114]*#15]



3. Examples of operation instructions

[1] Main program and argument specification	G65 P100 A10 B20. #101=100.000 #102=200.000	#1 10.000 #2 20.000 #101 100.000 #102 200.000
[2] Definition, replacement =	#1=1000 #2=1000. #3=#101 #4=#102 #5=#5081	#1 1000.000 #2 1000.000 #3 100.000 #4 200.000 #5 -10.000 Data of common variables Offset amount
[3] Addition, subtraction + -	#11=#1+1000 #12=#2-50. #13=#101+#1 #14=#5081-3. #15=#5081+#102	#11 2000.000 #12 950.000 #13 1100.000 #14 -13.000 #15 190.000
[4] Logical addition OR	#3=100 #4=#3OR14	#3 = 01100100 14 = 00001110 #4 = 01101110 = 110
[5] Exclusive OR XOR	#3=100 #4=#3XOR14	#3 = 01100100 14 = 00001110 #4 = 01101010 = 106
[6] Multiplication, Division * /	#21=100*100 #22=100.*100 #23=100*100. #24=100.*100. #25=100/100 #26=100./100 #27=100/100. #28=100./100. #29=#5081*#101 #30=#5081/#102	#21 10000.000 #22 10000.000 #23 10000.000 #24 10000.000 #25 1.000 #26 1.000 #27 1.000 #28 1.000 #29 -1000.000 #30 -0.050
[7] Surplus MOD	#31=#19MOD#20	#19 = 48 #20 = 9 = 5 surplus 3
[8] Logical product AND	#9=100 #10=#9AND15	#9 = 01100100 15 = 00001111 #10 = 00000100 = 4
[9] Sine SIN	#501=SIN[60] #502=SIN[60.] #503=1000*SIN[60] #504=1000*SIN[60.] #505=1000.*SIN[60] #506=1000.*SIN[60.] Note: SIN[60] is equal to SIN[60.].	#501 0.866 #502 0.866 #503 866.025 #504 866.025 #505 866.025 #506 866.025
[10] Cosine COS	#541=COS[45] #542=COS[45.] #543=1000*COS[45] #544=1000*COS[45.] #545=1000.*COS[45] #546=1000.*COS[45.] Note: COS[45] is equal to COS[45.].	#541 0.707 #542 0.707 #543 707.107 #544 707.107 #545 707.107 #546 707.107

[11] Tangent TAN	#551=TAN[60]	#551	1.732
	#552=TAN[60.]	#552	1.732
	#553=1000*TAN[60]	#553	1732.051
	#554=1000*TAN[60.]	#554	1732.051
	#555=1000.*TAN[60]	#555	1732.051
	#556=1000.*TAN[60.]	#556	1732.051
	Note: TAN[60] is equal to TAN[60.].		
[12] Arc-tangent ATAN	#561=ATAN[173205/1000000]	#561	60.000
	#562=ATAN[173.205/100.]	#562	60.000
	#563=ATAN[1.732]	#563	59.999
[13] Arc-cosine ACOS	#521=ACOS[100000/141421]	#521	45.000
	#522=ACOS[100./141.421]	#522	45.000
	#523=ACOS[1000/1414.213]	#523	45.000
	#524=ACOS[10./14.142]	#524	44.999
	#525=ACOS[0.707]	#525	45.009
[14] Square root SQRT	#571=SQRT[1000]	#571	31.623
	#572=SQRT[1000.]	#572	31.623
	#573=SQRT[10.*10.+20.*20.]	#573	22.361
	#574=SQRT[#14*#14+#15*#15]	#574	190.444
	Note: For enhanced accuracy, perform operations within [] as far as possible.		
[15] Absolute value ABS	#576=-1000	#576	-1000.000
	#577=ABS[#576]	#577	1000.000
	#3=70.		
	#4=-50.		
	#580=ABS[#4-#3]	#580	120.000
[16] BIN, BCD	#1=100		
	#11=BIN[#1]	#11	64
	#12=BCD[#1]	#12	256
[17] Rounding into the nearest whole number ROUND	#21=ROUND[14/3]	#21	5
	#22=ROUND[14./3]	#22	5
	#23=ROUND[14/3.]	#23	5
	#24=ROUND[14./3.]	#24	5
	#25=ROUND[-14/3]	#25	-5
	#26=ROUND[-14./3]	#26	-5
	#27=ROUND[-14/3.]	#27	-5
	#28=ROUND[-14./3.]	#28	-5
[18] Cutting away any decimal digits FIX	#21=FIX[14/3]	#21	4.000
	#22=FIX[14./3]	#22	4.000
	#23=FIX[14/3.]	#23	4.000
	#24=FIX[14./3.]	#24	4.000
	#25=FIX[-14/3]	#25	-4.000
	#26=FIX[-14./3]	#26	-4.000
	#27=FIX[-14/3.]	#27	-4.000
	#28=FIX[-14./3.]	#28	-4.000
[19] Counting any decimal digits as 1s FUP	#21=FUP[14/3]	#21	5.000
	#22=FUP[14./3]	#22	5.000
	#23=FUP[14/3.]	#23	5.000
	#24=FUP[14./3.]	#24	5.000
	#25=FUP[-14/3]	#25	-5.000
	#26=FUP[-14./3]	#26	-5.000
	#27=FUP[-14/3.]	#27	-5.000
	#28=FUP[-14./3.]	#28	-5.000
[20] Natural logarithm LN	#101=LN[5]	#101	1.609
	#102=LN[0.5]	#102	-0.693
	#103=LN[-5]	Alarm 860 CALCULATION IMPOSSIBLE	
[21] Exponent EXP	#104=EXP[2]	#104	7.389
	#105=EXP[1]	#105	2.718
	#106=EXP[-2]	#106	0.135

4. Operation accuracy

The errors listed in the table below are generated by one arithmetic operation, and the error rate increases each time an operation is performed.

Operation format	Mean error	Max. error	Kind of error
$a = b + c$ $a = b - c$	2.33×10^{-10}	5.32×10^{-10}	Min. $\left \frac{\varepsilon}{c} \right $, $\left \frac{\varepsilon}{b} \right $
$a = b \cdot c$	1.55×10^{-10}	4.66×10^{-10}	Relative error $\left \frac{\varepsilon}{a} \right $
$a = b/c$	4.66×10^{-10}	1.86×10^{-9}	
$a = \sqrt{b}$	1.24×10^{-9}	3.73×10^{-9}	
$a = \sin b$ $a = \cos b$	5.0×10^{-9}	1.0×10^{-8}	Absolute error $\left \varepsilon \right $ degree
$a = \tan^{-1} b/c$	1.8×10^{-6}	3.6×10^{-6}	

Note: The function TAN (Tangent) is calculated as SIN/COS (Sine/Cosine).

5. Notes on deterioration of accuracy

A. Addition/subtraction

As for additional-type operations, if an absolute value is subtracted from the other, the relative error cannot be reduced below 10^{-8} .

For example, when the true values (such values, by the way, cannot be substituted directly) of #10 and #20 are as follows:

#10 = 2345678988888.888

#20 = 2345678901234.567

then #10 – #20 = 87654.321 would not result from calculation of #10 – #20. This is because, since the effective number of digits of the variable is eight (decimal), the approximate values of #10 and #20 are:

#10 = 2345679000000.000

#20 = 2345678900000.000

More strictly, internal binary values slightly differ from these values. Actually therefore, a significant error results as follows:

#10 – #20 = 100000.000.

B. Logical relationship

As for EQ, NE, GT, LT, GE and LE, the processing is executed in a similar manner to addition and subtraction, so be careful to errors. For example, to judge whether #10 is equal to #20 of the above example, the conditional expression

IF [#10EQ#20]

is not appropriate due to the errors. In such a case, therefore, give a macro-instruction as shown below to allow for an acceptable tolerance in the judgement on the equality of two values.

IF [ABS[#10 – #20] LT200000]

C. Trigonometric functions

For trigonometric functions, although the absolute error is guaranteed, the relative error is not below 10^{-8} . Be careful, therefore, when carrying out multiplication, or division after trigonometric function operations.

13-10-6 Control commands

The flow of a program can be controlled using IF ~ GOTO ~ and WHILE DO ~ commands.

1. Branching

Format: IF [conditional expression] GOTO n

where n is a sequence number in the same program.

The branching will occur to the block headed by sequence number ' n ' if the condition holds, or if the condition does not hold, the next block will be executed.

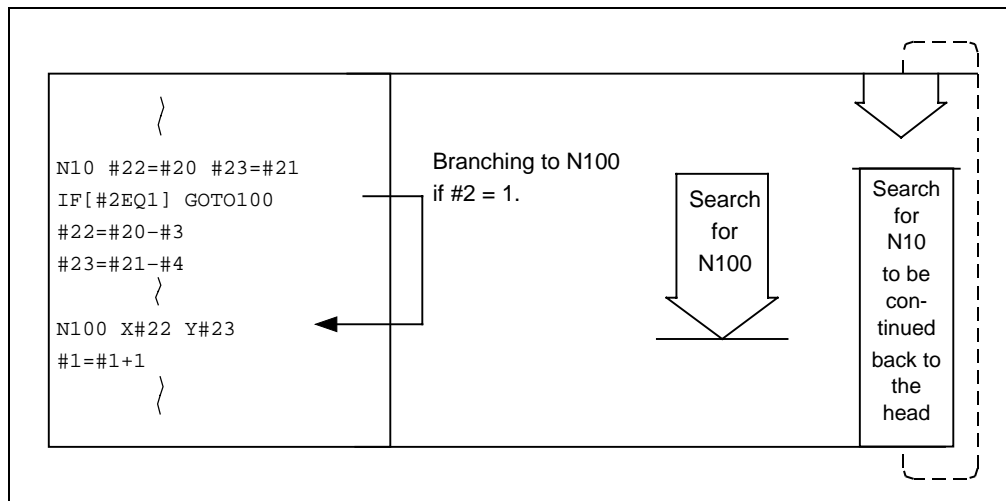
An independent setting of GOTO statement without IF [conditional expression] will perform unconditional branching to the specified block.

The [conditional expression] consists of the following six types:

#i EQ #j	=	(#i is equal to #j.)
#i NE #j	≠	(#i is not equal to #j.)
#i GT #j	>	(#i is larger than #j.)
#i LT #j	<	(#i is smaller than #j.)
#i GE #j	≥	(#i is equal to #j, or larger than #j.)
#i LE #j	≤	(#i is equal to #j, or smaller than #j.)

For GOTO n, n must be a sequence number within the same program. If the sequence number does not exist in that program, an alarm **843 DESIGNATED SNo. NOT FOUND** will occur. An expression or a variable can be used instead of #i, #j, or n.

Sequence number designation Nn must be set at the beginning of the destination block. Otherwise, an alarm **843 DESIGNATED SNo. NOT FOUND** will result. If, however, the block begins with "/" and Nn follows, the program can be branched into that sequence number.



Note: During search for the branching destination sequence number, if the area from the block after "IF ..." to the program end (% code) is searched (forward search) in vain, then the area from the head down to the block before "IF ..." will be searched next (backward search). It will therefore take more time to execute backward search (searching in the opposite direction to the flow of the program) than to execute forward search.

2. Looping

Format:

```

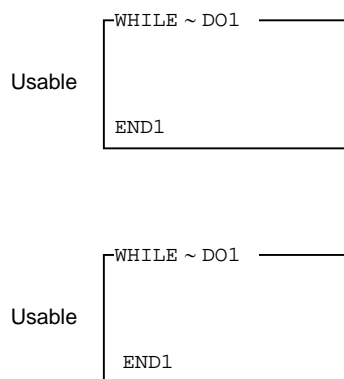
WHILE [Condition expression] DOM  (m = 1, 2, 3 ... 127)
  }
ENDm
  
```

The area from the next block to the ENDm block loops while the conditional expression holds. If the conditional expression does not hold, control will be transferred to the block after ENDm. In the format shown above, DOM can precede WHILE.

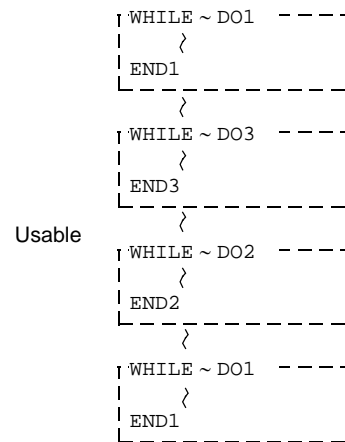
You must always use WHILE [conditional expression] DOM and ENDm in pairs. If you omit WHILE [conditional expression], the area from DOM to ENDm will endlessly loop. In DOM, m (1 to 127) identifies the number of looping. (DO1, DO2, DO3, and so on up to DO127)

The maximum available number of degrees of multiplicity is 27.

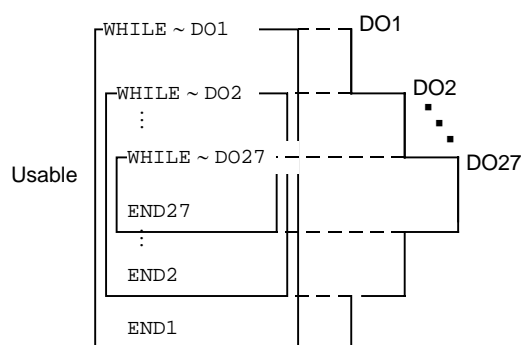
[1] Same identifying No. can be used repeatedly.



[2] The identifying No. of WHILE ~ DOM is arbitrary.

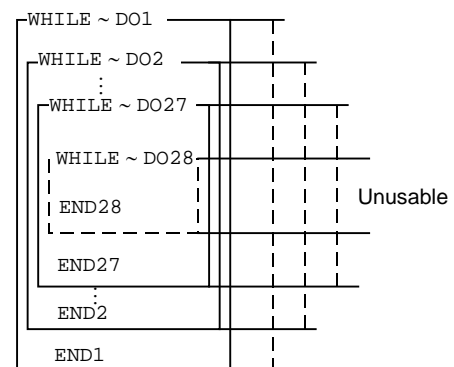


[3] Up to 27 levels of WHILE ~ DOM can be used.
m can be 1 to 127, independent of the depth of nesting.

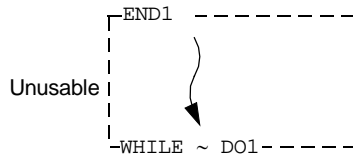


Note: For nesting, m once used cannot be used again.

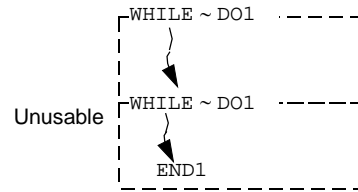
[4] The total number of levels of WHILE ~ DOM must not exceed 27.



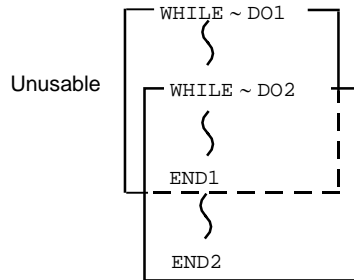
[5] WHILE ~ DOm must precede ENDm.



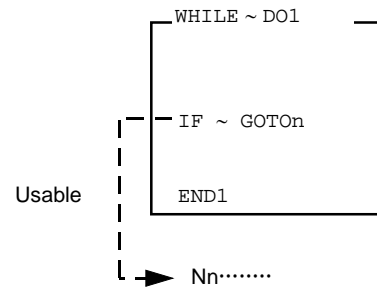
[6] WHILE ~ DOm must correspond to ENDm one-to-one in the same program.



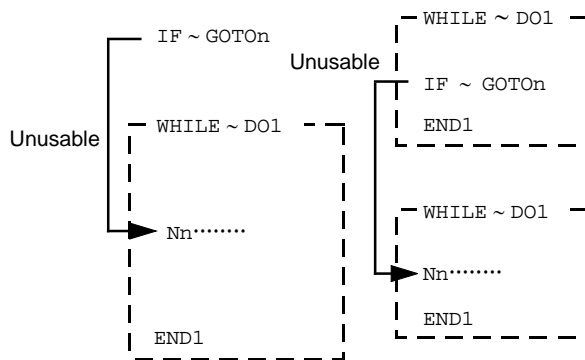
[7] WHILE ~ DOm must not overlap.



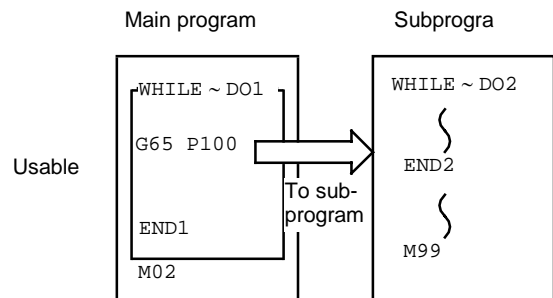
[8] Outward branching from the range of WHILE ~ DOm is possible.



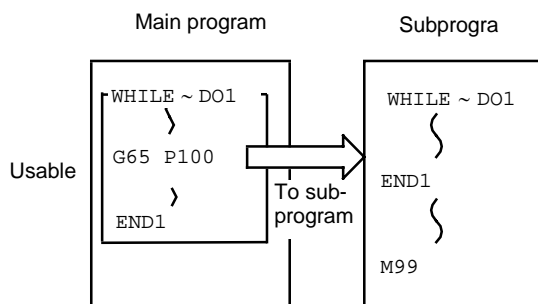
[9] Branching into WHILE ~ DOm is not allowed.



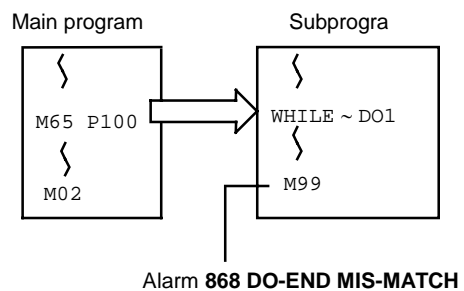
[10] Subprogram can be called using M98, G65, G66, etc. from the midway of WHILE ~ DOm.



[11] The looping can be independently programmed in a subprogram which is called using G65/G66 from the midway of WHILE ~ DOm. Up to 27 levels of nesting for both programs can be done.



[12] If WHILE and END are not included in pairs in subprogram (including macro subprogram), a program error will result at M99.



13-10-7 External output commands (Output via RS-232C)

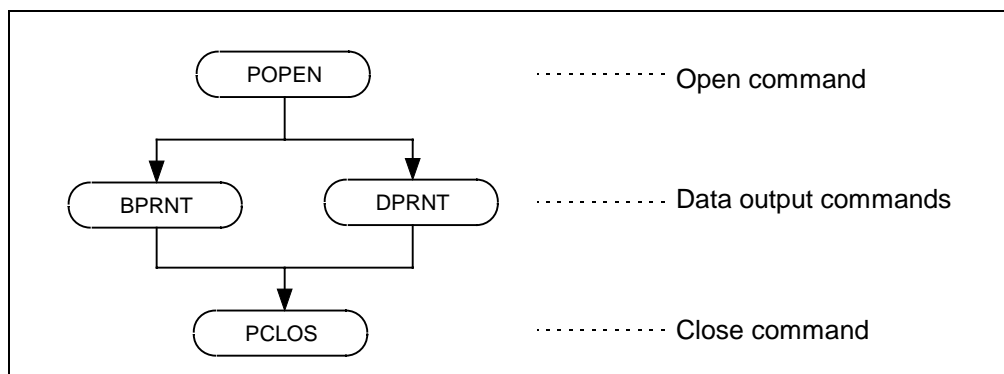
1. Overview

In addition to standard user macros, the types of macros listed below are provided as external output commands. These external output macros can be used to output character data or the numerical data in variables to an external unit via an RS-232C interface. The data are outputted in a data length of 7 bits with an even-parity bit added.

A. Types and functions of external output macros

POPEN	Setup processing for data output
PCLOS	Termination processing of data output
BPRNT	Printout of character data or binary printout of variable data
DPRNT	Printout of character data or numerical printout of variable data on a digit-by-digit basis

B. Programming order



2. Open command POPEN

Programming format: POPEN

Detailed description

- The command code POPEN must be included before a series of data output command codes.
- The control code for DC2 and the percentage code % are output from the NC unit to an external output unit.
- Once POPEN has been set, it will remain valid until PCLOS is set.

3. Close command PCLOS

Programming format: PCLOS

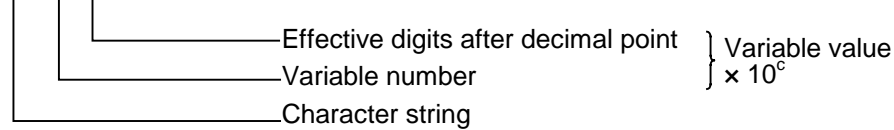
Detailed description

- The command code PCLOS must be included after all data output command codes.
- The control code for DC4 and the percentage code % are output from the NC unit to an external output unit.
- This command must be used together with POPEN. This command code must be included only after POPEN.
- This command must be set at the end of the program even after data output has been aborted using, for example, the NC reset switch.

4. Data output command BPRNT

Programming format:

BPRNT[ℓ 1#v1[c1] ℓ 2#v2[c2].....]



Detailed description

- The command BPRNT can be used to output characters or to output variable data in binary form.
- The designated character string is output directly in the ISO coded format. Alphanumerics (A to Z, and 0 to 9) and/or special characters (+, -, *, /) can be used. Of these characters, only the asterisk (*) is output as a space code.
- Since all variables are saved as those having a decimal point, the necessary number of decimal digits must be enclosed in brackets ([]).

All variables are handled as data of four bytes (32 bits), and each byte is output as binary data in the order of the most significant byte first. Minus data is processed as the complement for that data.

Example 1: If three digits are specified for 12.3456, then
 $[12.346 \times 10^3] = 12346$ (0000303A)
 will be output as binary data.

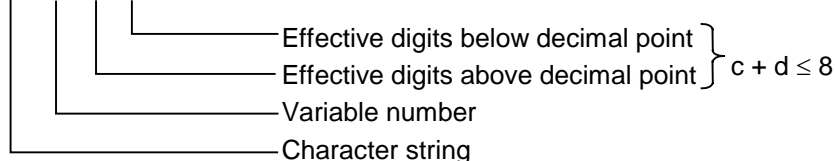
Example 2: If no digits are specified for -100.0, then
 -100 (FFFFFF9C)
 will be output as binary data.

- After the specified data has been output, the EOB (End Of Block) code is output in the format of the appropriate ISO code.
- Variables containing <empty> are interpreted as 0s.

5. Data output command DPRNT

Programming format:

DPRNT[ℓ 1#v1[d1 c1] ℓ 2#v2[c2].....]



Detailed description

- Output of character data or decimal output of variable data is performed in the format of ISO codes.
- The designated character string is output directly in the ISO coded format. Alphanumerics (A to Z, and 0 to 9) and/or special characters (+, -, *, /) can be used. Of these characters, only the asterisk (*) is output as a space code.
- Of the data contained in a variable, the necessary number of digits above the decimal point and that of digits below the decimal point must each be enclosed in brackets ([]). The variable data will then have its total specified number of digits, including the decimal point, output in the ISO coded format in the order of the most significant digit first. No trailing zeros will be left out in that case.

13-10-8 External output command (Output onto the hard disk)

1. Overview

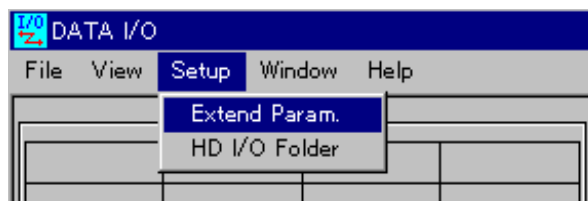
External output macros can also be used to output data in text file format into the predetermined directory on the hard disk.

2. Related parameters

- **DPR14**: Selection of an output destination port
Set **DPR14** to "4" (Output onto the hard disk) in the **PARAM.** window provided for data input/output operation.
- **DPR15**: Number of lines in feed section
Set the required number of lines to be fed.
- **DPR8**: Output file size
Use this parameter to specify the maximum permissible output file size.
Maximum permissible file size: $\text{DPR8} \times 100$ (KB)
A command for outputting a greater file will cause a corresponding alarm.
However, the file size limit is 100 KB if the value in **DPR8** is 0.

Note: Output of a file of smaller size than the limit, however, may not be possible due to a shortage of available area on the hard disk.

The **PARAM.** window (used to set extended parameters for data communication) can be called up from the menu bar on the **DATA I/O** display.



See the Parameter List for details of the parameters.

3. Output file

The text file will be automatically outputted with a particular file name into the predetermined directory.

Output directory: c:\ MC_sdg\ print\

Output file name: print.txt

(A file of this name will be automatically created, if required, or the text data will be added to the current contents of the file.)

File contents:

Given below on the right is an example of text file contents created by the execution of the program shown on the left under the particular parameter settings.

[Program]	[Output example]
G28XYZ	print.txt
POPEN	%
DPRNT[OOOOOOOOOOOO]	OOOOOOOOOOOO
DPRNT[XXXXXXXXXXXXXX]	XXXXXXXXXXXXXX
DPRNT[IIIIIIIIIIIII]	IIIIIIIIIIII
PCLOS	%
G0X100.Y100.Z100.	
M30	

[Parameter]
DPR14: 4
DPR15: No setting

4. Related alarms

The alarm given for text file output is described below.

No.	Message	Argument 1		Argument 2	Argument 3
887	TAPE I/O ERROR	-100	File open error	0	0
		-111	File write error	0	0
		-112	File size too great	0	0

13-10-9 Precautions

Use of user macro commands allows a machining program to be created by combining arithmetic operation, judgment, branching, or other macro commands with conventional NC commands such as move commands, M-, S-, T-commands, etc. The statement defined by these macro commands and that of conventional NC commands are taken as a macro statement and an NC execute statement, respectively. The treatment of a macro statement has no direct relations with machine control. Its treatment as short as possible is effective for shortening machining time. Parallel processing of the NC execute statement and the macro statement becomes possible according to the setting of bit 6 of parameter **F93**.

(It becomes possible to process all macro statements in batch form by setting the parameter bit to OFF when machining the workpiece, or to execute the macro statements block-by-block by setting the parameter bit to ON when checking the program. Therefore, set the parameter bit according to your requirements.)

Sample program

N1 G91G28X0Y0Z0

N2 G92X0Y0Z0

N3 G00X-100.Y-100.

N4 #101=100.*COS[210.]

N5 #102=100.*SIN[210.]

N6 G01X#101Y#102F800

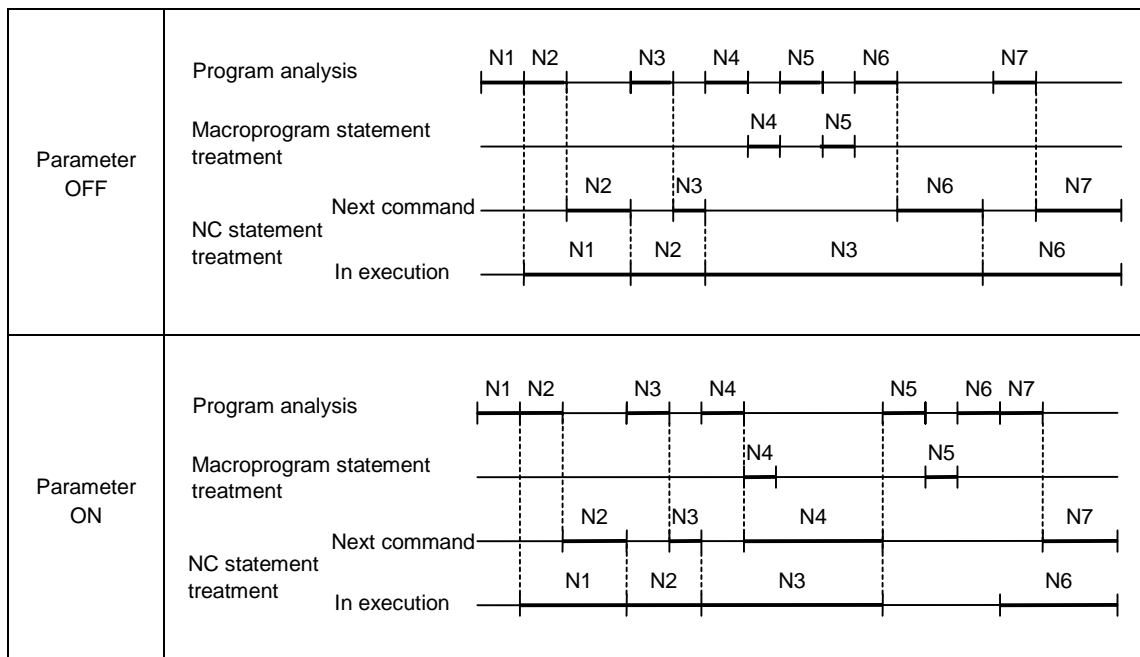
← Macro statements

A macro statement refers to a statement that consists of the following blocks:

- Arithmetic operation command block (compassing the equal sign =)
- Control command block (compassing GOTO, DO ~ END, etc.)
- Macro call command block (compassing macro call or cancellation G-code commands G65, G66, G66.1, or G67)

An NC execute statement refers to a non-macro statement.

The flow of processing of these two types of statements is shown below.



Machining program data is displayed as follows:

Parameter OFF	(In execution) N3 G00X-100.Y-100. (Next command) N6 G01X#101Y#102F800	N4, N5 and N6 are treated in parallel with NC execution sentence of N3, and N6 is displayed as next command because it is NC execution sentence. When N4, N5, and N6 are analyzed during NC execution sentence of N3, machine control continues.
Parameter ON	(In execution) N3 G00X-100.Y-100. (Next command) N4 #101=100.*COS[210.]	N4 is treated in parallel with the control of NC execution sentence of N3, and is displayed as next command. After N3 is completed, N5 and N6 are analyzed so the machine control is forced to wait by the analyzing time of N5 and N6 before N6 can be executed.

13-10-10 Specific examples of programming using user macros

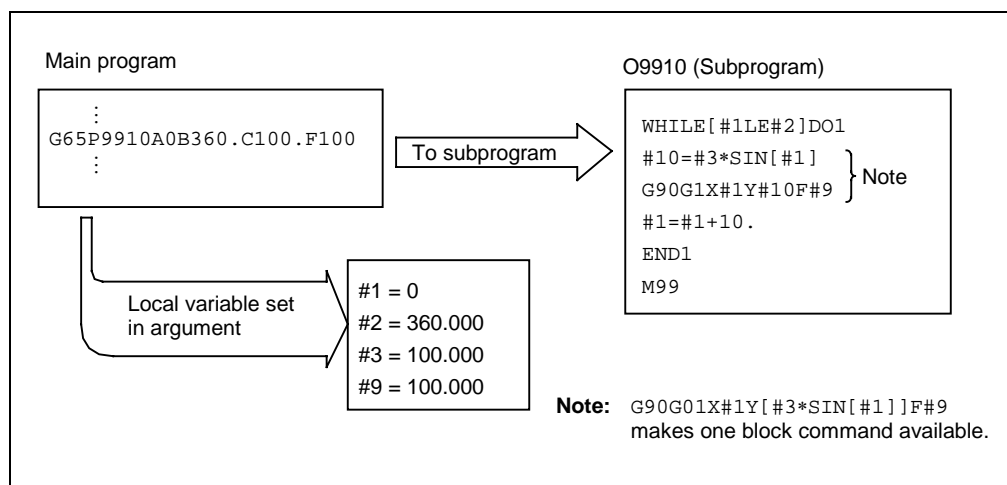
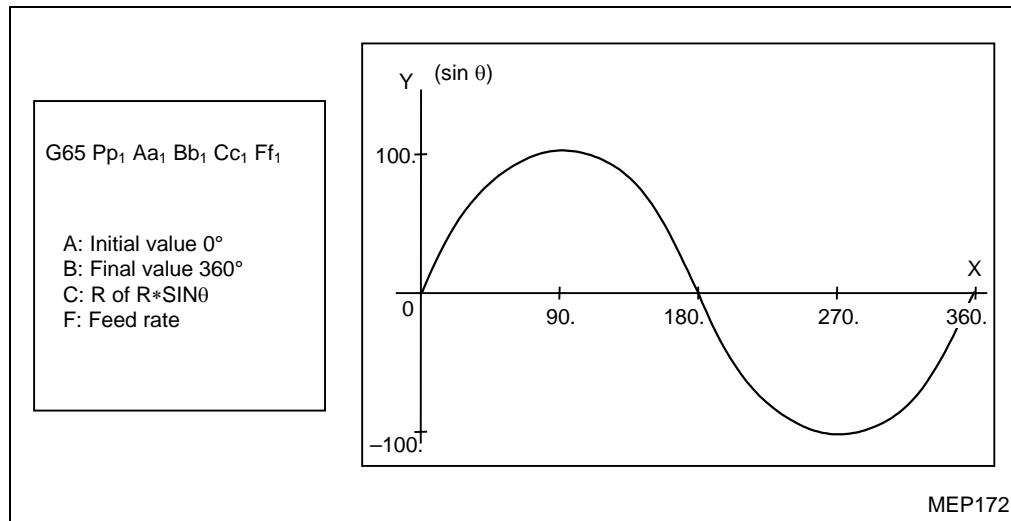
The following three examples of programming are shown here:

Example 1: SIN curve

Example 2: Bolt-hole circle

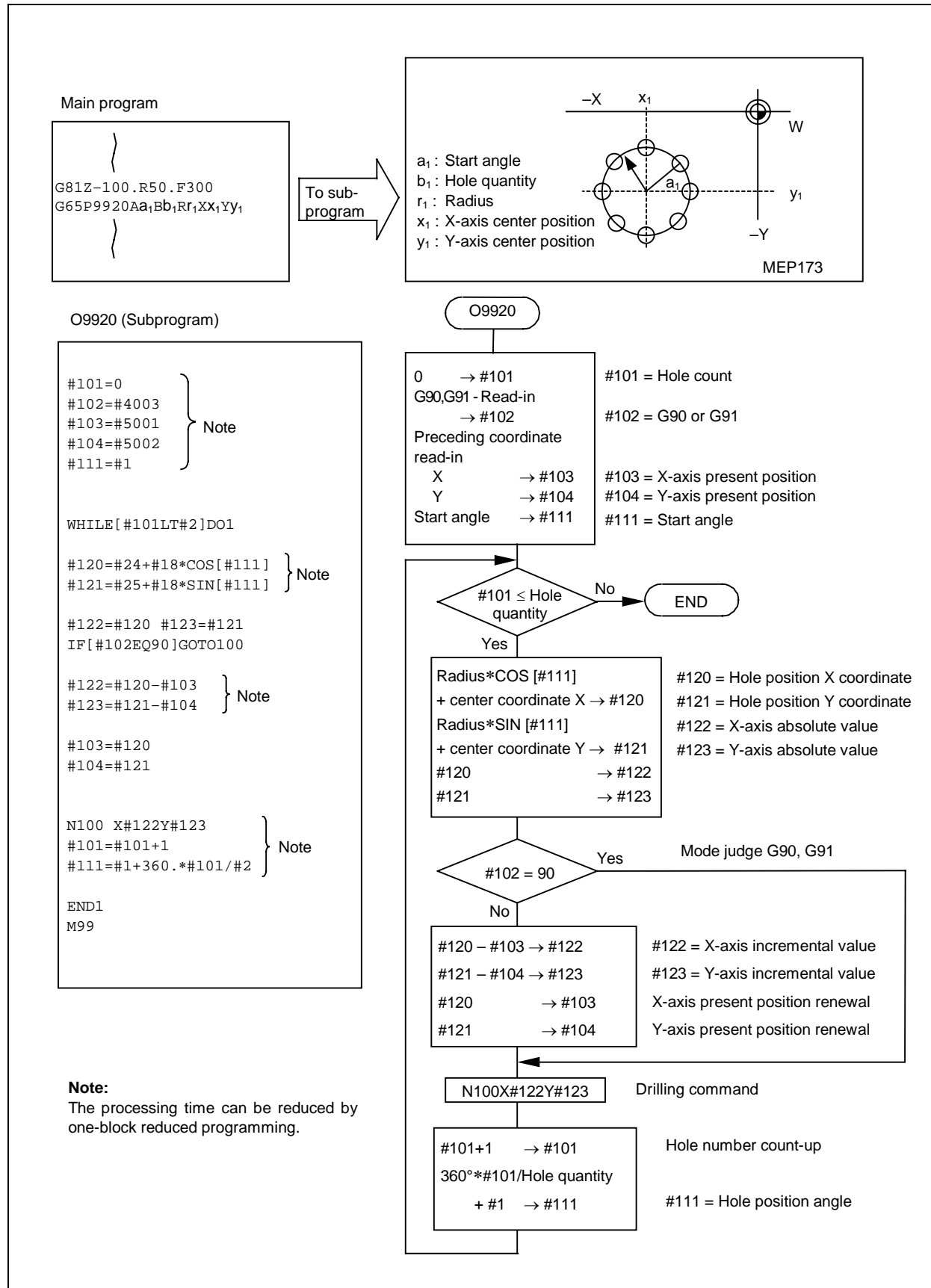
Example 3: Grid

Example 1: SIN curve



Example 2: Bolt-hole circle

After hole data has been defined using fixed-cycle machining commands G72 to G89, hole positions are to be designated using macro commands.

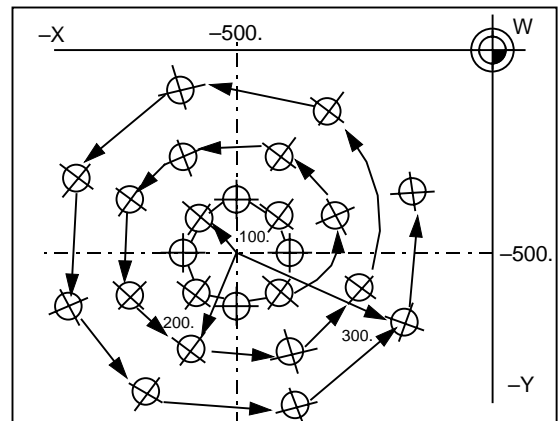
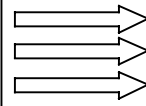


```

G28X Y Z
T1 M06
G90 G43 Z100. H01
G54 G00 X0 Y0
G81 Z-100. R3. F100 M03
G65 P9920X-500. Y-500. A0 B8R100.
G65 P9920X-500. Y-500. A30. B8R200.
G65 P9920X-500. Y-500. A60. B8R300.

```

To subprogram



MEP174

Example 3: Grid

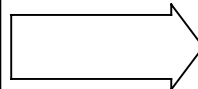
After hole data has been defined using fixed-cycle machining commands G72 to G89, hole positions are to be designated using macro call commands.

```

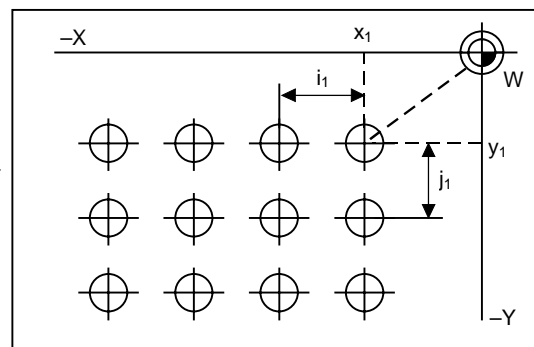
G81 Zz1 Rr1 Ff1
G65Pp1 Xx1 Yy1 Ii1 Jj1 Aa1 Bb1

```

X : X-axis hole position
Y : Y-axis hole position
I : X-axis distance
J : Y-axis distance
A : X direction hole quantity
B : Y direction hole quantity



Subprogram is shown on the next page.

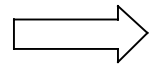


MEP175

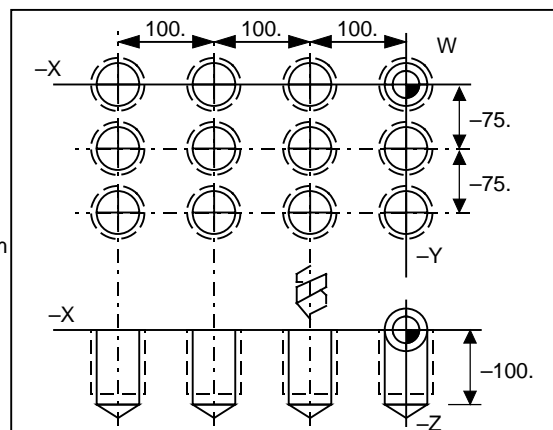
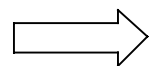
```

G28X Y Z
T1 M06
G90 G43 Z100. H01
G54 G00 X0 Y0
G81 Z-100. R3. F100M03
G65P9930 X0 Y0 I100. J-75. A5B3

```



To subprogram



MEP176

O9930 (Subprogram)

```
#101=#24
#102=#25
#104=#5
#105=#1
#106=#2-1
#110=0
#111=0
#112=0
```

Note

```
N2 #113=0
#103=#4
```

Note

```
WHILE[#105GT0]DO1
#101=#101+#113
#105=#105-1
X#101Y#102
```

Note

```
IF[#112EQ1]GOTO10
IF[#111NE1]GOTO10
```

Note

```
#103=0-#103
#112=1
```

```
N10 #113=#103
END1
```

```
N100 #106=#106-1
#112=0
#110=#110+1
```

Note

```
IF[#106LT0]GOTO200
```

```
#105=#1
#102=#102+#104
#111=#110
```

Note

```
#111=#111AND1
```

```
GOTO2
```

```
N200 M99
```

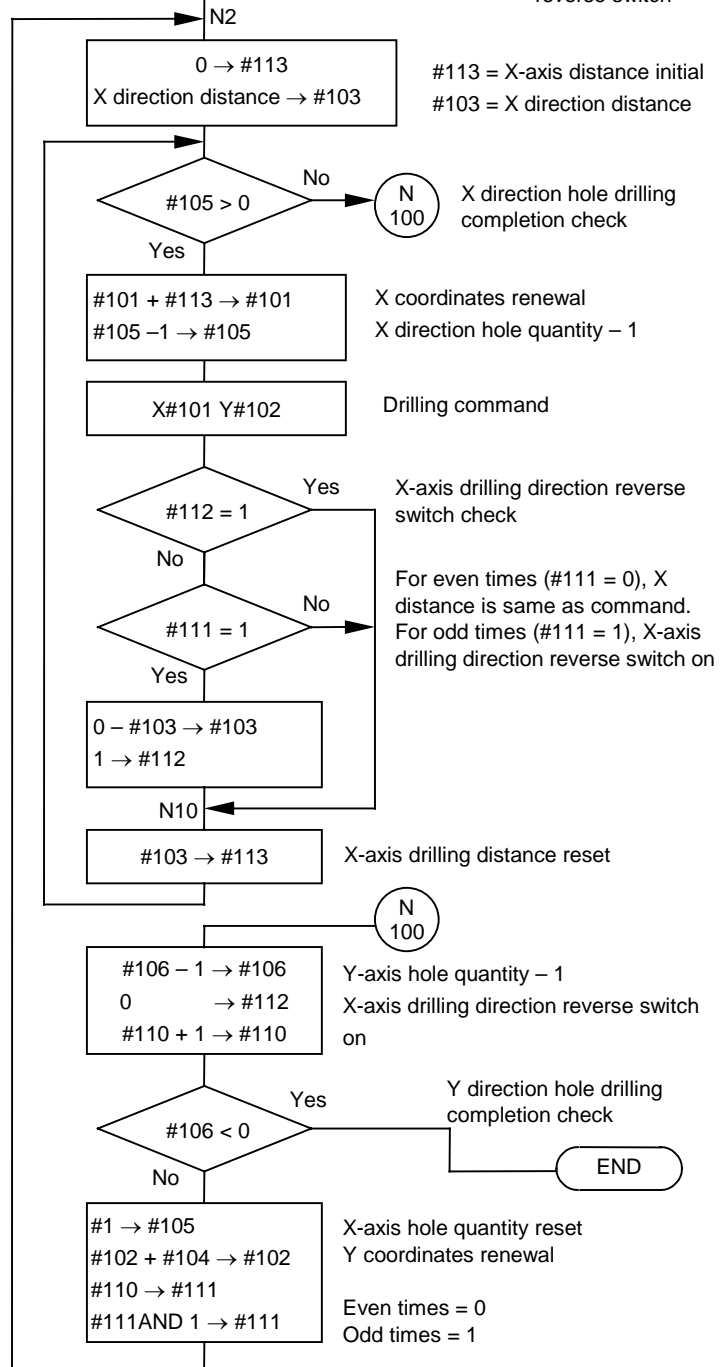
Note:

The processing time can be reduced by one-block reduced programming.

O9930

```
Start point X coordinate → #101
Start point Y coordinate → #102
Y-axis direction distance → #104
X-axis hole quantity → #105
Y-axis hole quantity -1 → #106
0 → #110
0 → #111
0 → #112
```

```
#101 = X-axis start point
#102 = Y-axis start point
#104 = Y direction distance
#105 = X-axis hole quantity
#106 = Y-axis hole quantity - 1
#110 = Y direction line count
#111 = Even/odd times judge
#112 = X-axis drilling direction
reverse switch
```



13-11 Scaling: G50, G51

1. General description

The shape specified in a machining program can be enlarged or reduced in size using scaling command G51. The range of scaling (enlargement/reduction) factors is from 0.000001 to 99.999999.

Use command G51 to specify a scaling axis, the center of scaling, and a scaling factor.

Use command G50 to specify scaling cancellation.

2. Programming formats

G51 Xx Yy Zz Pp Scaling on (specify a scaling axis, the center of scaling (incremental/absolute), and a scaling factor)

G50 Scaling cancel

3. Detailed description

A. Specifying a scaling axis

The scaling mode is set automatically by setting G51. Command G51 does not move any axis; it only specifies a scaling axis, the center of scaling, and a scaling factor.

Scaling becomes valid only for the axis to which the center of scaling has been specified.

Center of scaling

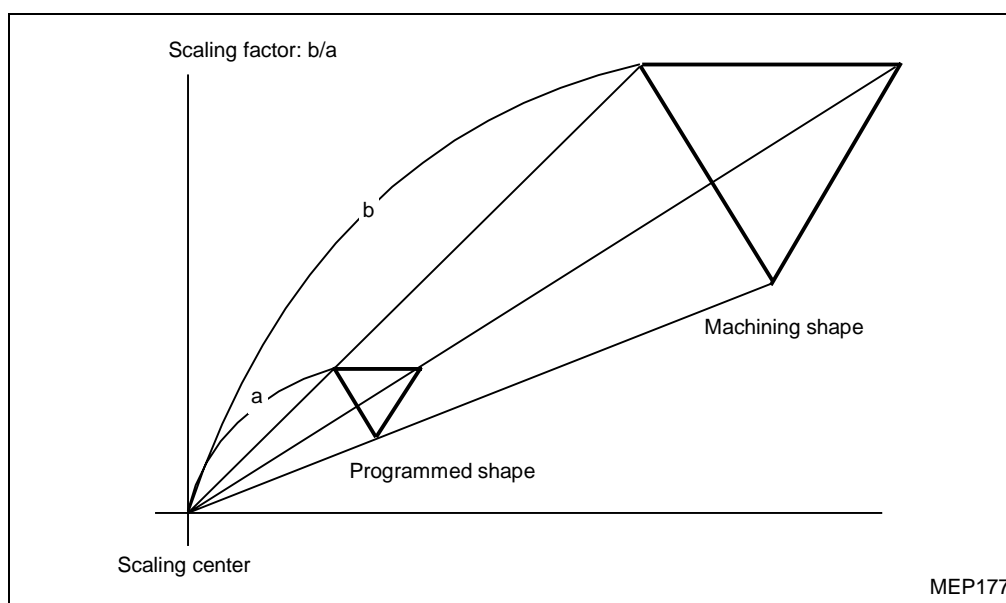
The center of scaling must be specified with the axis address according to the absolute or incremental data command mode (G90 or G91). This also applies even when specifying the current position as the center.

Scaling factor

Use address P to specify a scaling factor.

Minimum unit of specification: 0.000001

Specifiable range of factors: 1 to 99999999 or 0.000001 to 99.999999 (times)
(Although both are valid, the latter with a decimal point must be preceded by G51.)



The scaling factor set in parameter **F20** will be used if you do not specify any scaling factor in the same block as that of G51. The current setting of this parameter will be used if it is updated during the scaling mode. That is, the parameter setting existing when G51 is set is valid. Data will be calculated at a scaling factor of 1 if neither the program nor the parameter has a specified scaling factor.

Program errors occur in the following cases:

- If scaling is specified for a machine not capable of scaling (Alarm **872 G51 OPTION NOT FOUND**)
- If a scaling factor exceeding its maximum available value is specified in the same block as that of G51 (Alarm **809 ILLEGAL NUMBER INPUT**) (All scaling factors less than 0.000001 are processed as 1.)

B. Cancellation of scaling

The scaling cancel mode is set automatically by setting G50. Setting this command code offsets any deviation between the program coordinates and the coordinates of the actual machine position. Even for axes that have not been designated in the same block as that of G50, the machine moves through the offset amount specified by scaling.

4. Precautions

1. Scaling does not become valid for tool diameter offsetting, tool length offsetting, or tool position offsetting. Offsets and other corrections are calculated only for the shape existing after scaling.
2. Scaling is valid only for move commands associated with automatic operation (tape, memory, or MDI); it is not valid for manual movement.
3. After-scaling coordinates are displayed as position data.
4. Scaling is performed on the axis for which the center of scaling is specified by G51. In that case, scaling becomes valid for all move commands associated with automatic operation, as well as for the parameter-set return strokes of G73 and G83 and for the shift strokes of G76 and G87.
5. If only one axis of the plane concerned is selected for scaling, circular interpolation is performed with the single scaling on that axis.
6. Scaling will be cancelled if either M02, M30, or M00 (only when M0 contains reset) is issued during the scaling mode. Scaling is also cancelled by an external reset command or any other reset functions during the reset/initial status.
7. Data P, which specifies a scaling factor, can use a decimal point. The decimal point, however, becomes valid only if scaling command code G51 precedes data P.

```
G51P0.5      0.5 time
P0.5G51      1 time (regarded as P = 0)
P500000G51   0.5 time
G51P500000   0.5 time
```

8. The center of scaling is shifted accordingly if the coordinate system is shifted using commands G92 or G52 during scaling.

