FINGER CNC

Open Designs Customize The Future

B-Series Lathe CNC Programming Manual

Version No：F202309LP-EN

Guangzhou Finger Technology Co., Ltd

CONTENTS

Part 1. G-Code Instruction Explanation

1.1 G-Code Instruction List

Code Values	Function Definitions	Code Values	Function Definitions
0	Rapid Linear Positioning	50	External Workpiece Coordinate Offset
1	Linear Cutting		Maximum Spindle Clamp Speed
2	Clockwise Circular Interpolation	52	Local Coordinate System Setup
2.1	Clockwise Ellipse Instruction	53	Mechanical Coordinate Positioning
2.2	Clockwise Parabolic Instruction	54	Work Coordinate System Setup
3	Counterclockwise Circular Interpolation	55	
3.1	Counterclockwise Ellipse Instruction	56	
3.2	Counterclockwise Parabolic Instruction	57	
4	Pause Instruction	58	
9	Single-Section Precise Positioning	59	
12.1	Polar Coordinate Interpolation Start	54.1-54.48	
13.1	Cancel Polar Coordinate Interpolation		

17	XY Plane	70	Fine Turning Machining Cycle Cancel
18	ZY Plane	71	Compound Cross Roughing Cycle
19	YZ Plane	72	Compound Longitudinal Roughing Cycle
20	Imperial Mode	73	Compound Profile Roughing Cycle
21	Metric Mode	74	Cross Grooving Cycle
28	Return to First Reference Point	75	Longitudinal Grooving Cycle
29	Auto Return from Reference Point	76	Compound Threading Cycle
30	Return to Second Reference Point	83	Face Drilling Cycle
32	Thread Cutting	84	Face Threading Cycle
34	Variable Lead Thread Cutting	85	Face Boring Cycle
35	Clockwise Circular Arc Thread Cutting	87	Side Drilling Cycle
36	Counterclockwise Circular Arc Thread Cutting	88	Side Threading Cycle
40	Cancel Tool Tip Radius Compensation	89	Side Boring Cycle
41	Left Tool Tip Radius Compensation	90	External/Internal Turning Cycle
42	Right Tool Tip Radius Compensation	92	Thread Cutting Cycle
		94	Longitudinal Cutting Cycle

		96	Constant Surface Speed Function
		97	Disable Constant Surface Speed
		98	Feed Per Minute Mode
		99	Feed Per Revolution Mode

Table 1.1

1.2 Rapid Positioning (G00)

The G00 command is a rapid movement positioning command used solely for point-to-point positioning. It should not involve any cutting action and is primarily used to save time during non-cutting movements. In lathe programming, it is often used for moving from the machine origin to the cutting start point of the workpiece or for moving from the workpiece's cutting end point back to the machine origin.

1.2.1 Command Format

G00 X(U)Z(W)

X, Z: Specify the endpoint coordinates (in absolute values).

U, W: Specify the endpoint coordinates (in incremental values).

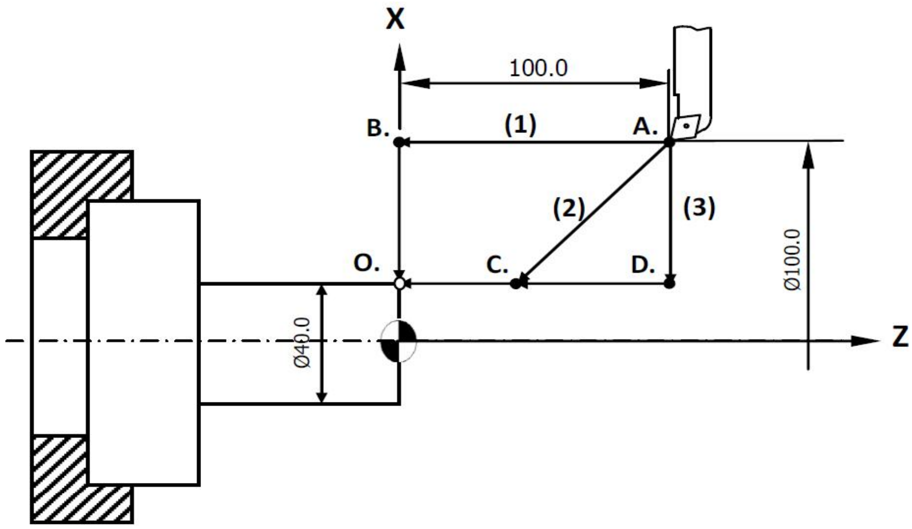


Figure 1.2-1

The movement of the tool from point A to point O is not limited to a single path; it can be chosen based on the specific requirements. There are three different approaches and three different paths that can be used when the tool approaches the workpiece.

1.3 Linear Cutting (G01)

When using the G01 command, the tool moves in a straight line from its current position to a specified location at the feed rate set by the F__ function. This mode is used for various machining operations, including external (internal) diameters, end faces, external (internal) tapers, external (internal) grooves, chamfers, and more.

1.3.1 Command Format

G01 X(U)__Z(W)F;

X, Z: Specify the endpoint coordinates (absolute mode).

U, W: Specify the endpoint coordinates (incremental mode).

F: Feed rate. In G98 mode, the metric unit is mm/min (or inch/min in imperial units); in G99 mode, the metric unit is mm/rev (or inch/rev in imperial units). The system defaults to G99 mode at startup (can be changed to G98 mode through parameter settings).

Parameter Setting Procedure: Machine Position Page → Parameter Setting → Comprehensive Parameters → Default G98/G99 Mode at Startup.

1.3.2 Notes

1. The maximum speed in G01 mode is limited by the maximum speed during cutting or the maximum speeds of individual axes during cutting.
2. In G98 mode, the default speed is 1000 mm/min (or inch/min); in G99 mode, the default speed is 1 mm/rev (or inch/rev).
3. The default settings for G98/G99 can be adjusted using SYS10032 (take effect after a restart).

1.3.3 Example

- **Diameter Programming**

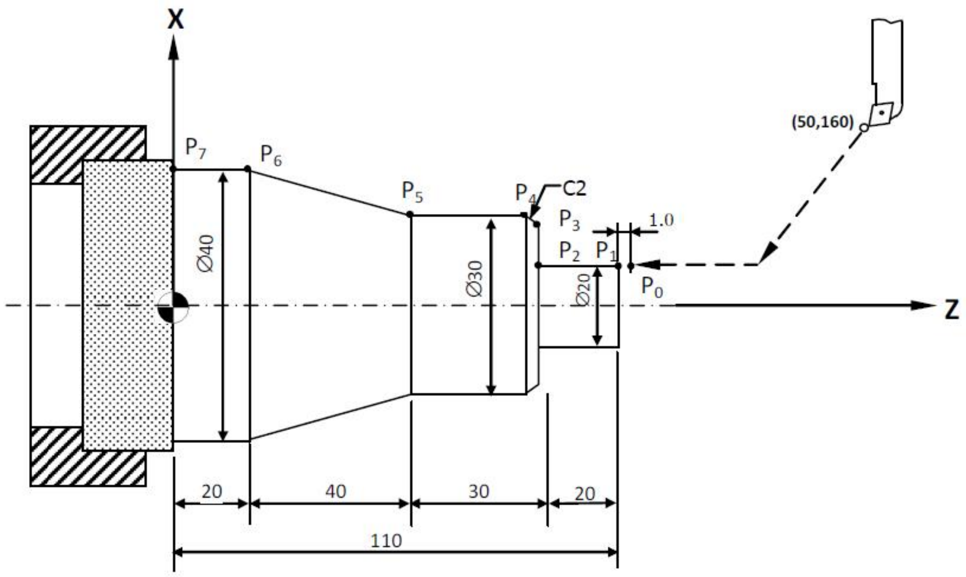


Figure 1.3-1

G50 S10000; // Program origin set, maximum speed 10,000 rpm;

```
M3 S1000;            // Start the first spindle in the forward direction;
T01;                 // Use tool No. 1;
M08;                 // Turn on the cutting fluid;
G00 X20.0 Z111.0;    // Rapid positioning to P0;
G01 Z90.0 F0.6;      // Linear cutting from P0 to P2;
X26.0;               // Move from P2 to P3;
X30.0 Z88.0;         // Move from P3 to P4;
Z60.0;               // Move from P4 to P5;
X40.0 Z20.0;         // Move from P5 to P6;
Z0.0;                // Move from P6 to P7;
G00 X50.0;           // Rapid tool retraction;
Z160.0;              // Return to the starting point;
G97;                 // Cancel constant spindle speed;
M05;                 // Stop the spindle;
M09;                 // Turn off the cutting fluid;
M30;                 // End of the program;
```

1.4 Circular Cutting (G02/G03)

In CNC lathe systems, the G02/G03 commands specify the tool's circular arc trajectory cutting in the X-Z plane.

1.4.1 Command Format

G02 (G03) X(U)Z(W)R(I__K)F__

G02: Specifies clockwise direction circular arc cutting with the tool.

G03: Specifies counterclockwise direction circular arc cutting with the tool.

X, Z: Absolute coordinates of the arc's endpoint.

U, W: Incremental values from the arc's starting point to the endpoint.

R: Arc radius (limited to within 180°).

I, K: Incremental values from the arc's starting point to the arc's center on the X, Z axes (I specifies the radius).

F: Cutting feedrate.

G02/G03 Cutting Direction:

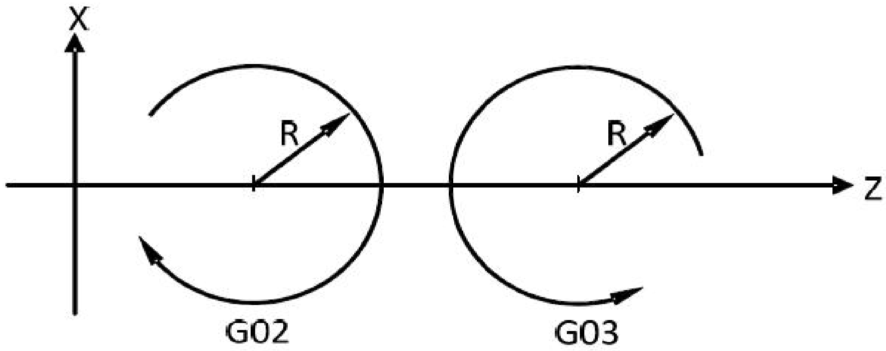


Figure 1.4-1

1.4.2 Additional Information

G02 Circular Arc Cutting:

1. Using the R value method.

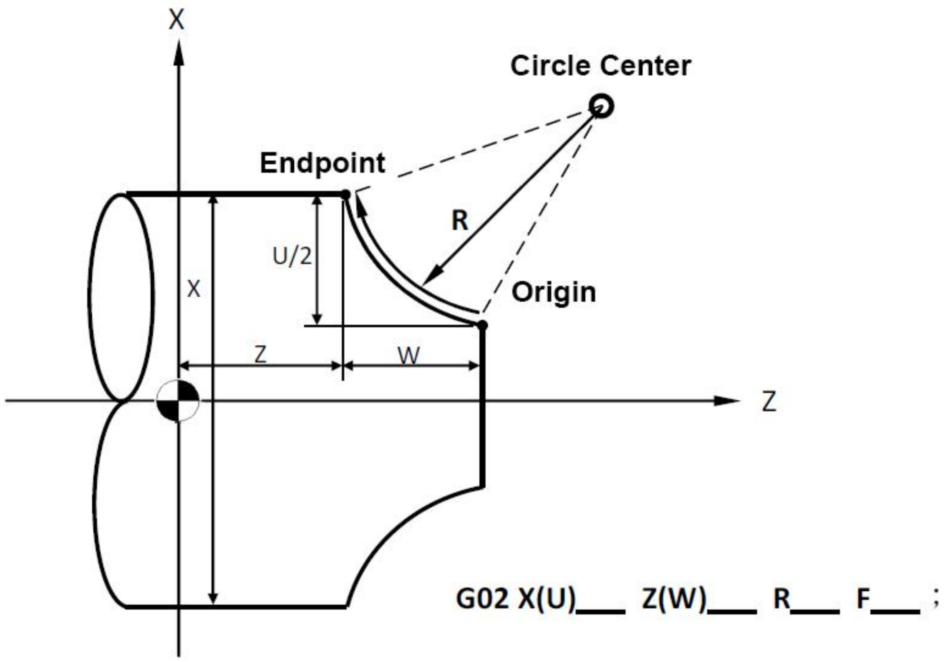


Figure 1.4-2

2. Using the I and K value method.

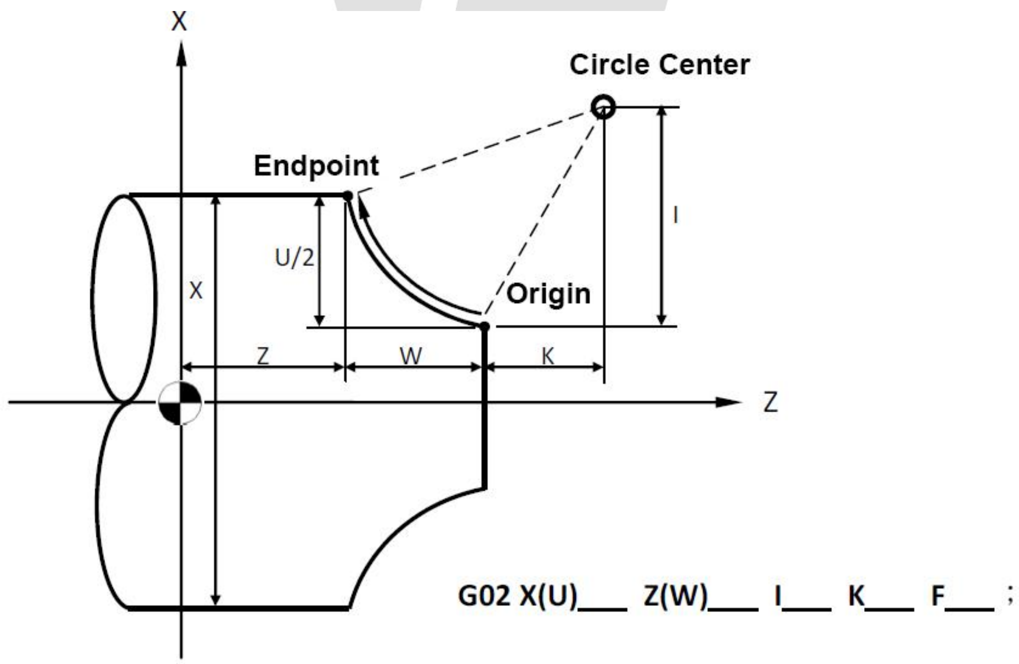


Figure 1.4-3

G03 Circular Arc Cutting:

3. Using the R value method.

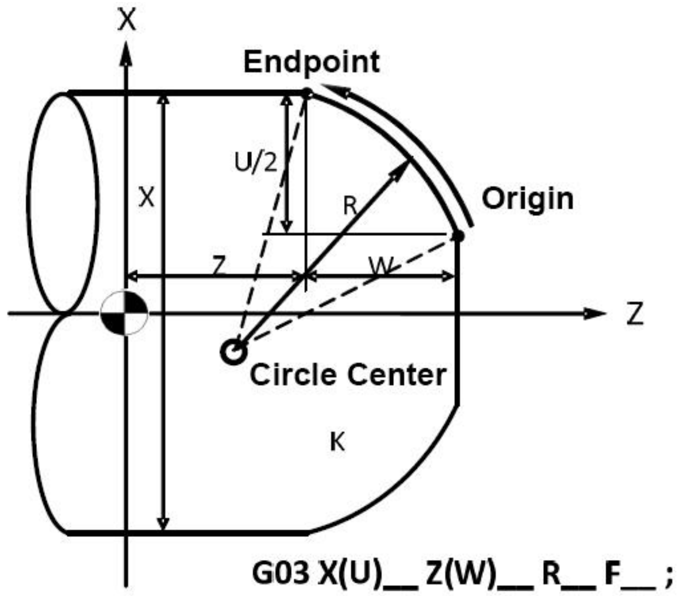


Figure 1.4-4

4. Using the I and K value method.

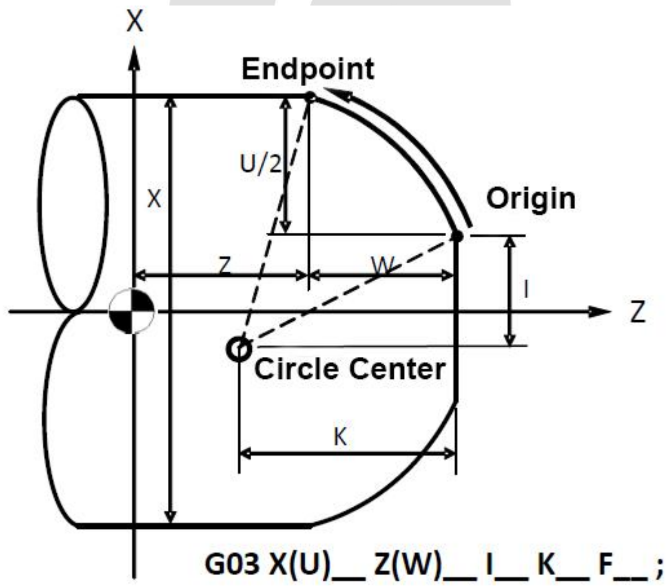


Figure 1.4-5

1.4.3 Note

1. When no R, I, J, K values are specified for G02/G03, it is treated as G01.

2. If the specified X, Z, I, K, R parameters for G02/G03 are incorrect, the system will

generate an alarm "Endpoint not on the arc," and this alarm can be adjusted within the alarm range through parameters.

1.4.4 Example 1

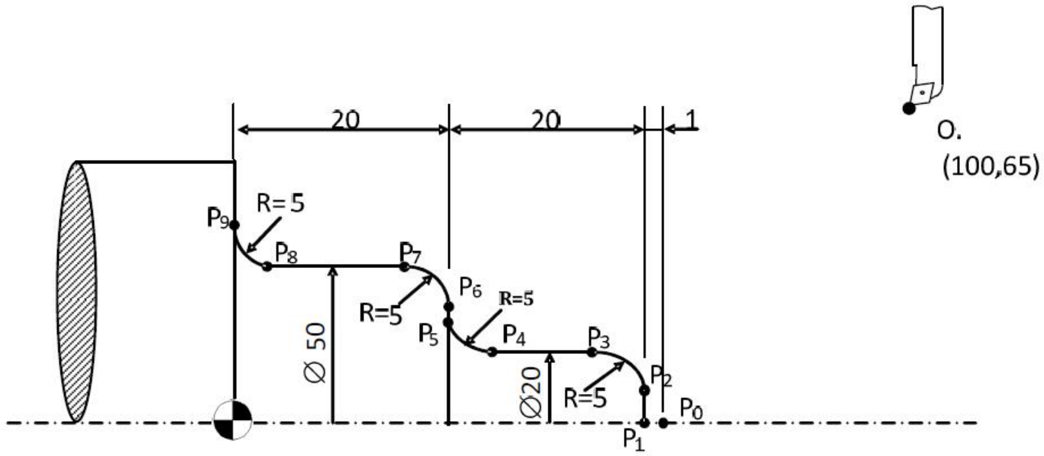


Figure 1.4-6

Diameter Programming

T01;	//Use Tool 1;
G99	//G99 mode
G50 S10000;	//Set maximum spindle speed limit to 10000rpm;
M3 S1000;	
M08;	//Turn on coolant;
G00 X0.0 Z41.0;	//Rapid positioning from O. to P0;
G01 Z40.0 F0.6;	//Linear cutting with a feed rate of 0.6mm/rev, from P0 to P1;
G00 X10.;	//Linear cutting with the feed rate set in the previous block, from P1 to P2;
G03 X20. Z35. R5.0;	//Counterclockwise circular arc cutting from P2 to P3 with a radius of 5mm;
G01 Z25.0;	//P3→P4;
G02 X30.0 Z20.0 R5.0;	//Clockwise circular arc cutting from P4 to P5 with a

	radius of 5mm;
G01 X40.0;	//P5→P6;
G03 X50.0 Z15.0 R5.0;	//Counterclockwise circular arc cutting from P6 to P7 with a radius of 5mm;
G01 Z5.0;	//P7→P8;
G02 X60.0 Z0.0 R5.0;	//Clockwise circular arc cutting from P8 to P9 with a radius of 5mm;
G00 X100.0 ;	//Rapid tool retraction, moving away from the workpiece;
G00 Z65.0;	//Return to the original point;
M09;	//Turn off coolant;
M05;	//Stop the spindle;
M30;	//End of program;

1.4.5 Example 2

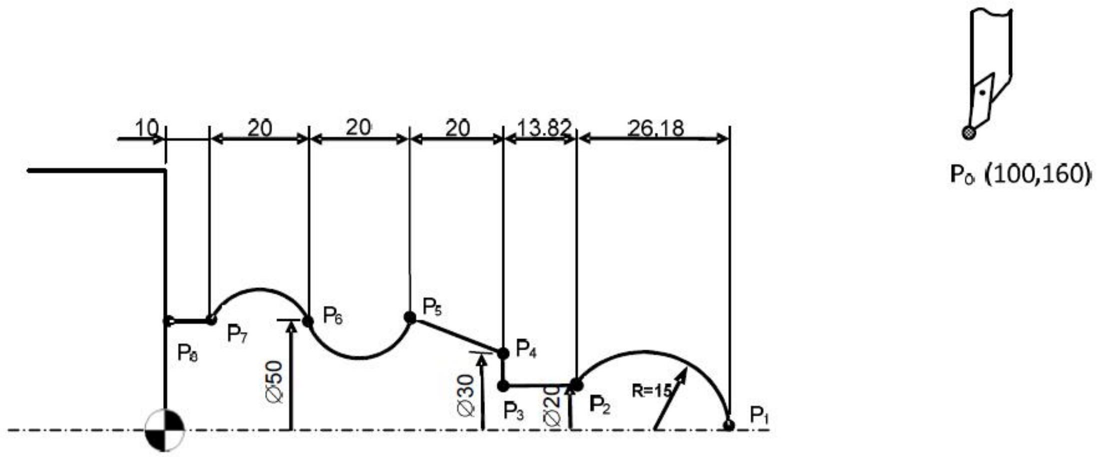


Figure 1.4-7

Diameter Programming

T01;	//Use Tool 1;
G99;	//G99 mode
G50 S10000;	//Set maximum spindle speed limit to 10000rpm;

M03 S1000; //Start the spindle, 1000rpm;

M08; //Turn on coolant;

G00 X0 Z110.5; //Rapid positioning, approaching the cutting start point;

G01 Z110.0 F0.5; //Linear cutting with a feed rate of 0.5mm/rev;

G03 X20.0 Z83.82 R15.0; //Counterclockwise circular arc cutting from P1 to P2, radius 15mm;

G01 Z70.0; //Linear cutting from P2 to P3;

X30.0; //P3→P4;

X50.0 Z50.0; //P4→P5;

G02 X50.0 Z30.0 R10.0; //Clockwise circular arc cutting from P5 to P6, radius 10mm;

G03 X50.0 Z10.0 R10.0; //Counterclockwise circular arc cutting from P6 to P7, radius 10mm;

G01 Z0.0; //Linear cutting from P7 to P8;

G00 X100.0; //Rapid tool retraction, moving away from the workpiece;

Z160.0; //Return to the original point;

M09; //Turn off coolant;

M05; //Stop the spindle;

M30; //End of program;

1.5 Elliptical Cutting(G02.1/G03.1)

In CNC lathe systems, the G02.1 and G03.1 commands specify elliptical arc cutting in the X-Z plane for the tool.

1.5.1 Instruction Format

G02.1 X(U)__Z(W)__R__P__Q__F__

Or

G03.1 X(U)__Z(W)__R__P__Q__F__

G02.1: Specifies clockwise elliptical arc cutting.

G03.1: Specifies counterclockwise elliptical arc cutting.

X(U) and Z(W): Absolute coordinate values of the endpoint of the ellipse (U and W are incremental coordinates from the starting point to the endpoint of the ellipse).

R: Length of the long semi-axis of the ellipse (specified as a radius, ignore the sign, non-modal).

P: Length of the short semi-axis of the ellipse (specified as a radius, ignore the sign, non-modal).

Q: Angle between the long semi-axis of the ellipse and the positive Z-axis (0° to 180°, unsigned, non-modal).

F: Cutting feed rate (modal command).

1.5.1.1 Elliptical Z/X Axis Length Definition

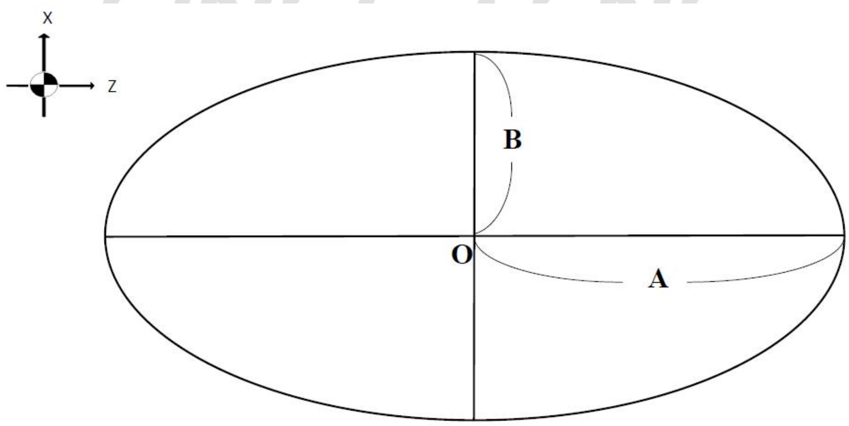


Figure 1.5-1

R: The longer of A and B in the example.

P: The shorter of A and B in the example.

For example, in the diagram:

1. If A > B: R is A, P is B, and Q is 0°.
2. If A < B: R is B, P is A, and Q is 90°.

1.5.1.2 Determining G02.1/G03.1 Direction

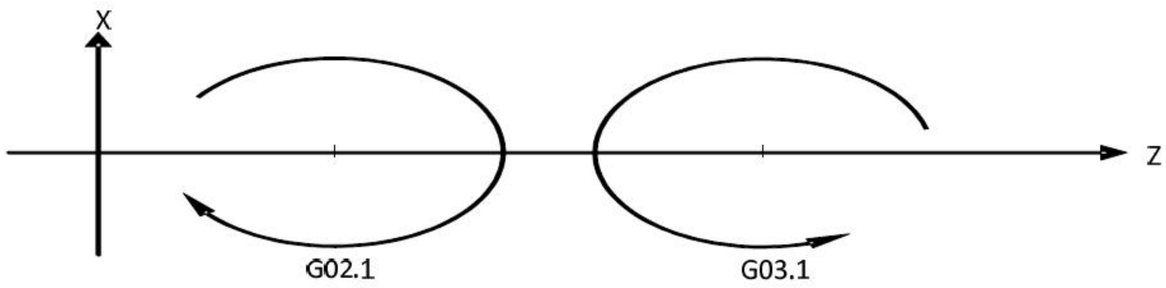


Figure 1.5-2

G02.1: Clockwise elliptical cutting.

G03.1: Counterclockwise elliptical cutting.

1.5.1.3 Specifying Parameters for Actual Machining

1. G02.1 Elliptical Arc Cutting

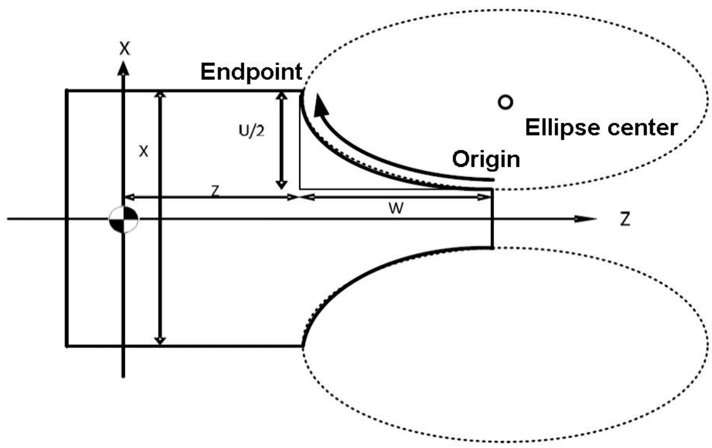


Figure 1.5-3

G02.1 X(U)__Z(W)__R__P__Q__F__

2. G03.1 Elliptical Arc Cutting

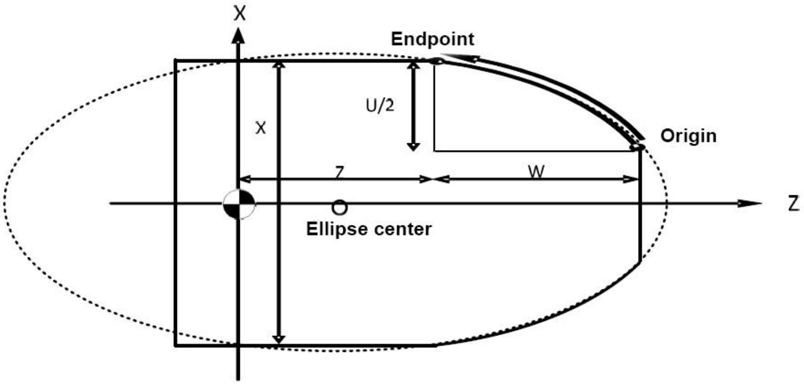


Figure 1.5-4

G03.1 X(U)__Z(W)__R__P__Q__F__

1.5.2 Notes

1. R and P are non-modal parameters, defaulting to 0. When R = 0 or P = 0, an alarm [102/103-03 (ellipse major or minor axis radius is 0)] is triggered. When R = P, it functions as circular arc machining (G02/G03).
2. Q value is non-modal, defaulting to 0.
3. When the programmed start and end points coincide, the system triggers an alarm (102/103-01).
4. When the distance between the programmed start and end points is greater than R, the system triggers an alarm (102/103-02).
5. This command only processes ellipses with angles less than or equal to 180°.
6. When the feed rate is equal to 0, the system triggers an alarm (102/103-04).
7. Q value should include a decimal point, e.g., Q10. represents a 10° deviation.
8. Q value can be set as negative, and when set as negative, it is treated as an absolute value.
9. Q value can be set to exceed 180°, and when set above 180°, the Q value is converted to its modulo 180° equivalent.

1.5.3 Example

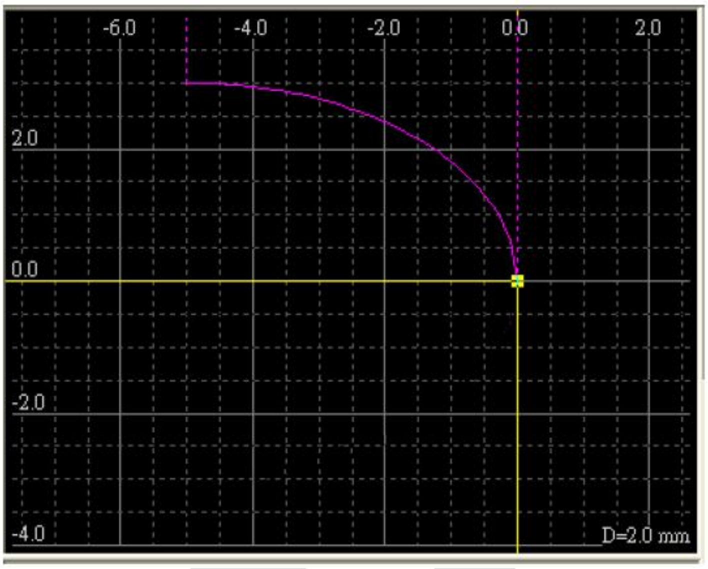


Figure 1.5-5

Turning Semi-Ellipse (Diameter Programming)

G99; //G99 mode

G00 X8.; //Starting point for elliptical turning - X coordinate

Z0; //Starting point for elliptical turning - Z coordinate

M03 S1000; //Start the main spindle in the forward direction at 1000 RPM

G00 X0.; //Starting point for elliptical turning - X coordinate

G03.1 Z-5. X6. R5. P3. F0.5; //Counterclockwise elliptical turning

G00 X8.; //Position at X8.0

M05; //Stop the main spindle

M30; //End of program

1.6 Parabolic Interpolation (G02.2/G03.2)

In CNC lathe systems, the G02.2 and G03.2 commands are used to specify toolpath cutting along a parabolic trajectory in the X-Z plane.

1.6.1Instruction Format

G02.2 X(U)__Z(W)__P__Q__F__;

G03.2 X(U)__Z(W)__P__Q__F__;

G02.2: Specifies clockwise parabolic cutting.

G03.2: Specifies counterclockwise parabolic cutting.

X(U), Z(W): Absolute value coordinates of the endpoint of the parabolic curve (U and W are incremental values from the starting point to the endpoint).

P: Parabolic focal length.

Q: Angle between the parabolic axis of symmetry and the positive Z-axis.

F: Cutting feed rate.

1.6.1.1 Definition of the Parabolic Focal Length

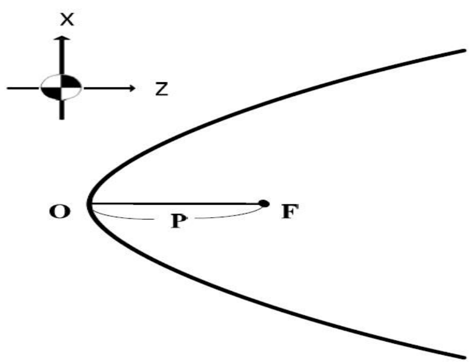


Figure 1.6-1

The definition of the parabolic focal length is the distance P from the vertex O of the parabola to the focus F.

1.6.1.2 Determining the Direction of G02.2/G03.2

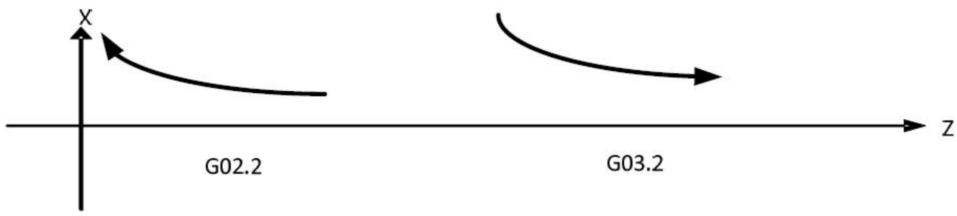


Figure 1.6-2

G02.2 specifies clockwise parabolic cutting.

G03.2 specifies counterclockwise parabolic cutting.

1.6.1.3 Specifying Actual Machining Parameters

1. G02.2 Parabolic Cutting:

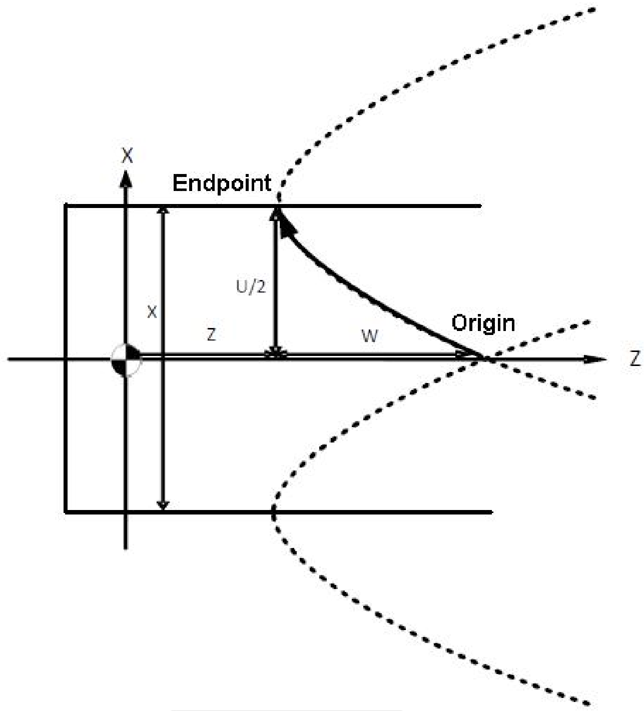


Figure 1.6-3

G02.2 X(U)__Z(W)__P__Q__F__

2. G03.2 Parabolic Cutting:

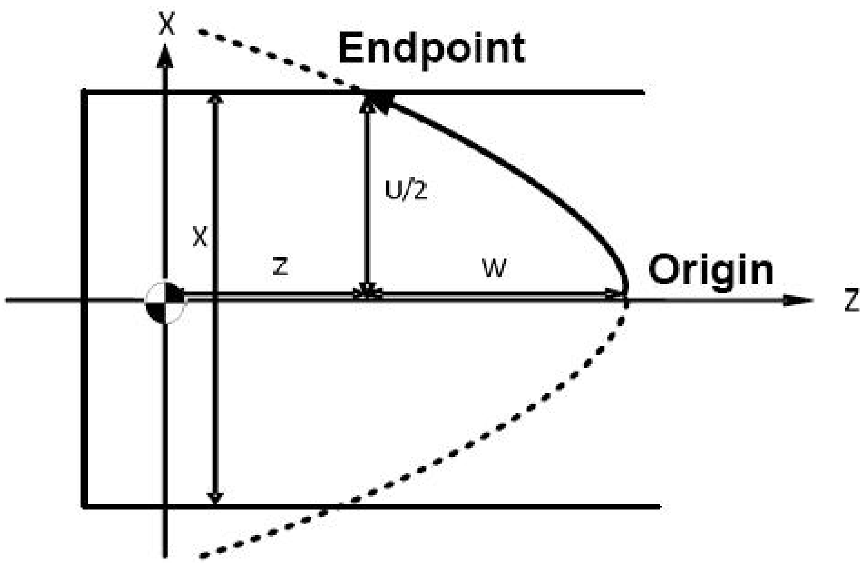


Figure 1.6-4

G03.2 X(U)__Z(W)__P__Q__F__

3. Explanation of Q Value:

The Q value refers to the angle, in the right-handed Cartesian coordinate system, when viewed from above the current plane in the direction of the positive third axis of the plane, the angle through which the positive first axis of the plane rotates in the clockwise (or counterclockwise) direction to coincide or align with the axis of symmetry of the parabola. Taking the G18 plane as an example, for the front tool holder, angle 'a' represents the angle through which it rotates counterclockwise from the Z-axis to align with the axis of symmetry of the parabola, as shown below:

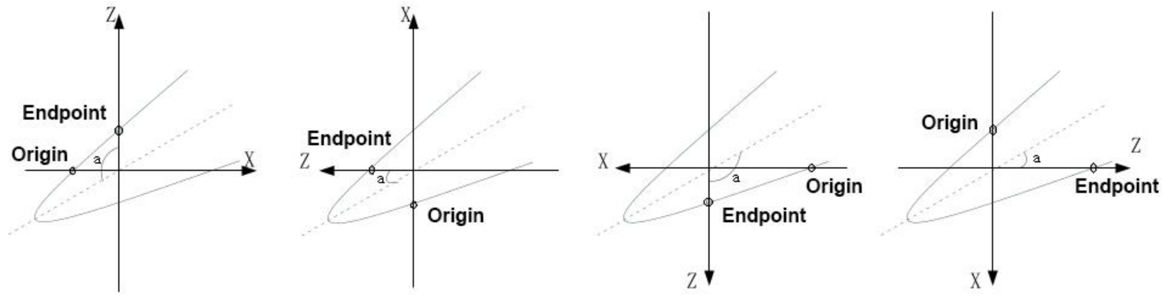


Figure 1.6-5

1.6.2 Notes

1. When X and Z both have no increments, the system will trigger an alarm (202/203-01).
2. When P value is 0 or omitted, the system will trigger an alarm (202/203-02).
3. If the line formed by the start and end points is parallel to the axis of symmetry of the parabola, the system will trigger an alarm (202/203-03).
4. When the feed speed is set to 0, the system will trigger an alarm (202/203-04).
5. P is not affected by the decimal point, and its value is unsigned. If a negative value is set for P, it will be treated as an absolute value.
6. Q value can be omitted. If omitted, it means that the centerline of the parabola is parallel to or coincides with the Z-axis. Q value is unsigned and is affected by the decimal point.

7. Q value can be set as a negative value. When a negative value is set, it is treated as an absolute value.
8. Q value can be set beyond 180°. When set beyond 180°, it is taken modulo 180°.

Warnings:

202/203-01: No increments for X and Z.

202/203-02: Parabola P value is 0 or omitted.

202/203-03: The line formed by the start and end points of the parabola is parallel to the axis of symmetry.

202/203-04: Parabola feed speed is set to 0.

202/203-05: The start and end points are not on the parabola.

1.6.3 Example

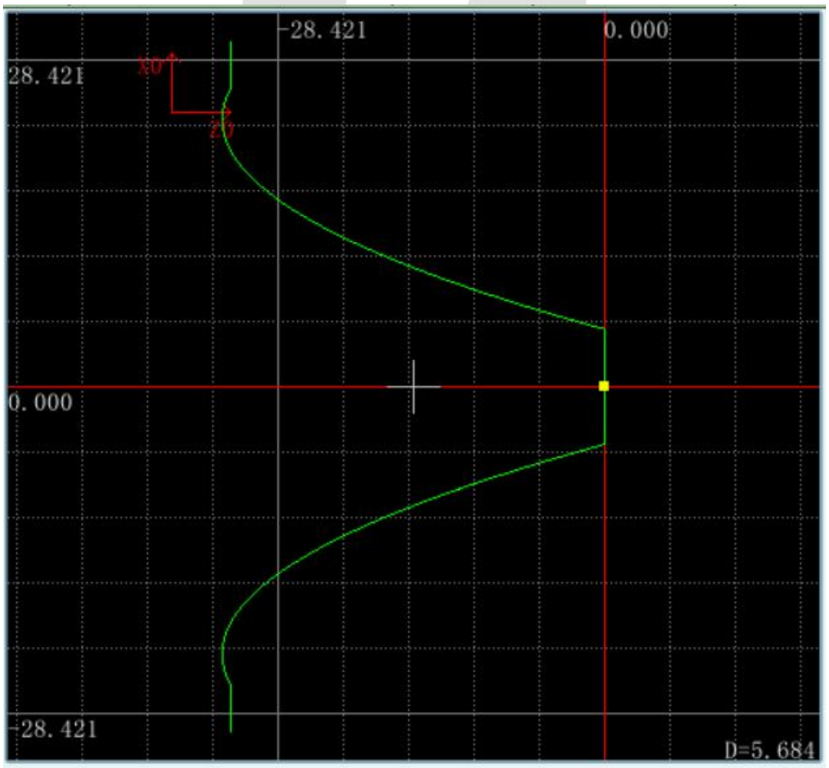


Figure 1.6-6

Turning Parabola (Diameter Programming):

G99;	//G99 mode
M03 S1000;	//Start the spindle in forward direction at 1000rpm
G00 X10.0;	//X-coordinate for the start of parabolic turning
Z0;	//Z-coordinate for the start of parabolic turning
G02.2 X52.0 Z-32.55 P5. F0.1;	//Counterclockwise parabolic turning
G00 X60.0;	//Position at X60
M05;	//Stop the spindle
M30;	//End of program

1.7 Three-Point Arc Interpolation（G02.4/G03.4）

1.7.1 Function and Purpose

This function calculates a circular arc interpolation based on the geometry of three known points in space. It enables the generation of a connected arc through these three points in the specified order. By providing the coordinates of the middle point of the arc, the endpoint of the arc (which is also the endpoint of the previous segment for G02.4 or G03.4), and the tangential feed rate, you can create a three-point circular arc in space.

1.7.2 Instruction Format

G02.4/3.4 X1(U)_ Y1(V)_ Z1(W)_ F_

X2(U)_ Y2(V)_ Z2(W)_

X1/Y1/Z1	First Segment (Middle Point of the Arc, Absolute Programming)
X2/Y2/Z2	Second Segment (Endpoint of the Arc, Absolute Programming)
U/V/W	Incremental values from the starting point to the middle point or from the middle point to the endpoint (Incremental Programming)
F	Tangential feed rate

1.7.3 Programming Example

- **Example 1**

```
G98 ;                    //Switch to G98 mode
G01 X0 Y0 Z0 F500;       //G1 positioning, starting point of the arc
G02.4 X5. Y5. Z8. F100;  //Three-point arc interpolation, midpoint of the arc
X10. Y10. Z0.;           //Three-point arc interpolation, endpoint of the arc
M30;                     //End of the program
```

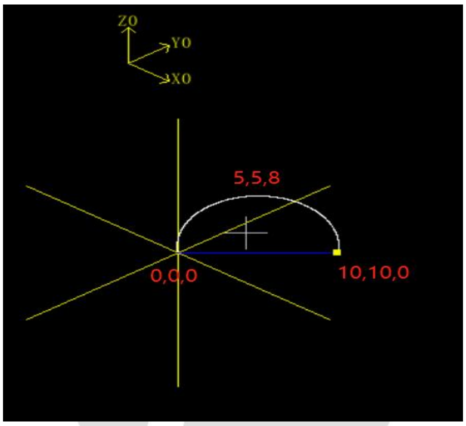


Figure 1.7-1

- **Example 2**


```
G98;                     //Switch to G98 mode
G01 X0 Y0 Z0 F5000;      //G1 positioning, starting point of the arc
G02.4 X5. Y5. F1000;     //Three-point arc interpolation, midpoint of the arc
Z10;                     //Three-point arc interpolation, endpoint of the arc
M30;                     //End of the program
```

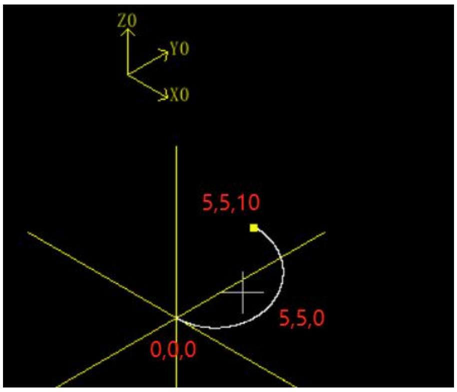


Figure 1.7-2

1.7.4 Notes

1. X, Y, Z: If any axis coordinate is omitted, it will take the same value as the previously specified point.
2. The G02.4 command and G03.4 command have the same actions and can be used interchangeably. They support both absolute and incremental instructions.
3. The endpoint of the previous motion is considered the starting point for the arc when using G02.4/G03.4.
4. If the three points that define the arc are in a straight line or if any two points coincide (e.g., specifying a full circle, causing the start and end points to be the same), linear interpolation (G01) will be used. The interpolation path will go from the start point to the midpoint and then from the midpoint to the end point.
5. When executed as a single block, it will move from the arc's starting point to the endpoint in one continuous motion, without stopping at the midpoint of the arc.
6. When using this function, the tool radius compensation function must be turned off, and commands such as A, C, and R are not supported. Otherwise, it will trigger the COR-133 alarm (Command not supported in G02.4/G03.4 interpolation mode).
7. In G02.4/G03.4 interpolation mode, G53 machine coordinate positioning cannot be used. Also, using a G02.4/G03.4 command immediately after G53 machine coordinate positioning will result in incorrect arc path.

8. If α/β instructions are omitted in the first program block but specified only in the second program block, the α/β axis will not move from the arc's starting point to the midpoint. However, when moving from the midpoint to the endpoint, the α/β axis will move to the specified position.
9. Similarly, if α/β instructions are omitted in the second program block but specified only in the first program block, the α/β axis will move to the specified position when moving from the arc's starting point to the midpoint, but it will not move from the midpoint to the endpoint.

1.8 Pause Command (G04)

This function allows pausing the machine's movement through a program command to introduce a time delay or wait state, effectively delaying the start of the next block.

1.8.1 Instruction Format

G04 X/U(P)__

X/U: Pause time, in seconds (In this context, X/U represents time, not position, and is subject to the influence of the "decimal point on/off" parameter).

P: Pause time, in milliseconds (Not affected by the "decimal point omitted/on" parameter).

1.8.2 Example

```
G04 X0.5;          //Pause for 0.5 seconds.
G04 U1.5;          //Pause for 1.5 seconds.
G04 P500;          //Pause for 0.5 seconds.
```

1.9 G06.2 NURBS Curve Interpolation

1.9.1 Function and Purpose

This function is used to specify parameters (degree, weights, knots, control points) for NURBS (non-uniform rational B-spline) curves used in surface or curve machining. It does not replace small line segments and is used for NURBS curve machining.

1.9.2 Instruction Format

G06.2 P__K__X__Y__Z__R__F__	
P	NURBS Curve Degree: The degree of the NURBS curve, which is (p-1) when omitted, holds the same meaning as p=4
K	Knots: Knots are values that are set equally from the first program segment to

	the p-th program segment. If there are program segments with only knots, the NURBS interpolation ends.
X/Y/Z _	Control Point Coordinates
R	"Control Point Weights," with a range from 0.0001 to 99.9999.
F	Interpolation Speed

1.9.3 Programming

```
G17;
G0 X0 Y0 Z0 ;
G06.2 P4 X0. Y0. R1. K0 F1000 ;
X1. Y2. R1. K0 ;
X2.5 Y3.5 R1. K0 ;
X4.4 Y4. R1. K0 ;
X6. Y0.5 R1. K1 ;
X8. Y0. R1. K2 ;
X9.5 Y0.5 R1. K3 ;
X11. Y2. R1. K4 ;
X10.5 Y4.5 R1. K5 ;
X8.0 Y6.5 R1. K6 ;
X9.5 Y8. R1. K7 ;
K8;
K8;
K8;
K8;
M30
```


1.10 Spline Curve Function (G06.3)

1.10.1 Instruction Format

G06.3 Q__ E__ D__ α__ β__ F__	
Q	Spline curve control point start address (excluding the starting point), unaffected

	by the decimal point. (When the Q letter is not specified, it defaults to 1000)
E	Spline curve control point end address (excluding the endpoint), unaffected by the decimal point. (When the E letter is not specified, it defaults to 1000)
D	D0 = Control point coordinate specified as absolute D1 = Control point coordinate specified as incremental (when D is not specified, it defaults to D0, absolute)
α/β	Spline curve interpolation axis endpoint coordinates (, can be any axis among X, Y, Z)
F	Feedrate (mm/min or mm/rev)

Storage of Control Points:

- ❖ Storage Range: User Variables V0~V9999
- ❖ One variable stores one interpolation axis coordinate, and control point coordinates are stored as groups (i.e., horizontal axis interpolation coordinates are stored first, followed by vertical axis interpolation coordinates). A maximum of 4500 sets of control points can be stored. A single G06.3 command can specify up to 4500 control points. When the starting point is not the same as the first control point, the starting point is also considered a control point; when the end point is not the same as the last control control point, the end point is also considered a control point; adjacent control points that are the same are treated as a single control point
- ❖ As shown in the Table below :

V1000	Horizontal interpolation axis coordinate	The first control point
V1001	Vertical interpolation axis coordinate	
V1002	Horizontal interpolation axis coordinate	
V1004	Vertical interpolation axis coordinate	The second control point
V1005	Vertical interpolation axis coordinate	
V1006	Vertical interpolation axis coordinate	
⋮	⋮	⋮
⋮	⋮	⋮

The control points are stored in the order they appear in the spline curve.

1.10.2 Programming Example

#8000=8.660

```
#8001=10.000
#8002=7.071
#8003=14.142
#8004=5.000
#8005=17.321
#8006=0.
#8007=20.000
#8008=-5.000
#8009=17.321
#8010=-7.071
#8011=14.142
#8012=-8.660
#8013=10.000;
G17 X0 Y0;
G0 X0. Y10.;
G06.3 X0. Y-10. Q8000 E8012 F3000;//Run a Spline Curve for Half Circle (Control Point Coordinates Placed in Variables #8000 to #8012)
M30
```

1.10.3 Explanation

Error Check:

❖ **Qxxxx and Exxxx Check**

The starting address and ending address must both be even numbers. E (ending address) and Q (starting address) have no size constraints. When E (ending address) is greater than Q (starting address), the spline curve interpolation direction goes from Q (starting address) to E (ending address). When E (ending address) is less than Q (starting address), the spline curve interpolation direction goes from E (ending address) to Q (starting address).

1) When Qxxxx is set beyond the storage range, Alarm 306-5 is triggered.
2) When Qxxxx/Exxxx is set as an odd number, Alarm 306-1 is triggered.
3) When Exxxx is set beyond the storage range, Alarm 306-6 is triggered."

1.11 G06.4 Spatial Spline Curve Function

1.11.1 Instruction Format

G06.4 Q__ E__ D__ α__ β__ F__	
Q	Control point starting address for spline curve (excluding the starting point), unaffected by the decimal point. (Defaults to 1000 if the letter "Q" is not provided)
E	Control point ending address for spline curve (excluding the end point), unaffected by the decimal point. (Defaults to 1000 if the letter "E" is not provided)
D	D0 = Control point coordinate specified as absolute D1 = Control point coordinate specified as incremental. Defaults to D0 (absolute) if "D" is not specified.
α/β	Spline curve interpolation axis endpoint coordinates. (Any one of the axes X, Y, or Z for the spline interpolation axis)
F	Feed rate (mm/min or mm/rev)

Storage of control points:

- Storage range: User variables V0 to V9999
- One variable stores one interpolation axis coordinate, and control point coordinates are stored in groups (i.e., first store the coordinates for the horizontal axis interpolation, and then the vertical axis interpolation coordinates). Up to 3333 groups of control points can be stored. A single G06.4 command can specify up to 3333 control points. When the starting point is different from the first control point, the starting point is also treated as a control point; when the end point is different from the last control point, the end point is also treated as a control point. Consecutive control points that are the same are treated as a single control point.
- As shown in the Table below :

V1000	X0 (Horizontal Interpolation Axis Coordinate)	First control point
V1001	Y0 (Vertical Interpolation Axis Coordinate)	
V1002	Z0 (Horizontal Interpolation Axis Coordinate)	
V1004	X1 (Vertical Interpolation Axis Coordinate)	Second control point
V1005	Y1 (Vertical Interpolation Axis Coordinate)	
V1006	Z1 (Horizontal Interpolation Axis Coordinate)	
⋮	⋮	⋮

⋮	⋮	⋮

The control points are stored in the order in which they appear in the spline curve.

1.11.2 Relevant parameters

#M11	Special curve (parabola, spline curve) precision setting (output shortest distance between two points)
#M12	Special curve (parabola, spline curve) precision setting (base axis increment value)

1.11.3 Programming Example

To achieve bidirectional machining along the same spline curve trajectory

Actual execution: In the program, N10 and N20 follow the same trajectory, but in opposite directions. When the program executes N10, it starts from the initial point (X10, Y10, Z5) and follows the specified spline curve control points to interpolate the trajectory, ending at the endpoint (X5, Y10, Z0). When the program executes N20, it starts from the initial point (X5, Y10, Z0) and follows the specified spline curve control points to interpolate the trajectory, ending at the endpoint (X10, Y10, Z5)

Note:

1. When setting control points in incremental mode (D1), it is not possible to achieve bidirectional machining.
2. If the first specified control point coincides with the starting point or the last specified control point coincides with the endpoint, it is counted as one control point, and the system does not generate an alarm.

```
#V1000=10.
#V1001=10.
#V1002=5.
#V1003=5.
#V1004=10.
#V1005=10.
#V1006=0.
#V1007=5.
#V1008=10.
#V1009=0.
```

```
#V1010=5.
#V1011=0.
#V1012=5.
#V1013=10.
#V1014=0.
G98
G0 X10. Y10. Z5.
N10 G06.4 X5.Y10. Z0 Q1000 E1012 F1000
N20 G06.4 X10. Y10. Z5. Q1012 E1000 F1000
M30
```

1.11.4 Explanation

Error Checks:

1. Qxxxx and Exxxx Checks

E (End address) and Q (Start address) have no size limits. When E (End address) is greater than Q (Start address), the spline curve interpolation direction goes from Q (Start address) to E (End address). When E (End address) is less than Q (Start address), the spline curve interpolation direction goes from E (End address) to Q (Start address).

1) When Qxxxx is set beyond the storage range, alarm 406-5 is generated.
2) When Exxxx is set beyond the storage range, alarm 406-6 is generated.

2. Curve Control Point Errors

1) Alarm is generated when the number of control points exceeds 3333.
2) The specified number of control points must be greater than or equal to 3.

Note: If the starting point is different from the first control point, it counts as two control points. Similarly, if the endpoint is different from the last control point, it also counts as two control points. Consecutive identical control points are counted as one control point.

- **Alarm Planning:**

Already Planned Alarms:

406-2, Spline curve point count is less than 3.

406-5, Spline curve first control point address error (exceeds user variable range).

406-6, Spline curve last control point address error (exceeds user variable range).

406-7, Spline interpolation failed, error reason: the total length of data points is 0.

406-8, Curve parameter error, cannot be split, error reason: the spline cannot be differentiated.

1.12 Cylindrical Interpolation

1.12.1 Function and Purpose

Cylindrical interpolation involves unfolding the shape of a cylinder's side into a flat plane and issuing program commands based on this flattened shape as plane coordinates. During machining on a machine tool, this is converted into movements in cylindrical coordinates, combining linear and rotary movements to control the toolpath for contour machining.

1.12.2 Instruction Format

G07.1 IP r; Enable/Cancel cylindrical interpolation mode
r : Specify the radius value for the cylinder
C≠0; Enable cylindrical interpolation mode
C=0; Cancel cylindrical interpolation mode

1.12.3 Detailed Explanation

1.12.3.1 Instruction Applications

1. Coordinate commands within the interval from the start to cancel of cylindrical interpolation mode are in cylindrical coordinate system.
2. G7.1C is for cylindrical radius value; it enables cylindrical interpolation mode.
3. ... ; Coordinate commands within this interval are in cylindrical coordinate system.
4. G7.1C0 ; Cancels cylindrical interpolation mode.
5. The G07.1 command must be specified in a separate program segment and cannot be specified in the same program segment as other G codes; otherwise, an alarm will occur.
6. Create a rotary axis by specifying an angle.
7. When using the G07.1 command, you must specify the plane selection command, which can be specified before or after G07.1 but must be before the motion command.

8. Coordinate commands can be either incremental or absolute.
9. Tool radius compensation can be specified in cylindrical interpolation mode.
10. As the cylindrical radius decreases, the accuracy of arcs will decrease, but there will be no cumulative error.

1.12.3.2 Plane Specification

The parallel axis of the rotary axis is set using parameters (1884), (1885), (1886), etc. The plane selection command is used to specify the axis for cylindrical interpolation.

1. When the rotary axis is parallel to the X-axis, the plane specification is G17Y0C0 or G18C0Z0.
2. When the rotary axis is parallel to the Y-axis, the plane specification is G17C0X0 or G19Z0C0.
3. When the rotary axis is parallel to the Z-axis, the plane specification is G18X0C0 or G19C0Y0."

1.12.3.3 Speed Handling

In cylindrical interpolation mode, the F command is determined based on the previous modal commands for units per minute (G98) or units per revolution (G99), and whether or not the F command is specified before the G07.1 command.

1. G98 Mode Before G07.1 Command:

1) If the F command is not specified during cylindrical interpolation, the previous F command's feedrate is continued.If the F command is specified, it uses the specified F command's feedrate.
2) After cylindrical interpolation is canceled, the feedrate returns to the feedrate specified by the last F command before the start of cylindrical interpolation or the F command specified within cylindrical interpolation.

2. G99 Mode Before G07.1 Command:

1) In cylindrical interpolation, the previous F command's feedrate cannot be used, so a new F command must be specified.
2) After cylindrical interpolation is canceled, the feedrate returns to the feedrate specified by the F command before the start of cylindrical interpolation in G99 mode.

1.12.3.4 Function Combinations

1. In cylindrical interpolation mode, you can perform circular arc interpolation, but circular arc interpolation can only be specified by an R value and cannot be specified by I, J, or K commands.
2. In cylindrical interpolation mode, tool tip radius compensation can be executed. However, tool tip radius compensation can only be enabled or disabled within cylindrical interpolation mode and cannot specify the G07.1 command within the tool tip radius compensation command, or else an error will occur.
3. G07.1 command cannot be specified in constant velocity control mode, or else an error will occur.
4. T code must be specified before the G07.1 command. Specifying it within cylindrical interpolation mode will result in an error.

1.12.3.5 Notes

1. Powering off and restarting or resetting the system will cancel the cylindrical interpolation mode.
2. When canceling the cylindrical interpolation mode, tool tip radius compensation needs to be canceled as well.
3. After canceling the cylindrical interpolation mode, the plane will return to the plane that was active before the cylindrical interpolation command.
4. Program restart function cannot be used within the cylindrical interpolation mode.
5. Cylindrical interpolation or polar coordinates cannot be specified within the cylindrical interpolation mode; doing so will result in an error.
6. Threads per revolution cannot be used in cylindrical interpolation commands; threads per minute should be used instead.

1.12.4 Program Example

```
N01 G28;                    //Return to reference point
N02 T0202;                  //Tool change to tool 2
N03 G97 M63 S100;           //Cancel constant surface speed, set second spindle speed
                              to 100 RPM
N04 G00 X50. Z0.;           //Rapid positioning
N05 G98 G01 X40. F100.;     //Switch to G98 mode, rapid positioning
```

```
N06 G19 C0 Z0;              //Specify the YZ plane, G01 positioning in C and Z axis
N07 G07.1 C20.;             //Cylindrical interpolation
N08 G41;                    //Enable tool radius compensation
N09 G01 Z-10. C80. F150.;   //G01 positioning
N10 Z-25. C90.;             //G01 positioning
N11 Z-80. C225.;            //G01 positioning
N12 G03 Z75. C270. R55.;    //Circular arc command
N13 G01 C360.;              //G01 positioning
N14 G40;                    //Cancel tool radius compensation
N15 G07.1 C0;               //Exit cylindrical interpolation mode
N16 G01 X50.;               //G01 positioning
N17 G0 X100. Z100.;         //Rapid positioning
N18 M65;                    //Stop the second spindle
N19 M30;                    //End of program
```

1.13 G07.7/7.8 Real-Time Compensation

1.13.1 Function and Purpose

The application of this command is to perform certain motion segment real-time compensation in the feedforward mode.

1.13.2 Instruction Format

G07.7 or G07.8 X_ Y_ Z_ A_ B_ C_ X1=_……A6=_ P_	
X/Y/Z/A/B/C/X1……/A6	Compensation axis setting
P	P0: The compensation will remain active continuously. P1: The next motion segment of G07.7/G07.8 will be affected, and subsequent segments will deduct the compensated incremental value.
G7.7: It calculates the final travel distance based on the original motion increment and the compensation value, maintaining a constant speed. This method is less efficient. If the compensation value is significant, the speed remains constant, but the time duration increases.	

G7.8: It adds compensation motion within the original motion cycle, with constant acceleration and deceleration times. This method is more efficient. If the compensation value is N times the original motion increment, the speed will increase by N times. However, excessive acceleration may lead to motor overcurrent or "motor bursting."

