CONTENTS

PROGRAMMING

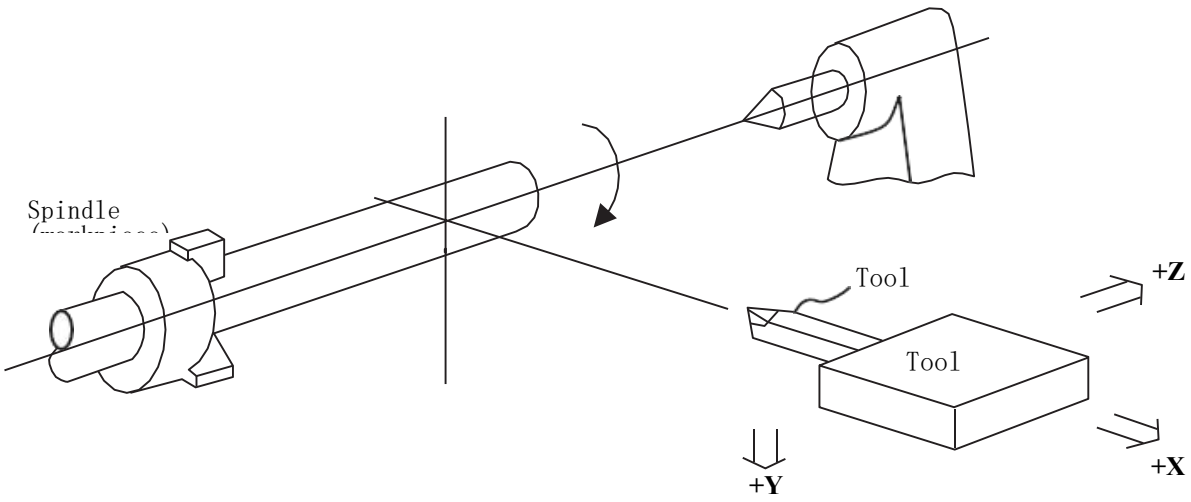
2.1 GENERAL

2.1.1 Axis Definition

For lathes, the axis names (coordinate terms) and directions are defined as follows:

- The axis perpendicular to the spindle, referred to as: Axis X
- The axis parallel to the spindle, referred to as: Axis Z

Definition of coordinate axes for general lathes:



Since the coordinates of the right-handed coordinate system are used for the lathe, the positive direction of the Y-axis, which intersects vertically with the X and Z-axes in the above figure, is the lower direction of the figure. It should be noted that the arcs on the X and Z planes exhibit clockwise and counterclockwise rotations when viewed from the positive direction of the Y-axis.

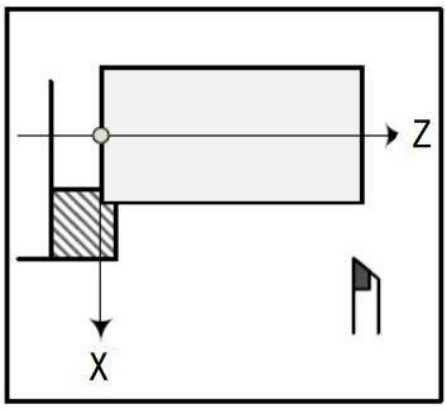


Figure: front tool post coordinate system

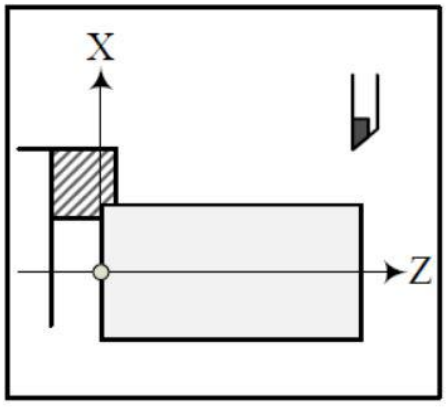
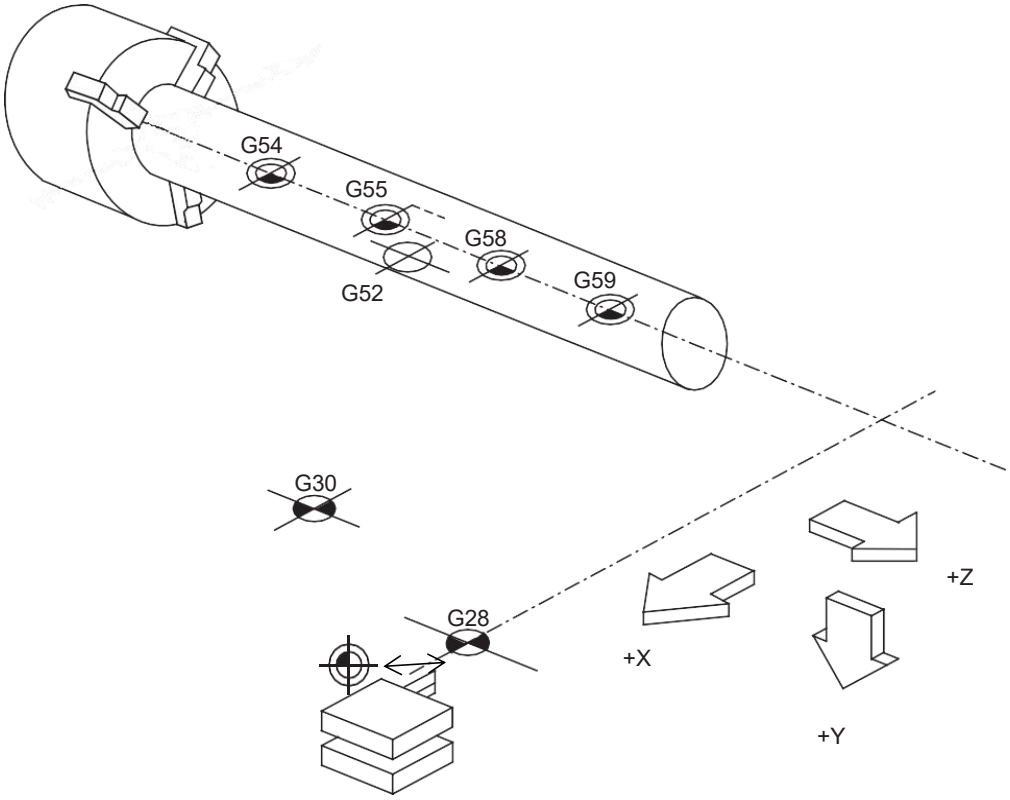


Figure: rear tool post coordinate system

X axis extends around front-rear direction of the horizontal plane, and Z axis extends around left-right direction of the horizontal plane. The direction close to workpiece is the negative direction, and the direction departing workpiece is the positive direction. As shown in the figure, for the coordinate system of front and rear tool posts, X direction is just the opposite, and Z direction is the same. For the figures and examples, the programming application will be described through front tool post, and the rear tool post lathe CNC can be analogized.

2.1.2 Coordinate System

The coordinate systems in the CNC system are associated:



The machine coordinate system origin (i.e., the position where the machine tool coordinate value of each axis is 0), is a position inherently specified in the machine tool, and the machine zero returning (mechanical zero returning) is usually performed by G28;

The reference point is usually used to establish the coordinate system and a specific position for tool change, and G30 is usually used to move the tool to a fixed position. For this series of CNC systems, the first reference point coincides with the origin of the machine coordinate system by default.

The workpiece coordinate system origin is established by executing G54-G59 to offset the machine coordinate system origin.

The local coordinate system origin is established by executing G52 to offset the workpiece coordinate system origin.

2.1.2.1 Machine coordinate system

A specific point on the machine tool as a processing reference is called the machine zero, the machine tool manufacturer sets the machine zero for each machine tool, and the coordinate system with the machine zero as the origin point is called the machine coordinate system. This series of CNC systems establish the machine coordinate system by:

① Zero point floating

The machine coordinate system is established through the power-off memory data of the CNC system. The pulse type system uses this method by default, and the bus type system needs to set the 4th bit of the CNC parameter 0003 to 1 (use the memory position to establish the coordinates). The machine coordinate system origin is established by the machine tool clearing operation.

② Machine zero returning proximity switch

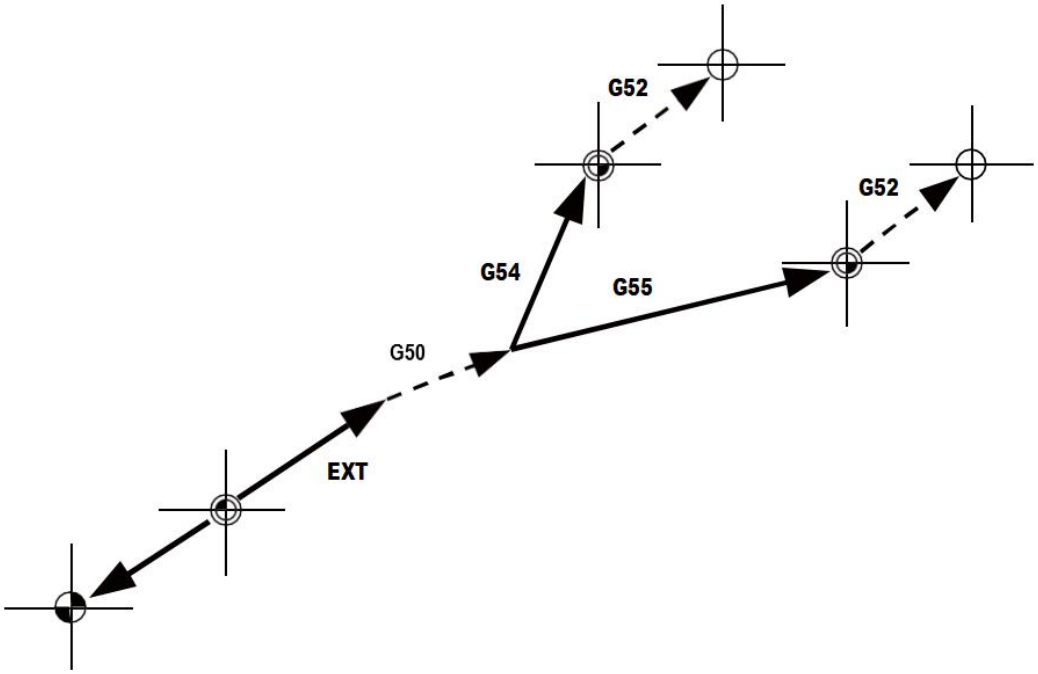
The zero returning operation needs to be performed to establish the machine coordinate system origin based on the mechanical position of the proximity switch mounted on the machine tool.

③ Servo motor position feedback data

Only the bus type system and absolute servo motor support this method. When the CNC is powered on, the gear ratio is calculated based on the servo position feedback data to obtain the current machine coordinate value of each axis. A new machine coordinate system origin can be established by the machine tool clearing operation.

2.1.2.2 Workpiece coordinate system

The workpiece coordinate system is the coordinate system used in the workpiece machining and is established based on the machine coordinate system. Different program origins are established by setting different offset values, or modified by coordinate system commands.



Machine reference point

Origin of machine coordinate system

Origin of workpiece coordinate system

Origin of local coordinate system

Offset through parameter settings

Offset through programming

G52	Local coordinate system offset
G54	Workpiece coordinate system (G54) offset
G55	Workpiece coordinate system (G55) offset
G50	Coordinate system offset
EXT	External workpiece coordinate offset

The local coordinate system G52 is valid in the coordinate system specified by the workpiece coordinate systems G54 to G59.

Furthermore, the basic machine coordinate system can be moved by the G50 command and used as a virtual machine coordinate system. The workpiece coordinate systems G54 to G59 are shifted at the same time.

2.1.3 Coordinate Unit and Range

By default, the minimum programming unit of this system is 0.001 mm, and the maximum programmed movement range is ±999999.999mm.

The minimum programming unit of the CNC is controlled by sys-parameter 0013, and the setting range is from 0.001 mm to 0.000001 mm.

2.1.4 Diameter and Radius Programming

There are diameter programming and radius programming for lathe CNC programming, see figure 2-7.

- Select diameter programming or radius programming by setting bit 1 of sys-parameter 0001 to 0 or 1;
- When the diameter programming is selected, the numerical value behind command X or U represents the diameter value;
- When the radius programming is selected, the numerical value behind command X or U represents the diameter value;

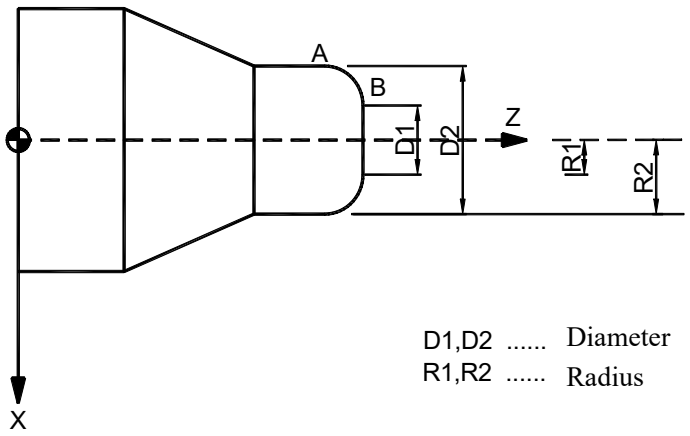


Figure 2-7

When the diameter programming is selected, attention should also be paid to the conditions in the following table:

Item	Cautions and Notes
Z-axis command (address **Z** or **W**)	Unrelated to diameter and radius programming
X-axis command (address **X** or **U**)	Carry out programming with diameter value
Coordinate system setting (**G50**)	Command X-axis coordinate value with diameter
X-axis cutter compensation value	Carry out setting with diameter value
X-axis cut depth of **G90, G92, G94**	Carry out setting with radius value
Radius command of circular interpolation (**R, I, K**)	Carry out setting with radius value
Feed speed in **X**-axis direction	Radius value variation

Note: When no diameter or radius is specified specifically in the following descriptions, the X-axis stands for the diameter value by default.

2.1.5 Programming Coordinate Values

After the workpiece coordinate system is established, the coordinate positions of all programming points are coordinate values relative to the zero point of workpiece coordinate system, but the programming value of certain positioning point or certain feeding point can employ absolute coordinate values (X, Y field), relative coordinate values (U, W field), or hybrid coordinate values (X/Z, U/W field, use absolute and relative coordinate simultaneously) to carry out programming.

1. Absolute coordinate value programming

"The distance from the coordinate system origin" is the coordinate position where the tool will move.

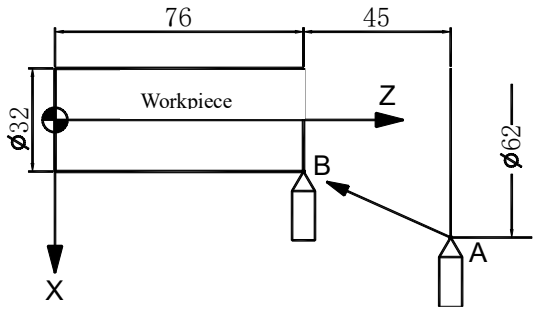


Figure 2-8

As shown in Figure 2-8, the tool moves from point A to point B, the coordinate value of point B is used, and the command is as follows:

X32.0 Z76.0;

2. Relative coordinate value programming

Command the distance from the current position to the next position (the plus or minus sign indicates the direction)

As shown above, the tool moves from point A to point B as well, and the commands are as below:

U-30.0 W-45.0;

3. Hybrid coordinate value programming

As shown above, the tool moves from point A to point B as well, and the commands are as below:

U-30.0 Z76.0; or X32.0 W-45.0;

2.1.6 Extended Format of incremental Programming

Generally, the absolute coordinates of lathe coordinate system are generally X, Z, Y and the fourth axis (A or B or C), and the corresponding incremental coordinates are U, W, V and H respectively. If the machine tool has more than 4 axes, such as X, Y, Z, A, B, the incremental programming H corresponds to axis B but not axis A; in the same way, if there are axes X, Y, Z, A, B and C, the incremental programming H corresponds to axis C but not axis A or axis B.

Therefore, the incremental programming is carried out in the following new ways:

Axis number	Absolute programming	Incremental programming (ISO)	Incremental programming (extension)
X	X	U	UX
Z	Z	W	UZ
Y	Y	V	UY
A	A		UA
B	B		UB
C	C	H	UC

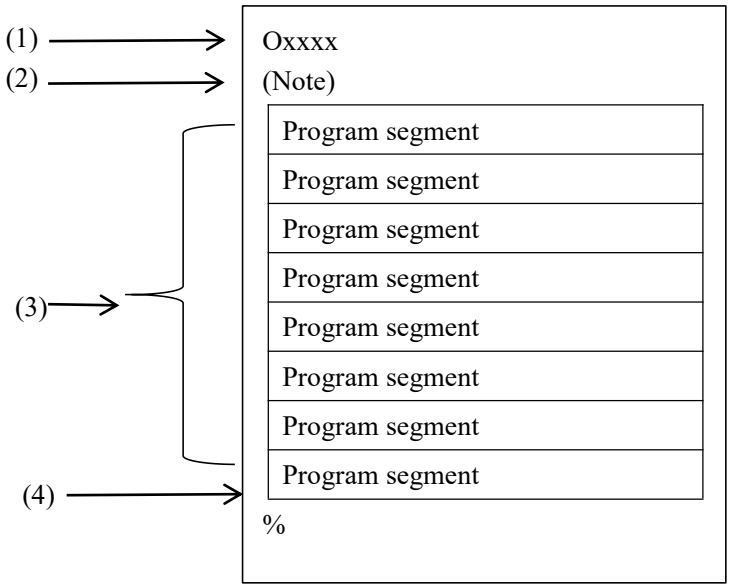
2.1.7 Composition of CNC Program

2.1.7.1 Program format

A collection of commands issued to the NC to run the machine tool is called a "program".

A program is a collection of "program segments", and each program segment specifies a machine action (sequence). Write down these commands (program segments) in the order in which the tool is actually moved.

A program segment is a collection of "words", and a "word" is used to specify a command for an operation. A word is a collection of characters (letters, numbers, symbols) that are arranged in a certain order.



(1) Program name

In order to distinguish the programs from each other, a program number in the format of ○□□□□ is used at the beginning of the program with the address O and the subsequent four-bit value. The program number is also conventionally referred to as the program name. In the program directory display screen, you can view all program names stored in the CNC.

(2) Note

The contents enclosed in parentheses "( )" or after the symbol "//" and the symbol ";" as notes, are ignored by the CNC during execution.

(3) Program section

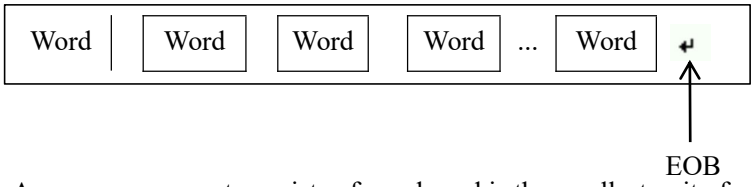
Consisting of multiple program segments.

(4) End of program

The CNC will automatically filter the text after the symbol "%".

2.1.7.2 Program segments and words

1. **Program segment**

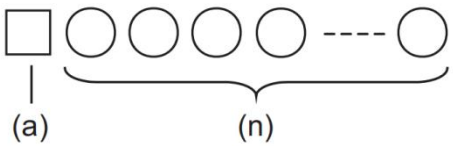


A program segment consists of words and is the smallest unit of a command.

It contains the information required to perform the characteristic actions of the machine tool and a complete command is composed of program segments.

A record terminator (EOB, which is replaced by the icon ↵ in the CNC for convenience) is added at the end of the program segment to indicate the end of the segment.

2. Word

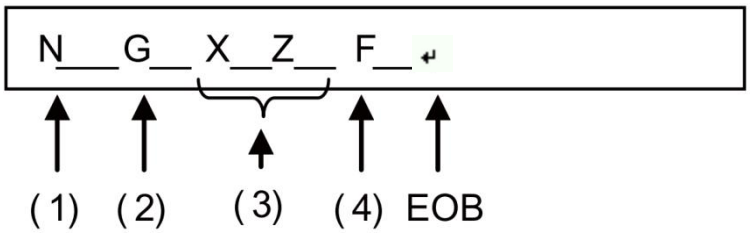


(a) character (address)
(n) number

The word consists of character and numbers (numeric information) called addresses.

The meaning of the numeric information and the number of valid bits varies from address to address.

The content of the main characters is shown below.



(1) Sequence number (program segment number)

The "sequence number" consists of the address N and up to 8 digits thereafter. It is used in the program as a flag to search for the desired program segment (the jump target of a branch). It does not affect the machine action.

(2) Preparatory function (G function)

The "preparatory function (G code, G function)" consists of the address G and the next 2 or 3 digits (which may also contain 1 decimal place). The G code is mainly used for functions such as movement of specified axes and setting of coordinate system. For example, G00 is used to perform positioning and G01 is used to perform linear interpolation.

(3) Coordinate term (axis name)

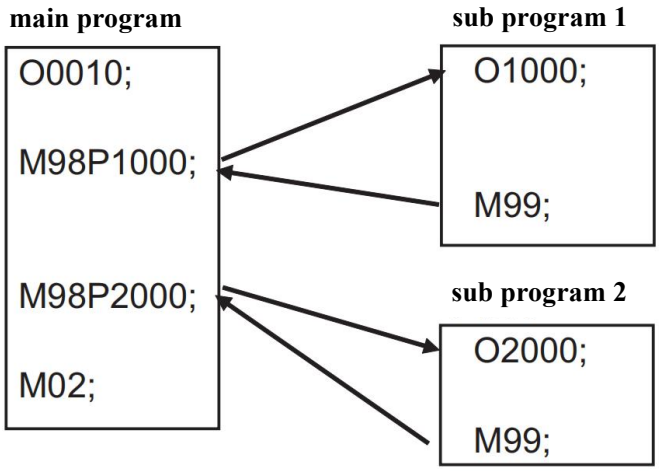
The "coordinate terms" are used to specify the position and the amount of movement of each axis of the machine tool. It consists of numeric information (positive and negative signs and numbers) indicating the address of each axis of the machine tool and the numeric information thereafter.

Addresses can use X, Y, Z, U, V, W, A, B, C, etc. There are two methods of specifying coordinate positions and movement amounts with numeric values: "absolute value command" and "incremental value command".

(4) Feed function (F function)

The "feed function (F function)" is used to specify the relative speed of the tool to the workpiece. It consists of the address F and the digit thereafter.

2.1.7.3 Main program and subprograms



A certain fixed sequence and repeatedly used parameters are prefixed and memorized into memory as a subprogram, which can be called from the main program when needed.

When the main program is executed, the subprogram is executed if the command to call the subprogram exists. At the end of the execution of the subprogram, it returns to the main program.

Details about subprogram execution.

2.1.7.4 End of program

The following code at the end of the program indicates the end of the program.

M02	The main program ends, without returning to the beginning of the program.
M30	The main program ends and returns to the beginning of the program.
M99	The subprogram ends and returns to the calling program.

During program execution process, if M30 or M99 is executed, the CNC will terminate the program that is executed currently. If M30 is executed, the document execution pointer will return to the start of program, and corresponding output control processing will be carried out according to parameter setting at the same time; if M99 is executed, the program execution process will return to the program that has called the subprogram.

2.2 INTERPOLATION FUNCTIONS

2.2.1 Fast Positioning (G00)

Format

G00 X/U_ Z/W_ α/β_;

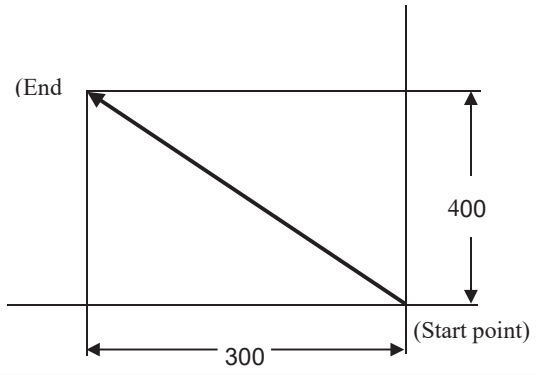
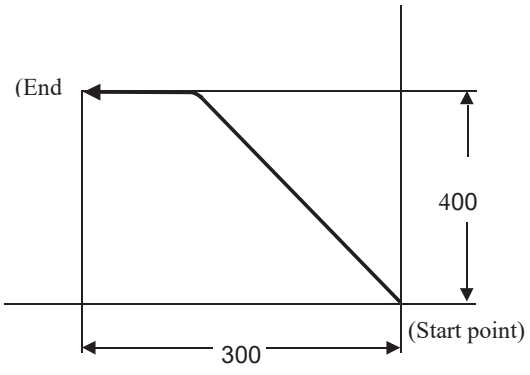
X/U : X-axis end point coordinates (X is absolute programming, and U is incremental programming)

Z/W : Z-axis end point coordinates (Z is absolute programming, and W is incremental programming)

α/β : additional axis end point coordinates
(α is absolute programming, and β is used for incremental programming)
(multiple additional axes can be specified)

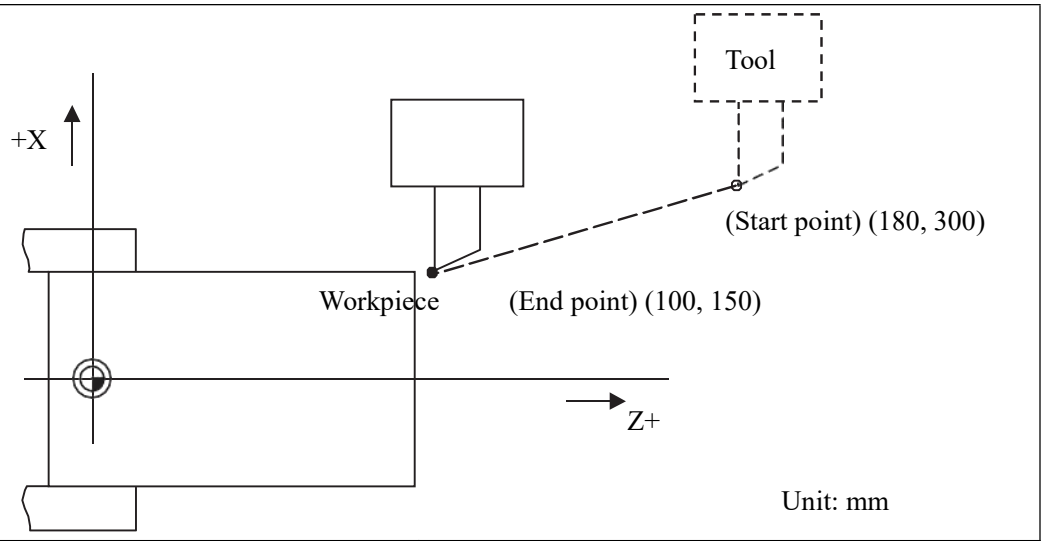
Description

1. The speed of G00 varies depending on the machine specifications and is specified by sys-parameters 0110 and 0112;
2. Controlled by rapid override , independent of the feed rate specified by the F value;
3. G00 motion is divided into motion of each axis at maximum speed and linear simultaneous motion of each axis, as specified by bit 2 of sys-parameter 2203;

	
Linear positioning	Positioning (non-linear positioning) is performed on each axis at its own independent speed.

Example

As shown below, fast positioning is programmed as below:



G00 X100 Z150	// Absolute value command
G00 U-80 W-150	// Incremental value command

2.2.1 Linear Interpolation (G01)

Format

G01 X/U_ Z/W_ α/β_ F_;

X/U : X-axis end point coordinates (X is absolute programming, and U is incremental programming)
Z/W : Z-axis end point coordinates (Z is absolute programming, and W is incremental programming)
α/β : additional axis end point coordinates
(α is absolute programming, and β is used for incremental programming)
(multiple additional axes can be specified)
F : Feed rate, linear axis unit: mm/min (in G98 mode), mm/r (in G99 mode)
Rotating axis unit: 36°/min (in G98 mode), 36°/r (in G99 mode)

Description

1. For G01 interpolation, the calculation of various axes' feed speed is as follows:

G01 Uα Wβ Hγ Ff

Moving distance : $L=\sqrt{(\alpha/2)^2+\beta^2+(\gamma/36)^2}$

Linear axis (diameter programming): X-axis feed speed : $Fx=\frac{a/2}{L}*f$

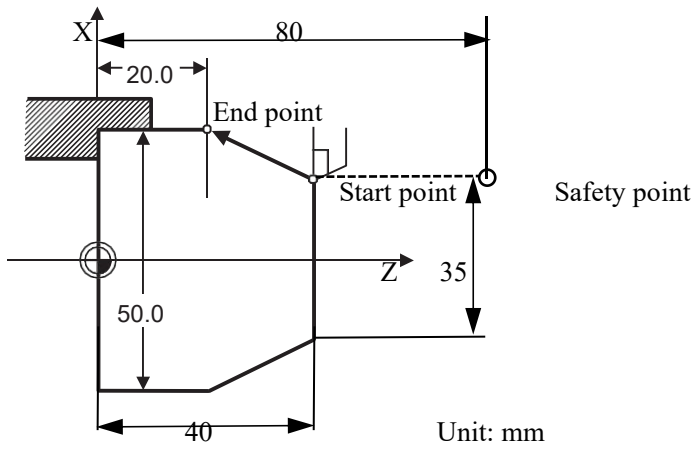
Linear axis (radius programming): Z-axis feed speed : $Fz=\frac{\beta}{L}*f$

Rotating axis: A-axis feed speed : $FA=\frac{\gamma/36}{L}*f$

2. The speed of G01 is specified by F and its upper limit is specified by sys-parameter 0120;
3. The acceleration and deceleration time of G01 is specified by sys-parameter 0121;

Example

The following toolpath performs linear interpolation (diameter programming):



G00 X35 Z80	// Safety point
G01 X40 Z50 F800	// Safety point -> start point
G01 X50 Z20 F300	// Start point -> end point

2.2.2 Circular Interpolation (G02/G03)

Format

Arc in the XpYp plane

$$G17 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Xp\_\ Yp\_ \begin{Bmatrix} I\_\ J\_ \\ R\_ \end{Bmatrix} F\_\ ;$$

Arc in the ZpXp plane

$$G18 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Zp\_\ Xp\_ \begin{Bmatrix} I\_\ K\_ \\ R\_ \end{Bmatrix} F\_\ ;$$

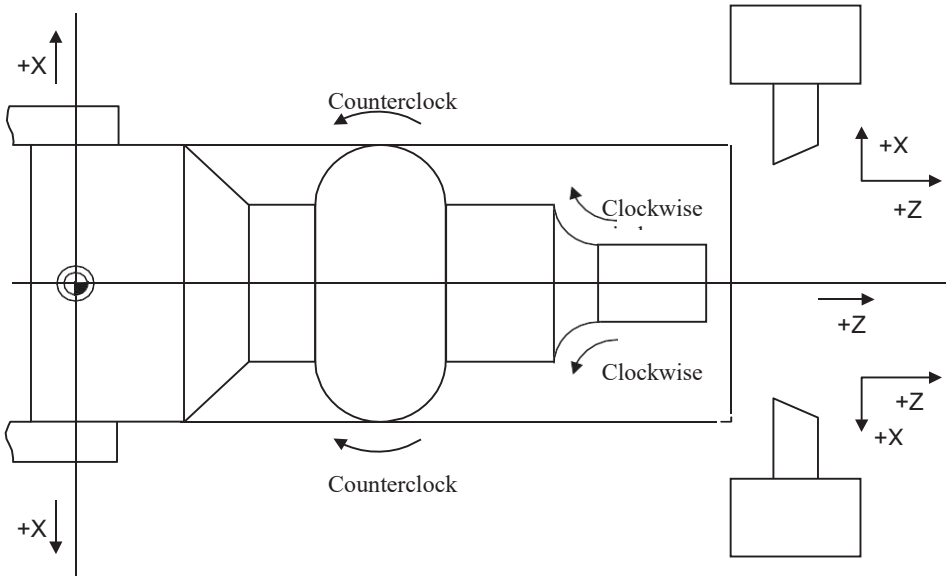
Arc in the YpZp plane

$$G19 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Yp\_\ Zp\_ \begin{Bmatrix} J\_\ K\_ \\ R\_ \end{Bmatrix} F\_\ ;$$

Command	Description
G17	Specification of arc on XpYp plane
G18	Specification of arc on ZpXp plane
G19	Specification of arc on YpZp plane
G02	Circular Interpolation : Clockwise direction (CW)
G03	Circular Interpolation : Counterclockwise direction (CCW)
Xp	Command values of X axis or its parallel axis (set by parameter No. 1022)
Yp	Command values of Y axis or its parallel axis (set by parameter No. 1022)
Zp	Command values of Z axis or its parallel axis (set by parameter No. 1022)
I_	Xp axis distance from the start point to the center of an arc with sign
J_	Yp axis distance from the start point to the center of an arc with sign
K_	Zp axis distance from the start point to the center of an arc with sign
R_	Arc radius (with sign, radius value for lathe cutting)
F_	Feedrate along the arc

Description

1. With regard to the clockwise direction and counterclockwise direction in the right rectangular coordinate system, it is viewed from the positive direction to the negative direction of axis Z.

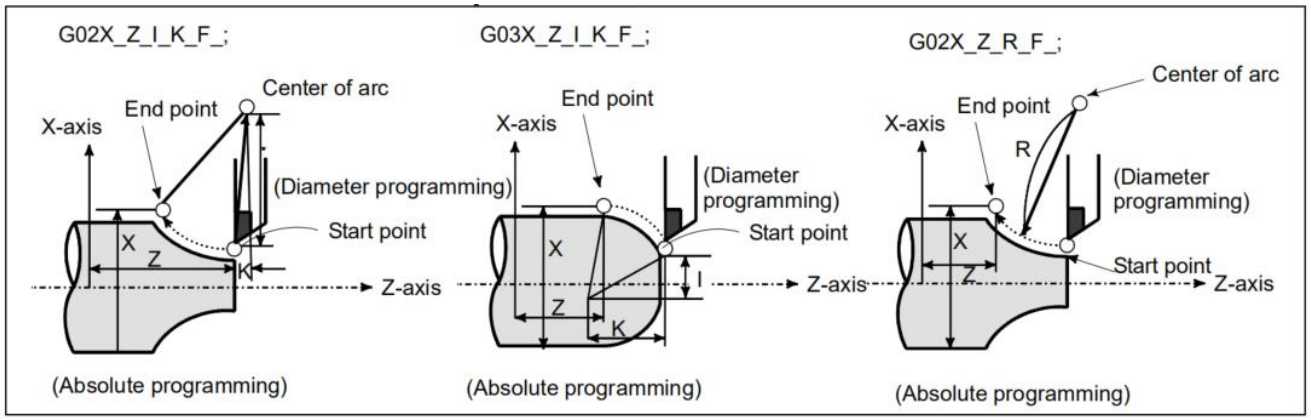


Front tool rest and rear tool rest programming

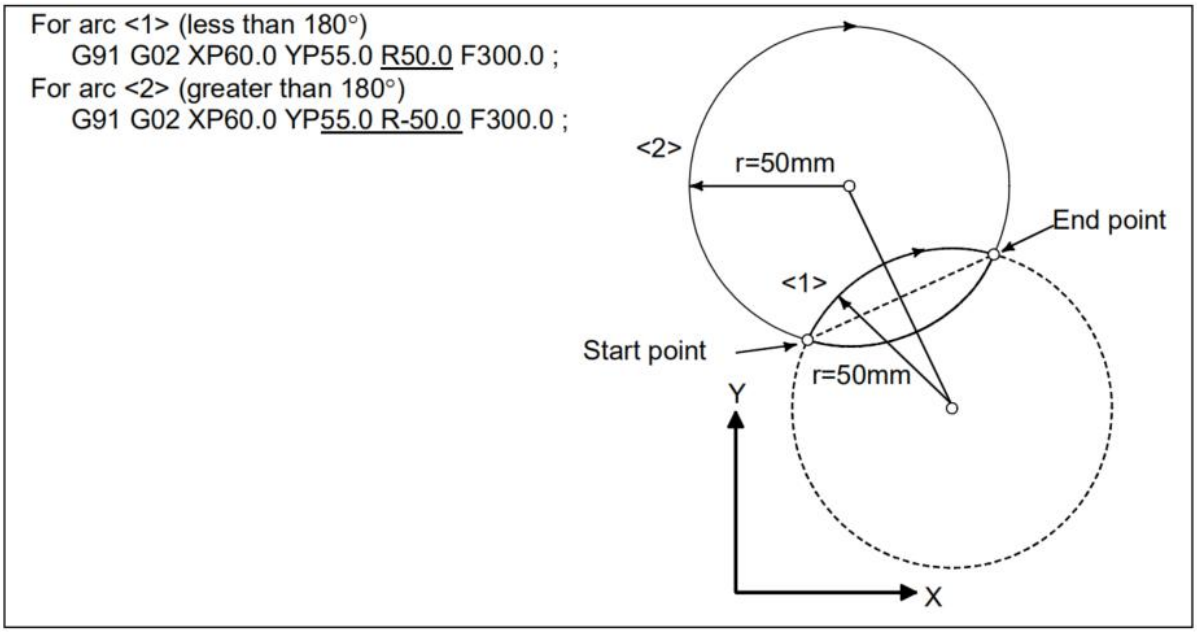
2. I and K programming applies to over quadrant circles and whole circles, but R programming does not apply to whole circles.

3. For I and K programming, I and K are incremental values from the start point to the center of the circle, independent of the diameter programming, but requiring consideration of the sign.

4. If the address X or Z is not programmed, it will be the coordinates of the last segment by default, and if I or K is not programmed, it will be 0 by default;



5. For R programming, R is a signed number, and "+" indicates that the arc angle is less than 180°; "-"means that the arc angle is greater than 180°.

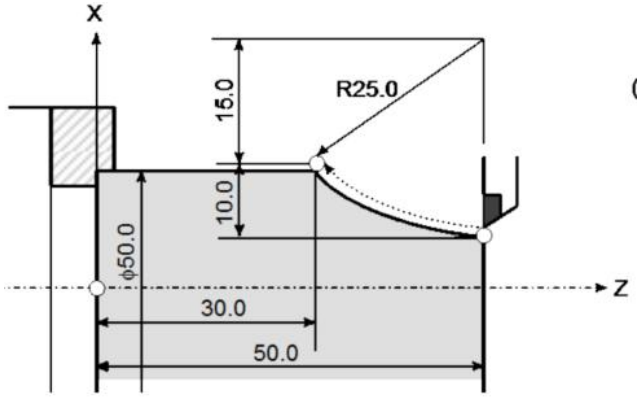


6. If the programming end coordinates or arc radius are specified incorrectly, an alarm will be generated to indicate that the arc radius is out of tolerance. Bit 1 of sys-parameter 2203 can be used to specify whether the CNC will give an alarm when the arc radius is out of tolerance.

7. When the out-of-tolerance judgment function of bit 1 of sys-parameter 2203 is turned on, the arc radius error can be adjusted by modifying the set value of sys-parameter 2210.

Example

As shown in the figure below, the tool performs circular interpolation:



(Diameter programming)
G02X50.0Z30.0I25.0F0.3; or
G02U20.0W-20.0I25.0F0.3; or
G02X50.0Z30.0R25.0F0.3 or
G02U20.0W-20.0R25.F0.3;

2.2.3 Helical Interpolation (G02/G03)

If an axis out of a plane is specified when specifying circular interpolation, the tool can be moved helically, which is called helical interpolation.

Format

[G02/G03 circular interpolation command] α/β_ L_ F_;

α/β : additional axis end point coordinates (α is absolute programming, and β is incremental programming); and multiple additional axes can be specified.

L : number of turns, it is 1 turn by default if not specified.

F : feed speed along the tool path, which is the composite speed of circular and linear motions.

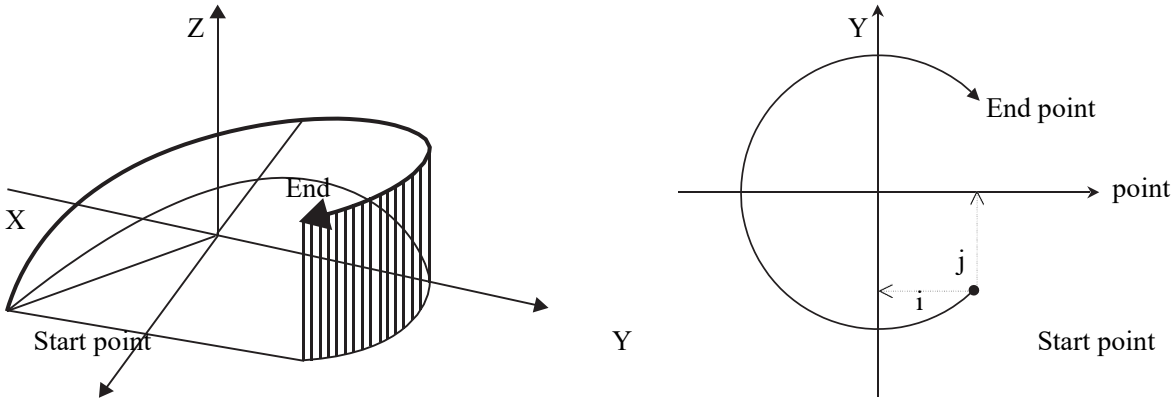
Description

① Interpolation movement based on axes other than X and Y as additional axes in the G17 plane;

② Interpolation movement based on axes other than X and Z as additional axes in the G18 plane;

③ Interpolation movement based on axes other than X and Y as additional axes in the G19 plane;

Helical interpolation path in the G17 plane



Three-dimensional path

Path on the XY plane (projection path)

Example

```
// Taking the turn-milling machine for example, axes X, Y and Z are linear axes.
O1000
G00 X100 Z0 Y0
M103 S2=1000      // Start the rotating milling cutter(2nd spindle)
G17               // Select XY circular interpolation plane
G03 X100 Y0 Z-10 I-50 J0 F500 L3
// Z0 to Z-10, the milling cutter moves helically in a circle of radius 50, for a total of 3 turns
G00 X80
G00 Z100
M105              // Turn off the rotating milling cutter(2nd spindle)
M30
```

2.2.4 Cylindrical Interpolation (G07.1/G107)

Format

G07.1/G107 IP r ;Starts the cylindrical interpolation mode

:

:

G07.1/G107 IP 0 ; The cylindrical interpolation mode is cancelled

IP : One of address for the rotary axis

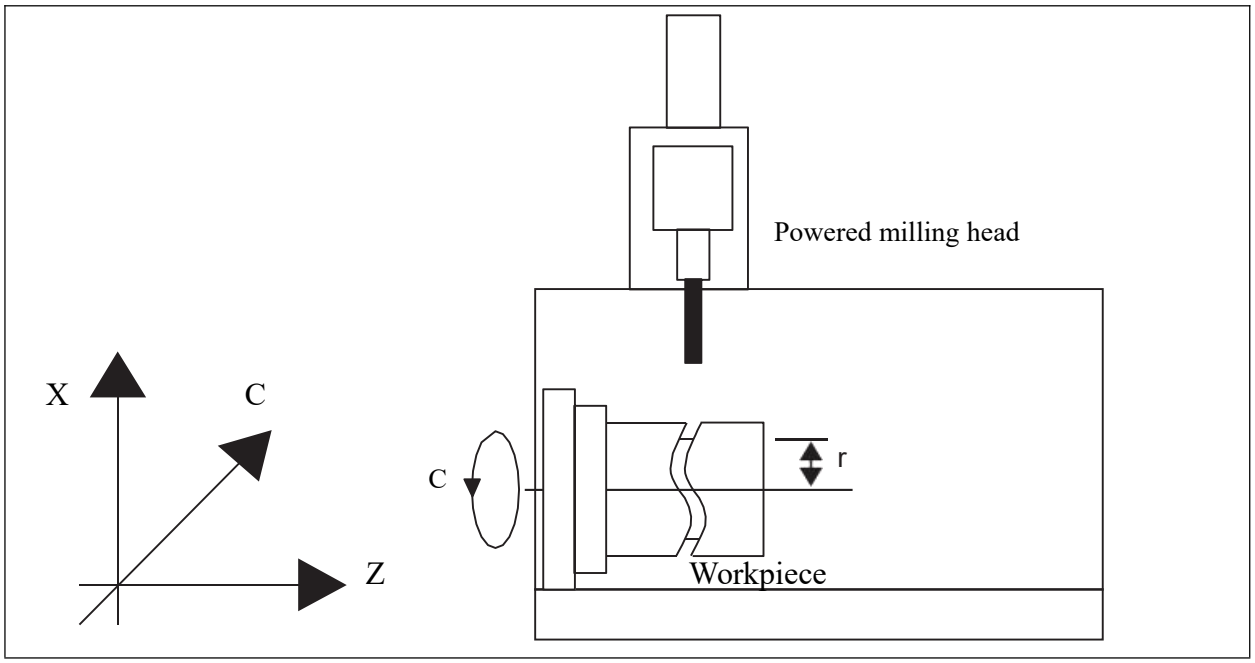
r : The radius of the workpiece

Specify G07.1 IPr; and G07.1 1P0; in separate blocks

G107 can be used instead of G07.1.

Description

This function unfolds the side of the cylinder (the shape on the cylinder coordinate system) into a plane. After the unfolded shape is programmed as a plane coordinate, it is converted into the movement of the cylinder coordinate on the linear axis and rotating axis (C axis) to perform contour control when machining is performed.



1. Before using the command G07.1, it is necessary to specify which the rotating axis is parallel to in the CNC parameter 0016;

2. The coordinate command in the interval from the beginning of cylindrical interpolation mode to the cancellation is the cylindrical coordinate system command;

3. G107 can be used to replace G07.1;

4. G07.1 must be a single line, and other G commands cannot be specified;

5. Programming the rotating shaft by angle;

6. In cylindrical interpolation, linear interpolation and circular interpolation commands are available. However, the plane selection command must precede the G07.1 program segment;

7. Coordinate commands can be absolute commands or incremental commands.

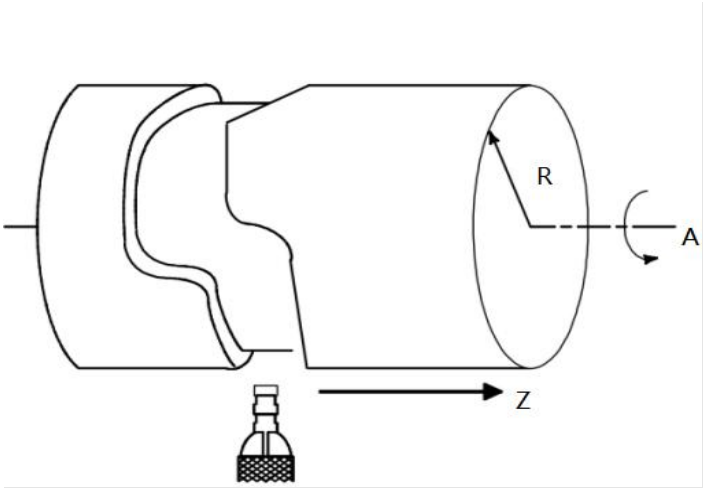
8. In the process of cylindrical interpolation, tool radius compensation G40/G41/G42 can be performed.

9. In the process of cylindrical interpolation, the feed speed of F is specified as the tangent speed on the plane of cylindrical expansion;

10. In the process of cylindrical interpolation, the movement of the rotating shaft is specified by angle, which is converted into the distance on the circumference, and then converted into angle again after linear and circular interpolation with other axes. Therefore, when the radius of the cylinder is small, the actual movement amount may be inconsistent with the specified value;

11. In the CNC graphical interface, it is not possible to perform program previews for programs with cylindrical interpolation commands;

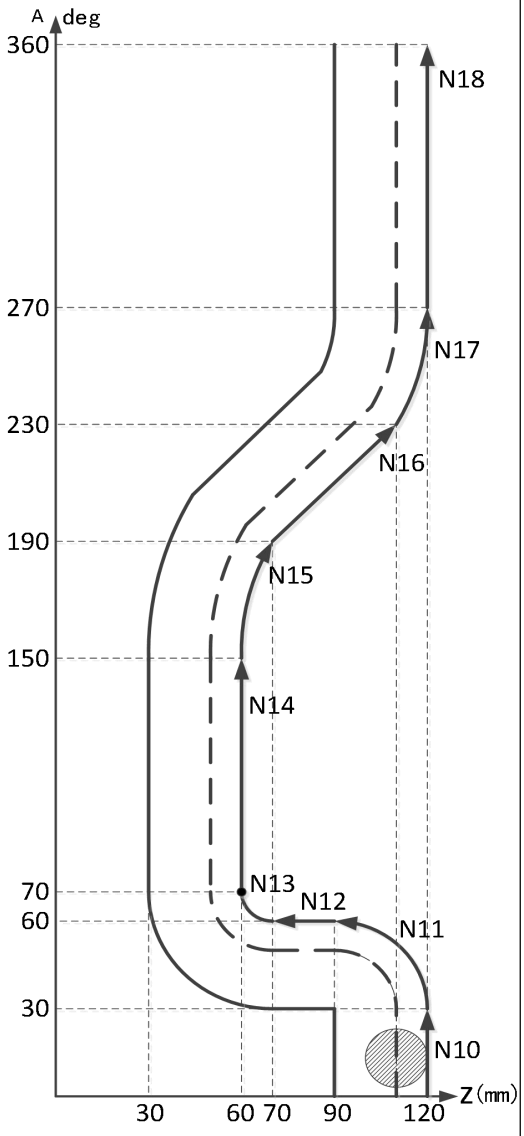
Example



Workpiece diameter: 125mm Milling depth: 2.5mm (radius value)

The A axis of the rotating axis is set as the parallel axis of the linear axis Y (sys-parameter 0016 A:6)

Expansion plan of cylindrical surface:




```
O1073 (cylindrical interpolation G19-ZOA plane)
N01 T0101
N02 G00 X125 Z100 A0 // Positioning
N03 M103 S2 = 1000 // Start the 2nd spindle
N04 G01 X120 F300
N05 G19 W0 H0 // Plane selection, interpolation is performed on axes Z and A
N06 G07.1 A60 // Cylindrical interpolation starts with a machining radius of 60
N07 G01 G42 Z120
N10 G01 A30
N11 G02 Z90 A60 R30.0
N12 G01 Z70
N13 G03 Z60 A70 R10
N14 G01 A150
N15 G03 Z70 A190 R75
N16 G01 Z110 A230
N17 G02 Z120 A270 R75
N18 G01 A360
N20 G40 Z100
N21 G07.1 A0 // Cylindrical interpolation ends
N22 G00 X125
N23 M30
```

2.2.5 Polar Coordinate Interpolation (G12.1/G13.1)

Format 1

G12.1;	: Polar interpolation mode starts
...	
G13.1;	: Polar interpolation mode is canceled

Format 2

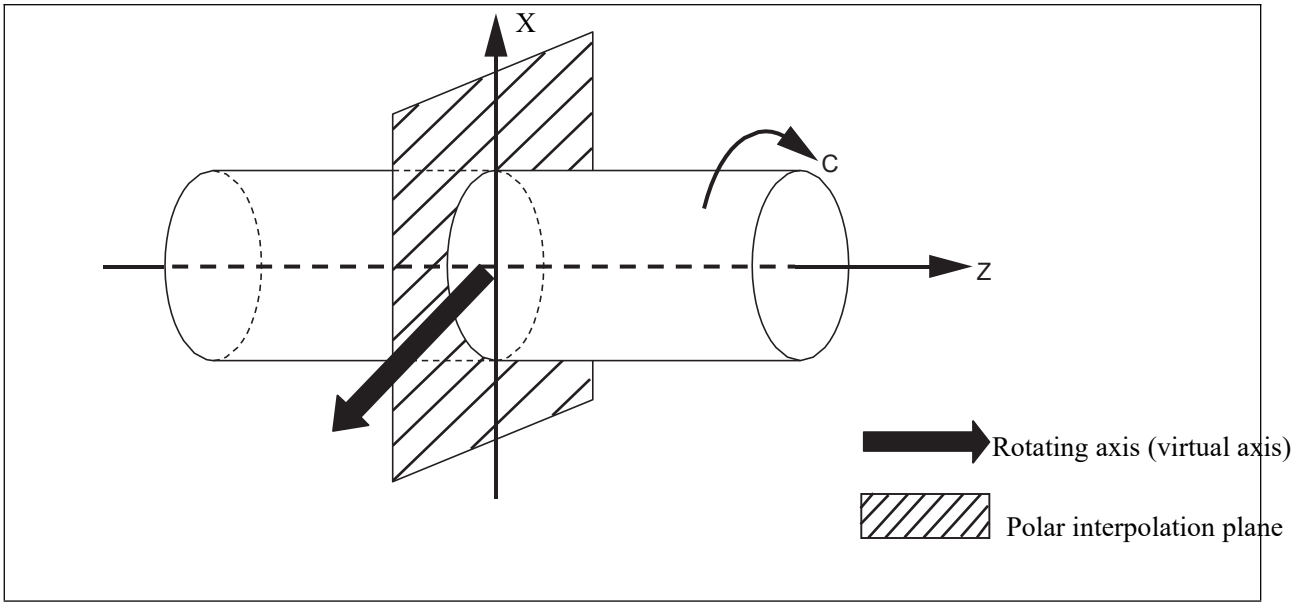
G16;	: Polar coordinate system programming mode starts
G12.1;	: Polar interpolation mode starts
...	
G13.1;	: Polar interpolation mode is canceled
G15;	: Polar coordinate system programming mode is canceled

Description

It converts the commands programmed in the vertically intersecting coordinate system into movement of the vertically intersecting axes (tool movement) and movement of the rotating axes (workpiece rotation) and performs contour control.

It specifies the linear axis as the vertically intersecting axis of the 1st axis of the plane, the vertically intersecting virtual axis as the plane of the 2nd axis of the plane, which is known as the polar interpolation plane, and performs polar interpolation in this plane.

It is generally used for milling of anisotropic workpieces, grinding of cams, etc.



Detailed description

1. Need to set the polar interpolation linear axis sys-parameter 5530, and polar interpolation rotating axis sys-parameter 5531;

2. G12.1 and G13.1 must be specified separately by a single line in the program section and other commands cannot be contained;

3. Either diameter programming or radius programming may apply to linear axes in polar coordinate commands;

4. In Format 1, the CNC will select the linear axis as the first axis of the plane and the rotating axis as the second axis of the plane (imaginary axis) to form a rectangular coordinate system as the polar interpolation plane. The specified unit of the second axis is the same as that of the first axis, but the second axis is subject to radius programming, and the radius or diameter programming of the first axis is specified by bit 1 of sys-parameter 0001.

5. In Format 2, the rotating axis is used as the second axis of the polar interpolation plane and is converted into a virtual linear axis, but is directly used as angular axis for programming. Its unit is degree.

6. G13.1 cancel polar interpolation, and restore the rotating axis unit. The plane before performing G12.1 is recovered;

7. The feed speed in polar interpolation is the tangent synthesis speed, and the speed unit is the same as the first axis;

8. G codes that can be executed in polar interpolation mode:

G01 (,C/R) (,A)	Linear interpolation, automatic chamfering, linear angle
G02, G03	Circular interpolation, you can specify the arc radius or arc with R or I, J.
G04	Time delay
G15, G16	Polar coordinate system
G40, G41, G42	Tool nose radius compensation
G65, M98	Subprogram call
G90, G91	G-code system B, absolute programming and incremental programming
G98, G99	Feed per revolution / Feed per minute

9. When tool nose radius compensation is used in the polar coordinate interpolation mode, it is necessary to ensure whether the tool nose phases match.

10. Regardless of the current coordinate value of the rotating axis as a virtual axis, the value of the virtual axis coordinates always starts from 0 when the G12.1 command is performed. Therefore, it should be ensured that the rotating axis is positioned before the execution of the G12.1 command.