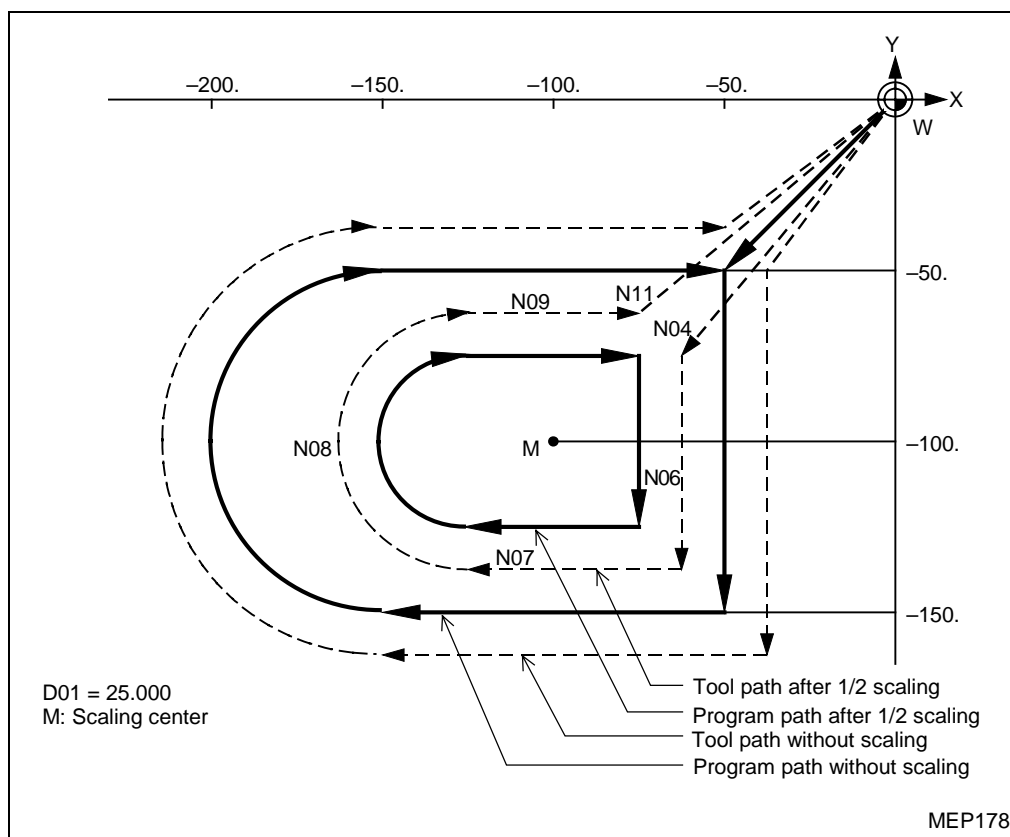
5. Sample programs

1. Basic operation I

```

N01G92X0Y0Z0
N02G90G51X-100.Y-100.P0.5
N03G00G43Z-200.H02
N04G41X-50.Y-50.D01
N05G01Z-250.F1000
N06Y-150.F200
N07X-150.
N08G02Y-50.J50.
N09G01X-50.
N10G00Z0
N11G40G50X0Y0
N12M02

```



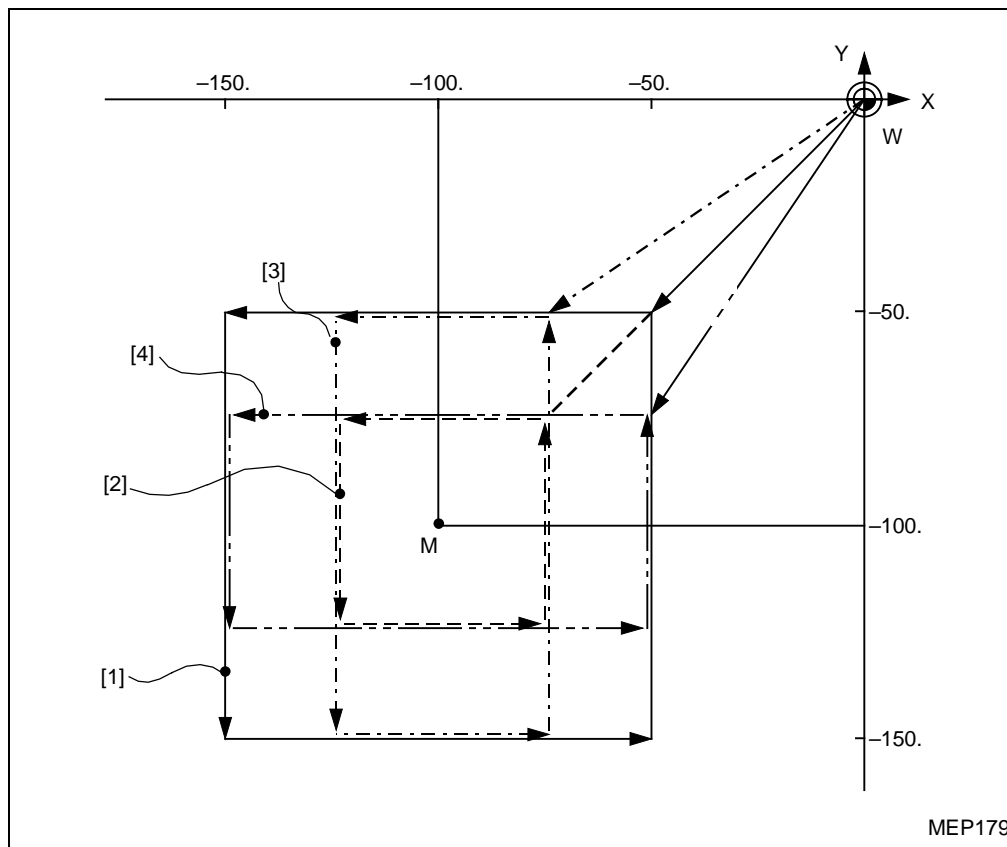
2. Basic operation II

```

N01G92X0Y0
N02G90G51P0.5 . . . . . See [1] thru [4] below.
N03G00X-50.Y-50.
N04G01X-150.F1000
N05Y-150.
N06X-50.
N07Y-50.
N08G00G50
N09M02

```

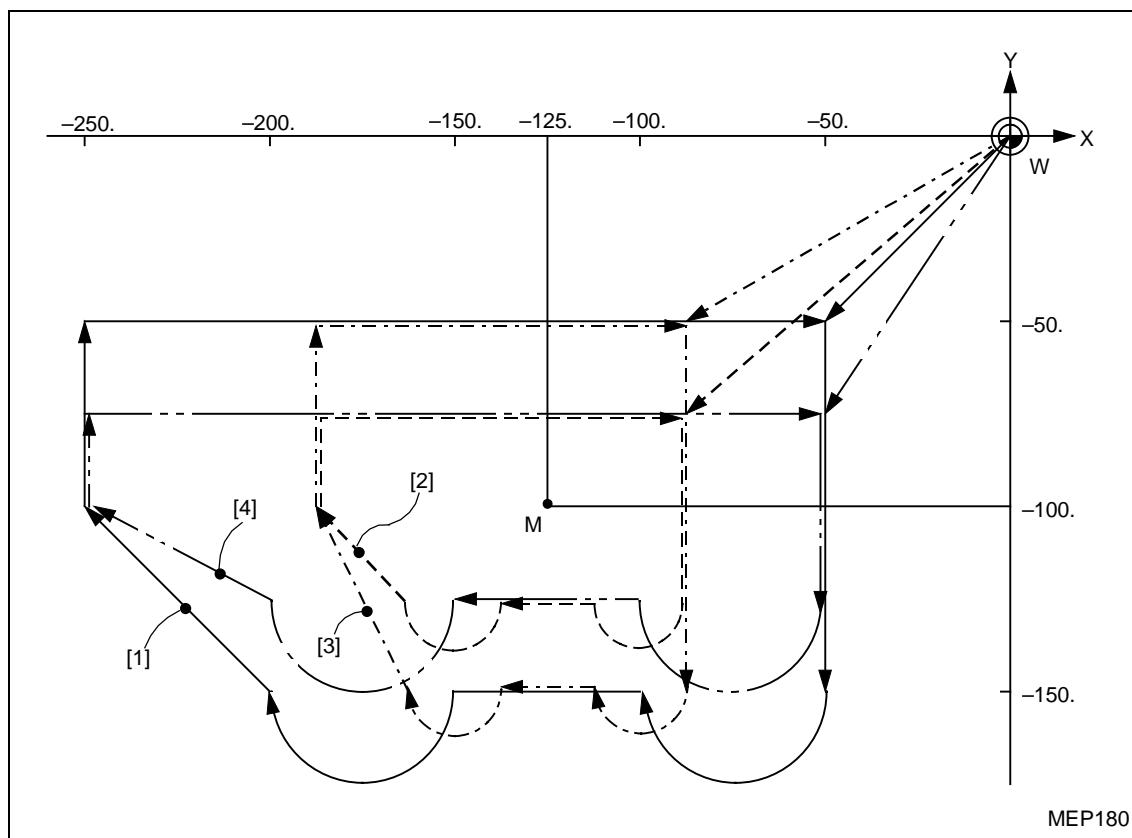
- | | |
|-----------------------------------------|---------------------------|
| [1] Without scaling | N02G90G51P0.5 |
| [2] If scaling is to be done for X, Y | N02G90G51X-100.Y-100.P0.5 |
| [3] If scaling is to be done for X only | N02G90G51X-100.P0.5 |
| [4] If scaling is to be done for Y only | N02G90G51Y-100.P0.5 |



3. Basic operation III

N01G92X0Y0
 N02G90G51P0.5 See [1] thru [4] below.
 N03G00X-50.Y-50.
 N04G01Y-150.F1000
 N05G02X-100.I-25.
 N06G01X-150.
 N07G02X-200.I-25.
 N08G01X-250.Y-100.
 N09Y-50.
 N10X-50.
 N11G00G50
 N12M02

- | | |
|-----------------------------------------|---------------------------|
| [1] Without scaling | N02G90G51P0.5 |
| [2] If scaling is to be done for X, Y | N02G90G51X-125.Y-100.P0.5 |
| [3] If scaling is to be done for X only | N02G90G51X-125.P0.5 |
| [4] If scaling is to be done for Y only | N02G90G51Y-100.P0.5 |



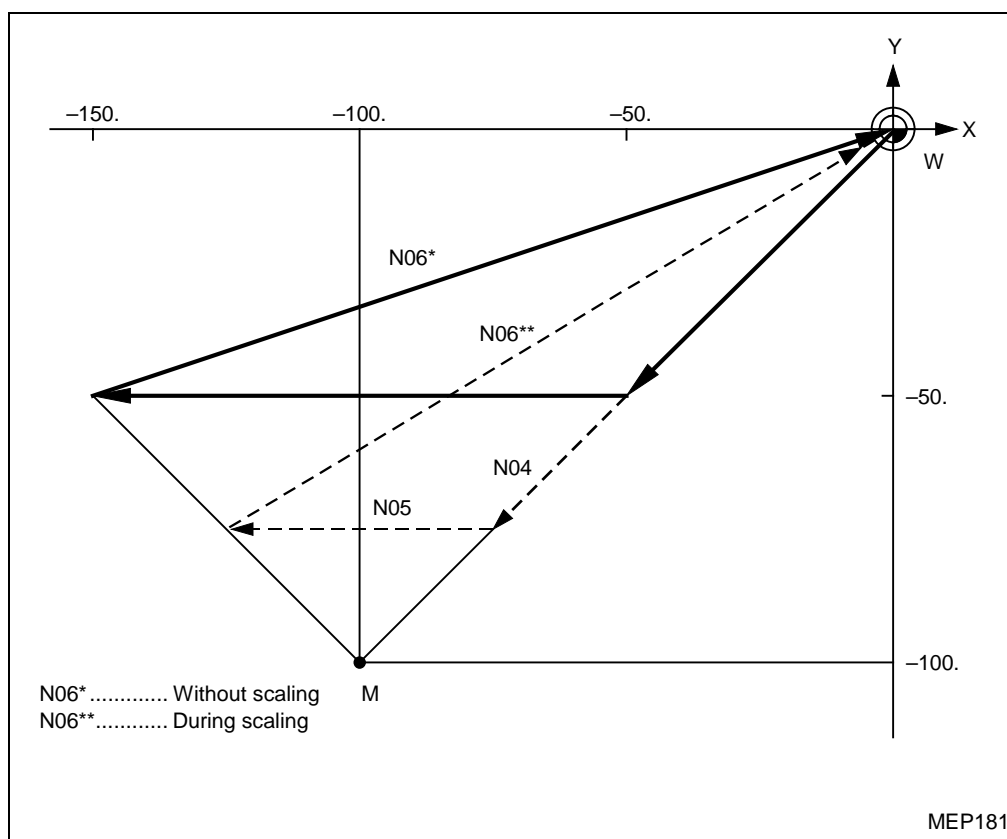
4. Reference-point (zero point) check (G27) during scaling

Setting G27 during scaling cancels the scaling mode after G27 has been executed.

```

N01G28X0Y0
N02G92X0Y0
N03G90G51X-100.Y-100.P0.5
N04G00X-50.Y-50.
N05G01X-150.F1000
N06G27X0Y0
      :
```

If a program is constructed in the manner that the reference point is reached under normal mode, it will also be reached even under scaling mode.

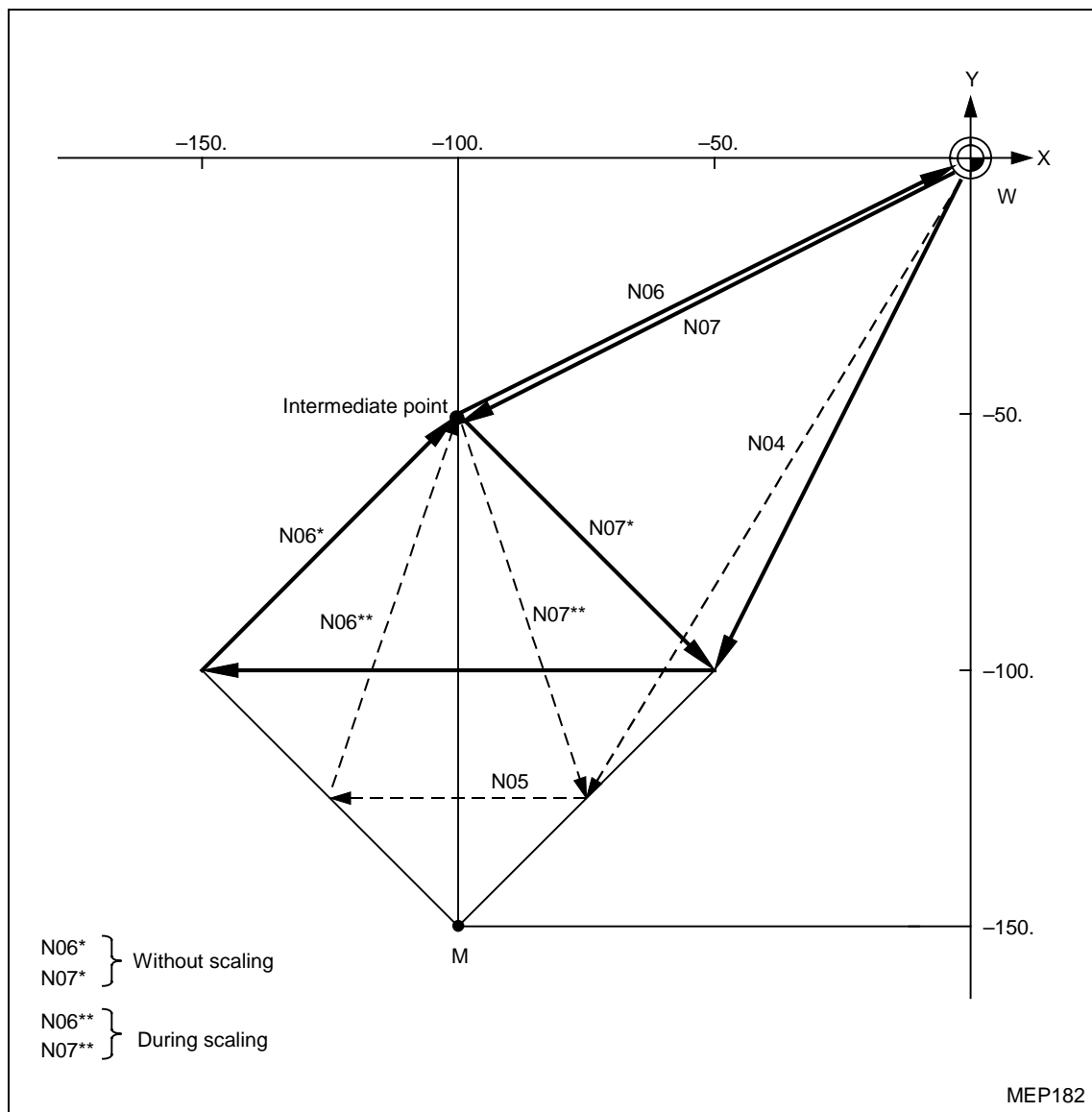


5. Reference-point (zero point) return (G28, G29, or G30) during scaling

Setting G28 or G30 during scaling cancels the scaling mode at the middle point and then executes the reference-point (zero point) return command. If the middle point has not been set, the reference-point (zero point) return command is executed with the point where scaling has been cancelled as middle point.

If G29 is set during the scaling mode, scaling will be performed for the entire movement after the middle point.

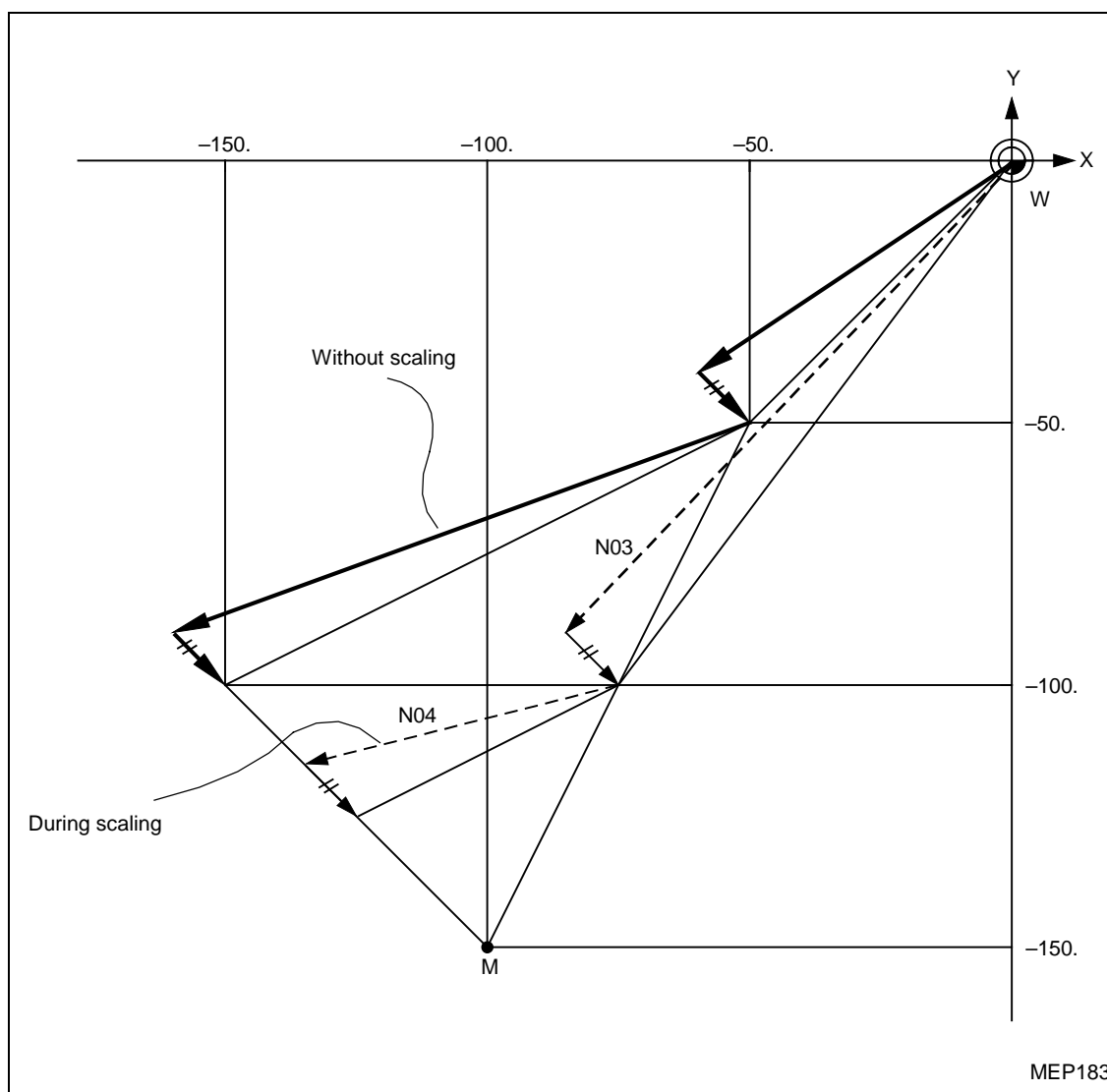
N01G28X0Y0
N02G92X0Y0
N03G90G51X-100.Y-150.P500000
N04G00X-50.Y-100. 0.5
N05G01X-150.F1000
N06G28X-100.Y-50.
N07G29X-50.Y-100.



6. One-way positioning (G60) during scaling

Setting G60 during the scaling mode executes scaling at the final point of positioning, and thus no scaling is performed for the parameter **I1** of creeping. That is, the amount of creeping remains constant, irrespective of whether scaling is valid.

```
N01G92X0Y0  
N02G91G51X-100.Y-150.P0.5  
N03G60X-50.Y-50.  
N04G60X-150.Y-100.
```



7. Workpiece coordinate system updating during scaling

Updating of the workpiece coordinate system during scaling causes the center of scaling to be shifted according to the difference in offset amount between the new workpiece coordinate system and the old one.

```

N01G90G54G00X0Y0
N02G51X-100.Y-100.P0.5
N03G65P100
N04G90G55G00X0Y0
N05G65P100

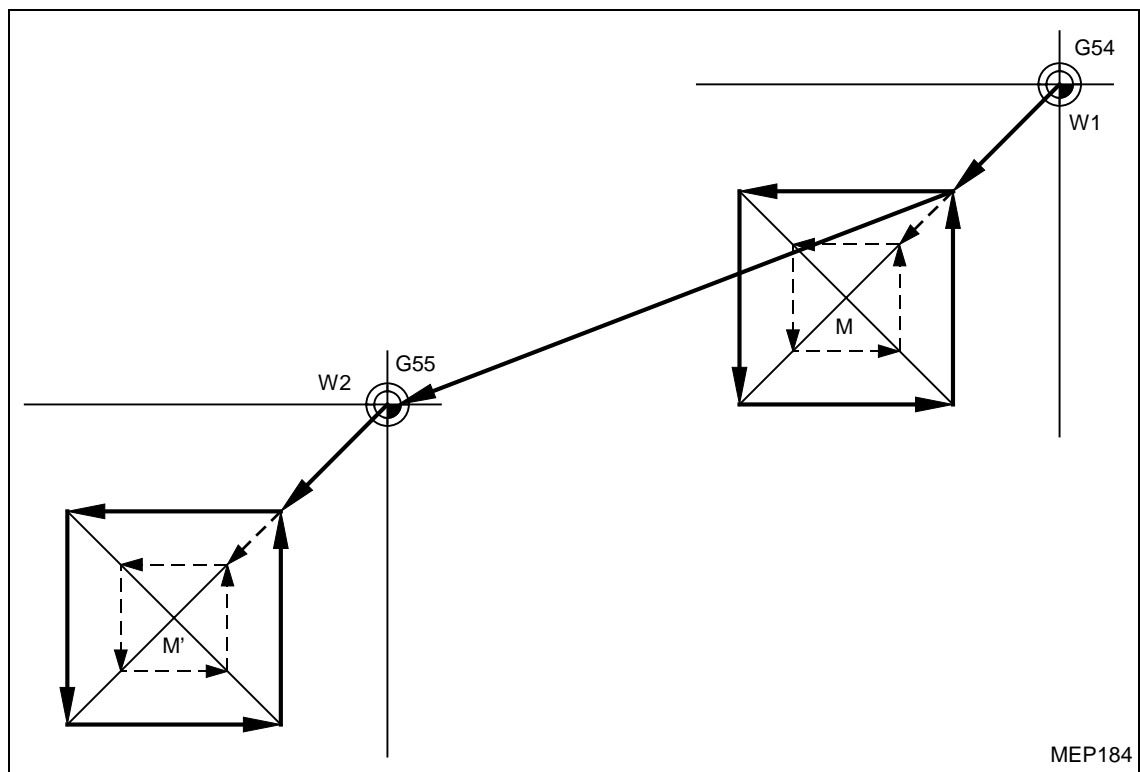
```

Subprogram

```

O100
G00X-50.Y-50.
G01X-150.F1000
Y-150.
X-50.
Y-50.
M99
%

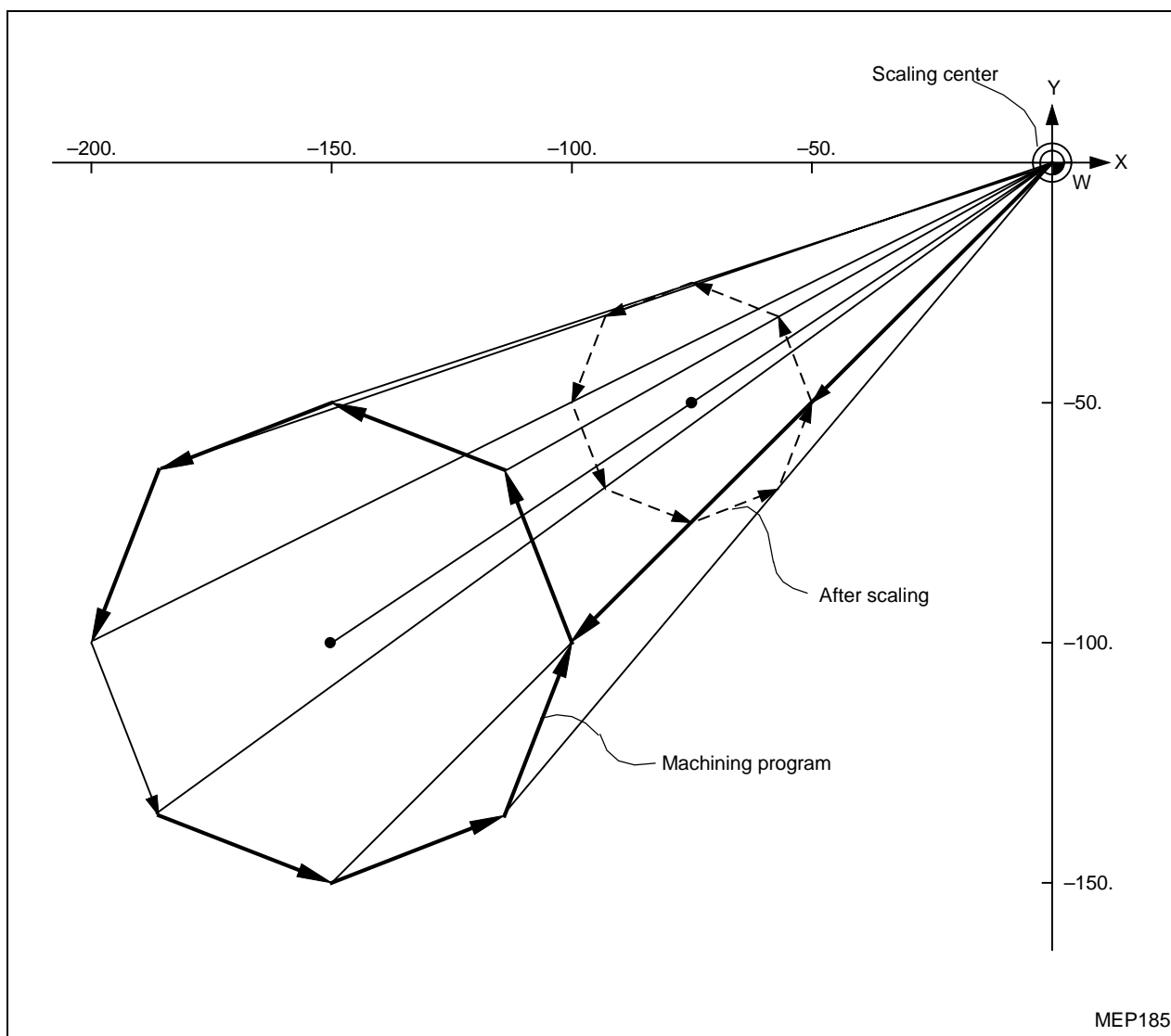
```



8. Figure rotation during scaling

Setting a figure rotate command during scaling executes scaling for both the center and radius of rotation of the figure.

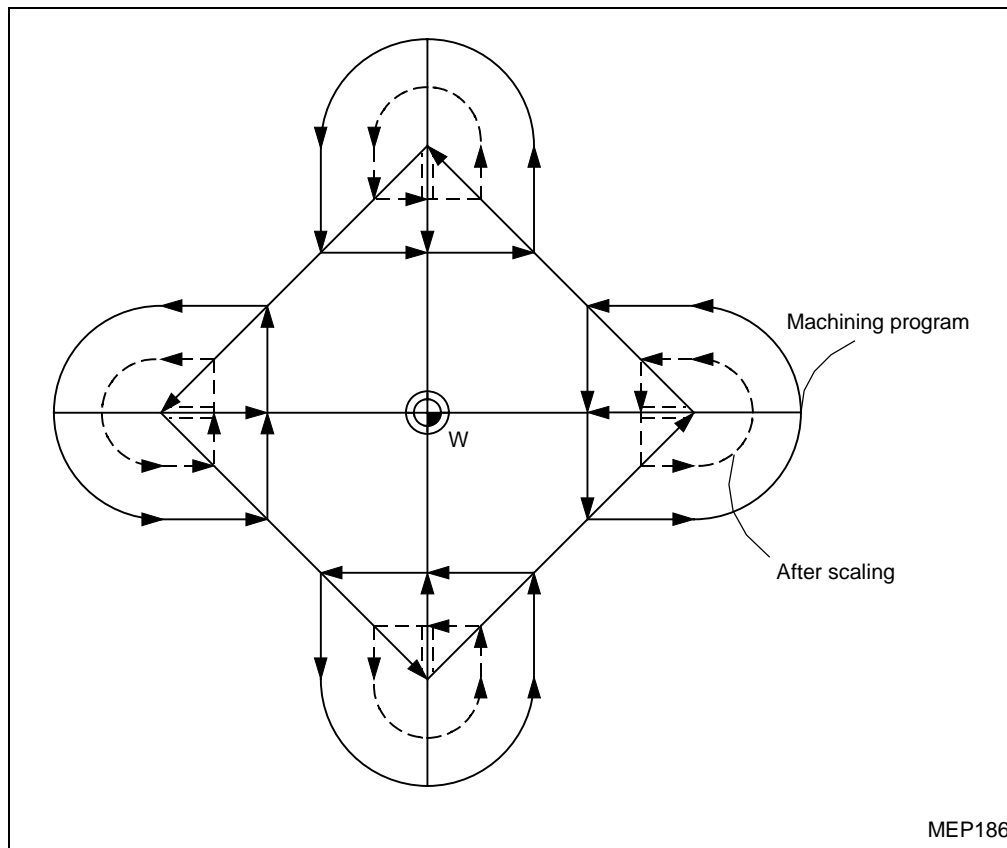
	Subprogram
N01G92X0Y0	O200
N02G90G51X0Y0P0.5	G91G01X-14.645Y35.355F1000
N03G00X-100.Y-100.	M99
N04M98P200I-50.L8	%



9. Scaling using a figure rotation subprogram

Setting a scaling command in a figure rotation subprogram executes scaling only for the shape predefined in the subprogram. Scaling is not executed for the radius of rotation of the figure.

	Subprogram
G92X0Y0	O300
G90G00X100.	G91G51X0Y0P0.5
M98P300I-100.L4	G00X-40.
G90G00X0Y0	G01Y-40.F1000
M02	X40.
	G03Y80.J40.
	G01X-40.
	Y-40.
	G00G50X40.
	X-100.Y100.
	M99
	%



MEP186

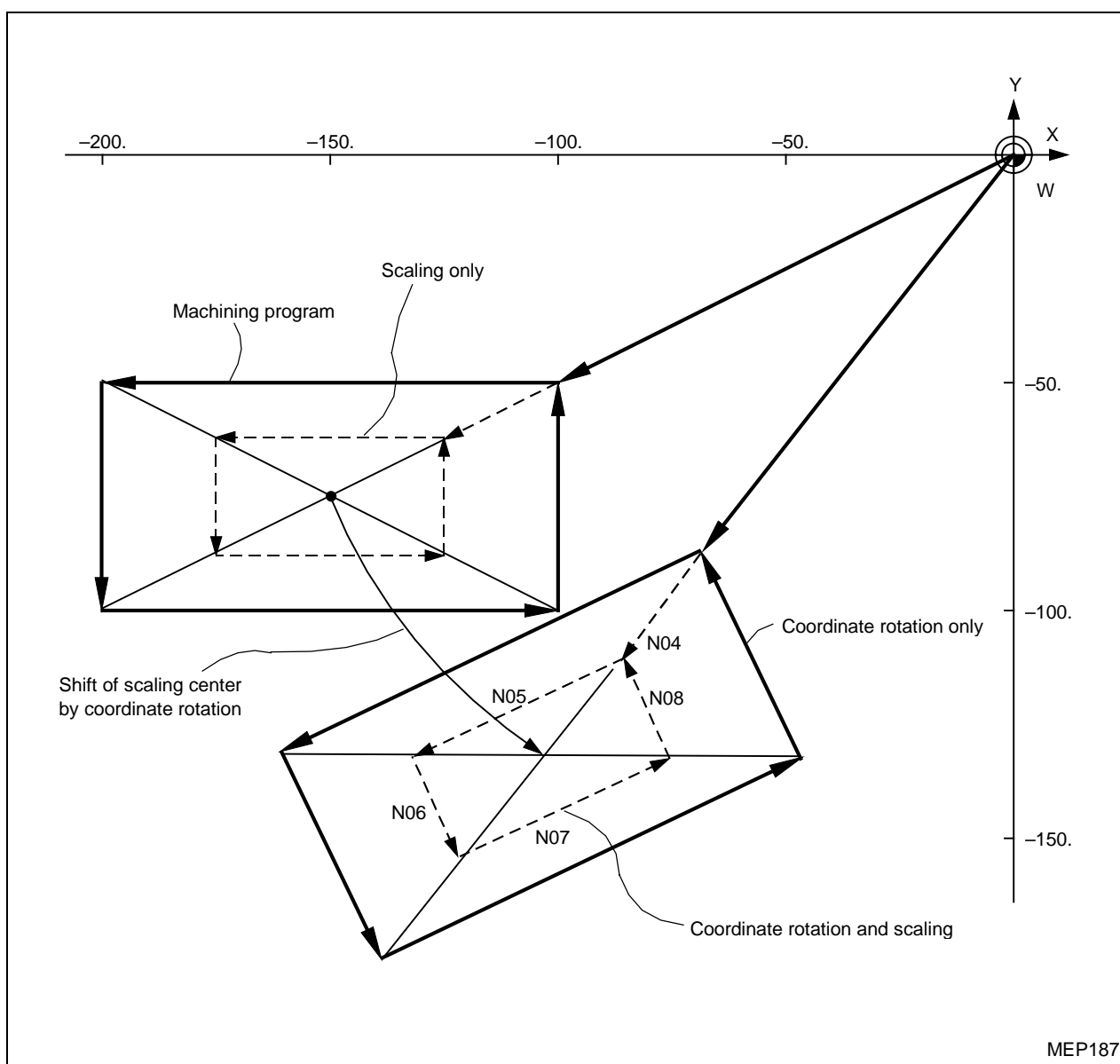
10. Scaling during coordinate rotation

If scaling during coordinate rotation is programmed the center of scaling will rotate and scaling will be performed at that rotated center of scaling.

```

N01G92X0Y0
N02M00                                     (Coordinate rotation data setting)
N03G90G51X-150.Y-75.P0.5
N04G00X-100.Y-50,
N05G01X-200.F1000
N06Y-100.
N07X-100.
N08Y-50.
N09G00G50X0Y0

```



MEP187

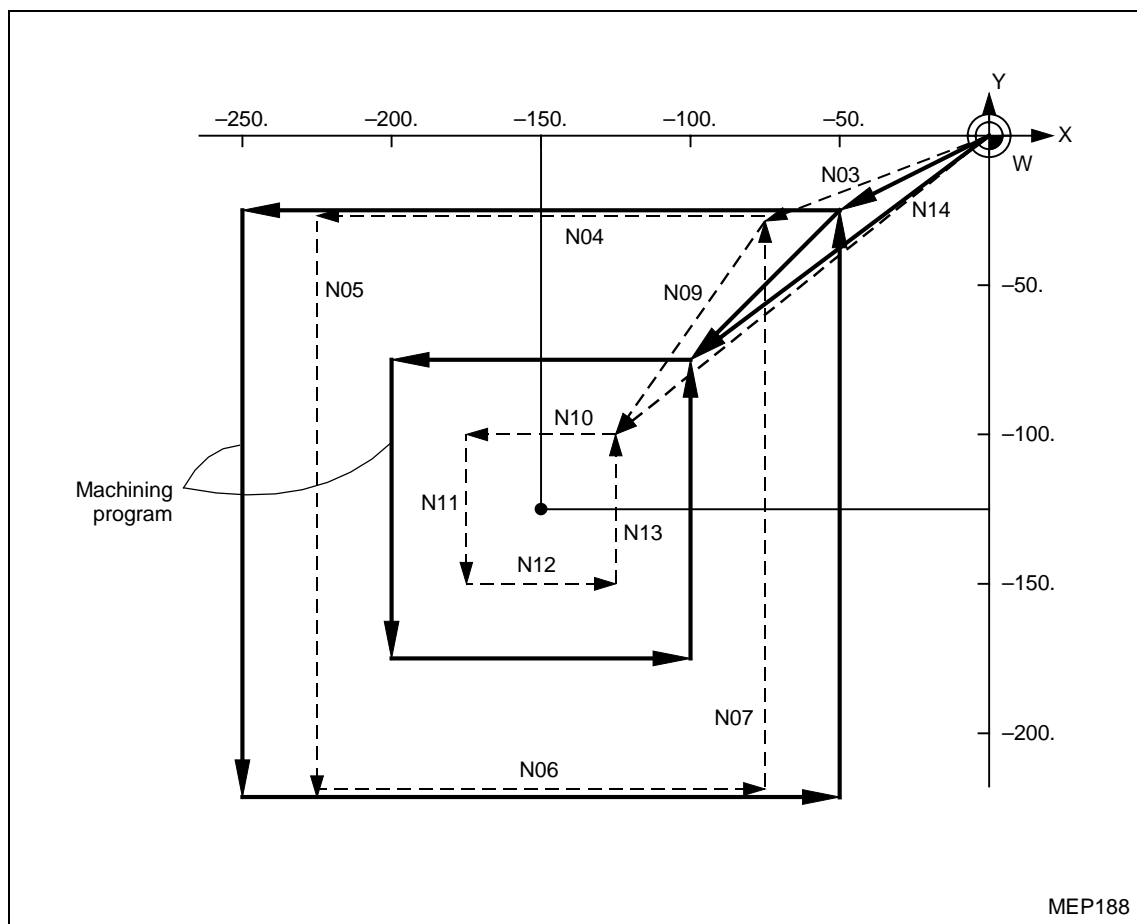
11. Setting G51 during scaling

If command G51 is set during the scaling mode, the axis for which the center of scaling is newly specified will also undergo scaling. The scaling factor specified by the latest G51 command becomes valid in that case.

```

N01G92X0Y0
N02G90G51X-150.P0.75      Scaling axis X; P = 0.75
N03G00X-50.Y-25.
N04G01X-250.F1000
N05Y-225.
N06X-50.
N07Y-25.
N08G51Y-125.P0.5          Scaling axes X and Y; P = 0.5
N09G00X-100.Y-75.
N10G01X-200.
N11Y-175.
N12X-100.
N13Y-75.
N14G00G50X0Y0             Cancel

```



MEP188

13-12 Mirror Image On/Off G-Codes: G50.1, G51.1

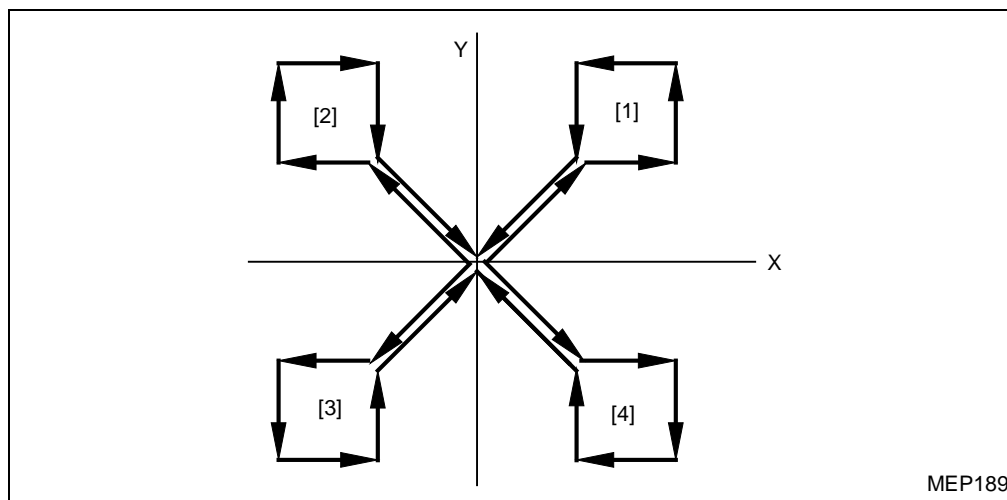
You can use G-code commands to turn the mirror image mode on or off for each axis. Higher priority is given to setting of the mirror image mode using G-code commands over setting using any other methods.

Programming format:

G51.1 Xx₁ Yy₁ Zz₁ (Mirror image ON)
 G50.1 Xx₂ Yy₂ Zz₂ (Mirror image OFF)

Detailed description

- When using command G51.1, the name of the axis for which mirror image processing is to be performed must be designated using the appropriate coordinate word, and the mirror image center coordinates must be designated using absolute or incremental data as the coordinate data.
- If the coordinate word is designated in G50.1, then this denotes the axis for which the mirror image is to be cancelled. Coordinate data, even if predefined, is ignored in that case.
- After mirror image processing has been performed for only one of the axes forming a plane, the rotational direction and the offset direction become reverse during arc interpolation, tool diameter offsetting, or coordinate rotation.
- Since the mirror image processing function is valid only for local coordinate systems, the center of mirror image processing moves according to the particular counter preset data or workpiece coordinate offsetting data.



MEP189

Specific examples

(Main program)

G00G90G40G49G80

M98P100

G51.1X0

M98P100

G51.1Y0

M98P100

G50.1X0

M98P100

G50.1Y0

M30

X Y

[1] OFF OFF

[2] ON OFF

[3] ON ON

[4] OFF OFF

OFF OFF

(Subprogram O100)

G91G28X0Y0

G90G00X20.Y20.

G42G01X40.D01F120

Y40.

X20.

Y20.

G40X0Y0

M99

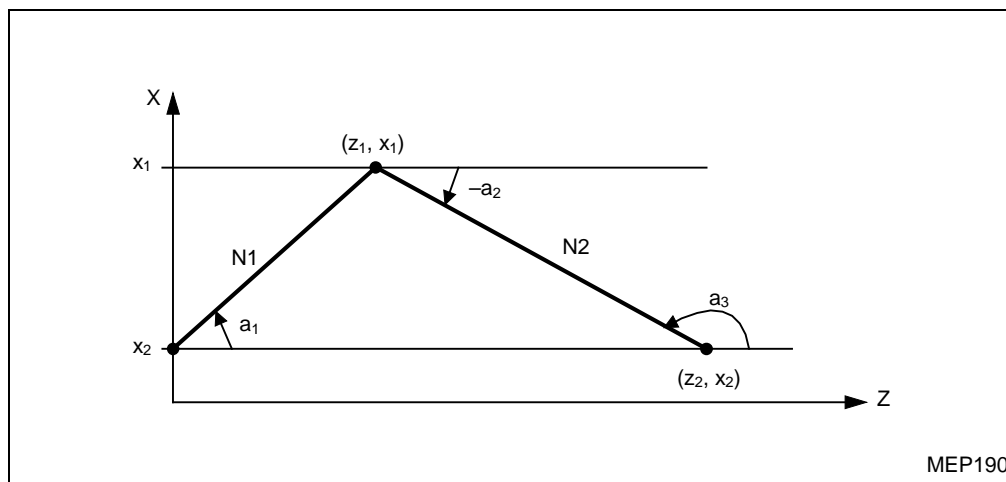
13-13 Linear Angle Commands

1. Function and purpose

Programming the linear angle and one of the coordinates of the ending point makes the NC unit automatically calculate the coordinates of that ending point.

2. Programming format

G18 Designate the plane using command G17, G18, or G19.
 N1 G01 Aa₁ Zz₁ (Xx₁) Designate the angle and the coordinates of the X-axis or the Z-axis.
 N2 G01 A-a₂ Zz₂ Xx₂ (Setting Aa₃ means the same as setting A-a₂.)



3. Detailed description

- The angle denotes that relative to the plus (+) direction of the first axis (horizontal axis) on the selected plane.
Assign the sign + for a counterclockwise direction (CCW) or the sign - for a clockwise direction (CW).
- Set the ending point on one of the two axes of the selected plane.
- Angle data will be ignored if the coordinates of both axes are set together with angles.
- If angles alone are set, the command will be handled as a geometric command.
- For the second block, the angle at either the starting point or the ending point can be specified.
- The linear angle command function does not work if address A is to be used for an axis name or for the No. 2 auxiliary function.

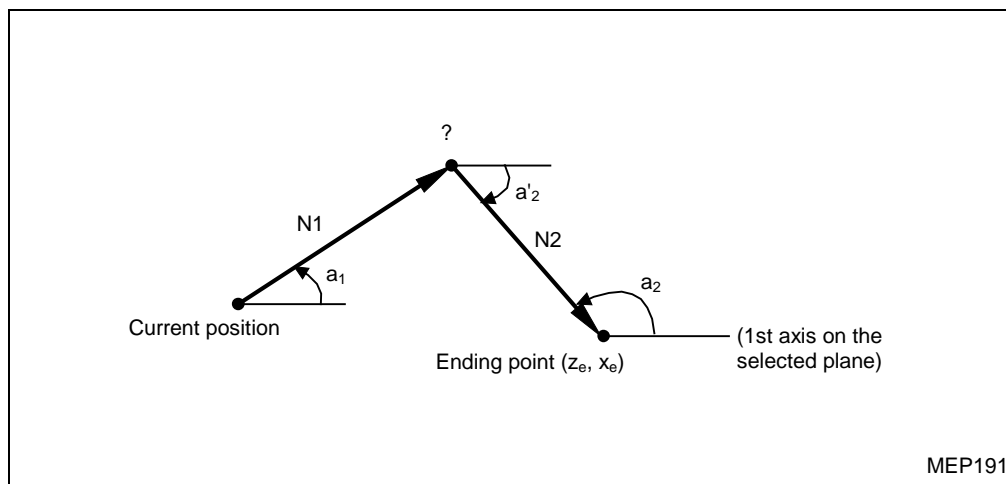
13-14 Geometric Commds

1. Function and purpose

Even if it is difficult to find the crossing point of two lines using linear interpolation commands, setting the slope of the first line and the absolute coordinates of the ending point of the second line and its slope will allow the NC unit to calculate the coordinates of the ending point of the first line and thus to control move commands.

2. Programming format

G18	Specify the intended plane using G17, G18, or G19.
N1 G01 Aa ₁ Ff ₁	Specify the angle and speed for the first block.
N2 Xx _e Zz _e Aa ₂ (a' ₂) Ff ₂	Specify the absolute coordinates of the ending point of the next block, angles, and a speed.

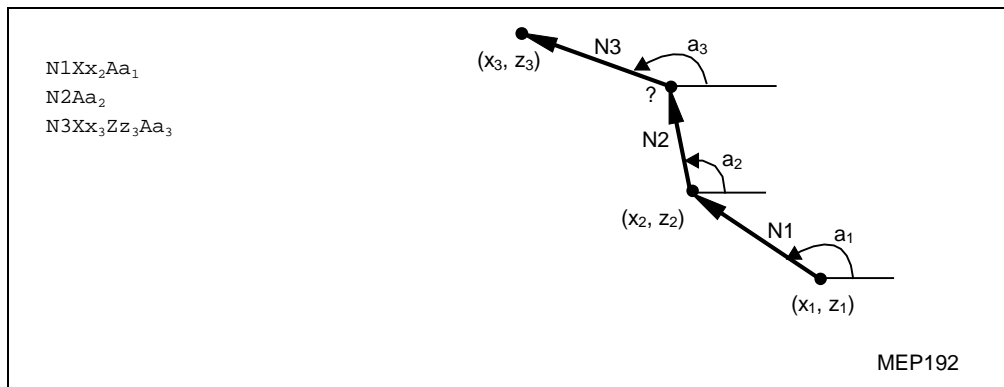


3. Detailed description

- The slope of a line denotes an angle relative to the plus (+) direction of the first axis (horizontal axis) on the selected plane. Assign the sign + for a counterclockwise direction (CCW), or the sign – for a clockwise direction (CW).
- The range of the slope a must be $-360.000^\circ \leq a \leq +360.000^\circ$.
- For the second block, the slope at either the starting point or the ending point can be set. The NC unit will identify whether the specified slope is for the starting point or for the ending point.
- The coordinates of the ending point of the second block must be specified using absolute data. Otherwise a program error will result.
- Any speed can be specified for each block.
- A program error will result if the angle of the crossing point of the two lines is 1 degree or less.
- A program error will result if the preselected plane for the first block is changed over at the second block.
- The geometric command function does not work if address A is to be used for an axis name or for the No. 2 auxiliary function.
- Single-block stop can be used at the ending point of the first block.
- A program error will result if the first block or the second block is not linear.

4. Correlationships to other functions

Geometric command can be set following a linear angle command.



13-15 Corner Chamfering and Corner Rounding Commands

Automatic chamfering (or rounding) at any angle becomes possible by adding characters “C_” or “, R_” at the end of the first block of the two blocks that use only lines to generate corners.

13-15-1 Corner chamfering (, C_)

1. Function and purpose

The sections before and after virtual corners, for which it is assumed that no chamfering is to take place, are chamfered over a distance which is specified in C_.

