SYNTEC TECHNOLOGY CO.,LTD.

Lathe Machine Program Manual (C Type)

匯出日期：2025-09-28
修改日期：2025-05-20

1 G Code Command Table of Lathe Machine.

Name Definition

Description	Definition	Example
Modal G Code	The following two conditions can be considered as modal G codes: 1. The G code function continues to be valid until the G code is turned off, and the axial index of the subsequent block is affected by this G code command. 2. The G code function continues to be valid before executing other G codes of the same group. The axial parameter of the subsequent block is affected by this G code command.	1. e.g. G43、G44、G49 After executing G43 and G44, if G49 is not commanded, the tool length compensation will continue to be effective. 2. e.g. G00、G01 **Machining Example** G00 X0. // Execute G00 Y0. // Execute G00 G01 X100. // Execute G01 Y100. // Execute G01
Non-Modal G Code	The G code command is valid only in a single block, and the axial argument is only affected by the block in the G code command.	e.g. G04 **Machining Example** G01 X1. // Execute G01 G04 X3. // Execute G04 X5. // Execute G01

* Modal G code displayed in green background

G Code Name	G code Type			Group	Dismiss Command
	Type A	Type B	Type C		
Rapid linear positioning (Rapid traverse)	G00	G00	G00	Interpolation mode	Execute other G code in the same group.
Linear interpolation	G01	G01	G01		
Arc interpolation（clockwise/ counterclockwise）	G02/G03	G02/G03	G02/G03		
Ellipse interpolation	G02.1/ G03.1	G02.1/ G03.1	G02.1/ G03.1		

G Code Name	G code Type			Group	Dismiss Command
	Type A	Type B	Type C		
Parabola interpolation	G02.2/ G03.2	G02.2/ G03.2	G02.2/ G03.2		
Dwell	G04	G04	G04		NA
Path synchronization	G04.1	G04.1	G04.1		
Cylinder interpolation	G07.1	G07.1	G07.1		G07.1 C0
Exact stop check	G09	G09	G09		NA
Programmable data input	G10	G10	G10		
Diameter/radius positioning switch	G10.9	G10.9	G10.9		
Enable polar coordinates interpolation	G12.1	G12.1	G12.1	Polar interpolation mode	G13.1
Disable polar coordinates interpolation	G13.1	G13.1	G13.1		
X-Y plane selection	G17	G17	G17	Working plane mode	Execute other G code in the same group.
Z-X plane selection	G18	G18	G18		
Y-Z plane selection	G19	G19	G19		
Process with imperial system	G20	G20	G70	Imperial, Metric Input dimension mode	Execute other G code in the same group.
Process with metric system	G21	G21	G71		
Enable second software stroke limit	G22	G22	G22	Stroke check mode	G23
Disable second software stroke limit	G23	G23	G23		

G Code Name	G code Type			Group	Dismiss Command
	Type A	Type B	Type C		
Return to reference position	G28	G28	G28		NA
Return from reference position	G29	G29	G29		
Return from any reference position	G30	G30	G30		
Skip function	G31	G31	G31		
Multi-Axis Multi-Signal Skip Function	G31.10/ G31.11	G31.10/ G31.11	G31.10/ G31.11		
Thread cutting	G32	G33	G33	Interpolation mode	Execute other G code in the same group.
Variable pitch thread cutting	G34	G34	G34		
Cancel tool nose radius compensation	G40	G40	G40	Cutter compensation mode	G40
Tool nose radius compensation (left)	G41	G41	G41		
Tool nose radius compensation (right)	G42	G42	G42		
Coordinate system setting / spindle maximum RPM limit	G50	G92	G92	Coordinate system setting mode	Execute other G code in the same group.
Disable Proportional Function	NA	G50	G50	Scaling mode	G50
Enable Proportional Function	NA	G51	G51		
Disable polygon cutting	G50.2	G50.2	G50.2		G50.2
Enable polygon cutting	G51.2	G51.2	G51.2		

G Code Name	G code Type			Group	Dismiss Command
	Type A	Type B	Type C		
Local coordinate system setting	G52	G52	G52		G52 X0.0 Z0.0
Axis removal	G52.1	G52.1	G52.1	Axis removal/ Axis borrowing function	G52.1
Axis borrowing	G52.2	G52.2	G52.2		
Machine coordinate system positioning	G53	G53	G53		NA
Workpiece coordinate system selection	G54~G59.9	G54~G59.9	G54~G59.9		Pr3229
Enable/Disable Workpiece Coordinate Rotation Functionality	G54.4	G54.4	G54.4		Pr3229、G54.4 P0
Exactly stop check	G61	G61	G61	Cutting feed control mode	Execute other G code in the same group.
Curved surface cutting mode	G62	G62	G62		
Tapping mode	G63	G63	G63		
Cutting mode	G64	G64	G64		
Single Marco call	G65	G65	G65		NA
Call Marco modal program	G66	G66	G66		G67
Cancel Marco modal program	G67	G67	G67		
Enable turret mirror function	G68	G68	G68		G69
Tilted plane machining	G68.2	G68.2	G68.2		

G Code Name	G code Type			Group	Dismiss Command
	Type A	Type B	Type C		
Disable turret mirror function Disable tilted plane machining	G69	G69	G69		
Fine cutting cycle	G70	G70	G72		Q argument specifies the ending block.
Lateral rough turning cycle	G71	G71	G73		
Radial rough facing cycle	G72	G72	G74		
Contour rough turning cycle	G73	G73	G75		
End face (Z axis) peck drilling cycle	G74	G74	G76		NA
Lateral (X axis) peck grooving cycle	G75	G75	G77		
Complex threading cycle	G76	G76	G78		
Complex mid-section threading cycle	G76.2	G76.2	G78.2		

G Code Name	G code Type			Group	Dismiss Command
	Type A	Type B	Type C		

Disable drilling cycle	G80	G80	G80	Interpolation mode	1. Execute G80
					Description : After dismissal, return to the previous interpolation mode.
					Example
					e.g.(C-type)
					G00
					G83 X5. Y5. Z-10. R-5. Q3.;
					X15.; // Execute G83
					G80;
					Y15.; // Execute G00
					2. Execute other G Codes
					in interpolation mode.
					Description : After dismissal, change into the commanded interpolation mode.
					Example
					e.g.(C-type)
					G00
					G83 X5. Y5. Z-10. R-5. Q3.;
					X15.; // Execute G83
					G01;
					Y15.; // Execute G01

Face drilling cycle	G83	G83	G83		
End face (Z axis) tapping cycle	G84	G84	G84		
Face boring cycle	G85	G85	G85		
Side drilling cycle	G87	G87	G87		
Side (X axis) tapping cycle	G88	G88	G88		
Side boring cycle	G89	G89	G89		

G Code Name	G code Type			Group	Dismiss Command
	Type A	Type B	Type C		

Outer/inner surface turning cycle	G90	G77	G20	Interpolation mode	Execute other G Codes in interpolation mode.
					Description : After dismissal, change into the commanded interpolation mode.
					Example
					e.g.(C-type)
					G00 X60.0 Z65.0;
					G20 X53.0 Z15.0 R-7.5 F0.6; // Execute G20
					X48.0; // Execute G20, second cycle
					G01;
					X42.0; // Execute G01
					X35.0; // Execute G01

Threading cycle	G92	G78	G21		
Reset absolute coordinate system	G92.1	G92.1	G92.1	Coordinate system setting mode	Execute other G code in the same group.

Mid-section threading cycle	G92.2	G78.2	G21.2	Interpolation mode	Execute other G Codes in interpolation mode.
					Description : After dismissal, change into the commanded interpolation mode.
					Example
					e.g.(C-type)
					G00 X50.0 Z55.0;
					G21.2 X39.0 Z15.0 R-10.0 F2.5; // Execute G21.2, first cycle
					X38.3; // Execute G21.2, second cycle
					X37.7; // Execute G21.2, third cycle
					G01;
					X37.3; // Execute G01
					X36.9; // Execute G01
					X36.75; // Execute G01

G Code Name	**G code Type**			**Group**	**Dismiss Command**
	Type A	**Type B**	**Type C**		
Inverse time feed	G93	G93	G93	Feed mode	Execute other G code in the same group.

End face cutting cycle	G94	G79	G24	Interpolation mode	Execute other G Codes in interpolation mode.
					Description : After dismissal, change into the commanded interpolation mode.
					Example
					e.g.(C-type)
					G00 X52.0 Z35.0;
					G24 X20.0 Z32.0 R-10.0 F0.6;// Execute G24
					Z28.0; // Execute G24
					G01;
					Z24.0; // Execute G01
					Z20.0; // Execute G01

Constant surface speed control	G96	G96	G96	Spindle speed mode	G97
Disable constant surface speed control	G97	G97	G97		
Feedrate per minute（mm/min）	G98	G94	G94	Feed mode	Execute other G code in the same group.
Feedrate per revolution（mm/rev）	G99	G95	G95		
Absolute command	NA	G90	G90	Abs, Inc Input command mode	Execute other G code in the same group.
Increment command	NA	G91	G91		
Return to initial point	NA	G98	G98		Execute other G code in the same group.
Return to R point	NA	G99	G99		

G Code Name	G code Type			Group	Dismiss Command
	Type A	Type B	Type C		
Cancel spindle synchronization/bearing function	G113	G113	G113		G113
Spindle synchronization function	G114.1	G114.1	G114.1		
Spindle bearing function	G114.3	G114.3	G114.3		

Note: NA means this function is not provided.

2 G Code Command Description (C-Type)

2.1 G00- Rapid Linear Positioning/Rapid traverse (C-Type)

2.1.1 Command Form

G00 [P1] X(U)__ Z(W)__ [F1=_];

X, Z: Target position (absolute value)

U, W: Target position (incremental value)

P1: active constant speed command

F1: feedrate mm/min or inch/min

Q：Overlapping distance

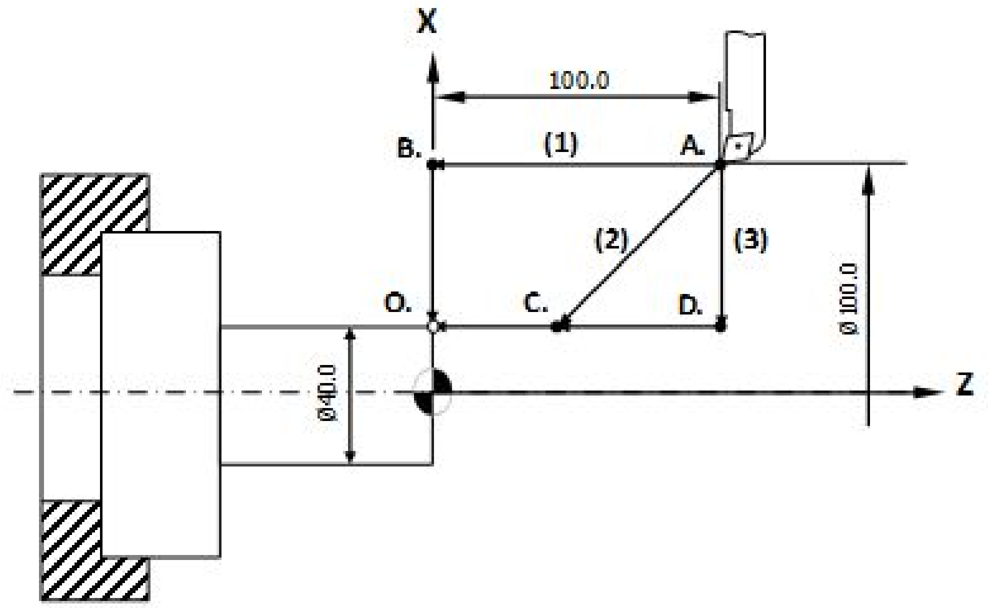
2.1.2 Description

G00 command is a rapid, point-to-point movement at a fast traverse rate without any cutting action, its main purpose is to save movement time when no cutting is needed. In Lathe program, it is usually used to move from machine zero point to cutting start point, or from cutting end point to machine zero point. In absolute mode (G90), tool moves to specified position in coordinate system; in increment mode(G91), tool moves from current position by specified distance.

2.1.3 Notice

1. The movement mode can decide by parameter #411 (0: linear, 1: each axle move in max speed independently)
2. Constant feedrate is legally with both P1 and F1 argument are determined. If P1 argument is missing, system wouldn't refer F1 argument as moving speed. Moving speed is refer to Pr461~Pr480.
3. Pr411=0, Resultant speed is limited by F1 and Pr461~Pr480 if executing with P1, F1. Each axis speed is limited by Pr461~Pr480.
4. R18 Rapid Traverse Override is still valid with P1 F1=_.
   The actually highest speed wouldn't exceed Pr461~Pr480 or F1, while the override is 0~100%.
   The actually highest speed wouldn't exceed Pr461~Pr480 over five times, while the override is F0(PR3207=2, R18=1).
5. G00 command doesn't support G10 L1100 P1002 R_.
6. G70/G71 supported unit of F1 can be mm/min or inch/min.
7. Unit of F1 can be mm/min, but always remains G94. G93/G95 is invalid.
8. F1 argument support version: 10.118.12 and after.
9. When using the G00 command for axial overlapping, the front and rear blocks need to be all G00 commands or G00 command followed by G01 command or G01 command followed by G00 command.

2.1.4 **Example**



There are several ways to move tool from point A to point O. Below are three program mode examples for three different movements.

Absolute Mode

G00 Z0.0 // A. → B.

X40.0 // B. → O.

G00 X40.0 Z0.0 //A. → C. → O.

G00 X40.0 //A. → D.

Z0.0 //D. → C. → O.

Incremental Mode

G00 W-100.0 //A. → B.

U-60.0 // B. → O.

G00 U-60.0 W-100.0 //A. → C. → O.

G00 U-60.0 //A. → D.

W-100.0 //D. → C. → O.

Absolute and Increment Mode Combined

G00 Z0.0; or G00 W-100.0;

U-60.0; or X40.0;

G00 X40.0; or U-60.0;

W-100.0; or Z0.0

G00 X40.0 W-100.0; or G00 U-60.0 Z0.0;

2.2 G01- Linear Cutting (C-Type)

2.2.1 Command Form

G01 X(U)___ Z(W)___ F___;

X ,Z: Specified position (absolute value)

U ,W: Specified position (increment value)

F: Feedrate
Unit under G94 mode: mm/min (inch/min)
Unit under G95 mode: mm/rev (inch/rev)

System default value: G95

2.2.2 Description

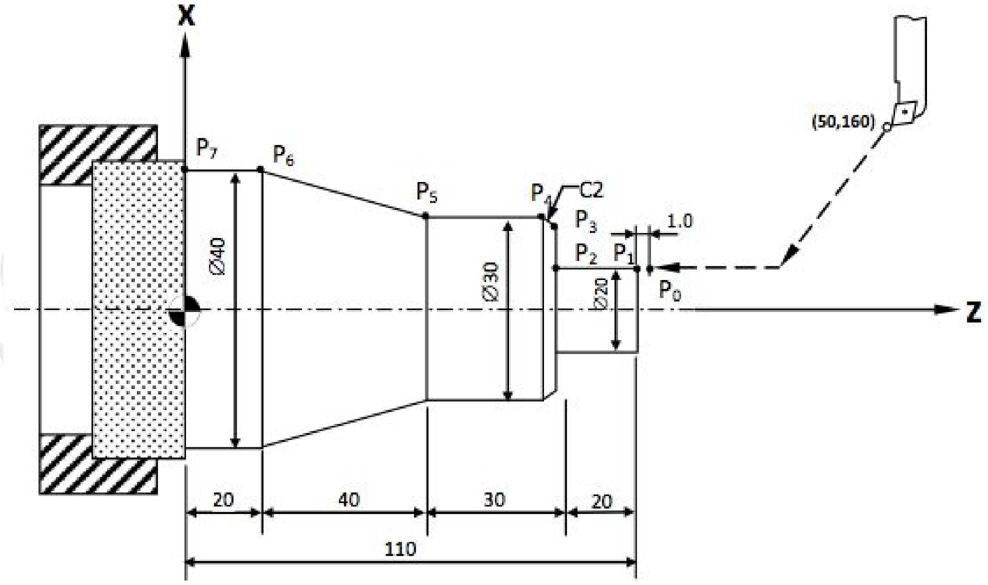
G01 executes linear interpolation, moves from current position to specified position with feed rate defined by F value.

It can process: outer (inner) diameter, end face, outer (inner) taper, outer (inner) groove, chamfer, ...etc.

2.2.3 Note

- The max. feed rate of G01 is defined by Pr405 maximum cutting feed rate or (Pr621~Pr636) axis maximum cutting feed rate.
- Default feed rate under G94 mode: 1000 mm/min (inch/min); default feed rate under G95 mode: 1.0 mm/rev (inch/rev).
- Default feed rate of G94/G95 can be changed in parameter Pr3836 (reboot controller to activate setting).

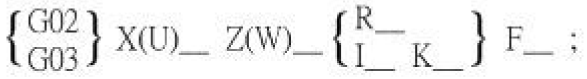
2.2.4 Example



```
G92 X50.0 Z160.0 S10000; // reset program zero point, spindle max 10000 rpm
T01;                    // use tool NO. 1
G96 S130 M03;             // constant surface speed, surface speed=130 m/min, spindle rotate CW
M08;                    //cutting liquid ON
G00 X20.0 Z111.0;         //positioning to specified point P0
G01 Z90.0 F0.6;           //linear interpolation P0-->P2
X26.0;                  //P2-->P3
X30.0 Z88.0;              //P3--> P4
Z60.0;                  //P4-->P5
X40.0 Z20.0;              //P5-->P6
Z0.0;                   //P6-->P7
G00 X50.0;                //return the tool
Z160.0;                   //return to zero point
M05 M09;                 //spindle stops, cutting liquid OFF
M30;                    //program end
```

2.3 G02/G03- Arc Interpolation (C-Type)

2.3.1 **Command Form**



$$\begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X(U)\_\ Z(W)\_ \begin{Bmatrix} R\_ \\ I\_\ K\_ \end{Bmatrix} F\_\ ;$$

G02: Specify the tool to make a clockwise arc interpolation

G03: Specify the tool to make a counterclockwise arc interpolation

X(U), Z(W): End point of the arc

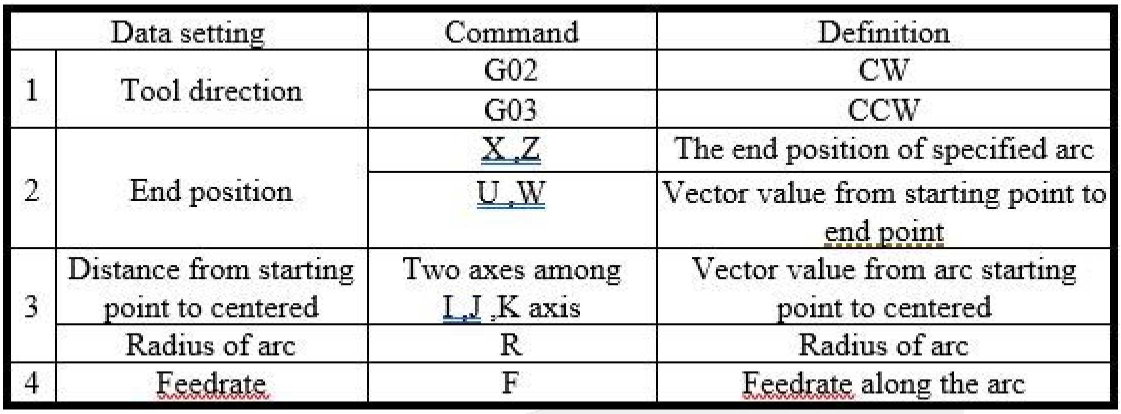
R: Radius of arc (under 180°)

I, K: X or Z direction distance from arc starting point to the center of arc (circle). Positive or negative is determined by the direction.

F: Feed rate of cutting

2.3.2 **Description**

The G02 ,G03 command of lathe numerical tool machine will move a specified tool along a circular arc on XZ plane. The setting of the parameters is given in the following table:



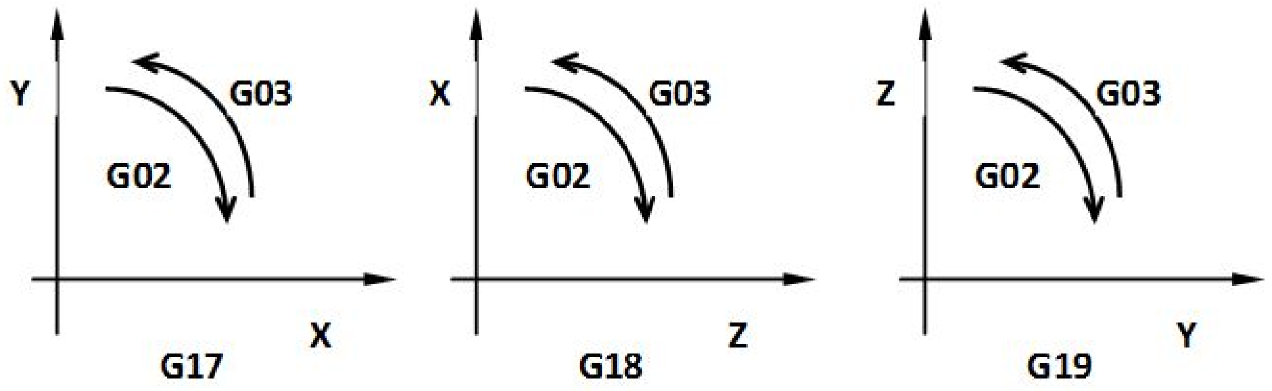
Data setting		Command	Definition
1	Tool direction	G02	CW
		G03	CCW
2	End position	X ,Z	The end position of specified arc
		U ,W	Vector value from starting point to end point
3	Distance from starting point to centered	Two axes among I ,J ,K axis	Vector value from arc starting point to centered
	Radius of arc	R	Radius of arc
4	Feedrate	F	Feedrate along the arc

Setting		Command	Definition
1	Tool direction	G02	Clockwise
		G03	Counter-clockwise
2	End point position	X, Z	The end point position of specified arc
		U, W	Vector value from start point to end point of the arc
3	Distance from start point to arc center	Two axes among I, J, K	Vector value from start point to center of the arc
	Arc radius	R	Arc radius
4	Feed rate	F	Tool move rate along the arc

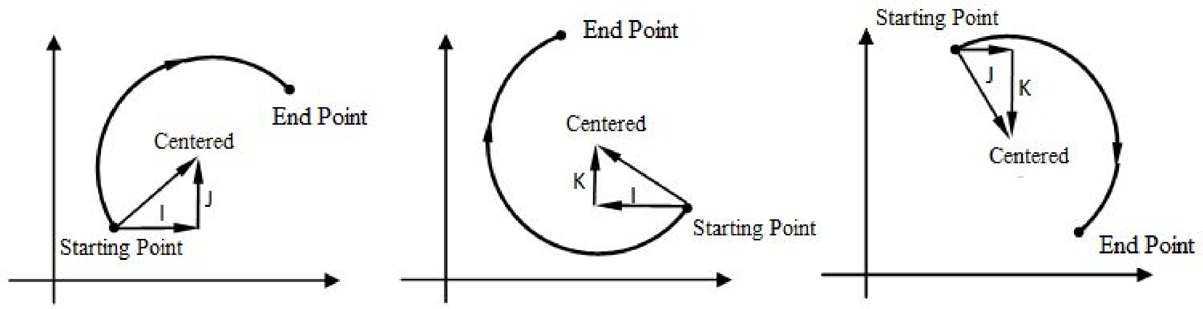
Illustration

1. G02/G03 Defined Arc

- Programming of XYZ/UVW arc coordinates will be affected by the diameter/radius position control setting.



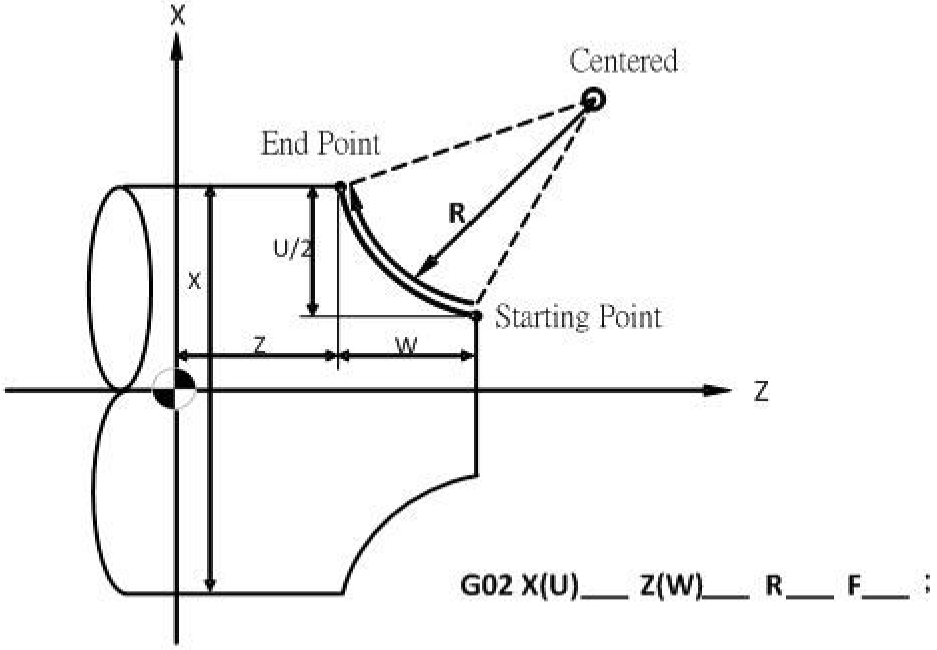
2. I, J, K Defined Arc
   - Vector IJK from arc starting point to arc center won't be affected by diameter/radius position control setting.