1.13.3 Programming Example

1. Basic Usage

1) Specifying X/Y/Z... Coordinate Values (#M36 BIT00=0)

```
G13.9 P9510 A0 B0 Q2;      // Wait until bit 0 of #U9510 is ON before proceeding
G07.7 X1. Y1. P1;          // Compensate X by 1.0, Y by 1.0
G01 X150. Y100. F5000;     // Compensation active (151.000, 101.000)
G01 X250. Y120. F5000;     // Compensate back (250.000, 120.000)
M30;                       // End of program
```

2) X/Y/Z... Using Variable Addresses (#M36 BIT00=1)

```
G13.9 P9510 A0 B0 Q2;      // Wait until bit 0 of #U9510 is ON before proceeding
G07.7 X#U10 Y#U10 P1;      // Compensate X and Y axes using the value of #U10
G01 X150. Y100. F5000;     // Compensation active (150.000 + value of #U10,
                               101.000 + value of #U10)
G01 X250. Y120. F5000;     // Compensate back (250.000, 120.000)
M30;                       // End of program
```

2. M, T, S Codes, G04, G11, G12, G10, G13.9 (Standard Common Modes)

```
G13.9 P9510 A0 B0 Q2;      // Wait until bit 0 of #U9510 is ON before proceeding
G07.7 X1. Y1. P1;          // Compensate X by 1.0, Y by 1.0
G04 P100;                  // Pause for 100ms
G10 L981 D9510 A0;         // Set the value of #U9510 variable to 0
G11 P11;                   // Output the software O-point 11
G01 X150. Y100. F5000;     // Compensation active (151.000, 101.000)
G01 X250. Y120. F5000;     // Compensate back (250.000, 120.000)
M30;                       // End of program
```

3. Integrated Compensation

```
G13.9 P9510 A0 B0 Q2;      // Wait until bit 0 of #U9510 is ON before proceeding
G07.7 X1. Y1. P1;          // Compensate X by 1.0, Y by 1.0
```

```
G04 P100;                      // Pause for 100ms
G10 L981 D9510 A0;             // Set the value of #U9510 variable to 0
G11 P11;                       // Output the software O-point 11
G01 X150. Y100. F5000;         // Compensation active (151.000, 101.000)
G07.7 X10. Y10. P1;            // Real-time compensation for X by 10.0 and Y by 10.0
G01 X250. Y120. F5000;         // Integrated compensation result (260.000, 130.000)
G01 X200. Y100. F5000;         // Compensate back (200.000, 100.000)
M30;
```

4. Single-Axis Compensation

```
G13.9 P9510 A0 B0 Q2;          // Wait until bit 0 of #U9510 is ON before proceeding
G07.7 X1. Y1. P1;              // Compensate X by 1.0, Y by 1.0
G04 P100;                      // Pause for 100ms
G10 L981 D9510 A0;             // Set the value of #U9510 variable to 0
G11 P11;                       // Output the software O-point 11
G01 X150. F5000;               // X-axis compensation active (X=151.000)
G01 X200. Y100. F5000;         // Compensate X-axis back (200.000, 100.000)
M30;                           // End of program
```

5. Synthetic Vector with Increment of 0 Compensation

```
G13.9 P9510 A0 B0 Q2;          // Wait until bit 0 of #U9510 is ON before proceeding
G07.7 X1. Y1. P1;              // Real-time compensation: Compensate X by 1.0, Y by 1.0
G04 P100;                      // Pause for 100ms
G10 L981 D9510 A0;             // Set the value of #U9510 variable to 0
G11 P11;                       // Output the software O-point 11
G01 X150. Y100. F5000;         // Compensation active (X=151.000, Y=101.000)
G07.7 X10. Y10. P1;            // Compensate X by 10.0, Y by 10.0
G01 X150. Y100. F5000;         // Result after composite compensation (X=160.000,
                                  Y=110.000)
G01 X200. Y100. F5000;         // Compensate back (X=200.000, Y=100.000)
M30;                           // End of program
```

6. Single-axis incremental compensation with zero offset and rollback:

1)

```
G13.9 P9510 A0 B0 Q2;          // Wait for bit 0 of #U9510 to be ON before executing the
next line
G07.7 X1. Y1. P1;              // X compensation 1., Y compensation 1.
```

```
G04 P100;                    // Pause for 100ms
G10 L981 D9510 A0 ;          // Set the value of #U9510 variable to 0
G11 P11 ;                    // Output software O point 11
G01 X150. Y100. F5000;       // Compensation effective (X=151.000, Y=101.000)
G07.7 X10. Y10. P1;          // Real-time compensation: X compensation 10., Y
                                compensation 10.
G01 X150. Y150. F5000;       // Combined compensation result (X=160.000,
                              Y=160.000)
G01 X150. F5000 ;            // In this line, X-axis increment is 0, but X compensation
                              Is deducted (X=150.)
G01 Y100. F5000 ;            // Deduct Y compensation (Y=100.)
M30;                         // End of program

2)
G13.9 P9510 A0 B0 Q2;        // Wait for bit 0 of #U9510 to be ON before executing the
next line
G07.7 X1. Y1. P1;            // X compensation 1., Y compensation 1.
G04 P100;                    // Pause for 100ms
G10 L981 D9510 A0;           // Set the value of #U9510 variable to 0
G11 P11 ;                    // Output software O point 11
G01 X150. Y100. F5000;       // Compensation effective (X=151.000, Y=101.000)
G07.7 Y10. P1;               // Real-time Y compensation 1.
G01 X150. Y100. F5000;       // Combined compensation result (X is rolled back to
                                150.000, Y is newly compensated to 110.000)
G01 X200. F5000 ;            // Roll back X compensation to X=200.
G01 Y100. F5000 ;            // Deduct Y compensation to Y=100.
M30;                         // End of program
```

7. Anti-anticipatory compensation

```
G13.9 P9510 A0 B0 Q2;        // Wait for bit 0 of #U9510 to be ON before executing
                                the next line
G07.7 X1. Y1. P1;            // Real-time compensation: X compensation 1., Y
                                compensation 1.
G04 P100;                    // Pause for 100ms
G10 L981 D9510 A0;           // Set the value of #U9510 variable to 0
G11 P11 ;                    // Output software O point 11
G65 L50;;                    // Anti-anticipatory
```

```
G01 X150. Y100. F5000;       // Compensation effective (X=151.000, Y=101.000)
G07.7 X10. Y10. P1;          // Real-time X compensation 10., Y compensation 10.
G01 X150. Y150. F5000;       // Compensation result (X=160.000, Y=160.000)
G01 X200. F5000;;            // Roll back X compensation to X=200.
G01 Y100. F5000;;;           // Roll back Y compensation to Y=100.
M30;                         // End of program
```

8. G08 or G31

```
G08 ;                        // Cancel compensation not rolled back when returning
                               from the loop
G01 X50. Y50. F5000 ;        // Positioning X and Y axes (50.000,50.000)
G13.9 P9510 A0 B0 Q2;        // Wait for bit 0 of #U9510 to be ON before executing
                               the next line
G07.7 X10. Y10. P1;          // Real-time compensation: X compensation 1., Y
                               compensation 1.
G04 P100;                    // Pause for 100ms
G10 L981 D9510 A0;           // Set the value of #U9510 variable to 0
G11 P11 ;                    // Output software O point 11
G01 X150. Y100. F5000;       // Compensation effective (X=160.000, Y=110.000)
M30;                         // End of program
```

9. Handling of Other Motion Axes

```
G01 X50. Y50. F5000 ;        // Position X and Y axes
G13.9 P9510 A0 B0 Q2;        // Wait for bit 0 of #U9510 to be ON before executing the
                               next line
G07.7 X1. Y1. P1 ;           // Real-time compensation: X compensation 1., Y
                               compensation 1.
G01 Z150. F5000;             // Compensation is not effective
G07.7 X1. Y1. P1;            // Real-time compensation: X compensation 1., Y
                               compensation 1.
G01 X50.Z150. F5000;         // Even if the X-axis increment is 0, X-axis
                               compensation is effective, Y compensation is not
                               effective (51.000, 50.000)
M30;                         // End of program
```

1.13.4 Explanation

1. This is only effective in the lead-in mode.
2. It is only effective for G00 or G01, G28, G30.
3. Adjusting the compensation and retracting the motion does not affect the feed rate.
4. The G08 command clears the compensation value, cancels the compensation command, and the subsequent motion commands move according to the command position.
5. The G31 anti-crawling command clears the compensation value, cancels the compensation command, and the motion commands move according to the command position.
6. The G65 L50 command clears the compensation value, cancels the compensation command, and the subsequent motion commands move according to the command position.
7. G13.9 jumps to an abnormal subroutine, cancels the compensation command, and the program is reinterpreted. The original compensation value is cleared, and recalculations are performed.
8. M99 jumps to a new main program, and the program continues to be pre-interpreted. The compensation value is not cleared and continues to be effective.
9. G13.9 Q2 indicates that the G13.9 condition must be met before real-time compensation values become effective and are pre-interpreted.
10. G07.7 is effective when the next line contains M, T, S code, G04, G11, G12, G10, G13.9 (non-abnormal jump and Q2 anti-real-time compensation pre-crawl) instructions, etc.
11. G07.7 becomes effective when the first line below is G00, G01, G28, G30. If the instruction contains compensation axes, the compensation values for those axes become effective (regardless of whether the instruction involves incremental motion), and P1 determines whether the compensation is added or needs to be retracted. If the instruction does not involve axes with compensation, real-time compensation for those axes is not effective, and subsequent instructions are canceled (the G07.7 instruction is no longer processed).
12. If compensation has already taken effect and has not been retracted, and there is a new compensation command, the new effective compensation is the new compensation minus the amount to be retracted. In the first G00, G01, G28, G30 instruction that follows the new compensation command, only axes that require compensation become effective (regardless of whether the instruction involves

incremental motion), axes without compensation are not effective, and subsequent instructions are canceled (the G07.7 instruction is no longer processed).

13. The compensation values in G7.7/7.8 commands cannot exceed 1000 mm, exceeding this value will trigger a system alarm message (1001-70701/1001-807).

1.14 G07.9 Angle Follow Function

1.14.1 Instruction Explanation

The purpose of this command is to ensure that the tool on the rotating axis follows the machining path at a certain angle.

Related parameters: #M1879~#M1889.

1.14.2 Instruction Format

G07.9 IPxx Qxx P1 Dxx Fxx - Activate angle tracking

G07.9 P0 - Deactivate angle tracking

IPxx	Angle tracking control for the following axis (counterclockwise is positive: C90; clockwise is negative: C-90.)
Qxx	Tool length
P1	Activate angle tracking, P0 to deactivate
Dxx	G-code used when positioning at the start of a single move. Range: 0 to 1 (0 or blank for G00, 1 for G01 at the start of a single move)
Fxx	F-value used when D1 is active

1.14.3 Programming Example

- **Example 1**

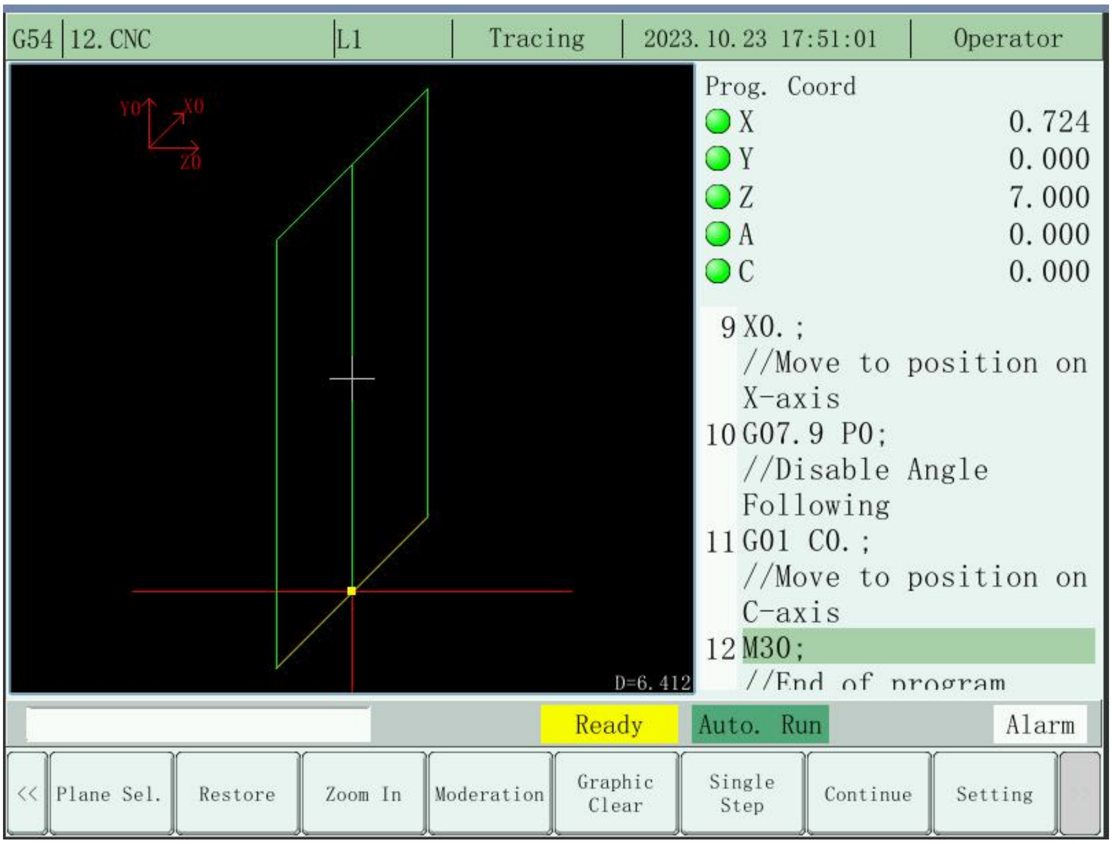


Figure 1.14.1

Diameter Programming

G17;	//Select XY Plane
M50;	//Switch the first spindle to Position Mode
G0 X0 Y0 Z0 C0;	//Rapid Positioning of X, Y, Z, and C axes
G1 Y100 F10000;	//Move to position on Y-axis using G01
G07.9 C-90. Q0. D1 P1 F10000;	//Enable Angle Following, Angle Following axis is C-axis, Angle to Follow is -90 degrees, Tool Length is 10.
G01 Y100.;	//Move to position on Y-axis using G01
X100. ;	//Move to position on X-axis
Y0.;;	//Move to position on Y-axis
X0.;	//Move to position on X-axis
G07.9 P0;	//Disable Angle Following
G01 C0.;	//Move to position on C-axis
M30;	//End of program

- **Example 2**

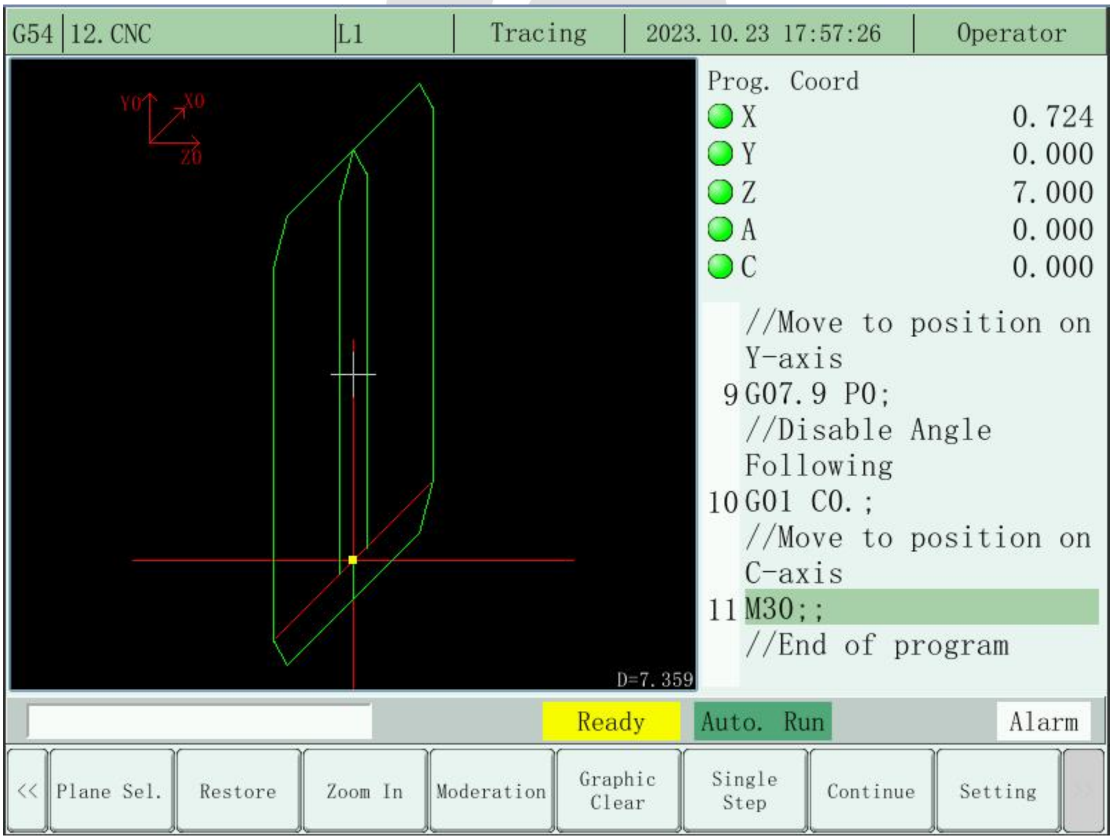


Figure 1.14.2

Diameter Programming

G17;	//Select XY Plane
M50;	//Switch the first spindle to Position Mode
G0 X0 Y0 Z0 C0;	//Rapid Positioning of X, Y, Z, and C axes
G07.9 C-90. Q10.P1 ;	//Enable Angle Following, Angle Following axis is C-axis, Angle to Follow is -90 degrees, Tool D-Length is 10.
G01 X100.;	//Move to position on X-axis using G01
Y100.;	//Move to position on Y-axis
X0.;	//Move to position on x-axis
Y0.;	//Move to position on Y-axis
G07.9 P0;	//Disable Angle Following
G01 C0.;	//Move to position on C-axis
M30;;	//End of program

- **Example 3**

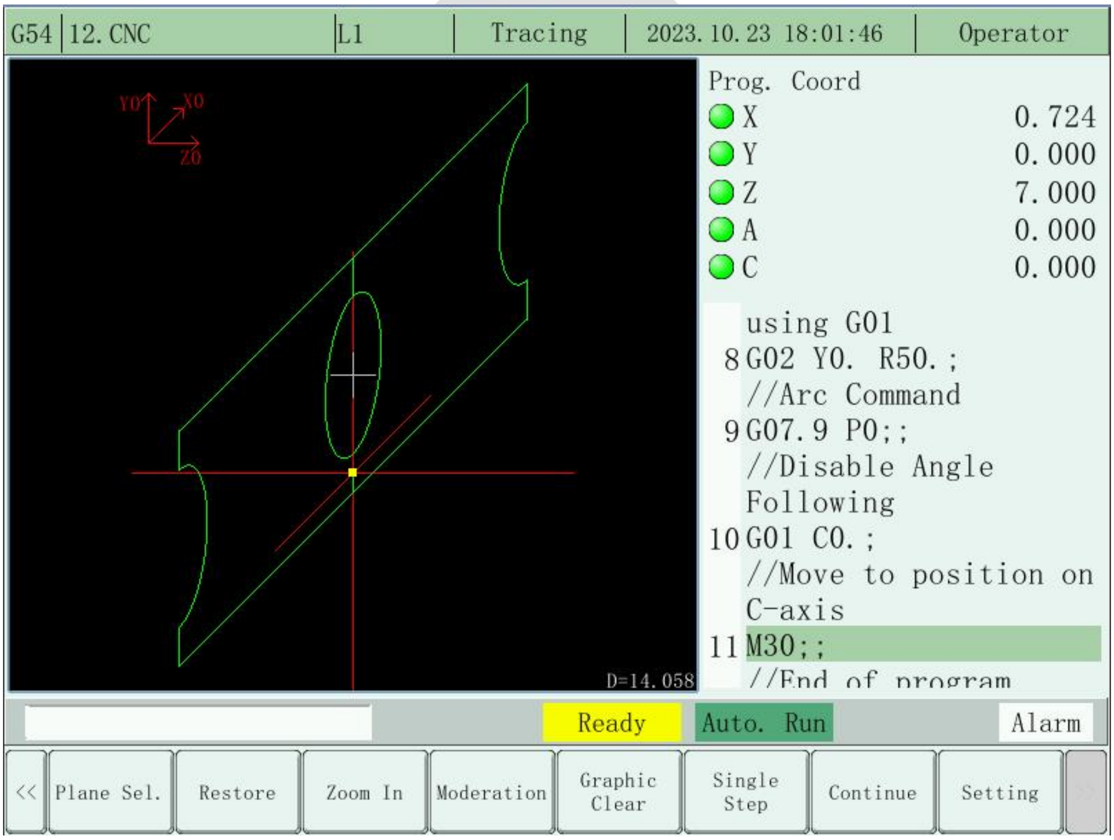


Figure 1.14.3

Diameter Programming

G17 ;	//Select XY Plane
M50;	//Switch the first spindle to Position Mode

```
G0 X0 Y0 Z0 C0;            //Rapid Positioning of X, Y, Z, and C axes
G07.9 C-90. Q10.P1 ;       //Enable Angle Following, Angle Following axis is
                            C-axis, Angle to Follow is -90 degrees, Tool
                            Length is 10.
G01 X500.;                 //Move to position on X-axis using G01
G02 Y100. R50.;;           //Circular Interpolation command
G01 X0.;;                  //Move to position using G01
G02 Y0. R50.;              //Arc Command
G07.9 P0;;                 //Disable Angle Following
G01 C0.;                   //Move to position on C-axis
M30;;                      //End of program
```

1.15 G09/G61 (Exact Stop Function)

1.15.1 Function and Purpose

1. When there is a sudden change in the tool's feed speed to mitigate mechanical collisions.
2. To prevent fillet cuts during angular cutting (or control the size of fillets).
3. To ensure the positioning state after mechanical deceleration stops or after a deceleration check time, before starting the next block of commands.
4. To accurately stop at the end of the program segment before continuing to execute the next program segment. This function is used for machining sharp corners. G09 is only effective within the specified program segment.
5. Execution after feedback check delay.

1.15.2 G09Instruction Format

G09 G01(G02 G03)

Or

G09 X__Y__Z__ Q1 Pxx.

1.15.3 Instruction Format

G61 - Activate exact stop mode.

G62 - Deactivate exact stop mode.

1.15.4 Special Notes

1. When using G09 X_ Y_ Z_.
2. It does not change the modal G code.
3. Exact stop is effective with G09 command or in G61 mode.
4. Q1 is for G09 feedback check, effective for the current line.
5. #S10124 BIT04=1 (default on, no manual setting required) enables fixed feedback check for G00 (if not needed, set a large positioning error for G00), and follows the feedback check with G09 motion commands.
6. Pxx sets the exact stop waiting time (using feedback as positioning completion signal). If the waiting time elapses and there is no feedback in position, the system will raise a warning for excessive positioning error. (Pxx takes precedence over #M4720.)

1.15.5 Example

```
M3 S500;                // Spindle rotation at 500 RPM
G01 X50. Z50.;          // G01 positioning
G01 X75.;               // G01 positioning
G09 X100. F2.0;         // Equivalent to G09 G01 X100.
Z100.;                  // G01 positioning
G83 X400. Z400. F3.;    // Drilling cycle
G09 X500. Z500.;        // Drilling cycle (equivalent to G09 G83 X500. Z500.)
G00 X600. Z600.;        // G00 positioning
M3 S500;                // G00 positioning
G92 X700. Z700. F2.;    // G92 threading cycle
X800.;                  // First pass
X900.;                  // Second pass
G09 X1000.;             // Third pass (equivalent to G09 G92 X1000 Z700. F2.)
G0 X0.;                 // G00 positioning
Z0.;                    // G00 positioning
M30;                    // Program end
```

1.16 G10.9 Diameter/Radius Programming Dynamic Switching

1.16.1 Function and Purpose

Dynamic switching between diameter and radius programming.

1.16.2 Instruction Format

G10.9 X__ Y__ Z__	
X，Y，Z	● Specify a specific axis to use diameter or radius programming mode. 0 = Radius 1 = Diameter

1.16.3 Explanation

1. When reset, it restores to the settings defined by parameters (MCM508 to MCM509).
2. Whether to restore to the diameter/radius programming settings defined by parameters MCM508 to MCM509 when the program starts is determined by parameter MCM07 BIT03.
3. When an axis switches from diameter programming to radius programming, the movement values will be doubled. Please ensure that the single-step movement values are correct to avoid tool interference and collisions.
4. When the system encounters a G10.9 command, it temporarily cancels tool compensation and resumes it in the next movement.
5. The G10.9 command is only effective for linear axes.
6. It is not allowed to use the G10.9 command in polar coordinate mode, RTCP (Rotary Tool Center Point) mode, TWPC (Tool Workpiece Point Control) mode, rotary mode, or scaling mode, as it will trigger an alarm message.

1.16.4 Programming Example

Assuming the current mode is lathe mode, and the X-axis is programmed in diameter

mode:

G98;
G01 X0 F5000;
G01 X-50. F5000; // Move to X=-50 (programmed value), mechanical position is -50, feedback is -25
G04 X2.;
G10.9 X0; // Switch X-axis to radius programming, programmed value is -25, mechanical position is -25, feedback is -25
G04 X2.
G01 X-100. F5000; // Move to X=-100 (programmed value), mechanical position is -100, feedback is -100
M30

1.17 G11 Output Point Control

1.17.1 Function and Purpose

This function is used to specify control over OUTPUT points.

1.17.2 Instruction Format

G11 P___ Q___	
P	● **Specify OUTPUT points (range 1~2048)** Positive values indicate that the specified O point is turned ON, while negative values indicate that the specified O point is turned OFF. When the P value is less than 10000, the G11 command clears the reset of the output points. When the P value is greater than 10000, the G11 command does not clear the reset of the output points.
Q	● **Specifying the channel (BIT00=1, Channel 1 BIT01 = Channel 2, etc.)** Q=-1 represents all channels. Q=0 or not specifying it designates the current channel.

1.17.3 Programming Example

Currently in a dual-channel system
1. G11 P10 Q3; Both channel 1 and channel 2 output O10 ON, and when reset, O10 turns OFF. G11 P-10 Q3; Both channel 1 and channel 2 output O10 OFF, and when reset, O10 remains OFF. 2. G11 P10010 Q3; Both channel 1 and channel 2 output O10 ON, and when reset, O10 remains in its current state. G11 P-10010 Q3; Both channel 1 and channel 2 output O10 OFF, and when reset, O10 remains in its current state.

1.18 G12 Input Point Control

1.18.1 Function and Purpose

Execution of the next block is initiated after the I-point signal is satisfied. (The awaited I-point is a software I-point.)

1.18.2 Instruction Format

G12 P___ L___ B___ A__ Q___	
P	● **Specify IUPUT point (range 1~2048):** Positive value: Execute the next block when the specified I-point is ON. Negative value: Execute the next block when the specified I-point is OFF. Note: G12 P0 and G12 P-0 are not usable.
L	● **When the letter L has a numerical value:** L=1: When the numerical value is positive, it waits for the specified I-point's rising edge and then executes the next block. L=-1: When the numerical value is negative, it waits for the specified I-point's falling edge and then executes the next block.
B	Detection time, an alarm will be triggered when the detection time is reached, unit: seconds (s)
A	● When the instruction contains the letter 'A', it means waiting for multiple I-point signals (waiting for a maximum of 32 I-point signals). 'P' represents the starting value for waiting for I-point signals, and 'A' is used to set the bits of the 32 I-points to be waited for, starting from 'P'. For example: G12 P15 A101 The binary value of 101 is 0000 0000 0000 0000 0000 0000 0110 0101. This means checking whether these 4 I-point signals - I15, I17, I20, and I21 - have all been ON at least once during one interpolation cycle before proceeding to the next block of instructions. If the waiting time elapses and these 4 I-point signals do not meet the condition, then SYS10166=15, S72=1 is generated.
Q	● **Signal Duration, measured in milliseconds (ms).** In other words, an I-point signal is considered valid only if it persists for the specified duration. If the signal doesn't persist for at least this specified time, it is regarded as interference or noise. This duration-based validation is not

	applicable when checking for rising or falling edges of the signal.

1.18.3 Programming Example

```
G0 X0
G12 P1;             //Execute the next block when the input signal, specifically software
                     input I1, is specified.
G01 U100.;
M30
```

1.19 Start/Cancel Polar Coordinate Interpolation (G12.1/G13.1)

Polar coordinates are a contour control method that converts programming instructions in Cartesian coordinates into linear axis movements (tool movement) and rotary axis movements (workpiece rotation). It is primarily used in turning for facing operations and grinding of camshafts.

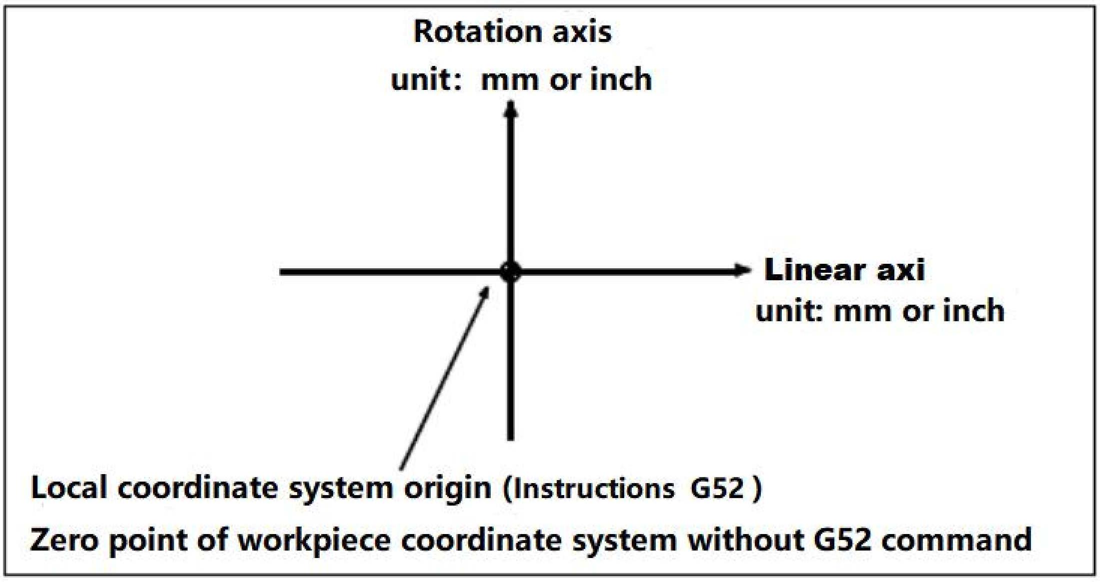


Figure 1.19.1

1.19.1 Instruction Format

- **Format 1:**

G12.1(G112); //Start polar coordinate interpolation;

…

… //Execute linear and arc interpolation using a Cartesian coordinate system consisting of linear axes and rotary axes (imaginary axes).

…

G13.1(G113); //Cancel polar coordinate interpolation;

- **Format 2**

G12.1 X__C__; //Polar coordinate interpolation based on X and C axes;

G12.1 Y__A__; //Polar coordinate interpolation based on Y and A axes;

G12.1 Z__B__; //Polar coordinate interpolation based on Z and B axes;

…

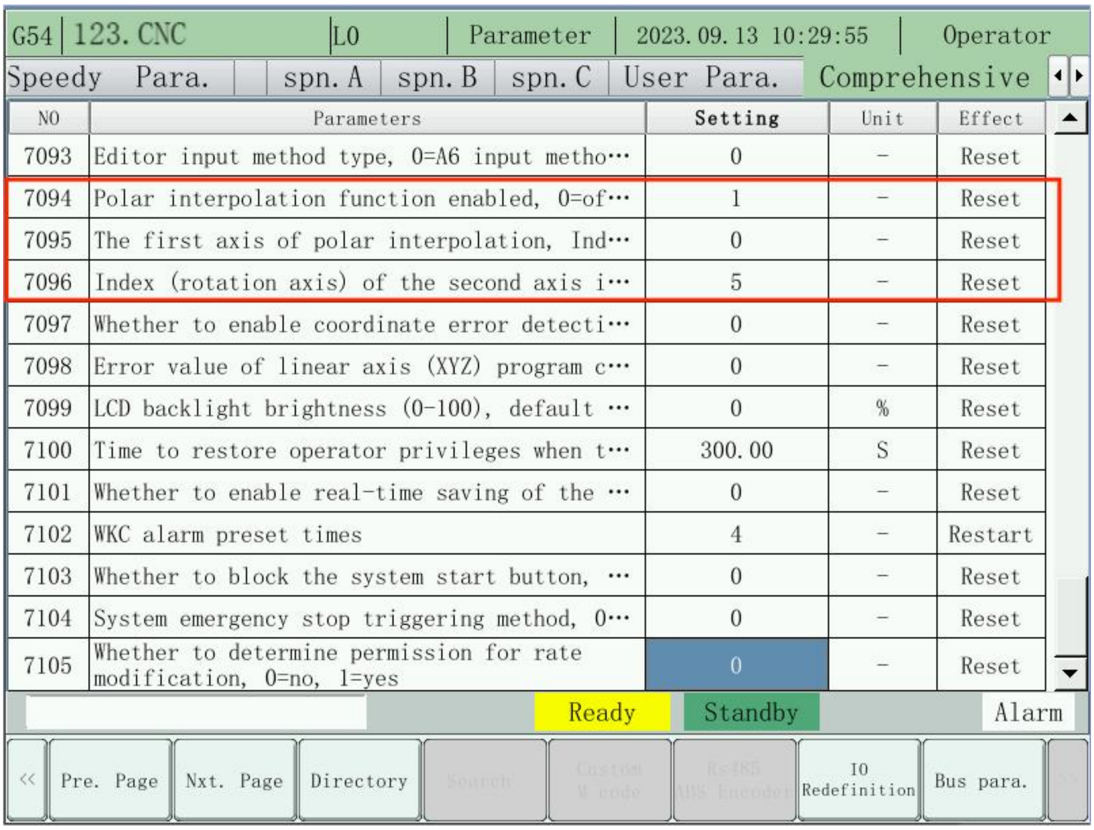
G13.1;

The axes for polar coordinate interpolation, including linear axes and rotary axes, are predefined using parameters in advance. The G12.1 command is used to activate polar coordinate interpolation mode. In this mode, the linear axis is selected as the first axis of the plane, and an imaginary axis orthogonal to the linear axis is chosen as the second axis of the plane. These two orthogonal axes together form the plane for polar coordinate interpolation

1.19.2 Notes

1. Prior to the G12.1 command, a local coordinate system or work coordinate system must be established. Plane selection commands like G17/G18/G19 used before G12.1 will be temporarily suspended and then restored after the execution of the G13.1 command.
2. In G12.1 mode, the coordinate system must not change (G50/52/53/54/55/56/57/58/59, relative coordinate reset, etc.), or else it will trigger an alarm message G5x-1.
3. Switching between polar coordinate G12.1/G13.1 is not allowed while in G41/42 mode. You can only execute G12.1/G13.1 after canceling G40, or else the system will generate alarm messages 112-1 or 113-1.

4. After a system reset (RESET), the polar coordinate interpolation mode is canceled and reverted to the plane specified by G17/G18/G19.
5. The second axis of the polar coordinate plane must be set as a rotary axis (virtual axis).
6. The positioning mode for the second axis, the rotary axis, should be set to the nearest positioning mode.
7. In polar coordinate interpolation, the speed component of the rotary axis may still exceed the range of the maximum cutting feed rate due to diameter or radius reduction. In such cases, the system will internally limit the rotary axis speed and simultaneously restrict the linear axis speed.
8. To enable polar coordinate functionality, you can configure parameters by going to Parameter Setting → Comprehensive Parameters → Polar Coordinate Parameters.



NO	Parameters	Setting	Unit	Effect
7093	Editor input method type, 0=A6 input metho…	0	-	Reset
7094	Polar interpolation function enabled, 0=of…	1	-	Reset
7095	The first axis of polar interpolation, Ind…	0	-	Reset
7096	Index (rotation axis) of the second axis i…	5	-	Reset
7097	Whether to enable coordinate error detecti…	0	-	Reset
7098	Error value of linear axis (XYZ) program c…	0	-	Reset
7099	LCD backlight brightness (0-100), default …	0	%	Reset
7100	Time to restore operator privileges when t…	300.00	S	Reset
7101	Whether to enable real-time saving of the …	0	-	Reset
7102	WKC alarm preset times	4	-	Restart
7103	Whether to block the system start button, …	0	-	Reset
7104	System emergency stop triggering method, 0…	0	-	Reset
7105	Whether to determine permission for rate modification, 0=no, 1=yes	0	-	Reset

Figure 1.19.2

A Cartesian-to-Polar coordinate interpolation program based on the X-axis (linear axis) and an imaginary axis would typically look something like this:

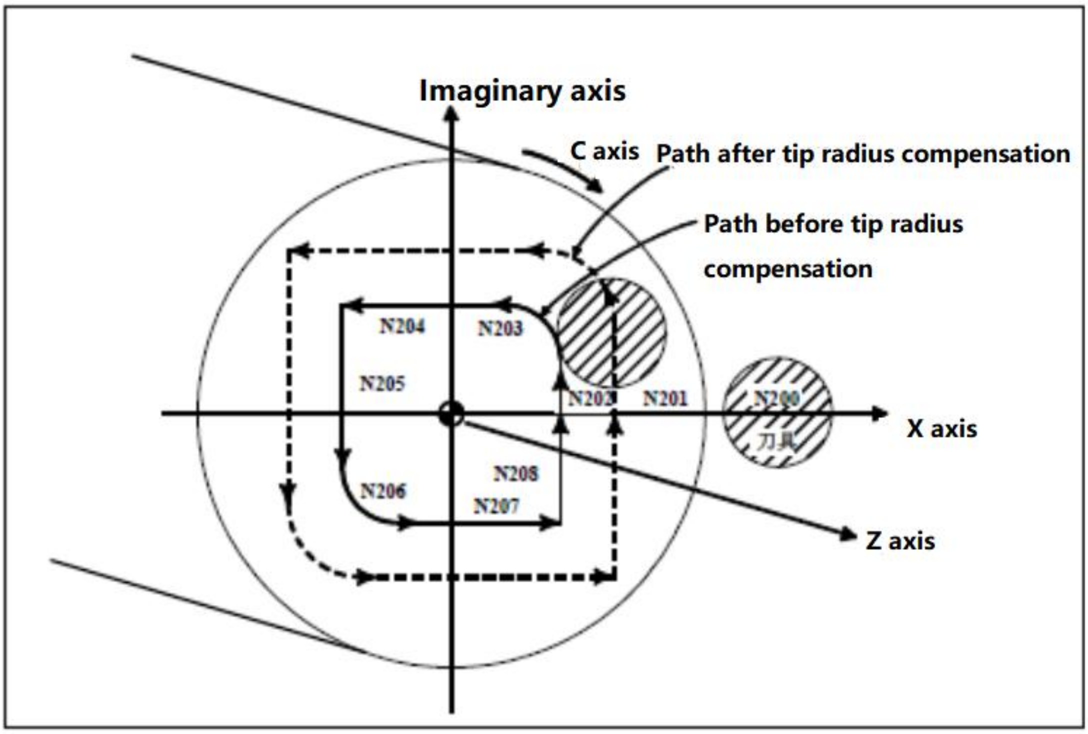


Figure 1.19.3

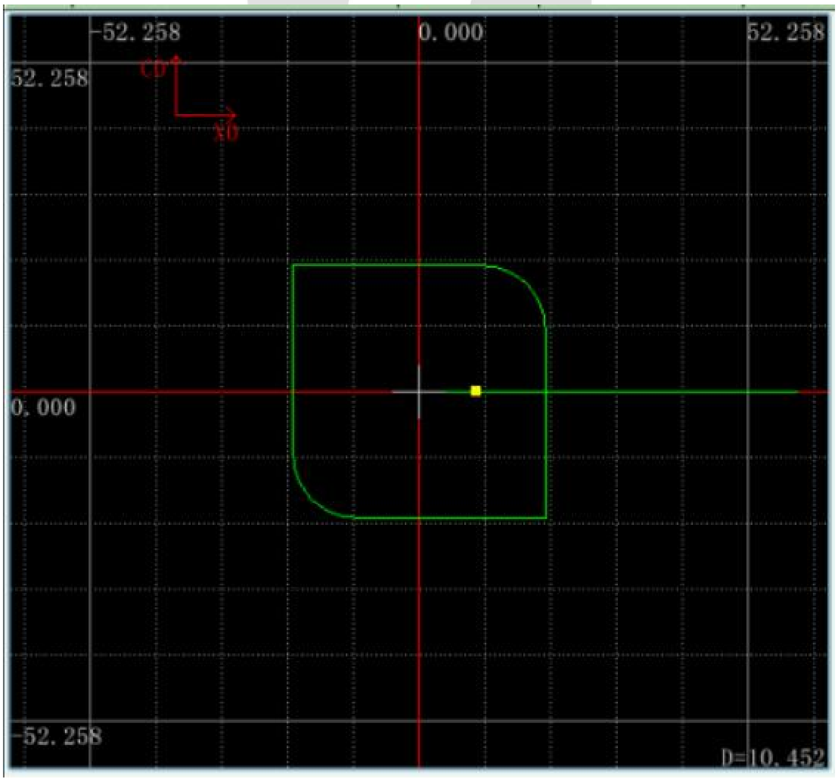


Figure 1.19.4

The X-axis is specified as the diameter, and the C-axis is specified as the radius

G98; // Switch to G98 mode

```
N010 T0101;;                         // Tool change to tool 1
…
M50;                                 // Switch the main spindle to position mode
N0100 G00 X120.0 C0 Z10.;            // Rapid positioning to the starting position
N0200 G12.1;                         // Enable polar coordinate interpolation
N0201 G42 G01 X40.0 F100;;           // Enable tool radius compensation and position
                                        X-axis
N0202 C10.0;;                        // Position the C-axis
N0203 G03 X20.0 C20.0 R10.0;;        // Circular arc command
N0204 G01 X-40.0;;                   // Position the X-axis
N0205 C-10.0;;                       // Position the C-axis
N0206 G03 X-20.0 C-20.0 I10.0 J0;;   // Circular arc command
N0207 G01 X40.0;                     // Position the X-axis
N0208 C0;                            // Position the C-axis
N0209 G40 X120.0;                    // Position the X-axis
N0210 G13.1;                         // Disable polar coordinate interpolation
N0300 Z20.;                          // Position the Z-axis
N0400 X120.;                         // Position the X-axis
…
N0900 M30;
```

Example: Machining Internal Hexagon

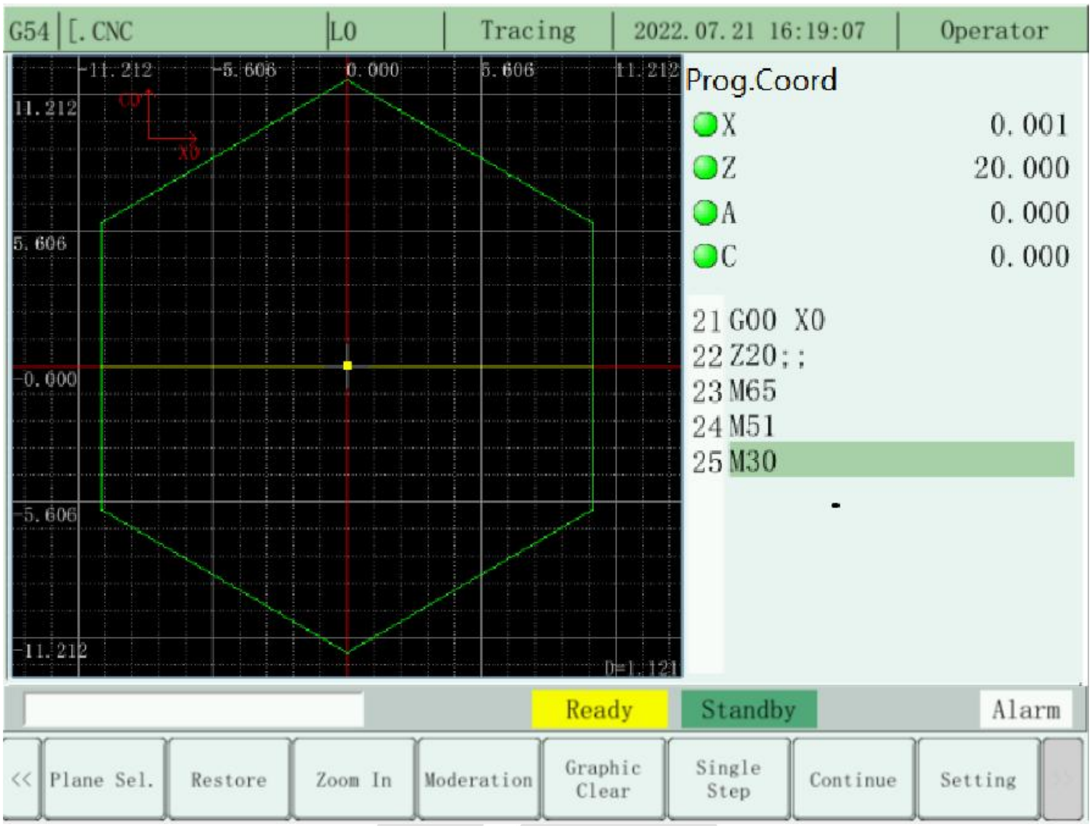


Figure 1.19.5

```
T01;                        //Tool change: Select Tool 1
M63 S2500;                  //Rotate the second spindle at 2500 RPM
G98;                        //Switch to G98 mode
G00 X00 Z10;                //Position to the starting point.
M50;                        //Switch the C-axis to positioning mode.
G0 C0 X5.5;                 //Rapidly position the C-axis to 0, and position the
                              X-axis to 5.5.
G01 Z5 F500;                //Position the Z-axis to...
G12.1;                      //Enable polar coordinate interpolation.
G41;                        //Enable tool tip radius compensation command.
G01 X20.5 C0 F150;          //Position the X-axis and C-axis.
G01 X20.5 C5.917 F150;      //Position the X-axis and C-axis.
X0 C11.835 F150 ;           //Position the X-axis and C-axis.
X-20.5 C5.917 F150;         //Position the X-axis and C-axis.
X-20.5 C0;                  //Position the X-axis.
X-20.5 C-5.917;             //Position the X-axis and C-axis.
X0 C-11.835;                //Position the X-axis and C-axis.
X20.5 C-5.917;              //Position the X-axis and C-axis.
```

```
X20.5 C0.;;        //Position the X-axis.
G40;               //Cancel the tool tip radius compensation command.
G13.1;;            //Cancel polar coordinate interpolation.
G00 X0;            //Rapid positioning of the X-axis.
Z20;;              //Rapid positioning of the Z-axis.
M65;               //Stop the second spindle.
M51;               //Switch the C-axis to speed mode.
M30;               //End of program.
```

1.20 Check Variable Bit Status (G13.9)

1.20.1 Function and Purpose

This function allows for the quick examination of the status of any variable bit. It enables the execution of specific actions based on the status of the object being checked. When the checked object's status meets the conditions specified by the G13.9 command, the system immediately proceeds to the next program block. Conversely, when the checked object's status does not meet the conditions set by G13.9, the system will remain in the G13.9 program block, waiting for the specified conditions to be met. This functionality is not influenced by pre-buffered program data and can provide real-time responses to the status conditions of the relevant objects.

1.20.2 Instruction Format

G13.9 P___ A___ B___ H___ D___ L___ C___ Q___ E___	
P	● **Specify the address corresponding to the object you want to check.** When the P value has more than 6 digits, the higher digits represent the channel number, i.e., P=xx yyy yyy, where xx represents the channel number. When xx=0, it denotes the current channel. For example: P=01 000 001, B=1, represents channel 1; 000 001 (decimal) indicates the address of the specified User Variable (USR) channel.

	P=02 000 001, B=2, represents channel 2; 000 001 (decimal) indicates the address of the specified Machine Parameter (MCM) channel. P=03 000 001, B=3, represents channel 3; 000 001 (decimal) indicates the address of the specified System Variable (SYS) channel. P=04 000 001, B=4, represents channel 4; 000 001 (decimal) indicates the address of the specified Register (REG) channel.
A	● **Specify which bit position of the corresponding address you want to check.** An address for a variable can store 32 different bit states. After determining the variable address for the object you want to check, you need to specify which bit position within that variable you want to check. When A has a positive value, the program will proceed only if the object's status is ON. When A has a negative value, the program will proceed only if the object's status is OFF.
B	● **Specify the address type for P:** 0 or blank: Absolute address. 1: User variable of the channel. 2: Mechanical parameter of the channel. 3: System variable of the channel. 4: Register of the channel. 10: Public BUS data. 11: Public user variable. 12: Public mechanical parameter. 13: Public system variable. 14: Public register.
H	● **The letter A checks the type as follows:** H=0 or not specified: Checks the bit value. It verifies whether the bit Axxx in the specified address P is in the ON state.

	H=1: Checks for equality. It verifies whether the value in the specified address P is equal to the value specified by A. If they are equal, the next instruction is executed. H=2: Checks for inequality. It verifies whether the value in the specified address P is not equal to the value specified by A. If they are not equal, the next instruction is executed.
D	● **When G13.9 conditions are not met, the program path to which it should jump can be specified with the letter D as follows:** If D is not specified, or D0, or D1, it means using the path specified in the [/home/root/hust/usr/sys00xx/config] file under the [M98] / D1 section.
L	● **When G13.9 conditions are not met, the program path to which it should jump can be specified with the letter D as follows:** If D is not specified, or D0, or D1, it means using the path specified in the [/home/root/hust/usr/sys00xx/config] file under the [M98] / D1 section.
C	● **When G13.9 conditions are not met, you can specify the program number to which it should jump using the letter C as follows:** C, followed by <XXX>, specifies the program name and extension in the same format as M98 Pxxxxx/<XXX>.
Q	BIT00=0 means that L__ specifies Nxx, and BIT00=1 means that L__ specifies the line number as the xx value.
E	● **The waiting time for G13.9 is in milliseconds and is only effective when C is specified.** If E is not specified or set to E=0, it means no waiting. In other words, when the condition is satisfied, the next line of instruction is executed. If the condition is not met, the program jumps to the program C__/<XXX>. If E is greater than 0, it means that if the condition is not met after waiting for E milliseconds, the program will jump to the program C__/<XXX>. If Q is greater than 0, it means that if the condition is not met after waiting for Q milliseconds, the program will jump to the program C__/<XXX>.

1.20.3 Programming Example

1.	G13.9 P10 A1;	// Check the status of BIT01 in variable 10 of the current channel.
		// When the status is ON, the program immediately proceeds to the next block of code.
		// When the status is OFF, the system remains in the G13.9 block, waiting for the status to turn ON before continuing execution.

2.	G13.9 P2 000 010 A-3 B4;	//Check the status of BIT03 in register 10 of channel 2.
		// When the status is OFF, the program immediately proceeds to the next block of code.
		// When the status is ON, the system remains in the G13 block, waiting for the status to turn OFF before continuing execution.

1.21 Spindle Positioning (G15.9)

1.21.1 Function and Purpose

When the spindle is running, the system will automatically perform dynamic positioning; otherwise, it will perform static positioning.

1.21.2 Instruction Format

G15.9 R__ P__ Q__ H__	
R	● **Positioning Angle** If not specified, the system defaults to positioning using the angle specified in parameter COM40118. Range: 0 to 360,000
P	● **Positioning Speed** If set to 0 or not specified, the system defaults to using the speed specified in the most recent COM40116 or COM40119. Range: 0 to 99,999,999
Q	● **Positioning Mode** Range: 0 to 3 (0 = position along the previous spindle rotation direction, 1 = position along the positive spindle direction, 2 = position along the negative spindle direction, 3 = position spindle closest), dynamic positioning settings do not affect the positioning mode, and it follows the current direction of spindle rotation.
H	● **Specify the Spindle ID for Positioning (1 = Spindle 1, 2 = Spindle 2, ...)** Range: 0 to the maximum spindle number
Note: After stopping the spindle with G15, the system does not automatically clear COM40115 BIT01. Before restarting the spindle, PLC must first check if COM40115 BIT01 is not 0. If it is not 0, it needs to be cleared to 0, and then a delay of 2 PLC cycles is required (since the spindle is in the common channel to ensure that it is executed at least once), and then the spindle can be restarted.	

1.21.3 Programming Example

```
G98;                    //Switch to G98 mode
M51;                    //Switch the first spindle to speed mode
M3 S100;                //Start the spindle in the forward direction at 100 RPM
G4 X3.0;                //Delay for 3 seconds
G15.9 R90 P50 Q0 H1;;   //Position the spindle at an angle of 90 degrees, at a
                        positioning speed of 50 RPM, in the direction of spindle
                        rotation, for the first spindle
M5;                     //Stop the spindle
M30;                    //End of program
```

1.22 Work Plane Setting (G17/G18/G19)

This command is used to select the control plane or the plane in which arcs are located.

1.22.1 Instruction Format

Command	Horizontal Axis	Vertical Axis
G17	X	Y
G18	Z	X
G19	Y	Z

Table 1.22.1

1.23 Inch/Metric Unit Setting (G20/G21)

The system will automatically handle unit conversion when switching between metric and imperial units. After the conversion, the following operation units will change accordingly:

1. Displayed coordinates and speed units.
2. Incremental jog units.
3. Manual pulse generator (MPG) jog units.

1.23.1 Instruction Format

```
G20;              // Set units to inches (imperial).
G21;              // Set units to millimeters (metric)
```

1.24 Return to Reference Point （G28）

- When the G28 command is executed, the tool moves at the rapid traverse rate (typically G00) to a specified intermediate point before automatically returning to the reference point, which is usually the machine's home position. The primary purpose of this is to ensure that the tool returns to the reference point while avoiding collisions with the workpiece.

- In the case of absolute positioning, X and Z are used to specify the absolute coordinates of the intermediate point. When using incremental positioning, U and W represent the incremental distances from the starting point to the intermediate point.

1.24.1 Instruction Format

G28 IP__

"IP" is used to specify the command for defining the intermediate point. You can specify this point using either absolute values or incremental values. You can omit the address codes for certain axes, which means those axes won't return to the reference point. If you omit all the address codes, the tool will remain stationary.

1.24.2 Additional Explanation

MCM2000: G28 Set First Reference Point for X-axis.
MCM2001: G28 Set First Reference Point for Y-axis.
MCM2002: G28 Set First Reference Point for Z-axis.
MCM2003: G28 Set First Reference Point for A-axis.
MCM2004: G28 Set First Reference Point for B-axis.
MCM2005: G28 Set First Reference Point for C-axis.

1.24.3 Notes

1. The position of the first reference point is set by MCM parameters [G28 First Reference Point] for the X and Z axes.
2. In the instruction format, the values for X and Z indicate the intermediate points that

the tool should pass through on the corresponding axis. If the G28 instruction is standalone, the tool will automatically return to the reference point position specified by MCM parameters X and Z. If both X and Z instructions are included, the tool will automatically return to the intermediate points specified by MCM parameters X and Z before returning to the reference point position.

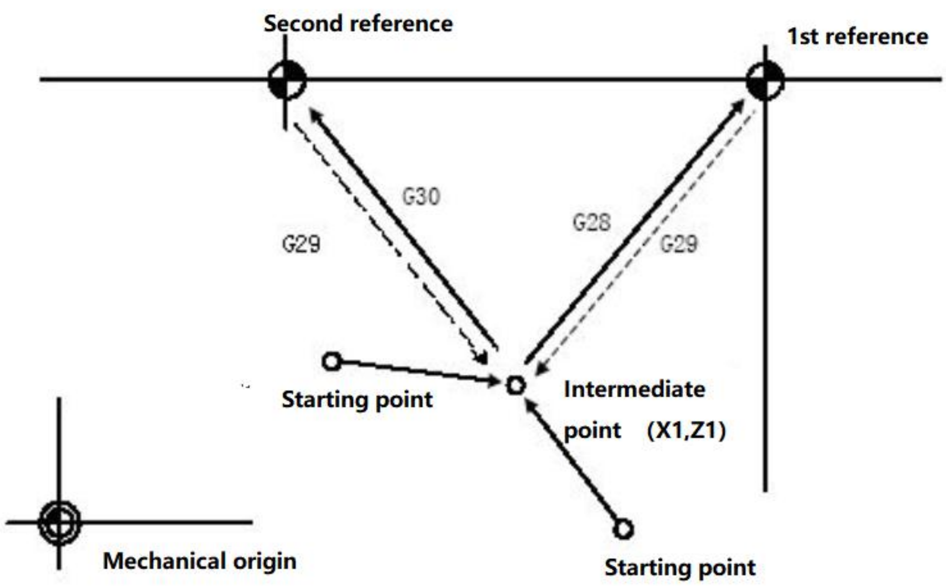
3. Before executing the G28 instruction, tool compensation must be canceled.

1.24.4 Example

Before executing the G28 command, tool compensation must be canceled.

T00; //Cancel tool compensation;

G28 X1.0 Z1.0; //Return to the middle point for X and Z (X1.0, Z1.0);



1.25 Automatic Return to Reference Point (G29)

- The G29 command is used to move rapidly from the reference point to a specified position point after using G28. It's important to note that G29 cannot be used on its own because it doesn't specify its own intermediate point; instead, it relies on the intermediate point specified in the G28 command. Therefore, before executing the G29 command, you must first execute the G28 command.

- In absolute value mode, X and Z represent the absolute coordinates of the target point you want to reach. In incremental value mode, U and W represent the incremental distances from the intermediate point to the target point.

1.25.1 Instruction Format

G29 IP__

IP: Instruction specifying the coordinates of the return position, either in absolute or incremental values. When IP is not specified, no action is taken.

1.25.2 Supplement Explanation

1. When the G29 command is specified, the midpoint position for the return of G29 is determined by the last executed G28 or G30 command. If there were no intermediate points set in the previous G28 or G30 commands, the G29 command will rapidly position to the location specified in the last positioning command preceding G28 or G30.

2. In single-block execution mode, and when intermediate points are set, running G28, G30, or G29 commands will pause at the intermediate point. It will resume and continue to the reference point position only after pressing the start button.

1.25.3 Example 1

```
N1 G00 X10.0 Z10.0;          //Axis reaches the specified position;
N2 G28 X-10.0 Y-10.0 Z-10.0  //First move to the intermediate point (X-10.0, Y-10.0,
                               Z-10.0), then to the reference point;
…
N3 G29;                      //Pause at the current position;
…
```

1.25.4 Example 2

```
N1 G00 X10.0 Z10.0;          //Axis reaches the specified position;
N2 G28 X-10.0 Y-10.0 Z-10.0  //First move to the intermediate point (X-10.0, Y-10.0,
```

Z-10.0), then to the reference point;

…

N3 G29 X30.0 Y20.0 Z10.0 // Axis moves through the intermediate point and then reaches the specified position (X30.0, Y20.0, Z10.0);

1.26 Second Reference Point (G30)

Similar to G28, the G30 command is used to instruct the axes to rapidly return to the second reference point.

1.26.1 Instruction Format

G30 P__IP__

P2: Return to the second reference point.
P3: Return to the third reference point.
P4: Return to the fourth reference point.
IP: Specify the command for the return position, either in absolute or incremental values.
The usage of this command is similar to G28, but the coordinates of the reference points are set by MCM parameters.

1.27 Skip Instruction (G31)

When specified standard input signals or fast input signals are detected by the system, the system rapidly stops motion for the specified axis or all axes, as required.

1.27.1 Instruction Format

G31 X__Y__Z__F__P__Q__E__

X, Y, Z: Specify the axes of motion (incremental values can be used, such as U, V, W).
F__: Set the interpolation feedrate.
P__: Set the function selection, unaffected by decimal points.
P=Blank or 10000: Indicates that Table uses the MPG interface's fast interrupt Input0 (5th

pin) as the external interrupt signal for G31.

P=10001: Indicates that Table uses the MPG interface's fast interrupt Input1 (16th pin) as the external interrupt signal for G31.

P=0-511: Indicates that Table uses a software-defined i-point as the external interrupt signal for G31.

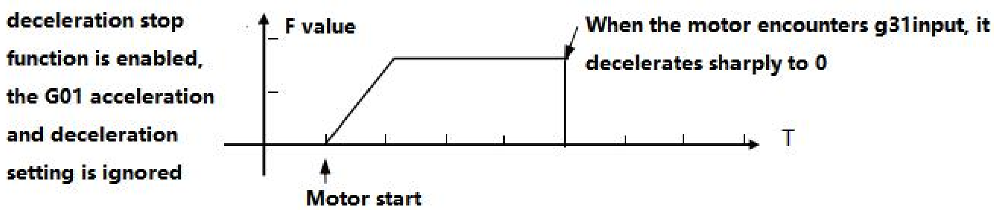
Q__: When the interrupt condition specified by P__ is encountered, this axis stops directly, while the remaining axes continue to move. When Q is not specified, Table indicates that all motion axes stop directly, regardless of decimal points (Q0 indicates stopping the X-axis).

E__: Set the composite velocity of the motion axes (including the axis specified by Q) when the interrupt condition specified by P__ is encountered.

1.27.2 Supplement Explanation

1. The G31 command is applied based on the specified conditions. It terminates motion on the specified axis or all axes when the specified condition is input, and the motion state is similar to the G01 motion command.
2. Response time: The time it takes for the specified axis to respond to the G31 signal. The response time for standard inputs is 6ms, and for fast inputs, it's 3ms.
3. There are three ways to stop the axis in response to a G31 signal, which are set by Mcm7202, Mcm7212, and Mcm7222:

   1) Instantaneous deceleration stop: The system interrupts the command directly, without acceleration or deceleration processing, and the motor stops on its own.



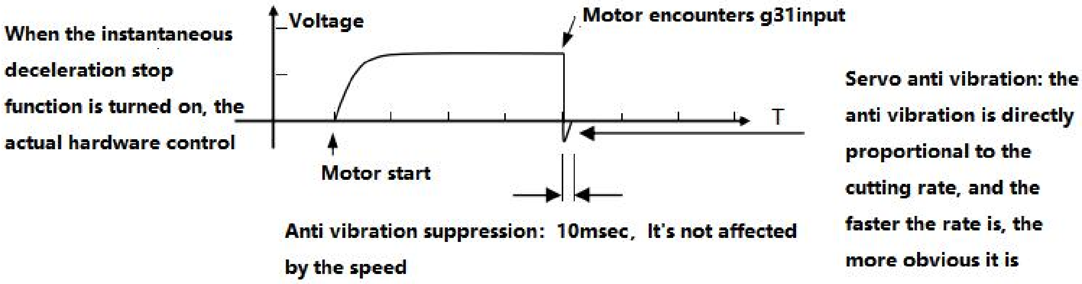


Figure 1.27-1

2) Stop according to acceleration and deceleration: The system stops the axis based on the G01 acceleration and deceleration time.

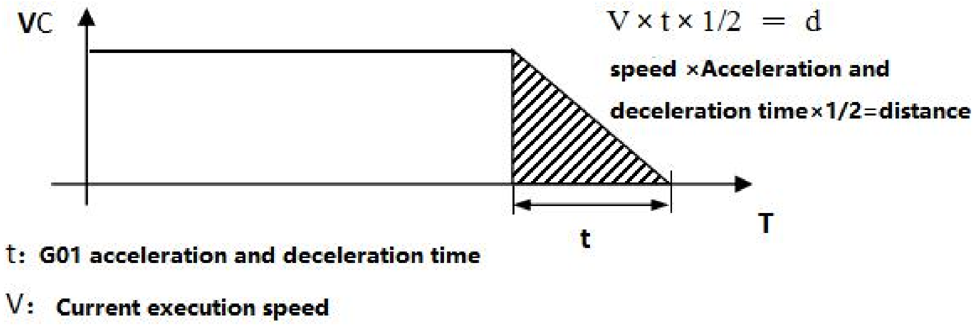


Figure 1.27-2

3) Stop according to deceleration distance: The system stops based on the deceleration distance set in the parameters.

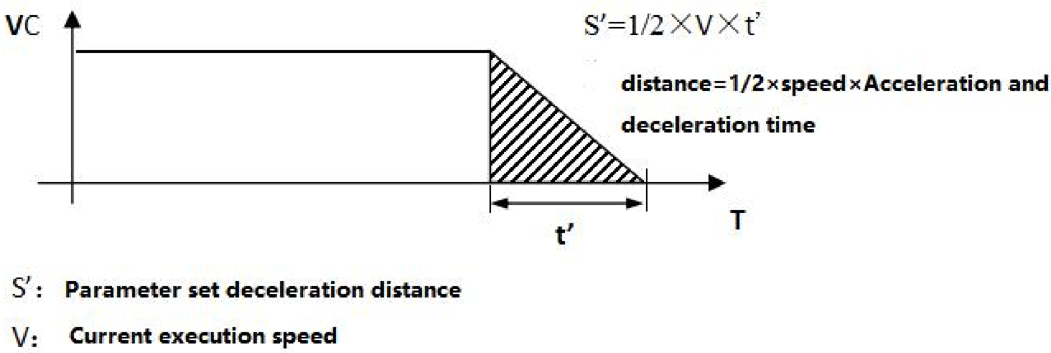


Figure 1.27-3

The minimum movement distance of the motor after G31 signal input, denoted as "s" (s = V × T).

s: The minimum movement distance of the motor after G31 signal input.

T: Response time.

Program command G31 X100. F12000. P01;

Using a standard Input, with a response time of 6ms, and an instantaneous deceleration stop.

s = V × T = 12000mm/min × 6ms = 1.2mm.

