Programming Manual for Milling Machine tool of AUCTECH controller

V2.0

Content

I. Programming

1 Overview

1.1 Fundamentals of programmed machining

1.1.1 Coordinate controller

There are two main coordinate controllers that are used during programmed machining: the workpiece coordinate controller and the machine tool coordinate controller.

The workpiece coordinate controller is the coordinate controller used for programming. The coordinate positions specified during programming are the values in the workpiece coordinate controller, which is why the workpiece coordinate controller is also called the programming coordinate controller.

The machine tool coordinate controller is the coordinate controller used to control the movement of the machine tool. To make the machine tool move according to the designed trajectory, a position command shall be sent to the servo motor, this position is the value in the machine tool's coordinate controller.

Therefore, in order for the machine tool to move in the path specified in programming in the workpiece coordinate controller, the commanded position in the workpiece coordinate controller must be converted to the machine tool coordinate controller in order to accurately control the movement of the machine tool. The conversion process is done automatically by the CNC controller and the user only needs to set the offset of the zero point of the workpiece coordinate controller. The offset of the zero point of the workpiece coordinate controller is the position of the zero point of the workpiece coordinate controller in the machine tool's coordinate controller.

The relationship between the workpiece coordinate controller and the machine tool coordinate controller is shown as Figure 1-1.

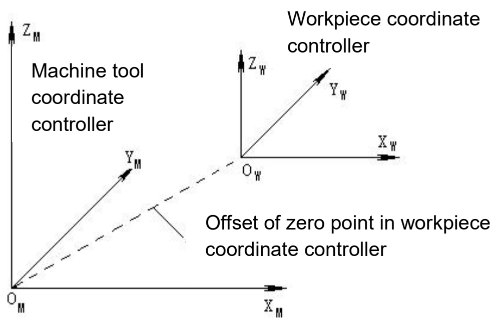


Figure 1- 1 Workpiece coordinate controller and machine tool coordinate controller

1.1.2 Setting of the zero point of the workpiece coordinate controller

The workpiece coordinate controller is fixed to the workpiece being machined. After the workpiece has been clamped to the table, the zero point of the workpiece coordinate controller is fixed in the machine tool's coordinate controller.

The position of the zero point of the workpiece coordinate controller can be determined in two ways after the workpiece has been clamped to the table.

- Positioning by using the datum point of the workpiece

A datum point on a workpiece is a defined point on the workpiece and the position of that point in the workpiece coordinate controller is known.

Once the workpiece has been clamped to the table, the position of the zero point of the workpiece coordinate controller can be positioned by means of datum surface and datum point on the workpiece.

Assuming the position of the datum point in the workpiece coordinate controller is (X_W , Y_W , Z_W), the calculation process of the zero point of the workpiece coordinate controller is described as follows.

1) Move the tip of the tool to the reference point position.
2) Read the machine tool coordinates of the tool tip from the CNC controller (X_M,Y_M, Z_M).
3) Calculate the position of the zero point of the workpiece coordinate controller in the machine tool coordinate controller.

   - $X_O = X_M – X_W$
   - $Y_O = Y_M – Y_W$
   - $Z_O = Z_M – Z_W$

- Positioning with clamps

The workpiece is clamped to the table by means of a fixture. If the mutual position of the workpiece and the fixture is fixed in each clamping (or if its position deviation is within the permissible error), the position of the zero point of the workpiece coordinate controller can be determined by the fixture.

As the clamping position of the fixture and the workpiece is fixed, a datum point can be determined on the fixture and the position of this datum point in the workpiece coordinate controller is determined unchanged after the workpiece has been clamped. In this way, the zero position of the workpiece coordinate controller can be determined from the datum point on the fixture.

Usually, for the machining of large quantities of parts, fixture shall be used for positioning. Since the fixture only need to be clamped once and the workpiece does not need to be repositioned when re-clamping, this method of positioning can greatly improve working efficiency.

1.1.3 Programming format of commands

The program consists of a number of command segments.

Each command segment consists of a command word and a number of parameter words.

Command segment programming format as Figure 1-2 shown.

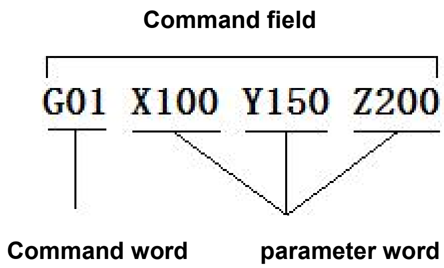


Figure 1-2　Programming format of commands

1.1.4 Program skip and comments

■ Program skip

The controller can selectively execute program segments that are preceded by the skip character '\'. These segments are ignored when the skip is valid, otherwise these segments are executed normally.

Example：

```
G0 X0 Y0 Z0
\G00 X20 Y30      ; this program segment is ignored if the skip is valid
G01 Z50 F300
M30
```

■ Program comments

Comment text is text in the program that explains the program, and is ignored when the program is run.

There are two types of program comments

- Line comment: The text following the comment character ';' in a program segment is the comment text.
- Block comments: Block comments can comment multiple lines of text, the content enclosed in brackets '(' and ')' is the comment text.

Example：

```
(Program name: test.nc
Tool radius: 3mm
Spindle speed: 3000rpm)

M03 S1000
;Quick positioning
G90 G00 X100 Y200
G17 G02 X150 I25 F300; circular interpolation
```

1.1.5 Program structure and execution process

A program consists of a main program and a subprogram. A program must have a main program, while subprograms are not required.

The main program shall be placed at the beginning of the program file and end with M30 or M99. Subprograms may only end with M99.

Subprogram names can only be numeric, and their storage location can take two forms:

- Subprograms are written in the same program file with the main program, and subprograms can only follow the main program. No sequential requirement between subprograms and subprograms.
- Subprograms are separate program files whose filenames are the names of the subprograms (they cannot have extensions).

The program execution process is shown in Figure 1-3.

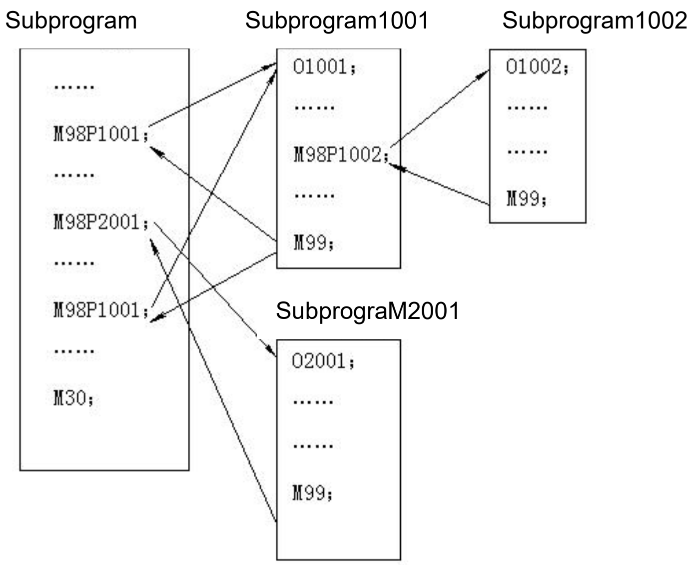


Figure 1- 3 The execution process of the program

1.2 Preparation function (G-code)

1.2.1 Modal G Code

G-codes can be divided into two types according to their validity period:

- Non-modal G-codes: valid only for the segment of the program for which the G-code is specified.
- Modal G-code: this type of G-code is executed once and then stored by the CNC controller and remains valid until other G-codes in the same group are executed.

Example.

G53 X0 Y0 Z0 ; G53 is a non-modal code, to be specified each time it is executed

G01 X-200 F300

Y-100 ;G01 is a modal code that is executed once and then stored without being specified again.

Z-250

M30

1.2.2 G-code grouping

G-codes are divided into groups according to their functional category, of which group 00 is a non-modal G-code and all other groups are modal G-codes. Multiple G-codes of different groups can be specified in the same segment. If more than one code of the same group is specified in the same segment, only the last specified code is valid.

The G-codes supported by this controller are shown in the table below.

No.	G-Code	Group	Function description
1	[G00]	01	Quick location
2	[G01]		Linear interpolation
3	G02		Circular interpolation/Helix interpolation (clockwise)
4	G03		Circular interpolation/Helix interpolation (counterclockwise)
5	G02.4		3D circular dimensional interpolation
6	G03.4		
7	G04	00	Pause, exact stop
8	G28		Return to reference point
9	G29		Return from the reference point
10	G30		Return to reference points 2, 3 and 4
11	G31		Jump commands
12	G52		Setting up the local coordinate controller
13	G53		Selecting the machine tool coordinate controller
14	G60		Unidirectional positioning
15	G65		Macro calls
16	G92		Setting the workpiece coordinate controller
17	[G15]	17	Polar coordinate command cancellation
18	G16		Polar coordinate commands
19	[G17]	02	Select XY plane
20	[G18]		Select the ZX plane
21	[G19]		Select YZ plane
22	[G40]	07	Tool radius compensation cancellation
23	G41		Tool radius compensation, left
24	G42		Tool radius compensation, right
25	G43	08	Positive tool length compensation

26	G43.4		Three-dimensional tool length compensation (RTCP)
27	G44		Negative tool length compensation
28	[G49]		Tool length compensation cancellation
29	[G50]	11	Proportional scaling cancelled
30	G51		Proportional scaling is valid
31	[G50.1]	22	Programming Mirror Cancellation
32	G51.1		Programming Mirror
33	G54.1	14	Select additional workpiece coordinate controller
34	G54		Select the 1st workpiece coordinate controller
35	G55		Select the 2nd workpiece coordinate controller
36	G56		Select the 3rd workpiece coordinate controller
37	G57		Select the 4th workpiece coordinate controller
38	G58		Select the 5th workpiece coordinate controller
39	G59		Select the 6th workpiece coordinate controller
40	[G61]	15	Exact stop
41	G64.1		High speed and high precision mode I
42	G64.2		High Speed High Precision Mode II
43	G68	16	Coordinate controller rotation valid
44	[G69]		Coordinate controller rotation cancellation
45	G73	09	Deep hole drilling cycles
46	G74		Reverse tapping cycle
47	G76		Fine boring cycle
48	[G80]		Fixed cycle cancellation
49	G81		Drilling cycle
50	G82		Drilling cycle (with stop)
51	G83		Chip removal drilling cycle
52	G84		Rigid tapping
53	G85		Boring cycle
54	G86		Boring cycle
55	G87		Anti-bore cycle
56	[G90]	03	Absolute value programming
57	[G91]		Incremental value programming

58	[G98]	10	Fixed cycle returns to initial point
59	G99		Fixed cycle back to point R

When the controller is powered up, the following groups of G-code default modal commands can be selected by the channel parameters (where XX represents the channel number).

Group 01: P1XX0300 = 0: G00 at power-up
=1: G01 at power-up

Group 02: P1XX0320 = 0: G17 at power-up
=1: G18 at power-up
=2: G19 at power-up

Group 03: P1XX0310 = 0: G90 at power-up
=1: G91 at power-up

1.2.3 Movement commands and status commands

In terms of the role of the commands, G-code can be divided into two types: Movement commands and status commands.

- Movement commands

Movement commands are commands that control the movement of the machine tool's coordinate axes, such as G00, G01, etc.
The Movement command, when executed, outputs a string of coordinate commands to control the position of the motor.
Depending on the form of the machine tool, there is a distinction between tool movement and workpiece movement, it is assumed to be tool movement in this manual.

- Status commands

A status command, when executed, only changes the modal command of the group it is in and does not output a coordinate command (it does not control the movement of the coordinate axes), but it does affect how subsequent movement commands are executed and their results.

For example, G90 and G91 are status commands. After the execution of G90, the coordinate values programmed by subsequent movement commands are all absolute positions; after the execution of G91, the coordinate values programmed by subsequent movement commands are all position increments.

2 Movement commands

2.1 Basic movement commands

2.1.1 Rapid positioning (G00)

■ Introduction to commands

When G00 is executed, the tool moves at a fast speed to the position in the workpiece coordinate controller specified by the absolute or incremental value command.
The G00 has two types of positioning: interpolated positioning and non-interpolated positioning.

- Interpolation positioning: The controller automatically adjusts the speed of each axis so that the tool is positioned along a straight line and each axis reaches the positioning endpoint at the same time. The synthetic rapid traverse speed during interpolation positioning is specified by the channel parameter P1XX0141.
- Non-interpolated positioning: the positioning speed of each axis is the maximum rapid traverse speed, the positioning path is not necessarily straight, and each axis does not necessarily reach the positioning endpoint at the same time. The rapid traverse speed of each axis in non-interpolated positioning is specified by the axis parameter P2XX0110.

The G00 positioning method is selected by the channel parameter P1XX0140.
P1XX0140 = 0: non-interpolated positioning method
=1: Interpolation positioning method

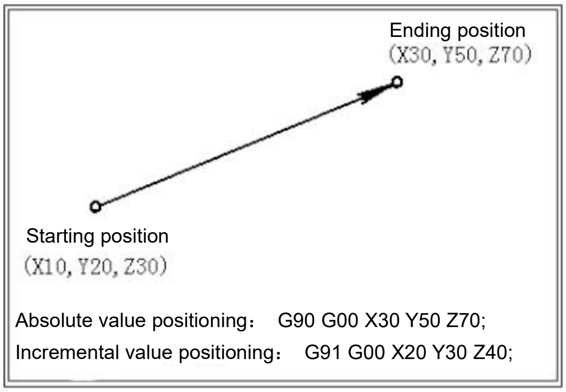


Figure 2- 1 G00 Interpolated positioning

■ Command Format

G00 IP_

IP_: coordinate value of the end point in the case of absolute commands; distance traveled by the tool in the case of incremental commands.

2.1.2 Linear interpolation (G01)

■ Introduction to commands

When G01 is executed, the tool moves in a straight line to a position in the workpiece coordinate controller specified by an absolute or incremental value command at a speed specified by the F command.

The feed speed specified by the F command is valid until a new value is specified, so there is no need to specify an F command for each program segment. The speed value specified by the F command shall be within the controller limit (0 to the maximum speed set by the controller parameters), if the speed specified by the F command exceeds the maximum speed limit, the controller uses the maximum speed as the feed speed.

The maximum feed speed limit is set by the channel parameter P1XX0130 (XX for the channel number).
The feed acceleration is set by the channel parameter P1XX0131 (XX stands for the channel number).

■ Command Format

G01 IP_ F_

IP_: coordinate value of the end point in the case of absolute commands; distance travelled by the tool in the case of incremental commands.
F_: Specify feed speed

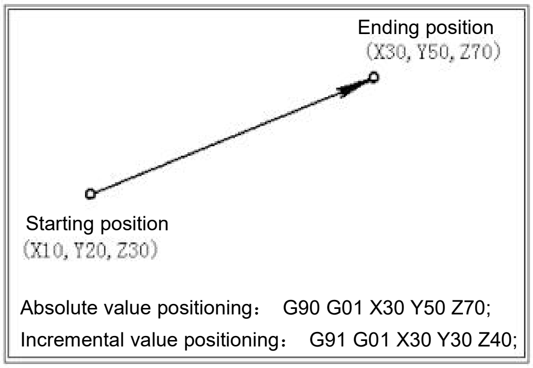


Figure 2- 2 G01 Linear interpolation

Feed speeds for linear axes are in millimeters per minute; for rotary axes, feed speed is in degrees per minute.

■ Example

- Interpolation of linear axes

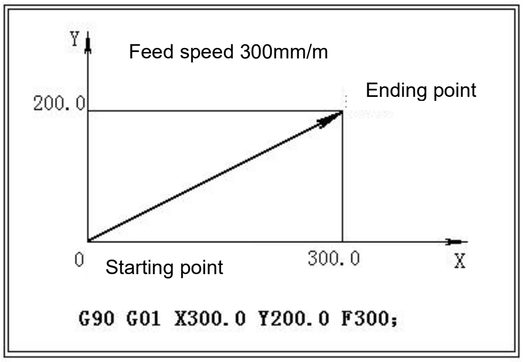


Figure 2- 3 G01 Linear axis interpolation

- Interpolation of rotary axes

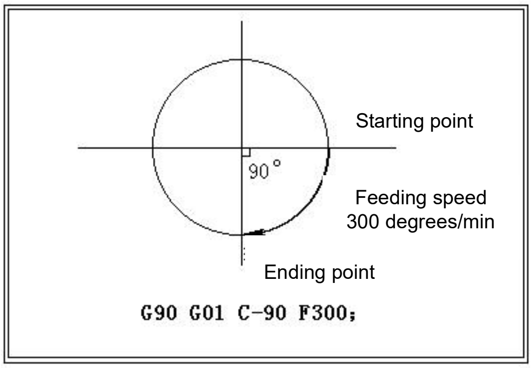


Figure 2- 4 G01 Rotary axis interpolation

2.1.3 Circular interpolation (G02/G03)

■ Introduction to commands

The circular movement command causes the tool to feed clockwise (G02) or counterclockwise (G03) along the circular path.

■ Direction of circular interpolation

The clockwise and counterclockwise directions in each plane are defined as:

- Looking at the XY plane along the Z axis from positive to negative determines the clockwise and anti-clockwise direction of the plane.
- Looking at the ZX plane along the Y axis from positive to negative determines the clockwise and anti-clockwise direction of the plane.
- Looking at the YZ plane along the X-axis from positive to negative determines the clockwise and anti-clockwise direction of that plane.

The clockwise and counterclockwise directions are defined as Figure 2- 5 shown.

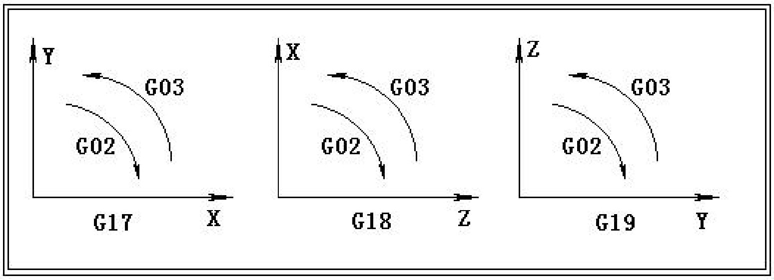


Figure 2- 5 Direction of circular interpolation

■ The end point of a circular arc is indicated

Specify the end point of a circular arc with the position command (X_, Y_, Z_). In the case of the absolute (G90) method, (X_, Y_, Z_) specifies the absolute position of the end point of the arc, in the case of the incremental (G91) method, (X_, Y_, Z_) specifies the distance from the start to the end point of the arc. As Figure 2-6 shown.

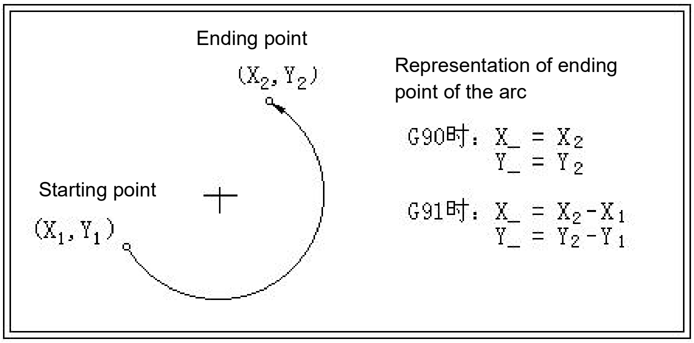


Figure 2-6 Representation of the end point of a circular arc

■ Distance from the starting point to the centre of the arc

Specify the position of the centre of the arc with the commands (I_, J_, K_). The parameters of the (I_, J_, K_) command are the vector components looking from the starting point towards the centre of the arc and are always incremental, whether G90 or G91.

(The command parameters of (I_, J_, K_) must specify their sign positive or negative depending on the direction.
If the position commands (X_, Y_, Z_) are all omitted when programming, this means that the start and end points coincide, in which case the programming with (I_, J_, K_) specifies a full circle.

The centre of the arc is represented as Figure 2-7 shown.