Restrictions:

1. In the polar interpolation mode, the coordinate system must not be changed (G50, G52, G53, G17~G19, G54~G59, etc.);

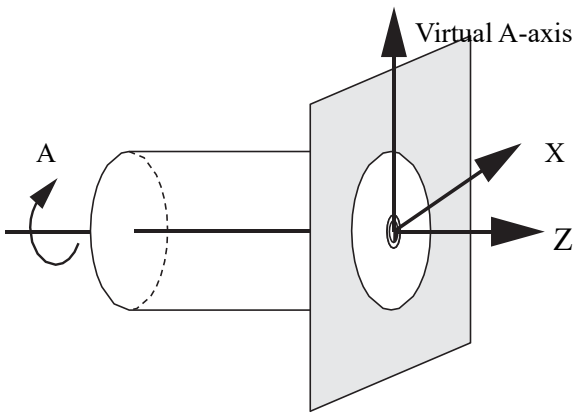
2. In the polar interpolation mode, the tool compensation number must not be changed;

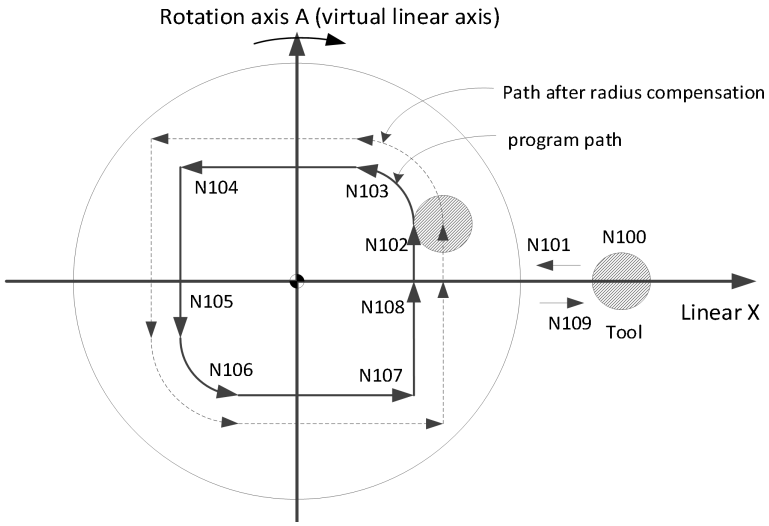
3. Under nose radius compensation (G41,G42), polar interpolation mode (G12.1,G13.1) cannot be switched. Nose radius

Commands G12.1 and G13.1 are executed only when the compensation is canceled (G40);

4. In the CNC graphical interface, it is not possible to perform a quick preview of the program with polar interpolation commands;

Example





Let: the rotating axis A as the second axis of the polar interpolation plane; A axis is ranked as 3 in the position interface

sys-parameter setting:

Bit 2 of 0001 sets A-axis as a rotating axis

5530 is set to 1 (X-axis is used as the first axis of the polar interpolation plane - linear axis (diameter programming))

5531 is set to 3 (A-axis is used as the second axis of polar interpolation plane - rotating axis)

O1000	
T0101	
G00 X120 A0 Z0	
G12.1	// Polar interpolation starts
N101 G42 G01 A0 X40 F500	
N102 G01 A10	
N103 G03 X20 A20 R10	// Specify the radius of the arc with R
N104 G01 X-40	
N105 G01 A-10	
N106 G03 X-20 A-20 I10 J0	// Specify the center of the arc with I and J
N107 G01 X40	
N108 G01 A0	
N109 G40 X120	
G13.1	// Cancel polar interpolation
M30	

2.2.6 Encoder Thread Feed (G32)

Format

G32 X/U_ Z/W_ F/I_ Q_;

X/U : X-directional coordinate position of the thread end point
(X is absolute programming, and U is incremental programming)

Z/W : Z-directional coordinate position of the thread end point
(Z is absolute programming, and W is incremental programming)

F : Metric thread lead, lead in the long axis direction, unit: mm

I : Inch screw thread lead, number of teeth per inch in the long axis direction

Q : The starting angle of thread processing defaults to 0° when not specified.

Extension characters

α/β : additional axis end point coordinates
(α is absolute programming, and β is incremental programming)
(multiple axes can be specified)

J : Retracted length in the long axis direction (valid only in high-precision mode), unit: mm

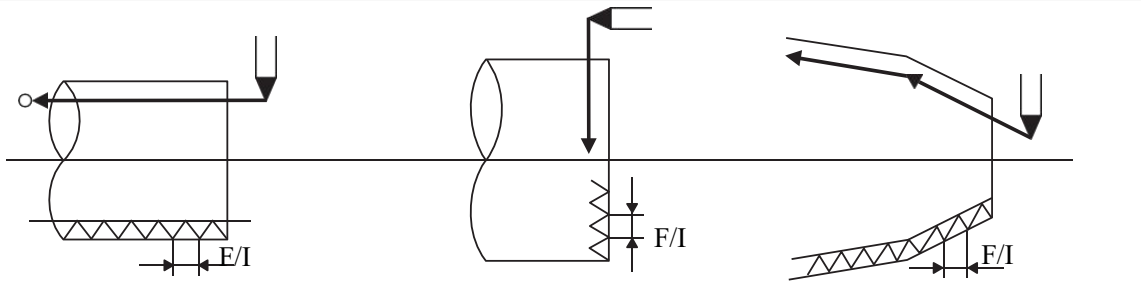
K : The amount of lead change per turn, in mm

NQ1 : When NQ1 is specified, the command does not detect the zero pulse of the encoder and is used for rollback of broken tap

Description

The command controls the feed axis for thread cutting through feedback from the machine's spindle rotary encoder.

1. G32 thread command applies to the processing of straight thread, taper thread, end plane thread, continuous thread, and variable lead thread;



Straight thread End plane thread
Continuous thread

2. G32 thread command execution flow:

① Spindle starts;

② Whether the spindle speed feedback is greater than the value set by sys-parameter 5025 (lower limit of thread processing spindle speed), and if the waiting time exceeds the time set by sys-parameter 5027, the CNC pauses to alarm;

③ Spindle speed feedback is compared with the speed specified by the S command, and if the speed difference exceeds the range set by sys-parameter 5026, the CNC pauses to wait; and if the waiting time exceeds the time set by sys-parameter 5027, the CNC pauses to alarm;

④ With regard to the judgment of the upper limit of thread cutting speed, if the spindle feedback

3. When the command in the next segment of the G32 command is the G00 command and it is in the high precision mode (bit 2 of sys-parameter 5000), there will be a thread vanishing action and the thread vanishing length in the long axis direction can be set in the following three ways:

① J in the G32 command segment specifies the thread vanishing length;

② When bit 1 of sys-parameter 5000 is 0 and J is not programmed, the thread vanishing length is specified by sys-parameter 012;

③ When bit 1 of sys-parameter 5000 is 1 and J is not programmed, the thread vanishing length is specified by sys-parameter 5016;

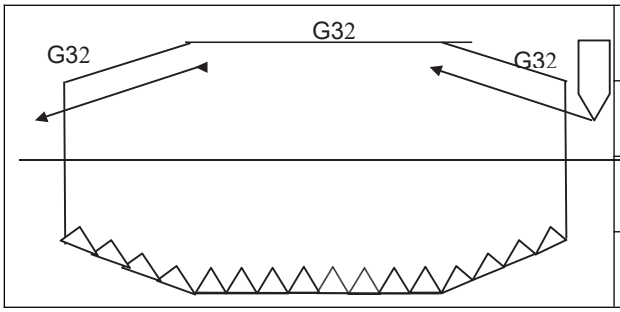
4. The thread vanishing angle is specified by sys-parameter 5017;

5. When the command in the next segment of the G32 command is the G01 command, there is no thread vanishing action;

6. G32 command applies to continuous thread cutting, G32 in the second segment can be omitted for continuous cutting, and different lead lengths can be specified;

7. When continuous thread cutting is performed, it needs to set bit 2 of sys-parameter 5000 to smooth mode;

8. When a continuous thread cutting command is executed, continuous thread cutting will not be performed if an command other than thread cutting is executed. However, if a command without axis movement (G04 pause command, MST command, etc.) is specified between the program segments of the thread cutting command, the G32 command will wait for the spindle 1-turn synchronization signal (encoder zero pulse) again starting from the second program segment;



G32 U10 W-20 F1;	// Thread cutting
U0 W-20 F2;	// Continuous thread cutting
U-10 W-20 F1;	// Continuous thread cutting
G00 U10;	// Tool retracting

9. the CNC will determine whether the long axis is the first axis 1 or the second axis of the interpolation plane based on the calculation of the current coordinate position and the end coordinate position;

10. The thread lead F/I corresponds to the lead in the long axis direction;

11. The long axis for thread cutting can be specified by means of the G17 to G19 plane selection commands;

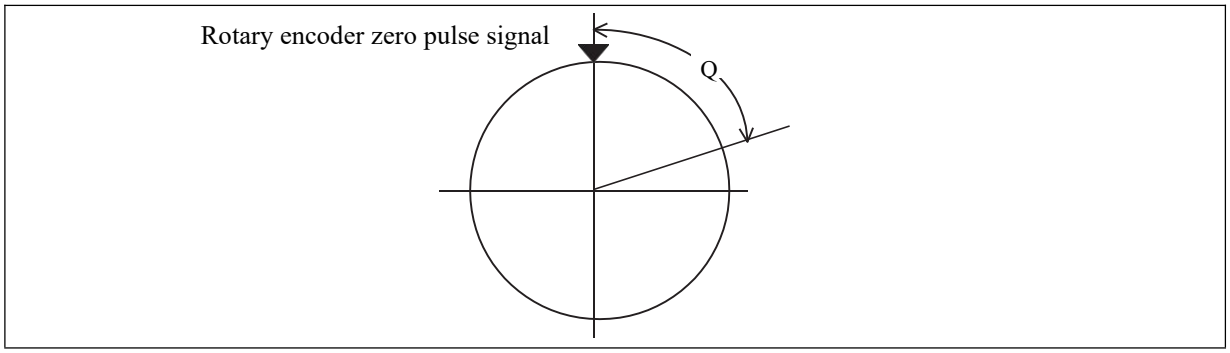
12. The G32 command can specify the movement of multiple additional axes, and the movement speed of the additional axes depends on the long axis of the G32 command;

G18 G00 X30 Z5	Select X-axis and Z-axis as plane interpolation axes (system default)
G32 U-5 W-10 F0.75	The distance moved on Z-axis is greater than the distance moved on X-axis The Z-axis is the long axis, and the command ends after estimating that the spindle has rotated for [distance moved on Z-axis / lead] turns.
G18 G00 X55 Z-2	Select X-axis and Z-axis as plane interpolation axes (system default)
G32 U-55 W2 F0.75	The distance moved on X-axis is greater than the distance moved on Z-axis The X-axis is the long axis, and the command ends after estimating that the spindle has rotated for [distance moved on X-axis / lead] turns.
G17 G00 X50 Y0 Z0	Select X-axis and Y-axis as plane interpolation axes
G32 V-50 U-5 Z-5 F2	The distance moved on Y-axis is greater than the distance moved on X-axis The Y-axis is the long axis, and the command ends after estimating that the spindle has rotated for [distance moved on Y-axis / lead] turns.

13. The unit of Q is 1° by default (can be specified as 0.001° by the bit 3 of sys-parameter 5000), and the CNC will automatically carry out modulo operation, if the specified angle of Q exceeds 360;

14. The CNC system establishes the spindle angular coordinate system, the start point of Q, by means of the rotary encoder 1-turn signal (encoder zero-pulse signal);

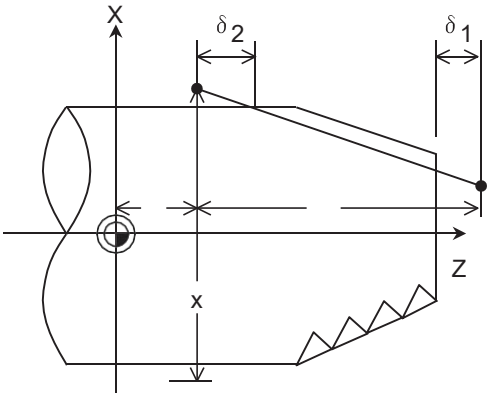
15. It starts moving after pausing the execution of G32 command to wait for the 1-turn signal from the rotary encoder (spindle angle of [0+Q]);



16. When G32 is executed in automatic mode, if the mode is switched to another mode (e.g., manual mode, handwheel mode), the CNC is in a suspended state after the current G32 command is executed;

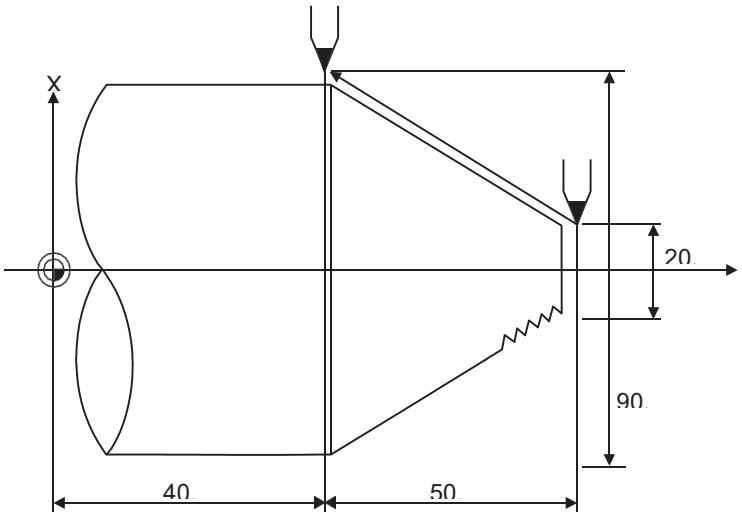
17. At the beginning/end of thread cutting, a lead error is usually caused by a delay in the servo drive. Therefore, when the thread length is specified, it must specify the length after the wrong lead length of δ1 and δ2 has been added to the required thread length.

δ2 is the thread vanishing length, which is calculated according to the calculation method specified by bit 1 of sys-parameter 5000, and bit 4 of sys-parameter 5000 specifies whether δ1 is equal to δ2.



18. The feed override, spindle override, feed hold key, single-segment function and hand pulse debugging function are invalid when the G32 command is running;

Example

General thread processing, thread lead: 1 mm	
	O1000; T0101; M03 S300; G00 X20 Z90; // Positioning **G32 X90 Z40 F1**; // Thread cutting G00 U10; // Retracting G00 Z90; M30;

Processing of oil groove with a length of 50 mm. (Need to set the bit 2 of sys-parameter 5000 to high precision mode)	
T0101;	
M03 S150	
G00 X49 Z1	
N10	
G32 U0.1 W-50 F100	
G32 U0.1 W50 F100	
M92 N10 L5	// Jump to N10, repeat the cycle 5 times, process until the diameter is 50 mm
G00 X30	// Tool retracting
G00 Z90;	
M30;	

2.2.7 Interpolation Thread Feed (G32.1)

Format

G32.1 X/U_ Z/W_ F/I_ S_;

X/U : X-directional coordinate position of the thread end point
(X is absolute programming, and U is incremental programming)

Z/W : Z-directional coordinate position of the thread end point
(Z is absolute programming, and W is incremental programming)

F : Metric thread lead, lead in the long axis direction, unit: mm

I : Inch screw thread lead, number of teeth per inch in the long axis direction

S : Cannot be omitted, used for spindle speed setting, unit: r/min

Extension characters

α/β : additional axis end point coordinates
(α is a programming command, and β is an incremental programming command);
(multiple axes can be specified)

J : The thread vanishing length in the direction of the long axis of the thread, unit: mm

Description

The function is basically the same as the G32 command, except that the G32.1 command does not require spindle encoder feedback, but interpolates the X and Z axes through the rotating axis set by the spindle to perform thread cutting.

1. Before the G32.1 command is used, the spindle must be in M05 and the spindle must be in position mode;
2. When the G32.1 command is used, S must be specified in the line where the command is located;
3. When the G32.1 command is running, the spindle speed is influenced by the feed override but not by the spindle override;
4. The running of G32.1 command is affected by the hand pulse debugging function, single segment function and feed hold key (pause key);
5. G32.1 cannot specify Q (starting machining angle), and the rotating axis can execute G00 positioning command to achieve the deflection of the machining starting angle before G32.1 is executed;

Example

Assume: The spindle is a servo spindle, and the rotating axis A is used as the interpolation axis of the spindle position mode for rotary milling	
O1000	
T0101	
G00 X80	// Locate the position according to the workpiece diameter
G00 Z5	
M05	// Make sure the 1st spindle stops
M103 S2=1000	//Turn on the 2nd spindle (powered milling head)
M18	// 1st spindle position mode switching
G00 A60	// 1st spindle positioning, start cutting from the position of 60 degrees
G01 U-2 F10	// Cutting feed
G32.1 W-20 F20 S30	// The lead of the first segment processing is 20, thread interpolation is carried out between A-axis and Z-axis, and the spindle (A-axis) speed is 30 r/min.
G32.1 W-30 F30 S30	// The lead of the second segment processing is 30, thread interpolation is carried out between A-axis and Z-axis, and the spindle (A-axis) speed is 30 r/min.
G00 U10	// Tool retracting
G00 Z5	
M17	// The 1st spindle switches back to speed mode
M30	

2.2.8 Circular Thread Cutting (G32.2/G32.3)

Format

G32.2/G32.3 X/U_ Z/W_ R_ F/I_ Q_ S_;

G32.2 : Clockwise circular thread
G32.3 : Counterclockwise circular thread
X/U) : X-directional coordinate position of the thread end point
(X is absolute programming, and U is incremental programming)
Z/W : Z-directional coordinate position of the thread end point
(Z is absolute programming, and W is incremental programming)
R : Radius of arc, unit: mm
F : Metric thread lead, lead in the long axis direction, unit: mm
I : Inch screw thread lead, number of teeth per inch in the long axis direction
Q : The starting angle of thread processing defaults to 0° when not specified.

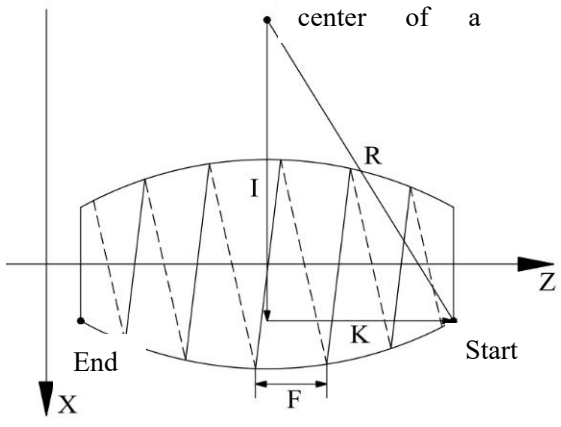
Extension characters

α/β : additional axis end point coordinates
(α is a programming command, and β is an incremental programming command);
(multiple axes can be specified)
J : The thread vanishing length in the direction of the long axis of the thread, unit: mm
K : The amount of lead change per turn, in mm

Description

1. The R value of the G32.2/G32.3 command must be greater than or equal to the thread length;
2. The lead on the arc is the value in the direction of the linear axis;
3. When S is not specified, the command is in encoder cutting mode (same as the G32 command);
4. When S is specified, the command is in interpolation cutting mode (same as G32.1 command);

Example




```
O1000
T0101
M03 S500
G00 X30 Z5
G32.3 W-10 R20 F2
G00 U10
G00 Z5
M30
```

2.2.9 Variable Pitch Thread Cutting (G34)

Format

G34 X/U_ Z/W_ F/I_ Q_ K_;

X/U : X-directional coordinate position of the thread end point
(X is absolute programming, and U is incremental programming)

Z/W : Z-directional coordinate position of the thread end point
(Z is absolute programming, and W is incremental programming)

F : Metric thread lead, lead in the long axis direction, unit: mm

I : Inch screw thread lead, number of teeth per inch in the long axis direction

Q : The starting angle of thread processing defaults to 0° when not specified.

K : The amount of lead change per turn, in mm

Command extension characters:

α/β : additional axis end point coordinates
(α is absolute programming, and β is incremental programming)
(multiple axes can be specified)

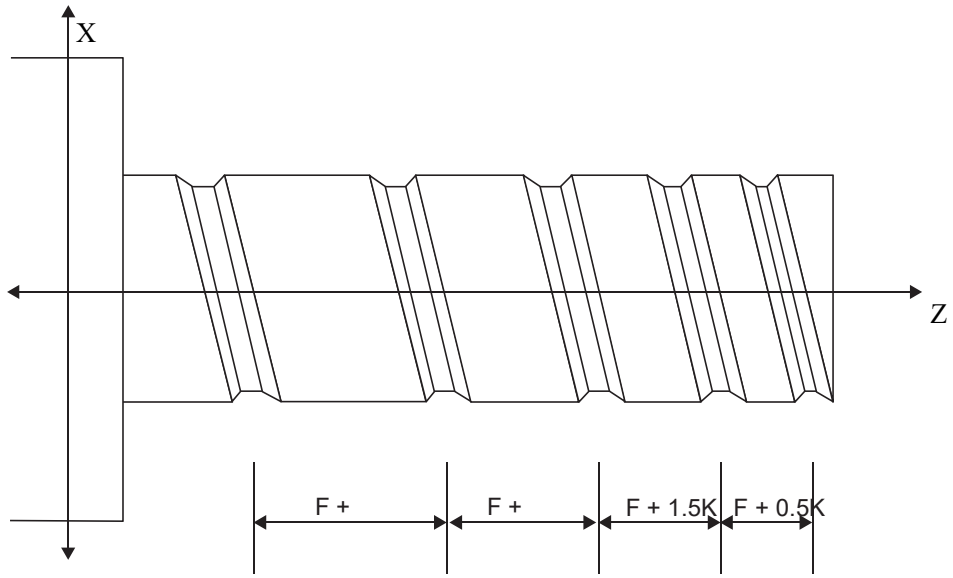
J : The thread vanishing length in the direction of the long axis of the thread, unit: mm

NQ :When NQ1 is specified, the command does not detect the zero pulse of the encoder.

Description

The function is the same as that of G32 command.

Schematic diagram of variable lead thread



The amount of change in the lead per turn is K

1. A positive value of K gradually increases with the lead per turn and a negative value of K gradually decreases with the lead per turn;

2. Relationship between K, F, and the length of the long axis:

The number of turns N, the lead F, and the amount of change K are known Find the length L of the thread long axis: Formula: $L = N^*(F + 0.5^*K) + N^*(N-1)^*K$	The length L of the thread long axis, the lead F, and the amount of change K are known Find the number of turns N: Formula: $endPitch = \sqrt{F^2 + 2 * K * L}$ $N = (-F + endPitch)/K$

2.3 PAUSE FUNCTIONS

2.3.1 Time Delay (G04)

Format

G04 X/U/P_;	
X	: delay time in seconds
U	: delay time in seconds
P	: delay time in 0.001 seconds

Description

Pause program delays the execution of the next program segment, and the delay time is the time of command.

2.3.2 Input Signal Detection (G04 I)

Format

G04 I_ J_ K_;	
I	: Input port settings, a positive value of I indicates the connection is valid, and a negative value of I indicates the disconnection is valid.
J	: Maximum waiting time for valid input signal, unit: seconds
K	: Effective width of input signal, unit: seconds

Description

This command allows you to detect the CNC input port status, and the function is similar to M01.

It can be specified at the same time with X/U/P of delay command G04. The delay time and input port detection have a logic relationship that if either of the delay time condition or input signal condition is satisfied, the execution of G04 command is completed.

Example

O1000	
T0101	
G04 X1	// Delay 1 seconds
M20 K1	// Output port with port number 1 is open (set to 1)
G04 X5 I2	// After a delay of 5 seconds or when the number 2 input port is detected, the CNC executes the next program segment
M21 K1	// The output port with port number 1 is closed (set to 0)
...	
M30	

2.4 SIMPLIFY / LOOP PROGRAMMING

2.4.1 Automatic Chamfering (,C and ,R)

Format

,C/R_ E_;	
,C	: Width of flat chamfering angle, unit: mm
,R	: Radius of arc chamfering angle, unit: mm
E	: Chamfering feed rate, linear axis unit: mm/min (in G98 mode), mm/r (in G99 mode)

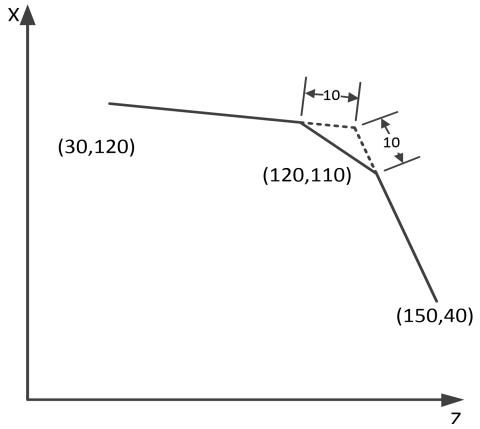
Description

The chamfering command can be generated automatically between program segments.

1. If both ",R" and ",C" are specified in the same command line, ",C" is valid;
2. Automatic chamfering commands are subject to the single segment function of the program;
3. The automatic chamfering command can specify E to specify the feed speed, or use the feed speed of F if E is not specified;

The program commands are described based on the G18 plane (ZOX) as below:

1. Straight line to straight line

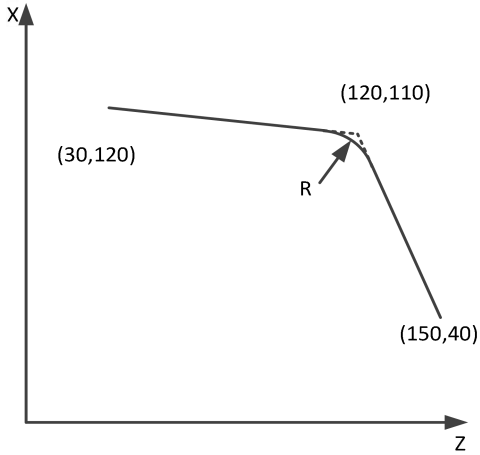


Absolute programming:

G00 X40 Z150;
G01 X110 Z120 F500 **,C10**;
G01 X120 Z30.0;

Relative programming:

G00 X40 Z150;
G01 U70 W-30 F500 **,C10;**
G01 U10 W-90;



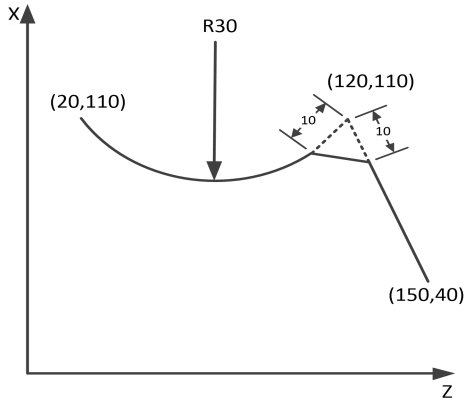
Absolute programming:

G00 X40 Z150;
G01 X110 Z120 F500 **,R10**;
G01 X120 Z30;

Relative programming:

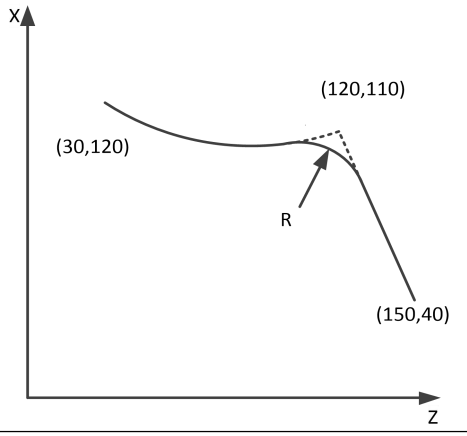
G00 X40 Z150;
G01 U70 W-30 F500 **,R10**;
G01 U10 W-90;

2. **Straight line to arc**



Absolute programming:
G00 X40 Z150;
G01 X110 Z120 F500 **,C10**;
G02 X110 Z20 R30;

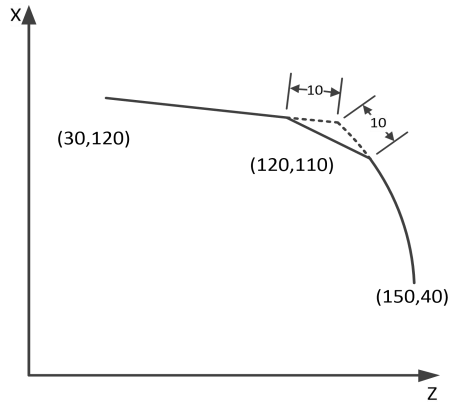
Relative programming:
G00 X40 Z150;
G01 U70 W-30 F500 **,C10**;
G02 W-100 R30;



Absolute programming:
G00 X40 Z150;
G01 X110 Z120 F500 **,R10**;
G02 X110 Z20 R30;

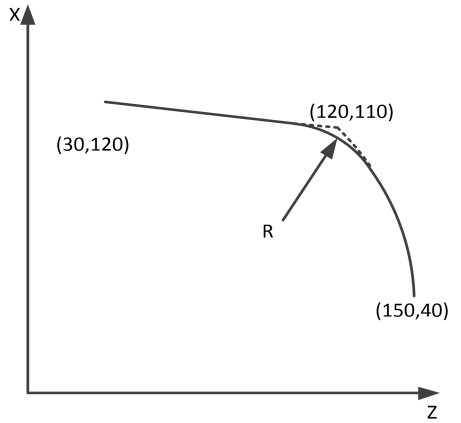
Relative programming:
G00 X40 Z150;
G01 U70 W-30 F500 **,R10**;
G02 W-100 R30;

3. **Arc to straight line**



Absolute programming:
G00 X40 Z150;
G03 X110 Z120 R50 F500 **,C10;**
G01 X120.0 Z30;

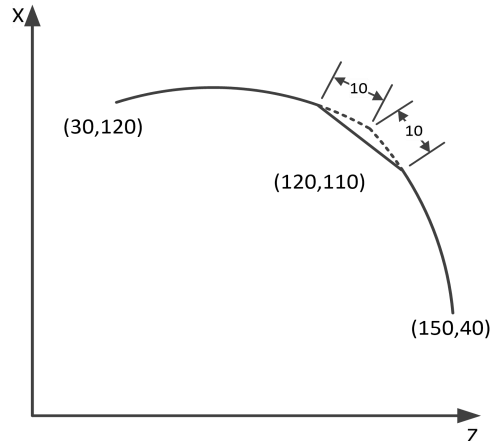
Relative programming:
G00 X40 Z150;
G03 U70 W-30 R50 F500 **,C10;**
G01 U10 W-90;



Absolute programming:
G00 X40 Z150;
G03 X110 Z120 R50 F500 **,R10;**
G01 X120.0 Z30;

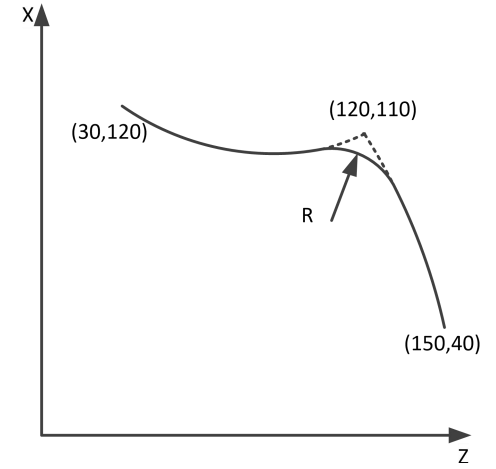
Relative programming:
G00 X40 Z150;
G03 U70 W-30 R50 F500 **,R10;**
G01 U10 W-90;

4. **Arc to arc**



Absolute programming:
G00 X40 Z150;
G03 X110 Z120 R50 F500 **,C10**;
G03 X120 Z30 R60;

Relative programming:
G00 X40 Z150;
G03 U70 W-30.0 R50 F500 **,C10**;
G03 U10 W-90 R60;



Absolute programming:
G00 X40 Z150;
G03 X110 Z120 R50 F500 **,R10**;
G02 X120 Z30 R60;

Relative programming:
G00 X40 Z150;
G03 U70 W-30.0 R50 F500 **,R10**;
G02 U10 W-90 R60;

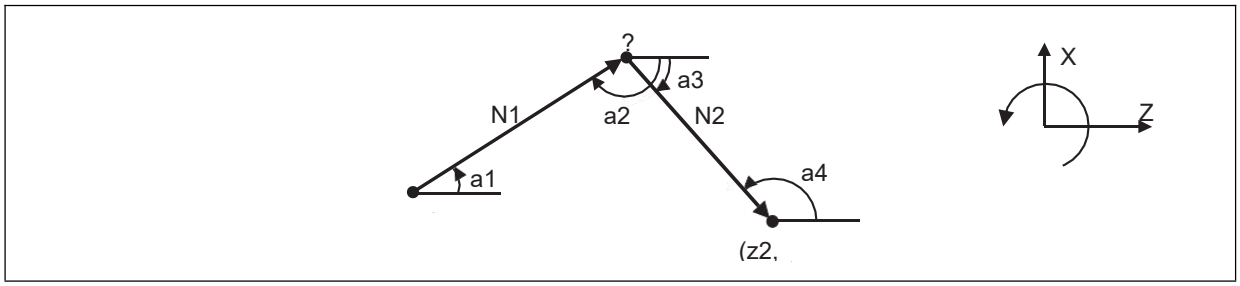
2.4.2 Linear Angle (,A)

Format

G01 X/U_ Z/W_ ,A_ F_;	
X/U	: X-axis end point coordinates (X is absolute programming, and U is incremental programming)
Z/W	: Z-axis end point coordinates (Z is absolute programming, and W is incremental programming)
,A	: Angle value, unit: degree
F	: Feed rate, unit: mm/min (in G98 mode), mm/r (in G99 mode)

Description

When it is difficult to calculate the intersection point of 2 lines in a continuous linear interpolation command, you can control the movement command by specifying the angle of the 1st line and the coordinate value and angle of the end point of the 2nd line, and automatically calculating the end point of the 1st line inside NC.



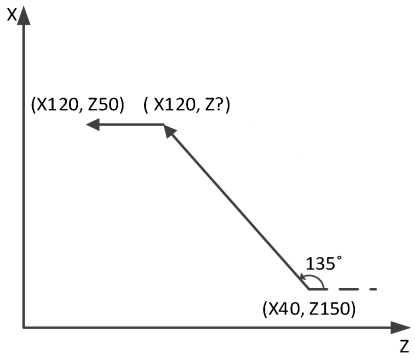
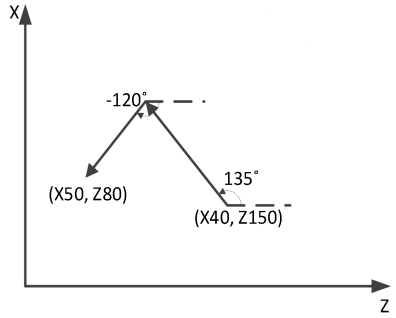
1. The angle is the angle with the direction of the 1st axis (horizontal axis) of the selected plane, with counterclockwise (CCW) as positive and clockwise (CW) as negative;

2. The angle range is -360.000 ≦ A ≦ 360.000, if a value outside this range is specified, the command value is the remainder of the original value divided by 360. For example, when it is specified to be 400, the command angle is the remainder 40 of 400/360;

3. The plane switching between the 1st and 2nd program segments (G17/G18/G19) is not allowed;

4.

Example

Case 1: 	**Absolute programming:** G00 X40 Z150 G01 X120 F500 **,A135** G01 X120 Z50 **Relative programming:** G00 X40 Z150 G01 U80 F500 **,A135** G01 X120 Z50
Case 2: 	**Absolute programming:** G00 X40 Z150 G01 ,A135 F500 G01 X50 Z80 **,A-120**

2.4.3 Finishing Cycle (G70)

Format

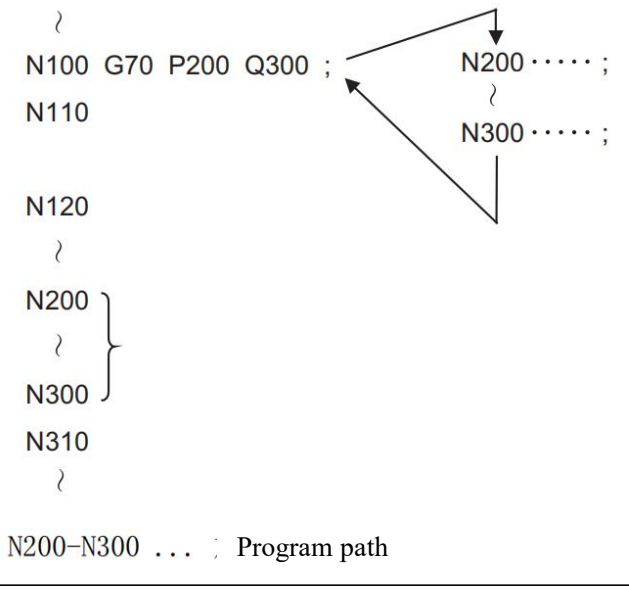
G70 P_ Q_ L_;

P	: Program segment number corresponding to the beginning of the machining path
Q	: Program segment number corresponding to the end of the machining path
L	: Number of calls, it is 1 by default when omitted.

Description

When this command is executed, the tool is machined from the start position along the workpiece machining path given in the program segment specified by P and Q.

It is generally used for finishing after rough machining by G71, G72 and G73.



Example

Case 1: for finishing after G71

```
O1000
T0101
G71 ...
G71 P10 Q20... // Rough turning cycle
N10
...          Program path
...
N20
T0202
G70 P10 Q20 // Finishing cycle
G00 X100 Z100
M30
```

Case 2: Used alone

```
O1000
T0101
...
G70 P10 Q20 ...// Call Program path 1
G70 P100 Q200 L2...// Call Program path 2, 2 times
M30

N10
...          Program path 1
...
N20

N100
...          Program path 2
...
N200
```

2.4.4 Stock Removal in Turning Cycle (G71)

Format

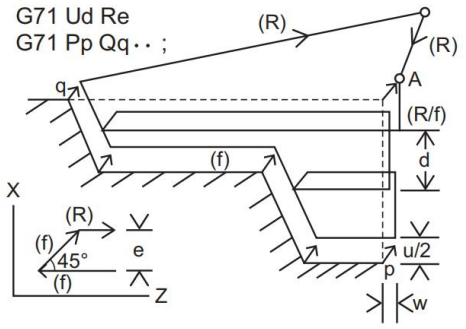
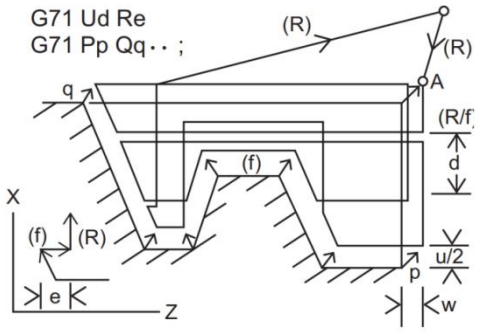
① **G71 U(d) R(e);** (J0/J1)
② **G71 P(p) Q(q) U(u) W(w) F S T ;**

U(d)	: Cut-in amount (radius value). Unit: mm, unsigned
R(e)	: Tool retracting amount (radius value). Unit: mm, unsigned

P(p)	: Program segment number corresponding to the beginning of the machining path
Q(q)	: Program segment number corresponding to the end of the machining path
U(u)	: The finishing allowance in the X-axis direction (the allowance is 0 when omitted). Unit: mm
W(w)	: The finishing allowance in the Z-axis direction (the allowance is 0 when omitted). Unit: mm
F	: Cutting feed speed
S	: Spindle speed
T	: Tool number, offset number

Description

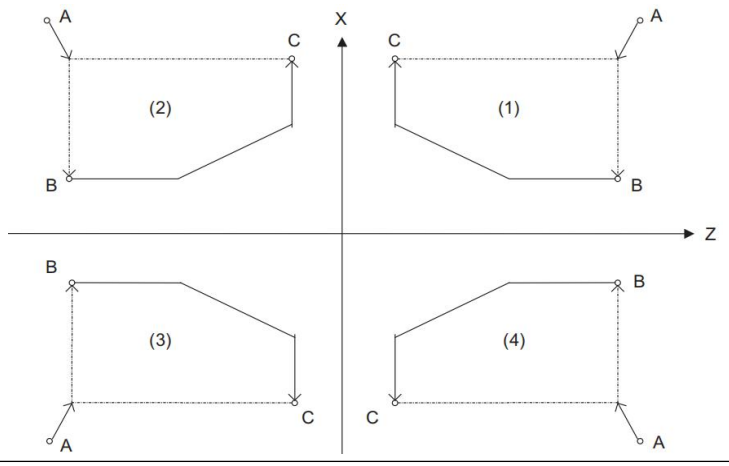
The roughing path is automatically calculated based on the machining path PQ, the cut-in amount U(d), and the tool retracting amount R(e) specified by the G71 command, as follows:

Example of machining path without cavities (G71 type I)	Example of machining path with cavities (G71 type II)
	
(R) Rapid feed (f) Cutting feed A: Start point of roughing	

1. Paragraph ① of G71 can specify J1 (cutting back along the contour) and J0 (cutting back at a 45 degree angle), which can be omitted and specified by the third bit of sys-parameter-5120.

2. Paragraph ① of G71 may be omitted, and when omitted, the cut-in amount U(d) is specified by sys-parameter-5130 and the tool retracting amount R(e) is specified by sys-parameter-5131.

3. Depending on the cut-in direction, the command path has the following four cases (shown in the figure below), and the symbols for the finishing allowance U(u) and W(w) are as follows:



A: Start point of G71 cycle

B: Start program segment of the machining path specified by P

C: End program segment of the machining path specified by Q

Execution track: A ➡ B ➡ C

4 cases:

(1)	U(u)+	W(w)+
(2)	U(u)+	W(w)-
(3)	U(u)-	W(w)-
(4)	U(u)-	W(w)+

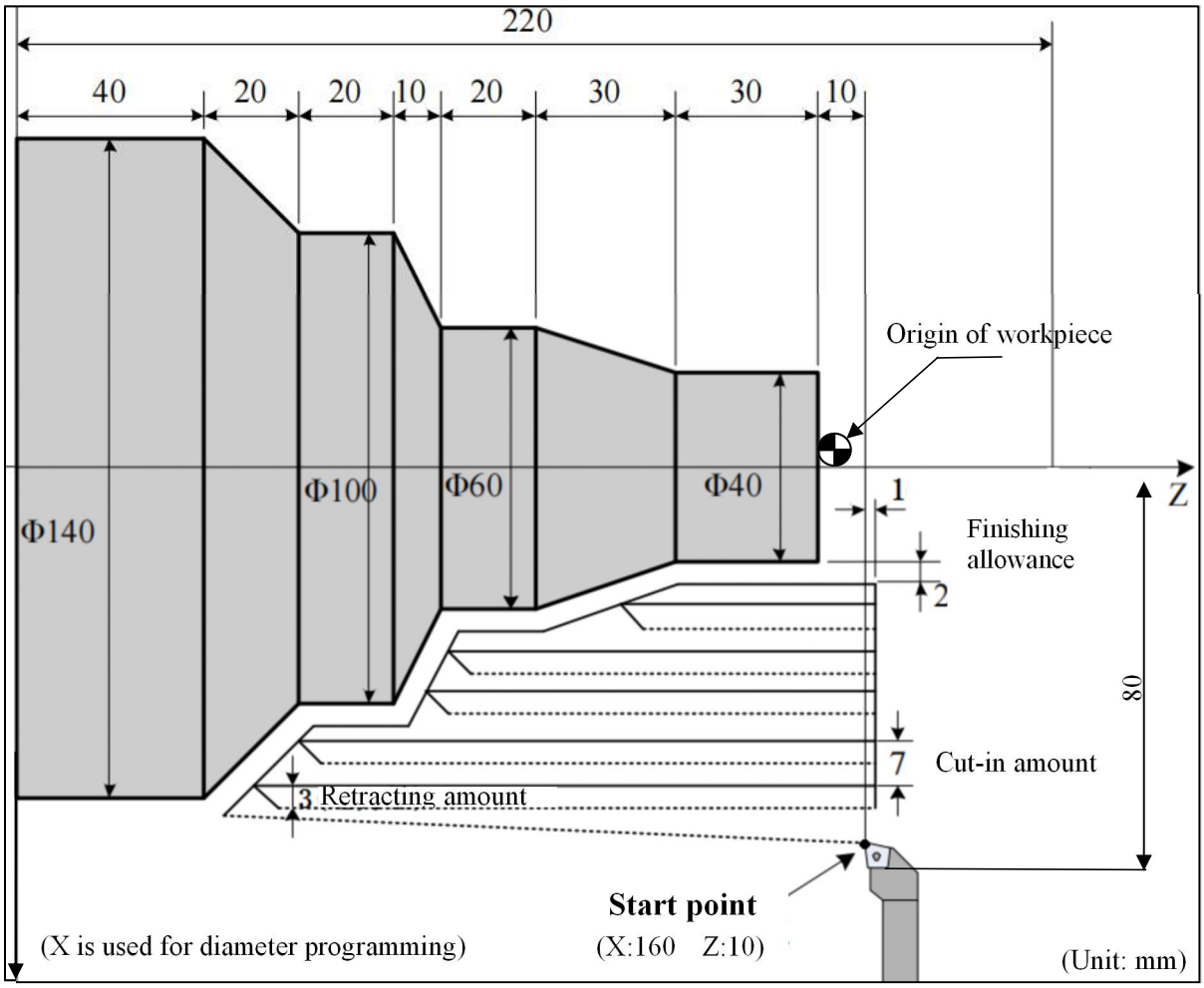
4. In G71 cycle, F, S, T functions of program segments between P and Q are all invalid and should be omitted. F, S, T functions of or before G71 program segment are all valid. F, S, T functions of program segments between P and Q are only valid for G70 code loops.

5. In G71 cycle, when constant linear velocity control function is available, G96 or G97 of program segments between P and Q is invalid.

6. Program segments between P and Q cannot call subprograms;

7. Program segments between P and Q can be used for tool nose compensations G40/G41/G42;

Example



O1000	T0202// Finishing tool
T0101// Roughing tool	G70 P10 Q20// Finishing call
G98 M03 S500	M30
G00 X160 Z10 // Positioned to the start point of the G71 cycle	
G71 U7 R3 // Set the cut-in amount and tool retracting amount	
G71 P10 Q20 U4 W1 F300// Set the machining path / finishing margin / rough turning feed speed	
N10 G00 X40 S1000	
G01 W-40 F100	
G01 X60 W-30	
G01 W-20	Machining path, during which F/S/T commands are not valid for the G71 cycle
G01 X100 W-10	
G01 W-20	
N20 G01 X140 W-20	

2.4.5 Stock Removal Facing Cycle (G72)

Format

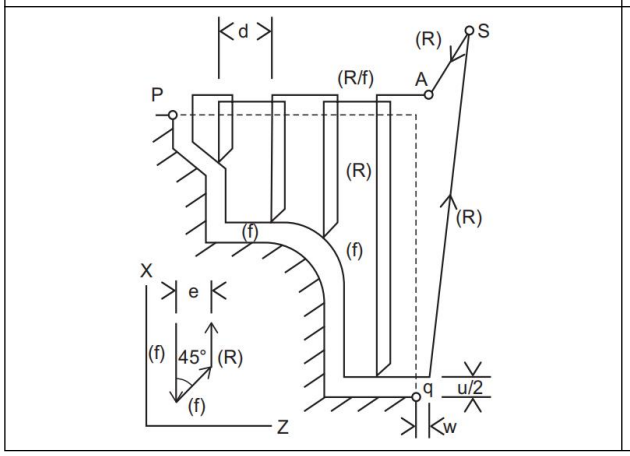
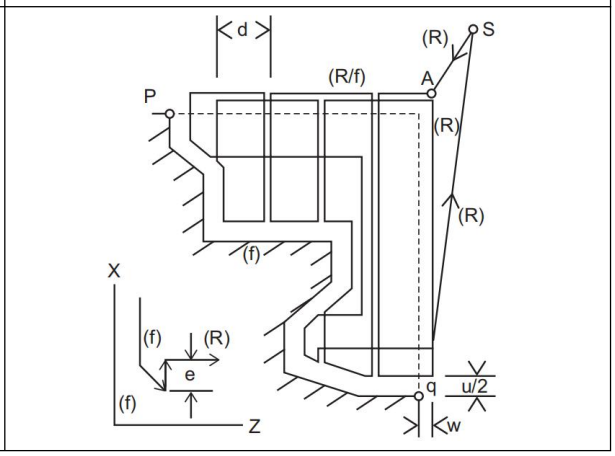
① **G72 W(d) R(e);** (J0/J1)
② **G72 P(p) Q(q) U(u) W(w) F_ S_ T_;**

W(d)	: Cut-in amount (modal) (radius value). Unit: mm, unsigned
R(e)	: Tool retracting amount (modal) (radius value). Unit: mm, unsigned

P(p)	: Program segment number corresponding to the beginning of the machining path
Q(q)	: Program segment number corresponding to the end of the machining path
U(u)	: The finishing allowance in the X-axis direction (the allowance is 0 when omitted). Unit: mm
W(w)	: The finishing allowance in the Z-axis direction (the allowance is 0 when omitted). Unit: mm
F	: Cutting feed speed
S	: Spindle speed
T	: Tool number, offset number

Description

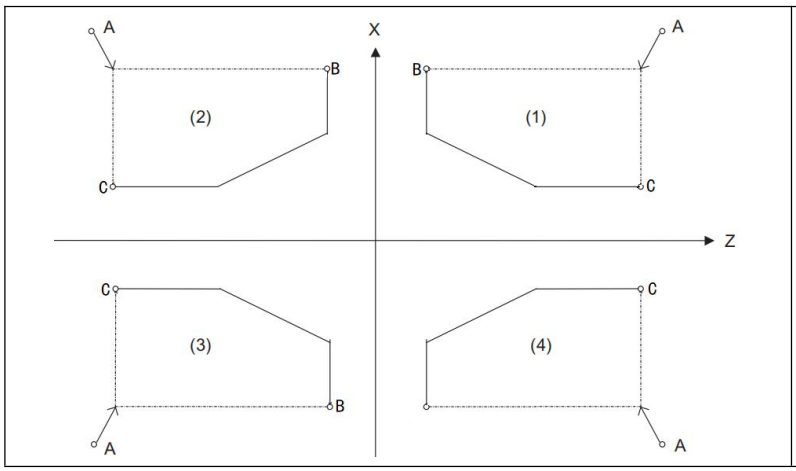
The roughing path is automatically calculated based on the machining path PQ, the cut-in amount U(d), and the tool retracting amount R(e) specified by the G72 command, as follows:

Example of machining path without cavities (G72 type I)	Example of machining path with cavities (G72 type II)
	

S: Start point of cycle (S) Rapid feed (f) Cutting feed A: Start point of roughing

1. Paragraph ① of G72 may be omitted, and when omitted, the cut-in amount W(d) is specified by sys-parameter-5130 and the tool retracting amount R(e) is specified by sys-parameter-5131;

2. Depending on the cut-in direction, the command path has the following four cases (shown in the figure below), and the symbols for the finishing allowance U(u) and W(w) are as follows:



A: Start point of G72 cycle

B: Start program segment of the machining path specified by P

C: End program segment of the machining path specified by Q

Execution track: A ➡ B ➡ C

4 cases:

(1)	U(u)+	W(w)+
(2)	U(u)+	W(w)-
(3)	U(u)-	W(w)-
(4)	U(u)-	W(w)+

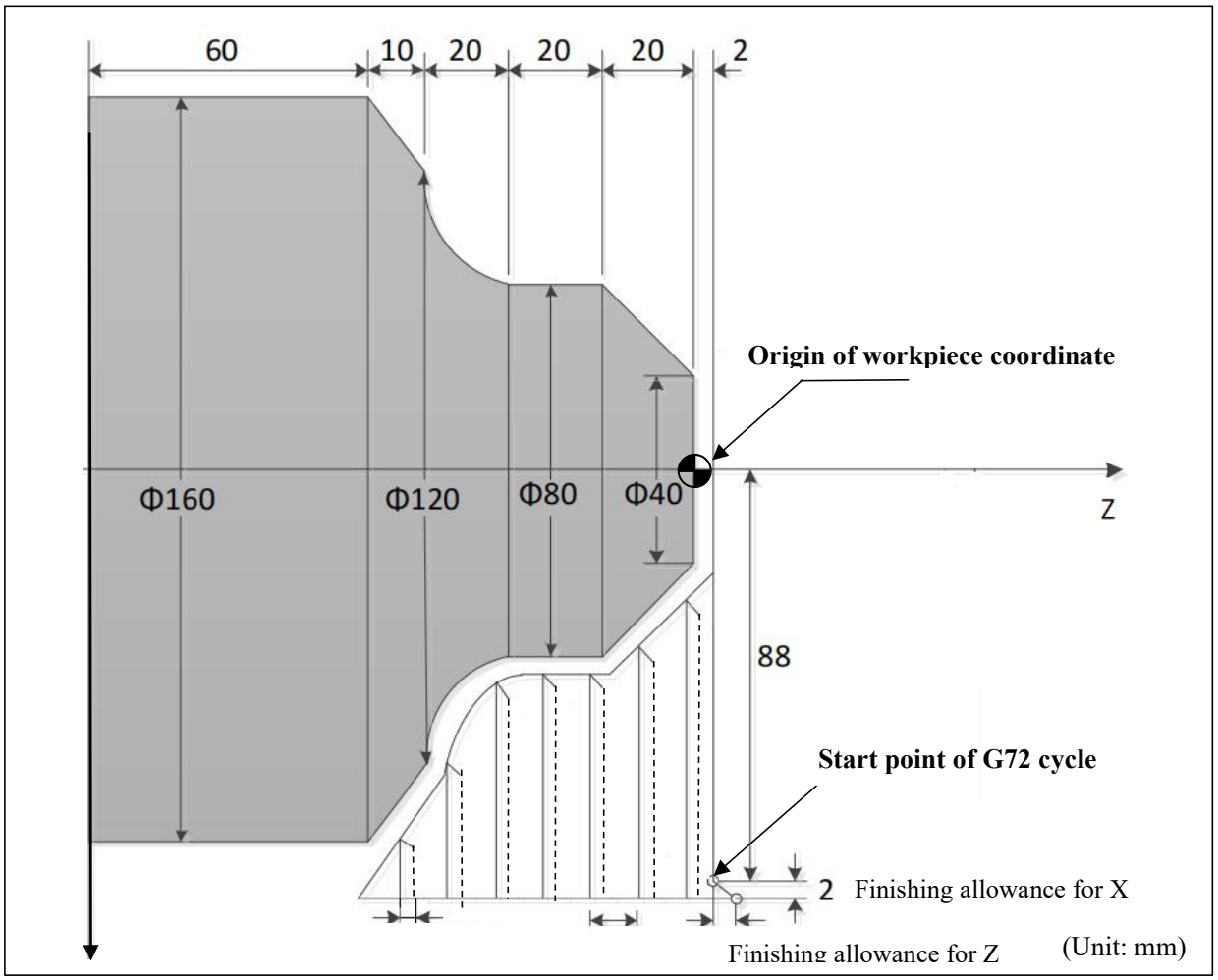
3. In G72 cycle, F, S, T functions of program segments between P and Q are all invalid and should be omitted. F, S, T functions specified by G71 program segment or previous commands are valid. F, S, T functions of program segments between P and Q are only valid for G70 code loops.

4. In G72 cycle, when constant linear velocity control function is available, G96 or G97 of program segments between P and Q is invalid.

5. Program segments between P and Q cannot call subprograms;

6. Program segments between P and Q can be used for tool nose compensations G40/G41/G42;

Example



O1000	T0202// Finishing tool
T0101// Roughing tool	G70 P10 Q20//Finishing call
G98 M03 S500	M30
G00 X176 Z2 // Positioned to the start point of the G72 cycle	
G72 W7 R2 // Set the cut-in amount and tool retracting amount	
G72 P10 Q20 U4 W2 F300// Set the machining path / finishing margin / rough turning feed speed	
N10 G00 Z-70 S1000	
G01 X160 F150	
G01 X120 W10	
G03 X80 W20 R20	Machining path, during which F/S/T commands are not valid for the G72 cycle
G01 W20	
N20 G01 X40 W20	

2.4.6 Pattern Repeating (G73)

Through this cycle command, repeated cutting can be carried out according to the path specified by NS-NF program segment, and the tool moves forward one time for each cutting. For the preliminarily formed blanks through rough machining, such as forging and casting, etc., they can be processed efficiently.