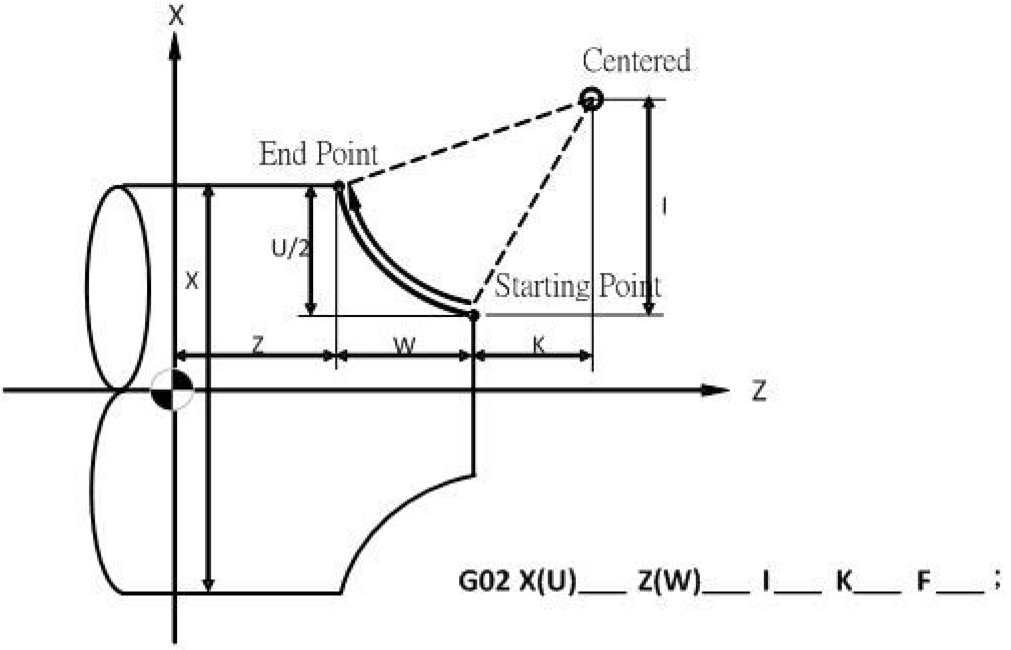
a. Arc on X-Y plane b. Arc on Z-X plane c. Arc on Y-Z plane

Parameter Setting Example

1. G02 Arc Interpolation
   a. Use R value

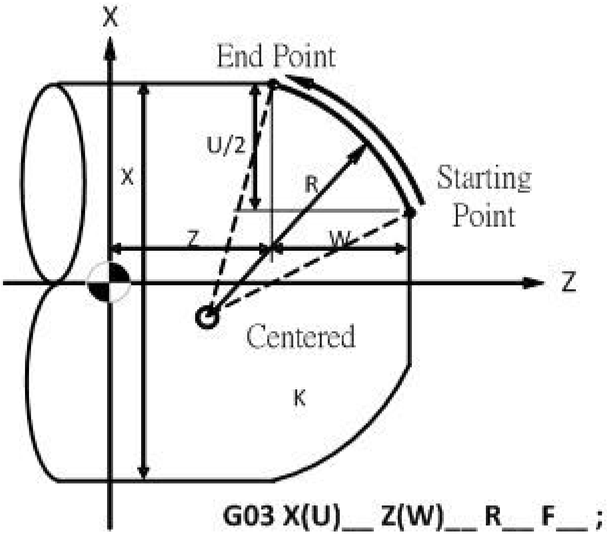


b. Use I, K

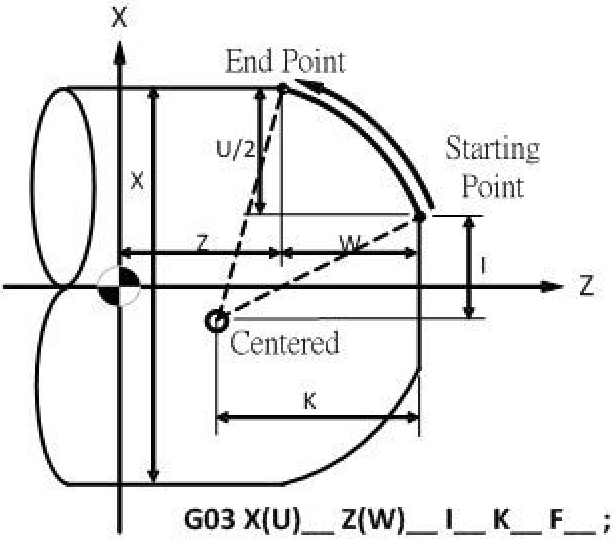


2. G03 Arc Interpolation

a. Use R value



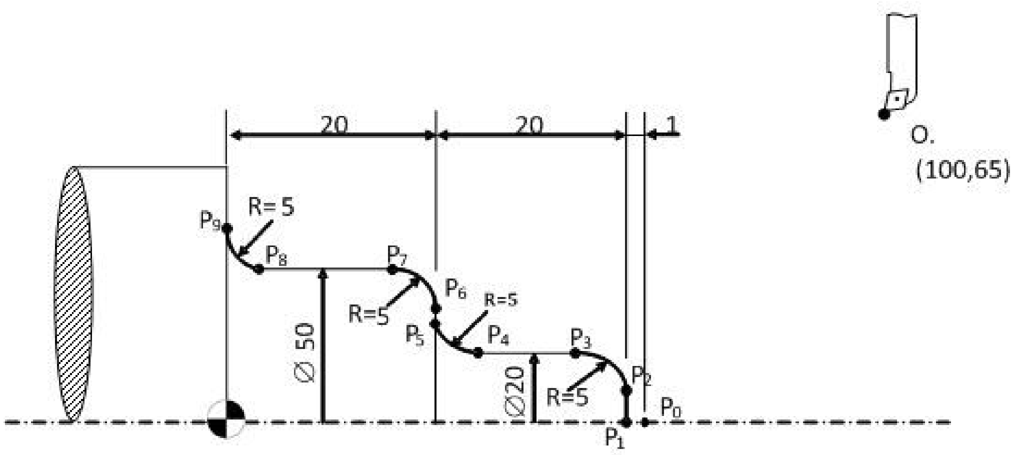
b. Use I ,K



2.3.3 Precautions

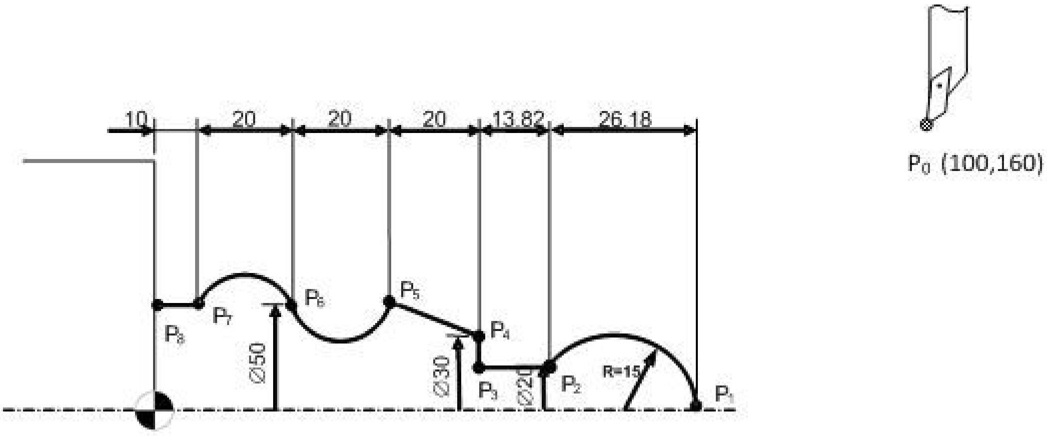
1. G02/G03 without any R, I, J, K will be executed as G01.
2. G02/G03 given improper X, Z, I, K, and R values will trigger COR-008 [Arc End Not on Arc] alarm. The alarm can be adjusted by Pr3807.
3. XYZ/UVW coordinate programming will be affected by diameter/radius position control setting.
4. IKJ vector programming will not be affected by diameter/radius position control setting.

Example 1



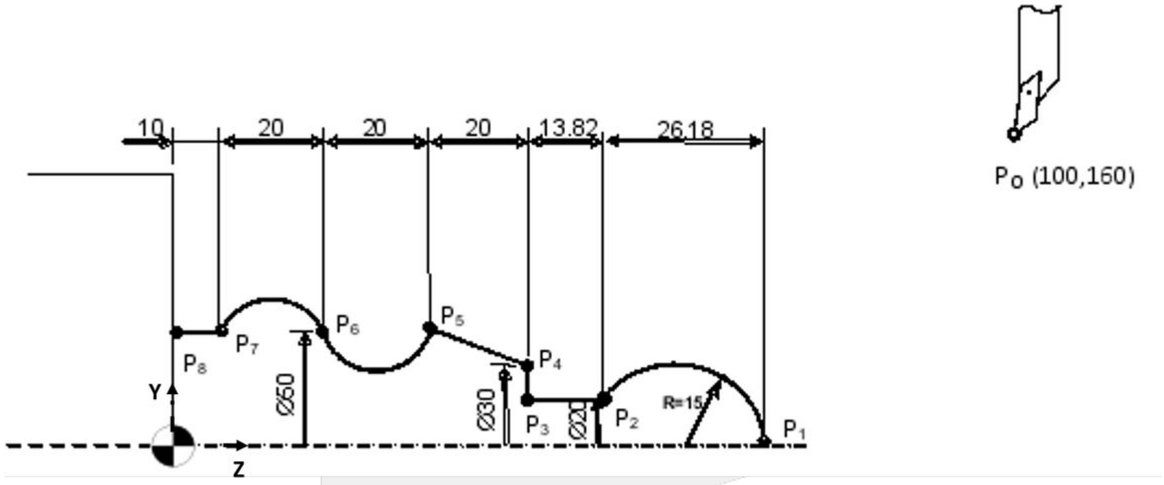
```
T01;                    //use tool NO.1
G92 S10000;             // max spindle speed 10000 rpm
G96 S130 M03;           //constant surface speed, surface speed =130m/min, spindle rotate CW.
M08;                    //cutting liquid ON
G00 X0.0 Z41.0;         //rapid positioning O.--->P0
G01 Z40.0 F0.6;         //linear interpolation, feedrate 0.6mm/rev, P0--->P1.
X10.0;                  //P1--->P2
G03 X20. Z35.0 R5.0;    //circular interpolation CCW P2--->P3, radius 5mm.
G01 Z25.0;              //P3--->P4
G02 X30.0 Z20. R5.0     //circular interpolation CW P4--->P5, radius 5mm.
G01 X40.0               //P5--->P6
G03 X50.0 Z15.0 R5.0;   //circular interpolation CCW P6--->P7, radius 5mm.
G01 Z5.0;               //P7--->P8
G02 X60.0 Z0.0 R5.0;    //circular interpolation CW P8--->P9, radius 5mm.
G00 X100.0;             //tool escape, escape from workpiece.
G00 Z65.0;              //return to initial point
M09;                    //cutting liquid OFF
M05;                    //spindle stops
M30;                    //program end
```

2.3.4 **Example 2**



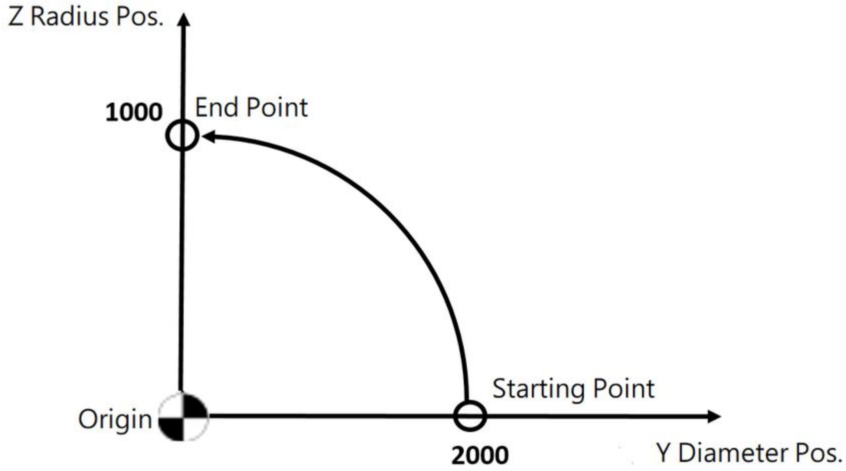
```
T01;                       //use tool NO.1
G50 S10000;                  // max spindle speed 10000 rpm
G96 S130 M03;                //constant surface speed, surface speed 130 mm/min, spindle rotate CW
M08;                       //cutting liquid ON
G00 X0.0 Z110.5;             //positioning, close to the starting point
G01 Z110.0 F0.5;             //linear interpolation, feedrate 0.5 mm/rev
G03 X20.0 Z83.82 R15.0;        //circular interpolation CCW, P1--->P2, radius 15 mm
G01 Z70.0;                   //linear interpolation, P2--->P3
X30.0;                      //P3--->P4
X50.0 Z50.0;                 //P4--->P5
G02 X50.0 Z30.0 R10.0;        //circular interpolation CW, P5--->P6, radius 10 mm.
G03 X50.0 Z10.0 R10.0;        //circular interpolation CCW, P6--->P7, radius10 mm
G01 Z0.0;                    //linear interpolation, P7--->P8
M09;                       //cutting liquid OFF
G00 X100.0;                  //tool escape, escape from workpiece
Z160.0;                     //return to initial point
M05;                       //spindle stops
M30;                      //program end
```

2.3.5 **Example 3 (YZ axis programming)**



```
// Changes to YZ axis processing
// In order to maintain the original processing shape, must rewrite G02/G03.
T01;   //use tool NO.1
G92 S10000; //max spindle speed 10000 rpm
G96 S130 M03; //constant surface speed, surface speed=130 m/min, spindle rotate CW
G19; //Change the processing plane to YZ plane
M08; //cutting liquid ON
G00 Y0.0 Z110.5; //positioning, close to the starting point
G01 Z110.0 F0.5; //linear interpolation, feed rate= 0.5 mm/rev
G02 Y20.0 Z83.82 R15.0; //circular interpolation CW，  P1 →P2, radius 15 mm
G01 Z70.0; //linear interpolation, P2 ->P3
Y30.0; //P3 ->P4
Y50.0 Z50.0; //P4 ->P5
G03 Y50.0 Z30.0 R10.0; //circular interpolation CCW，  P5 →P6, radius 10 mm
G02 Y50.0 Z10.0 R10.0; //circular interpolation CW，  P6 →P7, radius 10 mm
G01 Z0.0; //linear interpolation，  P7 ->P8
M09; //cutting liquid OFF
G00 Y100.0; //tool escape, escape from workpiece
Z160.0; //return to initial point
M05; //spindle stops
M30; //program end
```

2.3.6 **Example 4 (YZ axis cutting 1/4 arc)**



```
// Y axis is in diameter position, Z axis is radius position
// 1/4 arc of radius 1000
G90 G00 Y2000 Z0;
G19 G03 Y0 Z1000 J-1000 K0 F200; //distance from starting point to the center of arc
(G19 G03 V-2000 W1000 J-1000 K0 F200; //arc end point use increment program)
(G19 G03 Y0 Z1000 R1000 F200; //arc radius)
```

2.4 G02/G03-Spiral interpolation(C-type)

2.4.1 **Command Format**

1.

$$G17 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X\_ \quad Y\_ \begin{Bmatrix} R\_ \\ I\_ \; J\_ \end{Bmatrix} Z\_ \quad F\_;$$

X, Y: Coordinates of the arc endpoint;
Z: Coordinate of the straight line endpoint;
R: Radius of the arc;
I, J: Vector from the starting point to the center of the circle;
F: Feedrate velocity;

2.

$$G18 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X\_ \quad Z\_ \begin{Bmatrix} R\_ \\ I\_ \; K\_ \end{Bmatrix} Y\_ \quad F\_;$$

X, Z: Coordinates of the arc endpoint;
Y: Coordinate of the straight line endpoint;
R: Radius of the arc;
I, K: Vector from the starting point to the center of the circle;
F: Feedrate velocity;

3.

$$G19 \begin{Bmatrix} G02 \\ G03 \end{Bmatrix} Y\_ \ Z\_ \begin{Bmatrix} R\_ \\ J\_K\_ \end{Bmatrix} X\_ \ F\_;$$

Y, Z: Coordinates of the arc endpoint;
X: Coordinate of the straight line endpoint;
R: Radius of the arc;
J,K: Vector from the starting point to the center of the circle;
F: Feedrate velocity;

2.4.2 **Explanation**

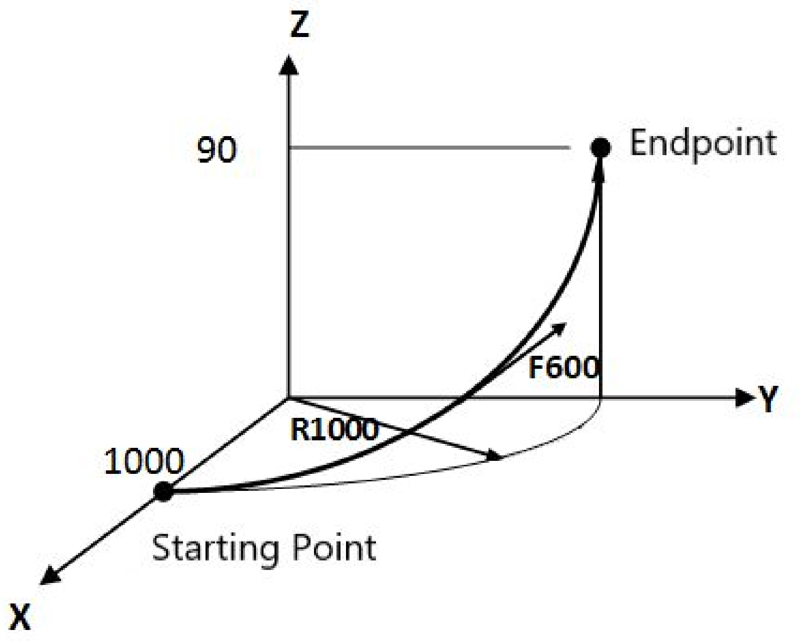
When there is a displacement on the third axis perpendicular to the plane of the arc, the action of the G02/G03 command is spiral interpolation. The selection of the arc plane for spiral interpolation is the same as that for arc interpolation. The command for spiral interpolation specifies the plane in which arc interpolation is performed using the plane selection G-codes (G17/G18/G19).

In G17 mode: The X-Y plane is the plane for arc interpolation, and the Z-axis is the axis for linear interpolation.
In G18 mode: The Z-X plane is the plane for arc interpolation, and the Y-axis is the axis for linear interpolation.
In G19 mode: The Y-Z plane is the plane for arc interpolation, and the X-axis is the axis for linear interpolation.

2.4.3 **Precautions**

1. When no R, I, J, or K is specified for G02/G03, the block is executed as G01.
2. If the arguments X, Y, Z, I, J, K, or R provided for G02/G03 are inappropriate, for example, in G17 mode when argument K is non-zero, the system will issue a [COR-006 Arc endpoint not on arc] alarm. This alarm can be adjusted via Pr3807 to set the alarm range.

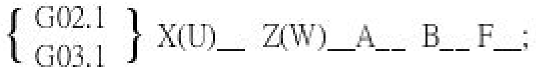
2.4.4 **Program Example**



Program Explanation:
G17 G03 X0.0 Y1000.0 R1000.0 Z90.0 F600;
// For X-Y plane arc, counterclockwise direction (**CCW**), Z-axis linear interpolation
// Cutting rate of 600 mm/min for helical cutting

2.5 G02.1/G03.1- Ellipse interpolation (C-Type)

Command Form



$\left\{\begin{matrix}\text{G02.1}\\\text{G03.1}\end{matrix}\right\}$ X(U)__ Z(W)__A__ B__ F__;

G02.1: Specify tool to make a clockwise ellipse interpolation
G03.1: Specify tool to make a counterclockwise ellipse interpolation
X(U), Z(W): End point of the arc
A: Ellipse Z semi-axis length
B: Ellipse x semi-axis length
F: Feedrate of cutting

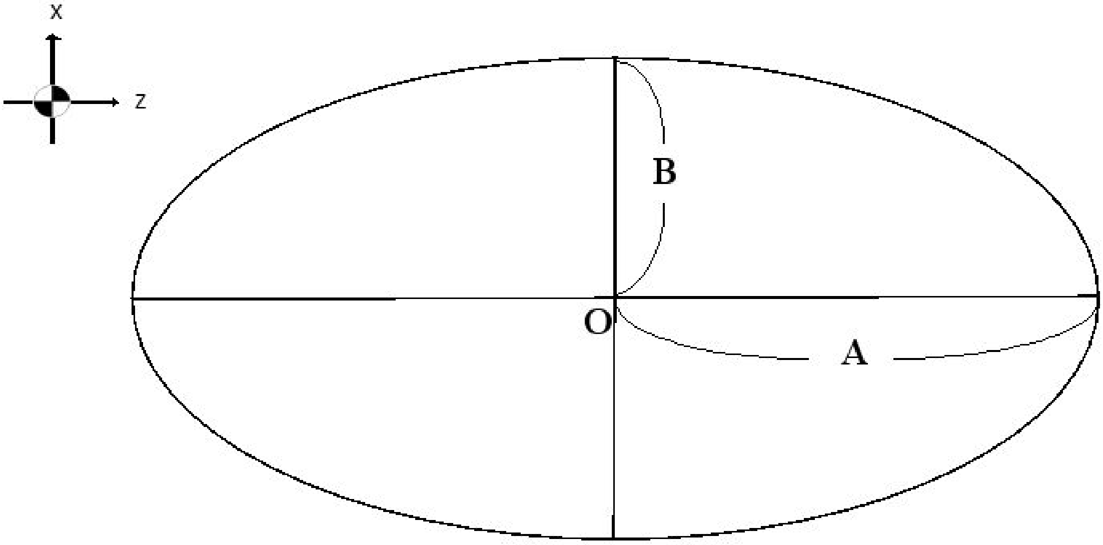
2.5.1 Description

The G02.2 ,G03.2 command of CNC lathe move tool to cut along an ellipse arc on X-Z plane. The setting of the parameters is in the table below:

Setting Data		Command	Definition
1	Tool path direction	G02.1	CW
		G03.1	CCW
2	End position	X, Z	End point of the arc
		U, W	Vector value from start to end point
3	Ellipse Z semi-axis length	A	The length from center of ellipse to the end point of Z axis
4	Ellipse x semi-axis length	B	The length from center of ellipse to the end point of X axis
5	Feedrate	F	Feedrate along the arc

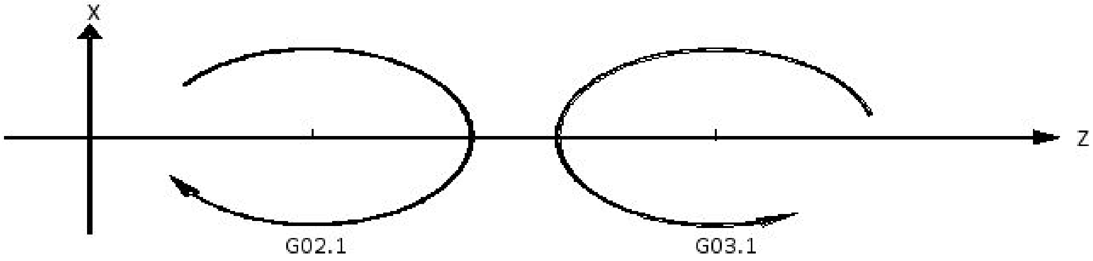
Illustration

Ellipse Z, X semi-axis length definition



As figure shows:
The Z semi-axis length is defined as the length A of the ellipse center O to the end of the Z-axis.
The X semi-axis length is defined as the length B of the ellipse center O to the end of the X-axis.