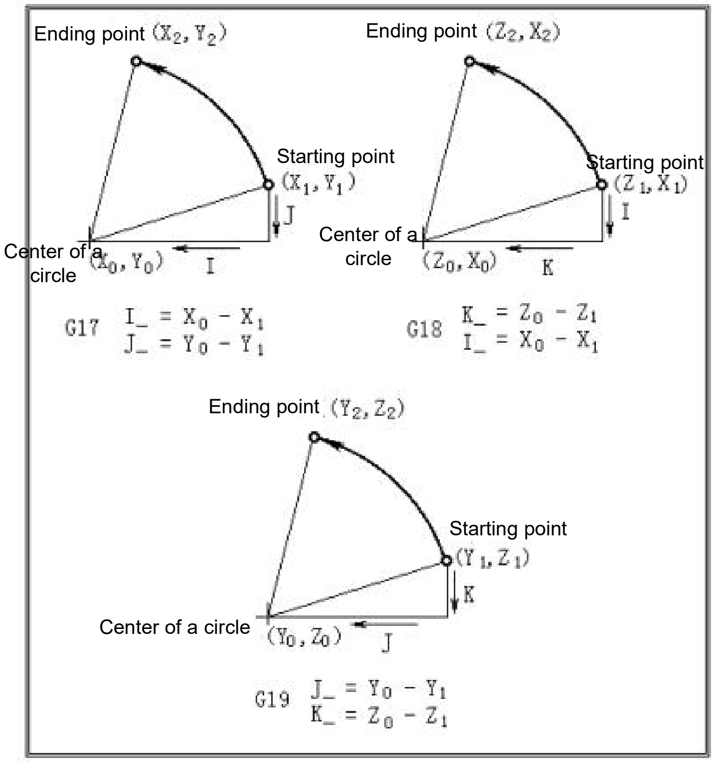


Figure 2-7　Representation of the centre of a circle

■　Radius of circle

The centre of a circular arc can be specified by the radius R of the arc in addition to (I_, J_, K_) as described above. When R is positive, arcs with a centre angle of less than 180 degrees are specified; when R is negative, arcs with a centre angle of greater than 180 degrees are specified. As Figure 2-8 shown.

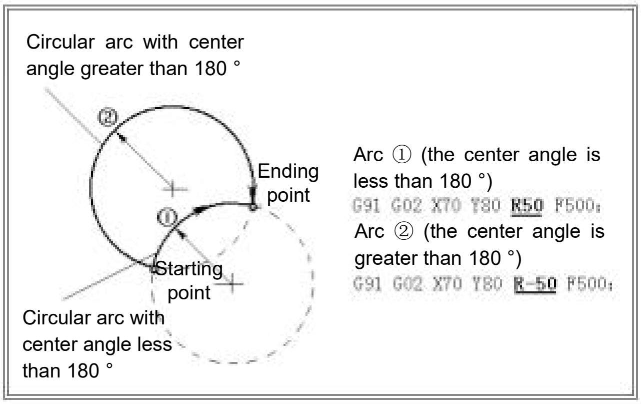


Figure 2-8　Positive and negative signs for the radius of a circle

■ Feed speed

The Feed speed for circular interpolation is the speed of the tool along the tangent of the arc, which is specified by the F command.

■ Command Format

Arc of XY plane: $G17 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X\_ \; Y\_ \begin{Bmatrix} I\_ \; J\_ \\ R\_ \end{Bmatrix} F\_;$

Arc of ZY plane: $G18 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X\_ \; Z\_ \begin{Bmatrix} I\_ \; K\_ \\ R\_ \end{Bmatrix} F\_;$

Arc of YZ plane: $G19 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Y\_ \; Z\_ \begin{Bmatrix} J\_ \; K\_ \\ R\_ \end{Bmatrix} F\_;$

Figure 2- 9　Arc command format

Command format description.

G17.	Specify the arc in the XY plane
G18.	Specify the arc of the ZX plane
G19.	Specify the arc of the YZ plane
G02.	Clockwise circular interpolation
G03.	Counterclockwise circular interpolation
X_/Y_/Z_.	The position of the end of the arc is the absolute position in the G90 mode and the distance from the start to the end of the arc in the G91 mode.
I_/J_/K_.	Distance from the start of the arc to the centre of the arc (with symbol)
R_.	Radius of circle (with symbol)
F_:.	Feed speed along the tangent of the arc

■ Example

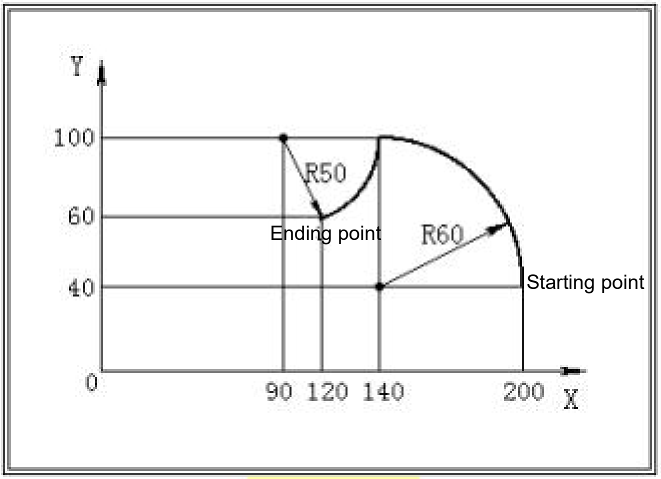


Figure 2- 10　Arc programming trajectory

The tool path shown above can be programmed in the following ways.

● Absolute value programming

G92 X200.0 Y40.0 Z0
G90 G03 X140.0 Y100.0 R60.0 F300.
G02 X120.0 Y60.0 R50.0

or

G92 X200.0 Y40.0Z0
G90 G03 X140.0 Y100.0 I-60.0 F300.
G02 X120.0 Y60.0 I-50.0

● Incremental value programming

G91 G03 X-60.0 Y60.0 R60.0 F300.
G02 X-20.0 Y-40.0 R50.0

or

G91 G03 X-60.0Y60.0 I-60.0 F300.
G02 X-20.0 Y-40.0 I-50.0

2.1.4 3D arc interpolation (G02.4/G03.4)

■ Introduction to commands

By specifying three non-overlapping points on a circular arc (start, middle and end points), circular arc interpolation can be performed in 3D space.

■ Command Format

G02.4/G03.4 X_ Y_ Z_ I_ J_ K_ F_

G02.4/G03.4: 3D arc interpolation command; G02.4 and G03.4 have the same function and can be substituted for each other.

X_/Y_/Z_: For absolute programming, command the position of the end point of the arc; for incremental programming, command the distance from the midpoint to the end point of the arc.

I_/J_/K_. For absolute programming, command the position of the middle point; for incremental programming, command the distance from the start of the arc to the middle point. The midpoint can be any point between the start and end points on the arc

F_. Feed speed along the tangent direction of the arc

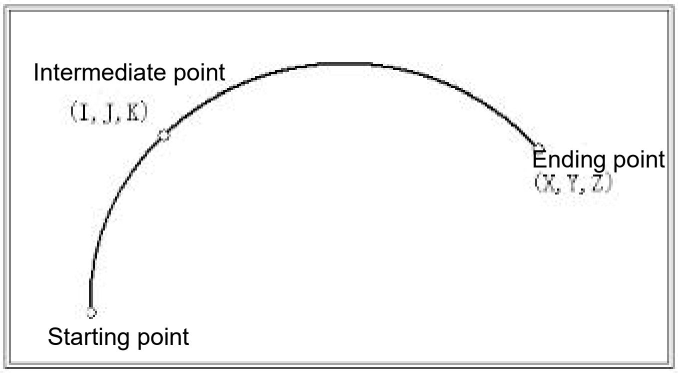


Figure 2- 11 Spatial arc interpolation

Restrictions

- Linear interpolation when the start, middle and end points are on the same line.

- Linear interpolation up to the end point when the start point and the middle point, the middle point and the end point, and the end point and the start point coincide.

- If the start, middle and end points are on the same line and the end point is between the start and middle points, first move in a linear interpolation from the start to the middle point, then perform a return operation in a linear interpolation from the middle to the end point.

As can be seen from the above limitations, the 3D arc movement command cannot program the whole circle (start and end point coincide), if you want to program the whole circle, you need to divide the whole circle into several arc segments and execute them in sections.

2.1.5 Helix interpolation (G02/G03)

■ Introduction to commands

When commanding circular interpolation, a further axis outside the plane of the arc is specified to move synchronously with the arc (follower axis) and the tool tip moves in a spiral line.

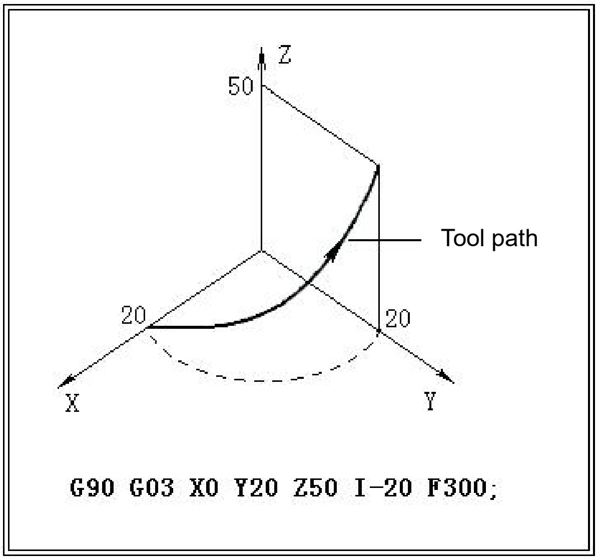


Figure 2- 12 Helix interpolation

■ Command Format

- XY planar arc, Z-axis follower

G17 g02/g03 x_ y_ z_ (i_ j_)/r_ l_ f_

G17. The XY plane is selected as the circular plane.
X_/Y_. Specify the arc end position for the absolute value method (G90) and the distance of the tool from the current position to the arc end for the incremental value method (G91).
Z_. Specify the end position of the follower axis for the absolute value method (G90) and the distance travelled by the follower axis for the incremental value method (G91).
I_ /J_. Specifies the distance (with sign) from the centre of the circle to the start of the arc.
R_. Specifies the radius of the arc (with sign).
L_. Number of spiral coils
F_. Specifies the feed speed along the helix.

- ZX plane arc, Y-axis follows

G18 g02/g03 x_ y_ z_ (i_ k_)/r_ l_ f_

G18. The ZX plane is selected as the circular plane.
X_/Z_. Specify the arc end position for the absolute value method (G90) and the distance of the tool from the current position to the arc end for the incremental value method (G91).
Y_. Specify the end position of the follower axis for the absolute value method (G90) and the distance travelled by the follower axis for the incremental value method (G91).
I_ /K_. Specifies the distance (with sign) from the centre of the circle to the start of the arc.
R_. Specifies the radius of the arc (with sign).
L_. Number of spiral coils
F_. Specifies the Feed speed along the helix.

- YZ plane arc, X-axis translation

G19 g02/g03 x_ y_ z_ (j_ k_)/r_ l_ f_

G19. The YZ plane is selected as the circular plane.
Y_/Z_. Specify the arc end position for the absolute value method (G90) and the distance of the tool from the current position to the arc end for the incremental value method (G91).
X_. Specifies the end position of the translation axis for the absolute value method (G90) and the distance travelled by the translation axis for the incremental value method (G91).
J_/K_. Specifies the distance from the centre of the circle to the start of the arc (with sign).
R_. Specifies the radius of the arc (with sign).
L_. Number of spiral coils
F_. Specifies the feed speed along the helix.

2.1.6 Skip (G31)

■ Introduction to commands

The skip command is used to implement the measurement function.

The G31 command is an axis movement command, similar to G01, whose speed is specified by F. During the execution of the G31 command, while moving towards the end position, if the controller detects the specified external input signal, it stops the execution of the current segment and jumps to the specified program segment to continue.

■ Command Format

G31 IP_ P_ L_ F_

IP_: Coordinate value of the end point in the case of absolute commands; distance travelled by the tool in the case of incremental commands.

P_: Probe number. The controller can support up to 6 probes, with head numbers 1 to 6.

May be omitted for P=1.

L_: The target segment to jump to when external input is detected, or if L is not specified, to the next segment automatically.

F_: Specifies the feed speed.

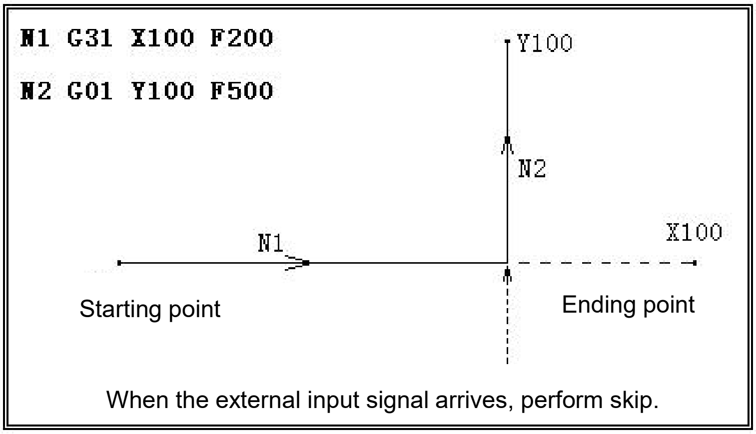


Figure 2- 13 Jump command execution process

2.1.7 Unidirectional positioning (G60)

■ Introduction to commands

In order to eliminate backlash in the machine tool, the coordinate axes can be made to converge to the command position from a fixed direction. This means that the final convergence to the end position is fixed, regardless of the direction of the end position relative to the start position. As Figure 2- 14 shown.

In order to achieve unidirectional positioning, an overshoot needs to be set.

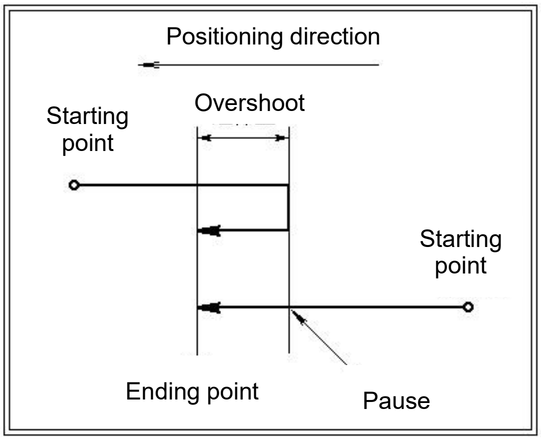


Figure 2- 14 Unidirectional positioning

■ Command Format

G60 IP_

IP_. End position for unidirectional positioning in absolute mode (G90), distance from the current position of the tool to the end position in incremental mode (G91).

■ Parameter settings

The convergence direction and overshoot for unidirectional positioning are set by the axis parameters (where XX is the physical axis number).

- The convergence direction is set by the axis parameter P2XX0191
- The overshoot is set by the axis parameter P2XX0190

- Cautions

  - The G60 command must be followed by a position command, otherwise the program alarms.
  - Execution of unidirectional positioning even if the tool travel distance is zero.
  - The overshoot setting for unidirectional positioning shall be greater than the backlash of the corresponding shaft, otherwise backlash cannot be completely eliminated in unidirectional positioning.

2.1.8 Return to reference points (G28, G29, G30)

■ Introduction to commands

Reference points are fixed points on the machine tool and this controller supports six reference points. The return to reference point command can be used to move the tool to these reference point positions.

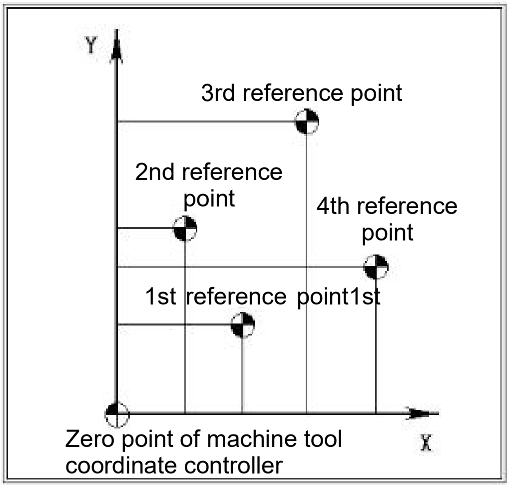


Figure 2- 15 Reference points

When returning to the reference point, the tool automatically moves quickly past the intermediate point to the position of the reference point, while the specified intermediate point is CNC controller stored and will be used when returning from the reference point.

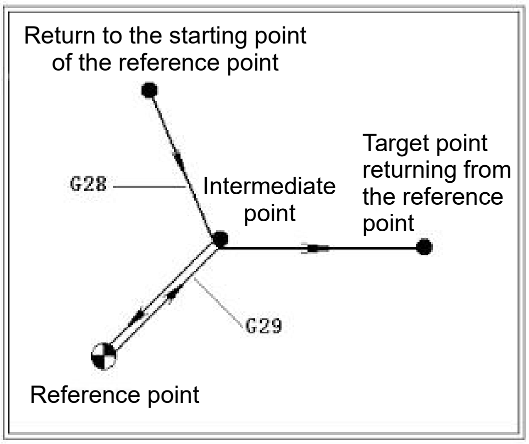


Figure 2- 16 Return to and from the reference point

■ Command Format

- Return to first reference point

G28 IP_

IP_. Specify the absolute position of the middle point in the absolute value mode (G90) and the distance of the middle point from the start point in the relative value mode (G91).
The coordinates of the IP_ command are the values in the workpiece coordinate controller.

- Back to the second, third, fourth, fifth and sixth reference points

G30 P_ IP_

P_: Specifies the reference point number to be returned (2 to 6, omitted if P=2).
IP_: Specifies the absolute position of the midpoint in the absolute value mode (G90) and the distance of the midpoint from the start point in the relative value mode (G91).
The middle point is the position in the workpiece coordinate controller.

- Return from the reference point

G29 IP_

IP_: Specifies the position of the returned target point for the absolute value method (G90) and the distance of the returned target point from the intermediate point for the relative value method (G91).
Returned target point as the value in the workpiece coordinate controller.

■ Cautions

- G28 or G30 shall be executed before G29 to return to the reference point, otherwise the controller reports an error.
- When the G28/G30 command is executed, only the axes for which the midpoint is commanded move; axes for which the midpoint is not commanded remain in their current position.
- Compensation functions such as tool radius compensation and length offset shall be removed before executing G28/G30.
- The intermediate point is the modal value, the intermediate point of the command in the G28/G30 program segment is stored by the controller, the axes without the command intermediate point will use the coordinates of the intermediate point of the command from the previous execution of G28/G30.
- The intermediate point is cleared when M30 is executed or when the controller is reset.
- G28, G30 and G29 are executed at fast travel speeds, which can be adjusted with the fast multiplier switch.

■ Example

N1 G28 X40.0
; The X-axis returns to the reference point and the intermediate point (X40.0) is stored

N2 G28 Y60.0
; The Y-axis returns to the reference point and the intermediate point (Y60.0) is stored

N3 G29 X10.0 Y20.0
; X-axis and Y-axis return from the reference point with an intermediate point of (X40.0,Y60.0)

2.1.9 Machine tool coordinate controller interpolation (G53)

■ Introduction to commands

A specific point at the machine tool used as a processing reference is called the machine tool zero point, the machine tool manufacturer for each machine tool set up machine tool zero point. The machine tool zero point is used as the origin to set up the coordinate controller called the machine tool coordinate controller.

The machine tool coordinate controller is established when the controller is powered up and a manual return to the reference point is performed, once established the machine tool coordinate controller remains unchanged until the power is removed.

When machine tool coordinate controller interpolation is commanded, the tool moves to the target position under the specified machine tool coordinate controller. The G53 command used to select the machine tool coordinate controller for interpolation is a non-modal G code, i.e. it is only valid for the program segment in which the machine tool coordinate controller interpolation is commanded.

When the tool is instructed to move to a special position on the machine tool (e.g. tool change position), the movement in the machine tool coordinate controller shall be programmed in G53.

■ Command Format

G53 IP_

IP_: Position command in the machine tool coordinate controller, the specified absolute position is used as the target position for G53 positioning.

■ Cautions

- The coordinates specified after G53 are always absolute coordinates, even in the incremental value mode (G91).
- Polar programming is not valid for G53, even if polar programming is valid (G16), the coordinates after G53 are always the absolute position in the machine tool's coordinate controller.
- G53 command speed designation: fast shift speed if currently in G00 mode, otherwise cutting feed speed (specified by F).

2.2 Fixed cycle commands

Fixed cycle commands combine a set of tool movements in a single command.

Fixed cycle commands make programming easy for the programmer. With fixed cycles, frequently used machining operations can be implemented in a single program segment using the G command, which would otherwise normally require multiple program segments to be written; in addition, fixed cycles shorten programs and save memory.

2.2.1 High-speed deep-hole machining cycle (G73)

■ Command introduction

The high speed deep hole machining cycle performs intermittent feeds along the Z axis to the bottom of the hole. When using this cycle, chips can be easily removed from the hole and a small setback value can be set, which allows efficient execution of the drilling.