Format

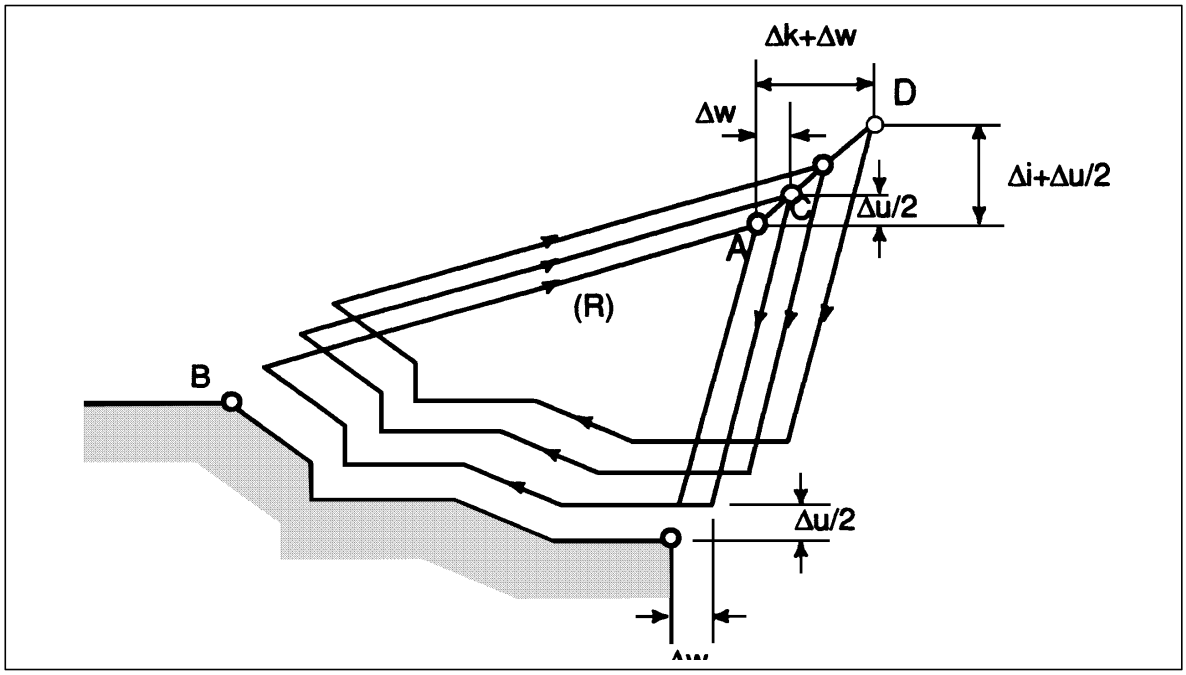
① G73 U(△i) W(△k) R(d);
② G73 P(NS) Q(NF) U(△u) W(△w) F_ S_ T_;

U(△i)
: return distance (modal) in the X direction (radius value). Unit: mm, the symbol indicates the direction

R(d)	: Number of close cutting (modal), unit: times
P(NS)	: Program segment number corresponding to the beginning of the machining path
Q(NF)	: Program segment number corresponding to the end of the machining path
U(△u)	: Finishing allowance in X-axis direction, unit: mm, the symbol indicates the direction
W(△w)	: Finishing allowance in Z-axis direction, unit: mm, the symbol indicates the direction
F	: cutting feed speed
S	: spindle speed
T	: tool number, offset number

Description

The machining path program is called to automatically calculate the intermediate path while rough machining is performed according to the machining path.



1. F, S, T functions on any one program segment from NS to NF are all invalid.

2. F, S, T functions are valid;

3. Δi, Δk, Δu and Δw are specified by addresses U and W, and the difference of them is determined depending on whether P and Q are specified by them.

4. The program segment of G73 from NS to NF cannot call the subprogram.

5. It realizes circular processing according to the program segment from NS to NF. Please pay attention to the symbol of Δi, Δk, Δu and Δw during programming. At the end of the cycle, the tool returns to point D as shown in the figure above;

2.4.7 End Face Peck Drilling Cycle (G74)

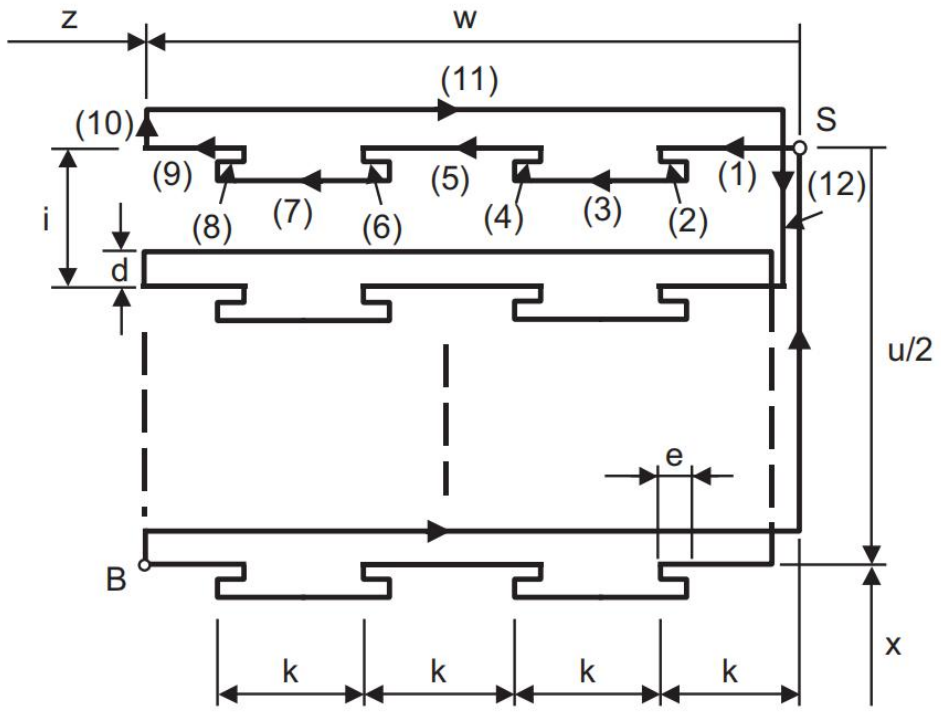
Format

① **G74 R(e);**
② **G74 X/U_ Z/W_ P(i) Q(k) R(d) F_;**

R(e)
: the retract amount after carrying out cutting for Q(k) along the direction of axis Z each time (unsigned), unit:

X/U : X-coordinate of point B (absolute/relative coordinates)
Z/W : Z-coordinate of point B (absolute/relative coordinates)
P(i) : The circulating moving amount of each time at the direction of axis X (unsigned) (radius value),
Q(k) : movement amount of each cutting at the direction of axis Z (unsigned) (radius value)
R(d) : movement amount of tool retracting at the X direction when cutting to the end point (unsigned) (radius value)
F : Cutting feed speed

Format



S:start point B:end point

The above path is illustrated. Paths (1)-(12) are subject to an action cycle, where paths (1), (5), (9) and (10) are subject to cutting feed, and paths (2), (4), (6), (8), (11) and (12) are subject to rapid movement.

1. When X/U_ and P(i) are omitted, only the Z-axis performs the action;
2. When Q(k) is omitted, it performs direct cutting feed to the end point coordinates of Z axis after G74 is executed;
3. When the program segment ① of G74 is omitted, the return amount is specified by sys-parameter-5136;
4. When X/U_ is specified, P(i) must be specified, otherwise, a system alarm will be generated;

2.4.8 Outer Diameter / Internal Diameter Drilling Cycle (G75)

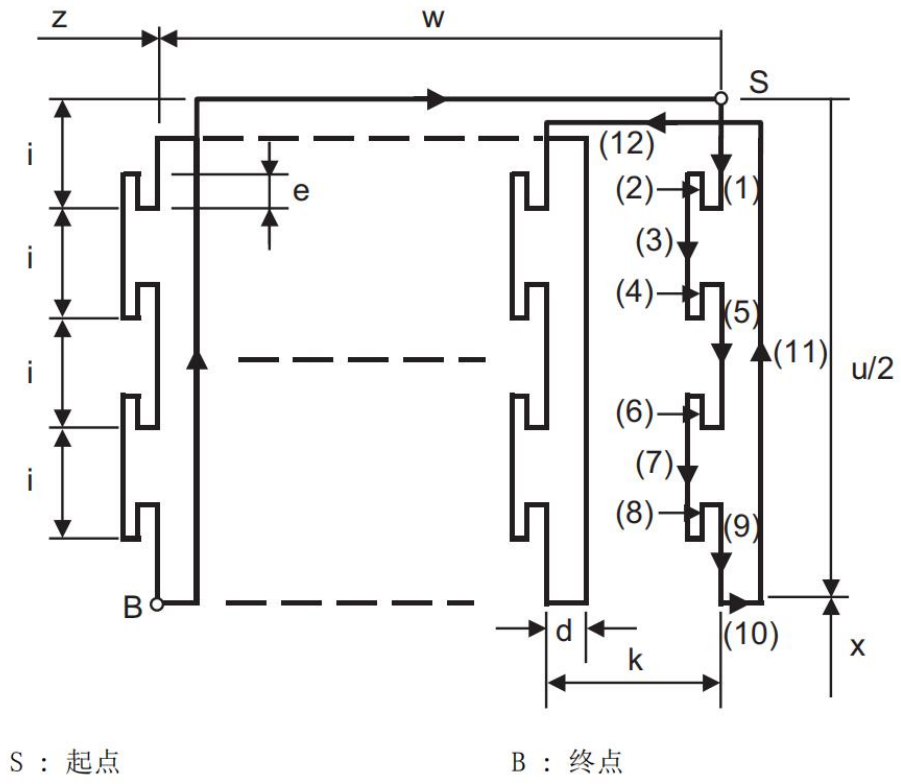
Format

① **G75 R(e);**
② **G75 X/U_ Z/W_ P(i) Q(k) R(d) F_;**

R(e) : the tool retracting amount after carrying out cutting for P(i) along the direction of axis X each time

X/U : X-coordinate of point B (absolute/relative coordinates)
Z/W : Z-coordinate of point B (absolute/relative coordinates)
P(i) : The cutting moving amount of each time at the direction of axis X (unsigned) (radius value),
Q(k) : movement amount of each cycle at the direction of axis Z (unsigned) (radius value), unit: mm
R(d) : The amount of tool retracting in the Z direction when cutting to the end point (unsigned) (radius value)
F : Cutting feed speed

Format



The above path is illustrated. Paths (1)-(12) are subject to an action cycle, where paths (1), (5), (9) and (10) are subject to cutting feed, and paths (2), (4), (6), (8), (11) and (12) are subject to rapid movement.

1. When Z/W_ and Q(k) are omitted, only the X-axis performs the action;
2. When P(i) is omitted, it performs direct cutting feed to the end point coordinates of X axis after G75 is executed;
3. When the program segment ① of G75 is omitted, the return amount is specified by sys-parameter-5136;

When Z/W_ is specified, Q(k) must be specified, otherwise, a system alarm will be generated;

2.4.9 Multiple Thread Cycle (G76)

Format

① G76 P(m)(r)(a) Q(△dmin) R(d);
② G76 X/U_ Z/W_ R(i) P(k) Q(△d) F/I_;

m	: Number of thread finishing cycles, value range: 00 to 99
r	: Chamfering amount, setting range: 00 to 99, unit: 0.1 × lead
a	: Tool nose angle, setting range: 00 to 99, unit: degree
Q(△dmin)	: Minimum cutting volume, when the automatically calculated cutting volume is less than this value, the value set by △dmin is used.
R(d)	: finishing allowance, unit: mm

X/U	: absolute (relative) coordinate of thread end point at X direction; unit: mm
Z/W	: absolute (relative) coordinate of thread end point at Z direction; unit: mm
R(i)	: radius difference value of the thread head relative to the thread tail; i＝0 represents that it is a straight thread; unit: mm
P(k)	: thread height (unsigned) (radius value), unit: mm Common empirical formula: thread height P ≈ 0.54 or 0.65 x lead
Q(△d)	: cut-in amount of the first cutting (unsigned) (radius value), unit: mm
F	: lead of metric thread, unit: mm
I	: lead of inch thread, number of teeth per inch

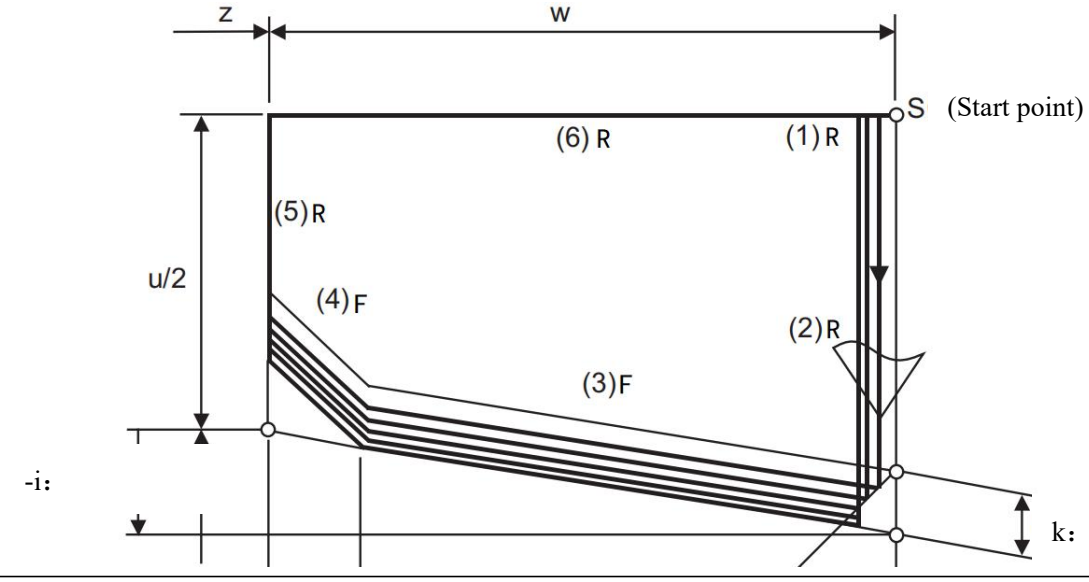
Command extension characters

In section ① of G76, additionally specify:	
J	: Type of feed, 0: standard 1: right center staggered 3: left and right staggered
PQ	: Thread chip-breaking interval, 1 thread chip-breaking action is performed per PQ cutting
In section ② of G76, additionally specify:	
K	: Lead variation per turn, unit: mm
L	: Multiple head thread count, if not specified, it is a single head thread

Description

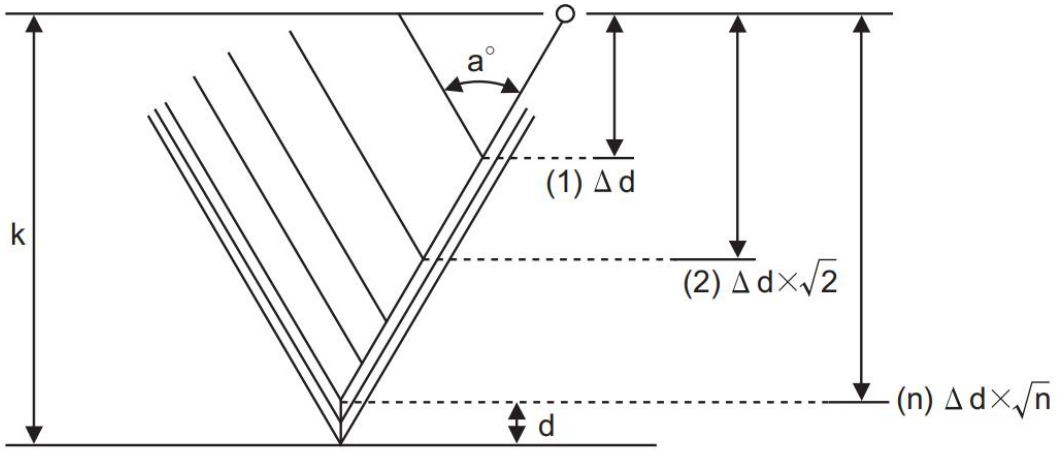
Composition of the cycle:

Paths (1), (2), (5) and (6) in 1 cycle are subject to rapid movement, and paths (3) and (4) are subject to the movement of cutting feed as specified by F.

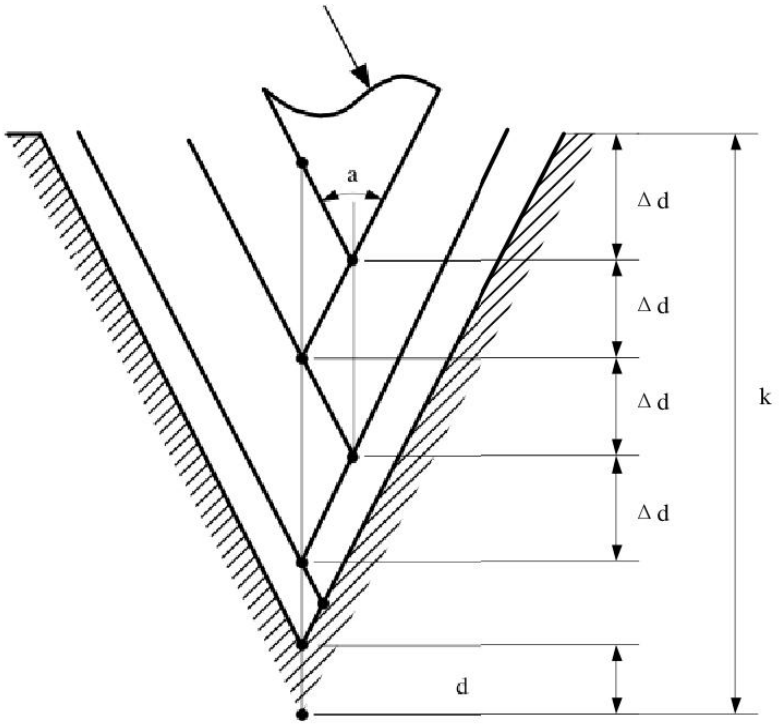


Cut-in path of G76 thread type:

Single-sided cutting with constant feed amount (when sys-parameter-5035 = 0)



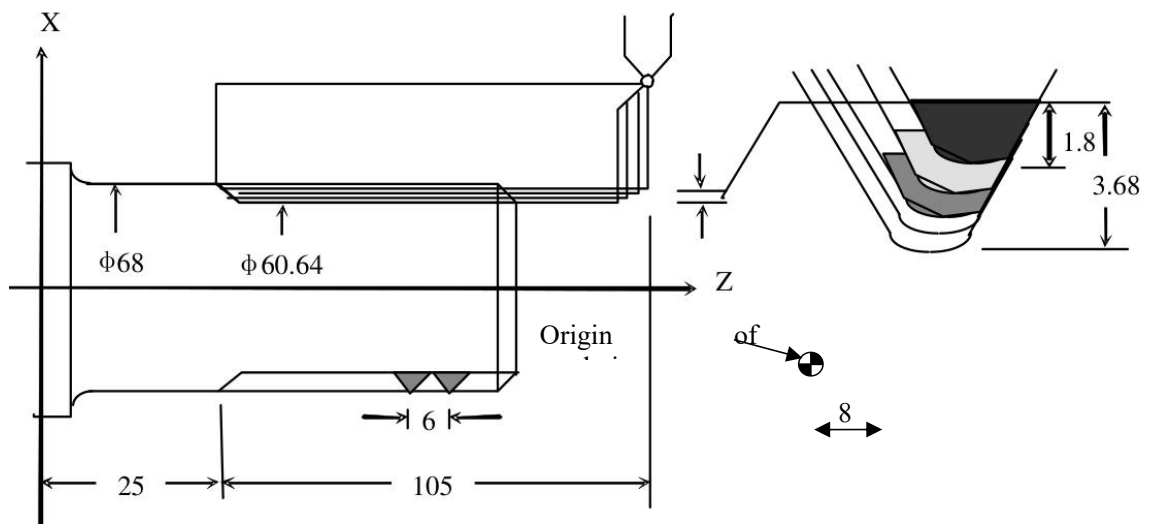
Staggered cutting with constant feed amount (when sys-parameter-5035 = 0)



Detailed description

1. The data specified by P, Q, R is distinguished according to X/U, Z/W;
2. In G76 mode, the long axis of pitch is always Z-axis, so end face (X-direction) threads cannot be machined;
3. The values of m, r, a are specified by the address P at one time; without programming the address P, they are set by sys-parameters 5032, 5033, and 5034;
4. Q(△dmin) and R(d) are specified by sys-parameters #5030 and #5031 if they are not specified or are 0;
5. The notes on screw cutting are the same as those on G32 screw cutting.
6. the CNC parameter-5035 can define the cutting mode of G76, including unilateral cutting and staggered cutting.
7. sys-parameter-5036 can define whether the start point of G76 coincides with the start point of the G32/G92 thread command;

Example




```
O1000
T0101 // Threaded tool
G00 X68 Z50 // Safety point positioning
G76 P010060 R0.2
//01 specifies the number of finishing turns
//00 indicates the default thread chamfering
//60 specifies angle of tool nose
//R0.2 specifies the finishing allowance
G76 X60.4 Z25 P3.68 Q1.8 F6
// P3.68 is the tooth height, Q1.8 is the depth of the first cut, and F6 is the pitch.
M30
```

2.4.10 Drilling Cycle (G83/G87)

Format

G83 Z/W_ α/β_ R_ Q_ F_;(Drilling cycle in the 1st axis direction of the plane)

Z/W	: indicates drilling depth (absolute programming / incremental programming)
α/β	: Positioning coordinates of additional axes other than Z axis (α absolute programming, β incremental programming), multiple axes can be specified.
R	: R point, fast positioning point before Z-axis drilling, specified by absolute coordinates
Q	: Single drilling depth (unsigned), unit: mm
F	: Drilling feed rate in mm/min (in G98 mode), mm/r (in G99 mode)

G87 X/U_ α/β_ R_ Q_ F_; (Drilling cycle in the 2nd axis direction of the plane)

X/U	: indicates drilling depth (absolute programming / incremental programming)
α/β	: Positioning coordinates of additional axes other than X axis (α absolute programming, β incremental programming), multiple axes can be specified.
R	: R point, fast positioning point before X-axis drilling, specified by absolute coordinates
Q	: Single drilling depth (unsigned), unit: mm
F	: Drilling feed rate in mm/min (in G98 mode), mm/r (in G99 mode)

Command extension characters

G83.5/G87.5	: High-speed drilling cycle mode, replace G83/G87
G83.6/G87.6	: Deep hole drilling cycle mode, replace G83/G87
G83.7/G87.7	: High-speed deep hole drilling cycle mode, replace G83/G87
K	: Number of drilling repetitions; it is 1 when omitted.
Mxx	: Specified by M code; Mxx is executed before each hole drilling, and Mxx+1 is executed after hole drilling is completed.
P	: dwell time at hole bottom, unit: ms

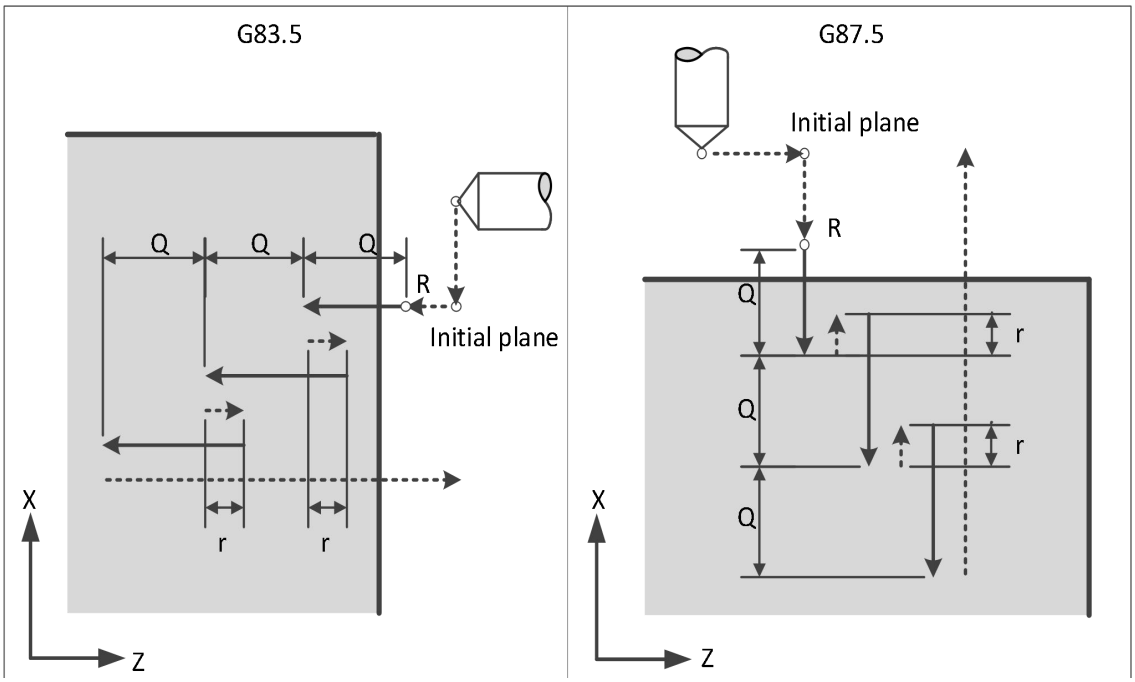
Description

Cycle type 1: High-speed drilling cycle

(set to 1 by bit 1 of sys-parameter-5100, or replacing G83/G87 with G83.5/G87.5)

Command action:

r: return amount per feed, defined by sys-parameter-5110

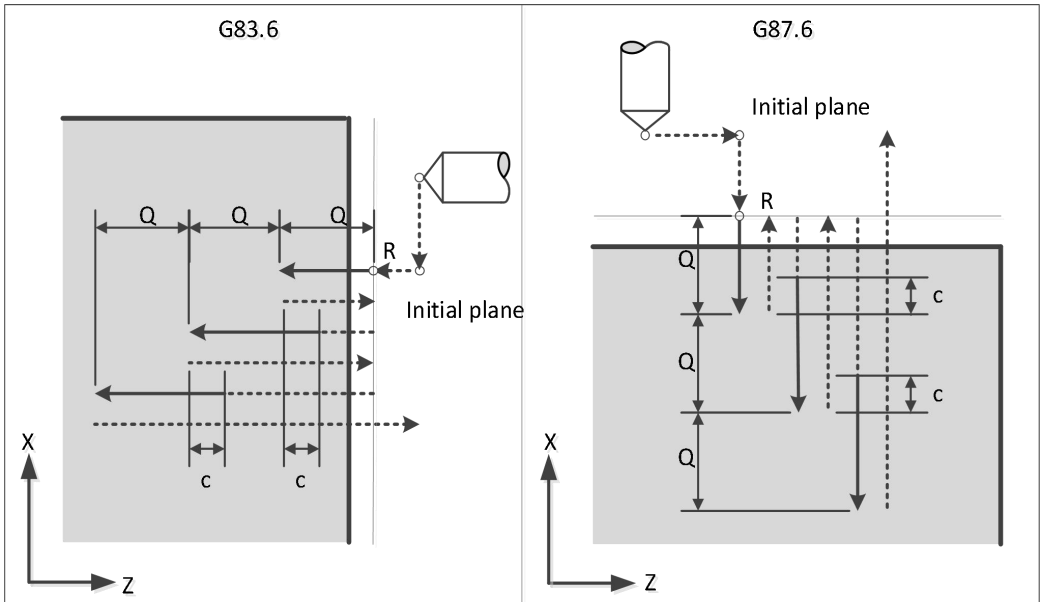


Cycle type 2: Deep hole drilling cycle

(set to 0 by bit 1 of sys-parameter 5100, or replacing G83/G87 with G83.6/G87.6)

Command action:

c: amount of tool feed clearance, defined by sys-parameter-5111



Cycle type 3: High-speed deep hole drilling cycle

The combination of type 1 and type 2, using both r (return amount) and c (clearance amount), is programmed with G83.7/G87.7 instead of G83/G87.

Example

Application in indexing drilling on workpiece end plane Assume: X and Z are linear axes, A axis is the spindle indexing axis, and 4 holes with 90 degree interval are drilled on the workpiece at the position where the diameter is 100 mm.	
O1000	
T0101 G98	
G00 Z5	
G00 X100	// Locate to the position where the workpiece diameter is 100 mm
M103 S2=1000	// Turn on the power head
M19	// Spindle orientation
M18	// Spindle position mode switch
G83.5 Z-10 A0 R1 Q4 F300 1 2 3 4 5	//1: G83.5 sets the high speed drilling cycle //2: Drilling depth: 10mm, starting from the 0 degree position of A-axis //3: Fast positioning point before drilling //4: carry out cutter lifting once for every 4mm of drilling, specified by the cutter lifting height sys-parameter-5110 //5: specify the drilling speed, feed per minute in G98 mode
UA90 K3	//Incrementally specify drilling at axis A every 90 degrees, perform 3 cycles
G00 Z100	//Cancel drilling cycle, and perform tool retracting
M17	//Switch back to speed mode of spindle
M30	

2.4.11 Rigid Tapping Cycle (G84/G88)

Format

M29/M28 P_ S_; (Set tapping spindle and tapping speed)

M29/M28	: Set tapping mode, M29 is the rigid tapping mode, M28 is the encoder tapping mode
P	: Set tapping spindle number
S	: Set the tapping spindle speed; when M29 is executed, a positive value of S indicates right-hand tapping, and a negative value of S indicates left-hand tapping. Example: M29 P2 S500 means setting the speed of the second spindle to 500 during tapping

G84 Z/W_ α/β_ R_ F/I_; (Tapping cycle in the 1nd axis direction of the plane)

Z/W	: indicates tapping depth (absolute programming / incremental programming)
α/β	: Positioning coordinates of additional axes other than Z axis (α absolute programming, β incremental programming); multiple axes can be specified.
R	: R point, fast positioning point before Z-axis tapping, specified by absolute coordinates
F	: lead of metric thread, unit: mm
I	: lead of inch thread, number of teeth per inch

G88 X/U_ α/β_ R_ F/I_; (Tapping cycle in the 2nd axis direction of the plane)

X/U	: indicates tapping depth (absolute programming / incremental programming)
α/β	: Positioning coordinates of additional axes other than X axis (α absolute programming, β incremental programming); multiple axes can be specified.
R	: R point, fast positioning point before X-axis tapping, specified by absolute coordinates
F	: lead of metric thread, unit: mm
I	: lead of inch thread, number of teeth per inch

Command extension characters

G84.5/G88.5	: High-speed deep hole tapping cycle mode
G84.6/G88.6	: Deep hole tapping cycle mode
K	: Number of tapping repetitions, it is 1 when omitted.
Mxx	: Specified by M code; Mxx is executed before each tapping and Mxx+1 is executed after the tapping is completed.
P	: dwell time at hole bottom, unit: ms
J	: Rotational speed during tapping tool retracting, unit: r/min

the CNC supports using M28/M29 Sn=xxxx instead of M29 P_ S_ to set the tapping spindle and tapping speed as shown in the table below:

1. Specified by M29

Example of use	Standard command	Extended command
Set the 1st spindle for tapping, with a spindle speed of 500	M29 P1 S500	M29 S1=500 or M29 S500
Set the 2nd spindle for tapping, with a spindle speed of 500	M29 P2 S500	M29 S2=500

2. Specified by M28

Example of use	Standard command	Extended command
Set the 1st spindle for tapping, with a spindle speed of 500	M28 P1 S500	M28 S1=500
		or M28 S500
		or SP1 S500
Set the 2nd spindle for tapping,	M28 P2 S500	M28 S2=500

with a spindle speed of 500		or SP2 S2=500

Description

1. The tapping mode is divided into synchronous mode and follow-up mode.

Synchronous mode	**Follow-up mode**	
Rigid tapping (Bit 1 of sys-parameter-5201 is set to 0)	Encoder tapping	Rigid tapping (Bit 1 of sys-parameter-5201 is set to 1)
1. The spindle is controlled by position 2. The tapping linear axis and spindle position control axis are used for interpolation tapping. 3. Specify tapping speed via M29 P_ S_ 4. No need to start the spindle via M03/M04	1. The spindle is controlled by analog 2. Spindle encoder feedback required 3. The spindle speed must be specified by M28 P_ S_. In addition, it also needs to use M03/M04 to start the spindle	1. The spindle is controlled by position 2. Spindle encoder feedback required 3. Specify tapping speed via M29 P_ S_ 4. No need to start the spindle via M03/M04

2. The lead value F during tapping is limited by sys-parameter-5210. If the lead is specified by I, the CNC will automatically convert it to F before making a determination.

3. P1 can be omitted from the M28/M29 command when tapping is performed with the 1st spindle;

4. M28/M29 specifies the modal state;

5. After power-on of the CNC, the default mode is M28 tapping mode.

6. The follow-up tapping mode is always used for M28, and it is independent of the set value of bit 1 of sys-parameter-5201.

7. The SPn (specifying current spindle) command can be used instead of M28 to set the current tapping spindle;

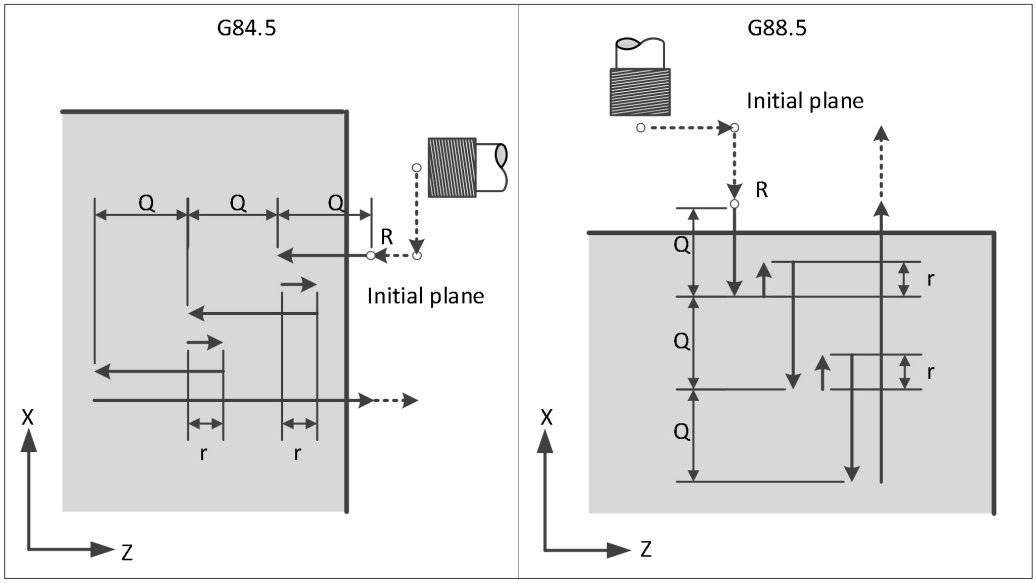
8. Support G17/G18/G19 to switch planes, when G84 corresponds to the 1st axis tapping command under the coordinate system plane, and G88 corresponds to the 2nd axis under the plane for tapping.

Cycle type 1: High-speed deep hole tapping cycle(The spindle must work in position mode**)**

(set to 1 by bit 1 of sys-parameter-5200, or replacing G84/G88 with G84.5/G88.5)

Command action:

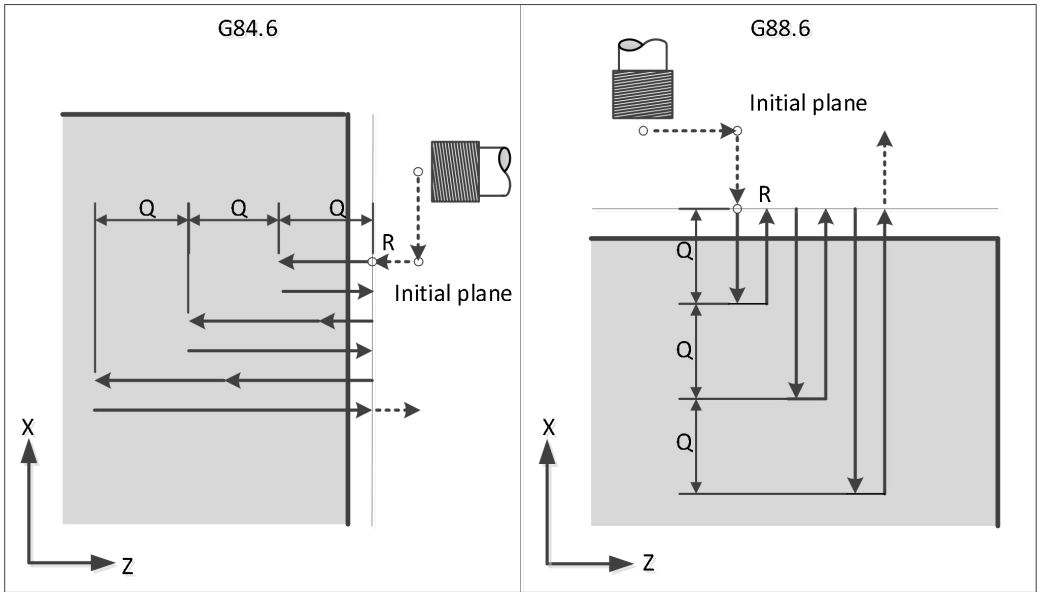
r: return amount per feed, defined by sys-parameter-5211



Cycle type 2: Deep hole tapping cycle

(set to 0 by bit 1 of sys-parameter 5200, or replacing G84/G88 with G84.6/G88.6)

Command action:



Example

1. Encoder follow-up tapping, the 1st spindle and 2nd spindle are controlled by the inverter Assume: X and Z are linear axes, the 1st spindle is responsible for X direction tapping, and the 2nd spindle is responsible for Z direction tapping.			
Since the spindle is controlled by analog, the tapping spindle is specified with M28			
O1000			
T0101			
G00 X5			
M28 P1 S500	or	SP1(default, can be omitted) M03 S500	//Set the 1st spindle as the tapping spindle and set the tapping speed S
M03			//Start the 1st spindle
G84 Z-10 R1 F1.25 1 2 3			//1: Tapping feed to the position of Z-10 mm //2: Fast positioning to absolute coordinates Z1 before tapping //3: Lead
G00 Z5			
M28 P2 S500	or	SP2(cannot be omitted) M103 S2=500	//Set the 2nd spindle as the tapping spindle and set the tapping speed S
M103			//Start the 2nd spindle
G88 X-10 R1 F1.25 1 2 3			//1: Tapping feed to the position of X-10 mm //2: Fast positioning to absolute coordinates X1 before tapping //3: Lead
M30			

2. Application in indexing tapping on workpiece end plane

Assume: X and Z are linear axes, A axis is the spindle indexing axis, Y axis is the tapping power head (2nd spindle), tapping 4 holes at 90 degree interval at the position where the diameter is 100 mm.

Bit 1 of sys-parameter-5201 is set to 0, and the synchronous mode is used when G84 command is executed.

O1000			
T0101 G98			
G00 Z5			
G00 X100			// Locate to the position where the workpiece diameter is 100 mm
M105			//Make sure the power head stops
M19			//Spindle (1st spindle) orientation
M18			// Spindle position mode switch
M29 P2 S400	or	M29 S2=400	//Setting the tapping speed of the tapping power head (2nd spindle)
G84.5 W-10 A0 R1 Q4 F1.25 1 2 3 4 5			//1: G84.5 is used to set the high-speed tapping cycle /2: The tapping depth is 10mm, and the tapping starts from the 0 degree position of A axis. //3: Fast positioning point before tapping //4: carry out cutter lifting once for every 4mm of tapping, specified by the CNC parameter-5211 for cutter lifting height //5: Lead
UA90 K3	or	A90	//One tapping action is performed after A-axis is positioned to 90 degrees.
		A180	//One tapping action is performed after A-axis is positioned to 180 degrees.
		A270	//One tapping action is performed after A-axis is positioned to 270 degrees.
G00 Z100			//Cancel tapping cycle, and perform tool retracting
M17			//Switch back to speed mode of spindle
M30			

2.4.12 Boring Cycle (G85/G89)

Format

G85 Z/W_ α/β_ R_ F_ P_; (Boring cycle in the 1st axis direction of the plane)

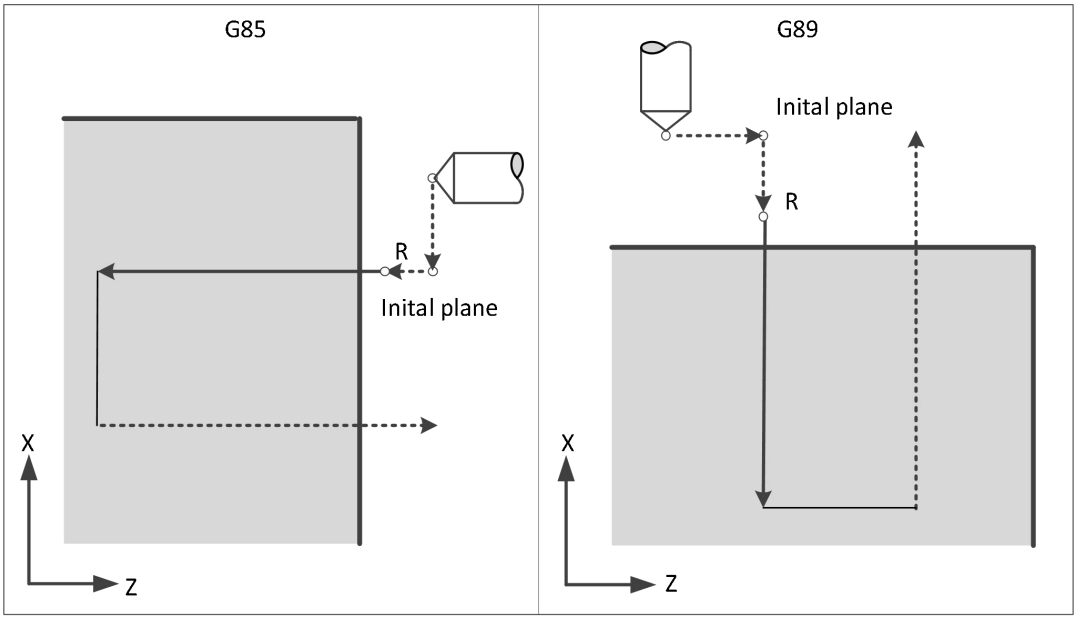
Z/W	: indicates boring depth (absolute programming / incremental programming)
α/β	: Positioning coordinates of additional axes other than Z axis (α absolute programming, β incremental programming), multiple axes can be specified.
R	: fast positioning point, specified by absolute coordinates
F	: Feed rate, unit: mm/min (in G98 mode), mm/r (in G99 mode)
P	: Dwell time at hole bottom, unit: ms

G89 X/U_ α/β_ R_ F_ P_; Boring cycle in the 2nd axis direction of the plane)

X/U)	: indicates boring depth (absolute programming / incremental programming)
α/β	: Positioning coordinates of additional axes other than X axis (α absolute programming, β incremental programming); multiple axes can be specified.
R	: fast positioning point, specified by absolute coordinates
F	: Feed rate, unit: mm/min (in G98 mode), mm/r (in G99 mode)
P	: dwell time at hole bottom, unit: ms

Description

Command action:



2.4.13 Canned Cycle (G90/G92/G94)

There are three canned cycles:the outer diameter/internal diameter cutting canned cycle (G90),the threading canned cycle(G92), and the end face turning canned cycle(G94).

NOTE

1 Explanatory figures in this section use the ZX plane as the selected plane, diameter programming for the X-axis, and radius programming for the Z-axis. When radius programming is used for the X-axis, change U/2 to U and X/2 to X.

2 A canned cycle can be performed on any plane (including parallel axes for plane definition). When G-code system A is used, however, U, V, and W cannot be set as a parallel axis.

3 The direction of the length means the direction of the first axis on the plane as follows:
ZX plane: Z-axis direction
YZ plane: Y-axis direction
XY plane: X-axis direction

4 The direction of the end face means the direction of the second axis on the plane as follows:
ZX plane: X-axis direction
YZ plane: Z-axis direction
XY plane: Y-axis direction

2.4.13.1 Outer Diameter/Internal Diameter Cutting Cycle (G90)

G90 can realize cylindrical surface and circular conical surface single cycle processing. After the cycle is finished, the tool returns to the cutting start point.

Format

G90 X/U_ Z/W_ R_ F_;

X/U : X-axis coordinate data

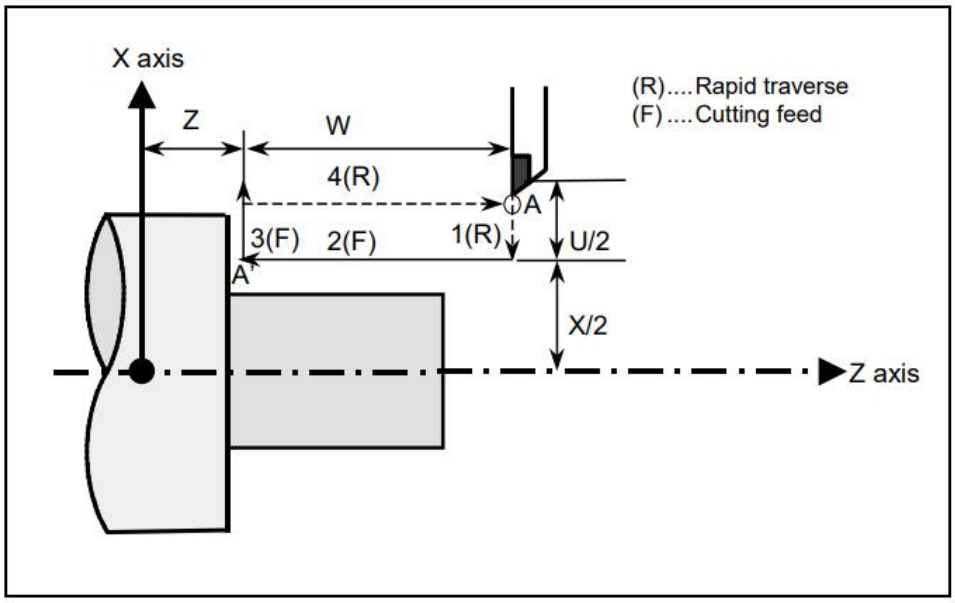
Z/W : Z-axis coordinate data

R : Radius difference between the conical surface start point and the conical surface end point;

F : Feed rate, unit: mm/min (in G98 mode), mm/r (in G99 mode)

Description

1. Straight cutting cycle



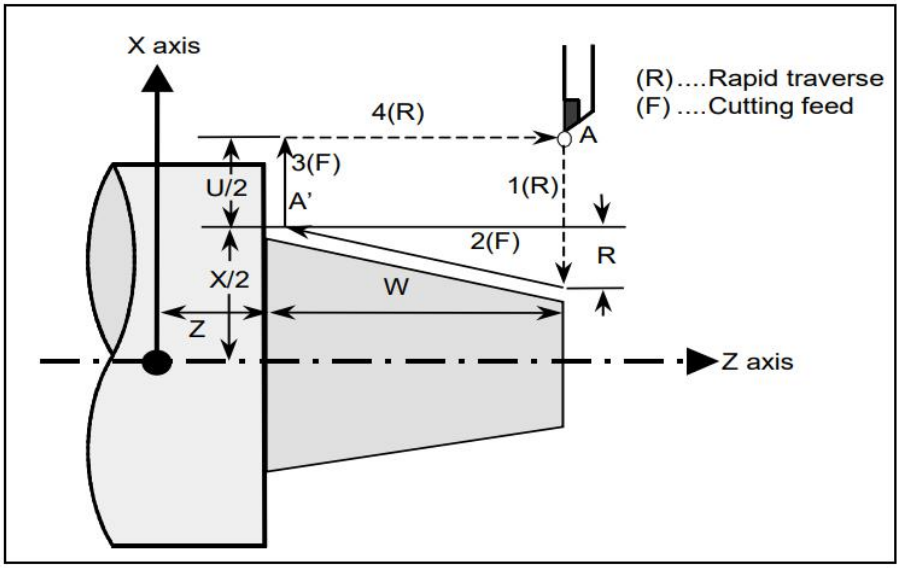
(1) In command G90, the figure behind X, Z is the coordinate of the end point of path 2. The positive and negative value of this position relative to the position of cutting start point (start point of path 1) in X direction determines the running direction of path 1; the positive and negative value of this position relative to the position of cutting start point in Z direction determines the running direction of path 2.

(2) The figure behind U, W of command G90 is the difference between the end point of path 2 and the cutting start point (the start point of path 1). The positive and negative value of U determines the running direction of path 1, and the positive and negative value of W determines the running direction of path 2.

(3) When the single segment function is on, press Start Cycle to execute operation according to the order 1→2→3→4→1 ……

(4) G90 is a modal command. For the programming of several continuous cycles, G90 and the modal command Z, W, F can be omitted, it just needs to carry out X(U) programming, thus simplifying the programming.

2. Circular conical cutting cycle

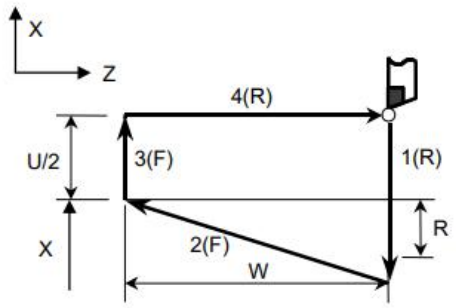
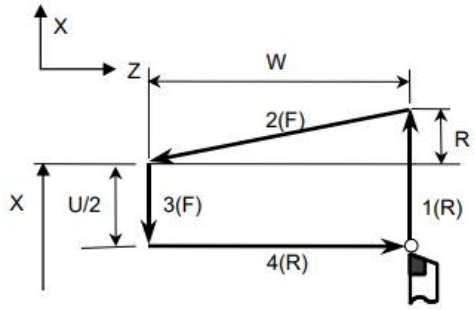
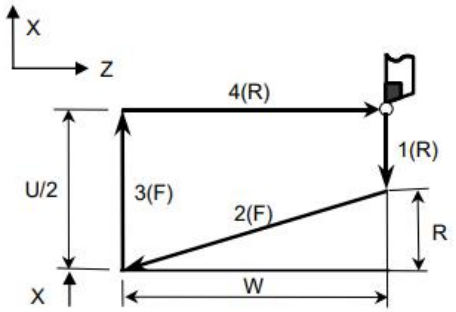
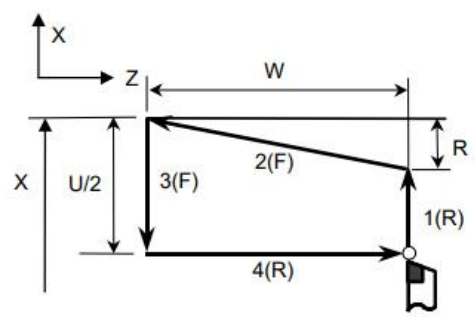


(1) The running path of G90 circular conical surface cutting command is the same with the cylindrical surface command, the the position of the circular conical surface initial cutting point at X direction is jointly determined by the value of X and R.

(2) When the single segment function is on, press Start Cycle to execute operation according to the order 1→2→3→4→1 ……

(3) G90 is a modal command. For the programming of several continuous cycles, G90 and the modal command Z, W, F can be omitted, it just needs to carry out X(U) programming, thus simplifying the programming.

For different cutting start points, there are four paths for code G90. The relation between U, W, R and the tool path is shown as below:

Outer diameter machining	Internal diameter machining
1. U < 0, W < 0, R < 0	**2. U > 0, W < 0, R > 0**
	
3. U < 0, W < 0, R > 0 at |R|≤|U/2|	**4. U > 0, W < 0, R < 0 at |R|≤|U/2|**
	

2.4.13.2 Threading Cycle (G92)

Format

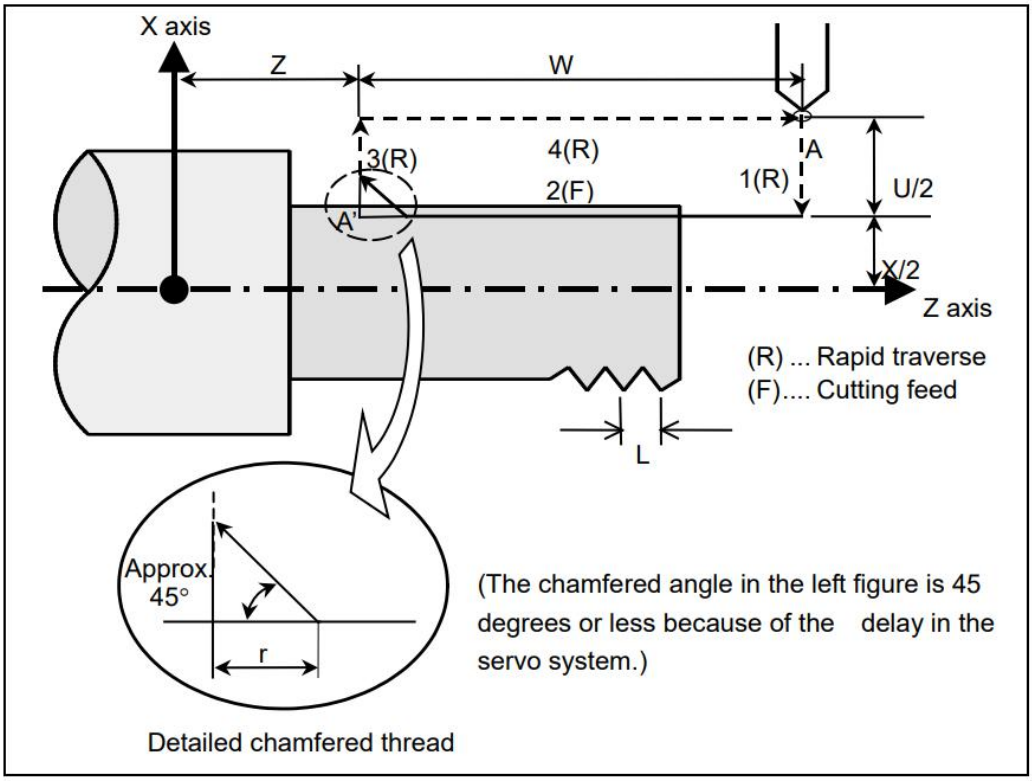
G92 X/U_ Z/W_ R_ F_;	
X/U	: X-axis coordinate data
Z/W	: Z-axis coordinate data
R	: Radius difference between the conical surface start point and the conical surface end point;
F	: Metric thread lead, lead in the long axis direction, unit: mm
I	: Inch screw thread lead, number of teeth per inch in the long axis direction
Q	: The starting angle of thread processing defaults to 0° when not specified

Extension characters

α/β	: Additional axis end point coordinates (α is absolute programming, and β is incremental programming) (multiple axes can be specified)
J	: Retracted length in the long axis direction (valid only in high-precision mode), unit: mm
K	: The amount of lead change per turn, in mm
P	: Thread feed starting height (radius specified), unit: mm

Description

Straight threading cycle



The ranges of thread leads and restrictions related to the spindle speed are the same as for threading with G32.