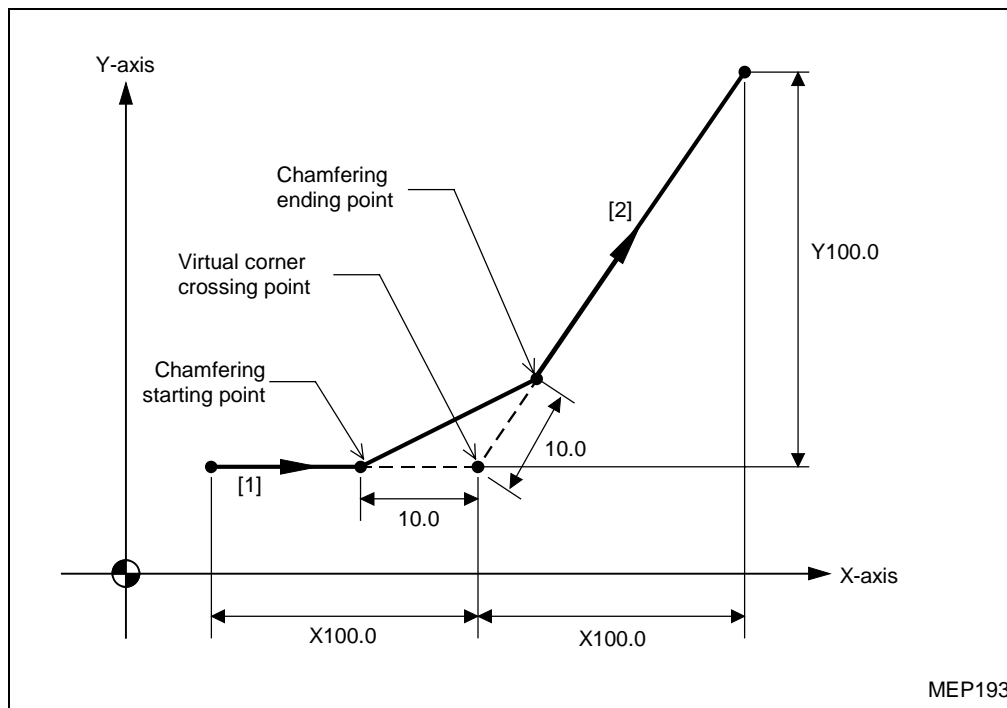
2. Programming format

$$\begin{array}{l} \text{N100 G01 X_Y_ , C_} \\ \text{N200 G01 X_Y_} \end{array} \left. \vphantom{\begin{array}{l} \text{N100 G01 X_Y_ , C_} \\ \text{N200 G01 X_Y_} \end{array}} \right\} \begin{array}{l} \text{Chamfering will be executed at the crossing point of N100 and} \\ \text{N200.} \end{array}$$

$$\left. \vphantom{\begin{array}{l} \text{N100 G01 X_Y_ , C_} \\ \text{N200 G01 X_Y_} \end{array}} \right\} \text{Distance from the virtual corner to the starting or ending point of} \\ \text{chamfering}$$

3. Sample program

```
[1] G91G01X100.,C10.
[2] X100.Y100.
```



4. Detailed description

- The starting point of the next block of corner chamfering is a virtual corner crossing point.
- If character C is not preceded by a comma (,), that command will be regarded as a C-code command.
- If one block contains both “, C_” and “, R_”, only the last issued command will become valid.
- Tool offset data is calculated for the shape existing after completion of corner chamfering.
- Scaling, if already set, also works for the amount of corner chamfering.
- Alarm **912 NO MOTION COMMAND AFTER R/C** results if the command in the block immediately succeeding that of the corner-chamfering command is not a linear-machining command.
- Alarm **913 INCORRECT R/C COMMAND** results if the distance of movement, specified in the corner-chamfering command block, is less than the amount of chamfering.
- Alarm **914 INCORRECT COMMAND AFTER R/C** results if the distance of movement, specified in the block immediately succeeding the corner chamfering command block, is less than the amount of chamfering.
- Alarm **911 CORNER R/C OPTION NOT FOUND** results if the command in the block immediately succeeding that of the corner chamfering command is an arc-machining command.

13-15-2 Corner rounding (,R_)

1. Function and purpose

The virtual corners to be used when it is assumed that no corner rounding is to take place for the corners that each consist of only lines, are rounded using the arcs whose radii are specified using R_.

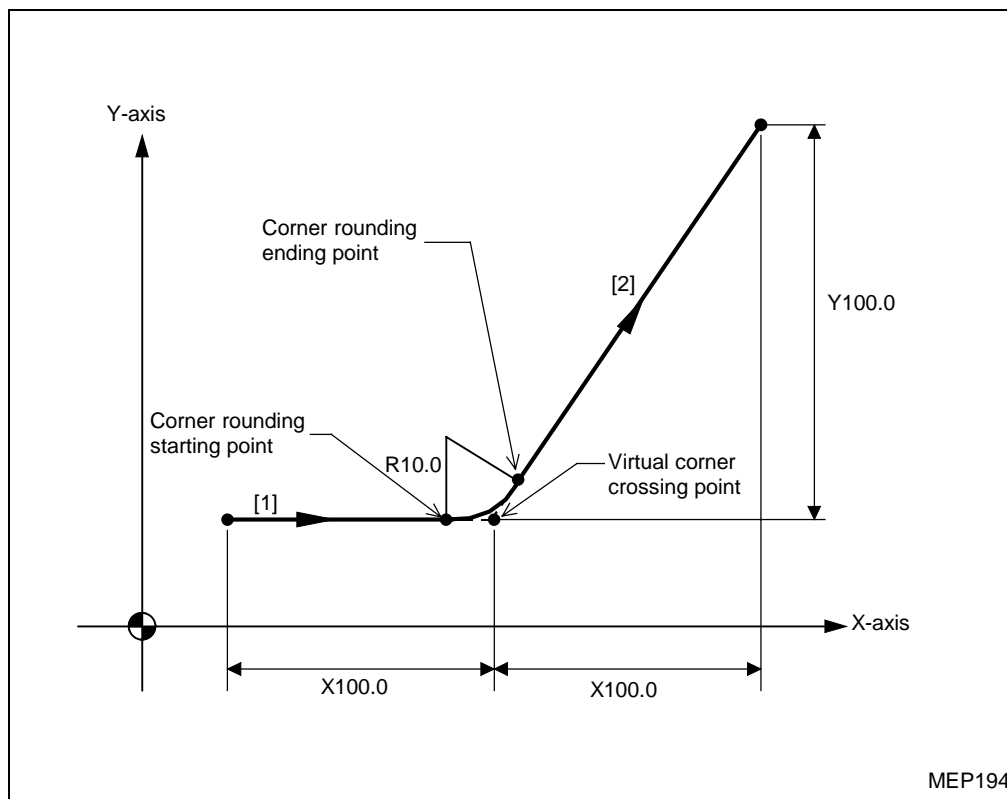
2. Programming format

N100 G01 X_Y_,R_	}	Rounding will be executed at the crossing point of N100 and N200.
N200 G01 X_Y_		
		Radius of the arc for corner rounding

3. Sample program

[1] G91G01X100.,R10.

[2] X100.Y100.



4. Detailed description

- The starting point of the next block of corner rounding is a virtual corner crossing point.
- If character R is not preceded by a comma (,), that command will be regarded as an R-code command.
- If one block contains both “, C_” and “, R_”, only the last issued command will become valid.
- Tool offset data is calculated for the shape existing after completion of corner rounding.
- Scaling, if already set, also works for the amount of corner rounding.
- Alarm **912 NO MOTION COMMAND AFTER R/C** results if the command in the block immediately succeeding that of the corner-rounding command is not a linear-machining command.
- Alarm **913 INCORRECT R/C COMMAND** results if the distance of movement, specified in the corner-rounding command block, is less than the amount of rounding.
- Alarm **914 INCORRECT COMMAND AFTER R/C** results if the distance of movement, specified in the block immediately succeeding the corner-rounding command block, is less than the amount of rounding.
- Alarm **911 CORNER R/C OPTION NOT FOUND** results if the command in the block immediately succeeding that of the corner-rounding command is an arc-machining command.

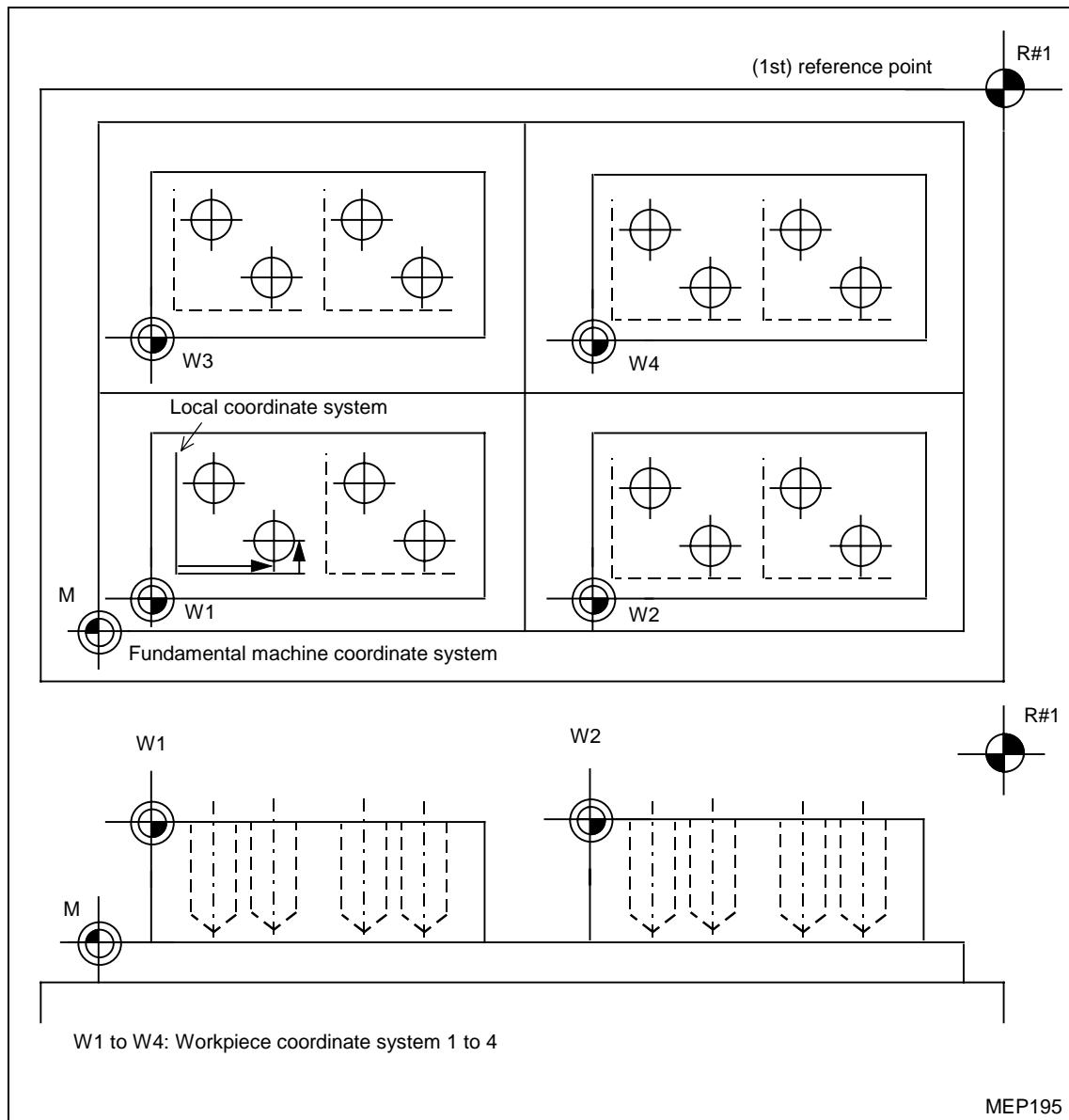
- NOTE -

14 COORDINATE SYSTEM SETTING FUNCTIONS

14-1 Fundamental Machine Coordinate System, Workpiece Coordinate Systems, and Local Coordinate Systems

The fundamental machine coordinate system, which is a fixed coordinate system for the machine, is used to designate the tool change position, stroke end position, etc. that are predetermined for the machine.

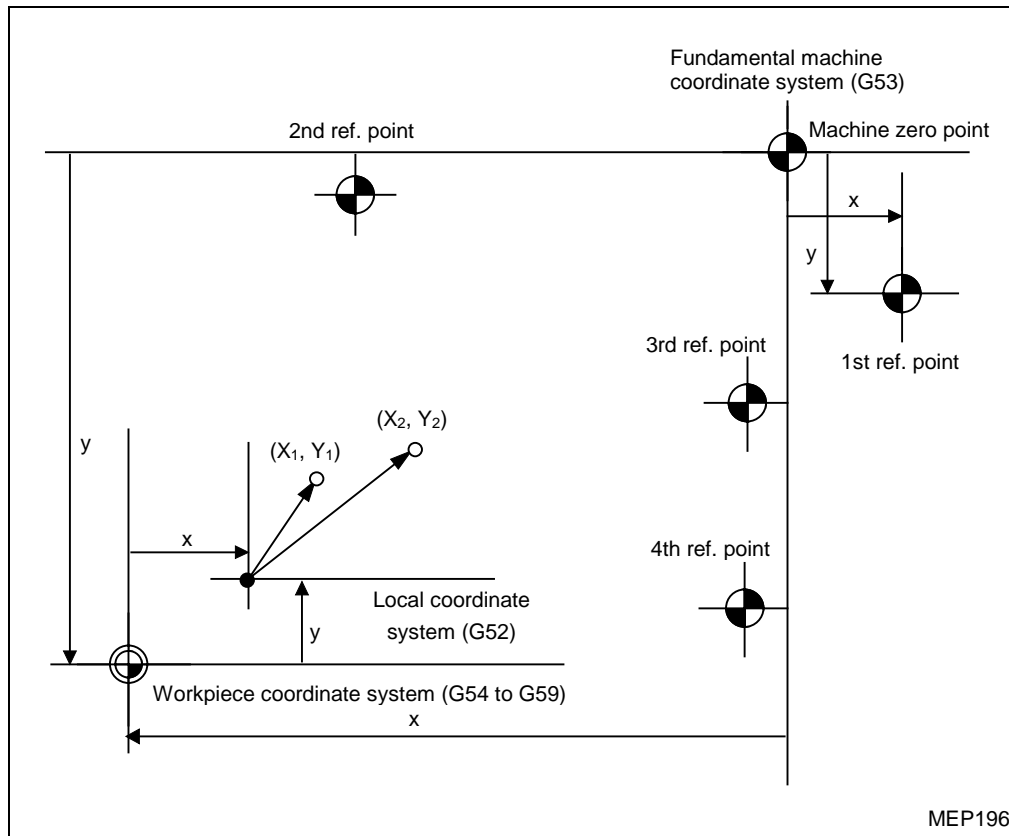
Workpiece coordinate systems are the coordinate systems to be used by a programmer when creating programs. These coordinate systems have their origins set at reference points on workpieces. Local coordinate systems are the coordinate systems to be set on the workpiece coordinate systems to facilitate the creating of partial-machining programs.



14-2 Machine Zero Point and Second, Third, and Fourth Reference Points

The machine zero point refers to a position that acts as a reference point for the fundamental machine coordinate system. That is, the machine zero point is a point specific to the machine, determined by return to the reference point (zero point).

The second, third, and fourth reference points (zero point) are the points whose positions in the fundamental machine coordinate system are parameter-preset.



14-3 Fundamental Machine Coordinate System Selection: G53

1. Function and purpose

The fundamental machine coordinate system is used to designate the tool change position, stroke end position, etc. that are predetermined for the machine.

Command G53 and its succeeding coordinate words move the tool at the rate of rapid traverse to the designated position on the fundamental machine coordinate system.

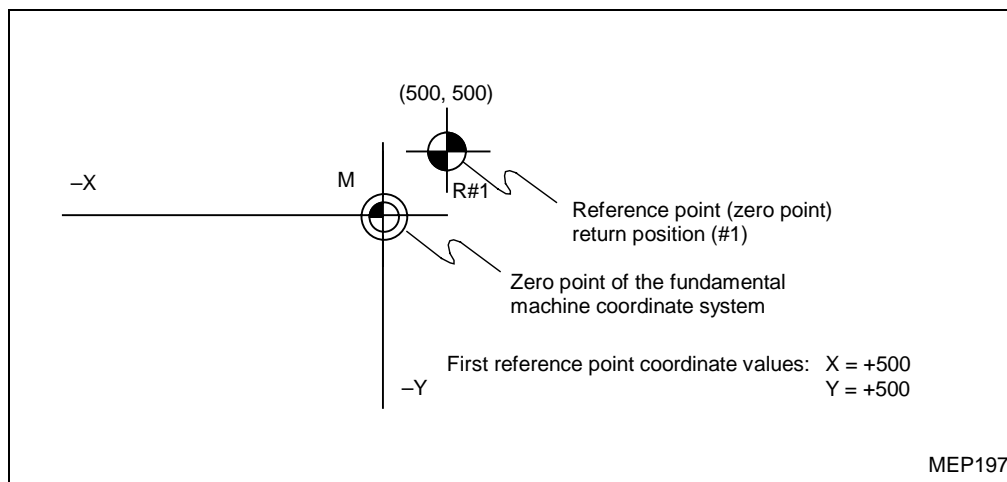
2. Programming format

Selection of the fundamental machine coordinate system:

(G90) G53 Xx Yy Zz $\alpha\alpha$ (α : Additional axis)

3. Detailed description

- When power is turned on, the fundamental machine coordinate system will be set according to the reference point (zero point) return position which is determined by a manual or automatic reference point (zero point) return command.
- The fundamental machine coordinate system is not updated by command G92.
- Command G53 is valid only for designated blocks.
- During the incremental data command mode (G91), command G53 moves the tool on the selected coordinate system according to incremental data.
- Setting of command G53 does not cancel the tool diameter offset data of the designated axis.
- The coordinate data of the first reference point represents the distance from point 0 on the fundamental machine coordinate system to the reference point (zero point) return position.



14-4 Coordinate System Setting: G92

1. Function and purpose

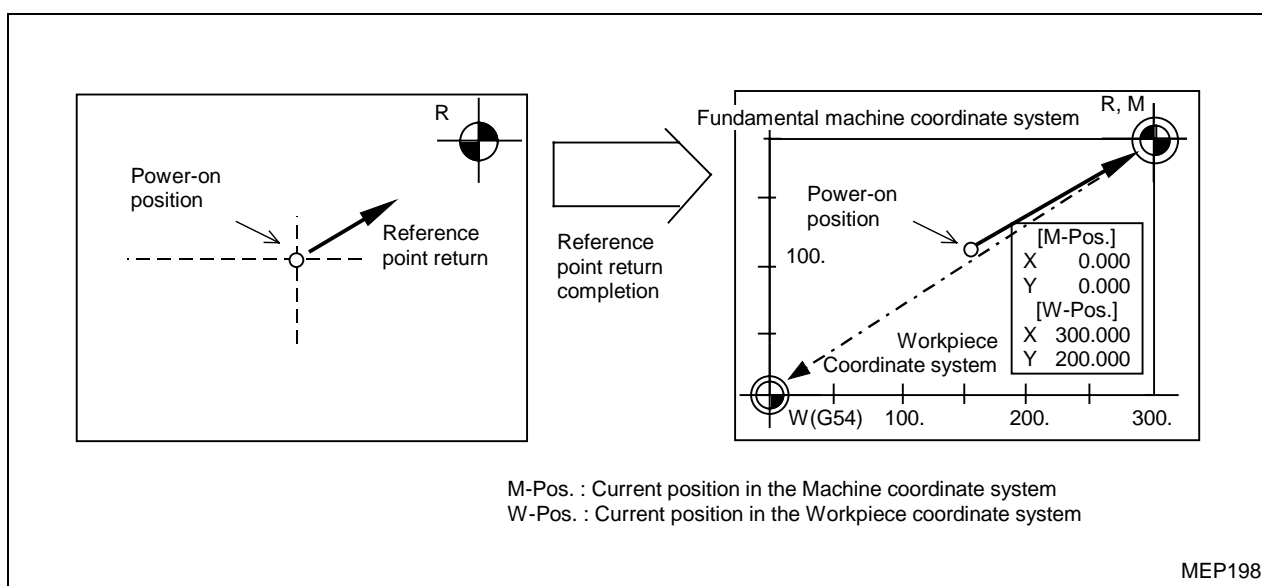
Setting of command G92 allows absolute coordinate system and current position display data to be newly preset exactly as programmed, without ever having to move the machine.

2. Programming format

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

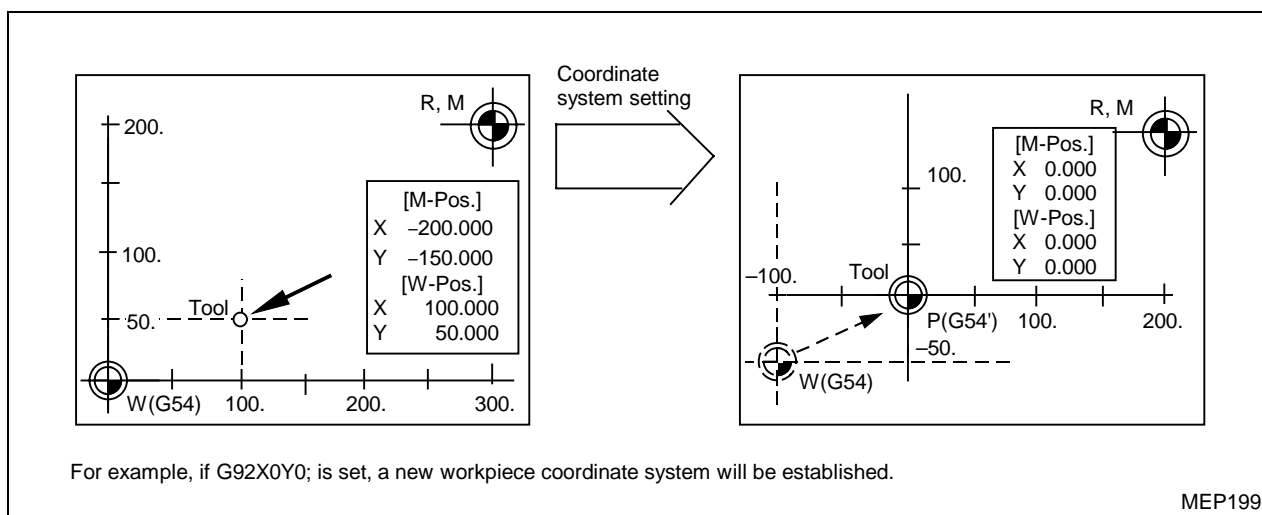
3. Detailed description

- After power-on, return to the reference point is initially performed using watchdogs. Coordinate systems are set automatically on completion of return.



Fundamental machine and workpiece coordinate systems are established at predetermined positions.

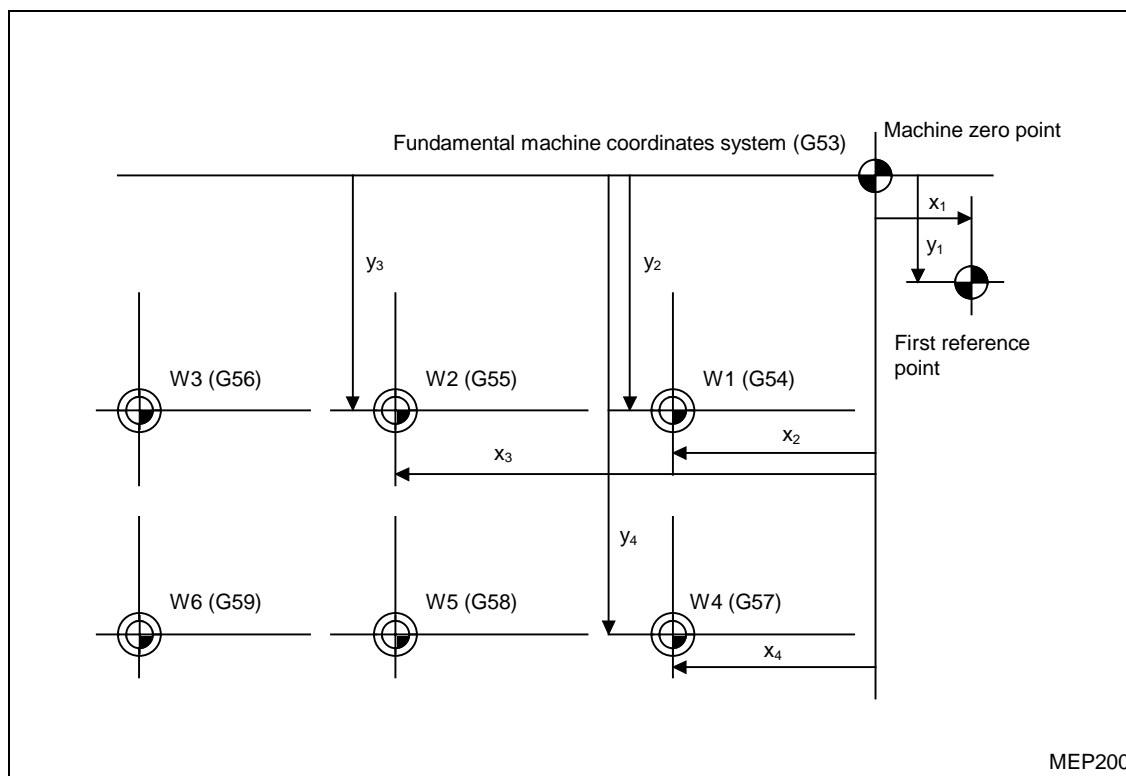
- Setting of command G92 allows absolute (workpiece) coordinate system and current position display data to be newly preset exactly as programmed, without ever having to move the machine.



14-5 Automatic Coordinate System Setting

The automatic coordinate system setting function allows various coordinate systems to be generated according to parameters previously set on the operation panel when the first manual reference point return or watchdog-based reference point return is completed.

Use the automatically set coordinate systems when creating machining programs.

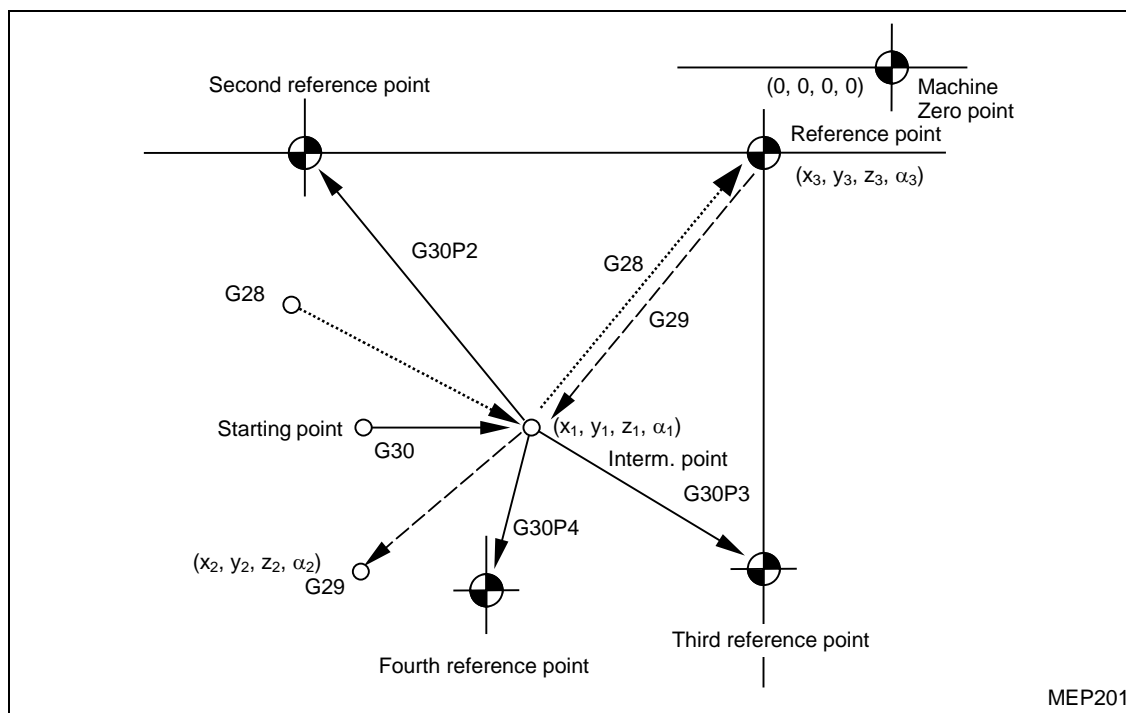


- This function generates the following coordinate systems:
 - 1) Fundamental machine coordinate system (G53)
 - 2) Workpiece coordinate systems (G54 through G59)
- Parameters for the NC unit must be the distance data from the origin of the fundamental machine coordinate system. You must therefore determine where on the fundamental machine coordinate system the first reference point is to be set, and then set the zero point of the workpiece coordinate system.

14-6 Reference Point Return: G28, G29

1. Function and purpose

- G28 command first performs a G00 (rapid) positioning to the specified intermediate position along the specified axes, and then a returning to the first reference point at the rapid feed rate independently along each specified axis.
- G29 command first performs a returning to the intermediate point of the last G28 or G30 command at the rapid feed rate independently along each specified axis, and then a G00 (rapid) positioning to the specified position.



2. Programming format

- G28 $Xx_1 Yy_1 Zz_1 \alpha\alpha_1$ (α : Additional axis) [Automatic reference point return]
- G29 $Xx_2 Yy_2 Zz_2 \alpha\alpha_2$ (α : Additional axis) [Start point return]

3. Detailed description

1. Command G28 is equivalent to the following commands:

$$\left[\begin{array}{l} G00 Xx_1 Yy_1 Zz_1 \alpha\alpha_1 \\ G00 Xx_3 Yy_3 Zz_3 \alpha\alpha_3 \end{array} \right]$$

where x_3, y_3, z_3 and α_3 denote the coordinates of the reference point, that are parameter-set as the distance from the zero point of the fundamental machine coordinate system.

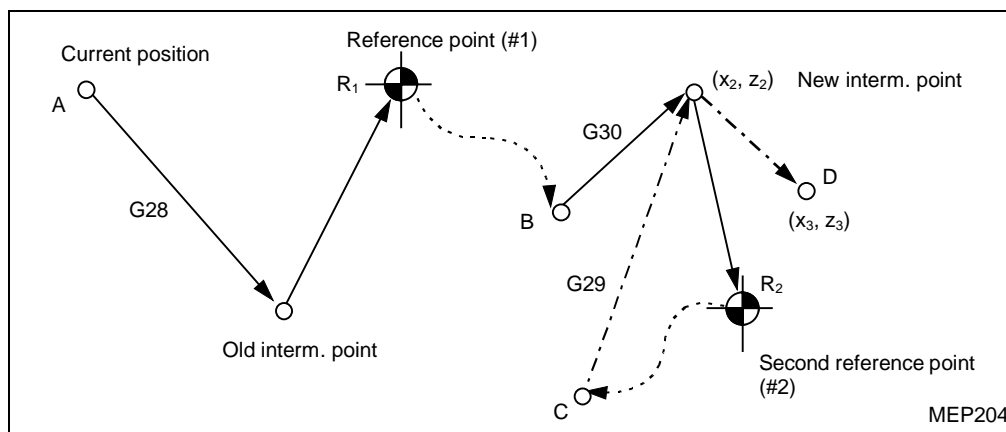
2. Axes that have not been returned to the reference point in manual mode after power-on are returned using the watchdog method. In that case, the direction of return is regarded as the same as the sign-designated direction. The direction of return will not be checked if it is of the straight-return type. For the second time onward, the axes are returned at high speed to the reference point that was stored into the memory by execution of the first return command the direction is not checked at this time, either.

3. When reference point (zero point) return is completed, a return complete output signal will be output and the CRT monitor of the operation panel will display “#1” in the display field of the axis name.
4. Command G29 is equivalent to the following commands:

$$\left[\begin{array}{l} G00 Xx_1 Yy_1 Zz_1 \alpha\alpha_1 \\ G00 Xx_2 Yy_2 Zz_2 \alpha\alpha_2 \end{array} \right] \quad \begin{array}{l} \text{Independent rapid feed on each axis} \\ \text{(not the same as for G0).} \end{array}$$

where x_1 , y_1 , z_1 and α_1 are the coordinates of the middle point specified by the last G28 or G30 command.
5. A program error will result if G29 is executed without any preceding G28 (automatic reference point return command) after turning-on.
6. Under machine locked status or Z-axis cancelled status, any movements of the Z-axis up to the middle point are ignored and only subsequent positioning is performed.
7. The coordinates of the intermediate point (x_1, y, z_1, α_1) must be given according to the type of dimensional data input (G90 or G91).
8. G29 command can refer to both G28 and G30, and the positioning along the specified axes is performed through the intermediate point of the last G28 or G30 command.
9. The tool offsetting, if left valid, is cancelled during execution of the return command and the offset data is also cleared.

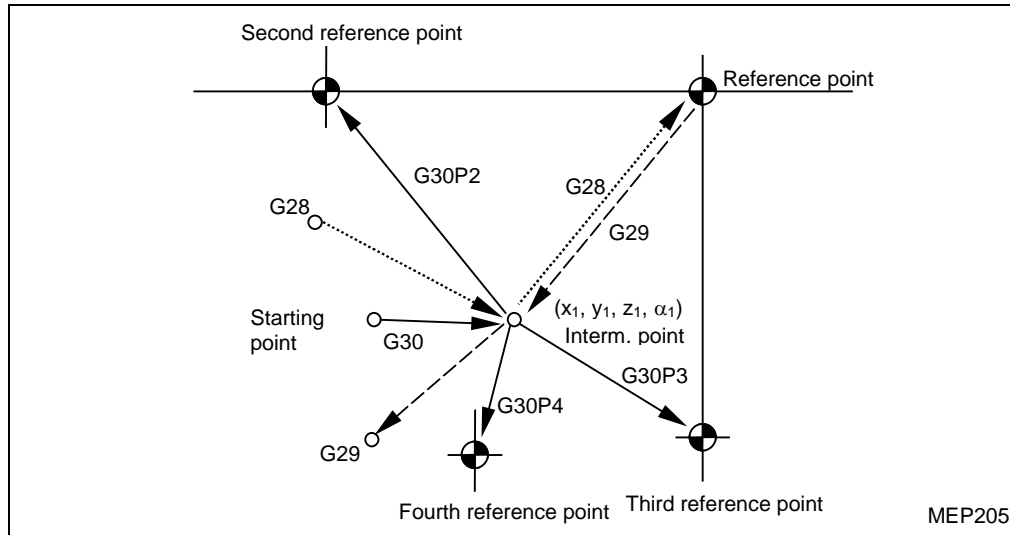
Example: G28X x_1 Z z_1 From point A to reference point
 G30X x_2 Z z_2 From point B to second reference point
 G29X x_3 Z z_3 From point C to point D



14-7 Second, Third, or Fourth Reference Point Return: G30

1. Function and purpose

The returning to the second, third, or fourth reference point can be programmed by setting "G30 P2 (P3, P4)".

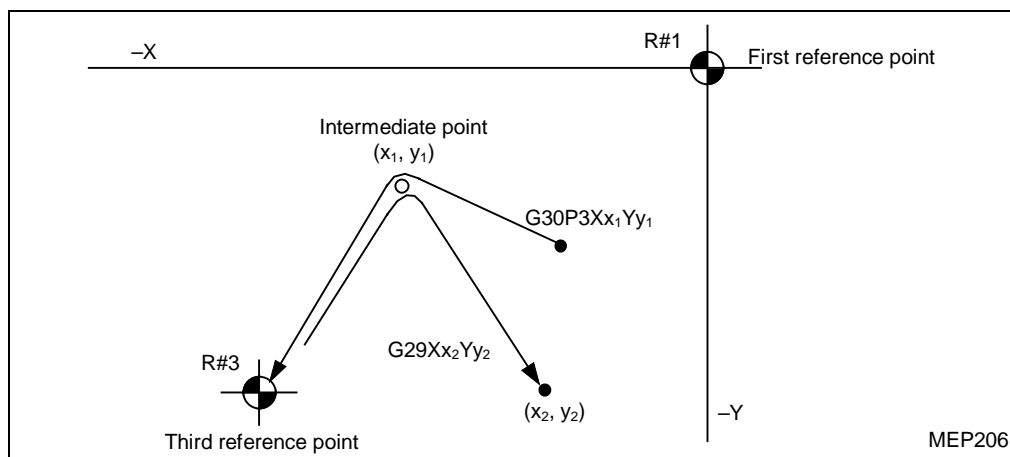


2. Programming format

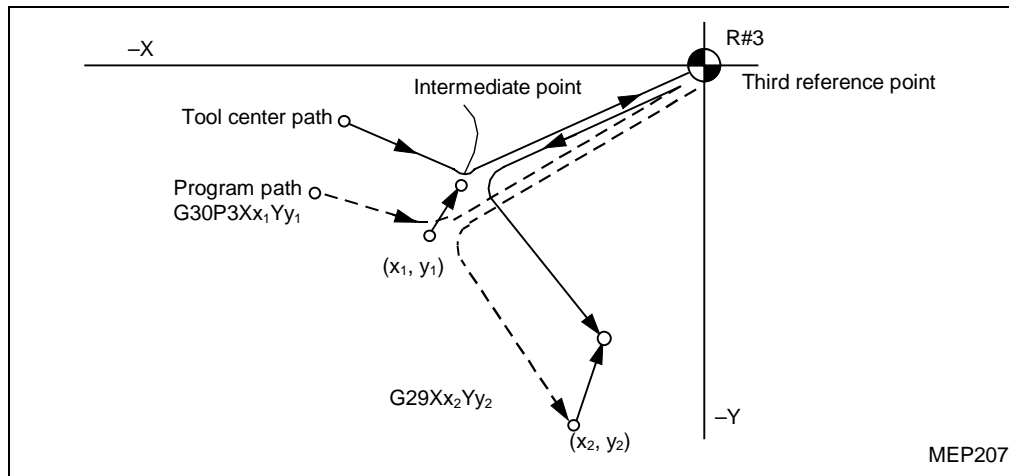
G30 P2 (P3, P4) Xx₁ Yy₁ Zz₁ αα₁ (α: Additional axis)

3. Detailed description

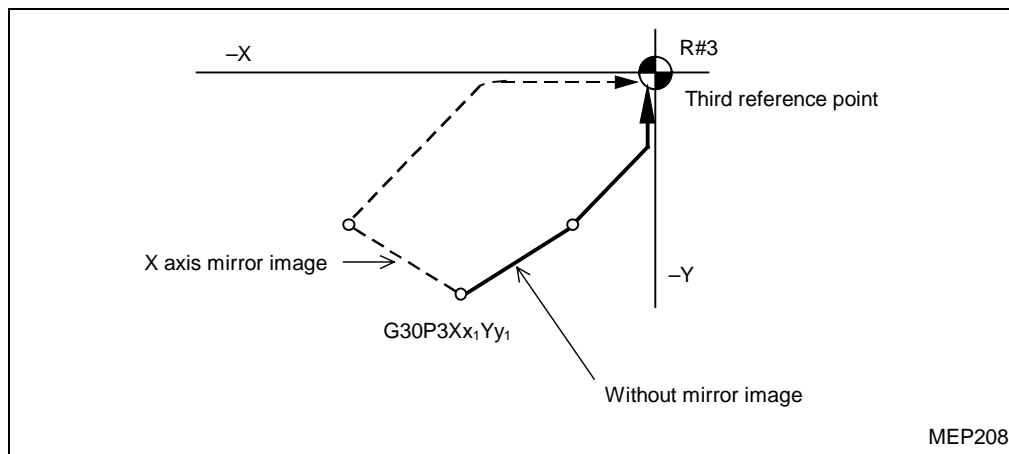
1. Use address P to specify the number of the required reference point (P2, P3 or P4). Return to the second reference point is automatically selected if P-command is omitted or zero, one, five or a greater integer is set at address P.
2. Return to the second, third or fourth reference point is performed through the specified intermediate point like the return to the first reference point.
3. The coordinates of the second, third, or fourth reference point represent the positions specific to the machine. The coordinates can be checked on the CRT monitor of the operation panel.
4. A command of G29 after return to the second, third or fourth reference point is carried out through the intermediate point of the last command for return to the reference point.



5. Tool diameter offsetting function is temporarily cancelled during return to a reference point in the offsetting plane and made valid again for a return (by G29) from there. The cancellation and resumption of offsetting dimensions occur during movement from the intermediate point to the reference point and vice versa.



6. Upon second, third, or fourth reference point return, the tool length offset data for the return axis is cancelled automatically.
7. For the second, third, or fourth reference point return under machine locked status, movement from the middle point to the reference point is skipped. The next block is executed after the designated axis has arrived at the intermediate point.
8. For the second, third, or fourth reference point return in the mirror image mode, the mirror image is valid for movement from the starting point to the intermediate point and the axis moves in an opposite direction to the designated one. For movement from the intermediate point to the reference point, however, the mirror image becomes invalid and thus the axis moves to the reference point.

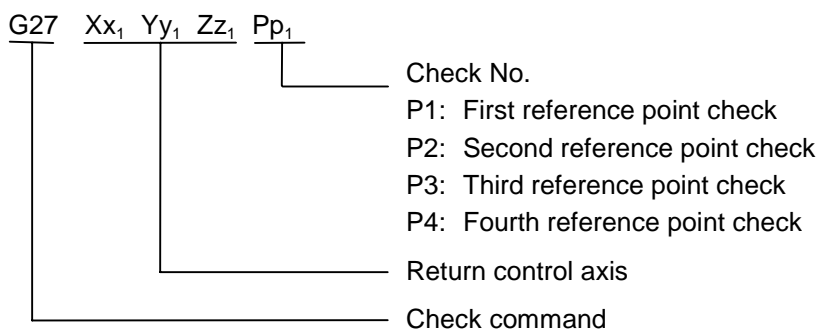


14-8 Reference Point Check Command: G27

1. Function and purpose

G27 command performs a rapid positioning to the specified position, checks the final position for agreement with the specified reference point and, upon confirmation of the agreement, outputs a reference point return complete signal to the machine side as is the case with G28. For a machining program, therefore, of which the starting and the ending positions are the reference point, this function is useful for checking whether the returning, i.e. the program itself, has been correctly executed to the end.

2. Programming format



3. Detailed description

- The first reference point check will occur if the P code is omitted.
- The number of axes for which reference point check can be done at the same time depends on the number of simultaneously controllable axes.
- An alarm will result if the designated reference point is not the final position on completion of this command.

14-9 Workpiece Coordinate System Setting and Selection: (G92) G54 to G59

1. Function and purpose

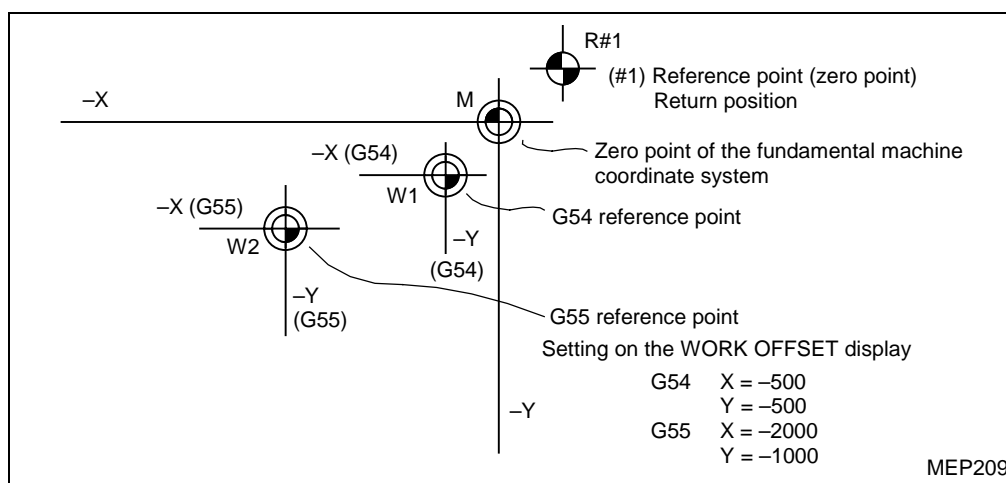
- Workpiece coordinate systems are set to facilitate creation of a workpiece machining program having its origin set at a machining reference point for the intended workpiece. Commands G54 through G59 move the designated axis to the designated position on the workpiece coordinate system corresponding to the command code number. These six types of commands generate respective workpiece coordinate systems.
- Command G92 can be used to modify the current workpiece coordinate system in order that the current tool position be indicated by the specified coordinates. (The current tool position refers to a position that has incorporated tool diameter, tool length, and tool position offset data.)
Command G92 also generates a virtual machine coordinate system according to which the current tool position is indicated by the specified coordinates. (The current tool position refers to a position that has incorporated tool diameter, tool length, and tool position offset data.)

2. Programming format

- Selecting a workpiece coordinate system (G54 to G59)
(G90) G54 Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$ (α : Additional axis)
- Setting the workpiece coordinate system (G54 to G59)
(G54) G92 Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$ (α : Additional axis)

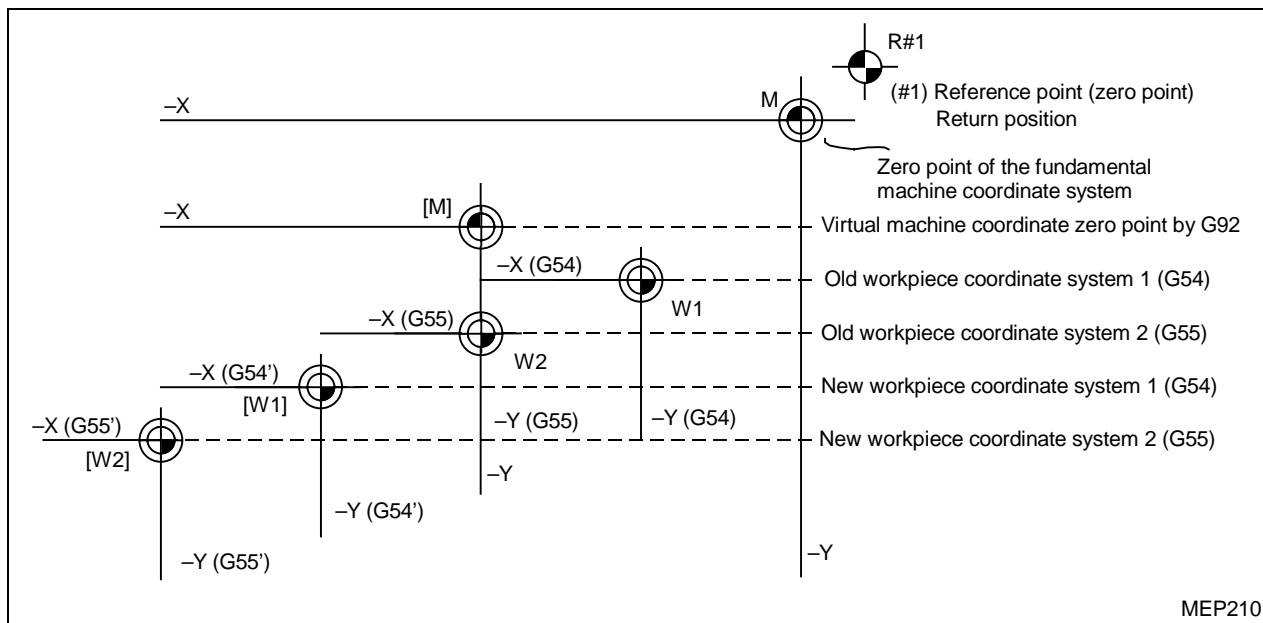
3. Detailed description

1. Tool diameter offset data for the designated axis is not cancelled by the selection of a new workpiece coordinate system using commands G54 to G59.
2. The coordinate system corresponding to G54 will always be selected when power is turned on.
3. Commands G54 through G59 are modal commands (of group 12).
4. Command G92 only moves the workpiece coordinate system.
5. The workpiece origins (reference points) of G54 to G59 must be externally preset with reference to the fundamental machine coordinate system.



6. Settings of workpiece origin can be updated repeatedly (manually or by programming "G10 L2 Pp₁ Xx₁ Yy₁ Zz₁").

7. New workpiece coordinate system 1 (G54) can be generated by setting G92 during the mode of G54. At the same time, other workpiece coordinate systems 2 through 6 (G55 through G59) are shifted parallel to the new workpiece coordinate system 1.



A virtual machine coordinate system is also formed in accordance with shifting of workpiece coordinate system by G92.

After power has been turned on, the virtual machine coordinate system will coincide with the fundamental machine coordinate system by the first automatic (G28) or manual reference point return.

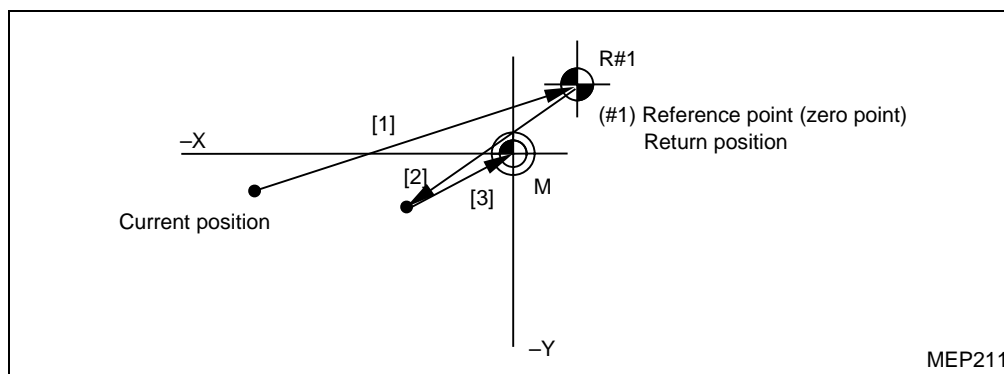
When a virtual machine coordinate system is generated, new workpiece coordinate systems will be set by using the preset origins with reference to the virtual machine coordinate system.

8. After power has been turned on, the fundamental machine coordinate system and workpiece coordinate systems will be set automatically according to the presets when the first automatic (G28) or manual reference point return is completed.

4. Sample programs

Example 1:

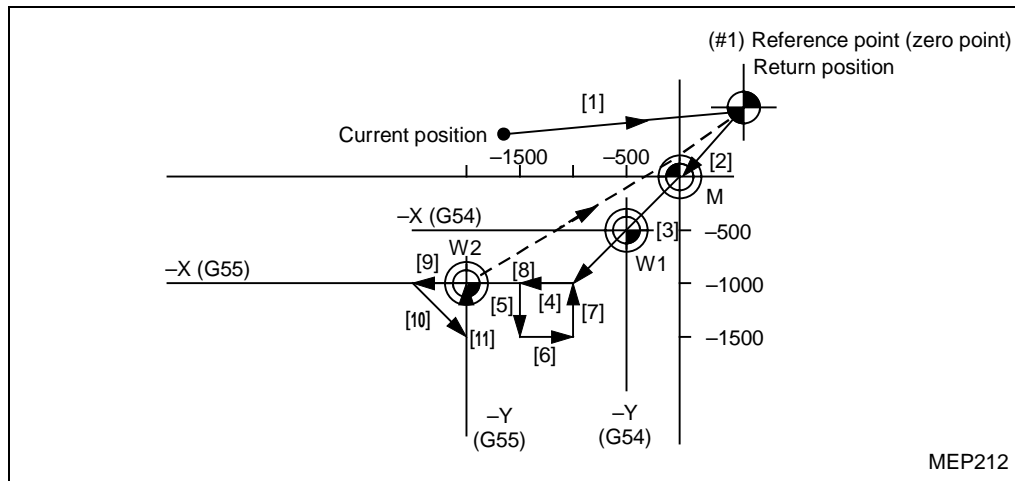
```
[1] G28X0Y0
[2] G53X-100.Y-50.
[3] G53X0Y0
```



If the coordinate data of the first reference point is 0 (zero), the zero point (origin) of the fundamental machine coordinate system and the reference point return position (#1) will agree.

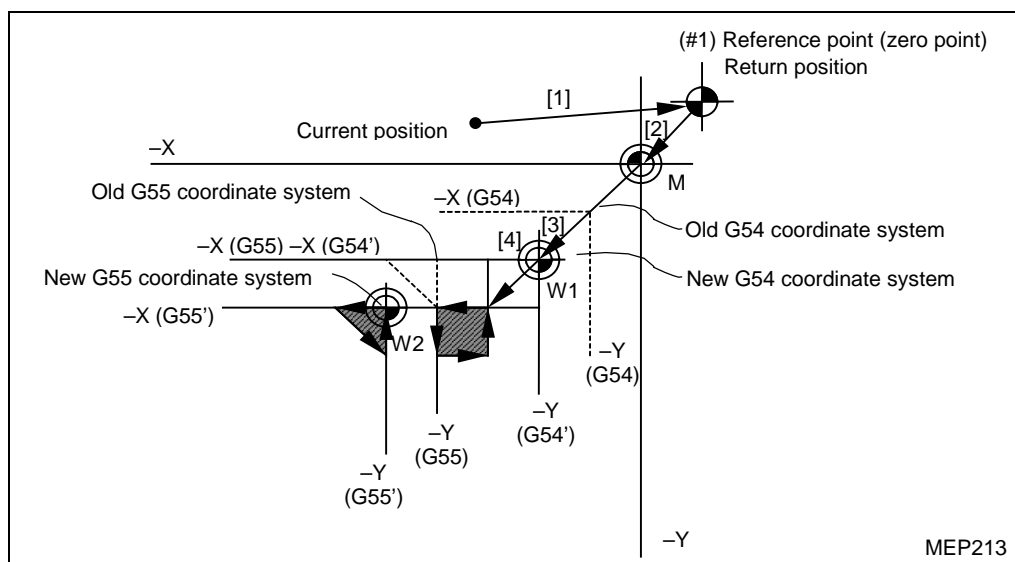
Example 2:

[1] G28X0Y0	[7] Y+500.
[2] G90G00G53X0Y0	[8] G90G00G55X0Y0
[3] G54X-500.Y-500.	[9] G01X-500.F200
[4] G01G91X-500.F100	[10] X0Y-500.
[5] Y-500.	[11] G90G28X0Y0
[6] X+500.	