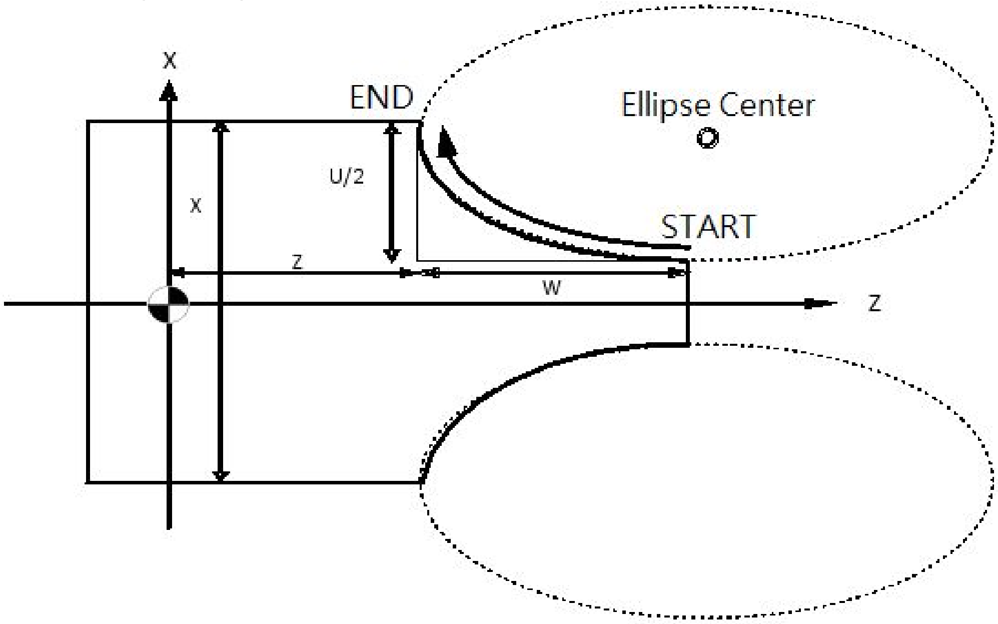
G02.1/G03.1 Defined Direction



G02.1 means ellipse interpolation CW
G03.1 means ellipse interpolation CCW

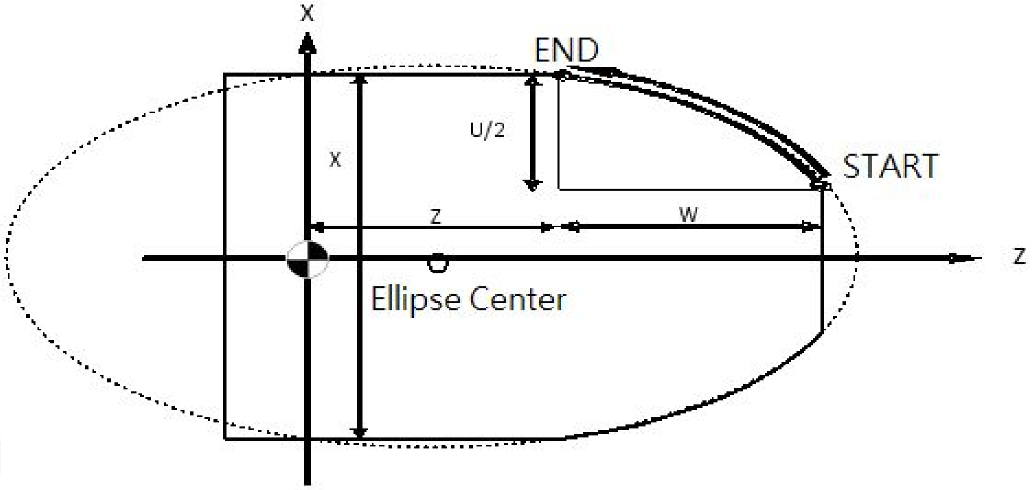
Parameter Setting Example

(1). G02.1 Ellipse Interpolation



G02.1 X(U)_*Z(W)ABF_*;

(2). G03.1 Ellipse Interpolation



G03.1 X(U)__Z(W)__A__B__F__;

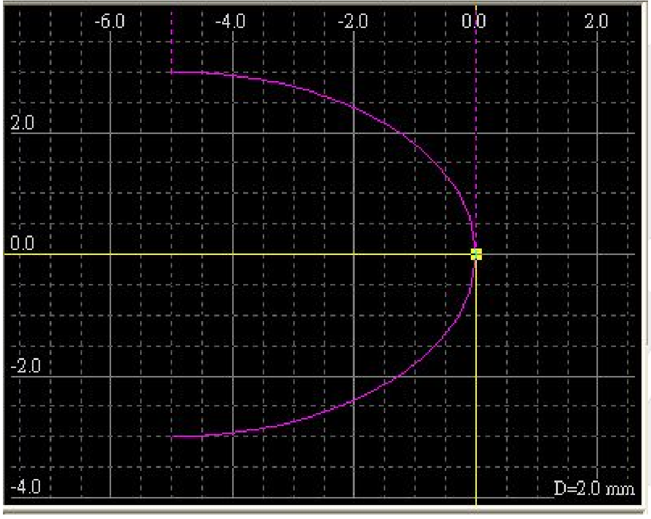
2.5.2 **Precautions**

1. G02.1/G03.1 are not modal G codes, which are valid only in a single block.
2. A, B values are not omissible and must be positive, otherwise system will trigger alarm MAR-022 [ellipse interpolation of Z, X semi-axis missing, less than or equal to zero].
3. The start to end point distance must be equal to or less than the long axis length of the ellipse, otherwise system will trigger alarm MAR-023 [the start and end point distance of the ellipse interpolation is greater than the long axis length of the ellipse].

4. The Z coordinate of start and end point can not be the same, otherwise the system will trigger alarm MAR-026 [ellipse & parabolic interpolation start and end point Z coordinates can not be the same].
5. G02.1/G03.1 can be used for turning commands such as G73, but it cannot appear in the ending number block of a turning cycle command, otherwise the system will trigger alarm COR-028 [System program error, resulting in failure to process normally].

Example 1

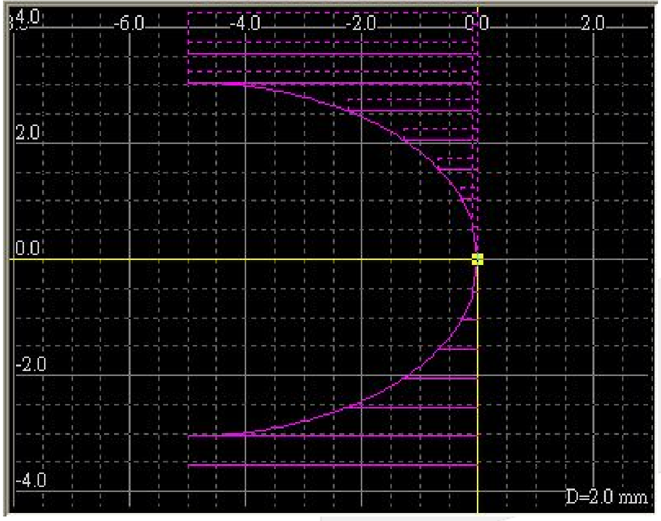
Turning half ellipse



```
G00 X8.;
Z0; // start point Z coordinate of ellipse interpolation
M03 S1000; // spindle rotates CW 1000rpm
G00 X0.; //  start point X coordinate of ellipse interpolation
G03.1 Z-5. X6.A5.B3.F0.5; // ellipse interpolation CCW
G00 X8.0;
M05;
M30;
```

2.5.3 **Example 2**

Half turn ellipse using G71 rough turning cycle



```
G00 X8.;
Z0;// start point Z coordinate of ellipse interpolation
M03 S1000; // spindle rotates CW 1000rpm
G71 U0.5 R0.2 H1;
G71 P01 Q02 U0.1 W0. F0.5;
N1 G00 X0.;// start point X coordinate of ellipse interpolation
G03.1 Z-5. X6. A5. B3. F0.2;// ellipse interpolation CCW
N2 G00 X8.0;
M05;
M30;
```

2.6 G02.2/G03.2- Parabola Interpolation (C-Type)

Command Form

$$\left\{\begin{matrix} \text{G02.2} \\ \text{G03.2} \end{matrix}\right\}\text{X(U)\_\_ Z(W)\_\_P\_\_F\_\_;}$$

G02.2: Specify the tool to make a clockwise parabola interpolation
G03.2: Specify the tool to make a counterclockwise parabola interpolation
X(U), Z(W): End point of the parabola
P: Focal length of parabola
F: Feedrate of cutting

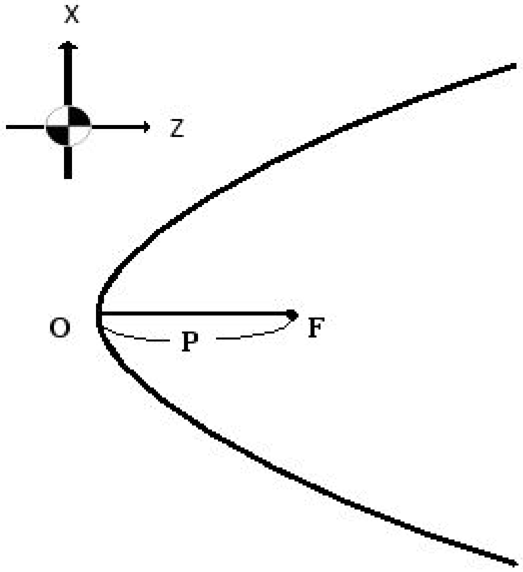
2.6.1 **Description**

The G02.2, G03.2 command of CNC lathe move tool to cut along a parabola arc on X-Z plane. The setting of the parameters is in the following table:

Setting Data		Command	Definition
1	Tool path direction	G02.2	CW
		G03.2	CCW
2	End position	X, Z	End point of the parabola
		U, W	Vector value from start to end point
3	Focal length of parabola	P	The distance from the vertex of the parabola to the focus
4	Feedrate	F	Feedrate along the parabola

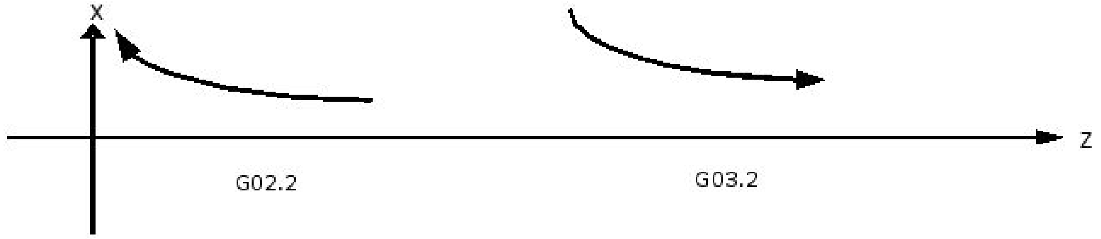
Illustration

Focal length of parabola definition



As figure shows:
The parabola focal length is defined as the distance P from the parabola vertex O to the parabola focus F.

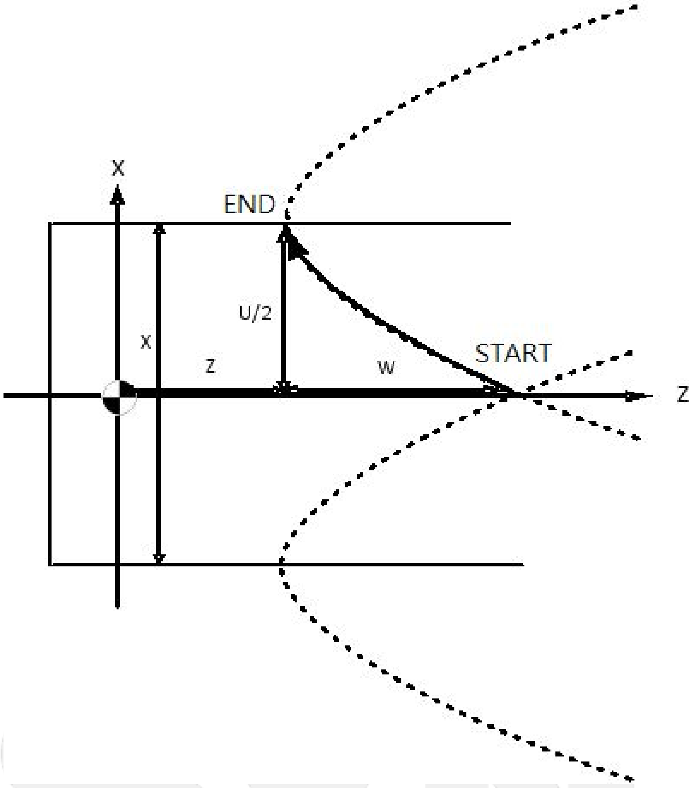
G02.2/G03.2 Defined Direction



G02.2 means CW parabola interpolation
G03.2 means CCW parabola interpolation

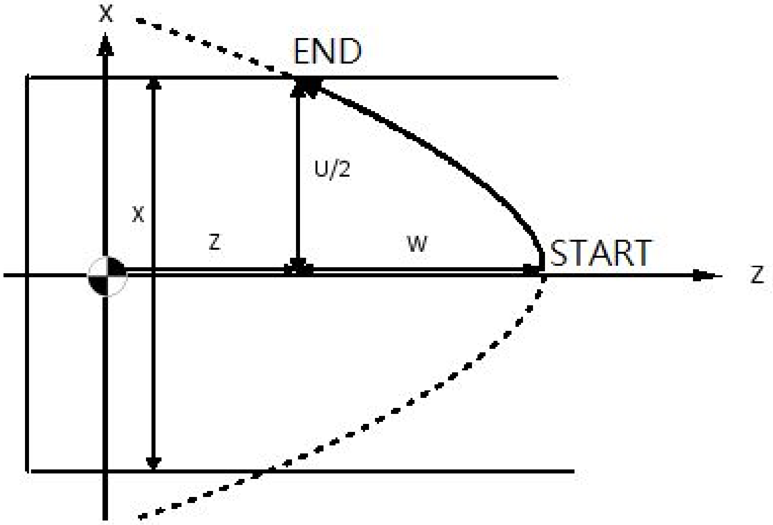
Parameter Setting Example

(1). G02.2 Parabola Interpolation



G02.2 X(U)__ Z(W)__P__ F__;

(2). G03.2 Parabola Interpolation



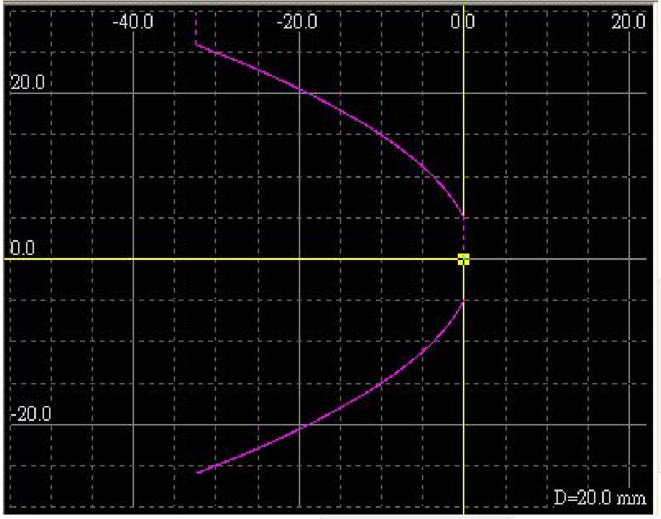
G03.2 X(U)__ Z(W)__P__ F__;

2.6.2 **Precautions**

1. G02.2/G03.2 are not modal G codes, which are valid only in a single block.
2. P values is not omissible and must be positive, otherwise system will trigger alarm MAR-024 [the focal length of parabola interpolation doesn't input or less than/equal to zero].
3. The line from start to end cannot be parallel to the Z symmetry axis. That is, the X coordinate of start and end point must be different, otherwise the system will trigger alarm MAR-025 [the parabola interpolation starting line is parallel to the symmetry axis].
4. The Z coordinate of start and end point cannot be the same, otherwise the system will trigger alarm MAR-026 [ellipse & parabolic interpolation start and end point Z coordinates can not be the same point].
5. G02.2/G03.2 can be used for turning commands such as G73, but it cannot appear in the ending number block of a turning cycle command, otherwise the system will trigger alarm COR-028 [System program error, resulting in failure to process normally].

2.6.3 **Example 1**

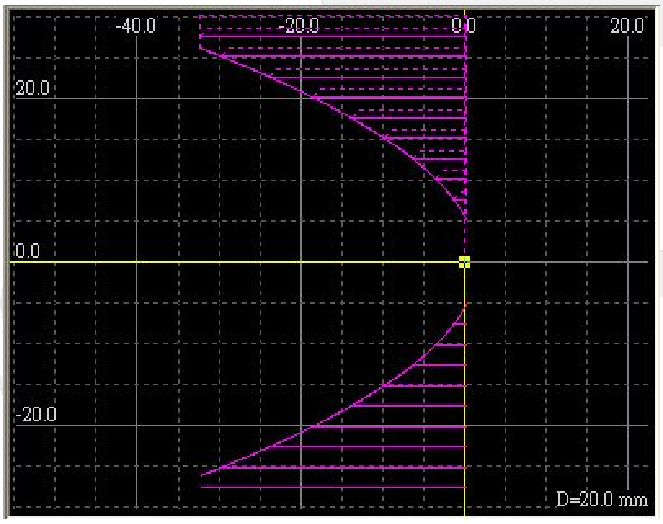
Turning parabola



```
M03 S1000;// spindle rotates 1000rpm
G00 X10.; // start point X coordinate of parabola interpolation
Z0.; //start point Z coordinate of parabola interpolation
G03.2 X52. Z-32.55 P5.;// parabola interpolation CCW
G00 X60.;
M05;
M30;
```

2.6.4 **Example 2**

Turning parabola using G71 rough turning cycle



```
M03 S1000; //spindle rotates CW 1000rpm
G00X60.;
Z0.;
G71 U2.5 R1.0 H0;
```

```
G71 P01 Q02 U0.2 W0.2 F0.5;
N1 G00 X10.;
Z0.;
G03.2 X52. Z-32.55 P5.;
N2 G00 X60.;
M05;
M30;
```

2.7 G04- Dwell (C-Type)

Command Form

$$G04 \begin{Bmatrix} X(U)\_ \\ P\_ \end{Bmatrix} Q\_$$

X(U): dwell time (With decimal point in second or revolution; no decimal point, in millisecond or 0.001 Rev.)

P: dwell time (in millisecond or 0.001 Rev. , decimal point not permitted)

Q: skip signal source, range: 101~132, corresponding to C101~C132 respectively

2.7.1 **Description**

During drilling or grooving, G04 lets tool to dwell a specified time when process to an appropriate position.

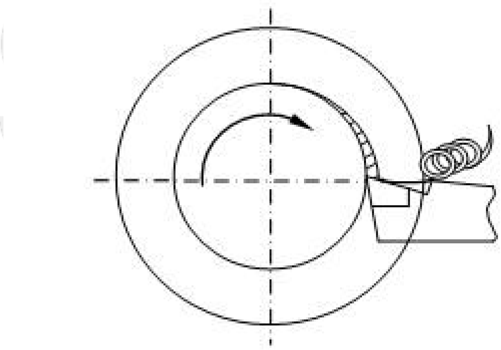
It can help cutting off chips, increase precision of cutting depth, and improve surface finish and achieve better roundness.

G04 command uses unit in second generally, under either G94/G95 in C-type or G98/ G99 in A-type.

2.7.2 **Precaution**

- G04 command is only effective in single block.
- With Q argument, G04 can be terminated before the specified time.
- In Lathe system, the unit can be changed to Rev. through Pr3801.

Example



G04 X0.5   //dwell 0.5s

G04 U0.5 //dwell 0.5s

G04 P500 //dwell 0.5s

2.8 G04.1 - Axis group synchronization (C-type)

2.8.1 **Command Form**

G04.1 P_ [Q_]

P: Numbering of the signal waited
Q: Enter the specified main system to synchronize, without Q argument means synchronizing all system paths. Value in decimal.

Q argument format description:

1. Q argument assign the paths to synchronize, support up to 4 paths. The corresponding Q arguments for each path are in the table below:

Path ID	Q Argument(decimal number)
1	1
2	2
3	3
4	4

2. Take 4 main system paths as an example (Pr731=4): To synchronize path 1, 2, and 4, Q argument must be a series of number 1, 2, and 4 such as Q124.
3. The Q argument number series has no order restriction, all six combinations make path 1, 2, and 4 to synchronize:
   Q124, Q142, Q241, Q214, Q412, Q421
4. The synchronization is effective when command's P are equal and Q assigned path exists, even if Q number orders are different in each path's command. The following cases are equivalent Q arguments, and path 1, 2, and 4 will synchronize:
   a. Processing program of 1st path G04.1 P2 Q124
   b. Processing program of 2nd path G04.1 P2 Q241
   c. Processing program of 3th path G04.1 P2 Q412

2.8.2 **Description**

- G04.1 is used for the need to synchronize different paths. For instance: Example 3, to use $1 to change $2 main spindle RPM. When $2 is under G95 mode, use G04.1 in $1 & $2 to update modes in both paths to prevent incorrect RPM and feed rate in $2.
- If there are 2 paths, G04.1 P1 [Q12] in path 1 and G04.1 P1 [Q12] in path 2 will wait until both paths are synchronized, and then execute then ext block.
- Similarly, G04.1 P2 [Q12] in path 1 and G04.1 P2 [Q12] in path 2 will wait until both paths are synchronized, and then execute then ext block. Others P values are in similar function.
- G04.1 with Q value must be used in the assigned path (no Q value as well), and P value must be use in order.

$1	$2	$3
G00 X0. **G04.1 P1** G01 X10. F1000 G04.1 P2 Q13 X20. G04.1 P3 Q13 X30. G04.1 P5 Q13 X40. **G04.1 P6** M30	G00 Y0. **G04.1 P1** G01. Y10. F1000 Y15. *G04.1 P4 Q23* Y20. **G04.1 P6** M99	G00 Z0. **G04.1 P1** G01 Z10. F1000 G04.1 P2 Q13 Z25. G04.1 P3 Q13 Z40. *G04.1 P4 Q23* Z55. G04.1 P5 Q13 Z70. **G04.1 P6** M99

*Program above: G04.1 w/o Q showed twice in each path; G04.1 Q13 showed 3 times in path 1 & 3; G04.1 Q23 showed once in path 2 & 3.

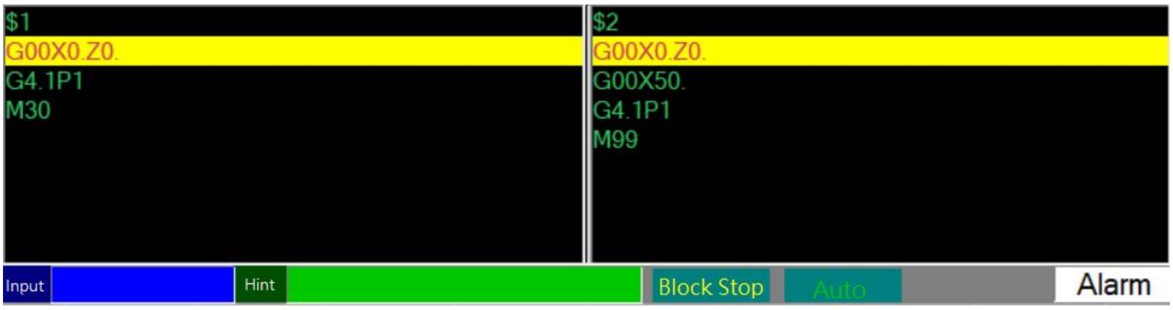
- To repeat program, use M99 at the end of path 1. Notice to add same G04.1 P_ before M99 in each path to ensure all paths repeat synchronously. Ex: the program above has G04.1 P6 before the end of program.

2.8.3 Notice

1. Alarm "COR-137 G04.1 P arguments in wrong order" will be triggered in following cases:
   a. Different P when no Q is used.
   b. Different P when same Q is used.
   c. Same P when different Q are used.
2. Alarm "COR–144 G04.1 Q argument value incorrect"will be triggered in following cases:
   a. Q is not positive integer.
   b. Q value assigned to a non-existing path.
   c. Q value excludes current path. Ex: G04.1 P1 Q23 under path 1.
3. For compatibility, G04.1 with no Q argument means all paths are assigned.
4. G04.1 is not supported under non-CNC main path and all its sub-programs.
5. When waiting at G04.1, system status is "Running". Examples below (shows status of path 1):

Environment: C40 On, programs runs one block at each Cycle Start hit.

Description: $1 and $2 finished single block, and path 1 is at "Block Stop" status.

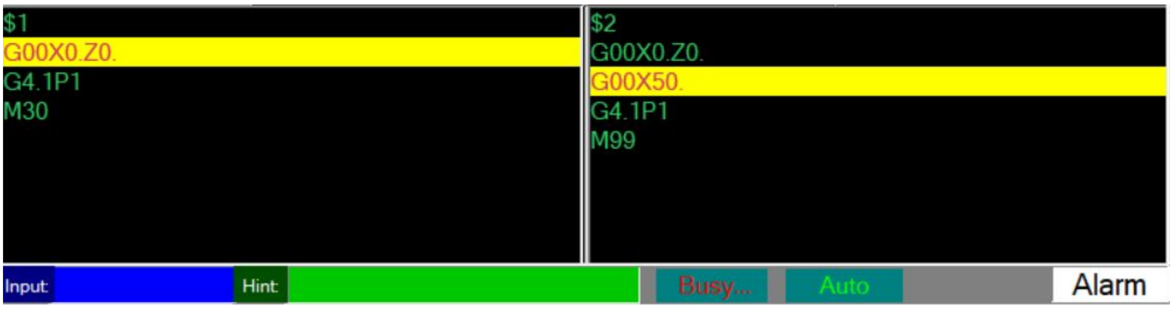


```
$1
G00X0.Z0.
G4.1P1
M30
```

```
$2
G00X0.Z0.
G00X50.
G4.1P1
M99
```

Environment: C40 On, programs runs one block at each Cycle Start hit.

Description: $1 waits for $2, path 1 status is "Busy".



```
$1
G00X0.Z0.
G4.1P1
M30
```

```
$2
G00X0.Z0.
G00X50.
G4.1P1
M99
```

2.8.4 Example

Program 1:

$1	**$2**

G04.1 P1;
G01 X50. F2000;
G04.1 P2;
Z100.;
G04.1 P3;
X0.;
G04.1 P4;
Z0.;
G04.1 P5;
M99;
	G04.1 P1;
	G01 X250. F3000;
	G04.1 P2;
	Z500.;
	G04.1 P3;
	X0;
	G04.1 P4;
	Z0;
	G04.1 P5;
	M99;

Program 2:

$1	**$2**	**$3**

G04.1 P1 Q12; // $1 & $2 sync
G01 X50. F2000;
Z100.;
X0.;
Z0.;
G04.1 P2; // all path sync, and repeat.
M99;
	G04.1 P1 Q12; // $1 & $2 sync
	G01 X25. F3000;
	Z50.;
	X0.;
	Z0.;
	G04.1 P2; // all path sync, and repeat.
	M99;
		G00 X10. Z10.;
		G04 X1.;
		X0. Z0.;
		G04 X1.;
		
		G04.1 P2; // all path sync, and repeat.
		M99;

Program 3： Two spindles first synchronize, and the two paths begin cutting separately. Notice the order of G04.1P_ and main spindle S argument, incorrect order will cause feed rate F of $2 abnormal.

$1	$2
G04.1 P1 // sync with $2 M03 S30 G114.1 R0 // enable spindle synchronize G04.1 P2 // sync with $2 G01 U10. 1. U-10. G04.1 P3 // sync with $2, change spindle RPM after $2 finishes M03 S60 // spindle syncing, spindle 1= 60RPM G04.1 P4 // sync with $2 G04.1 P5 // sync with $2, disable spindle synchronize after $2 finishes G113 // disable spindle synchronize M05 G04.1 P6 // sync with $2, keep $1 from executing M30 while $2 not finishes.	G04.1 P1 // sync with $1, prevent M99 to restart M13 S15 G04.1 P2 // sync with $1, spindle 2 sync to 30 RPM. G01 U10. F2. // G95 mode, feed rate = 30*2 = 60 mm/min U-10. G04.1 P3 // sync with $1 G04.1 P4 // sync with $1, spindle 2 sync to 60 RPM. G01 U10. // G95 mode, feed rate = 60*2 = 120 mm/min U-10. G04.1 P5 // sync with $1 M15 G04.1 P6 // sync with $1 M99

2.9 G07.1- Cylinder Interpolation (C-Type)

2.9.1 **Command Form**

```
G19 Z0 C0;    //select CZ working surface
G07.1 C__ [P_];    //start cylinder interpolation, C_ the cylinder radius, P_ option of axial limit
...
…              //description of cutting route
…
G07.1C0;      //end the cylinder interpolation
```

Definition of C argument:

Start the cylinder interpolation along rotary axis C, and the radius of cylinder equals to the C arguement.