A straight threading cycle performs four operations:

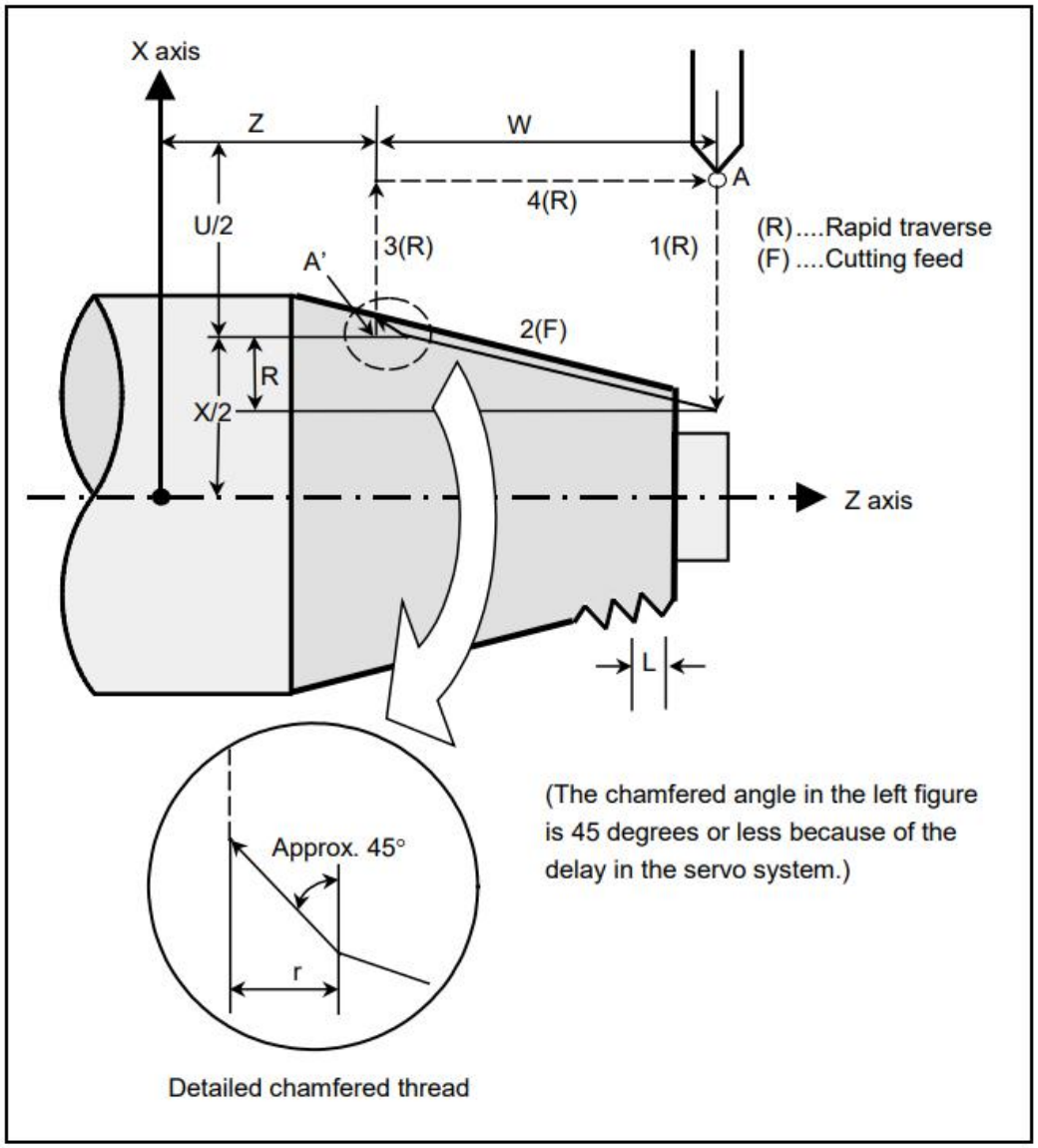
(1) Operation 1 moves the tool from the start point (A)to the specified coordinate of the second axis on the plane(specified X-coordinate for the ZX plane)in rapid traverse.

(2) Operation 2 moves the tool to the specified coordinate of the first axis on the plane (specified Z-coordinate for the ZX plane)in cutting feed.At this time,thread chamfering is performed.

(3) Operation 3 moves the tool to the start coordinate of the second axis on the plane (start X-coordinate for the ZX plane)in rapid traverse.(Retraction after chamfering)

(4) Operation 4 moves the tool to the start coordinate of the first axis on the plane(start Z-coordinate for the ZX plane)in rapid traverse.(The tool returns to the start point (A).)

Taper threading cycle



The ranges of thread leads and restrictions related to the spindle speed are the same as for threading with G32.

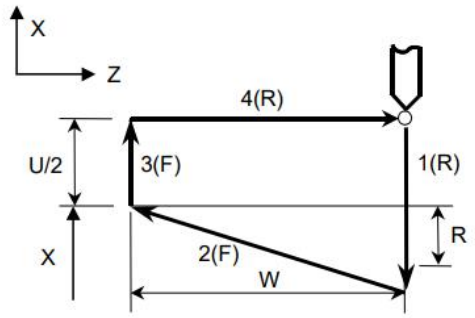
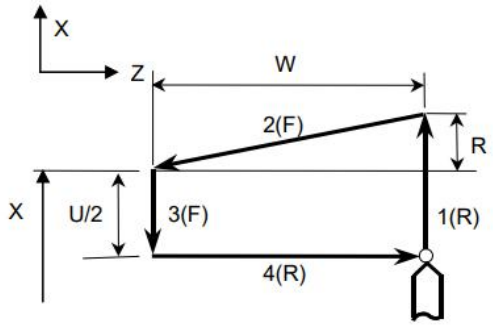
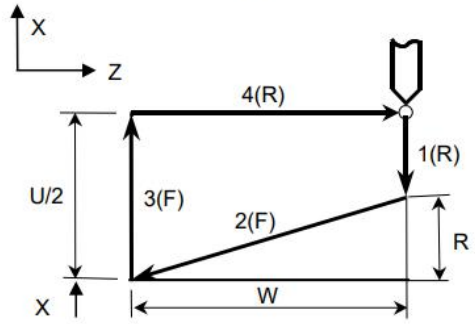
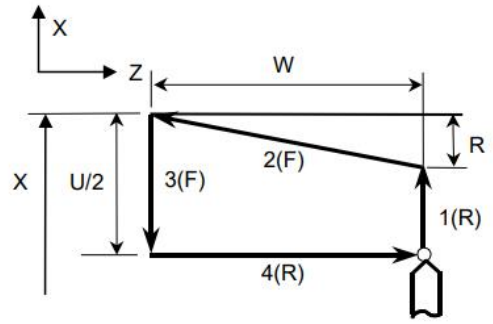
The figure of a taper is determined by the coordinates of the cuting end point (A') in the direction of theength and the sign of the taper amount (address R). For the cycle in the Fig), a minus sign is added to the taper amount.

A taper threading cycle performs the same four operations as a straight threading cycle.However, operation 1 moves the tool from the start point (A) to the position obtained by adding the taperamount to the specified coordinate of the second axis on the plane (specified X-coordinate for the ZX plane)in rapid traverse.

Operations 2, 3, and 4 after operation 1 are the same as for a straight threading cycle.

Relationship between the sign of the taper amount and tool path

The tool path is determined according to the relationship between the sign of the taper amount (address Rand the cutting end point in the direction of the length in the absolute or incremental programming as Table .

Outer diameter machining	Internal diameter machining
1. U < 0, W < 0, R < 0	**2. U > 0, W < 0, R > 0**
	
3. U < 0, W < 0, R > 0 at |R|≤|U/2|	**4. U > 0, W < 0, R < 0 at |R|≤|U/2|**
	

2.4.13.3 End Face Turning Cycle (G94)

Format

G94 X/U_ Z/W_ R_ F_;

X/U : X-axis coordinate data

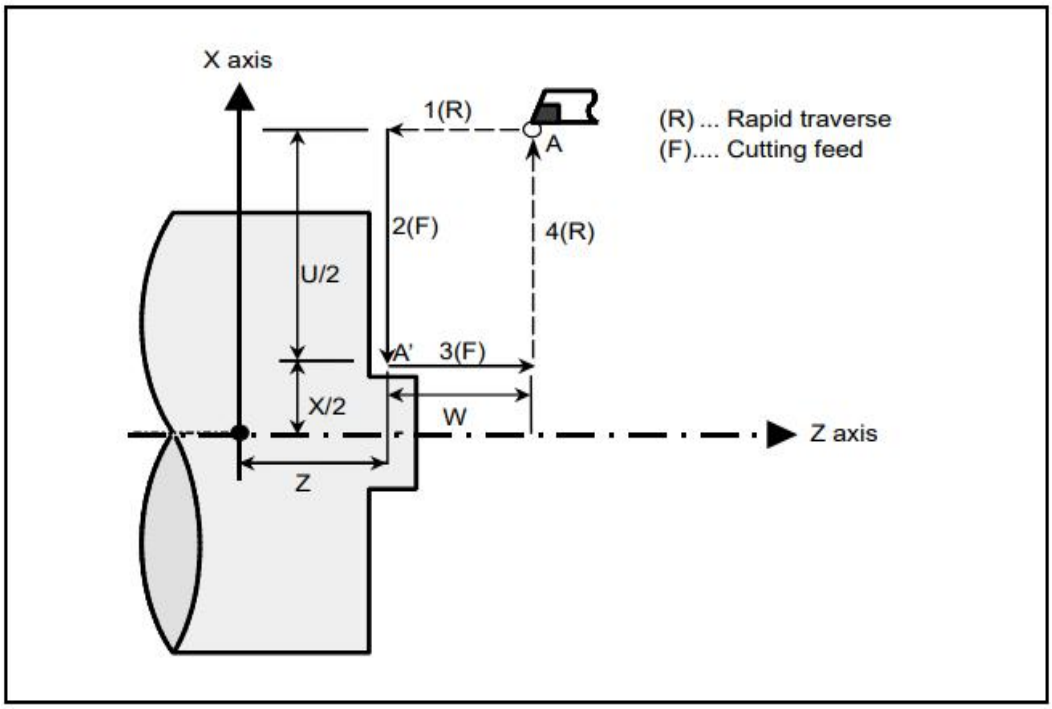
Z/W : Z-axis coordinate data

R : radius difference between the conical surface start point and the conical surface end point;

F : Feed rate, unit: mm/min (in G98 mode), mm/r (in G99 mode)

Description

1. Face cutting cycle



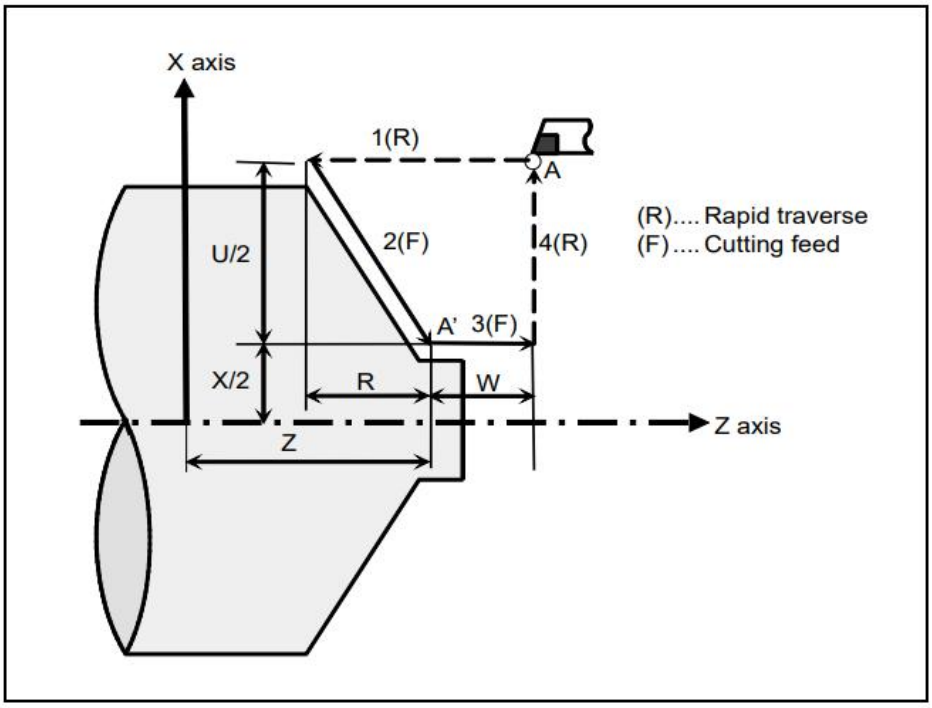
(1) In command G94, the figure behind X, Z is the coordinate of the end point of path 2. The positive and negative value of this position relative to the position of cutting start point (start point of path 1) at Z direction determines the running direction of path 1; the positive and negative value of this position relative to the position of cutting start point (start point of path 1) at X direction determines the running direction of path 2.

(2) The figure behind U, W of command G94 is the difference between the end point of path 2 and the cutting start point (the start point of path 1). The positive and negative value of U determines the running direction of path 2, and the positive and negative value of W determines the running direction of path 1. In the above cycle, U is a negative value and W is also a negative value.

(3) When the single segment function is on, press Start Cycle to execute operation according to the order 1→2→3→4→1 ……

(4) G94 is a modal command. For the programming of several continuous cycles, G94 and the modal command X, U, F can be omitted, it just needs to carry out Z/W programming, thus simplifying the programming.

2. Taper cutting cycle

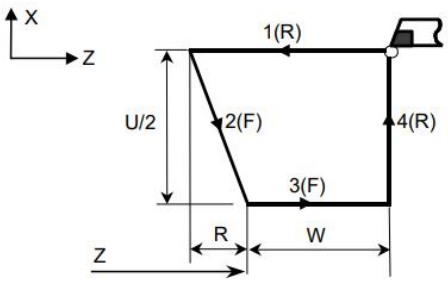
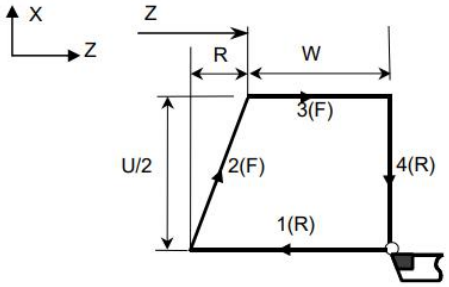
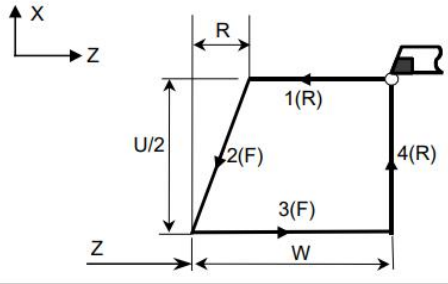
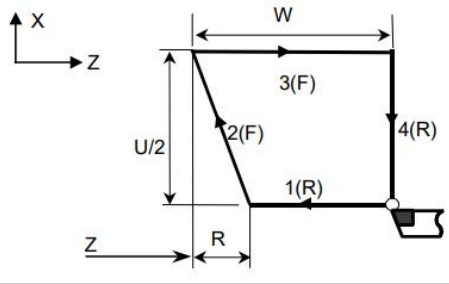


(1) The running path of circular conical end cutting command G94 is the same with the end plane command, the position of the circular conical end plane initial cutting point at Z direction is jointly determined by the value of Z and R.

(2) When the single segment function is on, press Start Cycle to execute operation according to the order 1→2→3→4→1 ……

(3) For the programming of several continuous cycles, G94 and the modal command Z, W, R, F can be omitted, it just needs to carry out Z/W programming, and it is the same as the simplified programming of end plane cutting.

For different cutting start points, there are four paths for code G94. The relation between U, W, R and the tool path is shown as below:

Outer diameter machining	Internal diameter machining
1. U < 0, W < 0, R < 0	**2. U > 0, W < 0, R < 0**
	
3. U < 0, W < 0, R > 0 at |R|≤|W|	**4. U > 0, W < 0, R > 0 at |R|≤|W|**
	

2.5 FEED FUNCTIONS

2.5.1 Cutting Feedrate Control

Function name	G code	Validity of G code	Description
Exact stop	G09	This function is valid for specified blocks only	The tool is decelerated at the end point of a block, then an in-position check is made.Then the next block is executed.
Exact stop mode	G61	Once specified, this function is valid until G64, is specified.	The tool is decelerated at the end point of ablock, then an in-position check is made.Then the next block is executed.
Cutting mode	G64	Once specified, this function is valid until G61, is specified.	The tool is not decelerated at the end point of a block, but the next block is executed.

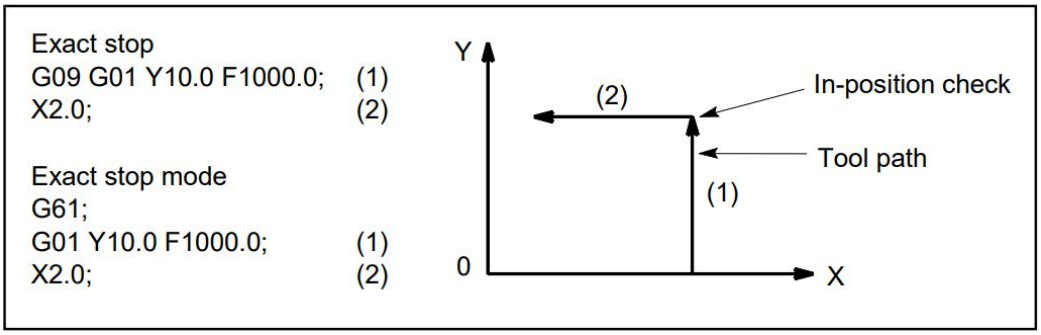
Format

G09 G01(G02,G03)...;	**Exact stop**
G61;	**Exact stop mode**
G64;	**Cutting mode**

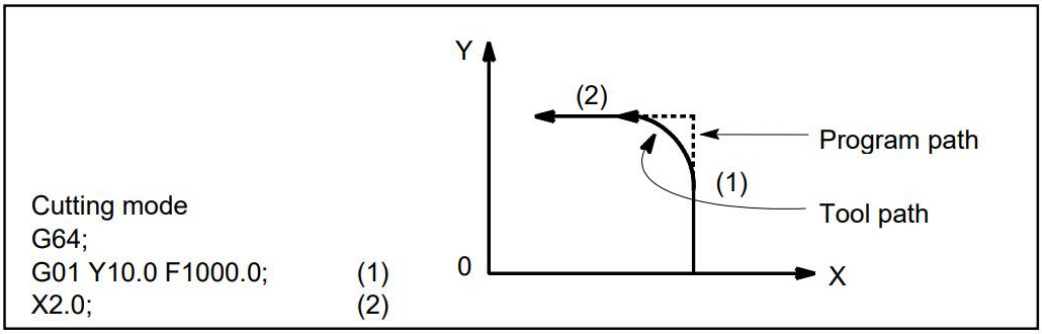
Description

The inter-block paths followed by the tool in the exact stop mode, cutting mode, and tapping mode are different.

At the end point of a block specifying exact stop (G09), or each block during exact stop mode (G61)cutting feed is decelerated and an in-position check is performed (Fig).



At the end of each block during tapping mode (G63) or cutting mode (G64), deceleration of cutting feed is not performed and execution proceeds to the next block (Fig).



2.5.2 High-precision Oscillation Function (G81.1/G80.1)

This function is an optional function, available in models: 21GD series, 31 series

Format

G81.1 α_ Q_ F_; Start of oscillation action

α : Position of the axis coordinates of the upper dead center, specified by absolute coordinate values

Q : Distance between the upper dead point and lower dead point

F : Oscillation reference speed, unit: mm/min

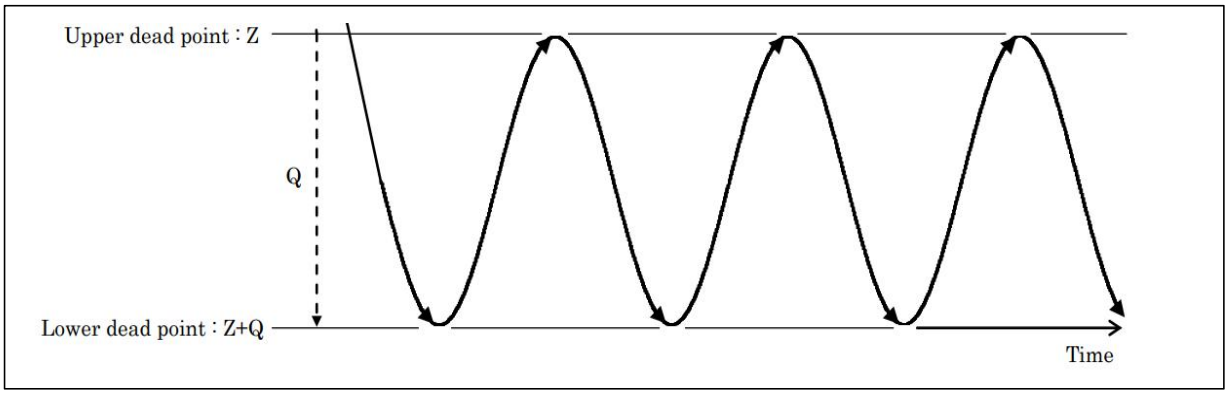
G80.1 α_; oscillation action canceled

α : Position of the axis after cancellation of oscillation, specified by absolute coordinates

Description

By making the movement speed of the oscillation axis vary sinusoidally, it is possible to ensure accurate movement toward the upper and lower dead centers. In addition, it is possible to make the prior feedforward function effective with respect to the oscillation action. As a result, the servo delay can be almost eliminated and the accuracy can be further improved when the position of the upper and lower dead centers and the oscillation speed have been changed.

It is generally used for surface grinding of shaft workpieces, which can improve the workpiece finish.



1. Before using this command, bit 1 of the CNC parameter-5700 must be set to open the oscillation function of each axis;
2. When resetting, the oscillation axis decelerates to a stop or stops at the upper dead center position, as determined by bit 2 of sys-parameter-5700;
3. In the execution of G81.1, the speed of the oscillating axis is not affected by the feed override, G99 per revolution of feed;
4. Axis positioning coordinates must be specified when executing G80.1 to cancel oscillation motion;

Example

O1000	
T0001	
G00 X51 Z0	
G81.1 Z0 Q100 F2000	// Move the Z-axis reciprocally between coordinates 0~100 at a speed of 2,000 mm/min
G01 X50 F10	//X-directional feed
G00 X100	// Tool retracting
G80.1 Z0	//Cancel Z-axis oscillation and move Z-axis to the Z0 position
M30	

2.5.3 Feed per minute (G98)

Format

G98;

Description

1. G98 is the per minute feed mode. Under G98 mode, the tool feed speed is specified by the figure behind F.
2. Unit of F in G98 mode: mm/min;
3. G98 is a modal value. Once G98 mode is specified, it will not be valid until command G99 (per revolution feed) is executed.
4. The default mode is G98/G99 after system power-on, which is specified by the 3rd bit of sys-parameter 2201.

2.5.4 Per revolution feed (G99)

Format

G99;

Description

1. G99 is the per revolution feed mode. Under G99 mode, the feed amount per revolution of the spindle is specified by the figure behind F.
2. Units of F in G99 mode: mm/r;
3. G99 is a modal value. If the status of G99 is specified, it will not be valid until command G98 (per minute feed) is executed.

	Feed per minute	Feed per revolution
Specified address	F	F
Specified code	G98	G99
Specified range	1～60,000 mm/min (F1～F60000)	0.01～500.00mm/rev (F1～F50000)

4. When the rotate speed of position encoder is below 1 revolution/minute, the processing will not be even. The slower the rotational speed is, the more uneven it will be.
5. When G99 is the per revolution feed mode, the spindle must have angular position feedback;

2.5.5 Automatic chip breaking (G98.1/G99.1)

Format

G98.1/G99.1 P_ Q_;

G98.1 : Chip breaking function in G98 mode

G99.1 : Chip breaking function in G99 mode

P : Number of chip-breaking turns, when P>0, pause type chip-breaking mode
When P<0, retract type chip-breaking mode

Q : The number of chip-breaking turns is judged, and during cutting feed, the spindle makes 1 chip-breaking action for each specified number of turns of Q

Description

This command is a modal command. After executing this command, the CNC will detect the spindle encoder signal during feeding, and judge the number of turns. After each feeding Q specifies the number of turns, chip breaking is performed, and the number of chip breaking turns is the number of turns specified by P.

1. Support chip breaking between G70~G75 compound cycle commands;
2. The execution of the G98/G99 command cancels the chip-break function;
3. When Q is specified as a negative number (retracting type), the spindle speed should not be set too fast, otherwise the retracting action will be filtered by the servo drive;
4. The chip-breaking mode will be canceled when a reset is performed;
5. Can be used in conjunction with SPSV commands to improve chip breaking;

Example

O1000	
G98.1 P-1.5 Q5	//Chip breaking in per minute feed mode, retracting type chip breaking mode
T0101	
M03 S500	
G00 X50	//Chip breaking actions will not be performed during fast positioning
G00 Z0	
G90 W-10 F200	//Chip breaking occurs - 1st cutting
X49.5	//Chip breaking occurs - 2nd cutting
G98 X49	//Cancel chip breaking - 3rd cutting for finish turning
M30	

2.6 SPINDLE FUNCTIONS

2.6.1 Spindle Speed Command (S)

The code signal is transmitted to the machine tool through address character S and the data behind it, and the signal is used to control the spindle speed of the machine tool.

Format 1: (used for speed control)

Sxxxx;
xxxx : Set speed in r/min

Description

1. S is a modal command that remains in effect until the next S command;
2. At power-on, the default spindle S value is set by sys-parameter-2221;

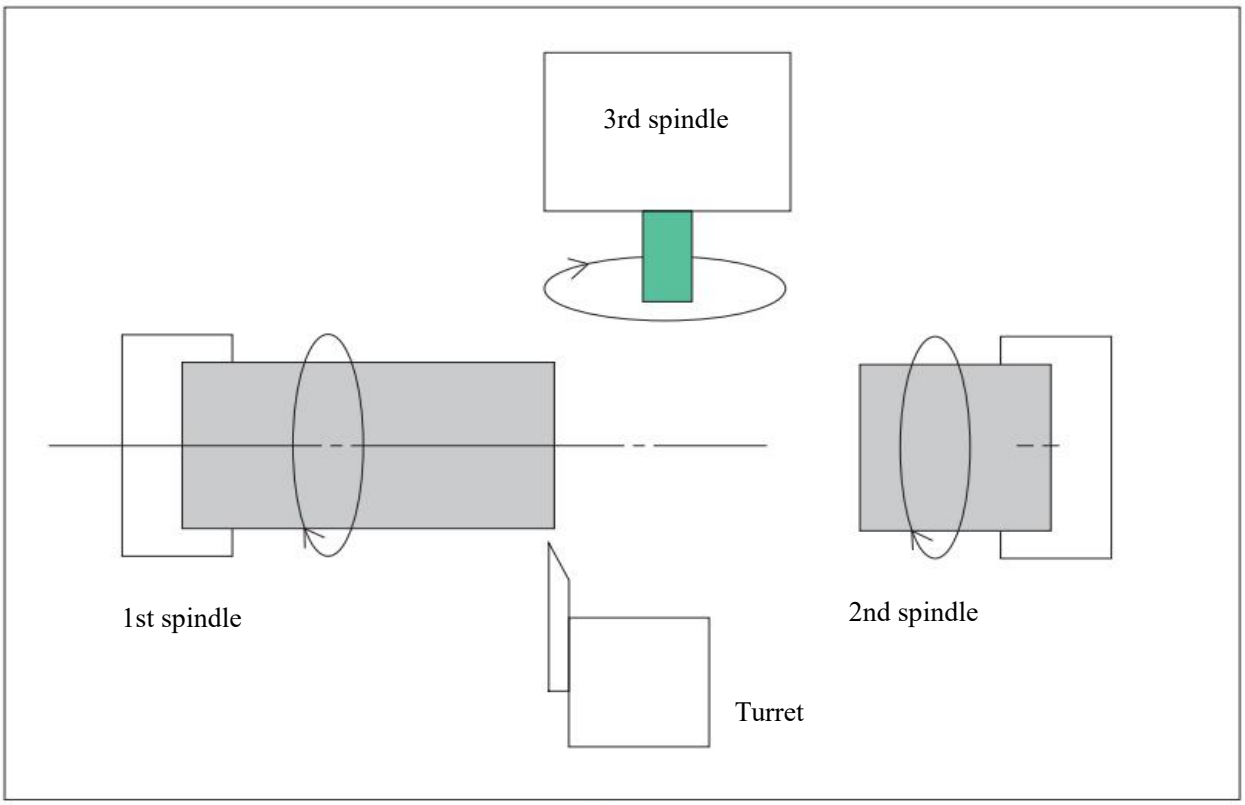
Format 2: (used for gear control)

Sn;
n : Gear number, default range: 0~4

Description

1. The spindle type needs to be set to gear control (sys-parameter-1011 is set to 1) for the command to take effect;
2. S0 to S4 are equivalent to M40 to M41;

2.6.2 Multi-spindle Speed Command (Sn=xxxx)



Format 1:

Sn=xxxx;	
n	: Spindle number, it is 1 when omitted, e.g., S1=500 is equivalent to S500
xxxx	: Set speed in r/min

Format 2:

S_ Paab;	
S	: Set speed in r/min
P	: indicates channel number, b indicates spindle number, when aa and b are omitted, it indicates the 1st spindle of the current channel

Description

The command is executed properly when bit 4 of sys-parameter-1000 is set to be valid for multiple spindles in the channel.

Example

M03 S500 (or S1=500 or S500 P1)	//Set the 1st spindle speed to 500 r/min //M03 is the command for the 1st spindle forward rotation
M103 S2=1500 (or S1500 P2)	//Set the 2nd spindle speed to 1,500 r/min //M103 is the command for the 2nd spindle forward rotation
M204 S3=50 (or S50 P3)	//Set the 3rd spindle speed to 50 r/min //M204 is the command for the 3rd spindle backward rotation

2.6.3 Specify the Current Spindle (SP)

Format

SPaab;

aa : Channel number, range: 1~64, it indicates the current channel when omitted

b : Spindle number, cannot be omitted, e.g., SP2 indicates that the 2nd spindle is set as the current spindle

Description

When the machine has multiple spindles, this command is used to switch the spindle currently in use.

1. The command is executed properly when bit 4 of sys-parameter-1000 is set to be valid for multiple spindles in the channel.

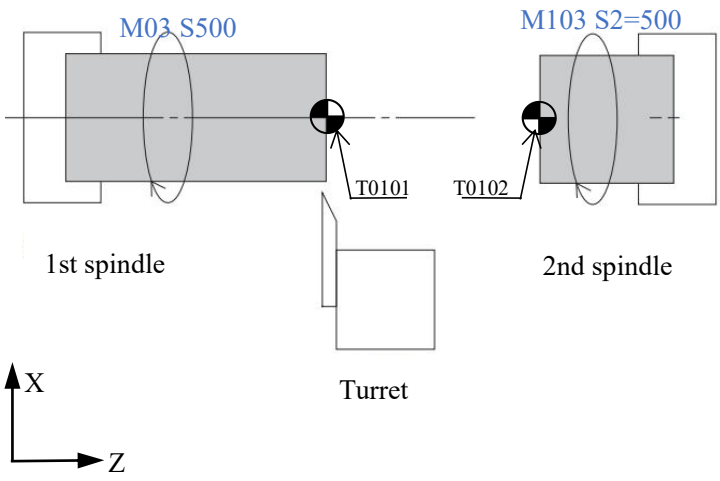
2. Impacts on the commands in the following table:

Functional commands	Description
Thread cutting G32	Specify G32 with SPn to follow the speed feedback of the nth spindle for thread cutting
Variable pitch thread cutting G34	Specify G34 with SPn to follow the speed feedback of the nth spindle for thread cutting
Tapping cycle G84/G88 (follow-up mode)	Specify the speed feedback of the nth spindle with SPn for tapping
Multiple thread cycle G76	Specify G76 with SPn to follow the speed feedback of the nth spindle for thread cutting
Threading cycle G92	Specify G92 with SPn to follow the speed feedback of the nth spindle for thread cutting
Spindle constant linear speed control G96/G97	Specify the reference spindle as the nth spindle with SPn
Per revolution feed G99	Specify the reference spindle of G99 as the nth spindle with SPn
Automatic chip breaking G98.1/G99.1	Specify the reference spindle at automatic chip breaking as the nth spindle with SPn
Spindle speed oscillation SPSV	Specify the nth spindle with SPn for speed oscillation processing

3. No impacts on the commands in the following table:

Functional commands	Description
Polygon machining G51.2 (follow-up mode)	SPn cannot switch the spindle used by these commands. these commands use the spindle specified by their respective sys-parameters
Electronic cam G51.8 (follow-up mode)	
Electronic gearbox EGB G81.4 (follow-up mode)	
Spindle synchronization G114.1 (follow-up mode)	

Example

Application in dual spindle thread cutting	
O1000	
T0101 M03 S500 SP1	
G00 X50 Z0	
G32 W-10 F1	
G00 X50	
T0102 M103 S2=500 SP2	
G00 Z0	
G32 W10 F1	
G00 X50	
G00 Z0	
M30	

2.6.4 Actual Spindle Speed Arrival (G04 PSn)

Format

G04 PSn=xxxx;

PSn : The spindle speed reaches the judgment value, n: spindle number,

e.g., G04 PS1=xxxx

Description

The command is used to detect whether the spindle speed fed back from the CNC arrives or to give timeout alarm according to the usage occasion and demand;

1. The n character of PSn when omitted indicates the 1st spindle, e.g., G04 PS1 = 500 is equivalent to G04 PS500;
2. The spindle speed range to be detected is set by sys-parameter-1057;
3. The time to be detected for spindle arrival timeout is set by sys-parameter-1059;
4. Can be used in conjunction with the G04 time delay command and the G04 time skip command;

Example

O1000	
T0101	
M03 S3000	
G04 PS3000	//Wait for the actual spindle feedback speed to reach 3,000 r/min
...	//Workpiece processing
M05	//Spindle stops
G04 PS0 (or G04 PS1=0)	//Wait for the actual spindle feedback speed to reach 0 r/min
M98 P9200	//Call the subprogram for automatic loading/unloading of workpieces
M30	

2.6.5 Programmable Spindle Precision Stop (M05 PN/PA)

Format

M05 PA_ PN_;

M05 : Standard M command (spindle stop command), other spindles can be specified, such as M105, M205, etc.

PA : Positioned to the angle specified by PA when the specified spindle executes the stop command

PN : Equivalent fraction, specifying the angle of positioning to 360°/PN when the spindle executes the stop command

Description

When the spindle is in position mode and in the M03/M04 state, this command is executed to stop the spindle at the specified angular position.

1. The spindle must be in position mode or the spindle type must be position spindle (sys-parameter-1011 is set to 2);
2. Bit 3 of sys-parameter-1004 needs to be set to 1 for the command to be executed properly;
3. When this command is executed, the spindle deceleration time is set by sys-parameter-1074;

Example

Example 1: The spindle (A-axis) quickly orientates to the origin of the coordinates	
M03 S3000	//Spindle rotates forward at 3,000
...	
M05 PA0	//Spindle stops at the position where the coordinate is 0°
Example 2: application of indexing drilling Assume that there is a 3-axis lathe with a servo spindle as the 3rd axis (A-axis)	
O1000	
T0101	
M18	//Switch position mode of the 1st spindle
M03 S2000	
...	//Excircle turning
M103 S2=1000	//Start the 2nd spindle (powered milling head)
M05 PN4	//Spindle stops at one of angles 0, 90, 180, and 270 according to the principle of proximity
G04 PS1=0 (or G04 PS0)	//Check the speed feedback to ensure that the A-axis (1st spindle) actually stops
N10	
M12	//A-axis (the 1st spindle) locks to prevent A-axis movement during drilling
G01 W-15 F200	//Z-directional drilling
G00 W15	// Tool retracting
M13	//A-axis (the 1st spindle) is unlocked
G00 UA90	//A-axis (the 1st spindle) incrementally rotates 90°
M92 N10 L4	//Jump to N10 and perform 4 cycles of the drilling action
M30	

2.6.6 Polygon Turning(G51.2/G50.2)

Format

G51.2 P_ Q_ R_; (Polygon machining begins)

P : Proportional coefficient (unsigned) (integer) of the active spindle speed

Q : Proportional coefficient (integer) of the speed of the slave axis;
the symbol is used to control the direction of the slave axis.

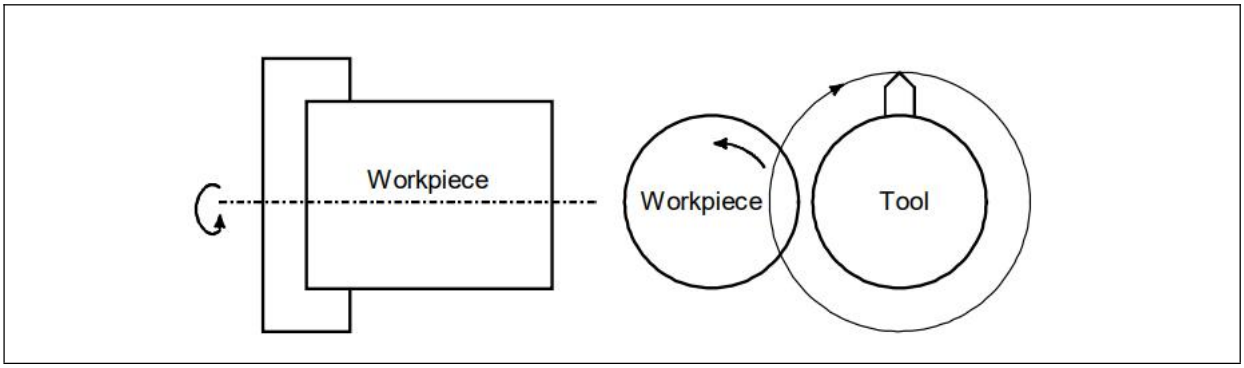
R : Start angle offset of the active spindle (unsigned), range: 0°~360.000°

G50.2 P_; (Cancel polygon machining)

P : Indicates P0, polygon machining is canceled without waiting for the slave axis to stop,
When it is omitted or when P1 is specified, G50.2 will wait for the slave axis to stop.

Description

By changing the rotation ratio between the workpiece and the cutting tool and the number of cutting tips installed in the tool, the workpiece can be machined into a quadrangle or hexagon.



1. G51.2 operates in follow-up mode and synchronous mode as follows:

Follow-up mode	Synchronous mode
Principle: The slave axis follows spindle speed feedback for position control	**Principle:** Interpolation control of slave axis and spindle
Requirements: (1) The spindle operates in speed mode or position mode (2) The spindle must be fed by speed feedback	**Requirements:** (1) The spindle must operate in position mode (2) The spindle does not require speed feedback
sys-parameter setting: Bit 1 of sys-parameter-5300 is set to 0 or Bit 1 of sys-parameter-5300 is set to 1 and the spindle is in speed mode	**sys-parameter setting:** Bit 1 of sys-parameter-5300 is set to 1 and the spindle is in position mode

2. The master axis must be a spindle. sys-parameter-5310 is used to set which spindle of the CNC is to be used as the active spindle of the G51.2.

3. The slave axis may be either a spindle or a common servo shaft. The control axis of the slave axis is set by sys-parameter-5311, and the slave axis must be specified as a rotating axis (bit 2 of sys-parameter-0001).

4. Matters related to G51.2 execution actions:

(1) If the slave axis is set to the spindle type, the spindle as the slave axis must stop in order to execute G51.2;

(2) G51.2 can be executed to synchronize the slave axis, whether the active spindle is at rest or running;

(3) The acceleration and deceleration of the slave axis for position synchronization during the execution of G51.2 is set by sys-parameter-5313;

(4) G51.2 cannot be executed properly if the setting value of PQ is not reasonable or if the speed of the active spindle is too high, causing the speed of the slave axis to be higher than the setting value of sys-parameter-5312;

(5) When G51.2 is executed in the follow-up mode, the CNC will check whether the difference between the speed feedback of the active spindle and the set speed is within the range of the set value of sys-parameter-5315, and generate an alarm if the speed difference is too large and the time exceeds the set value of sys-parameter-5316.

(6) When bit 5 of sys-parameter-5300 is set to 0, the execution of G51.2 will wait for the synchronization of the active spindle and slave axis to be completed before the CNC executes the next program segment;

(7) When bit 5 of sys-parameter-5300 is set to 1, the G51.2 command does not wait for the active spindle and the driven axis to complete synchronization, and the CNC immediately executes the next program segment, and when it executes cutting commands such as G01, the CNC will wait for the active spindle and the slave axis to complete synchronization.

5. Matters related to G50.2 execution actions:

(1) When G50.2 is executed, only the slave axis is released from the synchronization state; the operating state of the active spindle is not affected by G50.2.

(2) The deceleration time of the slave axis for the execution of G50.2 is set by sys-parameter-5313;

(3) If P1 is specified in the program line in which G50.2 is located or P is omitted, the G50.2 command will wait for the completion of the slave axis stop before the CNC can execute the next program segment; if P0 is specified in the program line in which G50.2 is located, the G50.2 command will not wait for the completion of the slave axis stop and the CNC will execute the next program segment immediately.

6. When bit 4 of sys-parameter-5300 is set to 0, the synchronization state of the active spindle and the slave axis is released after the active spindle stops in case of an emergency stop or reset.

7. When bit 4 of sys-parameter-5300 is set to 1, the active spindle and the slave axis are immediately released from the synchronization state during an emergency stop or reset, and the slave axis is decelerated to a stop in accordance with the acceleration and deceleration times set by sys-parameter-5313.

8. Superimposed motion of the slave axis during the synchronization between the active spindle and the slave axis can be used for helical tooth processing;

9. Changing the speed of the active spindle during G51.2 synchronization is not recommended due to the presence of servo-following errors;

Example

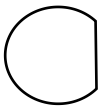
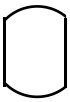
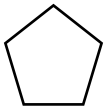
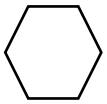
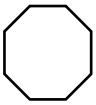
The machine is a 3-axis lathe, X and Z are linear axes, A axis is the 2nd spindle (tool disc), and the 1st spindle is the analog spindle.

Square machining with 2 tools in the cutterhead

Since the spindle is controlled by analog, G51.2 is executed in follow-up mode (encoder feedback for the 1st spindle)

O1000	
T0101 M105	//Ensure that the 2nd spindle is at a stop
M03 S500	//The 1st spindle speed is set to 500 r/min
G00 X50 Z2	//Safe positioning
G51.2 P1 Q2	//Synchronization between the tool axis and workpiece axis; at this time the slave axis speed is increased to 1,000 r/min
G01 X45 F100	
G01 W-15 F200	//Cutting
G00 U10	//Tool retracting X
G00 Z100	//Tool retracting Z
M05	//The 1st spindle stops
G50.2	//Synchronization between the tool axis and workpiece axis is canceled
M30	

References for tool selection according to workpiece shape:

Shape	Number of tools in the slave axis	Active spindle and slave axis Rotational speed ratio P/Q	Plane effect
	1	1:1	Serious bulge
	2	1:1	There are bulges
	1	2:1	Very suitable, almost no bulge
	3	1:1	There are bulges
	2	1.5:1	Suitable, a few bulges
	1	3:1	Very suitable almost no bulge
	2	2:1	Very suitable, almost no bulge
	1	4:1	There are bulges
	3	1.66:1	Suitable, a few bulges
	2	2.5:1	Very suitable almost no bulge
	1	5:1	There are bulges
	3	2:1	Very suitable, almost no bulge
	2	3:1	There are bulges
	4	2:1	Very suitable almost no bulge
	2	4:1	There are bulges

2.6.7 Electronic Cam Function (G51.8/G50.8)

Need model support, models: 21GD series, 31 series

Format

G51.8 P100 L360 R_; Electronic cam mode starts

P : The first address of the macro variable of the cam lift point data
e.g., P100 means starting from macro variable #100

L : The number of cam lift point data, e.g., L360 represents 360 point data

R : Start angle offset (unsigned), range: 0°~ 360.000°

G50.8; Electronic cam follow-up mode cancellation

Description

The electronic cam is a method that uses mathematical algorithms to implement a "cam curve" instead of a mechanical cam, hence it is named electronic cam. Electronic cams offer better flexibility than mechanical cams. The cam curve can be freely modified in the program to achieve different process requirements.

The control of electronic cams is subject to three elements:

Master axis: the reference axis to be used for synchronization control; the CNC parameter-5350 is used to set the spindle number, and bit 1 of sys-parameter-5340 is used to set whether to follow the encoder or the spindle coordinates.

Slave axis: servo shaft that follows the motion according to the cam table and the spindle position; the slave axis is set by sys-parameter-5351.

Data points in cam lift table: data describing the angle-position relationship corresponding to the spindle and slave axis;

Detailed description

1. This function is only available for 21GD and 31 series CNC systems;
2. In the electronic cam mode, the other axes can be moved freely and the slave axiss can be programmed for superposition of movements;
3. The number of data points is determined by the number of system macro variables, e.g., #1 to #999 indicates a maximum of 999 data points;
4. The accuracy of the cam follow-up depends on the number of data points, e.g., 360 data points means that one revolution of the spindle is divided into 360 equal parts and the value of each data point represents the amount of change in the slave axis;
5. When G51.8 is executed, the slave axis will be synchronized with the angle at which the current spindle is located, and then the slave axis will move the corresponding value incrementally according to the lift table data corresponding to the spindle angle relationship;
6. Bit 2 of sys-parameter-5340 is used to set whether the reset cancels the follow-up mode;
7. G51.8 belongs to group 00 of G code;

Example

Let the cam base circle diameter be 50 mm, the spindle (1st spindle) be the servo-controlled A axis, and X be the slave axis (follow-up axis) Bit 1 of sys-parameter is set to 1, sys-parameter-5350 is set to 1 (spindle A), and sys-parameter-5351 is set to 1 (X axis)	
O1000 (main program)	O8000 (track data point assignment)
G00 X55 //Safety point positioning	#100=0
M98 P8000 //Load data points	#101=0
G51.8 P100 L360 //X-axis follow-up starts	#101=0.02
M03 S100 //Spindle starts	#102=0.08 360 data points in total
G01 U-1 F1 //Grinding feed	...
G04 X5 //Polishing	#459=0
G00 U10 //Tool retracting	M99
G50.8 //Cancel X follow-up	
M30	

2.6.8 Cam Data Radius Compensation Processing (CAMRP)

Format

CAMRP[P, L, BR, GR]; Track data point C tool compensation processing

P	: The first address of the macro variable of the cam lift point data, e.g., P100 means starting from macro variable #100
L	: The number of cam lift point data, e.g., L360 represents 360 point data
BR	: base circle radius
GR	: Grinding wheel radius

Description

This command is usually used in conjunction with the electronic cam command for the optimization of track point data.

When the radius of the grinding wheel or the radius of the base circle of the workpiece changes, the new data track point is regenerated by this command.

Compensation method is based on data compensation when the tool nose type is 0.

2.6.9 Ellipse Follow-up Function (G51.8 H1)

Need model support, models: 21GD series, 31 series

Format

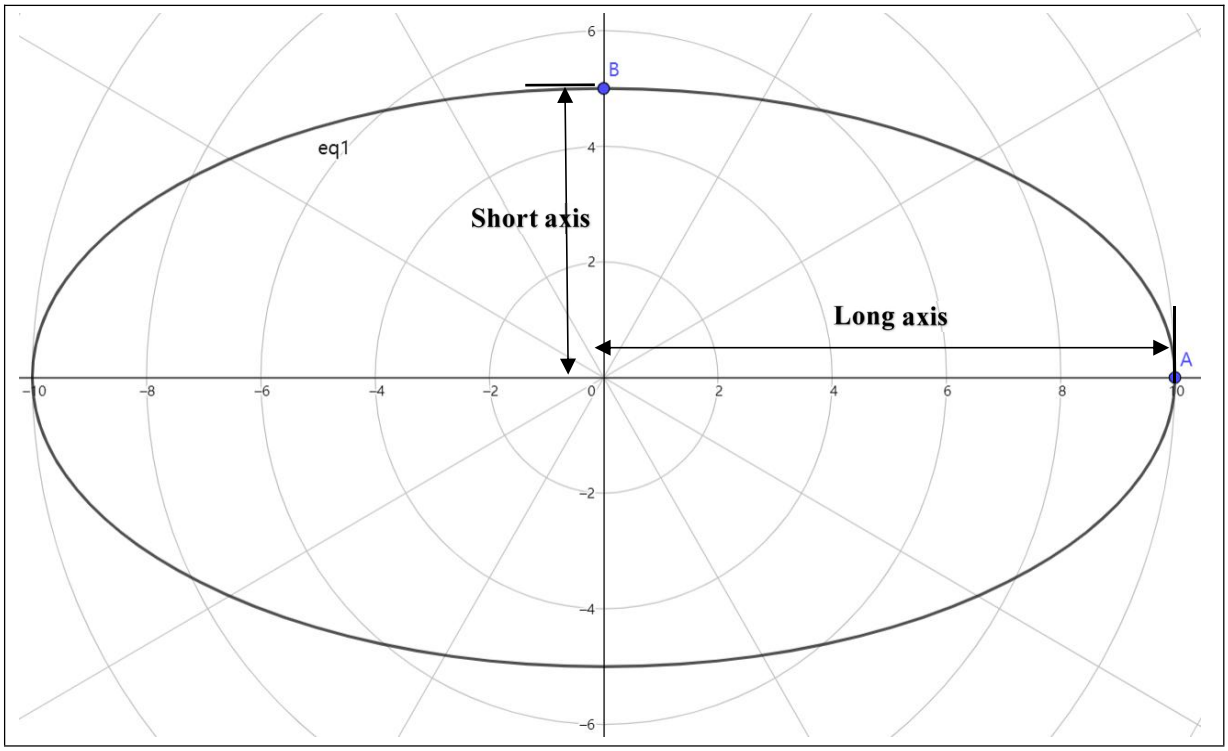
G51.8 H1 A_ B_ R_; Ellipse follow-up mode

A	: Long axis radius of the ellipse
B	: Short axis radius of the ellipse
R	: Start angle offset (unsigned), range: 0˚~ 360.000˚

G50.8; Follow-up mode cancellation

Description

This command is an extension of the electronic cam command, and the other axes can be moved freely when the follow-up mode is activated.



Example

Longitudinal elliptical turning, the ellipse long axis is 10 mm and the short axis is 8 mm	
Let the spindle (1st spindle) be the servo-controlled A axis, and X be the slave axis (follow-up axis) Bit 1 of sys-parameter is set to 1, sys-parameter-5350 is set to 1 (spindle A), and sys-parameter-5351 is set to 1 (X axis)	
O1000	
G00 X55	// Safety point positioning
G51.8 H1 A10 B8	//X-axis follow-up starts
M03 S200	// Spindle starts
G01 U-1 F1	//Grinding feed
G04 X1	//Polishing tool
G00 U10 //Tool retracting	
G50.8 //Cancel X follow-up	
M30	

2.6.10 EccentricCircle Follow-up Function (G51.8 H2)

Need model support, models: 21GD series, 31 series

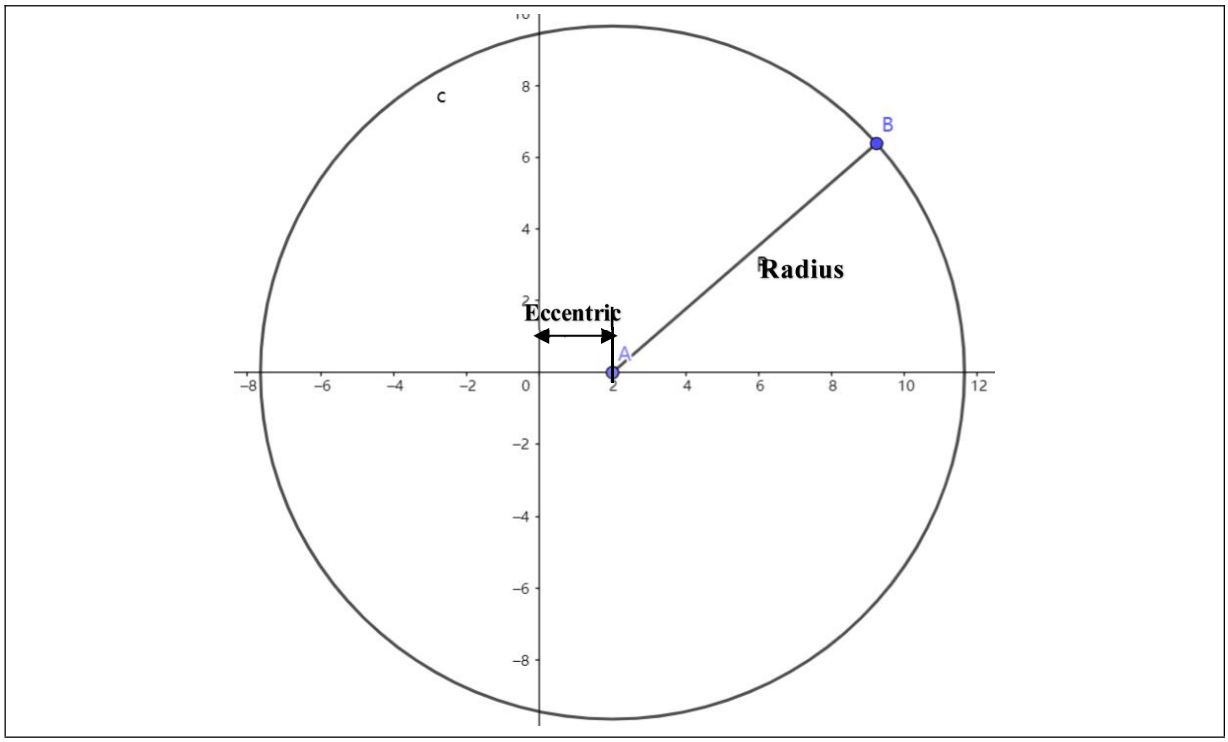
Format

G51.8 H2 D_ P_ R_; Eccentric circle follow-up mode	
D	: Eccentric circle diameter
P	: Eccentric distance
R	: Start angle offset (unsigned), range: 0˚~ 360.000˚

G50.8; Follow-up mode cancellation

Description

This command is an extension of the electronic cam command, and the other axes can be moved freely when the follow-up mode is activated.



Example

Longitudinal eccentric circle turning; the circle diameter is 10 mm and the eccentric distance is 2 mm	
Let the spindle (1st spindle) be the servo-controlled A axis, and X be the slave axis (follow-up axis) Bit 1 of sys-parameter is set to 1, sys-parameter-5350 is set to 1 (spindle A), and sys-parameter-5351 is set to 1 (X axis)	
O1000	
G00 X55	// Safety point positioning
G51.8 H2 D10 P2	//X-axis follow-up starts
M03 S200	// Spindle starts
G01 U-1 F1	//Grinding feed
G04 X1	//Polishing tool
G00 U10 //Tool retracting	
G50.8 //Cancel X follow-up	
M30	

2.6.11 Electronic Gearbox Function (G81.4/G80.4)

Need model support, models: 31 series

Format

G81.4 T_ L_ P_ Q_ R_; Synchronization starts

T : Number of teeth of the workpiece (positive integer), it is 1 when omitted

L : Number of hob heads (integer), it is 1 when omitted, and its sign determines the direction of rotation of the workpiece axis

P : Gear helix angle, range: -90° to 90°

Q : Module of gear, range: 0.1 to 25 mm

R : Phase angle offset of workpiece axis, range: 0 to 359.999°, it will not enter phase synchronization when omitted

G80.4; Cancel synchronization

Description

This function enables synchronized control of the tool axis and the workpiece axis for high-precision machining of gears, threads, etc.

Which axis is the master axis and which axis is the driven axis will vary depending on the machine specifications. Generally in hobbing machines, the tool axis is used as the master axis and the workpiece axis is used as the driven axis.

1. The relationship between the rotational speed of the tool axis and the workpiece axis is as follows:

$$S(Workpiece) = S(Tool) \times \frac{L}{T}$$

L: Number of hobbing heads

T: Number of workpiece teeth

2. the CNC needs to compensate the angle of the workpiece axis when the axial feed axis of the hobbing cutter (generally Z-axis) moves during helical gear machining, forming a differential motion;

$$\text{Compensation angle of tool axis} = \frac{Z \times \sin(P)}{\pi \times T \times Q} * 360°$$

Z: Movement of Z-axis during G81.4 synchronization

P: Gear helix angle specified by G81.4

π: Pi

T: number of gear teeth specified by G81.4

Q: module of gear specified by G81.4

3. G81.4 operates in follow-up mode and synchronous mode as follows:

Follow-up mode	Synchronous mode
Principle: The workpiece axis follows the tool axis speed feedback for position control	**Principle:** Interpolation control of workpiece axis and tool axis
Requirements: (1) Tool axis operates in speed mode or position mode (2) The tool axis must be fed by speed feedback	**Requirements:** (1) The tool axis must operate in position mode (2) The tool axis does not require speed feedback
sys-parameter setting: Bit 1 of sys-parameter-5320 is set to 0 or Bit 1 of sys-parameter-5320 is set to 1 and the tool axis is in speed mode	**sys-parameter setting:** Bit 1 of sys-parameter-5320 is set to 1 and the tool axis is in position mode

4. The tool axis must be a spindle. sys-parameter-5330 is used to set which spindle of the CNC is to be used as the master axis of the G81.4.