4. When using the G31 function, the system will reorganize the coordinate system after detecting the G31 signal input to ensure correct program execution.

   Example program:

   G0 X0. Z0.;

   G31 Z100. F3000. P1;

   G01 U30.;

   G01 X50. Z120.;

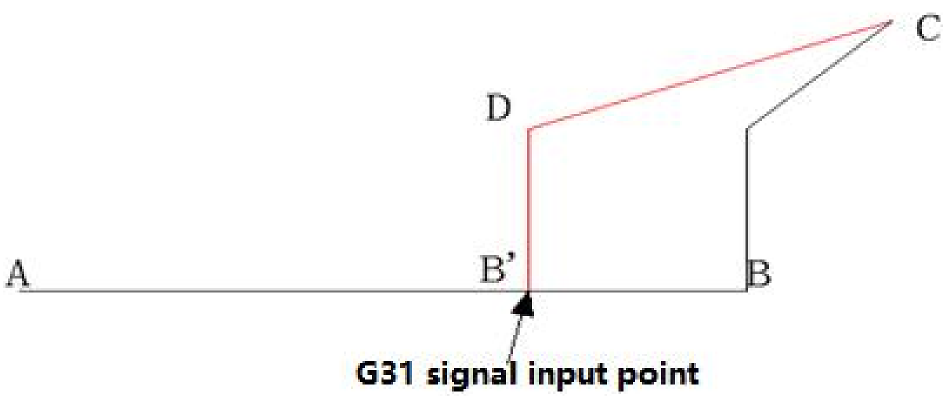


Figure 1.27-4

   G31 signal not input: Execute the path A → B' → B → C.

   G31 signal input: Execute the path A → B' → D → C.

   Coordinate reorganization is performed at point B'.

5. G31 0 signal: Short-circuit MPG pins 5 and 26, making G31 0 signal an input state.

   G31 1 signal: Short-circuit MPG pins 16 and 26, making G31 1 signal an input state.

   G31 standard INPUT signal: A standard INPUT signal is present.

   G31 signal has four triggering modes, each set by Mcm7201, Mcm7211, Mcm7221.

   Using Mcm7201 as an example:

Mcm7201 = 0: Rising edge trigger.

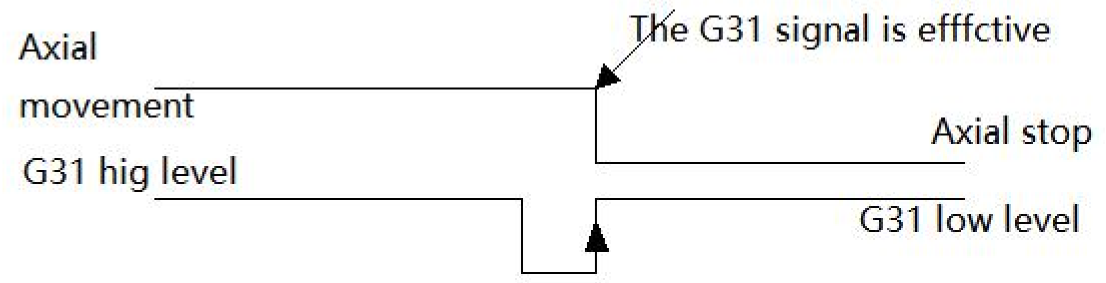


Figure 1.27-5

Mcm7201 = 1: Falling edge trigger.

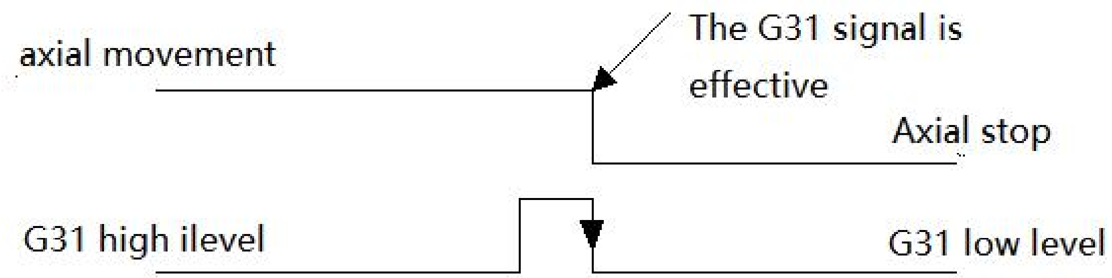


Figure 1.27-6

Mcm7201 = 2: High-level trigger.

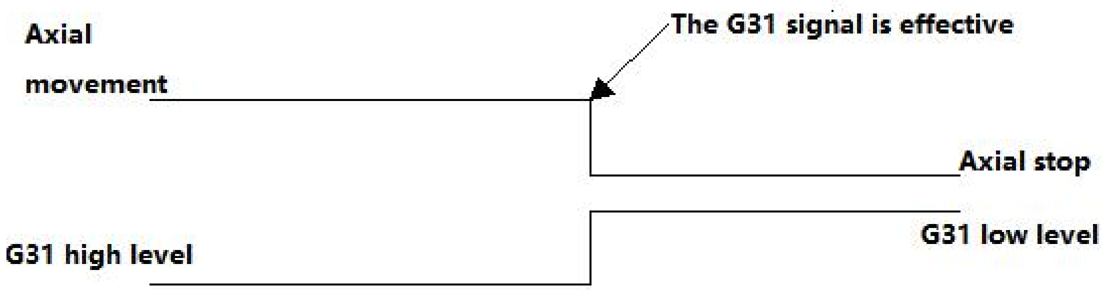


Figure 1.27-7

Mcm7201 = 3: Low-level trigger.

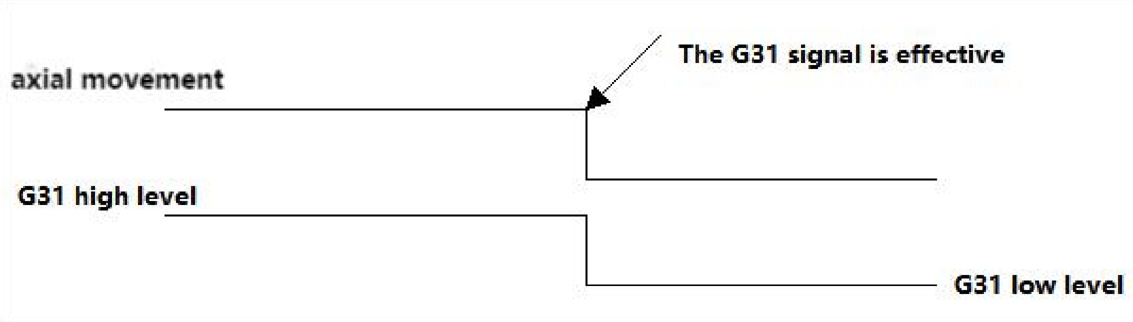


Figure 1.27-8

6. When the G31 signal is a fast Input signal, the system can filter the signal, which means how long the G31 fast Input signal must be active to be considered valid. The filter time is set by Mcm7200 and Mcm7210. When set to 0, no filtering is applied, and the maximum filtering time can be set to 510μs.

1.27.3 Example

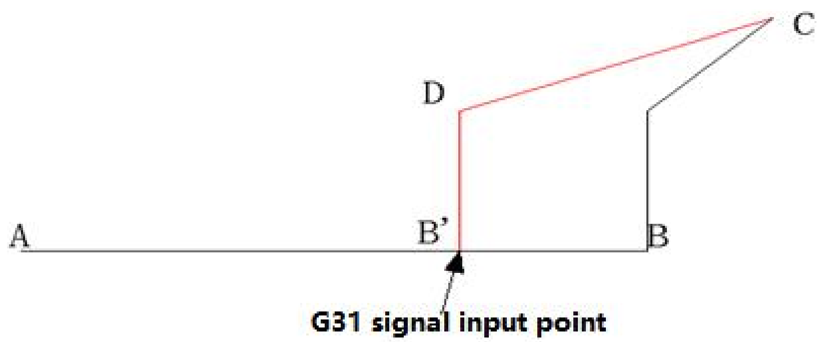


Figure 1.27-9

Based on the specified conditions, when the specified condition is input, it stops the motion along the specified axis, with the motion pattern similar to a G01 motion command.

```
G00 X0 Z0;
G31 Z100.0 F3000 P1;
G01 U30.0;
G01 X50.0 Z120.0;
```

When the G31 signal is not input, it follows the path: A → B' → B → C.

When the G31 signal is input, it follows the path: A → B' → D → C.

1.28 Thread Cutting (G32)

The G32 command is used for controlling the feed of a tool that rotates synchronously with the main spindle. This allows for machining processes such as cutting fixed-pitch straight threads, tapered threads, face milling spiral threads, and continuous threading.

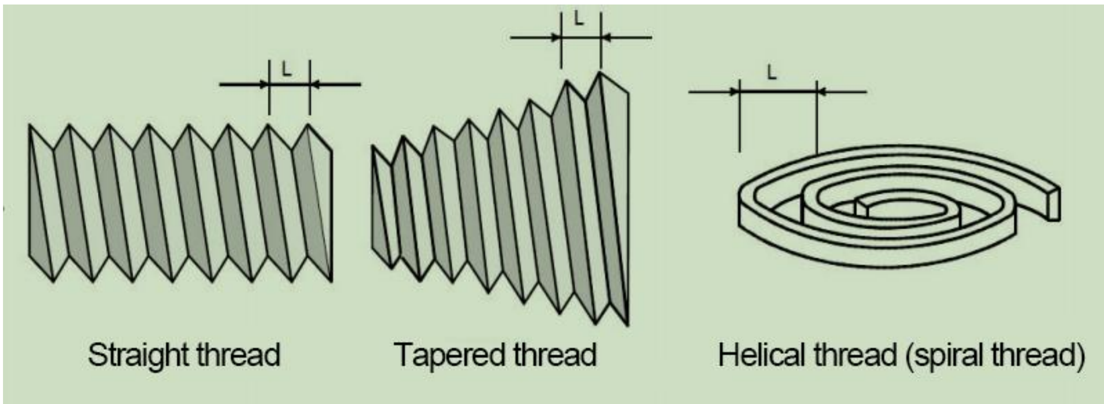


Figure 1.28-1

1.28.1 Instruction Format

G32 Z(W)__X(U)__F(E)__Q__

Z(W), X(U): The absolute (or incremental) coordinates of the thread endpoint.
F(E): The lead in the long-axis (the axis with the most movement) direction, specified in inches per thread (can be a decimal).
Q: The threading start offset angle (0.000° to 359.999°). This represents the offset angle between the spindle's one-turn signal and the thread cutting starting point. The valid range is 0 to 360 degrees, with 1-degree increments. The Q value is non-modal, meaning it must be specified each time it's used. If not specified, the system assumes a starting angle of 0 degrees.

1.28.2 Code Explanation

1. G32 is a modal G code.
2. When the X or U coordinates of the start and endpoint are the same, it performs straight thread cutting.

3. When the Z or W coordinates of the start and endpoint are the same, it performs helical thread cutting.
4. When both the X and Z coordinates of the start and endpoint are different, it performs tapered thread cutting.

1.28.3 Notes

1. Do not use constant linear velocity control in the tapered thread cutting and straight thread cutting commands.
2. Keep the spindle speed constant during the transition from roughing to finishing. In thread cutting, spindle override and cutting feed override are ineffective and fixed at 100%.
3. At the beginning and end of thread cutting, errors in lead due to servo system delays can occur. Therefore, when specifying the thread length to be cut, you must specify the length after adding the error lead lengths δ1 and δ2 to the desired thread length.
4. Thread cutting commands do not support pause, single-block execution, handwheel prediction, or MFO adjustment.
5. The thread cutting starting angular displacement angle is non-modal. When Q is not specified, it is assumed to be Q0. Changing the Q value allows multiple thread cutting operations within the same command. Q values less than 0 or exceeding 360.000 result in a program error.
6. If the previous block had thread cutting, even in the current block being threaded, tool synchronization is not checked at the beginning of the cut; it starts cutting directly.
7. Thread cutting blocks and the preceding move blocks cannot specify chamfers or corners.

1.28.4 Example 1

Equal diameter threading:

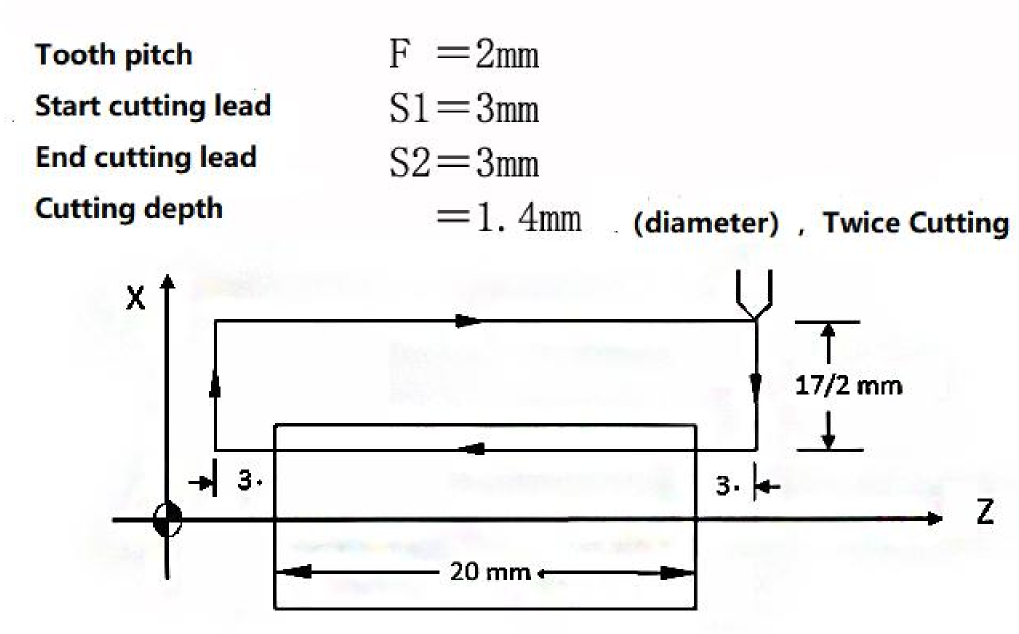


Figure 1.28-2

Diameter Programming

```
N01 G99;                      // Switch to G99 mode
N10 G00 X30.0 Z50.0;          // Rapid positioning
N20 M03 S2000;                // First spindle forward at 2000 RPM
N30 G00 U-17.000;             // First cut 1.0/2mm;
N40 G32 W-26.000 F2.00;       // Threading the first pass
N50 G00 U17.000;              // Retract X-axis
N60 W26.000;                  // Retract Z-axis
N70 G00 U-17.400;             // Second cut 0.4/2mm;
N80 G32 W-26.000 F2.00;       // Threading the second pass
N90 G00 U17.400;              // Retract X-axis
N100 W26.000;                 // Retract Z-axis
N110 M05;                     // Stop the spindle
N120 M02;                     // Program end
```

1.28.5 Example 2

Variable diameter helical thread cutting:

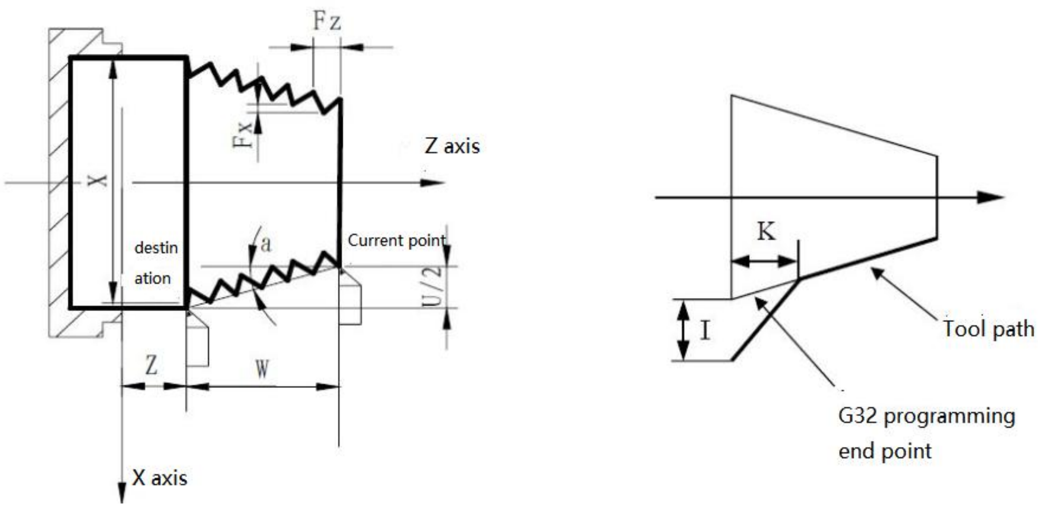


Figure 1.28-3

Diameter Programming

```
N01 G99;                                   // Rapid positioning
N20 M03 S2000;                             // Start the first spindle at 2000 RPM
N30 G00 X23.000 Z72.000;                   // First cutting pass, 1.0/2mm;
N40 G32 X32.000 Z28.000 F2.00 R-4.5;       // First thread cutting pass
N50 G00 X40.000;                           // Retract the X-axis
N60 Z72.000;                               // Retract the Z-axis
N70 G00 X22.600;                           // Second cutting pass, 0.4/2mm;
N80 G32 X31.600 Z28.000 F2.00;             // Second thread cutting pass
N90 G00 X40.000;                           // Retract the X-axis
N100 Z72.000;                              // Retract the Z-axis
N110 M05;                                  // Stop the spindle
N120 M30;                                  // Program end
```

1.28.6 Example 3

Multi-segment continuous thread cutting:

```
G00 Z0.0;              // Rapid positioning to the starting point;
M03 S2000;             // Start the first spindle at 2000 RPM
G32 Z-50.0 F1.0;       // First thread segment, find spindle encoder sync point,
                         seamlessly connect with the following G32 segment
G32 Z-100.0 F2.0;      // Second thread segment, seamlessly connect with the first
                         segment
G32 Z-150.0 F3.0;      // Third thread segment, seamlessly connect with the second
```

```
                             segment
G00 X60.0;                  // Retract from the threaded section, seamlessly connect with
                               the retraction
Z20.0;                      // Rapid positioning of the Z-axis
X20.0;                      // Rapid positioning of the X-axis
M05;                        // Stop the first spindle
M30;                        // Program end
```

In the above three segments of thread machining, the Z-axis does not pause, resulting in a smooth connection of the machined threads.

1.29 Variable Lead Threading (G34)

Variable lead thread cutting can be performed by specifying the increase or decrease in lead per revolution of the thread.

1.29.1 Instruction Format

G34 Z(W)__X(U)__F(E)__K__Q__

Z(W)、X(U)：Thread endpoint.

F, E: Thread pitch.

K: Incremental change in thread pitch per revolution.

Q: Starting angular displacement for thread cutting (0.000° to 359.999°).

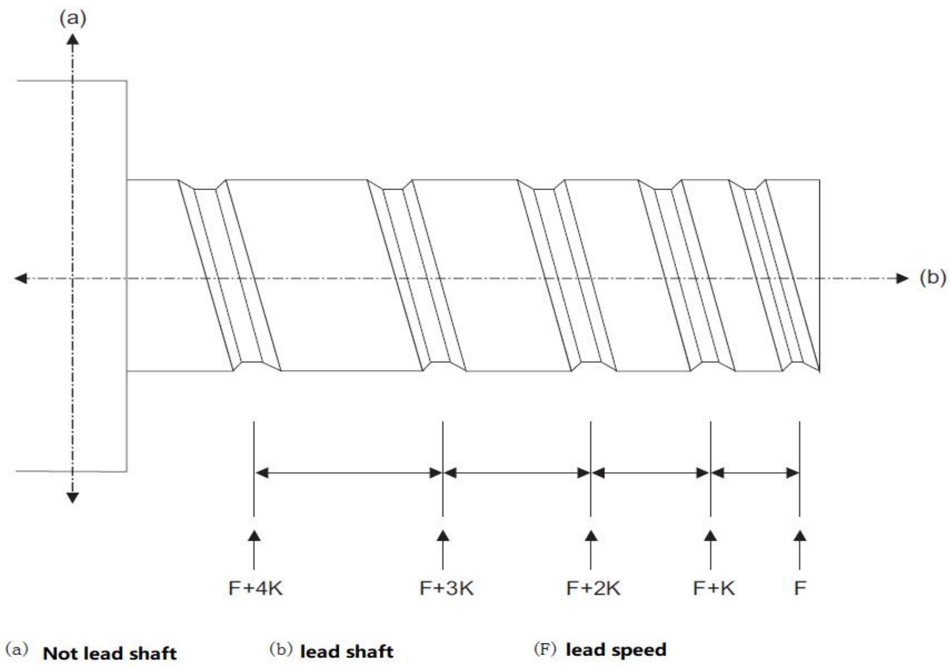


Figure 1.29-1

When K is positive, the pitch increases with each revolution, and when K is negative, the pitch decreases with each revolution.
G34 can also be used for machining continuous threads, tapered threads, and end face vortex threads, and its usage is similar to G32.

1.29.2 Notes

1. When the input for the reduction in pitch per revolution results in a pitch smaller than 0, a [Thread pitch cannot be negative] alarm will be generated. If the calculated feed rate per minute exceeds the maximum cutting feed rate and the speed is restricted, the pitch will shrink, leading to a [Thread exceeds maximum cutting speed] alarm.
2. Total cumulative movement in a single segment: F(n+1) + [n + n(n-1)/2]. Where n is the number of revolutions.
3. Other notes are the same as for G32.

1.29.3 Example

Diameter Programming

G99; // Tool change to tool 3
M03 S1000; // Start the main spindle in the clockwise direction at

```
                                    1000 RPM
M08;                                // Turn on coolant
G00 X20.0 Z0.0;                     // Rapid positioning to the starting point
G34 Z-4.5 F1.0 K0.5;                // First segment of variable pitch threading
G34 Z-9.0 F1.0 K0.5;                // Second segment of variable pitch threading
G32 Z-13.0 F2.0;                    // Third segment of constant pitch threading
G00 X40.0;                          // Rapid positioning along the X-axis
Z10.0;                              // Rapid positioning along the Z-axis
X20.0;                              // Rapid positioning along the X-axis
M09;                                // Turn off coolant
M05;                                // Stop the main spindle
M30;                                // End of program
```

1.29.4 Related Alarms

32/34-1: Neither X nor Z axis has an increment.

32/34-2: F value is 0 (1003-32/34 G32/G34 speed is 0).

32/34-3: The Nth variable pitch is less than 0.

32/34-4: Q value must be in the range of 0° to 360°.

32/34-5: Both F and E values are specified simultaneously (1003-132/134 G32/G34 F and E values are specified together).

1.30 Circular Thread Cutting (G35/G36)

Allows for the cutting of threads with a longitudinal lead.

1.30.1 Instruction Format

G35 Z(W)__X(U)__ R(I__K__)__F(E)__Q__

X(U): X-axis arc endpoint coordinates (absolute value in the X work coordinate system, increment from the current position U).

Z(W): Z-axis arc endpoint coordinates (absolute value in the Z work coordinate system, increment from the current position W).

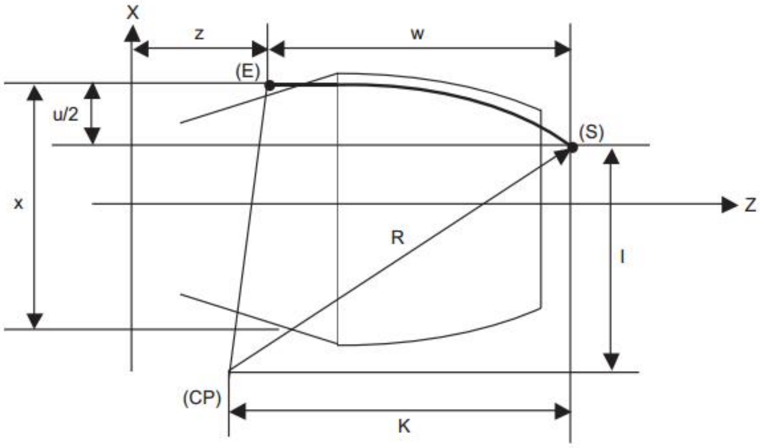
I: Increment value from the arc starting point to the center.

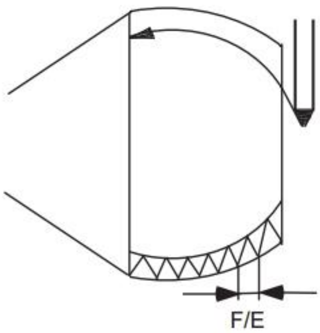
K: Increment value from the arc starting point to the center.

R: Arc radius.

F(E): Feedrate in the direction of the long axis (the axis with the largest movement).

Q: Thread cutting starting displacement angle (0.000° to 359.999°).





(a) Circular arc thread

Figure 1.30-1

Diameter Programming

```
G99;                              // Switch to G99 mode
M3 S1000;                         // Start the first spindle at 1000 RPM
G18;                              // Specify the ZX plane
G0 Z10.;                          // Rapid positioning of the Z-axis to 10.0
X10.;                             // Rapid positioning of the X-axis to 10.0
G36 X10. Z-20. R15. F0.1;         // Execute arc thread cutting
G32 Z-30.;                        // Execute thread cutting
G36 X10. Z-60. R15. F0.1;         // Execute arc thread cutting
G36 X10. Z-70. R5. F0.1;          // Execute arc thread cutting
M5;                               // Stop the first spindle
G0 X100.;                         // Rapid positioning of the X-axis to 100.0
```

Z10.; // Rapid positioning of the Z-axis to 10.0
M30; // End of program

1.30.2 Notes

1. Alarm when the start and end positions are the same or when the central angle of the arc is greater than 180°.
2. When the start and end radii positions are different: if the error is greater than the set arc error parameter, an alarm will occur. If the error is less than the set arc error parameter, the position where the start and end radii are the same is taken as the center of the arc.
3. Alarm when the sign of R__ is negative.
4. Alarm when I, K, and R instructions are not specified.
5. When thread cutting begins, if the cutting speed is greater than the clamping speed, an alarm will occur.
6. During thread cutting, in order to maintain the lead, there may be instances where the cutting feed rate exceeds the clamping speed. In this case, an error message is recorded, but thread cutting continues. Automatic stop and corresponding alarms occur before the arc thread cutting instruction in the second program.
7. Alarm when G35/G36 instructions are issued for axes other than the selected plane.
8. Alarm during thread cutting or when there is a rotation angle R, and a C instruction for thread cutting in the next program segment.

1.31 Tool Nose Radius Compensation Commands (G40/G41/G42)

The tool's tip is typically rounded, so when a program is executed, an imaginary tool tip point is used as the front end of the tool. Therefore, during inclined or arc cutting, there may be errors between the shape that the program cuts and the actual cutting shape due to the curved shape of the tool tip. Tool nose radius compensation is a function that automatically calculates errors and performs compensation by setting the tool nose radius.

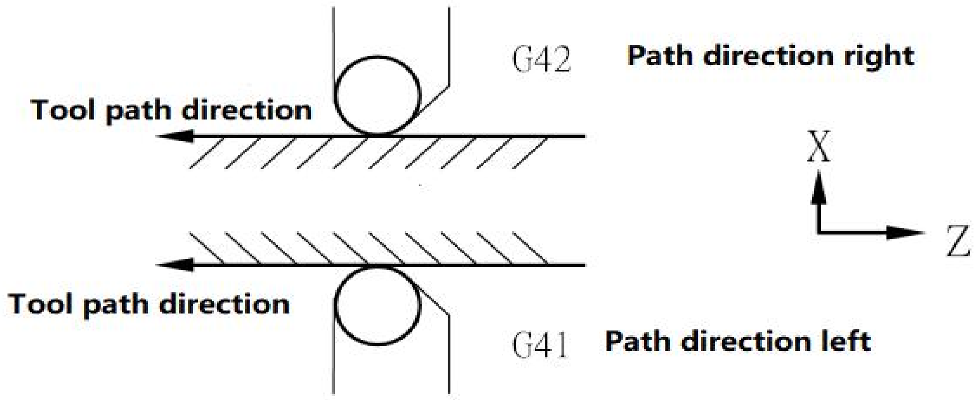


Figure 1.31-1

1.31.1 Instruction Format

T*******	// Call the compensation tool number, tool length compensation, and tool tip wear compensation;
G41 (G42)X/U_Z/W_	// Compensation settings;
G40	// Cancel compensation;

T__: Select the tool number before use.
G41: Compensate the tool nose radius to the left.
G42: Compensate the tool nose radius to the right.
G40: Cancel tool nose radius compensation.
X, Z: Specify position coordinates (absolute mode).
U, W: Specify position coordinates (incremental mode).

Note: When using tool nose radius compensation, you need to specify a tool number. Refer to [1.36 Tool Compensation Commands: T-code Instructions] for detailed T-code usage explanations.

1.31.2 Tool Nose Radius Compensation

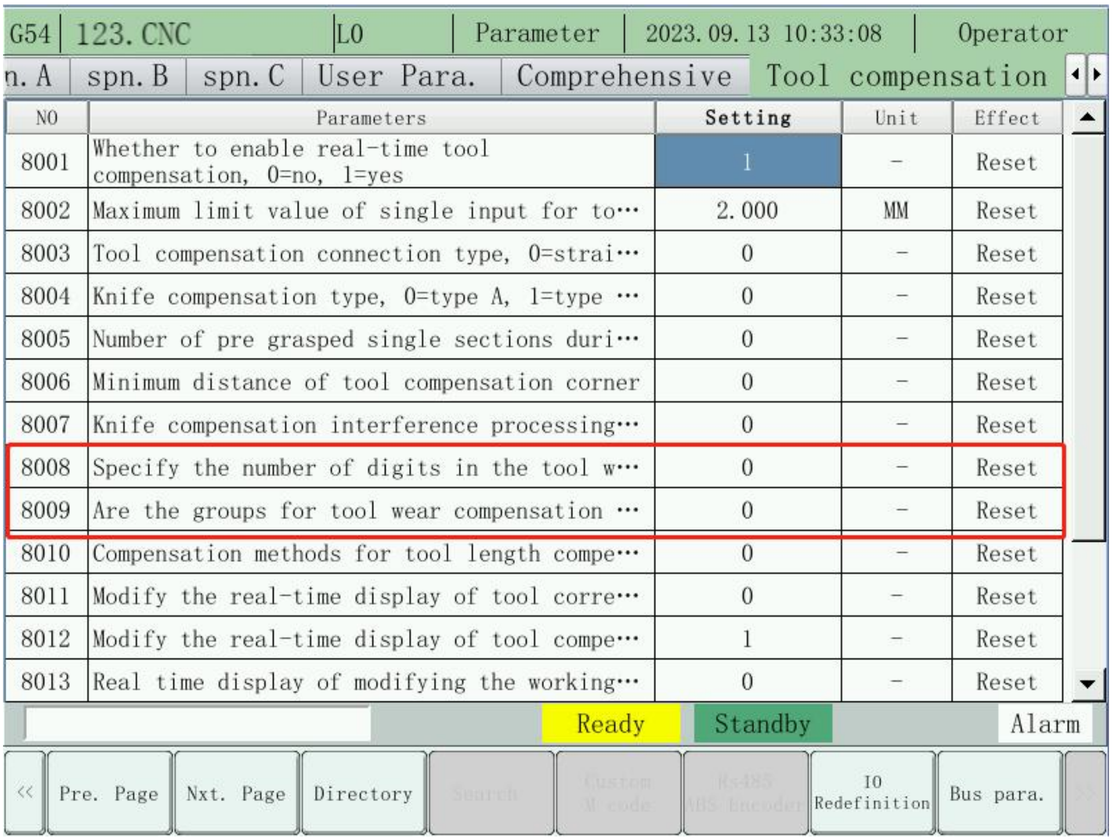
Tool nose radius compensation, also known as the T function, is used to specify the tool number and tool compensation number. Tool compensation includes tool length compensation and tool tip wear compensation. Through Mcm1705, both tool length compensation and tool tip wear compensation can be specified with tool numbers or T specified for the lower N bits of tool tip wear compensation, and the first M bits specified for tool length compensation. N is specified by Mcm1706.

1. Parameter Setting:

Use Mcm1705 and Mcm1706 to set the format specified by the T command.

Mcm1705: Sets the number of bits occupied by the tool tip wear group, i.e., N.
Mcm1706: Specifies whether tool tip wear compensation and tool length compensation are uniformly specified.

Machine Position → Parameter Setting → Tool Compensation Parameters



NO	Parameters	Setting	Unit	Effect
8001	Whether to enable real-time tool compensation, 0=no, 1=yes	1	–	Reset
8002	Maximum limit value of single input for to···	2. 000	MM	Reset
8003	Tool compensation connection type, 0=strai···	0	–	Reset
8004	Knife compensation type, 0=type A, 1=type ···	0	–	Reset
8005	Number of pre grasped single sections duri···	0	–	Reset
8006	Minimum distance of tool compensation corner	0	–	Reset
8007	Knife compensation interference processing···	0	–	Reset
8008	Specify the number of digits in the tool w···	0	–	Reset
8009	Are the groups for tool wear compensation ···	0	–	Reset
8010	Compensation methods for tool length compe···	0	–	Reset
8011	Modify the real-time display of tool corre···	0	–	Reset
8012	Modify the real-time display of tool compe···	1	–	Reset
8013	Real time display of modifying the working···	0	–	Reset

Figure 1.31-2

2. Detailed Explanation

1) Maximum Tool Numbers and Groups: FINGER CNC can have a maximum of 160 tool numbers. Each tool number can be divided into up to 40 groups, and each group can correspond to an effective axis channel. Mcm2560 to Mcm2599 are used to set the corresponding axes for each group. For example, if you set X, Y, and Z axes as effective axes, and Mcm2560 = 2, Mcm2561 = 5, Mcm2562 = 7, then X, Y, and Z

axes will correspond to the 2nd, 5th, and 7th groups of each tool number.

2) Tool Length and Tool Tip Wear Compensation: Tool length compensation and tool tip wear compensation for each tool are set using common variables Com10000 to Com22799.

3) Alarm for Tool Number Exceedance: If you set a tool number greater than 160, the system will generate an alarm.

4) Multiple Channels and Compensation: In a multi-channel setup, since each tool can have up to 40 groups, different channels can share the same tool number while executing different tool length and tool tip wear compensations.

5) Activation of Tool Compensation: Tool compensation can be activated in three different ways, determined by Mcm1707.

   Mcm1707 = 0: Compensation based on movement commands, meaning compensation occurs when there are displacement commands.
   Mcm1707 = 1: Compensation based on coordinate offsets, meaning compensation occurs when executing T codes.
   Mcm1707 = 2: Tool tip wear compensation occurs when executing T codes, and tool length compensation occurs when there are displacement commands.

1.31.3 Tool Length Compensation

Tool length compensation is applied to the program's reference position, which is typically located at either the center position of the tool turret or the tool tip position of a reference tool.

1. **Tool Length Compensation Setup:**
1) Setup for Tools Located at the Center Position of the Tool Turret.

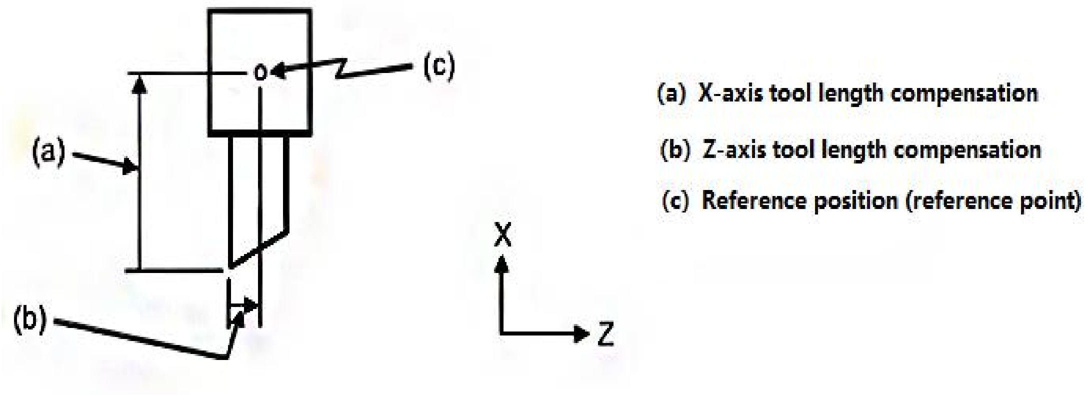


Figure 1.31-2

2) Setup for Tools Located at the Tool Tip Position of a Reference Tool.

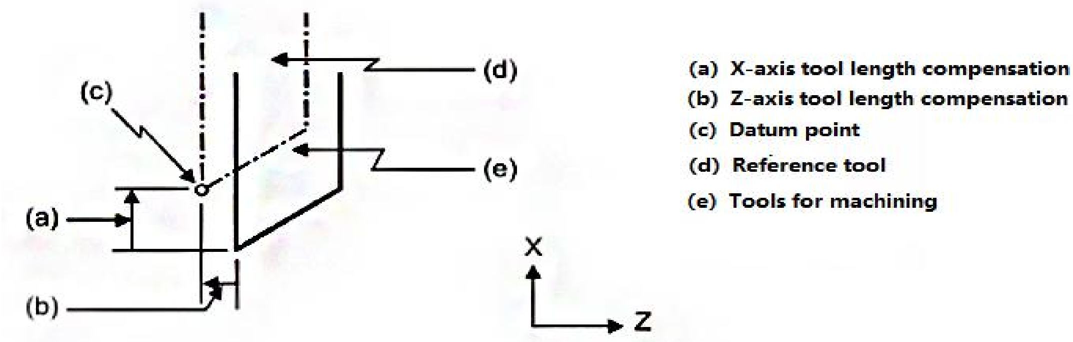


Figure 1.31-3

2. Changing Tool Length Compensation Number:

When changing the tool number, the compensation amount associated with the new tool number is added to the toolpath of the machining program.

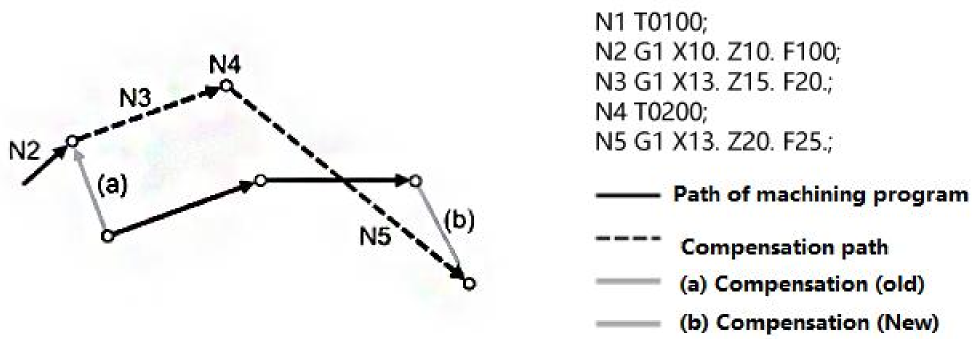


Figure 1.31-4

The diagram shows the compensation action of changing the tool number compensation during the program segment with motion commands.

3. Canceling tool length compensation:

1) Specify compensation number T0.

When the tool length compensation number in the T command is set to 0, the compensation is canceled.

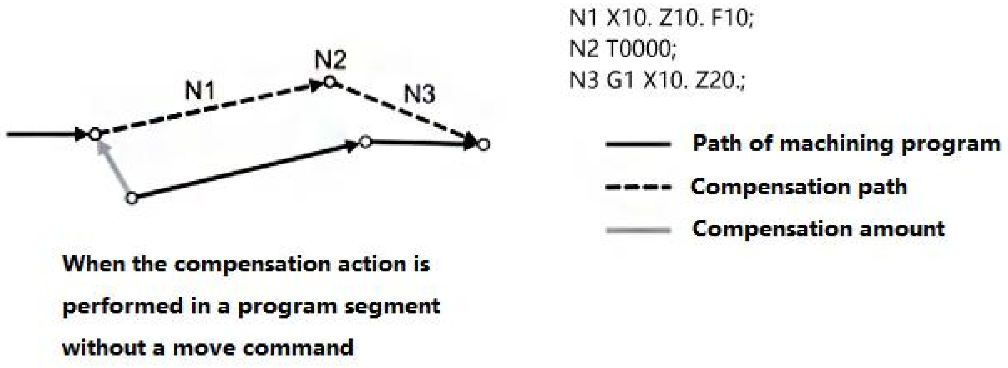


Figure 1.31-5

2) When the specified compensation amount is 0.

When the compensation amount for the tool length compensation number in the T command is set to 0, the compensation is canceled.

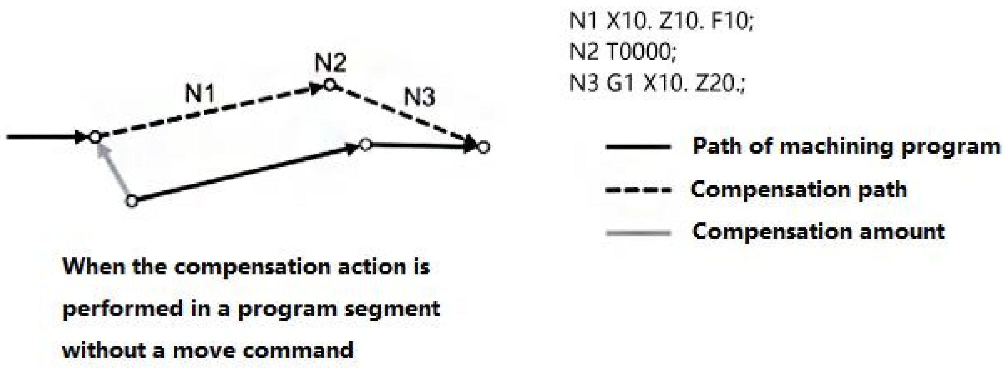


Figure 1.31-6

3) Cancel the compensation by pressing "Reset."

1.31.4 Tool Tip Wear Compensation

1. Tool Tip Wear Compensation Setting:

When wear occurs at the tip of the tool, compensation can be applied.

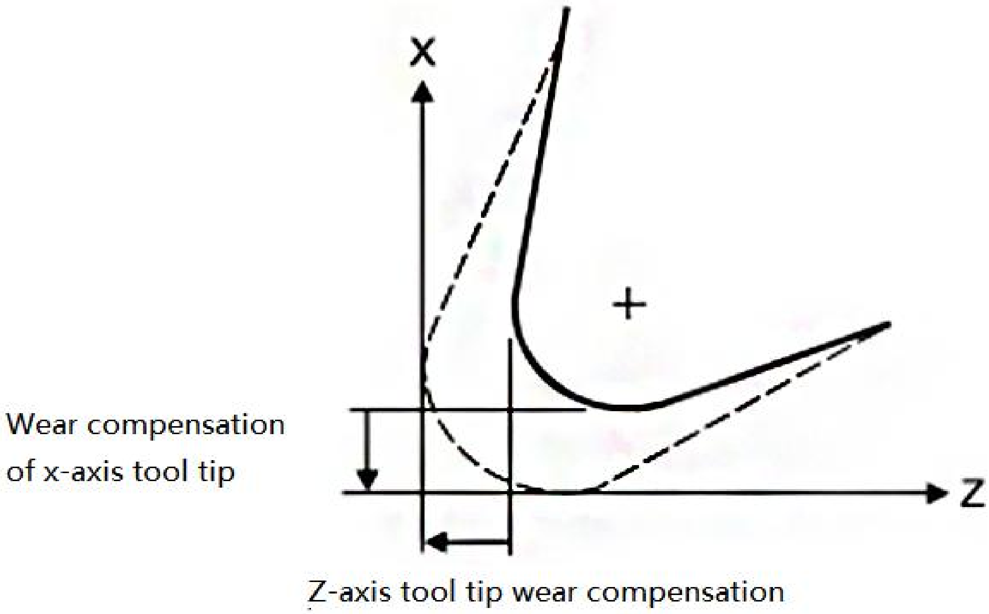


Figure 1.31-7

2. Tool Tip Wear Compensation Cancellation:

1) When the tool tip wear compensation number is set to 0, the compensation is canceled.

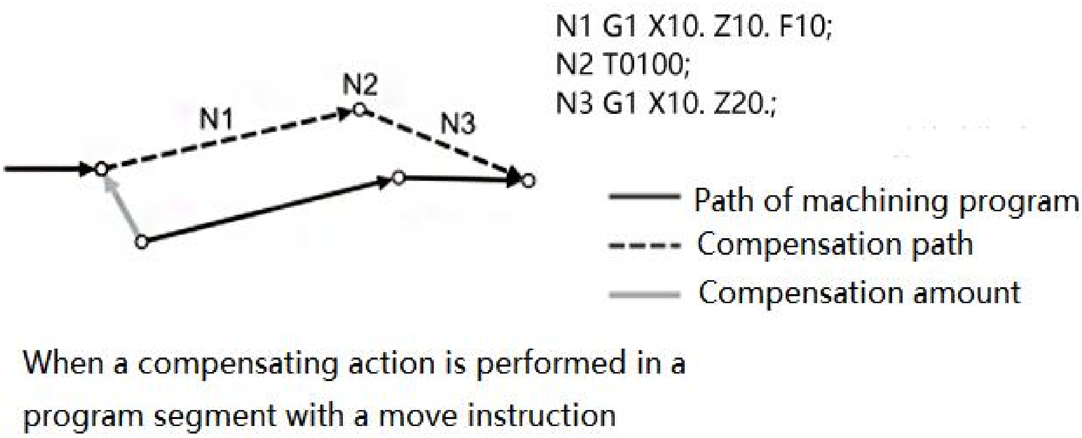


Figure 1.31-8

2) When the compensation amount for tool tip wear specified in the T command is set to 0, the compensation is canceled.
3) Tool tip wear compensation can be canceled by pressing the "Reset" button.

1.31.5 Tool Tip Radius Compensation

Due to the roundness of the tool tip during arc and taper cutting, there can be discrepancies between the shape cut by the program and the actual cut shape. Tool tip radius compensation is a feature that automatically calculates errors and applies compensation by setting the tool tip radius.

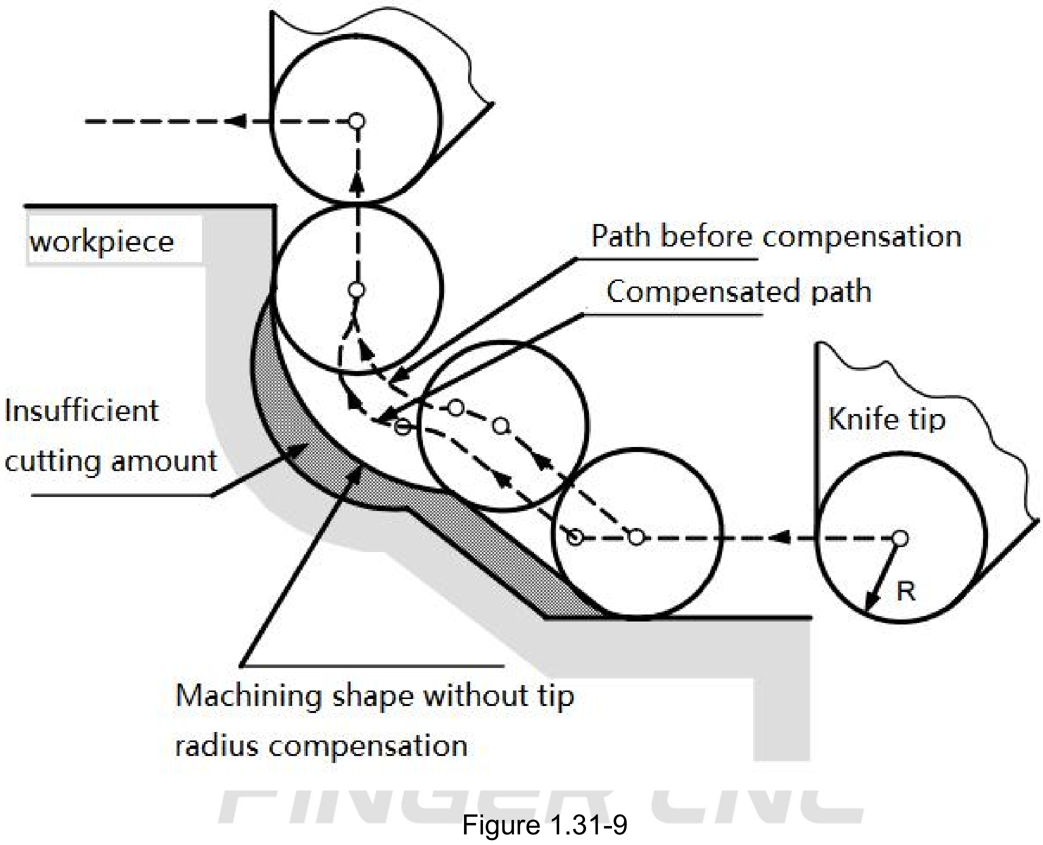


Figure 1.31-9

1.31.5.1 Supplement Explanation

Tool Offset Range:

G40: Moves along the programmed path, canceling tool compensation.

G41: Moves to the left side in the forward direction of the programmed path.

G42: Moves to the right side in the forward direction of the programmed path.

Tool offset is on the opposite side of the workpiece.

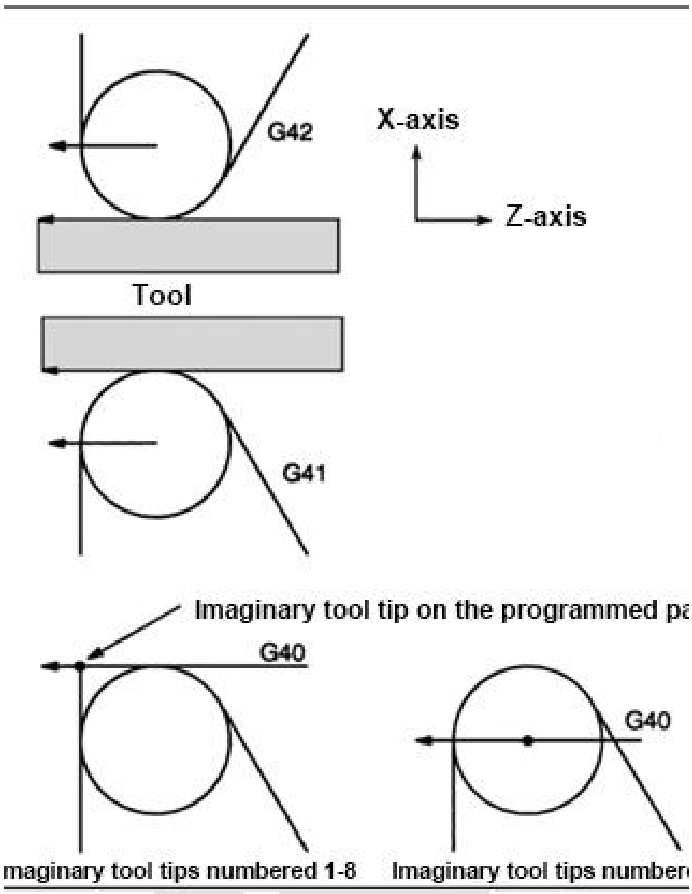


Figure 1.31-10

Setting the coordinate system can change the position of the workpiece.

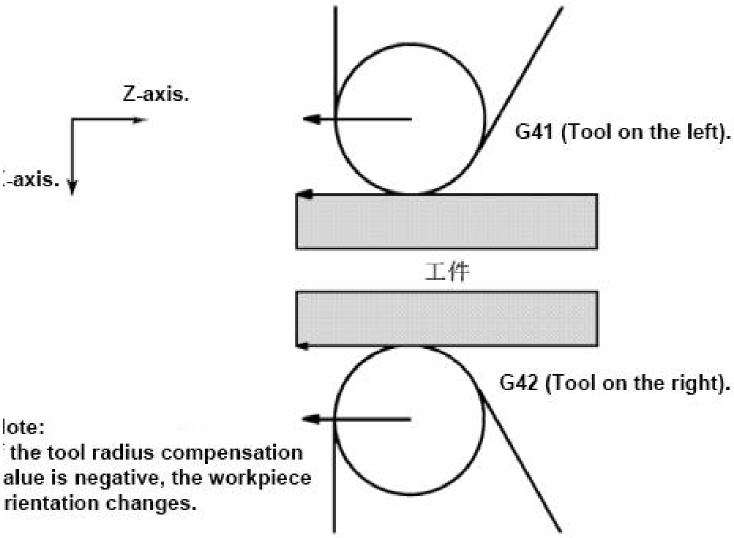


Figure 1.31-11

1.31.5.2 Tool Tip Point and Compensation Operation

1. Tool tip radius center for machining at the starting position:

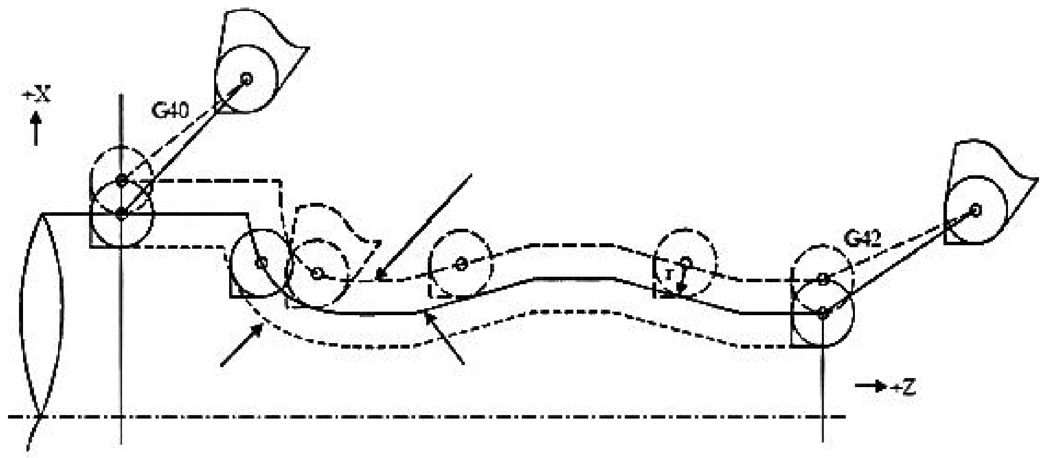


Figure 1.31-12

2. Tool tip point for machining at the starting position:

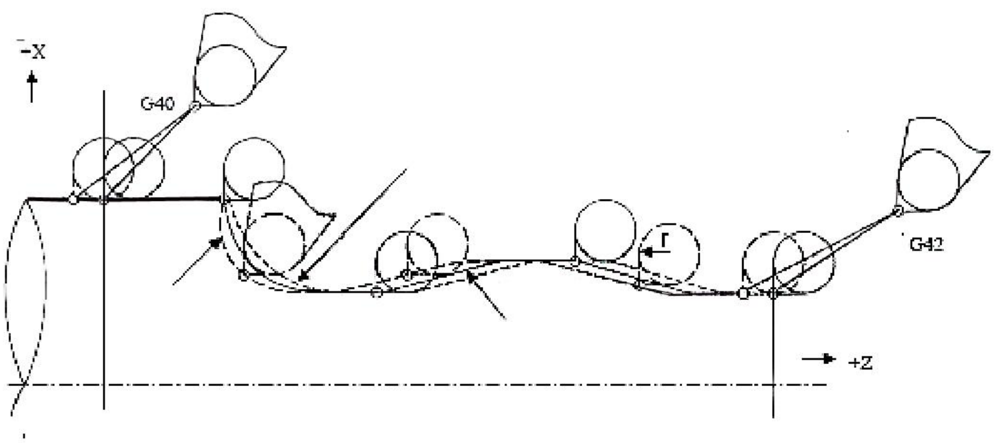


Figure 1.31-13

1.31.5.3 Tool Tip Radius Compensation Pre-processing

Regardless of whether in continuous or single-step execution mode, when compensation starts, it is necessary to continuously read N program segments first in order to perform intersection calculations (pre-read a maximum of N program segments when there are no movement commands, with N specified by MCM1702).

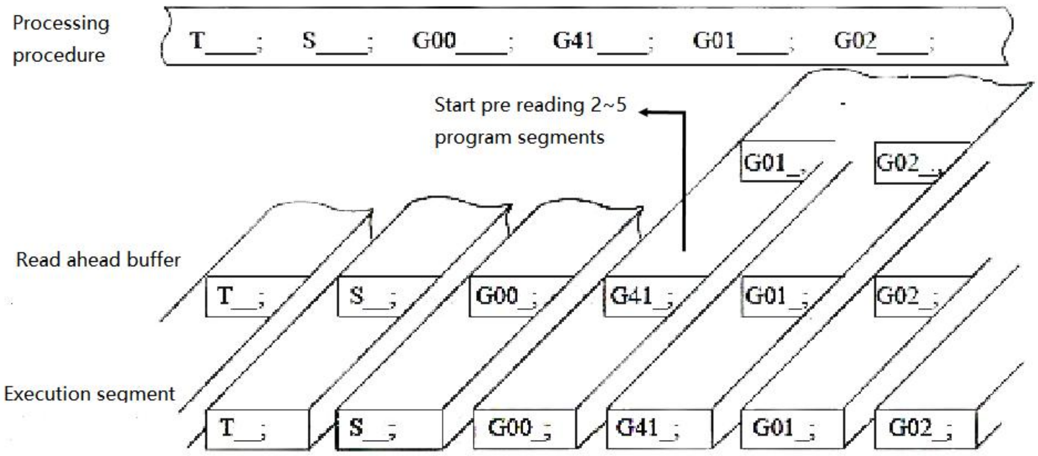


Figure 1.31-14

When encountering program segments without movement, the system performs the following steps:

1. At the start of compensation, continuously specify N program segments without movement, and do not establish tool tip compensation at program segments without movement.

   Assuming N is 4, as shown below:

   N01 U60. W30. T01;

   N02 G41;

   N03 G04 X2.;

   N04 F1000.;

   N05 M03 S500;

   N06 G04 X0.5;

   N07 U-50. W20.;

   N08 U-20. W50.;

   Execution path:

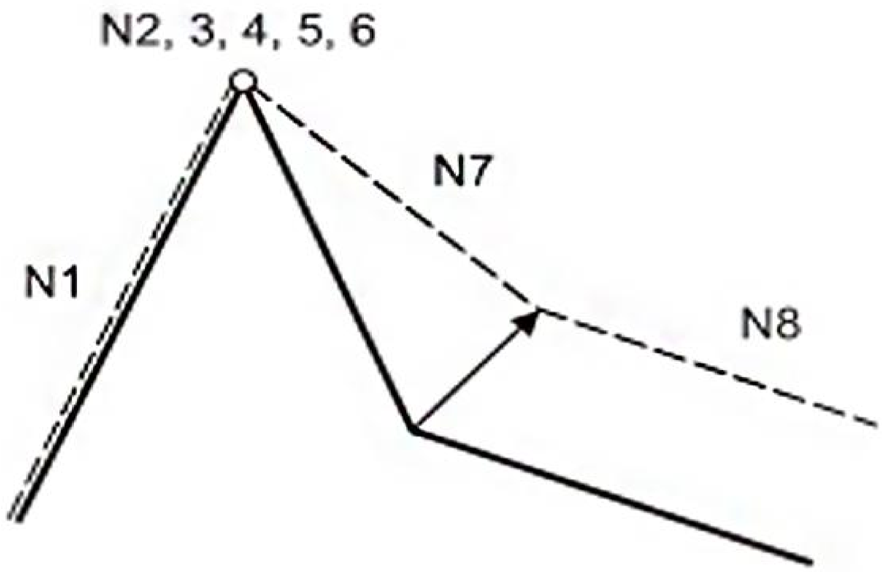


Figure 1.31-15

Tool compensation is only established in segment N7.

2. While staying in compensation mode and consecutively waiting for N program segments with no motion, a perpendicular compensation vector is established at the endpoint of the preceding program segment.
N06 U200. W100.;
N07 G04 X2.;
N08 F1000.;
N09 S500;
N10 M4;
N11 W100.;

Execution path:

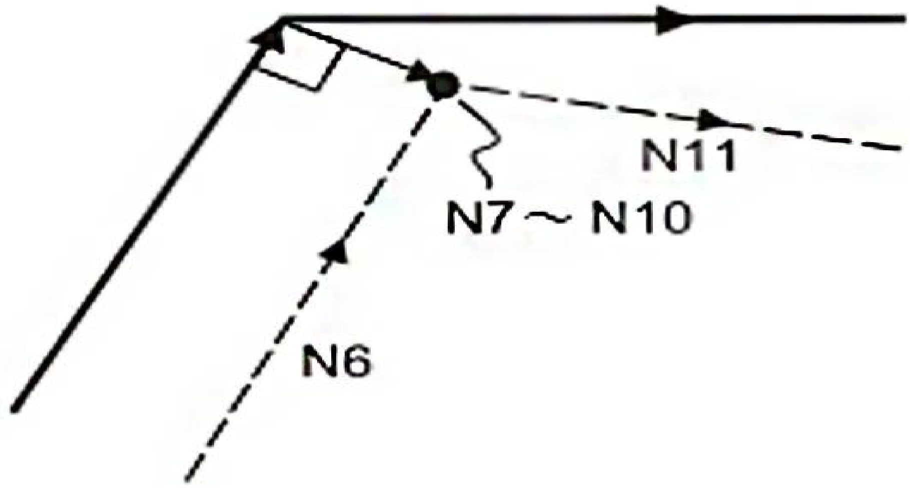


Figure 1.31-16

Create a vertical compensation vector in segment N7.

3. Specify a non-moving program segment when canceling compensation, i.e., cancel compensation in the same block as G40 and a non-moving program.
N06 U200. W100.;
N07 G40 M5;
N08 U50. W100.;

Execution path:

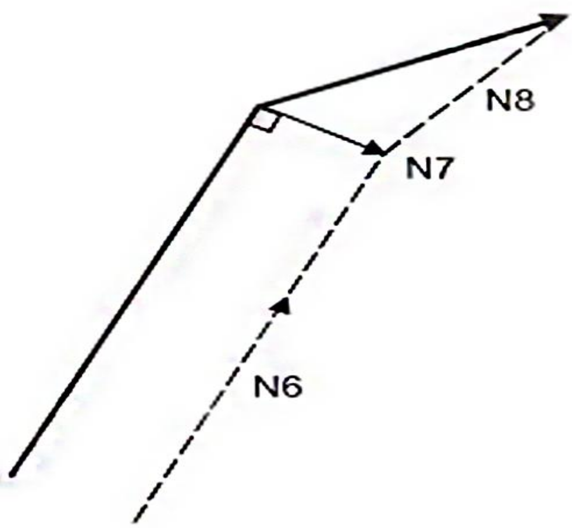


Figure 1.31-17

1.31.5.4 Tool Radius Compensation Types

Tool radius compensation start and end types can be divided into the following four types: Type A, Type B, Type C, Type D, as set by Mcm1701.

1. Tool Start

Programming：

1) G41 and the motion of the current plane axis are in the same block.

2) G41 is in a separate block or the plane axis increment is 0.

Type A: Output a vertical compensation vector in the next program block after the tool start.

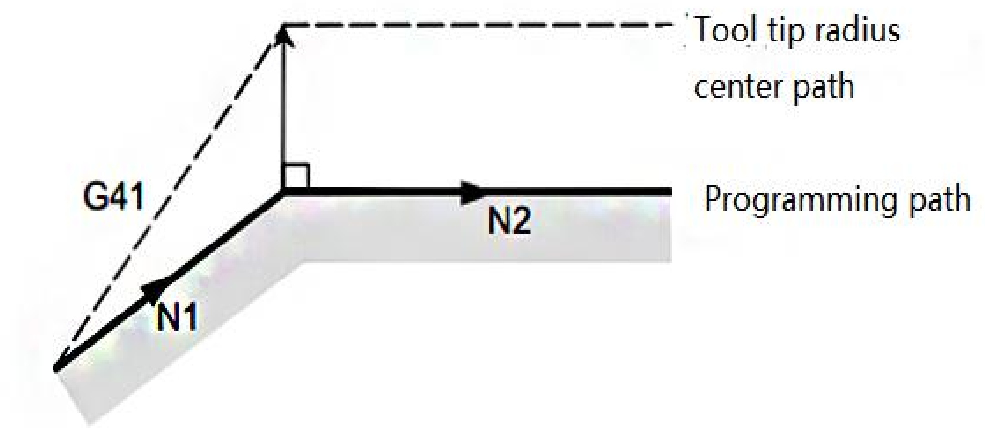


Figure 1.31-18

Programming1)、2）Run the toolpath.