End the cylinder interpolation along rotary axis C by setting C argument equals to zero.

Definition of P argument:

0(default): The effect on axial limits of velocity, acceleration, and jerk caused by the radius will "not" be considered.

1: The effect on axial limits of velocity, acceleration, and jerk caused by the radius will be considered.

P argument can be omitted, and the behavior will be identical to which under P0.

Description

G07.1 command to start cylinder interpolation.

Since the center vector is not easy to calculate, this function converts the angular movement of rotary axis into the movement along its radial surface, and perform linear and circular interpolation with other axes.

The G07.1 function allows programming on the side (radial surface) of the cylinder, so it is easy to create a cylindrical cam grooving program.

Feedrate

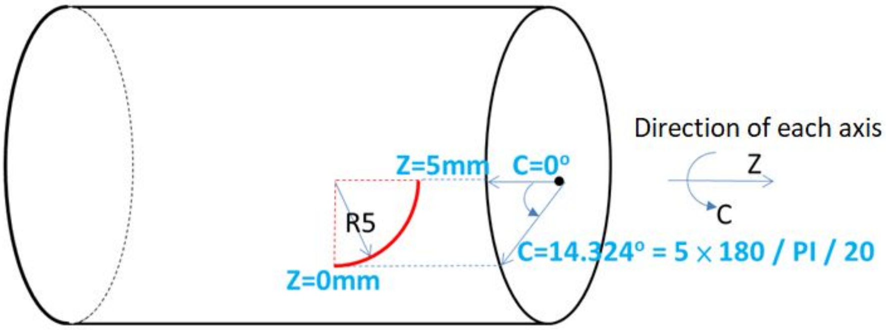
When using lathe, switch to G98 move with F argument allows correct feedrate cutting on cylinder radial surface.

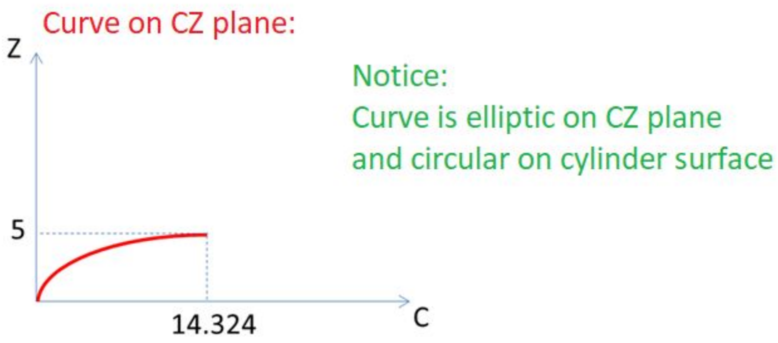
Cylindrical surface transformation

Given that cylinder radius is r and the C-axis rotation is θ, then the tool movement s on the cylindrical surface is s = θ / 180.0 × π × r.

How to draw a quarter circle on a cylindrical surface

Given that cylinder radius is 20 mm and the arc radius to be cut on the cylinder surface is 5 mm. We must first obtain the required rotation angle θ of the C axis, and θ should move tool on the cylinder surface for 5 mm. In this case, we solve equation 5 = θ / 180.0 × π × 20 and get required rotation angle θ=14.324 degrees. This value can be used to draw a quarter circle arc. See the example for detail:

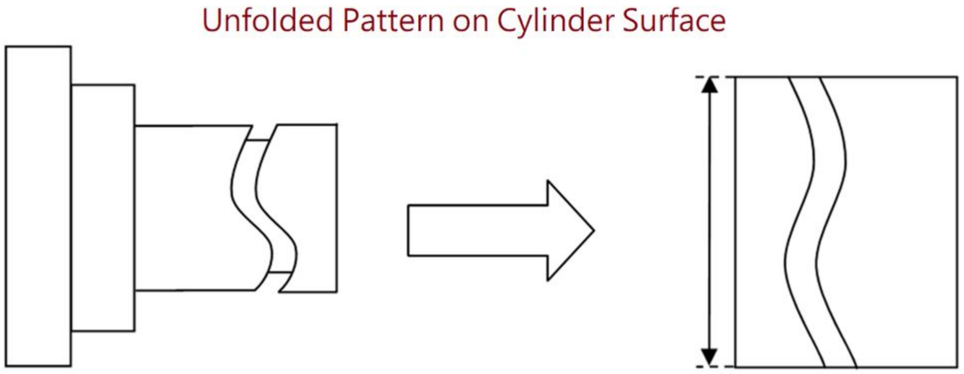


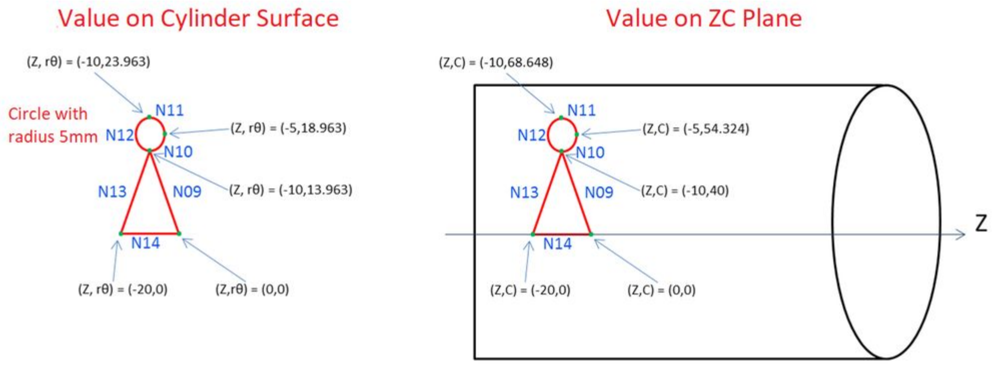


2.9.2 Precautions

- In G07.1 mode, tool radius compensation (G40/G41/G42) is applicable.
- In G07.1 mode, the feed rate must be G98 mode.
- In G07.1 mode, G02/G03 only allow center position & radius method: G02/G03 Z_ C_ R_ "Not allow" the center vector method: G02/03 Z_C_ I_J_(K_)
  - Example: G3 Z-18. C180. R7. F100; //Use R radius
- Please announce the G19 plane before using G07.1
  - Example: G19 C0 Z0;
    or: G19 H0 W0;
- When G07.1 encounters (G00) and rapid positioning related commands (such as positioning related: G28, G53, cycle related: G70-76, G80-89..., etc.), the speed of rapid positioning may be different from G07.1 disabled because system transformed the coordinate system from radial to linear. User can try the following methods If above situations occur :
  - Use G07.1 P1, the effect on axial limits of velocity, acceleration, and jerk caused by the radius will be considered, so the constraints of axial limits won't be break. the related specification can be find in the introduction of P argument.
  - Use G00 and rapid positioning related commands (such as positioning related: G28, G53, cycle related: G70-76, G80-89..., etc.). It is recommended to cancel G07.1 mode first when using these commands.
    - Note: If machining is mostly under G07.1, adjust "Pr461~Pr480 Axis max. rapid travel (G00) feedrate" to set fast positioning speed to meet user's requirements. (This method will cause the positioning speed to change again after disabling G07.1 mode, it is not recommended if not machining constantly under G07.1 mode.)
- In G07.1 mode, avoid using G54-G59.9 "Workpiece coordinate" and G50 "coordinate system setting" to avoid confusion since coordinate system is transformed. Similar applications can be done with G52 "local coordinate system" instead.
  - Note: "Pr3229 Disable workpiece coordinate setting screen display" can close the Workpience Coordinate setting (G54-59.9): 0-start 1-off.

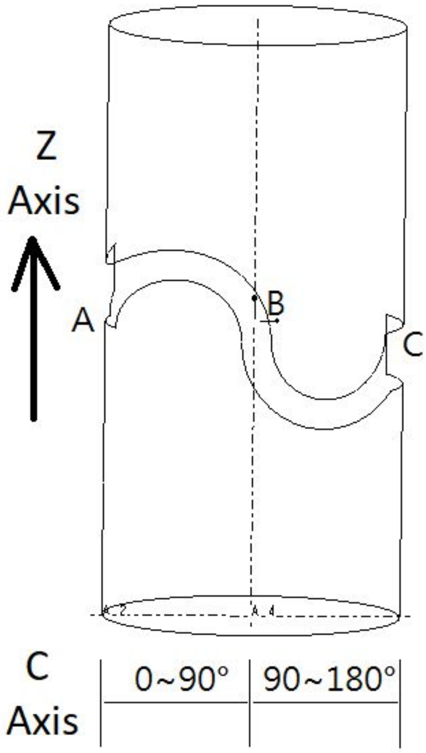
2.9.3 Example 1



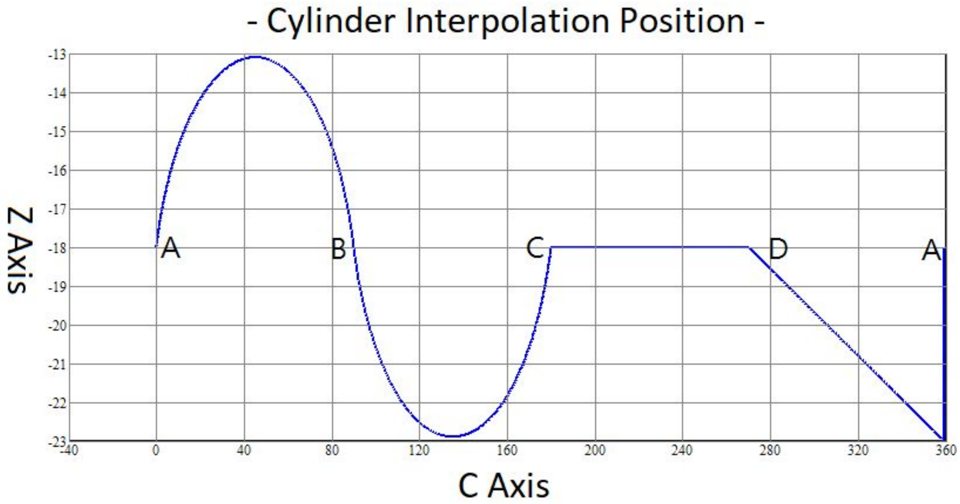


```
N01 G28 U0 W0;
N02 T0202;
N03 G97 S1000;// Set driven tool RPM
N04 G00 X50.0 Z0.;
N05 G98 G01 X40.0 F100.;
N06 G19 C0 Z0;//  Select CZ as the working plane (C is horizontal and Z is vertical)
N07 G07.1 C20.0;//  Start cylindrical interpolation mode, the cylinder radius is 20.0. Since P is omitted, the radius of cylinder will not be included in the axial limits of velocity, acceleration, and jerk.
N08 G41; // Start machining path
N09 G01 Z-10.0 C40.0 F150.0; // Triangle, first side
N10 G02 Z-5.0 C54.324 R5.0; // Quarter circle
N11 G02 Z-10.0 C68.648 R5.0; // Quarter circle
N12 G02 Z-10.0 C40.0 R5.0;// Semicircle
N13 G01 Z-20.0 C0.0;//  Triangle, second side
N14 G01 Z0.0 C0.0; //  Triangle, third side
N15 G40;// End machining path
N16 G07.1 C0;// Disable cylinder interpolation mode
N17 G01 X50.0;
N18 G00 X100.0 Z100.0;
N19 M30;
```

2.9.4 **Example 2**



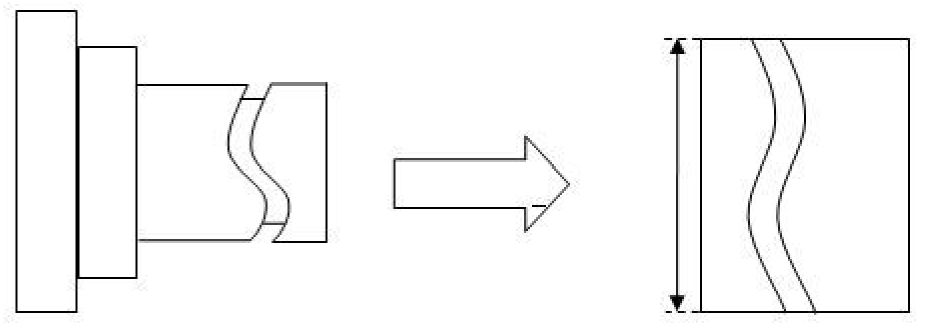
→



```
T0202;
G97 S1000;// Set driven tool RPM
M19; // Spindle OFF
G1 Z-18. F1000.;
C0. X15.;
G19 W0 H0; //  Specify plane
G07.1 C8.5 P1; // Start G7.1 function by using workpiece R8.5, Since P is 1, the radius of cylinder will be included in the axial limits of velocity, acceleration, and jerk.
```

```
G2 Z-18. C90. R7.; // A -> B
G3 Z-18. C180. R7.; // B -> C
G1 Z-18. C270.; // C -> D
Z-23. C360.; // D -> A
Z-18.;
G07.1 C0; //End G7.1 function
M20;
Z35.;
M30;
```

2.9.5 **Example 3**



```
G28 U0 W0
T0202
G97 S1000             // set up spindle RPM
G00 X50.0 Z0.
G94 G01 X40.0 F100.
G19 C0 Z0             // choose CZ the working platform
G07.1 C20.0           // start G07.1, the radius is 20.0
G41                   // enable tool nose compensation
G01 Z-10.0 C80.0 F150.0 //begin machining
G01 Z-25.0 C90.0
G01 Z-80.0 C225.0
G03 Z-75.0 C270.0 R55.0
G01 Z-25.0
G02 Z-20.0 C280.0 R80.0
G01 C360.0              //end machining
G40                   // disable tool nose compensation
G07.1 C0              // cancel G07.1
G01 X50.0
G00 X100.0 Z100.0
M30
```

2.10 G09-Exact Stop Check (C-Type)

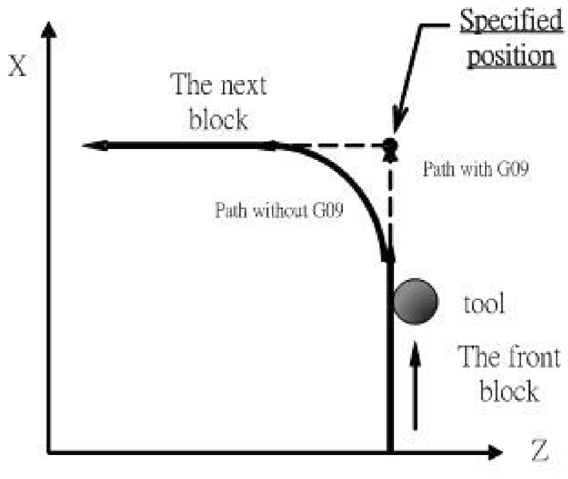
2.10.1 **Command Form**

G09 X__ Z__
X, Z: specified corner position

Description

When machining through a corner, machining tolerance can cause round corner instead of ideal sharp corner due to excessive cutting speed or servo delay. However, when perfectly sharp accuracy is required, the G09 function can be used to decelerate the tool when approaching corner until it reaches a certain range (within the wide range set by the parameter), and then the next block of command will be executed.

2.10.2 **Illustration**



2.11 G10-Programmable Data Input (C-Type)

2.11.1 **Command Form**

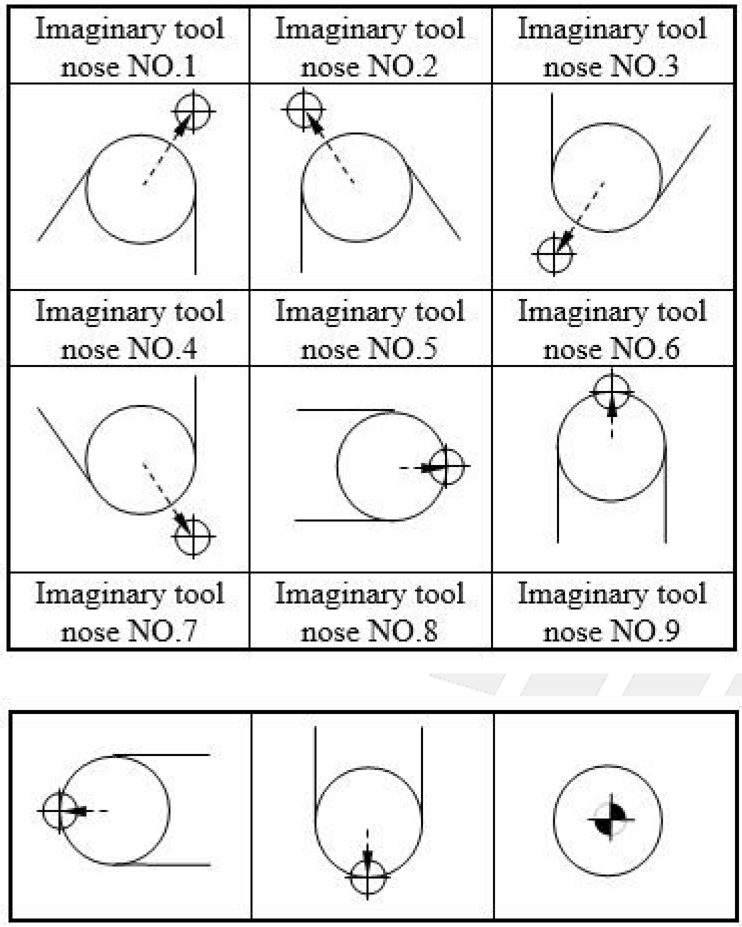
G10 P__X__Z__R__Q__
or
G10 P__U__W__C__Q__

P: offset number
Tool wear offset value: P = number of tool wear offset
Tool geometry offset value: P = 10000 + number of tool geometry offset
X: offset value on X axis (absolute)
Y: offset value on Y axis (absolute)
Z: offset value on Z axis (absolute)
U: offset value on X axis (incremental)
V: offset value on Y axis (incremental)
W: offset value on Z axis (incremental)
R: tool nose radius offset value (absolute)
C: tool nose radius offset value (incremental)
Q: imaginary tool nose number (setting method shows as follow)

Description

1. G10 command is programmable data input command. We can use this command to change the tool offset value when programming.
2. PLC axis control components (PLC Axis) are not supported.

2.11.2 **Imaginary Tool Nose Setting**



2.12 G10.9- Diameter/radius positioning switch (C-Type)

2.12.1 **Command Format**

G10.9 X_ Y_ Z_ Diameter/radius axis programming switching
X、Y、Z：Assign the specific axis to program in diameter/radius axis
0：Program in radius axis
1：Program in diameter axis

2.12.2 **Description**

Users can give G10.9 command in the program to specify all the axial commands in diameter or radius axis afterwards.

2.12.3 **Notice**

1. Diameter/radius axis program switching(G10.9) is valid after version 10.118.8.
2. Please program G10.9 X_ Y_ Z_ command in a line and only, without other commands.

3. With the G10.9 command given, the reset mode is determined by Pr3256. If not retained, the system will revert to the Pr281~Pr300 diameter/radius axis setup after a reboot or reset.
4. The diameter/radius axis program switching(G10.9) setup is effective to the assigned axis controlled by each path. Which means, if X axis belongs to 2 paths, $1 and $2, and it's defined as a radius axis by the parameter; when $1 is given command G10.9 X1, the X axis will also be applied with diameter axis programming when $2 executing command with X axis.
5. When the axis is switched to radius axis programming from diameter axis, the block movements will be doubled. Please make sure if the block movements is correct to avoid tool interference or crashes.
6. The system will deactivate the tool compensation temporary when executing diameter/radius program switching (G10.9), return after new movements.
7. This function only affects the linear axis(Pr221~Pr240 axis type set 0)
8. In polar coordinate interpolation mode(G12.1), diameter/radius axis program switching command(G10.9) is illegal, alarm COR-325 would be issued.
9. If given G10.9 X_ first then activates the polar coordinate interpolation function(G12.1), the X axis will follow the setup value of Pr4020 program under polar coordinate interpolation. After deactivated the polar coordinate interpolation function(G13.1), the X axis will return to the initial parameter setup value. Please assign G10.9 X_ again, if needs to return the G10.9 setup value.
10. If giving G10.9 command without any axis assigned or assigned with the value besides 0 and 1, alarm COR-326 would be issued.
11. Tool length/cutter radius compensation, offset, workpiece coordinate offset, and the values related to parameter coordinates and other machine settings are determined by the PR28x setup value when ready, switching G10.9 during machining won't change the actual machine compensation amount.
12. The values related to program coordinates and machining programming follow the G10.9 dynamic switching.
13. The values related to machine coordinates are determined by the Pr3257 setting.

2.12.4 **Example**

1. Set Y axis as a radius axis with parameters.
   ```
   T0101;
   G90 G00 Y0.;        //move to the orientation point, coordinate 0.0.
   G01 Y5.;            //Y axis actually moves 5mm, coordinate 5.0.
   G10.9 Y1;           //Y axis switched to diameter axis programming, coordinate 10.0.
   Y20.;               //Y axis actually moves 5mm, coordinate 20.0.
   M30;                //after Reset, Y axis switched to radius axis, coordinate 10.0.
   ```

2. Set X axis as a diameter axis and Z axis as radius axis with parameters.
   ```
   T0101;
   G90 G00 X0.;        //move to the orientation point, coordinate 0.0.
   X10. Z10.           //X axis actually moves 5mm, Z axis actually moves 10mm, coordinate (10.0,10.0).
   G10.9 X0 Z1;        //X axis switched to radius axis programming; Z axis switched to diameter axis
   programming, coordinate (5.0,20.0).
   X40. Z40.           //X axis actually moves 35mm, Z axis actually moves 10mm, coordinate (40.0,40.0).
   M30;                //after Reset, X axis switched to diameter axis, Z axis switched to radius axis, diameter
   (80.0,20.0).
   ```

2.13 G12.1/G13.1 - Polar Coordinates Interpolation (C-Type)

2.13.1 **Command Form**

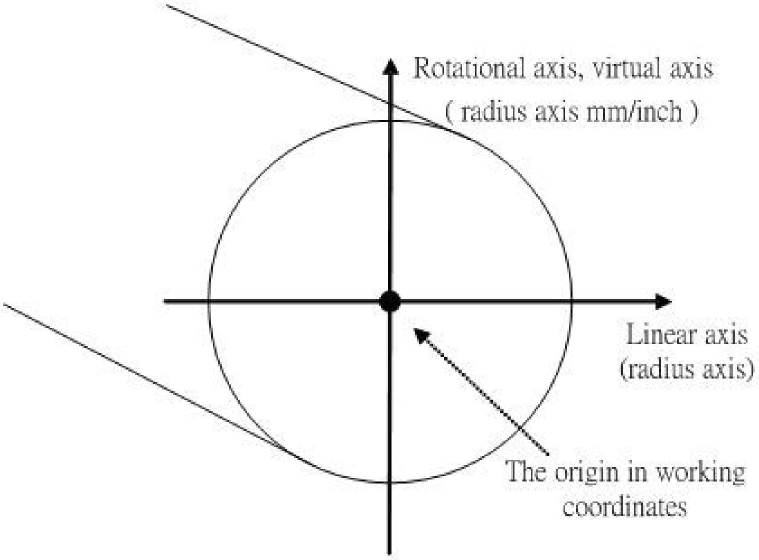
G12.1 X__C__: Enable polar coordinates interpolation
(Enable linear or circular interpolation in a Cartesian coordinate format, command includes a linear and a rotational axes.)