5. The workpiece shaft may be either a spindle or a common servo shaft. The control axis of the slave axis is set by sys-parameter-5331, and the slave axis must be specified as a rotating axis (bit 2 of sys-parameter-0001).

6. Axial feed axis of the workpiece (sys-parameter-5332) must be set when machining helical gears, otherwise the CNC will generate an alarm;

7. Matters related to G81.4 execution actions:

(1) If the workpiece shaft is set to the spindle type, the spindle as the workpiece shaft must stop in order to execute G81.4;

(2) G81.4 can be executed to synchronize the tool shaft, whether the tool shaft is at rest or running;

(3) The acceleration and deceleration of the workpiece shaft for position synchronization during the execution of G81.4 is set by sys-parameter-5313;

(4) G81.4 cannot be executed properly if the tool axis speed is too high, resulting in a higher tool axis speed than the set value of sys-parameter-5332;

(5) When G81.4 is executed in the follow-up mode, the CNC will check whether the difference between the speed feedback of the tool axis and the set speed is within the range of the set value of sys-parameter-5336, and generate an alarm if the speed difference is too large and the time exceeds the set value of sys-parameter-5337.

(6) When bit 5 of sys-parameter-5320 is set to 0, the execution of G81.4 will wait for the synchronization of the tool axis and workpiece axis to be completed before the CNC executes the next program segment;

(7) When bit 5 of sys-parameter-5320 is set to 1, the G81.4 command does not wait for the active spindle and the driven axis to complete synchronization, and the CNC immediately executes the next program segment, but when it executes cutting commands such as G01, the CNC will wait for the active spindle and the slave axis to complete synchronization.

8. Matters related to G80.4 execution actions:

(1) When G80.4 is executed, only the workpiece axis is released from the synchronization state; the operating state of the tool axis is not affected by G80.4.

(2) The deceleration time of the workpiece axis for the execution of G80.4 is set by sys-parameter-5333;

(3) If P1 is specified in the program line in which G80.4 is located or P is omitted, the G80.4 command will wait for the completion of the workpiece axis stop before the CNC can execute the next program segment; if P0 is specified in the program line in which G80.4 is located, the G80.4 command will not wait for the completion of the workpiece axis stop and the CNC will execute the next program segment immediately.

9. In addition to changing the direction of rotation of the workpiece axis by means of the plus or minus sign of L in G81.4, it can also be set by bit 2 of sys-parameter-5320;

10. In addition to changing the direction of helix compensation of the workpiece axis by means of the plus or minus sign of Q in G81.4, it can also be set by bit 3 of sys-parameter-5320;

11. When bit 4 of sys-parameter-5320 is set to 0, the synchronization state of the tool axis and the workpiece axis is released after the tool axis stops in case of an emergency stop or reset.

12. When bit 4 of sys-parameter-5320 is set to 1, the tool axis and the workpiece axis are immediately released from the synchronization state during an emergency stop or reset, and the workpiece axis is decelerated to a stop in accordance with the acceleration and deceleration times set by sys-parameter-5333.

13. Phase matching relationship between tool axis and workpiece axis:

Perform phase matching	No phase matching
G81.4 needs to specify R	G81.4 does not specify R
When G81.4 is executed, the workpiece axis will be moved according to the current tool axis coordinates and the T and L values will be specified by G81.4 for the shortest path movement. At this time, the workpiece axis movement is less than or equal to 360*(L / T)	When G81.4 is executed, the tool axis and feed axis do not make any movements.
Phase matching can be performed with the tool axis in rotation or at rest	Can only be used when the tool axis is at rest It needs to perform pre-positioning of the tool axis and feed axis, and then perform G81.4 for synchronization.

2.6.12 Constant Liner Speed Control (G96/G97)

Format

G96 S_ P_ ; (Constant linear speed enabled)

S : Linear speed, unit: m/min

P : Linear axis number, default X axis when omitted (requires software version support 2.80.32+)

G97 S_; (Constant linear speed canceled)

S : The speed to which the spindle returns after the cancellation of constant linear speed

G50 S_; (Maximum spindle speed limit at constant linear speed)

S : Maximum spindle speed in G96 mode, unit: r/min

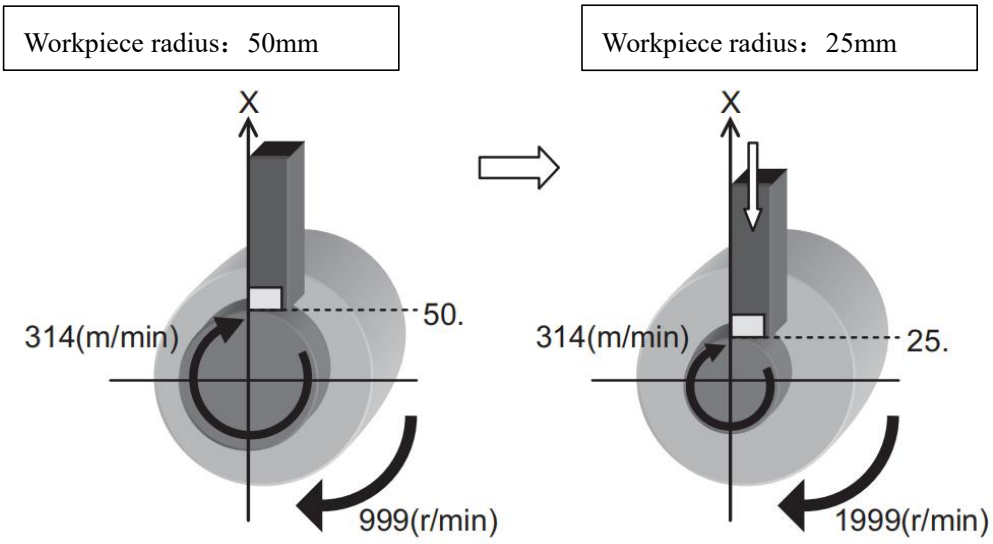
Description

In this function, the spindle speed is adjusted with the movement of the tool nose position (constant surface speed control) so that the speed of the cutting point is always constant (fixed cutting speed).

By using it in the cutting process, etc., it has improved machining time and tool life.

However, when the tool nose moves to the workpiece origin, it may rotate at the maximum speed specified by the machine specification, which is very dangerous. Be sure to set the limit of maximum speed with the spindle speed limit setting command (G50 S_).

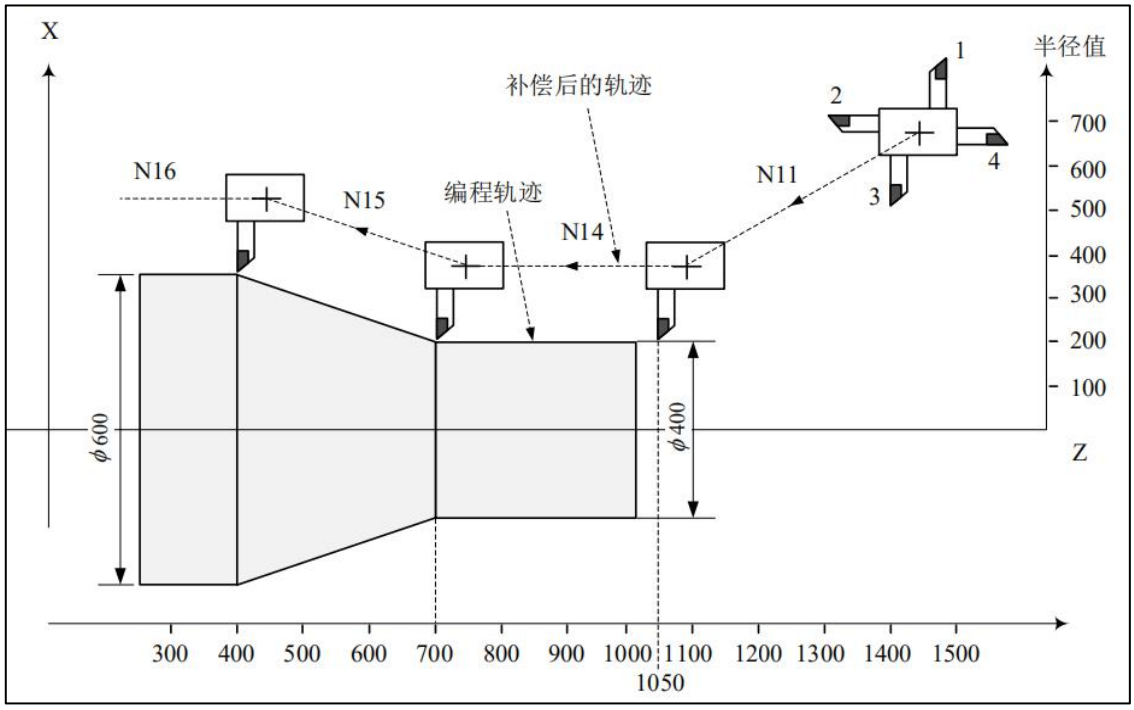
The constant surface speed control specified by the constant surface speed control command G96 S314 (m/min)



Because the surface speed is constant, the spindle speed is calculated as the tool nose moves so that it changes automatically.

In the above example, because the surface speed (314 (m/min)) is constant, the speed changes from 999 (r/min) to 1999 (r/min) as the radius of the workpiece changes (50 mm → 25 mm).

1. In G96 mode, the maximum spindle speed is set by G50 S_ and the minimum spindle speed is set by sys-parameter-1105;
2. In G96 mode, the actual maximum spindle speed is limited by the CNC parameters #1025 to #1028, even if the speed set by G50 S_ is greater than that limited by the CNC parameters, the speed limit of the CNC parameters shall prevail;
3. During resetting, bit 3 of sys-parameter-2202 is used to set whether or not the G96 mode is canceled;
4. In G96 mode, the fast positioning command will also affect the spindle speed variation;



N8 G0 X1000 Z1400;

N9 T0202;

N11 X400 Z1050;

N12 G50 S3000; (set the maximum revolving speed of the spindle as 3000 revolutions)

N13 G96 S250; (set the linear velocity as 250 meter/minute)

N14 G01 Z700 F100;

N15 X600 Z400;

N16 …

…

N28 G97; (cancel constant linear velocity)

…

N42 M30;

The spindle speed calculated by CNC is proportional to the surface speed at the position specified by the coordinate values on the axis X, but it is not calculated after offsetting X coordinate values. In the above example, the speed at the end point of N15,where the diameter is 600 (not the center of the turret but the tool nose), is 200 m/min. If the X axis coordinate values are negative, CNC will use absolute values.

2.6.13 Spindle Synchronization (G114.1/G113)

Format

G114.1 P_Q_K_R_; (Spindle synchronization)

P : Active spindle speed proportional coefficient (positive integer); it is 1 when omitted;

Q : Proportional coefficient of slave spindle (integer); the sign indicates the direction, and it is 1 when omitted.

K : Synchronous axis group number, with a maximum of 3 groups; if omitted, it is group 1;

R : Angle difference between the active spindle and the driven spindle, range: 0° to 359.000°; it is 0 when omitted;

G113; (Cancel spindle synchronization)

Description

If the machine tool has more than two spindles, it can use the two spindles together to complete some special applications. For example, the two spindles do the material receiving action, at this time, it is necessary to synchronize the rotating speeds of the two spindles and maintain the same angular phase or a fixed deflection.

Detailed description

1. sys-parameter-1420 sets the spindle number of the driving shaft and slave axis of each group;
2. The driving shaft and slave axis must be able to execute the position mode;
3. The driving spindle can be the encoder feedback spindle or the position spindle, which is specified by the CNC parameter-1400.
4. As far as possible, the driving spindle and the driven spindle with the same configuration should be adopted. In this way, the synchronization performance will be high.
5. The encoder synchronization mode causes lag of the driven spindle, so it is not recommended to use this mode to process square workpieces simultaneously.
6. When encoder synchronization is selected, please confirm whether the change trend of spindle angle in system diagnosis number 0214 is consistent with the actual rotation direction of spindle.
7. When the spindle is synchronized, the motor torque load limit of the driven spindle can be set to improve the coordination performance of spindle synchronization;
8. When the CNC is reset, the synchronization status of the spindle will not be released, and the synchronization must be released through command G113.

Example

Assume: the left side of the workpiece is channel 1, the right side of the workpiece is channel 2	
Channel 1	**Channel 2**
O1000	O1001
CH1	CH2
T0001	T0002
G00 X100 Z0	G04.1 P1 Q12
M03 S500 (rotation of spindle in channel 1)	G00 X0 Z0 (spindle 2 is positioned at the workpiece)
... (workpiece processing program)	M11 (Release the spindle in channel 2)
G144.1 (synchronize the slave axis, the spindle in channel 2 will start to turn)	G01 Z-5 F100 (ready for receiving)
T0002	G04.1 P2 Q12
G0 G0 X0 Z0 (located to the position to coordinate with spindle)	M10 (clamp the spindle in channel 2)
G04.1 P1 Q12 (channel synchronization)	G04.1 P3 Q12 (channel synchronization)
G04.1 P2 Q12 (channel synchronization)	T0001
M11 (Release channel 1 chuck)	G04.1 P4 Q12 (channel synchronization)
G04.1 P3 Q12(channel synchronization)	M03 S1000
G113 (cancel the spindle synchronization to make the spindle of channel 2 is out of the control of channel 1)	G0 X100 Z0
G04.1 P4 Q12(channel synchronization)	...(Machine the right end of the workpiece)
G0 X100 Z0	M30
M30	

2.6.14 Spindle Speed Floating Function (SPSV)

Format

SPSV[t, sv];

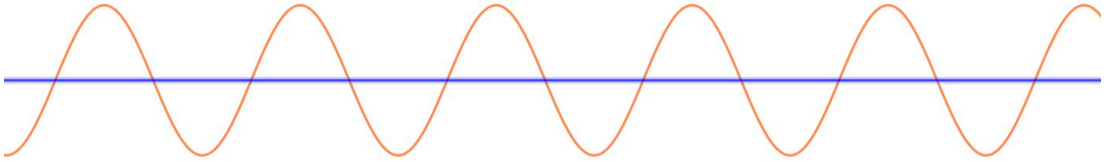
t	: Speed floating period, unit: second
s	: amount of spindle speed floating, unit: rpm

Description

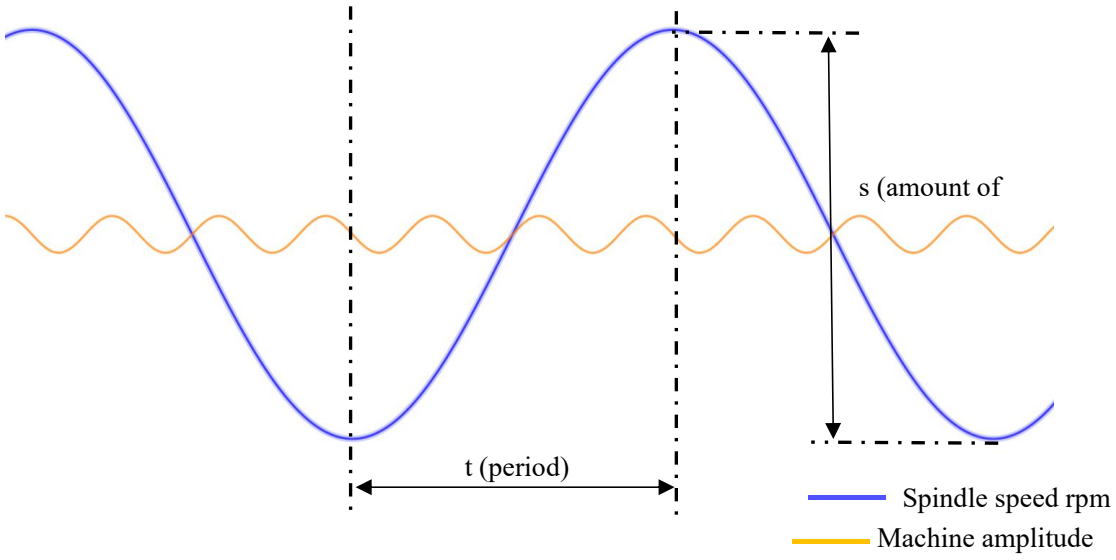
The command can continuously increase and decrease the spindle speed, thus significantly reducing the impact of tool shaking (e.g., improving the finish) and improving the chip entanglement.

It is a modal command. It cancels the speed floating when t or s is specified as 0, and the speed floating is automatically canceled after reset.

Spindle speed curve and machine amplitude curve without SPSV:



Spindle speed curve and machine amplitude curve with SPSV:



Example

O1000	
M03 S1000	//Set spindle speed to 1,000 r/min
SPSV [2, 200]	//After starting the speed floating function, the speed change period is 2 seconds, and the amount of speed floating is 200 rpm //At this time, the spindle speed is 800~1200
G00 X100	
G00 Z0	
...	//Machining program
M30	

2.7 TOOL FUNCTIONS(CHANGE / COMPENSATION)

2.7.1 T Function

Format

T AA BB	
AA	: Tool number, corresponding to the value of PLC's F0026 register address
BB	: Tool compensation number, range: 1 ~ 64

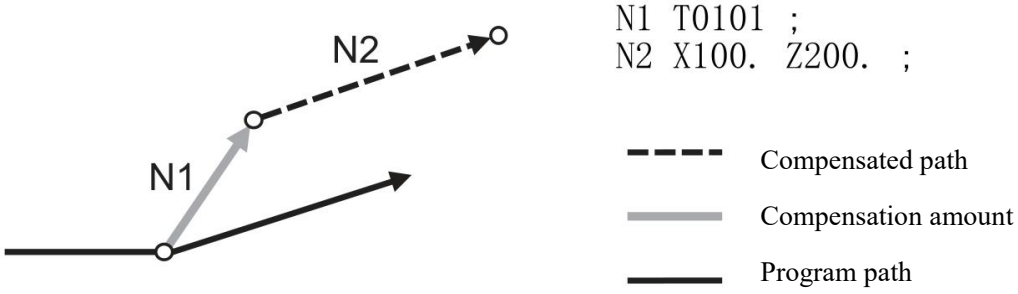
Description

The T function, also known as the tool function, specifies the tool number and the tool compensation number. The T command used varies from machine to machine, so please refer to the manual issued by the machine manufacturer.

1. One group of T commands can be specified in a program segment;
2. When the T function is specified at the same time as the move command program segment is specified, by default both are executed at the same time, and the PLC can be programmed to specify that the T tool change action is executed during the movement or after the movement is completed;
3. The type of tool post / tool magazine is specified by sys-parameter-1510;
4. The number of tools is set by sys-parameter 1511 for limiting the value of AA in TAA BB;
5. Only the standard PLC logic for electric tool post is built into the CNC (sys-parameter-1510 = 1), and other tool posts/magazines require additional PLC programming;
6. Built-in PLC parameters related to electric tool posts:

Parameter number	Parameter content
1500 Bit 1	After the tool change is completed, the tool number is not detected / detected
1500 Bit 5	Tool number error real-time detection, on/off
Bit 1 to bit 8 of 1501	
Bit 1 to bit 8 of 1502	
1520~1553	Electric tool post related settings

7. When TAA BB (BB is the compensation number) is executed, the program coordinates are compensated:



8. Bit 2 of sys-parameter 1500 is used to set whether the value of BB in TAA BB is allowed to be 0 or not;
9. When the movement command after the T command is programmed relatively, bit 3 of the CNC parameter-1500 is used to specify that the movement coordinate is relative to the coordinates before the T command is executed or after the T command is executed.

2.7.2 Programmable Tool Offset / Wear Input (G10)

Format

G10 P_ α/β_ R/C_ Q_;	
P	: Tool offset / wear serial number 1~64: Set tool wear data 10000+(1~64): Set tool offset data
α	: set the absolute value of the tool compensation data for the axis, and multiple values can be specified.
β	: set the incremental value of the tool compensation data for the axis, and multiple values can be specified.
R	: set the absolute value of tool nose radius
C	: set the incremental value of tool nose radius
Q	: set the tool nose phase value

Description

The tool offset/wear amount can be set and changed according to the G10 command. When the command is executed with absolute values (X,Y,Z), the compensation amount becomes the new amount; and when the command is executed with incremental values (U,V,W), the compensation amount becomes the new amount by adding the specified compensation amount to the currently set compensation amount.

Example

1. The compensation of No.1 tool compensation is realized by a program	
G10 P1 U0.01	// X-axis wear value for No. 1 tool wear + 0.01

2. Set the absolute coordinates at the time of No. 1 tool compensation as the specified values	
#1=100	//Set the preset value of the current absolute coordinate when G10 is executed
T0001	//Switch No. 1 tool compensation
G50.3 X0	//Restore the origin of the current absolute coordinates
G4I0	//Prevent CNC pre-reading to ensure that the next line of coordinate values are read correctly
#2=#5021	//Read the machine coordinate value of the 1st axis (X-axis) in the current channel
G10 P1 X0	//Clear data for No.1 tool compensation wear
G10 P10001 X[#2 - #1]	//Write the calculated tool offset data into the tool offset of No. 1 tool, so that the absolute X-axis coordinates at T0001 become the value set by #1

2.7.3 Tool Nose Radius Compensation (G40/G41/G42)

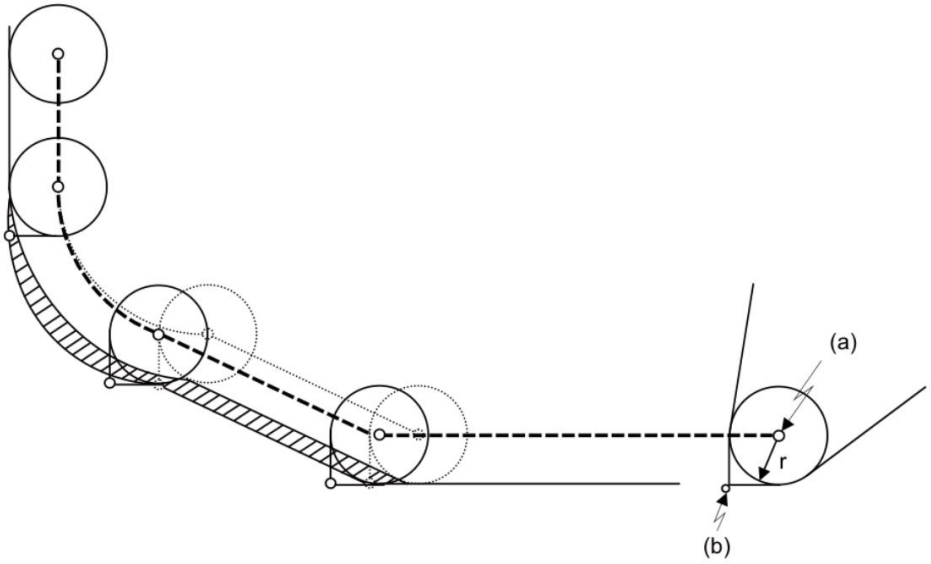
Format

G40 X/U_ Z/W_ α/β_; Cancel tool nose R compensation
G41 X/U_ Z/W_ α/β_; Establish tool nose R compensation Left
G42 X/U_ Z/W_ α/β_; Establish tool nose R compensation Right

X/U : X-axis end point coordinates (X is absolute programming, and U is incremental programming)
Z/W : Z-axis end point coordinates (Z is absolute programming, and W is incremental programming)
α/β : Additional axis end point coordinates
(α is used for absolute programming, and β is used for incremental programming); and multiple additional axes can be specified.

Description

Since the tool nose generally has roundness, the virtual nose point is considered as the tip of the tool for the program execution. Thus, in taper cutting and arc cutting, the error is caused by the roundness of the tool nose between the programmed shape and the cutting shape. Tool nose R compensation is a function that automatically calculates and makes the computation by setting the tool nose R value.



(a) Tool nose center (b) Virtual tool nose (r) Tool nose radius

Compensation，tool nose center path

No compensation, program path (The shadow is the over cut part)

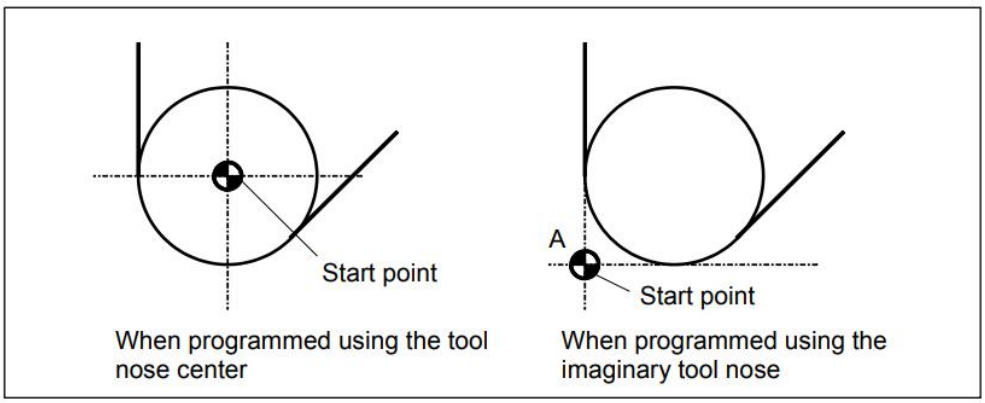
Imaginary Tool Nose

The tool nose at position A in Fig does not actually exist.

The imaginary tool nose is required because it is usually more difficult to set the actual tool nose radiuscenter to the start point than the imaginary tool nose.

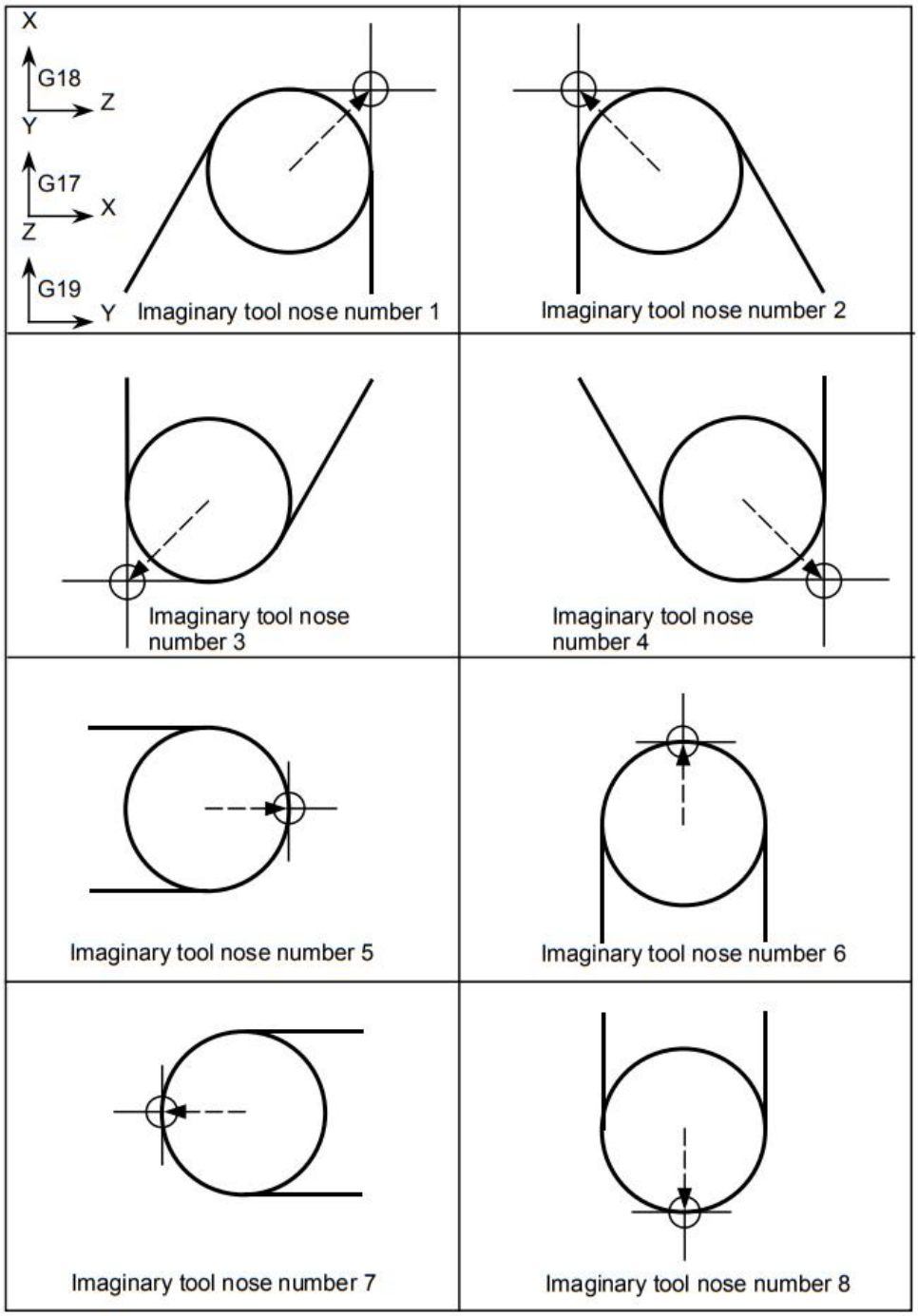
Also when imaginary tool nose is used, the tool nose radius need not be considered in programming.

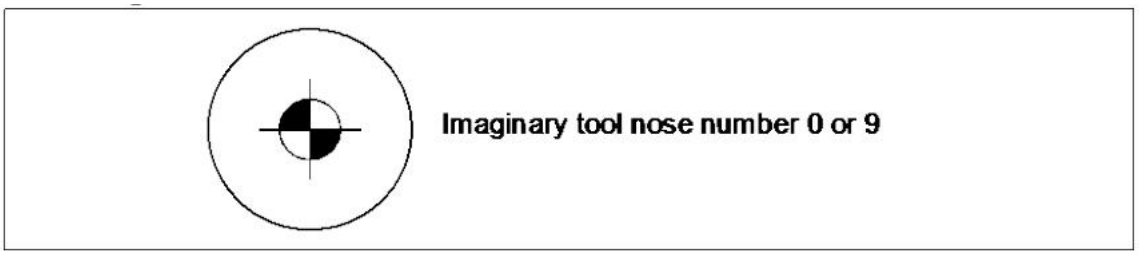
The position relationship when the tool is set to the start point is shown in Fig.



2.8.1.1. Tool nose point and compensation direction

The direction of the imaginary tool nose viewed from the tool nose center is determined by the direction of the tool during cutting, so it must be set in advance as well as offset values.The direction of the imaginary tool nose can be selected from the eight specifications shown in the Fig below together with their corresponding codes. Fig illustrates the relation between the tool and the start point. The following apply when the tool geometry offset and tool wear offset option are selected.





2.8.1.2. Establishment of tool nose R compensation

The tool nose R compensation starts when all of the following conditions are met with the compensation canceled.

(1) Movement command after execution of G41 and G42 commands

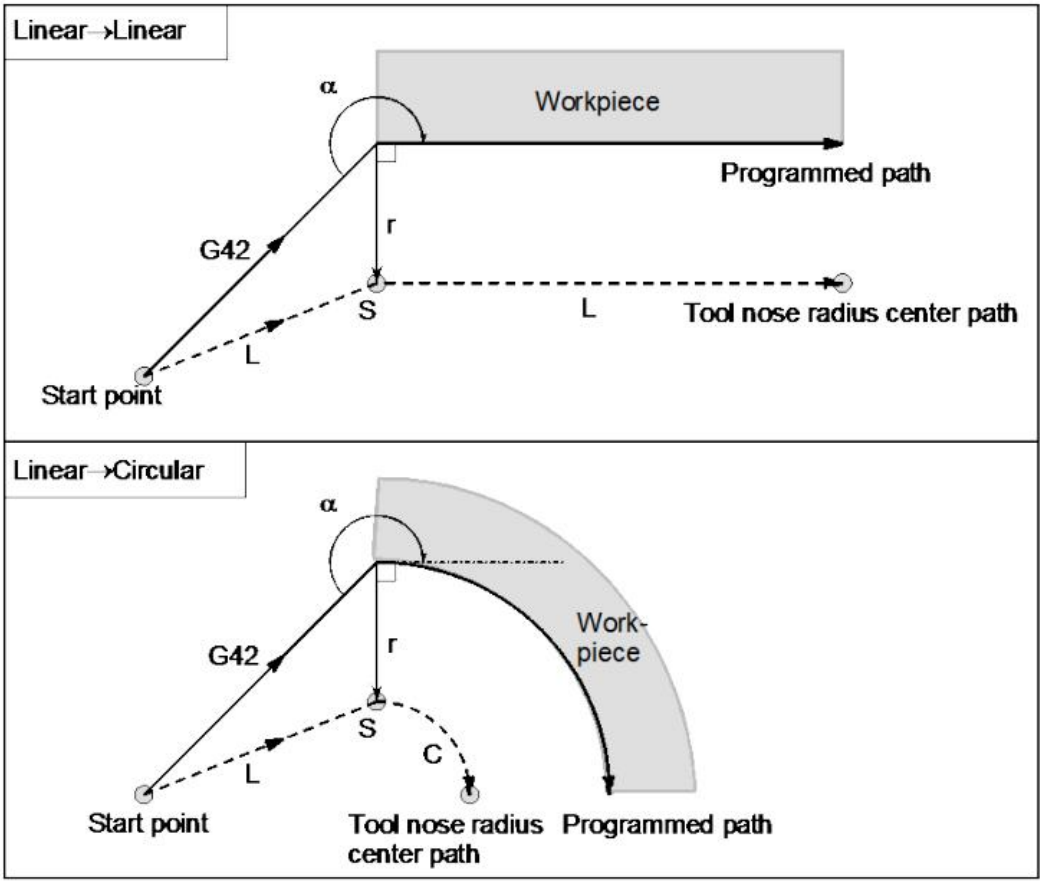
(2) Movement commands other than arc commands

At the beginning of compensation, regardless of continuous run or single segment run, 2 to 5 program segments are to be read in succession before execution in order to perform the intersection calculation. (Two program segments are pre-read if there is a movement command, or up to 5 program segments are pre-read if there is no move command.)

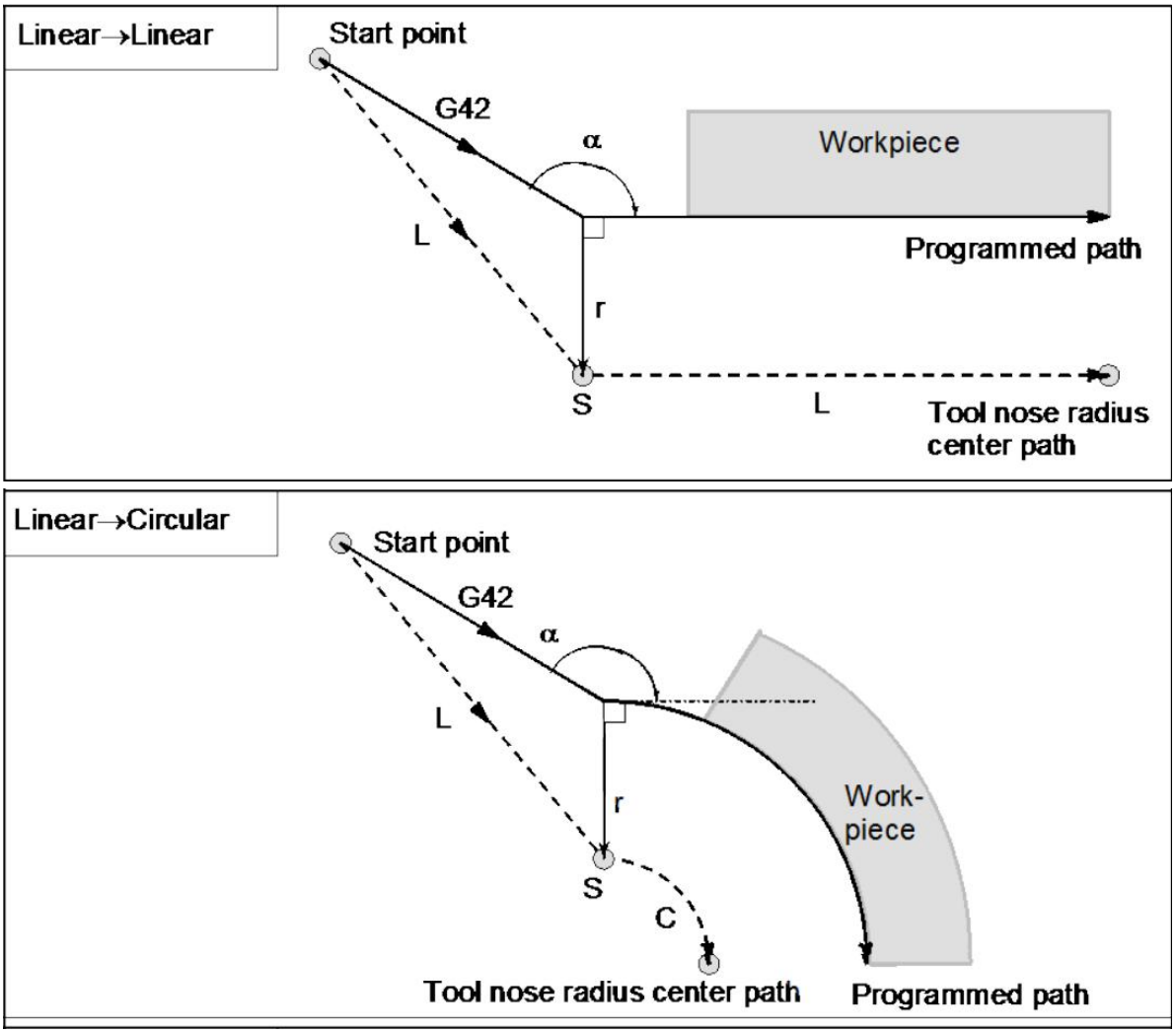
In addition, up to 5 program segments are also pre-read in the compensation mode before performing the compensation operation.

The G41/G42 commands cannot move the tool nose R compensation amount alone when starting the tool nose R compensation. The G00 command does not perform tool nose R compensation. The tool nose R compensation is performed by G01, G02, and G03 commands. However, even with an axis command, if there is no movement, no tool nose R compensation will be performed.

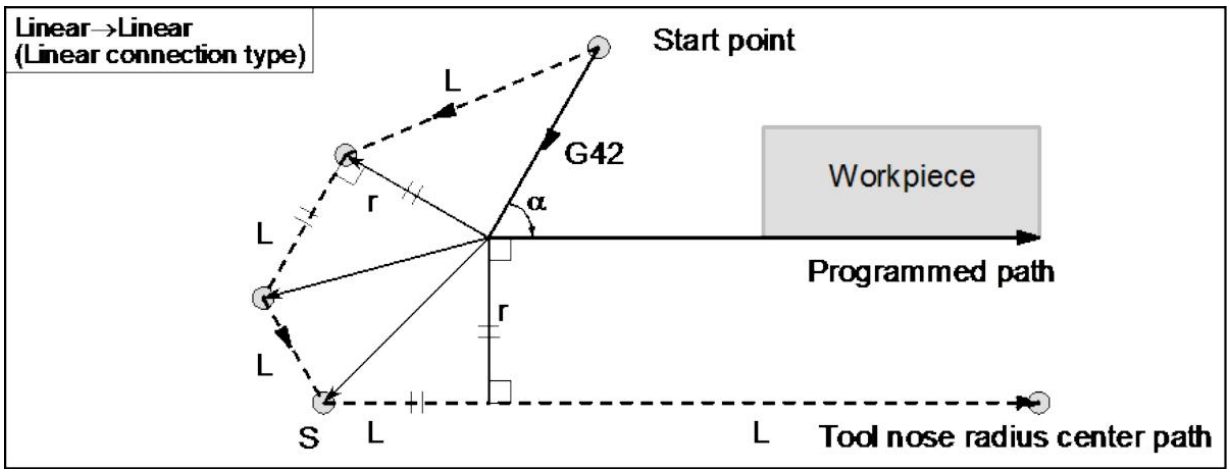
(1) When G41/G42 executes the command on the inside of the corner(180° <= α)

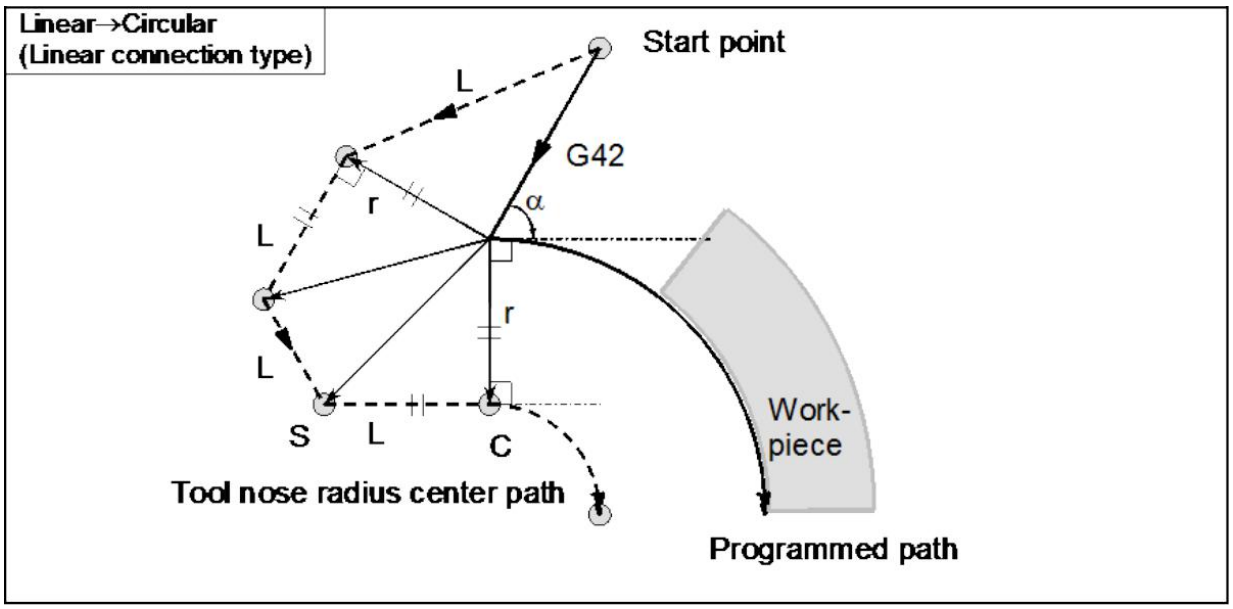


(2) On the outside of the corner (obtuse angle) [90° ≦ θ < 180°].



(3) On the outside of the corner (acute angle) [θ < 90°].

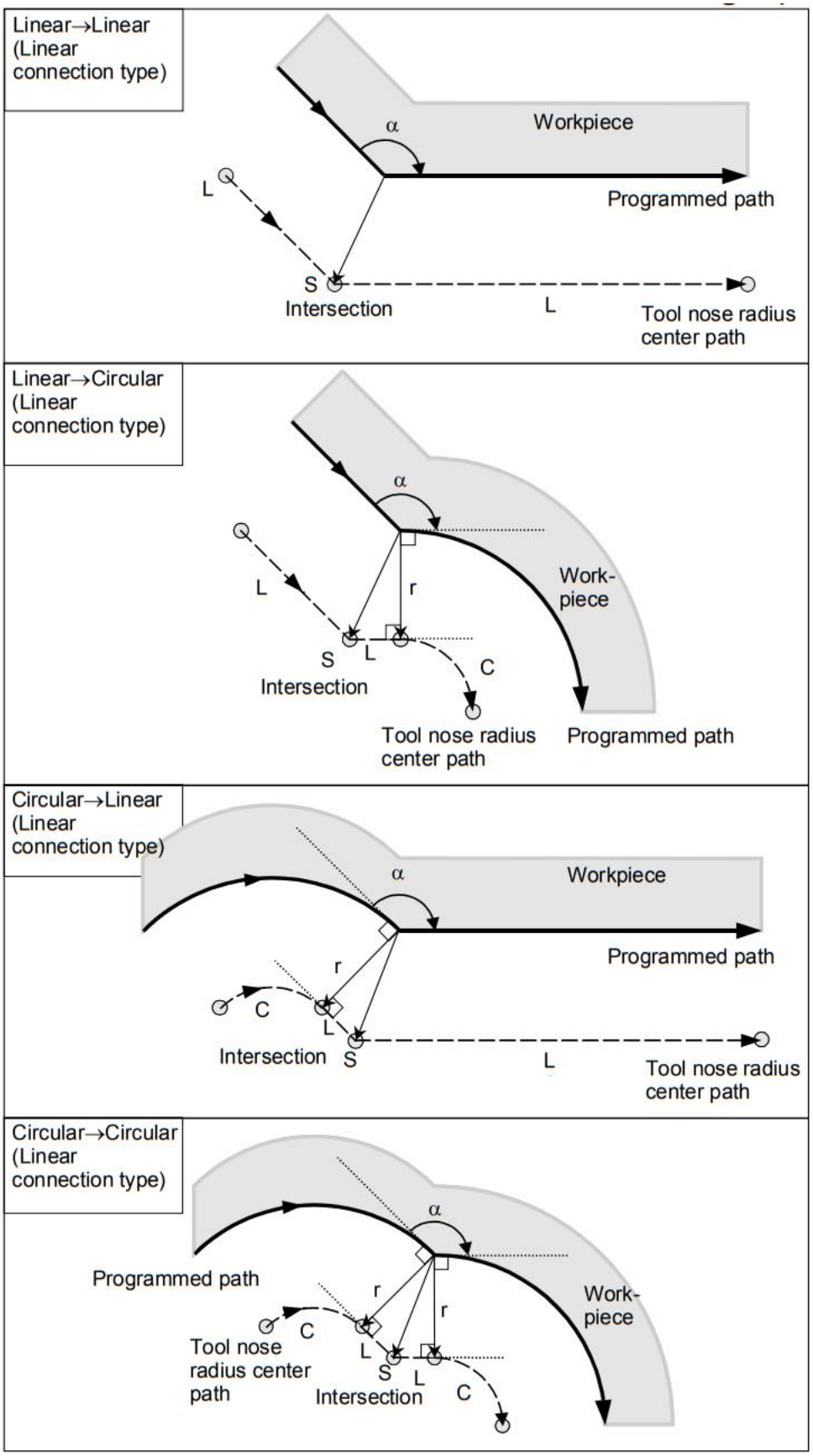




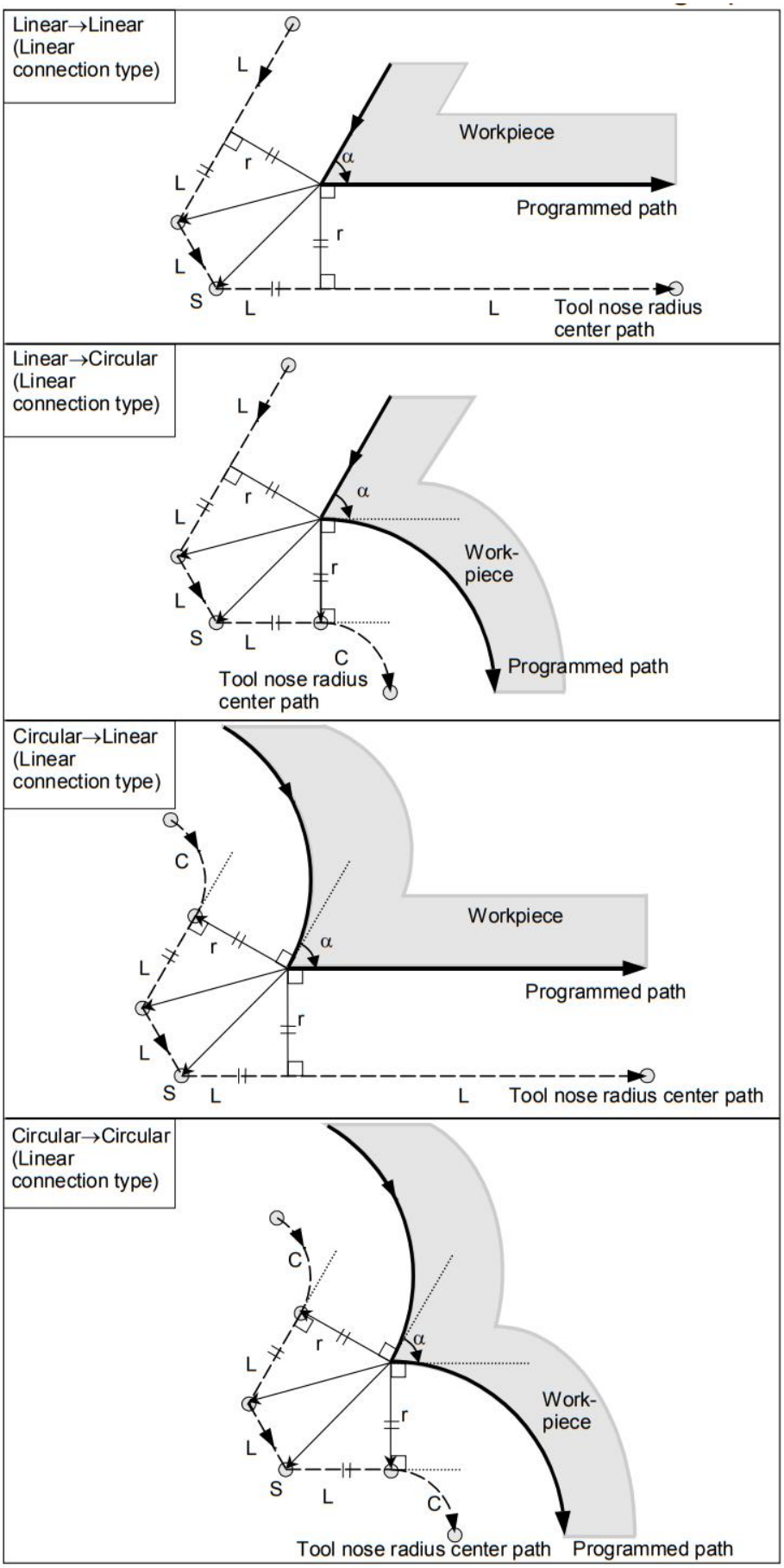
2.8.1.3. Action in tool nose R compensation mode

The tool center path is obtained from the line/arc to compensate the program path (G00, G01, G02, G03).

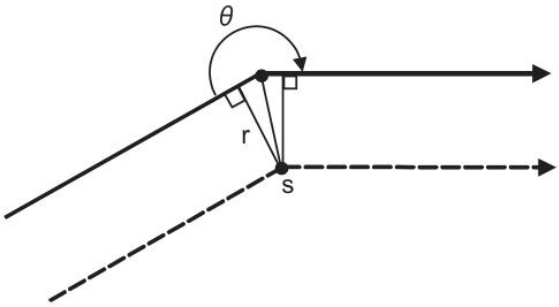
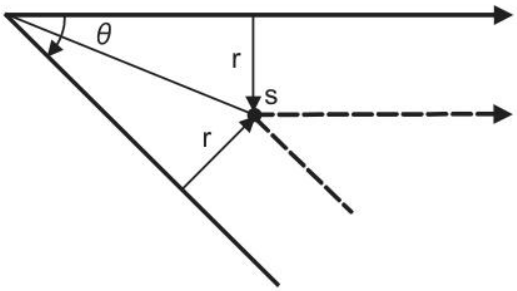
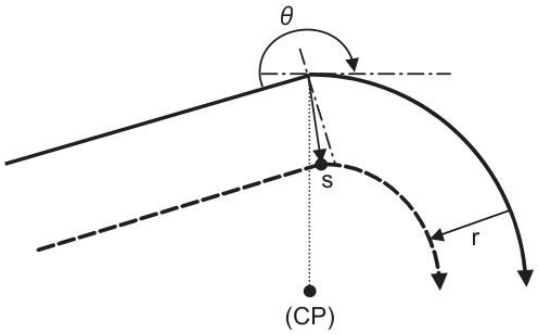
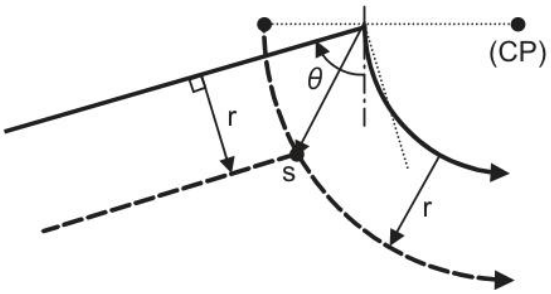
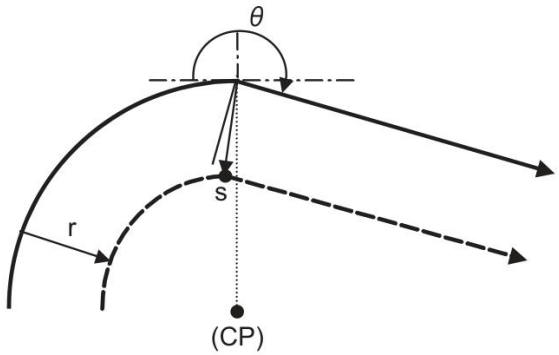
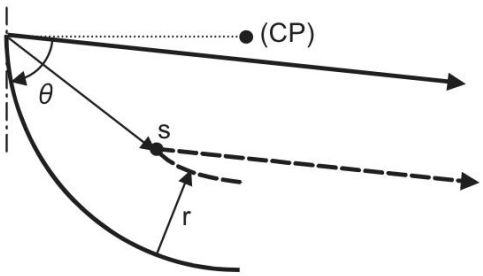
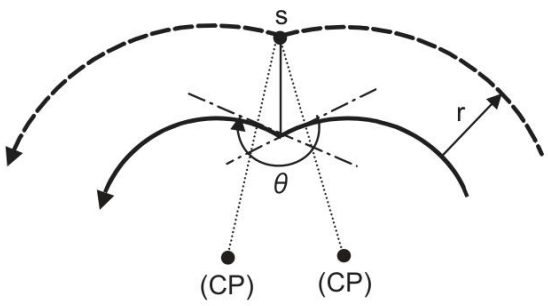
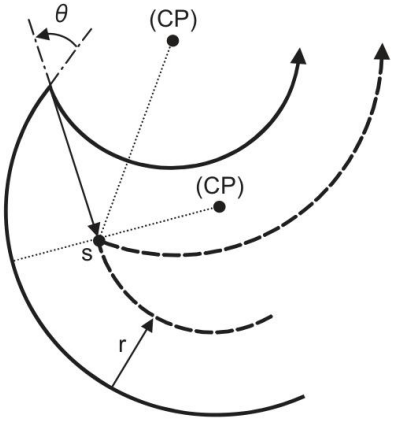
（1） When rotating the outer side of the corner**(90° ≦ θ ＜ 180°)**



（2） When rotating the outer side of the corner**(θ ＜ 90°)**

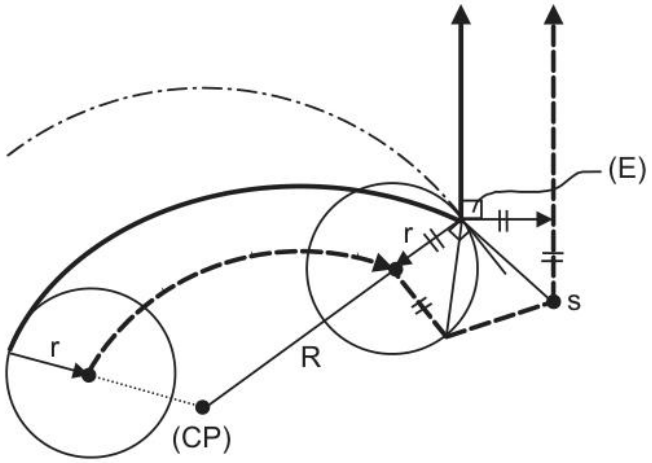


（3） When rotating the inner side of the corner

Straight line → straight line (obtuse angle)	**Straight line → straight line (obtuse angle)**
	
Straight line → arc (obtuse angle)	**Straight line → arc (obtuse angle)**
	
Arc → straight line (obtuse angle)	**Arc → straight line (obtuse angle)**
	
Arc → Arc (90° ≦ θ ＜ 180°)	**Arc → Arc (0° < θ < 90°)**
	

(3) When there is no arc end point on the circle

When the error after compensation is within the set value of sys-parameter-2210, the vortex-like arc interpolation will be performed from the start point to the end point of the arc.