Rotate the spindle with the auxiliary function (M code) before specifying G73. When the G73 code and the M code are specified in the same program segment, the M code is executed at the same time as the first positioning action, and then the controller processes the next drilling action. When the number of repetitions K is specified, the M code is only executed for the first hole and not for the second and subsequent holes.

The action sequence of the G73 is shown inFigure 2- 17 shown. The dashed line in the diagram indicates the rapid positioning, Q indicates the depth of each feed and D the value of each retraction.

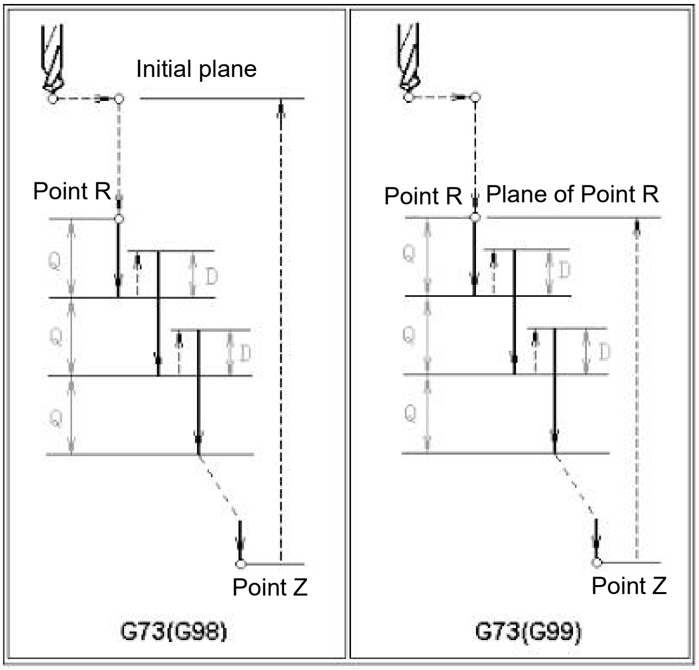


Figure 2- 17 G73 action sequence

Command Format

(G98/G99) G73 X_ Y_ Z_ R_ Q_ F_ K_

X_ Y_. Absolute position of the hole position in the absolute mode (G90), distance of the tool from the current position to the hole position in the incremental mode (G91).
Z_. Specifies the position of the bottom of the hole. The absolute position of the hole bottom in the Z-direction in the absolute mode (G90) and the distance from the hole bottom to the R-point in the incremental mode (G91).
R_. Specifies the position of the R point. The absolute position of the R point in the z-direction for the absolute value mode (G90) and the distance from the R point to the initial plane for the incremental value mode (G91).
Q_. Depth of cut at each cutting feed, the last depth of cut near the bottom of the hole may be less than this value. Q values are the same for positive and negative.
F_. Drill feed speed.
K_. Number of repetitions (can be omitted if K=1)

Cautions

- The drilling axis must be the Z axis.
- Point Z must be below the plane of point R, otherwise the program alarms.
- D: amount of retraction per upward movement; specified by channel parameter 1000310.
- The G73 command data is stored as modal data and the same may be omitted.

Example

M3 S2000 ;Spindle starts to rotate
G90 G99 G73 x300 y-250 z-150 r-100 q15 k5 f120
;position, drill 1, then return to point R
Y-550 ;position, drill 2, then return to point R
Y-750 ;position, drill 3, then return to point R
X1000 ;position, drill 4, then return to point R
Y-550 ;position, drill 5, then return to point R
G98 Y-750 ;Positioning, drilling 6, then return to initial position plane
G80 G28 G91 X0 Y0 Z0 ;Drilling cancelled, return to reference point
M5 ; Spindle stops rotating

2.2.2 Reverse tapping cycle (G74)

■ Introduction to commands

The G74 is similar to the G84, except that the G74 taps a counter-thread (the spindle starts reversing at the R point, stops at the bottom of the hole, and then exits in forward rotation).

In the rigid tapping method a spindle motor is used to control the tapping process. The spindle motor works in the same way as a servo motor. The tapping is performed by interpolation between the tapping axis and the spindle. In the rigid tapping method, the spindle feeds a distance of one thread guide per revolution along the tapping axis, and this feed relationship remains unchanged even during acceleration and deceleration.

The action of the G74 is shown in Figure 2- 18 shown. After positioning along the X and Y axes and moving quickly to point R, the spindle then reverses and performs a tapping from point R to point Z. When tapping is complete the spindle stops and executes a pause, then the spindle turns forward, the tool returns to point R, the spindle stops and, in the case of the G98 method , also moves quickly to the initial position.

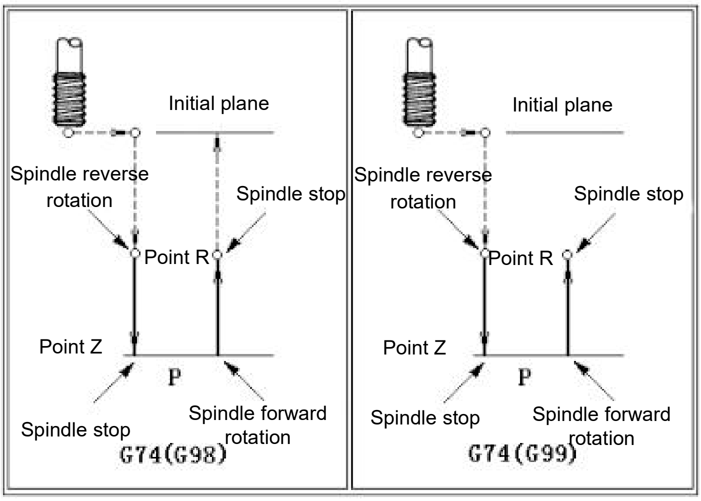


Figure 2- 18 G74 action sequence

■ Command Format

G74 X_ Y_ Z_ R_ Q_ P_ F_ S_ K_

X_ Y_. In the absolute value mode (G90), the absolute position of the hole is specified; in the incremental value mode (G91), the distance of the tool from the current position to the hole position is

specified.
Z_. For the absolute value method (G90), specifies the absolute position of the bottom of the hole; for the incremental value method (G91), specifies the distance from the bottom of the hole to the R point.
R_. For the absolute value method (G90), specifies the absolute position of the R point; for the incremental value method (G91), specifies the distance of the R point from the initial plane.
Q_: depth of cut at each cutting feed, the last depth of cut near the bottom of the hole may be less than this value. positive or negative Q values are the same.
P_. Specifies the pause time in milliseconds when tapping to the bottom of the hole.
F_. Specifies the thread guide.
S_. Specifies the spindle speed for tapping.
K_. Number of repetitions (can be omitted if K=1)

■ Cautions

- The tapping axis must be the Z axis.
- Point Z must be below the plane of point R, otherwise the program alarms.
- The G74 command data is stored as modal data and the same may be omitted.
- For rigid tapping, F is specified in the program as the thread guide and the Feed speed along the tapping axis is calculated by the following equation.

Feed speed = spindle speed x thread lead

2.2.3 Fine boring cycle (G76)

■ Introduction to commands

For fine boring, the spindle moves in the opposite direction of the tool tip after an orientation stop at the bottom of the hole, and then quickly retires the tool. The amount of tool tip reverse displacement is specified by Q and the direction of tool tip offset is determined by filling in the channel parameters.

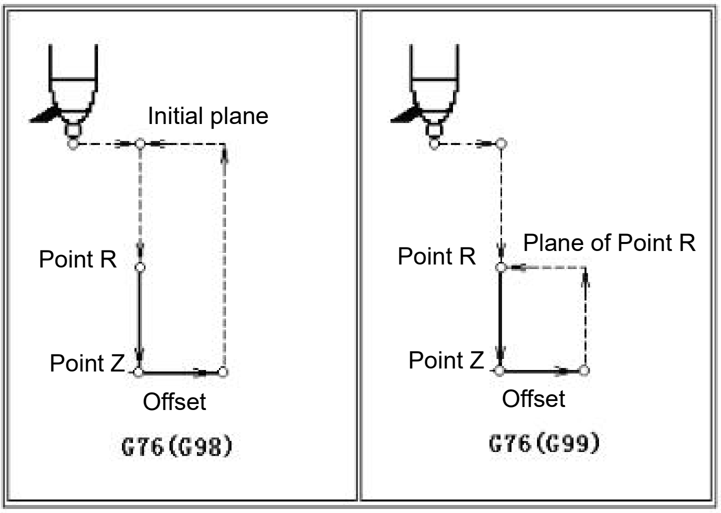


Figure 2- 19 G76 action sequence

■ Command Format

(G98/G99) G76 X_ Y_ Z_ R_ Q_ P_ F_ K_

X_ Y_. Hole position data, absolute position of the hole in the absolute value mode (G90), distance of the tool from the current position to the hole in the incremental value mode (G91).

Z_. Specifies the position of the bottom of the hole. The absolute position of the hole bottom in the Z-direction in the absolute mode (G90) and the distance from the hole bottom to the R-point in the incremental mode (G91).

R_. Specifies the position of the R point. The absolute position of the R point in the z-direction for the absolute value mode (G90) and the distance from the R point to the initial plane for the incremental value mode (G91).

Q_. Tool offset.

P_. Pause time at the bottom of the hole (in milliseconds).

F_. Cutting Feed speed.

K_. Number of repetitions (can be omitted if K=1)

■ Cautions

- The drilling axis must be the Z axis.
- Point Z must be below the plane of point R, otherwise the program alarms.
- The offset Q value at the bottom of the hole is the modal value held in the fixed cycle and must be carefully specified, as it is also used as the depth of cut for G73 and G83.
- The direction of tool offset is specified by the channel parameter 1000315.
- The G76 command data is stored as modal data and the same may be omitted.

2.2.4 Drilling cycle (G81)

■ Command introduction

This cycle is used for normal drilling. The cutting feed is executed to the bottom of the hole and then the tool moves back quickly from the bottom of the hole.

After positioning along the X and Y axes, a rapid movement is made to point R. From point R to point Z, drilling is performed and then the tool moves back quickly. When the G81 command and the M code are specified in the same program segment, the M code is executed at the same time as the first positioning movement and the controller then processes the next movement.

When the number of repetitions K is specified, the M code is only executed before the first hole is drilled, and no further M codes are executed when subsequent holes are drilled.

The action sequence of G81 is shown inFigure 2- 20 shown. The dotted line in the diagram indicates the fast positioning.

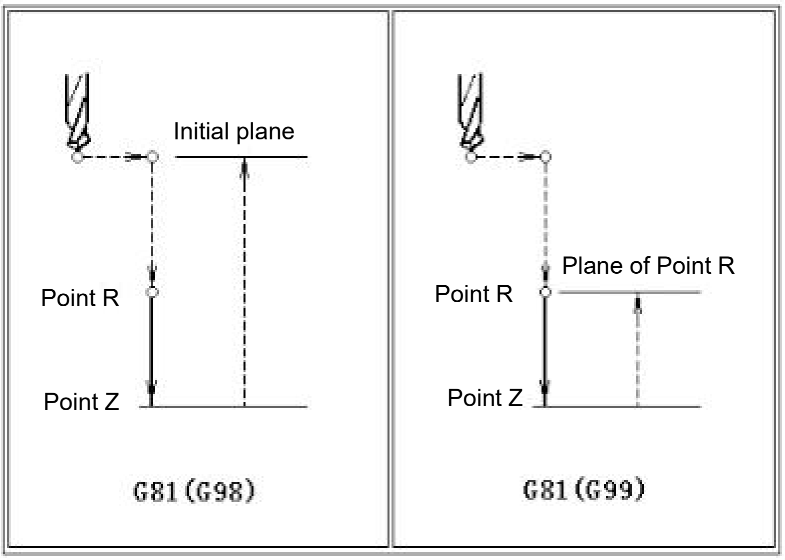


Figure 2- 20 G81 action sequence

■ Command Format

(G98/G99) G81 X_ Y_ Z_ R_ F_ K_

X_ Y_. Absolute position of the hole position in the absolute mode (G90); distance of the tool from the current position to the hole position in the incremental mode (G91).
Z_. Specifies the position of the bottom of the hole. The absolute position of the bottom of the hole in the Z direction in the absolute mode (G90); the distance from the bottom of the hole to the R point in the incremental mode (G91).
R_. Specifies the position of the R point. The absolute position of the R point in the z-direction for the absolute value mode (G90); the distance from the R point to the initial plane for the incremental value mode (G91).
F_. Cutting Feed speed.
K_. Number of repetitions (can be omitted if K=1)

■ Cautions

- The drilling axis must be the Z axis.
- the Z point must be below the R plane, otherwise the procedure alarms.
- The G81 command data is stored as modal data and the same may be omitted.

■ Example

M3 S2000 ;Spindle starts to rotate
G90 G99 G81 x300 y-250 z-150 r-100 f120
;Position, drill 1 and return to point R
Y-550 ;Position, drill 2, then return to point R
Y-750 ;Position, drill 3, then return to point R
X1000 Positioning, drilling 4, then return to point R
Y-550 ;Position, drill 5, then return to point R
G98 Y-750 ;Positioning, drilling 6, then return to initial position plane
G80 G28 G91 x0 y0 z0;Drill cancelled, return to reference point
M5; Spindle stops rotating

2.2.5 Drilling cycle with stop (G82)

■ Introduction to commands

This cycle is used for normal drilling. After positioning along the X and Y axes, the tool moves quickly to point R and then drills from point R to point Z. When the bottom of the hole is reached, a pause is executed and the tool moves quickly back.

Rotate the spindle with the auxiliary function M code before specifying G82. When the G82 command and the M code are specified in the same program segment, the M code is executed at the same time as the first positioning action, and then the controller processes the next drilling action. When the number of repetitions K is specified, the M code is only executed when the first hole is machine toold and no further M codes are executed for subsequent holes.

When tool length offset G43, G44 or G49 is specified in the fixed cycle, the offset is added while positioning to the R point.

The action sequence of G83 is shown inFigure 2- 21 shown.

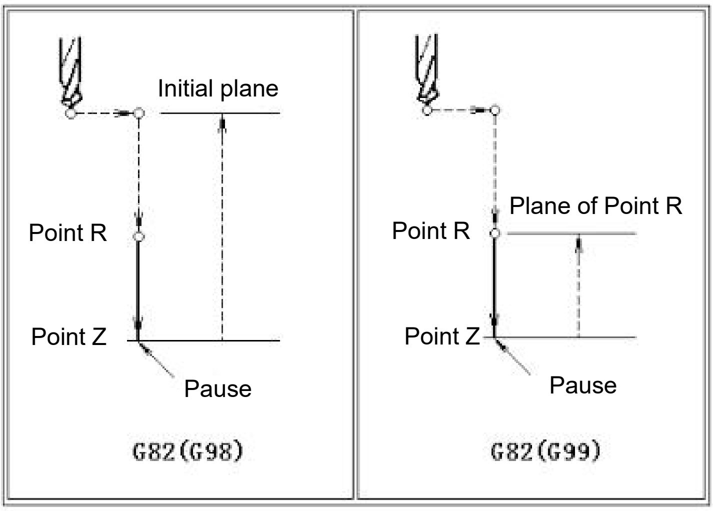


Figure 2-21 G82 action sequence

■ Command Format

(G98/G99) G82 X_ Y_ Z_ R_ P_ F_ K_

X_ Y_. In the absolute value mode (G90), the absolute position of the hole is specified; in the incremental value mode (G91), the distance of the tool from the current position to the hole position is specified.
Z_. For the absolute value method (G90), specifies the absolute position of the bottom of the hole; for the incremental value method (G91), specifies the distance from the bottom of the hole to the R point.
R_. For the absolute value method (G90), specifies the absolute position of the R point; for the incremental value method (G91), specifies the distance of the R point from the initial plane.
P_. Specifies the pause time (in milliseconds) at the bottom of the hole.
F_. Specifies the cutting feed speed.
K_. Number of repetitions (can be omitted if K=1)

■ Cautions

- The drilling axis must be the Z axis.
- Point Z must be below the plane of point R, otherwise the program alarms.
- The G82 command data is stored as modal data and the same may be omitted.

■ Example

M3 S2000;Spindle starts to rotate
G90 G99 G82 x300. y-250 z-150 r-100 p1000 f120
;Position, drill 1 and return to point R
Y-550 ;Position, drill 2, then return to point R
Y-750 ;Positioning, drilling 3, then return to point R
X1000 Positioning, drilling 4, then return to point R
Y-550;Position, drill 5, then return to point R
G98 Y-750;Positioning, drilling 6, then return to initial plane
G80 G28 G91 X0 Y0 Z0 ;Drilling cancelled, return to reference point
M5 Spindle stops rotating

2.2.6 Deep hole machining cycle (G83)

■ Introduction to commands

This cycle performs deep hole drilling. Intermittent cutting feeds are executed to the bottom of the hole and chips are removed from the hole during drilling.

When the G83 code and the M code are specified in the same segment, the M code is executed at the same time as the first positioning action, and then the controller processes the next drilling action. When the number of repetitions K is specified, the M code is only executed before the first hole is drilled and no further M codes are executed for subsequent holes.

Q indicates the depth of each cutting feed, which must be specified in increments. After each cutting depth Q, the R point plane is quickly returned to remove the chips, then quickly positioned to point D in the previous drilling position, after which the cutting feed is executed with a feed depth Q.

D indicates the positioning height for each cutting feed and point D is the point at which the height from the last drilling end position is D.

When specifying a tool length offset in a fixed cycle, the offset is added while positioning to the R point.

The action sequence of G83 is shown inFigure 2- 22 shown.

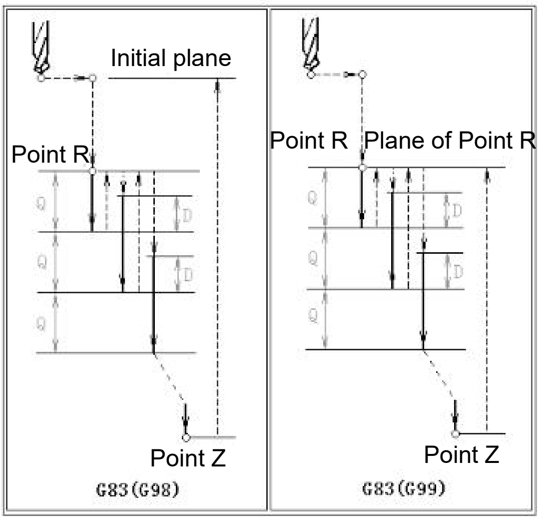


Figure 2- 22 G83 action sequence

■ Command Format

(G98/G99) G83 X_ Y_ Z_ R_ Q_ F_ K_

X_ Y_. In the absolute value mode (G90), the absolute position of the hole is specified; in the incremental value mode (G91), the distance of the tool from the current position to the hole position is specified.
Z_. For the absolute value method (G90), specifies the absolute position of the bottom of the hole; for the incremental value method (G91), specifies the distance from the bottom of the hole to the R point.
R_. For the absolute value method (G90), specifies the absolute position of the R point; for the incremental value method (G91), specifies the distance from the R point to the initial plane.
Q_. The depth of cut at each cutting feed, the last depth of cut near the bottom of the hole may be less than this value. positive or negative Q values are the same.
F_. Specifies the cutting feed speed.
K_. Number of repetitions (can be omitted if K=1)

■ Cautions

- The drilling axis must be the Z axis.
- Point Z must be below the plane of point R, otherwise the program alarms.
- D: Positioning height (distance from last drilling position); specified by channel parameter 1000320.
- The G83 command data is stored as modal data and the same may be omitted.

■ Example

```
M3 S2000            ;Spindle starts to rotate
G90 G99 G83 x300 y-250 z-150 r-100 q15 k3 f120
                    ;Position, drill 1 and return to point R
Y-550               ;Position, drill 2, then return to point R
Y-750               ;Position, drill 3, then return to point R
X1000               Positioning, drilling 4, then return to point R
Y-550               ;Position, drill 5, then return to point R
G98 Y-750           ;Positioning, drilling 6, then return to initial plane
G80 G28 G91 x0 y0 z0        ;Drill cancelled, return to reference point
M5                  ;Spindle stops rotating
```

2.2.7 Tapping cycle (G84)

■ Command introduction

In the rigid tapping method a spindle motor is used to control the tapping process. The spindle motor works in the same way as a servo motor. The tapping is performed by interpolation between the tapping axis and the spindle. In the rigid tapping method, the spindle feeds a distance of one thread guide per revolution along the tapping axis, and this feed relationship remains unchanged even during acceleration and deceleration.