Type B: Output the compensation vector perpendicular to the starting toolpath segment and the intersection vector.

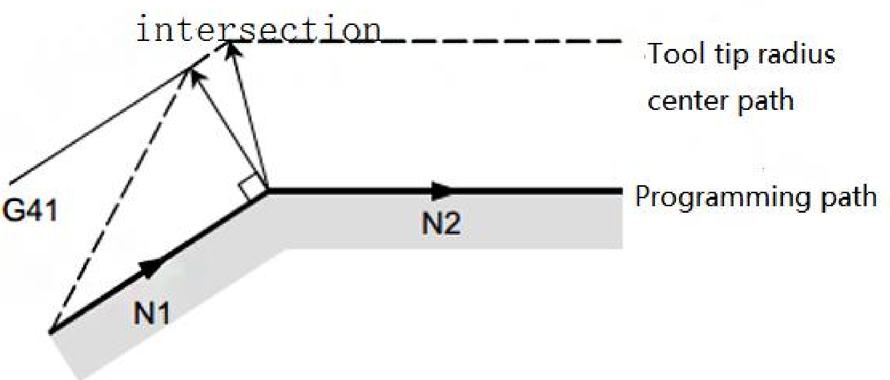


Figure 1.31-19

Programming1)、2）Run the toolpath.

Type C: The tool moves an amount along the direction perpendicular to the starting toolpath segment, equivalent to the tool radius compensation amount.

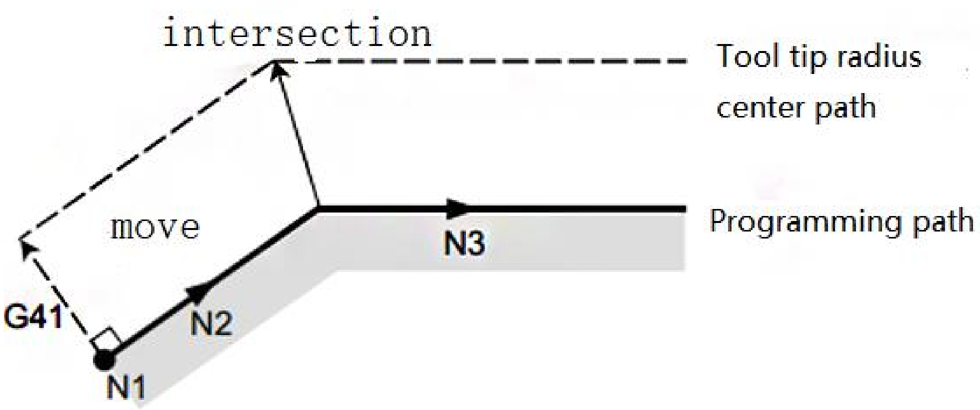


Figure 1.31-20

Programming1)、2）Run the toolpath.

Type D: The system pre-reads the preceding block before G41/G42, establishes tool compensation at the end of the preceding block, and does not involve a vertical movement.

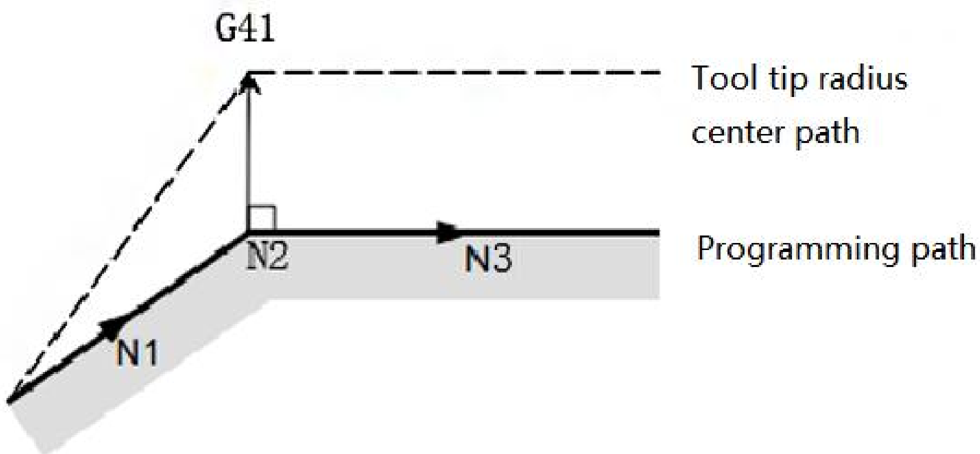


Figure 1.31-21

Note: Block N2 is a G41 block.

2. End

Programming：

3) G40 is in the same block as the motion of the current plane axis.

4) G40 is in a block by itself or has a plane axis increment of 0.

Type A: Cancel the vertical compensation vector in the Program block just before the cancellation.

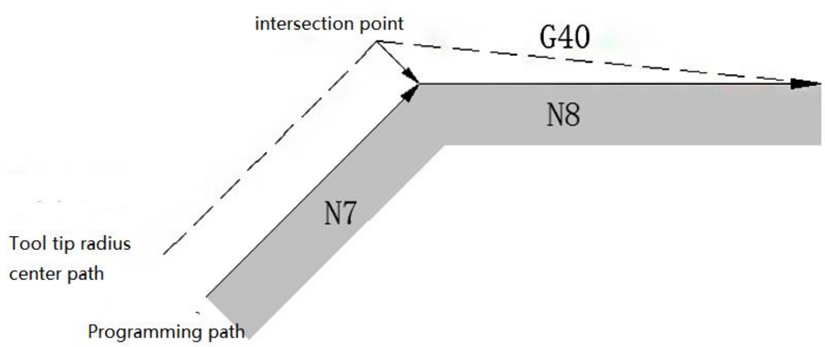


Figure 1.31-22

Programming3)、4）Run the toolpath.

Type B: Outputs a compensating vector perpendicular to the cancelled program block, as well as an intersection vector.

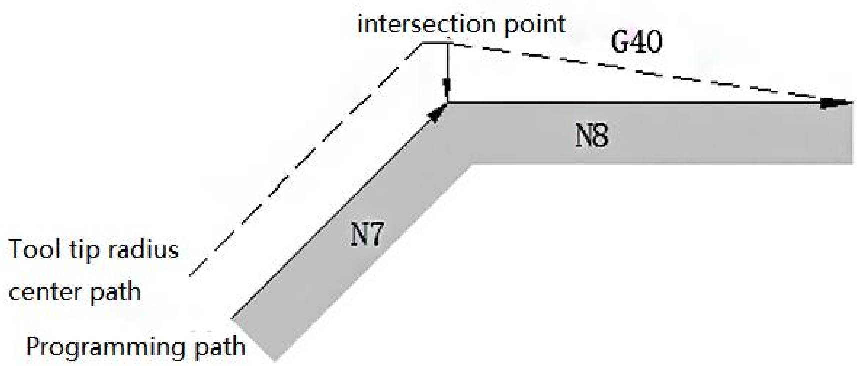


Figure 1.31-23

Programming3)Run the toolpath.

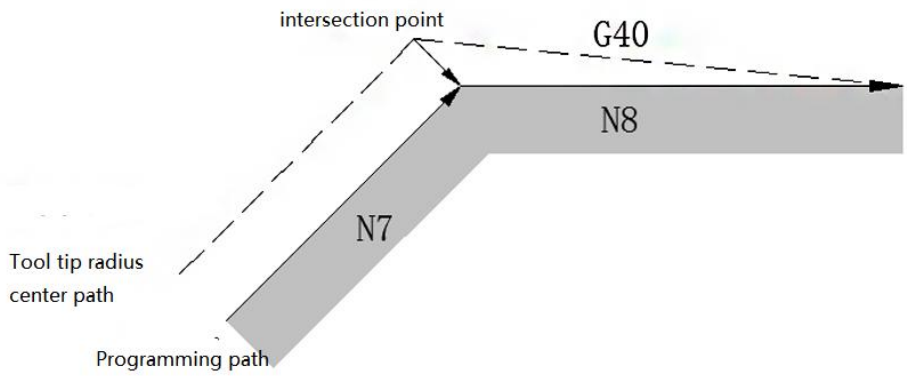


Figure 1.31-24

Programming4）Run the toolpath.

Type C: The tool moves an amount in a direction perpendicular to the cancelled program block, equivalent to the tool tip radius compensation.

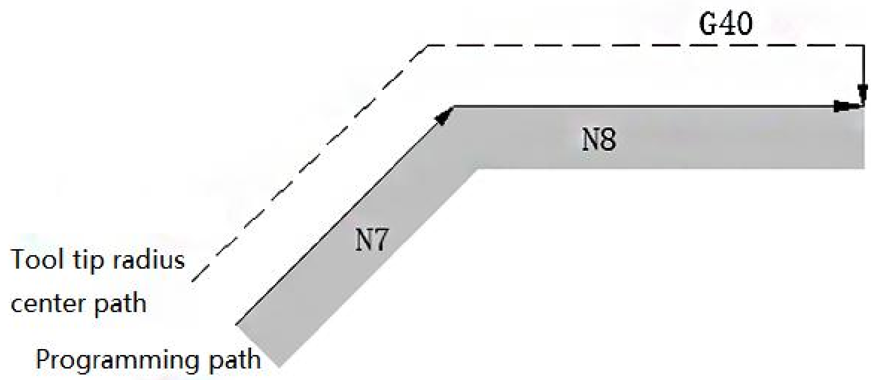


Figure 1.31-25

Programming3)Run the toolpath.

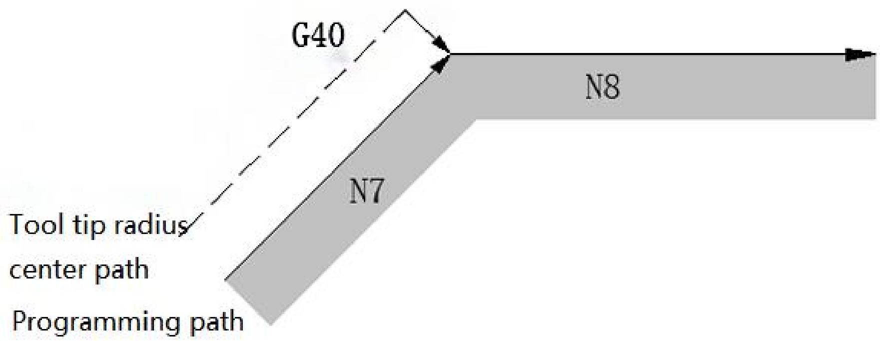


Figure 1.31-26

Programming4)Run the toolpath.

Type D: The system pre-reads the previous program block before G40 and cancels the tool compensation at the end of the program block before G40, without any vertical movement.

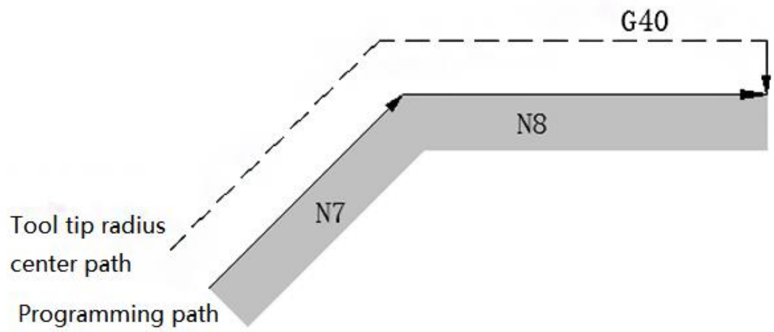


Figure 1.31-27

Programming3)Run the toolpath.

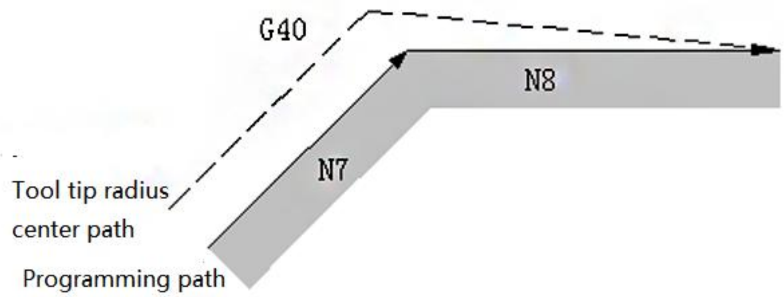
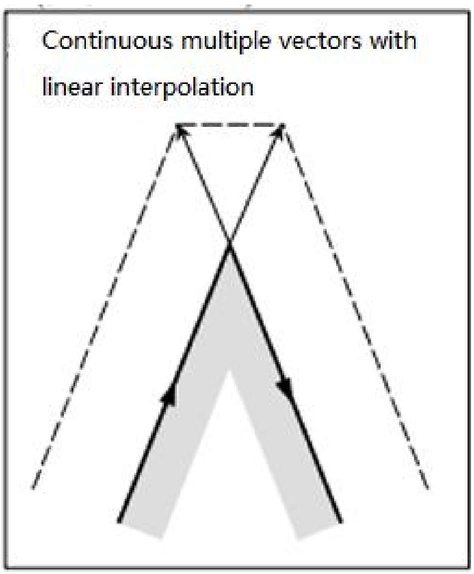


Figure 1.31-28

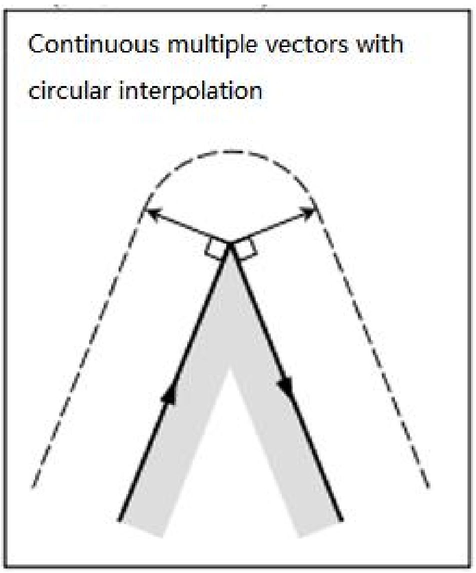
Programming4）Run the toolpath.

1.31.5.5 Tool tip radius compensation linkage

When rotating outer corner in tool tip radius compensation mode, you can use MCM1700 to choose between linear or arc interpolation to connect multiple compensation vectors.



Linear Connection (Mcm1700=0)



Arc Connection (Mcm1700=1)

Figure 1.31-29

1.31.5.6 Corner Handling

When two or more offset vectors are generated at the end of a program block, the tool moves linearly from one vector to another. This type of movement is referred to as a corner rounding move.

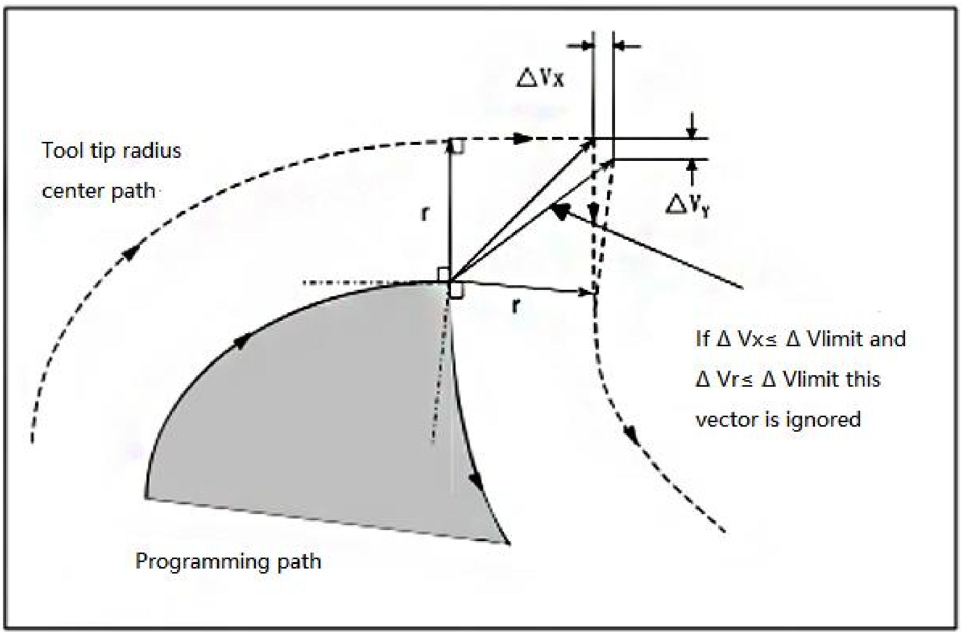


Figure 1.31-30

If △Vx ≤ △Vlimit and △Vr ≤ △Vlimit, the subsequent vectors are ignored, where △ is set by Mcm1703.

The cornering speed is determined by the next block of instructions following the corner, as shown in the figure.

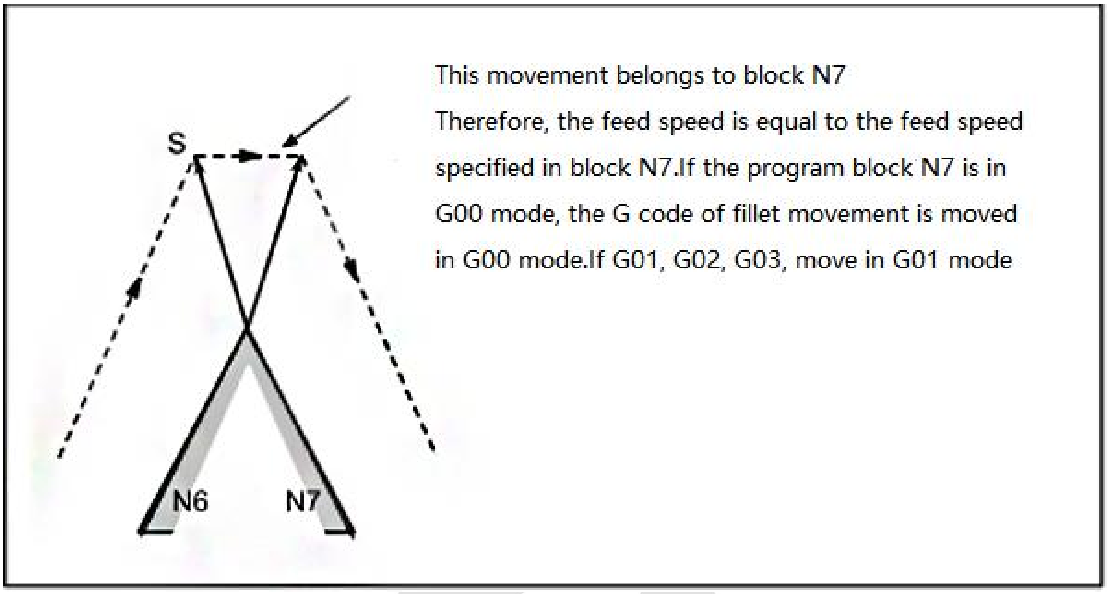


Figure 1.31-31

1.31.5.7 Tool tip radius compensation motion

The tool movement at the beginning of tool entry:

The program segment that transitions from G40 mode to G41 or G42 mode is referred to as the tool entry program segment.

When compensation is canceled and all the following conditions are met, tool tip radius compensation begins:

1) The execution of a G41 or G42 command.
2) The execution of a movement command other than arc commands. If used in G02 or G03 arc cutting modes, the system will generate an error signal (40/41/42-1).
3) When G41 or G42 and a movement with zero displacement are in the same block, no tool compensation starting action is taken (G41/G42 U0. W0.).
   - ❖ Inner: The included angle between the intersection of two movement segments is greater than or equal to 180°.
   - ❖ Outer: The included angle between the intersection of two movement segments is between 0° and 180°.

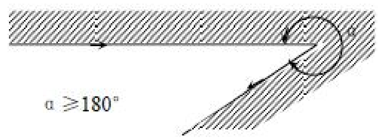
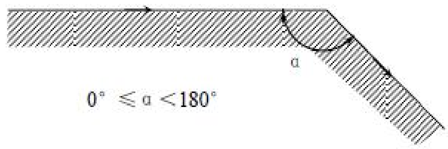
	

Figure 1.31-32

Tool moves along the inside edge (180° ≤ α).

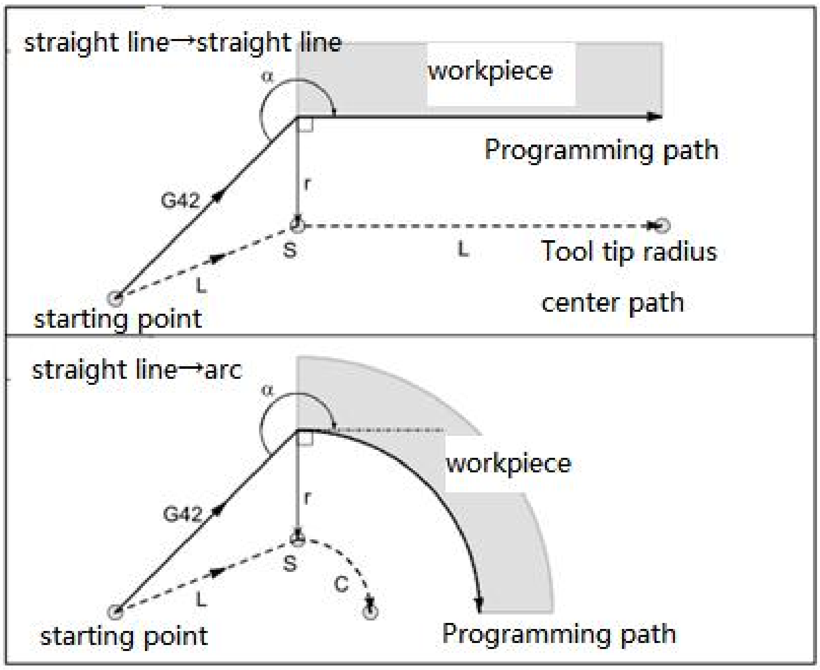


Figure 1.31-33

Tool moves along the outside edge (90° ≤ α < 180°).

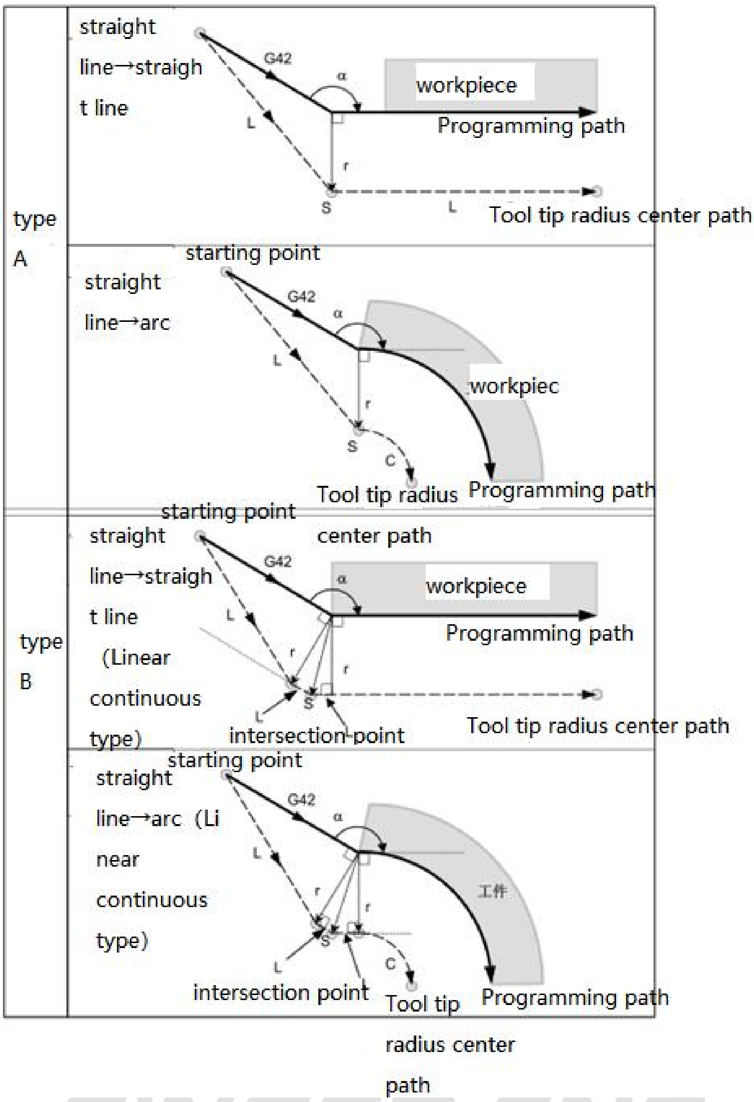


Figure 1.31-34

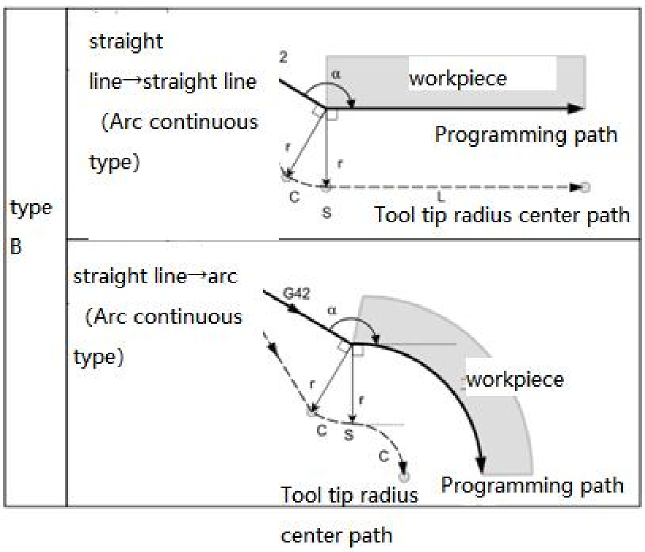
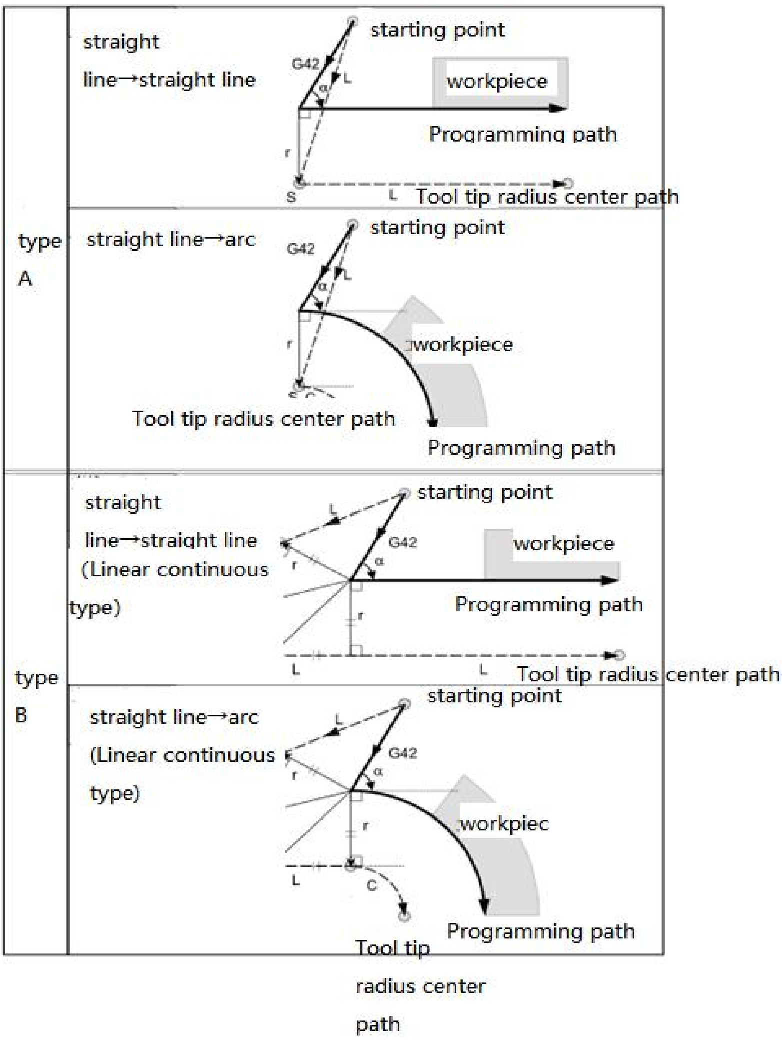


Figure 1.31-35

Tool moves around the acute angle (α < 90°).



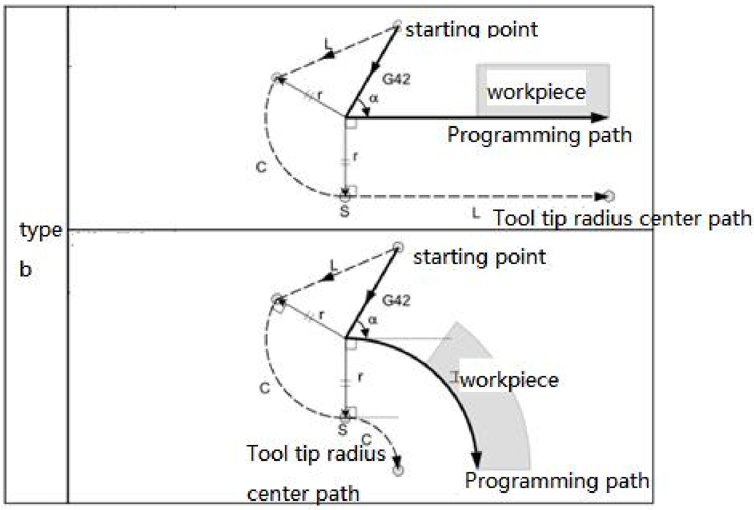


Figure 1.31-36

Tool movement during compensation:

Tool moves around the inside corner (α ≥ 90°).

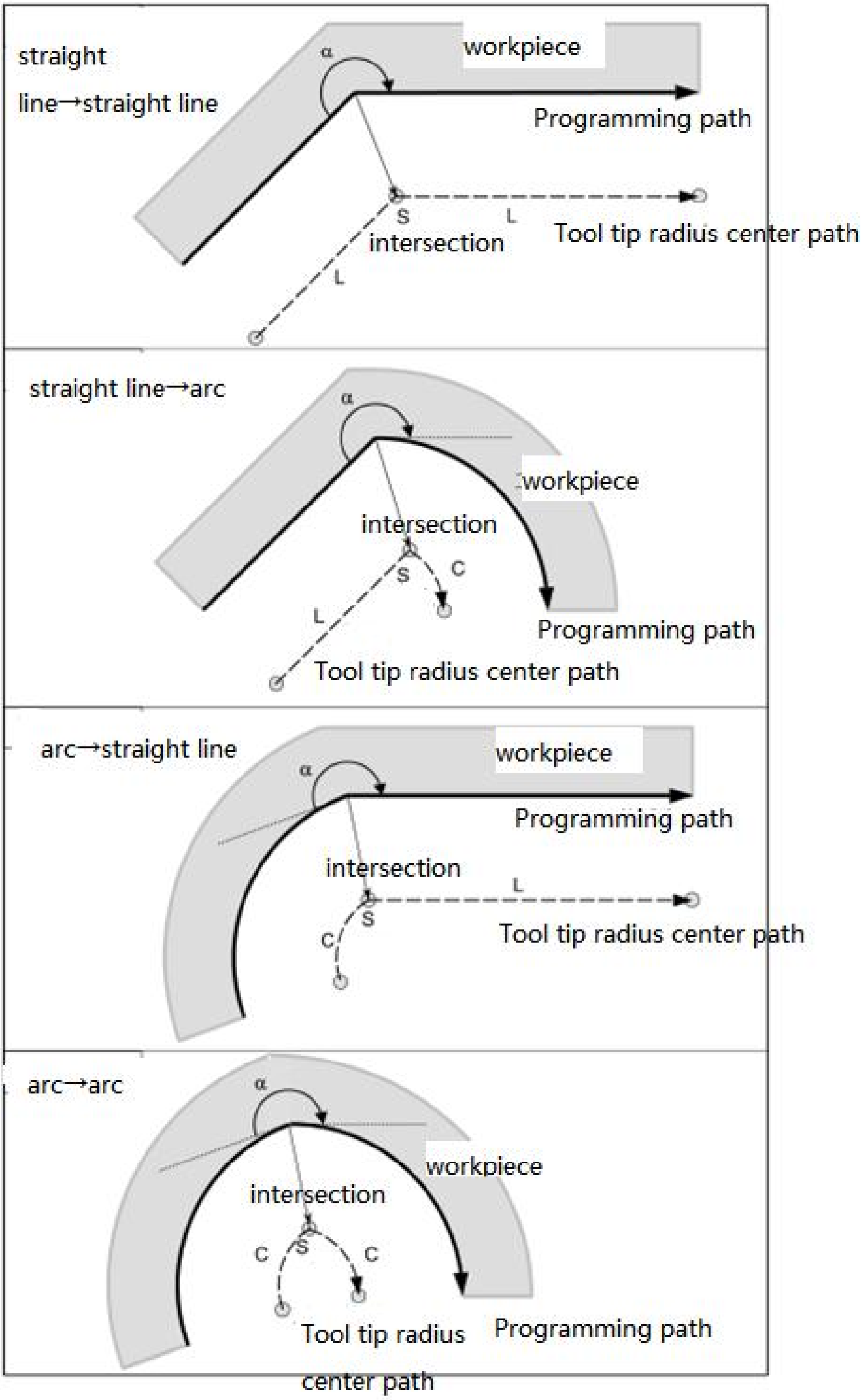


Figure 1.31-37

The tool moves around the outside corner (90° ≤ α < 180°).

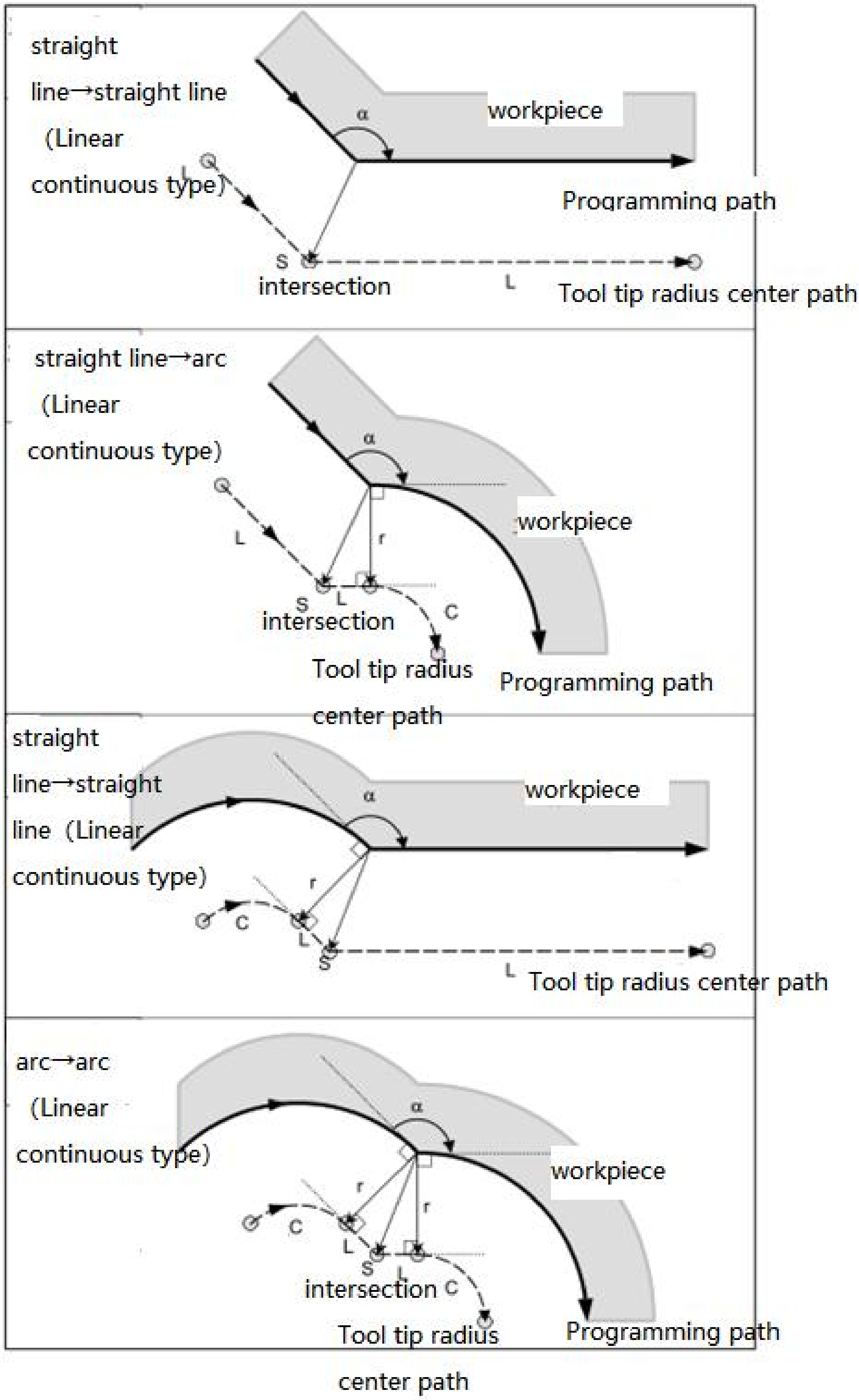


Figure 1.31-38

The tool moves around the acute corner (α < 90°).

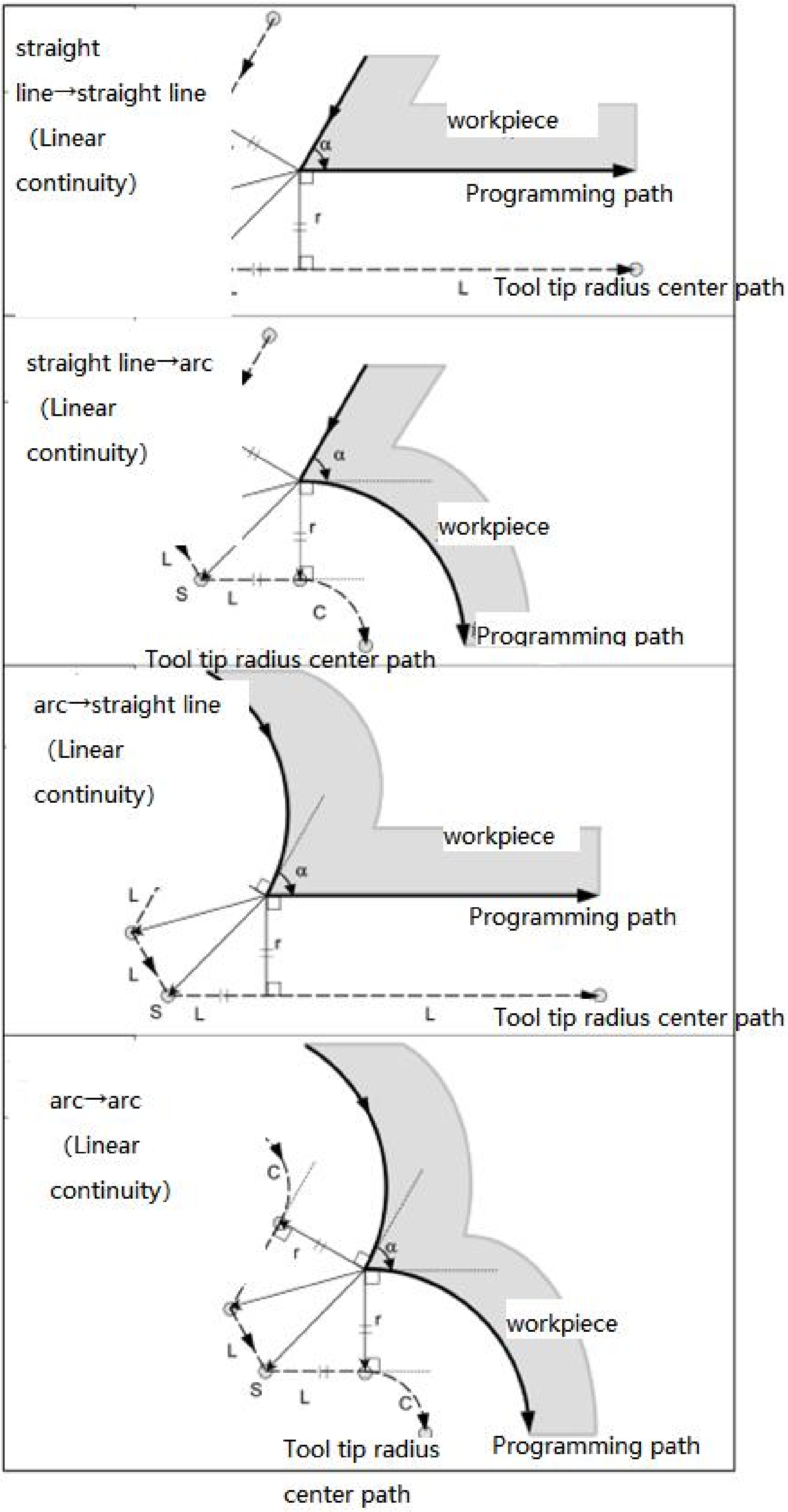


Figure 1.31-39

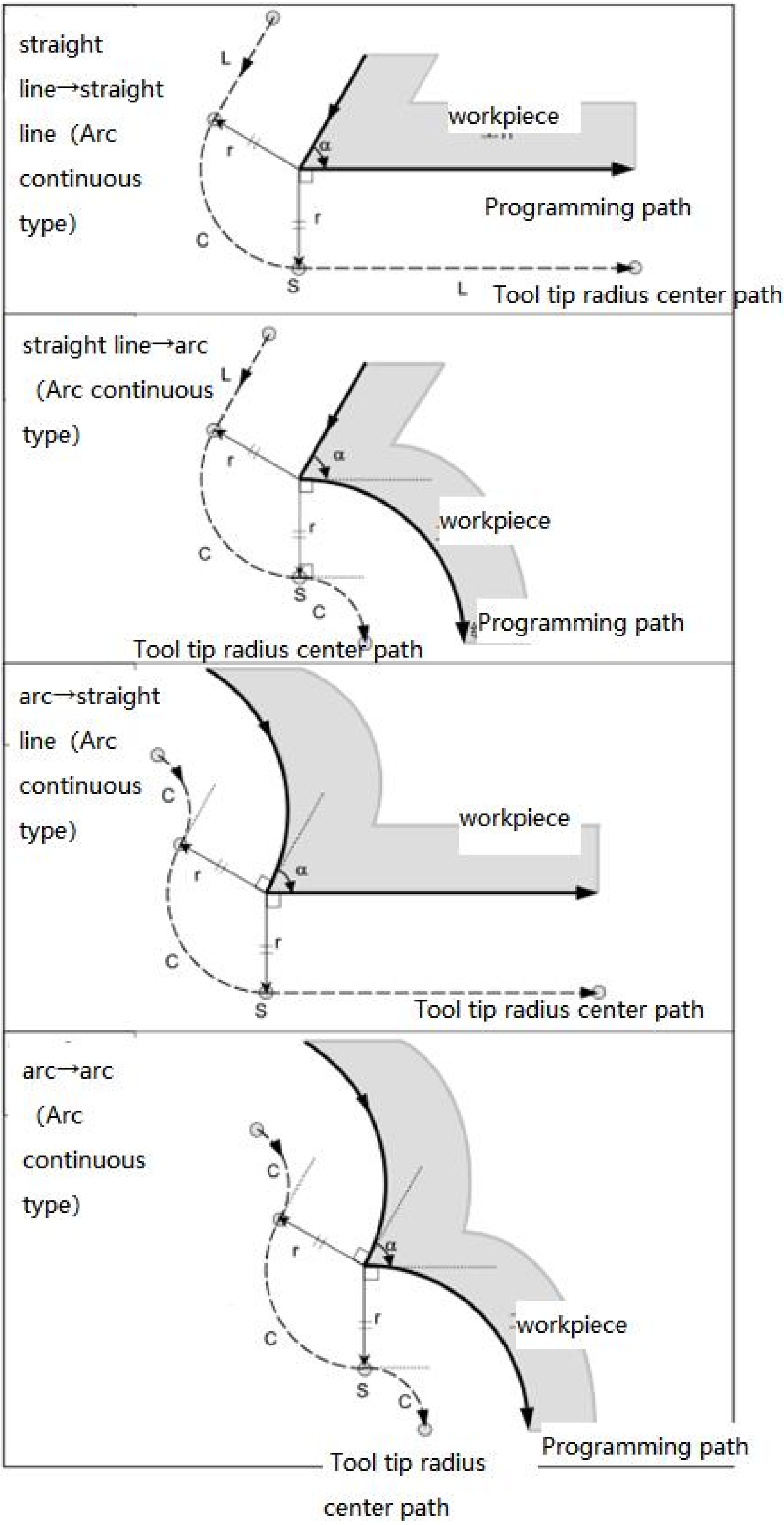


Figure 1.31-40

Tool Motion during Compensation Cancelation

In tool tip radius compensation mode, tool tip radius compensation can be canceled under the following conditions:

1. G40 command has been executed.
2. Any motion command other than arc commands has been executed.

Tool radius compensation cancelation takes place under the following conditions:

1. Cancellation upon power-up.
2. Cancellation upon reset.
3. Cancellation when executing M02 or M30 to end the program.
4. Cancellation when executing a G40 command.

Tool motion around inner edge (α≥90°).

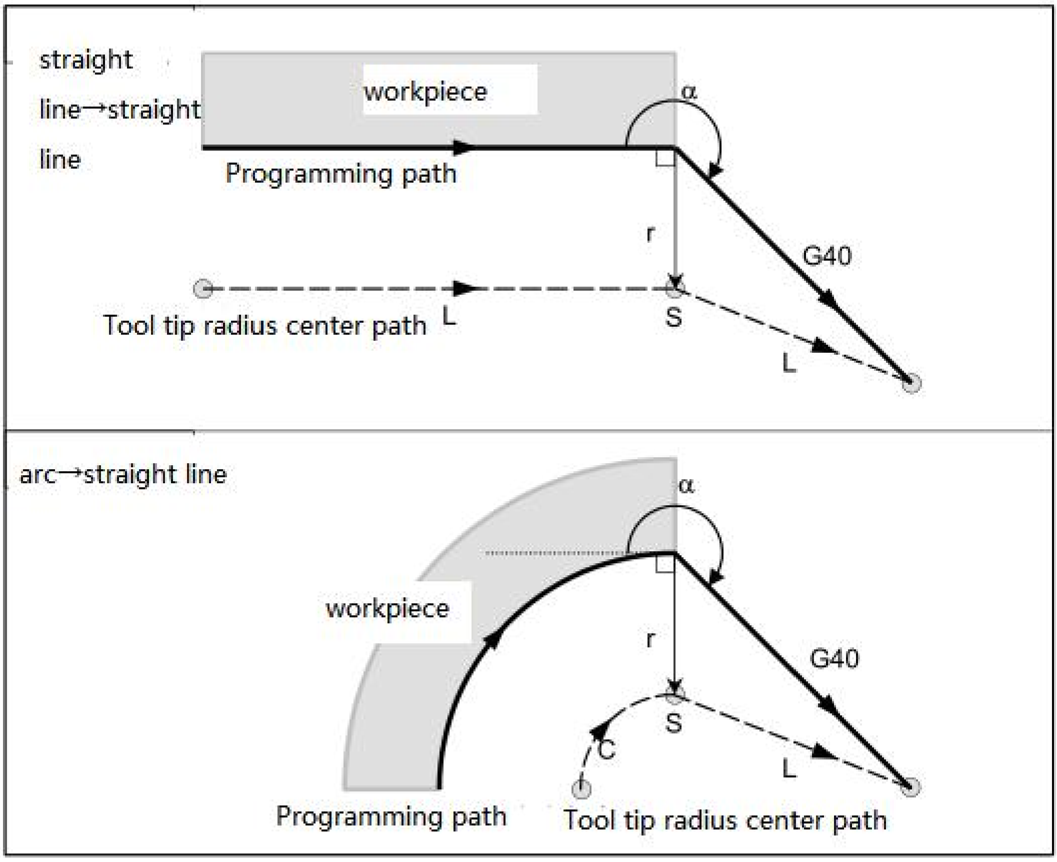
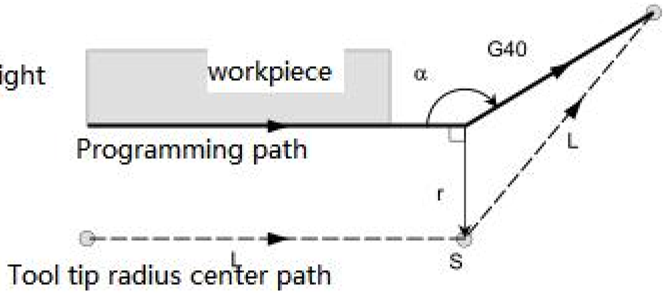
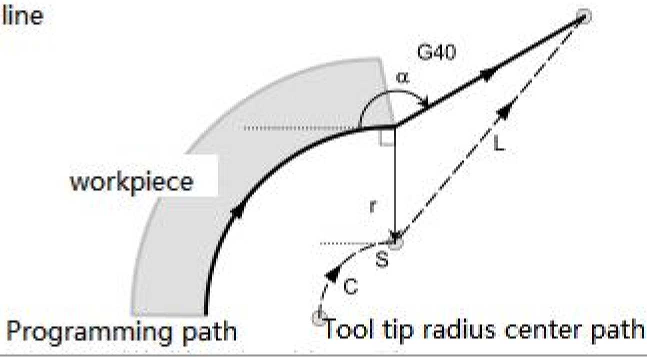
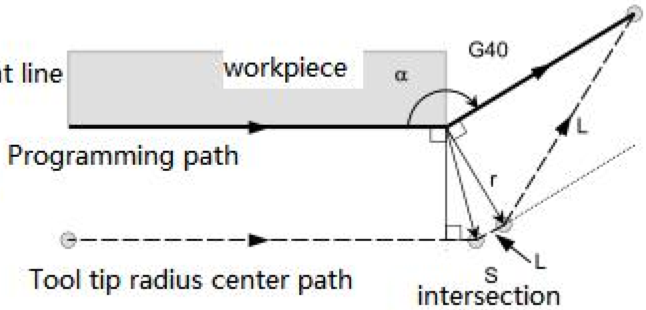
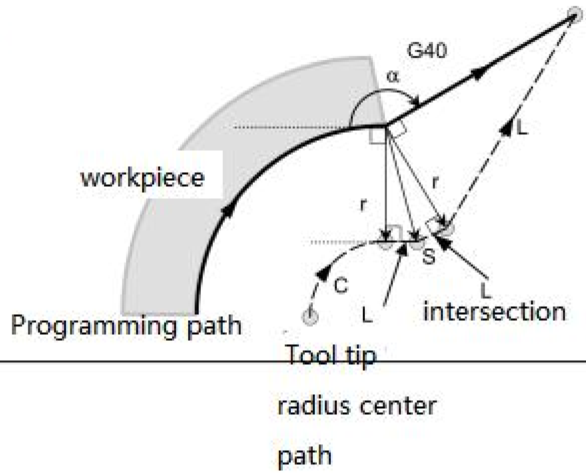


Figure 1.31-41

Tool Motion around Outer Edge (90°≤α<180°)

For the following type B compensation cancelations, G40 and the motion occur in the same block.

<table>
<tr>
<td rowspan="2">type A</td>
<td>straight line→straight line</td>
<td>
</td>
</tr>
<tr>
<td>arc→straight line</td>
<td>
</td>
</tr>
<tr>
<td rowspan="2">type B</td>
<td>straight line→straight line（Linear continuity)</td>
<td>
</td>
</tr>
<tr>
<td>arc→straight line（Linear continuity)</td>
<td>
</td>
</tr>
</table>

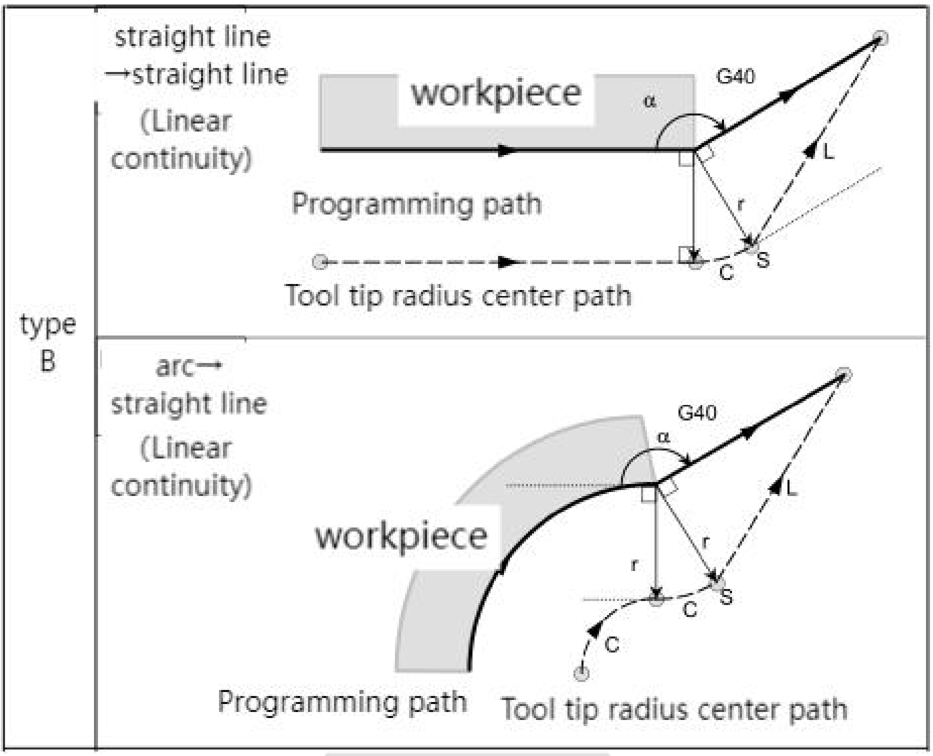
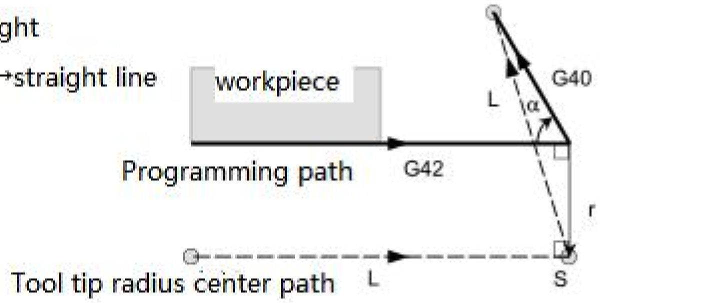
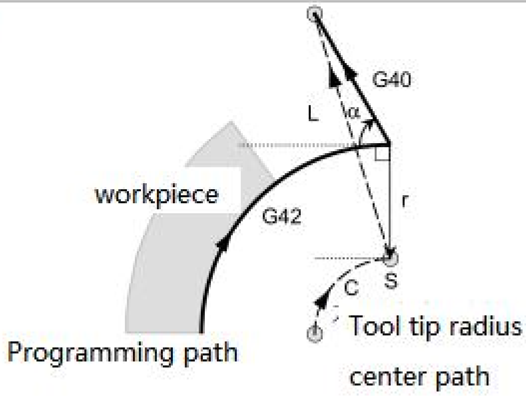
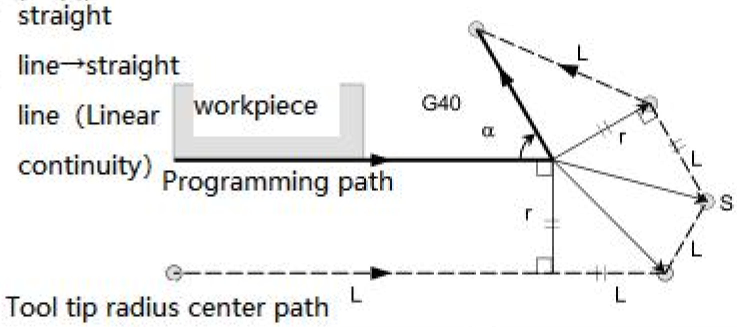
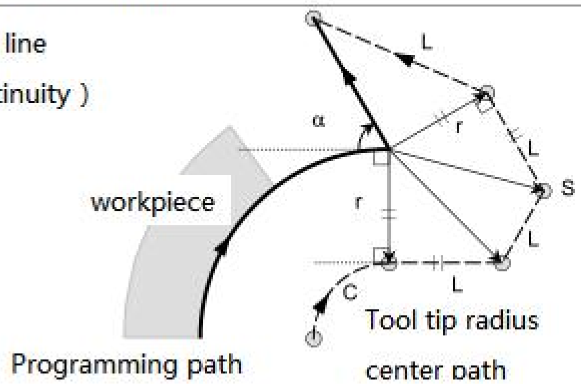


Figure 1.31-42

Tool Movement around Acute Angle (α＜90°)

type A	straight line→straight line	
	arc→straight line	
type B	straight line→straight line（Linear continuity）	
	arc→straight line（Linear continuity）	

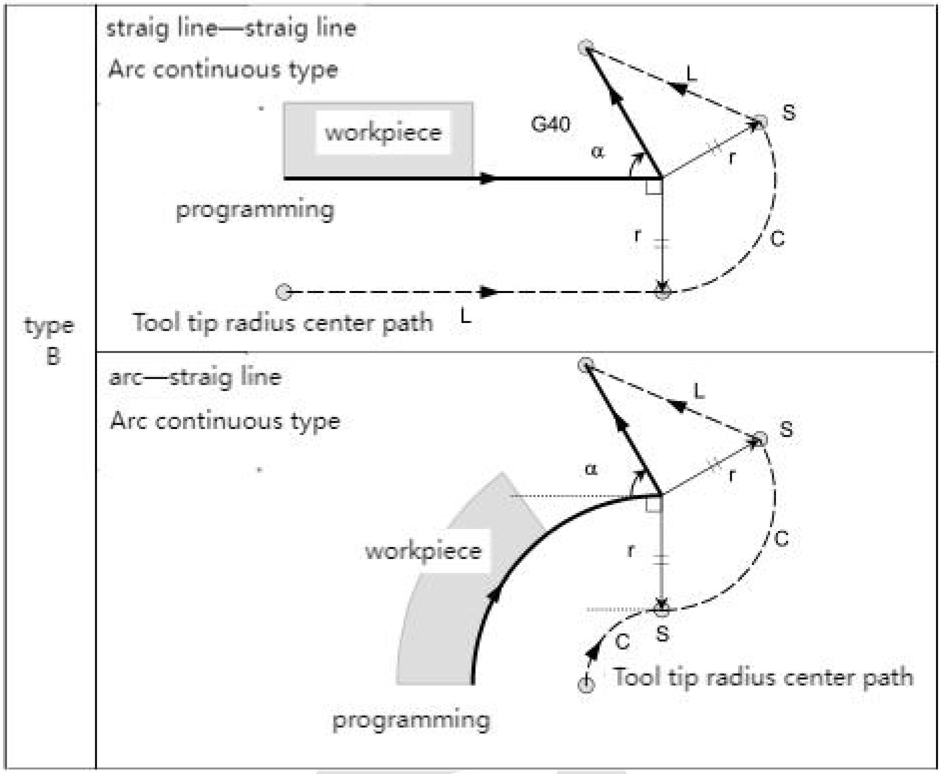


Figure 1.31-43

1.31.5.8 G40 with I_J_K_ specified

If G40 and I_J_K_ are specified in the program segment, the system's tool radius compensation trajectory is as follows.

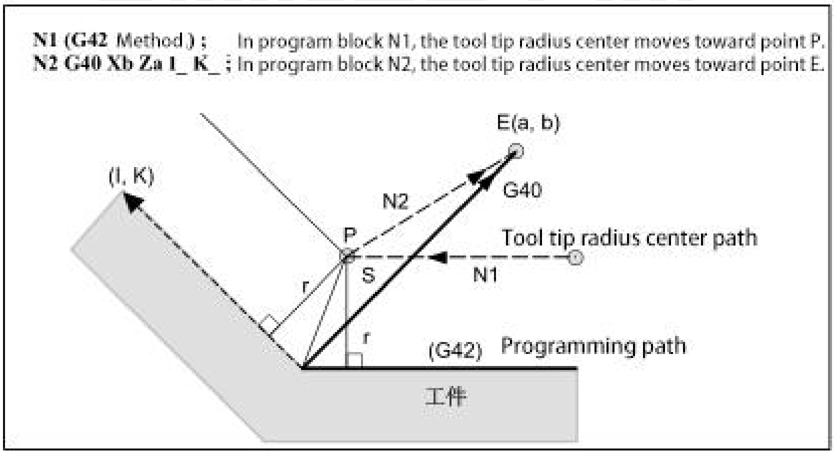


Figure 1.31-44

Note: Point P is the intersection point between the G42 segment and the I and K directions.

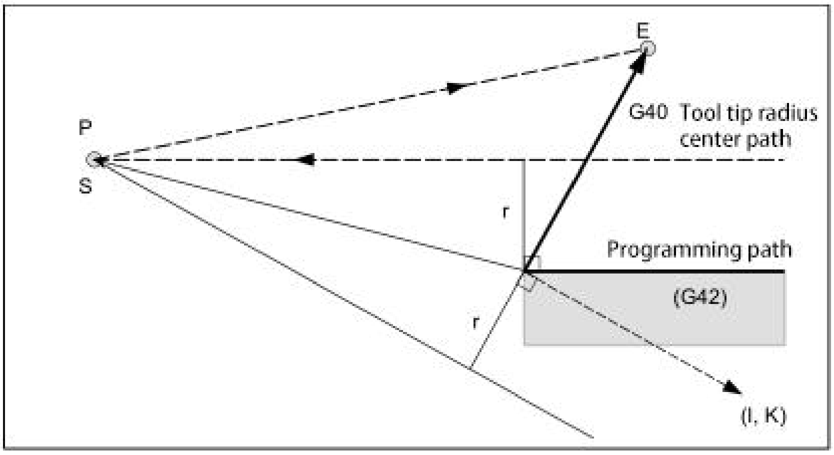


Figure 1.31-45

Note: The point P is the intersection of the G42 block and the I, K directions.

When the intersection cannot be obtained, the tool moves to the endpoint of the program block before G40 and intersects it orthogonally, as shown in the figure below.

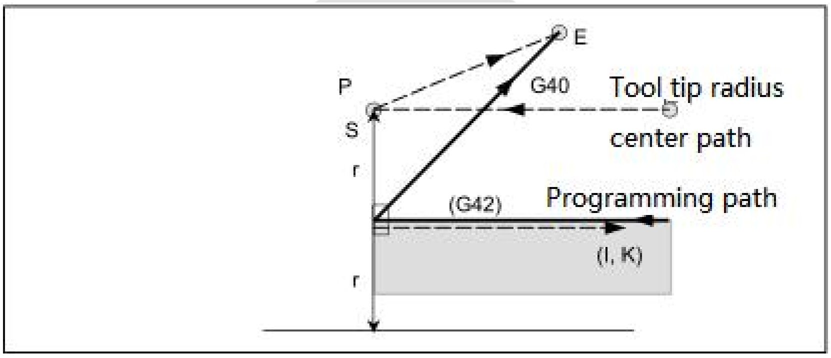


Figure 1.31-46

1.31.5.9 Interference Handling

- Tool entering the workpiece is called interference, and the interference check function detects tool interference in advance.
- Interference Check Conditions:

  To perform interference checks, you must read in three or more Program blocks that involve movement. Therefore, in tool compensation mode, if you issue continuous auxiliary function commands, pauses, or other Program blocks without movement, and you cannot read in three or more Program blocks with movement, it becomes difficult to perform interference checks. In some cases, this can lead to overcutting or insufficient cutting. The number of Program blocks to read in the tool compensation mode is determined by Mcm1702. Mcm1702 must be set to a value greater than or equal to 3 to enable interference processing. Otherwise, interference processing will not occur, even if overcutting or similar issues are present.

- The enable and disable states of interference checking are determined by Mcm1704, which dictates the interference checking method.