Example 3: Same machining as is given in **Example 2** after shifting through $(-500, -500)$ the workpiece coordinate system of G54 (provided that [3] through [10] in Example 2 have been registered in subprogram O1111):

[1] G28X0Y0	
[2] G90G00G53X0Y0	(Not required if #1 ref. pt. = machine origin)
[3] G54X-500.Y-500.	Shift amount of workpiece coordinate system
[4] G92X0Y0	New workpiece coordinate system setting
[5] M98P1111	



Note: If blocks [3] through [5] are used repeatedly, reference point return command G28 must be set at the end of the program since the workpiece coordinate systems will shift each time those blocks are executed.

Example 4: If one and the same machining is to be done for each of six identical workpieces placed on the coordinate systems of G54 through G59:

A. Setting of workpiece origins

Workpiece	1	X = -100.000	Y = -100.000	G54
	2	X = -100.000	Y = -500.000	G55
	3	X = -500.000	Y = -100.000	G56
	4	X = -500.000	Y = -500.000	G57
	5	X = -900.000	Y = -100.000	G58
	6	X = -900.000	Y = -500.000	G59

B. Machining program (Subprogram)

```

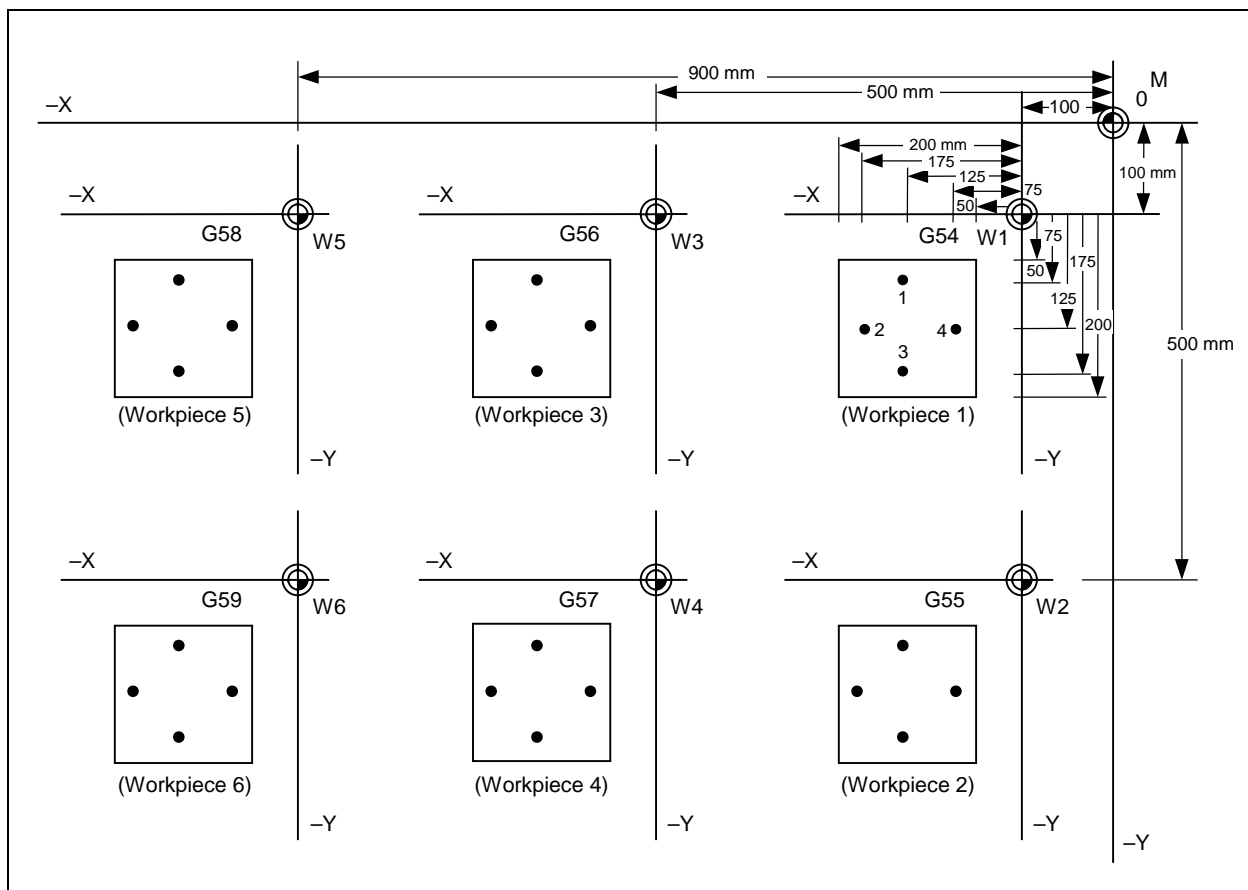
O100
N1G90G00G43X-50.Y-50.Z-100.H10      } Positioning
N2G01X-200.F50
      Y-200.
      X-50.
      Y-50.
N3G28X0Y0Z0
T□□M06
N4G98G81X-125. Y-75.Z-150.R-95.F40    } Drilling 1
      X-175. Y-125.                    } Drilling 2
      X-125. Y-175.                    } Drilling 3
      X-75.  Y-125.                    } Drilling 4
      G80
N5G28X0Y0Z0
      {
N6G98G84X-125. Y-5.Z-150.R-95.F40    } Tapping 1
      X-175. Y-125.                    } Tapping 2
      X-125. Y-175.                    } Tapping 3
      X-75.  Y-125.                    } Tapping 4
      G80
M99

```

C. Positioning program (Main program)

```

G28X0Y0Z0      } At power-on
N1 G90G54M98P100
N2      G55M98P100
N3      G57M98P100
N4      G56M98P100
N5      G58M98P100
N6      G59M98P100
N7 G28X0Y0Z0
N8 M02
%
```



14-10 Additional Workpiece Coordinate System Setting and Selection: G54.1 (Option)

1. Function and purpose

In addition to the six standard systems G54 to G59, up to 48 sets of workpiece origin data can be used to facilitate program creation.

Note 1: Local coordinate system setting is not available in G54.1 mode.

Note 2: Setting a G52-command during G54.1 mode will cause the alarm **949 NO G52 IN G54.1 MODE**.

2. Programming format

A. Selection of a workpiece coordinate system

G54.1 Pn (n = 1 to 48)

Example: G54.1 P48 Selection of P48 system

Note: Omission of P and setting of "P0" function the same as "P1". Setting a value other than integers from 0 to 48 at address P causes the alarm **809 ILLEGAL NUMBER INPUT**.

B. Movement in a workpiece coordinate system

G54.1 Pn (n = 1 to 48)

G90 Xx Yy Zz

Example: G54.1 P1 Selection of P1 system
G90 X0 Y0 Z0 Movement to P1-system origin (0, 0, 0)

C. Setting of workpiece origin data

G10 L20 Pn Xx Yy Zz (n = 1 to 48)

Example: G90 G10 L20 P30 X-255.Y-50. Data at addresses X and Y are set as data of P30-system origin.
G91 G10 L20 P30 X-3.Y-5. Data at addresses X and Y are added to data of P30-system origin.

3. Detailed description

A. Remarks on omission of P and/or L

G10 L20 Pn Xx Yy Zz When n = 1 to 48: Correct setting of data for Pn-system origin
Otherwise: Alarm **809 ILLEGAL NUMBER INPUT**

G10 L20 Xx Yy Zz Correct setting of workpiece origin data for the current system, except for G54- to G59-system (in which case: Alarm **807 ILLEGAL FORMAT**)

G10 Pn Xx Yy Zz or Correct setting of workpiece origin data for the current system
G10 Xx Yy Zz

B. Precautions for programming

- Do not set together in a block of G54.1 or L20 any G-code that can refer to address P.

Such G-codes are for example:

G04 Pp Dwell
 G30 Pp Reference-point return
 G72 to G89 Fixed cycle
 G65 Pp, M98 Pp Subprogram call

- Setting the G54.1-command without the option will cause the alarm **948 NO G54.1 OPTION**.
- Setting "G10 L20" without the option will cause the alarm **903 ILLEGAL G10 L NUMBER**.
- Local coordinate system setting is not available in G54.1 mode. Setting a G52-command during G54.1 mode will cause the alarm **949 NO G52 IN G54.1 MODE**.

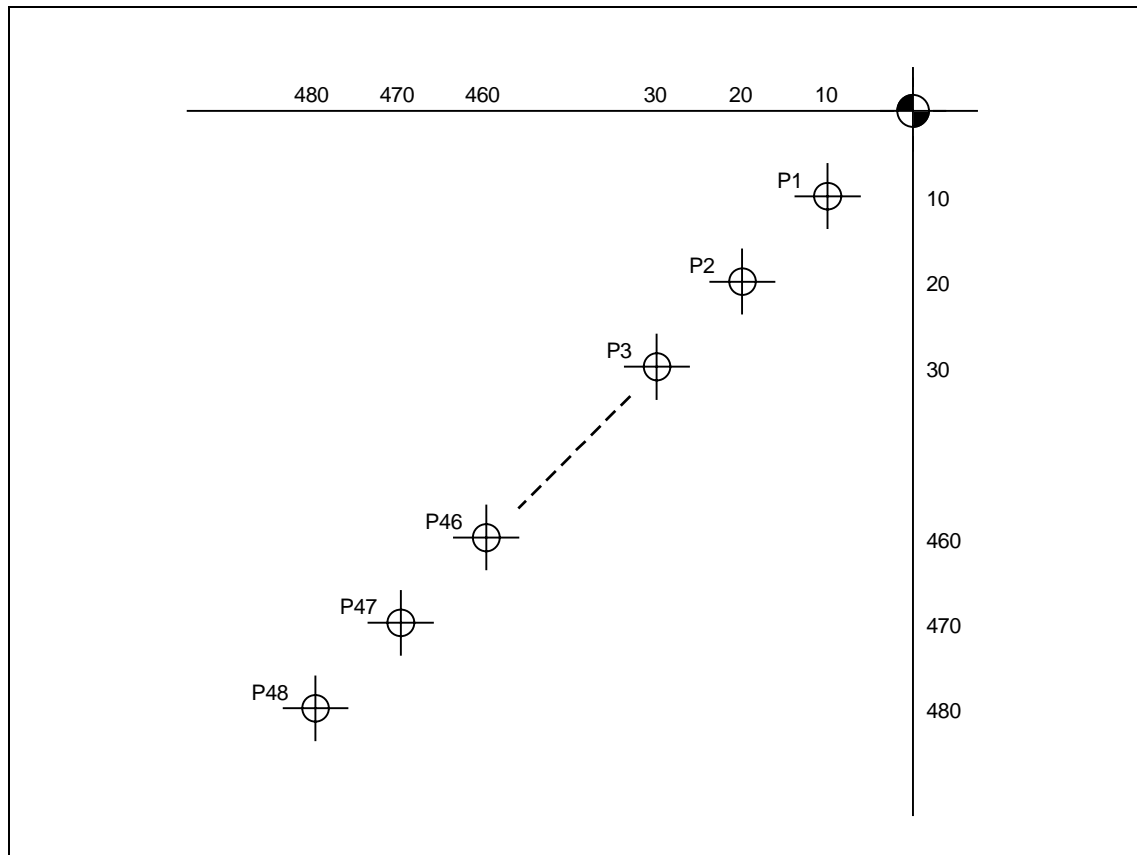
C. Related system variables

The origin data of additional workpiece coordinate systems are assigned to system variables as listed-up in the following table:

	1 axis	to	6 axis		1 axis	to	6 axis		1 axis	to	6 axis
P1	#7001	to	#7006	P17	#7321	to	#7326	P33	#7641	to	#7646
P2	#7021	to	#7026	P18	#7341	to	#7346	P34	#7661	to	#7666
P3	#7041	to	#7046	P19	#7361	to	#7366	P35	#7681	to	#7686
P4	#7061	to	#7066	P20	#7381	to	#7386	P36	#7701	to	#7706
P5	#7081	to	#7086	P21	#7401	to	#7406	P37	#7721	to	#7726
P6	#7101	to	#7106	P22	#7421	to	#7426	P38	#7741	to	#7746
P7	#7121	to	#7126	P23	#7441	to	#7446	P39	#7761	to	#7766
P8	#7141	to	#7146	P24	#7461	to	#7466	P40	#7781	to	#7786
P9	#7161	to	#7166	P25	#7481	to	#7486	P41	#7801	to	#7806
P10	#7181	to	#7186	P26	#7501	to	#7506	P42	#7821	to	#7826
P11	#7201	to	#7206	P27	#7521	to	#7526	P43	#7841	to	#7846
P12	#7221	to	#7226	P28	#7541	to	#7546	P44	#7861	to	#7866
P13	#7241	to	#7246	P29	#7561	to	#7566	P45	#7881	to	#7886
P14	#7261	to	#7266	P30	#7581	to	#7586	P46	#7901	to	#7906
P15	#7281	to	#7286	P31	#7601	to	#7606	P47	#7921	to	#7926
P16	#7301	to	#7306	P32	#7621	to	#7626	P48	#7941	to	#7946

4. Sample programs

- Consecutive setting of origin data for all the 48 sets of additional workpiece coordinate system



Setting in format "G10L20PpXxYyZz"

O100

#100=1

#101=10.

WHILE[#100LT49]DO1

G90G10L20P#100X#101Y#101

#100=#100+1

#101=#101+10.

END1

M30

%

P-No. initial.

Origin setting

P-No. count-up

Setting in assignment of variables

O200

G90

#100=7001

#101=10.

#102=1

WHILE[#102LT49]DO1

#103=0

WHILE[#103LT2]DO2

#[#100]=#101

#100=#100+1

#103=#103+1

END2

#100=#100+19

#101=#101+10.

#102=#102+1

END1

M30

%

Sys.-var.-No. initial.

Counter initial.

Counter initial.

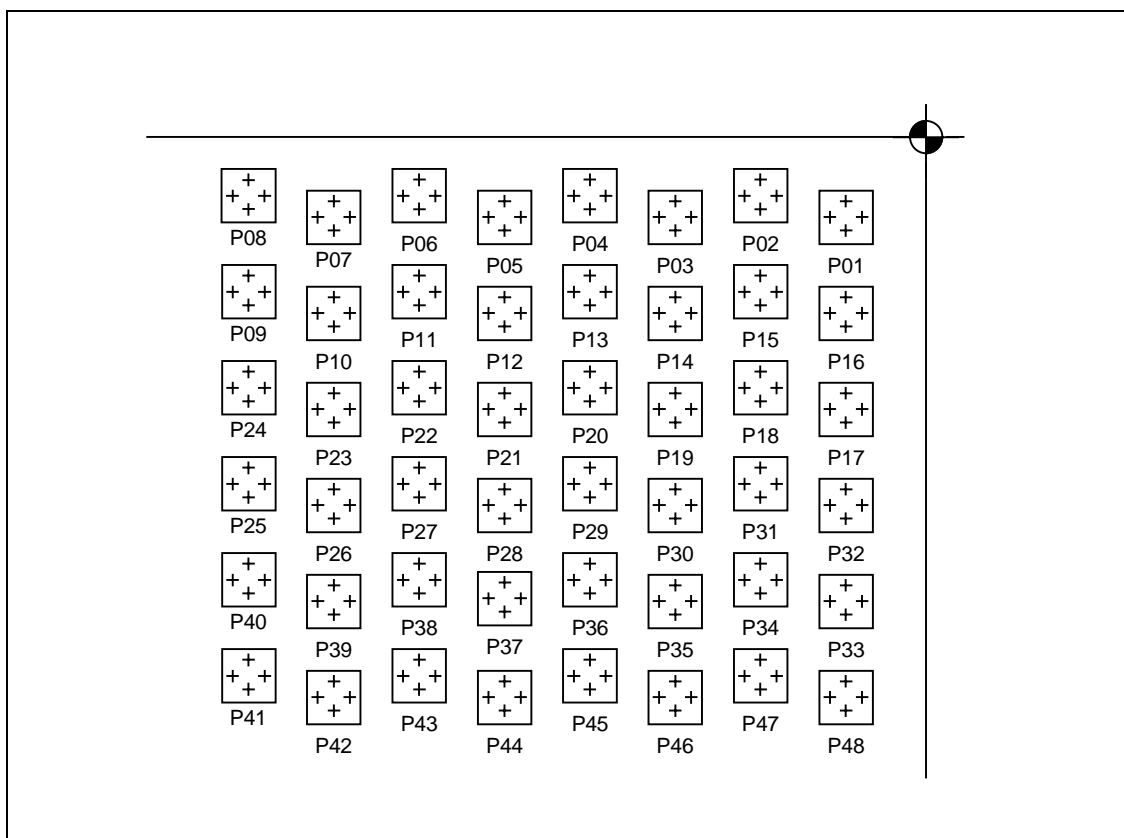
Sys.-var. Setting

Sys.-var.-No. count-up

Counter count-up

2. Consecutive application of all the 48 sets of additional workpiece coordinate system

Provided that preparatory setting of origin data in P1 to P48 is completed in accordance with the 48 workpieces fixed on the table in the arrangement shown in the figure below:



O1000 (Main prg.)

```
G28XYZ
#100=1
G90
WHILE[#100LT49]DO1
G54.1P#100
M98P1001
#100=#100+1
END1
G28Z
G28XY
M02
%
```

Reference-pt. return
P-No. initial.
Absolute data input
Repeat while P-No.<49
Wpc. coordn. sys. Setting
Subprogram call
P-No. count-up

Reset to reference-pt.

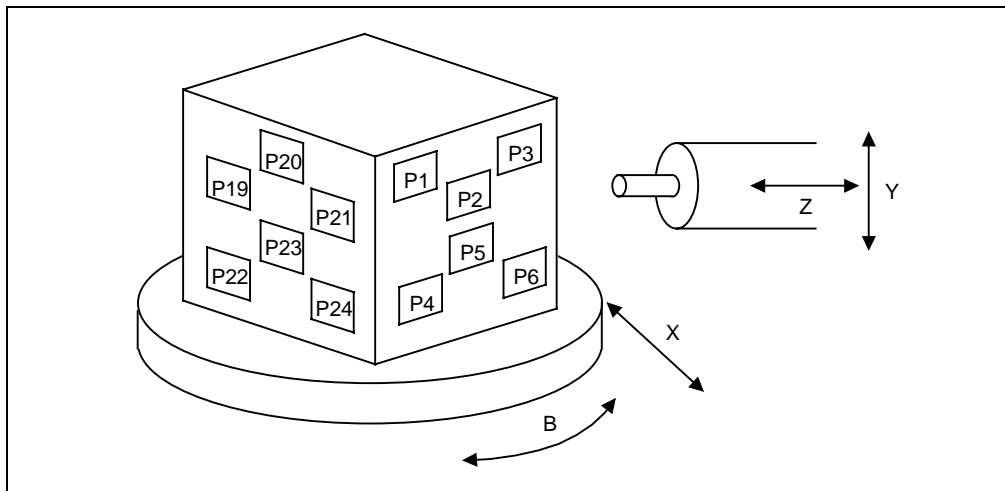
O1001 (Subprg.)

```
G43X-10.Y-0.Z-100.H10
G01X-0.
Y-30.
X-10.
Y-10.
G00G40Z10.
G98G81X-20.Y-15.Z-150.R5.F40
X-25.Y-20.
X-20.Y-25.
X-15.Y-20.
G80
M99
%
```

Profile
Drilling

3. Application of additional systems via transmission into G54 to G59

Provided that preparatory setting of origin data in P1 to P24 is completed in accordance with the 24 sections of a workpiece fixed on the rotary table as shown in the figure below:



O2000 (Main prg.)

```
G28 XYZB      Reference-pt. return
G90           Absolute data input
G00 B0        Table indexed f. 1. surf.
G65 P2001A1   Origin-data loading
M98 P2002     Drilling-subprg. call
G00 B90.      Table indexed f. 2. surf.
G65 P2001A7   Table indexed f. 3. surf.
M98 P2002     Table indexed f. 3. surf.
G00 B180.     Table indexed f. 4. surf.
G65 P2001A13  Table indexed f. 4. surf.
M98 P2002     Table indexed f. 4. surf.
G00 B270.     Table indexed f. 4. surf.
G65 P2001A19  Reset to reference-pt.
M98 P2002
G28 XYB
M02
%
```

O2001 (Origin-data transference)

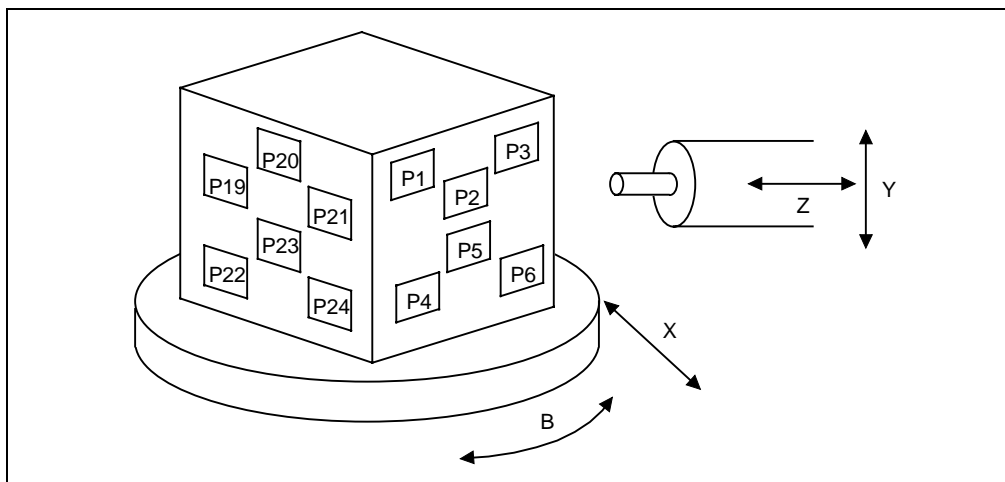
```
#2=5221      Init.-var.-No. of 1. G-sys.
#3=[#1-1].20+7001 Init.-var.-No. of 1. P-sys.
#5=0         Sys.-qty-counter cleared
WHILE[#5LT6]DO1 Check f. sys.-qty.
#6=#2       1.-ax.-var.-No. of receiver
#7=#3       1.-ax.-var.-No. of transm.
#4=0        Axis-qty-counter cleared
WHILE[#4LT6]DO2 Check f. axis-qty.
#[#6]=#[#7] Transmission of var.-data
#6=#6+1     Axis-qty count-up f. recvr.
#7=#7+1     Axis-qty count-up f. transm.
#4=#4+1     Axis-qty counter up
END2
#2=#2+20    Init.-var.-No. of next G-sys.
#3=#3+20    Init.-var.-No. of next P-sys.
#5=#5+1     Sys.-qty-counter up
END1
M99
%
```

O2002 (Drilling subprg.)

```
G54 M98 H100      Drilling in G54-system
G55 M98 H100      Drilling in G55-system
G56 M98 H100      Drilling in G56-system
G57 M98 H100      Drilling in G57-system
G58 M98 H100      Drilling in G58-system
G59 M98 H100      Drilling in G59-system
G28 Z0
M99
N100G98G81X-0.Y-15.Z-50.R5.F40 Fixed cycle for drilling
X-25. Y-20.
X-20. Y-25.
X-15. Y-20.
G80
G28Z
M99
%
```


4. Simplified version of **Example 3** program in application of "G54.1 Pp"

Provided that preparatory setting of origin data in P1 to P24 is completed in accordance with the 24 sections of a workpiece fixed on the rotary table as shown in the figure below:



O3000	
G28 XYZB	Reference-point return
G90	Selection of absolut data input
G00 B0	Indexing of table for 1st surface
G65 P3001A1	
G00 B90.	Indexing of table for 2nd surface
G65 P3001A7	
G00 B180.	Indexing of table for 3rd surface
G65 P3001A13	
G00 B270.	Indexing of table for 4th surface
G65 P3001A19	
G28 XYB	Reset to reference-point
M30	
%	
O3001	
#100=#1	Initialization of P-No.
#101=0	Initialization of counter
WHILE[#101LT6]DO1	
G54.1P#100	Setting of additional workpiece coordinate system
M98H100	Call for drilling subroutine
#100=#100+1	P-No. count-up
#101=#101+1	Checking counter count-up
END1	
G28Z0	
M99	
N100G98G81X-20.Y-15.Z-150.R5.F40	Fixed cycle for drilling
X-25. Y-20.	
X-20. Y-25.	
X-15. Y-20.	
G80	
G28Z	
M99	
%	

14-11 Local Coordinate System Setting : G52

1. Function and purpose

A local coordinate system in which the designated position becomes the program origin can be set on the current workpiece coordinate system by setting command code G52.

Command code G52 can also be used instead of G92 to update any positional shift between the machining program origin and the workpiece origin.

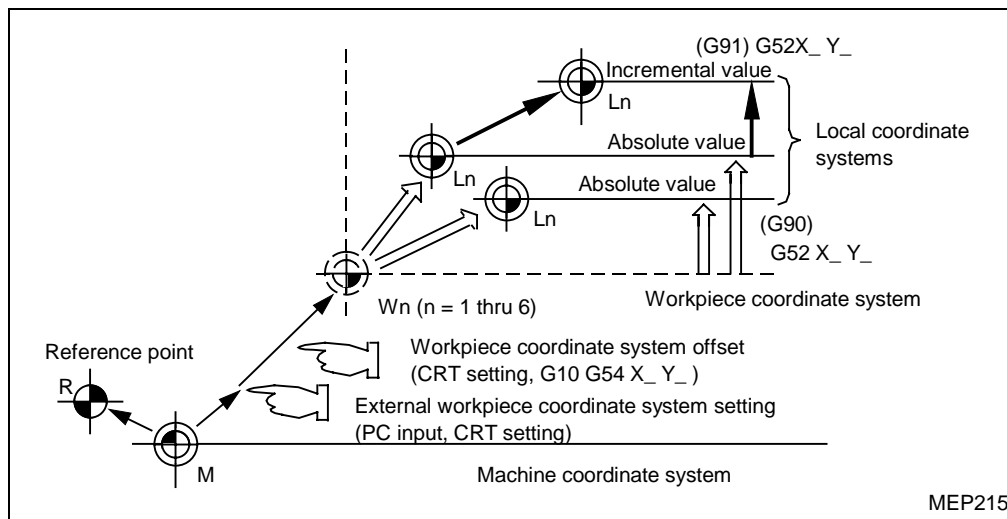
2. Programming format

G52 X_{α_1} Y_{α_1} Z_{α_1} α_{α_1} (α : Additional axis)

3. Detailed description

Command G52, which causes no machine movement, is valid until a new G52 is issued. Command G52 is therefore convenient for using a new coordinate system without changing the origin of the workpiece coordinate system.

The local coordinate system offset data is cleared by execution of either an automatic reference-point return operation, or a manual reference-point (zero point) return operation using the watchdog method following power-on.



Coordinates given in the absolute data input mode (G90) refer to the local coordinate system.

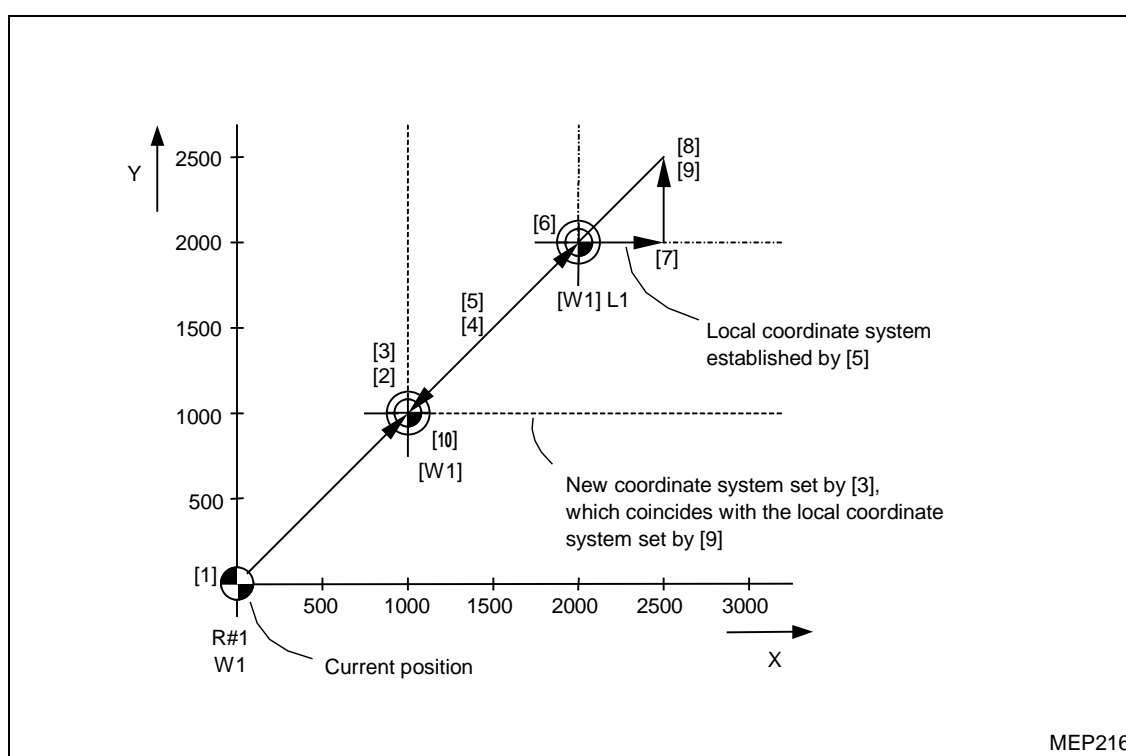
4. Sample programs

- Setting local coordinates in the absolute data input mode (local coordinate system offset data is not integrated)

```

[1] G28X0Y0
[2] G00G90X1.Y1.
[3] G92X0Y0
[4] G00X500.Y500
[5] G52X1.Y1.
[6] G00X0Y0
[7] G01X500.F100
[8] Y500.
[9] G52X0Y0
[10] G00X0Y0

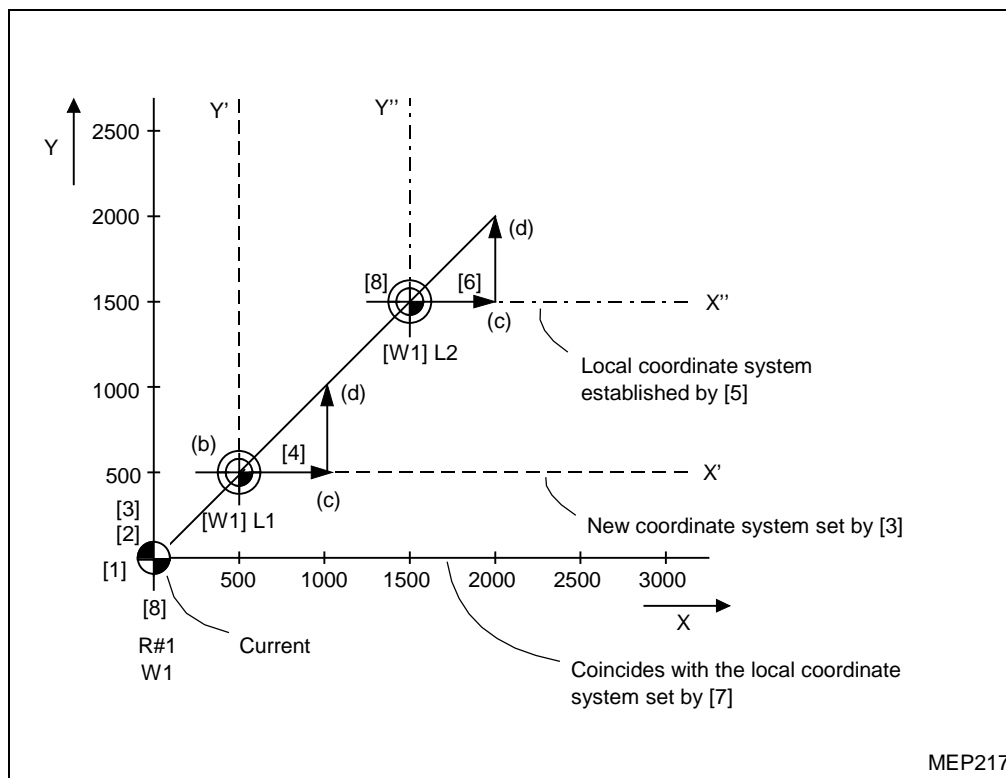
```



The local coordinate system is established by [5] and cancelled by [9], and coincides with the coordinate system of [3].

2. Setting local coordinates in the incremental data input mode (local coordinate system offset data is integrated)

[1] G28X0Y0	(a) O100
[2] G92X0Y0	(b) G90G00X0Y0
[3] G91G52X500.Y500.	(c) G01X500.
[4] M98P100	(d) Y500.
[5] G52X1.Y1.	(e) G91
[6] M98P100	(f) M99
[7] G52X-1.5Y-1.5	
[8] G00G90X0Y0	
⋮	
M02	



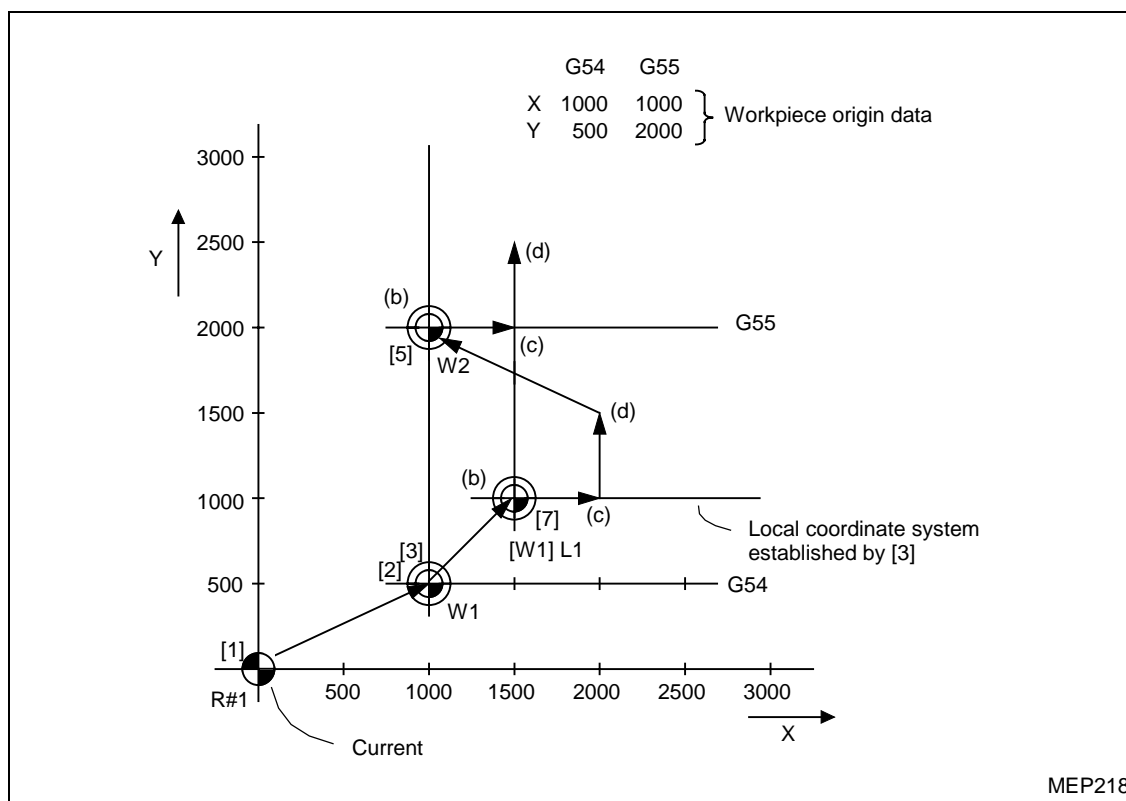
Block [3] generates local coordinate system $X'-Y'$ at the position of (500, 500) in the X-Y coordinate system.

Block [5] generates local coordinate system $X''-Y''$ at the position of (1000, 1000) in the $X'-Y'$ coordinate system.

Block [7] generates a local coordinate system at the position (-1500, -1500) in the $X''-Y''$ coordinate system. That is, the local coordinate system and the X-Y coordinate system coincide, which means that the former has been cancelled.

3. Combined use of local coordinate system and workpiece coordinate system

[1] G28X0Y0	(a) O200
[2] G00G90G54X0Y0	(b) G00X0Y0
[3] G52X500.Y500.	(c) G01X500.F100
[4] M98P200	(d) Y500.
[5] G00G90G55X0Y0	(e) M99
[6] M98P200	%
[7] G00G90G54X0Y0	
⋮	
M02	



Block [3] generates a local coordinate system at the position of (500, 500) in the G54 coordinate system no local coordinate system is generated in the G55 coordinate system.

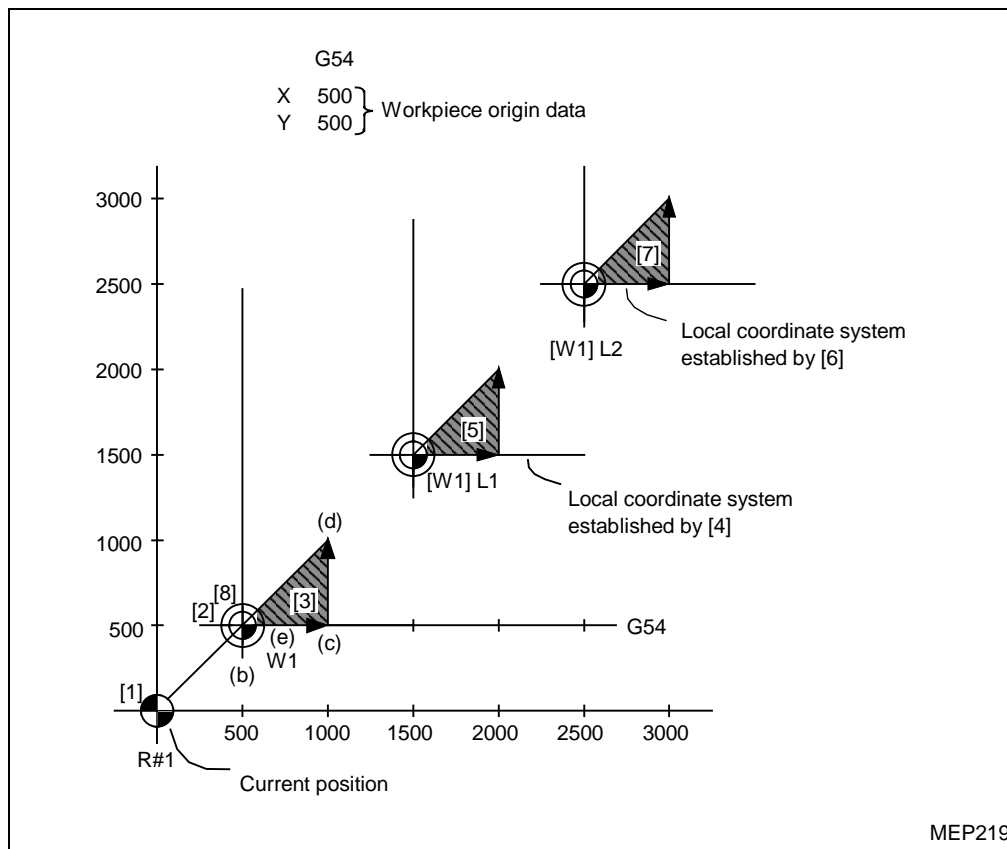
Block [7] performs a movement to the local coordinate system reference point (zero point) of G54.

The local coordinate system can be cancelled using this format:

G90 G54 G52 X0 Y0

4. Combined use of G54 workpiece coordinate system and multiple local coordinate systems

[1] G28X0Y0	(a) O300
[2] G00G90G54X0Y0	(b) G00X0Y0
[3] M98P300	(c) G01X500.F100
[4] G52X1.Y1.	(d) Y500.
[5] M98P300	(e) X0Y0
[6] G52X2.Y2.	(f) M99
[7] M98P300	%
[8] G52X0Y0	
⋮	
M02	



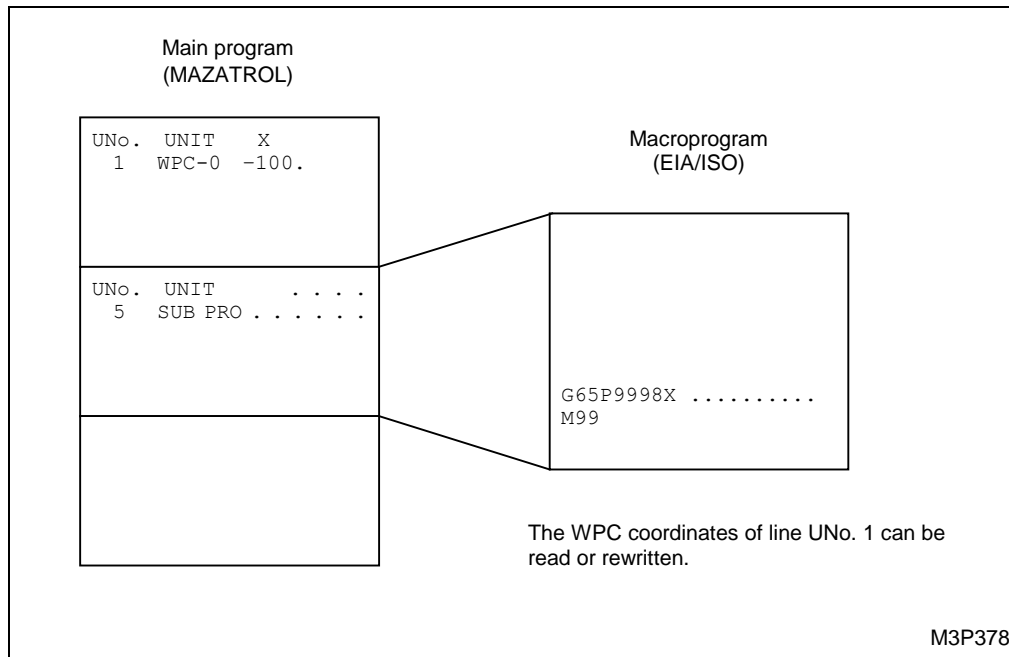
Block [4] generates a local coordinate system at the position of (1000, 1000) in the G54 coordinate system.

Block [6] generates a local coordinate system at the position of (2000, 2000) in the G54 coordinate system.

Block [8] makes the local coordinate system coincide with the G54 coordinate system.

14-12 Reading/Writing of MAZATROL Program Basic Coordinates

The basic coordinates of a MAZATROL program can be read or rewritten by calling a user macroprogram in the subprogram unit of the MAZATROL program.



Note 1: For updating the basic coordinates, do not forget to select the **[MEASURE MACRO]** menu function in entering a subprogram unit for the macroprogram concerned (WNo. 9998); otherwise the new coordinates may not always be used in time for the execution of the machining unit immediately following the subprogram unit.

Note 2: On the other hand, do not select the **[MEASURE MACRO]** menu function unless the above macroprogram is to be used.

14-12-1 Calling a macroprogram (for data writing)

To rewrite the basic coordinates data, call the specific user macroprogram from a subprogram unit of the MAZATROL program (macroprogram call is not required for data reading).

Refer to the section of subprogram unit in the Programming Manual (MAZATROL Programming) for details on data setting for subprogram call.

14-12-2 Data reading

System variables can be used to read the MAZATROL basic coordinates that are effective during macroprogram execution.

System variables for MAZATROL basic coordinates (WPC)

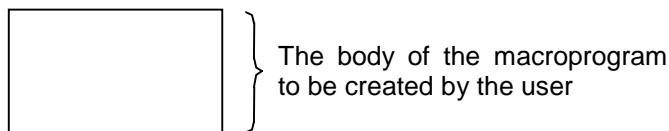
Variables number	Function	Variables number	Function
#5341	WPC-X	#5344	WPC-4
#5342	WPC-Y	#5345	WPC-5
#5343	WPC-Z	#5347	WPC-th

14-12-3 Rewriting

Same as reading, you are to use system variables when rewriting the basic coordinates.

The basic coordinates, however, cannot be rewritten just by entering the data into #5341 through #5347. It is therefore necessary to create a macroprogram in the following format:

1. Macroprogramming format



G65 P9998 X_Y_Z_D_B_C_

M99

- At the end of macroprogramming, call the rewriting macroprogram (WNo. 9998). At this time, assign new coordinates as arguments. The relationship between each argument and axis is as follows:

X: WPC-X Y: WPC-Y Z: WPC-Z

D: WPC-th B: WPC-4 C: WPC-5

- Only the coordinates assigned with the respective argument are rewritten. The argument data is handled as data having the decimal point.

2. Rewriting macroprogram

The rewriting macroprogram (WNo. 9998) is shown below.

O9998	N30	#50467=#50467OR-65536
IF[#50600EQ#0]GOTO60	IF[#7EQ#0]GOTO40	#50499=#50499OR1
IF[#24EQ#0]GOTO10	#5347=#7	N60
#5341=#24	#50441=#7	M99
#50449=#24	#50467=#50467OR512	%
#50467=#50467OR32	N40	
N10	IF[#2EQ#0]GOTO45	
IF[#25EQ#0]GOTO20	#5344=#2	
#5342=#25	#50443=#2	
#50447=#25	#50467=#50467OR256	
#50467=#50467OR64	N45	
N20	IF[#3EQ#0]GOTO50	
IF[#26EQ#0]GOTO30	#5345=#3	
#5343=#26	#50453=#3	
#50445=#26	#50467=#50467OR1024	
#50467=#50467OR128	N50	

Note: An alarm will occur when executing the macroprogram if no basic coordinates of MAZATROL program are currently validated.

14-13 Workpiece Coordinate System Rotation

1. Function and purpose

The function refers to rotating the workpiece coordinate system around the position of the specified machine coordinates. The machining program can be rotated in its entirety as required for the actual inclination of the workpiece.

2. Programming format

(G17) G92.5 Xx Yy Rr X-Y plane

(G18) G92.5 Zz Xx Rr Z-X plane

(G19) G92.5 Yy Zz Rr Y-Z plane

or

(G17) G92.5 Xx Yy li Jj X-Y plane

(G18) G92.5 Zz Xx Kk li Z-X plane

(G19) G92.5 Yy Zz Jj Kk Y-Z plane

x, y, z : Coordinates of the rotational center.

The position along the two axes of the previously selected X-Y, Z-X, or Y-Z plane must be designated in machine coordinates.

The designation for an axis not corresponding to the plane will be ignored.

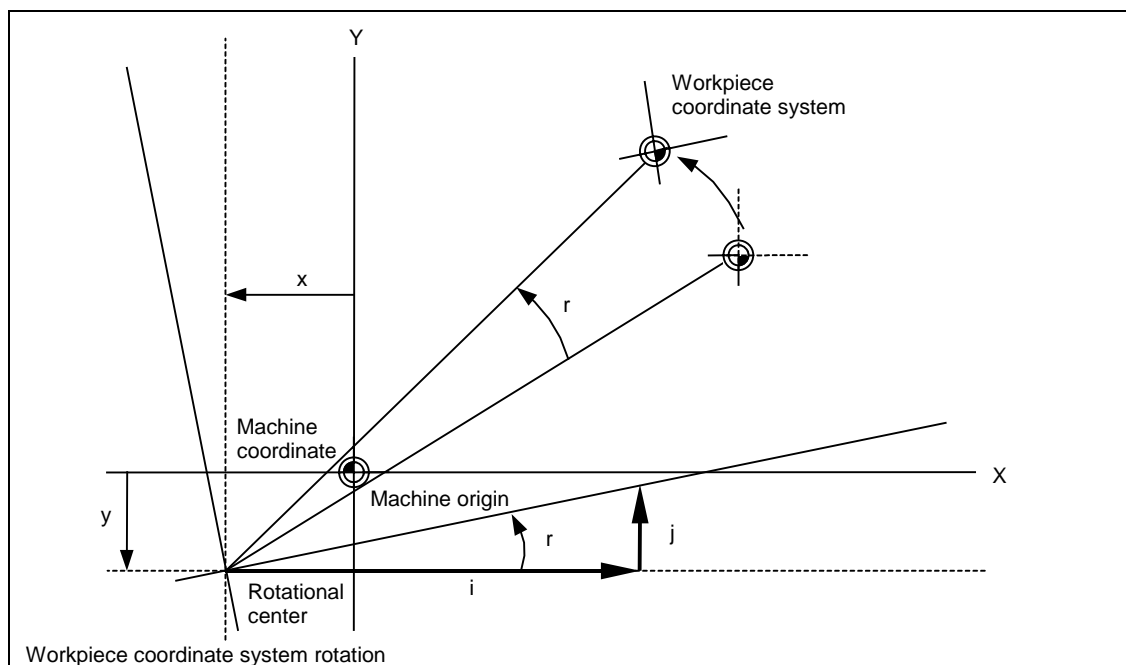
R : Angle of rotation.

Designate the rotational angle for the coordinate system. A positive value refers to a counterclockwise rotation.

i, j, k : Axial component vectors.

The angle for coordinate system rotation can also be designated in axial component vectors corresponding to the previously selected plane.

The designation for an axis not corresponding to the plane will be ignored.



Range and unit for angle data setting

Setting method		Setting range	Setting unit
Axial component vectors (i, j, k)	Metric system	0 to ± 99999.999	0.001 mm
	Inch system	0 to ± 9999.9999	0.0001 in.
Angle of rotation (r)	Metric system	0 to ± 99999.999°	0.001°
	Inch system		

3. Detailed description

1. Irrespective of the actual mode for incremental or absolute data input, the values at addresses X, Y, Z, or I, J, K as well as R are always referred to the machine coordinate system.
2. Two methods are available to designate a rotational angle:
 - (a) Designation in rotational angle (r), or
 - (b) Designation in axial component vectors (i, j, k).
3. If angle data are entered using both methods (a) and (b) above, the rotational angle (at address R) will govern.
4. If, during rotation of the workpiece coordinate system, a rotational angle of zero degrees is designated (by setting G92.5 R0, for example), the coordinate system rotation will be cancelled, irrespective of the data input mode of G90 (absolute) or G91 (incremental). The next move command will then be executed for the ending point in the original (not rotated) workpiece coordinate system (refer to Article 1 in Item 5. Precautions).
5. The rotational center coordinates will be retained and automatically applied for a succeeding rotation command without data designation at addresses X, Y, and/or Z.