The action of the rigid tapping is shown in Figure 2-23 shown. After positioning along the X and Y axes and moving quickly to point R, the spindle then turns positively and performs a tapping from point R to point Z. When tapping is complete the spindle stops and executes a pause, then the spindle rotates in the opposite direction, the tool returns to point R, the spindle stops and, in the case of the G98 method , also moves quickly to the initial position.

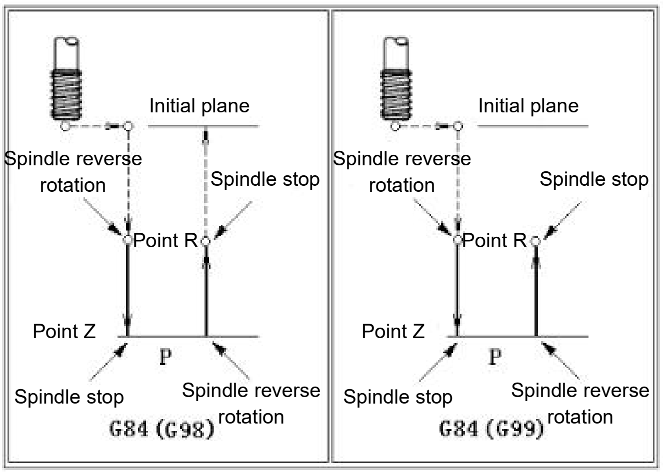


Figure 2-23 Rigid tapping (G84) action sequence

■ Command Format

G84 X_ Y_ Z_ R_ Q_ P_ F_ S_ K_

X_ Y_. In the absolute value mode (G90), the absolute position of the hole is specified; in the incremental value mode (G91), the distance of the tool from the current position to the hole position is specified.
Z_. For the absolute value method (G90), specifies the absolute position of the bottom of the hole; for the incremental value method (G91), specifies the distance from the bottom of the hole to the R point.
R_. For the absolute value method (G90), specifies the absolute position of the R point; for the incremental value method (G91), specifies the distance from the R point to the initial plane.
Q_: depth of cut at each cutting feed, the last depth of cut near the bottom of the hole may be less than this value. positive or negative Q values are the same.
P_. Specifies the pause time in milliseconds when tapping to the bottom of the hole.
F_. Specifies the thread guide.
S_. Specifies the spindle speed for tapping.
K_. Number of repetitions (can be omitted if K=1)

■ Cautions

- The tapping axis must be the Z axis.
- the Z point must be below the R plane, otherwise the procedure alarms.
- The G84 command data is stored as modal data and the same may be omitted.
- In rigid tapping, the Feed speed along the tapping axis is calculated by the following equation.

  Feed speed = spindle speed x thread lead

2.2.8 Boring cycle (G85)

■ Introduction to commands

After positioning along the X and Y axes, move quickly to point R. Then, from point R to point Z, execute the boring, reach the bottom of the hole and execute the cutting feed until returning to point R. After reaching point R, if the G98 method is used, return quickly to the initial plane.

Rotate the spindle with the auxiliary function M code before specifying G85. When the G85 command and the M code are specified in the same program segment, the M code is executed at the same time as the first positioning action, and then the controller processes the next boring action. When the number of repetitions K is specified, the M code is only executed for the first bore and no further M codes are executed for subsequent bores.

When tool length offsets G43, G44 or G49 are specified in the fixed cycle, the offsets are added while positioning to the R point.

The action sequence of G85 is shown inFigure 2- 24 shown.

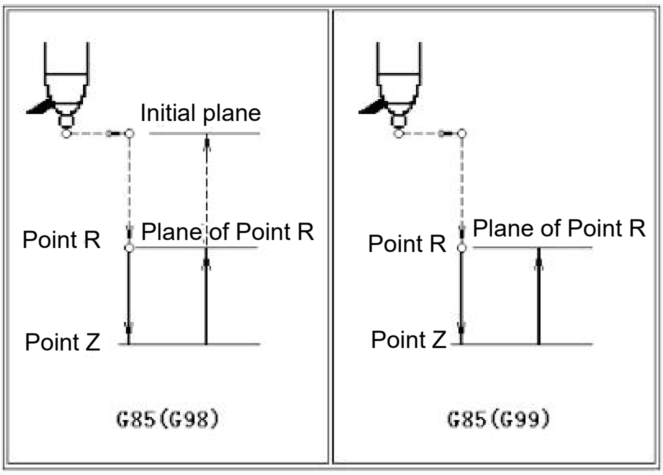


Figure 2- 24 G85 action sequence

Command Format

(G98/G99) G85 X_ Y_ Z_ R_ P_F_ K_

X_ Y_. In the absolute value mode (G90), the absolute position of the hole is specified; in the incremental value mode (G91), the distance of the tool from the current position to the hole position is specified.
Z_. For the absolute value method (G90), specifies the absolute position of the bottom of the hole; for the incremental value method (G91), specifies the distance from the bottom of the hole to the R point.
R_. For the absolute value method (G90), specifies the absolute position of the R point; for the incremental value method (G91), specifies the distance of the R point from the initial plane.
P_. Specifies the pause time (in milliseconds) at the bottom of the hole.
F_. Specifies the cutting feed speed.
K_. Number of repetitions (can be omitted if K=1)

Cautions

- The drilling axis must be the Z axis.
- Point Z must be below the plane of point R, otherwise the program alarms.
- The G85 command data is stored as modal data and the same may be omitted.

Example

M3 S1000 ;Spindle starts to rotate
G90 G99 G85 x300 y-250 z-150 r-120 p3000 f120
Positioning, boring 1, pause at the bottom of the hole for 3 seconds, then return to point R
Y-550 Positioning, boring 2, 3 seconds pause at the bottom of the hole, then return to point R
Y-750 Positioning, boring 3, pause at the bottom of the hole for 3 seconds, then return to point R
X1000 Position, bore 4, pause at bottom of bore for 3 seconds, then return to point R
Y-550 Positioning, bore 5, pause at the bottom of the hole for 3 seconds, then return to point R
G98 Y-750 Positioning, bore 6, pause at the bottom of the hole for 3 seconds, then return to the initial plane
G80 G28 G91 X0 Y0 Z0 ;Cancel boring, return to reference point
M5 ;Spindle stops rotating

2.2.9 Boring cycle (G86)

■ Command introduction

The G86 performs the same action as the G81, but stops at the bottom of the bore spindle and then quickly retreats, mainly for bore machining where accuracy is not too critical.

■ Command Format

(G98/G99) G86 X_ Y_ Z_ R_ F_ K_

X_ Y_. Absolute position of the hole position in the absolute mode (G90); distance of the tool from the current position to the hole position in the incremental mode (G91).
Z_. Specifies the position of the bottom of the hole. The absolute position of the hole bottom in the Z-direction in the absolute mode (G90); the distance from the hole bottom to the R-point in the incremental mode (G91).
R_. Specifies the position of the R point. The absolute position of the R point in the z-direction for the absolute value mode (G90) and the distance from the R point to the initial plane for the incremental value mode (G91).
F_. Cutting Feed speed.
K_. Number of repetitions (can be omitted if K=1)

2.2.10 Reverse Boring Cycle (G87)

This command is generally used to bore a hole that is small at the bottom and large at the top, with the bottom Z point of the hole generally above the reference point R, unlike other commands.

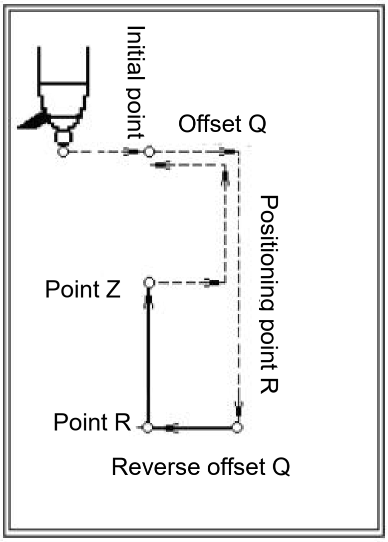


Figure 2- 25 G87 action sequence

■ Command Format

(G98/G99) G87 X_ Y_ Z_ R_ Q_ P_ F_ K_

X_ Y_. Hole position data, absolute position of the hole in the absolute value mode (G90), distance of the tool from the current position to the hole in the incremental value mode (G91).
Z_. Specifies the position of the bottom of the hole. The absolute position of the hole bottom in the Z-direction in the absolute mode (G90) and the distance from the hole bottom to the R-point in the incremental mode (G91).
R_. Specifies the position of the R point. The absolute position of the R point in the z-direction in the absolute mode (G90) and the distance from the R point to the initial plane in the incremental mode (G91).
Q_. Tool offset.
P_. Pause time at the bottom of the hole (in milliseconds).
F_. Cutting Feed speed.
K_. Number of repetitions (can be omitted if K=1)

Cautions

- The drilling axis must be the Z axis.
- Point Z must be above the plane of point R, otherwise the program alarms.
- The offset Q value at the bottom of the hole is the modal value held in the fixed cycle and must be carefully specified, as it is also used as the depth of cut for G73 and G83.
- The direction of tool offset is specified by the channel parameter 1000315.
- The G87 command data is stored as modal data and the same may be omitted.
- The G87 command can only be used with G98 and is not valid even if G99 is written.

2.2.11 Boring cycle (G89)

■ Command introduction

The cycle is almost identical to G86, the difference being that the cycle executes a pause at the bottom of the hole. The spindle is rotated with the auxiliary function M code prior to specifying G89. When the G89 command and the M code are specified in the same program segment, the M code is executed at the same time as the first positioning action and the controller then processes the next boring action. When the number of repetitions L is specified only the M code is executed for the first bore and no further M codes are executed for subsequent bores.

When tool length offsets G43, G44 or G49 are specified in the fixed cycle, the offsets are added while positioning to the R point.

The action sequence of G89 is shown inFigure 2-26 shown.

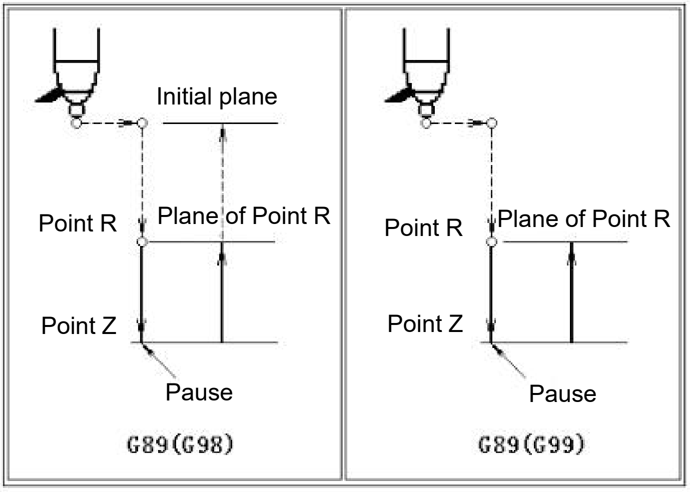


Figure 2-26 G89 action sequence

■ Command Format

(G98/G99) G89 X_ Y_ Z_ R_ P_ F_ K_

X_ Y_. In the absolute value mode (G90), the absolute position of the hole is specified; in the incremental value mode (G91), the distance of the tool from the current position to the hole position is specified.
Z_. For the absolute value method (G90), specifies the absolute position of the bottom of the hole; for the incremental value method (G91), specifies the distance from the bottom of the hole to the R point.

R_. For the absolute value method (G90), specifies the absolute position of the R point; for the incremental value method (G91), specifies the distance of the R point from the initial plane.
P_. Pause time at the bottom of the hole (milliseconds).
F_. Specifies the cutting feed speed.
K_. Number of repetitions (can be omitted if K=1)

■ Example

M3 S1000 ;Spindle starts to rotate
G90 G99 G89 x300 y-250 z-150 r-120 p1000 f120
; Position, bore 1 hole, pause at bottom of hole for 1 second, then return to point R
Y-550 ; Positioning, boring 2 holes, pause at the bottom of the hole for 1 second, then return to point R
Y-750 ; Position, bore 3 holes, pause at bottom of hole for 1 second, then return to point R
X1000 ; position, bore 4 holes, pause at bottom of hole for 1 second, then return to point R
Y-550 Positioning, boring 5 holes, pause at the bottom of the hole for 1 second, then return to point R
G98 Y-750 Positioning, boring 6 holes, 1 second pause at the bottom of the hole, then return to the initial plane
G80 G28 G91 X0 Y0 Z0 ;Cancel boring, return to reference point
M5 ;Spindle stops rotating

■ Cautions

- The drilling axis must be the Z axis.
- Point Z must be below the plane of point R, otherwise the program alarms.
- The G89 command data is stored as modal data and the same may be omitted.

2.2.12 Rectangular continuous drilling (Gxx.1/Gxx.2)

■ Introduction to commands

This cycle is used as a rectangular continuous drilling. Depending on the number of holes specified for each side, continuous drilling is carried out on each side of the rectangle.

The drilling methods Gxx include G73, G74, G81, G83 and G84.

■ Command Format

(G98/G99) Gxx.1/Gxx.2 X_ Y_ Z_ R_ A_ B_ J_ Q_ P_ F_ S_ K_

Gxx.1. Clockwise rectangular continuous drilling.
Gxx.2. Counterclockwise rectangular continuous drilling.
X_ Y_. Coordinates of the end point of the first rectangular edge.
Z_. Specifies the position of the bottom of the hole.
R_. Specifies the position of the R point.
A_: Number of holes on sides 1 and 3.
B_: Number of holes on sides 2 and 4.
J_: Length of the 2nd edge.
Q_: depth of cut at each cutting feed, the last depth of cut near the bottom of the hole may be less than this value. positive or negative Q values are the same.
P_. Specifies the pause time in milliseconds when tapping to the bottom of the hole.
F_. Cutting feed speed in non-tapping commands.
Thread guide in tapping commands.
S_: Specifies the spindle speed for tapping.
K_. Number of repetitions (can be omitted if K=1)

■ Cautions

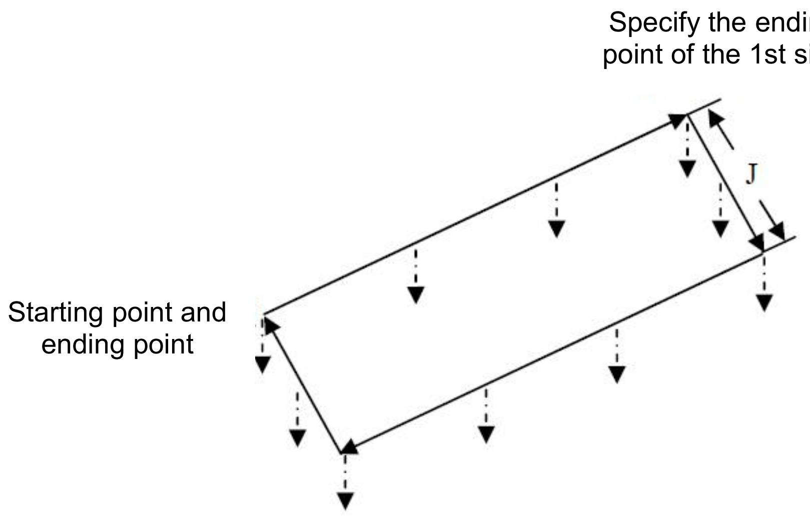
- The drilling axis must be the Z axis.
- the Z point must be below the R plane, otherwise the procedure alarms.
- The valid range of command input for each side drilling number A and B is 0~9999; when the command is negative, it is treated as absolute value. For command decimals, the decimal part is rounded off; if there is no command A or B then the default is 0.

- The rectangle is determined by the current start point, the end point of the 1st side and the length of the 2nd side; if the end point of the 1st side is not specified, the default is equal to the current start point and an alarm is generated; if the length of the 2nd side is not specified (i.e. J is not specified) an alarm is generated. When negative values are commanded, they are treated as absolute values.
- During continuous drilling, no holes are drilled when the number of holes is 0.
- Command words A and B are parameters of a Fixed cycle command and will not move the 4th and 5th axes.
- Gxx.1/Gxx.2, A, B and J are all non-modal commands and are only valid for the current program segment.

■ Example

Rectangular trajectory drilling with end coordinates X90, Y40 on side 1; length 20mm on side 2; machining using G81 drilling method requiring 3 holes on each of sides 1 and 3 and 2 holes on each of sides 2 and 4; hole depth 25mm.

Programming is as follows.
M03 S1500.
G90 G17 g0 x0 y0 z25.
G81.1 x90 y40 r5 z-25 a3 b2 j20 f800.
G80 g0 x100 y100.
M05.



2.2.13 Continuous drilling of circular arcs (Gxx.3/Gxx.4)

■ Introduction to commands

This cycle is used as continuous drilling of circular arcs. Depending on the number of holes specified, continuous drilling is carried out on the specified arc.

The drilling methods Gxx include G73, G74, G81, G83 and G84.

■ Command Format

(G98/G99) Gxx.3/Gxx.4 X_ Y_ Z_ R_ B_(I_ J_) C_ Q_ P_ F_ S_ K_

Gxx.3. Continuous drilling in clockwise circular arc.
Gxx.4. Counterclockwise circular continuous drilling.
X_ Y_. Coordinates of the end point of the arc, fixed to the G17 plane.
Z_. Specifies the position of the bottom of the hole.
R_. Specifies the position of the R point.
B_: The radius of the arc, specified as a superior arc when negative.
I_ J_: When no B value is specified, same as I and J definitions for G02/G03.
C_: Number of holes drilled.
Q_: depth of cut at each cutting feed, the last depth of cut near the bottom of the hole may be less than this value. positive or negative Q values are the same.
P_. Specifies the pause time in milliseconds when tapping to the bottom of the hole.
F_. Cutting feed speed in non-tapping commands.
Thread guide in tapping commands.
S_: Specifies the spindle speed for tapping.
K_. Number of repetitions (can be omitted if K=1)

■ Cautions

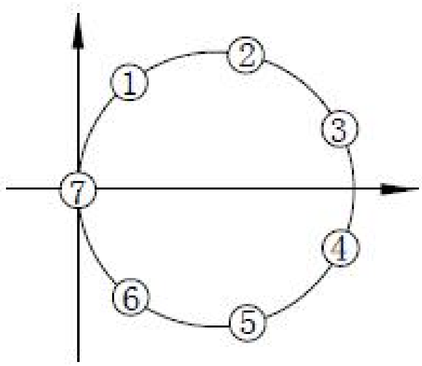
- The drilling axis must be the Z axis.
- the Z point must be below the R plane, otherwise the procedure alarms.
- During continuous drilling, when the start and end points are the same and are programmed with I and J, then the full circle is drilled on.

- The valid input range of the command for drilling number C is 0 to 9999; when the command is negative, it is treated as an absolute value. The command is decimal and the decimal part is rounded off.
- When no C is commanded or when C is commanded to be 0, go straight to the end without drilling.
- Command words B and C are parameters of the Fixed cycle command and will not move the 4th and 5th axes.
- Gxx.3/Gxx.4, B, C, I and J are all non-modal commands and are only valid for the current program segment.

■ Example

G84 tapping of 7 holes on a full circle, starting and ending at the origin of the coordinates, with a radius of 50; hole depth 50.

Programming is as follows:
G90 G17 g0 x0 y0 z0.
G98 G84.3 r0 z-50 i50 j0 c7 q5 f1 s1000.
M30.



2.2.14 Checkerboard continuous drilling (Gxx.5)

■ Introduction to commands

This cycle is used as continuous drilling on a chessboard. Starting from the starting point, holes are drilled at fixed intervals in the parallel direction of the X and Y axes, according to the specified direction, with a machining path like a checkerboard grid.

The drilling method Gxx includes G73, G74, G81, G83 and G84.