2.8.1.4. Cancel tool nose R compensation

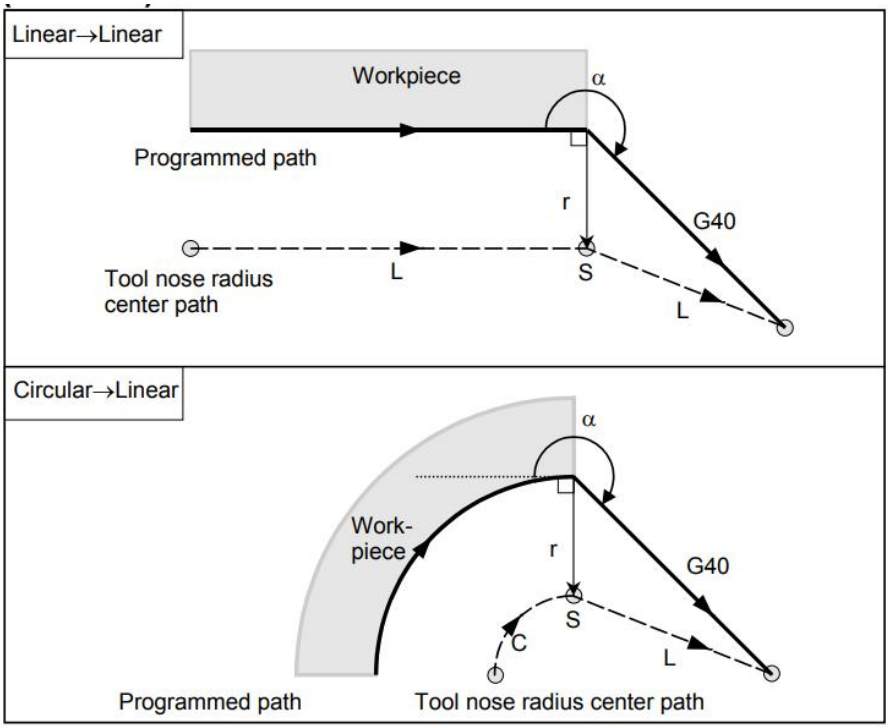
The tool nose R compensation is canceled under any of the following conditions:

(1) After power-on
(2) Press Reset or Emergency Stop
(3) After execution of M02 and M30 that contain the reset function
(4) After execution of the compensation cancellation command G40
(5) Execution of T command

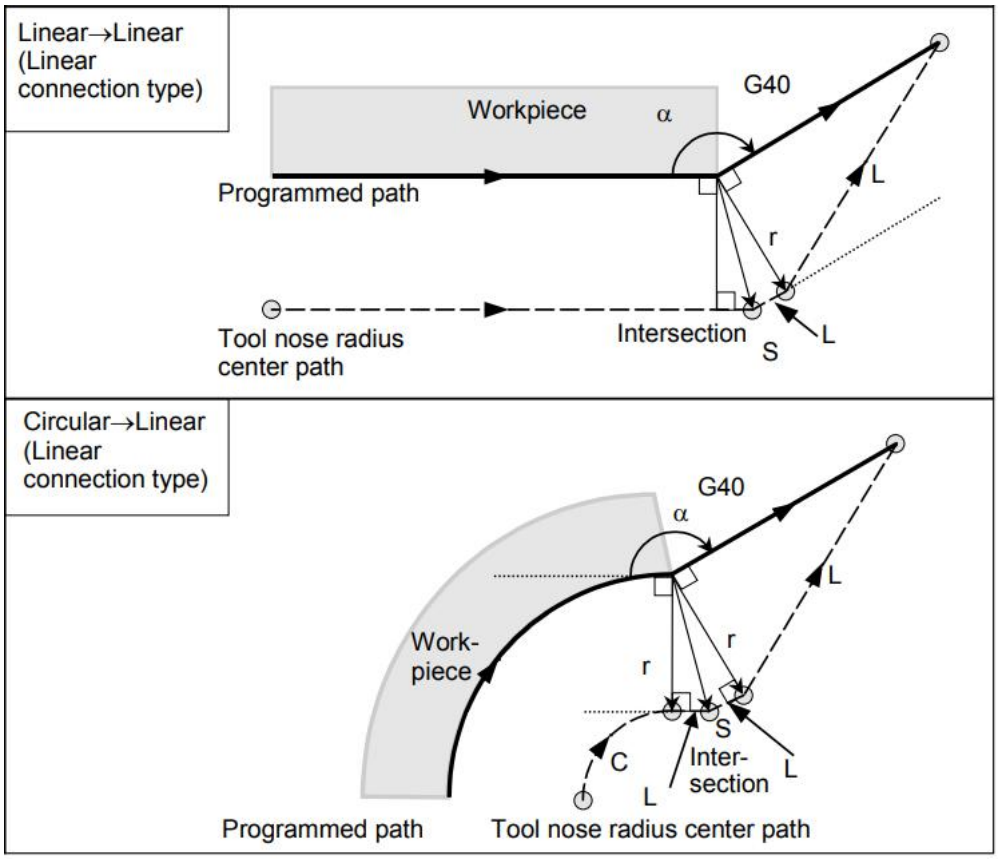
In the compensation cancellation mode, the compensation vector is 0 and the tool nose point path is the same as the program path.

Be sure to terminate the program that contains tool nose R compensation in the compensation cancellation mode.

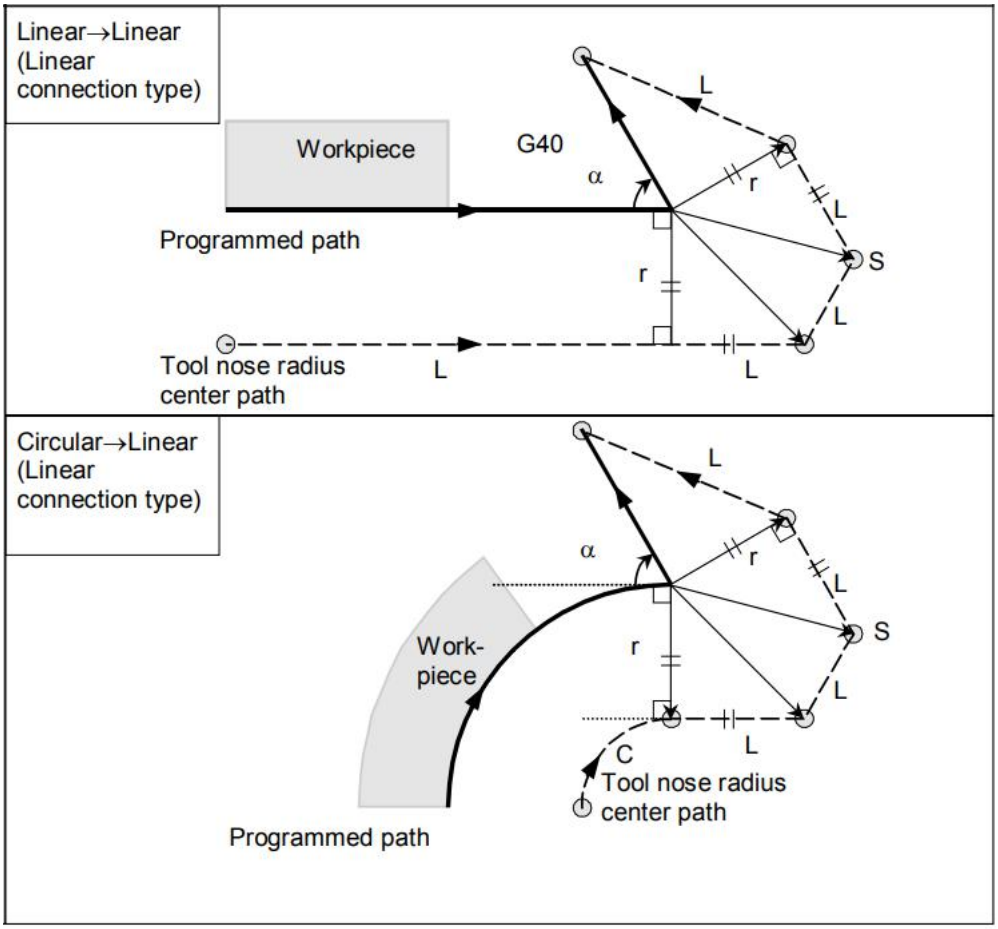
(1) Canceled on the inside of the corner（**180° <= α**）



(2) On the outside of the corner (obtuse angle)（**0° <= α < 180°** ）



(3) On the outside of the corner (acute angle)（ **α < 90°** ）



2.7.4 Normal control (G40.1/G41.1/G42.1)

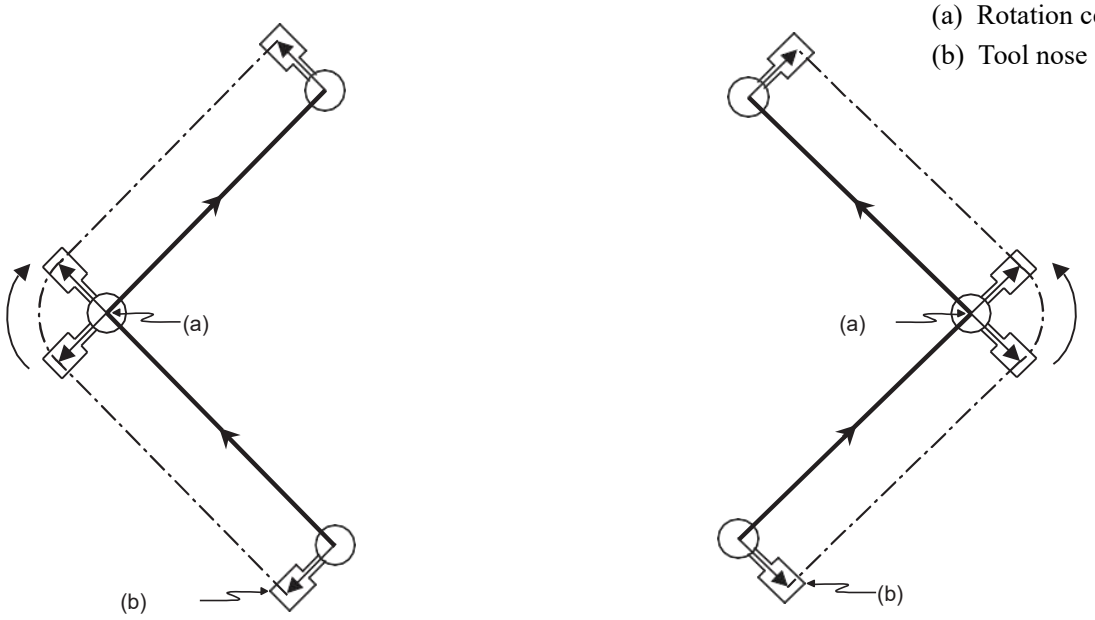
Format

G41.1; Enable normal control (left)
G42.1; Enable normal control (right)
G40.1; Cancel normal control

Description

This function is to control the rotation of the specified rotating axis relative to the axis movement in the plane selection during the program operation, so that the tool always faces the normal direction. Rotate the specified rotating axis at the joint of the program segment, so that the tool will face the normal direction at the starting point of the next program segment.

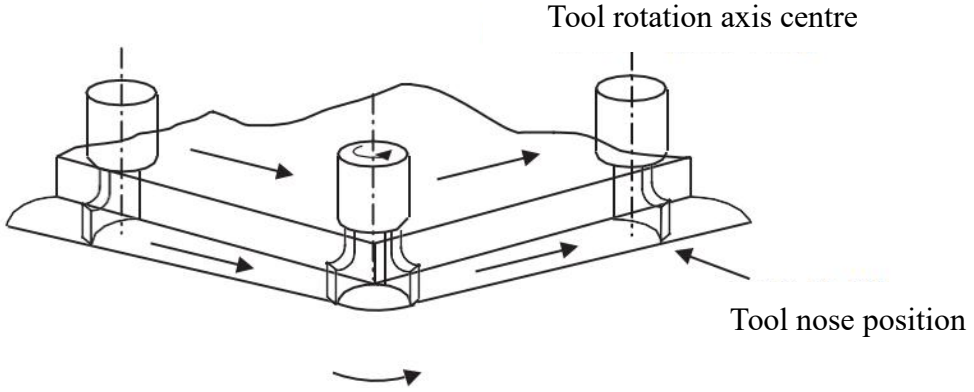
It is necessary to specify the rotating axis for normal control by sys-parameter-5550 and the follow-up speed of the rotating axis for normal control by sys-parameter-5551.



G41.1 normal control (left)

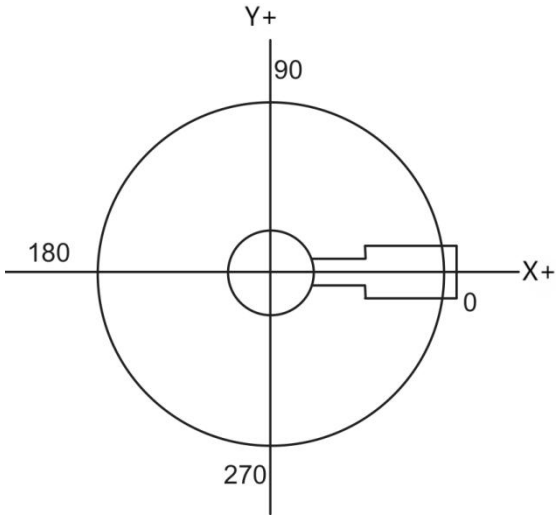
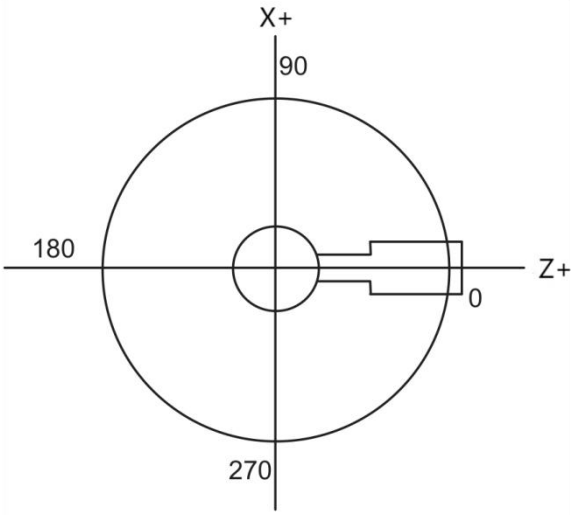
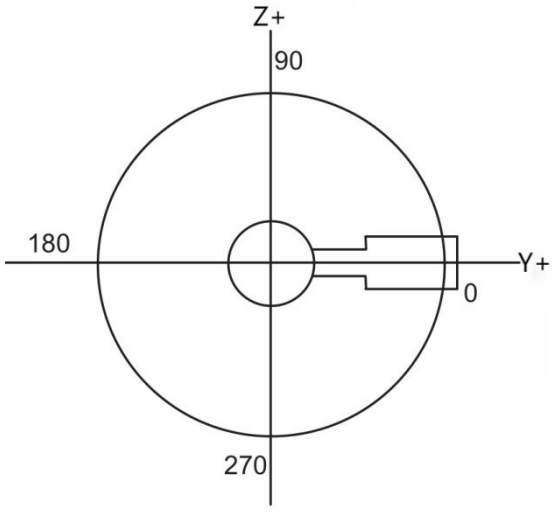
G42.1 normal control (left)

In the state of circular interpolation, the rotation of the rotating shaft is controlled to synchronize with the motion of circular interpolation.



Define the normal control angle

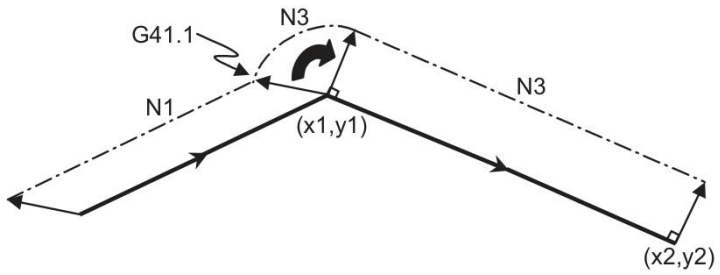
In the plane where normal control is executed, the axis movement direction selected in the plane is normally controlled. The angle of rotating axis for normal control takes the positive direction of the tool towards the first axis of the plane as 0°, the counterclockwise rotation as positive, and the clockwise rotation as negative.

G17 plane (XOY) It is 0° when the rotating axis rotates in the positive direction of the X-axis	
G18 plane (ZOX) It is 0° when the rotating axis rotates in the positive direction of the Z-axis	
G19 plane (YOZ) It is 0° when the rotating axis rotates in the positive direction of the Y-axis	

Rotation action of normal control corresponding to movement command

(1) Start

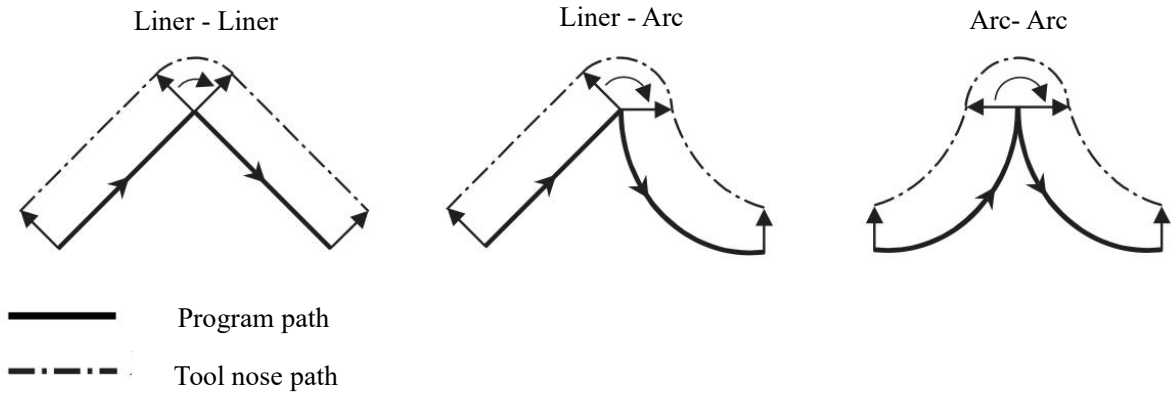
After the normal control axis is rotated to be perpendicular to the direction of travel, at the start of the normal control command segment, the axis for the plane selection starts moving, but the direction of rotation of the normal control axis at start-up follows the direction below 180° (shortcut direction) in the normal control type.

Let the rotating axis be C-axis  After executing G41.1/G42.2, the NC will make a vector judgment based on the movement command of the next segment and will automatically insert a movement command of the rotating axis.	.. N1 G01 Xx1 Yy1 Ff1 N2 G41.1//There will be no axis movement N3 G01 C [Automatic calculation] F [sys-parameter-5551] N3 G01 Xx2 Yy2 ..

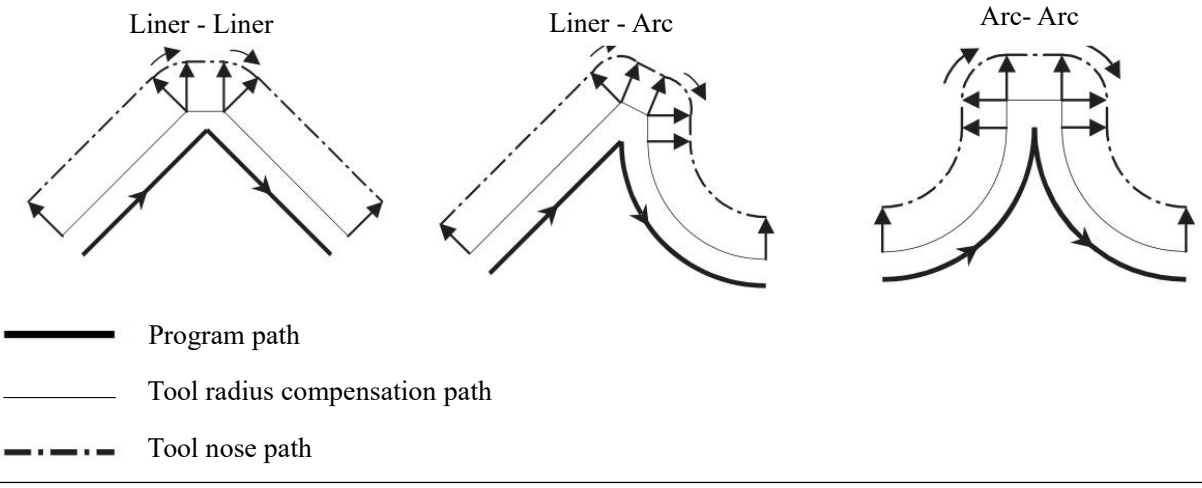
(2) In normal control mode

1. The connection of program segments

Without tool radius compensation, the normal control axis for rotation makes the movement direction selected by the next program segment perpendicular, and the movement direction is subject to the program segment.



After having the tool radius compensation executed, normal control is executed along the path of the tool diameter compensation.



2. In program segment movement
The angle of the normal control axis will remain the same and the normal control axis will not rotate when a straight line command is given. The normal control axis rotates synchronously with the action of circular interpolation when an arc command is given.

(3) Normal control mode cancellation

After G40.1 is executed, the subsequent movement command does not judge the rotating axis, and the rotating axis angle remains the same as before.

Example

Let the tool rotating axis be A axis (A axis is the 4th axis in the channel), and the CNC parameter-5550 is set to 3	
O1000	
G00 X0 Z0	//Positioning
G41.1	// Normal control (left) starts
G01 Z10 F300	//Rotate A to 90° first, then execute G01 Z10 F300
G01 X10 F300	//Rotate A to 0° first, then execute G01 X10 F300
G40.1	// Normal control cancellation
M30	

2.8 COORDINAT SYSTEM FUNCTIONS

2.8.1 Programmable Workpiece Coordinate Offset Input (G10 L2/L20)

Command format ①:

G10 L2 P_ α/β_; G54~G59 coordinate system offset input

P : Coordinate system offset number

0: Set the offset of EXT coordinate system

1~6: Set the offset of G54~G55 coordinate system

7~105: Set the offset of G54P1~G54P99 extended coordinate system

α : set absolute coordinate data and define multiple axis data.

β : set incremental coordinate data and define multiple axis data.

Command format ②:

G10 L20 P_ α/β_; Extended coordinate system offset input

P : Extended coordinate system offset number, P1~P99 correspond to extended coordinate G54P1~G54P99

α : set the absolute value of the coordinate system axis data, and multiple values can be specified.

β : set the incremental value of the coordinate system axis data, and multiple values can be specified.

Description

The G10 L2/L20 command allows you to set and change the workpiece coordinate system data such as G54. When the command is executed with absolute values (X,Y,Z), the compensation amount becomes the new amount; and when the command is executed with incremental values (U,V,W), the compensation amount becomes the new amount by adding the specified compensation amount to the currently set compensation amount.

Example

1. The program to achieve X-axis's coordinate system offset value +0.01 in G54 mode	
G10 L2 P1 U0.01	

2. Set the absolute coordinate X-axis under the current G55 coordinate to 0	
G55	//Switch G55 coordinate system
G10 L2 P2 X0	//Clear the X-axis offset value of the G55 coordinate system to zero
G4I0	//Prevent system pre-reading to ensure that the absolute coordinate value of the next line is read correctly
#2=#5001	//Read the absolute coordinate value of the 1st axis (X-axis) in the current channel
G10 L2 P2 U[#2]	//Set the G55 coordinate system offset value to change the absolute coordinate to 0

2.8.2 Plane selection (G17/G18/G19)

Format

G17 Xp_ Yp_; Select XY plane
G18 Zp_ Xp_; Select ZX plane
G19 Yp_ Zp_; Select YZ plane

Xp	: indicates axis X or the parallel axis of axis X;
Yp	: indicates axis Y or the parallel axis of axis Y;
Zp	: indicates axis Z or the parallel axis of axis Z;

Description

The G-code is used to select the plane for circular interpolation, tool diameter compensation, coordinate rotation, and the plane for drilling.

(1) Xp, Yp, and Zp are determined by the axis addresses in the program section containing G17, G18 or G19;

(2) If the address of one axis or two axes is omitted in the program section G17, G18 or G19, the omitted axis is considered as the basic axis, namely axis X, axis Y or axis Z;

(3) Set the coordinate axis of the axis in the basic coordinate system through sys-parameter-0016 to specify each axis as the basic axis or parallel axis of the coordinate system. If the parameter specifies the parallel axis incorrectly, the default axis in this plane will be automatically defaulted.

(4) After power-on, the lathe defaults to G18 plane, and the milling machine defaults to G17 plane.

Example

Let the A-axis in sys-parameter 0016 be 7, indicating that the A-axis is the parallel axis of the Z-axis in the basic coordinate system	
G00 X0 A0	// G00 to locate to X0 and A0.
G18 G02 X20 A10 R10	// Select ZX plane, axis Z is replaced by axis A, // Perform circular interpolation in ZA plane
G00 X0 A0	//G00 to locate to X0 and A0
G18 G02 X20 Z10 R10	// Select ZX plane, axis Z is recover, // Perform circular interpolation in ZX plane

2.8.3 Automatic Return o Reference Position (G28)

Format

G28 X/U_ Z/W_ α/β_;

X/U : X-axis coordinates of the intermediate point
(X is absolute programming, and U is incremental programming)

Z/W : Z-axis coordinates of intermediate point
(Z is absolute programming, and W is incremental programming)

α/β : Additional axis coordinates of intermediate point
(α is used for absolute programming, and β is used for incremental programming);

Making use of the above command, it can make the command axis return to the reference point. X/U__ Z/W__ specifies the center point of the path returning to the reference point through absolute value command or increment value command.

(1) the CNC parameter-0302 enables the zero returning function of each axis;

(2) It locates from current position to the center point of the command axis (point A - point B) at quick zero return speed set by the CNC parameter-0311. See Figure 2-11.

(3) Performing the machine zero returning action (point B - point R);

(4) When the zero returning action is executed, the NC will clear the G50 and the coordinate offset resulting from the machine clearing operation;

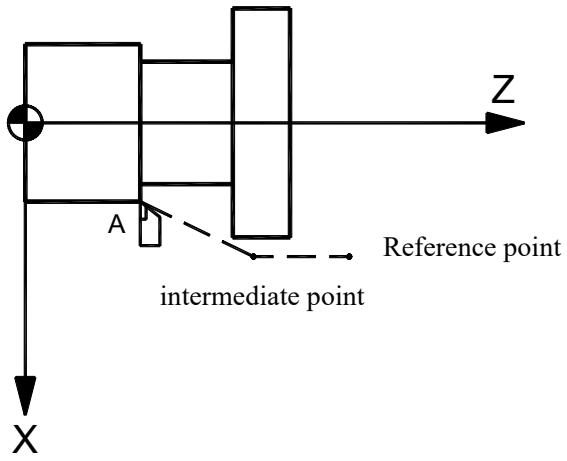


Figure 2-11

Example

```
O1000
T0101
G0 X50 Z3
... //Machining program
G28 X50 Z100    //Execute G0 X50 Z100 inside the CNC first, and then return to the zero point of axis X and axis Z of the machine tool.
T0202
M30
```

2.8.4 Automatic Return o Reference Position (G30)

Format

G30 P2 X/U_ Z/W_; Return to the 2nd reference point
G30 P3 X/U_ Z/W_; Return to the 3rd reference point
G30 P4 X/U_ Z/W_; Return to the 4th reference point

X	: the absolute coordinate of axis X of the intermediate point;
U	: the relative coordinate of axis X of the intermediate point;
Z	: the absolute coordinate of axis Z of the intermediate point;
W	: the relative coordinate of axis Z of the intermediate point;

Description

1. Starting from the start point, move to the intermediate point specified by X/U, Z/W at a fast moving speed, and then return to the 2nd, 3rd and 4th reference points of the machine tool.
2. When returning to the second reference point of the machine tool, the code address P2 can be omitted.
3. G30 is a non-modal G code;
4. G30 is normally used when the tool change position is different from the reference point;

Example

```
O1000
T0101
G0 X50 Z3
... //Machining program
G30 X50 Z0 P2 // Perform G0 X50 Z0 internally first, and then return the second reference point
T0202
...
M30
```

2.8.5 Setting a workpiece coordinate system (G50)

Format

G50 X/U_ Z/W_ α/β_;

X : Specify the workpiece coordinate of axis X of the current tool in the workpiece coordinate system;

Z : Specify the workpiece coordinate of axis Z of the current tool in the workpiece coordinate system;

U : Workpiece coordinate system shifting in the X direction;

W : Workpiece coordinate system shifting in the Z direction;

α/β : Additional axis coordinates;

(α is used for absolute programming, and β is used for incremental programming); and multiple additional axes can be specified.

Description

If G50 is used to set the coordinate system in tool compensation, a coordinate system is to be set. In this coordinate system, the position before applying tool compensation is the position specified by G50.

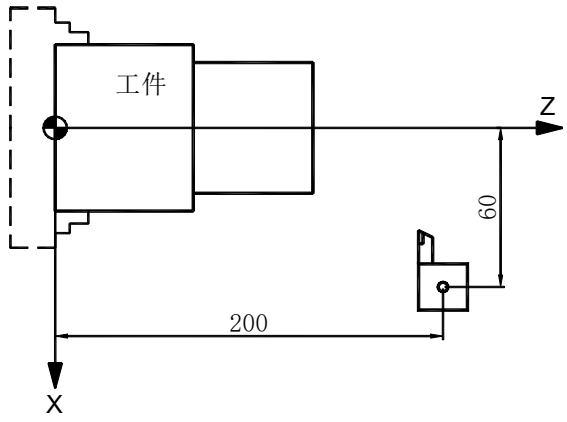
1. Absolute value coordinate system setting

The command G50 X_ Z_ is used to establish a coordinate system, so that the coordinate of a certain point on the tool (such as the tool nose) in this coordinate system is. This coordinate is called workpiece coordinate system. Once the workpiece coordinate system is established, the position of the subsequent absolute value command or relative value command is represented according to the origin of coordinates of this coordinate system.

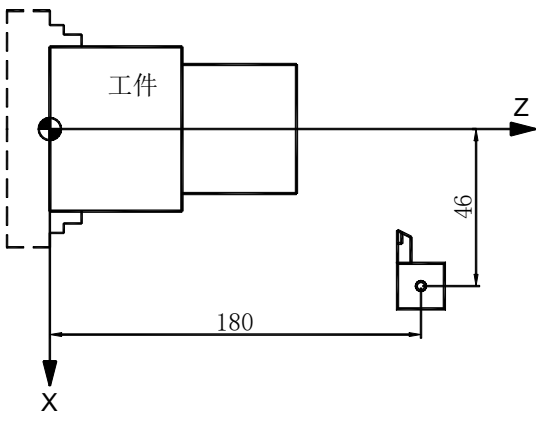
2. Relative value coordinate system setting

The command G50 U_ W_ is used to realize the translation of coordinate system.

After executing the G50 U_ W_ command, the physical position of the tool post remains the same and the absolute position value is increased by (U, W). `



Absolute coordinates (X60, Z200) before G50 U-14 W-20 is executed



Absolute coordinates (X46, Z180) after execution of G50 U-14 W-20

2.8.6 Programmable Mirror Image (G50.1/G51.1)

Format

G51.1 X_Z_; Set the programmable image
G50.1 X_Z_; Cancel the programmable mirroring

Description

For the symmetry axis specified in the program, a new image can be generated at the position specified by the program.

Detailed description

1. G51.1 specifies the position of the symmetry axis of the mirror image or the position of the mirror image point;
2. In the rows specified in G51.1 and G50.1, the axis does not move;
3. The path coordinates must be subject to absolute programming.

Example

O1000	
G50.1 X0	// Cancel mirroring
M98 Q100	///Call track (subprogram segment of N100)
G51.1 X0	// Take the position X=0 as the mirror axis to start mirroring
M98 Q100	// Call track (subprogram segment of N100)
G50.1 X0	// End mirroring
M30	
N100	//subprogram segment
G00 X100.0 Z0	//Positioning (During mirroring, the actual X-axis coordinates are X-100)
G01 Z-10 F200	//Feed
G01 X90 Z-20 F300	//Feed (During mirroring, the actual X-axis coordinates are X-90)
G0 Z0	//Positioning
M99	

2.8.7 Workpiece Coordinate System Preset(G50.3)

Format

G50.3 X_ Z_ α_;

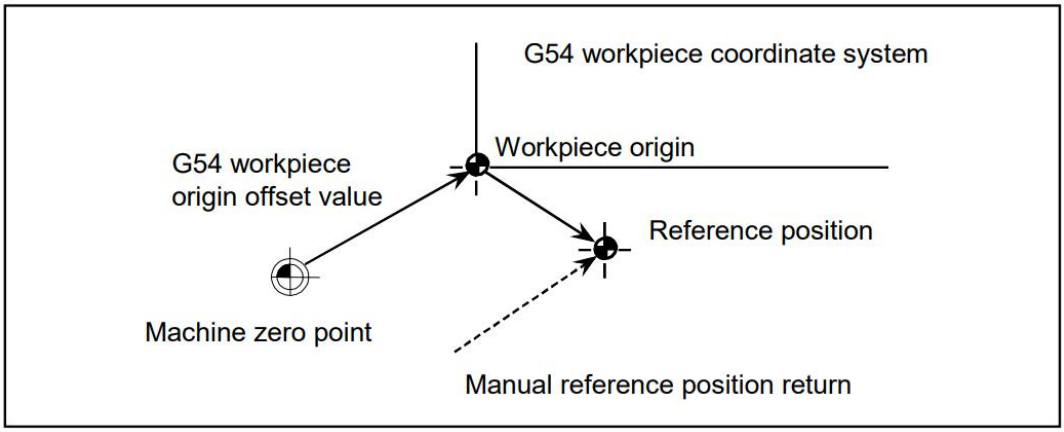
X : Set the origin of the X-axis of the workpiece coordinate system (absolutely specified)

Z : Set the origin of the Z-axis of the workpiece coordinate system (absolutely specified)

α : Set the origin of the additional axes of the workpiece coordinate system (absolutely specified);

Description

When manual reference position return operation is performed in the reset state, a workpiece coordinate system is shifted by the workpiece origin offset value from the machine coordinate system zero point Suppose that the manual reference position return operation is performed when a workpiece coordinate system is selected with G54. In this case, a workpiece coordinate system is automatically set which has itsorigin displaced from the machine zero point by the G54 workpiece origin offset value: the distance from the origin of the workpiece coordinate system to the reference position represents the current position in the workpiece coordinate system.



This function is similar to the reference point return function and can restore the workpiece coordinate system that has been offset by operations such as G50 and manual clearing of machine coordinates.

Usually used in conjunction with G10.

Detailed description

1. The implementation of G50.3 will cancel the G52 temporary coordinate system;

Example

1. Restore the origin of the workpiece coordinate system (i.e., the absolute coordinate value is the same as the machine coordinate value when there is no tool compensation value)	
G50.3 X0 Z0 Y0	// Restore the origin of the workpiece coordinate system for axes X, Y and Z

2.8.8 Local Coordinate System (G52)

Format

G52 X/U_ Z/W_; Establish local coordinate system

G52 X0 Z0; Cancel local coordinate system

X : Specify the position of the origin of X-axis local coordinate system under the current workpiece coordinate system.

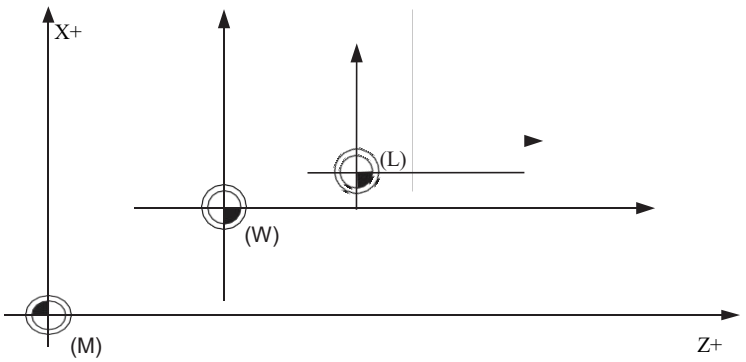
Z : Specify the position of the origin of Z-axis local coordinate system under the current workpiece coordinate system.

U : Shift the position of the origin of X-axis local coordinate system under the current workpiece coordinate system.

W : Shift the position of the origin of Z-axis local coordinate system under the current workpiece coordinate system.

Description

When programming on the workpiece coordinate system, for convenience, a sub-workpiece coordinate system can be created in the workpiece coordinate system. Such a sub-coordinate system is called a local coordinate system.



(L) G52 local coordinate system

(W) Workpiece coordinate system

(M) Machine coordinate system

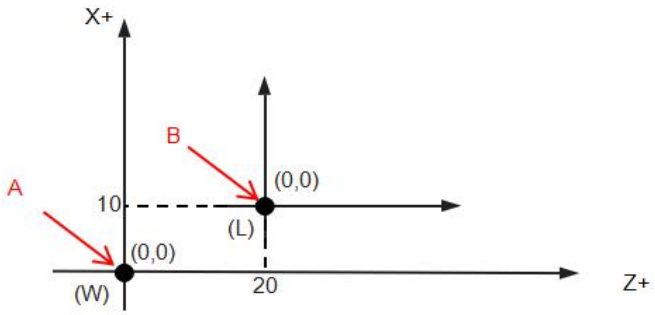
- Using G52 command, local coordinate systems can be set in all workpiece coordinate systems (G54 to G59), and the origin of each local coordinate system becomes the position of IP_ in each workpiece coordinate system.
- Once the local coordinate system is set, the specified axis movement command becomes the coordinate value in the local coordinate system. When you want to change the local coordinate system, you can specify the origin position of the new local coordinate system in the workpiece coordinate system while performing G52.
- To cancel the local coordinate system or specify coordinate values in the workpiece coordinate system, the origin of the local coordinate system should coincide with the origin of the workpiece coordinate system;

Detailed description

1. The setting of local coordinate system does not change the workpiece coordinate system and machine coordinate system;
2. To cancel the local coordinate system, the zero point of the local coordinate system should be consistent with the zero point of the workpiece coordinate system, that is, to perform G52 X0 Z0;
3. When the machine tool returns to zero, the local coordinate system of the axis will be cancelled;
4. Whether to cancel the local coordinate system during resetting, or execution of M30, it can be set by bits 4 and 5 of sys-parameter-0400.
5. When G52 is executed, the nose radius compensation will be temporarily cancelled.
6. After the execution of G52, the local coordinate system remains valid until the next G52 command is executed, and no movement is generated when the G52 command is specified;
7. When the workpiece coordinate system is set with G50, the local coordinate system under all workpiece coordinate systems of the specified axis will be cancelled, and the local coordinate system of the axis without specified coordinate value will not be cancelled.

Example

```
O1000
T0001
G00 X0 Z0      // point A
G52 X10 Z20 // set local coordinate system origin
G00 X0 Z0      // move to point B
G52 X0 Z0      // cancel the local coordinate system
G00 X0 Z0      // move to point A
M30
```



(W) Workpiece coordinate system
(L) Local coordinate system

2.8.9 Machine Soordinate System Selection (G53)

Format

G53 [G00/G01];

Description

When program segments with G53 commands, such as G53 G00 X100, are executed, the CNC system performs positioning based on the machine coordinate system.

1. G53 is a non-modal command, which is only valid in the program segment where G53 is specified.
2. In G53 mode, X/Y/Z must be absolutely programmed, and U/V/W cannot be incrementally programmed;
3. When the program segment in which G53 is located omits the movement commands such as G00 and G01, the CNC system will select G00 by default for coordinate positioning.

Note:

Versions prior to CNC software version 2.80.28 (not included) only support G00 positioning.

Example

```
O1000
G53 Z00              // Fast positioning to machine coordinate Z100
T0101                // Tool change
...                  //Processing
G53 G01 Z100 F10     //slowly move to the position of machine coordinate Z100
T0202                // Tool change
...                  //Processing
M30
```

2.8.10 Workpiece coordinate system (G54～G59)

Format

G54; Workpiece coordinate system 1
G55; Workpiece coordinate system 2
G56; Workpiece coordinate system 3
G57; Workpiece coordinate system 4
G58; Workpiece coordinate system 5
G59; Workpiece coordinate system 6

Description

1. The workpiece coordinate system is a coordinate system with the datum point of the machined workpiece as the origin and is used to simplify programming on the workpiece.
2. This command allows you to move to a position on the workpiece coordinate system. The workpiece coordinate system is the coordinate system used by the programmer for programming. In addition to the six groups G54 to G59, there is an extended workpiece coordinate system (G54.1).

The number of groups varies according to the machine manufacturer's specifications.

Detailed description

1. The amount of tool nose R compensation for the specified axis is not canceled in the commands G54 to G59, even if a switch of the workpiece coordinate system is specified;

2. Bit 1 of sys-parameter-0400 is used to specify the selection of the G54 coordinate system or the coordinate system before power-off when the machine is energized.

3. G54 to G59 are modal commands (group 12);

4. The set offset amount of the workpiece coordinate system indicates the distance to point 0 of the basic machine coordinate system;

2.8.11 Additional workpiece coordinate system (G54/G54.1 P)

Format

G54/G54.1 Pn;

Pn	: designation codes of additional workpiece coordinate system
n	: 1～99

Description

Except for the 6 workpiece coordinate systems (standard workpiece coordinate systems) of G54 to G59, 99 workpiece coordinate systems (additional workpiece coordinate systems) can also be used.

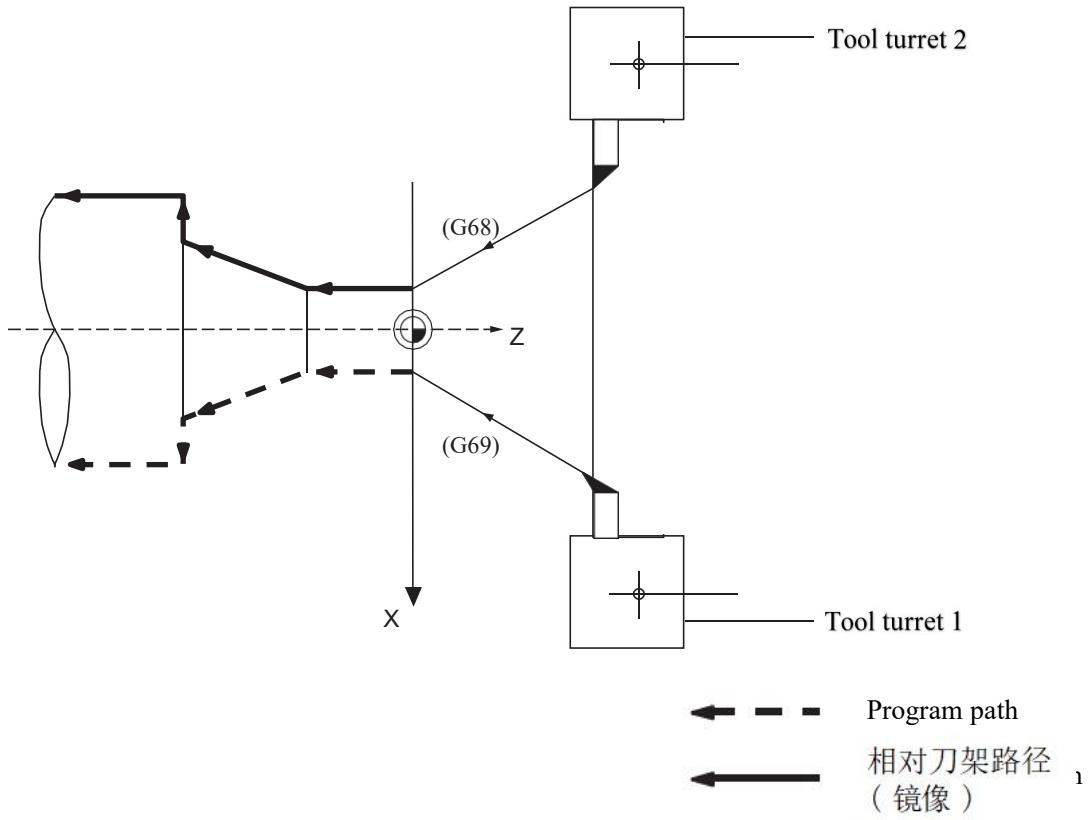
2.8.12 Opposite tool mirror image (G68/G69)

Format

G68; Opposite tool mirroring starts **G69; Cancel tool mirroring**

Description

If the machine tool has opposite tool stands composed of two opposite tool stands located on the same control axis, the mirror image can be applied to the X axis through the command of G code, which is like creating the machining program on the opposite tool stands in the same coordinate system and making symmetrical cutting.



Detailed description

1. After G68 is specified, the CNC will mirror X0 as the symmetry axis and must adopt absolute programming.

2. Command G53 will not be mirrored;

Example

O1000	T0002
T0001	G68 //Mirror starts; after execution of this command, the X coordinate is reversed
G69 // Cancel mirroring	G00 X20 Z5
G00 X20 Z5	G01 Z-10
G01 Z-10	G01 Z-20 X40
G01 Z-20 X40	G00 X50 Z10
G00 X50 Z10	M30

2.8.13 Rotation of coordinate system (G68.1/G69.1)

Format

G68.1 α_ β_ R_; Rotating workpiece coordinate system

α : The absolute coordinate of the 1st axis in the coordinate system of the coordinate rotation center plane, such as the Z-axis in the G18 plane

β : The absolute coordinate of the 2nd axis in the coordinate system of the coordinate rotation center plane, such as the X-axis in the G18 plane

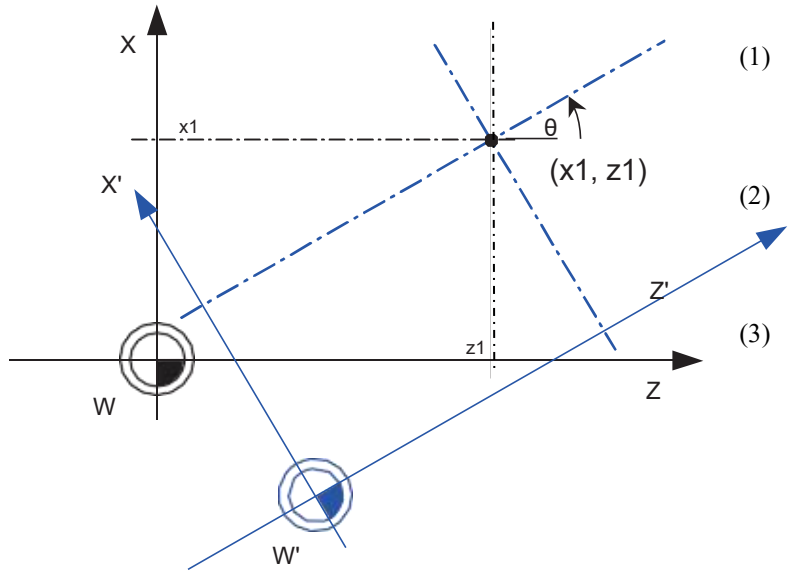
R : The angle of rotation of the coordinate system (in degrees), in the selected plane, counterclockwise is positive

G69.1; Cancel rotation of the workpiece coordinate system

Description

When machining complex shapes in positions rotating around the coordinate system, the shape before rotation can be specified in the local coordinate system, and the rotation angle can be specified by the program coordinate rotation command to machine the rotated shape, mainly for drilling and tapping of inclined planes.

The command plane is first selected by G17 to G19, and the following figure illustrates the G18 plane (*ZOX*) of the lathe as an Example



(1) Specify the center of rotation coordinates (x1,z1) in absolute values;

(2) Rotation in the counterclockwise direction according to the angle specified in the rotation angle θ;

(3) If the command value is outside the set range, the remainder of the command value divided by 360 degrees is used. Example When it is specified to be 400, the command angle is the remainder 40 degrees of 400/360;

W: Workpiece coordinate system before rotation — W': Workpiece coordinate system after rotation

θ: rotation angle — (x1, z1): coordinates of of rotation center

Detailed description

1. The coordinates of the center of rotation must be specified in absolute values;

2. In the coordinate rotation mode, G68.1 is displayed in the modal information screen, or G69.1 when coordinate rotation is canceled;

3. During resetting, the rotation of the coordinate system is forced to cancel;

4. When the rotation center coordinates (x1, z1) are omitted, the current coordinate value is used as the rotation center for the G68.1 command;

5. Relationship with other functions:

(1) Unable to switch planes in the coordinate system rotation mode (G17/G18/G19);

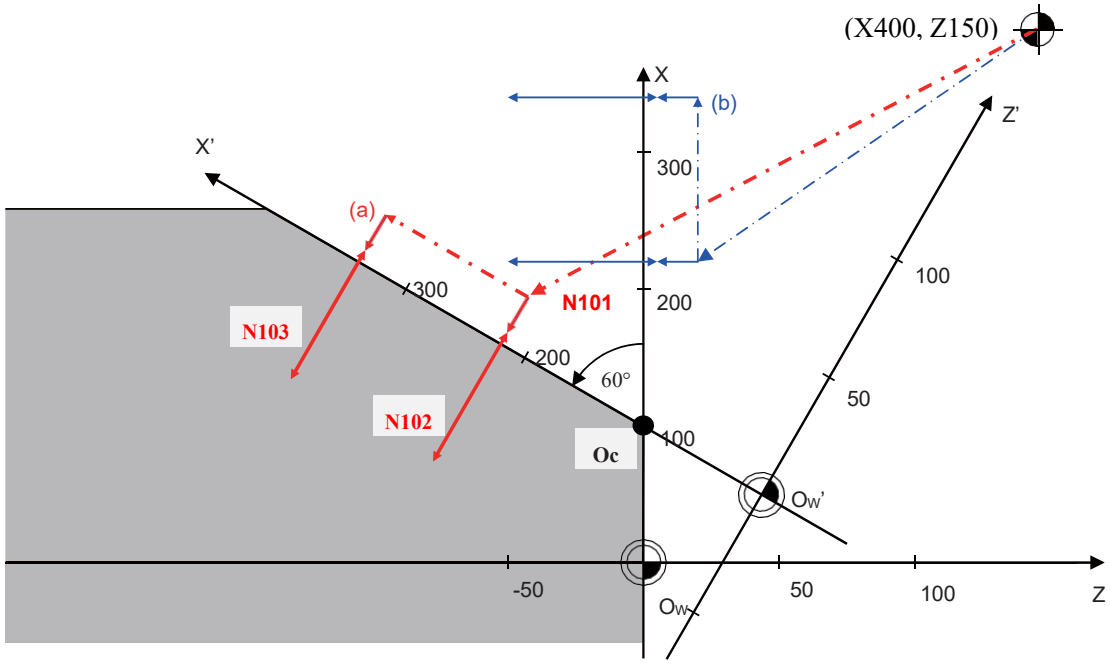
(2) Polar interpolation (G12.1) and cylindrical interpolation (G07.1) cannot be performed in the coordinate system rotation mode, which generates unpredictable errors;

(3) In the coordinate system rotation mode, the coordinates read by the G31 skip command are the rotated coordinate values;

(4) G53 is not affected by the rotation of the coordinate system;

Example

Drilling cycle on the inclined plane



(Oc) rotation center

(Ow) Workpiece coordinate origin before rotation (Ow') Workpiece coordinate origin after rotation

(a) subprogram path after rotation (b) subprogram path before rotation

O0001	
T0001	
G98 G69.1 G18 M13	//Modal recovery, spindle unlocking
G00 X400 Z150	//Start point positioning
M19	//Spindle orientation
G68 .1 X100 Z0 R60	// Rotate the workpiece coordinates by 60 degrees counterclockwise with the absolute coordinates (X100, Z0) as the center of the circle
M103 S2=1000	// Start the 2nd spindle
N101 G00 X210 Z20	// Fast positioning to the position of the 1st hole of the inclined plane
N102 G83 Z-50 R1 F100 M12	// Drill hole 1
N103 X340	// Drill hole 2
G69.1	// Modal recovery
G00 X400 Z150	// Return to the start point
M105	// Stop the 2nd spindle
M30	

2.9 MEASUREMENT / SKIP FUNCTIONS

2.9.1 Torque detection during axis position control (G10 L14 P0~P6)

Format ①:

G10 L14 P_ α_;

α : Axis torque detection reference value, multiple axes can be specified;
α indicates data specified in absolute coordinates, such as X/Y/Z

P : Detection method
P=0 Detect all the motion loads of the specified shaft, and give an alarm if the limit is exceeded
P=1 Detect G01 motion loads of the axis, and give an alarm if the limit is exceeded
P=2 Detect G01 motion loads of the axis, and do not give an alarm if the limit is exceeded
P=3 Detect G01 motion loads of the axis, and do not give an alarm if the limit is exceeded
P=4 Detect all the motion loads of the axis, and do not give an alarm if the limit is exceeded
P=5 Detect all the motion loads of the axis, and do not give an alarm if the limit is exceeded
P=6 Set the torque skip value of G31 command

Format ②:

G10 L14 P_ α_ R_;

α : Axis torque detection time (seconds), α indicates data specified in absolute coordinates, such as X/Y/Z/A/B/C

P : Detection method
P=20 Judge the time when the actual torque exceeds the specified torque, and give an alarm when it is less than the set value
P=21 Judge the time when the actual torque exceeds the specified torque, and give an alarm when it exceeds the set value
P=30 Judge the time when the actual torque exceeds the specified torque, and give an alarm when it is less than the set value
P=31 Judge the time when the actual torque exceeds the specified torque, and give an alarm when it exceeds the set value
P=40 Judge the time when the actual torque exceeds the specified torque, and give an alarm when it is less than the set value
P=41 Judge the time when the actual torque exceeds the specified torque, and give an alarm when it exceeds the set value
P=50 Judge the time when the actual torque exceeds the specified torque, and give an alarm when it is less than the set value
P=51 Judge the time when the actual torque exceeds the specified torque, and give an alarm when it exceeds the set value

R : The macro variable number; the axis number will be written to the macro variable specified by R as a binary number when the judgment condition is met.