G13.1: Disable polar coordinates interpolation

X: The offset of the program zero point compared to the X-direction of rotation center
C: The offset of the program zero point compared to the C-direction of rotation center

2.13.2 **Description**

1. The function of the polar coordinates interpolation transfers the program in cartesian coordinate into linear axis motion (driven tool motion) and rotational motion (lathe workpiece motion). The command is usually used in cutting end face and milling cam shaft in lathe.
2. The plane of the polar coordinates interpolation: G12.1 enables polar coordinates interpolation and selects a plane to apply transferred pattern. (See picture below)



3. After G12.1, the absolute coordinate of C is displayed as the negative C-direction eccentricity; the absolute coordinate of X is affected by diameter or radius positioning and Pr4020 (G12.1 X axis programming). The details are as follows:
   a. X in radius positioning, Pr4020 = 0, X absolute coordinate is displayed as coordinate before G12.1 minus X-direction eccentricity:
      G0 X50. C90. // absolute coordinate X = 50, C =90
      G12.1 X10. C5. // absolute coordinate X = 50 - 10 = 40, C = 0 - 5 = -5
      G13.1 // absolute coordinate X = 50, C =90
   b. X in diameter positioning, Pr4020 = 0, X absolute coordinate is displayed as half the coordinate before the G12.1 minus X-direction eccentricity:
      G0 X50. C90. // absolute coordinate X = 50, C =90
      G12.1 X10. C5. // absolute coordinate X = 50/2 - 10 = 15, C = 0 - 5 = -5
      G13.1 // absolute coordinate X = 50, C =90

c. X in diameter positioning, Pr4020 = 1, X absolute coordinate is displayed as coordinate before G12.1 minus X-direction eccentricity：
G0 X50. C90. // absolute coordinate X = 50, C =90
G12.1 X10. C5. // absolute coordinate X = 50 - 10 = 40, C = 0 - 5 = -5
G13.1 // absolute coordinate X = 50, C = 90

d. X in radius positioning, Pr4020 = 1, X absolute coordinate is displayed as double of the coordinate before G12.1 minus X-direction eccentricity：
G0 X50. C90. // absolute coordinate X = 50, C =90
G12.1 X10. C5. // absolute coordinate X = 50*2 - 10 = 90, C = 0 - 5 = -5

G13.1 // absolute coordinate X = 50, C = 90

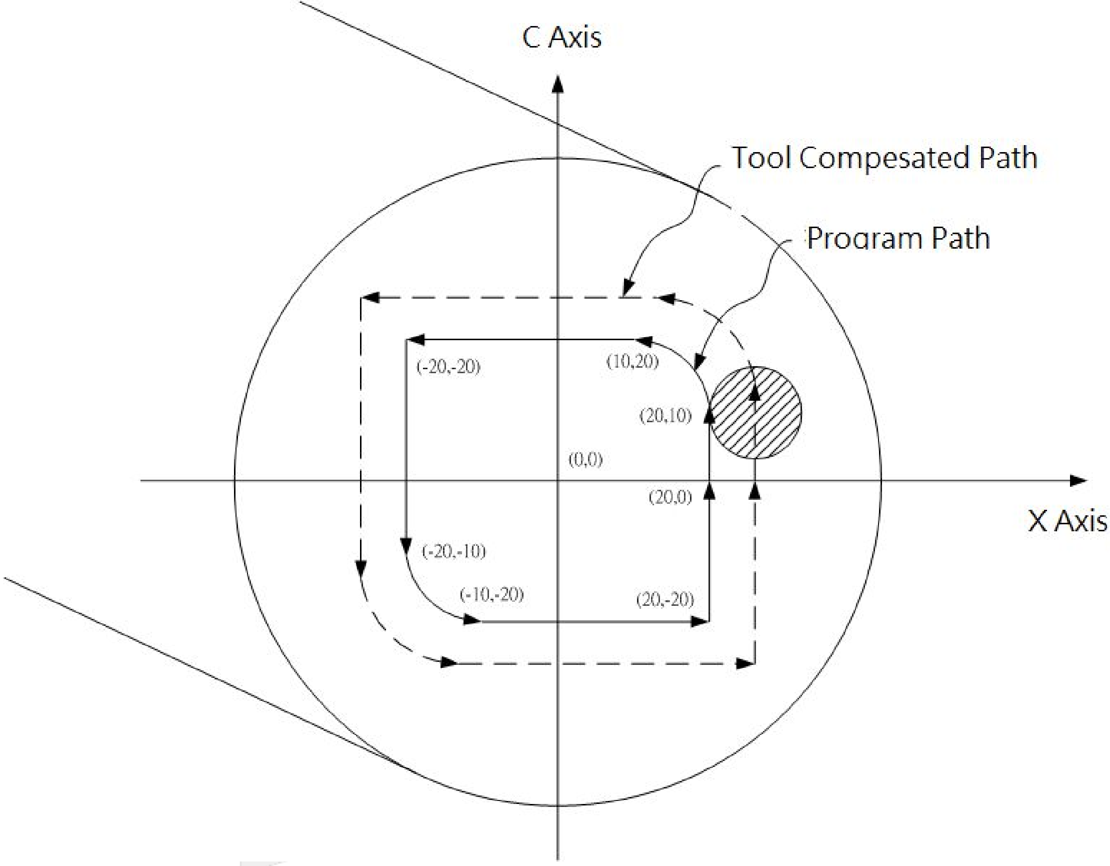
2.13.3 Precautions

1. The eccentricity argument function starts at ver 10.116.11.
2. Linear axis (X) command in diameter positioning is valid from version 10.114.51, the selection mode is determined by Pr4020: =0 is radius positioning; =1 is diameter positioning.
3. Polar coordinate interpolation is cancelled after shutdown or system reset.
4. The coordinate display in the polar coordinate interpolation mode: the linear axis (X) and the rotary axis (C) display the actual position with the radius axis, and the other axes display the actual position with the parameter setting.
5. The following G codes can be used in polar coordinate interpolation:
G01 Linear interpolation
G02/G03 Circular interpolation (I, J, R argument are same as general format)
G04 Dwell
G40/G41/G42 Tool nose radius compensation
G65/G66/G67 User program call (Macro)
6. After the polar coordinate interpolation is started, the plane mode (G17/G18/G19) will be canceled; after the polar coordinate interpolation is canceled or the system is reset, the system restore to the previous defined plane mode.
7. After the polar coordinate interpolation is enabled, Workpience coordinates cannot be changed (G50/G52/G53/G54~G59).
8. Polar coordinate interpolation cannot be enabled or disabled when tool radius compensation (G41/G42) is enabled, only use it when the tool radius compensation function is canceled (G40).
9. When polar coordinate interpolation is enabled and need tool radius compensation function (G41/G42), a 0 movement block of defined tool must be added to ensure the correctness of the path.
10. In the polar coordinate interpolation mode, the tool radius compensation cannot set as preview mode (PR3815=1).
11. Program restart: For programs in G12.1 mode, the program cannot restart to avoid path errors.
12. After switching to the polar coordinates, the current C-axis angle is assumed to be 0. Therefore, perform a C-axis positioning before G12.1 to ensure the correct feed angles (see example)
13. To use this function without turning on the Z axis is not supported. Otherwise, the G02/G03 path may be incorrect.
14. Cannot use polar coordinate interpolation function with the five-axis tool nose point function (G43.4/G43.5).
15. When polar coordinate interpolation function (G12.1) is enabled, do not usediameter/radius positioning switch (G10.9) or alarm COR-325 will pop up.
16. When G10.9 X_ command is followed by the polar coordinate interpolation function (G12.1), the X axis will take Pr4020 value in the polar coordinate interpolation. After canceling the polar coordinate interpolation (G13.1), the X axis will return to the original parameter setting. To resume the setting of G10.9, please re-specify G10.9 X_ command.

17. If the tool moves from the center position of the C-axis, the movement can be divided into two stages: first C-axis positioning and then X-axis movement. The separate action is to avoid singular point at center of C Axis due the nonlinear conversion between Cartesian and polar coordinates.
18. This feature doesn't support Multi-Axis Multi-Signal Skip Function(G31.10/G31.11).

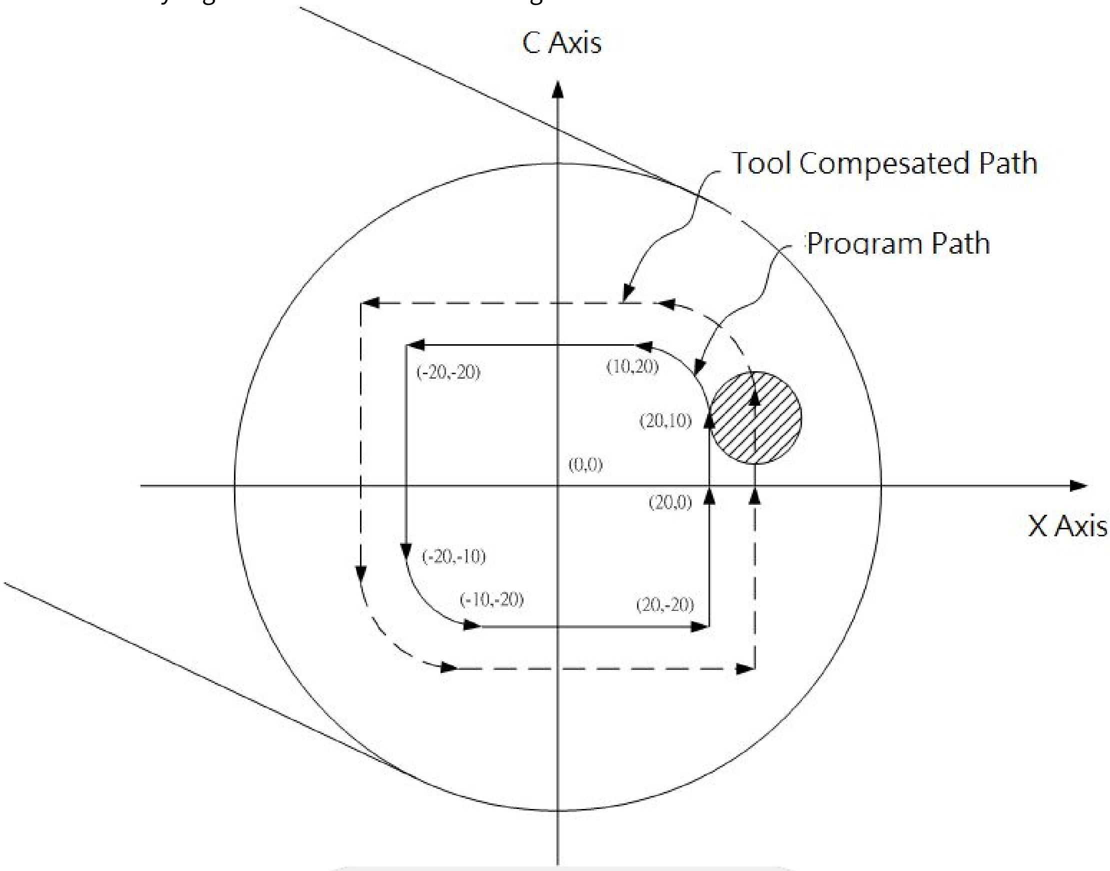
2.13.4 **Example**

1. Eccentricity Argument - Radius Positioning



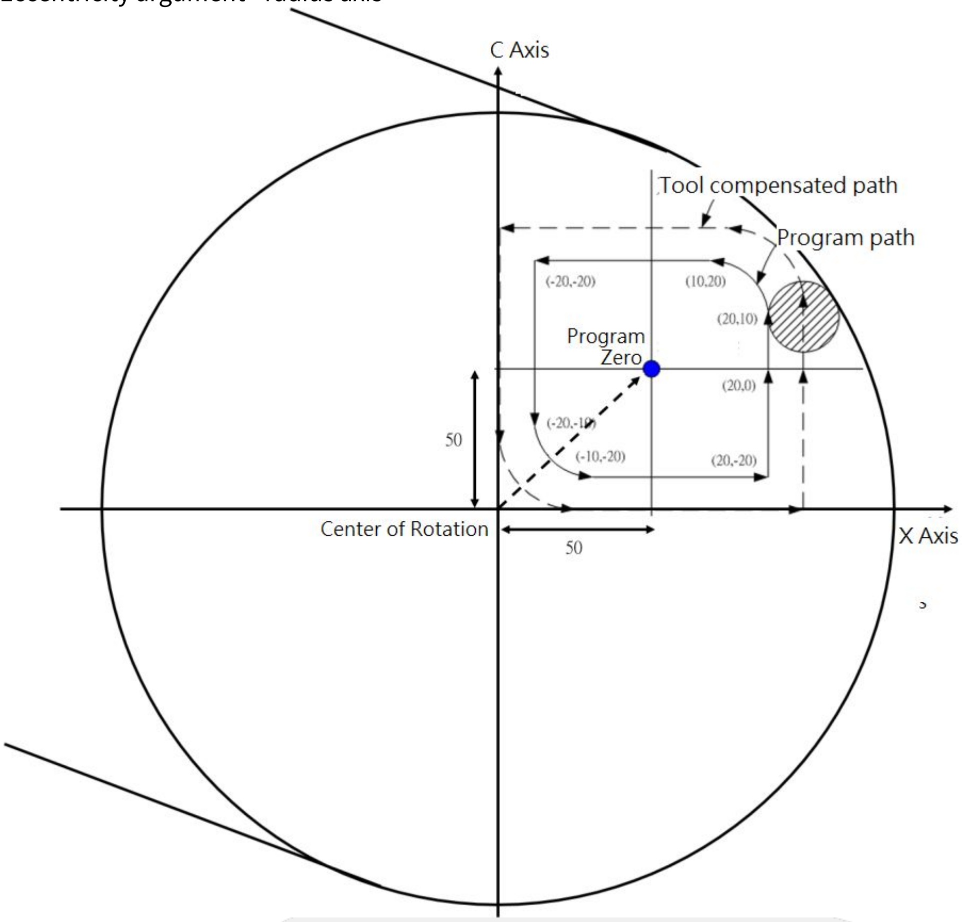
```
T0101;
G00 X110. C0. Z0.; //Move to positioning point
G40 G94;
G12.1;//Start polar polar coordinate interpolation
//Program with the Cartesian coordinate system X-C plane
G42 X55. ;//Add a block with a movement amount of 0
G01 X20. F100.;
C10.;
G03 X10. C20. R10.;
G01 X-20.;
C-10.;
G03 X-10. C-20. R10.;
G01 X20.;
C0;
G40 X55.;
G13.1;//Cancel polar coordinate interpolation
M30;
```

2. No Eccentricity Argument - Diameter Positioning



```
T0101;
G00 X110. C0. Z0.; //Arrive positioning point
G40 G94;
G12.1;//Start polar polar coordinate interpolation
//Program with the Cartesian coordinate system X-C plane
G42 X110. ; //Add a block with a movement amount of 0
G01 X40. F100.;
C10.;
G03 X20. C20. R10.;
G01 X-40.;
C-10.;
G03 X-20. C-20. R10.;
G01 X40.;
C0;
G40 X110.;
G13.1;//Cancel polar coordinate interpolation
M30;
```

3. Eccentricity argument - radius axis



```
T0101
G00 X110. C0. Z0.; //Arrive positioning point
G40 G94;
G12.1 X50. C50.;//Start polar polar coordinate interpolation, eccentricity(50, 50)
//Edit program with the Cartesian coordinate system X-C plane
G42 X55.;//Add a block with a movement amount of 0
G01 X20. F100.;
C10.;
G03 X10. C20. R10.;
G01 X-20.;
C-10.;
G03 X-10. C-20. R10.;
G01 X20.;
C0;
G40 X55.;
G13.1;//Cancel polar coordinate interpolation
M30;
```

Appendix

1. Specify arbitrary axis + tool radius compensation
   For G12.1, X axis is the default linear axis and C axis is the default rotary axis. Due to different machine

configurations, sometimes it might need to change the linear axis to Y axis, for the situation customized G12.1 is needed.

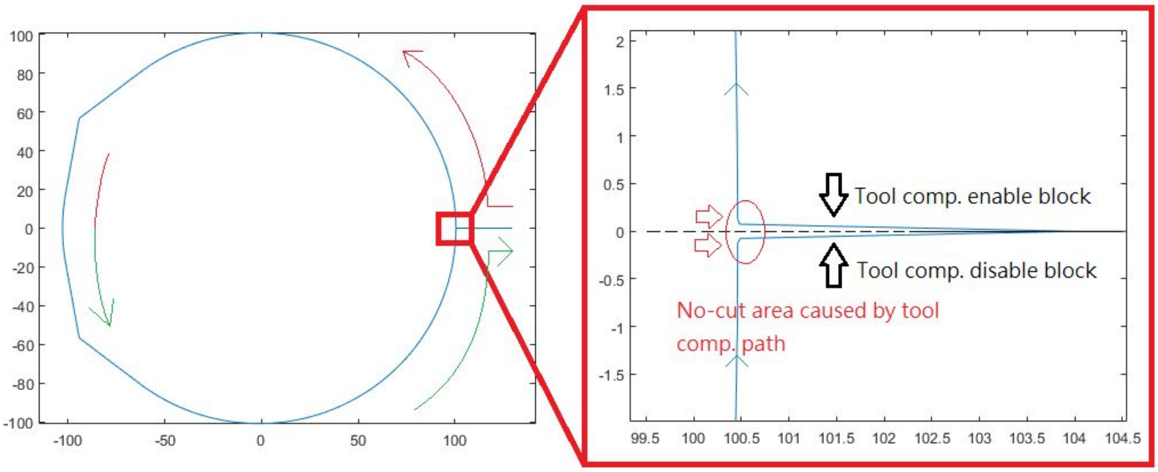
G10 L1301 X_ C_ R_;
X__ linear axis ID
C__ rotary axis ID
I_: The offset of the program zero point compared to the X-direction of rotation center
J_: The offset of the program zero point compared to the C-direction of rotation center
R_: 0:Disable、1:Enable(diameter axis uses radius programming)、2:Enable(diameter axis uses diameter programming)

Example (customized G012001, Y linear axis C rotary axis):

```
%@MACRO
IF (#1012<>40) THEN
   ALARM( 17 );
END_IF;
IF (#1018=96) THEN
   ALARM( 18 );
END_IF;

#30:=AXID(Y); // get Y axis ID
#31:=AXID(C); // get C axis ID
#33:=1+#4020; // diameter/radius programming parameter
// get Y axis diameter/radius programming before G12.1 enable
#34 := ROUND( POW( 2, #30 ) );
#35 := #1814 AND #34;

IF (#25 = #0) THEN
   #25 := 0;
END_IF;
IF (#3 = #0) THEN
   #3 := 0;
END_IF;
IF ((#30=#0) OR (#30<=0) OR (#31=#0) OR (#31<=0)) THEN
   ALARM( 19 );
   M99;
END_IF;

// State backup
#32:=#1004;
#2048:=#1002;
#2049:=#1008;
// When using the arc interpolation, tool radius compensation or polar coordinate interpolation, must set cutting plane with G17/G18/G19 first.
G91 G19 Y0 C0;
G94;
G90 G10 L1301 X#30 C#31 I#25 J#3 R#33; // Start polar coordinate interpolation mode, specified as Y-C
IF ( #35 = 0 AND #4020 = 1 ) THEN
   Y( #1412 * 2.0 - #25 ) C-#3;
ELSEIF ( #35 > 0 AND #4020 = 0 ) THEN
   Y( #1412 / 2.0 - #25 ) C-#3;
ELSE
```

```
  Y( #1412 - #25 ) C-#3;
END_IF;

WAIT();
G#32;
M99;
```

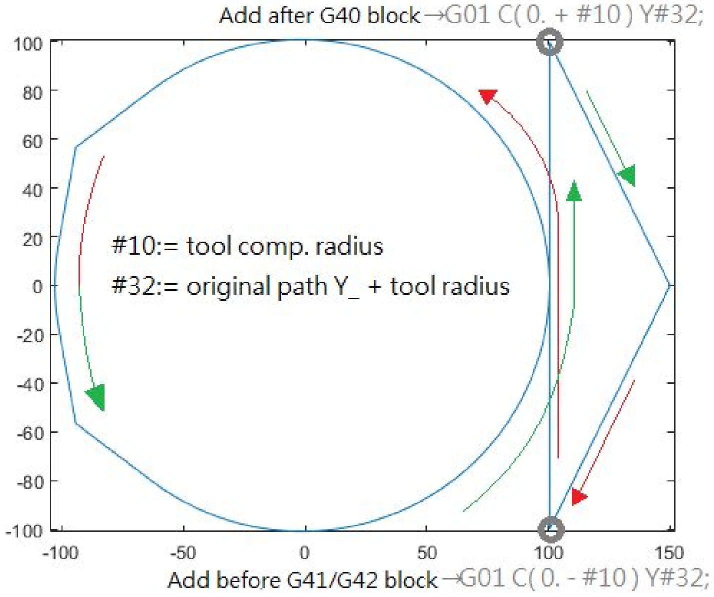
When tool compensation is enabled under polar coordinate interpolation, directly giving contour path may still cause no-cut at tool starting point of the workpiece (see following figure).



Therefore, it is recommended to add a block with tool command to allow the path to enter along the contour tangent. Assume the begin and end position commands are all C0, then the simplest arrangement is:

Before G41/G42 and after G40, add one moving block along the cutting contour tangent path plus the position after the tool radius compensation, and C_ axis moves at least one radius of the tool distance along the path direction (determined by the original path). Detail refer to the following example, Y-C polar coordinate + tool radius compensation (adding a single block with tool)

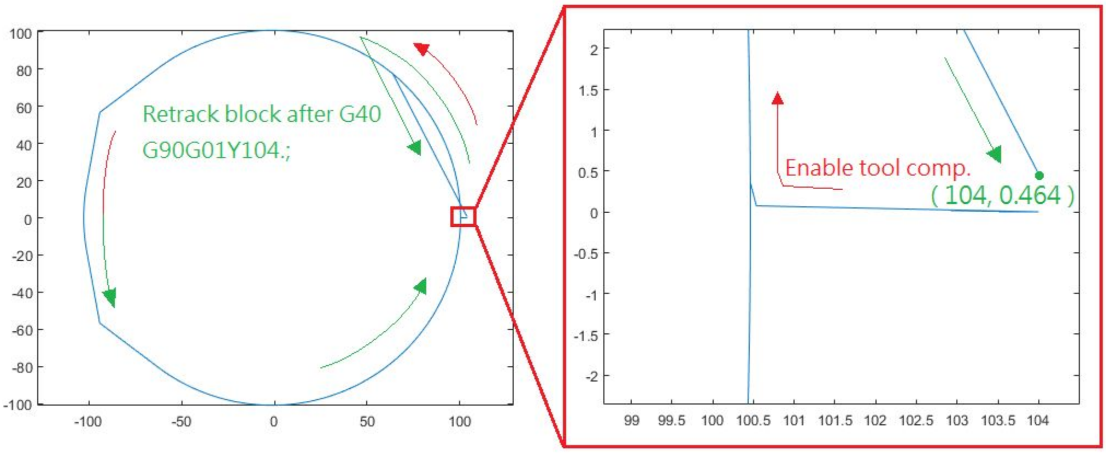
Example (Use customize G12.1 + tool radius compensation)



```
%@MACRO // Orange can be replaced by blue MACRO code
G90 G00X0.Y150. C0. Z0. F1500 ;
M03 S3000;
M08;
G40;

G90 G10 L12 P13 R99.86;   // #10 := 99.86; tool radius 99.86
                          // G90 G10 L12 P13 R#10; set the tool radius of T13
G12.1;                     // Start customize Y-C polar coordinate interpolation

G90G01 F1600;
G01 C-99.86 Y100.462;      // #32 := 0.602 + #10; cutting contour path plus tool radius, 99.86 + 0.602 = 100.462
                          // #G01 C( 0. - #10 ) Y#32;  Proper tool move block, Y_ moves to 100.462, and C_ moves
the radius of the tool
G42 D13 C0. Y0.602;
//  cutting contour
// ====================
G03 C0.464 Y0.383 R0.602;
G03 C0.536 Y-2.952 R2.69;
G03 C-0.536 Y-2.952 R3.0;
G03 C-0.464 Y0.383 R2.69;
G03 C0. Y0.602 R0.602;
// ====================
G40;
G01 C99.86 Y100.462;       // G01 C( 0. + #10 ) Y#32;  Tool retract block, C_ moves the radius of the tool
                          // Retract block is still affected by tool compensation, should be before G13.1.
// ====================
G90 G01 C0. Y150.;          // Return to positioning point
G13.1;                     // Close polar coordinate interpolation
M30;
```

※Note: Avoid claiming any work plane under tool compensation, or it may temporarily closes the tool compensation and generates wrong path.

2. Retraction path causes overcutting
   According to the third point, there is a problem that there is no way to cut the workpiece under the knife. If you move to C≠0. by the repeated path of the plan, you need to pay attention to the path of Y-C while retreating. In this example, after G40, G90 G01 Y104., the intuition will think that the path should be extended to 104.. However, the single section is for the single block, and there is no command to release C, so C will move to the front command under a single block (G03 C0.464 Y0.383 R0.602;), making it possible to cancel the tool radius compensation and move to the position of (104.0, 0.464). From the figure below, you can see the cutting path moves from inside and cause the workpiece to be overcut. Therefore, if you really want the path to be moved horizontally in the Y direction, the block should be placed in front of the G40, but it should be noted that this involves the path of the tool offset, and the tool may not be able to be compensated properly (in this case, the tool radius is too large)). Please refer to the example of the third point for the way to correct the path.



```
%@MACRO
G90 G01X0.Y104.C0.Z0.F1500 ;
M03 S3000;
M08;
G40;

G90 G10 L12 P13 R99.86;    // set the tool radius of T13
G12.1;

G90G01 F1600;
G42D13 C0. Y0.602 ;
// cutting profile
// =====================
G03 C0.464 Y0.383 R0.602;
G03 C0.536 Y-2.952 R2.69;
G03 C-0.536 Y-2.952 R3.0;
G03 C-0.464 Y0.383 R2.69;
G03 C0. Y0.602 R0.602;
G03 C0.464 Y0.383 R0.602;  //  Additional cutting path of the uncut part due to tool radius compensation
// =====================
G40;                         // Disable tool offset
G90 G01 Y104.;                 // Retract block, since no specified C coordinate
                             // Retract to ( 104.0, 0.464 ) cause overcutting
// =====================
G13.1;                        // Disable polar coordinate interpolation
M30;
```

2.14 G17/G18/G19- Work Plane Selection (C-Type)

2.14.1 **Command Form**

G17:   X-Y plane selection
G18: Z-X plane selection （Controller default）
G19: Y-Z plane selection

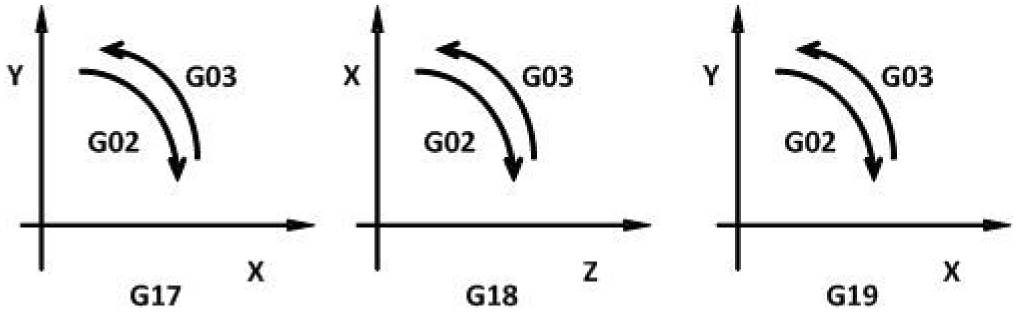
2.14.2 **Description**

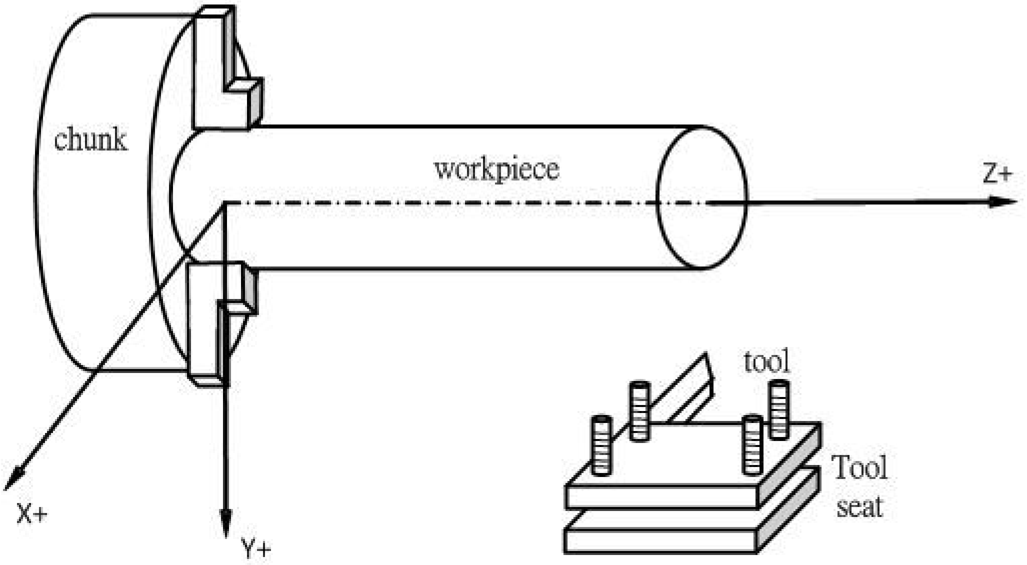
1. When using the arc command and tool radius compensation command, G17, G18, and G19 must be used first to set to inform the controller of the machining plane.
2. the actual corresponding axes in the X, Y, and Z directions of the cutting plane are called geometry axes. The selection rules are as follows:
    a. The controller will classify the axis into three categories according to the axis name.
        i. X categories: X, X1 ~ X99, U, U1 ~ U99, A, A1 ~ A99.
        ii. Y category: Y, Y1~Y99, V, V1~V99, B, B1~B99.
        iii. Z category: Z, Z1~Z99, W, W1~W99, C, C1~C99.
    b. The axis in X category is eligible to be selected as the X axis; Y and Z axes are in similar fashion.
    c. If there are multiple axes in same category, follow the order above as the line of selection.
    d. If there is no corresponding axis in a category, then choose the axis with the smallest ID among unselected axes as the geometry axis.
    e. If the number of system declared axes is less than three, there will be no geometry axis selected for certain category. In this case, the arc command, tool radius correction command or polar coordinate command will be limited in use.
        i. Taking a two-axis lathe (Z, X-axis) as an example, only the G18 work plane can be used.
    f. One axis can not be selected as two geometry axes at the same time.