■ Command Format

(G98/G99) Gxx.5 X_ Y_ Z_ R_ I_ J_A_ B_ Q_ P_ F_ S_ K_

Gxx.5. Checkerboard continuous drilling.
X_ Y_. Start and end coordinates.
Z_. Specifies the position of the bottom of the hole.
R_. Specifies the location of the R point.
I_: Specify the interval distance for drilling in the X-axis direction, if a negative value is specified, it means the negative direction of the X-axis.
A_: Number of holes to be machine toold in the X-axis.
J_: Specify the interval distance for drilling in the Y-axis direction, if a negative value is specified, it means the negative direction of the Y-axis.
B_: Number of holes to be machine toold in the Y-axis.
Q_: depth of cut at each cutting feed, the last depth of cut near the bottom of the hole may be less than this value. positive or negative Q values are the same.
P_. Specifies the pause time in milliseconds when tapping to the bottom of the hole.
F_. Cutting feed speed in non-tapping commands.
Thread guide in tapping commands.
S_: Specifies the spindle speed for tapping.
K_. Number of repetitions (can be omitted if K=1)

■ Cautions

- The drilling axis must be the Z axis.
- the Z point must be below the R plane, otherwise the procedure alarms.
- The valid range of command input for the number of holes drilled on each side, A and B, is 0 to 9999; when the command is negative, it is treated as an absolute value. If the command is decimal, the decimal part is rounded off; if there is no command A or B then the default is 0. If A and B are specified as 0, no checkerboard drilling is carried out. If

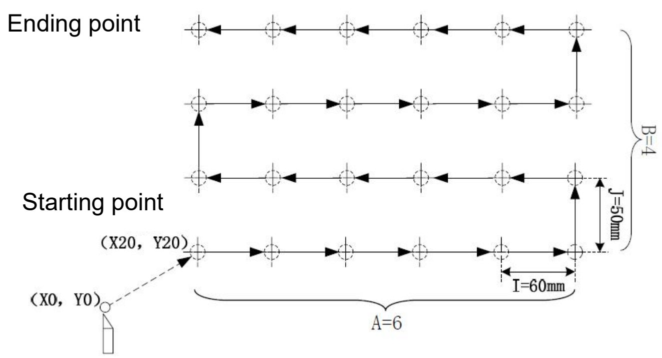
only either A or B is specified, then only straight-line method processing is carried out.

- If an I value is not specified when specifying the number of holes to be punched in the X-axis direction A, the controller will generate an alarm: "I value not specified". If the A and I values are specified and the B value is not also specified, the controller will perform straight punching in the X-axis direction.
- If a J value is not specified when specifying the number of holes to be punched in the Y-axis direction, B, the controller will generate an alarm: "J value not specified". If the B and I values are specified and the A value is not also specified, the controller will perform straight punching in the Y-axis direction.
- Command words A and B are parameters of a Fixed cycle command and will not move the 4th and 5th axes.
- Gxx.5, I, J, A and B are all non-modal commands and are only valid for the current program segment.

■ Example

The need to machine tool 6*4 holes in the positive X and Y directions on a workpiece with a starting position of (X20, Y20) by means of deep hole drilling, requiring an interval of 60 mm in the X-axis direction and 50 mm in the Y-axis direction. the starting plane of machining is -2 mm, the depth is 8 mm and the depth of each cut is 4 mm.

Programming is as follows:
M03 S2000.
G90 G17 g0 x0 y0 z0.
G83.5 x20 y20 r-2 z-10 i60 a6 j50 b4 q4 f1000.
G80 G0 X0 Y0.
M05.



2.2.15 Fixed cycle cancellation (G80)

■ Introduction to commands

All fixed cycles are cancelled and the normal operation is performed. r-point and z-point are also cancelled. This means that in the incremental mode (G91) R=0 and Z=0 and the other drilling data are also cancelled and cleared.

■ Command Format

G80

■ Example

M3 S1000 ;Spindle starts to rotate
G90 G99 G81 x300 y-250 z-150 r-120 f120.
Position, bore 1 hole and return to point R
Y-550 Position, bore 2 holes, then return to point R
Y-750 Position, bore 3 holes, then return to point R
X1000 Position, bore 4 holes, then return to point R
Y-550 Position, bore 5 holes, then return to point R
G98 Y-750 Positioning, boring 6 holes, then return to initial plane
G80 G28 G91 x0 y0 z0 ;Cancel fixed cycle, return to reference point
M5 ;Spindle stops rotating

3. Status commands

3.1 Absolute value and incremental value programming (G90/G91)

■ Introduction to commands

There are two ways of commanding the movement of the tool: the absolute and the incremental way. In the absolute method the absolute coordinate value of the end point of the tool movement is programmed; in the incremental method only the distance of the tool movement is programmed.

G90 and G91 are used to command the absolute or incremental value method respectively.

■ Command Format

G90/G91

(......)

■ Example

The toolpaths shown below can be programmed in absolute and incremental values as follows.

- Absolute approach
  G90 G00 X100 Y200

- Incremental value approach
  G91 G00 X-50 Y100

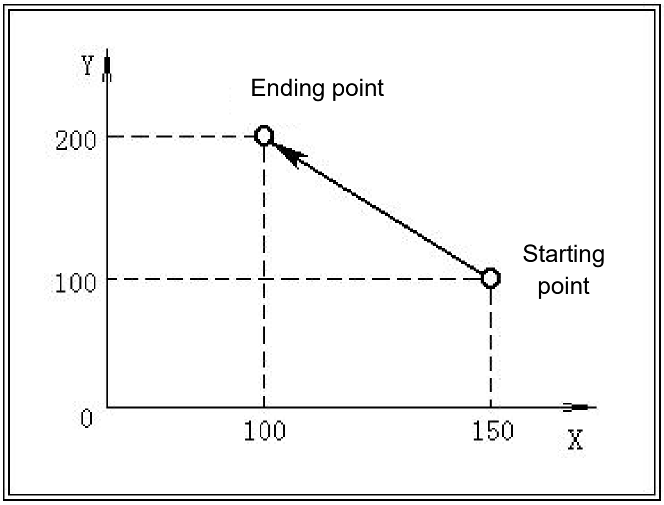


Figure 3- 1 Example of G90/G91

3.2 Polar coordinate commands (G15/G16)

■ Command introduction

When programming, coordinate values can be entered in polar radius and angle, with the positive direction of the angle being the counterclockwise turn in the positive direction of the 1st axis of the selected plane and the negative direction being the clockwise turn. Both radius and angle can be specified using the absolute value command (G90) or the incremental value command (G91).

There are two ways to set the zero point of a polar coordinate controller.

- **Set the zero point of the workpiece coordinate controller as the origin of the polar coordinate controller**

When specifying the radius (the distance between the zero point of the polar coordinate controller and the command position) in absolute terms (G90), the zero point of the workpiece coordinate controller is set to the origin of the polar coordinate controller. This is shown in the diagram below.

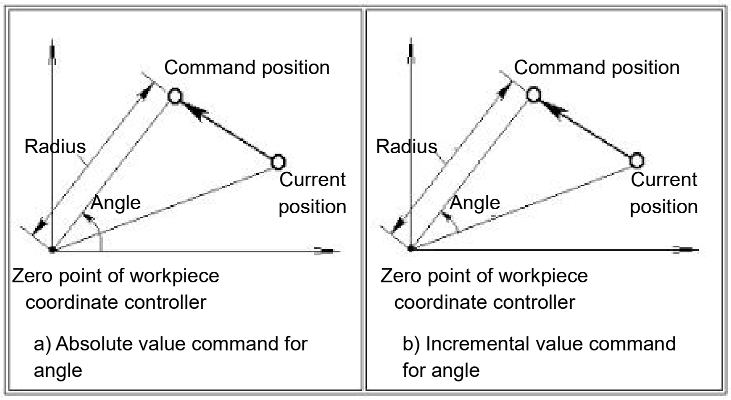


Figure 3-2 Setting the zero point of the workpiece coordinate controller as the zero point of the polar coordinate controller

- **Set the current position as the origin of the polar coordinate controller**

When specifying the radius (the distance between the current position and the command position) in the incremental value method (G91), the current position is set to the origin of the polar coordinate controller. This is shown in the diagram below.

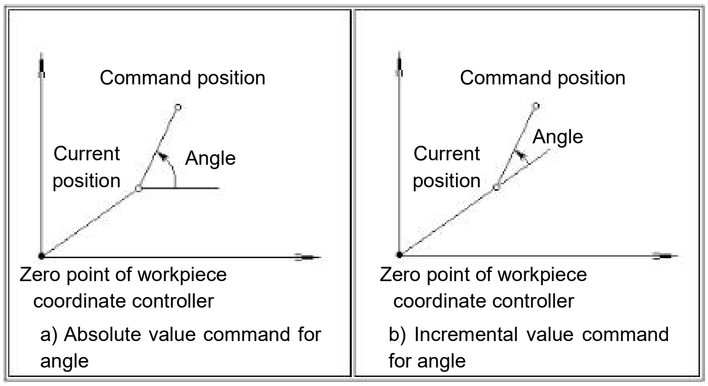


Figure 3-3 Set the current position as the origin of the polar coordinate controller

■ Command Format

(G17/G18/G19) (G90/G91) G16

......

G15

G17/G18/G19: Select the plane in which the polar coordinate controller is positioned.

G17 - XY plane. radius specified on the X axis, angle specified on the Y axis.

G18 - ZX plane. radius specified on the Z axis, angle specified on the X axis.

G19-YZ plane. Y-axis specifies radius, Z-axis specifies angle.

G90/G91. G90 designates the zero point of the workpiece coordinate controller as the origin of the polar coordinate controller, from which the radius is measured.

G91 specifies the current position as the origin of the polar coordinate controller from which the radius is measured.

G16. Valid for programming in polar coordinates.

.... Polar coordinate programming commands.

G15. Polar coordinate programming cancelled.

■ Example (bolt hole round)

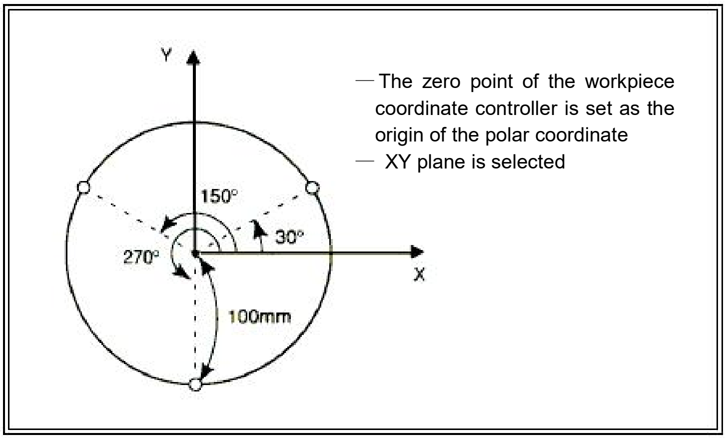


Figure 3-4 Example of polar coordinate programming

● Specifying angles and radii with absolute value commands

N1 G17 G90 G16 ; Specify polar programming, select the XY plane and set the zero point of the workpiece coordinate controller as the origin of the polar controller.

N2 G81 X100.0 Y30.0 Z-20.0 R-5.0 F200.0
; Specify a distance of 100mm and an angle of 30 degrees
N3 Y150.0 ;Specify a distance of 100mm and an angle of 150 degrees
N4 Y270.0 ; Specify a distance of 100mm and an angle of 270 degrees
N5 G15 G80 ;Cancellation of polar programming

● Specify the angle in increments and the pole diameter in absolute values

N1 G17 G90 G16 ; Specify polar programming, select the XY plane and set the zero point of the workpiece coordinate controller as the origin of the polar coordinates

N2 G81 X100.0 Y30.0 Z-20.0 R-5.0 F200.0
; Specify a distance of 100mm and an angle of 30 degrees
N3 G91 Y120.0 ;Specify a distance of 100mm and an angle increment of +120 degrees
N4 Y120.0 ;Specify a distance of 100mm and an angle increment of +120 degrees
N5 G15 G80 ;Cancellation of polar programming

3.3 Scaling (G50/G51)

■ Introduction to commands

When scaling is performed the programmed toolpath is scaled up or down by a given scale factor (scaling). The dimensions specified with X_, Y_ and Z_ can be scaled up and down by the same or different proportions.

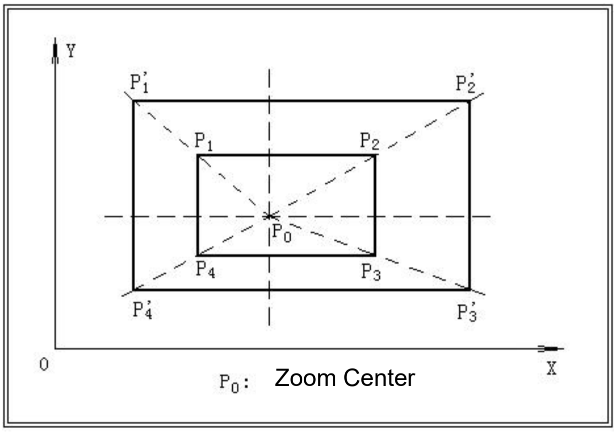


Figure 3- 5 Scaling P_1 P_2 P_3 P_4 →P_1' P_2' P_3' P_4'

● All axes scaled to the same scale

When P_ is specified in the proportional scaling command (G51), all axes are scaled up or down by the same proportion (the scale factor is specified by the P command parameter).

● Each axis scaled to a different scale

When I_, J_ and K_ are specified in the scale scaling command (G51), each axis is scaled at a different rate.

- I: X-axis scaling
- J: Y-axis scaling
- K: Z-axis scaling

● Mirroring function when scaling

When each axis is scaled at different scales, if the scaling is negative, a point-symmetrical mirror image is formed with the point of symmetry being the centre of the scaling.

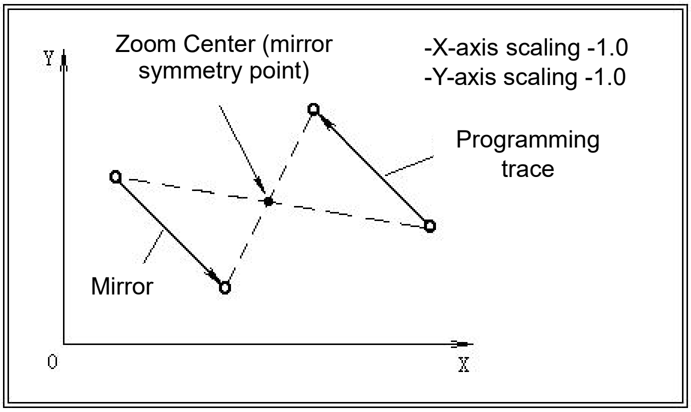


Figure 3- 6　Mirror function for scaling

■ Command Format

● All axes scaled to the same scale

G51 X_ Y_ Z_ P_
......
G50

X_ Y_ Z_.　　Specifies the coordinates of the scaled scaling centre, the scaling centre of an unspecified axis is the current position. This command always specifies the absolute position of the scaling centre in the workpiece coordinate controller, whether in absolute (G90) or incremental (G91) mode.
P_: Specifies the scaling of each axis. All axes (X, Y, Z) are scaled to this ratio.
......: Toolpath programming command.
G50: Proportional scaling cancelled.

- Each axis scaled to different proportions

G51 X_ Y_ Z_ I_ J_ K_
......
G50

X_ Y_ Z_. Specifies the coordinates of the scaled scaling centre, the scaling centre of an unspecified axis is the current position. This command always specifies the absolute position of the scaling centre in the workpiece coordinate controller, whether in absolute (G90) or incremental (G91) mode.
I_: Specifies the scale of the X-axis.
J_: Specifies the scale of the Y-axis.
K_: Specifies the scale of the Z-axis.
......: Toolpath programming command.
G50: Proportional scaling cancelled.

■ Cautions

- When both P and I/J/K are specified in a proportional scaling command, the P command is valid and I/J/K is ignored.
- Axes without specified scaling default to unscaled (i.e. scaled to 1)

3.4 Coordinate controller rotation (G68/G69)

■ Introduction to commands

Using the rotation command, the programmed machining path can be rotated by a specified angle around the centre of rotation. If the shape of the workpiece consists of many identical shapes, the graphic units can be programmed as Subprograms which can then be called up using the rotation commands of the main program. This simplifies programming and saves both time and storage space.

The rotation angle can be programmed in absolute values or in increments. This is shown in the diagram below.

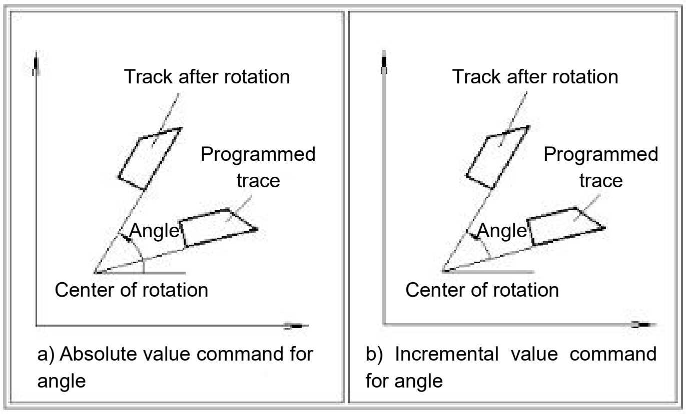


Figure 3- 7 Coordinate controller rotation

■ Command Format

(G17/G18/G19) G68 α_ β_ R_

......

G69

G17/G18/G19. Selecting the plane of coordinate rotation

G68. Coordinate controller rotation is valid.

α_ β_. The absolute position of the centre of rotation. α_/β_ is always the absolute position of the centre of rotation in the workpiece coordinate controller, independent of G90/G91. If α_/β_ is not specified, the centre of rotation defaults to the current position of the tool.

R_. Specify the rotation angle in absolute or incremental values. Positive values indicate counterclockwise rotation, negative values indicate clockwise rotation. When the R_ command is not specified.

.... Toolpath programming commands.

G69: Coordinate controller rotation cancellation.

■ Cautions

- (a) After G68 has been specified, the centre of rotation of the incremental value command is the current position of the tool after G68 and before the absolute command.
- The plane selection code (G17/G18/G19) shall be specified before G68 is specified. If the plane selection code is specified when G68 is valid, the program alarms.

■ Example

The solid line below shows the programmed trajectory and the dashed line shows the tool trajectory after rotation, including two cases.

- Rotation trajectory 1, with "centre of rotation (70, 30)" as the centre of rotation.
- Rotation trajectory 2, with the tool position as the centre of rotation.

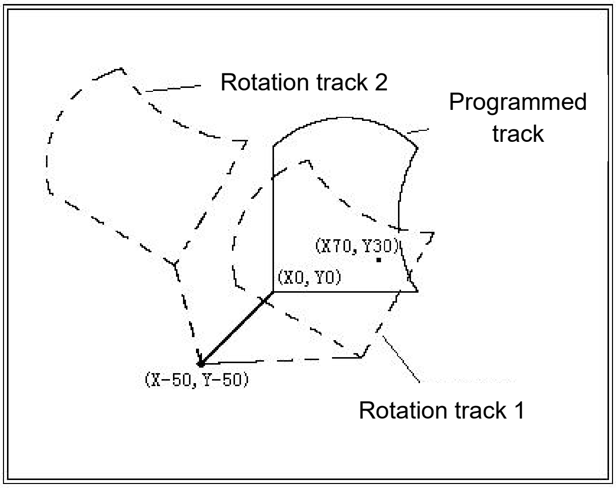


Figure 3- 8 Example of rotation of a coordinate controller

The above two toolpaths are programmed as follows.

Rotation trajectory 1.

```
%1001
G92 x-50 y-50 G69 G17
G68 x70 y30 r60
G90 g01 x0 y0 f2000
G91 X100
G02 Y100 R100
G03 X-100 I-50 J-50
G01 Y-100
G69 G90 X-50 Y-50
M30
```

Rotation trajectory 2.

```
%1002
G92 x-50 y-50 G69 G17
G68 x70 y30 r60
G91 X50 Y50
G91 X100
G02 Y100 R100
G03 X-100 I-50 J-50
G01 Y-100
G69 G90 X-50 Y-50
M30
```

3.5 Exact stop (G61)

■ Command introduction

Once G61 is specified, the tool moves along the programmed path and stops exactly at the end of the programmed segment to cut out the edges precisely.

When High Speed High Precision mode (G64.1/G64.2) is specified, there will be a smooth transition at adjacent trajectory connections.

3.6 Programmable mirror image (G50.1/G51.1)

■ Introduction to commands

The mirroring function enables symmetrical machining of the tool programming path (axis or point symmetry). The following diagram shows that the diagram.

- Mirror trajectory 1 is axisymmetric to the programmed trajectory, with an axis of symmetry of α = 50.
- Mirror trajectory 2 is point symmetrical to the programmed trajectory with a symmetry point of (50, 60).
- Mirror trajectory 3 is axisymmetric with the programmed trajectory and the axis of symmetry is β = 60.