Example:

N1 G17	Selection of the X-Y plane
N2 G92.5X100.Y100.R45.	Rotation of the workpiece coordinate system through 45 deg around the point of (X, Y) = (100, 100)
:	
N3 G92.5R0	Cancellation of the workpiece coordinate system rotation
:	
N4 G17G92.5R90.	Rotation of the workpiece coordinate system through 90 deg around the center last programmed (X100, Y100)
:	
%	

6. Omission of addresses R and I, J, K is regarded as a rotational angle designation of zero degrees.

Example: "G92.5 X0. Y0." is equivalent to "G92.5 X0. Y0. R0".

7. Alarm No. **809 ILLEGAL NUMBER INPUT** will be displayed if the specified axial component vectors (i, j, k) or rotational angle (r) oversteps the effective setting range.
8. Plane selection (by codes G17, G18, and G19) need not be included in the block of G92.5, if the rotation shall be performed on the currently active plane.
9. The designation for an axis not corresponding to the selected plane will be ignored. The designations at addresses Z and K in a block of G92.5, for example, will be ignored in the mode of G17 (X-Y plane).

Example: The second block shown below rotates the workpiece coordinate system through 63.435°, calculation from $\tan^{-1}(2/1)$, around the point of (X, Y) = (10, 20) on the X-Y plane, and the values at Z and K are ignored for the rotation.

```
G17
G92.5X10.Y20.Z30.I1.J2.K3.
```

Even the ignored axial values at X, Y, and Z in a G92.5 block are retained as well as the values actually used (see Article 5 above) and, for example, if the G92.5 block shown above is followed by

```
G19
G92.5J2.K3.
```

then the workpiece coordinate system will be rotated around the point of (Y, Z) = (20, 30) through 56.301°, calculation from $\tan^{-1}(3/2)$, on the Y-Z plane (G19).

4. Examples of operation

1. Rotation around the machine origin.

N1 G28X0Y0

N2 G17

N3 G90

N4 G55

N5 G92.5X0Y0R90. (or G92.5X0Y0I0J1.)

N6 G0X0Y0

N7 G1X100.F1000.

N8 Y200.

N9 X0.

N10 Y0.

N11 M30

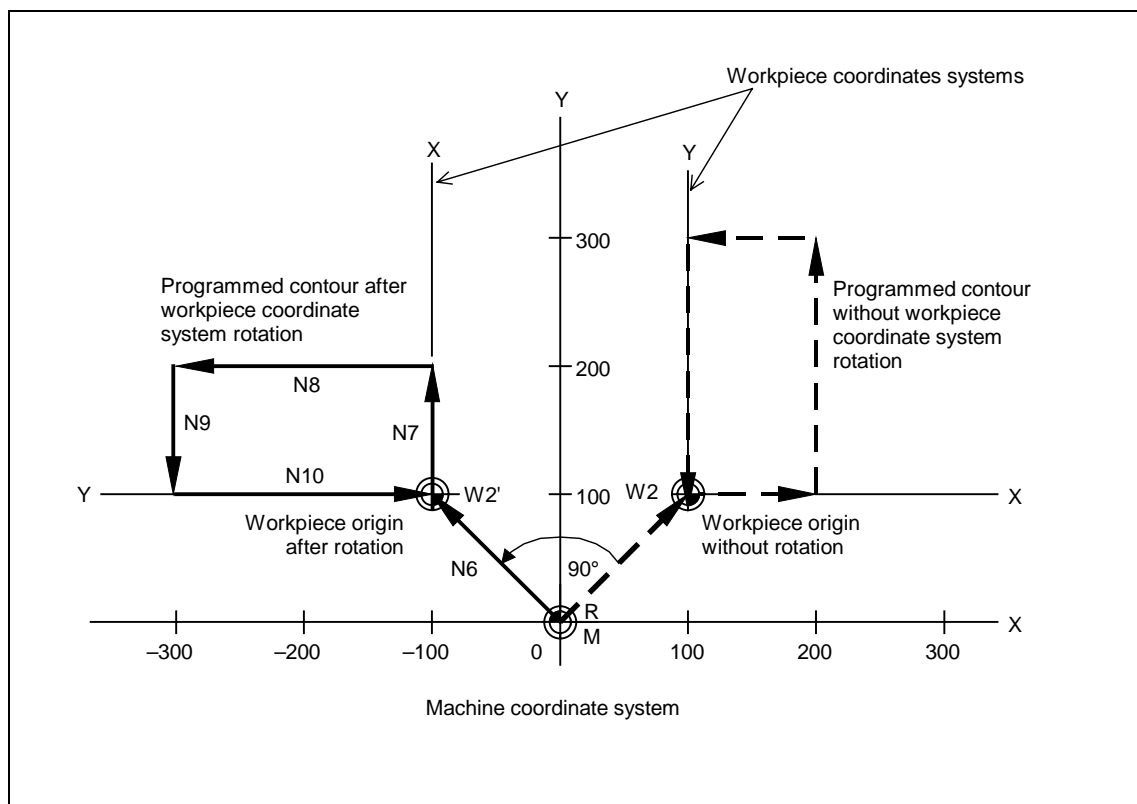
%

G55 (Work Offset)

X100.

Y100.

Rotation around the machine origin
through 90 deg



- The block of G92.5 under N5 rotates the workpiece coordinate system through 90 degrees around the origin of the machine coordinate system. For N6 onward, the machine operates according to the rotated workpiece coordinate system.

- The above example of the vector setting method for the same 90-deg rotation is based on the following calculation:

$$= \tan^{-1} (J/I) = \tan^{-1} (1/0) = 90^\circ.$$

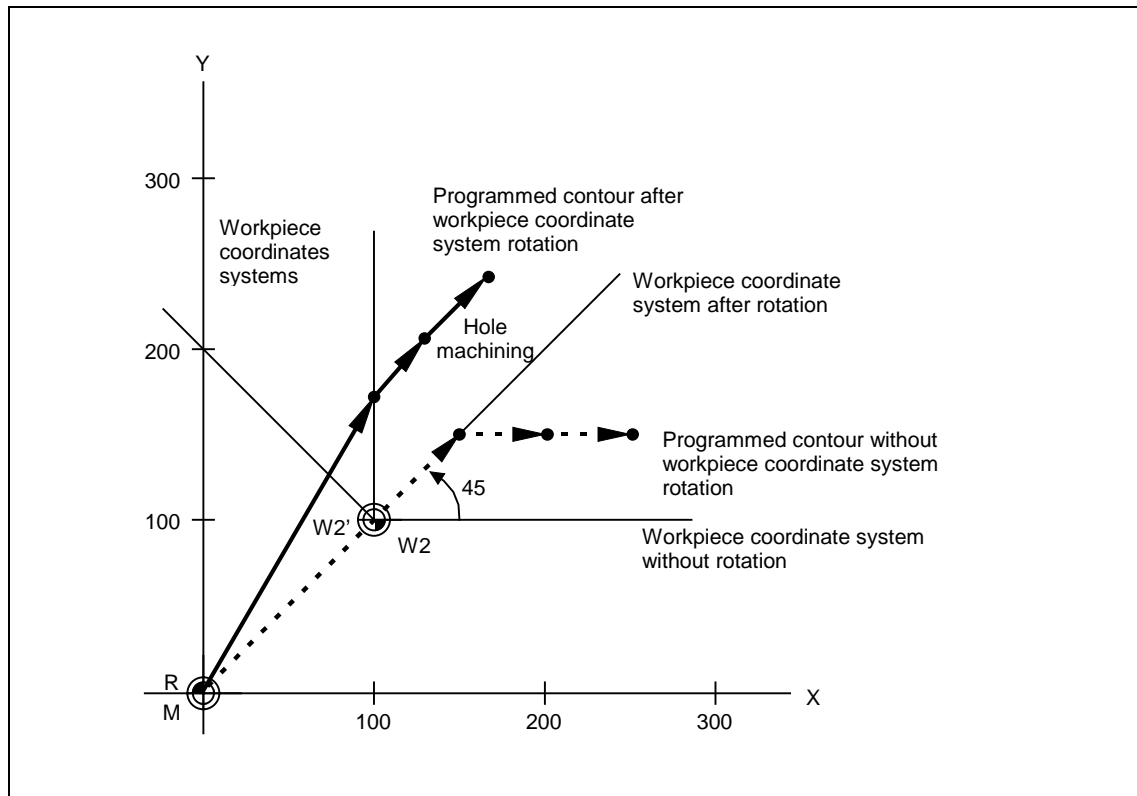
2. Rotation around the workpiece origin.

```

N1 G28X0Y0Z0
N2 G17
N3 G55
N4 G90
N5 G92.5X100.Y100.R45.....
N6 G81X50.Y50.Z-25.R-5.F500
N7 X100.
N8 X150.
N9 M30
%
```

G55 (Work Offset)
X100.
Y100.

Rotation through 45 deg around the point of machine coordinates X=100 and Y=100 (that is, the origin of the G55 workpiece coordinate system).



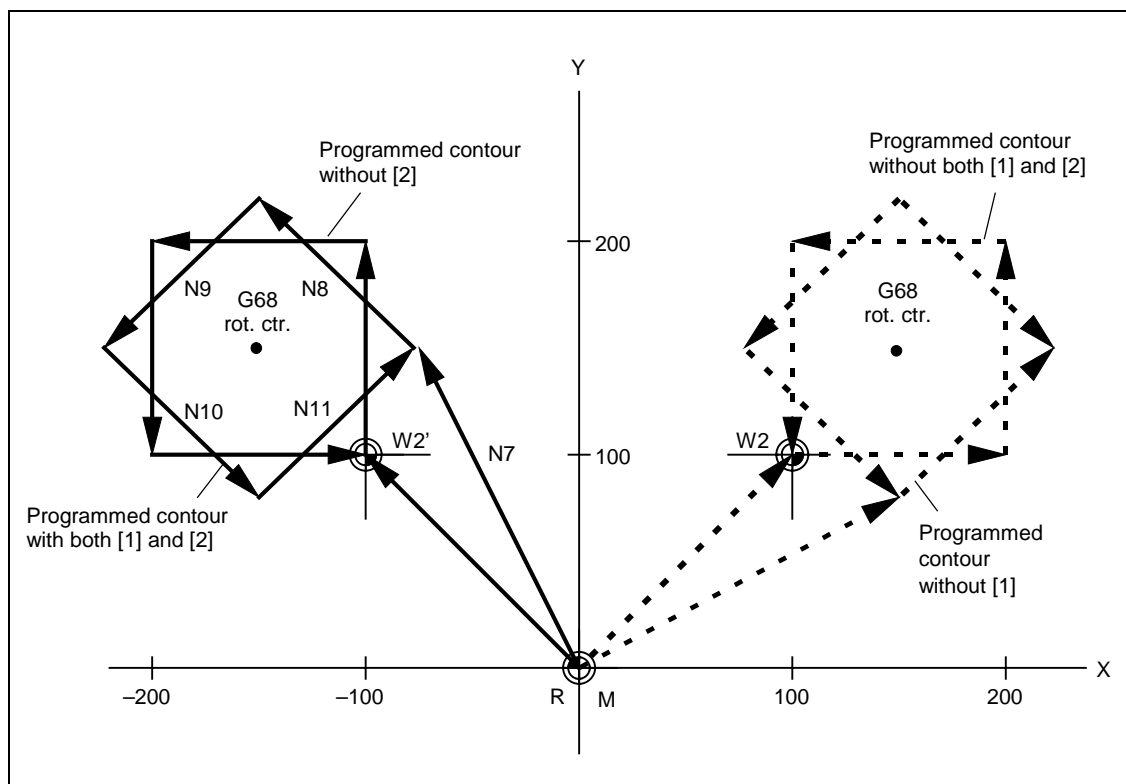
- The block of G92.5 under N5 rotates the workpiece coordinate system around its own origin through 45 degrees. For N6 onward, the machine operates according to the rotated workpiece coordinate system.
- Set the rotational center on the workpiece origin, as shown in this example, to rotate the current workpiece coordinate system around its own origin.

3. Programmed coordinate rotation (G68) in the mode of G92.5

```

N1 G28X0Y0
N2 G17
N3 G55
N4 G90
N5 G92.5X0Y0R90. .... [1]
N6 G68X50.Y50.R45. .... [2]
N7 G0X0Y0
N8 G1X100.F500
N9 Y100.
N10 X0
N11 Y0
N12 M30
%
```

G55 (Work Offset)
X100.
Y100.



In a combined use with G92.5, the center of programmed coordinate rotation by G68 will be a position which corresponds with the workpiece coordinate system rotation designated by the G92.5 command.

It will not affect operation even if the order of the program blocks marked [1] and [2] above is reversed.

4. Figure rotation (M98) in the mode of G92.5

N1 G28X0Y0

N2 G17

N3 G55

N4 G90

N5 G92.5X0Y0R90.

N6 G0X0Y0

N7 M98H10I-50.J50.L4

N8 M30

N9

N10 G1X100.Y50.F500

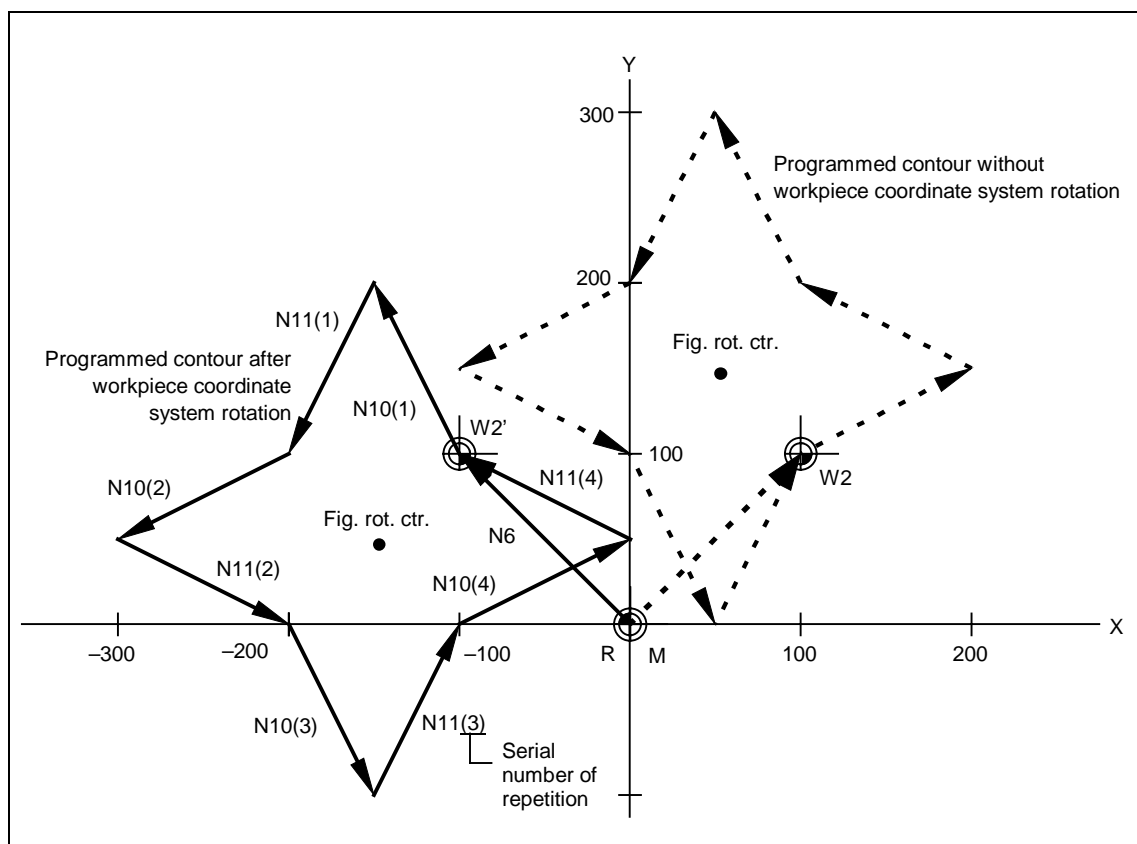
N11 X0Y100.

N12 M99

%

G55 (Work Offset)
X100.
Y100.

Rotation through 90 deg around the origin of the machine coordinate system



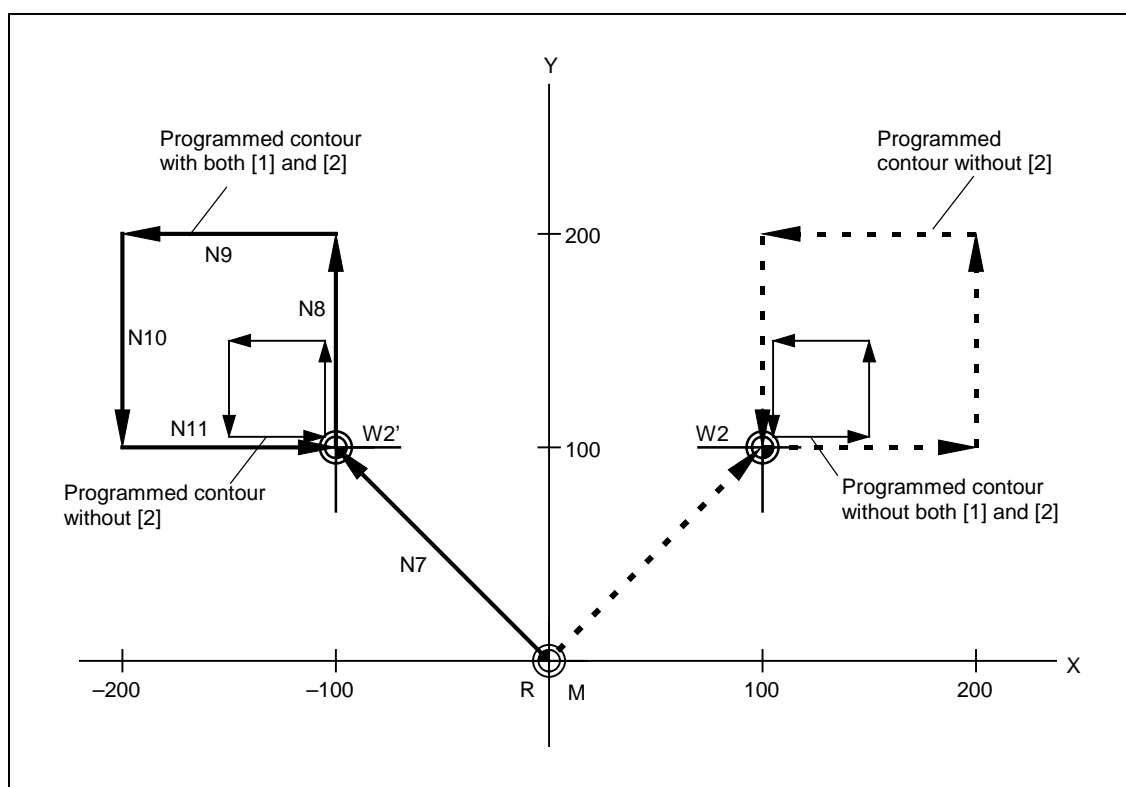
In a combined use with G92.5, the center of figure rotation by M98 will be a position which corresponds with the workpiece coordinate system rotation designated by the G92.5 command.

5. Scaling (G51) in the mode of G92.5

```

N1 G28X0Y0
N2 G17
N3 G55
N4 G90
N5 G92.5X0Y0R90. .... [1]
N6 G51X0Y0P2. .... [2]
N7 G0X0Y0
N8 G1X50.F500
N9 Y50.
N10 X0
N11 Y0
N12 M30
%
```

G55 (Work Offset)
X100.
Y100.



In a combined use with G92.5, the scaling center will be a position which corresponds with the workpiece coordinate system rotation designated by the G92.5 command.

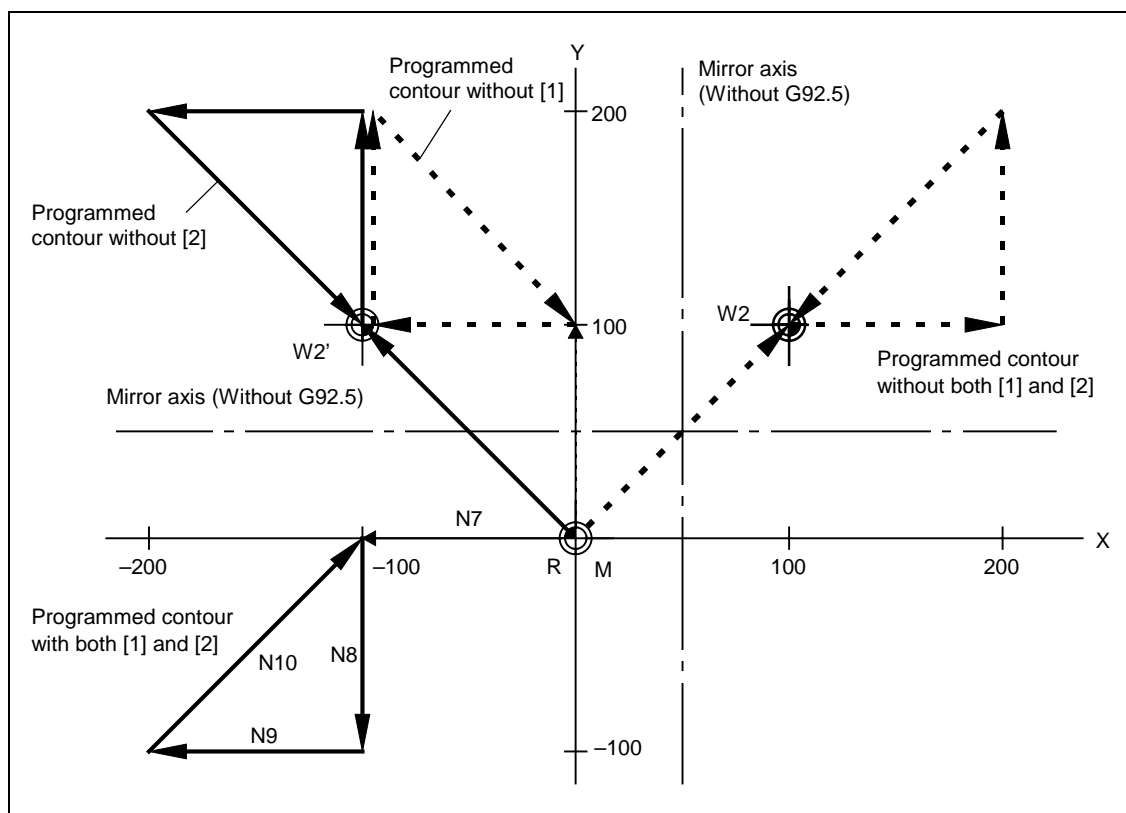
6. Mirror image in the mode of G92.5

a) G-code mirror image

```

N1  G28X0Y0
N2  G17
N3  G55
N4  G90
N5  G92.5X0Y0R90. .... [1]
N6  G51.1X-50. .... [2]
N7  G0X0Y0
N8  G1X100.F500
N9  Y100.
N10 X0Y0
N11 M30
%
```

G55 (Work Offset)
X100.
Y100.

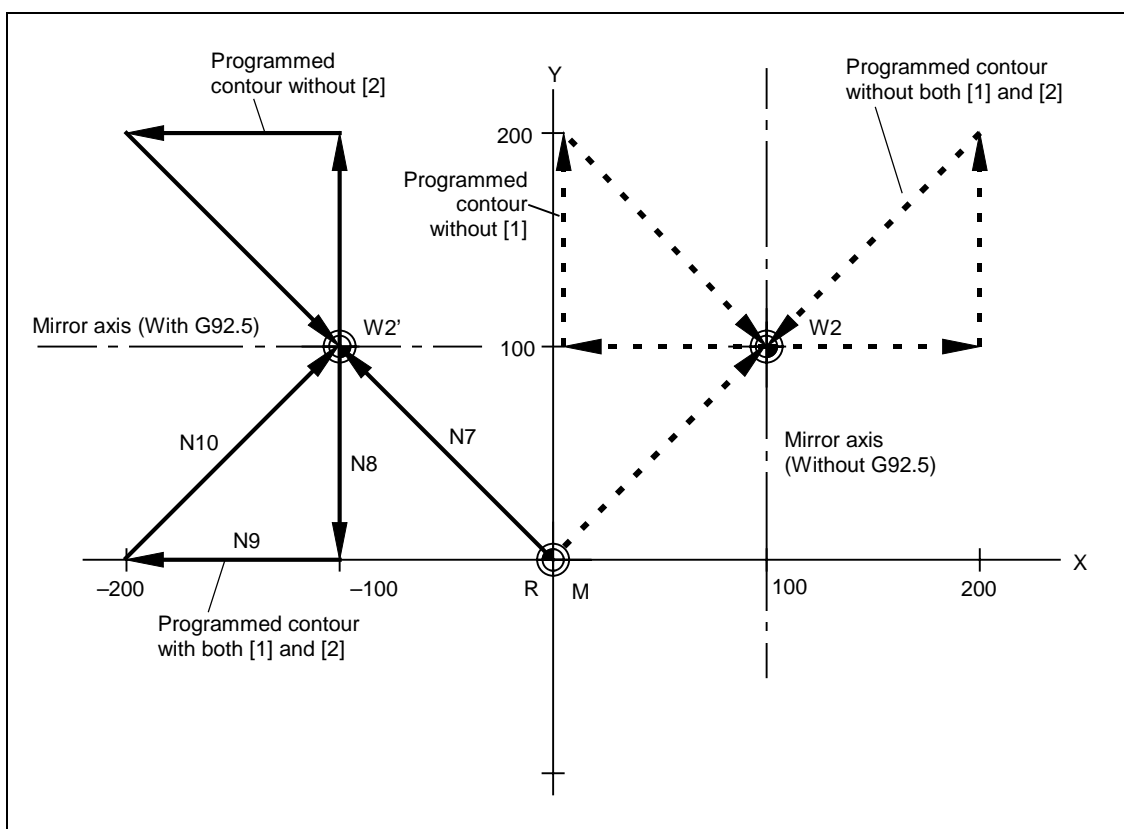


b) M-code mirror image

```

N1  G28X0Y0
N2  G17
N3  G55
N4  G90
N5  G92.5X0Y0R90. .... [1]
N6  M91. .... [2]
N7  G0X0Y0
N8  G1X100.F500
N9  Y100.
N10 X0Y0
N11 M30
%
```

G55 (Work Offset)
X100.
Y100.

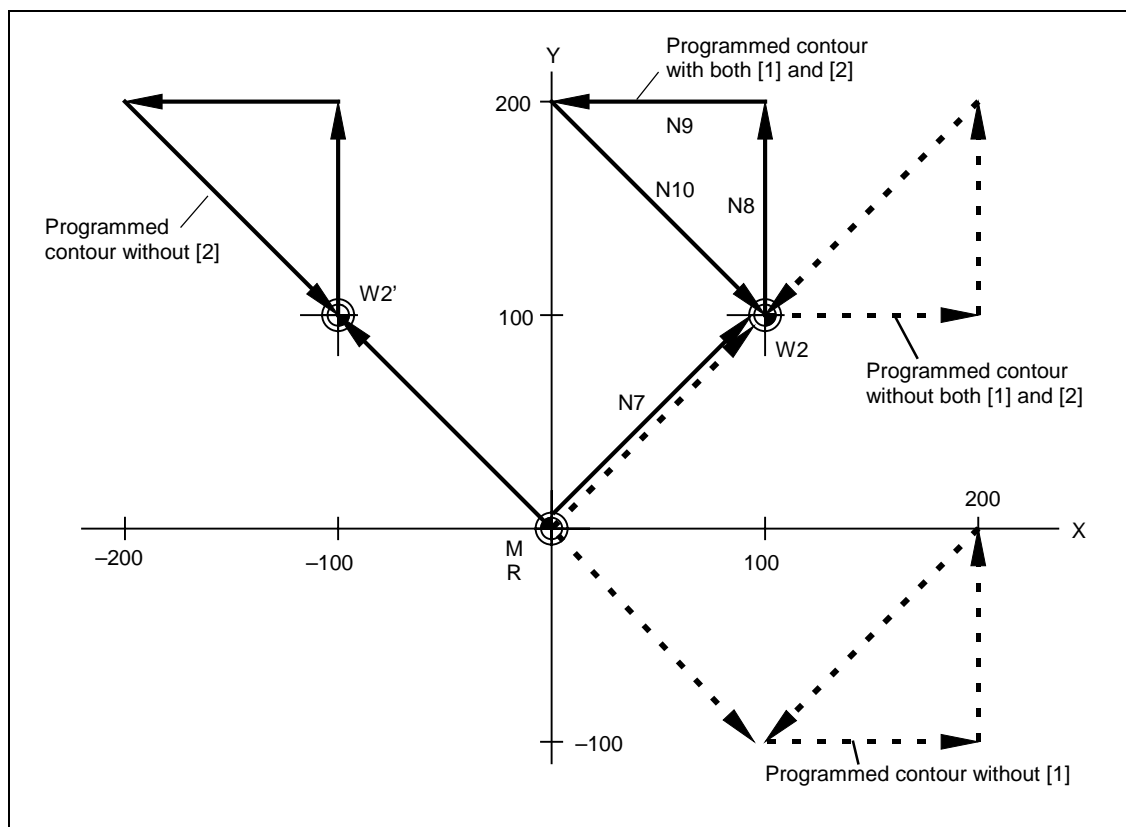


In a combined use with G92.5, the axis of symmetry for G-code or M-code mirror image will be set in accordance with the workpiece coordinate system rotation designated by the G92.5 command.

7. Coordinate system setting (G92) in the mode of G92.5

N1	G28X0Y0	
N2	G17	
N3	G55	
N4	G90	
N5	G92.5X0Y0R90.	[1]
N6	G92X-100.Y100.	[2]
N7	G0X0Y0	
N8	G1X100.F500	
N9	Y100.	
N10	X0Y0	
N11	M30	
%		

G55 (Work Offset)
X100.
Y100.



Coordinate system setting by a G92 block after G92.5 will be performed in reference to the coordinate system rotation designated by the G92.5 command.

5. Precautions

1. If, during rotation of the workpiece coordinate system, a rotational angle of zero degrees is designated (by setting G92.5 R0, for example), the coordinate system rotation will be cancelled, irrespective of the data input mode of G90 (absolute) or G91 (incremental). The next move command will then be executed for the ending point in the original (not rotated) workpiece coordinate system.

Example 1: For incremental data input

```

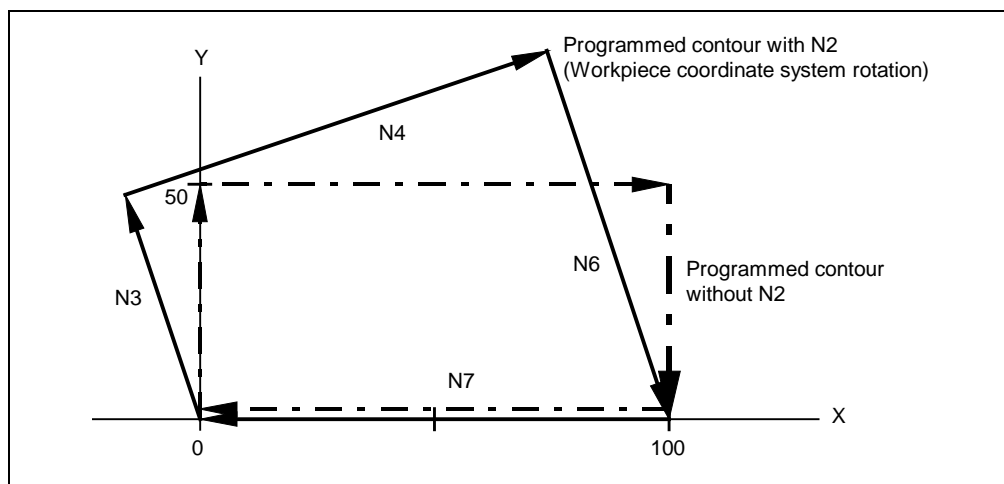
N1 G28X0Y0
N2 G17G92.5X0Y0R20.
N3 G91G01Y50.F1000.
N4 X100.
N5 G92.5R0..... Command for 0-deg rotation
N6 Y-50. .... Motion to (X100, Y0)
N7 X-100.
N8 M30
%
```

Example 2: For absolute data input

```

N1 G28X0Y0
N2 G17G92.5X0Y0R20.
N3 G90G01Y50.F1000.
N4 X100.
N5 G92.5R0..... Command for 0-deg rotation
N6 Y0..... Motion to (X100, Y0)
N7 X0
N8 M30
%
```

Programmed contour for **Examples 1 and 2** above



2. Use a linear motion command (with G00 or G01) for the first movement to be executed after G92.5 command.

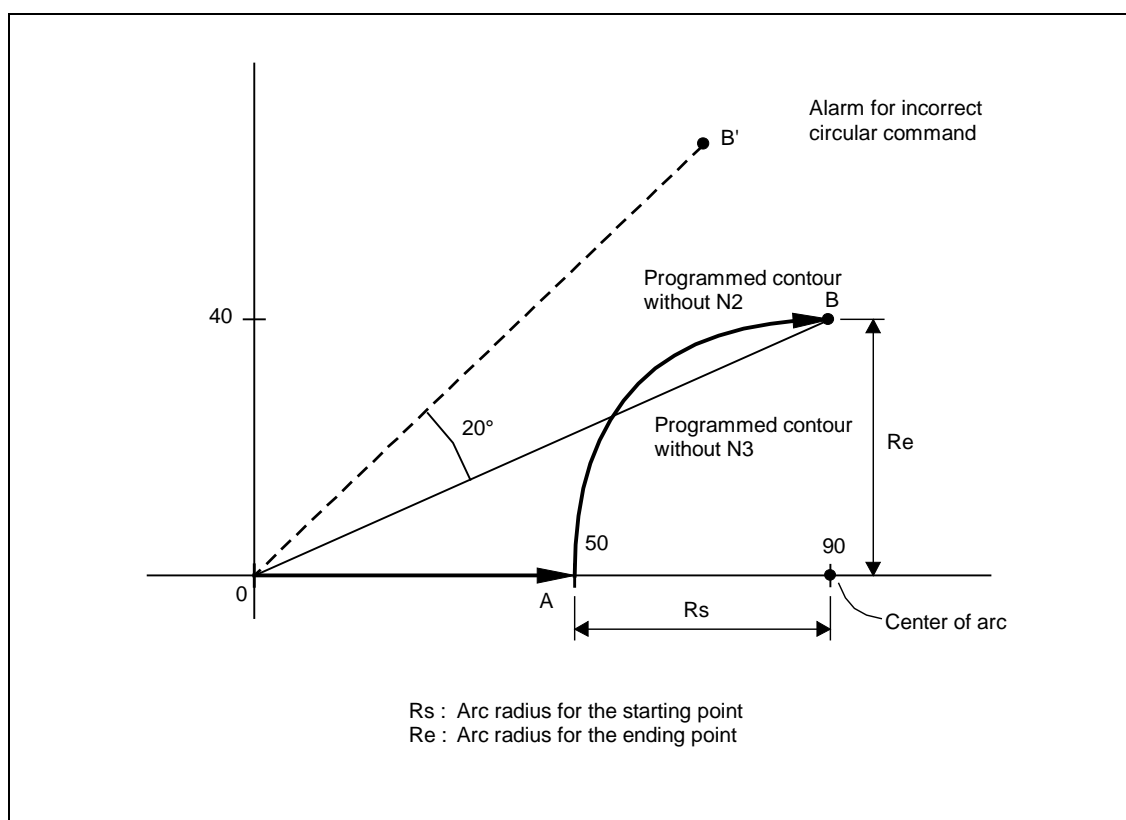
Circular interpolation in such a case, as shown below, would have to take place from the current position A, which refers to the original workpiece coordinate system, to the ending point B' to which the point B should be shifted in accordance with the rotation. As a result, the radii of the starting and ending points would differ too significantly and the alarm **No. 817 INCORRECT ARC DATA** would be caused.

Example:

```

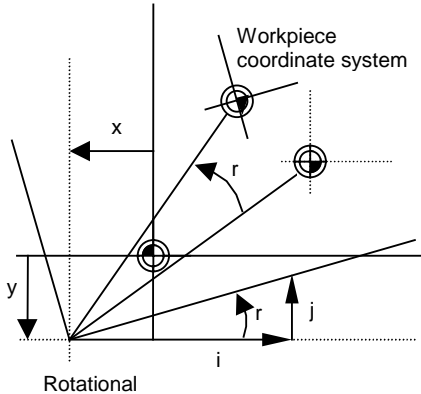
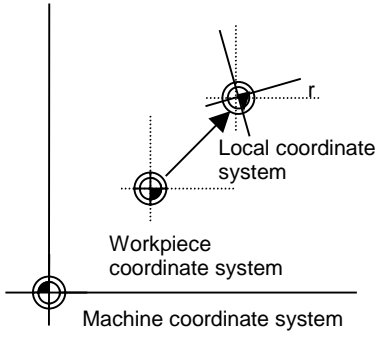
N1  G28X0Y0
N2  G91G01X50.F1000.
N3  G17G92.5X0Y0R20.
N4  G02X40.Y40.I40.
N5  M30
%
```

Circular interpolation as the first motion after G92.5



3. Set a G92.5 command in the mode of G40.
4. The machine will operate on the rotated coordinate system for an MDI interruption during the mode of G92.5.
5. For a manual interruption during the mode of G92.5 using the JOG or handle feed mode, the machine will operate independently of the coordinate system rotation.

6. Differences between workpiece coordinate system rotation and programmed coordinate rotation.

Function name		Workpiece coordinate system rotation	Programmed coordinate rotation
System to be rotated		Workpiece coordinate system	Local coordinate system
Programming format		(G17) G92.5 Xx Yy Rr (G18) G92.5 Yy Zz Rr (Angle) (G19) G92.5 Zz Xx Rr or (G17) G92.5 Xx Yy Ii Jj (G18) G92.5 Yy Zz Jj Kk (Vector comp.) (G19) G92.5 Zz Xx Kk Ii	(G17) G68 Xx Yy Rr (G18) G68 Yy Zz Rr (G19) G68 Zz Xx Rr
Operation			
Rotational center coordinates		Designation at addresses X, Y, Z	Designation at addresses X, Y, Z
Angle of rotation		Designation at R (angle) or at I, J, K (vector components)	Designation at R (angle)
Information on center and angle of rotation cleared?	Power-off → on	Retained	Cleared
	M02/M30	Retained	Cleared
	Reset key	Retained	Cleared
	Resumption of readiness after emergency stop	Retained	Cleared

Note: Resetting or M02/M30 cancels the G92.5 mode itself, while the information on the rotational center, etc., at related addresses is retained as indicated above.

- NOTE -

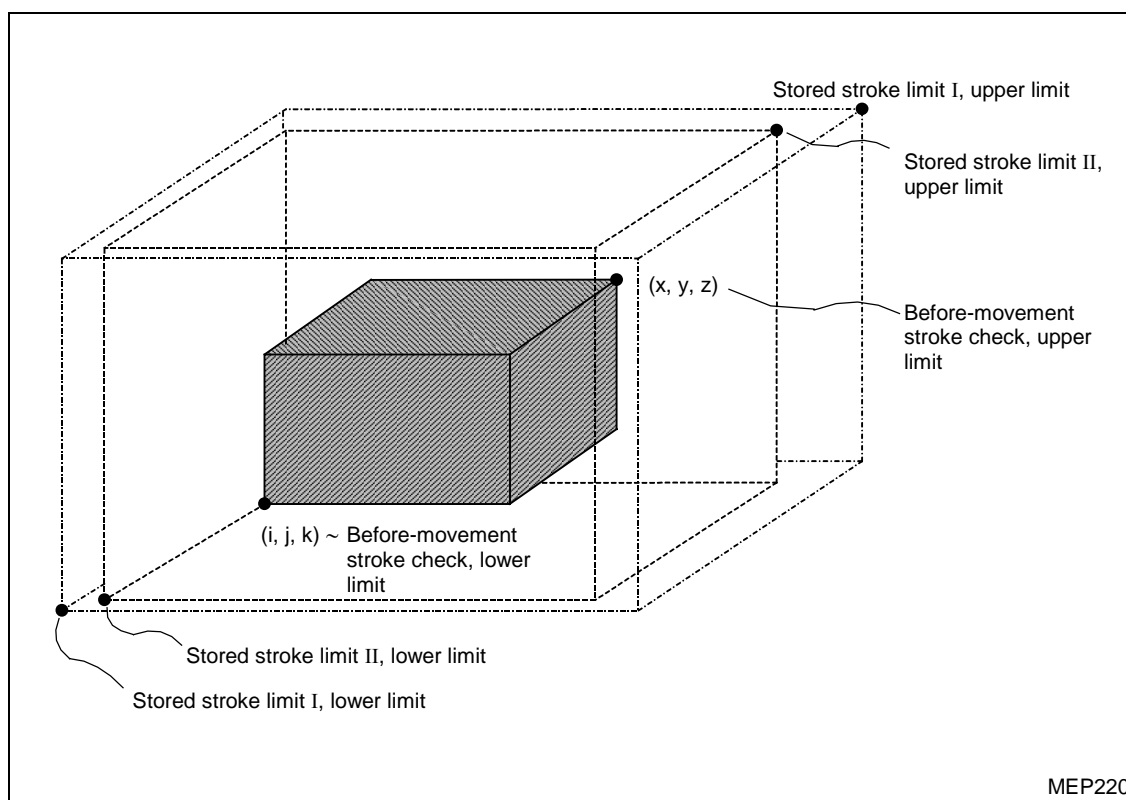
15 PROTECTION FUNCTIONS

15-1 Before-Movement Stroke Limit Check: G22, G23

1. Function and purpose

While stored stroke limit check generates an outside machining prohibit area, before-movement stroke limit check generates an inside machining prohibit area (shaded section in the diagram below).

An alarm will result if you set a move command code that brings an axis into contact with (or moves it through) the shaded section.



2. Programming format

G22 $\underbrace{X_Y_Z_}_{\text{Upper limit specification}} \underbrace{I_J_K_}_{\text{Lower limit specification}}$ (Inside machining prohibit area specification)

G23 (Cancel)

3. Detailed description

- Both upper-limit and lower-limit values must be data present on the machine coordinate system.
- Use X, Y, Z to set the upper limit of the prohibit area, and I, J, K to set the lower limit. If the value of X, Y, Z is smaller than that of I, J, K, then the former will become the lower-limit value and, the latter, the upper-limit value.

3. No stroke limit checks will be performed if the upper- and lower-limit values that have been assigned to the axis are identical.

G22X200.Y250.Z100.I200.J-200.K0



The X-axis does not undergo the stroke check.

4. The before-movement stroke limit check function will be cancelled if you set G22.
5. If, for example, G23 X_Y_Z_ is set, it will be regarded as G23 X_Y_. After cancellation of before-movement stroke limit check, therefore, X_Y_ will be executed according to the modal move command code last set.

Note: Before setting G22, move the machine to a position outside the prohibit area.

16 SKIP FUNCTION: G31

16-1 Skip Function

1. Programming format

G31 Xx Yy Zz $\alpha\alpha$ Ff (α : Additional axis)

x, y, z, α : The coordinates of respective axes. These coordinates must be designated using absolute or incremental data, depending on the type of currently valid modal command mode (G90 or G91).

f: Feed rate (mm/min)

Linear interpolation becomes possible by setting G31 as shown above. When any one of external skip signals 1 through 3 is input, the machine will stop and all remaining commands will be cancelled and then the program will skip to the next block.

2. Detailed description

1. For the feed rate, if Ff is included in the program, the value you have set as f will be used. If Ff is not included, the "G31 skip" value set by the parameter **K41** will be used as the feed rate. F-modal command data, however, will not be updated.
2. Automatic acceleration/deceleration is not applied to command block G31. The maximum feed rate that can be defined with G31 depends on the machine specifications.
3. When G31 is set, the feed rate will be fixed at 100% since the feed override function becomes invalid. Dry run becomes invalid, either. However, stop conditional functions, such as Feed Hold, Interlock, Override Zero, and Stroke End, will be valid. External deceleration is also valid.
4. Command G31 is unmodal, and thus it must be set each time.
5. The execution of command G31 will immediately terminate if a skip signal (either type 1, type 2, or type 3) is input at the beginning.
Also, if a skip signal is not input until the end of command block G31, execution of this command will terminate on completion of execution of move commands.
6. Setting this command code during tool diameter offsetting results in a program error.
7. A program error will also result if the G31 command code does not include F and the parameter **K41** is 0.
8. Under a machine locked status or if the Z axis alone is designated, any skip signals will be ignored and the program will be executed through to the end of that block.

3. Parameter setting

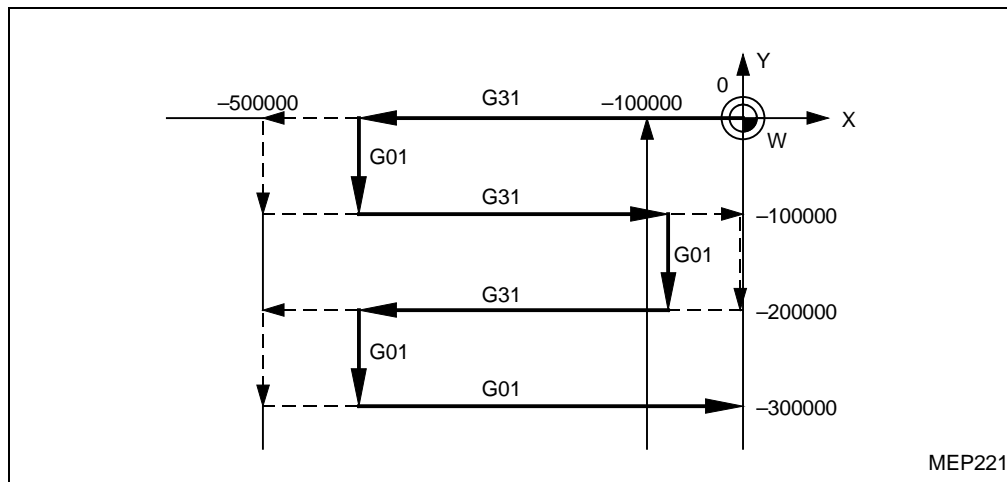
Set the "G31 skip conditions" on the **PARAMETER** display.

4. A sample program for executing G31

```

G90 G00 X-100000 Y0
      G31 X-500000 F100
      G01 Y-100000
      G31 X0
            Y-200000
      G31 X-500000
            Y-300000
      X0

```



16-2 Skip Coordinate Reading

The coordinates of the positions where skip signals are input will be stored into system variables #5061 (first axis) through #5066 (six axis). These coordinates can be called using user macros.

```

:
G90 G00 X-100.
G31 X-200. F60 — Skip command
#101=#5061 — Skip signal input coordinate value (in the workpiece coordinates
               system) is stored into variable #101.
:

```

16-3 Amount of Coasting

The amount of coasting of the machine from the time a skip signal is input during G31 command code setting until the machine stops differs according to the G31-defined feed rate or the F-command data contained in G31.

Accurate machine stop with a minimum amount of coasting is possible because of a short time from the beginning of response to a skip signal to the start of deceleration.

The amount of coasting is calculated as follows:

$$\begin{aligned}\delta_0 &= \frac{F}{60} \times T_p + \frac{F}{60} (t_1 \pm t_2) \\ &= \underbrace{\frac{F}{60} \times (T_p + t_1)}_{\delta_1} \pm \underbrace{\frac{F}{60} \times t_2}_{\delta_2}\end{aligned}$$

δ_0 : Amount of coasting (mm)

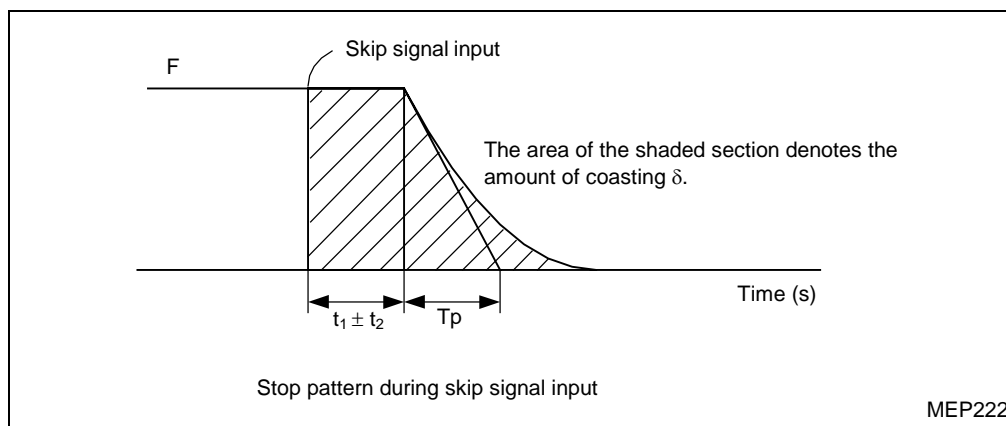
F : G31 skip rate (mm/min)

T_p : Position loop time constant (sec) = (Position loop gain)⁻¹

t_1 : Response delay time (sec) = (The time from skip signal detection until arrival at NC through PC)

t_2 : Response error time = 0.001 (sec)

Use of command G31 for measurement purposes allows any errors in measured data to be corrected using δ_1 . Such corrections, however, cannot be performed for δ_2 .



16-4 Skip Coordinate Reading Error

1. Reading the skip signal input coordinates

Skip signal input coordinate data does not include the amounts of coasting, defined by position loop time constant T_p and cutting feed time constant T_s .

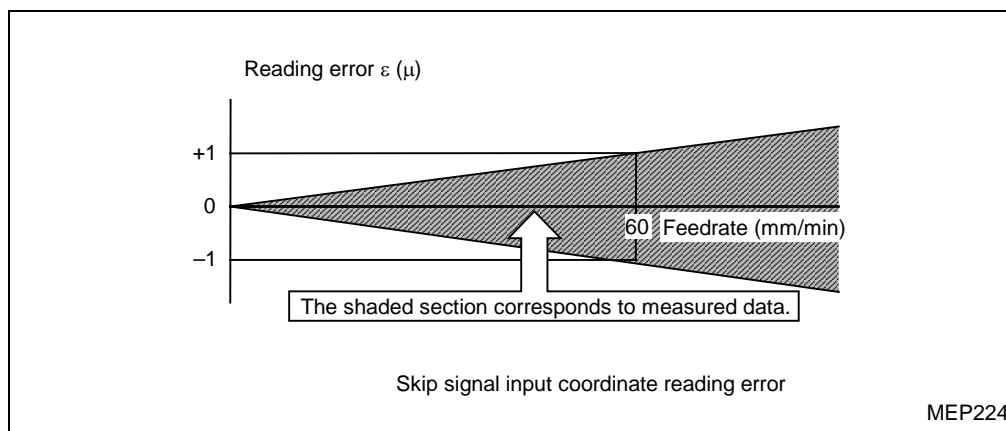
Thus, skip signal input coordinates can be checked by reading within the error range shown in the diagram below the workpiece coordinates existing when skip signals were input. The amount of coasting that is defined by response delay time t_1 , however, must be corrected to prevent a measurement error from occurring.

$$\varepsilon = \pm \frac{F}{60} \times t_2$$

ε : Reading error (mm)

F : Feed rate (mm/min)

t_2 : Response delay time 0.001 (sec)



2. Reading coordinates other than those of skip signal inputs

Coordinate data that has been read includes an amount of coasting. If, therefore, you are to check the coordinate data existing when skip signals were input, perform corrections as directed previously in section 16-3. If, however, the particular amount of coasting is that defined by response delay time t_2 , then a measurement error will occur since such data cannot be calculated.