2.14.3 **Precaution**

1. When using G17, G18, G19 to switch machining plane with movement command in the same block, movement and plane switch will be execute in simultaneously. Pay attention to the action of the machine to avoid danger.

2.14.4 **Illustrations**





2.14.5 **Example**

Given controller parameters are set as follows:

Pr21, 22, 23, 24, 25 = [1, 2, 3, 4, 5]

Pr321, 322, 323, 324, 325 = [101, 100, 800, 302, 301]

Therefore, there are five axes in the system named X1, X, V, Z2, and Z1 (axis ID from small to large)

According to the above rules, the three geometric axes that form the space are: X, V, Z1.

Example 1:

```
G17;
G91 G02 X5. R20. F2000;//  After G17 command, the arc will be displayed on the X-V plane.
```

Example 2:

```
G18;
G91 G02 X5. R20. F2000;// After G18 command, the arc will be displayed on the Z1-X plane.
```

2.15 G20- Outer (Inner) Surface Tuning Cycle (C-Type)

2.15.1 **Command Form**

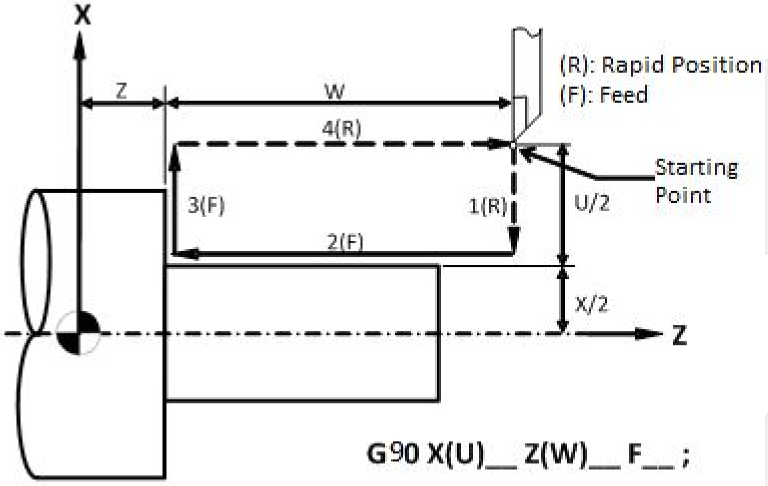
1. Straight turning cycle: G20 X(U)__ Z(W)__ F__ ;
2. Tapered turning cycle: G20 X(U)__ Z(W)__ R__ F__ ;
X, Z: Turning endpoint coordinates (absolute value mode)
U, W: Turning endpoint coordinates (incremental value mode)
R: Radius difference between the starting point and the endpoint
F: feed rate

2.15.2 **Description**

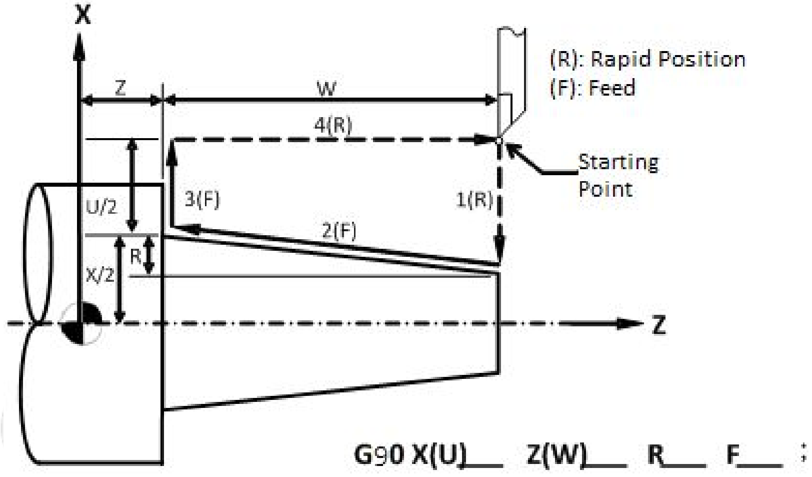
The G20 command is an outer (inner) surface straight and tapered turning cycle. The **cycle** is used to represent turning movements of several blocks into one block so that the program is simplified.

X axis only supports setting as diameter axis, and Z axis only supports setting as radius axis.

Axial Straight Turning Cycle



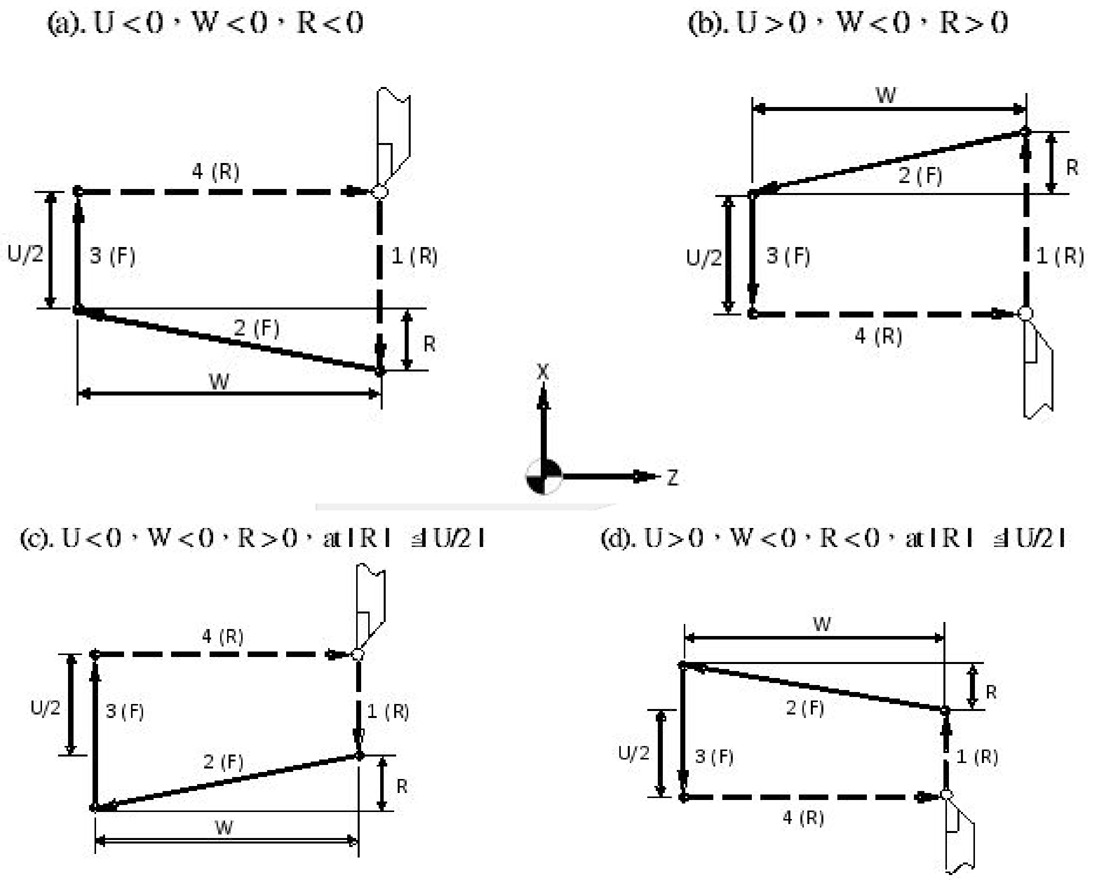
Axial Rapered Turning Cycle



Action description

1. Rapid position the tool to starting point before cycle;
2. When executing the G20 command, the tool first rapid position to the coordinate of the X (U) to begin feed.
3. The tool then feeds to specified X (U), Z (W) position at a specified F feed rate;
4. At the end of the feed, the tool automatically returns to the starting point in rapid positioning;
5. After arriving the starting point, the tool continues to repeat the cycle of the feed by the next X(U) value;
6. When the specified size reached, the tool will stop at the starting point and wait for the next cycle.

※ When the incremental value is used, the relationship between the sign of argument U/W/R and the tool path are as follows:



(a). U<0，W<0，R<0

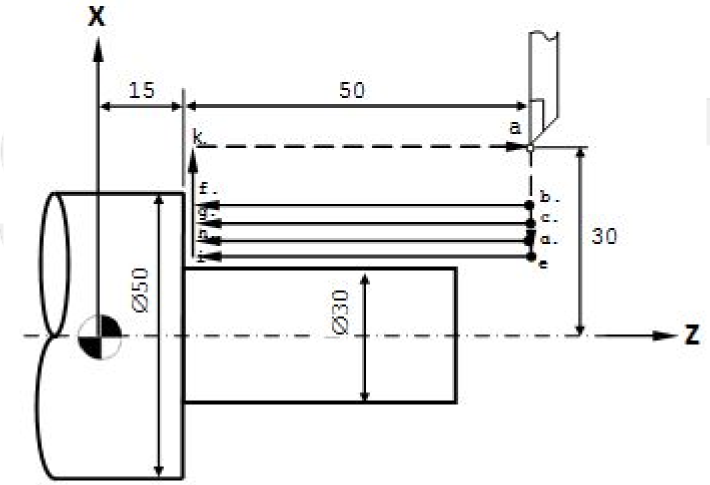
(b). U>0，W<0，R>0

(c). U<0，W<0，R>0，at |R| ≦ |U/2|

(d). U>0，W<0，R<0，at |R| ≦ |U/2|

2.15.3 **Example 1**

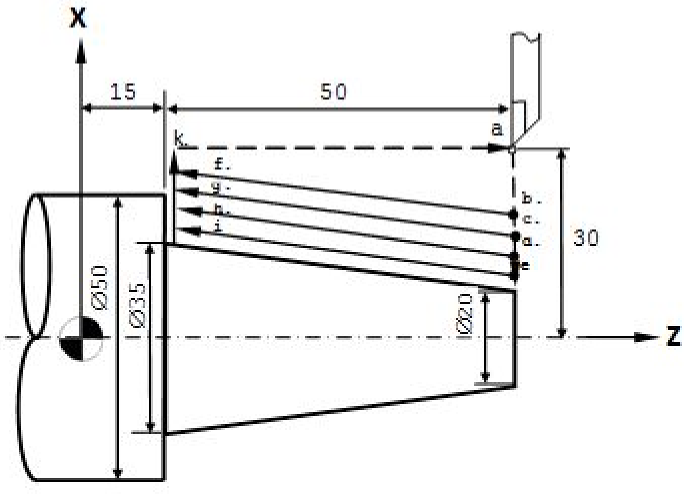
Axial Straight Turning Cycle



```
G50 S5000; //maximum speed 5000 rpm
T01; //use tool NO. 1
G96 S130 M03;//constant surface speed at 130 m/min, spindle rotate CW
M08; //cutting liquid ON
G00 X60.0 Z65.0;//position to  a.(starting point)
G20 X45.0 Z15.0 F0.6;//perform axial turning cycle, feed rate 0.6 mm/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;//positioning to specified mid-point, then return to machine zero point
M09; //cutting liquid OFF
M05; //spindle stops
M30; //program ends
```

2.15.4 **Example 2**

Axial Tapered Turning Cycle



```
G50 S5000;        //maximum speed 5000 rpm
T01;              //use tool NO. 1
G96 S130 M03;     //constant surface speed at 130 m/min, spindle rotate CW
M08;              //cutting liquid ON
G00 X60.0 Z65.0;  //position to  a.(starting point)
G20 X53.0 Z15.0 R-7.5 F0.6;  //perform axial turning cycle, feed rate
                             // 0.6 mm/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;  //positioning to specified mid-point, then return to machine zero point
M09;              //cutting liquid OFF
M05;              //spindle stops
M30;              //program ends
```

2.16 G21- Threading Cycle (C-Type)

2.16.1 **Command Form**

1. Straight threading cycle:
   G21 X(U) Z(W) H (F___ or E___);
2. Tapered threading cycle:
   G21 X(U) Z(W) R H (F___ or E___);

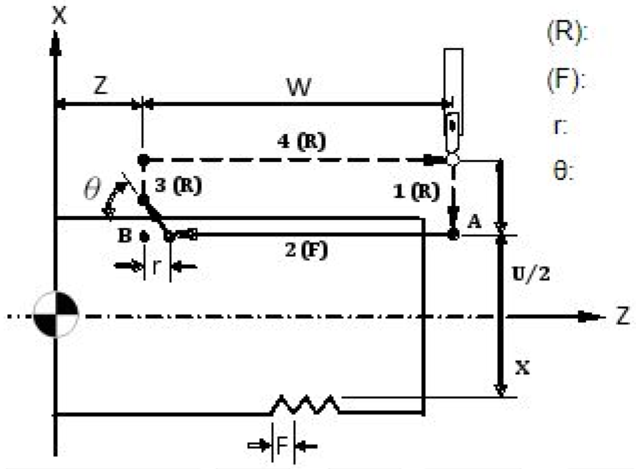
X, Z: turning end point (absolute value mode)
U, W: turning end point (Incremental value mode)
R: Taper radius difference
F: Thread pitch in metric (unit: mm/thread)
E: Thread per inch (unit: thread/inch)
H: The number of multi-start thread (ex: H3 means 3 starts. When H is given, F means the pitch of adjacent threads)

Description

The G21 command is a threading cycle which simplifies thread turning, retract, and rapid positioning into single block.

Straight Threading Cycle

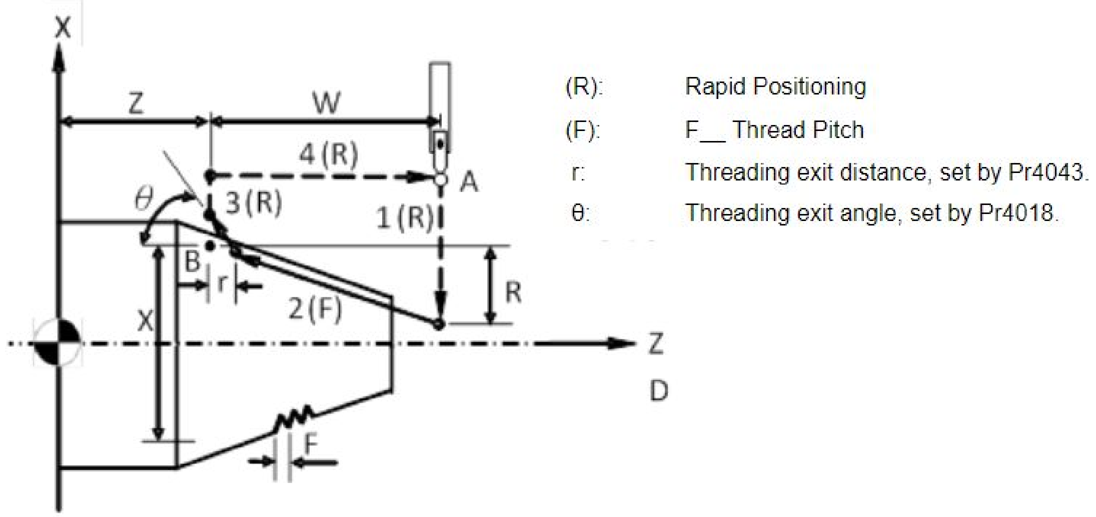
G21 X(U)___ Z(W)___ F___;



(R):	Rapid Positioning
(F):	F__ Thread Pitch
r:	Threading exit distance, set by Pr4043.
θ:	Threading exit angle, set by Pr4018.

Tapered Threading Cycle

G21 X(U)___ Z(W)___ R___ F___;



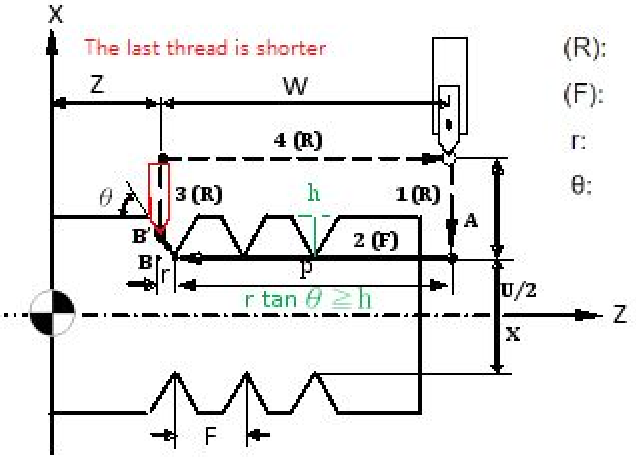
(R): Rapid Positioning
(F): F__ Thread Pitch
r: Threading exit distance, set by Pr4043.
θ: Threading exit angle, set by Pr4018.

Action description

1. Positioning the tool to start point before cycle starts;
2. When executing G21 command, tool moves in X axis to the X(U) position to be fed;
3. Then tool start cutting to the specified X(U), Z(W) position by specified F pitch;
4. After feed ends, the tool returns to start point.
5. After arriving the starting point, the tool will continue to repeat the path according to the X (U) thread depth each time (The difference is the amount of each feed. Refer to the infeed reference table in the G32 Threading command of manual) ;
6. When specified size reached, the tool will stop at starting point.

2.16.2 **Precaution**

1. After 10.114.56E/10.116.0E/10.116.5 (include), the spindle override during threading cycle is locked to the setting before entering the cycle. That is, the override knob in the threading cycle is invalid until the cycle ends.
2. Moreover, before 10.114.56E/10.116.0E/10.116.5, the spindle override is locked to 100% during the infeed and resume knob setting when retracting. So if threading under non-100% spindle override, the spindle will frequently accelerate and decelerate.
3. Value of Pr4018-Threading exit angle (θ) must meet the actual thread cutter angle. For example, the actual thread cutter angle is 60 degrees, then Pr4018 is set to 60;
4. Value of Pr4043 -Threading exit distance (r) must meet the condition r tan θ ≧ h (where h is the thread depth) . If r is too large, it will affect the total length of the thread (W = r + p) . If r is too small, the end point B' of the retraction be on the thread, and the last thread will be lower (refer to the figure below) .



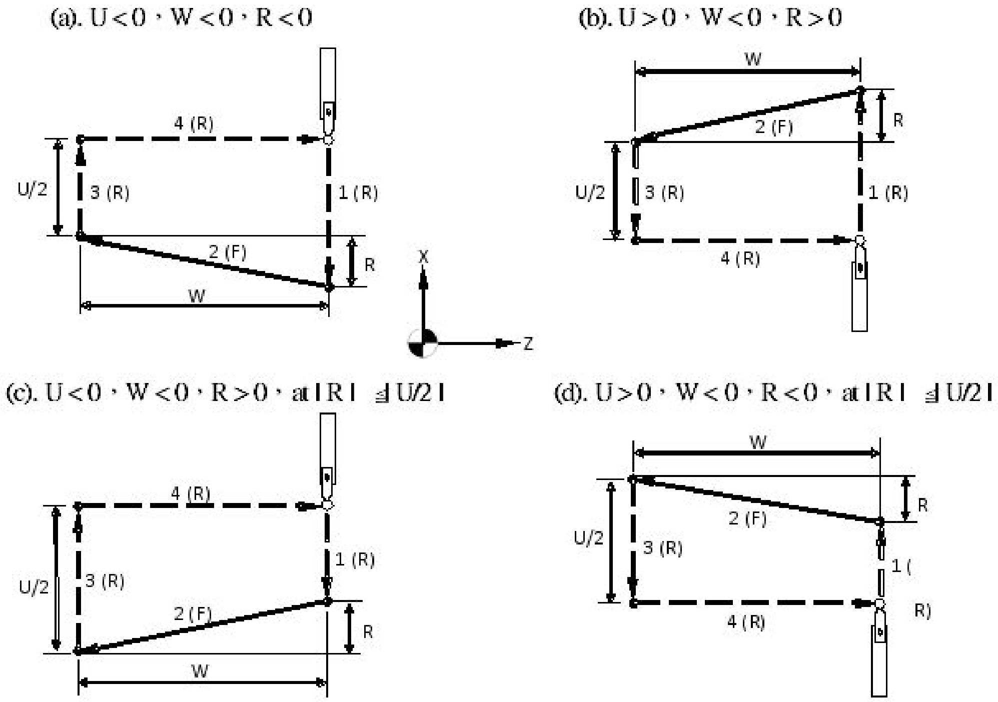
(R): Rapid Positioning

(F): F__ Thread Pitch

r: Threading exit distance, set by Pr4043.

θ: Threading exit angle, set by Pr4018.