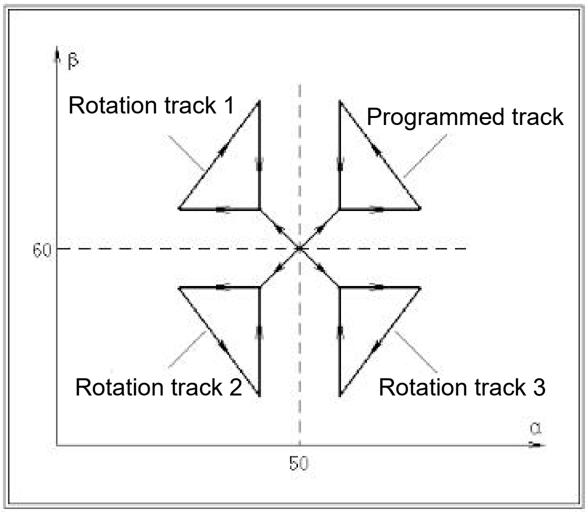


Figure 3- 9 Programmable mirror

■ Command Format

(G17/G18/G19) G51.1 α_ β_
......
G50.1

G17/G18/G19: Select the mirror plane which shall contain the tool programming path.
α_ β_: specifies the axis or point of symmetry of the mirror image.
When only one axis is specified for α_ and β_, it is an axisymmetric mirror.
Point-symmetric mirroring when α_ and β_ specify two axes.
.... Tool programming trajectory.
G50.1. Mirroring function removed.

3.7 Feed hold (G04)

■ Introduction to commands

When G04 is executed, the tool feed is suspended.

G04 can be started to run after suspension by

- Automatic start after a specified time pause
- Triggered by waiting register to start

■ Command Format

G04 P_/X_R_

P_/X_. Specifies the pause time.

P - milliseconds

X-seconds

Only one P_ command and one X_ command can be specified (and one must be specified), otherwise the program alarms.

R_: Specifies the R register signal to wait for after a pause. when the R signal is high, the program continues to run.

■ Cautions

- When both the pause time and the wait register are specified, the pause time starts when the wait register signal goes high.

3.8 Programmable input (G10)

■ Introduction to commands

The user can dynamically modify data such as the zero position of the coordinate controller, the tool data, the value of the registers, etc. in the program, and the changes take effect instantly when G10 is executed. This command is a non-modal command.

■ Command Format

(1), G10 L10 E/K/P_ X_Y_Z_ I_ Write to the zero position of the coordinate controller

E_: specifies the coordinate controller as an external zero offset, usually with an E value of 0.
K_: specifies the coordinate controller as the workpiece coordinate controller G54 to G59, with K taking the values 54 to 59.
P_: designates the coordinate controller as the additional workpiece coordinate controller P1 to P93, with P taking the values 1 to 93.
X_Y_Z_: coordinate offset. Used to specify the required offset in the coordinate controller to be written.
I_: specifying the writing mode as absolute or incremental.
When I takes the value 0, the set value is written to the specified coordinate controller in absolute terms.
When I takes the value 1, the set value is accumulated incrementally to the specified coordinate controller.

(2), G10 L20 P_ K_ Q_ I_ Write to tool data

P_: designation of the tool number.
K_: specifies the data type, with K taking values from 0 to 3.
K0 - tool length.
K1 - tool length wear.
K2 - tool radius.
K3 - tool radius wear.
Q_: the value of the data written in.
I_: specifying the writing mode as absolute or incremental.
When I takes the value 0, the set value is written to the specified

coordinate controller in absolute terms.
When I takes the value 1, the set value is accumulated incrementally to the specified coordinate controller.

(3), G10 L30 D_ Q_ Write register

D_: D register number.

Q_: the value of the data written in.

3.9 Channel synchronization function (G110/G111)

■ Introduction to commands

G110: Waiting for channel synchronisation.

The channel executing G110 will pause in operation until the synchronisation signal from the other channels.

G111: Sets the channel sync signal.

■ Command Format

G110 K_ P_Q_R_

K_. Specify the synchronization register number, the value range is 0~3.

P_: Specifies the channel check code.
For example P0001 for channel 0
P0101 represents channel 0 and channel 2

Q_: No Q written or Q=0. Wait for all synchronization signals
Q=1: Waiting for only one sync signal

R_. Position 1 corresponding to the channel for which the synchronization signal has been received.

G111 K_

K_. Specify the synchronization register number, the value range is 0~3.

■ Example

Channel 0 procedure.

g01 x10 y20 z30 f1000
G04 X2
G111 K0 ;Set channel 0 sync signal
G110 K1 P1000 ;Wait for channel 3 sync signal, otherwise the program is suspended on this line

```
X0 Y0 Z0
M30
```

Channel 3 procedure.

```
g01 x10 y10 f1000
G110 K0 P0001 ;Wait for channel 0 synchronization signal, otherwise the program is suspended on this line
X0 Y0 F2000
G111 K1 ;Set channel 3 sync signal
M30
```

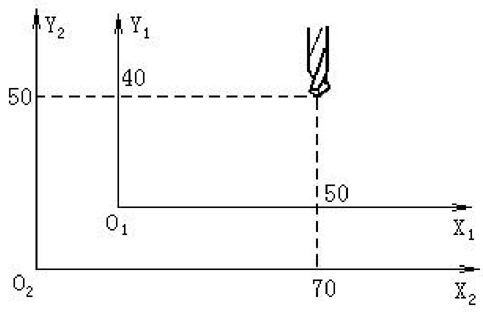
3.10 Coordinate controller

3.10.1 Setting up the workpiece coordinate controller (G92)

■ Introduction to commands

The workpiece coordinate controller is set by setting the point on the tool (e.g. the tool tip) at the specified coordinate position.

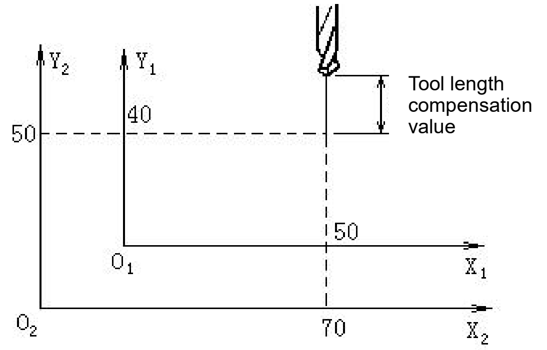
As shown in the figure below, when the tool tip position is set to (X=50, Y=40), the workpiece coordinate controller X_1 O_1 Y_1 is established, and if the tool tip position is set to (X=70, Y=50), the workpiece coordinate controller X_2 O_2 Y_2 is established.



Establish coordinate controller $X_1O_1Y_1$:G92 X50 Y40;
Establish coordinate controller $X_2O_2Y_2$:G92 X70 Y50;

Figure 3- 10 Setting up the workpiece coordinate controller

When setting the workpiece coordinate controller with G92, the IP value after G92 shall contain the compensation value for the tool length if the tool length compensation is valid. This is shown in the diagram below.



Establish coordinate controller $X_1O_1Y_1$: G92 X50 Y40;

Establish coordinate controller $X_2O_2Y_2$: G92 X70 Y50;

Figure 3-11 G92 Setting the workpiece coordinate controller (including tool length compensation values)

- Command Format

G92 IP_

IP_. Specifies the coordinate position of the point on the tool in the workpiece coordinate controller (includes tool length compensation values).

The zero position of the workpiece coordinate controller for axes not specified in the IP_ command remains unchanged, so that G92 has the function of setting the zero position of the workpiece coordinate controller for each axis individually.

- Cautions

The workpiece coordinate controller set using the G92 is volatile and will be lost when the controller is powered off.

3.10.2 Selection of the workpiece coordinate controller (G54/G55/G56/G57/G58/G59)

■ Introduction to commands

In addition to the workpiece coordinate controller, which can be set using G92, it is also possible to use the G command to select a pre-set coordinate controller in CNC controller as the workpiece coordinate controller to be used for machining in the program. 6 standard workpiece coordinate controllers can be set in CNC controller, selected using the following code.

- G54: Selecting the first workpiece coordinate controller
- G55: Selecting the second workpiece coordinate controller
- G56: Selecting the third workpiece coordinate controller
- G57: Selecting the fourth workpiece coordinate controller
- G58: Selecting the fifth workpiece coordinate controller
- G59: Selecting the sixth workpiece coordinate controller

■ Cautions

- External zero offset is valid.

3.10.3 Select additional workpiece coordinate controller (G54.1)

■ Introduction to commands

In addition to the 6 sets of standard workpiece coordinate controllers selectable from G54 to G59, 93 additional sets of workpiece coordinate controllers can be used.

The additional workpiece coordinate controller can be used with the same effect as the standard workpiece coordinate controller.

■ Command Format

G54.1 P_

P_: Specify the additional workpiece coordinate controller number, in the range [1 to 93].

3.10.4 Setting up the local coordinate controller (G52)

■ Command introduction

When programming in a workpiece coordinate controller, it is possible to set up subcoordinate controllers of the workpiece coordinate controller in order to make programming easier. Sub-coordinate controllers are called local coordinate controllers.

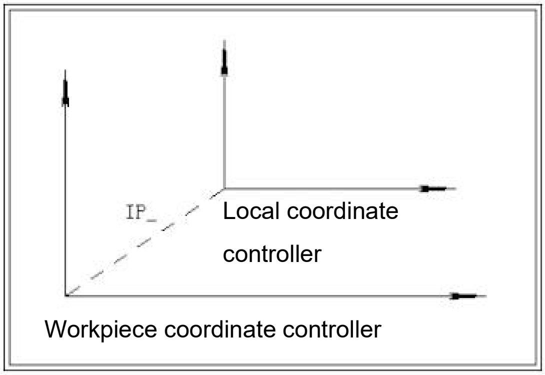


Figure 3-12　Local coordinate controller

■ Command Format

G52 IP_ (I_ J_ K_ R_)

IP_: offset of the zero point of the local coordinate controller relative to the zero point of the workpiece coordinate controller.
I_/J_/K_: specifies the direction of the axis of rotation when the coordinate controller is rotated (can be omitted when the coordinate controller is not rotated).
R_: Specifies the angle of rotation of the coordinate controller (can be omitted if the coordinate controller is not rotated).

■ Change/clear local coordinate controller

- The local coordinate controller can be changed by setting a new local coordinate controller with G52.
- The local coordinate controller is cleared when the IP_ specified by G52 is all zeroes.
- After re-executing the workpiece coordinate controller command (G54 to G59/G54.1/G92), the local coordinate controller is cleared.

■ Cautions

- The local coordinate controller set by G52 is determined by the coordinate controller of the workpiece in which it is currently positioned.
- Once the local coordinate controller has been set up, subsequent programming is carried out in that local coordinate controller.

3.11 Plane selection (G17/G18/G19)

■ Introduction to commands

For functions such as interpolation of arcs and execution of rotation and mirroring of the coordinate controller, it is necessary to select the plane on which the code will be executed.

Three planes are included: the XY plane, the ZX plane and the YZ plane, each selected by the following codes.

- G17: Select XY plane, X-axis is the first axis, Y-axis is the second axis
- G18: Select the ZX plane, with the Z axis as the first axis and the X axis as the second axis
- G19: Select the YZ plane, with the Y-axis as the first axis and the Z-axis as the second axis

The default selection plane at controller power-up can be set by the channel parameters (see the parameter description section for details).

3.12 Tool compensation function

3.12.1 Tool length compensation (G43/G44/G49)

■ Command introduction

If the tool length specified during programming is not the same as the actual tool length used, the difference between the two can be set in the tool offset table and compensated for with a tool length compensation code. In this way, after a tool change, there is no need to modify the machining program, only the tool offset table can be modified for normal machining.

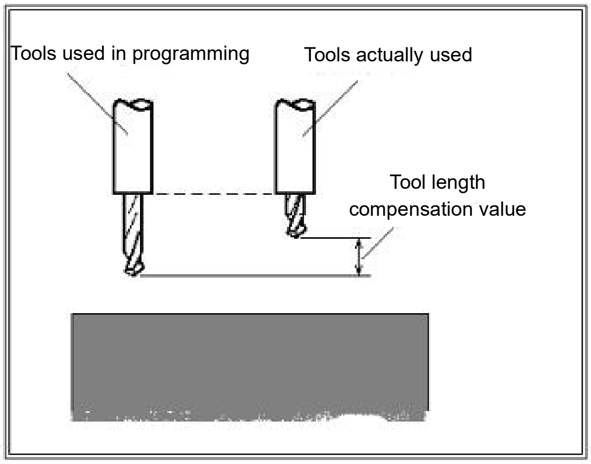


Figure 3-13 Tool length compensation

■ Command Format

G43/G44 H_

......

G49

G43/G44.	Tool length compensation commands
	G43 - Positive compensation
	G44 - Negative compensation
H_.	Compensation number for tool length
....	Tool movement commands
G49.	Tool length compensation cancelled.

- Cautions
  - G43/G44 can only compensate along the Z-axis direction (positive or negative).
  - The H code must be specified together with G43/G44, the program alarms when specified separately.

3.12.2 Tool radius compensation (G40/G41/G42)

■ Introduction to commands

Usually only the tool centre path is programmed during programming (i.e. the tool radius is assumed to be 0), while for actual machining, the tool centre path needs to be offset to a certain extent (offset distance is equal to the tool radius, offset direction can be left offset or right offset, depending on the specific workpiece offset) due to the impact of the tool radius not being 0. The tool radius compensation function is needed at this point.

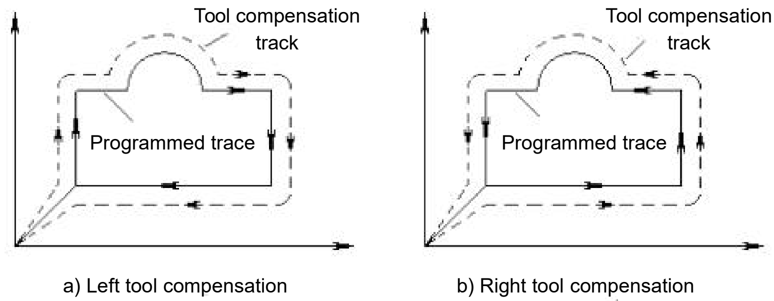


Figure 3-14 Tool radius compensation

■ Command Format

G41/G42 D_
......
G40

G41/G42. Tool radius compensation valid
G41-Left blade complement
G42-Right Blade Patch
D_. Specifies the compensation number of the tool radius. the parameter after the D command specifies the compensation number (or tool number) in the tool parameter table where the compensation value of the tool radius is set. the D command shall be specified together with G41 or G42, the program alarms if specified separately.
.... Tool movement command.
G40. Tool radius compensation cancelled.

■ Cautions

The D code must be specified together with G41/G42, the program alarms when specified separately.

3.13 Subprogram call (M98)

■ Introduction to commands

Subprograms can either be in the same program file as the main program, or they can be separate program files.

The Subprogram name is in the same program file as the main program and the Subprogram name must start with O or %.

Subprograms are separate program files, which must have a four-digit file name and cannot have an extension.

The M99 command must be specified at the end of the Subprogram.

The controller gives priority to calling Subprograms that are in the same program file as the main program.

■ Command Format

M98 P_

P_: Program name (with number of calls)
The lower four digits are the name of the Subprogram, the higher digits other than that are the number of calls, which can be omitted if the number of calls is 1.
For example: M98P121001, i.e. the Subprogram is named 1001 and the number of calls is 12.

■ Example

- Subprograms in the same program file as the main program

```
G00 X0 Y0 Z0
g01 x25 y40 f1000
M98 P52001 ; Call Subprogram 2001, call number 5
G00 X0 Y0
M30

O2001 ; Subprogram name, must be defined with O or %
M3 S2000
G00 Z-10
G01 Z-50 F100
G00 Z0
M05
M99
```

- Subprograms as separate program files

Main program file name: main.nc

```
G00 X0 Y0 Z0
g01 x25 y40 f1000
M98 P52001 ; Call Subprogram 2001, call number 5
G00 X0 Y0
M30
```

Subprogram filename (no extensions allowed): 2001

```
%2001
M3 S2000
G00 Z-10
G01 Z-50 F100
G00 Z0
M05
M99
```

4 Macro program

4.1 Overview

A certain function implemented by a set of commands is stored in memory in advance like a Subprogram, and these functions are called with a single command. These functions are implemented in the program by simply writing out the calling command. The set of commands is called the user macro program ontology, and the commands that call the set of commands are called "user macros". The user macro program ontology is sometimes referred to as a macro program. User macro commands are also known as macro call commands.

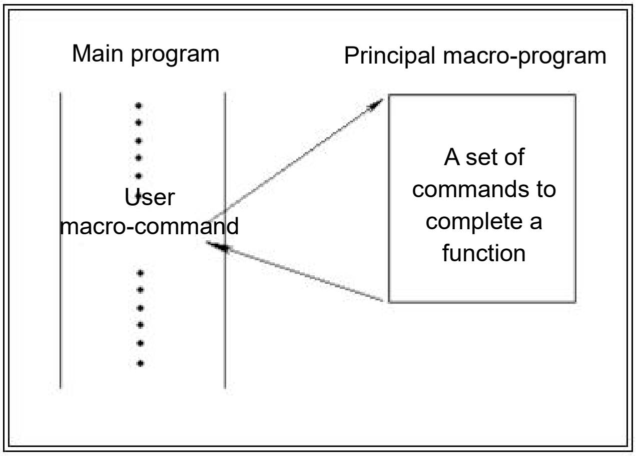


Figure 4- 1 Macro procedure call

The programmer does not have to remember the user macro body, just the macro command that called the macro.

The most important feature of user macros is the ability to use variables in the body of the user macro. Variables can be operated on and can be assigned values using the macro command.
The following program segments are macro program statement segments.

- A program segment containing the assignment operator (=).
- A program segment containing a conditional statement (IF).
- A program segment containing a circular statement (WHILE).
- A program segment containing G65.
- A program segment containing a reference to a variable (#).
- A program segment containing program control block boundary characters ({ and }).

4.2 Variables

Normal machining programs specify G-codes and travel distances directly with values, e.g. G01 and X100.0. When using user macros, values can be specified directly or with variables. When using variables, the variable value can be changed using the macro's assignment statement.

4.2.1 References to variables

The user macro uses the symbol "#" plus the variable name to reference the variable, e.g. G00X#201, which means that the value of variable 201 is used as the programming command coordinate for the X-axis. If you want to change the symbol of the referenced variable, you shall put the minus sign in front of the "#", e.g. G00X-#1.

4.2.2 Types of variables

Variable type			Variable name	Variable address	Properties
Channel variables	Channel 0 variable	Local variables	0 to 99	0 to 99	Read/write
		Public variables	100 to 499	100 to 499	Read/write
		Memory variables	500 to 1499	500 to 1499	Read/write, power off save
		controller variables	1500 to 1999	1500 to 1999	Read only
	Channel 1 variables	Local variables	0 to 99	2000 to 2099	Read/write
		Public variables	100 to 499	2100 to 2499	Read/write
		Memory variables	500 to 1499	2500 to 3499	Read/write, power off save
		controller variables	1500 to 1999	3500 to 3999	Read only
	Channel 2 variables	Local variables	0 to 99	4000 to 4099	Read/write
		Public variables	100 to 499	4100 to 4499	Read/write
		Memory variables	500 to 1499	4500 to 5499	Read/write, power off save
		controller variables	1500 to 1999	5500 to 5999	Read only
	Channel 3 variables	Local variables	0 to 99	6000 to 6099	Read/write
		Public variables	100 to 499	6100 to 6499	Read/write
		Memory variables	500 to 1499	6500 to 7499	Read/write, power off save
		controller variables	1500 to 1999	7500 to 7999	Read only
Axis variables	Axis 0 variable		8000 to 8099	8000 to 8099	Read only
	Axis 1 variable		8100 to 8199	8100 to 8199	Read only
	Axis 2 variables		8200 to 8299	8200 to 8299	Read only
	Axis 3 variables		8300 to 8399	8300 to 8399	Read only
	Axis 4 variables		8400 to 8499	8400 to 8499	Read only
	Axis 5 variables		8500 to 8599	8500 to 8599	Read only
	Axis 6 variables		8600 to 8699	8600 to 8699	Read only
	Axis 7 variables		8700 to 8799	8700 to 8799	Read only
	Axis 8 variables		8800 to 8899	8800 to 8899	Read only
	Axis 9 variables		8900 to 8999	8900 to 8999	Read only
	Axis 10 variable		9000 to 9099	9000 to 9099	Read only
	Axis 11 variables		9100 to 9199	9100 to 9199	Read only
	Axis 12 variable		9200 to 9299	9200 to 9299	Read only
	Axis 13 variables		9300 to 9399	9300 to 9399	Read only
	Axis 14 variables		9400 to 9499	9400 to 9499	Read only
	Axis 16 variables		9500 to 9599	9500 to 9599	Read only
Spindle	Main axis 0 variable		9600 to 9649	9600 to 9649	Read only

variables	Main axis 1 variable	9650 to 9699	9650 to 9699	Read only
	Main axis 2 variables	9700 to 9749	9700 to 9749	Read only
	Main axis 3 variables	9750 to 9799	9750 to 9799	Read only
	Spindle 4 variables	9800 to 9849	9800 to 9849	Read only
	Main axis 5 variables	9850 to 9899	9850 to 9899	Read only
	Spindle 6 variables	9900 to 9949	9900 to 9949	Read only
	Spindle 7 variables	9950 to 9999	9950 to 9999	Read only

4.3 Arithmetic expressions

4.3.1 Arithmetic Operators

Arithmetic operators are symbols used to perform basic arithmetic operations, i.e. symbols used to handle the four operations.