3. Examples of correction for the amount of coasting

A. Correcting skip signal input coordinates

#110 = Skip feed rate

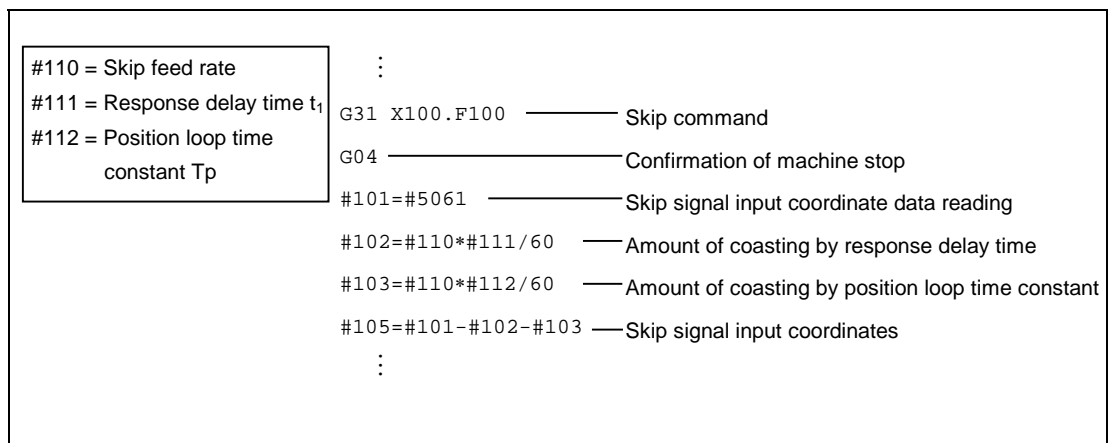
#111 = Response delay time t_1

```

:
G31 X100.F100  —— Skip command
G04  —— Confirmation of machining stop
#101=#5061  —— Skip signal input coordinate data reading
#102=#110*#111/60  —— Amount of coasting by response delay time
#105=#101-#102  —— Skip signal input coordinates
:

```

B. Correcting workpiece coordinates



16-5 Multi-Step Skip: G31.1, G31.2, G31.3, G04

1. Function and purpose

Conditional skipping becomes possible by previously setting a combination of skip signals that are to be input. Skipping occurs in the same manner as that done with G31.

The skip function can be designated using commands G31.1, G31.2, G31.3, or G04. The relationship between each of these G commands and the type of skip signal can be set by the parameters **K69** to **K73**.

2. Programming format

G31.1 Xx Yy Zz αα Ff (Same as for G31.2 or G31.3 Ff is not required for G04)

Feed rate (mm/min)

Axis address and target coordinate data

Using this programming format, you can execute linear interpolation in the same manner as that done using command G31.

During linear interpolation, the machine will stop when the previously set skip signal input conditions are satisfied, and then all remaining commands will be cancelled and the next block will be executed.

3. Detailed description

- For feed rates set by the parameter **K42** to **K44**, the following relationship holds:
 - G31.1 G31.1 skip feed rate
 - G31.2 G31.2 skip feed rate
 - G31.3 G31.3 skip feed rate
- The program will skip when the skip signal input conditions appropriate for each of these G commands are satisfied.
- Except for items other than 1 and 2 above, the description of command code G31 in section 16-3 also applies.

4. Parameter setting

1. The feed rate appropriate for each of the G31.1, G31.2, and G31.3 command codes can be set by the parameters **K42 to K44**.
2. The skip conditions appropriate for each of the G31.1, G31.2, G31.3 and G04 command codes are to be set by parameters **K69 to K73**. (The skip conditions refer to the logical sum of previously set skip signals).

A parameter setting of 7 is equivalent to G31.

Parameter setting value	Valid skip signals		
	1	2	3
1	○		
2		○	
3	○	○	
4			○
5	○		○
6		○	○
7	○	○	○

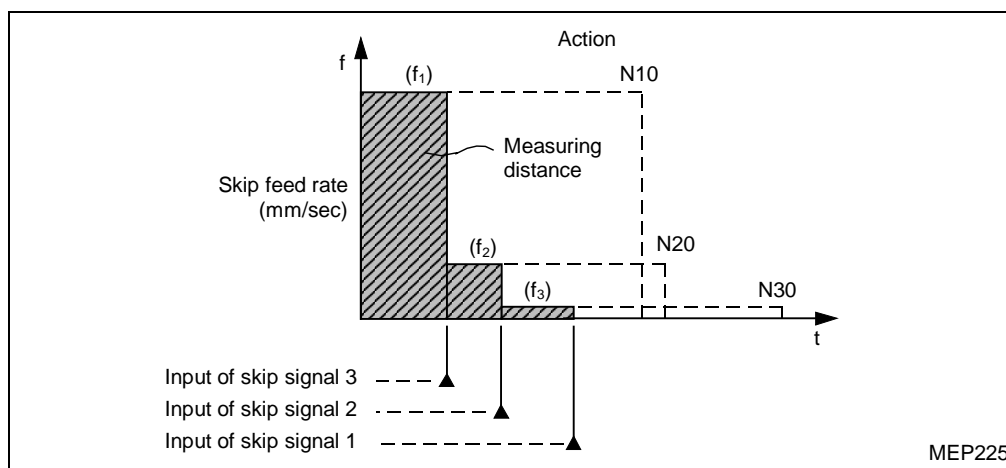
5. Machine action

1. Use of the multi-step skip function allows the following type of machine action control, and hence, reduction of the measurement time with improved measurement accuracy.
If parameter settings are as shown below.

Skip condition	Skip feed rate
G31.1 = 7	20.0 mm/min (f_1)
G31.2 = 3	5.0 mm/min (f_2)
G31.3 = 1	1.0 mm/min (f_3)

Sample program

```
N10 G31.1X200.0
N20 G31.2X40.0
N30 G31.3X1.0
```



Note: During the machine action shown above, if input of skip signal 1 precedes that of skip signal 2, the remaining distance of N20 will be skipped and N30 will also be ignored.

2. If the skip signal corresponding to the conditions previously set is input during dwell (command G04), the remaining time of dwell will be cancelled and the next block will be executed.

17 THREADING: G33 (Option)

17-1 Equal-Lead Threading

1. Function and purpose

Setting a G33 command in the program enables equal-lead threading under tool feed control synchronized with spindle rotation.

Also, multiple-thread screws can be machined by specifying the starting angle of threading. A d'ANDREA tool is required for fully automatic threading.

2. Programming format

A. Standard lead threading

G33 Zz Ff Qq

z: Threading direction axis address and the thread length

f: Lead in the direction of the long axis (the axis to be moved through the longest distance among all axes)

q: Threading start shift angle (0 to 360 degrees)

(The starting angle of threading becomes 0 degrees if omitted.)

B. Precise lead threading

G33 Zz Ee Qq

z: Threading direction axis address and the thread length

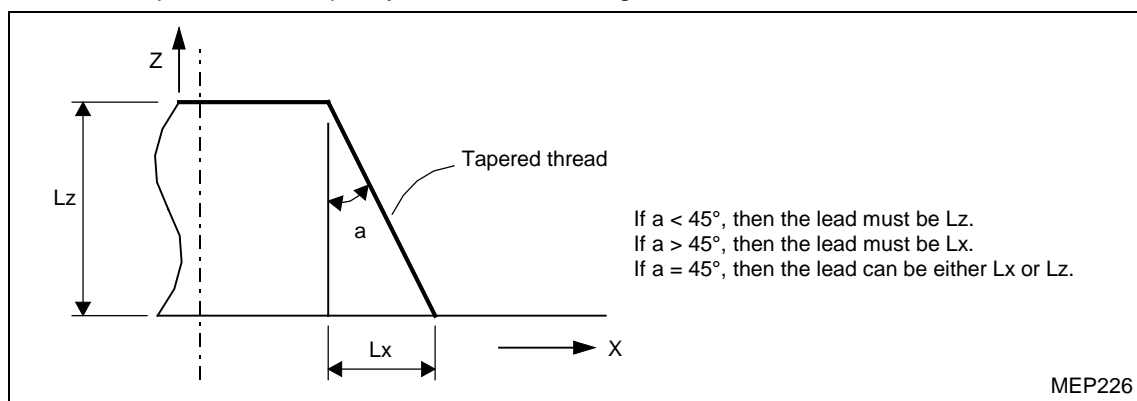
e: Lead in the direction of the long axis (the axis to be moved through the longest distance among all axes)

q: Threading start shift angle (0 to 360 degrees)

(The starting angle of threading becomes 0 degrees if omitted.)

3. Detailed description

- For a tapered screw, specify the lead in the long-axis direction.



The setting ranges of leads F and E are as follows:

Input unit	F-command lead data range (6-digit)	E-command lead data range (8-digit)
mm	0.001 to 99.9999 mm/rev.	0.00002 to 99.999999 mm/rev.
Inch	0.0001 to 9.99999 in./rev.	0.000002 to 9.9999999 in./rev.

Note: Alarm **134 SPINDLE ROTATION EXCEEDED** will be displayed if the feed rate, after being converted into a feed rate per minute, is in excess of the maximum cutting feed rate.

2. The E-command data will also be used as number-of-threads command data for threading in inches. Whether the command data is to be used only as precise lead data or number-of-threads data can be selected using bit 7 of parameter **F91**.
3. Keep the spindle speed constant during the entire machining cycle from rough cutting to finish cutting.
4. During threading, feed hold does not work. Pressing the feed hold button during threading brings the program to a block stop at the end of the block which immediately succeeds the block under the control of the G33 mode (that is, the block at which the threading operation has ended).
5. For tapered screws, since machining cannot be stopped in the middle of threading, the cutting feed rate after being converted may exceed the maximum cutting feed according to the particular command data.
To prevent this from occurring, therefore, the command lead data must be set according to the maximum feed rate obtained after conversion, not for the starting point of threading.
6. Usually, the leads at the beginning of threading and at the end of threading become incorrect because of a delay in the operation of the servo system. A thread length must therefore be specified that has a probable, incorrect lead length added to the required thread length.
7. The spindle speed undergoes the following restriction:

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

where R must be smaller than or equal to the maximum allowable encoder revolutions (rpm), and

R: Spindle speed (rpm)
 Thread lead: mm or inches
 Maximum feed rate: mm/min or inches/min

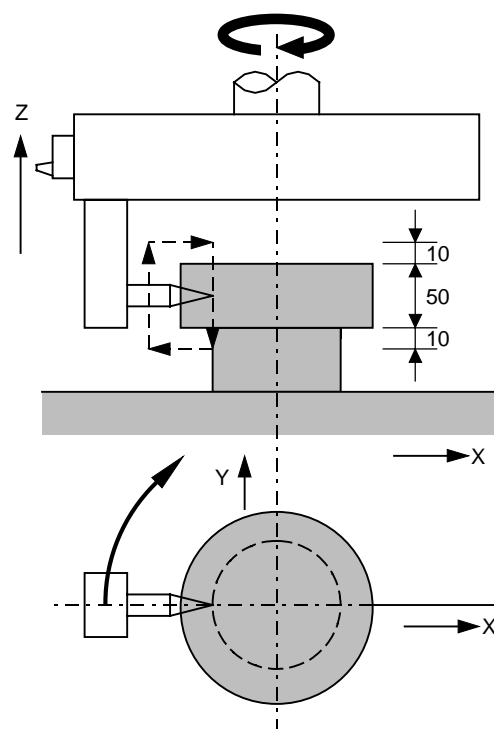
8. The threading start shift angle must be specified using an integer from 0 to 360.
9. The cutting feed override value is fixed at 100%.

4. Sample program

```

N110 G90G0X-200.Y-200.S50M3
N111 Z110.
N112 G33Z40.F6.0
N113 M19
N114 G0X-210.
N115 Z110.M0
N116 X-200.
      M3
N117 G04X5.0:
N118 G33Z40.

```



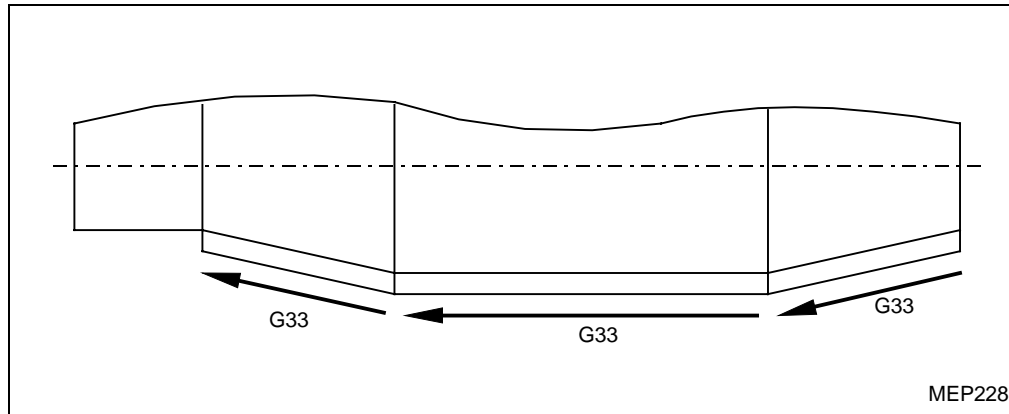
MEP227

<Operation description>

- | | |
|------------|------------------------------------------------------------------------------------------------------------------------|
| N110, N111 | The spindle center is positioned at the workpiece center and the spindle rotates forward. |
| N112 | The first threading operation is performed.
Tread lead = 6.0 mm (0.236 in.) |
| N113 | Spindle orientation based on an M19 command is performed. |
| N114 | The tool moves away in the X-axis direction. |
| N115 | The tool returns to a position above the workpiece, and an M00 command stops the program. Adjust the tool as required. |
| N116 | Preparations for the second threading operation are made. |
| N117 | Set a dwell time, as required, to stabilize the spindle speed. |
| N118 | The second threading operation is performed. |

17-2 Continuous Threading

Continuous threading becomes possible by setting threading command codes in succession in the program. Thus, special threads that change in lead and/or shape during threading can be machined. A D'ANDREA tool is required for continuous threading.



17-3 Inch Threading

1. Function and purpose

Including in a G33-command format the number of threads per inch in the long-axis direction enables tool feed to be controlled in synchronization with spindle rotation, and thus enables equal-lead straight threading and tapered threading.

2. Programming format

G33 Zz Ee Qq

- z: Threading direction axis address and the thread length
- e: Number of threads per inch in the direction of the long axis (the axis to be moved through the longest distance among all axes)
(A decimal point can be included.)
- q: Threading start shift angle (0 to 360 degrees)

3. Detailed description

1. The number of threads per inch must be that existing in the long-axis direction.
2. E-command data will also be used as command data for precise lead threading. Whether the command data is to be used only as number-of-threads command data or precise lead command data can be selected using bit 7 of parameter **F91**.
3. E-command data value must be in the setting range for the command.

4. Sample program

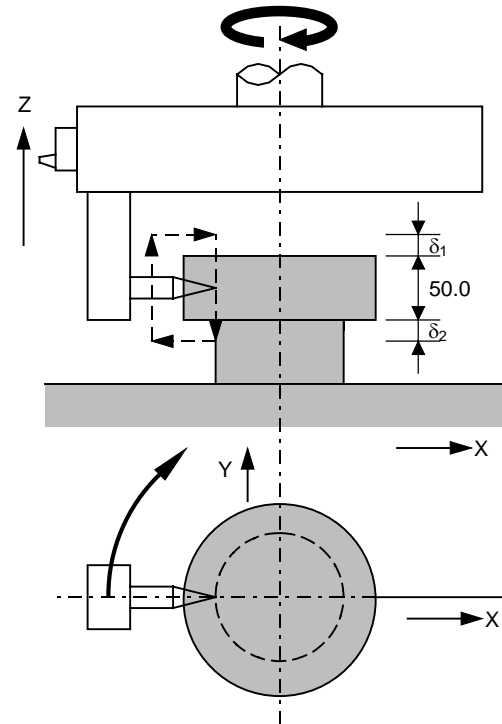
Thread lead3 thread leads/inch (= 3.46666...)

$\delta_1 = 10 \text{ mm}$ $\delta_2 = 10 \text{ mm}$

If programmed in millimeters:

```

N210 G90G0X-200.Y-200.S50M3
N211 Z110.
N212 G91G33Z-70.E3.0      (First threading)
N213 M19
N214 G90G0X-210.
N215 Z110.M0
N216 X-200.
      M3
N217 G04X2.0
N218 G91G33Z-70.          (Second threading)
  
```



MEP227

- NOTE -

18 AUTOMATIC TOOL LENGTH MEASUREMENT: G37 (Option)

1. Function and purpose

When the tool for which command data has been assigned moves to a programmed measurement position, the NC system will measure and calculate any differential data between the coordinates at that time and those of the programmed measurement position. Data thus obtained will become offset data for that tool.

Also, if offsetting has already been performed for the tool, the current offset data will be further offset, provided that after movement of that tool under an offset status to the required measurement position, the measurements and calculations of any differential coordinates show some data to be further offset.

At this time, further offsetting will occur for the tool offset data if only one type of offset data exists, or for the tool wear offset data if two types of offset data exist (tool length offsets and tool wear offsets).

2. Programming format

G37 Z_ (X_, Y_) R_ D_ F_

X, Y, Z: Address of the measurement axis and the coordinate of the measurement position

R: Distance from the starting point of movement at a measurement feed rate, to the measurement position

D: The area where the tool is to stop moving

F: Measurement feed rate

If R, D, or F is omitted, respective parameter values will become valid.

3. Description of parameters

Parameter	Description
F42	R-code command. Deceleration area
F43	D-code command. Measurement area
F44	F-code command. Measurement feed rate f
F72	Conditions for skipping based on EIA G37

See the Parameter List for further details.

4. Example of execution

If $H01 = 0$

T01T00M06

G90G00G43Z0H01

G37Z-600.R200.D150.F300

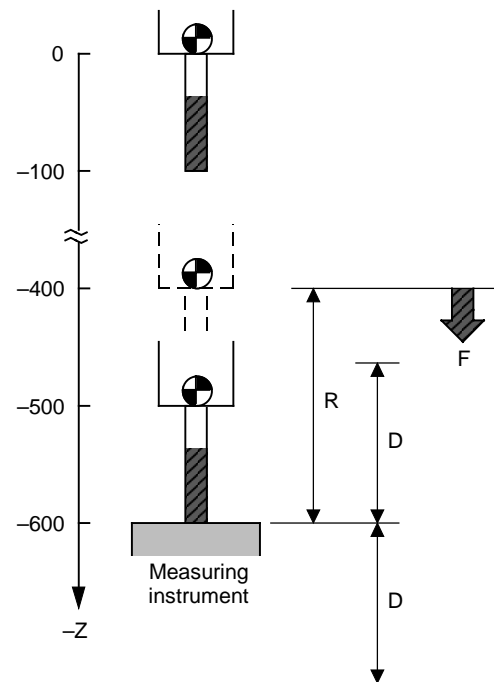
Coordinate to reach the measurement position

= -500.01

$-500.01 - (-600) = 99.99$

$0 + 99.99 = 99.99$

Thus, $H01 = 99.99$



MEP229

If $H01 = 100$

T01T00M06

G90G00G43Z-200.H01

G37Z-600.F300

Coordinate to reach the measurement position

= -600.01

$-600.01 - (-600) = -0.01$

$100 + (-0.01) = 99.99$

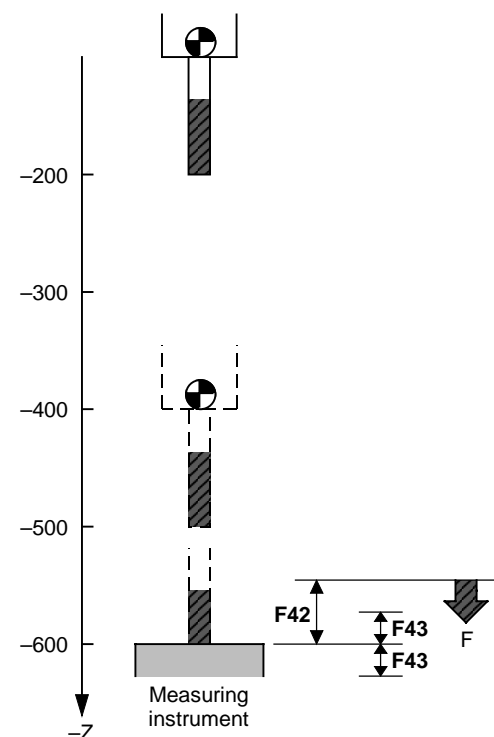
Thus, $H01 = 99.99$

<Supplement>

When the program shown above is executed, parameter **F42** and **F43** are set as follows:

F42 (R-code command) : 25000 (25 mm)

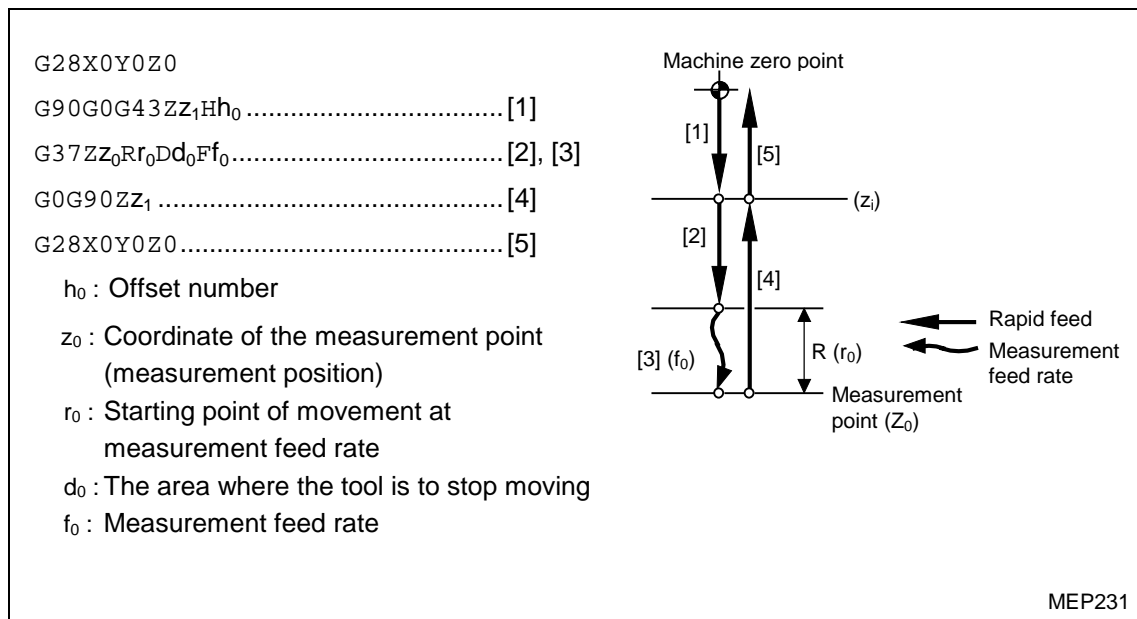
F43 (D-code command) : 2000 (2 mm)



MEP230

5. Detailed description

1. Machine action based on command G37



2. Sensor signals (Measurement Position Reached) also act as skip signals.
3. If the F-code value is 0, the feed rate becomes 1 mm/min.
4. Update offset data becomes valid from the Z-axis (measurement axis) command codes that succeed the block of G37.
5. The delay and dispersion in processing of sensor signals, except for the PLC side, is from 0 to 0.2 msec for the NC side alone. Accordingly, the following measurement error may occur:

Maximum measurement error [mm]

$$= \text{Measurement feed rate [mm/min]} \times \frac{1}{60} \times \frac{0.2 [\text{ms}]}{1000}$$

6. When a sensor signal is detected, although the coordinates of the machine position at that time will be read, the machine will stop only after overrunning through the distance equivalent to a servo droop.

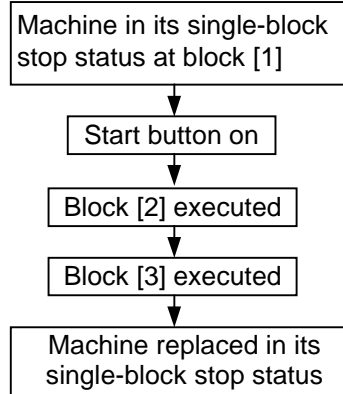
Maximum amount of overrun [mm]

$$= \text{Measurement feed rate [mm/min]} \times \frac{1}{60} \times \frac{30.3 [\text{ms}]}{1000}$$

30.3 [msec] if the position loop gain is 33.

7. If command G37 is executed in the single-block operation mode, the machine will come to a single block stop after execution of the block that immediately succeeds the G37-containing block.

Example: G0G90G43Z-200.H01 [1]
 G37Z-600.R25.D2.F10 [2]
 G0G90Z-200. [3]



6. Precautions

- Alarm **889 G37 OPTION NOT FOUND** will result if G37 is set for a machine that does not have a mounted option for automatic tool length measurement.
- Alarm **923 ILLEGAL COMMAND G37 AXIS** will result if the block of G37 does not contain axis data or contains data of two or more axes.
- Alarm **924 G37, H COMMANDS SAME BLOCK** will result if an H code exists in the block of G37.
- Alarm **925 H CODE REQUIRED** will result if G43 H_ does not exist before the block of G37.
- Alarm **926 ILLEGAL G37 SIGNAL** will result if input sensor signals occur outside a predetermined allowable measurement range or if a sensor signal is not detected on arrival of the tool at the ending point of movement.
- If a manual interruption operation has been carried out during movement of the tool at a measurement feed rate, the program must be restarted only after returning that tool to the position existing when the interruption operation was carried out.
- Set G37 data or parameter data so that the following condition is satisfied:

$$\text{Measurement point} - \text{Starting point} > \begin{matrix} \text{R-code value} \\ \text{or parameter r} \end{matrix} > \begin{matrix} \text{D-code value} \\ \text{or parameter d} \end{matrix}$$

- If the R-code value, the D-code value and parameter d, mentioned in Item G above, are all 0s, the program will come to a normal end only when the designated measurement point and the sensor signal detection point agree. Alarm **926 ILLEGAL G37 SIGNAL** will result in all other cases.
- If the R-code value, the D-code value, parameter r, and parameter d, mentioned in Item G above, are all 0s, alarm **926 ILLEGAL G37 SIGNAL** will result after the tool has been positioned at the designated measurement point, irrespective of whether a sensor signal is detected.
- Set G37 (automatic tool length measurement code) together with G43 H_ (offset number assignment code).

G43 H_
 G37 Z_R_D_F_

11. If the offset data is tool offsets of type A, then automatic correction of tool data occurs, or if the offset data is tool offsets of type B, then automatic correction of tool wear offsetting data occurs.

Example: The **TOOL OFFSET** displays in both cases after offsetting of H1 = 100

	TOOL OFFSET (Type A)				TOOL OFFSET (Type B)		
	No.	OFFSET	No.	OFFSET	TOOL LENGTH		
					No.	GEOMETRY	WEAR
Before measurement	1	100	17	0	1	100	0
	2	0	18	0	2	0	0
	3	0	19	0	3	0	0
	No.	OFFSET	No.	OFFSET	TOOL LENGTH		
					No.	GEOMETRY	WEAR
After measurement	1	110	17	0	1	100	10
	2	0	18	0	2	0	0
	3	0	19	0	3	0	0

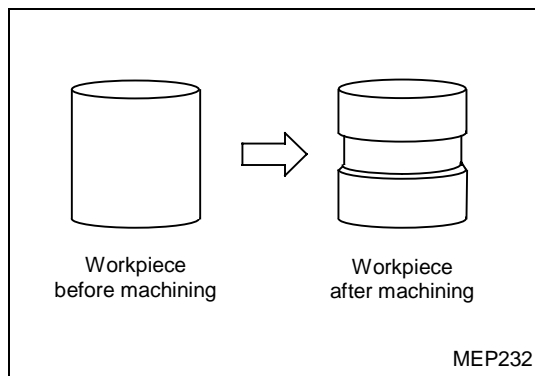
12. The distance from the machine zero point to the measurement point (skip sensor) is preset in register R2392 or R2393. Use this value as reference to set a coordinate using Z-, X-, or Y-code command.
13. When this function is used for tool offsets of type B, the correct data will not be displayed if the wear offset value exceeds 100.
14. When executing this function in the presence of offset data, set the value of a D code to 2mm or less to prevent damaging the measuring instrument.
15. When executing this function in the absence of offset data (offset data = 0), set the values of an R code and a D code to those larger than the tool length of the tool to be measured. Also, in that case, before executing this function, make sure that the skip sensor in the measuring instrument correctly operates.

- NOTE -

19 DYNAMIC OFFSETTING: M173, M174 (Option)

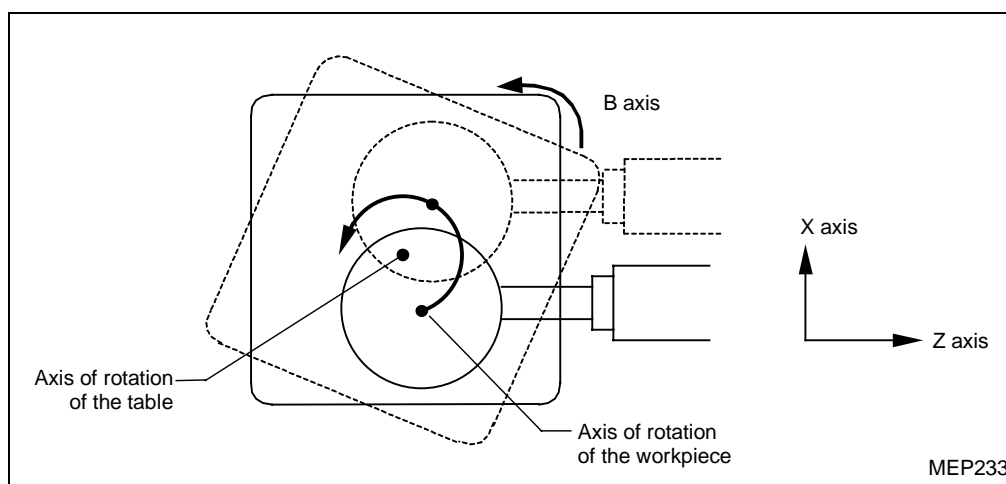
1. Function and purpose

Machining with rotation of the rotary table (B-axis) requires in principle the axis of rotation of the workpiece to be completely aligned with the axis of rotation of the table.



In practice, however, this is very difficult to implement for the reasons of fixture design, unless a very precise fixture is used.

Dynamic offsetting is a function that internally compensates continuous deviation due to the misalignment in question. As a result, machining program can easily be prepared on the assumption that the alignment is ideal.



M173

Dynamic offsetting ON

G01 B360. F500

M174

Dynamic offsetting OFF

2. Detailed description

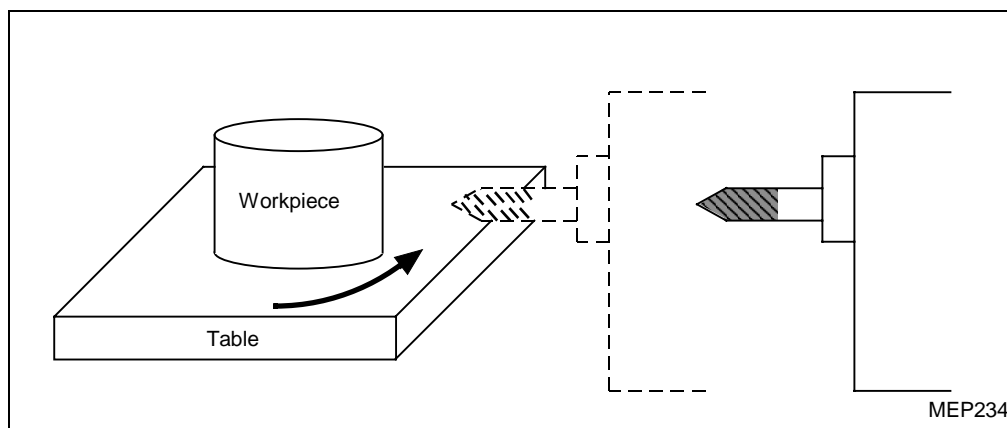
- Automatic limitation by the software does not occur even if dynamic offsetting may cause the stroke limit to be overstepped.
- Reduce to 3 mm or less the eccentricity of an actually mounted workpiece from the axis of rotation of the table; otherwise the alarm **137 DYNAMIC COMPENSATION EXCEEDED** is caused.

Automatic operation is executed on the assumption that the origin of the workpiece coordinates lies on the axis of rotation of the workpiece. Manual operation uses the data in parameter **I11** (as described later).

- The workpiece origin must be set on the axis of rotation of a workpiece when it is to be machined using dynamic offsetting.
- Dynamic offsetting is not effective in the 3-dimensional coordinates conversion mode (G68).
- The related parameters are as follows:

Address	Name	Description
S5	Axis of table rotation	Set the machine coordinates of the axis of the table rotation for the controlled axes concerned.
I11	Axis of workpiece rotation	Set the machine coordinates of the axis of the workpiece rotation, existing at a table angle of 0 degrees, for the controlled axes concerned. (This parameter is valid only for manual operation.)

- Dynamic offsetting is provided for the type of machining that can in principle be achieved by turning the workpiece only with the tool in a fixed position.



3. Sample program

```

G55 ..... The workpiece axis is assumed to go through the origin of the G55 system.
G0X_Y_Z_ ..... Approach
M173 ..... Dynamic offsetting ON
G1Z_F_ ..... Start of cutting
  B_F_ ..... B-axis rotation
  Z_F_ ..... Relief on the Z-axis
M174 ..... Dynamic offsetting OFF
M30 ..... End of machining

```

20 HIGH-SPEED MACHINING MODE FEATURE (OPTION)

1. Function and purpose

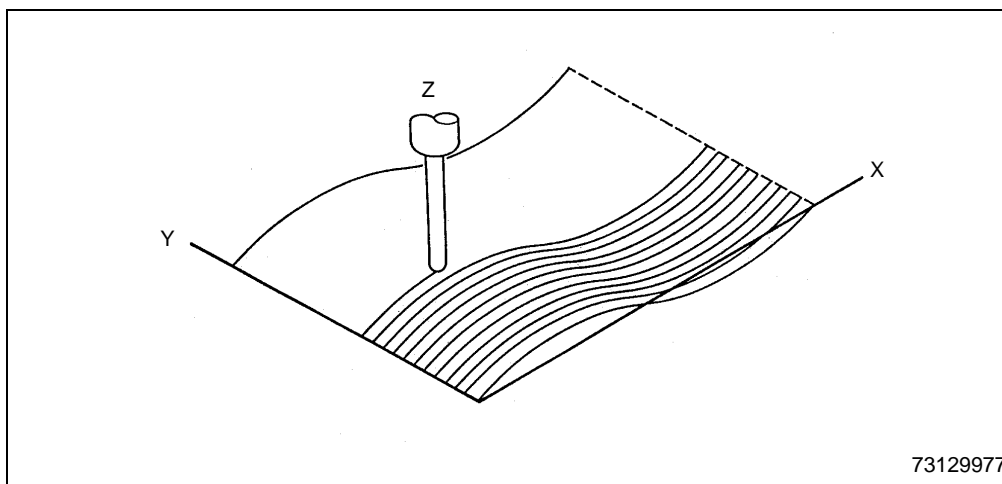
The high-speed machining mode feature allows high-speed execution of programs used for the machining of free-curved surfaces that have been approximated using very small lines.

In high-speed machining mode, microsegment machining capabilities improve by several times, compared with conventional capabilities. This allows the same machining program to be executed at several times the original feed rate, and thus the machining time to be reduced significantly.

Conversely, a machining program that has been approximated using lines of several fractions of the original segment length, can also be executed at the same feed rate, so more accurate machining is possible.

Combined use of the high-speed machining mode and the shape correction function allows more accurate machining to be implemented.

If, moreover, a protruding section exists in the microsegment machining program, smooth interpolation can be conducted automatically by removing this illegal path.



High-speed machining is available in the automatic operation modes: Memory, HD (Hard Disk), IC card and Ethernet.

Even in the high-speed machining mode can be applied various operational functions: override functions, cutting feed rate limit function, single-block operation function, dry run function, graphic trace function and high-precision control function.

The microsegment machining capability in the high-speed machining mode is as follows:

Operation mode	Max. speed	Conditions required
Memory operation	270 m/min (10630 IPM)	None
HD operation	16.8 m/min (661.4 IPM)	None
	270 m/min (10630 IPM)	Optional function for the Ethernet operation With the POSITION display selected on the screen (see Note 2)
Ethernet operation	270 m/min (10630 IPM)	Avoid unusual key operations (see Note 3)
IC card operation	270 m/min (10630 IPM)	None

The microsegment machining capability is restricted further by the functions used in, or applied to, the program as shown below:

Preparatory functions		Faring function	
		Not applied	Applied
G01	Linear interpolation only	270 m/min (10630 IPM)	168 m/min (6614 IPM)
G02/G03	Circular interpolation included	67 m/min (2638 IPM)	
G6.1	Fine-spline interpolation included	200 m/min (7874 IPM)	135 m/min (5315 IPM)

Note 1: The microsegment machining capabilities shown above refer to the case where three-axis simultaneous motion commands consist of 32 characters per block for a segment length of 1 mm.

Note 2: If the **POSITION** display should be changed to any other display during operation, program reading from the hard disk may be aborted to damage the surface to be machined.

Note 3: If unusual operations, such as holding down any cursor/page key or a mouse button, are performed, program reading from the network may be aborted to damage the surface to be machined.

Note 4: Before executing a microsegment machining program for hard disk operation or Ethernet operation, terminate the commercially available software if it is being used.

Note 5: Since optimum corner deceleration occurs during the shape correction mode, the machining time may be longer than in other modes.

2. Programming format

G5 P2 High-speed machining mode ON
G5 P0 High-speed machining mode OFF

Note: Both commands must be given in a single-command block.

3. Commands available in the high-speed machining mode

Only axis motion commands with the corresponding preparatory functions (G-codes) and feed functions (F-codes), and designation of sequence number are available in the high-speed machining mode. Setting data of any other type will result in an alarm (**807 ILLEGAL FORMAT**).

1. G-codes

The available preparatory functions are G00, G01, G02 and G03.

The circular interpolation can be programmed with R (radius designation) as well as with I and J (center designation). If the machining program includes circular commands, however, make bit 2 of the **F96** parameter valid.

F96 bit 2: Type of control for circular commands in the high-speed machining mode:

- 0: Control for the specified speed (with acceleration/deceleration)
- 1: Control for a uniform feed

2. Axis motion commands

The three linear axes (X, Y, Z) can be specified.

Absolute data input as well as incremental data input is applicable, indeed, but the former input mode requires the validation of bit 5 of the **F84** parameter.

F84 bit 5: Type of position data input in the high-speed machining mode:

- 0: Always incremental data input
- 1: According to the input mode before selection of the high-speed machining mode

3. Feed functions

Feed rate can be specified with address F.

4. Sequence number

Sequence number can be specified with address N. This number, however, is skipped as a meaningless code during reading.

5. Sample program

```
G28 X0 Y0 Z0
G90 G0X-100.Y-100.
G43 Z-5.H03
G01 F3000
G05 P2 _____ High-speed machining mode ON
X0.1
X0.1 Y0.001
X0.1 Y0.002
      ⋮
X0.1 F200
G05 P0 _____ High-speed machining mode OFF
G49 Z0
M02
```

When **F84** bit 5 = 0:
Incremental motion under G01

When **F84** bit 5 = 1:
Absolute motion under G01

Note 1: Either 0 or 2 is to be set with address P (P0 or P2). Setting any other value will result in an alarm (**807 ILLEGAL FORMAT**).

Note 2: No other addresses than P and N must be set in the same block with G05.

Note 3: A decimal point must not be appended to address P.

Note 4: Unclamp the table concerned beforehand (by the corresponding M-code, etc.) to give a command for rotational axis motion.

4. Additional functions in the high-speed machining mode

A. Fairing function

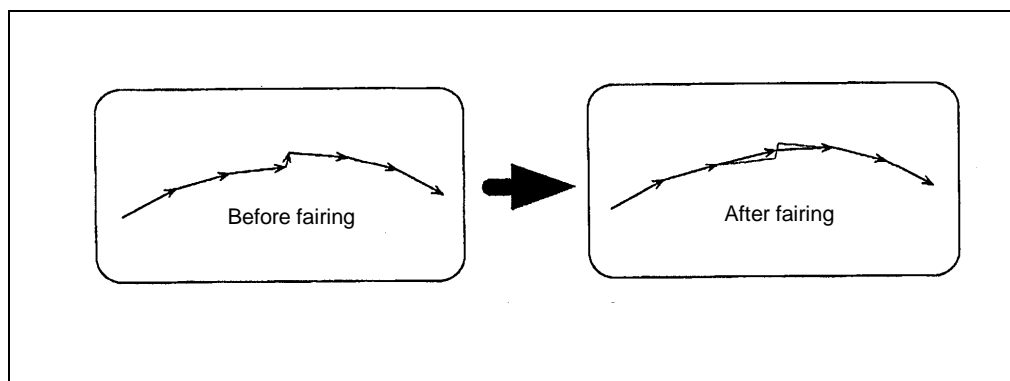
If, in a series of linear paths, a protruding section exists in the CAM-created microsegment machining program, this protruding path can be removed and the preceding and following paths connected smoothly by setting parameter **F96** bit 1 to "1".

F96 bit 1: Faring function for the microsegment machining program

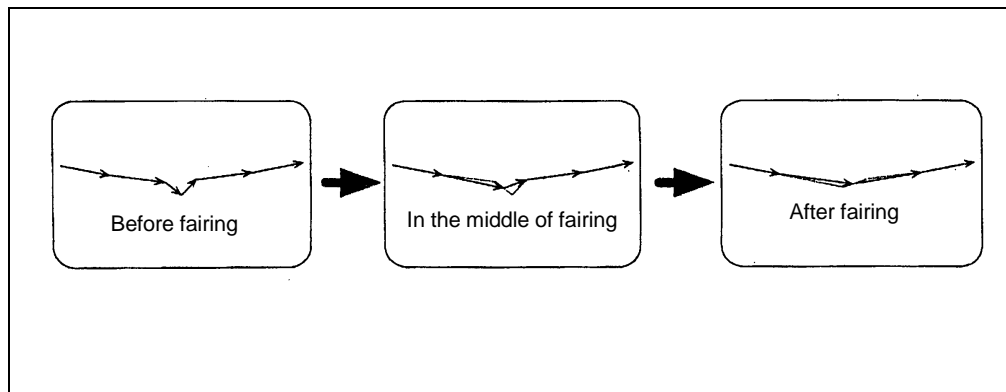
0: No faring

1: Faring for a protruding path

F103: Maximum length of a block to be removed for fairing



Fairing is also valid for a succession of protruding paths as shown below:

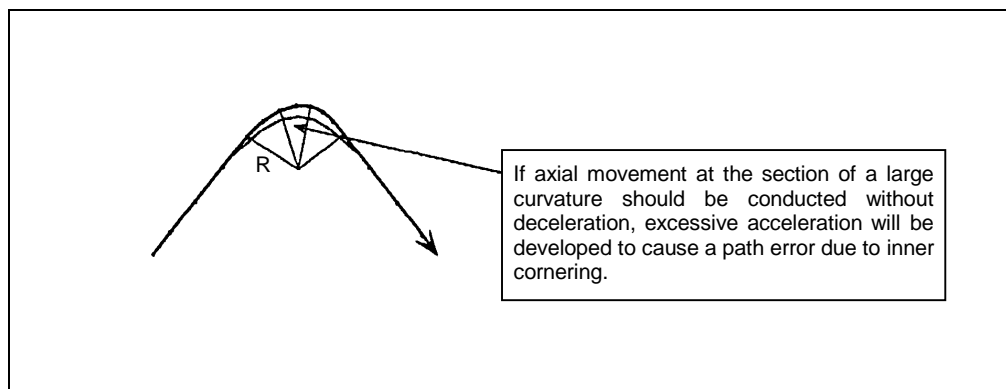


B. Cutting feed limiting speed

In shape correction mode, the minimum of the cutting feed limiting speeds of the movable axes is set as the cutting feed limiting speed in the high-speed machining mode. Setting parameter **F96** bit 5 to "1", however, allows the curvature of every curved section to be judged for limiting the speed so as not to exceed the maximum available acceleration.

F96 bit 5: Type of cutting feed limiting speed for the high-speed machining mode

- 0: Minimum of the cutting feed limiting speeds of the movable axes
- 1: Limiting speed based on the radius of curvature



C. Deceleration at corners in the high-speed machining mode

In shape correction mode, automatic deceleration at corners of significantly large angle is provided in general to ensure that the acceleration developed during cornering shall fall within the predetermined tolerance.

A micro-length block between relatively longer blocks intersecting each other in a large angle in CAM-created microsegment machining programs, in particular, may cause the cornering speed to mismatch the surroundings and thus affect surface quality.

Setting parameter **F96** bit 4 to "1" will now allow corner judgment and deceleration without suffering any effects of such a microblock.

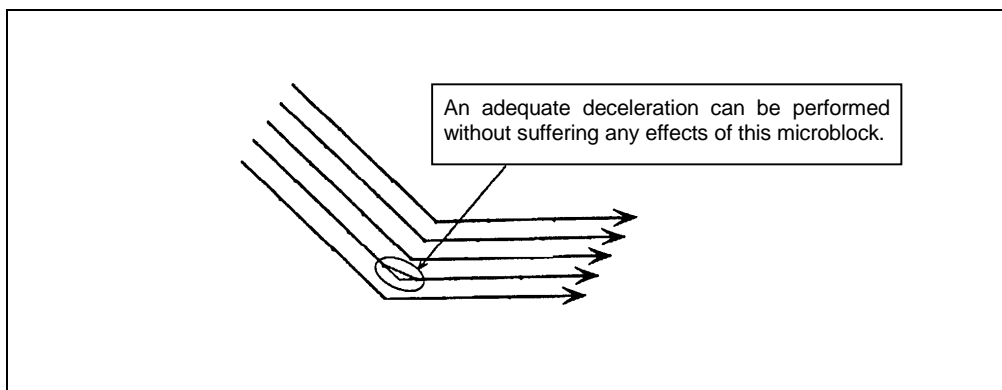
To use this function, however, the high-accuracy control option is required in addition to the optional high-speed machining function.

F96 bit 4: Type of corner judgment in the high-speed machining mode

0: Always judging from the angle between adjacent blocks

1: Judging after removing any microblock (if present between large-angle blocks)

F107: Reference length for microblock judgement



5. Restrictions

- The modal functions other than that of G-code group 01 will be saved during, and restored upon cancellation of, the high-speed machining mode, indeed; but the modal functions for tool diameter offset, mirror image, scaling, coordinate system rotation, virtual axis interpolation and three-dimensional diameter offset should have been cancelled beforehand to give a G05 P2 command. Otherwise, an alarm may be caused or the modal function unexpectedly cancelled.

Example: Main program

```
G28 X0 Y0 Z0
```

```
G90 G92 X0 Y0 Z100.
```

```
G00 X-100.Y-100.
```

```
G43 Z-10.H001
```

Movement under the conditions of G90, G00 and G43

```
M98 H001
```

```
G49 Z0
```

Movement under the conditions of G90 and G01

```
G28 X0 Y0 Z0
```

```
M02
```

Subprogram (O001)

```
N001 F3000
```

```
G05 P2
```

```
G01 X0.1
```

```
X-0.1 Y-0.001
```

```
X-0.1 Y-0.002
```

```
⋮
```

```
X0.1
```

```
G05 P0
```

```
M99
```

High-speed machining mode ON

When **F84** bit 5 = 0:

Incremental motion under G01

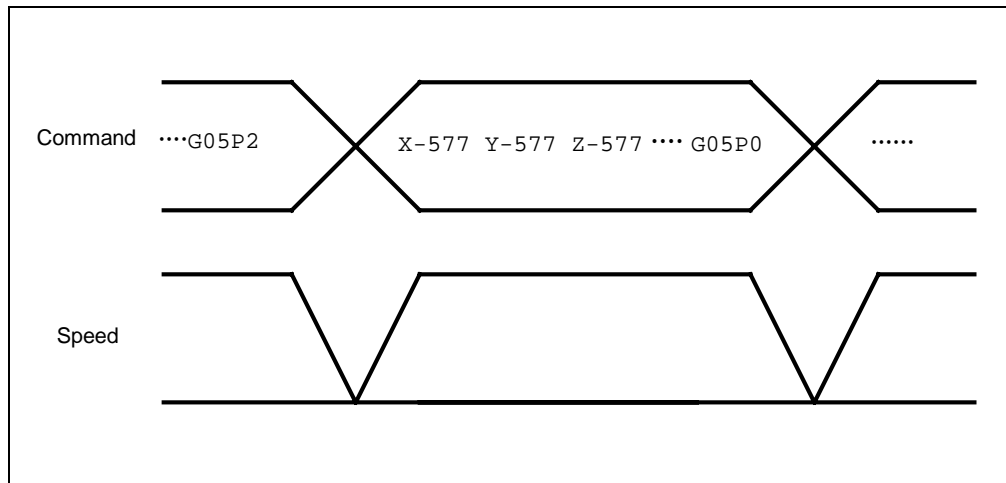
When **F84** bit 5 = 1:

Absolute motion under G01

High-speed machining mode OFF

- In the high-speed machining mode there may occur a delay in display response since priority is always given to the processing for the automatic operation.