Parameter Setting:

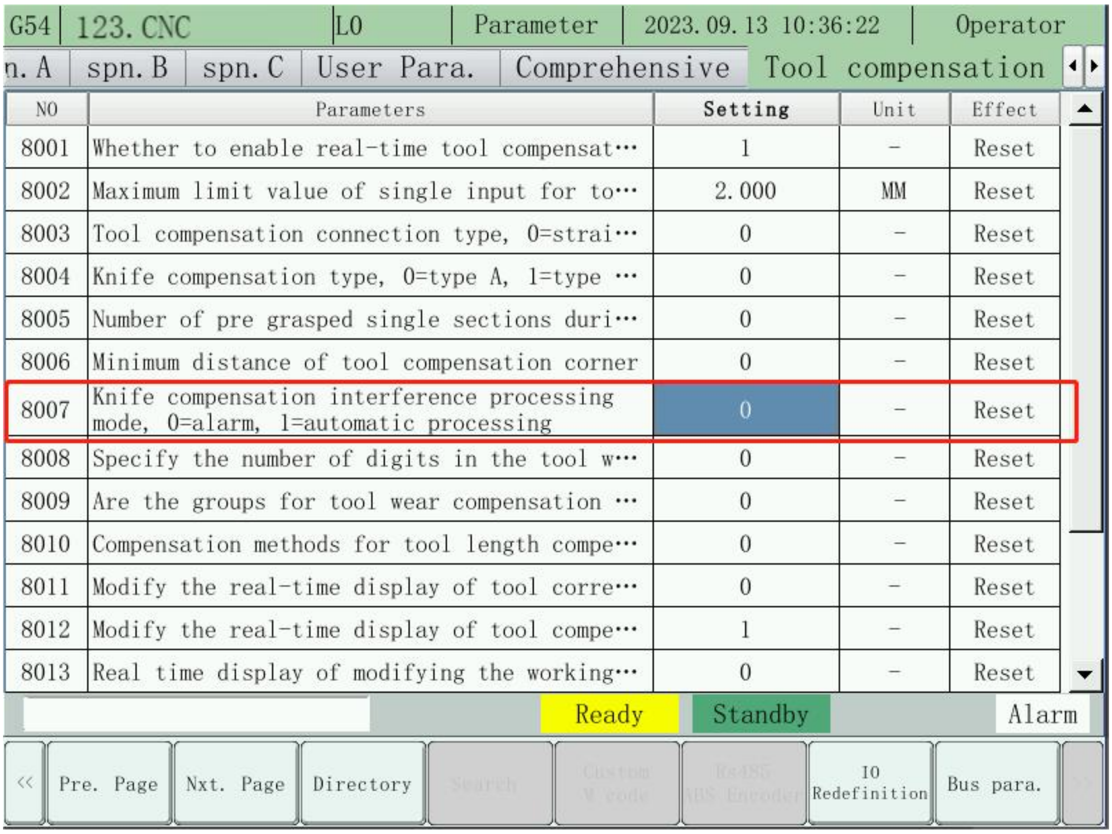


Figure 1.31-47

Mcm1704	Function Definitions
0	The system issues an alarm when there is interference in the tool compensation trajectory.
1	If Mcm1702 is less than or equal to 2, interference optimization processing is not performed.
	If Mcm1702 is greater than 2, interference optimization processing is performed.

1.31.6 Tool Tip Point

The imaginary tool tip is a point that does not actually exist but is used because it can be difficult to align the actual tool tip radius center at the starting point or reference position.

Therefore, an imaginary tool tip is used.

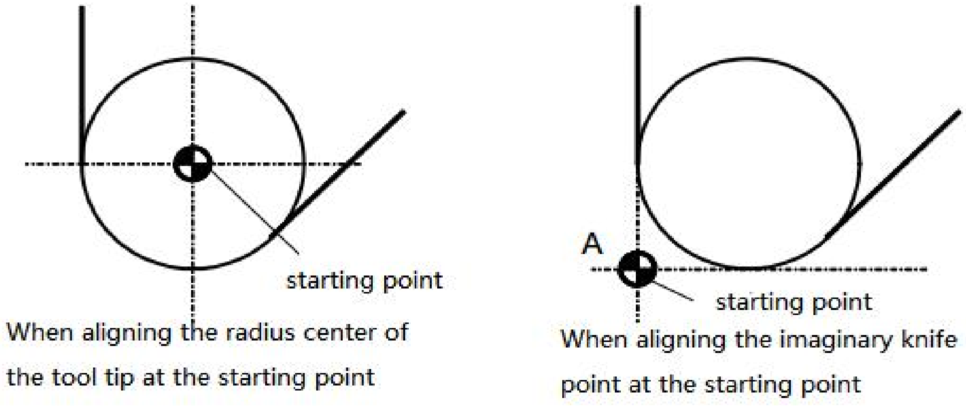


Figure 1.31-48

1.31.7 Imaginary Tool Tip Direction

The imaginary tool tip direction, as seen from the tool tip radius center, is determined by the orientation of the tool during the cutting process. Therefore, it must be specified in advance, just like the compensation amount. You can choose from the 8 different ways shown in the figure below and select the corresponding TCode along with it.

Direction of the Imaginary Tool Tip:

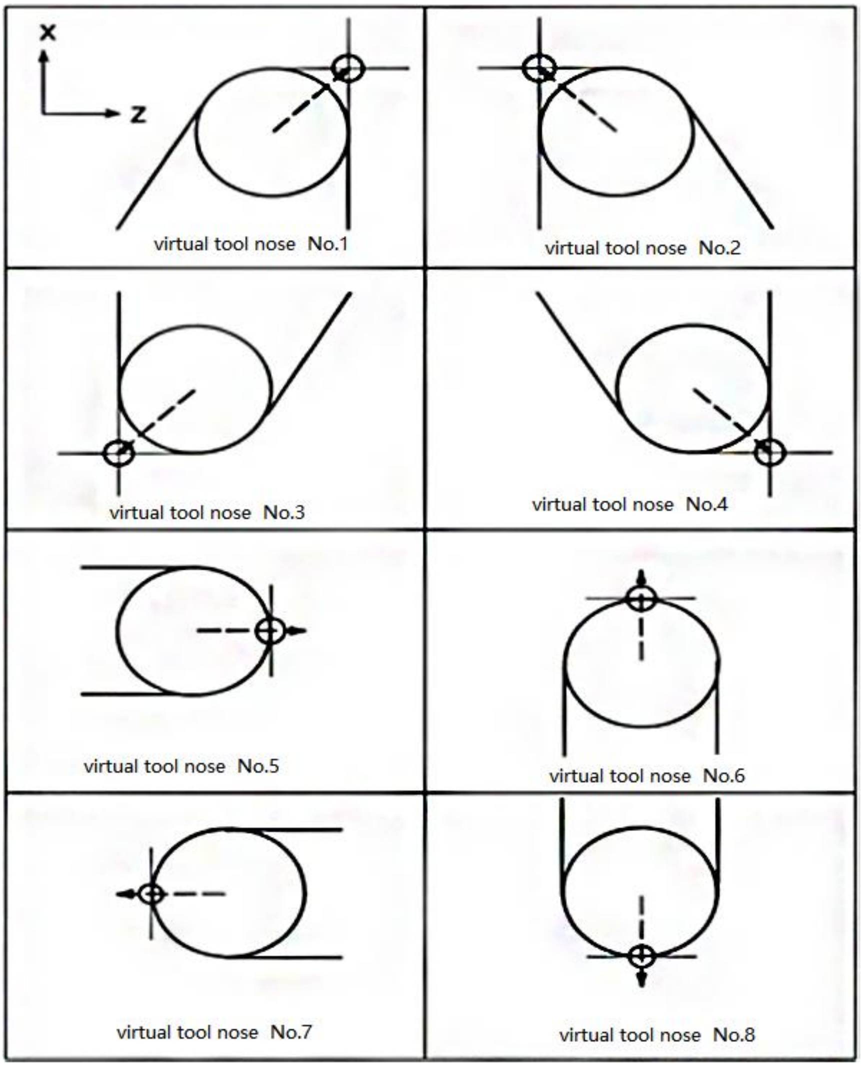


Figure 1.31-49

When the tool tip center coincides with the starting point position, use imaginary tool tip numbers 0 and 9:

Figure 1.31-50

The setting of the tool tip direction is determined by parameters such as Sys2202.

1.31.8 Tool Tip Radius Compensation Value

The tool tip radius value (OFR) is comprised of two components: the tool tip length compensation value (OFGR) and the tool tip wear compensation value (OFWR).

OFR = OFGR + OFWR

When OFR is a positive value, G41 is used for left compensation, and G42 is used for right compensation.

When OFR is a negative value, G41 is used for right compensation, and G42 is used for left compensation.

The tool tip length compensation value (OFGR) is configured using Sys2201.

1.31.9 Example 1

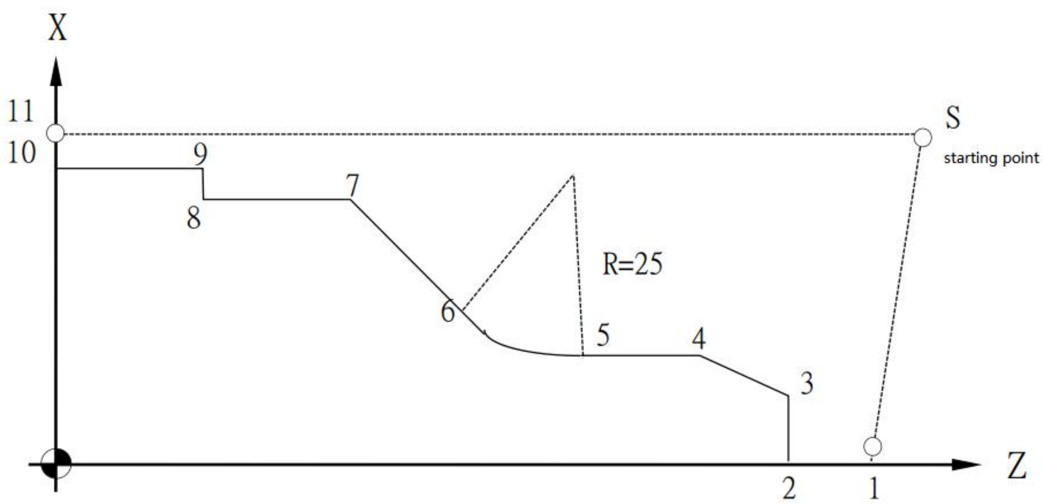


Figure 1.31-51

Diameter Programming

N10 T1;	// Select tool number 1.
N20 G00 X100.0 Z120.0;	// Point S.
N30 G00 X0 Z110.0;	// Point 1.
N40 M3 S2000;	// Start spindle at 2000 RPM.
N50 G42 Z100.0 T02 F3.0;	// Point 2, tool radius compensation active.

```
N60 G01 X20.0;                          // Point 3.
N70 X30.0 Z91.35;                       // Point 4.
N80 Z75.0;                              // Point 5.
N90 G02 X44.644 Z57.322 I25.0 F1.5;     // Point 6, arc cutting.
N100 G01 X76.0 Z37.644 F3.0;            // Point 7.
N110 Z20.0;                             // Point 8.
N120 X80.0;                             // Point 9.
N130 Z0;                                // Point 10.
N140 G40 X90.0;                         // Point 11, tool radius compensation cancelled.
N150 G00 X100.0 Z120.0;                 // Point S.
N160 M05;                               // Stop the first spindle.
N170 M30;                               // End of the program.
```

1.32 Setting External Workpiece Coordinate System (G50)

This command is used to set the external workpiece coordinate system.

1.32.1 Instruction Format

G50 IP__

IP__: Specifies the axis position, which represents the tool's position in the current workpiece coordinate system (the IP setting value is written in real-time to variables Sys2160 to Sys2165).

1. Absolute Specification: IP__ specifies the position, which is the current position of the tool.
2. Incremental Specification: The specified increment value is added to the current tool coordinate values before the specification, resulting in the current position of the tool.

Note: G50 does not change the values of G54 to G59 and G54.1Pxx work coordinate systems. It offsets all work coordinate systems by modifying the external workpiece coordinate system's values.

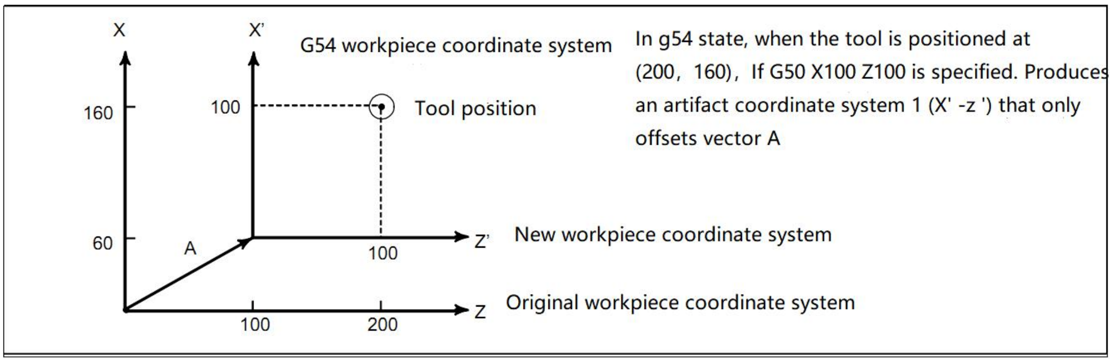

X
X'
G54 workpiece coordinate system
In g54 state, when the tool is positioned at (200，160)，If G50 X100 Z100 is specified. Produces an artifact coordinate system 1 (X' -z ') that only offsets vector A
160
100
Tool position
60
A
Z'
100
New workpiece coordinate system
100
200
Z
Original workpiece coordinate system

Figure 1.32-1

Program	Mechanical Coordinates	Program Coordinates	Workpiece Coordinate System: X-axis SYS2160, Z-axis SYS2162
G0 X0. Z0.	0.000，0.000	0.000，0.000	0，0
G0 X100. Z50.	100.000，50.000	100.000，50.000	0，0
G50 X0. Z20.	100.000，50.000	0.000，20.000	100000，30000
G0 X50. Z50.	150.000，80.000	50.000，50.000	100000，30000
G50 U20. W-10.	150.000，80.000	70.000，40.000	80000，40000
G0 X0. Z0.	80.000，40.000	0.000，0.000	80000，40000

1.33 Spindle Maximum Clamping Speed (G50)

The G50 command is used to set the maximum and minimum clamping speeds of the spindle during machining, in conjunction with G96.

1.33.1 Instruction Format

G50 S__Q__ (Modal Command)

S: The spindle's maximum clamping speed, used when controlling at a constant speed with a lower control axis position.
Q: The spindle's minimum clamping speed, used when controlling at a constant speed with a higher control axis position.

1.33.2 Supplement Explanation

1. G50 S__Q__ spindle maximum/minimum speed control is only effective in G96 mode and not in G97 mode.
2. The S and Q values specified in the G50 S__Q__ command are modal and will be saved. When switching to G96 mode again, the saved S and Q values will be used as the maximum/minimum speed limits for the spindle without the need for re-specification.
3. In G50 S__Q__ M3: During the execution of program segments, the functionality of limiting the maximum/minimum spindle speed set by S and Q in G50 S__Q__ is active. The M3 command is effective but S and Q are not used as spindle speed values.
4. G50 S__Q__X__Z__: During the execution of program segments, G50 S__Q__ sets the maximum/minimum spindle speed limits for constant linear speed control. G50 X__Z__ sets the current workpiece coordinate system.
5. S and Q values set with G50 are stored and will only be cleared when power is lost.
6. Resetting or ending a program does not cancel the G50 state. Resetting or ending a program cancels the G96 state.

1.34 Local Coordinate Setting (G52)

The G52 command sets a local coordinate system. This local coordinate system can be independently set on various workpiece coordinate systems (G54 to G59). When editing a program within a workpiece coordinate system, local coordinate systems can be established for ease of programming. These local coordinate systems are considered sub-coordinates within the workpiece coordinate system.

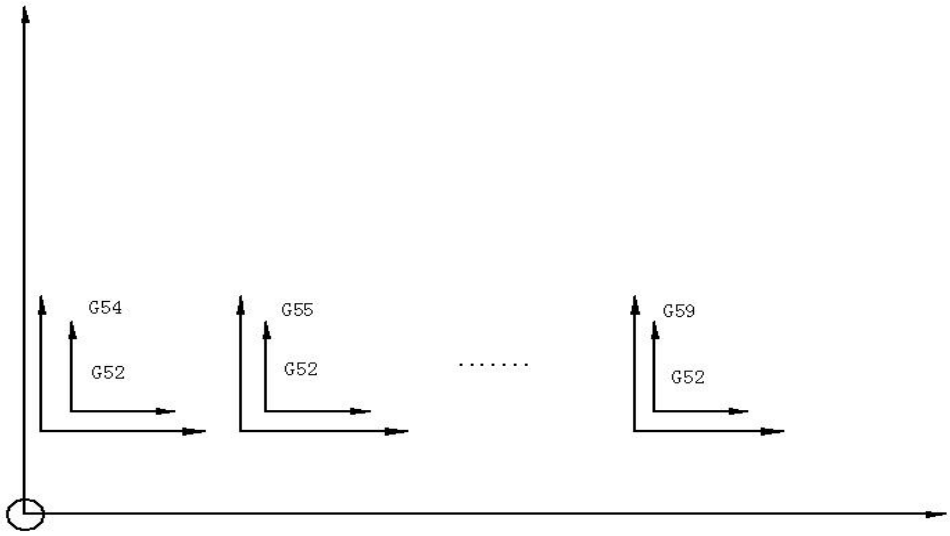


Figure 1.34-1

1.34.1 Instruction Format

G52 IP__: Set the local coordinate system.

IP: Specifies the axis for setting the local coordinate. When IP = 0, it cancels the local coordinate setting for that axis.

Note: A standalone G52 command cannot cancel the setting.

1.34.2 Supplement Explanation

Setting a local coordinate system with G52:

1. Absolute specification: When absolute coordinate values are used in the command, the local coordinates are set at the specified coordinate position within the current local coordinate system.
2. Incremental specification: When incremental coordinate values are used in the command, the local coordinate system currently in use is offset as the current local coordinate system.

Notes:

1. Local coordinate systems can be established within all workpiece coordinate systems (G54 to G59, G54.1 Pxx). The origin of a local coordinate system is defined by IP__ within the workpiece coordinate system.
2. Once a local coordinate system is set, it remains in effect for subsequent commands

until canceled.

3. When switching between workpiece coordinate systems (G54 to G59), local coordinate systems remain active.

Canceling a local coordinate system set with G52:

1. Cancel a specific axis's local coordinate offset when IP is set to 0 for that axis:
   N1 G52 X2. Z2.;
   N2 G0 X100. Z100.;
   N3 G52 X0.;
   In N3, the X-axis cancels the local coordinate offset while the Z-axis remains active.
2. Canceling local coordinate offsets upon resetting:
   MCM06 BIT00=1：Set to cancel local coordinate offsets upon resetting.
3. Canceling local coordinate offsets upon program start.

1.34.3 Example

Program Commands	Mechanical Coordinates（X，Z）	Program Coordinates（X，Z）	Local Coordinates：Sys11620，Sys11622
G0X0Z0	0.000,0.000	0.000,0.000	0,0
G0X100.Z50.	100.000,50.000	100.000,50.000	0,0
G52X10.Z10.	100.000,50.000	90.000,40.000	10000,10000
G0X0.Z0.	10.000,10.000	0.000,0.000	10000,10000
G52 U20.W-20.	10.000,10.000	-20.000,20.000	30000,-10000
X50.Z80.	80.000,70.000	50.000,80.000	30000,-10000

Program Commands	Mechanical Coordinates（X、Y、Z）	Program Coordinates（X、Y、Z）	Local Coordinates：X-Axis：Sys11620，Y-Axis：Sys11621，Z-Axis：Sys11622
G0 X0.Y0.Z0.	0.000, 0.000, 0.000	0.000, 0.000, 0.000	0,0,0
G54	0.000, 0.000,	0.000, 0.000,	0,0,0

	0.000	0.000	
G52 X10.Y20.Z30.	0.000, 0.000, 0.000	-10.000, -20.000, -30.000	10000, 20000, 30000
G0 X100.Y80.Z50.	110.000, 100.000, 80.000	100.000, 80.000, 50.000	10000, 20000, 30000
G55	110.000, 100.000, 80.000	100.000, 80.000, 50.000	10000, 20000, 30000
G0 X-100.Y-100. Z-100.	-90.000, -80.000, -70.000	-100.000, -100.000, -100.000	10000, 20000, 30000
G52 U-50.V30.W0.	-90.000, -80.000, -70.000	-50.000, -130.000, -100.000	-40000, 50000, 30000
G0 X-50. Y-50. Z-50.	-90.000, 0.000, -20.000	-50.000, -50.000, -50.000	-40000, 50000, 30000
G52 X0. Y0.	-90.000, 0.000, -20.000	-90.000, 0.000, -50.000	0,0,30000
G0X0.Y0.Z0.	0.000, 0.000, 30.000	0.000, 0.000, 0.000,	0,0,30000

Program Commands	Mechanical Coordinates（X、Y、Z）	Program Coordinates（X、Y、Z）	Local Coordinates： X-Axis：Sys11620， Y-Axis：Sys11621， Z-Axis：Sys11622
G0 X0.Y0.Z0.	0.000, 0.000, 0.000	0.000, 0.000, 0.000	0,0,0
G54	0.000, 0.000, 0.000	0.000, 0.000, 0.000	0,0,0
G52	0.000,	-10.000,	10000,

X10.Y20.Z30.	0.000, 0.000	-20.000, -30.000	20000, 30000
G0 X100.Y80.Z50.	110.000, 100.000, 80.000	100.000, 80.000, 50.000	10000, 20000, 30000
G55	110.000, 100.000, 80.000	100.000, 80.000, 50.000	10000, 20000, 30000
G0 X-100.Y-100. Z-100.	-90.000, -80.000, -70.000	-100.000, -100.000, -100.000	10000, 20000, 30000
G52 U-50.V30.W0.	-90.000, -80.000, -70.000	-50.000, -130.000, -100.000	-40000, 50000, 30000
G0 X-50. Y-50. Z-50.	-90.000, 0.000, -20.000	-50.000, -50.000, -50.000	-40000, 50000, 30000
G52 X0. Y0.	-90.000, 0.000, -20.000	-90.000, 0.000, -50.000	0,0,30000
G0X0.Y0.Z0.	0.000, 0.000, 30.000	0.000, 0.000, 0.000,	0,0,30000


```
G00 X0.0 Y0.0 Z0.0;;             // Rapid positioning
G54;;                            // Change coordinate system
G52 X10.0 Y20.0 Z30.0;           // Set X, Y, Z-Axis Local Coordinates;
G00 X100.0 Y80.0 Z50.0;          // X, Y, Z-Axis Local Coordinates active;
G55;                             // X, Y, Z-Axis Local Coordinates active;
G00 X-100.0 Y-100.0 Z-100.0;     // X, Y, Z-Axis Local Coordinates active;
G52 U-50.0 V30.0 W0.0;;          // Set Local Coordinates system for U, V, W
G00 X-50.0 Y-50.0 Z-50.0;;       // Rapid positioning
G01 X0.0 Y0.0 Z0.0;;             // G01 positioning
G52 X0.0 Y0.0;                   // Cancel X, Y-Axis Local Coordinates; Z-Axis Local
                                    Coordinates still active
G00 X0.0 Y0.0 Z0.0;;             // Rapid positioning
```

1.35 Mechanical Coordinates Positioning(G53)

The G53 command, in conjunction with the feed mode (G01 or G00), and followed by coordinate instructions, allows the tool to move to a specified position on the basic Mechanical Coordinates system.

1.35.1 Instruction Format

G53 IP__

IP: Specifies the axis for moving to the designated position in Mechanical Coordinates (can specify incremental or absolute coordinates).

1.35.2 Supplement Explanation

1. G53 is only effective for the current instruction block.
2. Using G53 does not cancel tool radius compensation and does not cancel the current work coordinate system.
3. IP__ can specify both incremental and absolute coordinates.
4. The G53 command uses the G00/G01 mode from the previous block for movement.

1.35.3 Example

Program	Mechanical Coordinates（X、Z）	Program Coordinates（X、Z）
G0 X100. Z100.	100.000,100.000	100.000,100.000
G53 X20. Z30.	20.000,30.000	20.000,30.000
G55	20.000,30.000	-30.000,-20.000
G01 X80.000，Z80.000 F50.	130.000,130.000	80.000,80.000
G53 U-20. W-20.	110.000,110.000	60.000,60.000
T01	110.000,110.000	10.000,160.000
G53 X0.Z0.	0.000,0.000	-100.000,50.000
G0 X0 Z0	100.000,-50.000	0.000,0.000

Note: The default is the G54 workpiece coordinate system, and the tool length compensation for T01 on each axis is (50.000, -100.000). The G55 workpiece coordinate system is set to (50.000, 50.000).

1.36 Work Coordinate System Setting (G54-G59)

The work coordinate system is a coordinate system used to simplify programming, with the reference point of the workpiece as the origin. You can use this command to move to a position within the work coordinate system and redefine the work coordinate system within the currently selected one, making the current tool position the specified coordinate location.

1.36.1 Instruction Format

G54-G59 IP__:Select work coordinate systems G54 to G59 and specify the axis position to move to.
G54.1 P__IP__:Select work coordinate systems G54.1 P1 to G54.1 P48 and specify the axis position to move to.

IP:Specify the axis position to move to within the work coordinate system, either as incremental or absolute coordinates.
P:Choose a group within G54.1 P1 to G54.1 P48.

Note: When IP is specified, first select the work coordinate system, then execute displacement instructions (speed is based on the speed of the previous movement section).

1.36.2 Example

Program Commands	Mechanical Coordinates（X、Z）	Program Coordinates（X、Z）
G0 X0.Z0.	0.000，0.000	0.000，0.000
G54 X100.Z100.	100.000，100.000	100.000，100.000
G55	100.000，100.000	50.000，50.000

G01 X0. Z0.F500.	50.000，50.000	0.000，0.000
G59 U50. W-50.	100.000，0.000	120.000，30.000
G0 X0. Z0.	-20.000，-30.000	0.000，0.000

Note:

G54 workpiece coordinate system is set to (0.000, 0.000), G55 workpiece coordinate system is set to (50.000, 50.000), and G59 workpiece coordinate system is set to (-20.000, -30.000).

After G54-G59 (G54.1P__) commands, you can use axis movement commands with speeds based on the speed of the previous movement section (G00 or G01).

G54 X100.0; is equivalent to G54;

X100.0;

X100.0 specifies the position of Program CoordinatesX100.0 within the G54 workpiece coordinate system.

1.37 Tilted Plane Machining (G68)

In the tilted plane machining function, you can define a new coordinate system (called the feature coordinate system) for the X, Y, and Z axes of the currently set original coordinate system (before the tilted plane machining command). This new coordinate system results from rotation and parallel movement. By using this function, you can define any plane in space and perform.

Classification of Methods for Specifying Feature Coordinate Systems:

GCode	Specifying Methods
G68.2 P0	Specifying via Euler Angles
G68.2 P1	Specifying via Roll, Pitch, and Yaw Angles
G68.2 P2	Specifying via 3 Points in a Plane
G68.2 P3	Specifying via 2 Vectors
G68.2 P4	Specifying via Projection Angles
G68.3	Specifying via Tool Axis Direction
G69	Cancel Tilted Plane Machining Mode

1.37.1 Instruction Format (G68.2 P0)

G68.2 P0 Xx___ Yy___ Zz___ Ia___ Jb___ Kc___

Tilted Plane Machining Mode Activation (Specifying Tilt Using Euler Angles) (P0 can be omitted)

X/Y/Z: Origin of the feature coordinate system specified in the absolute values of the coordinate system before the tilted plane machining command.

I/J/K: Euler Angles (-360.0° to 360.0°)

Note:

1. When X, Y, Z addresses are omitted, they are assumed to be "0".
   When X, Y, and Z are all "0", the origin of the coordinate system before the tilted plane machining command becomes the origin of the feature coordinate system.
2. When I, J, K addresses are omitted, they are assumed to be "0".
3. If addresses other than P, X, Y, Z, I, J, K are specified, a program error (P954) will occur.
   Commands with addresses other than the above will be ignored during processing.

1.37.2 Instruction Format (G68.2 P1)

G68.2 P1 Qq___ Xx___ Yy___ Zz___ Ia___ Jb___ Kc___

Tilted Plane Machining Mode Activation (Specifying Tilt Using Roll, Pitch, and Yaw Angles)

X/Y/Z: Origin of the feature coordinate system specified in the absolute values of the coordinate system before the tilted plane machining command.

Q: Rotation Sequence.

Q value	The first	The second	The third
123	X	Y	Z
132	X	Z	Y
213	Y	X	Z
231	Y	Z	X
312	Z	X	Y
321	Z	Y	X
Note: When the Q address is omitted, the default value for Q is 123.			

I: Angle of rotation around the X-axis (roll angle) (set range: -360.0° to 360.0°).
J: Angle of rotation around the Y-axis (pitch angle) (set range: -360.0° to 360.0°).
K: Angle of rotation around the Z-axis (yaw angle) (set range: -360.0° to 360.0°).

Note:

1. When addresses X, Y, Z are omitted, they are considered as "0".
   If X, Y, and Z are all "0", the origin of the coordinate system before the tilted plane command becomes the origin of the feature coordinate system.
2. When addresses I, J, K are omitted, they are considered as "0".
3. If an address other than P, Q, X, Y, Z, I, J, K is specified, a Program error (P954) will occur.
   Non-specified address commands other than the above will be ignored.
4. If the value of "q" is not one of the specified values mentioned above, a Program error (P954) will occur.
   The TWPC_A7 library returns the error TWPC_A7_STV_ERRORS_P1_Q_ERROR in the setTiltedVars function.

1.37.3 Instruction Format (G68.2 P2)

G68.2 P2 Q0 Xx0___ Yy0 ___ Zz0 ___ Ra___
G68.2 P2 Q1 Xx1___ Yy1___ Zz1___
G68.2 P2 Q2 Xx2___ Yy2 ___ Zz2___
G68.2 P2 Q3 Xx3___ Yy3___ Zz3___

❖ Tilted plane machining mode activation (specifying a tilted plane through three points in a plane)
  Q: Select the specified point from the first point to the third point or an offset.
  0: Offset
  1: First point
  2: Second point
  3: Third point
❖ X0/Y0/Z0: Offset from the first point to the origin of the feature coordinate system.
❖ Specified using increments relative to the feature coordinate system before parallel movement.
❖ a: Angle of rotation to make the feature coordinate system rotate around the Z-axis (-360.0° to 360.0°).
❖ X1/Y1/Z1: Position of the first point in the workpiece coordinate system, specifying

the origin of the feature coordinate system.

- X2/Y2/Z2: Position of the second point in the workpiece coordinate system, setting a point on the X-axis (positive direction) of the feature coordinate system.
- X3/Y3/Z3: Position of the third point in the workpiece coordinate system, specifying a point on the Y-axis.

Note:

1. When address Q is omitted, it is considered as specifying "0".
2. If addresses X, Y, Z are omitted between Q0 and Q3, the omitted addresses are considered as specifying "0".
3. When address R is omitted, it is considered as specifying "0".
4. If an address other than P, Q, X, Y, Z, R is specified, a Program error (P954) will occur.
   - Non-specified address commands other than the above will be ignored.
5. A Program error (P954) occurs in the following cases:
   - There are other commands between G68.2 P2 Q0 and Q3.
   - Any one of G68.2 P2 Q1 to Q3 is missing.
   - G68.2 P2 Q0 to Q3 are repeated.
   - A value other than 0 to 3 is specified for address Q.
   - R is specified in multiple Program sections.
6. A Program error (P955) occurs in the following cases:
   - Two or more points among the first point to the third point are specified as the same point.
   - All three points among the first point to the third point lie on the same straight line.
   - The distance from one of the points connecting the first point to the third point to the other two points is less than 0.1 [mm].
   - The TWPC_A7 library returns an error TWPC_A7_STV_ERRORS_P2_TITLEDVARS_INPUT_ERROR.

1.37.4 Instruction Format (G68.2 P3)

G68.2 P3 Q1 Xx___ Yy___ Zz___ Iix___ Jjx___ Kkx___

G68.2 P3 Q2 Iiz___ Jjz___ Kkz___

- Tilted Plane Machining Mode Activation (Specifying Tilted Plane with 2 Vectors)
- Q: Select a vector. Choose the vector direction along the X-axis or the Z-axis.

- ❖ 1: X-axis direction vector
- ❖ 2: Z-axis direction vector
- ❖ X/Y/Z: The origin of the feature coordinate system. Use the absolute values of the coordinate system before the tilted plane machining command.
- ❖ Ix/Jx/Kx: Specify the direction of the X-axis vector in the feature coordinate system on the coordinate system before the tilted plane machining command. The range is the same as the axis setting range, with no physical units.
- ❖ Iz/Jz/Kz: Specify the direction of the Z-axis vector in the feature coordinate system on the coordinate system before the tilted plane machining command. The range is the same as the axis setting range, with no physical units.

Note:

1. When addresses X, Y, Z are omitted, it is considered as specifying "0". When X, Y, Z are all "0", the origin of the coordinate system before the tilted plane machining command becomes the origin of the feature coordinate system.
2. If addresses I, J, K are omitted between G68.2 P3 Q1 and Q2, the omitted addresses are considered as specifying "0".
3. If an address other than P, Q, I, J, K is specified, a Program error (P954) will occur. (X, Y, Z can also be used with G68.2 P3 Q1.) Non-specified address commands other than the above will be ignored.
4. A Program error (P954) occurs in the following cases:
   - ❖ There are other commands between G68.2 P3 Q1 and Q2.
   - ❖ Any one of G68.2 P3 Q1 to Q2 is missing.
   - ❖ G68.2 P3 Q1 to Q2 are repeated.
   - ❖ A value other than 1 to 2 is specified for address Q.
   - ❖ Address Q is omitted.
5. A Program error (P955) occurs in the following cases:
   - ❖ The values of Ix, Jx, Kx are all "0".
   - ❖ The values of Iz, Jz, Kz are all "0".
   - ❖ The X-axis direction vector and Z-axis direction vector of the feature coordinate system deviate more than 5 degrees in the vertical direction.
   - ❖ The TWPC_A7 library returns an error TWPC_A7_STV_ERRORS_P3_TITLEDVARS_INPUT_ERROR.

1.37.5 Instruction Format (G68.2 P4)

G68.2 P4 Xx___ Yy___ Zz___ Ia___ Jb___ Kc

❖ Tilted Plane Machining Mode Activation (Specifying Tilted Plane with Projection Angle)
❖ X/Y/Z: The origin of the feature coordinate system. Use the absolute values of the coordinate system before the tilted plane machining command.
❖ I: Angle to rotate the X-axis around the Y-axis of the coordinate system before the tilted plane machining command (-360.0° to 360.0°).
❖ J: Angle to rotate the Y-axis around the X-axis of the coordinate system before the tilted plane machining command (-360.0° to 360.0°).
❖ K: Angle to rotate around the Z-axis of the feature coordinate system (-360.0° to 360.0°).

Note:

1. When addresses X, Y, Z are omitted, it is considered as specifying "0". When X, Y, Z are all "0", the origin of the coordinate system before the tilted plane machining command becomes the origin of the feature coordinate system.
2. If addresses I, J, K are omitted, it is considered as specifying "0".
3. If an address other than P, X, Y, Z, I, J, K is specified, a Program error (P954) will occur. Non-specified address commands other than the above will be ignored.
4. If the angle between X-axis after rotating a degrees around the Y-Axis and Y-axis after rotating b degrees around the X-Axis is less than 1 degree, a Program error (P955) will occur.
   ❖ The TWPC_A7 library returns an error TWPC_A7_STV_ERRORS_P4_TITLEDVARS_INPUT_ERROR.

1.37.6 Instruction Format (G68.3)

❖ G68.3 Xx Yy Zz Ra; ... Tilted Plane Machining Mode Activation (Specifying Tilted Plane Based on Tool Axis Direction)
❖ X/Y/Z: The origin of the feature coordinate system. Use the absolute values of the coordinate system before the tilted plane machining command.
❖ a: Angle to rotate the feature coordinate system around the Z-axis (-360.0° to 360.0°).

Note:

1. When addresses X, Y, Z are omitted, it is considered as specifying "0". When X, Y, Z are all "0", the origin of the coordinate system before the tilted plane machining command becomes the origin of the feature coordinate system.
2. When address R is omitted, it is considered as specifying "0".

3. If an address other than X, Y, Z, or R is specified, a Program error (P954) will occur. Non-specified address commands other than the above will be ignored.

1.37.7 Explanation

1. Permissible Commands in Tilted Plane Machining

In tilted plane machining, if any command other than the following is executed, a Program error (P953) will occur.

Commands	Functionality	Commands	Functionality
G00, G01 G02, G03 G02.1, G03.1	Positioning.Linear Interpolation.Circular Interpolation.Helical Interpolation.Spiral Interpolation	G43, G44, G49 G43.1 G43.4, G43.5	Tool Length Compensation Cancel Positive/Negative/Cancel.Tool Axis Length Compensation.Tool Tip Point Control Type I/II
G04	Pause	G45,G46,G47,G48	Tool Position Offset
G05 P0, P1, P2, P10000	High-Speed Machining Mode.High-Speed High-Precision Control II	G50,G51	Scale Cancel/Activate
G05.1 Q0, Q1	High-Speed High-Precision Control I	G50.1, G51.1	G-Code Mirroring Cancel/Activate
G08 P1	High-Precision Control	G53	Mechanical Coordinates System Selection
G09	Exact Stop Check	G53.1	Tool Axis Direction Control
G10/G11	Program Parameter Input/Cancel.Compensation Input	G61 G61.1 G62 G64	Exact Stop Check Mode.High Precision Control I Activate.Automatic Angle Rate.Cutting Mode
G12, G13	Circular Cutting	G65	User Macro Program Single Call

G17, G18, G19	Plane Selection	G66, G66.1, G67	User Macro Program Modal Call A/B/Cancel
G22/G23	Forward Travel Check Function Enable/Disable	G69	Coordinate Rotation/Tilted Plane Machining Cancel
G28	Automatic Return to the 1st Reference Point	G70～ G76, G80 ～G89	Fixed Cycle for Drilling *Including Rigid Tapping
G29	Starting Point Return	G90, G91	Absolute Value Instruction/Incremental Value Instruction
G30	Return to Reference Points 2 to 4	379 G93 G94 G95	Inverse Time Feed.Feed per Minute. Feed per Revolution
G30.1～ G30.6	Tool Change Position Return	G98, G99	Fixed Cycle Initial Level Return/R-Point Level Return
G34, G35, G36, G37.1	Special Fixed Cycle	M98, M99	Subprogram Call/Main Program Return
G40, G41, G42	Tool Radius Compensation Cancel/Left/Right	M,S,T,B	M,S,T,B Instructions
F	Feedrate Command	Macro Program Commands	Local Variables, Global Variables, Arithmetic Instructions (Addition, Subtraction, Multiplication, Division, etc.), Control Instructions (IF-THEN-GOTO, WHILE-DO, etc.)

2. Modes for Tilted Plane Machining (Including Cancel Commands)

Machining G68.2 and G68.3 commands in modes other than the ones listed below will result in a Program error (P952).

Modes	Functionality	Modes	Functionality
G00, G01	Positioning, Linear Interpolation	G50	Cancel Scaling
G05 P0, P1, P2	High-Speed Machining Mode	G50.1	Disable G Code Mirroring
G05.1 Q0, Q1	High-Speed High-Precision Control I	G54 ～G59, G54.1	Workpiece Coordinate System, Extended Workpiece Coordinate System Selection
G08 P1	High-Precision Control	G54.4 Pp	Workpiece Setting Error Compensation
G13.	Polar Coordinate Interpolation Cancel	G61 G61.1 G64	Exact Stop Check Mode.Enable High-Precision Control I.Cutting Mode
G15	Polar Coordinate Command Cancel	G67	Disable User Macro Program Modal Calling
G17, G18, G19	Plane Selection	G69	Cancel Coordinate Rotation, Three-Dimensional Coordinate Transformation
G20, G21	Imperial Units Instruction, Metric Units Instruction	G80	Cancel Fixed Cycle
G22/G23	Enable/Disable Advance Stroke Check Function	G90, G91	Absolute Value Instruction / Incremental Value Instruction
G40	Cancel Tool Radius Compensation	G93	Inverse Time Feed Rate
G43, G44 G49	Tool Length Compensation.Cancel Tool Length	G94 G95	Feed Rate per Minute.Feed Rate per Revolution

	Compensation		
		G97	Disable Constant Surface Speed Control
		G98, G99	Cancel Fixed Cycle Initial Level Return, R-Point Level Return

1.37.8 Example

G00 X0 Z0 Y0；

G68.2 P1 J45 X0 Z0；In P1 mode, specify X0 Z0 as the origin of the rotation coordinate system and rotate around the Y-Axis by 45 degrees.

G0 Z100；

M30；

1.38 Multiple repetitive cutting cycles (G70-G76)

Simplify the creation of CNC programs by allowing the CNC machine to automatically determine the tool path for rough machining based on the shape of the workpiece. These cycles are also suitable for thread cutting and are particularly useful for processing cylindrical materials.

1.38.1 Fine Turning Cycle Cancel (G70)

After rough machining with commands like G71, G72, or G73 for semi-finished products, the final precision machining is done using G70 to achieve precise workpieces.

1.38.1.1 Instruction Format

G70 P(ns)___Q(nf)___

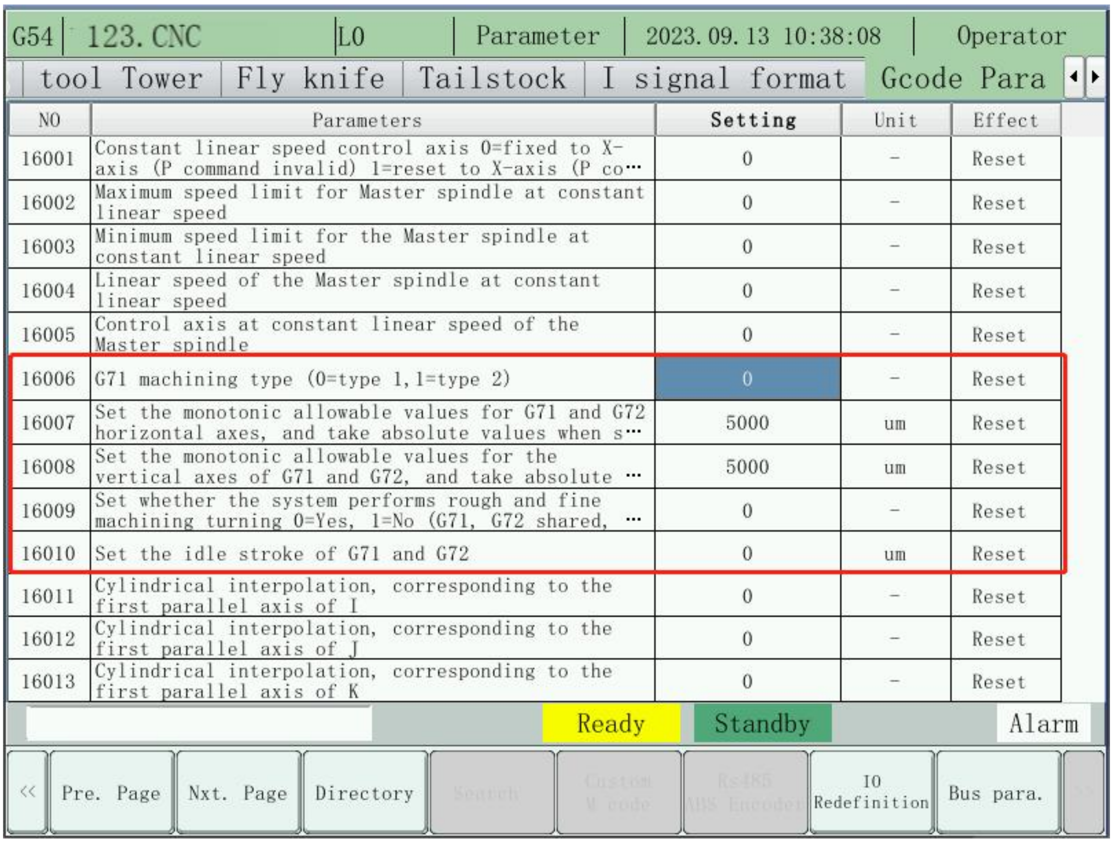
P(ns): The first block number of the precision machining cycle program.

Q(nf): The last block number of the precision machining cycle program.

1.38.2 Compound Horizontal Rough Turning Cycle (G71)

This command calls a shaping program, automatically calculates the machining path, and simultaneously executes transverse rough machining. It is typically used in conjunction with G70.

Parameter Setting:



NO	Parameters	Setting	Unit	Effect
16001	Constant linear speed control axis 0=fixed to X-axis (P command invalid) 1=reset to X-axis (P co…	0	-	Reset
16002	Maximum speed limit for Master spindle at constant linear speed	0	-	Reset
16003	Minimum speed limit for the Master spindle at constant linear speed	0	-	Reset
16004	Linear speed of the Master spindle at constant linear speed	0	-	Reset
16005	Control axis at constant linear speed of the Master spindle	0	-	Reset
16006	G71 machining type (0=type 1, 1=type 2)	0	-	Reset
16007	Set the monotonic allowable values for G71 and G72 horizontal axes, and take absolute values when s…	5000	um	Reset
16008	Set the monotonic allowable values for the vertical axes of G71 and G72, and take absolute …	5000	um	Reset
16009	Set whether the system performs rough and fine machining turning 0=Yes, 1=No (G71, G72 shared, …	0	-	Reset
16010	Set the idle stroke of G71 and G72	0	um	Reset
16011	Cylindrical interpolation, corresponding to the first parallel axis of I	0	-	Reset
16012	Cylindrical interpolation, corresponding to the first parallel axis of J	0	-	Reset
16013	Cylindrical interpolation, corresponding to the first parallel axis of K	0	-	Reset

Figure 1.38-1

1.38.2.1 Instruction Format

G71 U(△d)___R(e)___H(h)___

G71 P(ns)___Q(nf)___U(△u)___W(△w)___F(f)___S(s)___T(t)___

N(ns)

… The program segments from "ns" to "nf" specify the fine turning shape from A to A1 to B.

N(nf)

U(△d): Cutting depth (specified by radius, specified as positive, absolute value taken when specified as negative).
R(e): Rough turning retreat amount (specified by radius, specified as positive, absolute value taken when specified as negative).
H(h): Retreat method selection. Defaults to zero when not specified. When set to other than 1, it is considered as 0, unaffected by decimals.

Explanation:
H=0:Type I
In each cycle, the tool retreats in the 45° direction and finally outlines the machining path. Both the first and second axes in the plane must be monotonically increasing or decreasing. An alarm is triggered in case of grooves.
H=1: Type II
In each cycle, cutting is done along the fine turning path. The first axis in the plane must be monotonically increasing or decreasing, but the second axis can be non-monotonic. When using H=1, the starting point calculation is done in the roughing program, and it's important to note the monotonicity, which is not related to the rough turning retreat amount R(e).

P(ns): The first single-segment number of the fine turning cycle program, unaffected by decimals.
Q(nf): The last single-segment number of the fine turning cycle program, unaffected by decimals.
U(△u): Allowance for precision machining in the X-direction.
W(△w): Allowance for precision machining in the Z-direction.
F(f), S(s), T(t): F=feed rate, S=spindle speed, T=tool selection number.

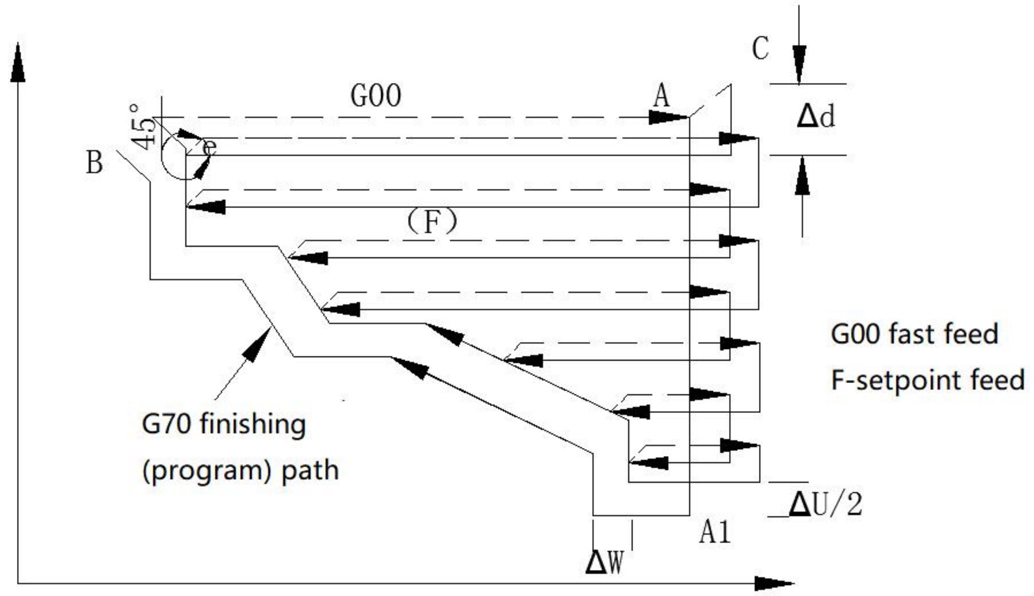


Figure 1.38-2

As shown in the above figure, the fine machining path is from A to A1 to B. The fine machining path, offset by the fine turning allowance (△u, △w), follows the trajectory formed by executing G71. The distance from A to C represents the fine machining allowance for tool retraction.

Note:

When Diameter Programming is used, the fine turning allowances are (△u/2, △w).

1.38.2.2 Example 1

Two-axis plane with monotonicity, H is 0 (Type I).

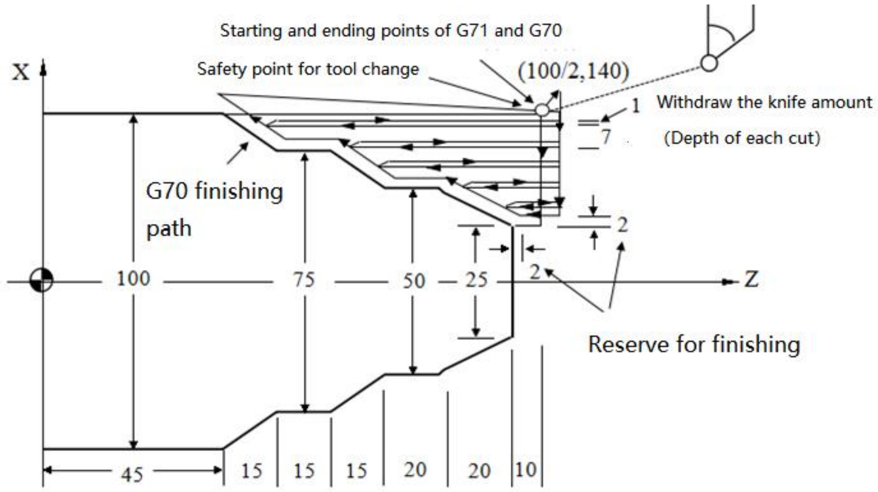


Figure 1.38-3

Diameter Programming

```
G99;                                  // Switch to feed per revolution
G28;                                  // Return to reference point
T0202;                                // Tool change to tool #2
M3 S3000;                             // Start spindle #1, forward rotation, speed
                                         3000 RPM
G00 X100.000;                         // Rapid positioning along X-Axis
Z140.000;                             // Rapid positioning along Z-Axis
G71 U7.000 R1.000 H0;                 // G71: Cutting depth = 7, Rough turning
                                         allowance = 1, Retract mode 1
G71 P100 Q200 U4.000 W2.000 F2.00;    // Start fine turning from line number 100 to
                                         200, X-axis fine turning allowance = 4,
                                         Z-Axis fine turning allowance = 2, Feed
                                         AA-rate = 2.00
N100 G01 X25.0 F1.50;                 // Fine turning path
W-10.000;                             // Fine turning path
X50.000 W-20.000;                     // Fine turning path
W-20.000;                             // Fine turning path
X75.000 W-15.000;                     // Fine turning path
W-15.000;                             // Fine turning path
N200 X100.000 W-15.000;               // Fine turning path
G00 X110.000;                         // Rapid positioning along X-Axis
Z150.000;                             // Rapid positioning along Z-Axis
T0303;                                // Tool change to tool #3
G00 X100.000;                         // Rapid positioning along X-Axis
```

Z140.000; // Rapid positioning along Z-Axis

G70 P100 Q200; // Cancel fine turning cycle

M05; // Stop spindle #1

M30; // End of program

Action Explanation

1. Before the cycle, the tool should be rapidly positioned (G00) to point A (starting point).
2. After executing the G71 command, the tool offsets itself by the specified fine turning allowances (ΔU/2 for X-Axis and ΔW for Z-Axis) to point C.
3. The tool then moves Δd distance along the X-Axis and begins feeding towards the contour.
4. After retracting e distance in the X-Axis direction and 45°, the Z-Axis feeds in the opposite direction, retracting to the adjacent starting point along the X-Axis.
5. After moving Δd distance along the X-Axis, the next repetition cycle continues.
6. When the last cycle ends, the tool follows the contour A'→B for one pass.
7. Upon completion, the tool will rapidly return to point A and wait for the next cycle of machining to begin.

This is for a plane with two monotonic axes, and H is set to 1 (Type II).

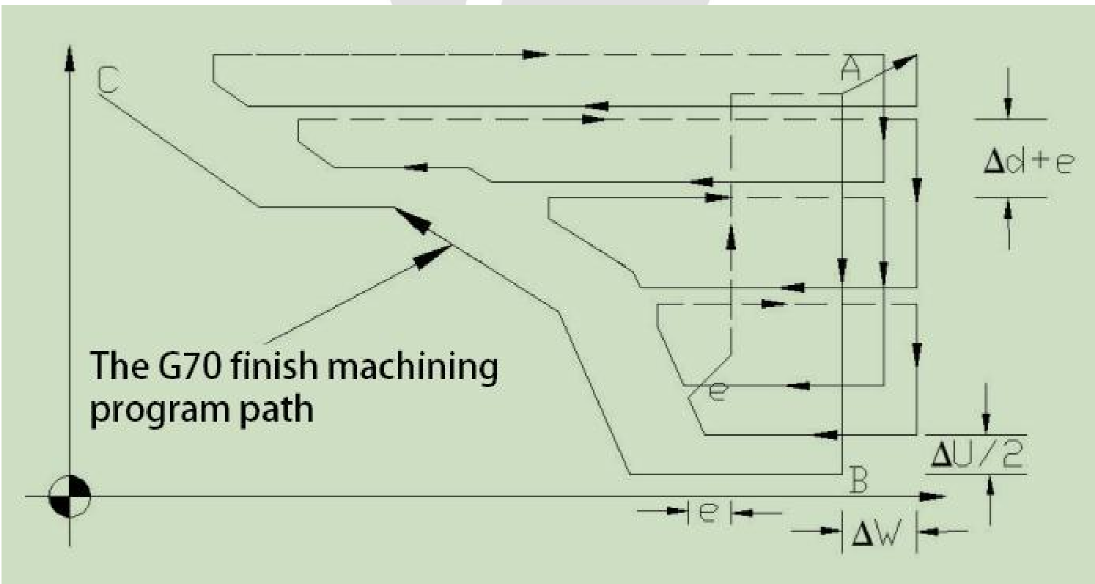


Figure 1.38-4

Diameter Programming

G99; // Switch to feed per revolution mode.

M03 S1000; // Start the first spindle at 1000 RPM

G0 X80. Z80.; // Rapid positioning to the starting point

G71 U10. R3. H1; // G71: Cutting depth 10, Rough cut retraction 3, Retraction type 2

G71 P10 Q20 U3. W1.5 F5.; // Start of fine machining, block numbers 10 to 20, X-axis allowance 3, Z-axis allowance 1.5, Feed rate 2

N10 G0 X15. Z65.; // Fine machining path

G1 Z55. F450.; // Fine machining path

G1 X30.; // Fine machining path

G3 X40. Z50. R5.; // Fine machining path

G1 Z42.; // Fine machining path

G1 X50.; // Fine machining path

G1 X55. Z35.; // Fine machining path

N20 G1 X60.; // Fine machining path

G70 P10 Q20; // Cancel fine machining cycle

M5; // Stop the first spindle

M30; // End of program

Plane first axis monotonic, second axis non-monotonic, H equals 1 (Type II).

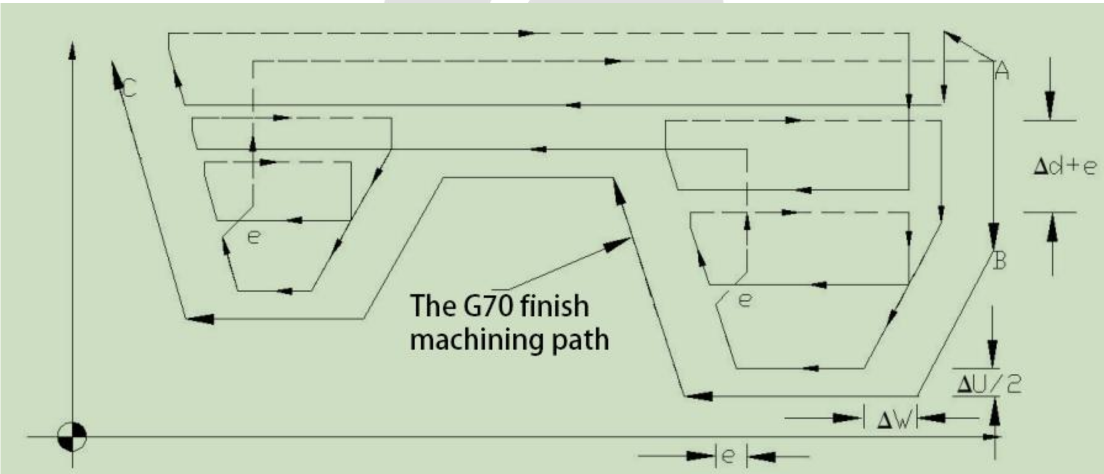


Figure 1.38-5

Program Example

G99; //Switch to feed per revolution mode.

M3 S1000; //The first spindle is rotating in the forward direction at a speed of 1000 RPM.

G0 X110. Z100.; //Rapid positioning of the X and Z axes.

G71 U5. R2. H1 ; //G71 with a cutting depth of 5, rough retraction of 2, and retraction type 2.

G71 U2. W2. P100 Q200 F5.; //Starting block number for fine machining: 100, ending block number: 200, X-Axis fine machining allowance: 2,

```
                                 Z-Axis fine machining allowance: 2, feed rate: 5.
N100 G01 X100.W0.;               //Fine machining path.
G01X0.Z90.;                      //Fine machining path.
G01 Z80. F3000.;                 //Fine machining path.
G01 Z65. X30.;                   //Fine machining path.
G03 Z45. X25. R30.;              //Fine machining path.
G01 Z40. X15.;                   //Fine machining path.
G01 Z30. Z25. X40. ;             //Fine machining path.
G02 Z10. X70. R50. ;             //Fine machining path.
N200 G01 Z0. X105. ;             //Fine machining path.
G01 X110. F4000. ;               //X-Axis positioning.
Z110.;                           //Z-Axis positioning.
M30;                             //Program end.
```

Note: If a program segment during processing is higher than the program segment at the end of processing, an alarm 71/72-5 will occur.

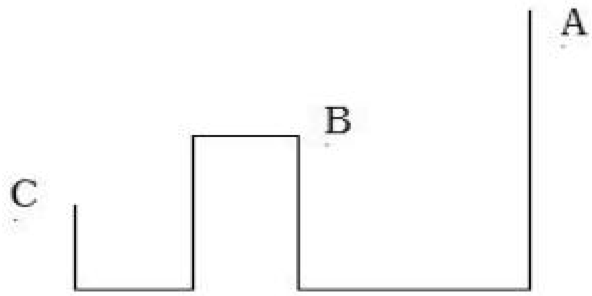


Figure 1.38-6

Sys3053 setting, whether the system alarms when the Z-axis changes non-monotonically, as shown in the figure below.

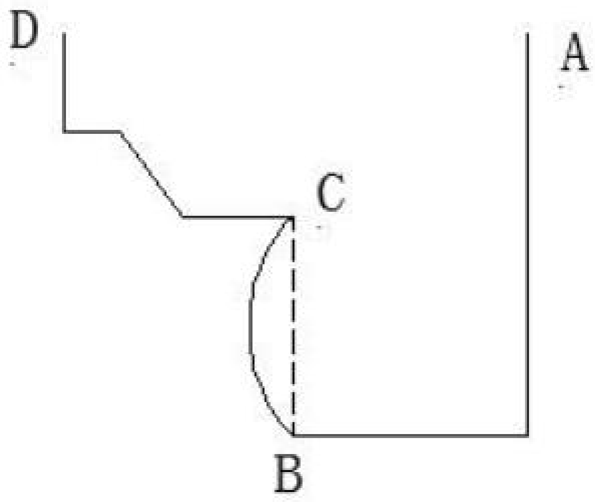


Figure 1.38-7

During the movement from B to C, there is a non-monotonic change in the Z-axis direction.

Based on sys3053 (allowable value for the horizontal axis monotonicity), it is determined that when the non-monotonic part of the horizontal axis does not exceed the value set by sys3053, the system does not trigger an alarm. If it exceeds this value, it triggers alarm 71-2.

1.38.3 Compound Longitudinal Rough Turning Cycle (G72)

Call a shaping program and automatically calculate the toolpath while performing longitudinal rough machining.

1.38.3.1 Instruction Format

G72 W(△d)___R(e)___H(h)___

G72 P(ns)___Q(nf)___U(△u)___W(△w)___F(f)___S(s)___T(t)___

N(ns)
…　　Program segments from ns to nf specify the finishing shape from A→A1→B.
N(nf)

W(△d): Cutting depth (specified as a radius, positive values specified, absolute value taken when specified as negative).

R(e): Rough tool retraction (specified as a radius, positive values specified, absolute value taken when specified as negative).

H(h): Retraction method selection, default to zero when not specified, set to anything other than 1 is treated as 0, unaffected by decimal point.

Explanation:

H=0: Type I

Each cycle retracts the tool by 45°, and finally, the machining path is drawn. Both the first and second axes in the plane must be monotonically increasing or decreasing. An alarm is triggered if there are concavities.

H=1: Type II

Each cycle cuts according to the finishing path. The first axis in the plane must be monotonically increasing or decreasing, while the second axis can be non-monotonic.

P(ns): The first single section number of the finishing cycle program, unaffected by the decimal point.

Q(nf): The last single section number of the finishing cycle program, unaffected by the decimal point.

U(△u): Reserved amount for precision machining in the X direction.

W(△w): Reserved amount for precision machining in the Z direction.

F(f), S(s), T(t): F=feed rate, S=spindle speed, T=tool selection.

In G72, any F, S, and T commands in this single section and before are effective, while any F, S, and T commands between N(ns) and N(nf) are ineffective for G72 and are only used for the precision machining command G70.