Description

Format① sets the torque load reference value at G00/G01 of each axis. When the value is greater than the set reference value, the CNC generates an alarm;

Format② is used to determine whether the time each axis exceeds the specified load is up to standard, and it generally is used for tool wear and tool damage detection;

1. Data for each axis must be specified in absolute coordinates, and the incremental format of the coordinate axes cannot be used;

2. Before using this function, it is necessary to determine whether the setting of sys-parameter-5522 is correct, otherwise, inconsistency between the torque value specified by G10 command and the actual servo torque will result in a detection error;

3. Format② needs to be used in conjunction with format①, and is valid only for P2~P5 of format ①;

4. P2~P5 of format① will activate system timer system diagnostics 0530~0533;

5. Format② load judgment is based on system timer system diagnostics 0530~0533, as shown in the following table:

System diagnosis number	Corresponding format ① P-value	Corresponding format ② P-value
0530 Load monitoring 2 arrival time for each axis (s)	P2	P20/P21
0531 Load monitoring 3 arrival time for each axis (s)	P3	P30/P31
0532 Load monitoring 4 arrival time for each axis (s)	P4	P40/P41
0533 Load monitoring 5 arrival time for each axis (s)	P5	P50/P51

Example

1. G01 load detection	
O1000	
G10 P1 L14 Z80	//Set the torque detection value of Z-axis torque G01 command to 80%, and give an alarm if the value is exceeded
G00 X0 Z0	// Fast positioning (do not detect when performing G00)
G01 W-10 F20	// Drilling holes in Z direction (detect the load when performing G01)
G00 Z0	// Fast positioning (do not detect when performing G00)
G10 P1 L14 X80	// Set the torque detection value of X-axis torque G01 command to 80%, and Z-axis is canceled
G01 U-10 F20	// Drilling holes in X direction (detect the load when performing G01)
G00 X0 Z0	// Fast positioning (do not detect when performing G00)
G10 P1 L14	// Cancel torque detection of all axes
M30	

2. For detection of automated rig testing	
O1000	
G10 P2 L14 Z80	//Set Z-axis torque detection to 80%; when the value is exceeded, do not give an alarm but activate timer 2
G10 P3 L14 Z10	//Set Z-axis torque detection to 10%; when the value is exceeded, do not give an alarm but activate timer 3
G00 X0 Z0	// Fast positioning (do not detect when performing G00)
G98 G01 W-10 F30	//Z-direction drilling, speed: 30 mm/min, expected to finish drilling in 20 seconds (detect the load when performing G01)
G00 Z0	// Fast positioning (do not detect when performing G00)
#100=0 // Initialization	// Judge whether the duration of Z load torque exceeding 80% is longer than the set value 15 seconds // If exceeded, a value of #100 is assigned, indicating that the drill bit may be worn out due to excessive load during drilling.
G10 P21 L14 Z15 R100	
#101=0 // Initialization	// Judge whether the duration of Z load torque exceeding 10% is shorter than the set value 15 seconds // If so, a value of #101 is assigned, indicating that the drill bit may have broken due to too little load during drilling
G10 P30 L14 Z15 R101	
IF[#100==4]#3000=1 (excessive load)	//Judging whether the Z-axis meets the overload condition by #100 and generating an alarm prompt
IF[#101==4]#3000=1 (drill bit is broken)	//Judging whether the Z-axis meets the tool breakage condition by #101 and generating an alarm prompt
M30	

2.9.2 Drilling/Tapping overload detection (G10 L14 P7~P8)

Format

G10 L14 P7/P8 R_ J_ α_;

P7 : Specify drilling command (G83/G87) overload detection

P8 : Specify tapping command (G84/G88) overload detection

R : Macro variable number, the value of the specified macro variable + 1 in case of overload

J : Overload filtering time (seconds); it is considered to be overload only when the motor load exceeds the set value and the J setting is maintained

α : Torque detection value of the axis, multiple axes can be specified, α indicates data specified in absolute coordinates, such as X/Y/Z

Description

For G83 drilling cycle command G84 (must be the rigid tapping mode specified by M29) tapping cycle command	
Case 1	During tapping or drilling, if the load reaches the set value, it immediately retracts to the safe position, and then the CNC generates an alarm and pauses operation.
Case 2	During tapping or drilling, if the load reaches the set value, it immediately retracts to the safe position without alarm or pause, ends the current program segment, and continues to execute the next segment.
Case 3	During tapping or drilling, if the load reaches the set value, it immediately retracts to the safe position, and repeats the current program segment until the current segment can be successfully executed.

Detailed description

Drilling cycle related sys-parameters		
Sys-par NO.	**Meaning**	**Remarks**
5100 Bit 2	Not return / Return when the drilling cycle is overloaded	Drill overload return master switch
5100 Bit 3	Not retry / Retry after drilling cycle overload return	
5100 Bit 4	Not give an alarm / Give an alarm after drilling cycle overload return	
5112	Upper limit of each load during drilling	When G10 omits the axis number, the axis load is specified by this parameter
5113	During drilling cycle, the overload status returns a variable number	When R is omitted in G10, the macro variable number is specified by this parameter

Tapping cycle related sys-parameters		
Sys-par NO.	**Meaning**	**Remarks**
5200 Bit 4	Not return / Return when the tapping cycle is overloaded	Tapping overload return master switch parameters
5200 Bit 5	Not retry / Retry after tapping cycle overload return	
5200 Bit 6	Not give an alarm / Give an alarm after tapping cycle overload return	
5250	Upper limit of each load during tapping	When G10 omits the axis number, the axis load is specified by this parameter
5251	During tapping cycle, the overload status returns a variable number	When R is omitted in G10, the macro variable number is specified by this parameter

Example

1. Drilling cycle	
Requirements: Set the overload judgment value of drilling feed axis (Z-axis) to 100%, and when the current drilling axis is overloaded, perform retracting and CNC alarm.	
sys-parameter setting: bit 2 of 5100 is set to 1 bit 3 of 5100 is set to 0 bit 4 of 5100 is set to 1	
O1000	
T0101	
G00 X0 Z10	
M03 S1000	
G10 L14 P7 Z100	// Set Z-axis overload detection to 100%
G98 G83 X0 R1 Z-10 F100	//If the Z-axis load exceeds 100% during the drilling cycle, it performs retracting immediately to the R point and the CNC generates an alarm.
X50	
X100	
G80	
G10 L14 P7	//Cancle the axis overload detection for drilling cycle
M30	

2. Tapping cycle (A-axis for spindle indexing, Y-axis for tapping power head (2nd spindle))	
Requirements: Set the overload judgment value of 120% for tapping feed axis (Z axis) and tapping rotating axis (Y axis); when the current specified axis is overloaded, perform retracting and pause alarm.	
sys-parameter setting: bit 2 of #5100 is set to 1 bit 3 of #5100 is set to 0 bit 4 of #5100 is set to 1	
O1000	
T0101	
G00 X50 Z10	//The position where the workpiece diameter is 50 mm
G10 L14 P8 Z120 Y120	//Setting Z-, and Y-axis overload detection to 120%
M29 S2=300	//If the load on either Z-axis or Y-axis exceeds 120% during tapping, the CNC will immediately perform a retreat to the Z10 position and generate an alarm.
G84 X50 A0 R1 Z-10 F1.25	
A180	
G80	
G10 L14 P8	//Cancel the axis overload detection for tapping command
M30	

3. Complex requirements, 4-axis lathe CNC (XYZA), tapping cycle (A-axis for spindle indexing, Y-axis for tapping power head (2nd spindle))	
Requirements: Set the feed axis (Z axis) overload judgment value to 100%; when the current feed axis is overloaded, perform retracting and retry; when the retry times are greater than 2, the alarm is suspended	
sys-parameter setting: bit 4 of #5200 is set to 1 bit 5 of #5200 is set to 0 bit 6 of #5200 is set to 0	
O1000	
T0101	
G00 X50 Z10	//The position where the workpiece diameter is 50 mm
M03 S300	
G10 L14 P8 Z100 R1	//Set Z-axis overload detection to 100%, the value of #1 +1 at overload condition
M29 S2=300	//If the Z-axis load exceeds 100% during the tapping cycle, it performs retracting immediately to the R point. //M98 Q10001 calls the subprogram segment to determine whether tapping is in place and the number of tapping repetitions
G84 X50 A0 R1 Z-10 F1.25	
M98 Q10001	
A180	
M98 Q10001	
G80	
G10 L14 P8	//Cancel the axis overload detection for tapping cycle
M30	
N10001	// Judgment subprogram
G04 I0	// Block program pre-reading
WHILE[#1!=0]DO1	//Determine whether there is overload, repeat tapping if overload occurs
A[#5004]	// Repeat tapping in situ
G04 I0	
IF[#1==2] #3000=1 (too many retries)	// If the 2nd tapping fails, an alarm will be generated
IF[#1==1]GOTO 99	// If the 2nd tapping is successful, the cycle is skipped
END1	
N99	
#1=0	// Overload judgment variable is cleared to zero
M99	// Return to the main program

2.9.3 Torque Limit during Axis Position Control (G10 L14 P9)

Format

G10 L14 P9 α_;

α : Axis torque limit percentage, multiple axes can be specified, α indicates data specified in absolute coordinates

Description

1. This command limits the percentage of torque output in position mode for the specified axis;
2. Servo Diagnostics 0440 is capable of displaying the current torque limit value for each axis;
3. When G00 and G01 commands are used, the torque limit mode will be forced out;
4. When an axis coordinate term specifies the torque limit value to be 0, it indicates that the torque limit is removed;

2.9.4 Axis Position Torque Control (G10 L14 P10~P12)

Format

G10 L14 P_ α_;

α : Axis torque related parameter, specified in absolute coordinates, such as X/Y/Z

When P=10, α indicates the percentage of torque output value of specified axis, and the sign indicates the direction of motor rotation.

When P=11, α indicates the motor speed at the specified axis (unsigned), unit: rpm

When P=12, α indicates the torque limit value when reverting to position mode after exiting torque mode

P : Mode

Position → Torque mode switching at P=10

Shaft motor speed setting in torque mode at P=11

Torque→position mode switching at P=12

Description

It directly controls the current value and output torque of the motor, and is generally used in tension control and motorized tail-stock control for stranded fabric machines, etc.

1. Bit 1 of sys-parameter-5520 is used to set whether to exit torque control mode upon reset;

2. When the motor speed of the torque mode is not specified with G10 P11, it will be specified by sys-parameter-5521;

3. The maximum torque output of the servo needs to be matched by sys-parameter-5522, otherwise, the torque output percentage specified by G10 P10 will deviate from the actual servo motor torque output;

4. When the torque output value is specified by the coordinate language to be 0 in G10 P10 mode, it indicates that the torque mode is canceled and the previous control mode (position or speed) is restored;

5. When the torque limit value is specified by the coordinate language to be <= 0 (-1 is recommended) in G10 P12 mode, it indicates that the torque limit value reverts to the default value (i.e., no limit);

Example

Application of electric tailstock	
Requirements: Execute M78/M79 to control the feeding and retracting of the electric (servo axis B) tailstock, so that the traditional hydraulic tailstock can be replaced	
sys-parameter setting: Bit 1 of #5520 is set to 0. The motor cannot exit torque control mode during reset or emergency stop because it is subject to tailstock control. #5521 sets the speed of the servo motor controlled by the electric tailstock for forward and backward movements #9060 set to 2 #9061 set to 9030 #9062 set to 78 #9063 set to 79	
O1000 (main program)	O9030 (subprogram called by M78)
T0101	G10 L14 P10 B50 // Motor outputs at 30% of rated torque and enters torque mode
G00 X0 Z10	M99
M03 S1000	
M78// Tailstock moves forward	O9031 (subprogram called by M79)
...	G10 L14 P12 B50 // The motor exits torque mode and the motor torque is limited to 50%
M79 //Tailstock moves backward	G160 P50 UB100 F500 // B-axis leaves the workpiece in the positive direction by 100 mm
M30	M99

2.9.5 Skip Function (G31)

Format ①:

G31 X/U_ Z/W_ α/β_ R_ F_;Detection input port
(The input port to be detected is specified by sys-parameter-5510)

X/U : X-axis end point coordinates (X is absolute programming, and U is incremental programming)
Z/W : Z-axis end point coordinates (Z is absolute programming, and W is incremental programming)
α/β : Additional axis end point coordinates
(α is used for absolute programming, and β is used for incremental programming)
R : Macro address, when the signal is detected, set the value of the macro address to 1
F : Feed rate, linear axis unit: mm/min (in G98 mode), mm/r (in G99 mode)
Rotating axis unit: 36°/min (in G98 mode), 36°/r (in G99 mode)

Format ②:

G31 X/U_ Z/W_ α/β_ F_ L/K_; Detection input port (specified by L/K)

L : Detected input port number (positive integer), the detection input signal is valid for connection
K : Detected input port number (positive integer), the detection input signal is valid for disconnection

Format ③:

G31 X/U_ Z/W_ α/β_ F_ I_; Detection input port (specified by I)

I : Port number of input port to be detected (integer)
When I > 0, the detection input signal is valid for connection.
When I < 0, the detection input signal is valid for disconnection.

Format ④:

G31 X/U_ Z/W_ α/β_ F_ P_; Detection of G address signal of PLC

P : Detect the G0014 address signal of PLC, range: 1~8,
corresponding to addresses G0014.0~G0014.7

Format ⑤:

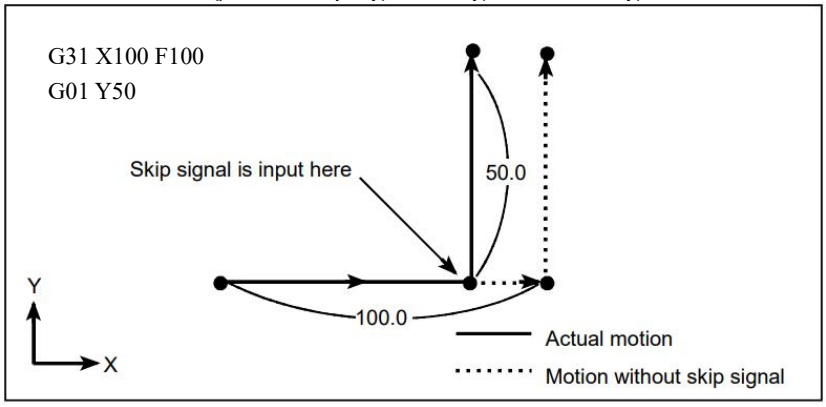
G10 L14 P6 α_; Set the detection load rate of the axis (the load of each axis is specified by sys-parameter-5511 when G10 is not programmed)
G31 X/U_ Z/W_ α/β_ F_ P98;

Format ⑥:

Define P99 additionally in command formats ①②③④; detect input signal or axis load rate
(the load of each axis is specified by sys-parameter-5511)

Description

In the linear interpolation of G31 command, if an external jump signal is input, the machine movement is stopped immediately, the coordinate value is read, the remaining distance is rounded off, and the next program segment command is executed. Generally used for programming of measuring instruments.



Detailed description

1. The acceleration and deceleration performance and speed limit of G31 are the same as those of G01;

2. In command format ①, if the value of sys-parameter-5510 is set to 0, a system alarm will be generated upon execution;

3. In command format ⑥, the G31 command generates a skip when either the load rate condition or the input signal condition is met;

4. When a skip is generated, its current absolute coordinate position value is stored in the CNC variables #5061 (axis 1 within the channel) ~ #506n (axis n) and can therefore be used in user macros;

Example

1. Positioning of rotating axis A by external sensor with input port number 13	
O9010	
G98	//Feed per minute
G31 UA-400 F300 L13	//First search the sensor signal reversely at a fast speed (coarse positioning), and the search range is 400 degrees
G31 UA-20 K13	// Ensure that A-axis is out of sensor detection range
G01 UA-3	// Ensure that A-axis is out of sensor detection range to a certain extent to prepare for forward fine positioning
G31 UA30 F50 L13 R100	// Seek sensor signal at slow forward speed (fine positioning)
IF[#100==0] #3000=1 (no signal detected) ENDIF	// Verify that sensor signal is detected, but no alarm signal is detected
G50 A0	// Set the current absolute coordinate of A-axis to 0
M30	

2.9.6 Flexible Skip(EGB Skip) (G31.8)

Format

G31.8 X/U_ Z/W_ α/β_ F_ K/L/I_ P_ Q_ R_;

X/U : X-axis end point coordinates (X is absolute programming, and U is incremental programming)

Z/W : Z-axis end point coordinates (Z is absolute programming, and W is incremental programming)

α/β : Additional axis end point coordinates
(α is used for absolute programming, and β is used for incremental programming)

F : Feed rate, linear axis unit: mm/min (in G98 mode), mm/r (in G99 mode)
Rotating axis unit: 36°/min (in G98 mode), 36°/r (in G99 mode)

I : Detected input port number, positive numbers indicate effective connection, negative numbers indicate effective disconnection

P : Macro address; used for the first address of macro to store the machine coordinates when the skip is triggered.

Q : Limit on the maximum number of signal skip, no limit if it is omitted

R : Address number; used for macro to store the number of input skip signals

Description

By executing G31.8, at the end of the skip command program segment, write in the user macro variable number specified by P, and the machine coordinate values when the skip signal is entered for the number of times specified by Q. In addition, in the user macro program variable number with R for commanding, the number of input times of the skip signal is written in whenever there is a skip signal input.

P must be specified and cannot be omitted, otherwise a system alarm will be generated;

Example

1. Gear measurement; let the gear workpiece be C axis, the number of gear teeth be 20, and the port number of the measurement input signal be 1	
G31.8 UA360 F100 I1 P500 Q5 R1	Rotate the workpiece at F100, and store the current C-axis absolute coordinate data to #500~#504 for a total of 5 teeth whenever the sensor touches the top of the tooth
IF[#1<=5] #3000=1 (abnormal tooth measurement)	#1 is the tooth count, if the number of teeth measured is less than 5, it will give an alarm

2.9.7 Axis torque limit skip (G160)

Format

G160 α/β_ P_ Q_ F_;

α/β : Specify the amount of axis movement,only 1 axis can be specified
(α is used for absolute programming, and β is used for incremental programming)

P : Torque limit value, unit: percentage. If P is -1, the torque limit of this axis is cancelled

Q : Axis position deviation during skipping, unit: mm

F : Feed speed

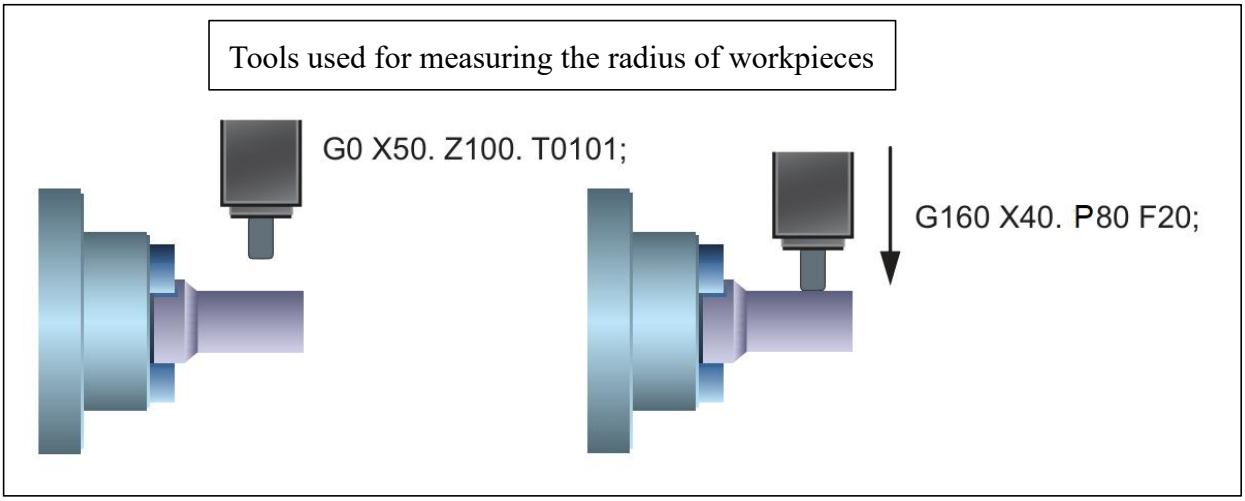
Description

This command enables the feed skip of servo axis in position mode with torque limit, similar to G31 command.

The G160 command segment generates a skip when both of the following conditions are met:

① The servo axis is obstructed by the load and the motor torque rises to the P value set by G160;

② The servo axis feedback position deviation value is greater than the Q value set by G160;



Detailed description

1. Before using this function, it is necessary to determine whether the setting of sys-parameter-5522 is correct, otherwise, the G10 command torque value will not match the actual servo torque resulting in a detection error;

2. Other axis displacement commands, such as G00, G01 and G02, will immediately cancel the torque limit of the shaft specified by G160;

3. The G160 command is a non-modal (group 00) command. When specifying G160 consecutively, be sure to specify G160 in each program segment;

4. When P is omitted, it is specified by sys-parameter-5515;

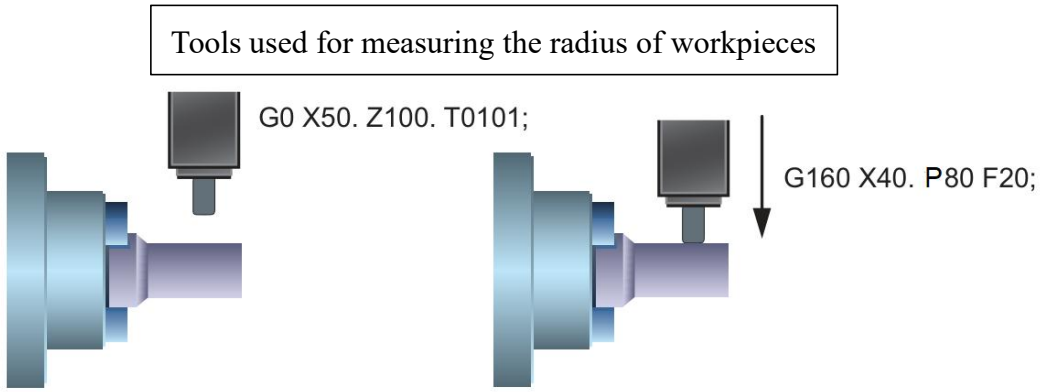
5. When Q is omitted, it is specified by sys-parameter-5516;

6. The absolute coordinate storage location at the time of skip generation is the same as for the G31 command (system macro variables #5061 ~ #5080);

Example

1. **Used as a detection instrument**

Let the workpiece diameter be 45 mm



O1000	
T0101	// Tool compensation for measurement
G00 X50 Z100	// Fast positioning (do not detect when performing G00)
G160 X40 P80 F20	//Slow speed feed measurement
#100=#5061	//Reading the absolute X-axis coordinates that generate the measurement jump
G160 X50 P80 F500	// Quickly leave the workpiece; G00 and G01 can not be used here for tool retracting
G10 P1 U[#100-45]	// Tool compensation correction to change the absolute coordinates of the jump position to 45 mm
M30	

2. Used as tailstock (the tailstock is controlled by the CNC's B-axis)	
O1000	
T0101	
G00 X0 Z10	
M03 S1000	
G160 B-200 Q2 P50	// Control the forward movement of tailstock; the torque limits 50% of the motor rated torque
...//Cutting	
G160 B0 P50	// Control the backward movement of tailstock; the torque limits 50% of the motor rated torque
M30	

2.10 MULTI-CHANNEL FUNCTIONS

2.10.1 Channel synchronization (G04.1)

Format

G04.1 Pxx (Qxx);

P : Group number, (1, 2, 3, 4, 5, 6.....); it must be specified;

Q : Specify the channel number;

e.g., Q12 indicates that channel 1 is waiting for synchronization with channel 2.

When Q is -1, it indicates that all channels are synchronized

Description

It is a multi-channel specific command for synchronizing programs between channels.

When neither P nor Q is specified, G04.1 is equivalent to G04i0, and it can be used for program pre-read suppression.

Example

1. Channel 2 is synchronized with channel 3: G04.1 Pxx Q23
2. Channels 1, 4, 5, 6, 7, 8, 9 are synchronized: G04.1 Pxx Q1456789
3. Channel 1 is synchronized with channel 10: G04.1 Pxx Q10
4. Channel 1 is synchronized with channel 12: G04.1 Pxx Q1 Q1=2

Channel 1	Channel 2	Channel 3
O1001 CH1 M03 S1000 G04.1 P1 Q123 // synchronization 1 ... G04.1 P2 Q12 // synchronization 2 M05 ... ... G04.1 P4 // synchronization 4 M30	O1002 CH2 M03 S1000 G04.1 P1 Q123 // synchronization 1 ... G04.1 P2 Q12 // synchronization 2 M05 ... G04.1 P3 Q23 // synchronization 3 G0 X0 Z0 ... ... G04.1 P4 // synchronization 4 M30	O1003 CH3 M03 S1000 G04.1 P1 Q123 // synchronization 1 ... ... G04.1 P3 Q23 // synchronization 3 G0 X0 Z0 ... M05 G04.1 P4 // synchronization 4 M30

2.10.2 Axis Synchronization (G51.4/G50.4)

Format

G51.4 P_ Q_; (Start of axis synchronization)

P : Master axis number (intra-system shaft number)

Q : Slave axis number (intra-system shaft number), cannot be omitted

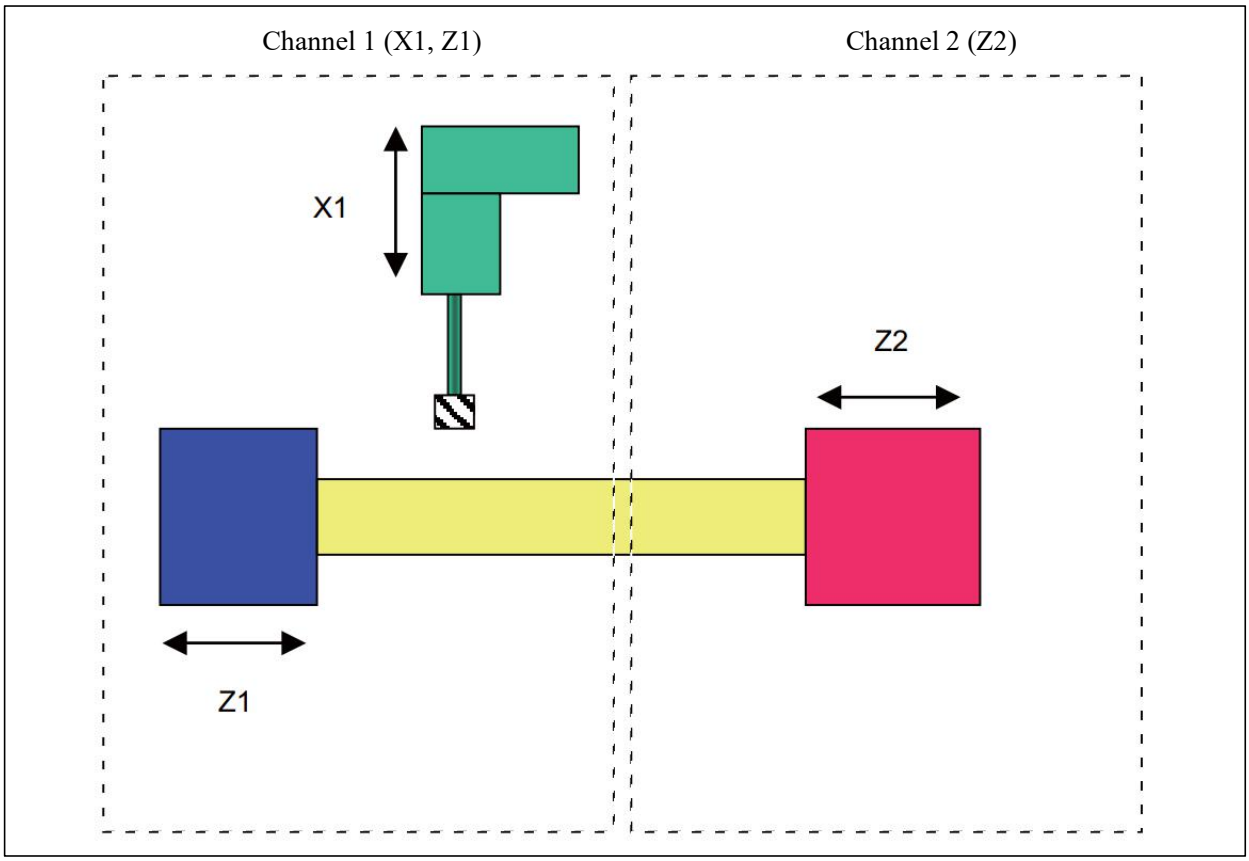
G50.4 Q_; (End of axis synchronization)

Q : Slave axis number, cannot be omitted

Description

Using this command, it is possible to synchronize the motion of an axis of the same type within a channel or in other channels, which can be used for gantry synchronizing shaft control, synchronizing cutting with dual tool rests, etc.

Example of synchronizing machining of Z1 and Z2 axes in channel 1 (X1, Z1) and channel 2 (Z2):



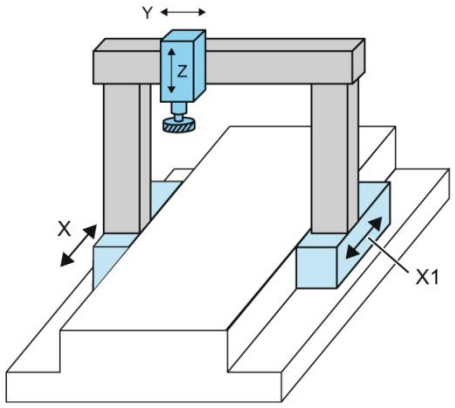
Detailed description

1. Bit 1 of sys-parameter-5600 corresponding to the slave axis must be set to 1 in order to use the G51.4 function;
2. When P is not specified in G51.4, the main control axis number set by sys-parameter-5610 is used.
3. During axis synchronization, the slave axis cannot be programmed and manually controlled;
4. The type of axis synchronization is incremental synchronization, and no absolute coordinate synchronization is performed;
5. One master axis can specify multiple slave axes;
6. The master axis and slave axis can be in the same channel or in different channels;

Example

1. Gantry synchronizing shaft

X-axis is always used as the master axis and X1-axis as the slave axis for synchronous control.



System axis configuration:

X (axis number 1)
Y (axis number 2)
Z (axis number 3)
X1 (axis number 4)

Parameter setting:

Bit 1 of sys-parameter-5600 is set to 1 to enable the synchronization function of X1 axis.

Bit 3 of sys-parameter-5600 is set to 0 to enable X1 axis to switch to the synchronization state after power-on.

sys-parameter-5610 is set to 1 (usually the axis number of X-axis is 1), set the master axis of X1-axis as X-axis

Bit 8 of sys-parameter-0001 is set to 1 **(not necessary)** to hide the display of X1 axis in the coordinate screen.

Note:

1. The position ring gain of the 2 servo motors of the gantry synchronizing shaft must be the same, otherwise, mechanical bending and tilting will occur;

2. If there is some deviation between the slave axis and the master axis, execute G50.4 Q4 in MDI mode to temporarily cancel the synchronization between X1 and X. Then, after removing the deviation between X1 axis and X axis by manual or hand wheel, execute G51.4 Q4 to re-establish the synchronization between X1 and X.

2. Multi-axis synchronous motion

Assume that there are axes (linear axes) X, Y and Z, the synchronization command makes axes Y and Z follow X axis to achieve the purpose of simplified programming.

T0001	
G00 X0 Y0 Z0	
G51.4 P1 Q2	// Synchronize X and Y axes, X is the master axis
G51.4 P1 Q3	// Synchronize X and Z axes, X is the master axis
G01 U10 F100	// Axes X, Y, and Z move synchronously
G50.4 P1 Q3	// Cancel the synchronization between axis X and axis Z
G01 U10 F100	// Axes X and Y move synchronously
G50.4 P1 Q2	// Cancel the synchronization between axis X and axis Y
G01 U10 F100	// X-axis motion

2.10.3 Axis Exchange (G140/G141/G142)

Format

G140 αn; (axis release)

α : Axis name, e.g., XYZABC;

n : The axis number for the axis exchange command;

The exchange number of each axis is defined by sys-parameter-5660.

G141 αn; (axis acquisition)

α : Axis name, e.g., XYZABC, etc;

n : The axis number for the axis exchange command;

The exchange number of each axis is defined by sys-parameter-5660.

G142; (axis recovery)

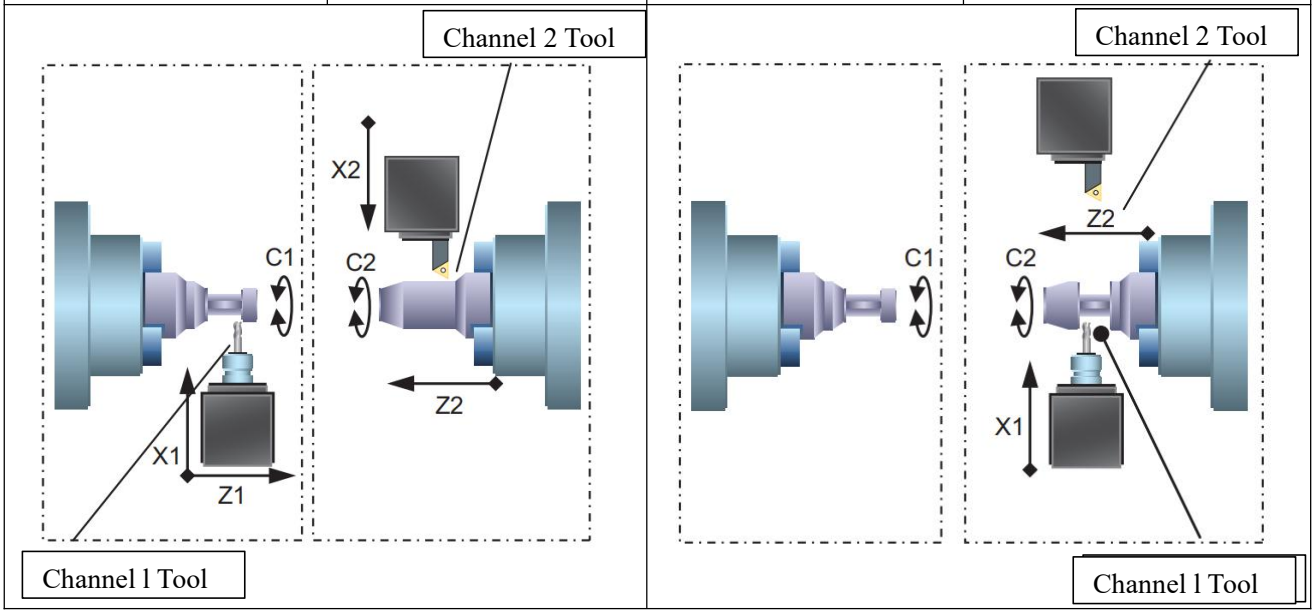
General process of axis exchange:

1. Execution of G140 to release the specified axis;
2. Execution of G141 to obtain the axis released by G140;
3. Execution of G142 to restore the axis to its initial state after use;

Description

With this function, you can freely exchange any axis. In a multi-channel CNC, the specified axis can be exchanged by the machining program of each channel for more flexible machining. This enables, for example, the tapping feed axis in channel 1 to be used as a tapping axis in channel 2 to perform tapping, etc., which is not possible with conventional axes.

Before axis exchange		After axis exchange	
Channel 1 (X1, Z1, C1) Machining in the 1st spindle	Channel 2 (X2, Z2, C2) Machining in the 2nd spindle	Channel 1 (Z1, C1) Machining in the 1st spindle	Channel 2 (X1, Z2, C2) Machining in the 2nd spindle



Note:

1. Axis exchange is a hold state, and the axis exchange state will not be restored by pressing reset or emergency stop;

2. The axes will return to their initial states after modification of parameters, or power recovery.

3. Axis exchange is the exchange of axis machine coordinates, without exchange of axis tool compensation data;

4. When the axis exchange function is used, the data for the following axes will be used:

- Machine coordinates of axes
- Diameter / Radius programming
- Rotating axis / Linear axis
- Gear ratio
- Soft limit value

Example

The A axis is the workpiece rotating axis, and by exchanging the A axis, it is possible to achieve rigid tapping of the A axis with the Z axis of channel 1 or the Z axis of channel 2; sys-parameter-5660 sets the A-axis of channel 1 to 14	
Position coordinates of channel 1: **X** (linear axis) **Y** (linear axis) **Z** (linear axis) A (workpiece rotating axis)	**Position coordinates of channel 2:** **X** (linear axis) **Y** (linear axis) **Z** (linear axis) A (virtual rotating axis)
O1000	O1001
CH1	CH2
T0001	T0001
G142// forced recovery of axis	**G142**// forced recovery of axis
G00 X0 Z100 A0 // Safety point positioning	G00 X0 Z100 A0 // Safety point positioning
G00 Z5	
M29 S300// Rigid tapping, specify the speed of A-axis to 300 r/min	
G84 Z-10 X0 R1 F1// Rigid tapping feed	
G00 X0 Z100 A0 // Safety point positioning	
G140 A14// Release A-axis of channel 1	
G04.1 P1 Q12 // channel synchronization P1	G04.1 P1 Q12 // channel synchronization P1
	G141 A14// Get the A-axis of channel 1
	M29 S300// Rigid tapping, specify the speed of A-axis to 300 r/min
	G84 Z-10 X0 R1 F1// Rigid tapping feed
	G00 X0 Z100 A0 // Safety point positioning
G04.1 P2 Q12 // channel synchronization P2	G04.1 P2 Q12 // channel synchronization P2
G142// forced recovery of axis	**G142**// forced recovery of axis
M30	M30

2.10.4 Channel subprogram call (G144/G145)

Format

G144 N_ P_ H_ L_ [Independent variable_]; (subprogram call)

N : Channel number; used to specify the channel to execute the subprogram, cannot be omitted

P : Number of called subprogram

H : H1 means waiting for subprogram execute G145, H0 means not waiting

L : Number of subprogram executions; valid only in H1, it is 1 time by default if omitted

Independent variable : Independent variable of subprogram local variables

G145; (Forcibly end G144 command's waiting for subprograms)

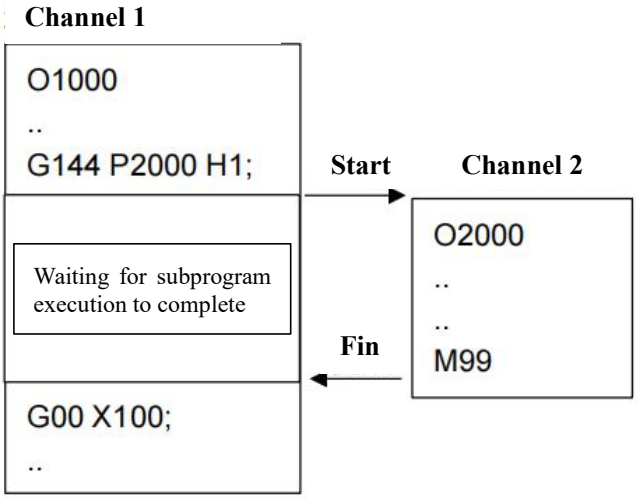
Description

This function starts a sub-channel (the called channel) by performing a G144 command on any channel. Parallel operation of the main channel and sub-channel is possible, etc., shortening processing time.

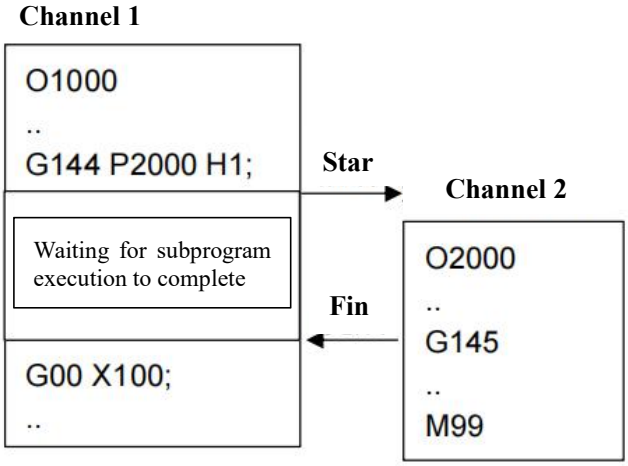
1. When H is specified as 1 or when command address H is omitted, the called channel needs to execute G145 before the main channel can execute the next program segment;

2. When H is specified as 0, the main channel does not wait for the execution of the called program to complete, but executes the next line of the program directly, and there is no need to edit G145 in the program of the called channel;

Case 1: (Only available for CNC software versions prior to 1.80.23)



Case 2:



Independent variables ↔ Local variables correspondence table:

Variable number	Independent variable address specified by G144
#1	A
#2	B
#3	C
#4	I
#5	J
#6	K
#7	D
#8	E
#9	F
#10	Not available
#11	Not available
#12	Not available
#13	M

Variable number	Independent variable address specified by G144
#14	Not available
#15	Not available
#16	Not available
#17	Q
#18	R
#19	S
#20	T
#21	U
#22	V
#23	W
#24	X
#25	Y
#26	Z

1. The addresses can be assigned in any order;
2. Addresses that do not need to be specified may be omitted;
3. Local variables are initialized at each start with an initial value of null (#0);

Example

1. Waiting form, channel 1 waiting for channel 2 to finish execution	
Channel 1	Channel 2
O1000	O2000
CH1 // Set that O1000 can only be executed in channel 1	CH2 // Set that O2000 can only be executed in channel 2
G00 X0 Z0	T0001
M03 S200	G00 X10
G144 N2 P2000 H1	..
G01 Z-10 F100	G145
M30	M99 or M30

2. No-waiting form, channel 1 waiting for channel 2 to finish execution		
Channel 1	Channel 2	Channel 3
O1000	O2000	O3000
CH1	CH2	CH3
G144 N2 P2000 H0	...	...
G144 N3 P3000 H0	...	...
...	...	...
G04.1 P1 Q123 //synchronization, ensure that the execution in all channels are completed	G04.1 P1 Q123 //synchronization, ensure that the execution in all channels are completed	G04.1 P1 Q123 //synchronization, ensure that the execution in all channels are completed
M30	M99	M99

3. Waiting form, channel 1 calls different process actions of channel 2 multiple times	
Channel 1	Channel 2
O1000	O2000
CH1 // Set that O1000 can only be executed in channel 1	CH2 // Set that O2000 can only be executed in channel 2
G00 X0 Z0	IF[#17==10] GOTO 10// judge the Q value of G144
M03 S200	IF[#17==20] GOTO 20// judge the Q value of G144
G144 N2 P2000 H1 Q10	IF[#17==30] GOTO 30// judge the Q value of G144
...	M30
G144 N2 P2000 H1 Q20	
...	N10// Action 1
G144 N2 P2000 H1 Q30	...
...	G145
M30	M99
	N20// Action 2
	...
	G145
	M99
	N30// Action 3
	...
	G145
	M99

2.10.5 Program Specified channel Operation Limit (CH)

Format

CHn; n : Channel number (positive integer)

Description:

As a multi-channel specific command, it is used to restrict the program to run only in the specified channel. If the program runs in other channels, it will generate a system alarm.

Example

Channel 1	Channel 2	Channel 3
O1001 CH1 // means it is specific for channel 1 ... M30	O1002 CH2 // means it is specific for channel 2 ... M30	O1002 CH3 // means it is specific for channel 3 ... M30

2.10.6 Channel Tracing（CHN）

Format

CHN; :Return channel number, range 1-255, corresponding to channels 1-255

Description

Obtain the channel number of the currently running program and process CHN as a variable value.

Example

Multiple channels call the same subprogram		
CH1	**CH3**	**CH3**
O1000	O2000	O3000
M98 P9020	M98 P9020	M98 P9020
...	...	...
M30	M99	M30

	O9020	
	G4I0	//synchronous
	IF[CHN= =1] G53 X0	//Executed by channel 1
	IF[CHN= =2] G53 Y0	//Executed by channel 2
	IF[CHN= =3] G53 Z0	//Executed by channel 3
	M99	

2.11 AUXILARY FUNCTIONS (M CODES)

The miscellaneous function (M function) mainly is used to control the switch-on and switch-off of machine tools electric, input status detection, and the running order of the processing process, etc. M function consists of M and two figures behind it. When the movement command and M command are in the same program segment, the movement command and M command start execution at the same time.

Example

N1 G01 X50.0 Z-50.0 F100 M05; while executing N1 segment, G01 function and M05 will be executed at the same time.

2.11.1 Standard PLC Control M Codes

The M commands in the following table are all defined by the built-in PLC, i.e., they are effective when bit 1 of sys-parameter 3000 is 0

M command	Function description	Related sys-parameters
M00	Program suspension	
M01	Condition suspension	
M02	Program stops, but does not return to the beginning	
M03	The 1st spindle rotates forward	1000~1299
M04	The 1st spindle rotates backward	
M05	Stop the 1st spindle	
M07	Enable air cooling	
M08	Enable water cooling	3340/3341
M09	Disable both air cooling and water cooling	
M10	Clamp spindle 1 chuck	1600~1699
M11	Release spindle 1 chuck	
M12	Lock spindle 1	1300~1399
M13	Unlock spindle 2	
M17	Switch speed mode of spindle 1	1000~1299
M18	Switch position mode of spindle 1	
M19	Spindle 1 orientation	
M30	End of program	
M32	Lubricating on	1800~1899
M33	Lubricating off	
M40	Set the 1st spindle to neutral gear	1025~1028 1100 1122~1161
M41	Switch the 1st gear of the 1st spindle	
M42	Switch the 2nd gear of the 1st spindle	
M43	Switch the 3rd gear of the 1st spindle	
M44	Switch the 4th gear of the 1st spindle	
M78	Feed the 1st tailstock	1700~1733
M79	Retract the 1st tailstock	

2.11.2 Input port detection (M01 L/k/i)

Format

M01 L(K)_ J_;
M01 I_ J_;

L : Input port number; its sign is ignored, and it will not be valid until the input port is connected

K : Input port number, its sign is ignored, and it will not be valid until the input port is disconnected

I : Input port number; it is signed; >0 means it will not be valid until the input port is connected, <0 means it will not be valid until the input port is disconnected.

J : The maximum waiting time (unit: seconds)

Description

M01 can pause the program to wait for the external input port signal. If a valid signal is detected, the program will continue running, otherwise, it will wait for the signal. If a valid signal is not detected within the time specified by J, the CNC will give an alarm. The program will be paused when the alarm is generated. If the input port is detected again after the alarm is given, the alarm will be automatically eliminated and the execution of the program can be continued by the start button.

Each input port has its own fixed port number within the CNC, which can be viewed via the Diagnostics button → Input Port Definition.

Example

M01 L7 //Wait for the No.7 input port to be connected

M01 I-7 //wait for input port 7 to be disconnected)

M01 K8 J5

//wait for input port 8 to be disconnected, if the signal is not disconnected within 5 seconds then an alarm is generated

M01 I8 J5

//wait for input port 8 to be connected, if the signal is not connected within 5 seconds then an alarm is generated)

2.11.3 Output port control (M20/M21)

Format

M20 K_ J_; Open the specified output port
M21 K_; Close the specified output port
K : Specify the output port number. Its sign is ignored
J : Output holding time. Its sign is ignored. If it is omitted or is 0, it keeps outputting all the time.

Description

Generally used for custom output port control.

Detailed description

Each output port has its own fixed port number within the CNC, which can be viewed via the Diagnostics button → Input Port Definition.

2.11.4 Rotating axis speed control (M26/M27)

Format①:

M26 P_ Q_; Rotating axis start
M27; Stop rotating axis
P : Speed of rotating axis 1; a positive or negative sign indicates the direction of rotation and the control axis is specified by sys-parameter-543
Q : Speed of rotating axis 2; a positive or negative sign indicates the direction of rotation and the control axis is specified by sys-parameter-5431

Format②:

M26 P_ K_; Start rotating axis
M27 K_; Stop rotating axis
P : Speed of rotating axis,a positive or negative sign indicates the direction of rotation, and the control axis is set by sys-parameter-5430
K : Axis number; multiple numbers can be specified and the control axis is set by sys-parameter-5432

Description

It is a special command to control the rotational speed of the rotating axis, and bit 1 of sys-parameter-5420 is used to switch between format ① and format ②.

2.11.5 PLC sequential control sequence (M34/M35)

Format

M34 K_ [i_] [J_] [P_] [L_] [N_] [Q_];

M35 K_ [i_] [J_] [P_] [L_] [Q_];

K : **Cannot be omitted.** K > 0: open the designated output port; K < 0: close the designated output port; K = 0: output nothing;

I : It is the detection signal input port. I>0 indicates it is valid when the signal is connected. I<0 indicates it is valid when the signal is disconnected. I=0 indicates there is no detection.

J : The maximum waiting time of the in-position input signal, unit: seconds; During this period, the port specified by K maintains its status. After it exceeds the time specified by J, the state of the output port specified by K will be reversed and the CNC alarm will be suspended.

P : When the time specified by J is up, repeatedly open the output port specified by K after a time delay specified by P. Unit: second.

L : Valid when JP is specified, number of repeat executions

N : It is the group number, up to 32 groups of continuous sequence commands;

Q : The trigger condition is aabbcc, and aabbcc is the group number. Q=0 indicates there is no waiting.

Function description:

The function is suitable for background action process, such as automatic loading in the background, which can replace traditional PLC programming and simplify programming.

Several typical usages:

1. Turn on/off the output signal
   M34 K_;
2. After outputting the signal, wait for the in-place signal to arrive
   M34 K_ I_ [J_];
3. Wait for the input signal only;
   M34 K0 I_ [J_];
4. After outputting the signal, the time delay is J
   M34 K_ J_;
5. The time delay is only J
   M34 K0 J_;
6. The number of pulse outputs is L
   M34 K_ J_ [P_] L_;
7. Repeated feeding
   M34 K_ I_ J_ [P_] L_;
8. Execute background command sequence
   M34 N1 K_ I_
   M34 N2 K_ I_ Q01 // wait for N1 to complete, and then execute N2
   M34 N3 K_ I_ Q01 // wait for N1 to complete, and then execute N3
   (note: N2 and N3 are executed at the same time)
   M34 N4 K_ I_ Q0203 // wait for N2 and N3 to complete, execute N4, output first and wait later
   M34 N5 K_ I_ J_ P_ L_ Q4 // wait for N4 to complete, execute N5 and repeat feeding
   M34 N6 K0 J_ Q5 // wait for N5 to complete, then execute N6 and delay
   M34 N6 K0 J_ Q5 // wait for N6 to complete, and then open/close the designated port

Example

O0001		
M34 N1 K49 I89 J3	Group N1: open No.49 output port, wait for No. 89 input port to be in place, and give an alarm after timeout of 3 seconds.	**Background operation**
M34 N2 K0 J1 Q1	Group N2: wait for group 1 to complete, and then execute time delay of 1 second	
M34 N3 K-49 Q2	Group N3: wait for group 2 to complete, and then close No. 49 output port	
M34 N4 K50 I90 J3 P1 L3 Q3	Group N4: wait for group 3 to complete 1. Open output port 50 and wait for input port 90 to be in position 2. If it is not in position after a timeout of 3 seconds, close output port 50 and open output port 50 again after a 1-second delay 3. Repeat up to 3 times, if input port 90 is not in position all the time, the CNC alarm will be suspended	
M34 N5 K-50 Q4	Group N5: wait for group 4 to complete, and then close No. 50 output port	
... (machining program)		**Foreground operation**
M35 K0 Q5	Wait for N5 to complete, and then the CNC will continue to execute the next program.	
...		
M30		

2.11.6 Program jump (M91)

Format

M91 N_ K (L or I)_;

N : Number of the skip target program segment;

I : Number of the input port ,unsigned; >0 means the port will not be valid until it is connected.
<0 means it will not be valid until the input port is disconnected.

Description

The program jumps and cycles are realized by judging the external PLC input signal.

Example

M91 Conditional jump	
O0020	
N10 M03 S1000	
T0101	
G0 X100	
Z0	
G1 Z-40 F100	
X120 Z-100	
X150	
M91 N10 L10	// If the external input signal (port 10) is connected, program jump will be performed Otherwise, the next line of program will be executed
M30	

2.11.7 Program jump cycle (M92)

Format

M92 N_ L_;

N : Number of the skip target program segment;

L : Number of skip repetitions. When it is <1, it will be regarded as 1.
When it is omitted, it will be an infinite skip cycle.

Description

When M92 is used to realize skip cycles, in order to ensure that the coordinate does not offset at the beginning of each cycle, it requires the command path of the cycling program segment to be a closed path. Otherwise, it will result in start point drift at each start, and it finally will move out of the workbench. When M92 N_ L_ is used to specify a limited skip cycle, execute the next program segment of M92 after l skips are executed.

M92 can be nested, and the maximum number of nesting times in each subprogram is 16.

Example

M92 infinite jump cycle	M92 limited jump cycle
O0020	O0020
N10 M03 S1000	N10 M03 S1000
T0101	T0101
G0 X100	G0 X100
Z0	Z0
G1 Z－40 F100	G1 Z－40 F100
X120 Z－100	X120 Z－100
X150	X150
M92 N10// Infinite cycle	M92 N10 L100// 100 times cycles
M30	M30;

2.11.8 Subprogram call and subprogram return (M98/M99)

Format ①:

M98 Pxxxxyyyy;

xxxx : the number of times the subprogram is repeatedly called. If omitted, it is called once.

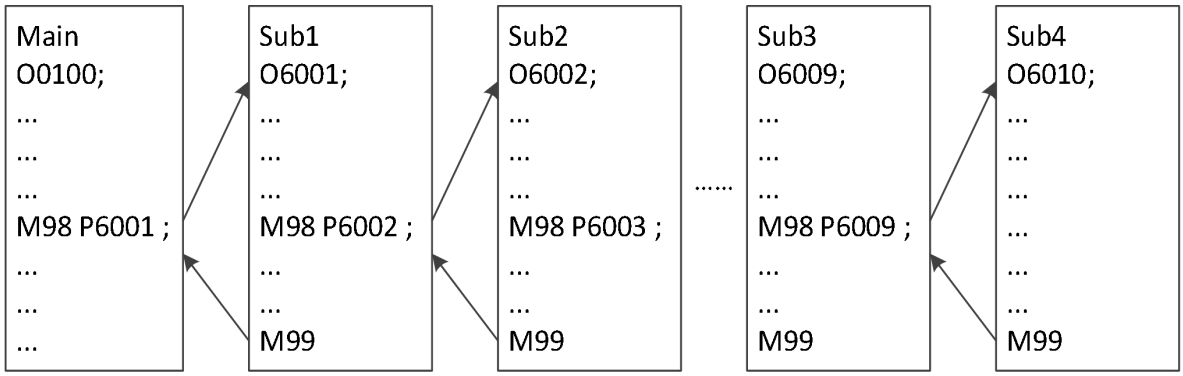
yyyy : subprogram number. When xxxx is designated as the number of repeated calls through P, yyyy must have four digits

M99 P_;

P : When the subprogram returns, it jumps to the program segment specified by P. If it is omitted, the next program segment of M98 is executed.

Description

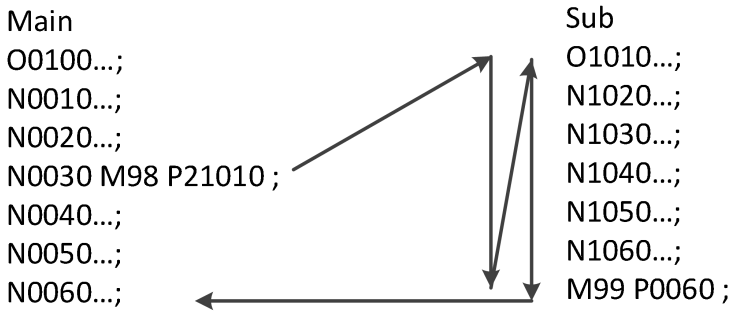
When the main program calls a subprogram, it is considered as a level 1 subprogram call, so that the subprogram call can be nested up to 16 levels. The last segment of the subprogram must be the subprogram return command M99. After executing command M99, the program will return to the next program segment of the command calling the subprogram. See the following figure:



Usage:

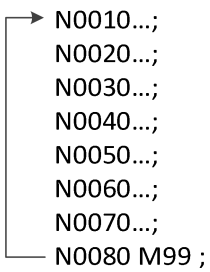
(1) Specify the sequence number for returning the target program segment of the main program;

Specify the parameter P behind M99 to specify the sequence number after the subprogram returns. When the subprogram is executed, it will not return to the program segment after the program segment called by the main program, but return to the program segment with the sequence number specified by P.



(2) Use M99 in the main program;

If M99 is executed in the main program, it will return to the beginning of the main program.



Format②(recommended):

M98 Pyyyy Q_ L_;

yyyy : Subprogram number

Q : Program segment number

L : Number of subprogram calls, can be repeated up to 999,999 times

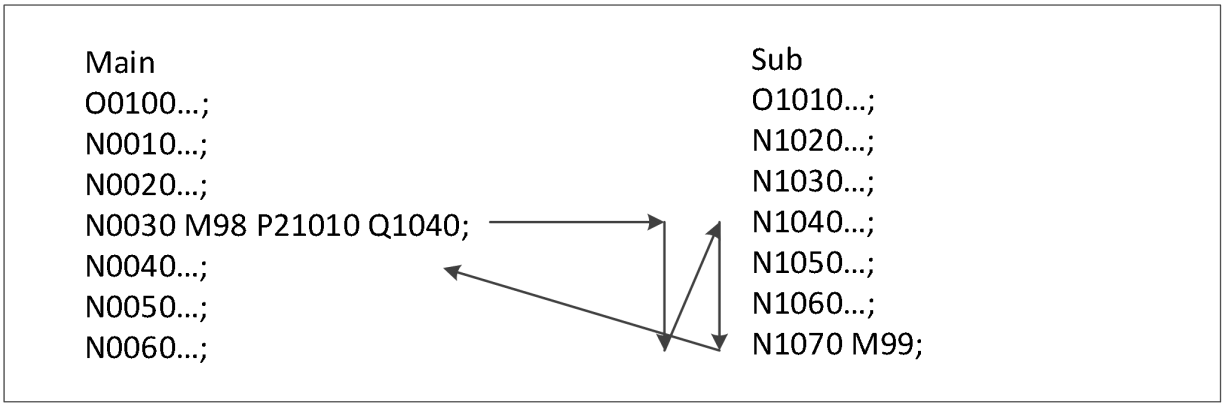
M99 P_;

P : When the subprogram returns, it jumps to the program segment specified by P.
If it is omitted, the next program segment of M98 is executed.