3. The high-speed machining mode should be selected and cancelled by using commands of G05 P2 and G05 P0, respectively, with the tool sufficiently cleared from the workpiece since the selection and cancellation always cause a deceleration of feed motions as shown below:



4. Restrictions on programming and machine operation are listed in the following table:

O: Valid, -: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)
Classification	Subclassification		
Control axes	Maximum controllable axis quantity	6	6
	Effective controllable axis quantity	6	6
	Simultaneously controllable axis quantity	4	3
	Axis name	O	O (O)
	CT axis	O	O (O)
Units of control	Unit of input	ABC	ABC
	Unit of programming	O	O
	Unit-of-programming × 10	O	O
Input formats	Tape code	ISO/EIA	ISO/EIA
	Label skip	O	- (-)
	ISO/EIA automatic identification	O	O (O)
	Parity H	O	O (O)
	Parity V	O	O (O)
	Tape format	O	Refer to the programming format.
	Program number	O	O (err)
	Sequence number	O	O (O)
	Control IN/OUT	O	O (err)
	Optimal block skip	O	O (err)
Buffers	Tape input buffer	O	O (O)
	Pre-read buffer	O	O (O)
Position commands	Absolute/incemental data input	O	O (err)
	Inch/metric selection	O	O (err)
	Decimal point input	O	O (O)

O: Valid, -: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)
Classification	Subclassification		
Interpolation functions	Positioning	O	O (O)
	One-way positioning	O	- (err)
	Linear interpolation	O	O (O)
	Circular interpolation	O	O (O)
	Helical cutting	O	- (err)
	Spiral interpolation	O	- (err)
	Virtual-axis interpolation	O	- (err)
	Threading	O	- (err)
	Plane selection	O	O (err)
	Fine-Spline interpolation	O	O (err)
	NURBS interpolation	O	- (err)
Feed functions	Rapid feed rate	O	O (O)
	Cutting feed rate	O	O (O)
	Synchronous feed	O	O (err)
	Automatic acceleration/deceleration	O	O (O)
	Linear acceleration/deceleration before cutting interpolation	O	O (err)
	Cutting feed rate limitation	Limitation in cutting direction	Minimum limiting speed of feed axes/ According to curvature
	Rapid feed override	O	O (O)
	No. 1 cutting feed override	O	O (O)
	No. 2 cutting feed override	O	O (O)
	Exact-stop mode	O	- (err)
	Cutting mode	O	O (err)
	Tapping mode	O	- (err)
	Automatic corner override	O	-
	Error detection	O	O (O)
	Override cancellation	O	O
Dwell	Dwell in time	O	- (err)
	Dwell in number of revolutions	O	- (err)
Miscellaneous function	M-command	O	O (err)
	M independent output command	O	- (err)
	Optional stop	O	- (err)
	No. 2 miscellaneous functions	O	O (err)
Spindle functions	S-command	O	O (err)
Tool functions	T-command	O	O (err)
	Tool operation time integration	O	O (O)
	Spare-tool selection	O	O (-)
Tool offset functions	Tool-length offset	O	O (err)
	Tool-position offset	O	- (err)
	Tool-diameter offset	O	- (err)
	3D-tool-diameter offset	O	- (err)
	Tool-offset memory	O	O (O)
	Number of tool offset data sets	O	O (O)
	Programmed tool-offset input	O	- (err)
	Tool-offset number auto selection	O	O (err)

O: Valid, -: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)
Classification	Subclassification		
Program auxiliary functions	Fixed cycle for drilling	O	- (err)
	Pattern cycle	-	- (-)
	Subprogram control	O	O (err)
	Variable command	O	- (err)
	Figure rotation	O	- (err)
	Coordinate rotation	O	- (err)
	User macro	O	O (err)
	User macro interruption	O	O (err)
	Scaling	O	- (err)
	Mirror image	O	- (err)
	Geometric function	O	- (err)
	Geometric function	O	- (err)
	Programmed parameter setting	O	err (err)
Coordinate system setting	Watchdog-based reference-point return	O	O (-)
	Memory-based reference-point return	O	O (-)
	Automatic reference-point return	O	- (err)
	#2/#3/#4 reference-point return	O	- (err)
	Reference-point check	O	- (err)
	Machine coordinate system offset	O	- (err)
	Workpiece coordinate system offset	O	- (err)
	Local coordinate system offset	O	- (err)
	Coordinate system setting	O	- (err)
	Coordinate system rotation setting	O	- (err)
	Program restart	O	O (err)
	Absolute data detection	O	O (O)
Machine error correction	Backlash correction	O	O (O)
	Lost-motion correction	O	O (O)
	Memory-based pitch error correction	O	O (O)
	Memory-based relative position correction	O	O (O)
	Machine coordinate system correction	O	O (O)
Protection functions	Emergency stop	O	O (O)
	Stroke end	O	O (O)
	Software limit	O	O (O)
	Programmed software limit	O	- (err)
	Interlock	O	O (O)
	External deceleration	O	O (O)
	Data protection	O	O (O)

O: Valid, -: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)
Classification	Subclassification		
Operation modes	Tape operation	O	O (-)
	Memory operation	O	O (-)
	MDI operation	O	O (O)
	Jog feed	O	- (O)
	Incremental feed	O	- (O)
	Handle feed	O	- (O)
	Manual rapid feed	O	- (O)
	Handle interruption	O	O (O)
	Auto/manual simultaneous	O	O (O)
	HD operation	O	O (-)
	IC card operation	O	O (-)
	Ethernet operation	O	O (-)
External control signals	Automatic-operation start	O	O (O)
	Automatic-operation halt	O	O (O)
	Single-block stop	O	O (O)
	NC reset	O	- (O)
	External reset	O	- (O)
	All-axis machine lock	O	O (O)
	Axis-by-axis machine lock	O	O (O)
	Dry run	O	O (O)
	Miscellaneous-function lock	O	O (O)
	Manual-absolute selection	O	O (-)
Status output signals	Control-unit ready	O	O (O)
	Servo-unit ready	O	O (O)
	Auto-run mode	O	O (O)
	Auto-run in progress	O	O (O)
	Auto-run halted	O	O (O)
	Cutting feed in progress	O	O (O)
	Tapping in progress	O	- (-)
	Threading in progress	O	- (-)
	Axis selected	O	O (O)
	Axis-movement direction	O	O (O)
	Rapid feed in progress	O	O (O)
	Rewind		O (O)
	NC alarm	O	O (O)
	Reset	O	O (O)
	Movement-command completed	O	O (O)
Measurement aid functions	Manual tool-length measurement	O	- (-)
	Automatic tool-length measurement	O	- (err)
	Skip	O	- (err)
	Multi-step skip	O	- (err)
	Manual skip	O	- (err)
Axis control functions	Servo off	O	O (O)
	Follow-up	O	O (O)
	Control-axis removal	O	O (O)

O: Valid, -: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)
Classification	Subclassification		
Data input/output	External data input I/F	O	O (O)
	External data output I/F	O	O (O)
	External data input/output	O	O (O)
Setting/display functions	Setting/Display unit	O	O (O)
	Settings display	O	O (O)
	Search	O	O (err)
	Check-and-stop	O	- (-)
	MDI	O	O (O)
	Program restart	O	O (err)
	Machining-time calculation	O	O (O)
	PC opening	O	O (O)
	Program-status display	O	O (O)
	Integrated-time display	O	O (O)
	Graphics display	O	O (O)
Program creation	Multi-step skip	O	- (err)
	Graphics check	O	O (O)
Self-diagnostics	Program-error display	O	O (O)
	Operation-error display	O	O (O)
	Servo-error display	O	O (O)
	Operation-stop-cause display	O	O (O)
	Servo monitor display	O	O (O)
	NC-PC I/O signal display	O	O (O)
	DIO display	O	O (O)
	Keyboard-operation record	O	O (O)

21 FIVE-SURFACE MACHINING FUNCTION (OPTION)

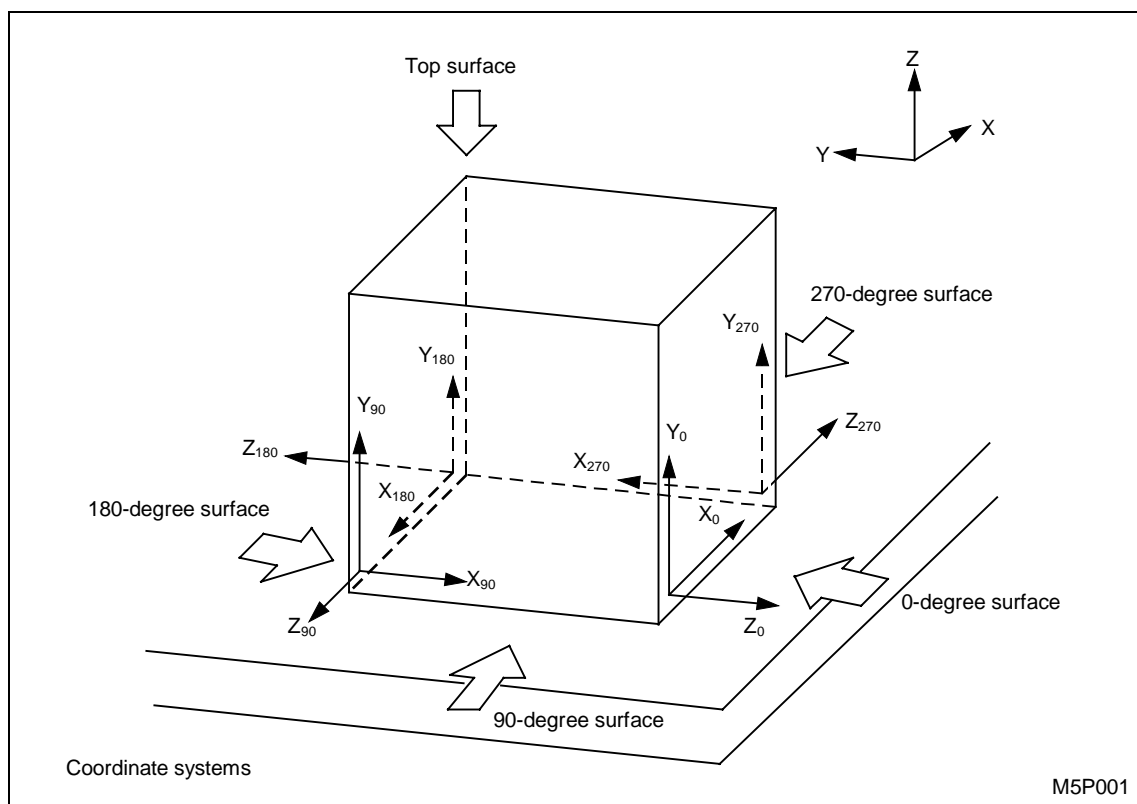
EIA/ISO programs for any surface-machining can be readily created since all surfaces can be regarded as X-Y planes.

Use G-codes to designate the surface for machining.

G-codes are also used to carry out head offsetting, and once a G-code is used, head offsetting will be performed in a direction corresponding to the selected surface-machining mode until the G-code is cancelled.

21-1 Coordinate Systems for Five-Surface Machining

When creating a five-surface machining program, divide the five surfaces into a top surface, a 0-degree surface, a 90-degree surface, a 180-degree surface and a 270-degree surface. Then, define workpiece coordinate systems, such as those shown in the figure below, for respective surfaces.



X, Y, Z : Machine coordinate system

X_n, Y_n, Z_n : Workpiece coordinate system for each surface (n = 0, 90, 180 or 270 degrees)

The workpiece coordinate system for the top surface is the same as the machine coordinate system.

21-2 Surface Selection Command

1. Function and purpose

Specifying a surface allows that surface to be handled as an X-Y plane. Otherwise, the head offset function cannot be used.

2. Programming format

G17.1 : Top-surface machining mode

G17.2 : 0-degree surface machining mode

G17.3 : 90-degree surface machining mode

G17.4 : 180-degree surface machining mode

G17.5 : 270-degree surface machining mode

G17.9 : Cancellation of the selected machining mode

3. Detailed description

After specifying the mode (surface), each axis will move in the workpiece coordinate system of the selected surface machining mode until it is cancelled.

Reference-point return cannot be programmed during the selected surface machining mode.

Remarks on G17.9 (Cancellation of surface machining mode)

- The surface machining mode is temporarily cancelled for a block containing G17.9. The axis movement command in such a block will be executed in reference to the machine coordinate system.
- The head offsetting amount is not cancelled by G17.9.
- The tool-length offsetting axis for a block containing G17.9 is the same as for the temporarily cancelled surface machining mode.
- The command G17.9 is not modal, and the surface machining mode temporarily cancelled will be revived in the next block.

21-3 Head Offsetting

21-3-1 Specification of head offsetting

1. Function and purpose

Specifying a head offset causes head offsetting based on the selected machining mode to remain effective until that command is cancelled.

2. Programming format

G45.1 Hh

h: Specify the head offset number, 1 to 16, to h.

The number corresponds to **OFFSET No.** on the **HEAD OFFSET** display. (For details on the display, see the related section of the Operating Manual.)

3. Detailed description

The head offset function is performed, based on **OFFSET X** through **OFFSET Z** of the head offset data. The direction of offsetting in that case depends on the selected machining mode (see the table below).

Head offsetting cannot be performed while the selected machining mode remains cancelled with G17.9.

Direction of offsetting according to the type of machining mode

Mode Offset	Top-surface	0-degree surface	90-degree surface	180-degree surface	270-degree surface
OFFSET X	+X	+X	−Y	−X	+Y
OFFSET Y	+Y	+Y	+X	−Y	−X
OFFSET Z	+Z	+Z	+Z	+Z	+Z

In the table, +X or −X means that offsetting is performed in the plus (+) or minus (−) direction of the X-axis in the machine coordinate system, respectively.

21-3-2 Cancellation of head offsetting

1. Function and purpose

Cancelling the head offsetting.

2. Programming format

G49.1

3. Remarks

Head offsetting cannot be performed while the selected machining mode remains cancelled with G17.9.

21-4 Relationship to Other Functions

The table below indicates the compatibility of each G-code with the five-surface machining function in the two columns on the right.

[1] Is the G-code available in the mode of five-surface machining?

[2] Is the five-surface machining mode selectable under the condition of the G-code?

G-code	Group	Function	[1]	[2]
00	01	Positioning	Yes	Yes
01	01	Linear interpolation	Yes	Yes
02	01	Circular interpolation CW	Yes	No
03	01	Circular interpolation CCW	Yes	No
02.1	00	Spiral interpolation CW	Yes	No
03.1	00	Spiral interpolation CCW	Yes	No
04	00	Dwell	Yes	
05	00	High-speed machining	No	No
06				
06.1	01	Spline interpolation	Yes	No
06.2	01	NURBS interpolation	Yes	No
07	00	Virtual-axis interpolation	No	No
08				
09	00	Exact-stop check	Yes	
10	00	Programmed parameter input	Yes	No
11	00	Programmed parameter input cancel	Yes	
12				
13				
14				
15				
16				
17	02	Plane selection X-Y	No	Yes
18	02	Plane selection Z-X	No	Yes
19	02	Plane selection Y-Z	No	Yes
20	06	Inch command	Yes	Yes
21	06	Metric command	Yes	Yes
22	04	Pre-move stroke check ON	No	No
23	04	Pre-move stroke check OFF	No	Yes
24				
25				
26				
27	00	Reference-point check	Yes	
28	00	Reference-point return	No	
29	00	Starting-point return	No	
30	00	No. 2 through 4 reference-point return	No	
31	00	Skip	Yes	
31.1	00	Multi-step skip 1	Yes	
31.2	00	Multi-step skip 2	Yes	
31.3	00	Multi-step skip 3	Yes	

G-code	Group	Function	[1]	[2]
32				
33	01	Threading	No	No
34				
35				
36				
37	00	Automatic tool-length measurement	No	
38	00	Tool-diameter offset vector selection	No	No
39	00	Tool-diameter offset corner arc	No	No
40	07	Tool-diameter offset cancel	Yes	Yes
40.1	15	Shaping cancel	No	No
41	07	Tool-diameter offset left	Yes	No
41.1	15	Shaping to the left	No	No
42	07	Tool-diameter offset right	Yes	No
42.1	15	Shaping to the right	No	No
43	08	Tool-length offset (+)	Yes	No
44	08	Tool-length offset (-)	Yes	No
45	00	Tool-position offset, extension	Yes	No
46	00	Tool-position offset, reduction	Yes	No
47	00	Tool-position offset, double extension	Yes	No
48	00	Tool-position offset, double reduction	Yes	No
49	08	Tool-length offset cancel	Yes	Yes
50	11	Scaling cancel	Yes	Yes
50.1	19	G-command mirror image cancel	Yes	Yes
51	11	Scaling on	Yes	No
51.1	19	G-command mirror image on	Yes	No
52	00	Local coordinate system setting	Yes	Yes
53	00	Machine coordinate system selection	Yes	
54	12	Workpiece coordinate system 1 selection	Yes	Yes
54.1	12	Additional workpiece coordinate system selection	Yes	Yes
55	12	Workpiece coordinate system 2 selection	Yes	Yes
56	12	Workpiece coordinate system 3 selection	Yes	Yes
57	12	Workpiece coordinate system 4 selection	Yes	Yes
58	12	Workpiece coordinate system 5 selection	Yes	Yes
59	12	Workpiece coordinate system 6 selection	Yes	Yes
60	00	One-way positioning	Yes	
61	13	Exact-stop check mode	Yes	No
61.1	13	Shape correction mode	Yes	No
61.2	13	Shape correction (Modal spline interpolation)	Yes	No
62	13	Automatic corner override	Yes	No
63	13	Tapping mode	Yes	No
64	13	Cutting mode	Yes	Yes
65	00	User macro simple call	Yes	
66	14	User macro modal call A	Yes	No
66.1	14	User macro modal call B	Yes	No
67	14	User macro modal call cancel	Yes	Yes
68	16	Program coordinates rotation	Yes	No
69	16	Cancellation of program coordinates rotation	Yes	Yes

G-code	Group	Function	[1]	[2]
70				
71.1	09	Fixed cycle (chamfering cutter 1)	Yes	No
72.1	09	Fixed cycle (chamfering cutter 2)	Yes	No
73	09	Fixed cycle (high-speed deep-hole drilling)	Yes	No
74	09	Fixed cycle (reverse tap)	Yes	No
75	09	Fixed cycle (boring)	Yes	No
76	09	Fixed cycle (boring)	Yes	No
77	09	Fixed cycle (Back facing)	Yes	No
78	09	Fixed cycle (boring)	Yes	No
79	09	Fixed cycle (boring)	Yes	No
80	09	Fixed cycle cancel	Yes	Yes
81	09	Fixed cycle (drill/spot drill)	Yes	No
82	09	Fixed cycle (drill)	Yes	No
83	09	Fixed cycle (deep hole drill)	Yes	No
84	09	Fixed cycle (tapping)	Yes	No
85	09	Fixed cycle (reaming)	Yes	No
86	09	Fixed cycle (boring)	Yes	No
87	09	Fixed cycle (back boring)	Yes	No
88	09	Fixed cycle (boring)	Yes	No
89	09	Fixed cycle (boring)	Yes	No
90	03	Absolute data input	Yes	Yes
91	03	Incremental data input	Yes	Yes
92	00	Machine coordinate system setting	Yes	
93				
94	05	Asynchronous feed (feed per minute)	Yes	Yes
95	05	Synchronous feed (feed per revolution)	Yes	Yes
96				
97				
98	10	Initial level return in fixed cycle	Yes	Yes
99	10	R-point level return in fixed cycle	Yes	Yes

22 ARBITRARY-SURFACE MACHINING FUNCTION (OPTION)

22-1 Three-Dimensional Coordinate Conversion: G68

1. Function and purpose

The three-dimensional (3D) coordinate conversion mode makes it possible for a new coordinate system to be defined by rotating the X-axis, Y-axis, or Z-axis of the currently valid workpiece coordinate system and moving the workpiece origin in parallel to that axis. Thus, definition of any plane on a space allows creation of a program assuming that the plane is an X-Y plane.

2. Programming format

G68 [Xx0 Yy0 Zz0] Ii Jj Kk Rr 3D coordinate conversion mode on

G69 3D coordinate conversion mode off

Where x_0, y_0, z_0 : Rotational center coordinates (Absolute)

i, j, k: Central axis of rotation (1: Valid, 0: Invalid)

I: X-axis

j: Y-axis

k: Z-axis

r: Angle and direction of rotation of the rotational center axis (the counter clockwise rotational direction when the center of rotation is seen from the positive side of the rotational axis, is taken as a plus direction)

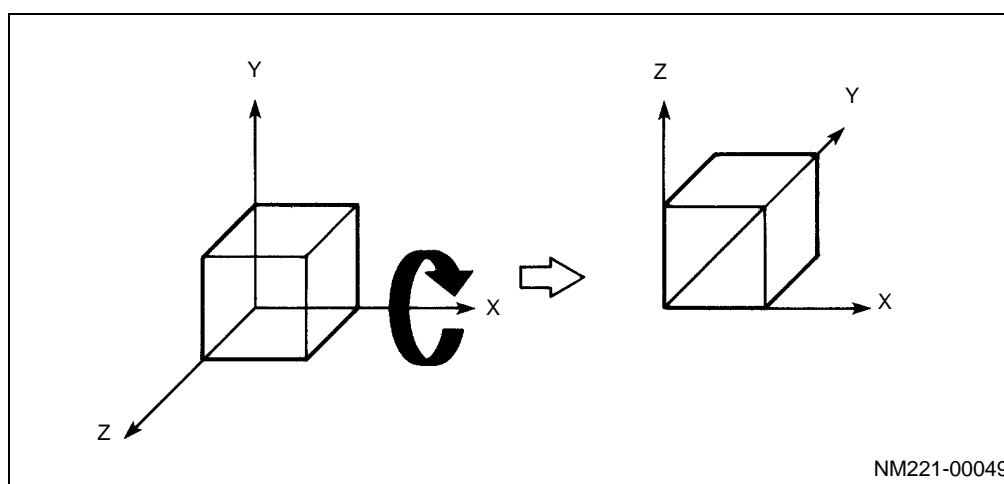
Example:

G68X0Y0Z0 I1J0K0 R-90.

Rotates the axis through 90 degrees in the clockwise direction.

Sets the X-axis as the central axis of rotation.

Sets the workpiece origin as the center of rotation.



<Precautions>

- If X, Y, or Z is omitted, the workpiece origin in the currently valid workpiece coordinate system will become rotational center coordinates.
- The characters I, J, and K must all be set. Omission of even one of these three characters results in alarm **807 ILLEGAL FORMAT**. Omission of all the three characters makes program coordinate rotation valid, instead of 3D coordinate conversion.

G68X0Y0Z0I1K0R-45. Format error

G68X0Y0Z0R-45. Program coordinate rotation ON

- Setting of 0 in all the arguments of I, J, and K also results in alarm **807 ILLEGAL FORMAT**.
- Setting of 1 in the arguments of two or more of the three characters (I, J, and K) also results in alarm **807 ILLEGAL FORMAT**.

G68X0Y0Z0I1J1K0R-90. Format error

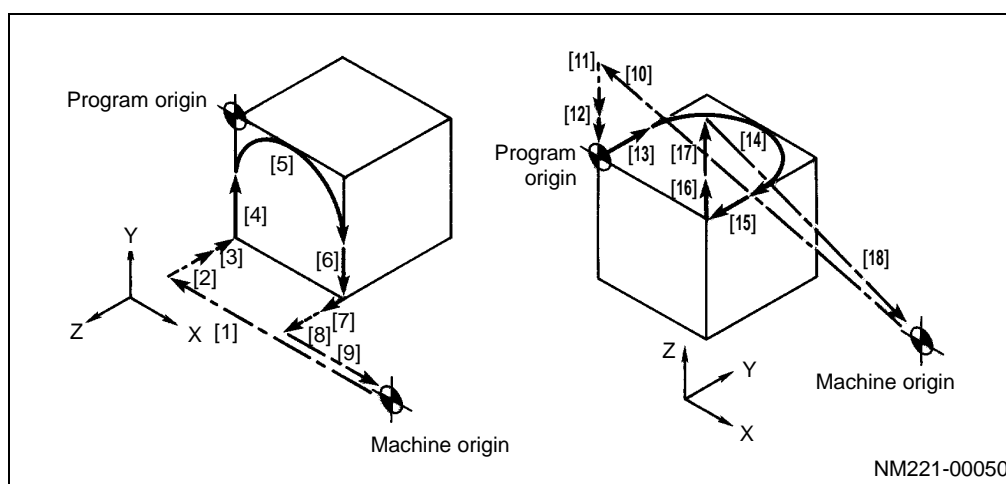
- Setting of command G68 in the absence of a 3D coordinate conversion option results in alarm **942 NO 3-D CONVERSION OPTION**.

- Setting of an illegal G-code during the 3D coordinate conversion mode results in alarm **943 CONVERTING IN 3-D CORDINATES**.

See Paragraph 5. Relationship to other functions, for the listing of illegal G-codes.

3. Sample program

N01	G90G00G40G49G80		G68I1J0K0R-90.	
	G54X0Y-100. [1]	G00X0Y0 [10]
	G43Z50.H01 [2]	G43Z50.H01 [11]
	G01Z-1.F1200 [3]	G01Z-1. [12]
	Y-50. [4]	Y50. [13]
	G02X100.R50. [5]	G02X100.R50. [14]
	G01Y-100. [6]	G01Y0 [15]
	G01Z50. [7]	G01Z50. [16]
	G91G28Z0 [8]	G69	
	G28X0Y0 [9]	G91G28Z0 [17]
N02	G90G00G40G49G80		G28X0Y0 [18]
	G54G00A90.		M30	

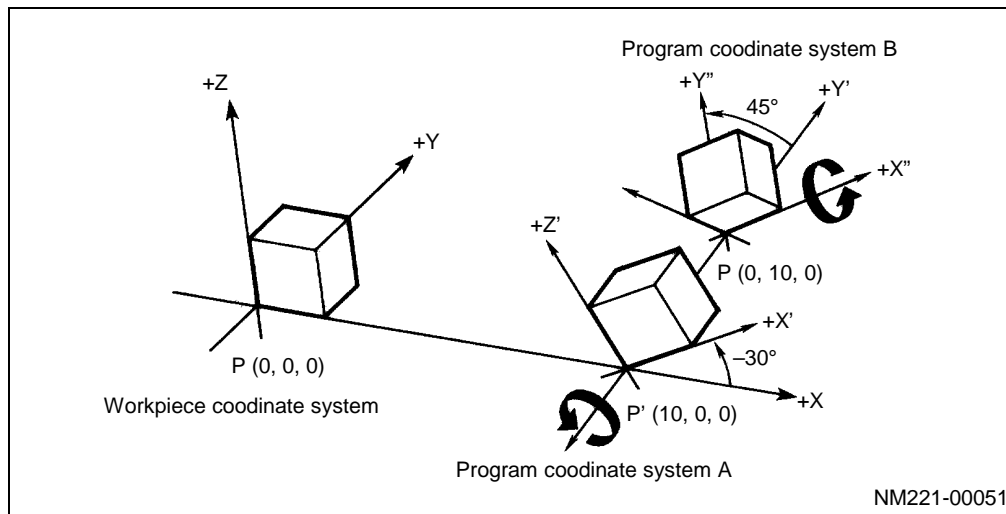


4. 3D coordinate conversion and a program coordinate system

Use of command G68 allows the coordinates of the rotational center in the currently valid coordinate system to be shifted to the designated position (X, Y, Z) and the corresponding graphics to be rotated through the designated angle of rotation (R) in the designated direction of rotational center (I, J, K). Thus, a new program coordinate system can be set.

Example: [1] G90 G54 X0 Y0
 [2] G68 X10.Y0 Z0 I0 J1 K0 R-30.
 [3] G68 X0 Y10.Z0 I1 J0 K0 R45.
 [4] G69

- [1] Set a workpiece coordinate system using workpiece offset command G54.
- [2] The program origin in the workpiece coordinate system that was set during step [1] above shifts to the position of (x, y, z) = (10, 0, 0) and the graphics rotates through 30 degrees in a clockwise direction around the Y-axis. Thus, new program coordinate system A is set.
- [3] The program origin in the workpiece coordinate system that was set during step [2] above shifts to the position of (x, y, z) = (0, 10, 0) and the graphics rotates through 45 degrees in a counter clockwise direction around the X-axis. Thus, new program coordinate system B is set.
- [4] Both the program coordinate systems that were set during steps [2] and [3] above are cancelled and the coordinate system that was set during step [1] above becomes valid once again.



5. Relationship to other functions

1. The table below indicates the compatibility of each G-code with the 3D coordinate conversion function in the two columns on the right.

[1] Is the G-code available in the mode of 3D coordinate conversion?

[2] Is the 3D coordinate conversion mode selectable under the condition of the G-code?

G-code	Group	Function	[1]	[2]
00	01	Positioning	Yes	Yes
01	01	Linear interpolation	Yes	Yes
02	01	Circular interpolation CW	Yes (*1)	No
03	01	Circular interpolation CCW	Yes (*1)	No
02.1	00	Spiral interpolation CW	No	No
03.1	00	Spiral interpolation CCW	No	No
04	00	Dwell	Yes (*5)	
05	00	High-speed machining	No	No
06				
06.1	01	Spline interpolation	No	No
06.2	01	NURBS interpolation	No	No
07	00	Virtual-axis interpolation	No	No
08				
09	00	Exact-stop check	Yes	
10	00	Programmed parameter input	Yes (*5)	No
11	00	Programmed parameter input cancel	Yes (*5)	
12				
13				
14				
15				
16				
17	02	Plane selection X-Y	Yes	Yes
18	02	Plane selection Z-X	Yes	Yes
19	02	Plane selection Y-Z	Yes	Yes
20	06	Inch command	No	Yes
21	06	Metric command	No	Yes
22	04	Pre-move stroke check ON	No	No
23	04	Pre-move stroke check OFF	No	Yes
24				
25				
26				
27	00	Reference-point check	No	
28	00	Reference-point return	Yes (*2)	
29	00	Starting-point return	Yes	
30	00	No. 2 through 4 reference-point return	Yes (*2)	
31	00	Skip	No (*4)	
31.1	00	Multi-step skip 1	No	
31.2	00	Multi-step skip 2	No	
31.3	00	Multi-step skip 3	No	
32				

G-code	Group	Function	[1]	[2]
33	01	Threading	No	No
34				
35				
36				
37	00	Automatic tool-length measurement	No	
38	00	Tool-diameter offset vector selection	No	No
39	00	Tool-diameter offset corner arc	No	No
40	07	Tool-diameter offset cancel	Yes	Yes
40.1	15	Shaping cancel	No	No
41	07	Tool-diameter offset left	Yes	No
41.1	15	Shapint to the left	No	No
42	07	Tool-diameter offset right	Yes	No
42.1	15	Shaping to the right	No	No
43	08	Tool-length offset (+)	Yes	No
44	08	Tool-length offset (-)	Yes	No
45	00	Tool-position offset, extension	Yes	No
46	00	Tool-position offset, reduction	Yes	No
47	00	Tool-position offset, double extension	Yes	No
48	00	Tool-positon offset, double reduction	Yes	No
49	08	Tool-length offset cancel	Yes	Yes
50	11	Scaling cancel	No	Yes
50.1	19	G-command mirror image cancel	Yes	Yes
51	11	Scaling on	No	No
51.1	19	G-command mirror image on	Yes	No
52	00	Local coordinate system setting	Yes	Yes
53	00	Machine coordinate system selection	Yes (*6)	
54	12	Workpiece coordinate system 1 selection	No	Yes
54.1	12	Additional workpiece coordinate system selection	No	Yes
55	12	Workpiece coordinate system 2 selection	No	Yes
56	12	Workpiece coordinate system 3 selection	No	Yes
57	12	Workpiece coordinate system 4 selection	No	Yes
58	12	Workpiece coordinate system 5 selection	No	Yes
59	12	Workpiece coordinate system 6 selection	No	Yes
60	00	One-way positioning	No	
61	13	Exact-stop check mode	No	No
61.1	13	Shape correction mode	No	No
61.2	13	Shape correction (Modal spline interpolation)	No	No
62	13	Automatic corner override	No	No
63	13	Tapping mode	No	No
64	13	Cutting mode	No	Yes
65	00	User macro simple call	Yes (*5)	
66	14	User macro modal call A	Yes (*5)	No
66.1	14	User macro modal call B	Yes (*5)	No
67	14	User macro modal call cancel	Yes (*5)	Yes
68	16	Program coordinates rotation	Yes (*3)	No
69	16	Cancellation of program coordinates rotation	Yes (*3)	Yes
70				

G-code	Group	Function	[1]	[2]
71.1	09	Fixed cycle (chamfering cutter 1)	Yes	No
72.1	09	Fixed cycle (chamfering cutter 2)	Yes	No
73	09	Fixed cycle (high-speed deep-hole drilling)	Yes	No
74	09	Fixed cycle (reverse tap)	Yes	No
75	09	Fixed cycle (boring)	Yes	No
76	09	Fixed cycle (boring)	Yes	No
77	09	Fixed cycle (Back facing)	Yes	No
78	09	Fixed cycle (boring)	Yes	No
79	09	Fixed cycle (boring)	Yes	No
80	09	Fixed cycle cancel	Yes	Yes
81	09	Fixed cycle (drill/spot drill)	Yes	No
82	09	Fixed cycle (drill)	Yes	No
83	09	Fixed cycle (deep hole drill)	Yes	No
84	09	Fixed cycle (tapping)	Yes	No
85	09	Fixed cycle (reaming)	Yes	No
86	09	Fixed cycle (boring)	Yes	No
87	09	Fixed cycle (back boring)	Yes	No
88	09	Fixed cycle (boring)	Yes	No
89	09	Fixed cycle (boring)	Yes	No
90	03	Absolute data input	Yes	Yes
91	03	Incremental data input	Yes	Yes
92	00	Machine coordinate system setting	No	
93				
94	05	Asynchronous feed (feed per minute)	Yes (*5)	Yes
95	05	Synchronous feed (feed per revolution)	Yes (*5)	Yes
96				
97				
98	10	Initial level return in fixed cycle	Yes	Yes
99	10	R-point level return in fixed cycle	Yes	Yes

*1: Setting of helical interpolation results in an alarm.

*2: Only the intermediate point has its coordinates converted.

*3: Setting of program coordinate rotation results in an alarm.

*4: To use G31 for measurement purposes, G68 (3D coordinate conversion mode) must be cancelled.

*5: The preparatory function is executed independently of the coordinate conversion.

*6: The operation is always performed in the original coordinate system (free from any conversion).

- Setting of a G-code other than G17, G18, and G19, in the block containing G68 or G69 results in alarm **807 ILLEGAL FORMAT**.
- G28 and G30 commands permit a return to the reference point through the 3D converted intermediate point, whereas a G53 command is performed without any conversion.
- Codes G41 (Tool diameter offset left), G42 (Tool diameter offset right), G51.1 (G-command mirror image on), and Fixed cycle codes must be present with their cancellation codes between G68 and G69 so that a nested relationship is established. Moreover, give a motion command with G90 (absolute data input) in the next block to that of the G68 command.

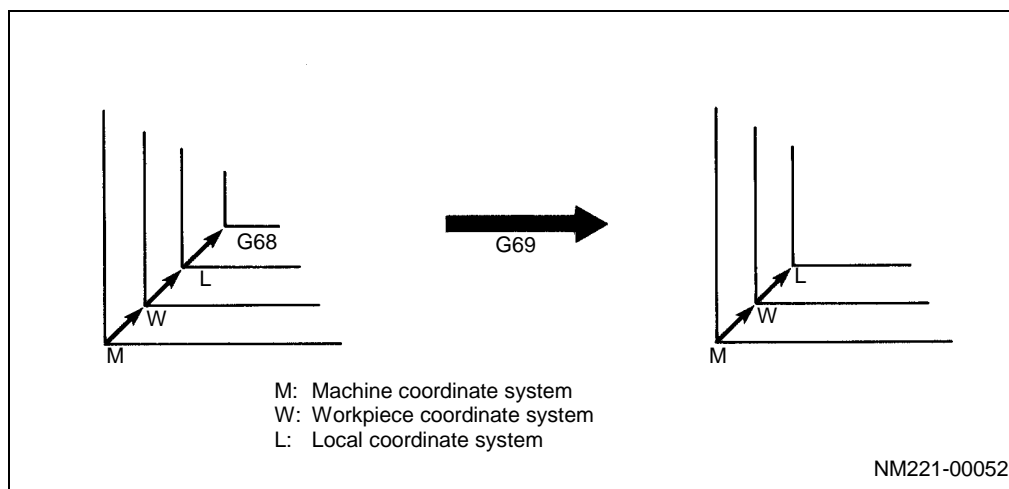
Example:

```

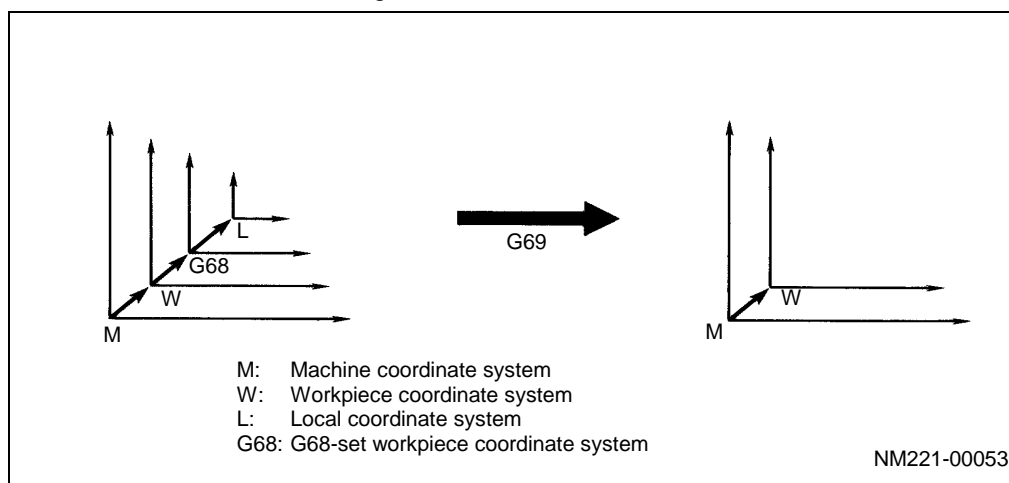
G68X50.Y100.Z150.I1J0K0R-90.
G90G00X0Y0Z0
G41G01X10.F1000
  ⋮
  ⋮
G40
G69

```

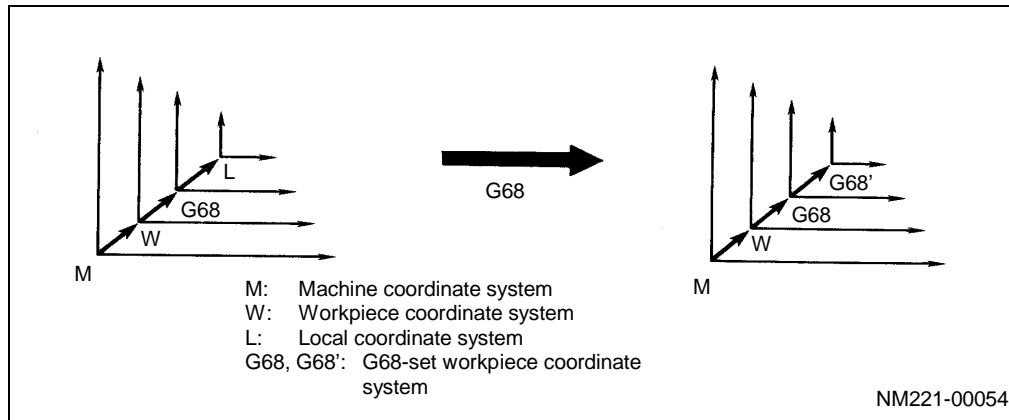
5. Tool-length, -diameter and -position offset is first processed, and the 3D conversion is conducted on the offset contour.
6. Mirror image can only be applied by G51.1 (the related M-codes are not available). The processing order is: mirror image first, then 3D conversion.
7. In mode G68, code G52 is handled as follows:
 - If code G68 is set in the coordinates that have been set using G52, that local coordinate system will be overridden with a new workpiece coordinate system by G68. Setting of G69 under that status will make the original local coordinate system valid once again.



- If code G52 is present in the data set using mode G68, a local coordinate system can be set on the workpiece coordinate system that has been set using G68. Setting of G69 under that status will cancel the local coordinate system and the G68-set workpiece coordinate system and thus the original coordinate system existing before G68 was set will become valid once again.



- If code G52 is present in the data set using mode G68, a local coordinate system can be set that starts from the workpiece coordinate system to be set using G68. Setting of G68 under that status will cancel the local coordinate system and then set a G68 program coordinate system.



- Setting of G68 during figure rotation results in alarm **850 G68 AND M98 COMMANDS SAME BLOCK**.
- Three-dimensional coordinate conversion is not valid for the axis of rotation.
- The G68 mode does not accept any external workpiece origin setting commands.

22-2 Simultaneous-Operation M-Codes

To machine the inclined and top surfaces of workpiece using an HV machining center, the positions of the α -axis (head) and the B-axis (table) must be programmed to ensure that the angle of a virtual A-axis is formed by rotation on the former two axes.

Simultaneous positioning on both the α -axis and the B-axis can be performed just by setting the desired angle data for the virtual A-axis using the following simultaneous-operation M-codes:

- M175 : Virtual A-axis command
Only an α -axis positioning by a virtual A-axis command.
- M176 : Virtual A-axis command for simultaneous B-axis operation
Simultaneous α -axis and B-axis positioning by a virtual A-axis command.
- M177 : α -axis direct command
Independent α -axis positioning to the angle specified at address A.

M175 mode is automatically set when power is turned on.

Example: N01 G90G00G40G49G80

N02 ← See items [1] to [3] listed in the table below.

N03 G90G00A45.

M-code to be set	α -axis (machine position)	B-axis (machine position)
[1] M175	114.47	0
[2] M176	114.47	-65.53
[3] M177	45.	0

Note 1: For the restart operation, these M-codes cannot be set automatically. Before executing restart operation, the required M-code must be set by MDI interruption.

Note 2: When the simultaneous-operation M-codes are to be used, set A-axis data as follows:

M-code to be set	Setting range (degrees)
M175	$-90^\circ \leq A \leq 90^\circ$
M176	$-90^\circ \leq A \leq 90^\circ$
M177	$-180^\circ \leq A \leq 180^\circ$

Note 3: The α -axis position is indicated at **A** in the display on the NC operating panel.

22-3 Table Rotational Machining (Option)

1. Function and purpose

To machine the top surface of workpiece using an HV machining center, it has formerly been necessary to divide the process into two sub-processes because of the limited machining range in the Z-axial direction. Machining with one process, however, becomes possible by rotating the table (B-axis) if the machining range in the Z-axial direction is overstepped.

Two types of table rotational machining are available: B-X interpolation, in which only the X-axis and the B-axis are to be actuated with the Z-axis remaining fixed, and B-Z interpolation, in which only the Z-axis and the B-axis are to be actuated with the X-axis remaining fixed.

2. Programming format

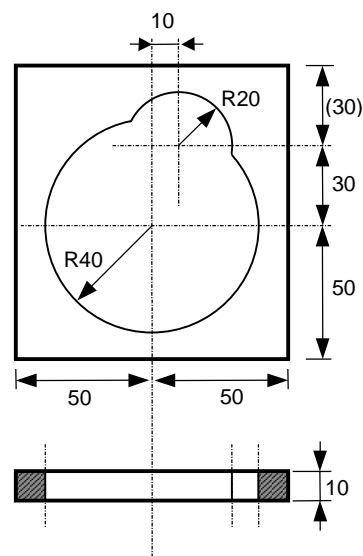
M144: Cancellation of table rotational machining

M145: Table rotational machining mode I (B-X interpolation)

M146: Table rotational machining mode II (B-Z interpolation)

3. Sample program

```
G90G00G40G49G80
G91G28Z0
G28X0Y0
G54S320M03
M175
G90G00A90.
G68I1J0K0R-90.
G00X0Y0
G43Z50.H01
M145 _____Table rotational machining mode I ON
G00Z-10.
X0Y-35
G01Y-40.F40
G03X29.735Y26.755R40.
G03X-7.735Y39.245R20.
G03X0Y-40.R40.
G01Y-35.
G00X0Y0
Z50.
M144 _____Cancellation of table rotational
      machining
G69
G91G28Z0
G28X0Y0
G28A0B0
M30
```



4. Detailed description

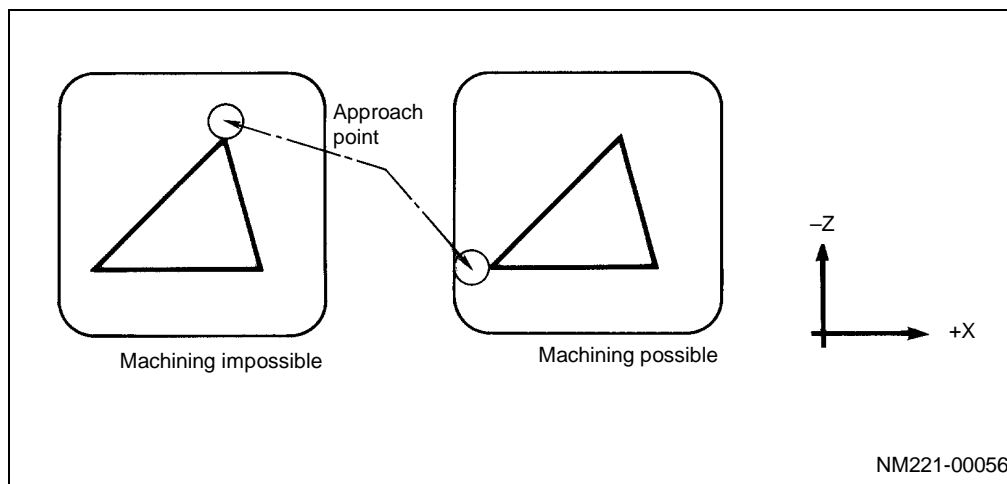
1. Table rotational machining is valid only for top-surface machining (α -axis = 180°).
2. Use parameter **S7** to set the allowable machining area. (Only for table rotational machining mode)
3. The desired method of table rotational machining can be selected using bit 0 of parameter **F85**. (Only for table rotational machining mode I)

F85 bit 0 = 1: Machining based on conventional X-Z interpolation occurs until the machining area has been overstepped, and then, rotational machining follows.

= 0: Only rotational machining occurs.

5. Precautions

1. Table rotational machining (M145, M146) should not be designated for inclined or the side surfaces as it does not occur correctly for them.
2. Inappropriate setting of an approach point may not allow table rotational machining if the selected method of machining is only-rotational machining (**F85** bit 0 = 0).



3. An incremental B-axis command during table rotational machining will perform a rotation on the B-axis through the designated distance from the current position.
4. Correctly designate the starting position for restart operation from a section of table rotational machining (refer to the following example).

```

      ⋮
N10  M145
N11  G01X100.Y100.F1000
      ⋮
      ⋮
      ⋮
  
```

Restart from N10:

Table rotational machining is executed.

Restart from N11:

Table rotational machining is not executed.

22-4 Workpiece Coordinate Conversion

1. Function and purpose

Machining for an inclined surface requires a preparatory indexing (angular positioning) of the spindle head (α -axis) and the machine table (B-axis). The position of the offset workpiece origin, which is indeed difficult to be measured after the preparatory indexing, can now be internally calculated from both the corresponding data of a measurement before indexing and the angular position required for the particular machining.

2. Programming format

Gg₀ Dd₀

g₀: Number of the macro-call G-code (set in the related machine parameter)

d₀: Code (4 to 9) for the workpiece origin data

4: G54 7: G57

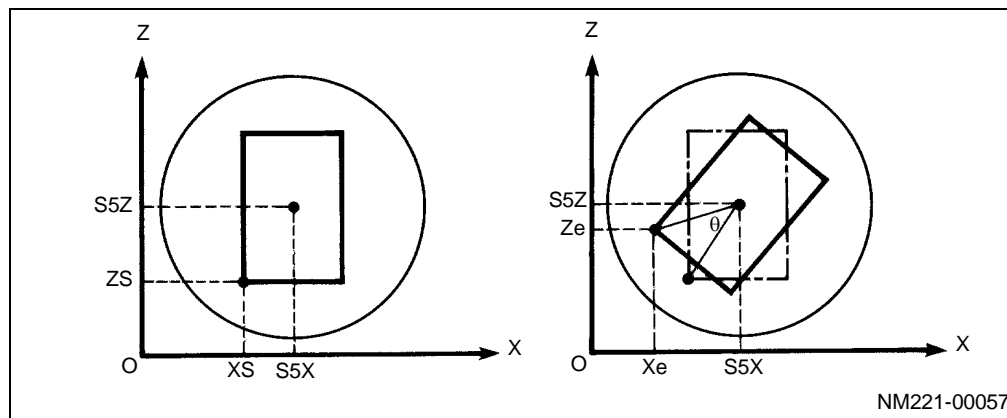
5: G55 8: G58

6: G56 9: G59

(If d₀ is omitted or assigned with another numeral than 4 to 9, conversions will be performed for the currently valid workpiece origin data.)

3. Detailed description

The macroprogram concerned updates the coordinates of the workpiece origin specified with argument D according to the actual angular position (on the B-axis) of the machine table.



S5X, S5Z : Coordinates of the center of table's rotation (set in parameter **S5**)

XS, ZS : Measured coordinates of workpiece origin

Xe, Ze : Workpiece origin after conversion

θ : Angle of table's rotation

4. Precautions

- Preparatory measurement of workpiece origin should be performed with only the table set in the required angular position (without particular indexing of the head; i. e. $\alpha = 0$ degrees).
- Workpiece coordinate conversion must be programmed after a positioning of the head (α -axis) and the table (B-axis) by an A-axis command given with the table positioned at 0° .
- The G-code specified with argument D will function as a modal one upon the macro call.
- After workpiece coordinate conversion, set the table to the angular position corresponding to the offset origin.

- Converted workpiece origin data are cancelled by an explicit command of G54 to G59.
- This function cannot be used for machines equipped with an indexing table.

5. Related parameters

Address	Description	Validation condition	Value to be set
J1	Work No. of the program to be called up	Power off and on	100009300
J2	G-code No. to be used for call	Power off and on	Any value from 0 to 255
J3	Call type	Power off and on	1
S5	Coordinates of the table center	Power off and on	Machine-dependent

Note 1: G-codes, such as G00, G01, G02, etc., that have their uses already clearly defined under the EIA Standards cannot be used for macro call.

Note 2: Parameters **J1** to **J3** must have been set for executing this function.

6. Sample program

The example given below refers to a machining for an inclined surface of 45° with the table positioned at 90° on the B-axis.

N01 G90G00G40G49G80

G91G30Y0Z0

T01T00M06

N02 G90G00B0.

N03 G54S320M03

N04 M176

G00A45.

N05 G168D4

G68X0Y0Z0I0J1K0R-[#5024]

G68X0Y0Z0I1J0K0R-45.

N06 G91B90.

N07 G90

G00X-10.Y-15.

Description (with reference to the figures below)

Fig. 1

Set beforehand under G54 on the **WORK OFFSET** display the coordinates (XS, ZS) of the origin obtained by a measurement with the table positioned at B=90° as required for machining.

Fig. 2

N02: Command for setting the table to B=0°.

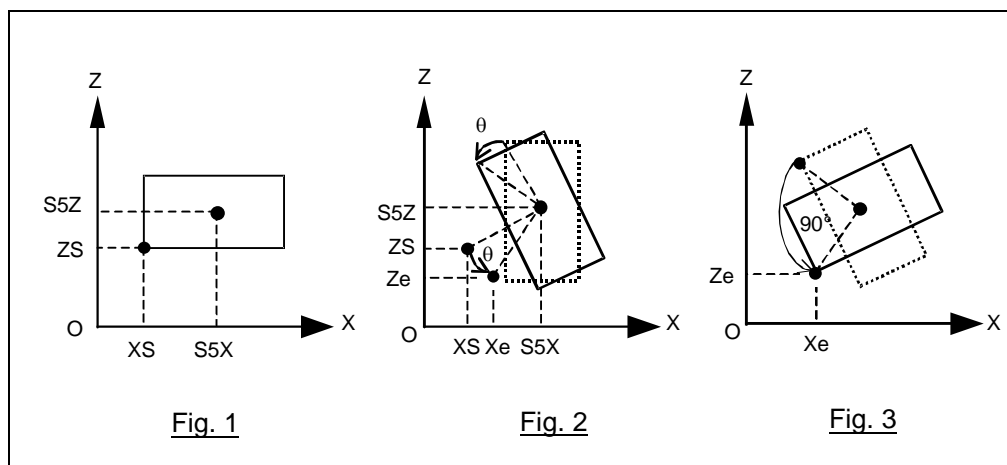
N04: A-axis command for positioning on the α - and B-axes.

N05: Command for converting the origin from (XS, ZS) to (Xe, Ze).

* "168" refers to the setting in parameter **J2**.

Fig. 3

N06: Command for rotating the table by 90° for a conformity to the offset origin (Xe, Ze).



22-5 Inclined-Surface Synchronous Tapping

As with the side of a workpiece, inclined and top surfaces can be subjected to synchronous tapping, but only during mode of G68 (3D coordinate conversion).