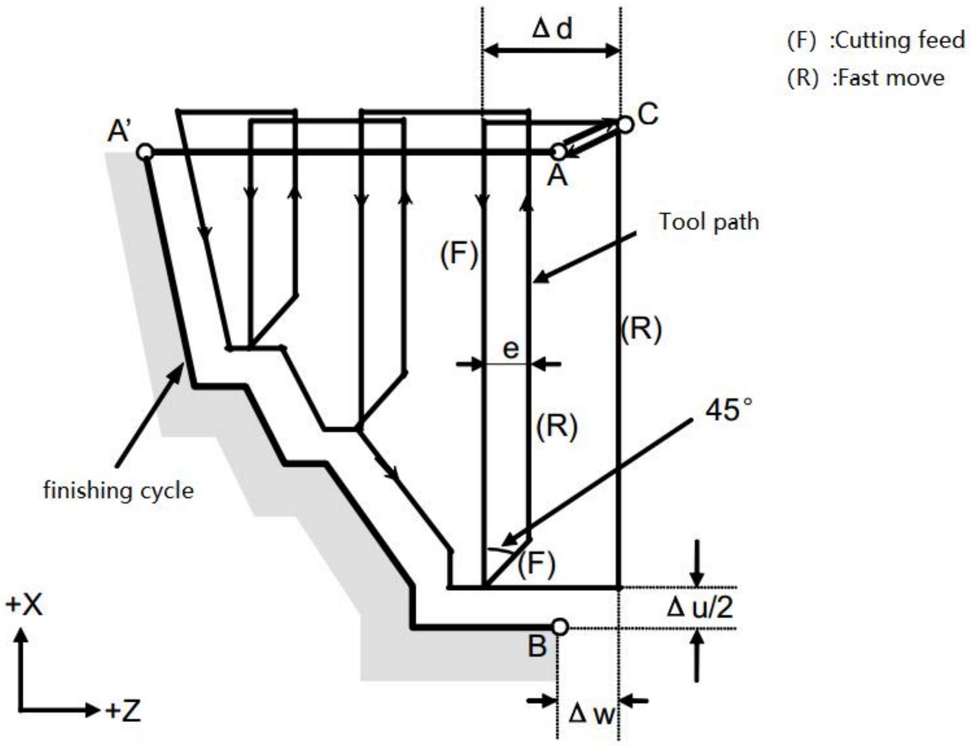


Figure 1.38-8

Track Explanation:

As shown in the figure below, the precision machining path is from C to A to B. The precision machining path is the trajectory offset by the precision turning allowance (Δu, Δw), and it is formed using G72.

Note: When in Diameter Programming, the precision turning allowance is (Δu/2, Δw).

1.38.3.2 Supplement Explanation

1. Sections N(ns) to N(nf) specify the machining path from C to B to A.
2. You can set up to 64 individual sections between N(ns) and N(nf).
3. The individual sections between N(ns) and N(nf) can call subprograms.
4. The speed between C and B can be either G00 or G01.
   - ❖ Type I: The feed mode (G00 or G01) for the first motion command between N(ns) and N(nf) is determined by the motion type.
   - ❖ Type II: The feed mode (G00 or G01) is determined by the feed mode of the second plane axis in the machining path. G00 and G01 can coexist.
5. Cutting depth U(Δd) and roughing tool retraction R(e) are modal commands and will

not change until another numerical command is given.

6. When H=0, Type I, G72 can be applied to four types of cutting cycles, all of which are parallel to the Z-axis. The positive or negative values of U and W depend on the direction of the reserved machining, as shown in the figure below:

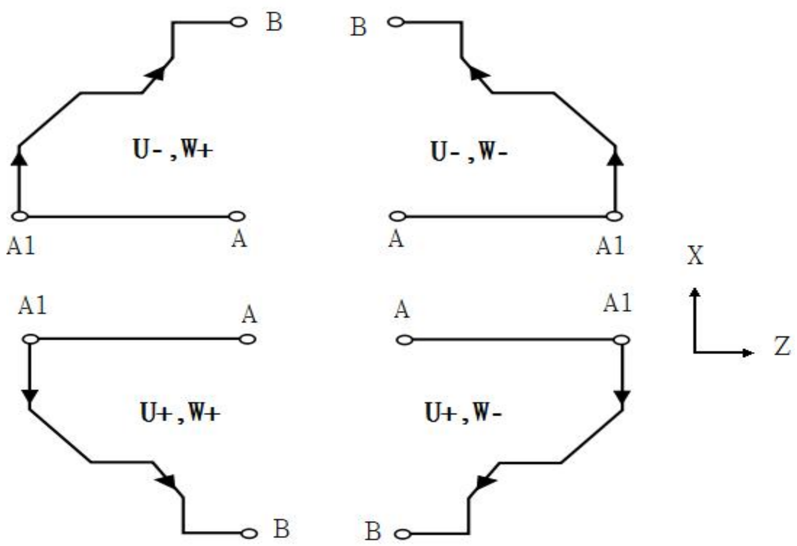


Figure 1.38-9

1) In the case of (W+), shapes located higher than the cycle starting point cannot be machined; otherwise, an alarm will occur. In the case of (W-), shapes located lower than the cycle starting point cannot be machined; otherwise, an alarm will occur.
2) The handling of rough and finish turning is determined based on Sys3055. (Please refer to the figure for the parameter setting method.)

Definition of Sys3055	Range.	Default value (=0).
Whether to execute rough and finish machining.	0～1	=0: Execute rough and finish machining. =1: Do not execute rough and finish machining.

3) Retract handling is consistent with G71:
   - ❖ Type I: Retract at 45°.
   - ❖ Type II: Retract following the finish machining shape.
4) Bottom-of-the-valley retract handling is consistent with G71.

1.38.3.3 Example 1

H=0 will execute TYPE I machining mode.

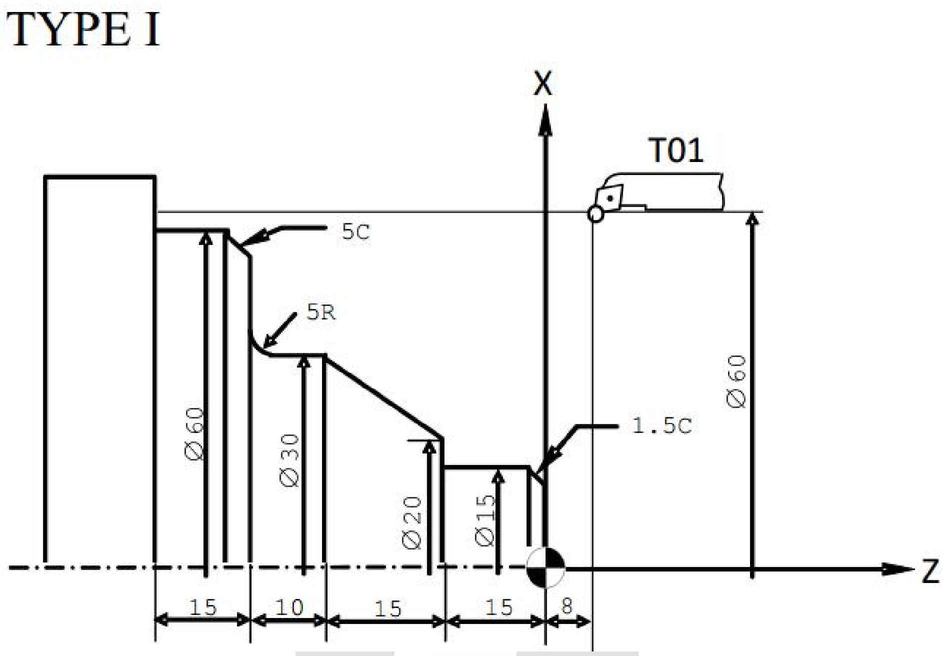


Figure 1.38-10

Diameter Programming

T01;	// Use tool No. 1;
G50 S5000;	// Speed limit set to 5000 rpm;
G96 M03 S130;	// Constant speed, table feed rate 130 m/min, spindle forward;
G00 X60.0 Z8.0;	// Rapid positioning to the starting point;
M08;	// Coolant on;
G72 W3.0 R1.0 H0;	// Z-axis cutting depth 3.0mm, retraction 1.0mm; H can be omitted, equivalent to G72 U3.0 R1.0;
G72 P01 Q02 U0.8 W0.2 F0.6;	// Perform face roughing cycle, with block from sequence N01 to N02, X-Axis machining allowance 0.8mm, Z-Axis machining allowance 0.2mm, feed rate 0.6mm/rev;
N01 G00 Z-55.0;	// Preparing for contour turning;
G01 X60.0;	
Z-45.0;	
X50.0 Z-40.0;	
X40.0;	

G03 X30.0 Z-35.0 R5.0;
G01 Z-30.0;
X20.0 Z-15.0;
X15.0;
Z-1.5;
N02 X12.0 Z0.0;
M09; // Coolant off;
G28 X60.0 Z10.0; // Tool rapid movement to the specified intermediate point, then return to machine zero;
M05; // Spindle stop;
M30; // Program end;

1.38.3.4 Example 2

H=1, the TYPE II machining method will be executed.

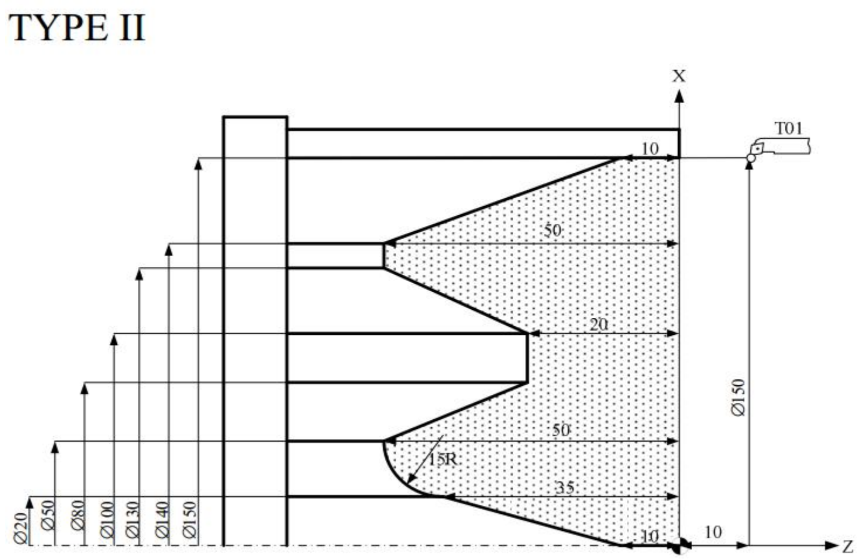


Figure 1.38-11

Diameter Programming

T01; // Use tool No. 1;
G50 S5000; // Speed limit set to 5000 rpm;
G96 M03 S130; // Constant speed, Table surface speed at 130 m/min, main spindle forward rotation;
M08; // Turn on cutting fluid;
G00 X150.0 Z10.0; // Rapid positioning to the starting point;

```
G72 W2.0 R1.0 H1;              // Cutting depth in Z-axis is 2.0mm, retraction amount
                                is 1.0mm, H can be omitted, e.g., G72 W2.0 R1.0 is
                                equivalent;
G72 P01 Q02 U0.8 W0.1 F0.6;    // Execute face rough turning cycle, the block is from
                                sequence N01 to N02, X-Axis fine turning allowance
                                is 0.8 mm, Z-Axis fine turning allowance is 0.1 mm,
                                feed rate 0.6 mm/rev;
N01 G00 X150.0 Z0.0;            // Prepare for turning the contour;
G01 Z-10.0;
X140.0 Z-50.0;
X130.0;
X100.0 Z-20.0;
X80.0;
X50.0 Z-50.0;
G03 X20.0 Z-35.0 R15.0;
G01 X20.0;
X0.0 Z-10.0;
N02 X0.0 Z0.0;
M09;                            // Turn off cutting fluid;
M05;                            // Stop the main spindle;
M30;                            // End of program;
```

1.38.4 Multiple Contour Roughing Cycle (G73)

The G73 command is used for workpieces that have been pre-machined using methods like forging or casting, and their shape is similar to the finished product. This command is used to save machining time.

1.38.4.1 Instruction Format

G73 U(△i)___W(△k)___R(d)___

G73 P(ns)___Q(nf)___U(△u)___W(△w)___F(f)___S(s)___T(t)___

U(Δi): Cutting amount in the X-axis (outer diameter). When not specified, use the parameter [G73 Total X Cut] setting.

W(Δk): Cutting amount in the Z-axis (length). When not specified, use the parameter [G73

Total Z Cut] setting.

R(d): Number of cutting passes, i.e., how many times the cutting in the X and Z-axes should be divided. When not specified, use the parameter [G73 Division Count] setting.

P(ns): Start of the loop sequence number.

Q(nf): End of the loop sequence number.

U(Δu): Fine finishing allowance in the X-axis (outer diameter).

W(Δw): Fine finishing allowance in the Z-axis (length).

F: Feed rate.

T: Tool number.

S: Spindle speed setting.

1.38.4.2 Supplement Explanation

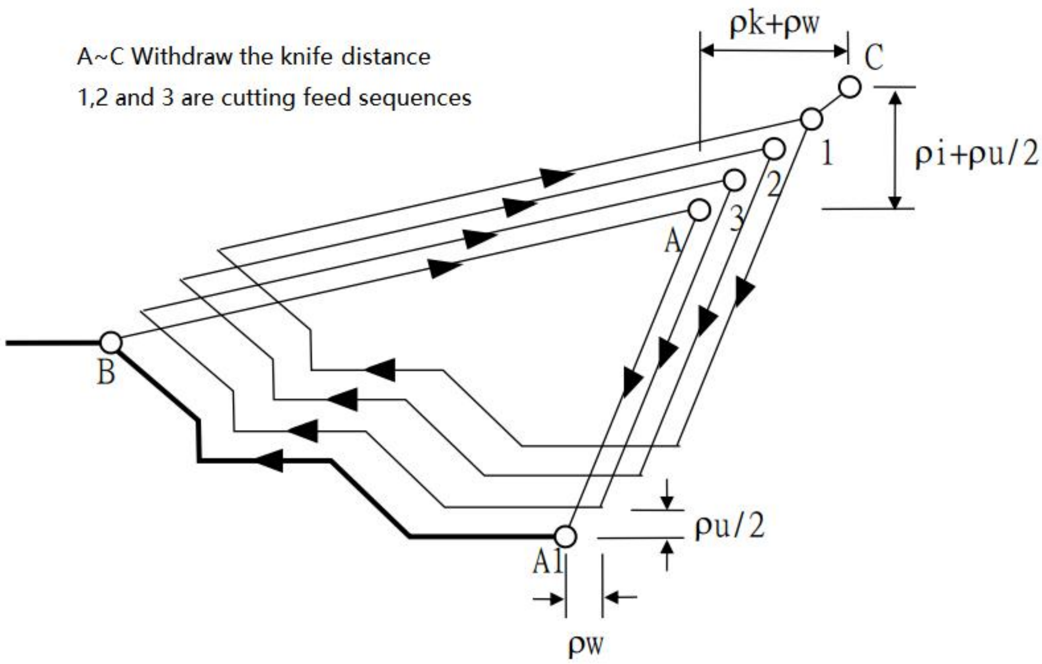


Figure 1.38-12

1. The sequence numbers N(ns)-N(nf) specify the machining path from A to A1 to B.
2. A maximum of 64 individual sequences can be defined between sequence numbers N(ns) and N(nf).
3. Subprograms cannot be called between sequence numbers N(ns) and N(nf).
4. When the machining cycle is completed, the tool returns to point A.
5. The X and Z-axis cutting amounts U(Δi) and W(Δk), as well as the number of divisions R(d), are modal instructions and won't change until another numeric value is

specified.

1.38.4.3 Example

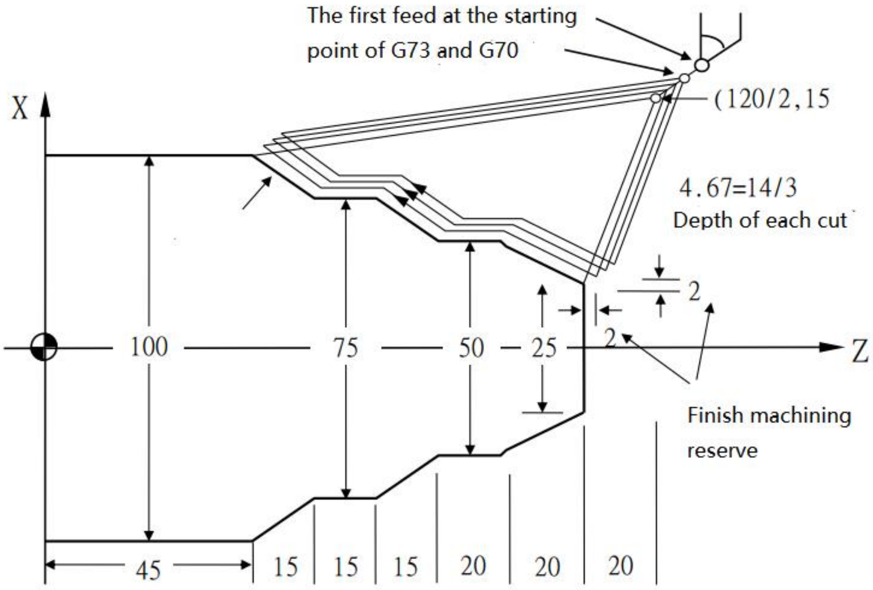


Figure 1.38-13

Diameter Programming

G28;

T0202;

M3 S3000;

G00 X120.000 Z150.000;

G73 U14.000 W14.000 R3;

G73 P100 Q200 U4.000 W2.000 F2.00;

N100 G00 X25.000 W-20.000;

G01 X50.000 W-20.000 F1.5;

W-20.000;

X75.000 W-15.000;

W-15.000;

N200 G01 X100.000 W-15.000;

G00 X130.000;

Z160.000;

T0303;

G00 X120.000;

Z150.000;

G70 P100 Q200;
M5;
M30;

1.38.5 Slotting Cycle (G74)

G74 is a fixed cycle used for slot machining on the end face of a bar material. It is defined by specifying the coordinates of the slot endpoint, the approach amount, the tool offset, and the tool retraction amount at the slot bottom.

1.38.5.1 Instruction Format

G74 R(e)__
G74 X(U)__Z(W)__P(i)__Q(k)__R(d)__F(f)__

- R(e): Return amount (specified in radius, modal instruction, unsigned). This can be set by Sys3010, and the parameter can be modified with Program Commands.
- X(U): X-Axis endpoint coordinates, where U is the incremental value relative to the starting point in the X direction.
- Z(W): Z-Axis endpoint coordinates, where W is the incremental value relative to the starting point in the Z direction.
- P(i): Tool offset (ignore sign, specified in radius, incremental value, without decimal point), in units of 0.001mm.
- Q(k): Cutting amount (ignore sign, specified in radius, incremental value, without decimal point), in units of 0.001mm.
- R(d): Tool offset at the slot bottom position (specified in radius).
- F: Feed rate.

1.38.5.2 Supplement Explanation

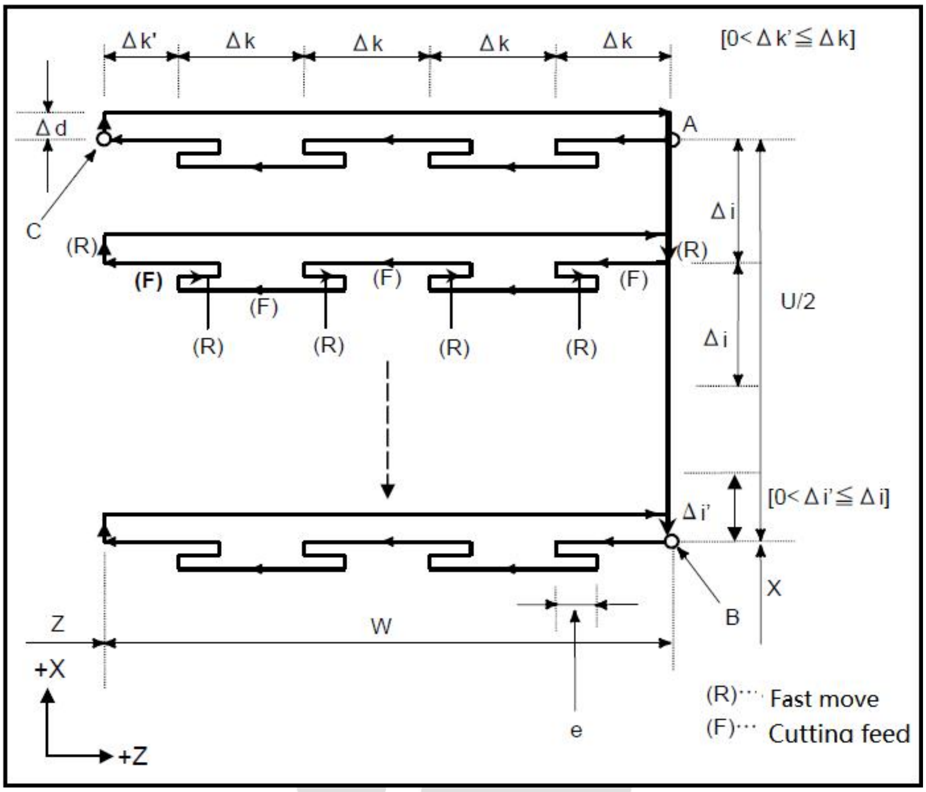


Figure 1.38-14

Action Explanation:

1. The bottom position offset (d) of the slot is typically specified as a positive value. When R(△d) is specified without a sign, the tool will offset on the first cut at the bottom. When R(△d) is specified with a sign, the tool won't offset on the first cut at the bottom, but it will start offsetting from the second cut onwards, and the direction of offsetting is not affected by the sign.
2. When the cutting amount is greater than the Z-axis increment, it performs one cut with the cutting amount equal to the Z-axis increment. If the offset of the other plane's axis is greater than its movement increment, it only moves once, and the offset amount is equal to that axis's increment.

Example - Part Figure

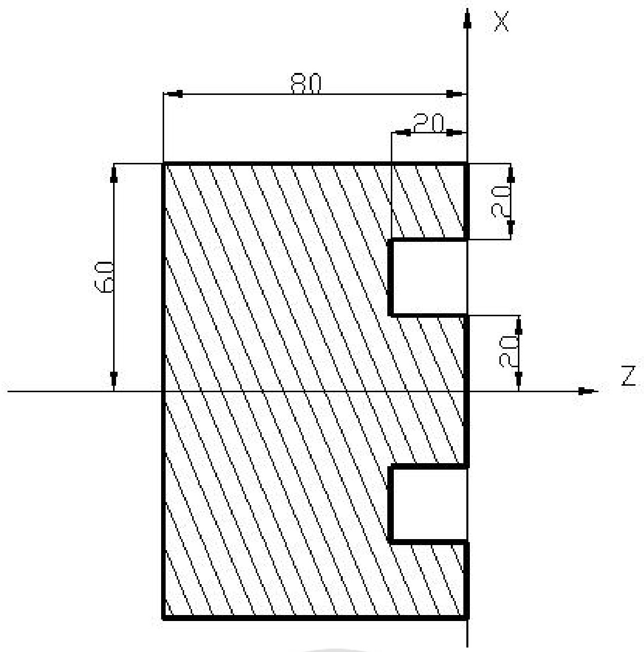


Figure 1.38-15

Program Example

G0 X100 Z50 ; (Rapid positioning)

T0101 ; (Tool change to tool #1, 2mm width)

M3 S500 ; (Start the main spindle, speed 500)

G0 X36 Z5 ; (Position to the machining start point)

G74 R1 ; (Specify Z-axis retract)

G74 X20 Z-20 P2000 Q3500 F50 ; (X-Axis moves 4mm per cycle, Z-Axis moves 3.5mm per cycle)

G0 Z50

X100

M5

M30

1.38.5.3 Related Alarms

74-1: The feed amount for the cutting axis is not specified or is set to 0.

74-3: The retract amount for the cutting axis is greater than the feed amount.

74-4: The feed amount for the other plane axis is not specified or is set to 0.

74-6: The retract amount for the other plane axis is greater than the feed amount.

74-7: Incorrect F value (≤0).

1.38.6 Longitudinal Slotting Cycle (G75)

G75 is used to automatically perform fixed-cycle slotting in the side direction of a bar material by specifying the end coordinates of the slot, the approach amount, the tool offset, and the tool's retract amount at the slot bottom.

1.38.6.1 Instruction Format

G75 R(e)__
G75 Z(W)__X(U)__P(i)__Q(k)__R(d)__F(f)__

- R(e): Retraction amount (specified as radius, modal command, unsigned). It can be set by Sys3010, and the parameters are modified with Program Commands.
- X(U): X-Axis endpoint coordinates. U is the incremental value relative to the starting point in the X direction.
- Z(W): Z-Axis endpoint coordinates. W is the incremental value relative to the starting point in the Z direction.
- P(i): Cutting amount (ignore the sign, specified as radius, incremental value, without decimal point), in units of 0.001mm.
- Q(k): Tool offset amount (ignore the sign, specified as radius, incremental value, without decimal point), in units of 0.001mm.
- R(d): Tool offset amount at the slot bottom position (specified as radius, reference the sign handling in the comments below).
- F: Feed rate.

1.38.6.2 Supplement Explanation

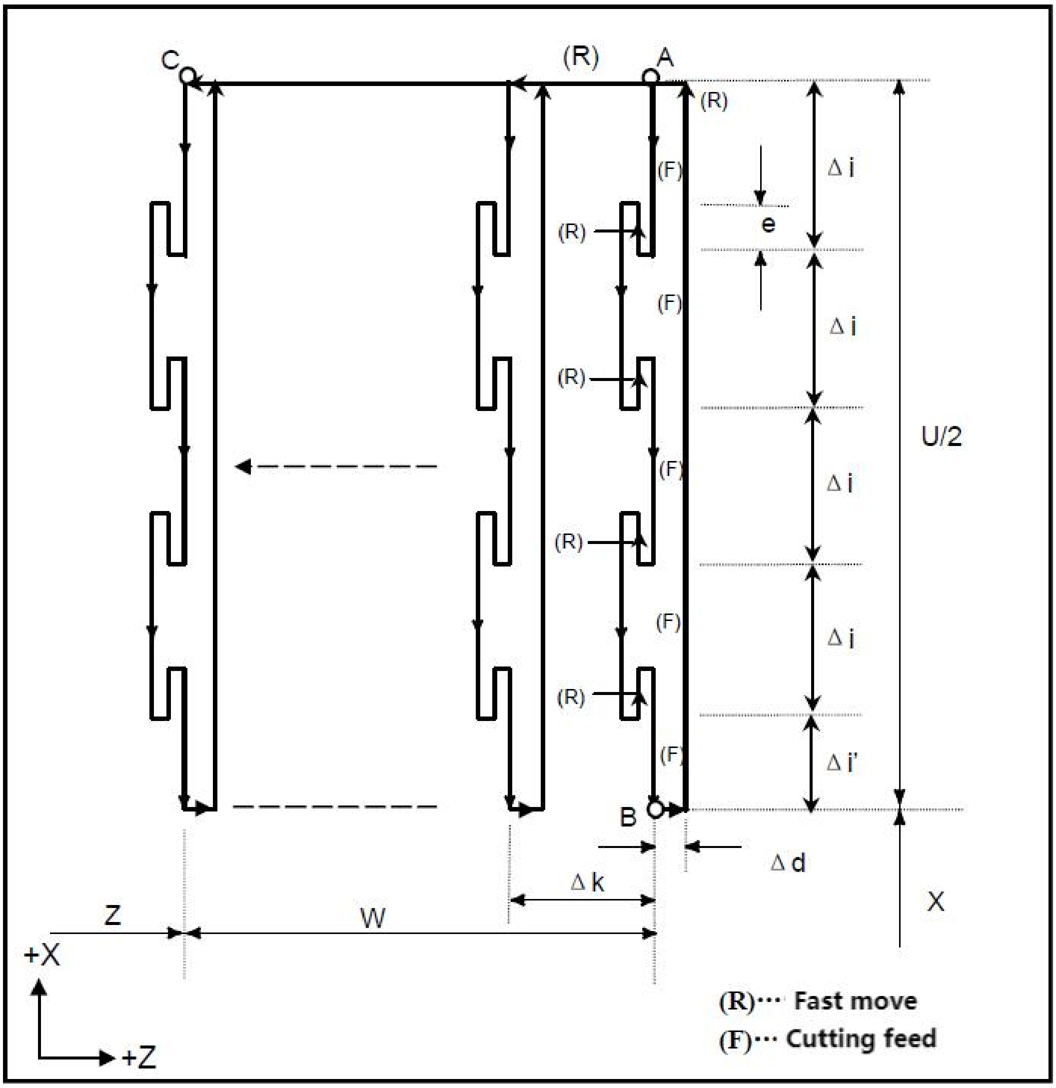


Figure 1.38-16

Action Explanation:

1. The offset at the slot bottom position (d) is typically specified as a positive value. When R(△d) is provided without a sign, the tool will offset on the first cut at the bottom. When R(△d) includes a sign, the tool will not offset on the first cut at the bottom; it will start offsetting from the second cut onwards, and the direction of offset is not influenced by the sign.
2. When the cutting amount is greater than the incremental value in the X direction, one cutting pass is executed, and the cutting amount corresponds to the X direction increment. If the offset in the other plane axis exceeds its movement increment, only one movement occurs, and the offset corresponds to that axis's increment.

1.38.6.3 Example

Diameter Programming

N10 G0 X80.0 Z0.0; // Rapidly move the tool to position X80.0 Z0.0 from the workpiece origin.

N20 M03 S2000; // Start the spindle in the forward direction at 2000rpm.

N30 G75 R2.0; // R2.0. The table indicates that after each drilling of 2500μm, retract 1000μm.

N40 G75 X60.0 Z3.0 P5000 Q5000 R1.0 F0.5; // Grooving cycle: "X60.0" indicates the absolute coordinate value of the end point of the drilling cycle in the X direction, "Z3.0" indicates the end coordinate value of the cycle movement in the Z direction, "P5000" indicates a drilling increment of 2500μm, "Q5000" indicates a cycle movement of 500μm in the Z direction, "R1.0" indicates a retract of 1000μm in the Z direction to the endpoint of cutting.

N50 M5; // Stop the spindle.

N60 M2; // End of the program.

1.38.6.4 Related Alarms

75-1: Cutting axis feedrate not specified or set to 0.

75-3: Cutting axis retract value greater than the feedrate.

75-4: Feedrate not specified or set to 0 for the other plane axis.

75-6: Retract value greater than feedrate for the other plane axis.

75-7: Incorrect F value (≤0).

1.38.7 Compound Threading Cycle (G76)

G76 is used to specify the starting and ending points for thread cutting, allowing cutting to occur at any angle and maintaining a constant cutting torque for each cycle. By considering the specified values for the thread endpoint coordinates and the taper angle, various longitudinal threads can be cut.

1.38.7.1 Instruction Format

G76 P(mra)__Q(△)__R(d)__

G76 X(U)__Z(W)__R(i)__P(k)__Q(△d)__F(l)__L__

Address		Meaning
P	m	Precision Machining's Number of Entries: 00-99 (modal), when set to 0, defaults to processing once.
	r	Chamfer Amount: 00-99 (modal), can also be set via parameters. Thread pitch 'l' serves as the reference. The range is from 0.001l to 9.9l, specified in integers.
	a	Tool Tip Angle (Thread Angle): 00-99 (modal), specified in 1° increments from 0° to 99°.
Q	△	Minimum Entry Amount (modal): When the calculated entry amount is less than this value, it is constrained to Δ [Δ allows setting the minimum cut amount (affected by Mcm1741, when Mcm1741 = 3, the minimum value is 1mm)]. Diameter specified, ignore the sign, can include a decimal point.
R	d	Precision Machining Allowance (modal, Diameter specified): Ignore the sign, can include a decimal point.
X(U)		X-Axis Endpoint Coordinates of the Threaded Portion: Specified as either absolute or incremental values. U is set as the difference in the X-direction from the thread endpoint to the thread start.
Z(W)		Z-Axis Endpoint Coordinates of the Threaded Portion: Specified as either absolute or incremental values. W is set as the difference in the Z-direction from the thread endpoint to the thread start.
R	i	Taper Height of the Threaded Portion (Diameter specified): 'i' can be positive or negative.
P	k	Thread Height: Specifies the thread height with a positive Diameter value (Diameter specified, can include a decimal point).
Q	△d	Entry Amount: Specifies the first entry amount with a positive Diameter value (Diameter specified, can include a decimal point).

F	I	Thread Lead (the axis with the maximum incremental length).
L		Number of Thread Starts: Ranges from 1 to 99.

1.38.7.2 Example

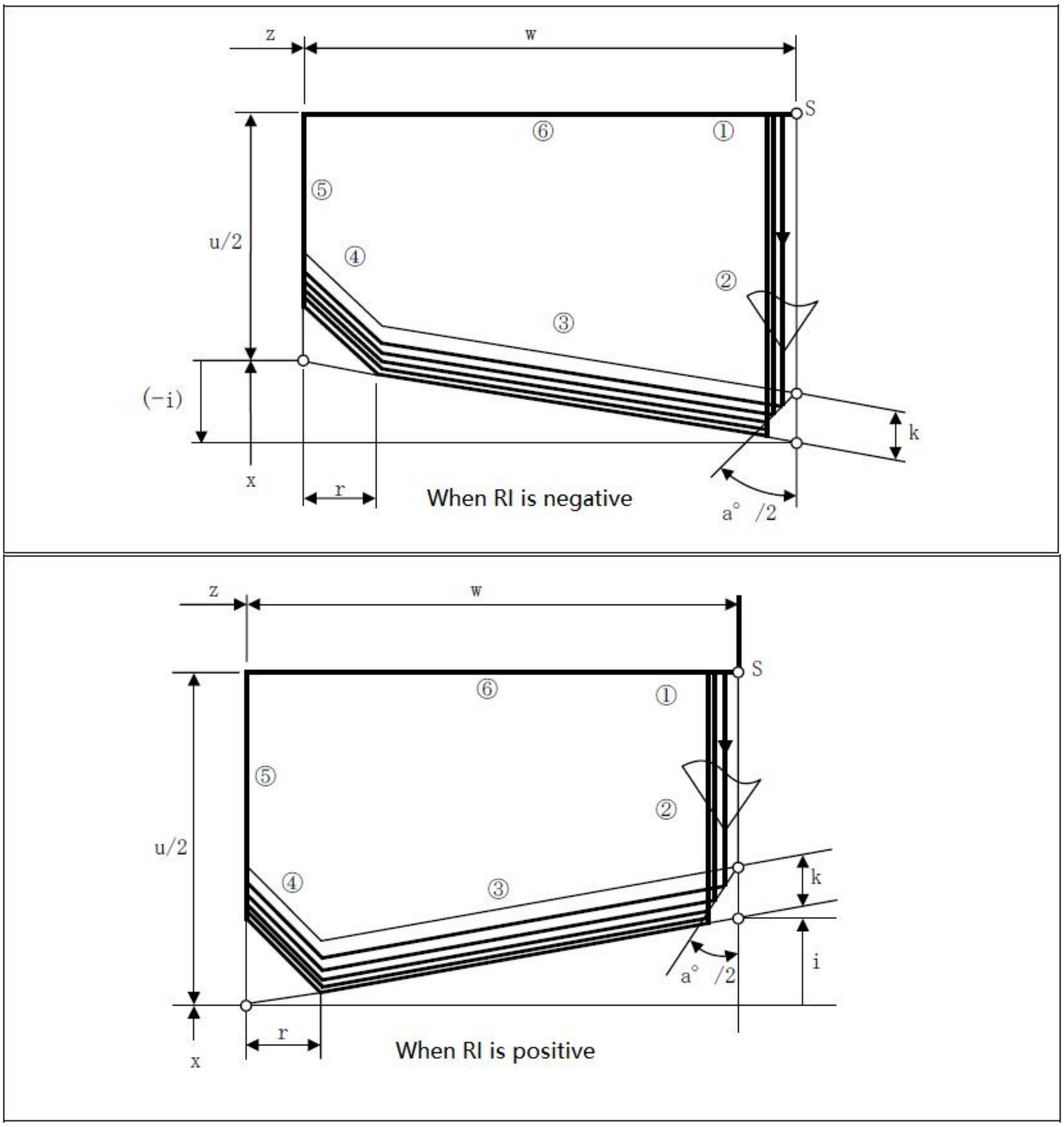


Figure 1.38-17

In one cycle, ①, ②, ⑤, and ⑥ are for rapid feed, while ③ and ④ move according to the cutting feed specified by F.

1.38.7.3 Notes

1. The two G76 commands mentioned above cannot be combined into one block. The automatic determination is made based on the data provided by P, Q, and R commands, considering the presence of axis addresses like X, Z, etc.
2. The specified lead of the long axis is used as the lead of the tapered thread.
3. In the first G76 command, the P command cannot have a decimal point, while the Q command can have a decimal point. In the second G76 command, both P and Q commands can have decimal points.
4. P and Q commands in the second G76 command are specified as positive values.
5. In the first G76 command, P__ specifies m, r, and a as follows:
    - ❖ m: Set to 1-2 digits, specifying the number of precision passes (m).
    - ❖ r: Set to 3-4 digits, specifying the chamfer amount (r).
    - ❖ a: Set to 5-6 digits, specifying the tool tip angle (a).
6. The threading section is divided based on the number of precision passes, and the feed amount for each pass is d/m.
7. Parameters such as P(mra), Q(Δ), R(d), etc., in the G76 command can be specified by parameters that may change with Program Commands. When not specified in the G76 command, the parameter values are used.
8. When both the X and Z-axis increments are 0 in the second G76 command, it is processed like the first G76 command. Parameters are set, but no actual machining is performed.
9. When Δ is set greater than Δd, the first cut is made with Δ cutting amount. When Δ is greater than k, only one roughing cycle is performed, with roughing cut amount = k - d.
10. The F value is specified based on the lead of the axis with the maximum incremental length, as shown in the figure. For the chamfered portion, the F value depends on the current thread axis. If the thread axis is Z-Axis, the Z-Axis speed for the chamfered portion is fixed at the F value specified in the program, and the X-Axis is automatically calculated based on the angle.

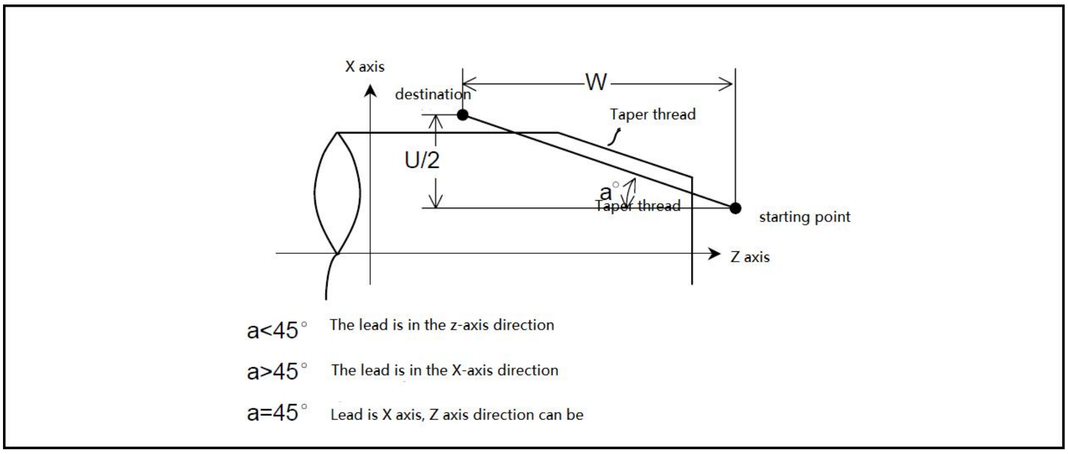


Figure 1.38-18

11. R(i) Thread Taper Height Setting: This value can be positive or negative. R(i) is set as the difference in the X-Axis direction between the start point and end point of the threaded section, as specified below.
    ❖ R value = Coordinate value in the X-Axis direction at the start of the G76 thread cycle - Coordinate value in the X-Axis direction at the end of the G76 thread cycle.
    ❖ U value = Coordinate value in the X-Axis direction at the end of the G76 thread cycle - Coordinate value at the start of the G76 cycle in the X-Axis direction.

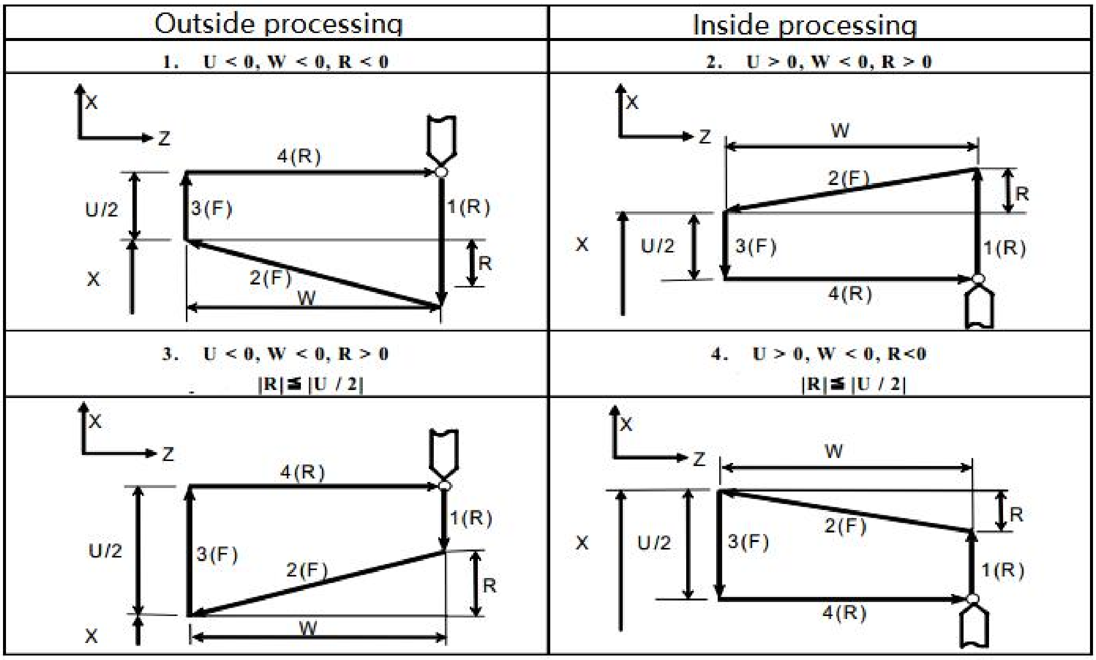


Figure 1.38-19

1. In the G76 command, the G00 factor, G01 factor, and spindle factor in the thread cutting section are fixed at 100%. They automatically revert to their state before the G76 command (MFO) after the cycle is completed.

2. In the G76 command, if the first entry amount is greater than the thread height, it will only execute one rough cutting pass, with the cutting amount being equal to k (thread height) - d (finish allowance).
3. In the G76 command, when Sys3000 (setting for the distance from the end point where the G76 thread retracts and chamfers in the direction of the spindle) is set to a value greater than 99, it doesn't produce an error. However, when the G76 thread tail length P(r) is greater than or equal to the Z-axis increment, it will produce an ERR1.7.2 warning (Sys3000 specifying range should be within 99 or below).
4. In the G76 command, when the taper angle is greater than the cutting direction increment, the system does not produce an error.
5. The finish allowance can be set to 0 in the G76 command. When set to 0, no finish machining is executed.

1.38.7.4 Example

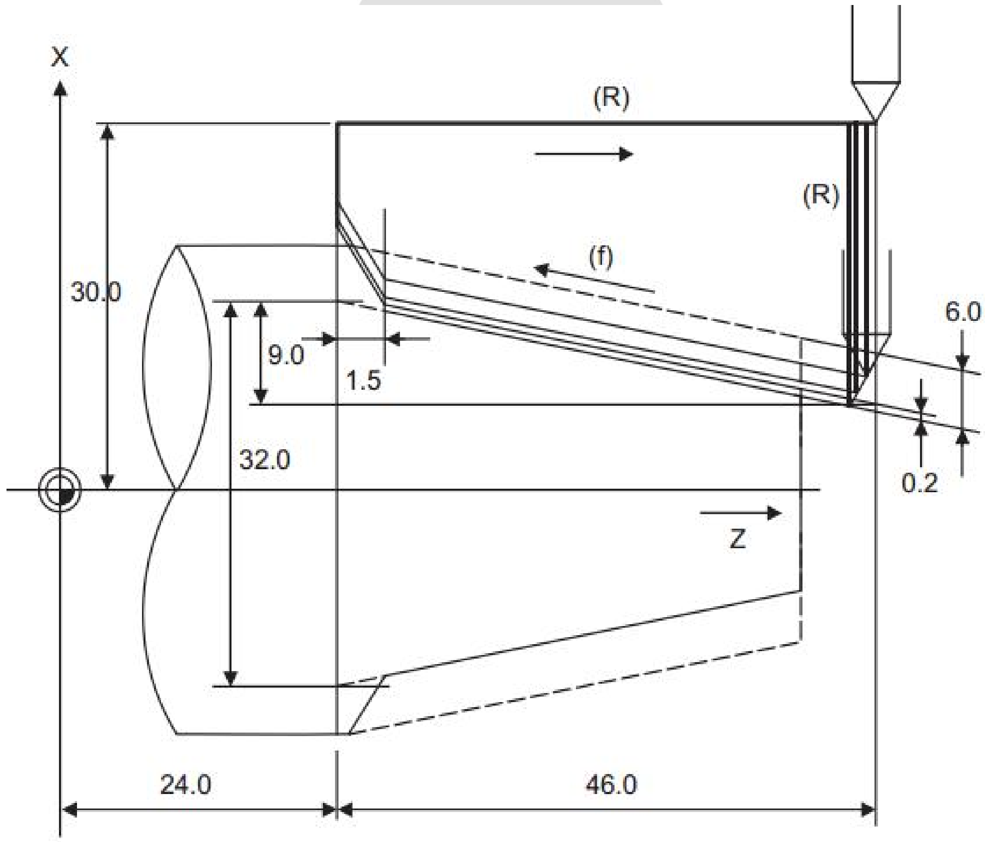


Figure 1.38-20

Diameter Programming

G0 X0 Z0;

G99;	
M03 S100;	
G00 X60.0 Z60.0;	
G76 P011560 Q0.1 R0.2 ;	//Precision machining once, thread retraction value of 15mm, tool angle 60 degrees, minimum cutting amount 0.1mm, precision turning allowance 0.2mm.;
G76 U-28.0 W-46.0 P6.0 Q3.5 F2. R-9.0 L2;	//The increment in the distance between the starting point and the cutting endpoint in the X-Axis direction is -14mm, and the increment in the distance between the starting point and the cutting endpoint in the Z-Axis direction is -46mm. The thread height is 3mm, the first cutting depth is 1.25mm, the thread pitch is 2mm, the difference between the thread start and end points in the X-Axis direction is 9mm, and there are 2 threads.;
G0 X80.0;	
Z72.0;	
M30;	

Explanation:

Thread Retract Distance (in mm) = Retract Setting (r) × 0.1 × Thread Pitch (F)

1.39 Face/Side Drilling Cycles (G83/G87)

1.39.1 Instruction Format

G83 X(U)__C(H)__Z(W)__R(r)__Q(q)__P(p)__F(f)__K(k)__M(m)__I(i)__;

Or

G87 Z(W)__C(H)__X(U)__R(r)__Q(q)__P(p)__F(f)__K(k)__M(m)__I(i)__;

- G83: Face Deep Hole Drilling Cycle (Modal Command).

- G87: Side Deep Hole Drilling Cycle (Modal Command).
- X(U): Specify the initial point of the loop (absolute/incremental value) (specify the position of the hole bottom).
- Z(W): Specify the initial point of the loop (absolute/incremental value) (specify the position of the hole bottom).
- R(r): Initial plane to R distance. (Normal format: Incremental programming, radius specified, ignore sign; Special format: Can specify half-Diameter, incremental programming, modal command).
- Q(q): Specifies the depth of each peck, with incremental radius value, in millimeters.
- P(p): Specifies the dwell time at the bottom of the hole, equivalent to G04 P__. (Modal) Unit: milliseconds.
- F(f): Specifies the drilling feed rate, a modal command.
- K(k): Specifies the number of repetitions; if not set, it defaults to 1. Not a modal command (parameter can be set to k=0 for system handling).
- M(m): M-code for C-axis clamping. Not a modal command.
- I(i): Specifies the drilling axis in the special format. When I is specified, all other axes are positioning axes (only effective when the special format is set).

1.39.2 Notes

1. The instruction for the hole's initial position is non-modal. When issuing consecutive G83 (G87) commands, you must specify the initial position for each program segment.
2. If the specified Q value is greater than the total cutting feed, it will perform a regular drilling fixed cycle (i.e., cutting feed continues to the bottom of the hole, and the tool retracts rapidly from the hole bottom).
3. When K is specified, the M-code is only executed when positioning to the initial plane for the first time; it is not executed during repeats.
4. There is a check for positioning at the hole bottom and when returning to the R-point or starting point.
5. Q(q) is a non-modal command. If Q is not specified in programming, it will result in a single plunge.

1.39.3 Regular Drilling Fixed Cycle

If the cutting depth Q is not specified, it will execute a regular drilling cycle, with cutting feed to the hole bottom, followed by a dwell of P1 second, and then the tool rapidly retracts from the hole bottom.

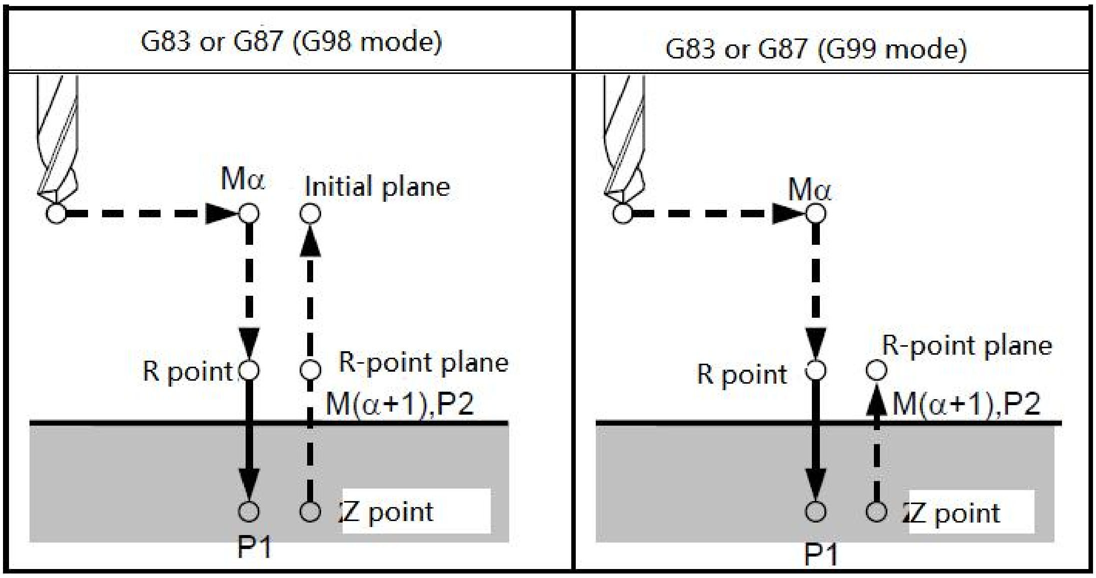


Figure 1.39-1

Mα: M-code for C-axis brake.

M(α+1): M-code for releasing the C-axis brake.

P1: Pause specified by the program.

P2: Pause time set by parameters (to ensure positioning check and spindle brake release are completed).

d: Parameter set for retract distance.

Note: Regarding the retract point, in a lathe, the final retraction is to the starting point; in a milling machine, it's controlled using G98 and G99.

1.39.4 Deep Hole Drilling Cycle

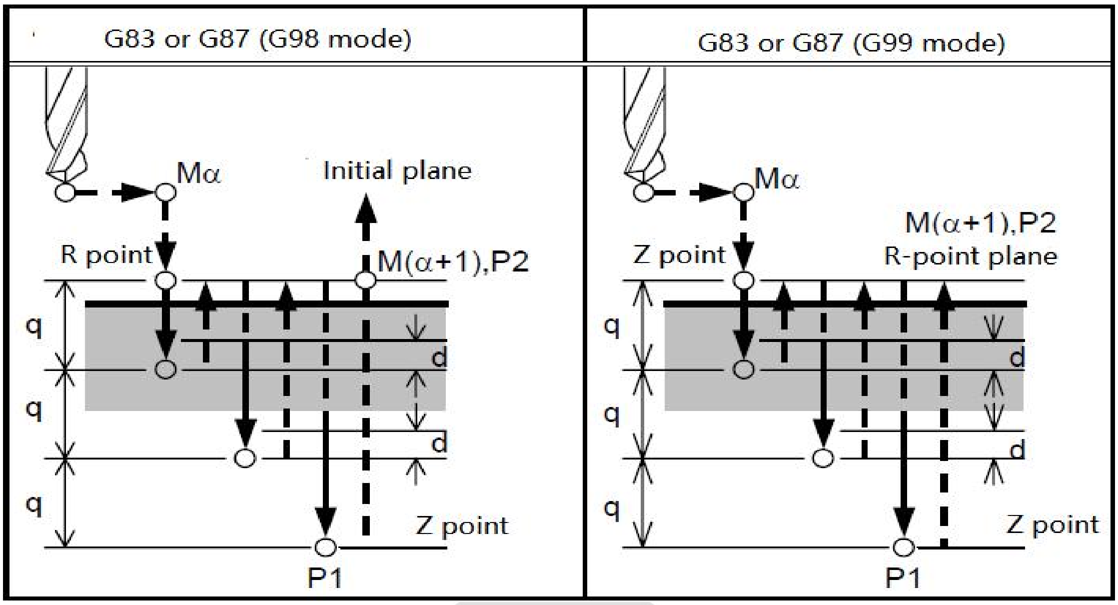


Figure 1.39-2

Mα: M-code for engaging the C-axis brake.

M(α+1): M-code for disengaging the C-axis brake.

P1: Pause specified by the program.

P2: Pause time set by parameters (to ensure positioning check and spindle brake release are completed).

d: Parameter-set retract distance.

Note: Regarding the retract point, in a lathe, the final retraction is to the starting point; in a milling machine, it's controlled using G98 and G99.

1.39.5 High-Speed Deep Hole Drilling Cycle

This cycle is used for high-speed deep hole drilling operations. It repeats the following process: intermittently cutting feed and rapidly retracting by a specified amount before reaching the hole bottom, all while evacuating chips from the hole during the machining process.

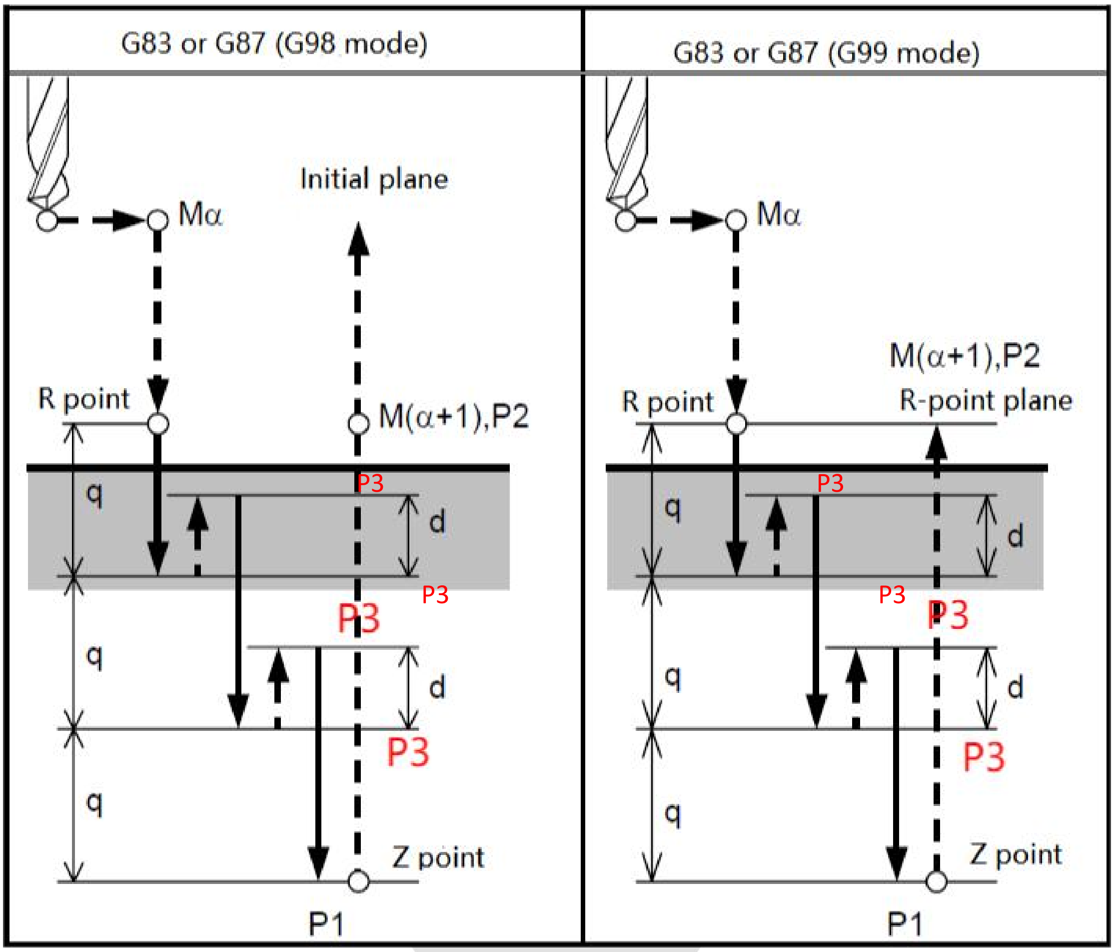


Figure 1.39-3

Mα: M-code for engaging the C-axis brake.

M(α+1): M-code for disengaging the C-axis brake.

P1: Pause specified by the program.

P2: Pause time set by parameters (to ensure positioning check and spindle brake release are completed).

d: Parameter-set retract distance.

P3: Pause time for each feed (used in special formats).

Note: Regarding the retract point, in a lathe, the final retraction is to the starting point; in a milling machine, it's controlled using G98 and G99.

Ordinary Drilling Cycle Mode Example：

G98

G0 X0 Z10 ; Rapid positioning to the starting point on the X and Z axes.

M03 S500 ; Spindle starts to rotate at 500 RPM.

G83 Z-60 R10 Q2000 P1000 F100	; G83 initiates a peck drilling cycle. Starting point is X0 Z10, hole bottom is at X0 Z-60, rapid positioning to Z0 (R) position, each cut is 2 mm deep, retracting to the R point after each cut, pause at the hole bottom for 1 second, cutting feed rate at F100.
G80	; Cancel fixed cycle (G80).
M30	; End of the program.

1.40 Face/Side Threading Cycles

The G84/G88 commands are threading cycles used for face and side threading operations on CNC lathes. These cycles are designed for threading work on a workpiece that is clamped to the spindle and cannot rotate. In this process, a rotating tool is used to perform threading on the workpiece, which remains fixed and cannot rotate.

- **Types of Threading Cycles**

Threading cycles can be classified into two types based on their working principles: flexible threading and rigid threading. These cycles can be further categorized into high-speed deep-hole threading cycles or deep-hole threading cycles.

- **Explanation**

Flexible Threading:

In flexible threading, the movement along the threading axis is synchronized with the spindle rotation, whether the spindle is rotating or stopped. Threading is performed using auxiliary functions (M03, M04, M05). When the spindle slows down at the bottom of the hole, there is an associated telescopic motion of the threading axis. Therefore, it's necessary to use a variable thread tap in this mode. Using a standard thread tap in this scenario can result in the thread tap breaking.

Rigid Threading:

In rigid threading, the control of the spindle motor operates similarly to a servo motor (position control). Threading is achieved through the coordination of the spindle motor and interpolation between the threading axis and the spindle. In this mode, for every revolution of the spindle, the threading axis advances a specific distance (pitch). Rigid threading doesn't require the use of a variable thread tap (spring-loaded tap).

1.40.1 Notes

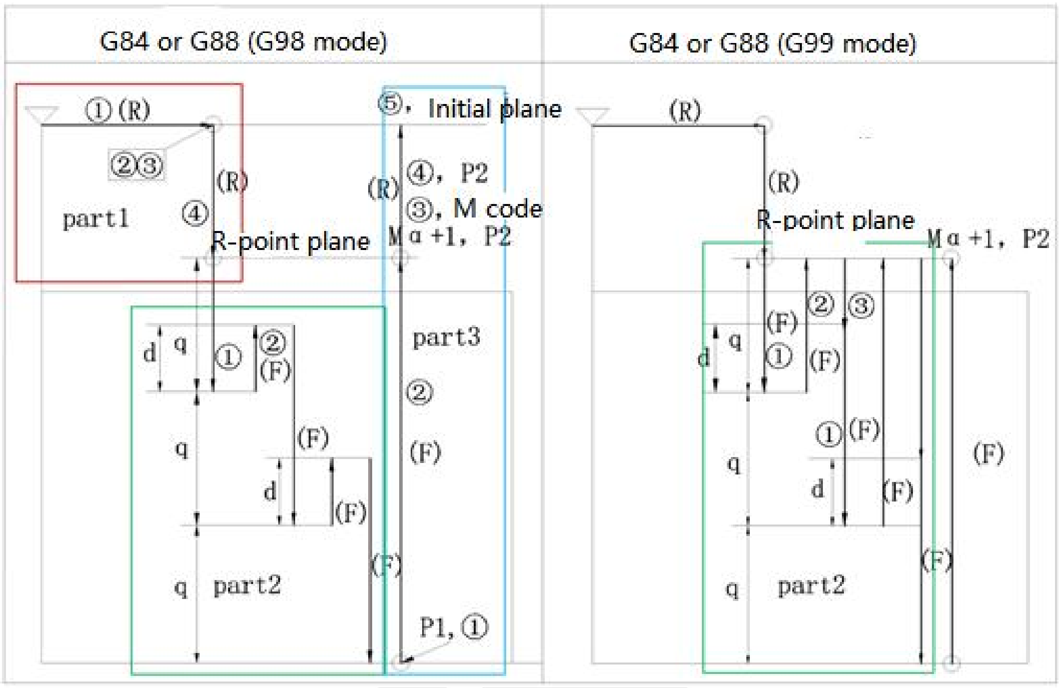


Figure 1.40-1

1. **Approach Phase**
   1) Position to the drilling start point (initial plane) (G00).
   2) Position the C-axis to the specified angle, skip this step if no C-value is specified.
   3) Execute Mα, skip this step if Mα is not specified.
   4) Position to the safe plane R-point (G00).

2. **Threading Preparation**

   When Q is not equal to 0:

   1) Feed to [n×q＋R], if [n×q＋R] exceeds the endpoint Z, then feed to Z and end this part of the process.

   2) In high-speed deep hole threading cycle mode, retract to distance d from the current point using G01; in deep hole threading cycle mode, retract to the R-point.

   3) Feed to [n×q＋R－d] position using G01, this step exists only in deep hole threading cycle mode. When Q is 0, feed to Z and end this part of the process.

3. **Retraction Preparation**

1) Pause with G04 P1.
2) Feed to the R-point using G01.
3) If Mα is specified, execute M(α+1); otherwise, skip this step.
4) Pause with G04 P2.
5) In G98 mode, retract to the initial plane (for milling machine mode).

Note:

1. Program restart function cannot be executed during the threading cycle.
2. The instruction for the hole's initial position is non-modal. When consecutive G84 (G88) instructions are used, specify the hole's initial position in each Program segment.
3. When K is reached, M-code is executed only during the first positioning to the initial plane. It's not executed during repetitions.
4. Specify the spindle number and spindle speed before the G84/G88 instructions.
5. The spindle's clockwise and counterclockwise rotation M-codes in the threading cycle are specified through parameters.
6. In milling machine mode, G98 returns to the initial plane, and G99 returns to the R-point plane. In lathe mode, it returns to the initial plane.

1.40.2 Instruction Format

G84 X/U_C/H_Z/W_Rr_Qq_Pp_Ff_Kk_Mm_Jj_Dd

G88 Z/W_C/H_X/U_Rr_Pp_Qq_Ff_Kk_Mm_Jj_Dd

G84(G88)	G84: Face Deep-Hole Threading Cycle; G88: Longitudinal Deep-Hole Threading Cycle;
X/U(Z/W) _C/H_	Specify the axis position (absolute/incremental) of the tapping plane; The angle at which the workpiece is positioned on the C-axis;
Z/W(X/U)_	Depth position (absolute value/incremental from R-point). If specified as Z_, it represents the absolute coordinate from the start point to the hole's bottom position. If specified as W_, it represents the incremental value from the R-point to the hole's bottom position;
Rr_	Specification of the R-point (modal, specified as an incremental radius value). If R-value is not set, the R-point plane coincides with the initial plane;

Qq_	Specification of each cut (modal, specified as an incremental radius value, ignore the sign);
Pp_	Specification of the pause time at the hole's bottom, equivalent to G04 P_ (modal), unit: ms;
Ff_	Specify the feed speed (mode); Pitch of metric teeth; Ee_ is the pitch of English teeth.
Kk_	Specification of the repetition count, 0 to 9999 (default is 1) (non-modal);
Mm_	Specification of the spindle brake command (non-modal, parameter planning, can flexibly set M-code);
Dd	Specification of the spindle used in the threading cycle (modal) (set range: 1 to the number of spindles).