Arithmetic operators include.

- Add (+)
- Less (-)
- Multiply (*)
- Divide (/)

4.3.2 Functions

■ Function Name

Trigonometric functions (angles are in degrees. For example, 90º30´ shall be specified as 90.5º).

- Sine: SIN
- Anyway, the chord: ASIN
- Cosine: COS
- Inverse cosine: ACOS
- Tangent: TAN
- Anyway, cut: ATAN

Other functions.

- Square root: SQRT
- Absolute value: ABS
- Rounded to the nearest whole number: ROUND
- Rounded up: FUP
- Lower rounding: FIX
- Natural logarithm: LN
- Exponential function: EXP

■ Cautions

- (a) In the functions ASIN[#j] and ACOS[#j], the value range of #j shall be -1 ≤ #j ≤ 1, otherwise the program alarms.
- function ATAN[#j], #j cannot take a value that is an odd multiple of 90º, otherwise the program alarms.
- The angular parameters of all trigonometric functions are given in degrees.
- function SQRT[#j], the value of #j cannot be negative,

otherwise the program alarms.

- function LN[#j), the value of #j cannot be zero or negative, otherwise the program alarms.
- function FIX[#j] and FUP[#j], so that the absolute value is greater than the original number of rounding for the upper rounding, so that the absolute value is less than the original number of rounding for the lower rounding, for the negative rounding shall be particularly careful. For example.
  - FIX[-23.3] = -23
  - FUP[-23.3] = -24

4.3.3 Arithmetic expressions

An arithmetic expression is a mathematical operation consisting of a number, a function and an arithmetic operator.

For example.

- #32=#11+#8
- #50 = #20*#30 - 30*SIN[30]/2

■ Order of operations

The usual order of arithmetic expressions is function → multiplication and division → addition and subtraction, but nesting in square brackets can change this order.
This is shown in the diagram below. Where ① → ⑤ indicates the order of operations.

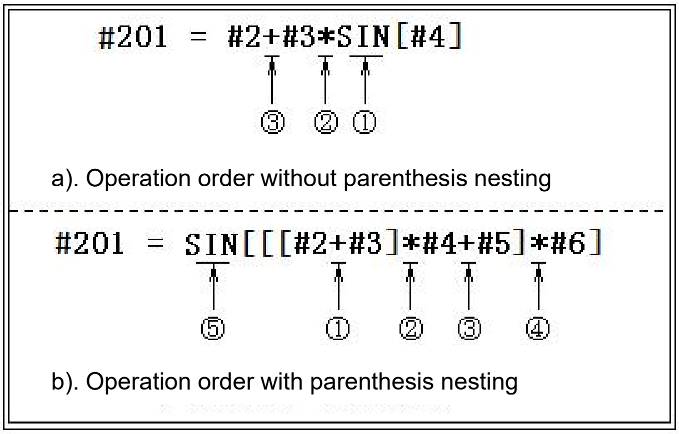


Figure 4- 2 Operation order of arithmetic expressions

4.4 Logical expressions

Logical expressions are used to determine whether a comparison is valid or not, and are usually used in IF and WHILE statements as a condition for execution.

The result of a logical expression is either TRUE (true) or FALSE (false).

4.4.1 Comparison Operators

The comparison operators are used for the comparison of data and include.

- Equivalent to (= =)
- Greater than (>)
- Less than (<)
- Not equal to (< >)
- Greater than or equal to (>=)
- Less than or equal to (<=)

4.4.2 Logical Operators

Logical operators are used for logical determination of multiple comparison operations, including.

- Logical with (&)
- Logical or (|)

4.4.3 Floating point comparison

When comparing floating point numbers, the accuracy of the comparison is set internally by CNC controller at 0.000001, i.e. two floating point numbers are considered equal if the absolute value of the difference is less than 0.000001.

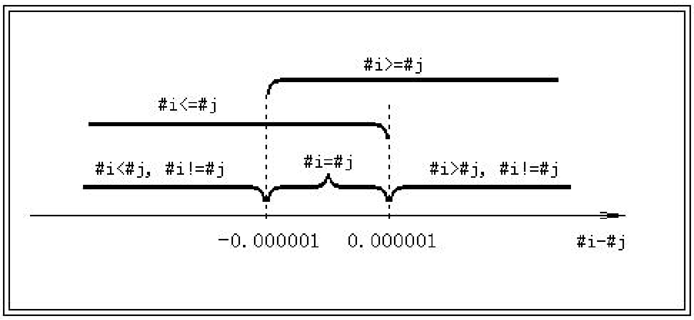


Figure 4-3 Comparison of variables #i and #j

4.4.4 Simple logical expressions

A simple logical expression is an expression that contains only one comparison operator, e.g.

- #30>#20

The result is TRUE if #30 is greater than #20, FALSE otherwise.

- #10+#20<=#30

The result is TRUE if the value of #10+#20 is less than or equal to #30, otherwise it is FALSE.

4.4.5 Compound logical expressions

■ Overview

Combining multiple simple logical expressions with logical operators constitutes a compound logical expression, e.g.

- #10>#20 & #30<#40

#10>#20 and #30< #40 The result is TRUE if both comparisons are valid, otherwise FALSE.

- #10+#20>#30 | #40==#50

#10+#20>#30 and #40==#50 The result of the two comparison operations is TRUE as long as one of them holds, otherwise it is FALSE.

■ Example

```
IF[#202>0 & #203>0]
{; #202 and #203 are both greater than 0 before the following statement is executed
G00 X50
X0
}

IF[#202>0 | #203>0]
{; whenever one of #202 and #203 is greater than 0, the following statement is executed
G00 X50
X0
}
```

4.5 Conditional control statements (IF/ELIF/ELSE)

■ Introduction to commands

Conditional control statements are used to control whether a program block is executed or not. A block is a sequence of program statements defined by the symbols "{" and "}".

■ Command Format

Format of the IF statement (where the ELIF and ELSE statement blocks may be omitted).

```
IF[logical expression]
{
    Program block 1
}
ELIF [logical expression]
{
    Program block 2
}
ELSE
{
    Program block 3
}
```

Execution process:

- Execution of statement block 1 when the IF condition holds, skipping the ELIF, ELSE statements.
- If the IF condition is not valid and the ELIF condition is valid, execute statement block 2, skipping the IF and ELSE statements.
- If neither IF nor ELIF condition holds, execute statement block 3 after ELSE, skipping the IF and ELIF statements.

■ Cautions

The IF, ELIF and ELSE statements must be followed by a block defined by "{" and "}" (even if the sequence of statements is empty or there is only one statement), otherwise the program alarms.

For example.

```
IF[#200<10] G00 X200            ; wrong, shall be '{' and '}'
IF[#200<10] { G00 X200 }    ; Correct
IF[#200< 10] { }      ; correct, sequence of statements is empty
```

4.6 Cycle control statements (WHILE)

■ Command introduction

Cycle control statements are used to control the cyclic execution of a block of programs.

■ Command Format

WHILE [logical expression]
{
  Statement 1
  Statement 2
......
    Statement n
}

When the logical expression holds, the cycle executes the block defined by the symbols '{' and '}' (statements 1 to statement n). Otherwise, the cycle is jumped out and the subsequent statements are executed.

■ Cautions

The WHILE statement must be followed by a block defined by "{" and "}" (even if the sequence of statements is empty or there is only one statement), otherwise the program alarms.

For example.
WHILE (#200<10) G00 X200 ; wrong, shall be '{' and '}'
WHILE (#200<10) { G00 X200 } ; correct
WHILE (#200< 10) { } ; correct, statement sequence is empty

4.7 Unconditional jump statement (GOTO)

■ Introduction to commands

After the jump statement is executed, the program will move to the specified program segment for execution.

■ Command Format

GOTO n

where n is used to specify the segment number of the target segment to jump to.

■ Cautions

When a GOTO statement jumps between program blocks, it can only jump from inside the block to outside or level jump (neither outward nor inward), not from outside the block to inside. Otherwise the program alarms.

For example.

```
GOTO 100 ; Error: The target segment is within the block and cannot be jumped inwards from outside the block

IF[#200==10]
{
n100 g00 x10 y20 z30
GOTO 200 ; correct, level jump
G00 Z150
n200 g01 z-50 f500
GOTO 300 ; correct, jumping outwards from inside the block
}
N300 G00 X0 Y0 Z0
```

4.8 Call of macro program

4.8.1 Non-modal call (G65)

■ Introduction to commands

The macro program is called with the G65 command, which is a non-modal command.

Macro procedure calls are also Subprogram calls, G65 vs. M98.

A set of parameters can be passed to the macro when called with the G65. The M98, on the other hand, has no parameters to pass.

Both G65 and M98 are Subprogram calls, and Subprograms can be nested up to a maximum of 20 levels.

■ Command Format

The format of the G65 command is

G65 P_ L_ (......)

G65. Macro procedure call command. This command must be specified before P_, L_ and the sequence of parameters.
P_. The name of the macro program to be called. The program name is an integer of no more than four digits.
L_. The number of calls. The number of calls shall be greater than or equal to 1, otherwise the program alarms. Can be omitted if the number of calls is 1.
.... Sequence of parameters.

4.8.2 Macro procedure call parameters

When making a macro call (G65), you can specify the parameters of the call, which are passed to the macro being called via local variables.

■ Parameter call method

Call parameters can be specified in two ways. The two methods can be mixed and are automatically recognised within CNC controller. When the two methods are mixed, the later specified parameter is valid (i.e. the later specified parameter will overwrite the previous specified parameter value). : The following is a list of the parameters

● Parameter designation Ⅰ

The independent variables are specified using letters other than G, L, O and N, each letter being specified once to represent a fixed local variable.

The correspondence between letters and local variables is shown in the table below.

Alphabet	**Local variables**	**Alphabet**	**Local variables**	**Alphabet**	**Local variables**
A	# 1	J	# 5	T	# 20
B	# 2	K	# 6	U	# 21
C	# 3	L	#12	V	# 22
D	# 7	M	# 13	W	# 23
E	# 8	P	#16	X	# 24
F	# 9	Q	# 17	Y	# 25
H	# 11	R	# 18	Z	# 26
I	# 4	S	# 19		

● Parameter designation II

The independent variables are specified using I, J and K in Independent Variable Designation II. I, J and K can be repeated 10 times and the program alarms if more than 10 times are specified. Independent variable designation II is used to pass variables such as 3D coordinate values.

The correspondence between I, J, K and the local variables in the independent variable designation II is shown in the table below. The subscripts for I, J and K in the table are used to determine the number of repetitions of I, J and K. They are not written in the actual programming.

Alphabet	Local variables	Alphabet	Local variables	Alphabet	Local variables
I_1	# 4	J_4	# 14	K_7	# 24
J_1	# 5	K_4	# 15	I_8	# 25
K_1	# 6	I_5	# 16	J_8	# 26
I_2	# 7	J_5	# 17	K_8	# 27
J_2	# 8	K_5	# 18	I_9	# 28
K_2	# 9	I_6	# 19	J_9	# 29
I_3	# 10	J_6	# 20	K_9	# 30
J_3	# 11	K_6	# 21	I_{10}	# 31
K_3	# 12	I_7	# 22	J_{10}	# 32
I_4	# 13	J_7	# 23	K_{10}	# 33

4.8.3 Level of local variables

■ Local variable stacks

A stack is a data storage space with only one entry and exit point, as shown below. Data can only enter from the top of the stack and can only exit from the top of the stack in order, so data that enters first exits later and data that enters later exits first (i.e. first in, then out).

There are two types of operations on the stack.

- Stacking: The operation of putting data on the stack is called stacking.
- Out of the stack: The operation of reading data off the stack (and removing it from the stack) is called out of the stack.

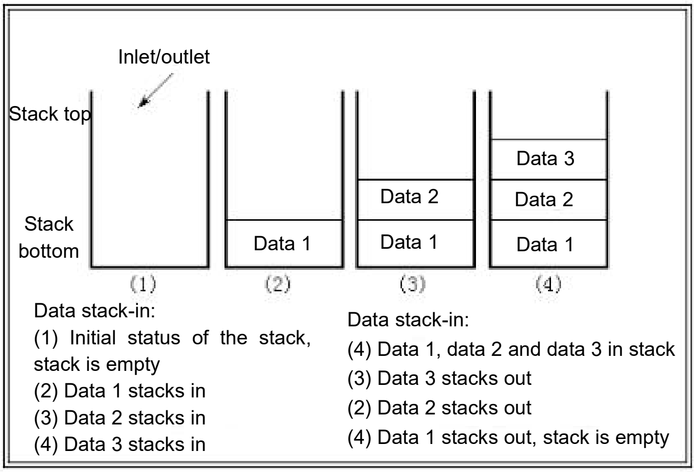


Figure 4-4 Stack-in/stack-out diagram

■ Stacking in and out of local variables

- Local variables on the stack: local variables are put on the stack once for each call to the macro program.
- Local variables out of the stack: Every time the macro program returns (M99), the local variables are out of the stack.

5 High-speed machining functions

5.1 Overview

For the machining of complex surfaces, they are usually discrete into small line segments for programming. In the normal machining mode, the lack of processing capability for small segments leads to low machining efficiency and uneven surfaces. The high-speed machining mode enhances the processing capability of small segments, effectively increasing the machining speed and surface quality of small segments.

The controller supports two high-speed machining modes.

- G05.1Q1 : High-speed machining mode I
- G05.1Q2 : High-speed machining mode II (intelligent contour smoothing)
- G05.1Q0 : High-speed machining mode closed (quasi-stop mode)

	High-speed machining mode I	High-speed machining mode II
Number of pre-reading paragraphs	100	100
Speed foresight function	Yes	Yes
High-speed interpolation of small line segments	Yes	Yes
Tiny line segment merge optimisation	None	Yes
Spline fitting/interpolation function	None	Yes

Normally, for small line segment programming processing, the best results can be obtained by using High Speed Machining Mode II; for long linear segments or circular trajectory processing, High Speed Machining Mode I shall be selected to achieve high precision contouring.

5.2 High-speed machining mode I (G05.1 Q1)

In high speed machining mode I, the controller automatically calculates the transition speed at the connection of adjacent line segments, so that the transition speed is maximised while ensuring smooth machine tool operation, thus achieving high speed machining.

In high speed machining mode I, the interpolation trajectory coincides with the programmed trajectory.

5.3 High-speed machining mode II (intelligent contour smoothing, G05.1 Q2)

5.3.1 Overview

High-speed machining mode II, i.e. spline strip smooth interpolation mode.

In this mode, the toolpath specified by G01 in the program is fitted to a spline curve for interpolation if certain conditions are met.

As shown in the diagram below, the dashed line is the programmed path and the solid line is the actual spline path machine toold by the tool. The proposed splines result in a smoother machining path and more efficient machining.

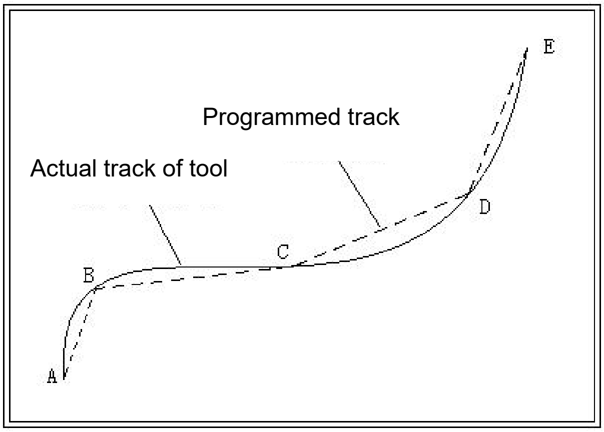


Figure 5- 1 Spline fitting

5.3.2 Spline fitting conditions

G01 programmed straight line trajectories need to meet all of the following conditions in order to be fitted to a spline for interpolation. Segments that do not satisfy the spline fitting condition will be interpolated as a straight programmed trajectory.

- Small line segment length limit

If the length of the small line segment programmed in G01 is less than the limit, the spline fitting condition is satisfied, otherwise it is not.

- Angle between adjacent line vectors

The definition of the angle between adjacent line vectors is shown below. If this angle is less than the limiting value, the spline fitting condition is satisfied, otherwise it is not.

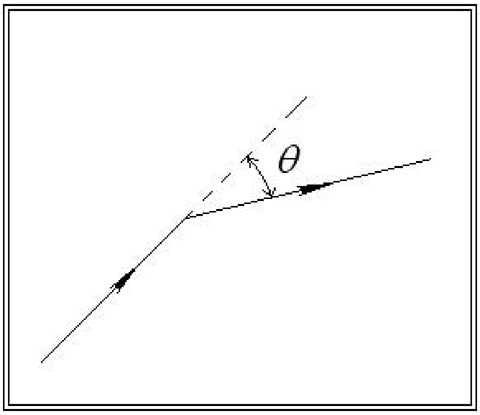


Figure 5- 2 Definition of the vector angle

- Ratio of the lengths of adjacent line segments

If the ratio of the lengths of the first two straight-line segments, L1 and L2, is less than a limited value, the spline fitting condition is satisfied, otherwise it is not.

Assuming a finite value of ε for the length ratio (ε > 1), the range of ratios allowed to fit the splines is

$$\frac{1}{\varepsilon} < \frac{|L1|}{|L2|} < \varepsilon,$$

- Spline fitting error

The spline fitting error is the bow height error between the fitted spline and the original programmed trajectory, as shown below.

If the spline fitting error is less than the limit, the spline fitting condition is satisfied, otherwise it is not.

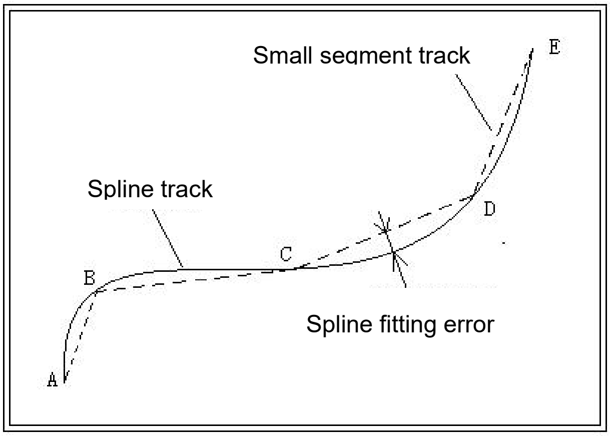


Figure 5- 3 Spline fitting error

6 Five-axis machining functions

6.1 Tool centre point control (RTCP)

6.1.1 Overview

RTCP functions include 3D tool length compensation and table coordinate controller programming.

3D tool length compensation is real-time tool length compensation while changing the attitude of the tool in a 5-axis machine tool. With this function, the tool centre point is guaranteed to move along the path specified in the program even if the tool changes direction to the workpiece.

Table coordinate controller programming means that toolpaths can be programmed in a coordinate controller that rotates with the table. The use of a table coordinate controller not only simplifies CAM programming, but also results in better quality and more efficient machining.

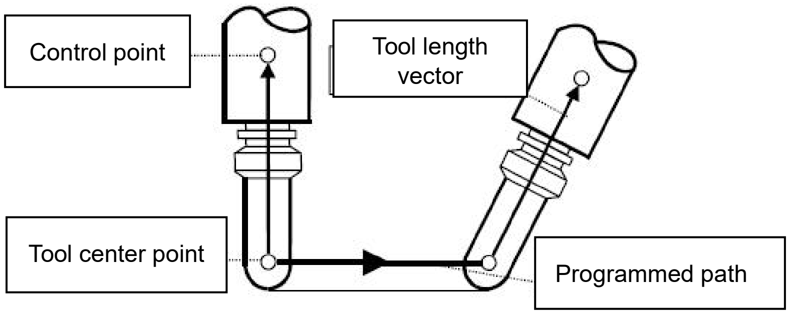


Figure 6- 1 RTCP control

6.1.2 Three-dimensional tool length compensation

Three-dimensional tool length compensation means that in a 5-axis machine tool, the compensation of the tool length is always carried out in the direction of the tool axis, regardless of the position to which the tool is rotated.