5. When the incremental value is used, the relationship between the sign (+/-) of the values after argument U, W and R and the tool path are as follows:


(a). U<0，W<0，R<0

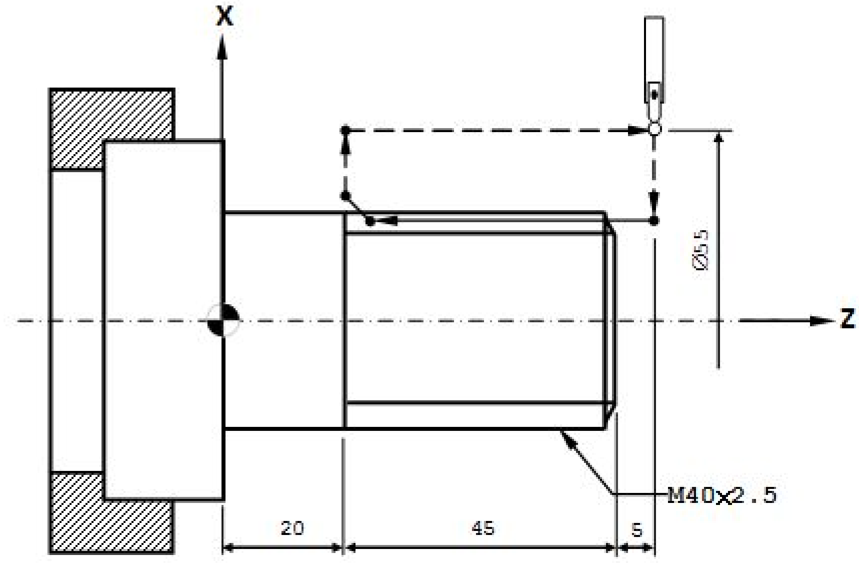
(b). U>0，W<0，R>0

(c). U<0，W<0，R>0，at |R| ≦|U/2|

(d). U>0，W<0，R<0，at |R| ≦|U/2|

2.16.3 **Example 1**

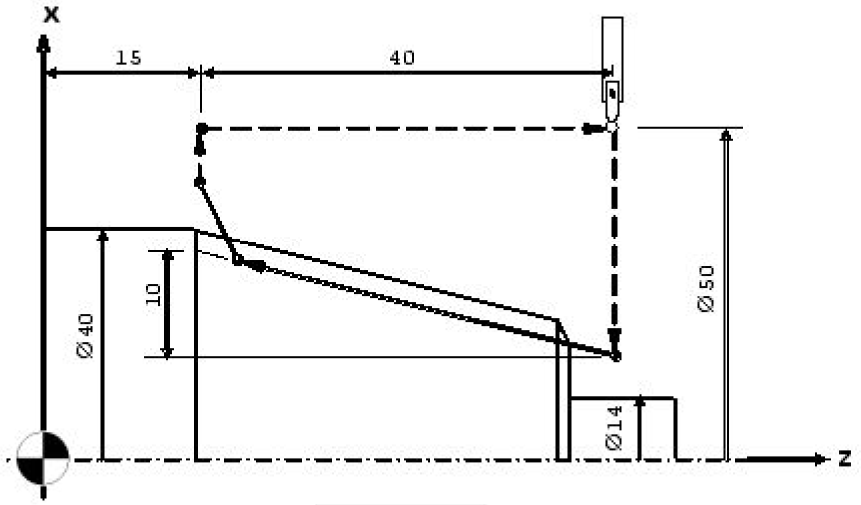
Straight Threading Cycle, **Three-Start Thread**



```
T03; //use tool NO.3
G97 S600 M03; //constant speed at 600 rpm CW
G00 X50.0 Z70.0; //positioning to the starting point of cycle
M08; //cutting liquid ON
G21 X39.0 Z20.0 H3 F2.5;//execute threading cycle, three starts, first cycle
X38.3; //second cycle
X37.7; //third cycle
X37.3; //fourth cycle
X36.9; //fifth cycle
X36.75; //sixth cycle
G28 X60.0 Z75.0; //positioning to specified mid-point and return to machine zero point
M09; //cutting liquid OFF
M05; //spindle stops
M30; //program ends
```

2.16.4 **Example 2**

Tapered Threading Cycle, **Single-Start Thread**

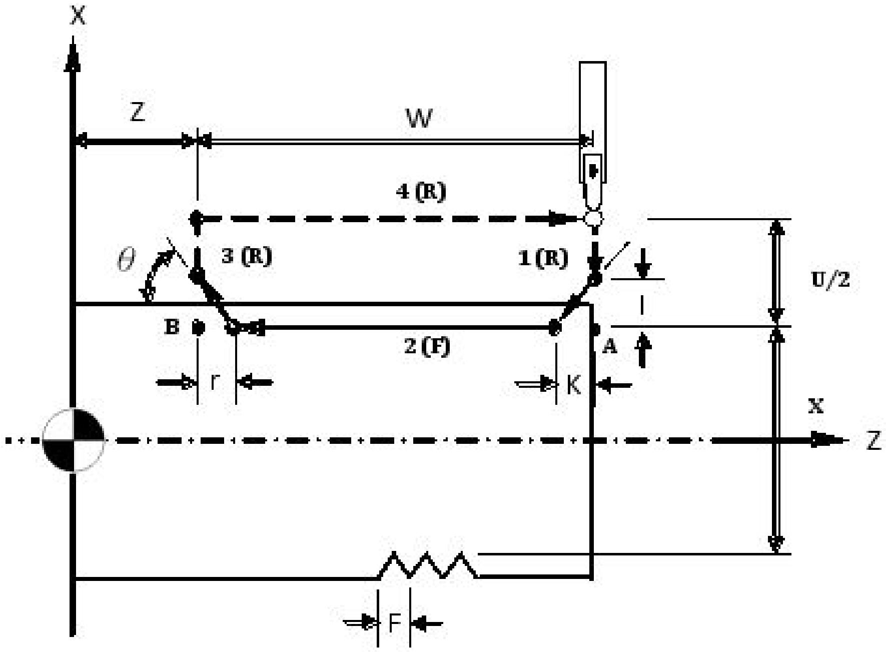


T03; //use tool NO.3
G97 S600 M03; //constant speed at 600 rpm CW
G00 X50.0 Z55.0; //positioning to the starting point of cycle
M08; //cutting liquid ON
G21 X39.0 Z15.0 R-10.0 F2.5; //execute threading cycle, first cycle
X38.3; //second cycle
X37.7; //third cycle
X37.3; //fourth cycle
X36.9; //fifth cycle
X36.75; //sixth cycle
G28 X60.0 Z70.0;//positioning to specified mid-point and return to machine zero point
M09; //cutting liquid OFF
M05; //spindle stops
M30; //program ends

2.17 G21.2- Mid-Section Threading Cycle (C-Type)

2.17.1 **Command Form**

1. Straight Threading Cycle:
   G21.2 X (U) __ Z (W) __ H__ (F__ or E__) ;
   X, Z: Turning end point (absolute value mode)
   U, W: Turning end point (Incremental value mode)
   F: Thread pitch in metric (unit: mm/thread)
   E: Thread per inch (unit: thread/inch)
   H: The number of multi-start thread (ex: H3 means 3 starts. When H is given, F means the pitch of adjacent threads)



(R): Rapid Positioning

(F): F__ Thread Pitch

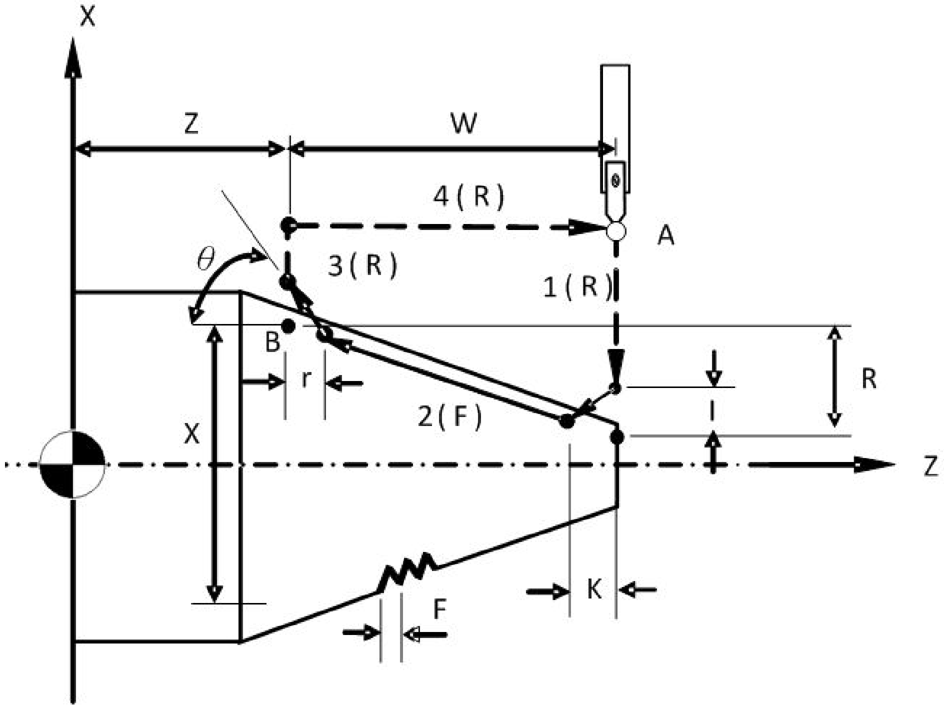
r: Threading exit distance, set by Pr4043.

θ: Threading exit angle, set by Pr4018.

K: Threading infeed distance, set by Pr4046.

I: Threading infeed height, set by Pr4047.

2. Tapered Threading Cycle:
   G21.2 X (U) Z (W) R H (F or E) ;
   X, Z: turning end point (absolute value mode)
   U, W: turning end point (Incremental value mode)
   R: Taper radius difference
   F: Thread pitch in metric (unit: mm/thread)
   E: Thread per inch (unit: thread/inch)
   H: The number of multi-start thread (ex: H3 means 3 starts. When H is given, F means the pitch of adjacent threads)



(R): Rapid Positioning

(F): F__ Thread Pitch

r: Threading exit distance, set by Pr4043.

θ: Threading exit angle, set by Pr4018.

K: Threading infeed distance, set by Pr4046.

I: Threading infeed height, set by Pr4047.

2.17.2 **Description**

The G21 command is a threading cycle which simplifies thread turning, retract, and rapid positioning into single command.
G21.2 is different from G21 in that it provides the infeed parameter setting for threading. When the threading segment is in the middle of a bar and has no clearance to accelerate, G21.2 can be used to avoid the damage of the first thread.
G21.2 provides straight and tapered threading cycles.

X axis only supports setting as diameter axis, and Z axis only supports setting as radius axis.

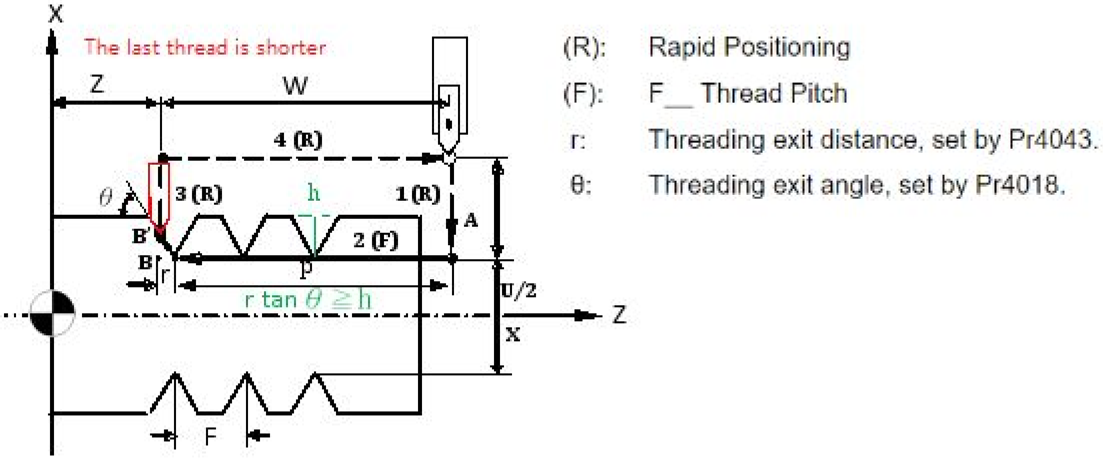
Action description

1. The tool should be rapid positioned to the starting point before cycle start;
2. When executing G21 command, the tool first rapid position in X axis to coordinate X (U) where feed will start.
3. The tool then feeds toward the specified X (U) and Z (W) coordinates at the specified F pitch feedrate;
4. At the end of the feed, the tool automatically rapid position to the starting point;
5. After arriving the starting point, the tool will continue to repeat the path according to the X (U) thread depth each time (The difference is the amount of each feed. Refer to the infeed reference table in the G33 Threading command of manual) ;

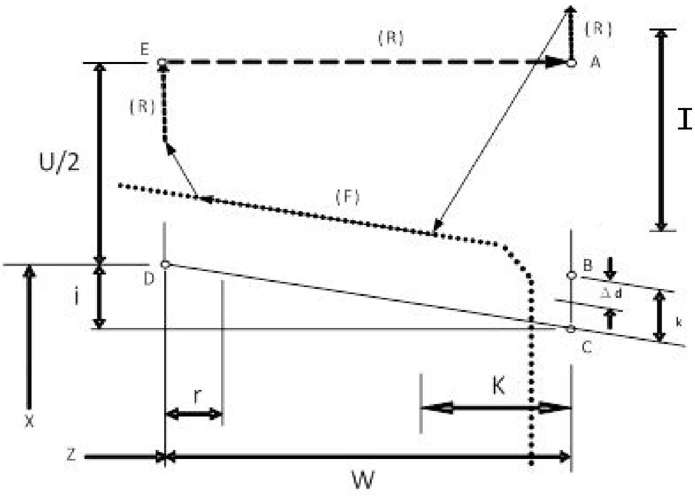
6. When the specified size reached, the tool will stop at the starting point and wait for the next cycle.

2.17.3 **Precaution**

1. When using G21.2 command, please set Pr4046 and Pr4047 correctly; if either 4046 or 4047 is 0, will trigger MACRO alarm 19 "Thread feed has no specified length or height".
2. Value of Pr4018-Threading exit angle (θ) must meet the actual thread cutter angle. For example, the actual thread cutter angle is 60 degrees, then Pr4018 is set to 60;
3. Value of Pr4043 -Threading exit distance (r) must meet the condition r tan θ ≧ h (where h is the thread depth) . If r is too large, it will affect the total length of the thread (W = r + p) . If r is too small, the end point B' of the retraction be on the thread, and the last thread will be lower (refer to the figure below) .

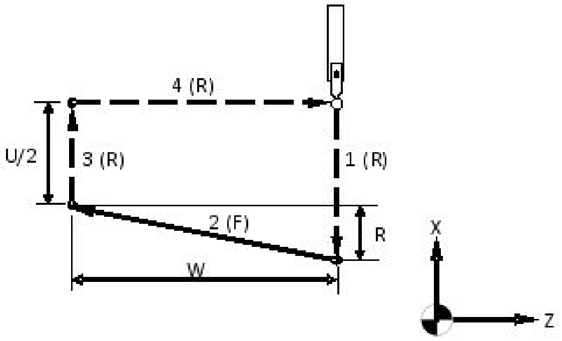

X
The last thread is shorter
Z
W
4 (R)
θ
3 (R)
h
1 (R)
B'
B
2 (F)
A
r
p
r tan θ ≧ h
U/2
X
Z
F
(R): Rapid Positioning
(F): F__ Thread Pitch
r: Threading exit distance, set by Pr4043.
θ: Threading exit angle, set by Pr4018.

4. Pr4046- Threading infeed distance is recommended to be 0.5 pitch and Pr4047 = Pr4046/ tan (0.5*Pr4018) .
5. If two G21.2 commands are used to cut two consecutive threads, besides setting Pr4046 and Pr4047, the second segment must meet the following conditions:
   a. Z-axis feed point position must be equal to a positive integer multiple of the pitch
   b. The distance between the last revolution of thread one and the first revolution of thread two must be a positive integer multiple of the pitch (refer to Example 3) .
6. If the distance of the infeed plus the retract exceeds the total movement of the Z axis, the MARCO alarm 20 "Thread feed/retraction chamfer distance exceeds total Z axis movement" will be triggered.
7. If the infeed position in the X-axis direction is higher than the starting point, MARCO alarm 92 "Threading X-axis feed position higher than the starting point" will be triggered to avoid interference. See the following figure:

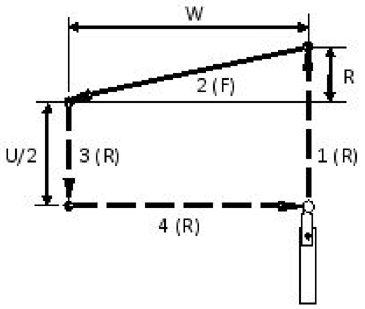


※ When the incremental value is used, the relationship between the sign of the values after argument U, W and R and the tool path are as follows:

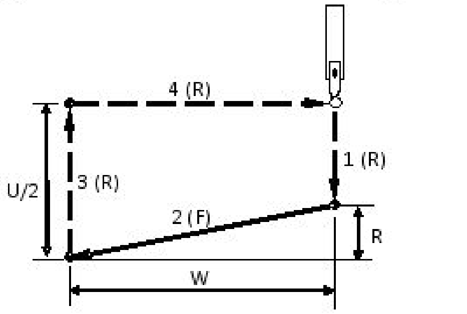
(a). U<0，W<0，R<0



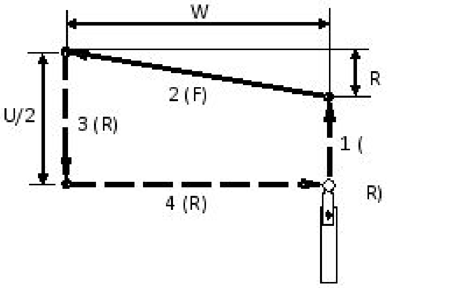
(b). U>0，W<0，R>0



(c). U<0，W<0，R>0，at | R | ≦ | U/2 |

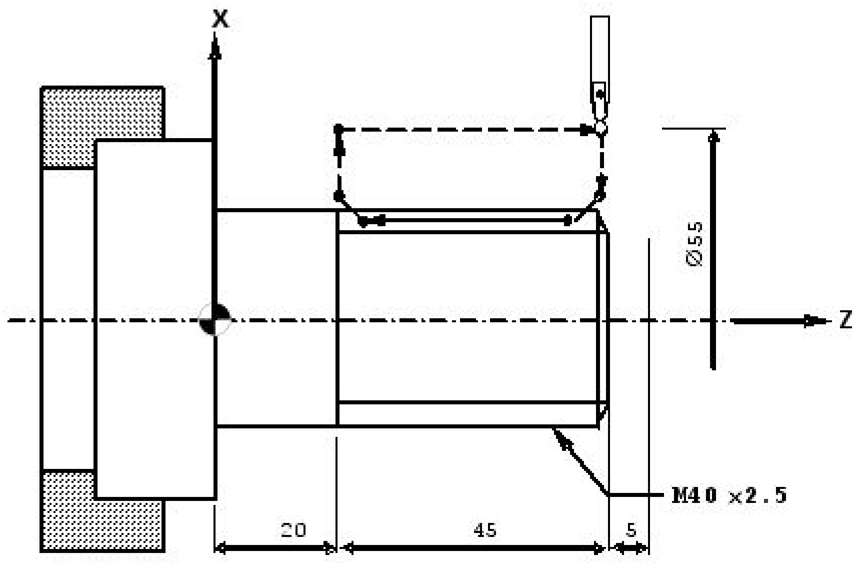


(d). U>0，W<0，R<0，at | R | ≦ | U/2 |



2.17.4 **Example 1**

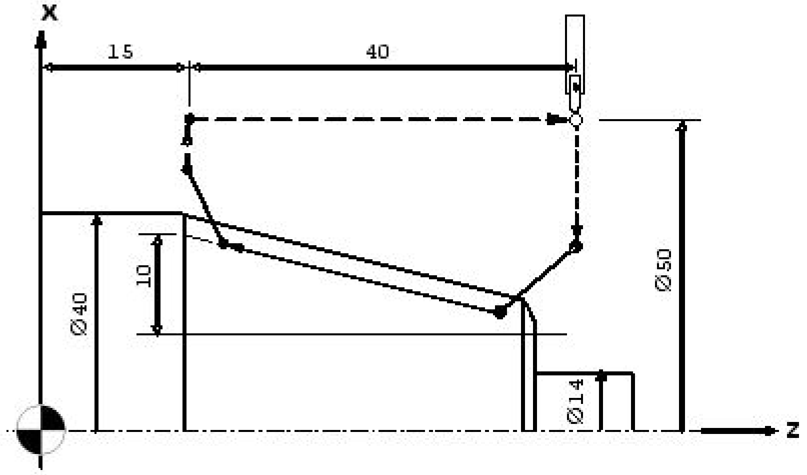
Straight Threading Cycle, Three-Start Thread



```
N03T03; //use tool NO.3
G97 S600 M03; //constant speed at 600 rpm CW
G00 X50.0 Z65.0; //rapid position to initial point
M08; //cutting liquid ON
G21.2 X39.0 Z20.0 H3 F2.5; //perform threading cycle, three starts, first cycle
X38.3; //second cycle
X37.7; //third cycle
X37.3; //fourth cycle
X36.9; //fifth cycle
X36.75; //sixth cycle
G28 X60.0 Z75.0; //positioning to specified mid-point, then return to machine zero
M09; //cutting liquid OFF
M05; //spindle stops
M30; //program ends
```

2.17.5 Example 2

Tapered Threading Cycle, Single-Start Thread



```
N03T03; //use tool NO.3
G97 S600 M03; //constant speed at 600 rpm CW
G00 X50.0 Z55.0; //position to initial point
M08; //cutting liquid ON
G21.2 X39.0 Z15.0 R-10.0 F2.5; //perform threading cycle, three starts, first cycle
X38.3; //second cycle
X37.7; //third cycle
X37.3; //fourth cycle
X36.9; //fifth cycle
X36.75; //sixth cycle
G28 X60.0 Z70.0; //positioning to specified mid-point, then return to machine zero
M09; //cutting liquid OFF
M05; //spindle stops
M30; //program ends
```

2.17.6 Example 3

For a round bar of 20mm long, two G21.2 are used cuts two segments (pitch 2 mm, angle 60) . The first segment is from Z2 to Z-12. The second segment is from Z-6 to Z-20.

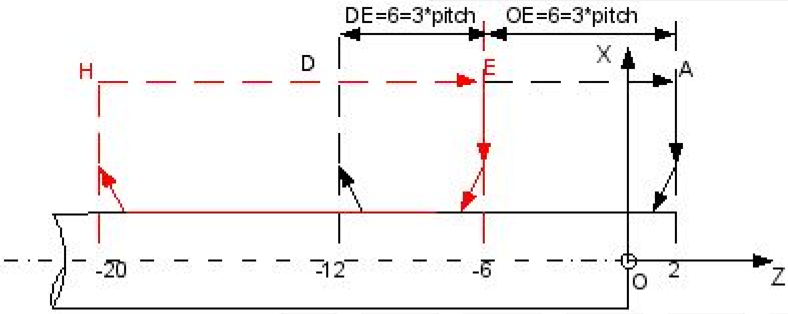
1. Processing program:

```
T0404 //use tool NO.4
M03 S1500 //spindle rotates CW at 1500 rpm
M98 H11 //call the subroutine, starting with N11
M98 H12 //call the subroutine, starting with N12
M30
N11
G0X50. Y0. //rapid position to initial point
Z2. //first infeed point Z2
G21.2X15.65Z-12. F2.0; //first threading cycle, exit (retract) point Z-12, first cycle
X15.25 //second cycle
X14.85 //third cycle
```

```
M99
N12
G0X50. Y0. //position to initial point
Z-6. //second infeed point Z-6
G21.2X15.65Z-20. F2.0; //second threading cycle, exit (retract) point Z-20, first cycle
X15.25 //second cycle
X14.85 //third cycle
M99
```

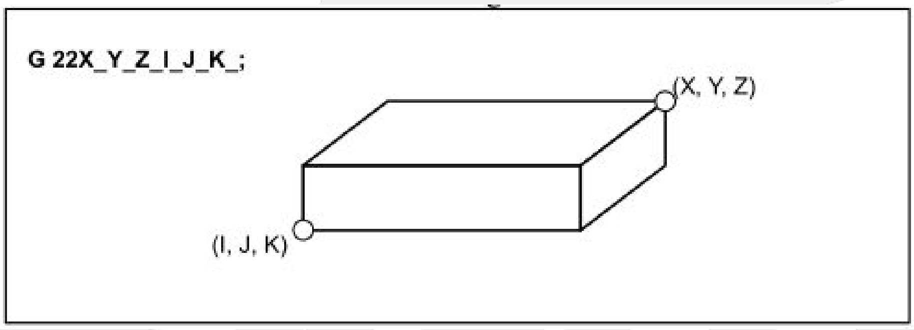
2. Parameter setting

```
Pr4018= 60 //thread cutter angle
Pr4046= 1000 //0.5 pitch= 1mm (unit: LIU)
Pr4043= 5 //0.5 pitch (unit: 0.1 pitch)
Pr4047=1732 //Pr4046*tan60 (unit: LIU)
```



2.18 G22/G23- Enable/Disable 2nd Software Stroke Limit (C-Type)

2.18.1 **Command Form**



G22 X_ Y_ Z_ I_ J_ K_;   // X_ Y_ Z_: Positive limit Machine Coordinate

// I_ J_ K_: Negative limit Machine Coordinate

G23   // Disable software stroke limit protection

2.18.2 **Description**

1. New feature of 10.116.x. The original 2nd software stroke limit was renamed to the 3rd software stroke limit.
2. The G22 can dynamically enables 2nd software stroke limit protection, and can modify the protection range of the XYZ axes in the program.

3. There are three sets of arguments in G22: X-I, Y-J, and Z-K, each of which corresponds to the positive and negative limits of the XYZ axes.
4. G23 is to disable second software stroke limit function.
5. The positive and negative limits of the second software stroke are set in Pr2501~2540.
6. The protection range can be determined by Pr2542 to the inside or outside of the set value.
7. The default state after power-on can be determined by Pr3838 as G22 or G23.
8. If there is an argument after G22, then the value specified by the argument is used as the range of stroke limit. Parameter value is unchanged. The following table clarifies the scope of protection for different command format.

Program command	X	Y	Z	Other axis
G22	parameter	parameter	parameter	parameter
G22 X_	COR-109 G22 wrong command, ail to enable.			
G22 X_ I_	command	parameter	parameter	parameter
G22 X_ Y_ Z_ I_ J_ K_	command	command	command	parameter

9. The arguments in the same group (X & I, Y & J, Z & K) can be reversed in order and the protection range is the same. For example, G22 X100. I200. has the same protection range as G22 X200. I100.
10. If difference of arguments in the same group is 0, the protection is not enabled even if the parameters are set. Example:
    G22 X0. I0. indicates X axis protection not enabled.
    G22 X10. I10. indicates X axis protection not enabled.
    G22 X0. I10. indicates X0. ~ X10. is the protection range.
    G22 X10. I0. indicates X0. ~ X10 is the protection range.. and so on.
11. When the G22 is used under main axis group ($1), it only apply stroke limit protection to first axis group. Axes of the second axis group are not affected, and vise versa.
12. In the second axis group, G22 X_ Y_ Z_ I_ J_ K_ command will apply protection range to X2, Y2, Z2 axes.

2.18.3 **Precaution**

1. Pressing the Reset button can not disable the G22 protection status, must use G23 to disable.
2. Enable/disable function is effective only in the next block of the G22/G23 command.
3. The positive limit of the 2nd software stroke protection of each axis needs to be greater than the negative limit, otherwise the protection is not effective in that axis.
4. Please use the version after 10.116.x
   P.S. For more instructions, see Software Stroke Limit Application Manual.