1.40.3 Threading Cycle Specification Methods

1.40.3.1 Rigid Threading

1. Specify the rigid threading with the M-code MxxSxxxx:
   1) The M-code is specified by Sys3034 (when Sys3034 = 0, it defaults to M29).
   2) The threading M-code can be set before threading or in the same Program segment as the threading instruction.
   ❖ M29 specifies rigid threading, canceling rigid threading when encountering G80 or G01. To specify rigid threading again, you need to specify M29 again.
2. Set Sys3031 BIT00=1 to define G84/G88 as rigid threading G-codes.
3. Specify the rigid threading's instruction format (along with parameter settings):
   1) Specify M29 before the G84/G88 threading Program segment:M29;
      G84 X_C_R_P_F_K_(M_);
      X_C_;
      G80;
   2) Specify it within the G84/G88 threading Program segment; in this case, you cannot specify the M-code for clamping the C-axis within the G84/G88 Program segment:
      G84 X_C_Z_R_P_F_K_M29 S_;
      X_C_;
      G80;
   3) Use G84/G88 as rigid threading G-codes (determined by Sys BIT00); in this case,

G84/G88 can only be used for rigid threading and not for regular threading.

```
G84 X_C_Z_R_P_F_K_M_;
X_C_;
G80;
```

Note: You cannot specify S or axis movement commands between M29 and the G84/G88 Program segment; otherwise, an alarm will occur.

1.40.4 Flexible Threading

Not specifying rigid threading means flexible threading.

1.40.4.1 Regular Flexible Threading Cycle

In flexible threading, when Q (cutting increment per cycle) is not specified in Program Commands, it is a regular flexible threading cycle. Regular threading mode is defined as threading where the threading axis threads from the initial plane or R-point plane to the bottom of the hole in one pass.

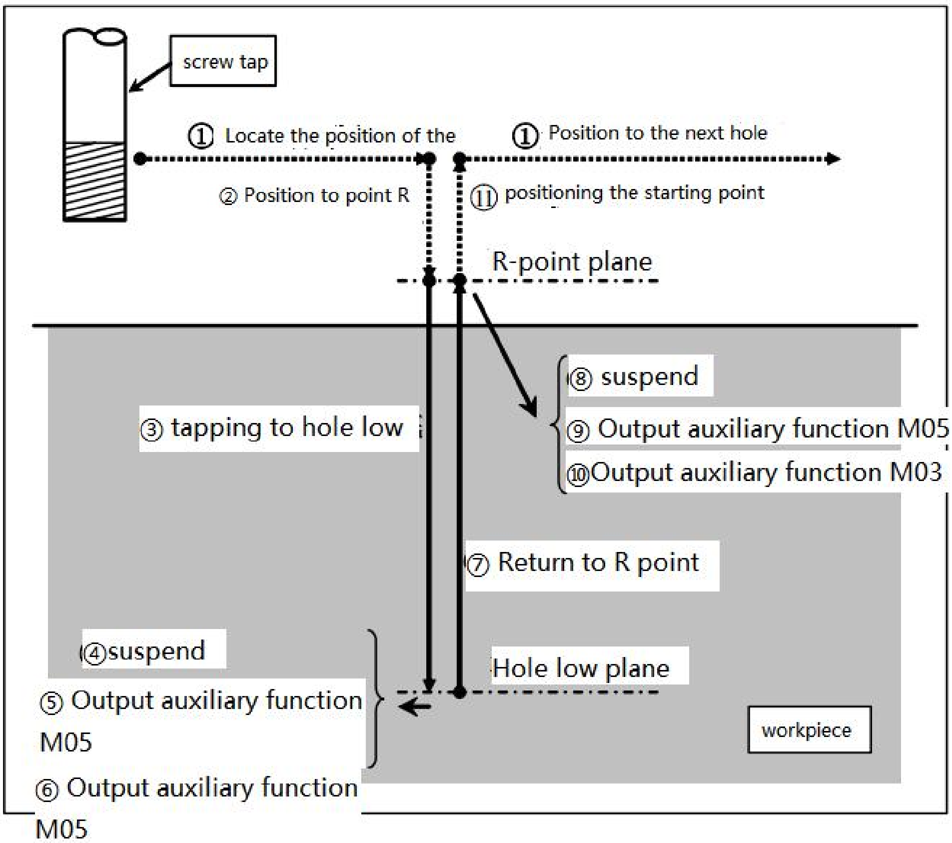


Figure 1.40-3

Action commands are as follows:

1. Flexible Threading per Revolution

Before specifying the threading command, follow these steps:

1) Position to the hole's starting point (initial plane) (G00).

2) When there is a spindle brake command (Mα), output the spindle brake (no need to output the spindle brake command when the current rotating spindle is the first spindle).

3) If the R-point is specified, perform positioning to the R-point (G00).

4) Wait for spindle synchronization and then thread the material in a cutting feed manner, using the spindle feedback until reaching the hole bottom. Before reaching the hole bottom, output the spindle stop command (M05).

5) If a pause time is specified (P1), pause.

6) Output the auxiliary function M04 for spindle reversal and wait for the M-code completion signal.

7) When the M-code completion signal returns, retract the tap in a cutting feed manner to the R-point (G01).

8) If a pause time is specified (P2), pause.

9) Output the auxiliary function M05 for spindle stop and wait for the M-code completion signal.

10) When the M-code completion signal returns, output the spindle forward signal M03 and wait for the M-code completion signal.

11) When the M-code completion signal returns, return to the starting point in a rapid movement (G00) (this action is executed only in G98 mode).

12) When there is a spindle brake command, release the spindle brake (Mα+1). If the repetition count is specified, repeat from step (1).

2. Flexible Threading per Minute

Before specifying the threading command, you don't need to specify spindle rotation (specify it according to the need, either the first spindle or other spindles).

1) Position to the hole's starting point (initial plane) (G00).

2) If there is a spindle brake command (Mα), output the spindle brake (no need to output the spindle brake command when the current rotating spindle is the first spindle).

3) If the R-point is specified, perform positioning to the R-point (G00).

4) Rotate the spindle and wait for synchronization. The threading axis will thread in a cutting feed manner based on spindle feedback until reaching the hole bottom. Before reaching the hole bottom, output the spindle stop command (M05).

5) If a pause time is specified (P1), pause.

6) Output the auxiliary function M04 for spindle reversal and wait for the M-code completion signal.

7) When the M-code completion signal returns, retract the tap in a cutting feed manner until reaching the R-point (G01).

8) If a pause time is specified (P2), pause.

9) Output the auxiliary function M05 for spindle stop and wait for the M-code completion signal.

10) When the M-code completion signal returns, output the spindle forward signal M03 and wait for the M-code completion signal.

11) When the M-code completion signal returns, return to the starting point in a rapid movement (G00) (this action is executed only in G98 mode).

12) If there is a spindle brake command, release the spindle brake (Mα+1). If the repetition count is specified, repeat from step (1).

1.40.4.2 Deep Hole Flexible Threading Cycle

When the Q command (cutting increment per cycle) is specified, and the threading method is set to Sys3031 BIT02=1, it becomes a deep hole threading cycle. In this cycle, the tool retracts to the R-point after each pass. (Parameters are set in the threading parameters page under the threading method settings).

Note: The threading method settings parameters are found on the threading parameters page.

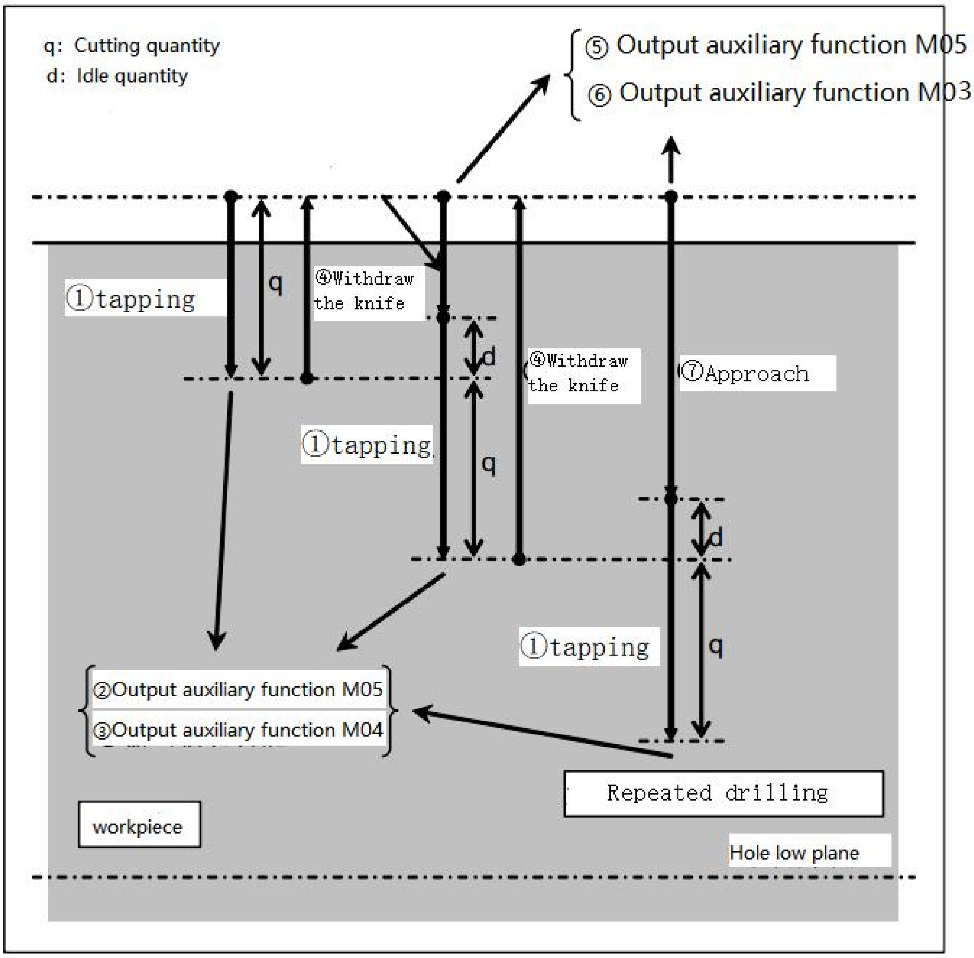


Figure 1.40-4

Action commands are as follows:

Action Commands for the Flexible Threading Cycle mentioned in Action ④ above are as follows. Other actions are the same.

1. Only cut the cutting amount q specified by Q until reaching q, then output spindle stop (M05).

2. When the M-code completion signal returns, output spindle reverse (M04) signal and wait for the M-code completion signal.
3. When the M-code completion signal returns, the spindle reverses, and the threading axis retracts to the R-point using cutting feed.
4. Output spindle stop (M05) signal before reaching the R-point and wait for the M-code completion signal.
5. When the M-code completion signal returns, output spindle forward (M03) signal and wait for the M-code completion signal.
6. When the M-code completion signal returns, the spindle reverses and waits for the synchronization point signal. The threading axis moves from the previous cutting point to a position just before d (clearance range, parameter setting) using cutting feed.
7. Cut the clearance range d + cutting amount q, and repeat the above steps for threading until reaching the hole bottom.

1.40.4.3 High-Speed Deep Hole Flexible Threading Cycle

When the Q command (cutting amount per cycle) is specified, and Sys3031 BIT02=0 (threading method setting), it becomes a high-speed deep hole threading cycle. In this cycle, the tool retracts to the d point (Sys3033) after each pass.

Note: The threading method settings parameters are located on the threading parameters page.

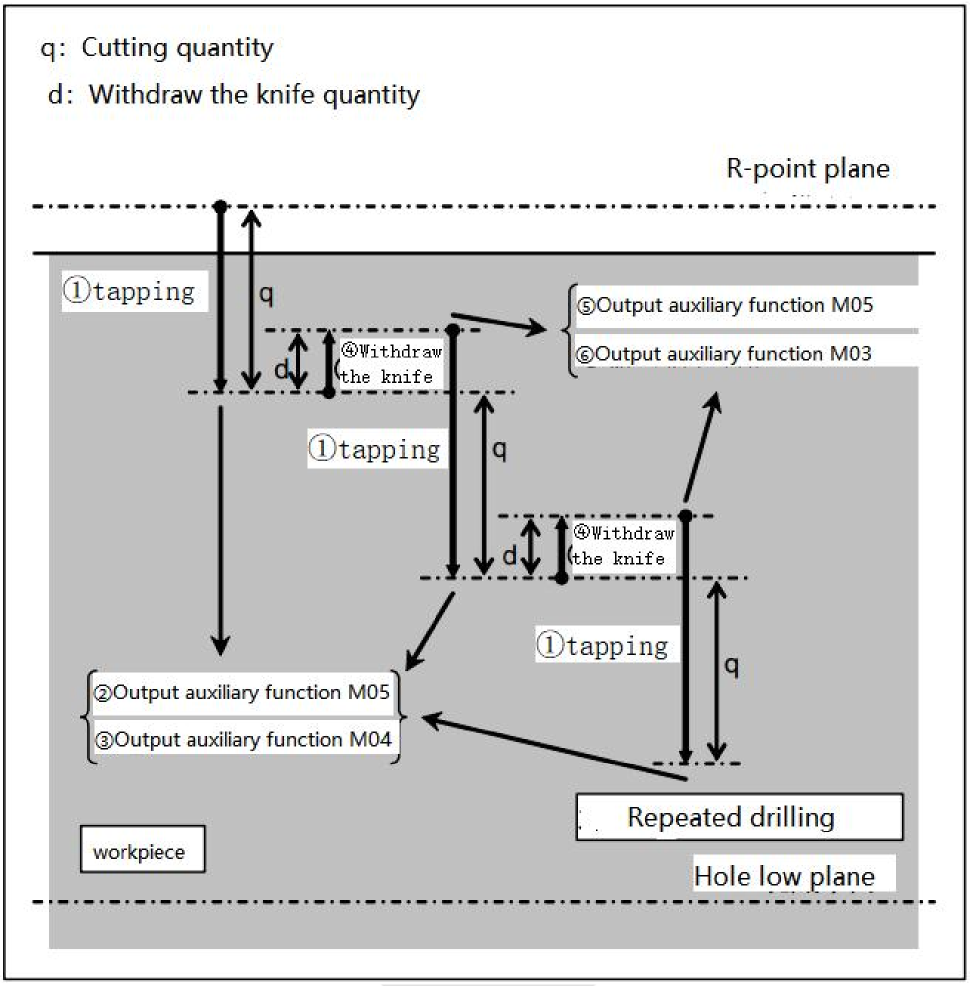


Figure 1.40-4

Action commands are as follows:

The actions for the flexible threading cycle mentioned in Action ④ above are as follows, with all other actions remaining the same:

1. Only cut the cutting amount q specified by Q until reaching q, then output spindle stop (M05).
2. When the M-code completion signal returns, output spindle reverse (M04) signal and wait for the M-code completion signal.
3. When the M-code completion signal returns, the spindle reverses, and the threading axis retracts to the d point using cutting feed.
4. Output spindle stop (M05) signal before reaching the d point and wait for the M-code completion signal.
5. When the M-code completion signal returns, output spindle forward (M03) signal and wait for the M-code completion signal.
6. When the M-code completion signal returns, the spindle reverses and waits for the synchronization point signal. The threading axis cuts by feeding in the cutting retreat

amount d + cutting amount q using cutting feed, repeating these actions until reaching the hole bottom.

1.40.5 Rigid Threading

1.40.5.1 Standard Normal Rigid Threading

No Q value is specified, and the threading is completed in a single pass.

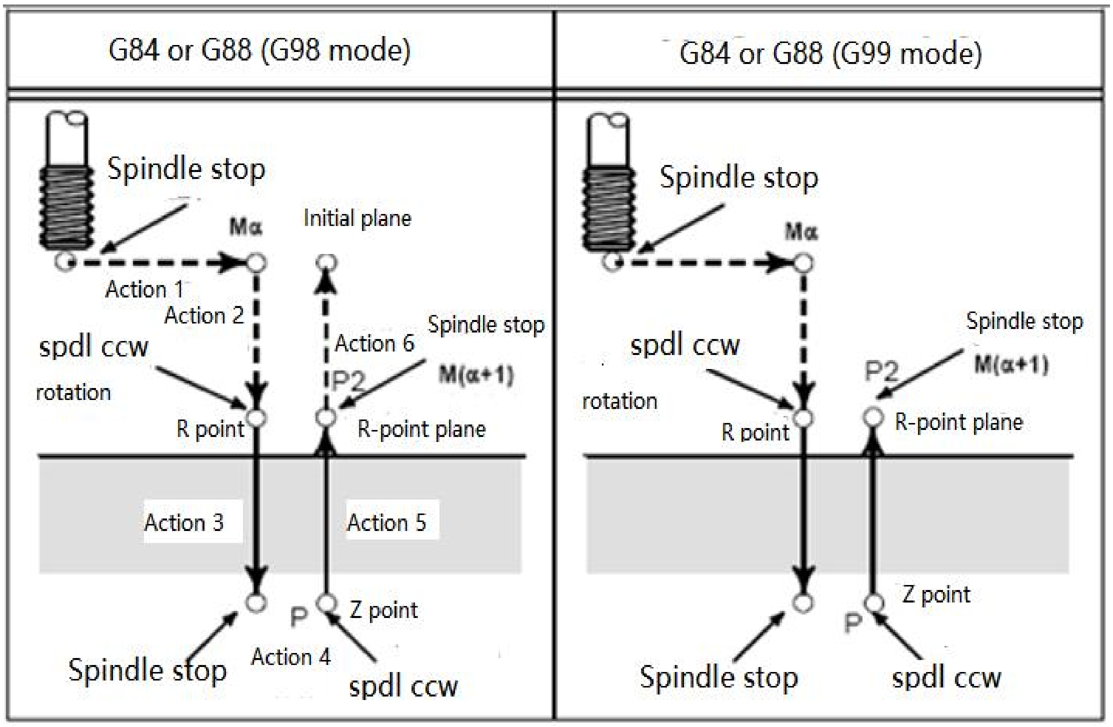


Figure 1.40-5

Program Example:

Thread M10×1.5

G99 ; Infeed per revolution

G0 X0 Z10 ; Position X and Z axes to the starting point

M29 S1000 ; Specify rigid tapping, spindle speed 1000 RPM

G84 Z-30 R-5 P2000 F1.5 ; G84 for rigid thread milling cycle, starting at X0 Z10, R plane at Z5, hole bottom at Z-30, hole bottom dwell time 2 seconds, thread pitch 1.5.

G80 ; Cancel fixed cycle

M30 ; End of Program

G98 ; Infeed per minute
G0 X0 Z10 ; Position X and Z axes to the starting point
M29 S500 ; Specify rigid tapping, spindle speed 500 RPM
G84 Z-40 R-4 P2000 F1.5 ; G84 for rigid thread milling cycle, starting at X0 Z10, R plane at Z6, hole bottom at Z-40, hole bottom dwell time 2 seconds, thread pitch 1.5.
G80 ; Cancel fixed cycle
M30 ; End of Program

Action commands are as follows:

1. Rigid Thread Cutting Per Revolution - Rigid Thread Following
   1) Stop the spindle.
   2) If there is a positioning command to the hole position, perform positioning action (G00); otherwise, skip.
   3) When a spindle brake command (Mα) exists, output the spindle brake (no need to output spindle brake command when the Master spindle is the first spindle).
   4) In case R-point is specified, execute positioning action towards the R-point (G00).
   5) Start the spindle, and the thread cutting axis performs thread cutting based on spindle feedback using a cutting feed until reaching the hole bottom. The spindle and thread axis stop simultaneously at the hole bottom.
   6) If a pause time (P1) is specified, pause.
   7) Output the spindle reversal auxiliary function M04 and wait for the completion of M-code.
   8) Upon the return of the M-code completion signal, retract the tap using a cutting feed until reaching the R-point (G01). The spindle and thread axis stop at the R-point.
   9) If a spindle brake command exists, output the spindle brake release command (Mα+1).
   10) If a pause time (P2) is specified, pause.
   11) Return to the starting point using rapid movement (G00) (only executed in G98 mode). If a repeat count is specified, repeat from (1).
   12) Restore the spindle to its state before thread cutting.

2. Rigid Thread Cutting Per Minute - Rigid Thread Following
   1) Stop the spindle.
   2) If there is a positioning command to the hole position, perform positioning action (G00); otherwise, skip.
   3) When a spindle brake command (Mα) exists, output the spindle brake command

(no need to output spindle brake command when the current rotating spindle is the first spindle).

4) In case R-point is specified, execute positioning action towards the R-point (G00).
5) Start the spindle, and the thread cutting axis performs thread cutting based on spindle feedback using a cutting feed until reaching the hole bottom. The spindle and thread axis stop simultaneously at the hole bottom.
6) If a pause time (P1) is specified, pause.
7) Output the spindle reversal auxiliary function M04 and wait for the completion of M-code.
8) Upon the return of the M-code completion signal, retract the tap using a cutting feed until reaching the R-point (G01). The spindle and thread axis stop at the R-point.
9) If a spindle brake command exists, output the spindle brake release command (Mα+1).
10) If a pause time (P2) is specified, pause.
11) Return to the starting point using rapid movement (G00) (only executed in G98 mode). If a repeat count is specified, repeat from (1).
12) Restore the spindle to its state before thread cutting.

Note: In milling machine mode, G98 returns to the initial plane, and G99 returns to the R-point plane. In lathe mode, it's determined by Sys3019 BIT04. (Parameter settings are in the thread cutting parameter → thread cutting retraction mode.)

3. Rigid Thread Cutting Per Revolution - Rigid Thread Interpolation

1) Stop the spindle.
2) If there is a positioning command to the hole position, perform positioning action (G00); otherwise, skip.
3) When a spindle brake command (Mα) exists, output the spindle brake (no need to output spindle brake command when the Master spindle is the first spindle).
4) In case R-point is specified, execute positioning action towards the R-point (G00).
5) The spindle and thread axis move together, performing interpolated cutting feed for thread cutting until reaching the hole bottom.
6) If a pause time (P1) is specified, pause.
7) The spindle and thread axis retract the tap using interpolated cutting feed until reaching the R-point and stop simultaneously at the R-point.
8) If a spindle brake command exists, output the spindle brake release command

(Mα+1).

9) If a pause time (P2) is specified, pause.

10) Return to the starting point using rapid movement (G00) (only executed in G98 mode). If a repeat count is specified, repeat from (1).

11) Restore the spindle to its state before thread cutting.

4. Rigid Thread Cutting Per Minute - Rigid Thread Interpolation

When specifying the thread cutting command, it's not necessary to specify the spindle rotation command (specify based on the requirement, whether it's the first spindle or another spindle rotation).

1) Stop the spindle.

2) If there is a positioning command for locating the hole, execute a positioning action (G00); otherwise, skip this step.

3) If there is a spindle brake command (Mα), output the spindle brake (if the currently rotating spindle is the first spindle, there's no need to output the spindle brake command).

4) If the R-point is specified, execute a positioning action towards the R-point (G00).

5) The spindle and the threading axis move simultaneously, cutting and feeding the thread using an interpolation method until reaching the bottom of the hole.

6) If a pause time is specified (P1), pause during this time.

7) The spindle and threading axis retract and feed the tap using an interpolation method until reaching the R-point, and both the spindle and threading axis stop at the R-point.

8) If there is a spindle brake command, output the spindle brake release command (Mα+1).

9) If a pause time is specified (P2), pause during this time.

10) Return to the starting point using rapid movement (G00) (only executed in G98 mode); if a repeat count is specified, repeat from step (1).

11) Restore the spindle's rotation state before threading.

1.40.5.2 Deep-Hole Rigid Threading Cycle

Specify the Q value (not zero) and set the parameters for deep-hole rigid threading.

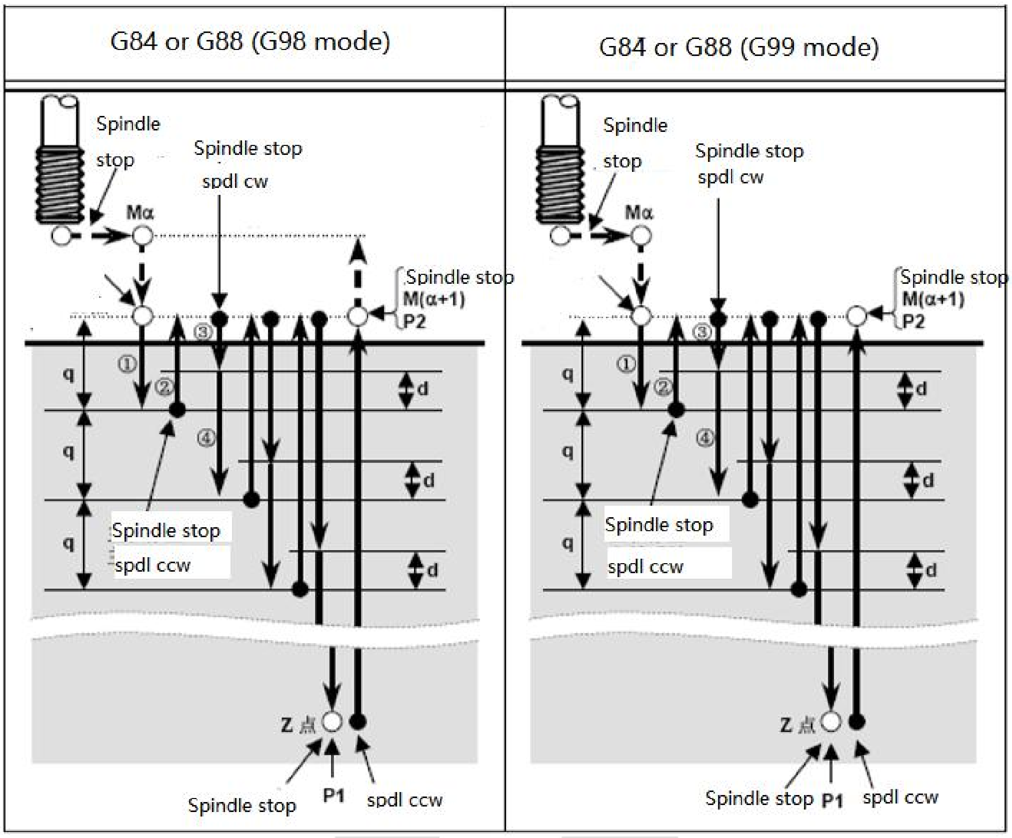


Figure 1.40-6

Action commands are as follows:

1. Cut only the specified amount q, as indicated by Q. When q is reached, both the main spindle and the threading axis stop simultaneously (Action ①).
2. Reverse the main spindle, and the threading axis retracts to the R point using cutting feed, where both the main spindle and threading axis stop (Action ②).
3. Start the main spindle in the forward direction, and the threading axis feeds to a position d from the last cut (Action ③, retraction ratio is effective).
4. The threading axis feeds d + q (Action ④) continuously until reaching the bottom of the hole.
5. After a stop time (P1) set, reverse for retraction.

Note:

1. If P1, Mα, M(α+1), or P2 is not specified or set, it will not be executed or output.
2. Actions ① and ④ use cutting speed and the time constant of rigid threading.
3. Actions ②, ③, and the movement from the hole bottom (Z point) to the R point use the retraction ratio of rigid threading and the time constant of rigid threading.

1.40.5.3 High-Speed Deep-Hole Rigid Threading Cycle

Specify a non-zero Q value and set parameters for high-speed deep-hole rigid threading.

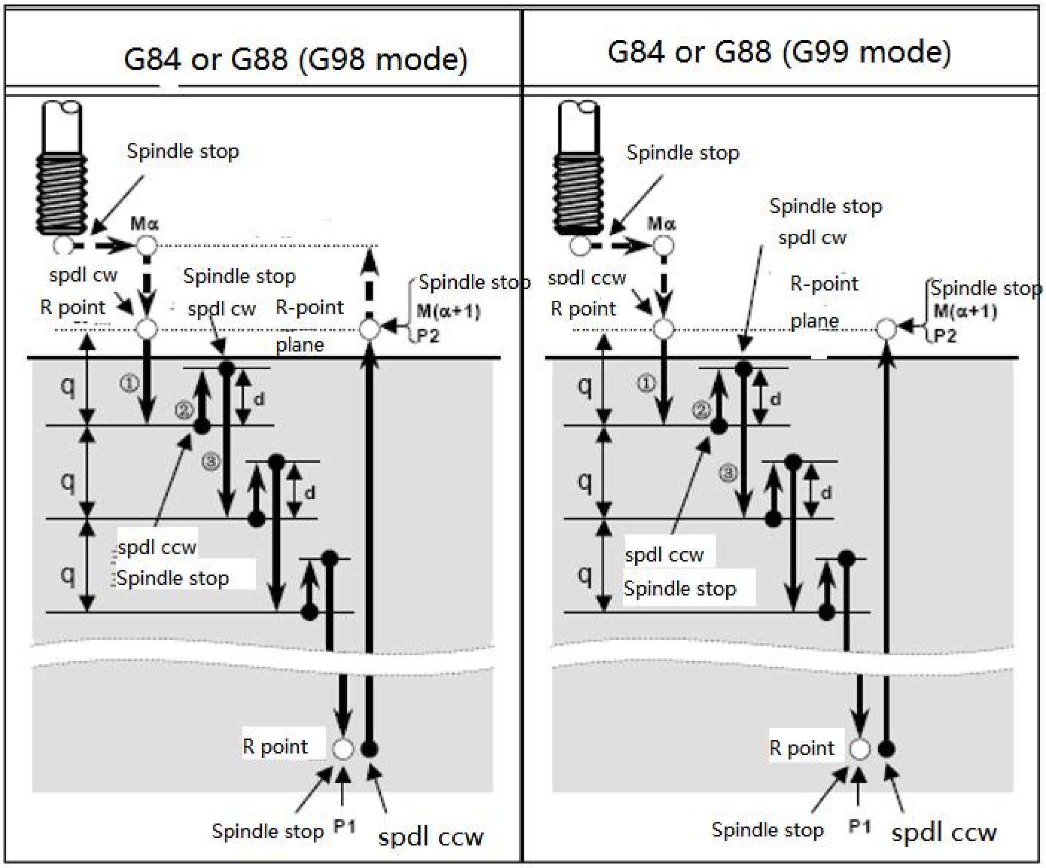


Figure 1.40-7

Action commands are as follows:

1. Only cut the specified amount q as indicated by Q. When q is reached, both the main spindle and the threading axis stop simultaneously (Action ①).
2. Reverse the main spindle, and the threading axis retracts to the d point using cutting feed, where both the main spindle and threading axis stop (Action ②).
3. Start the main spindle in the forward direction, and the threading axis feeds d + q.
4. Continue this cycle until reaching the bottom of the hole.
5. After a stop time (P1) set, reverse the main spindle for retraction.

Note:

1. If P1, Mα, or P2 is not specified or set, it will not be executed or output.
2. Actions ① and ③ use cutting speed and the time constant of rigid threading.
3. Action ② and the movement from the hole bottom (Z point) to the R point use the retraction ratio of rigid threading and the time constant of rigid threading.

1.40.6 Reverse End/Face Threading Cycle (G84.1/G88.1)

1.40.6.1 Instruction Format

G84.1 X/U_C/H_Z/W_Rr_Qq_Pp_Ff_Kk_Mm_Jj_Dd

G88.1 Z/W_C/H_X/U_Rr_Pp_Qq_Ff_Kk_Mm_Jj_Dd

G84.1(G88.1)	G84 - Face Deep Hole Threading Cycle; G88 - Longitudinal Deep Hole Threading Cycle;
X/U(Z/W) _C/H_	Specification of the Initial Point of the Cycle (Absolute/Incremental); Angle of the C-axis;
Z/W(X/U)_	Bottom of Hole Position (Absolute/Incremental from R-point) - Here, if Z_ is specified, it denotes the absolute coordinate from the start point to the bottom of the hole. If W_ is specified, it denotes the incremental value from R-point to the bottom of the hole;
Rr_	Specify the R-point (Modal, specified as an incremental radius value) - If R-value is not set, the R-point plane coincides with the initial plane;
Qq_	Specify the Entry Amount for Each Pass (Modal, ignore sign, specified as an incremental radius value);
Pp_	Specify the pause time at the bottom of the hole, similar to G04 P_ (Modal) - Unit: ms;
Ff_	Specify the feed rate (Modal);
Kk_	Specify the number of repetitions, ranging from 0 to 9999 (default is 1) (Non-modal);
Mm_	Specify the main spindle brake command (Non-modal, parameter planning, can flexibly set M-code);
Dd	Specify the main spindle used in the threading cycle (Modal) - (Range: 1 to the number of main spindles);

1.40.6.2 Notes

Reverse threading cycles have no significant differences in format compared to normal threading cycles. The only distinction lies in the threading process: normal threading involves a forward motion for threading in and a reverse motion for threading out, whereas reverse threading employs a reverse motion for threading in and a forward motion for threading out.

1.41 External/Internal Turning Cycle (G90)

The G90 command can execute longitudinal straight and tapered cutting cycles.

1.41.1 Instruction Format

G90 X(U)__Z(W)__F__R__

X, Z: Absolute coordinates of the longitudinal cutting endpoint.
U, W: Incremental values of the cutting endpoint relative to the cycle start point.
R: Taper amount (defaults to flat cutting if not specified).
F: Cutting feed rate.

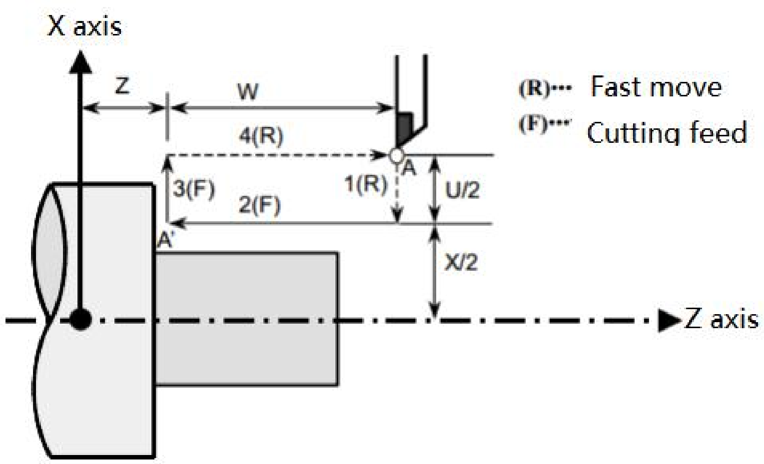


Figure 1.41-1

Action Explanation：

1. The first action, move the tool from the starting point (A) to the specified coordinate

values on the second axis in rapid movement mode (G00).

2. The second action, move the tool to the specified coordinate values on the first axis in cutting feed mode (G01).
3. The third action, move the tool to the starting coordinate values on the second axis in cutting feed mode (G01).
4. The fourth action, move the tool to the starting coordinate values on the first axis in rapid movement mode (G00).

1.41.1.1 Tapered Cutting Cycle

Execute a tapered cutting cycle when the R value is set to a non-zero value.

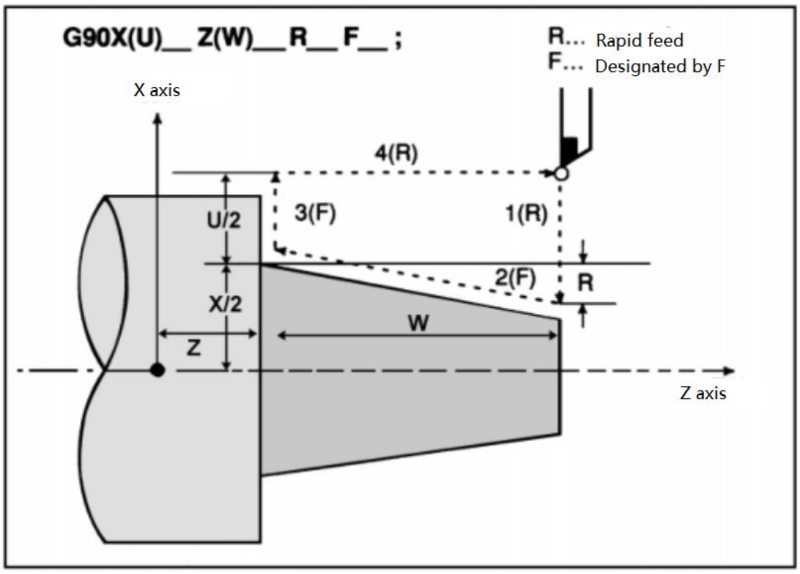


Figure 1.41-2

1.41.1.2 Relationship Between Taper Symbol and Tool Path

The relationship between the taper symbol and the longitudinal cutting endpoint based on absolute and incremental instructions.

The tool path is as shown in the following figure:

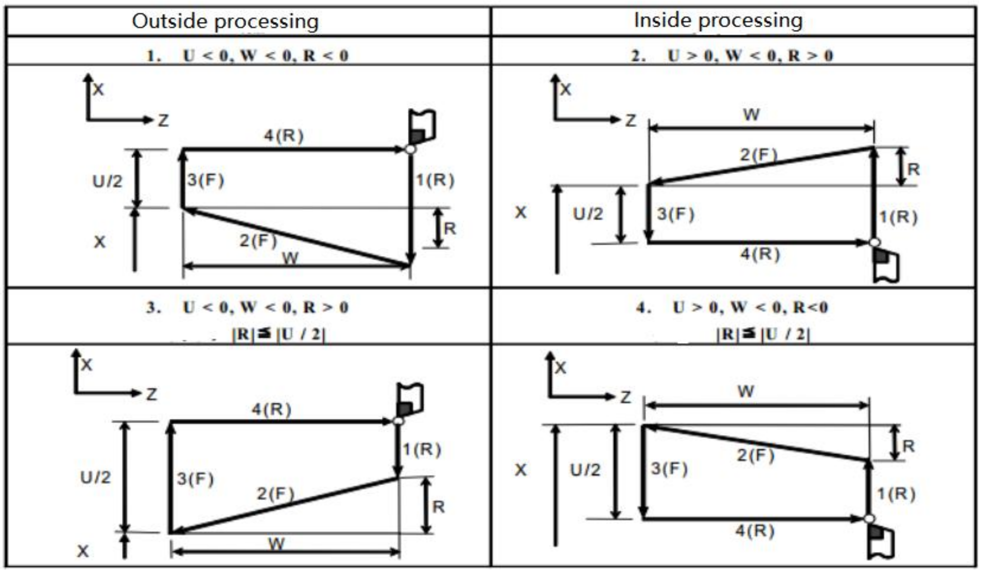


Figure 1.41-3

Note:

1. In the G90 cycle command, you can execute it step by step, program preview, or pause.
2. Single-step execution is effective for each segment.
3. Since the values of X(U), Z(W), and R are modal during the fixed cycle, if X(U), Z(W), or R is not re-specified, the originally specified data remains valid.

1.41.2 Example 1

Axial straight turning cycle.

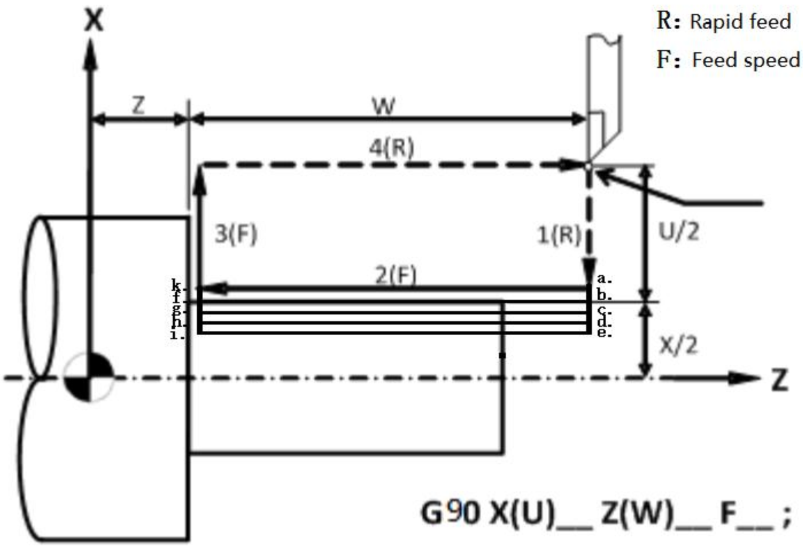


Figure 1.41-4

Diameter Programming

G50 S5000; // Maximum speed 5000rpm;

T01; // Use tool number 1;

G96 M03 S130; // Constant surface speed, Table speed 130m/min, spindle clockwise rotation;

G00 X60.0 Z65.0; // Rapid positioning to point a. (starting point);

G90 X45.0 Z15.0 F0.6; // Execute axial turning cycle, feed rate 0.6mm/rev, a.→b.→f.→k.→a;

X40.0; // a.→c.→g.→k.→a;

X35.0; // a.→d.→h.→k.→a;

X30.0; // a.→e.→i.→k.→a;

G28 X60.0 Z70.0; // Rapidly position to the specified midpoint, then return to the machine origin;

M05; // Spindle stop;

M30; // Program end;

1.41.3 Example 2

Axial Taper Turning Cycle.

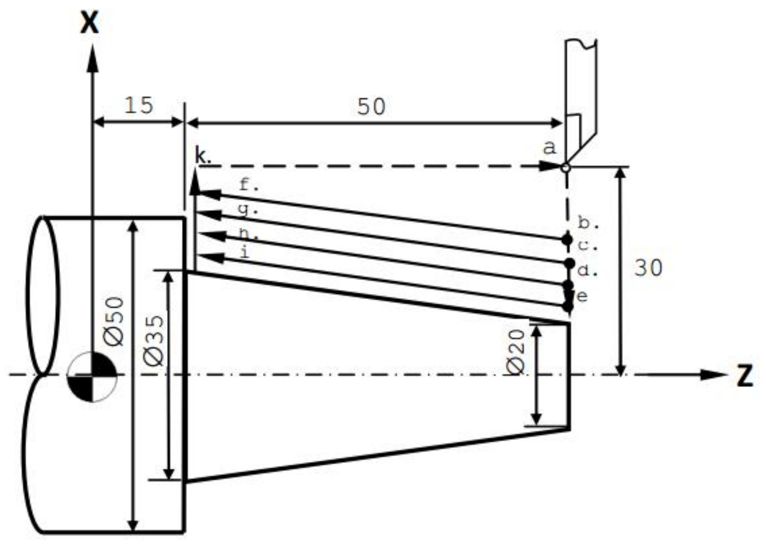


Figure 1.41-5

G50 S5000; // Maximum spindle speed 5000rpm

T01; // Use tool 1

G96 M03 S130; // Constant surface speed, table speed 130m/min, spindle forward

M08; // Turn on coolant

G00 X60.0 Z65.0; // Rapid positioning to point a (start point)

G90 X53.0 Z15.0 R-7.5 F0.6; // Execute axial turning cycle, feed rate 0.6mm/rev, a. → b. → f. → k. → a;

X48.0; // a. → c. → g. → k. → a;

X42.0; // a. → d. → h. → k. → a;

X35.0; // a. → e. → i. → k. → a;

G28 X60.0 Z70.0; // First rapid positioning to the specified intermediate point, then return to machine zero;

M09; // Turn off coolant

M05; // Stop spindle

M30; // End of program

1.42 Incremental command G91.9 / Absolute command G90.9

- There are two ways to specify tool movement: incremental and absolute.
- Regarding the coordinates of the moving point, when specifying incremental values, the command is issued based on the distance from the current point, while absolute values are specified based on the distance from the coordinate origin.
- Axes can be specified in incremental or absolute modes in the following ways:
  1. U, V, W specify incremental displacement for X, Y, Z-Axis (there are no specific letters for other axes).
  2. G90.9, G91.9 specify certain axes for absolute or incremental programming (can be used in all modes).

Mode ①

As shown in the figure below, P1 is positioned to P2:

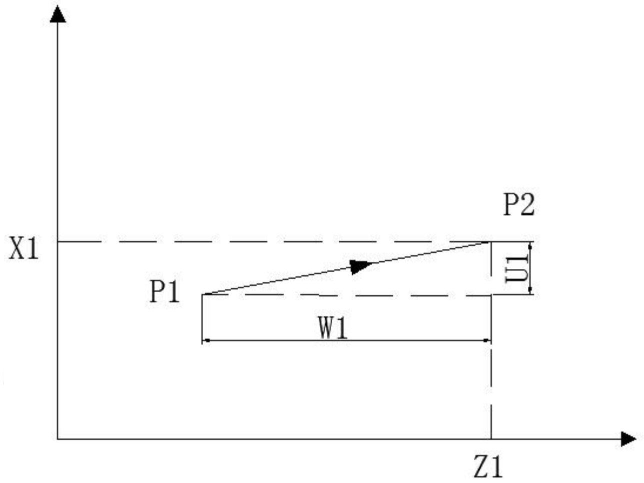


Figure 1.42-1

❖ As shown in the figure above, the distance from point P1 to P2 is (U1, W1), and the absolute coordinates of point P2 are (X1, Z1).

❖ Assuming: U1 = 10.0, W1 = 40.0, X1 = 30.0, Z1 = 60.0, the specification for P1→P2 is as follows:

❖ Incremental specification: G00 U20. W20.

❖ Absolute specification: G00 X30. Z60.

Mode ②

As shown in the figure below, P1 is positioned to P2:

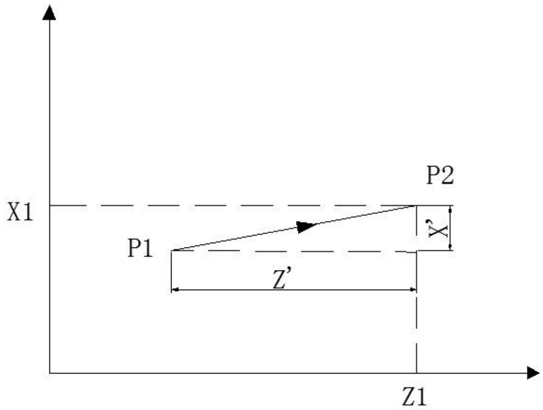


Figure 1.42-2

- Assuming: X' = 10.0, Z' = 40.0, X1 = 30.0, Z1 = 60.0, the specification for P1→P2 is as follows:
- Incremental specification: G91.9 X__ Z__ (Incremental specification in the XZ-Axis, fill in any number for the underscores, otherwise it will trigger an alarm) G00 X20. Z20.
- Absolute specification: G90.9 X__ Z__ (Absolute specification in the XZ-Axis, fill in any number for the underscores, otherwise it will trigger an alarm) G00 X30. Z60.

1.43 Thread Cutting Cycle (G92)

The G92 thread cutting cycle is used for executing fixed cycles of straight and tapered thread cutting.

1.43.1 Instruction Format

G92 X(U)__Z(W)__Q__L__F(E)__R__J____K____

- X(Z): Absolute coordinates of the cutting endpoint C.
- U(W): Incremental values of the cutting endpoint C relative to the starting point A.
- F(E): Pitch of the longitudinal axis (the axis with the most movement), F and E cannot be specified simultaneously.
- Q: Set the starting angle offset of the thread, with a set value range of 0.000° to 359.999°.

- L: Multi-thread setting item, with a set value range of 1 to 99.
- J: Chamfer angle of the G92 thread retraction point. When the letter J is not present, SYS3043 determines the chamfer angle of the G92 thread retraction point. (Parameter settings in: Thread parameters → Chamfer angle of G92)
- K: Distance set for chamfering along the main spindle direction at the G92 thread retraction point (unit 0.1L, range 0.1L to 12.7L). When the letter K is not present, SYS3042 determines the distance of chamfering along the main spindle direction at the G92 thread retraction point. (Parameter settings in: Thread parameters → Distance of chamfering along the main spindle direction at G92 thread retraction point)
- R: Taper amount, the absolute coordinates difference between the thread cutting starting point B and the cutting endpoint C. When R = 0 or R is not set, it is for straight thread cutting; when R ≠ 0, it is for tapered thread cutting, and it is a modal command.

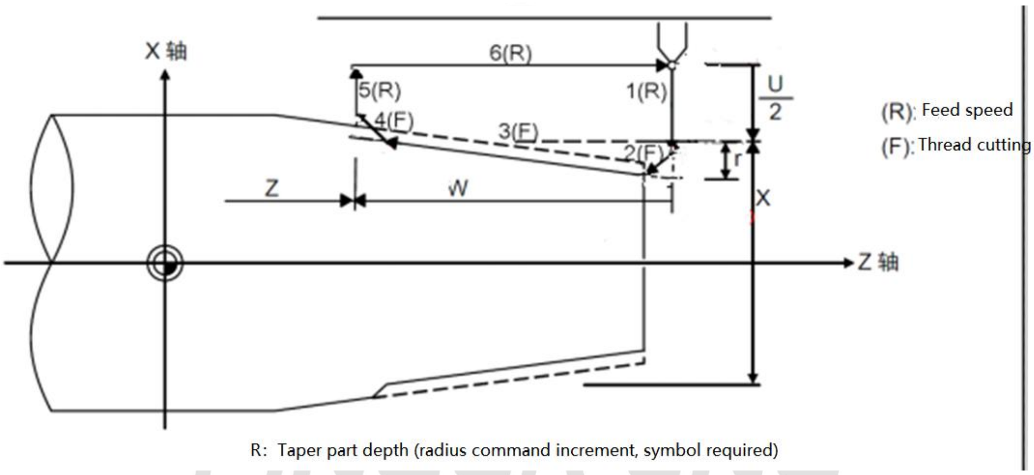


Figure 1.43-1

Action Explanation:

1. The X-Axis moves rapidly from the starting point to the cutting starting point (2).
2. After waiting for synchronization, it performs cutting feed to the starting point of step (3) based on Sys3042 (Distance set for chamfering along the main spindle direction at the G92 thread entry point), Sys3043 (Chamfer angle of the G92 thread entry point), and G92's acceleration and deceleration time.
3. Using G92's acceleration and deceleration time and the set F value, it performs cutting feed to the starting point of step (4).
4. Using Sys3040 (Distance set for chamfering along the main spindle direction at the thread retraction point), Sys3041 (Chamfer angle of the thread retraction point), and G92's acceleration and deceleration time, it performs cutting feed to the starting point of step (5).

5. It rapidly positions to the starting point of step (6) at G00 speed.
6. It rapidly positions to the cycle starting point at G00 speed.

1.43.2 Notes

1. L (number of thread heads) and Q (thread starting angle) can coexist. When L and Q exist simultaneously, the thread angle set by Q is 0°, and it divides 360° into L equal parts (360°/L).
   - For example, G92 X10.0 Z10.0 L3 Q90.0 F2.0.
   - The first thread starts at 90°.
   - The second thread starts at 90° + 120° = 210° (90° + 360° × 1/3).
   - The third thread starts at 90° + 240° = 330°.
2. L can have a decimal point, and having or not having a decimal point has the same effect.
3. G92 modal is not canceled when there are M, T, or S codes after the G92 command (except for M02 and M30), but other G codes cancel the G92 modal.

1.43.3 Example 1

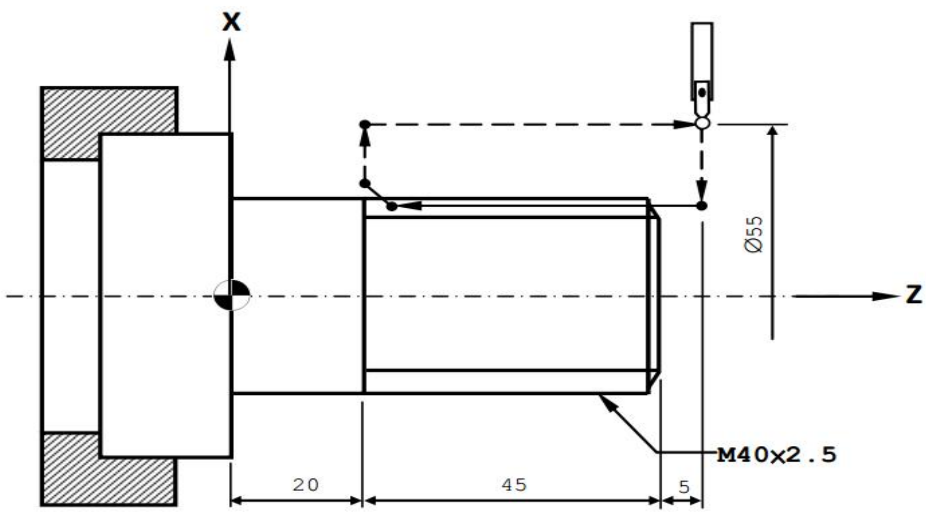


Figure 1.43-2

Diameter Programming

T03; // Use tool #3;

G96 M03 S100; // Set constant surface speed, Table speed 100 m/min, Spindle

```
                                forward rotation;
G00 X50.0 Z70.0;               // Rapid positioning to the starting point of the cycle;
M08;                           // Turn on cutting fluid;
G92 X39.0 Z20.0 L3 F2.5;       // Execute threading cycle, triple-threaded, 1st cycle;
X38.3;                         // 2nd cycle;
X37.7;                         // 3rd cycle;
X37.3;                         // 4th cycle;
X36.9;                         // 5th cycle;
X36.75;                        // 6th cycle;
G97;                           // Cancel fixed spindle speed;
G28 X60.0 Z75.0;               // Rapid positioning to the specified middle point and then
                                  return to machine zero point;
M09;                           // Turn off cutting fluid;
M05;                           // Stop the spindle;
M30;                           // Program end;
```

1.43.4 Example 2

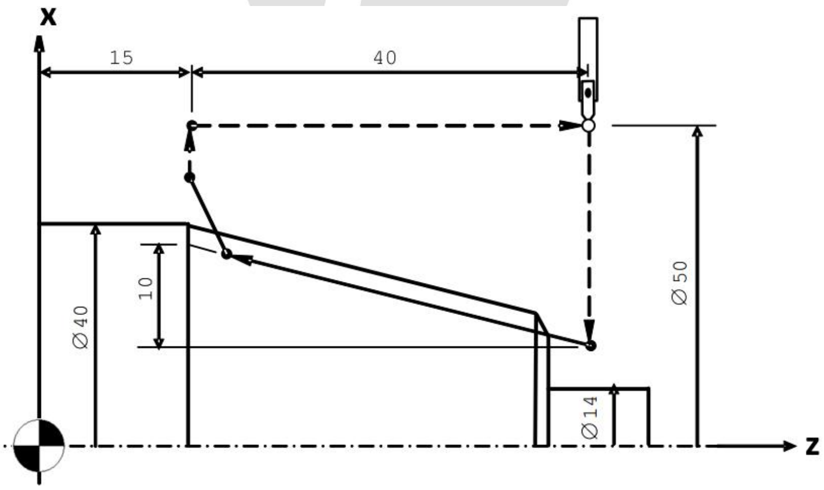


Figure 1.43-3

Diameter Programming

```
T03;                        // Use tool #3;
G96 M03 S100;               // Set constant surface speed, Table speed 100 m/min, Spindle
                               forward rotation;
G00 X50.0 Z55.0;            // Rapid positioning to the starting point of the cycle;
M08;                        // Turn on cutting fluid;
```

G92 X39.0 Z15.0 R-10.0 F2.5; // Execute tapered threading cycle, 1st cycle;
X38.3; // 2nd cycle;
X37.7; // 3rd cycle;
X37.3; // 4th cycle;
X36.9; // 5th cycle;
X36.75; // 6th cycle;
G97; // Cancel fixed spindle speed;
G28 X60.0 Z70.0; // Rapid positioning to the specified middle point and then return to machine zero point;
M09; // Turn off cutting fluid;
M05; // Stop the spindle;
M30; // Program end;

1.44 Longitudinal Cutting Cycle (G94)

The G94 command is used for performing longitudinal cutting cycles, allowing for both straight-line and tapered cutting operations.

1.44.1 Instruction Format

G94 X(U)__Z(W)__F__R__

X(Z): The absolute coordinate value of the bottom-side cutting endpoint.
U(W): The incremental value of the cutting endpoint relative to the starting point.
R: Taper amount (not specified for flat cutting).
F: Cutting feed rate.

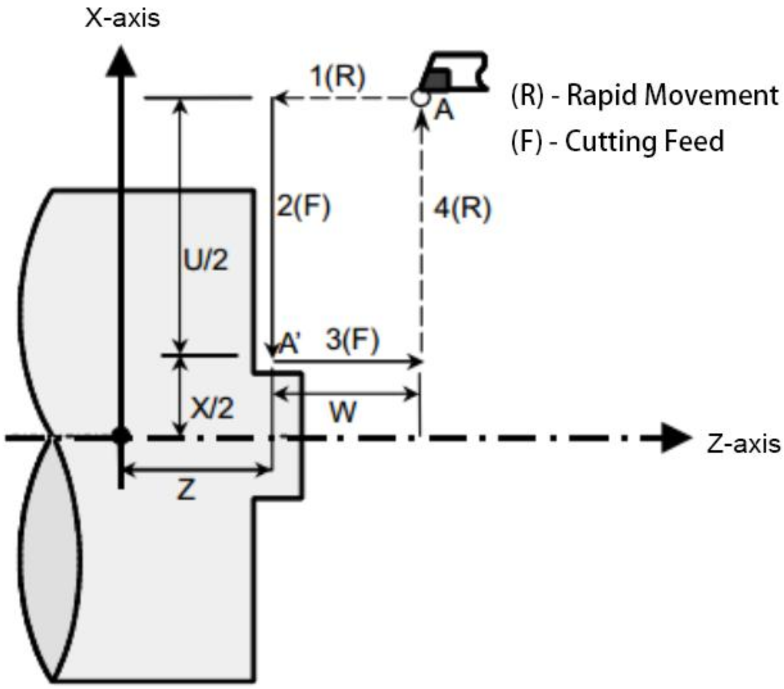


Figure 1.44-1

Action Explanation:

1. First Action: Move the tool from the starting point (A) to the specified command coordinates on the first axis using the rapid movement mode (G00).
2. Second Action: Move the tool to the specified command coordinates on the second axis using cutting feed mode (G01).
3. Third Action: Move the tool to the starting coordinates on the first axis using cutting feed mode (G01).
4. Fourth Action: Move the tool to the starting coordinates on the second axis using rapid movement mode (G00).

1.44.2 Example 2

When R is set to a non-zero value, perform tapered surface cutting cycle.

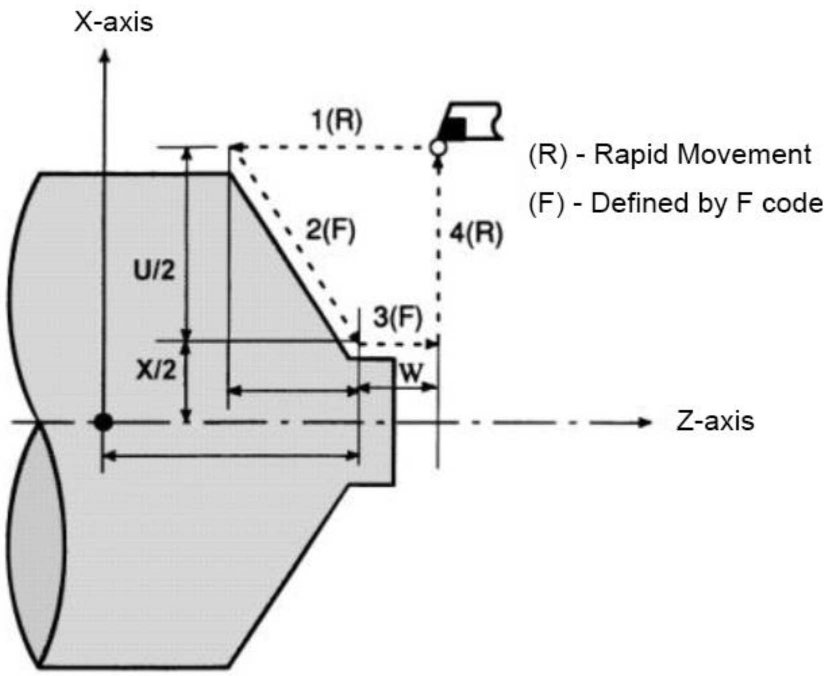


Figure 1.44-2

1.44.3 Relationship Between Taper and Toolpath

Based on the sign of the taper and the relationship between longitudinal cutting endpoints using absolute and incremental instructions, the toolpath is illustrated in the following Figure:

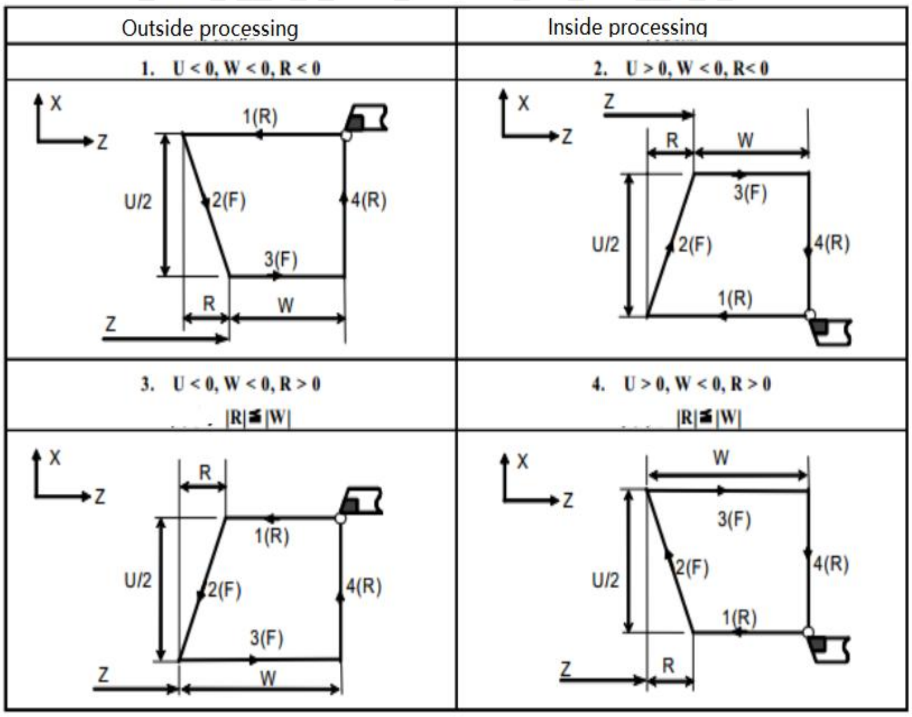


Figure 1.44-3

Note:

1. The G94 cycle command can be executed step by step, predicted by the program, and paused.
2. Single-step execution is effective in each segment.
3. Since the values of X(U), Z(W), and R are modal during the fixed cycle, the originally specified data remains effective if X(U), Z(W), or R are not re-specified.

1.44.4 Example 1

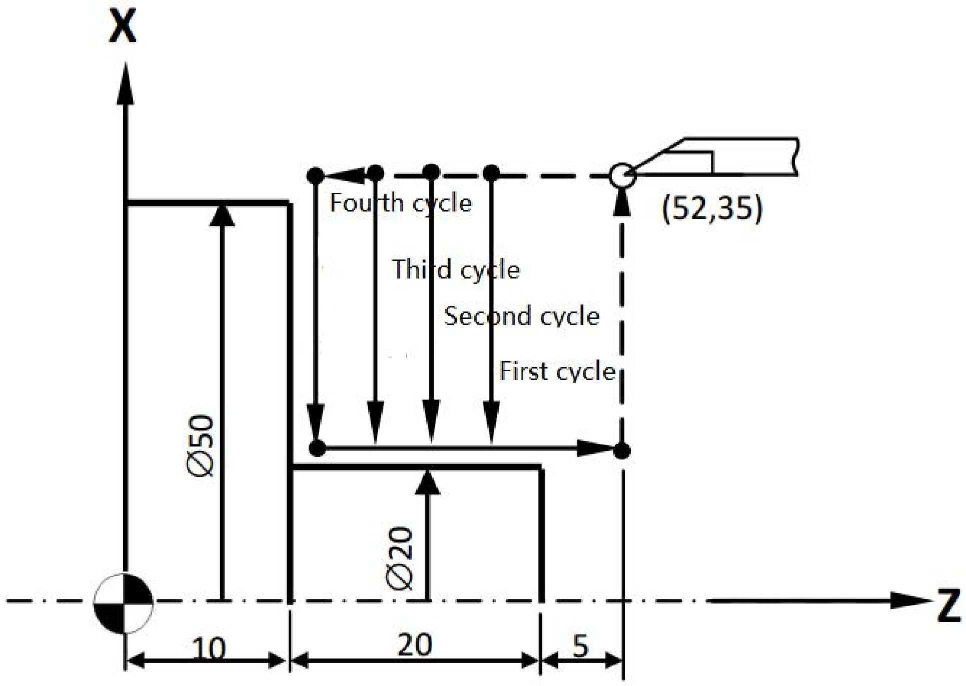


Figure 1.44-4

Diameter Programming

G50 S3000; // Limit the spindle speed to 3000 rpm.