6.1.3 Programming of the table coordinate controller

In five-axis machining, there are three main types of coordinate controller: the machine tool, the workpiece and the table. The workpiece coordinate controller is always parallel to the machine tool coordinate controller, while the table coordinate controller is fixed to the table and rotates with it. This is shown in the diagram below.

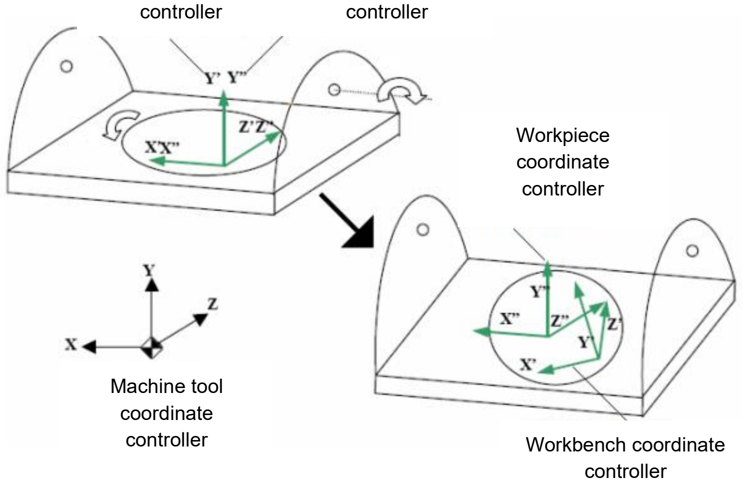


Figure 6-2 Machine tool coordinate controller, workpiece coordinate controller and workbench coordinate controller

6.1.4 RTCP function commands

G43.4 H_ ; Enables RTCP function

......

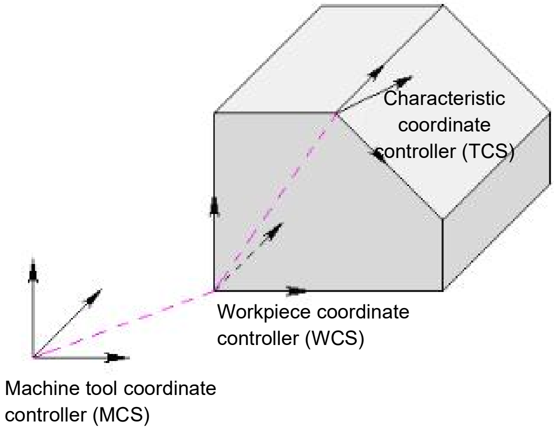
G49 ; Disable RTCP function

The type of construction and parameters of the 5-axis machine tool must be set correctly in order to perform the RTCP function correctly.

6.2 Inclined surface machining commands

6.2.1 Overview

For machining on inclined surfaces, a characteristic coordinate controller (TCS) can be created on that inclined surface and programmed in that coordinate controller. As the characteristic coordinate controller is adapted to the inclined plane, programming on the inclined plane is as simple as programming on the flat plane. This is shown in the diagram below.



6- 3 Inclined surface characteristic coordinate controller

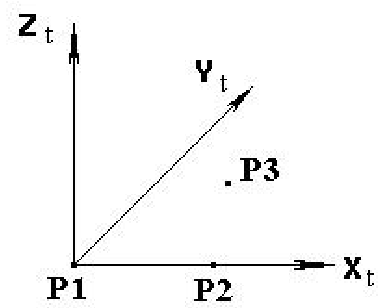
6.2.2 Establishing the characteristic coordinate controller

■ Three point method for establishing the characteristic coordinate controller

The characteristic coordinate controller is established by specifying three points in the characteristic coordinate controller. These three points are.

- Zero point of the characteristic coordinate controller
- Any point in the positive direction of the X-axis of the characteristic coordinate controller (cannot coincide with the zero point)
- Any point in the first or second quadrant of the XY plane of the characteristic coordinate controller (cannot be on the X axis)

Figure 6-4 Three points to establish the characteristic coordinate controller



G68.1 X_ Y_ Z_ I_ J_ K_ U_ V_ W_ ; Set characteristic coordinate controller
...... ; Commands in the characteristic coordinate controller
G69 ; Cancellation of the characteristic coordinate controller

Of which.
X_/Y_/Z_. Zero point of the characteristic coordinate controller
I_/J_/K_. Any point in the positive direction of the X-axis of the characteristic coordinate controller
U_/V_/W_. Any point in the first or second quadrant of the XY plane of the characteristic coordinate controller, which cannot be on the X-axis line.

■ Euler angle setting

The characteristic coordinate controller is established by specifying the Euler angle that determines the direction of the characteristic coordinate controller.
The Euler angles are rotated in the order ZXZ, as shown below.

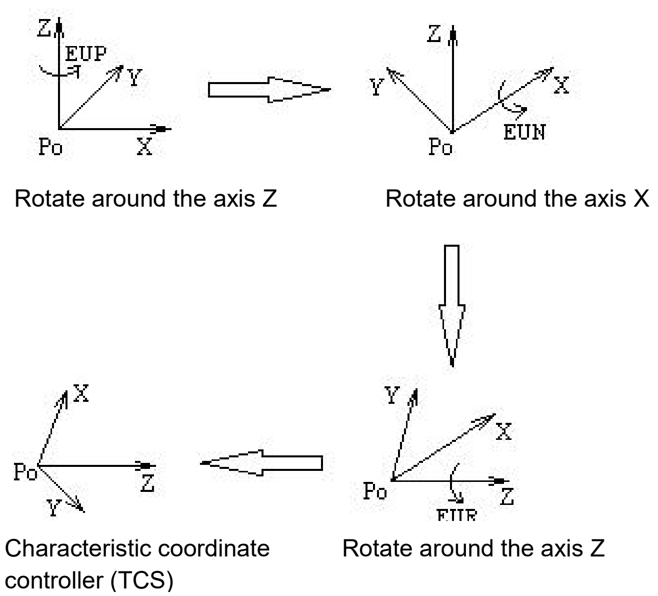


Figure 6- 5　Euler angles

G68.2 X_ Y_ Z_ I_ J_ K_　; Set characteristic coordinate controller
......　; Commands in the characteristic coordinate controller
G69　; Cancellation of the persistent coordinate controller

Of which.
X_/Y_Z_: zero point of the characteristic coordinate controller
I_/J_/K_: Euler angles determining the direction of the characteristic coordinate controller

6.2.3 Tool axis direction control (G53.1/G53.2)

■ Overview

After specifying G68.1 to establish the characteristic coordinate controller, G53.1 or G53.2 can be commanded to control the tool axis swing to a direction parallel to the Z axis of the characteristic coordinate controller.

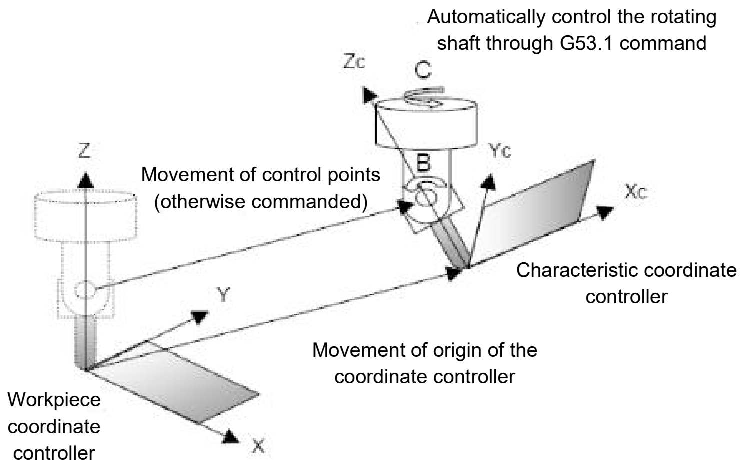


Figure 6- 6 Tool axis direction control

■ Command Format

G53.1/G53.2

- When the G53.1 command is executed, only the rotary axis moves, the linear axis does not make a compensating movement.
- When the G53.2 command is executed, the rotary axis is moved while the linear axis makes a compensating movement, keeping the relative position of the tool tip to the workpiece unchanged.

■ Speed designation

If the current mode is G00, it is the rapid traverse speed; otherwise, it is the cutting feed speed (specified by F).

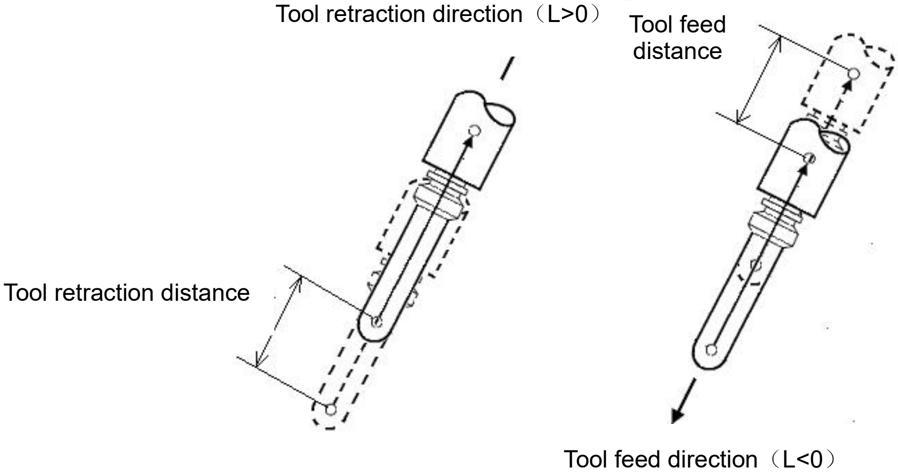
■ Cautions

- G53.1/G53.2 must be specified when G68.1 is valid, otherwise an alarm is raised.
- RTCP must be switched on before executing G53.2, otherwise it will result in an action error.

6.3 Advancing and retracting tool in normal direction (G00.4/G01.4)

■ Overview

Normal tool entry and exit means that the tool is entered or exited in the direction of the tool axis, as shown in the diagram below.



■ Command format.

G00.4/G01.4 L_

Where L specifies the distance between entering and leaving the knife.

- movement of the tool away from the workpiece (retraction) at L>0.
- movement of the tool in the direction of the workpiece (feed) when L< 0.

The feed and retract speed of the command G00.4 is the G00 speed; the feed and retract speed of the command G01.4 is specified by F.

7 Auxiliary functions

7.1 M commands

When the auxiliary function M code is executed, the controller outputs a given signal (high or low) to the specified I/O address, which can be used by the user to control various switches on the machine tool.

M-codes are defined by the machine tool shop, except for the following M-codes which are handled internally by the controller

- M00: Program suspended
- M01: Program selection stop
- M02: Program ends but does not return program header
- M03: spindle positive rotation
- M04: Spindle reversal
- M05: Spindle stop
- M30: End of program and execution of reset.
- M98: Subprogram call
- M99: Subprogram return. If M99 is executed in the main program, it automatically jumps to the program header cycle and runs.

7.2 S commands

The S command is used to specify the speed value of the spindle. After executing M03 (spindle forward), M04 (spindle reverse) or manually starting the spindle, the spindle will rotate at the speed value specified by this S command.

When the S command is executed, the spindle speed value is changed immediately if the spindle is started, otherwise, if the spindle is not started when the S command is executed, only the speed value is stored without starting the spindle.

The spindle speed specified after the S command shall be greater than zero, the minus sign is ignored when the specified speed is less than zero.

7.3 T commands

The T command is used for tool selection in the tool magazine. The value specified after the T command is the tool number of the selected tool.

8 Program alarms and descriptions

Alarm Codes	Alarm messages	Details
-1	Invalid decoder	Please contact the manufacturer of the CNC controller department.
-2	Function not registered	For use of unregistered functions, please contact the CNC controller manufacturer.
-3	Exceeding the maximum number of linked axes	The number of axes programmed in a single program segment exceeds the maximum number of linked axes.
-9	Program segment exceeds maximum length limit	The maximum length allowed for a single program segment is 128 characters.
-10	Illegal characters	An illegal character appears in the program segment.
-11	Wrong program format	
-12	Mismatch of machine tool types	Commands are used that do not match the current machine tool, e.g. lathe-specific commands cannot be used on a milling machine tool.
-13	Invalid G command	An illegal G command was used.
-14	Invalid M command	An illegal M command was used.
-15	Invalid S command	An illegal S command was used.
-16	Invalid T command	An illegal T command was used.
-17	Invalid F command	An illegal F command was used.
-18	Too many single program segment commands	A maximum of 20 G and M commands can be programmed in a single program segment.
-19	Too many single program segment control commands	A maximum of 20 control commands can be programmed in a single program segment.
-26	Change undo order wrong	Coordinate transformation commands such as G51/G51.1/G68, etc., which are programmed with nesting, shall be undone in the reverse order of when the coordinate transformation was established.
-30	GOTO markings are invalid	Illegal markings were used.
-31	IF command formatted wrong	
-32	WHILE command formatted wrong	
-33	No IF and ELSE matching	
-34	Excessive nesting of program blocks	IF/WHILE allows a maximum of 20 levels of nesting.

-35	Too many levels of Subprogram calls	Subprogram calls are allowed to be nested at a maximum of 20 levels.
-36	Subprogram does not exist	
-37	Subprogram not returned	The end of the Subprogram is not specified M99 return.
-40	Jump target does not exist	The target of the GOTO command jump does not exist.
-41	Jump to target illegal	When a GOTO statement jumps between blocks, the direction of the jump can only be from inside the block to outside or level jump (neither outward nor inward), not from outside the block to inside.
-50	Missing block command (IF/WHILE)	Missing program block after IF/WHILE.
-51	Missing block starter '{'	The block definers '{' and '}' must appear in pairs.
-52	Missing block terminator '}'	The block definers '{' and '}' must appear in pairs.
-53	The block terminator '{' is redundant	The block definers '{' and '}' must appear in pairs.
-54	Expressions too long	The maximum length allowed for mathematical expressions is 128 characters.
-55	Incorrectly positioned block comment character	Block comments must be written on a separate line and not on the same line as the program commands.
-56	Incomplete commenting of program blocks	The block comment is missing '(' or ')'.
-66	Invalid measurement head number	The controller supports a maximum of 6 probes with a range of 1 to 6 stylus numbers, beyond which stylus numbers are invalid.
-70	Circular arcs do not exist	The programmed arc does not exist.
-71	Circular arcs are not unique	The programmed arc data corresponds to multiple arcs.
-72	G02/G03 command format error	
-73	3D arcs cannot be programmed when G68 is active	
-74	3D arcs with unspecified midpoints	
-75	Invalid reference point number	The controller supports up to 6 reference points, numbered in the range 1 to 6; numbers outside the range are invalid.
-76	Execution of the command requires the channel to return to zero on all axes	
-80	Not equipped with tapping swivel	The controller must be equipped with a positional spindle to perform tapping commands.

-81	Undefined tapping guide	When executing a tapping command, the tapping pitch must be specified.
-82	Too many in-segment movement commands	
-85	Register address error	
-90	Suspension of time is invalid	The G04 pause time range is 0 to 999999 seconds, any time outside this range is invalid.
-91	G04 missing parameter word	G04 must be followed by a P or X parameter word specifying the pause time.
-93	M98 missing parameter word	The M98 must be followed by the P parameter word specifying the Subprogram.
-95	Invalid tool compensation number	
-96	Reference point return command cannot be executed when the tool radius is valid	
-97	Reference point return command cannot be executed when scaling is active	
-101	G43 missing parameter word	The mandatory H parameter word after G43 specifies the knife complement number.
-102	No plane switching when G68 is in effect	When G68 is in effect, the G17/G18/G19 plane selection commands cannot be executed.
-103	G41/G42 mode cannot switch planes	When G41/G42 is in effect, the G17/G18/G19 plane selection commands cannot be executed.
-104	G43/G44 mode cannot switch planes	When G43/G44 is in effect, the G17/G18/G19 plane selection commands cannot be executed.
-105	G41/G42 mode cannot change the workpiece coordinate controller	When G41/G42 is in effect, the coordinate controller setting commands (G54 to G59/G92/G52, etc.) cannot be executed.
-109	G51 scaling factor too small	The G51 scaling factor must not be less than 0.001.
-110	Mirror axis invalid	G51.1 cannot specify an axis other than the current plane as a mirror axis.
-111	Lack of mirror axis	G51.1 No mirror axis is specified.
-115	G52 unspecified rotation angle	When setting up a local coordinate controller in G52, if the coordinate controller has a rotation, both the axis of rotation and the angle of rotation must be specified.
-116	G52 unspecified axis of rotation	When setting up a local coordinate controller in G52, if the coordinate controller has a rotation, both the axis of rotation and the angle of rotation must be specified.

-117	The axis of rotation specified by G52 is invalid	The I/J/K of the specified axis of rotation cannot be all 0s.
-118	G65 missing Subprogram name	G65 must be followed by a P parameter word specifying the Subprogram to be called.
-120	Coordinate controller does not exist	The standard workpiece coordinate controller is G54 to G59, additional workpiece coordinate controllers are numbered in the range 1 to 93, and coordinate controllers outside this range are not valid.
-121	Invalid coordinate controller	Only orthogonal coordinate controllers can be specified, i.e. where the three axes are perpendicular to each other.
-130	Wrong position of the jump charm	The jump character "\" can only be at the beginning of a program segment.
-131	O command in wrong position	The O command must be at the beginning of the program segment.
-132	Wrong N command position	The N command must be at the beginning of the program segment.
-133	Missing command word/parameter word value	
-134	Invalid value	
-135	Parameter word redundant	
-150	Wrong format for mathematical expressions or values	
-151	Missing front brackets	The operators '[' and ']' in mathematical and logical expressions must appear in pairs.
-152	Missing back brackets	The operators '[' and ']' in mathematical and logical expressions must appear in pairs.
-153	except 0	A division by zero error occurred in a mathematical expression.
-154	Illegal variables	
-155	Variables are read-only	
-170	Illegal function names	
-171	TAN function parameter error	The value of the parameter of the TAN function cannot be an odd multiple of 90 degrees.
-172	ASIN function parameter error	The parameter values of the ASIN function range from -1.0 to 1.0.
-173	Wrong ACOS function parameter	The ACOS function has a range of parameter values from -1.0 to 1.0.
-174	Wrong SQRT function parameter	The SQRT function cannot have a negative number of parameters.
-175	Wrong LN function parameter	The value of the argument to the LN function must be greater than 0.
-200	Macro command format error	
-201	Macro command parameter error	

-250	Invalid characteristic coordinate controller data	When using three points to establish a characteristic coordinate controller, the three given points cannot coincide and cannot be on the same line.
-251	Characteristic coordinate controller does not exist	Before executing the G53.1 (tool orientation) command, the characteristic coordinate controller must be established.
-300	Tool radius compensation - buffer overflow	Please contact the CNC controller manufacturer.
-301	Tool radius compensation - creation of segments invalid	Tool radius compensation can only be established for straight lines.
-302	Tool radius compensation - end segment invalid	The tool radius compensation undo command must be preceded by a Movement command.
-304	Tool radius compensation - invalid transfer type	
-305	Tool radius compensation - offset arc not present	The arc does not exist after performing tool radius compensation (the tool radius must be smaller than the arc radius when machining an internal arc).
-306	Tool radius compensation - offset straight line without intersection with circular arc	In a trajectory where a straight line meets an arc, there is no intersection between the straight line and the arc after performing tool radius compensation.
-307	Tool radius compensation - offset arcs without intersection with circular arcs	The intersection of two arcs does not exist in the trajectory where the arc meets the arc, after performing the tool radius compensation.
-308	Tool radius compensation - offset trajectory reversal	The trajectory after performing tool radius compensation is in the opposite direction to the trajectory before compensation (the tool radius must be smaller than the trajectory length).
-400	No R-plane specified	
-401	Unspecified hole bottom coordinates	
-402	The R plane must be above the bottom of the hole	
-403	Uncommanded depth of cut Q	
-407	Tapping guide F not commanded	
-408	The R-plane must be below the hole	
-409	Uninstructed Offset Q	