<Sample program>

```

N01  G90G00G40G49G80
      G91G30Y0Z0
      T01T00M06
      M176
      G90G00A-45.
      G54S390M03
N02  G68X0Y0Z0I0J1K0R-65.53
N03  G68X0Y0Z0I1J0K0R-45.
N04  G00X100.Y100.
      G43Z50.H01
N05  G84Z-12.R5.F1.25H100
      G80G00Z50.
      G91G28Z0
      G28X0Y0
      G28A0B0
      M30

```

N02: Sets a coordinate system rotated clockwise through 65.53 degrees around the Y-axis.

N03: Sets a coordinate system rotated clockwise through 45 degrees around the X-axis.

N04: Positioning on the X-axis and the Y-axis.

N05: Executes a synchronous tapping cycle.

<Precautions>

1. Set parameters as follows to perform synchronous tapping operations for the inclined and top surfaces of workpiece:
 - Set the tapping-position loop gain level to the same value for all three axes (X, Y and Z).
 - Set the acceleration/deceleration mode and the time constant for rapid feed to the same values for all three axes (X, Y and Z).
 - Set the acceleration/deceleration mode and the time constant for cutting feed to the same values for all three axes (X, Y and Z).
2. Tap return overriding and other related machine actions are the same as for the conventional synchronous tapping cycle.
3. The programming format is the same as for the side surface.

22-6 Inclined-Surface Boring

As with the side of a workpiece, inclined and top planes can be subjected to direction-of-escape control of the tool nose during boring. The direction of escape of the tool nose, however, can be converted during mode G68 (3D coordinate conversion).

<Sample program>

```

N01  G90G00G40G49G80
      G91G30Y0Z0
      T01T00M06
      M176
      G90G00A-45.
      G54S390M03
N02  G68X0Y0Z0I0J1K0R-65.53
N03  G68X0Y0Z0I1J0K0R-45.
N04  G00X100.Y100.
      G43Z50.H01
N05  G76Z-20.R5.Q0.5F30
      G80G00Z50.
      G91G28Z0
      G28X0Y0
      G28A0B0
      M30

```

N02: Sets a coordinate system rotated clockwise through 65.53 degrees around the Y-axis.

N03: Sets a coordinate system rotated clockwise through 45 degrees around the X-axis.

N04: Positioning on the X-axis and the Y-axis.

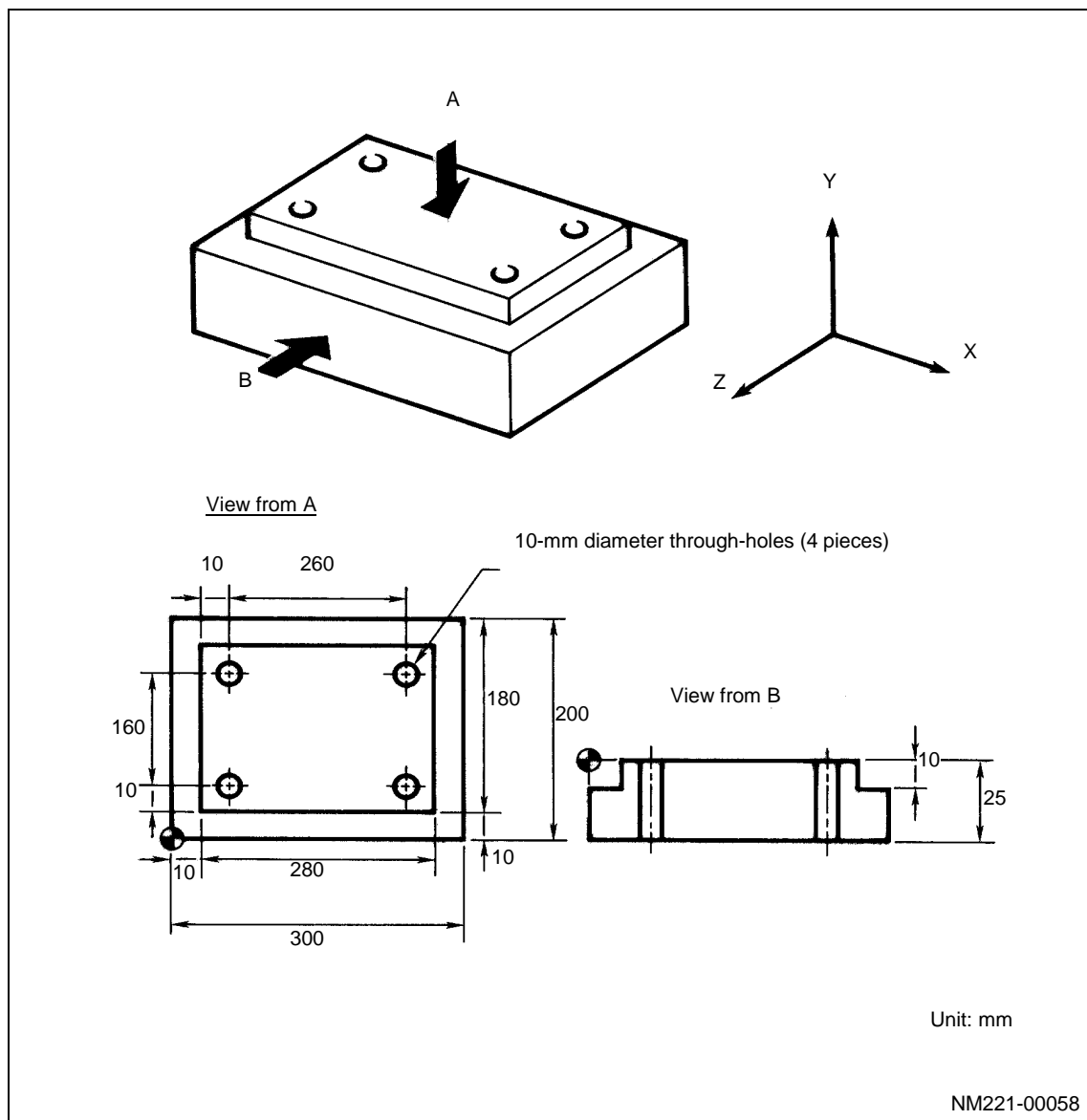
N05: Executes a boring cycle.

<Precautions>

1. Use parameter **I14** to set whether and in which direction the tool nose is to be made to escape during boring. Use argument Q during fixed-cycle call to set how far the tool nose is to be made to escape. Refer to the Operating Manual for further details on the direction of escape.
2. The programming format is the same as for the side surface.

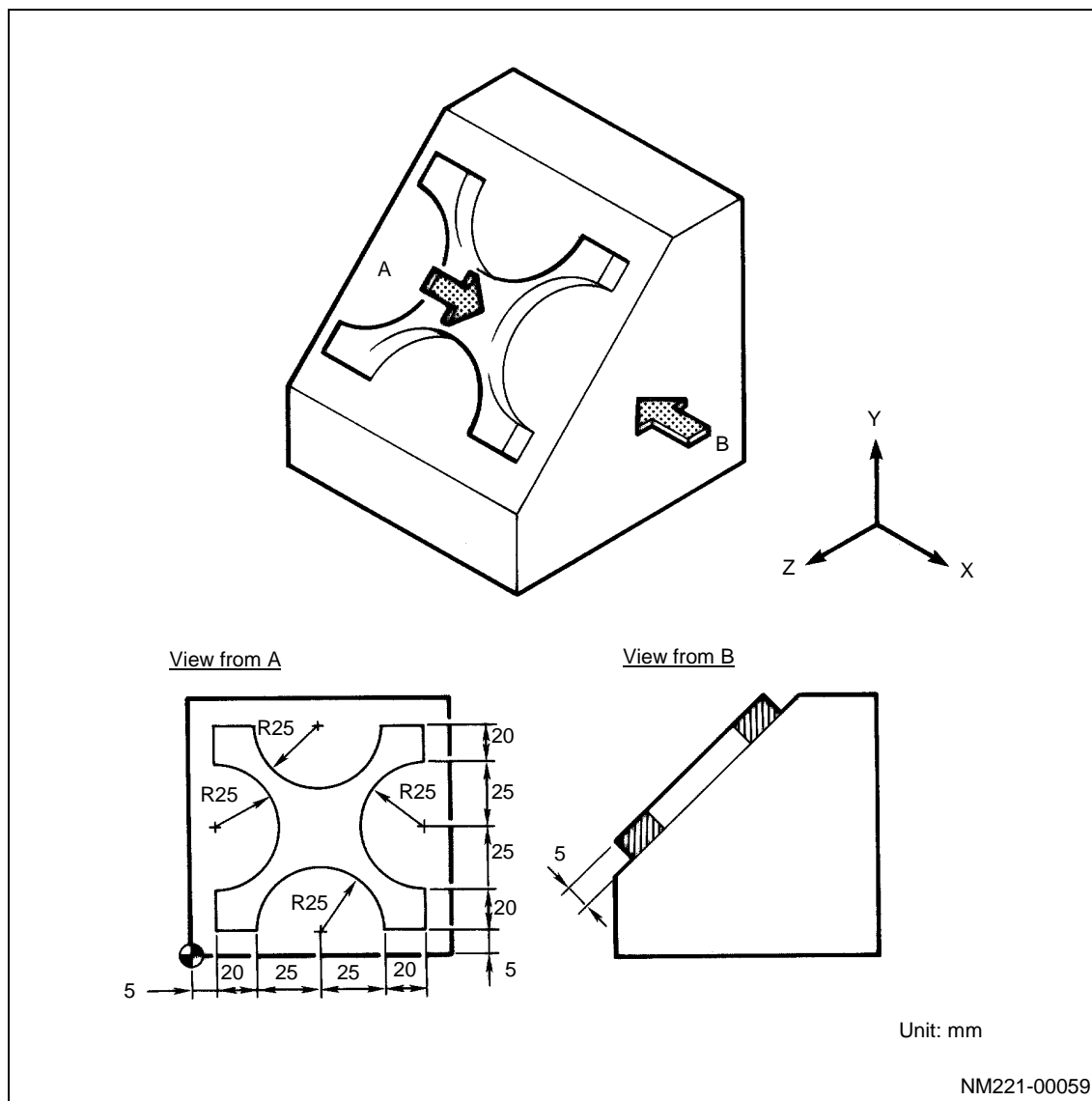
22-7 Sample Program

1. Top-surface machining (Facial angle: 90 degrees)



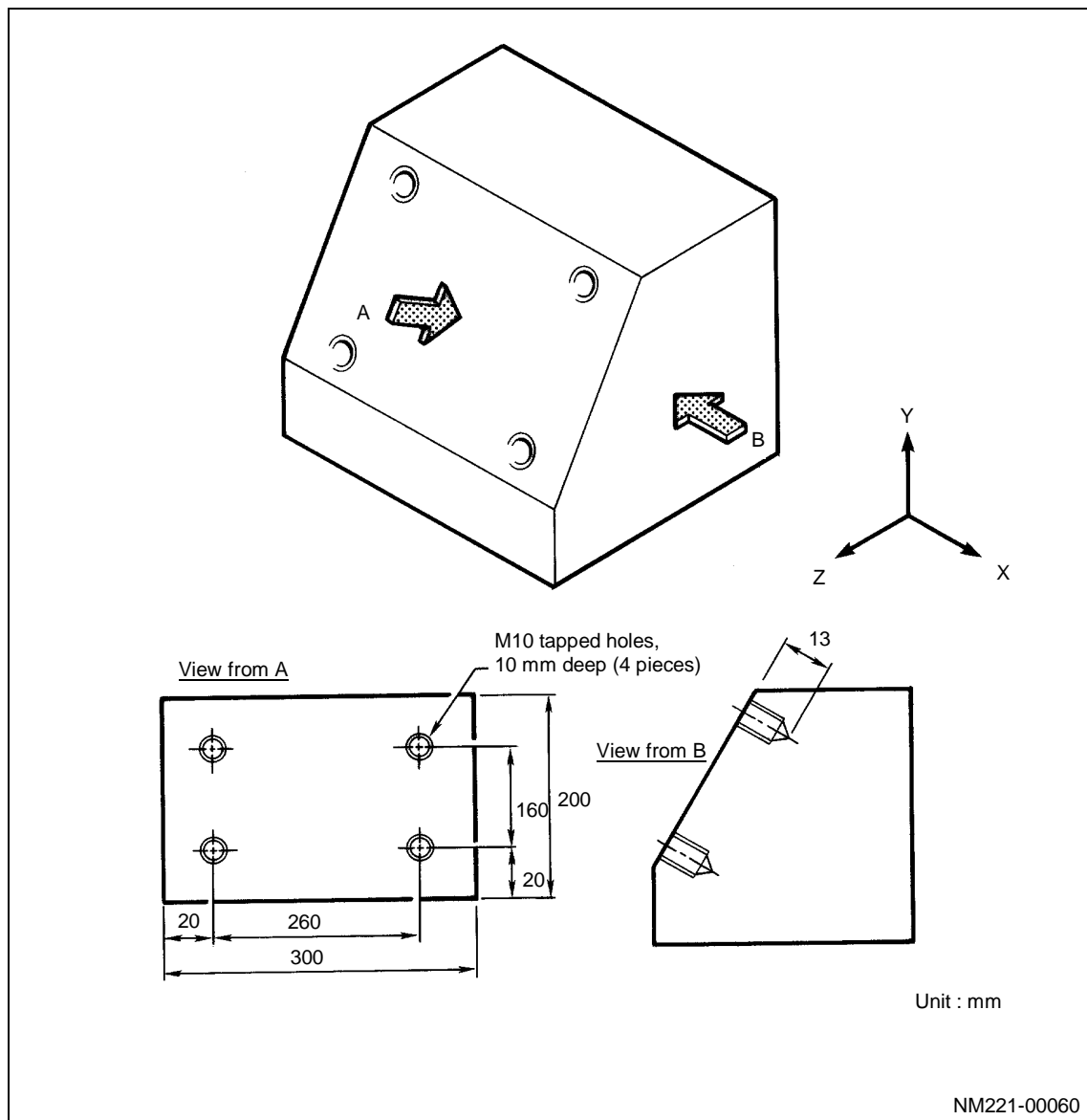
<p>N01 (F-MIL D200)</p> <p>G90G00G40G49G80</p> <p>G91G30Y0Z0</p> <p>T01T00M06</p> <p>G90G54S225M03</p> <p>M175</p> <p>G00A90.</p> <p>G68I1J0K0R-90.</p> <p>G00X-110.Y160.</p> <p>G43Z50.H01M08</p> <p>G00Z0</p> <p>G01X410.F160</p> <p>G00Z50.</p> <p>X-110.Y40.</p> <p>Z0</p> <p>G01X410.</p> <p>G00Z50.M09</p> <p>G69</p> <p>G91G28Z0</p> <p>G91G28Z0Y0</p> <p>M01</p>	<p>N03 (SPOT D16)</p> <p>G90G00G40G49G80</p> <p>G91G30Y0Z0</p> <p>T03T00M06</p> <p>G90G54S157M03</p> <p>M175</p> <p>G00A90.</p> <p>G68I1J0K0R-90.</p> <p>G00X20.Y20.</p> <p>G43Z50.H03M08</p> <p>G99G82R5.Z-6.F127</p> <p>M98H06</p> <p>G80G00Z50.M09</p> <p>G69</p> <p>G91G28Z0</p> <p>G91G28X0Y0</p> <p>M01</p>
<p>N02 (E-MIL D20)</p> <p>G90G00G40G49G80</p> <p>G91G30Y0Z0</p> <p>T02T00M06</p> <p>G90G54S640M03</p> <p>M175</p> <p>G00A90.</p> <p>G68I1J0K0R-90.</p> <p>G00X10.Y-20.</p> <p>G43Z50.H02M08</p> <p>G00Z-10.</p> <p>G41G01Y0.F1100D12</p> <p>Y190.</p> <p>X290.</p> <p>Y10.</p> <p>X-20.</p> <p>G40G00Z50.M09</p> <p>G69</p> <p>G91G28Z0</p> <p>G91G28X0Y0</p> <p>M01</p>	<p>N04 (DRL D10)</p> <p>G90G00G40G49G80</p> <p>G91G30Y0Z0</p> <p>T04T00M06</p> <p>G90G54S1270M03</p> <p>M175</p> <p>G00A90.</p> <p>G68I1J0K0R-90.</p> <p>G00X20.Y20.</p> <p>G43Z50.H03M08</p> <p>G99G82R5.Z-28.F470</p> <p>M98H06</p> <p>G80G00Z50.M09</p> <p>G69</p> <p>G91G28Z0</p> <p>G91G28X0Y0</p> <p>M01</p>
	<p>N05 G91G28A0B0</p> <p>M30</p>
	<p>N06 (SUB PRO)</p> <p>X280.</p> <p>Y180.</p> <p>X20.</p> <p>M99</p>

2. Inclined-surface machining (Facial angle: 45 degrees)



N01 (E-MIL D20)	X25.
G90G00G40G49G80	G03X75.R25.
G91G30Y0Z0	G01X95.
T01T00M06	Y75.
G90G54S320M03	G03Y25.R25.
M176	G01Y5.
G00A45.	X75.
G68I0J1K0R-[#5024]	G03X25.R25.
G68I1J0K0R-45.	G01X-10.
G00X-10.Y-15.	G40G00Z50.M09
G43Z50.H01M08	G69
Z-5.	G91G28Z0
G41G01X0Y-5D11F64	G28X0Y0
Y25.	G28A0B0
G03Y75.R25.	M30
G01Y95.	

#5024 : Fourth-axis machine coordinate

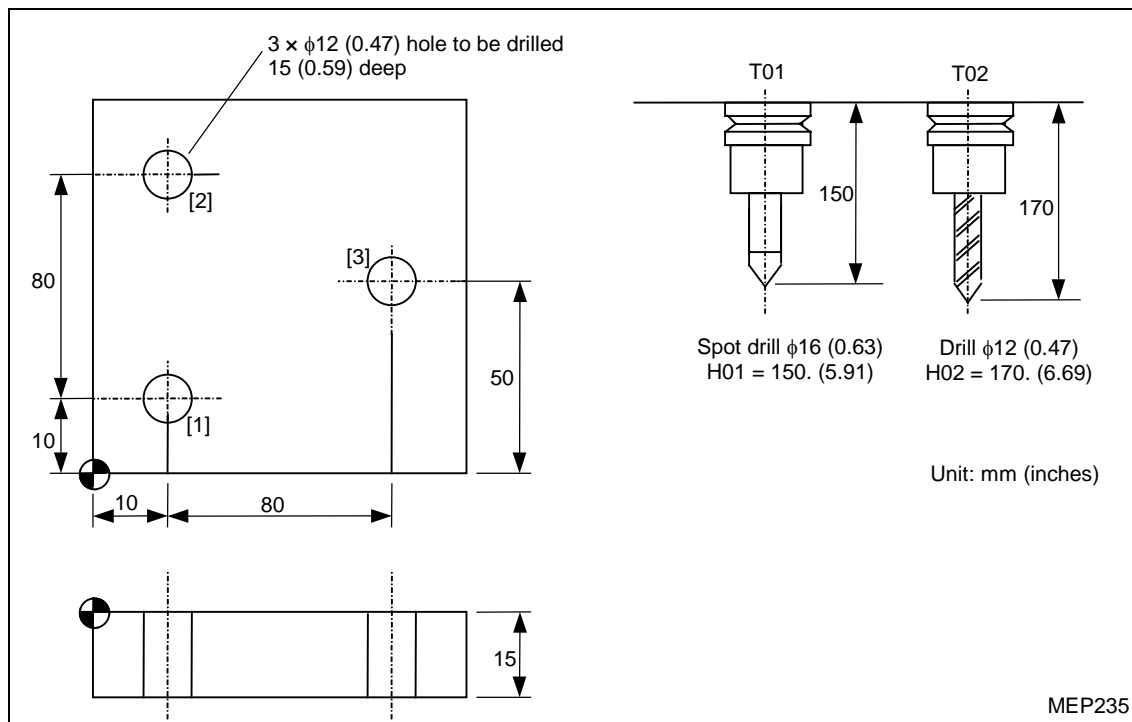
3. Inclined-surface machining (Facial angle: 30 degrees)

<p>O200</p> <p>N01 (F-MIL D200) G90G00G40G49G80 G91G30Y0Z0 T01T00M06 G90G54S225M03 M176 G00A30. G68I0J1K0R-[#5024] G68I1J0K0R-30. M98P201H01 G69 G91G28Z0 G28X0Y0 M01</p> <p>N02 (SPOT D16) G90G00G40G49G80 G91G30Y0Z0 T02T00M06 G90G54S1570M03 M176 G00A30. G68I0J1K0R-[#5024] G68I1J0K0R-30. M98P201H02 G69 G91G28Z0 G28X0Y0 M01</p>	<p>N03 (DRL D8.5) G90G00G40G49G80 G91G30Y0Z0 T03T00M06 G90G54S1600M03 M176 G00A30. G68I0J1K0R-[#5024] G68I1J0K0R-30. M98P201H03 G69 G91G28Z0 G28X0Y0 M01</p> <p>N04 (TAP M10 P1.5) G90G00G40G49G80 G91G30Y0Z0 T04T00M06 G90G54S250M03 M176 G00A30. G68I0J1K0R-[#5024] G68I1J0K0R-30. M98P201H04 G69 G91G28Z0 G28X0Y0 M01</p> <p>N05 G91G28A0B0 M30</p>
<p>O201</p> <p>N01 (F-MIL D200) G90G00S225M03 G00X-110.Y160. G43Z50.H01M08 G00Z0 G01X410.F160 G00Z50. X-110.Y40. Z0 G01X410.G00Z50.M09 M99</p> <p>N02 (SPOT D16) G90G00S1570M03 G00X20.Y20. G43Z50.H02M08 G99G82Z-5.5R5.F127 M98H05 G80G00Z50.M09 M99</p>	<p>N03 (DRL D8.5) G90G00S1600M03 G00X20.Y20. G43Z50.H03M08 G99G82Z-5.5R5.F520 M98H05 G80G00Z50.M09 M99</p> <p>N04 (TAP M10 P1.5) G90G00S250M03 G00X20.Y20. G43Z50.H04M08 G99G84Z-13.R5.H100F1.5 M98H05 G80G00Z50.M09 M99</p> <p>N05 (SUB PRO) X280. Y180. X20. M99</p>

#5024 : Fourth-axis machine coordinate

23 PROGRAMS EXAMPLES

Example 1:



N001	(SPOT D16)	
G90G80G40G49G00	Initial setting	
G91G28Z0	Zero point return (Z-axis)	
G28X0Y0	Zero point return (X- and Y-axes)	
T01T00M06	Tool change	
G90G54S1590M03	Coordinate system setting, Spindle start	
G00X10.Y10.	Positioning at hole position [1]	
G43Z50.H01M08	Movement to initial point, Tool length offset	
G99G82R5.Z-5.F127	Hole [1] drilling	
Y90.	Positioning, Hole [2] drilling	
X90.Y50.	Positioning, Hole [3] drilling	
G80G00Z50.M09	Fixed-cycle cancellation, Movement to initial point	
G91G28Z0	Zero point return (Z-axis)	
G28X0Y0	Zero point return (X- and Y-axes)	
M01	Optional stop	
N002	(DRL D12)	
G90G80G40G49G00	Initial setting	
G91G28Z0	Zero point return (Z-axis)	
G28X0Y0	Zero point return (X- and Y-axes)	
T02T00M06	Tool change	
G90G54S1590M03	Coordinate system setting, Spindle start	
G00X10.Y10.	Positioning at hole position [1]	
G43Z50.H02M08	Movement to initial point, Tool length offset	
G99G73R5.Z-19.Q4.F76	Hole [1] drilling	
Y90.	Positioning, Hole [2] drilling	
X90.Y50.	Positioning, Hole [3] drilling	
G80G00Z50.M09	Fixed-cycle cancellation, Movement to initial point	
G91G28Z0	Zero point return (Z-axis)	
G28X0Y0	Zero point return (X- and Y-axes)	
M30		

Example 2:

```

N001 (F-MIL D100)
G55G90G94G40G49
G91G28Y0Z0
T01T00M06
G90G00S480M03
X-110.Y30.
G43Z50.H01M08
G00Z.3
G01X110.F290
Y-30.
X-110.
G00Z50.
X-110.Y30.
Z0
G01X110.
G00Z50.
X-110.Y-30.
Z0
G01X110.
G00Z50.
G91G28Y0Z0
M01

```

```

N002 (SPOT D16)
G55G90G94G40G49
G91G28Y0Z0
T02T00M06
G90G00S1500M03
X-35.Y30.
G43Z50.H02M08
G99G81Z-5.R3.F127
X35.
X0Y0
G80M09
G91G28Y0Z0
M01

```

```

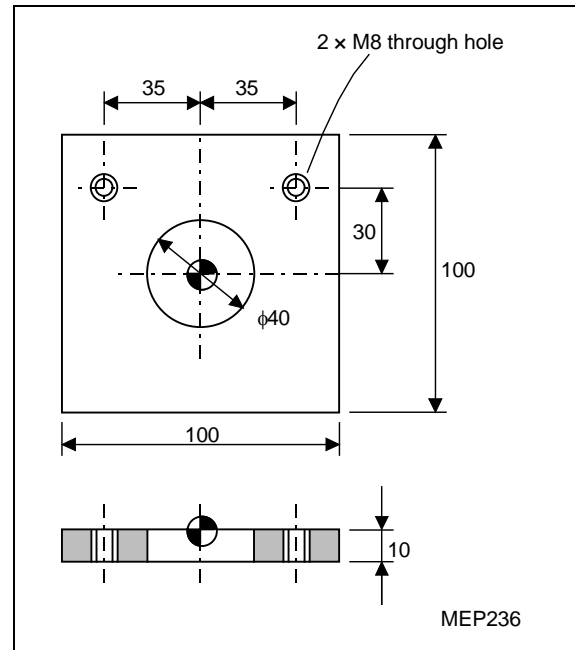
N003 (DRL D6.8)
G55G90G94G40G49
G91G28Y0Z0
T03T00M06
G90G00S935M03
X-35.Y30.
G43Z50.H03M08
G99G82Z-12.R3.F75
X35.
G80M09
G91G28Y0Z0
M01

```

```

N004 (DRL D39.5)
G55G90G94G40G49
G91G28Y0Z0
T04T00M06
G90G00S185M03
X0Y0
G43Z50.H04M08
G99G82Z-22.R3.F56
G80M09
G91G28Y0Z0
M01

```



```

N005 (BOR D40.)
G55G90G94G40G49
G91G28Y0Z0
T05T00M06
G90G00S720M03
G43Z50.H05M08
G99G76Z-10.R3.F80
G80M09
G91G28Y0Z0
M01

```

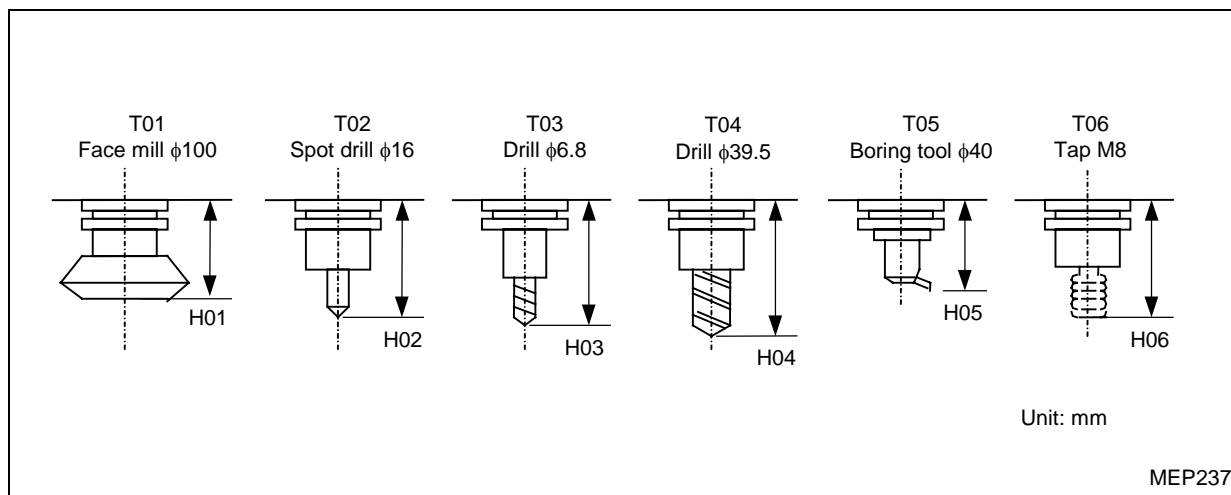
```

N006 (TAP M8)
G55G90G94G40G49
G91G28Y0Z0
T06T00M06
M00
G90G00S280M03
X-35.Y30.
G43Z50.H06
G99G84Z-16.25R3.F350H0
X35.
G80M09
G91G28Y0Z0
G28X0
T00T00M06
M30

```

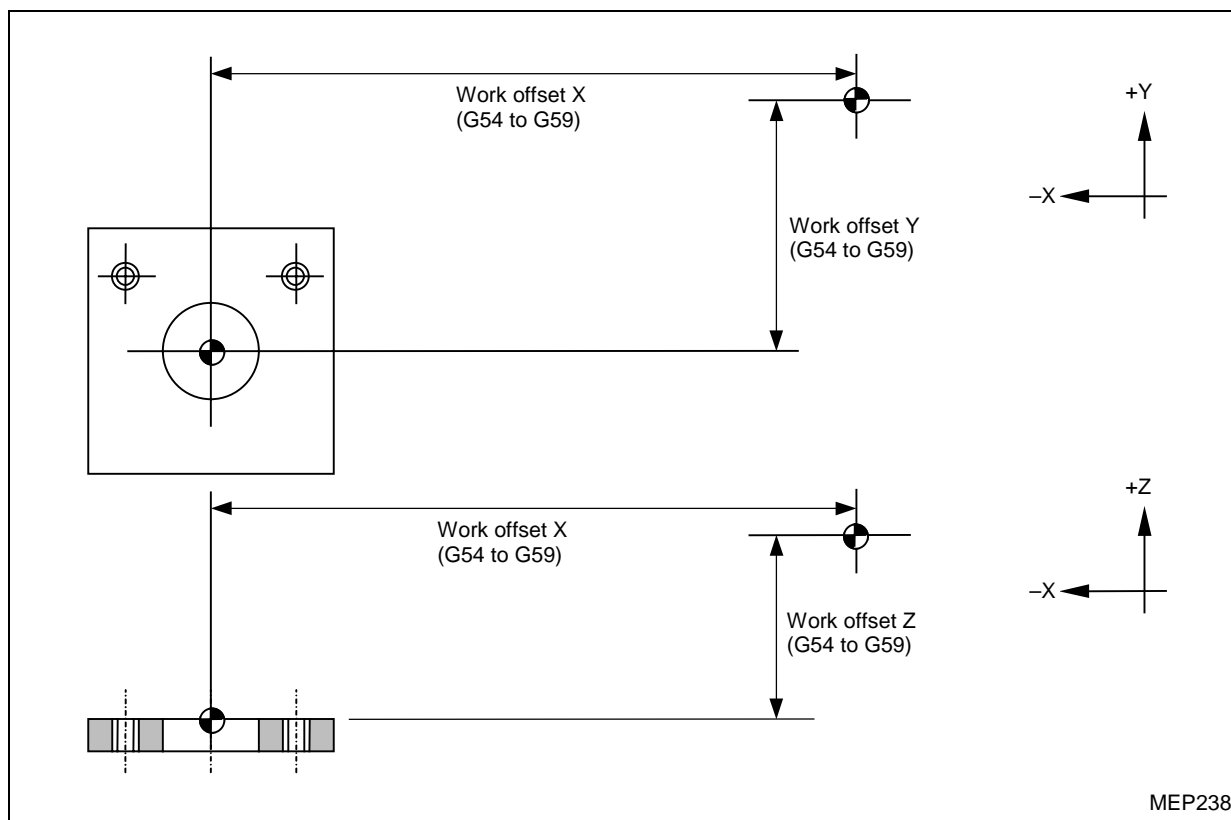
<Tool offset>

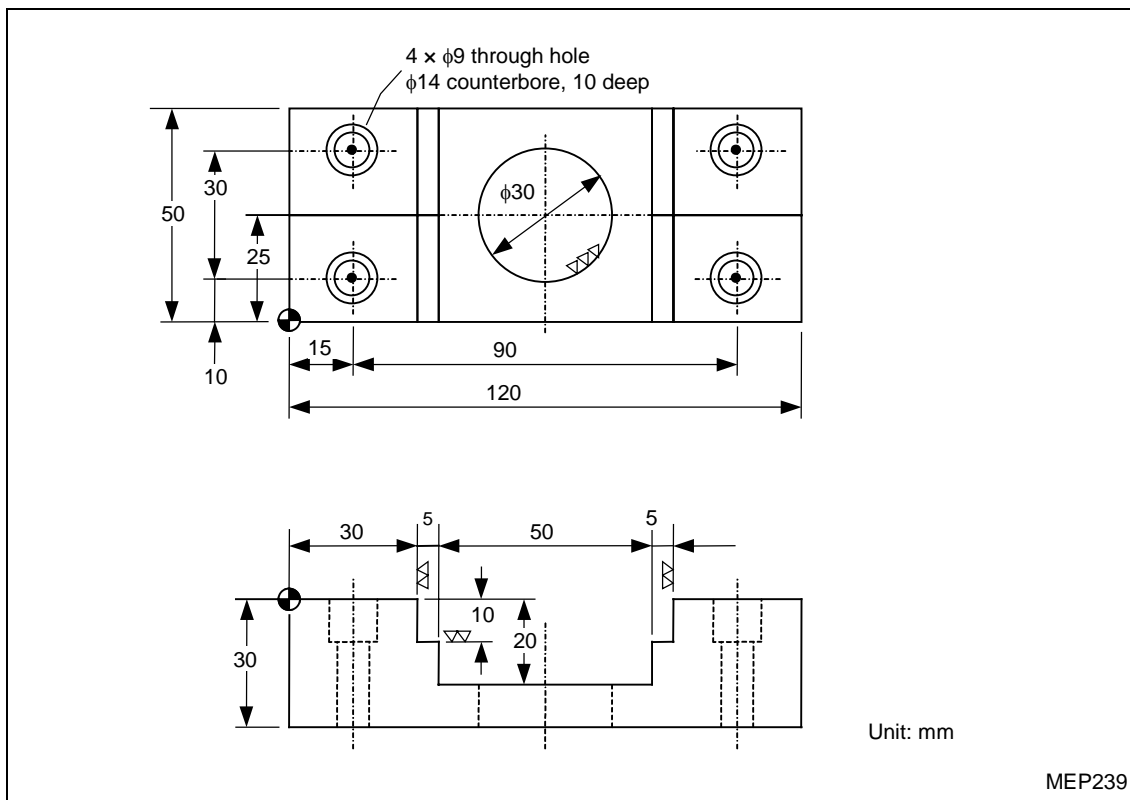
Set tool length offset data and tool diameter offset data.



<Work offset>

Set in the workpiece coordinate system (G54 to G59) the distance from the machine zero point to the workpiece zero point.



Example 3:

N001 (E-MIL D20)
G90G80G40G00G49
G91G28Y0Z0
T01T00M06
G90G54G00X40.Y-20.
S320M03
G43Z50.H01M08
G00Z-10.
G42G01X303Y10.D11F64
Y70.
X89.7F1000
Y-20.F64
X30.F1000
S340M03
G01Y70.F40
X90.F1000
Y-20.F40
G40G00Z50.M09
G91G28Y0Z0
M01

N002 (SPOT D16)
G90G80G40G00G49
G91G28Y0Z0
T02T00M06
G90G54G00X15.Y40.
S1590M03
G43Z50.H02M08
G99G81Z-5.R3.F127
M98H008
G80G00Z50.
X60.Y25.
G98G81Z-25.R-17.F127
G80G00Z50.M09
G91G28Y0Z0
M01

N003 (DRL D9.)
G90G80G40G00G49
G91G28Y0Z0
T03T00M06
G90G54G00X15.Y40.
S750M03
G43Z50.H03M08
G99G83Z-32.7R3.Q5.I5.B70.
M98H008
G80G00Z50.M09
G91G28Y0Z0
M01

N004 (DRL D29.5)
G90G80G40G00G49
G91G28Y0Z0
T04T00M06
G90G54G00X60.Y25.
S250M03
G43Z50.H04M08
G98G83Z-38.85R-17.Q7.I10.B70.
G80G0Z50.M09
G91G28Y0Z0
M01

N005 (E-MIL D14)
G90G80G40G00G49
G91G28Y0Z0
T05T00M06
G90G54G00X15.Y40.
S420M03
G43Z50.H05M08
G99G81Z-10.R3.F30
M98H008
G80G00Z50.
G91G28Y0Z0
M01

N006 (BOR D30.)
G90G80G40G00G49
G91G28Y0Z0
T06T00M06
G90G54G00X60.Y25.
S900M03
G43Z50.H06M08
G98G76Z-30.R-17.F72
G80G00Z50.M09
G91G28Y0Z0
M01

N007 (MENTORI D30)
G90G80G40G00G49
G91G28Y0Z0
T07T00M06
G90G54G00X15.Y40.
S2100M03
G43Z50.H07M08
G99G81Z-2.5R3.F530
M98H008
G80G00Z50
G00X60.Y25.
G71.1Z-25.R-17.Q1.F530
G80G00Z50.M09
G91G28Y0Z0
G28X0
M30

N008 (SUB PRO)
Y10.
X105.
Y40.
M99

- NOTE -

24 EIA/ISO PROGRAM DISPLAY

This chapter describes general procedures for and notes on constructing an EIA/ISO program newly, and then editing functions.

24-1 Procedures for Constructing an EIA/ISO Program

- (1) Press the display selector key.
- (2) Press the **[PROGRAM]** menu key.
 - ➔ The PROGRAM display will be selected.
- (3) Press the **[WORK No.]** menu key.
 - ➔ **WORK No.** is displayed in reverse to show the window of work number list.

Remark: Refer to the Operating Manual for the window of work number list.

- (4) Enter the new work number of a program to be constructed.
 - Specifying a work number of a program registered already in NC unit allows the program to be displayed on the screen. Therefore, constructing a new program requires specifying a work number which has not been used. The conditions how work numbers are used are displayed on the window of work number list.
- (5) Press the **[EIA/ISO PROGRAM]** menu key.
 - Press the **[PROGRAM EDIT]** menu key instead of **EIA/ISO PROGRAM** if a work number of the program already registered has been set in Step (4).



- (6) Enter the required programming data.
 - Set data using alphabetic keys, numeric keys and INPUT key ().
 - When INPUT key is pressed, the cursor is moved to the top of the next line, and then the data of the next block can be entered.
- (7) Press the **[PROGRAM COMPLETE]** menu key to end the editing.

24-2 Editing Function of EIA/ISO PROGRAM Display

24-2-1 General

Establishing a constructing mode on the **PROGRAM (EIA/ISO)** display allows the following menu to be displayed as an initial one.

PROGRAM COMPLETE	SEARCH	COPY	ALTER	ERASE	MOVE	FIND & REPLACE		MACRO INPUT	MACRO VARIABLE
	[1]	[2]	[3]	[4]	[5]	[6]			

Terms [1] to [6] represent functions related to the program editing. Use of the functions permits the following operations:

- Inserting and altering data at any position
Data can be inserted and altered at any position on the display.
- Erasing the data
Data displayed on the display can be erased.
- Searching for the data
Data can be searched in the following four ways.
 - 1) Searching for the top line of the program
 - 2) Searching for the bottom line of the program
 - 3) Searching for any required line of the program
 - 4) Searching for any character string
- Copying the data
Other EIA/ISO programs registered in the NC unit can be copied into the selected program, or any data character string in the selected program can be copied into a given position of the selecting program or a new EIA/ISO program.
- Moving the data
Any data character string can be moved to a given position of the selecting program or a new EIA/ISO program.
- Replacing the data
Any data character string can be replaced by another character string.

24-2-2 Operation procedure

The procedure for each operation is described below.

(Given that EIA/ISO program, in which several lines of data are already provided, is selected, and editing mode is established, and also that **[ALTER]** menu item is not displayed in the reverse status in the operations 3 and onward.

1. Inserting the data

- (1) Press the **[ALTER]** menu key as required to obtain the display status ALTER.
 - When **ALTER** is displayed, press the menu key to cancel the reverse-display status.
- (2) Move the cursor to the position where data must be inserted.
 - The cursor can be moved to any direction (vertically and horizontally).
- (3) Enter the required data.
 - Data is inserted in sequence into the position where the cursor is placed.
 - Data previously set behind the cursor position are moved behind the inserted data.

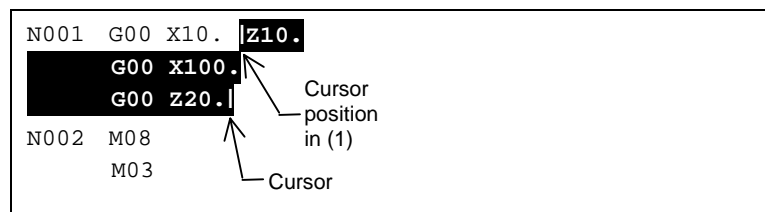
2. Altering the data

- (1) Press the **[ALTER]** menu key to display **ALTER**.
 - When ALTER is displayed, press the menu key to reverse the display status.
- (2) Move the cursor to the position where data must be altered.
 - The cursor can be moved to any direction (vertically and horizontally).
- (3) Enter the required data.
 - Data is altered in sequence from the position where the cursor is placed.
 - The character previously set at the cursor position is replaced in sequence by the new data.

3. Erasing the data

- (1) Move the cursor to the head of the character string to be erased.
- (2) Press the **[ERASE]** menu key.
 - ➔ The character at the cursor position is displayed in reverse and the **[ERASE]** menu item is also displayed in reverse.
- (3) Move the cursor to the position next to the end of the character string to be erased.
 - ➔ The portion from the head of the character string specified in (1) to the position before the cursor is displayed in reverse, which indicates that the reversed portion provides the object of erasure.

Example:



- (4) Press the input key.
 - ➔ The character string displayed in reverse in (3) is erased.

Example:

```

N001  G00 X10.
N002  M08
      M03
  
```

4. Searching for the data

A. Searching for the top line of the program

- (1) Press the **[SEARCH]** menu key.
- (2) Press the **[PROG HEAD]** menu key.
- ➔ The cursor moves to the top line.

B. Searching for the bottom line of the program

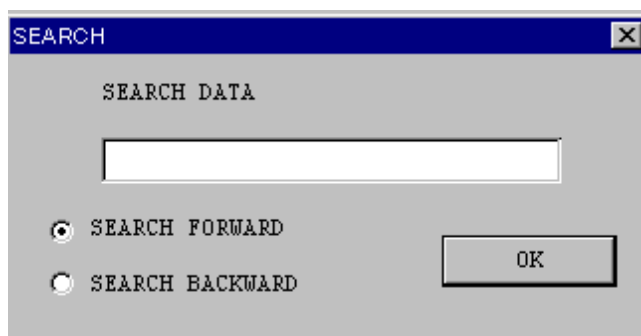
- (1) Press the **[SEARCH]** menu key.
- (2) Press the **[PROG END]** menu key.
- ➔ The cursor moves to the bottom line.

C. Searching for any required line of the program

- (1) Press the **[SEARCH]** menu key.
- (2) Press the **[SEARCH LINE No.]** menu key.
- ➔ **SEARCH LINE No.** is displayed in reverse.
- (3) Set the line No. of the line to be searched for.
- Enter the line No. with numeric keys, and press the input key.
- ➔ The cursor moves to the specified line.

D. Searching for any character string

- (1) Press the **[SEARCH]** menu key.
- (2) Press the **[SEARCH FORWARD]** menu key or **[SEARCH BACKWARD]** menu key.
- ➔ **SEARCH FORWARD** or **SEARCH BACKWARD** is displayed in reverse.



- To search for a character string in the area before the cursor position, press the **[SEARCH FORWARD]** menu key, and for the area after the cursor position, press **[SEARCH BACKWARD]** menu key.
- (3) Set the character string to be searched for and press the input key.
- ➔ The cursor moves to the head of the character string which has been found first.
- Press the data cancellation key (CANCEL) to stop halfway the searching operation, whose running state is indicated by the message **CNC BUSY** on the display.

Remark: Pressing the input key in sequence allows the cursor to move to the character string which has been found next.

5. Copying the data

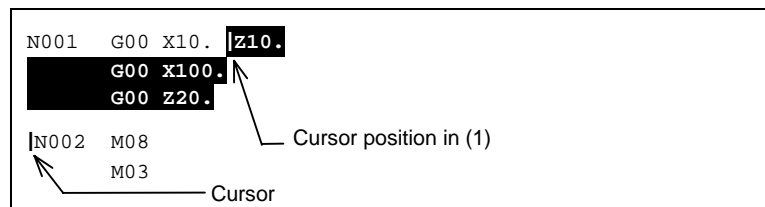
A. Copying a program

- (1) Move the cursor to the position where the program is to be copied.
 - (2) Press the **[COPY]** menu key.
 - (3) Press the **[PROGRAM COPY]** menu key.
 - ➔ The window of work number list is displayed and the **[PROGRAM COPY]** menu item is displayed in reverse.
 - (4) Set the work number of the program to be copied and press the input key.
 - ➔ The program is inserted into the cursor position.
- Note:** MAZATROL programs cannot be copied.

B. Copying any character string into the selected program

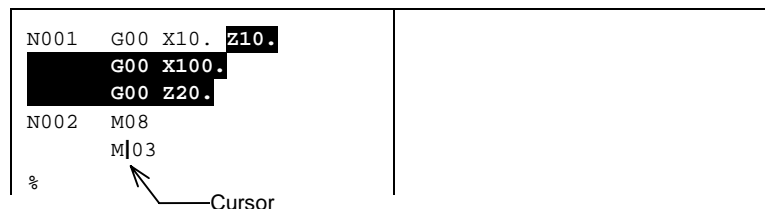
- (1) Move the cursor to the head of the character string to be copied.
- (2) Press the **[COPY]** menu key.
- (3) Press the **[LINE(S) COPY]** menu key.
 - ➔ The character at the cursor position is displayed in reverse and the **[LINE(S) COPY]** menu item is also displayed in reverse.
- (4) Move the cursor to the position next to the end of the character string to be copied.
 - ➔ The portion from the head of the character string specified in (1) to the position before the cursor is displayed in reverse, which indicates that the reversed portion provides the object of copying.

Example:



- (5) Press the input key.
 - ➔ The area displayed in reverse is established as the object to be copied.
- (6) Move the cursor to the position where the character string is to be copied.
 - ➔ The cursor only moves, and the area displayed in reverse does not change.

Example:



- (7) Press the input key.
 - ➔ The character string displayed in reverse is copied at the cursor position.

Example: (Continued)

N001	G00 X10. Z10.
	G00 X100.
	G00 Z20.
N002	M08
	Z10.
	G00 X100.
	G00 Z20.
	M03

C. Copying any character string into a new program

- (6) First, carry out Steps (1) to (5) of **B**, then set the workpiece number of a new program where the character string is to be copied and press the input key.

➔ The character string is copied in the new program, and the area displayed in reverse is returned to normal display.

Remark: Pressing the **[PROGRAM FILE]** menu key allows the window of program list to be displayed.

6. Moving the data**A. Moving the selected program to any position**

- (1) Move the cursor to the head of the character string to be moved.

- (2) Press the **[MOVE]** menu key.

➔ The character at the cursor position and the **[MOVE]** menu item is also displayed in reverse.

- (3) Move the cursor to the position next to the end of the character string to be moved.

➔ The portion from the head of the character string specified in (1) to the position before the cursor is displayed in reverse, which indicates that the reversed portion provides the object of moving.

N001	G00 X10. Z10.
	G00 X100.
	G00 Z20.
N002	M08
	M03

Cursor position in (1)

Cursor

- (4) Press the input key.

➔ The area displayed in reverse is established as the object to be moved.

- (5) Move the cursor to the position where the character string is to be moved.

- The cursor only moves, and the area displayed in reverse does not change.

Example: (Continued)

N001	G00 X10. Z10.
	G00 X100.
	G00 Z20.
N002	M08
	M03

Cursor

- (6) Press the input key.

➔ The character string displayed in reverse is moved to the cursor position.

Example: (Continued)

N001 G00 X10.	
N002 M08	
Z10.	
G00 X100.	
G00 Z20. M03	

B. Movement to a new program

- (5) First, carry out Steps (1) to (4) of **A**, then set the work number of a new program where the character string is to be moved and press the input key.

- Pressing the **[PROGRAM FILE]** menu key allows the window of program list to be displayed.

➔ The character string is moved to the new program.

7. Replacing the data


- (1) Move the cursor to the starting position of data replacement.

- Replacement is made downward from the cursor position. To make replacement throughout the program, therefore, move the cursor to the first character of the top line.

- (2) Press the **[FIND & REPLACE]** menu key.

➔ **FIND & REPLACE** is displayed in reverse.

- (3) Set the character string before replacement.

- Enter the character string to be replaced using alphanumeric keys, and press the tab key ()

- (4) Set the new character string after replacement using alphanumeric keys, and press the input key.

➔ The cursor moves to the head of the character string before replacement that has been found first after the cursor position specified in (1).

- (5) Press the **[REPLACE]** menu key.

➔ The character string before replacement at the cursor position is replaced by the character string after replacement, and the cursor moves to the head of the next character string before replacement. Pressing the **[REPLACE]** menu key in sequence allows the character string before replacement to be replaced in order of being found.

Remark 1: When replacing the special character string at the cursor position is not required, press the **[NO REPLACE]** menu key in place of **[REPLACE]** menu key.

Remark 2: To stop the replacement, press the **[END]** menu key.

Remark 3: To replace all the character strings in the program, press the **[NEXT]** menu key.

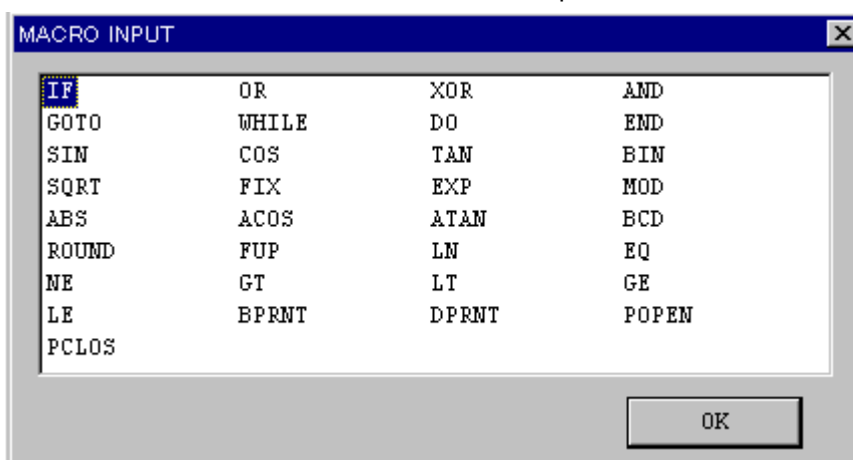
Remark 4: Press the data cancellation key (CANCEL) to stop halfway the total replacement by the NEXT menu function, whose running state is indicated by the message **CNC BUSY** on the display.

24-3 Macro-Instruction Input

This function permits entering the macro-instruction word by word for editing the EIA/ISO program efficiently.

(1) Press the **[MACRO INPUT]** menu key.

➔ The **MACRO INPUT** window will be opened.



- The characters selected with the cursor are usable.

(2) Move the cursor to the characters corresponding to the required macro-instruction and press the input key.

➔ The macro-instruction is entered in the editing zone of the program.

Press the menu selector key to display the menu for normal data input, and continue program editing.