T01; // Use tool No. 1.

G96 M03 S130; // Constant speed, table speed 130 m/min, spindle forward rotation.

M08; // Turn on the coolant.

G00 X52.0 Z35.0; // Rapid positioning to the starting point of the cycle.

G94 X20.0 Z25.0 F0.6; // Execute radial straight line turning cycle, feed rate 0.6 mm/rev, first cycle.

Z20.0; // Second cycle.
Z15.0; // Third cycle.
Z10.0; // Fourth cycle.
G28 X70.0 Z40.0; // Rapid move to the specified intermediate point, then return to the machine origin.
M09; // Turn off the coolant.
M05; // Stop the spindle.
M30; // End of the program.

1.44.5 Example 2

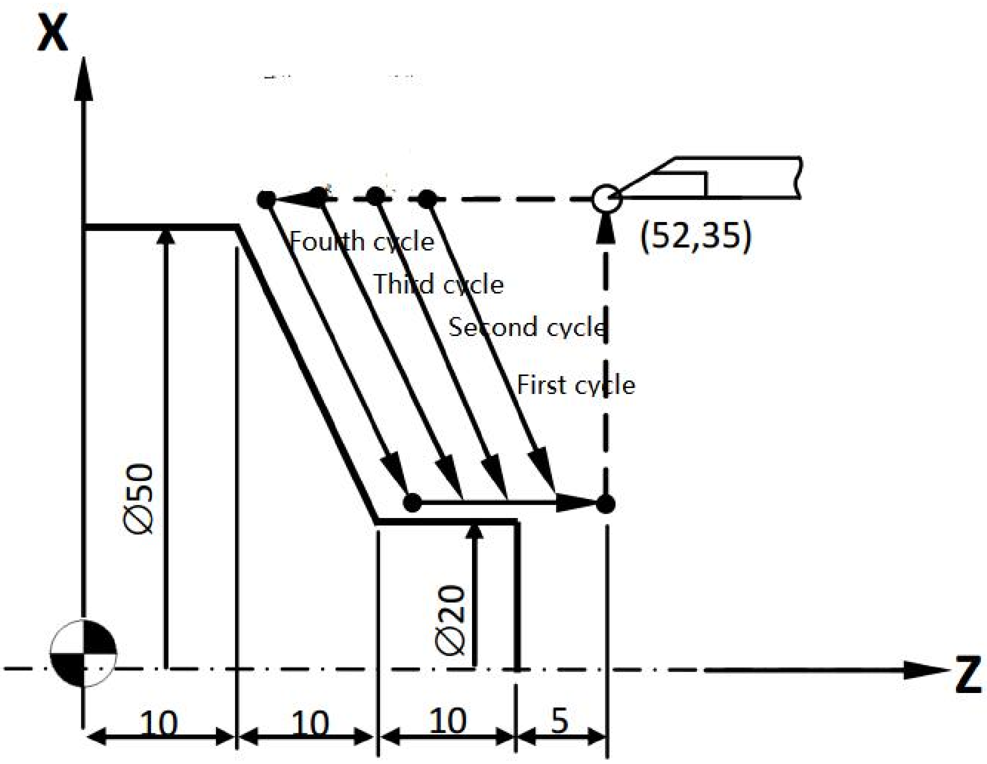


Figure 1.44-5

Diameter Programming

G50 S3000; // Limit the spindle speed to 3000 rpm.
T01; // Use tool No. 1.
G96 M03 S130; // Constant speed, table speed 130 m/min, spindle forward rotation.
M08; // Turn on the coolant.

```
G00 X52.0 Z35.0;        // Rapid positioning to the starting point of the cycle.
G94 X20.0 Z32.0 R-10.0 F0.6;// Execute radial taper turning cycle, feed rate 0.6 mm/rev,
                                first cycle.
Z28.0;                  // Second cycle.
Z24.0;                  // Third cycle.
Z20.0;                  // Fourth cycle.
G28 X70.0 Z35.0;        // Rapid move to the specified intermediate point, then return to
the machine origin.
M09;                    // Turn off the coolant.
M05;                    // Stop the spindle.
M30;                    // End of the program.
```

1.45 Constant Linear Speed Function (G96/G97)

In radial cutting operations, automatic control of the spindle speed is carried out based on changes in coordinate values, ensuring constant speed machining at the cutting point. The constant linear speed function is activated by the G96 command, which is modal, and it is turned off by the G97 command. It is used in conjunction with the G50 command to set the upper and lower speed limits for the spindle during machining.

Algorithm

V = ПDN

V = Table surface cutting speed, i.e., the value of S in G96 (unit: m/min)

D = Diameter of the workpiece being cut on the table (unit: meters)

N = Spindle speed

П = 3.14159 (pi)

1.45.1 Instruction Format

G96 S_ P_

S: Table surface linear speed, in m/min, with a valid range from 0 to 2147483.648, no decimal points allowed

P: Axis for constant speed control, no decimal points allowed

=1: Specifies the X-axis

=2: Specifies the Y-axis

......

G97 S_: Cancels constant speed control and saves the set table surface speed.

G50 S_ Q_, modal command

S: Spindle's maximum limiting speed, used when the constant speed control axis is at a lower position.

Q: Spindle's minimum limiting speed, used when the constant speed control axis is at a higher position.

1.45.2 Notes

1. The spindle is determined by the master spindle of the current channel, and the constant linear speed control axis can be determined by the P or #M03 command as follows:
   P specification: P=1: Specifies the X-axis
   =2: Specifies the Y-axis
   ...
   #M03 specification: #M03=0: Fixed X-axis (P specification is ineffective)
   =1: Specifies the X-axis
   =2: Specifies the Y-axis
2. The G96 state is generally only effective during cutting feed.
3. The S and Q values specified in the G50 S_ Q_ command are modal values and will be saved. When switching back to the G96 state, the saved S and Q values from the G50S_Q_ command will be used as the maximum/minimum spindle speed limits, and you don't need to specify them again.
4. When executing a program segment with G50 S_ Q_ M3; the functionality of setting the maximum and minimum spindle speeds with S and Q is effective. However, S and Q are not used as spindle speed values.
5. When G96 is canceled by G97, and there is no new spindle speed command S code in the G97 instruction, the spindle speed at the end of the G96 state is used as the spindle speed in the G97 state.
6. The S and P values in the G96 command are not allowed to have decimal points.
7. Multiple channels specifying the same spindle as a constant linear speed control spindle in G96 result in a system alarm (Program Error -96-5). In G96, S cannot be specified as <= 0, otherwise, it triggers an alarm (Program Error -96-1). P in G96 cannot be specified as <= 0 or > the number of physical axes, otherwise, it triggers an alarm (Program Error -96-2). P in G96 cannot be specified as an invalid axis, otherwise, it triggers an alarm (Program Error -96-3).
8. When using the same master spindle, if one channel is running a G96 command,

other channels running G96 commands will result in a system alarm (Program Error -96-4).

9. Resetting or ending a program does not cancel the G50 state, while resetting or ending does cancel the G96 state.

1.45.3 Example

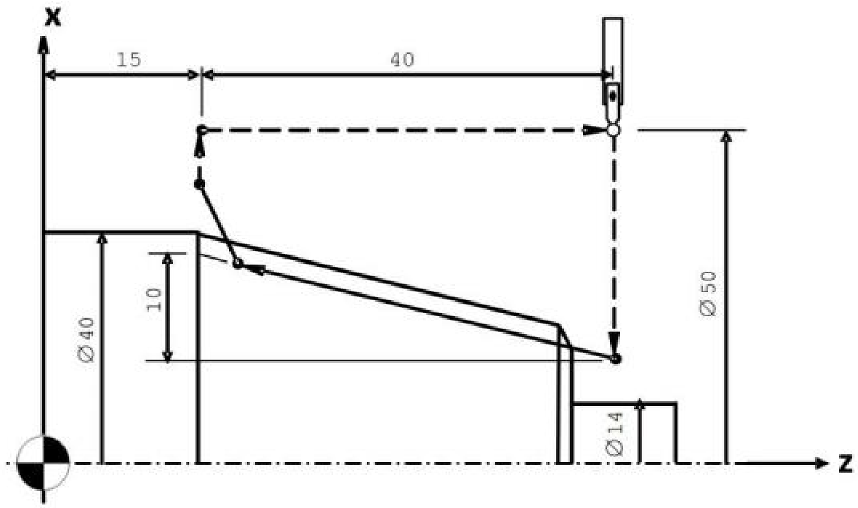


Figure 1.45-1

Diameter Programming

T03;	// Use tool No. 3
G96 M03 S100;	// Constant speed, table speed 100 m/min, spindle forward rotation
G00 X50.0 Z55.0;	// Rapid positioning to the starting point of the cycle
M08;	// Turn on the coolant
G92 X39.0 Z15.0 R-10.0 F2.5;	// Execute tapered threading cycle, first cycle
X38.3;	// Second cycle
X37.7;	// Third cycle
X37.3;	// Fourth cycle
X36.9;	// Fifth cycle
X36.75;	// Sixth cycle
G97;	// Cancel fixed spindle speed
G28 X60.0 Z70.0;	// Rapid positioning to the specified intermediate point, then return to the machine origin
M09;	// Turn off the coolant
M05;	// Stop the spindle

M30; // End of the program

1.46 Feed Mode Setting (G98/G99)

1.46.1 Instruction Format

G98: Feedrate per minute mode (mm/min).

G99: Feedrate per spindle revolution mode (mm/rev).

Conversion formula between G98 and G99:

Fm = Fr × S.

Fm: Feedrate per minute, mm/min.

Fr: Feedrate per spindle revolution, mm/rev.

S: Spindle speed, rev/min.

1.46.2 Notes

1. G99 instruction must be used in G01 mode; it's invalid in G00 rapid feed mode.
2. In G99 mode, the machine tool must have a rotating spindle device and an encoder feedback system.
3. When using G99, you must confirm the following parameter settings:
   1) Acceleration and deceleration time: =0 (exponential), =1 (linear), =2 ("S" type).
   2) Setting of the encoder command for the rotation axis.
   3) Setting of the encoder pulses for the rotation axis.

1.47 Automatic Chip Breaking (G165)

1.47.1 Instruction Format

G165 P__ Q__;

This command adds oscillatory motion with a certain frequency and amplitude on the linear feed motion to create a chip-breaking effect.

P: Oscillation frequency, range: 0, 1, 3 (unit: chip-breaking times per spindle revolution).

When set to 0, oscillation functionality ends.

Q: Oscillation amplitude, can be omitted, range: 15~100 (unit: % of feedrate per spindle revolution). If omitted or out of range, the value from MCM48 setting is used.

1.47.2 Notes

1. The chip-breaking function is effective only during G01 linear feed motions and compound cycles.
2. It automatically deactivates when executing M30, M99 (used in the main program), or CNC reset.
3. When performing an emergency stop or pausing feed hold, the oscillation motion stops immediately after completing the current oscillation cycle.
4. The chip-breaking function is inactive when the spindle speed is less than or equal to 10 revolutions.
5. When the CNC is approaching the end of program execution, the chip-breaking function automatically pauses, preserving smooth transitions between program segments.
6. The effectiveness of chip breaking depends on the material being machined and servo characteristics. If the chip-breaking effect is unsatisfactory, you can adjust the oscillation amplitude and frequency using P and Q to achieve the desired result.

1.47.3 Example

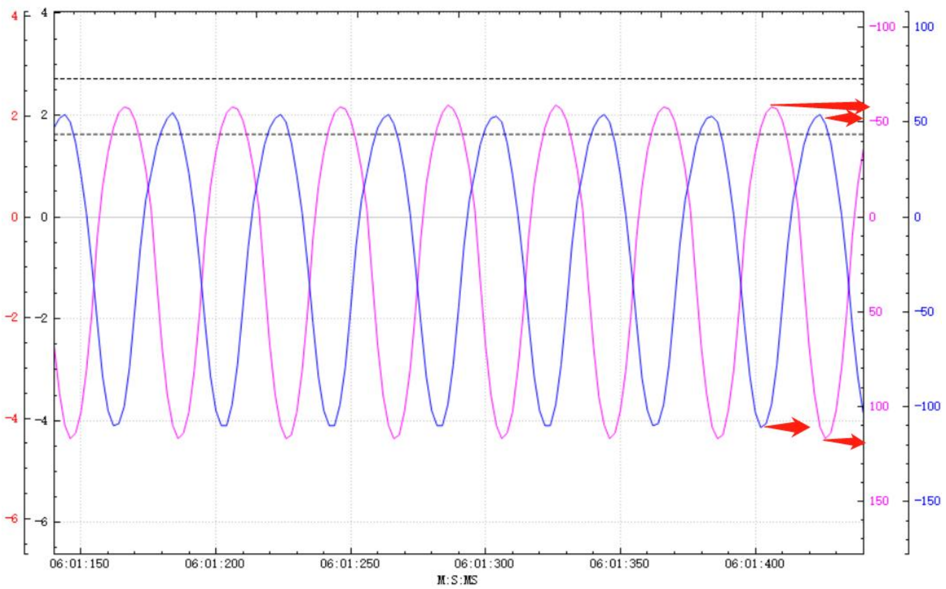


Figure 1.47-1

```
T0101;                  // Call tool No. 1
G99;                    // G99 mode
M03 S2000;              // Spindle forward at 2000 RPM
G00 X50 Z2;             // Rapid positioning in XZ-Axis
G01 Z-0.2 F0.12;        // Move to Z-axis position
G165 P1 Q40;            // Activate chip-breaking, 1 chip break per revolution,
                           40% amplitude
G01 X-0.5 F0.2;         // Face milling 0.2mm
G165 P0;                // Deactivate chip-breaking
G00 W1;                 // Rapid positioning
X41;                    // Rapid positioning
G165 P3 Q70;            // Activate chip-breaking, 3 chip breaks per revolution,
                           70% amplitude
G01 X43 Z-5 F0.12;      // Cut chamfer
G01 Z-100 F0.12;        // Cut the outer circle 0.5mm deep
U2;                     // Retract in the X-axis
G0 Z50;                 // Retract in the Z-axis
G165 P0;                // Deactivate chip-breaking
M30;                    // End of program
```


Tips for Use:

1. If the cutting amount is larger, Q should be increased. However, increasing the Q value can lead to motor impact and increased vibration. If this is unacceptable, you may need to reduce the spindle speed, which will decrease machining efficiency.
2. If the cutting amount is smaller, you can use a smaller Q value, which will reduce the impact on the motor and decrease vibration.
3. A larger P value results in shorter chip breaking, which can improve chip breaking performance. However, it also places higher demands on the motor and machine, leading to increased vibration. If this is unacceptable, you may need to lower the spindle speed, again affecting machining efficiency.
4. It's generally preferable to select smaller P values for machining, unless there are specific requirements for chip length.

1.48 Corner Chamfer, Corner Fillet, Linear Angle (,C, R, A)

The ,C, ,R, ,A, and other commands are used to achieve chamfering, filleting, and other functions during machining.

1.48.1 Corner Chamfer (,C_)

In two consecutive blocks, the ",C_" command can be used in the first block to execute corner chamfering. ",C_" indicates the length from the imaginary chamfer starting point to the chamfer end point.

1.48.1.1 Instruction Format

N1 G0x X__Z__,C__
N2 G0x X__Z__

G0x: Can be one of G00, G01, G02, or G03.
C__: Indicates the length from the imaginary chamfer starting point to the chamfer end point.

1.48.1.2 Example 1

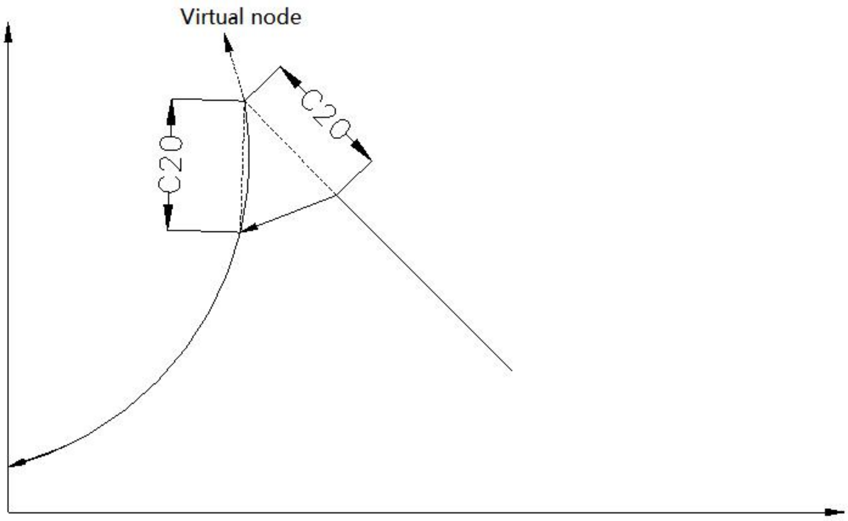


Figure 1.48-1

Straight Line to Arc (Diameter Programming)

- **Absolute Value Command**

G28; // Return to reference point

G00 X25. Z100.; // Rapid positioning

G01 ,C20. X75. Z50. F100.; // Chamfer at an angle

G02 X25. Z0. R50; // Arc command

M30; // End of the program

- **Incremental Value Command**

G28; // Return to reference point

G00 U25. W100.; // Rapid positioning

G01 ,C20. U50. W-50. F100; // Chamfer at an angle

G02 U-50. W-50. R50; // Arc command

M30; // End of the program

1.48.1.3 Example 2

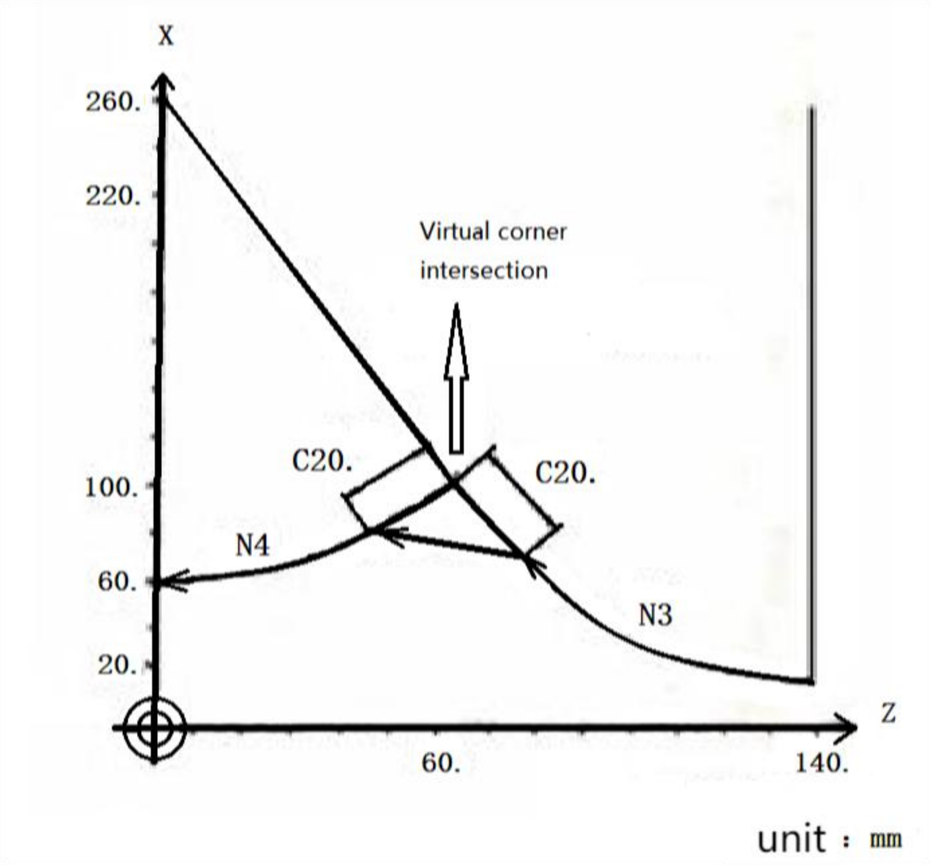


Figure 1.48-2

Circular Arc to Circular Arc (Radius Programming)

- **Absolute Programming**

G28 X_ Z_;

G00 X10. Z140.;

G02 X50. Z60. I100. K0. ,C20. F100;

G02 X30. Z0. I80. K-60.;

M30;

- **Incremental Programming**

G28;

G00 U10. W140.;

G02 U40. W-80. I100. K0. ,C20. F100.;

G02 U-20. W-60. I80. K-60.;

M30

1.48.2 Rounded Corners (,R_)

In two consecutive blocks, in the first block, you can execute rounded corners using ",R_" with a Table that indicates the radius of the rounded corner.

1.48.2.1 Instruction Format

N1 G0x X__Z__,R__
N2 G0x X__Z__

G0x: Can be G00, G01, G02, or G03.
,R__: Table indicating the radius of the rounded corner.

1.48.2.2 Example

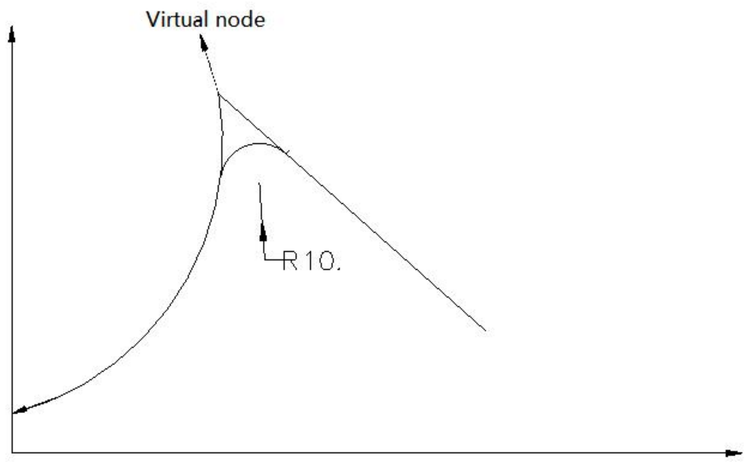


Figure 1.48-3

Straight Line to Circular Arc (Radius Programming)

- **Absolute Programming**

G28;

M03 S2000

G00 X30. Z100.;

G01 X80. Z50. ,R10. F100.;

G02 X30. Z0. I0. K-50.;

M30;

1.48.2.3 Example 2

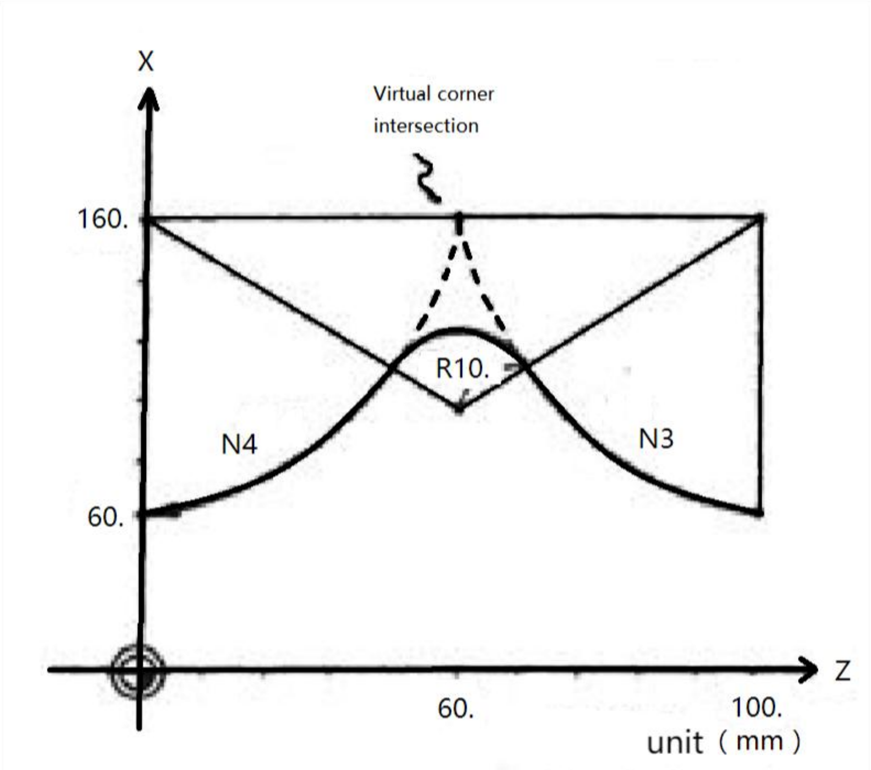


Figure 1.48-4

Circular Arc to Circular Arc (Radius Programming)

- **Incremental Programming**

G28;

G00 U30. W100.;

G02 U50. W-50. I50. K0. ,R10. F100.;

U-50. W-50. I0. K-50.;

M30;

1.48.3 Angle Function for Linear Movement (,A_)

Given the angle of the straight line and the endpoint coordinates of any one axis, the endpoint coordinates of the other axis are automatically calculated.

N1 G01 X(Z)__,A__

1.48.3.1 Instruction Format

The angle table is as follows: Starting from the positive direction of the first axis in the plane, counterclockwise is considered positive, and clockwise is considered negative.

The angle range is -360.000° to 360.000°. If it exceeds the range of 360.000°, it will be taken as the remainder after division by 360.000°.

1.48.3.2 Example

G00 X50.0 Z0;

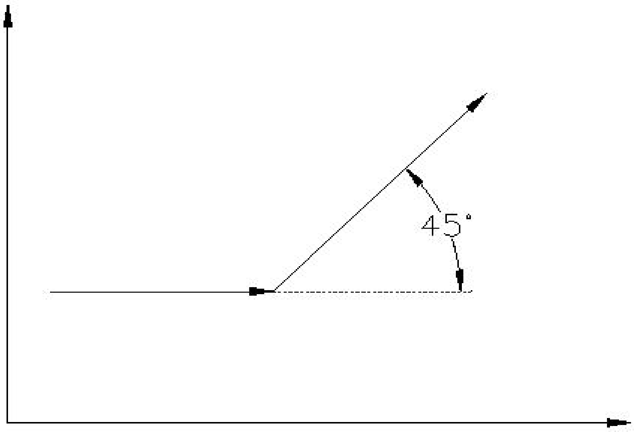


Figure 1.48-4

G01 X50.0 Z50.0 F100.;
G01 Z100.0 ,A45.0;
M30;

1.48.4 ,C, R, A Combination

When combining C, R, and A commands, you can execute operations involving corners, fillets, and angular adjustments in your machining process, simplifying your programming.

1.48.4.1 ,A→,A Combination

If it's challenging to determine the intersection point of two lines, the system can automatically decide the endpoint of the first line and control the movement path based on the slope of the first line, the absolute endpoint coordinates of the second line, and its slope.

- **Instruction Format**

  G01 ,A__

  G01 X__Z__,A__

- **Example**

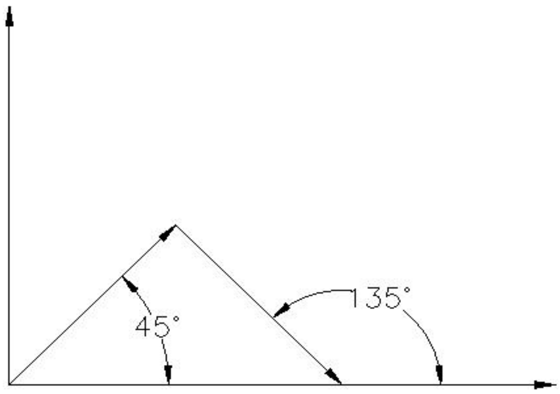


Figure 1.48-5

G00 X0 Z0;

G01 ,A45. F100.;

G01 Z90. X0 ,A135.;

M30;

Note:

1. An error is triggered when relative coordinates are used for the endpoint of the second segment.
2. An error occurs when there is no intersection point between the two lines or when the intersection angle is less than 1°.

1.48.4.2 ,A→,C Combination

- **Instruction Format**

G01 X__Z__,A__,C__

G0x X__Z__I__J__

Or

G01 X__Z_ ,A__,C__

G01 X__Z__,A__

- **Example**

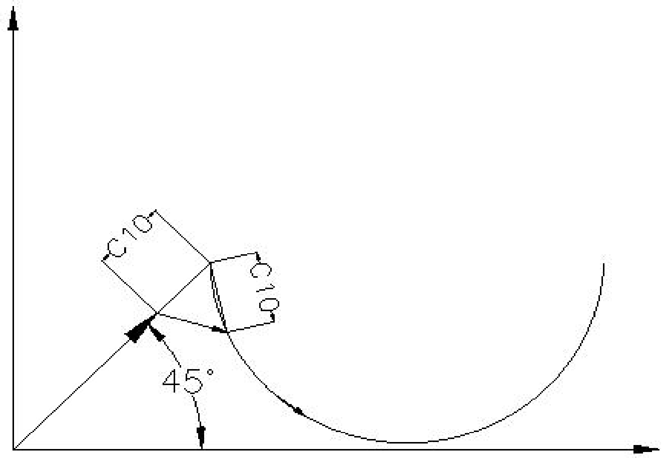


Figure 1.48-6

Radius Programming

G00 X0 Z0; //

G01 X50. ,A45. ,C10. F100.; //

G03 X50. Z150. R50.; //

M30; //

1.48.4.3 ,A→,R Combination

- **Instruction Format**

G01 X__Z__,A__,R__

G0x X__Z__I__J__

Or

G01 X__Z__,A__,R__

G01 X__Z__,A__

- **Example**

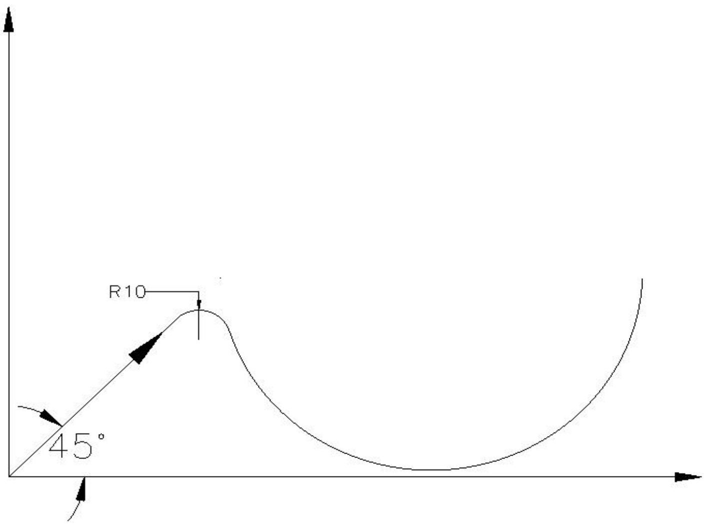


Figure 1.48-7

Radius Programming

G00 X0 Z0;

G01 X50. ,R10. ,A45. F100.;

G03 X50. Z150. R50.;

M30

1.49 Tool Compensation Instruction: T-code

Tool compensation, also known as T-code functionality, is used to specify the tool number and tool compensation identifier. Tool compensation includes tool length compensation and tool tip wear compensation.

1.49.1 Instruction Format

The T instruction format is specified through Mcm1705 and Mcm1706.

- Mcm1705: Specifies the number of digits reserved for the tool tip wear group, denoted as N.
- Mcm1706: Specifies whether tool length compensation and tool tip wear compensation are specified together.

1. When Mcm1706 = 0, both tool length compensation and tool tip wear compensation are specified using the tool number:

T * * * * * * $\underline{* *}$

N

Note: FINGER CNC can set up to 160 tool numbers in this mode. Therefore, there are only 3 valid digits for specifying T, and the value should not exceed 160.

2. When Mcm1706 = 1 and Mcm1705 = 0 or 2, tool length compensation and tool tip wear compensation are specified separately. The last 2 digits specify tool tip wear compensation, and the first M digits specify tool length compensation:

   T * * * $\underline{* * *}$ $\underline{* *}$

   M N

3. When Mcm1706 = 1 and Mcm1705 = 1, tool length compensation and tool tip wear compensation are specified separately. The last 1 digit specifies tool tip wear compensation, and the first M digits specify tool length compensation:

   T * * * * $\underline{* * *}$ $\underline{*}$

   M N

4. When Mcm1706 = 1 and Mcm1705 = 3, tool length compensation and tool tip wear compensation are specified separately. The last 3 digits specify tool tip wear compensation, and the first M digits specify tool length compensation.

   T * * $\underline{* * *}$ $\underline{* * *}$

   M N

1.49.2 Example

The relationship between T-codes and the workpiece coordinate system:

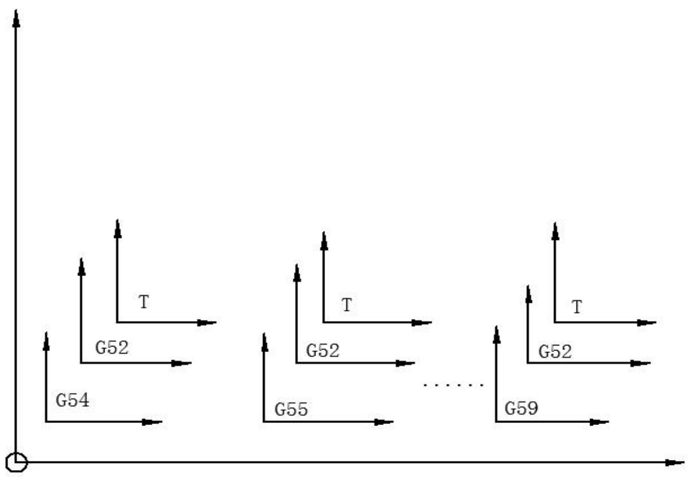


Figure 1.49-1

1.49.3 Supplement Explanation

1. FINGER CNC can have a maximum of 160 tool numbers, and each tool number can be divided into a maximum of 40 groups. Each group can correspond to a valid axis for a channel. The assignment of each group to axes is configured using Mcm2560 to Mcm2599. For example, if X, Y, and Z-Axis are designated as valid axes, and Mcm2560=2, Mcm2561=5, Mcm2562=7, then X, Y, and Z-Axis correspond to the 2nd, 5th, and 7th groups of each tool number.
2. The length compensation and tool tip wear compensation for each tool are configured using common variables Com10000 to Com22799.
3. The system generates an alarm if a tool number is configured beyond 160.
4. In multi-channel configurations, each channel can share the same tool number while executing different tool length and tool tip wear compensations.
5. Tool compensation can be implemented in three ways, determined by Mcm1707:
   - ❖ Mcm1707=0: Compensation based on movement instructions, i.e., compensation occurs when displacement instructions are present.
   - ❖ Mcm1707=1: Compensation based on coordinate offset, i.e., compensation occurs when executing T-codes.
   - ❖ Mcm1707=2: Tool tip wear compensation is applied when executing T-codes, and tool length compensation is applied when displacement instructions are present.

1.50 Main Spindle Speed Command: S-code Instruction

The S-code is used for specifying the spindle speed command, determining the spindle's revolutions per minute or constant surface speed.

1.50.1 Instruction Format

S__

S: Used to control the selected spindle's revolutions per minute (RPM). It is usually preceded by an M-code, such as M03, indicating the spindle action. The maximum allowable value is S999999.

Example: M03 S1000 specifies the first spindle to rotate at 1000 RPM.

1.50.2 Notes

When switching between different spindles during machining, if the current spindle is the second spindle and you want to specify the first spindle's forward rotation at 150 RPM, you should use M03 S150 to set the first spindle to rotate at 150 RPM per minute.

1.50.3 Example

M03 S150;   //Spindle maintained at 150 RPM.

1.51 Feed Instruction: F-code Instruction

When cutting a workpiece, the speed at which the reference point on the tool moves relative to the workpiece along the tool's trajectory is called the feed rate. The feed rate can be set using two methods: per minute feed (G98) and per revolution feed (G99). In the G98 mode, the feed rate can be directly specified as F300 for a tool with a feed rate of 300 mm/min. In G99 mode, F0.5 indicates a feed rate of 0.5 mm/rev.

1.51.1 Instruction Format

F__

Specifies the feed rate during machining.
Note: The F-code instruction is affected by 'Programming automatic decimal point.'

1.51.2 Example

G98 G01 X100.0 Y100.0 F300.; // Tool moves in a straight line, with a feed rate of 300 mm/min.
G99 G01 X100.0 Y100.0 F0.5; // Tool moves in a straight line, with a feed rate of 0.5

mm/rev."

Part 2.M-code Instructions Explanation

2.1 M-code Functions Table

M-Code	Function	M-Code	Function
00	Program Pause	44	Feeder Waiting for Material Change Completion
01	Select Stop	45	Enable Selective Skipping
02	Program End	46	Disable Selective Skipping
03	First Spindle Forward	47	Spindle and Chuck Independent Rotation
04	First Spindle Reverse	48	Spindle and Chuck Linked Rotation
05	First Spindle Stop	50	First Spindle Indexing Mode
08	Coolant On	51	First Spindle Rotation Mode
09	Coolant Off	55	Enable Anti-Crash
10	Spindle Clamp Released	60	Second Spindle Indexing Mode
11	Spindle Clamp Engaged	61	Second Spindle Rotation Mode
12	Tailstock Advance	63	Second Spindle Forward
13	Tailstock Retract	64	Second Spindle Reverse
15	Increment Machine Count	65	Second Spindle Stop
16	Reset Machine Count	70	Third Spindle Indexing Mode
17	Close Safety Door	71	Third Spindle Rotation Mode
18	Open Safety Door	73	Third Spindle Forward
19	Spindle Static/Dynamic Indexing	74	Third Spindle Reverse

20	Cancel M19	75	Third Spindle Stop
21	Air Blast On	80	Temporary Suppression of Home Return Alarm
22	Air Blast Off	84	Spindle Brake
26	Material Feeder On	85	Release Spindle Brake
27	Material Feeder Off	91	Single Block Jump (M91 Pxx)
29	Enable Spindle Rigid Tapping Mode	93	Forward Tool Comp (Polygon Cutting)
30	Program End	94	Reverse Tool Comp (Polygon Cutting)
31	Spindle Simulation Feedback On (For Pulse-Type Spindles Only)	95	Tool Comp Stop
32	Spindle Simulation Feedback Off	98	Call Subprogram
40	Chip Conveyor Forward	99	Return from Subprogram to Main Program Loop
41	Chip Conveyor Reverse	361	Set First Spindle as the Reference Spindle
42	Chip Conveyor Stop	362	Set Second Spindle as the Reference Spindle
43	Feeder Start	363	Set Third Spindle as the Reference Spindle

2.2 Program Pause (M00)

When the M00 command is executed, the machining program is paused.

2.3 Stop Selection (M01)

The M01 function is similar to M00 but is controlled by the "Stop Selection" switch. When the switch is ON, M01 is effective, causing the program to pause. If the switch is OFF, M01 is not effective.

2.4 Program End (M02)

If there is an M02 command at the end of the main program, when this command is executed, the machine will stop all actions. To restart the program, you must first press the "RESET" key and then press the "CYCST" start button for it to be effective.

2.5 First Spindle Clockwise/Counterclockwise (M03/M04)

M03 is the command for the first spindle to rotate clockwise and must be used in speed mode with the S command specifying the speed. M04 is the command for the first spindle to rotate counterclockwise, also used in speed mode with the S command specifying the speed.

2.6 First Spindle Stop (M05)

The M05 command stops the rotation of the main spindle.

2.7 Cutting Fluid On/Off (M08/M09)

The M08 command turns on the cutting fluid, while the M09 command turns it off.

2.8 Spindle Clamp Release/Clamp (M10/M11)

The M10 command releases the spindle clamp, and the M11 command clamps the spindle. The assignment of this M code can be configured in the system parameters for the clamp.

2.9 Tailstock In/Out (M12/M13)

The M12 command moves the tailstock in, and the M13 command moves it out.

2.10 Incremental Count +1 (M15)

The M15 command increments the machining count by 1 (seen on the machining monitoring page).

2.11 Reset Machining Count (M16)

The M16 command resets the machining count to zero (seen on the machining monitoring page).

2.12 Safety Door Open/Close (M17/M18)

The M17 command opens the safety door, and the M18 command closes it.

2.13 Spindle Static/Dynamic Indexing (M19/M20)

M19 R__S__

R: Static positioning angle of the spindle.

S: Static positioning speed of the spindle.

(This function needs to be redefined in custom jumps.) If R is not specified or has no value, it will result in an alarm.

Example: M19 R90S100 // Static positioning of the spindle to 90°.

M20 // Close M19.

2.14 Air Blow On/Off (M21/M22)

The M21 command turns on the air blow, and the M22 command turns it off.

2.15 Material Receiver On/Off (M26/M27)

The M26 command turns on the material receiver, and M27 turns it off.

2.16 Enable Spindle Rigid Tapping Mode (M29)

G84/G88 rigid tapping cycles require specifying rigid tapping. Refer to section 1.27 for details on G84/G88 tapping cycles.

2.17 Program End (M30)

The M30 command indicates the end of the program. When the program reaches the M30 command, all actions stop, and the execution cursor moves to the beginning of the program.

2.18 Chip Conveyor Forward/Reverse/Stop (M40/M41/M42)

M40 turns on the chip conveyor forward, M41 turns on the chip conveyor in reverse, and M42 stops the chip conveyor.

2.19 Feeder Start (M43)

The M43 command starts the feeder.

2.20 Feeder Wait for Material Change Signal (M44)

M44 is used to wait for the feeder to signal that a material change is complete. If the material change isn't completed, M44 will hold, and it will end when the material change is finished, allowing the program to proceed to the next line.

2.21 Enable/Disable Optional Stop Function (M45/M46)

M45 enables the optional stop function, allowing selective skipping of specific program segments marked with "/" or "/n" in the program. This feature allows one program to machine two or more different workpieces by creating a program with "/n" code.

- **Detailed Explanation**

1. Using "/" for Selective Skipping Function:

1) The use of "/" is specified to enable the selective skipping function. When "/" is

used at the beginning of a program segment, and the selective skipping function is active, the segment will be ignored and not executed. When the selective skipping function is turned off, the segment with "/" will be executed.

2) When "/" is used in the middle of a program segment, it is used for user macro program division operations.

❖ Incorrect: N20 G1 X25. /Z25.5. ; ... (This is a user macro program division command, and an error occurs at this point)

❖ Correct: /N20 G1 X25. Z25.25. ; ...

2. Using "/n" for Selective Skipping Function, n=1~9:

1) "/n" can be specified at the beginning or middle of a program segment (the parameter for "/n" to be valid in the middle of a segment needs to be enabled).

❖ Scenario I: "/n" specified at the beginning of a program segment, and when "/n" is valid, the single-line skipping function is enabled, and the program segment is ignored. Example: /2 N2 G0 X100. #S14999 BIT02=0, when "/2" is valid, the N2 program segment is ignored.

❖ Scenario II: "/n" specified in the middle of a program segment (parameter for "/n" to be valid in the middle of a segment needs to be enabled). Example: N2 G0 X100. /2 Z100. #S14999 BIT02=0, when "/2" is valid, the Z100 program segment is ignored.

2) Multiple "/n" can be specified within a program segment.

❖ In cases where multiple "/n" are specified at the beginning of a program segment, as long as any one of them is active, the program segment is ignored.

N01 M03 S1000

/1/2 N02 G00 X50.

/1/2 N03 G01 Z-20. F100.

```
/1/2 N04 G00 Z3.
/1/3 N05 G00 X30.
/1/3 N06 G01 Z-20. F100.
/1/3 N07 G00 Z3.
/2/3 N08 G00 X10.
/2/3 N09 G01 Z-20. F100.
/2/3 N10 G00 Z3.
N11 G28 X0.
N12 M30
```

- ❖ "/1" is active, "/2" and "/3" are inactive: Execution sequence: N01 → N08 → N09 → N10 → N11 → N12 program segments.
- ❖ "/2" is active, "/1" and "/3" are inactive: Execution sequence: N01 → N05 → N06 → N07 → N11 → N12 program segments.
- ❖ "/3" is active, "/1" and "/2" are inactive: Execution sequence: N01 → N02 → N03 → N04 → N11 → N12 program segments.

3) When multiple "/n" are specified within the same program segment, only the part specified by the active "/n" signals is ignored.

- ❖ "/1" is active, "/2" is inactive: Y1. /2 Z1. is ignored.
- ❖ "/2" is active, "/1" is inactive: Z1. is ignored.

Notes

1. Before prefetching, enabling the selective skipping function won't skip program segments loaded into the prefetch buffer.
2. When "/" or "/n" is specified at the beginning of a program segment, functions related to sequence numbers (Nxxxx) still remain active.

   Example:

N1 IF[#2 GT 100]GOTO 2;

#1=#1+#2;

#2=#2+2;

GOTO 1;

/1 N2 M98 P1;

When /1 is present, it still jumps to N2 M98 P1.

3. When "/" is used in the middle of a program segment and does not conform to the programming format, it triggers an alarm.
4. When the parameter is set to enable "/n" only at the beginning of a program segment, if customers insert "/n" in the middle of a program segment, it will be treated as a division operation.

2.22 Enable/Disable Independent Spindle Rotation and Chuck Action (M47/M48)

The M47 command enables independent spindle rotation and chuck action. When this function is activated, whether the chuck is loosened is independent of spindle rotation. Spindle rotation is also independent of whether the chuck is clamped. The M48 command disables independent spindle rotation and chuck action.

2.23 First Spindle Position Mode (M50)

Switches the first spindle to position mode, allowing indexing functions.

2.24 First Spindle Velocity Mode (M51)

Switches the first spindle to velocity mode, allowing velocity commands such as M3 Sxxx.

2.25 Enable Anti-Preload Function (M55)

When modifying tool compensation values during program execution, you need to prefix the tool number with M55 to ensure real-time tool compensation takes effect.

2.26 Second Spindle Position Mode (M60)

Switches the second spindle to position mode, allowing indexing functions.

2.27 Second Spindle Rotation Mode (M61)

Switches the second spindle to velocity mode, allowing velocity commands like M63 Sxxx.

2.28 Second Spindle Forward/Reverse (M63/M64)

M63 is the command for forward rotation of the second spindle. It requires velocity mode and an S command to specify the speed. M64 is the command for reverse rotation of the second spindle, also requiring velocity mode and an S command to specify the speed.

2.29 Stop Second Spindle (M65)

This command stops the rotation of the second spindle.

2.30 Third Spindle Position Mode (M70)

Switches the third spindle to position mode, allowing G00 B___ (degree) indexing.

2.31 Third Spindle Velocity Mode (M71)

Switches the third spindle to velocity mode, allowing velocity commands like M73 Sxxx.

2.32 Third Spindle Forward/Reverse (M73/M74)

M73 is the command for forward rotation of the third spindle. It requires velocity mode and an S command to specify the speed. M74 is the command for reverse rotation of the third spindle, also requiring velocity mode and an S command to specify the speed.

2.33 Stop Third Spindle (M75)

This command stops the rotation of the third spindle.

2.34 Temporarily Disable Homing Alarms (M80)

This command disables all axis homing alarms temporarily, allowing axes that were previously in a homing-required state to become ready.

2.35 First Spindle Brake/Release Brake (M84/M85)

M84 is the command for activating the main spindle brake output (similar to pressing the main spindle brake button on the auxiliary board), which halts the currently running program.

2.36 Single-Block Skip (M91) and Selective Jump (M92)

M91 Pxx is used to specify jumping to a particular program segment, which is defined within the program.

Example

```
G00 X10 Z0;        // Rapid positioning
G01 Z-30;          // G01 positioning
M91 P10;           // Single-block skip
G00 Z0;            // Rapid positioning
X0;                // Rapid positioning
N10 ;              // Block number
M30;               // Program ends
```

In this example, using N__ specifies the program location, and P__ takes the value of N. As shown in the example, during program execution, it will skip the block G00 Z0 X0.

2.36.1 Selective Jump Function M92

The format for M92 Pxxx Ixxx，

P specifies the N block to jump to.

I specifies the signal to check for jumping.

When the signal specified by I is detected (software input point), the program jumps to the N block specified by P; otherwise, it continues executing downward. The signal point "I" is represented as "-" for normally closed, and "I0" can only be normally open.

Example

G00 X10 Z0; //Rapid positioning

G01 Z-30; //G01 Positioning

M92 P10 I6; //Selective Jump, Program jumps when i-point 6 is detected

G00 Z0; //Rapid positioning

X0; //Rapid positioning

N10 ; //Block Number

G0 X100; //Rapid positioning

M30; //Rapid positioning

When the specified i-point signal 'I6' is detected, as shown in the example, the Program will skip the section 'G00 Z0 X0' and execute 'G0 X100'.

2.37 Polygon Cutting(Forward/Reverse Tool Compaction) (M93/M94)

M93 P___ Q___ R___ S___

M94 P___ Q___ R___ S___

Note:

When using the flying cutter function, set the reference axis and synchronized axis in the flying cutter function parameters. You can directly execute M93/M94 again to make changes when altering P/Q/R/S; there's no need to use M95 to desynchronize and stop the spindle.

P___: Number of cutting edges (decimal point omitted);

Q___: Number of edges in the shape (decimal point omitted);

S___: Active axis speed (decimal point omitted);

R___: Synchronization angle (decimal point omitted);

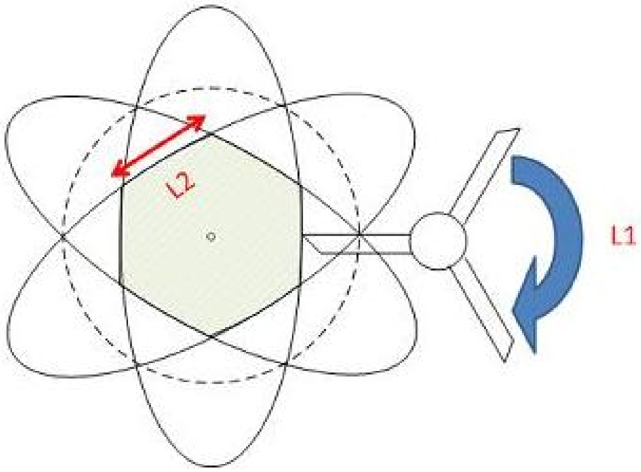


Figure 2.37-1

The flying cutter is a new machining process that utilizes the speed difference between the tool and the workpiece to rapidly machine polygonal workpieces. However, due to the physical cutting dynamics, changes in machining conditions can often affect the flatness of the machined surface, resulting in a non-flat surface. The following judgment criteria are primarily used to assess the flatness of the machined surface.

speed ratio i	Machining results by external cutting method		
>2	K>L	K=L	K<L
	Convex	plane	concave
	K = C/(i-1)^2		
=2	Convex		
<2	Convex		

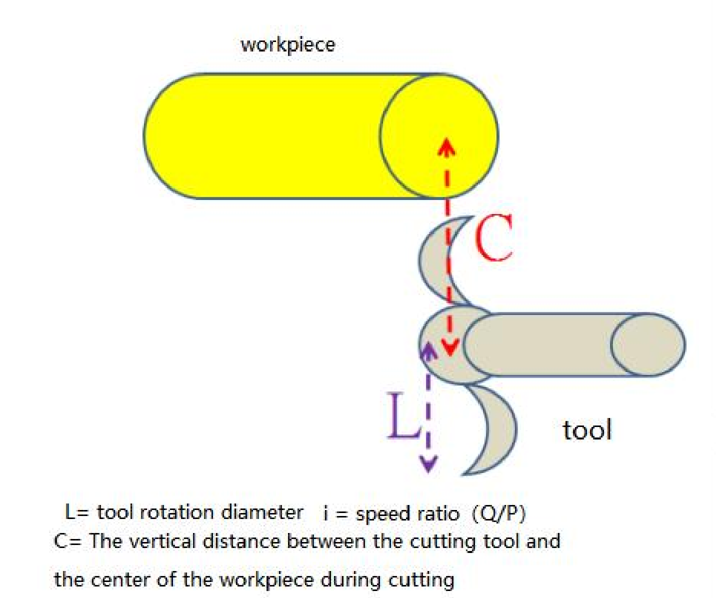


Figure 2.37-1

Example

When the flying cutter disk has only one blade, it flies symmetrically to both sides.

```
T0101;                      // Call tool #1
G0 X0 Z0 ;                  // Rapid positioning
M93 P1 Q4 S1000 R0. ;       // Active axis speed 1000, Slave axis 4000,
                               Synchronization angle 0 degrees
G01 Z100.;                  // Machining in the Z-Axis direction to 100
X10.;                       // Lift the tool
M95 ;                       // Stop the flying cutter function
M30 ;                       // End of program
```

2.38 Flying Cutter Function Stop (M95)

Stops the flying cutter function.

2.39 Call Subprogram / Return from Subprogram (M98 / M99)

2.39.1 M98 Call Subprogram

2.39.1.1 Instruction Format

M98 P____H____L____D____

M98 <File Name> H____L____D____

P: The file name of the subprogram to be called (P takes precedence over L).

<File Name>:

1. The file name of the subprogram to be called, consisting of <File Name> + the file extension ".CNC".
2. The file name can be composed of characters or numbers, up to a maximum of 32 characters. For example: M98 <AA123> calls the AA123.CNC subprogram; M98 <TEST.CN> calls the TEST.CNC subprogram.
   - ❖ H: The sequence number (Nxxxxx) within the subprogram to start from. Maximum 5 digits. When omitted, it starts from the beginning of the subprogram.
   - ❖ L: The number of times the subprogram should be executed. If unspecified or L0, it indicates a single execution.
   - ❖ D: The folder containing the subprogram to be called, ranging from 0 to 10. If D=0 or omitted, it calls the subprogram from the current folder (this feature is primarily for developers, and users can typically omit D).

2.39.1.2 Notes

1. M98 P123 calls the 123.CNC subprogram.
2. M98 P0123 calls the 123.CNC subprogram.
3. M98 P0 calls the 0.CNC subprogram.
4. When both P and <File Name> are omitted, it calls the current program.

2.39.2 M99 Subprogram End and Return / Main Program Loop

2.39.2.1 Instruction Format

M99 P__

M99: Subprogram return command.

P: Sequence number of the program to return to. Example: M99 P200 would return to program line N200.

2.39.2.2 Notes

When using M99 P__, the main program returns to the program line with the corresponding number, and the subprogram returns to the program line with the corresponding number of the previous program.

2.39.3 Example

O000.CNC Program (Main Program)

```
N1 G98;                          //Switch to per minute feed
N2 G00 X10.0 Z10.0;              //Rapid positioning
N3 G01 U10.0 W10.0 F1000.;       //G01 positioning
N4 M98 <O001> L3 H3;             //Jump to subprogram
N5 G00 X-100.0 Z-100.0;          //Rapid positioning
M30;                             //End of program
```

O001.CNCProgram（Subprogram）

```
N1 G00 Y0;                       //Rapid positioning
N2 G00 Y100.0;                   //Rapid positioning
N3 G00 Y-100.0;                  //Rapid positioning
N4 G00 V10.0;                    //Rapid positioning
M99 P5;                          //Return to main program line N5
```

When executing the O000.CNC program, it jumps to the subprogram O001.CNC at line N4 and continues executing downward. Since the subprogram ends with M99 P5 instead of M99, it doesn't execute the O001.CNC program three times but instead jumps directly to line N5 of the main program.

2.39.4 Main Program Loop (M99)

In the main program, adding M99 at the end of the program will cause the program to jump back to the program header and start executing again when it reaches M99.

2.40 Spindle Speed Control (M140-M144)

M140: Spindle at neutral position (idle).

M141: Spindle in first gear.

M142: Spindle in second gear.

M143: Spindle in third gear.

M144: Spindle in fourth gear.

2.41 External Spindle Precise Stop (M505)

When the spindle stops, it simultaneously sends an O-point signal to the drive, allowing the drive to perform a precise stop based on this signal.

2.42 First Spindle as Reference Spindle (M361)

Set the first spindle as the reference spindle (per revolution feed reference).

2.43 Second Spindle as Reference Spindle (M362)

Set the second spindle as the reference spindle (per revolution feed reference).

2.44 Third Spindle as Reference Spindle (M363)

Set the third spindle as the reference spindle (per revolution feed reference).

Part 3. Appendix

3.1 Manual Operation Instructions

3.1.1 Returning to Machine Zero Point

Every time the controller is powered on, it's necessary to perform a return to the machine zero point to establish the reference position for all coordinate settings during machining.

1. Use the axis buttons such as 1 , 2 , 3 , 4 , 5 , and 6 to control the machine to move towards the zero point position in the respective axis.
2. After the axis has completed the return to the zero point movement, the system will reset the machine coordinates for that axis to zero.
3. You can configure parameters to set the return to zero point method, direction, and speed, among other settings.

3.1.2 Selective Skipping

Selective skipping is set using the "/" character in the program. When the selective skipping function is enabled and a program block begins with "/", the content of that block is skipped, and the system continues executing the next block without "/" in it.

1. The program block contains the "/" instruction.
2. Press the "Selective Skip" button to activate the selective skipping function.
3. Press the "Auto Execute" and "Start" buttons to initiate program execution.
4. When the program encounters a block starting with "/", it skips that block and proceeds to execute the next block without "/" in it.

3.1.3 Single Block Execution in Auto Mode

The single block execution function allows operators to execute machining step by step, following the order of program blocks.

1. Select the machining program you want to execute.
2. Switch to "Auto Execute" mode.
3. Press the "Single Block Execution" button to activate single block execution.
4. Press the "Start" button, and the controller will execute the first program block. Pressing "Start" again will execute the next block, and so on, until the program is completed.

3.1.4 Program Restart

Program restart is used to specify a particular block within a program as the restart block. During machining, you can start execution from the specified block.

1. Switch the mode to "Auto Execute" mode.
2. In the "Processing Information" section, input the block number for restart.
3. Press the F6 "Restart" button. When the restart block is successfully identified, the system will automatically switch to "MDI" mode. At this point, press the "Start" button to run the content currently in the MDI.
4. Switch back to "Auto Execute" mode.
5. Press the "Start" button to begin execution from the specified block.

3.1.5 Selective Stop Function

When the selective stop function is active, if an M01 instruction is encountered in the program, the program will pause until the "Start" button is pressed again to continue to the next block.

1. The program includes an M01 instruction.
2. Press the "M01 Stop" button to activate the selective stop function.
3. Press the "Auto Execute" and "Start" buttons to initiate program execution.
4. When the program encounters an M01 or M1 instruction, it will pause.
5. Press the "Start" button to resume program execution.

3.1.6 MPG Test Program Mode

The MPG test program mode is suitable for use before mass-producing finished products. Customers can use this mode to create actual samples. If there are errors in the program or differences in the samples, customers can easily modify and test the program until the produced samples meet the requirements. Additionally, during testing, it can effectively prevent collisions.

1. Press the "Program Preview" button to enable the MPG test program mode.
2. Press the "Auto Execute" and "Start" buttons to execute the program. At this point, the program will not start immediately.
3. Choose the MPG pulse multiplier to determine the feed rate.
4. Now, rotate the MPG to start executing the test program. When you stop turning the MPG, the feed will also stop. If you continue turning the MPG, the program will resume execution.
5. If you want to cancel the MPG test program mode at this point, the program will continue at normal speed.

3.2 System Alarm Handling

To prevent incorrect operations that could pose risks to individuals and the machine, the system and PLC have numerous safeguards. When these protective conditions are triggered, the system will issue warnings or alarms to alert users.

3.2.1 Emergency Stop

In case of machine malfunction or unexpected actions that may pose a safety risk to individuals or the machine, pressing the "Emergency Stop" button will immediately halt the machine's operations. This button is typically locked in place after being pressed, and to release the lock, you usually need to rotate the button. It serves to cut off the machine's actions, and before unlocking it, any issues or malfunctions must be resolved.

3.2.2 Alarm Display

Alarms are divided into "Current Alarms" and "Alarm History."

3.2.2.1 Current Alarms

1. The current alarm status of the system.
2. When an alarm occurs, the controller will display the current alarm content in a pop-up window.
3. Clicking the leftmost button "<" can cancel the pop-up window.
4. If the alarm has not been cleared, clicking "Reset" will bring up the alarm window again.
5. Switching to the "Alarms" page will automatically display the current alarms.

3.2.2.2 Historical Alarms

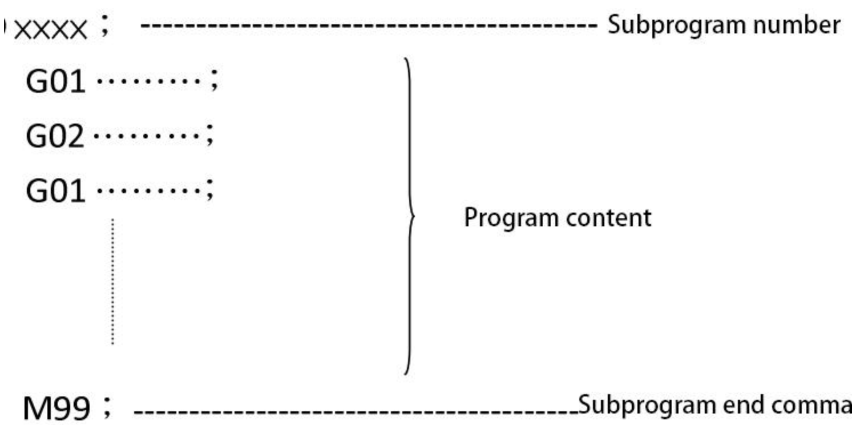
1. This page allows you to determine the possible causes of alarms that the system has

previously experienced.

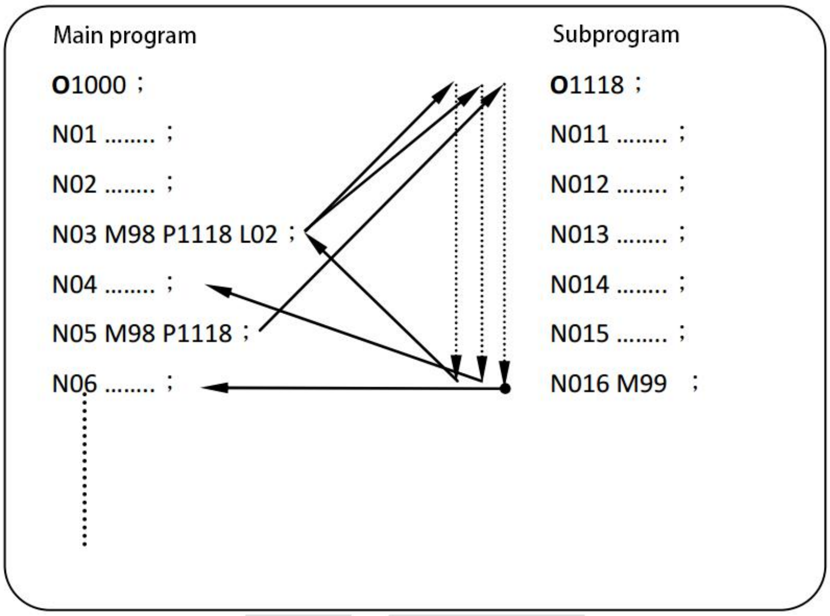
2. Switch to the [Alarm] page and click on [Historical Alarms] to display the historical alarms.
3. When multiple alarms occur, they are listed in descending order based on the time of occurrence, from top to bottom.

3.3 Creating and Executing Subprograms

3.3.1 The program format for subprograms is as follows.



3.3.2 Execution sequence of the main program in conjunction with subprogram call commands.



Guangzhou Finger Technology Co.,Ltd

Hotline: 020-39389901

Repair Helpline: 18127931302

Fax: 020-39389903

Postal Code: 511495

E-mail: finger@fingercnc.com

Website: www.finger-cnc.com

Address: 201,No. 8, Chengding Street, Zhongcun Street, Panyu District, Guangzhou City, Guangdong Province

Official Website

Official Wechat