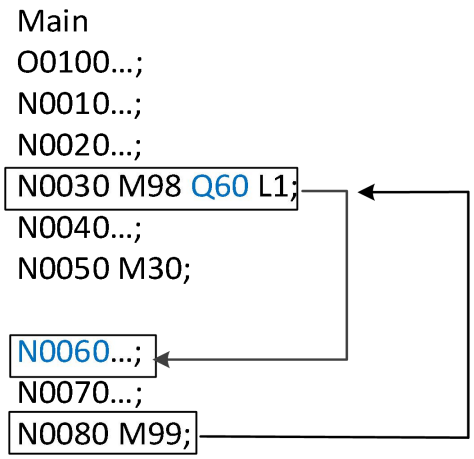
Description

M98 can not only call other subprograms, but also call the same program number as the main program. When calling the same program number of the main program, the program number P can be omitted. The execution process is as follows:

The program segment that calls the target subprogram:



Calling its own program segment:



2.11.9 Multi-spindle related commands

This system supports the multi-spindle function, which means that spindle-related commands also need to be extended.

Basic command		Extended command		
M Code	**Function**	**2nd spindle**	**3rd spindle**	**4th spindle**
M03	The 1st spindle rotates forward	M103	M203	M303
M04	The 1st spindle rotates backward	M104	M204	M304
M05	Stop the 1st spindle	M105	M205	M305
M10	Tighten the 1st chuck	M110	M210	M310
M11	Release the 1st chuck	M111	M211	M311
M12	Lock the spindle 1	M112	M212	M312
M13	Unlock spindle 1	M113	M213	M313
M17	Switch the speed mode of spindle 1	M117	M217	M317
M18	Switch the position mode of spindle 1	M118	M218	M318
M19	The 1st spindle orientation	M119	M219	M319
M40	Switch the neutral gear of spindle 1	M140	M240	M340
M41	Switch the 1st gear of the 1st spindle	M141	M241	M341
M42	Switch the 2nd gear of the 1st spindle	M142	M242	M342
M43	Switch the 3rd gear of the 1st spindle	M143	M243	M343
M44	Switch the 4th gear of the 1st spindle	M144	M244	M344
M78	Feed the 1st tailstock	M178	M278	M378
M79	Retract the 1st tailstock	M179	M279	M379

Note: The above functions are only valid when the built-in PLC is in effect (sys-parameter-3000=0).

2.12 CUSTOM MACRO

2.12.1 Macro Variables

Ordinary user processing programs directly specify G codes, displacement distance, feed speed, and etc. with numerical values. Take G01 and X100.0 for example, while using the user macro program, the numerical value can be specified through macro-variable, and the value of macro-variable is specified by the program, e.g.,

```
#101=1;
#102=100;
#103=500;
G[#101] X[#102*SIN[20]] F[#103];
```

When the user macro program specifies the macro-variable, the macro-variable is represented with the variable symbol # and the variable number after it. For example, #100

The variable number can also be represented with an expression, e.g., #[#100 + #102 + 2]

1. Classification of macro variables

#aa bbbbbb

aa : channel number; When it is 0 or omitted, it indicates the current channel;
when it is 01-99, it indicates channel 1 to channel 99.

Variable number	Meaning & Type	Description
#0	Null value (read only)	We call the state where the variable value has not been defined "null value". It has no numerical value and is not equal to 0. It cannot be written, but can be read.
		(a) **Reference variables** When referring to an undefined variable, the address itself will also be ignored. Example G00 X100 Y[#0] is equivalent to G00 X100
		(**b) Assignment, addition multiplication** When a local variable or a public variable is directly assigned to "null value", the result is also "null value". When using the "null value" for computation, its variable value is regarded as 0. Example #1=#0 ; #1 is an empty variable #2=#0+1 ; #2 为 1 #3=#0*3 ; #3 is 0
		(c) Comparison computation In the event of == and !=, "null value" and 0 are regarded as different values. In the event of >=, >, <= and <, "null value" and 0 are regarded as the same value. Example #1=#0 ; #1 is null #1==#0 ; true #1==0 ; false #1!=0 ; true #1>=0 ; true
#1~#99	Local variables (read-write)	Local variables can only be used to store data in one program, and the values and meanings of subprograms called by macro and other programs may be different. Local variables can be used to transmit independent variables. Local variables without transmitted independent variables are "null" in the initial state and can be freely used by users.
#100～#499 #500～#999	Global variable	The numerical value and meaning of global variables in different subprograms are the same. ● Variables #100~#499 are NULL when powered on, that is, empty variables; they are independent of each other for various channels; ● #500 to #999 are variables with memory, they will hold the values before power-off and are shared across channels;

Variable number	Meaning & Type	Description
#1000~#1031	Input signal (read only)	Corresponding to G0054.0 ~ G0057.7 of PLC
#1032	Input signal (read only)	Convert 32-bit binary numbers of G0054.0 to G0057.7 to decimal values
#1033	Input signal (write only)	Convert 32-bit binary numbers of G0276.0 to G00279.7 to decimal values
#1034	Input signal (write only)	Convert 32-bit binary numbers of G0280.0 to G00283.7 to decimal values
#1035	Input signal (write only)	Convert 32-bit binary numbers of G0284.0 to G00287.7 to decimal values
#1100~#1115	Output signal (write only)	Corresponding to F0054.0 ~ F0055.7 of PLC
#1132	Output signal (write only)	Convert 32-bit binary numbers of F0054.0 to F0057.7 to decimal values
#1133	Output signal (write only)	Convert 32-bit binary numbers of F0276.0 to F00279.7 to decimal values
#1134	Output signal (write only)	Convert 32-bit binary numbers of F0280.0 to F00283.7 to decimal values
#1135	Output signal (write only)	Convert 32-bit binary numbers of F0284.0 to F00287.7 to decimal values
#2001～#2064	Tool wear (read-write)	X-axis tool wear value
#2101～#2164		Z-axis tool wear value
#2401～#2449		Y-axis tool wear value
#2701～#2764	Tool offset (read-write)	X-axis tool offset value
#2801～#2864		Z-axis tool offset value
#2451～#2499		Y-axis tool offset value
#2201～#2264	Tool radius wear (read-write)	Wear value of tool radius
#2901～#2964	Tool radius offset (read-write)	Tool radius offset value
#2301～#2364	Tool nose phase (read-write)	
#3000	Macro alarm (write only)	When the CNC system executes #3000=1 (alarm content), the CNC generates an alarm, and the CNC displays the alarm content in parentheses "()".
#3003	Program single-segment suppression (write only)	=0: No suppression of program single segment function =1: Suppression of program single-segment function **Note:** The value of #3003 is forced back to 0 when reset
#3004	Invalid feed override (write only)	=0: valid feed override =2: invalid feed override **Note:** The value of #3004 is forced back to 0 when reset
#3101 or Pi	Π (read only)	π: 3.14159265358979323846
#3102	e (read only)	Basic number of natural logarithm, e = 2.7182818284590452353 6
#3901	Piece count (read-write)	Piece count for position interface
#3902	Target piece count (read-write)	Target piece count for position interface
#4000	Main program number (read only)	Number of the main program executed by the CNC system
#4001~#4030	G modal (read only)	G code modal information: 1 group ~ 30 groups
#4108	E modal (read only)	Modal information of the chamfering speed specified by E during automatic chamfering
#4109	F modal (read only)	Modal information of feed speed F
#4120	T modal (read only)	Modal information of tool change command
#4201~#4230	Current G modal (read only)	G code modal information of current execution segment: 1 group ~ 30 groups
#4308	Current E modal (read only)	Program segment currently executed; the chamfering speed modal information specified by E during automatic chamfering
#4309	Current F modal (read only)	Program segment currently executed; modal information of feed speed F
#4320	Current T modal (read only)	Program segment currently executed; modal information of tool change command

Variable number	Meaning & Type	Description
#5001~#5020	Workpiece (absolute) coordinates (read only)	Absolute coordinates of the program segment end point of the 1st to 20th axes in the channel, which can be read when the axis moves
#5021~#5040	Machine tool coordinates (read only)	Machine coordinates of the program segment end point of the 1st to 20th axes in the channel, which cannot be read when the axis moves
#5041~#5060	Workpiece (absolute) coordinates (read only)	Machine coordinates of the program segment end point of the 1st to 20th axes in the channel, which cannot be read when the axis moves
#5061~#5080	Workpiece (absolute) coordinates (read only)	Workpiece coordinates when skipping G31 from Axis 1 to Axis 20 in the channel, which can be read when the axis moves
#5121~#5140	Handwheel interruption compensation amount (read only)	Handwheel interruption compensation amount for the 1st to 20th axes in the channel, which can be read when the axis moves
#5181~#5200	Remaining movement amount (read only)	Remaining movement amount of the 1st to 20th axes in the channel, which cannot be read when the axis moves
#5221~#5240	G54 coordinate system offset (read-write)	Coordinate offset values of the 1st to 20th axes in the channel in the G54 coordinate system
#5241~#5260	G55 coordinate system offset (read-write)	Coordinate offset values of the 1st to 20th axes in the channel in the G55 coordinate system
#5261~#5280	G56 coordinate system offset (read-write)	Coordinate offset values of the 1st to 20th axes in the channel in the G56 coordinate system
#5281~#5300	G57 coordinate system offset (read-write)	Coordinate offset values of the 1st to 20th axes in the channel in the G57 coordinate system
#5301~#5320	G58 coordinate system offset (read-write)	Coordinate offset values of the 1st to 20th axes in the channel in the G58 coordinate system
#5321~#5340	G59 coordinate system offset (read-write)	Coordinate offset values of the 1st to 20th axes in the channel in the G59 coordinate system
#7001~#7020	G54 P1 coordinate system offset (read only)	Coordinate offset values of the 1st to 20th axes in the channel in the G54 P1 coordinate system
#7021~#7040	G54 P2 coordinate system offset (read only)	Coordinate offset values of the 1st to 20th axes in the channel in the G54 P2 coordinate system
...	...	...
#7981~#7999	G54 P50 coordinate system offset (read only)	Coordinate offset values of the 1st to 19th axes in the channel in the G54 P50 coordinate system

Extended macro variables:

Variable number	Meaning & Type	Description
#10001～#10064	Tool wear (read-write)	X-axis tool wear value
#11001～#11064		Z-axis tool wear value
#14001～#14064		Y-axis tool wear value
#15001～#15064	Tool offset (read-write)	X-axis tool offset value
#16001～#16064		Z-axis tool offset value
#19001～#19064		Y-axis tool offset value
#12001～#12064	Tool radius wear (read-write)	
#17001～#17064	Tool radius offset (read-write)	
#13001～#13064	Tool nose phase (read-write)	
#20000～#20999	Extended global variables (read-write)	The function and usage are the same as those of #100 to #499, but shared across various channels.

2. Format of macro variables

- While using macro variables or variable expressions in CNC program segment, "["and"]" needs to be added, and the detailed format is described as below:
  (1) The format of macro variable used is: [# variable number];
  (2) The format of macro expression used is: [expression];
  (3) Constants and macro variables connected by operators constitute expressions;

Example

G01 X[#101+#102] F[#103]

G01 X[100*COS[50]＋20]

- Rounding of the value of reference variable can be automatically carried out according to the minimum setting unit of address

Example

When G00 X[#101] is executed in units of 1/1000 mm, the CNC system assigns 12.3455 to variable #101 and the actual command is G00 X12.345.

- To change the sign of the reference variable value, the negative sign "－" shall be placed in front of #

e.g., G00 X[－#101]

Note: While using the macro expression, please take care of the operator precedence, brackets can be used to change the order of operation, if necessary.

3. Assignment of macro variables

Assigning the value of constant or expression to macro variable is called assignment.

Format:

```
# variable number = constant
# variable number = # variable number
# variable number = expression
```

Example

O1000	
#101=3	// The value of #101 becomes 3
#102=#101+1	// The value of #102 becomes 3+1=4
#103=SQRT[#101*#101+#102*#102]	// The value of #103 becomes $\sqrt{3*3+4*4}$ = 5
G00 X0	// X-axis positioning origin
G01 X[#103] F100	// X-axis is oriented to 5 mm
M30	

2.12.2 Logical and Mathematical Operations

Operational characters listed in the table can be executed in variables or constants, and both sides of operational characters can be constants, variables or expressions consisting of functions or operational characters, i.e., variables #j and #k can be constants, variables or expressions. I can be a constant, a variable or an expression.

Type	Complete format	Function	Remarks
Assignment	#i = #j	Definition or replacement of variables	
Basic computation	#i = #j + #k	Addition	
	#i = #j - #k	Subtraction	
	#i = #j * #k	Multiplication	
	#i = #j / #k	Division	
	#i = #j MOD #k	Remainder (modulo)	
Logical operation	#i = #j && #k or #i = #j AND #k	Logical AND	Logical operation, mainly used for conditional judgment
	#i = #j || #k or #i = #j OR #k	Logical OR	
	#i = ! # j	Logical negation	
Bit operation	#i = #j | #k	or	Bit operators can also be used as logical judgments
	#i = #j & #k	and	
	#i = ~#j	not	
	#i = #j ^ #k or #i = #j XOR #k	xor	
Comparison	#j == #k or #j EQ #k	equal to	Relational operators are mostly used in conditional judgments IF and circular judgments WHILE
	#j != #k or #j NE #k	not equal to	
	#j > #k or #j GT #k	more than	
	#j >= #k or #j GE #k	more than or equal to	
	#j < # k or #j LT #k	less than	
	#j <= #k or #j LE #k	less than or equal to	

Type	Complete format	Function	Remarks
Trigonometric function	#i = SIN[#j]	Sine	Unit: degree, 90°30′ is represented to be 90.5°
	#i = COS[#j]	Cosine	
	#i = TAN[#j]	Tangent	
	#i = ASIN[#j]	Arc sine	
	#i = ACOS[#j]	Arc cosine	
	#i = ATAN[#j, #k]	Arc tangent	
Other functions	#i = SQRT[#j]	Square root	After rounding, three digits are reserved after the decimal point, and the fourth digit is rounded off ROUND2 can specify the exact number of decimal places INT rounded to an integer
	#i = ABS[#j]	Absolute value	
	#i = ROUND[#j]	rounded to an integer	
	#i = ROUND2[#j, #k]	Rounding to the nearest integer (exact)	
	#i = INT[#j]	rounded to an integer	
	#i = FIX[#j]	Round down	
	#i = FUP[#j]	Round up	
	#i = LN[#j]	Napierian logarithm	
	#i = EXP[#j]	Exponential function	
	#i = POW[#j, #k]	Power	
Constant	PI	Pi	Pi constant π

Functional items	Description	
#i = ASIN[#j]	–1 < #j < 1 -90°< #i <90°	
#i = ACOS[#j]	–1 < #j < 1 180° < #i < 0°	
Round up Round down Rounding	If the decimal part of the operand is not 0, the absolute value of the integer generated after up-rounding operation is greater than the integer part of the original number; if the integer part is smaller than the original number, it is down-rounding; if the absolute value of the integer generated after rounding operation is greater than or equal to the integer part of the original number, be careful when dealing with negative numbers.	
	Provided that #1=1.2 and #2= -1.2 2.0 is assigned to #3 when executing #3=FUP[#1] -2.0 is assigned to #3 when executing #3=FUP[#2] 1.0 is assigned to #3 when executing #3=FUP[#1] -1.0 is assigned to #3 when executing #3=FUP[#2]	Provided that #1=1.5 and #2= -1.5 2.0 is assigned to #3 while executing #3=INT[#1] -2.0 is assigned to #3 while executing #3=INT[#2]
Operator precedence	(1) Multiplication and division (*, /) (2) Addition and subtraction (+, –) (3) Relational operations (==, !=, >, etc.) (4) Logical operations (| |, &&, etc.) (5) Bit operation (|, ^, &)	
The use of []	In the macro expression, brackets can be used to change the order of operation For Example # 1 = 3 * 20-10, then the value of # 1 is 50 # 1 = 3 * [20-10], then the value of # 1 is 30 the CNC will automatically change the order of operation according to operator precedence Example In case of #1=10+2*10, #1＝30 When the bracket or operator priority in macro expression changes too many times (including less than 10 times), the CNC will prompt an error.	

2.12.3 Dealing with Macro Statements

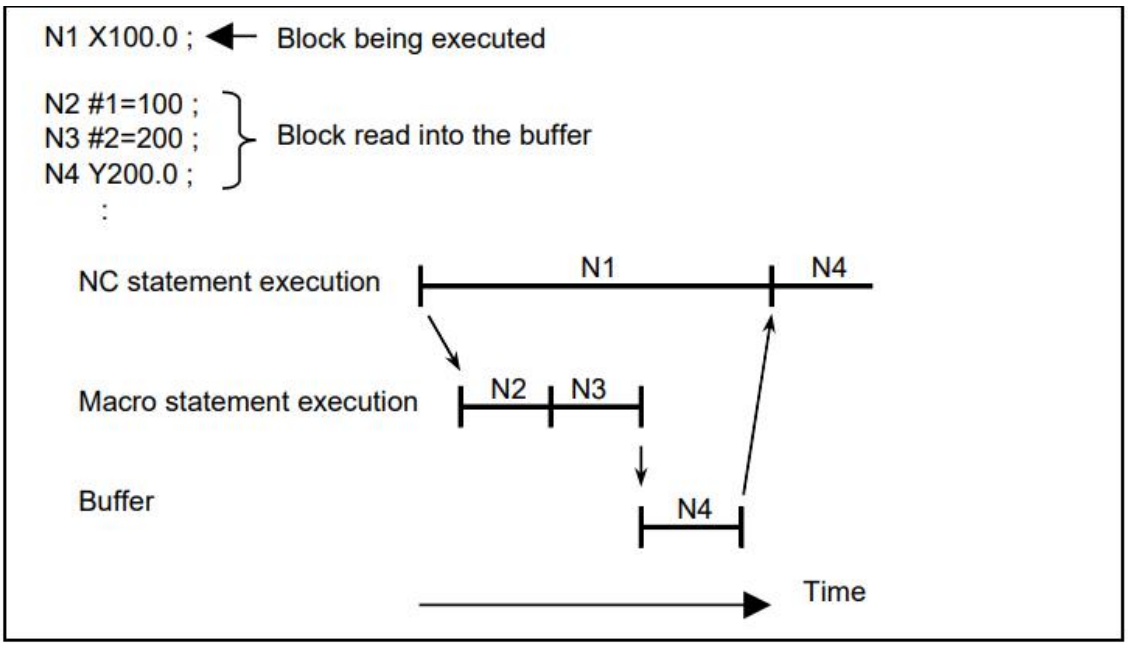
For smooth machining, the CNC performs the following NC statement pre-reading. Call it a buffer. For example, up to 200 program segments can be buffered for the pre-reading based on AI profile control.

In addition, in the tool path compensation (G41, G42) mode, the NC statement after pre-reading at least 3 program segments performs the intersection point calculation.

However, arithmetic equations and conditional transfer macro statements can be processed once they are buffered (i.e., read into the buffer). Therefore, the timing of executing macro statements does not necessarily follow the specified order.

On the contrary, pre-reading is not performed in the program segment that is designated to control the G-code set in M00, M01, M02, M30, M91, or to control the buffer of G04.1, G04I0, G31, G53, etc. Therefore, you can ensure that the macro statements that follow are not executed until you have finished executing these M codes and G codes.

Description:



Note:

When you want to execute a macro statement just after executing the previous NC statement, specify the M code and G code used to control the buffer before the macro statement. In particular, during reading or writing of system variables used to control signals, coordinate values, or compensation values, the timing of NC statement execution may result in different system variables, so specify the M codes and G codes for buffer suppression as needed.

2.12.4 Jump (GOTO/GOTOB/GOTOF)

Format

GOTO/GOTOF/GOTOB n;	
GOTO	: Searching is carried out first towards the end of the program and then from the beginning of the program
GOTOB	: Search from the beginning of the program
GOTOF	: Search backwards
n	: Program segment number

Description

The commands GOTOF, GOTOB, GOTO are used to jump from other locations to the target segment number within the same program. The program processing is then continued by this command, which follows directly after the jump marker. So it is possible to implement branches within the program.

If a jump condition exists before the jump command (IF ...), the program jump is performed only if the jump condition is satisfied.

Example

1. Infinite cycle	
O1000	
T0101	
N10	
...(machining program)	
GOTO 10	
M30	

2. Conditional jump	
O1000	
T0101	
G4I0	//Prevent program pre-reading
#1=RDDI[1]	// Read input port 1 of the CNC to determine if it is connected
IF[#1==1]**GOTO 10**	//Determine whether the input port is connected; if it is connected, then it jumps to the end of the program to end the program.
...(machining program)	
N10	
M30	

2.12.5 Conditional Judgment (IF Statement)

Format ①:

IF [conditional expression] THEN <#n=expression>;
#n : Macro variables, e.g., #100

Format ②:

IF [condition expression] GOTO n;
n : program segment number

Format ③:

IF [conditional expression] <expression / G command / M command>;

Format ④:

IF [conditional expression 1]
<Program segment>
<Program segment>
...
ELIF [conditional expression 2]
<Program segment>
<Program segment>
...
ELSE
<Program segment>
<Program segment>
...
ENDIF

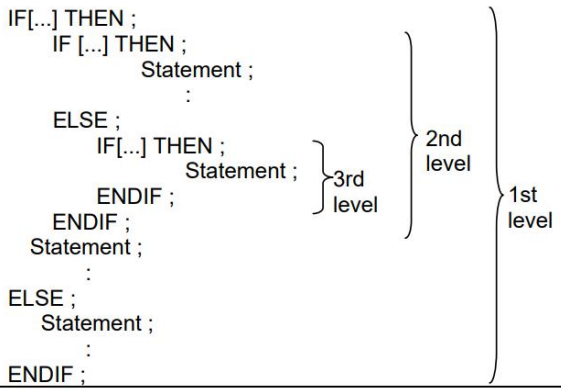
IF	: Conditional judgment start statement
ELIF	: else if (can be omitted), is equivalent to IF
ELSE	: Otherwise (can be omitted); when both IF and ELIF are not satisfied, execute the statement after ELSE
ENDIF	: Conditional judgment end statement

Description

When the condition expression after IF is true, the CNC will execute the sentence after "IF [condition expression]", otherwise, the next program segment will be executed.

Format ④, ELIF and ELSEIF are equivalent. IF and ENDIF are required in this format, and ELIF and ELSE can be added as needed.

EXAMPLE
The following is an example of 3 levels of nesting, but up to 10 levels of nesting are allowed.



Example

1.Infinite loop	
#10=0	//Counter initialization
N10	
G00 X[#10*20]	// Perform punching actions at positions 0 mm, 20 mm, 40 mm, 60 mm, 80 mm, and 100 mm of X coordinate, respectively.
(Drilling action program...)	
#10=#10+1	// Counter + 1
IF[#10<=5]**GOTO 10**	// When the current counter is less than or equal to 5, jump and repeat the cycle processing

2. Branch jump	
O1000	
IF[#500==1]**GOTO 300**	// Call different machining programs by modifying the value of macro variable #500
IF[#500==2]**GOTO 200**	
IF[#500==3]**GOTO 100**	
M30	
N100	// Branch 3
(...)	
M30	
N200	// Branch 2
(...)	
M30	
N300	// Branch 1
(...)	
M30	

3. Multi-conditional judgment	
O1000	
IF[#500==1]	// Judgment 1: Judge whether #500 is equal to 1, if the condition is met,execute (...)
(...)	
ELIF[#500==2]	// Judgment 2: Judge whether #500 is equal to 2, if the condition is met,execute (...)
(...)	
ELIF[#500==3]	// Judgment 3: Judge whether #500 is equal to 3, if the condition is met,execute (...)
(...)	
ELSE	// When judgments 1/2/3 are not satisfied, execute the statement after ELSE
#3000=1(#500 setting error)	//When the current #500 parameter setting does not meet judgment 1/2/3, a macro alarm will be generated
ENDIF	
M30	

2.12.6 Repetition (WHILE Statement)

Format

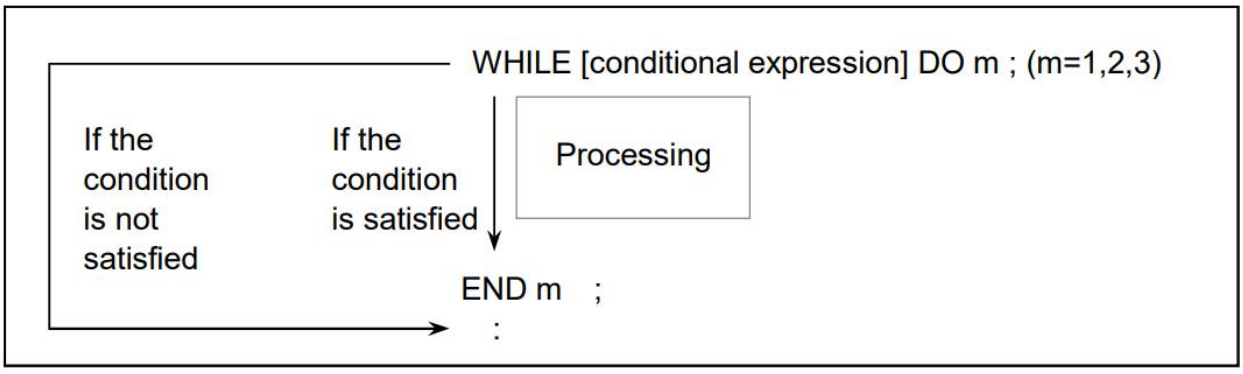
WHILE [conditional expression] Do m
<Program segment>
<Program segment>
...
END m
m : sign number (positive integer); the m of each level must be different and m cannot be crossed in use

CONTINUE; Skip the remaining statements and immediately execute the next cycle

BREAK; exit the cycle

Description

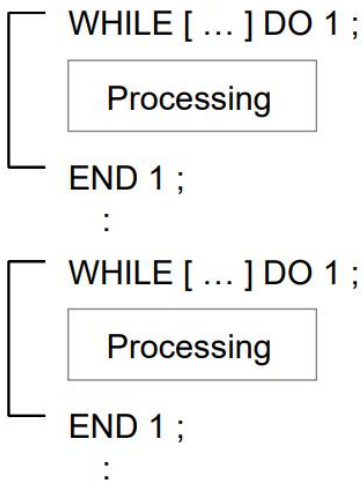
Specify a conditional expression after WHILE. When the specified conditional expression is satisfied, the programs between DO to END will be executed. When the specified conditional expression is not satisfied, enter the program segments behind END.



1. Some use cases:

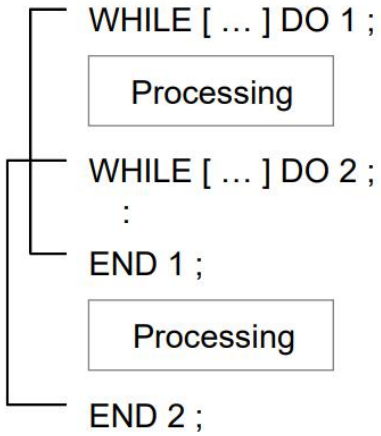
√

1. The identification numbers (1 to 3) can be used as many times as required.



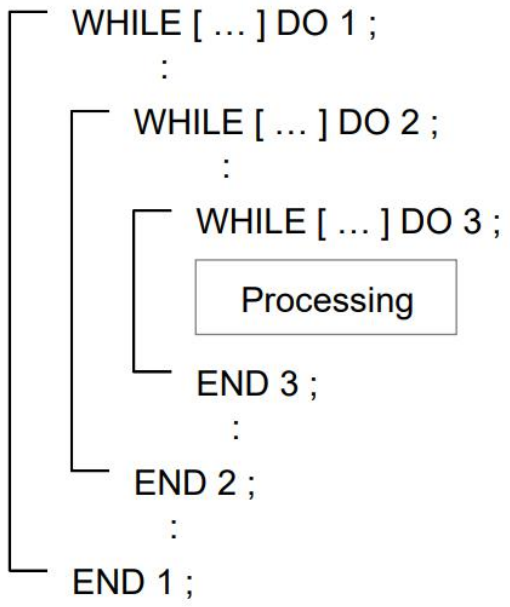
×

2. DO ranges cannot overlap.



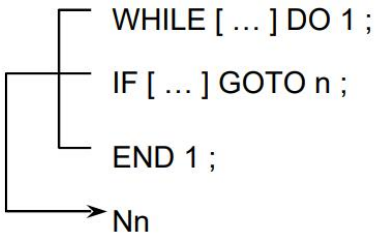
√

3. DO loops can be nested to a maximum depth of three levels.



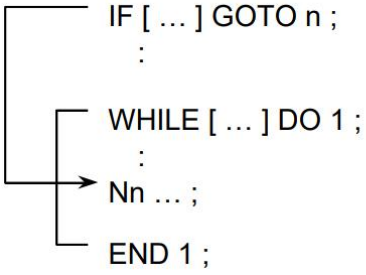
√

4. Control can be transferred to the outside of a loop.



×

5. Branches cannot be made to a location within a loop.



2. WHILE is nested up to 5 levels and GOTO can be used to jump out of the cycle and end the cycle;
3. CONTINUE can be abbreviated to CONT and BREAK can be abbreviated to BRK;

Example

1. Infinite cycle, conditional jump-out	
WHILE[1]**DO1**	
G01 W-10 F100	// Incremental feed
G4I0	
IF[RDDI[5] == 1] **BREAK**	// If the input port 5 is connected, it will jump out the cycle
END1	

1. See the following program example. Calculate the sum of 1 to 10.	
O0001	
#1=0	
#2=1	
WHILE[#2 <= 10]**DO 1**	
#1=#1＋#2	
#2=#2+1	
END 1	
M30	

2.12.7 Read Data (RD)

Format

Macro variable = RDDI [port number of input port]
Macro variable = RDDO[port number of output port]

Macro variable = RDX [high bit of PLC address X, low bit of PLC address X]
Macro variable = RDX8 [high bit of PLC address X]
Macro variable = RDX16 [high bit of PLC address X]
Macro variable = RDX32 [high bit of PLC address X]

Macro variable = RDY [high bit of PLC address Y, low bit of PLC address Y]
Macro variable = RDY8 [high bit of PLC address Y]
Macro variable = RDY16 [high bit of PLC address Y]

Macro variable = RDR [high bit of PLC address R, low bit of PLC address R]
Macro variable = RDR8 [high bit of PLC address R]
Macro variable = RDR16 [high bit of PLC address R]
Macro variable = RDR32 [high bit of PLC address R]

Macro variable = RDF [high bit of PLC address F, low bit of PLC address F]

Macro variable = RDG [high bit of PLC address G, low bit of PLC address G]

Macro variable = RDDGN [system diagnostic number, diagnostic data serial number]

Description

Through the above commands, the values of various system data are obtained in the form of macro variable assignments to complete the conditional judgments of some complex functions.

Detailed description

1. A synchronization (G04I0) is to be performed before using the read command to prevent data read errors due to system pre-reading;

2. Difference between RDX/RDX8/RDX16/RDX32:

	Description
RDX	Read data from PLC address X by bit **Example** #100=RDX[3,2] means assign the signal state of X3.2 to #100 Assuming X3.2 indicates connection state, #100 is assigned with 1
RDX8	Read the data of PLC address X by 8 bits, convert the 8-bit binary value to a decimal value, and assign it to the macro variable **Example** #100=RDX8[3] means to assign the signal states of X3.0~X3.7 to #100 Assuming X3.0, X3.1, and X3.3 indicate connection states, #100=$1\times 2^0+1\times 2^1+1\times 2^3$=11
RDX16	Read the data of PLC address X by 16 bits, convert the 16-bit binary value to a decimal value, and assign it to the macro variable **Example** #100=RDX16[3] means to assign the signal states of X3.0~X3.7 and X4.0~X4.7 to #100 Assuming X3.0 and X4.0 indicate connection, #100=$1\times 2^0+1\times 2^8$=257
RDX32	Read the data of PLC address X by 32 bits, convert the 32-bit binary value to a decimal value, and assign it to the macro variable

2.12.8 Write Data (WR)

Format

WRDO [port number of output port]

WRY [high bit of PLC address Y, low bit of PLC address Y, value]
WRY8 [high bit of PLC address Y, value]
WRY16 [high bit of PLC address Y, value]

WRR [high bit of PLC address R, low bit of PLC address R, value]
WRR8 [high bit of PLC address R, value]
WRR16 [high bit of PLC address R, value]
WRR32 [high bit of PLC address R, value]

Description

Generally used for decimal to binary conversions.

Detailed description

1. Before the command is used, a synchronization G04I0 needs to be executed to prevent the command from being written early due to system pre-reading;
2. Difference between WRR/WRR8/WRR16/WRR32:

	Description
WRR	Write to PLC address R by bit, the maximum value written is 1 **Example** WRR[3, 2, 1] means to set the coil of R3.2 to 1
WRR8	Convert decimal values to 8-bit binary numbers and write them to PLC address R. The maximum value written is 255. **Example** WRR8[3, 64] means to convert decimal 64 data to binary data and write it to R3.0~R3.7
WRR16	Convert decimal values to 16-bit binary numbers and write them to PLC address R. The maximum value written is 65535. **Example** WRR16[3, 64] means to convert decimal 64 data to binary data and write it to R3.0~R4.7
WRR32	Convert decimal values to 32-bit binary numbers and write them to PLC address R. The maximum value written is [2^{32} -1] **Example** WRR32[3, 64] means to convert decimal 64 data to binary data and write it to R3.0~R6.7

2.12.9 File Output (POPEN/DPRNT/PCLOS)

Format

POPEN [NC program number]; Open file

DPRNT[a #b[d] ...]; write data

a : Characters, such as G01, M01 and other functional characters

#b : Macro variable

[d] : Specify the data format of the macro variable, with d indicating the number of significant digits after the decimal point

PCLOS [NC program number]; Close file

Example

Execute the program named O1000	
O1000	
POPEN[6000]	// Create and open the program named O6000
#1=123.456	// Set variable
#2=11.22	// Set variable
DPRNT[G01X #1[3] Y #2[1] F100] 1 2 3 4 5	Write data content "G01 X123.456 Y11.2" //1: Character, value G01X //2: Variable [keep 3 decimal places], value 123.456 //3: Character, value Y //4: Variable [keep 1 decimal places], value 11.2 //5: Character, value F100
DPRNT[M30]	// Write data content "M30"
PCLOS[6000]	// Close the O6000 file
M30	

The contents of the CNC-generated program named O6000 are as follows:	
O6000	
G01 X123.5 Y11.2 F100	
M30	

2.12.10 Read sys-parameter (PRM)

Format

PRM[Par, n]	
Par	: Sys-parameter number
n	: The nth parameters in the parameter number

Description

Read sys-parameters, such as axis name definition, axis type, 2nd reference point coordinate value setting, etc., to facilitate custom application of user macro programs.

Example

Automatic determination of the name of the 1st axis in the channel by sys-parameter-0120	
O9020	
#1=100	
IF[**PRM[12,1]**==1] G00 X#1	// Judged to be X-axis, perform X-axis positioning
IF[**PRM[12,1]**==2] G00 Y#1	// Judged to be Y-axis, perform Y-axis positioning
IF[**PRM[12,1]**==3] G00 Z#1	// Judged to be Z-axis, perform Z-axis positioning
IF[**PRM[12,1]**==4] G00 A#1	// Judged to be A-axis, perform A-axis positioning
IF[**PRM[12,1]**==5] G00 B#1	// Judged to be B-axis, perform B-axis positioning
M30	

2.12.11 Macro Program Call (G65)

Format

G65 P_ L_ <argument specification>;

P : Number of called program;

L : Number of repetitions, it is1 when omitted

argument : Data passed to macro program. Its value can be substituted into corresponding local variable (optional).

Description

Through G65 macro call, you can transfer data (arguments) when calling the user macro program, and the user macro program can realize different processing by judging the data (variables).

1. G65 may not be nested more than 5 levels and must be specified before any independent variable;
2. When the arguments is specified, you can omit the address that is unnecessary to specify. The local variable corresponding to the omitted address is set to null;
3. There are two ways to specify arguments:

Argument specification I					
Address	**Variable number**	**Address**	**Variable number**	**Address**	**Variable number**
A	#1	J	#5	S	#19
B	#2	K	#6	T	#20
C	#3	L	Number of repetitions	U	#21
D	#7	M	#13	V	#22
E	#8	N	#14	W	#23
F	#9	P	Not available	X	#24
H	#11	Q	#17	Y	#25
I	#4	R	#18	Z	#26

- Addresses G, L, N, O, P cannot be used as arguments, or treated as program segment numbers if N is before G65;
- It does not need to be specified in alphabetical order, but in word address format;

Argument specification II					
Address	**Variable number**	**Address**	**Variable number**	**Address**	**Variable number**
A	#1	I4	#13	I8	#25
B	#2	J4	#14	J8	#26
C	#3	K4	#15	K8	#27
I1	#4	I5	#16	I9	#28
J1	#5	J5	#17	J9	#29
K1	#6	K5	#18	K9	#30
I2	#7	I6	#19	I10	#31
J2	#8	J6	#20		
K2	#9	K6	#21		
I3	#10	I7	#22		
J3	#11	J7	#23		
K3	#12	K7	#24		

- For type II designation method, the addresses A, B and C are only used once, and I, J and K are used as a group, which can be designated repeatedly for up to 10 times.
- The subscript (indicating the order specified by the arguments) of I, J and K are not written in the actual program.

- Call nesting

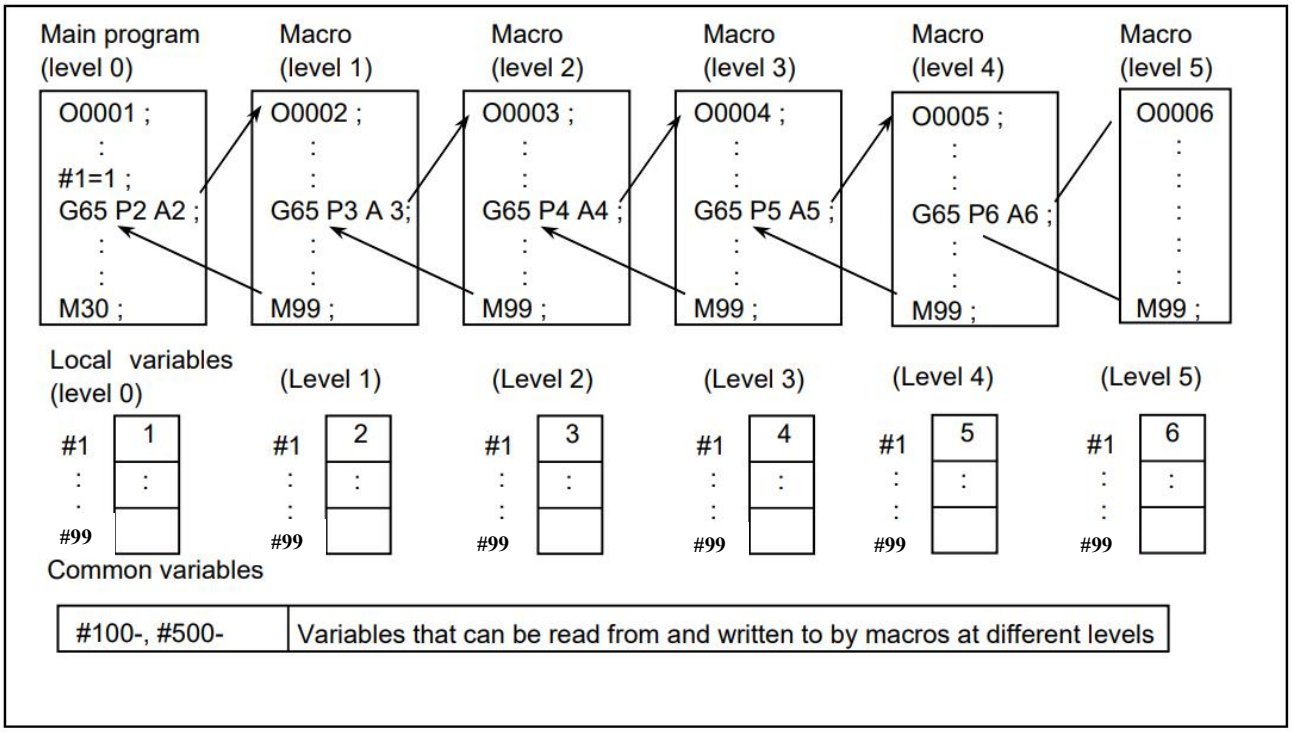
Macro calls can be nested to a depth of up to five levels including simple calls (G65) and modal calls(G66/G66.1). Subprogram calls can be nested to a depth of up to 15 levels including macro calls.A macro program can also be called during DI operation in the same way.

- Local variable levels

Local variables from level 0 to 5 are provided for nesting.

The level of the main program is 0.Each time a macro is called (with G66, G66), the local variable level is incremented by one. The values of the local variables at the previous level are saved in the CNC.

When M99 is executed in a macro program, control returns to the calling program. At that time, the local variable level is decremented by one, the values of the local variables saved when the macro was called are restored.



Example

Main program	Subprogram
O1000 T0101 M03 S1000 G00 X0 Z0 **G65 P9200 L1 A10 B20 C30** // Call the macro program, set #1=10 #2=20 #3=30 M30	O9200 G01 X**#1** F500 // X to 10 mm G01 W-10 F200 // Z Drilling G00 Z0 // Tool retracting G01 X**#2** F500 // X move to 20 mm G01 W-10 F200 // Z Drilling G00 Z0 // Tool retracting G01 X**#3** F500 // X to 30 mm G01 W-10 F200 // Z Drilling G00 Z0 // Tool retracting M99 // Return to the main program

2.12.12 Modal Call:Call After the Move Command (G66/G67)

Format

G66 P_ L_ <argument specification>**;**
P : Number of called program;
L : Number of repetitions, it is1 when omitted
argument : Data passed to macro program. Its value can be substituted into corresponding local variable (optional).

G67; Cancel the modal call

Description

If a program segment with a movement command between G66 and G67 is specified, the subprogram of the specified user macro program is executed after the movement command is executed. The number of times the user macro program subprogram is executed is specified in L. The argument is the same as that of G65.

Detailed description

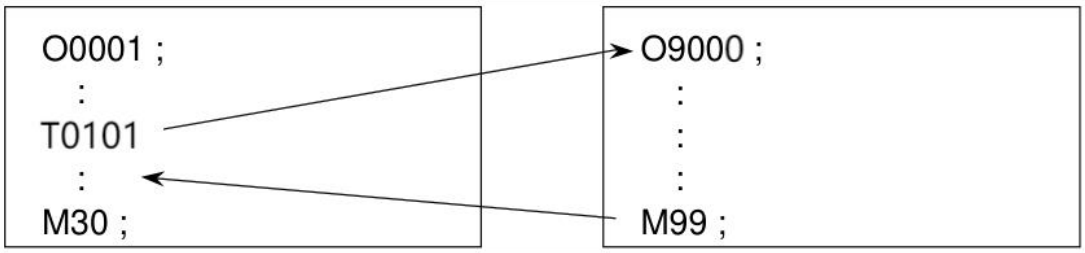
1. When the G66 command is entered, the specified user macro program subprogram is called after the execution of the program segment with the movement command before the G67 (cancel) command is entered;
2. Commands G66,G67 must appear in pairs in the same program;
3. The program segment of G66 does not make macro calls, but local variables (arguments) are set;
4. When G67 is specified, the modal macro program call will not be executed in the next program segment.
5. No macro calls are made in program segments without axis movement commands;

Example

Main program	Subprogram
O0001 G00 X100.0 Y50.0 **G66 P9600 Z-20.0 R5.0 F500** X20.0 Y20.0// drilling X50.0// drilling Y50.0// drilling X70.0 Y80.0// drilling **G67** X100.0 Y50.0 // without drilling M30	O9600 (drilling subprogram) #1=#4001// save G00/G01 #4=#4109// save cutting feed speed #5=#5003// Z coordinate is stored at the beginning of drilling G00 Z#18// locate to point R G01 Z#26 F#9// locate to point Z G00 Z#18// locate to point R G00 Z#5// locate to the starting point G#1 F#4// restore modal information M99

2.12.13 T Code Subprogram Call

Format



Description

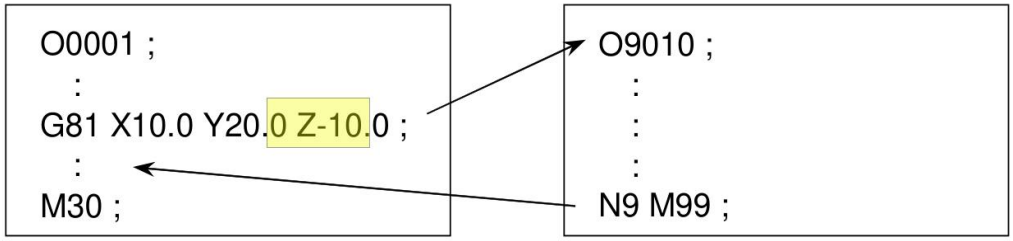
By setting the target program number in the parameters in advance, the subprogram can be called when the CNC executes the T command, and the calling method is similar to M98, which is generally used for tool change control of tool magazine and turret.

Detailed description

1. Bit 1 of sys-parameter-9000 is used to enable T command to call subprograms;
2. sys-parameter-9010 sets the number of the subprogram to be called;
3. When a subprogram is called by the execution of a T command, #149 is assigned a value equal to the value of the executed T command. Take T0202 as an example, the value of #149 at the time of the call would be 202;

2.12.14 G Code Macro Call (Usually)

Format



Description

The macro program can be called by setting a G code number in the parameter in advance to call the macro program, and the call method is similar to the G65.

Detailed description

1. First set the CNC parameter-9020, which sets the number of programs that can execute the call;
2. First set sys-parameter-9021, which sets the opening program number for execution of the macro call;
3. Correspondence between parameter number and program number:

sys-parameter number	Corresponding G-code	Corresponding program number
9022	G [Value set by 9022]	O [Value set by 9021 + 0]
9023	G [Value set by 9023]	O [Value set by 9021 + 1]
9024	G [Value set by 9024]	O [Value set by 9021 + 2]
9025	G [Value set by 9025]	O [Value set by 9021 + 3]
9026	G [Value set by 9026]	O [Value set by 9021 + 4]
9027	G [Value set by 9027]	O [Value set by 9021 + 5]
9028	G [Value set by 9028]	O [Value set by 9021 + 6]
9029	G [Value set by 9029]	O [Value set by 9021 + 7]
9030	G [Value set by 9030]	O [Value set by 9021 + 8]
9031	G [Value set by 9031]	O [Value set by 9021 + 9]
9032	G [Value set by 9032]	O [Value set by 9021 + 10]
9033	G [Value set by 9033]	O [Value set by 9021 + 11]

2.12.15 G Code Macro Call (Interval)

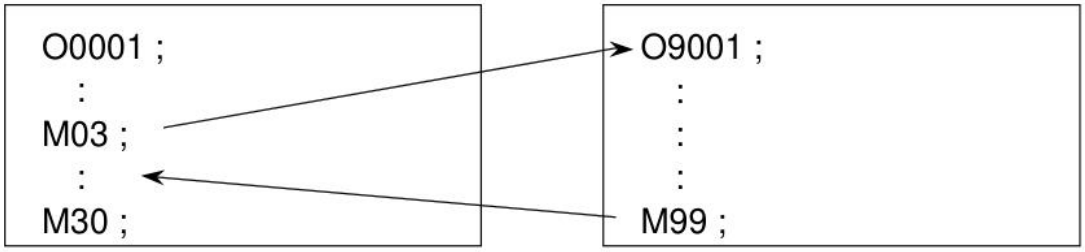
You can define multiple G-code macro calls by setting the G-code number used in the macro call, the opening program number of the call, and the number of definitions in advance.

Example

sys-parameter-9034 is set to 99, sys-parameter-9035 is set to 9200, and sys-parameter-9036 sets the G-code start number to 200	
M Code	**Corresponding program number**
G200	O9200
G201	O9201
G202	O9201
G203	O9201
...	...
G299	O9299

2.12.16 M Code Subprogram Call (Usually)

Format



Description

Macro programs can be called by setting in advance an M code number in the parameter for calling subprograms (macros) in the same way as subprogram call (M98).

Detailed description

1. First set the CNC parameter-9040, which sets the number of programs that can execute the call;
2. First set sys-parameter-9041, which sets the opening program number for execution of the subprogram call;
3. Correspondence between parameter number and program number:

sys-parameter number	Corresponding M code	Corresponding program number
9042	M [Value set by 9042]	O [Value set by 9041 +0]
9043	M [Value set by 9043]	O [Value set by 9041 +1]
9044	M [Value set by 9044]	O [Value set by 9041 +2]
9045	M [Value set by 9045]	O [Value set by 9041 +3]
9046	M [Value set by 9046]	O [Value set by 9041 +4]
9047	M [Value set by 9047]	O [Value set by 9041 +5]
9048	M [Value set by 9048]	O [Value set by 9041 +6]
9049	M [Value set by 9049]	O [Value set by 9041 +7]
9050	M [Value set by 9050]	O [Value set by 9041 +8]
9051	M [Value set by 9051]	O [Value set by 9041 +9]
9052	M [Value set by 9052]	O [Value set by 9041 +10]
9053	M [Value set by 9053]	O [Value set by 9041 +11]

2.12.17 M Code Subprogram Call (Interval)

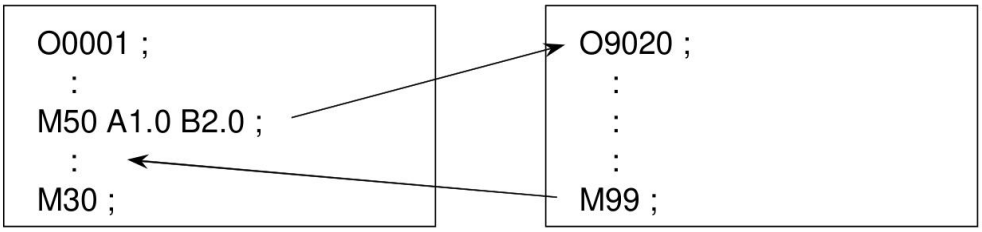
You can define multiple M code subprogram calls by setting the M code number used in the macro call, the opening program number of the call, and the number of definitions in advance.

Example

sys-parameter-9054 is set to 99, sys-parameter-9055 is set to 9500, and sys-parameter-9056 sets the M-code start number to 500	
M Code	**Corresponding program number**
M500	O9500
M501	O9501
M502	O9501
M503	O9501
...	...
M599	O9599

2.12.18 M Code Macro Call (Usually)

Format



Description

The macro program can be called by setting a M code number in the parameter in advance to call the macro program, and the call method is similar to G65.

Detailed description

1. First set the CNC parameter-9060, which sets the number of programs that can execute the call;
2. First set sys-parameter-9061, which sets the opening program number for execution of the macro call;
3. Correspondence between parameter number and program number:

sys-parameter number	Corresponding M code	Corresponding program number
9062	M [Value set by 9062]	O [Value set by 9061 +0]
9063	M [Value set by 9063]	O [Value set by 9061 +1]
9064	M [Value set by 9064]	O [Value set by 9061 +2]
9065	M [Value set by 9065]	O [Value set by 9061 +3]
9066	M [Value set by 9066]	O [Value set by 9061 +4]
9067	M [Value set by 9067]	O [Value set by 9061 +5]
9068	M [Value set by 9068]	O [Value set by 9061 +6]
9069	M [Value set by 9069]	O [Value set by 9061 +7]
9070	M [Value set by 9070]	O [Value set by 9061 +8]
9071	M [Value set by 9071]	O [Value set by 9061 +9]
9072	M [Value set by 9072]	O [Value set by 9061 +10]
9073	M [Value set by 9073]	O [Value set by 9061 +11]

2.12.19 M Code Macro Call (Interval)

You can define multiple M code macro calls by setting the M code number used in the macro call, the opening program number of the call, and the number of definitions in advance.

Example

sys-parameter-9074 is set to 99, sys-parameter-9075 is set to 9800, and sys-parameter-9076 sets the M-code start number to 800	
M Code	**Corresponding program number**
M800	O9800
M801	O9801
M802	O9801
M803	O9801
...	...
M899	O9899

2.12.20 Modbus Read Hold Register (03H)

CNC Software version support is required, version number: 2.80.33+

Format

MBRD03[slv, reg, num, dst, retvar]	
slv	:Slave device number
reg	:Read the starting register address, range 0~99 (0x00~0x63)
num	:Read the number of data, with one register having a size of 16 bits
dst	:Data storage address, such as X0, R0,500 (macro variable address)
retvar	:Execution result returns the macro address, which can be omitted. =1/0 indicates true/false

Description

Read the register values of the slave device through the serial port.

Detail Description

1. Before use, it is necessary to set the serial port baud rate system parameter 2120, serial port parity check system parameter 2121, and main station port system parameter 2135;
2. When the parameter 'retvar' is omitted, a CNC alarm will be generated directly if the read fails. If not omitted, the success of the read will be determined by specifying the macro address;

Example

Example 1	
MBRD03[2, 0x00, 2, 500, 1]	//Starting from the 0x00 address of device slave 2, read two consecutive data and assign them to CNC # 500, and write the result into #1(true/false)
IF[#1!=1] #3000=1(read fail)	//Determine if the reading was successful

Example 2	
MBRD03[2, 0x10, 1, 100]	//Starting from the 0x10 address of device slave 2, read 1 data and assign it to CNC # 100. If the read fails, an CNC alarm will be generated

2.12.21 Modbus Write Coil(05H)

CNC Software version support is required, version number: 2.80.33+

Format

MBWR05[slv, reg, set, retvar]	
slv	:Slave device number
reg	:Write the starting coil address, range 0~99 (0x00~0x63)
set	:Coil reset and setting, 0 or 1
retvar	:Execution result returns the macro address, which can be omitted.=1/0 indicates true/false

Description

Write the coil of the slave device through the serial port.

Detail Description

1. Before use, it is necessary to set the serial port baud rate system parameter 2120, serial port parity check system parameter 2121, and main station port system parameter 2135;

2. When the parameter 'retvar' is omitted, a CNC alarm will be generated directly if the read fails. If not omitted, the success of the read will be determined by specifying the macro address;

Example

Example	
MBWR05[10, 10, 0] or MBWR05[10, 0x0A, 0]	//Write 0 to coil address 10 (0x0A) from the slave device at station 10
...	
MBWR05[10, 10, 1] or MBWR05[10, 0x0A, 1]	//Write 1 to coil address 10 (0x0A) from the slave device at station 10

2.12.22 Modbus Write Hold Register(06H)

CNC Software version support is required, version number: 2.80.33+

Format

MBRD06[slv, reg, src, retvar]	
slv	:Slave device number
reg	:Read the starting register address, range 0~99 (0x00~0x63)
src	:Data source address, such as X0, R0,500 (macro variable address)
retvar	:Execution result returns the macro address, which can be omitted.=1/0 indicates true/false

Description

Read the register values of the slave device through the serial port.

Detail Description

1. Before use, it is necessary to set the serial port baud rate system parameter 2120, serial port parity check system parameter 2121, and main station port system parameter 2135;
2. When the parameter 'retvar' is omitted, a CNC alarm will be generated directly if the read fails. If not omitted, the success of the read will be determined by specifying the macro address;
3. The size of a single register value written is 16 bits of data;

Example

Example 1	
MBWR06[2, 0x00, 500, 1]	//Write the value of CNC #500 to the device at 0x00 address of slave station 2,, and write the result into #1(true/false)
IF[#1!=1] #3000=1(fail)	//Determine if the writing was successful

Example 2	
MBWR06[2, 0x10, 100]	//Write the value of CNC #100 to the device at 0x10 address of slave station 2,and automatically determine whether it is successful. If it fails, immediately generate a CNC alarm