2.19 G24- End Face Turning Cycle (C-Type)

2.19.1 **Command Form**

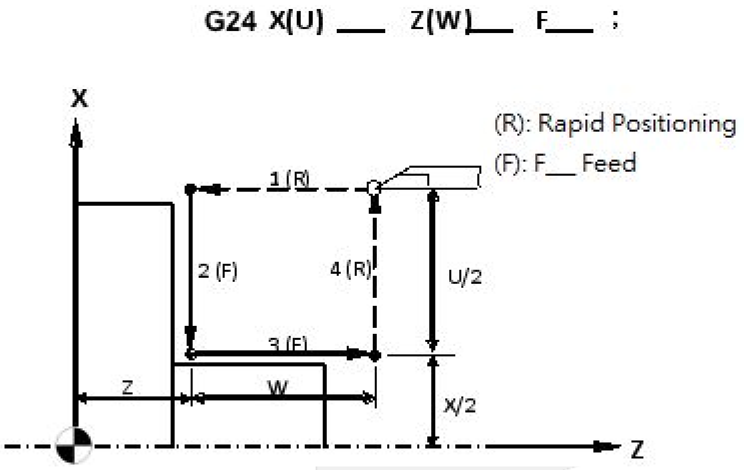
1. Radial straight turning cycle: G24 X(U)___ Z(W)___ F___;
2. Radial tapered turning cycle: G24 X(U)___ Z(W)___ R___ F___;

X, Z: End position coordinate of turning (absolute)
U, W: End position coordinate of turning (incremental)
R: Difference Z length from starting point to end point
F: Feedrate

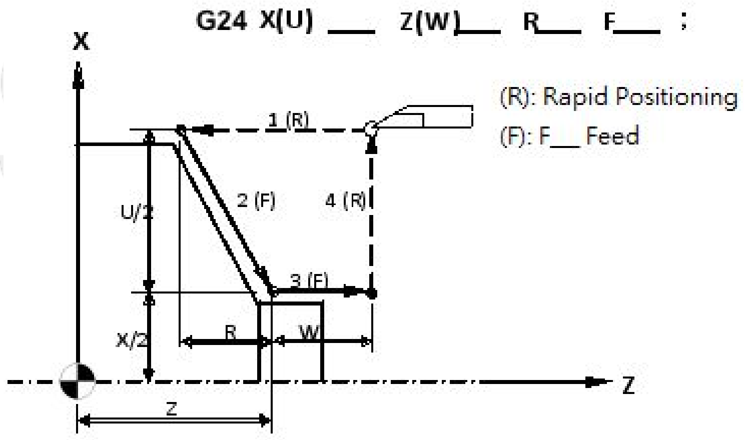
2.19.2 **Description**

G24 command is end face turning cycle. It simplifies several repeating end face turning blocks into one single block.

Radial Straight Turning Cycle



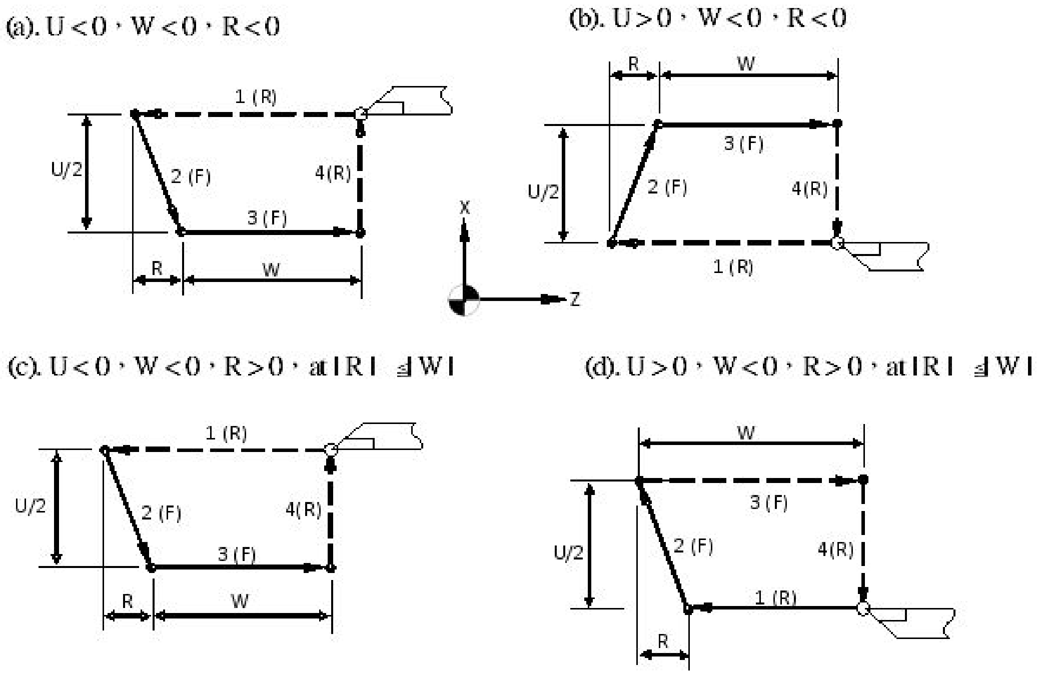
Radial Tapered Turning Cycle



2.19.3 **Action description**

1. Positioning the tool to start point before cycle starts;
2. When executing G24 command, the tool will move along Z direction and reach the specified Z(W) position;
3. Then the tool will feed toward X(U), Z(W) position by specified feedrate;
4. After feed finishes, the tool returns to start point;
5. After reaching the start point, tool will repeat turning path by the next Z(W) value.
6. When reaching the specified size, the tool will stop at starting point and wait for next cycle.

※ In increment mode, the relationship between the signs of U, W, R arguments (+/-) and the tool path are as below:


(a). U<0，W<0，R<0

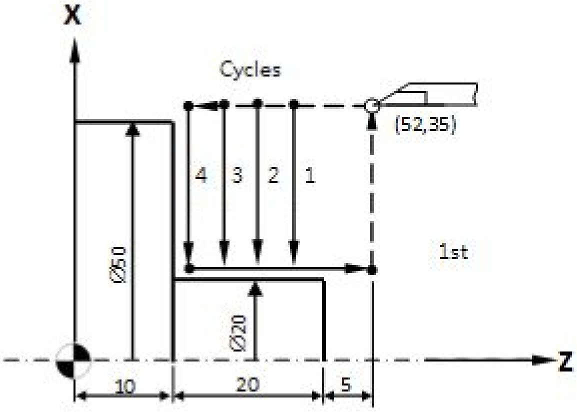
(b). U>0，W<0，R<0

(c). U<0，W<0，R>0，at |R| ≦|W|

(d). U>0，W<0，R>0，at |R| ≦|W|

2.19.4 **Example 1**

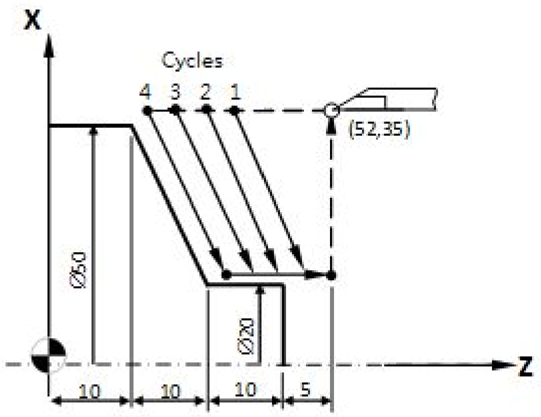
Radial Straight Turning Cycle



```
G50 S3000; //maximum speed 3000 rpm
T01; //use tool NO.1
G96 S130 M03;//constant speed, surface speed 130 m/min
M08; //cutting liquid ON
G00 X52.0 Z35.0; //positioning to the starting point of cycle
G24 X20.0 Z25.0 F0.6; //perform radial straight 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; //positioning to specified mid-point and return to machine zero point
M09; //turning liquid OFF
M05; //spindle stops
M30; //program ends
```

2.19.5 **Example 2**

Radial taper turning cycle



```
G50 S3000; //maximum speed 3000 rpm
T01; //use tool NO.1
G96 S130 M03; //constant speed, surface speed 130 m/min
M08; //cutting liquid ON
G00 X52.0 Z35.0; //positioning to the starting point of cycle
G24 X20.0 Z32.0 R-10.0 F0.6; //perform 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; //positioning to specified mid-point and return to machine zero point
M09; //turning liquid OFF
M05; //spindle stops
M30; //program ends
```

2.20 G28- Return To Reference Point (C-Type)

2.20.1 **Command Form**

G28 X(U)__Z(W)__

X , Z: Specified mid point (absolute mode)

U , W: Specified mid point (incremental mode)

2.20.2 **Description**

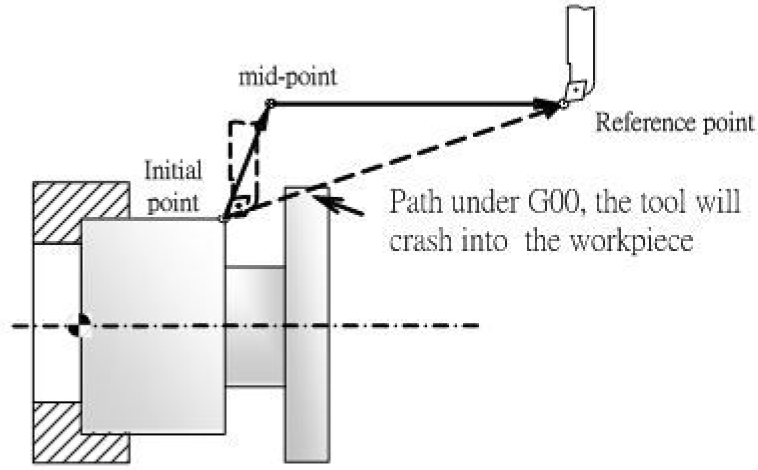
When G28 command is executed, tool will move to specified mid point and then return to reference point (machine zero point) by the speed of G00. To prevent interference between tool and workpiece, G28 keep tool clear of the workpiece when returning.

In absolute mode, it use absolute coordinate of the mid point. In increment mode, it is the increased value from start point to the mid point.

※ Notice:

1. Tool compensation must be disabled prior to G28 command to insure the returning action correct.
2. After the software version 10.116.10P & 10.116.16G, the block will not execute and skipped directly if G28 has no argument. If G29 is used after G28, it will also be skipped because G29 cannot find the mid point.

2.20.3 **Illustration**



2.20.4 **Additional Remark**

If the Axis Type (Pr221~236) is rotational axis, see the attachment “parameter manual” for your reference.

2.21 G29- Return From Reference Point (C-Type)

2.21.1 Command Form

G29 X(U)_Z(W)_ ;

X , Z: Specified point (absolute mode)

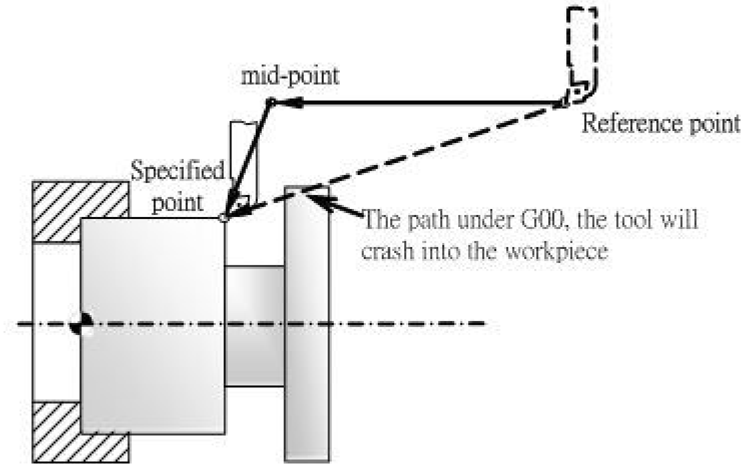
U , W: Specified point (incremental mode)

2.21.2 Description

G29 command is used in conjunction with return to reference point return (G28), to quickly move to specified point via mid-point. Noted that **G29 cannot to be executed alone** because it does not specify its own mid point, instead it uses the mid point from G28. So G29 can only be executed after G28 command.

In absolute mode, it is the absolute value to the specified point. In increment mode, it is the incremental value from starting point (reference point) to specified point.

Illustration



2.22 G30- Any Reference Point Return (C-Type)

Command Form

G30 Pn X(U)_Z(W)_

X , Y , Z: Mid-point coordinate

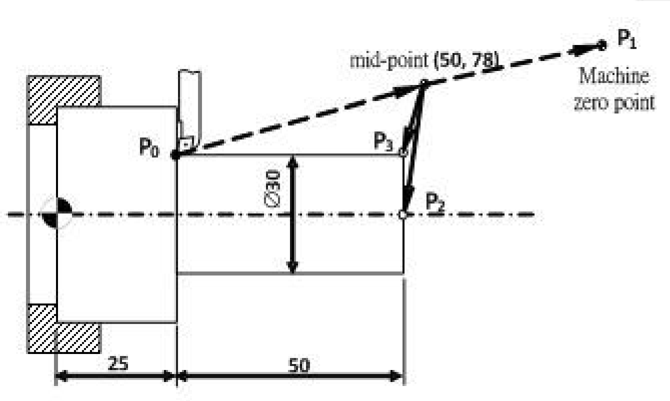
Pn: Reference point (setting parameter #2801 ~ #2856)
P1: Machine zero point
P2: Second reference point
Default is P2 when P__ neglected

2.22.1 **Description**

In the convenience of tool change and inspection, a reference point is specified by parameters a distance away from the machine zero point so the machine has no need to return to machine zero point to change tool and efficiency is increased. It is very similar to G28 except the different Reference Point. G30 is usually used in the case which tool changing position differs from machine zero. The movement is G00 (rapid positioning) mode.

<Notice> This command is usually used in auto tool change. Always disable tool compensation before executing G30 for safety.

2.22.2 **Example**



Return Path 1:
G30 P01 X50.0 Z78.0 // P_0 →mid-point→ P_1

Return Path 2:
G30 P02 X50.0 Z78.0 // P_0→ mid-point→ P_2
or
G30 X50.0 Z78.0 //default P_2

Return Path 3:
G30 P03 X50.0 Z78.0 // P_0→ mid-point→ P_3

2.23 G31 - Skip Function (C-Type)

2.23.1 **Command Form**

G31 X(U)__ Z(W)__ F__ Q__ P__;

X, Z: Specified position (absolute mode)

U, W: Specified position (incremental mode)

F: Feedrate

Q: "101~132" specifies the C BIT of corresponding skip signal; "201~216" specifies signal from Yaskawa servo drive EXT1 from axis 1~16. C62 is default signal source if no Q argument given.

P: Deceleration time (ms). G31 movement will stop immediately if no P argument or P0.

Note: Please consult the machine builder to check signal source is C62, C101~C132, or Yaskawa serial driver EXT1. .

2.23.2 **Description**

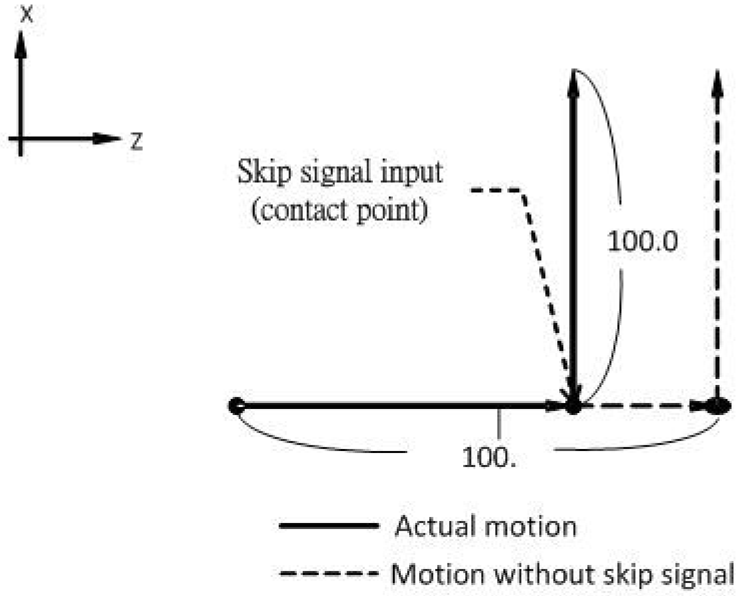
1. The skip function is used probe an unknown target by assigning an approximate end point. When the **probe** touches measurement target and triggers PLC **C BIT** signal, system will records the current mechanical position, skips the unfinished movement, and continuing to the next block. The feed rate of the G31 is the same as that of the G01, specified by the F code.
2. When Q is not specified, C62 is preset as the skip signal.
3. Q101 specifies C101s the skip signal, and so on.
4. If skip signal is triggered at the beginning of the block, it is regarded as encountering the skip signal. G31 movement will stop immediately to execute next block.
5. When P argument is not specified or specified as zero, there is no deceleration and the command is directly interrupted.
6. When P argument is specified, decelerated by deceleration time. It will stop at the block end if the deceleration time is not enough.
7. The P argument must be a non-negative integer, otherwise alarm COR-64 will be issued.

Additional Remark

a. To only skip one G31 command when continuous G31 are used with the PLC interface, upper edge trigger of C BIT can avoid multiple G31 being skipped.
b. If the G31 end point is too close to the skip signal position, it may sometimes cause G31 block to finish first, and then PLC C-bit signal or the driver's signal come in later. This is a case that G31 has no time to skip while skip signal was correctly triggered. Take tool measuring with probe as example, it is recommended to move G31 end point lower to avoid this case.

2.23.3 **Example 1**

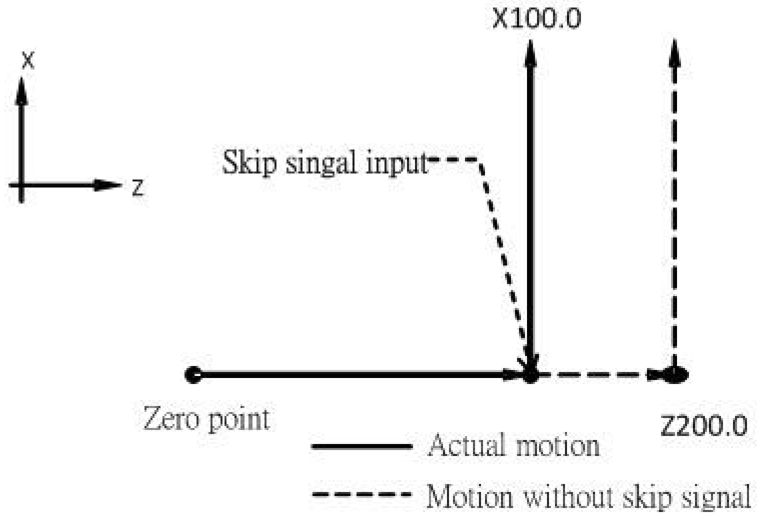
Increment mode, single axis movement.



G31 W100.0 F0.1; //Original path until touches contact point
U100.0; //Skip previous block, use contact point to be the relative position and change path to specified distance

Example 2

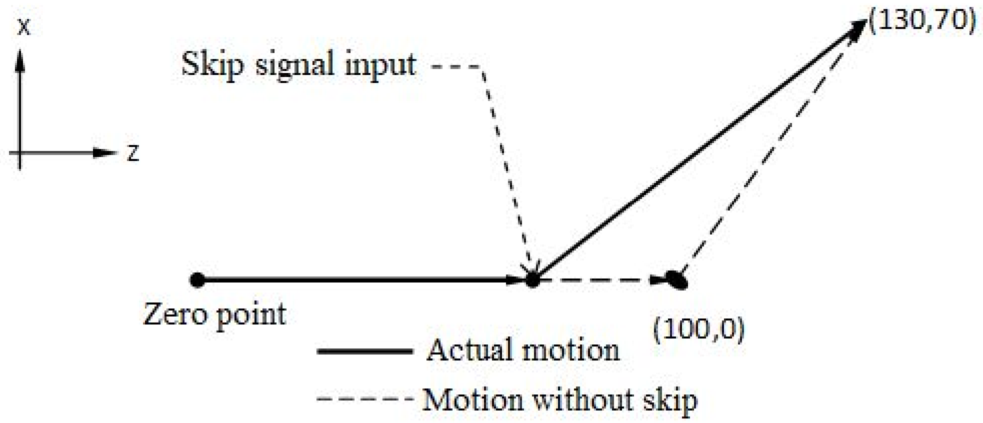
Absolute mode, single axis movement.



G31 Z200.0 F0.1；　//Original path until touches contact point
X100.0；　//Skip previous block, move from contact point to specified position

2.23.4 Example 3

Absolute mode, two axes movement.



G31 Z100.0 F1.0; //Original path until touches contact point
Z130.0 X70.0; //Don't wait for previous block finishing execution. From the signal point
//move to specified position

2.24 G31.10/G31.11： Multi-Axis Multi-Signal Skip Function(C-Type)

2.24.1 **Command Form**

G31.10 X_ Y_ Z_ F_ Q_ P_

Set multi-axis multi-signal skip function

X, Y, Z: Specified Position

F: Feedrate

Q: Skip source reference

P: Deceleration time (ms)

G31.11

Execute multi-axis multi-signal skip function

2.24.2 **Description**

1. A multi-axis multi-signal skip function is composed of one G31.10 at least and one G31.11. Set first then execute.
   Both commands should be used in a multi-axis multi-signal skip function, and it is not allowed to insert other commands in between.
   a. Set multi-axis multi-signal skip function(G31.10)：
      i. Supports up to six G31.10 commands in a set of multi-axis multi-signal skip function. Skip source and deceleration time can be different in each G31.10 command.
      ii. In each G31.10, every axis specified in this command will move, stop and skip at the same time.
      iii. If the specified axis is the virtual axis set by G10 L801, all settings of F, P and Q will be applied to corresponding axis.
   b. Execute multi-axis multi-signal skip function(G31.11)：
      Execute multi-axis multi-signal skip function with previous settings of G31.10.
      When the skip signal is triggered, the axis corresponding to the skip source skips.
2. The unit of position can be mm or inch, depends on which unit setup is used(G70/G71).
3. Feedrate F：
   a. Previous feedrate will be considered while F argument is not specified.
   b. unit：
      mm/min(inch/min) in G94.
      mm/rev(inch/rev) in G95.
4. Skip source reference Q：
   a. C62 is the default skip source while Q argument is not specified
   b. The signal source of **Q101~Q132** is C-bit, corresponds to **C101~C132** respectively.
   c. The signal source of **Q201~Q218** is the **external signal (EXT) source** of serial drivers, it detect the signal of the 1st ~ 18th axis respectively. The supporting serial drivers are listed below.

Supporting Serial Drivers	**External Signal (EXT) Source**
M2	EXT1

Supporting Serial Drivers	External Signal (EXT) Source
M3	EXT1
RTEX	EXT1
EtherCAT	EXT1

Note: Syntec M2 doesn't support.

5. Deceleration time P：
    a. When P argument is not specified or P0 is given, there is no deceleration and the command will be suspended directly.
    b. When P argument is given, deceleration time will be considered in deceleration planning.
    Axis reaches the position specified by G31.10 when deceleration time is so long that the brake distance exceeds the destination of this block.

2.24.3 **Notice**

1. #1361~#1378, #1441~#1458, #1608 are set to 0 when CNC just powered on, RESET, system executes G31 or G31.11 or G28.1 again.
   To avoid showing the previous escape location before skip signal comes in and causes misjudgment.
2. If the final point location of G31.10 & G31.11 is too close to the skip signal triggering location, it might occasionally lead to the situations that the PLC scanned the C-bit signal after G31.11 is finished. In the situations, although the signal is triggered, but it's too late for G31.11 to execute the skip. Take the tool measuring action as example, when the situation happened, it's better to move the G31.10 & G31.11 block deeper to avoid the situation.
3. The P argument should be bigger than 0 or an integer, or alarm COR-064 will issued.
4. Alarm COR-362 issues when only G31.10 or G31.11 command is given, or inserting other commands between G31.10 and G31.11.
5. Alarm COR-362 issues when the same axis is used in different G31.10.
6. Alarm COR-362 issues when G31.10 is given more than 6 times in a multi-axis multi-signal skip function.
7. The functions below are not supported in multi-axis multi-signal skip function：
    a. G5.1 (Path Smoothing)
    b. G12.1/G13.1 (Activate/Deactivate Polar Coordinate Interpolation)
    c. G15/G16 (Polar Coordinates Command Mode)
    d. G40/G41/G42 (Cutter Radius Compensation)
    e. G43.4/G43.5 (Rotate Tool Center Point (RTCP) Type1 & 2)
    f. G10 L16 (Virtual circle radius)
8. While executing multi-axis multi-signal skip function, F(command) displays the latest F command given by NC file, F(actual) displays the value of current resultant velocity. Therefore, F(actual) may be greater than F(command). The block feedrate command of G31.11 while interpolation can be obtained by using variable K62.
   example

   sample code

   ```
   G90 G71
   ```

```
G31.10 Z1=10. F300. Q101 P100   // set Z1 goes to 10 by F300, the skip source
is C101, deceleration time is 100 ms
G31.10 Z2=20. F400. Q102 P100   // set Z2 goes to 20 by F400, the skip source
is C102, deceleration time is 100 ms
G31.11                          // execute multi-axis multi-signal skip
function
M30
```

When Z1 and Z2 reach F300 and F400 respectively, the display result：

F(command)：400 mm/min

F(actual)：500 mm/min

The block feedrate command of G31.11 while interpolation can be obtained by variable K62.

The actual compound feedrate and axis feedrate can be obtained by R700 and R701~718：

- R700：Actual compound feedrate command, unit: LIU/min.
- R701~718：Velocity of each axis, Servo On mode: according to command value; Servo Off mode: according to feedback value, unit : BLU/min.

R700：500 IU/min = 500000 LIU/min

R701：300 IU/min = 300000 BLU/min (if Z1 is the first axis)

R702：400 IU/min = 400000 BLU/min (if Z2 is the second axis)

9. If an escape signal is triggered while **cross-coordinate axis coupling** is enabled, the command of the master axis will be interrupted, **and the slave axis commands will also be interrupted accordingly** **(applicable only to coupling types 2 to 5)**. However, note that the escape machine position recorded by the slave axis group will not be correct in this case.
   - To ensure the recorded position is correct, you can use the axis borrowing function to borrow the coupled axis into the same coordinate, and then use the G31.11 to trigger the escape signal.
10. Valid version : after 10.118.40G, 10.118.44 (included).
11. The 3rd-party EtherCAT drive support version: 10.118.82N, 10.118.86G, 10.118.90C, 10.118.94 and later versions.

2.24.4 **Example**

Example 1: each axis moves and skips with the settings of G31.10.(the feedrate of all G31.10 refers the previous one)

example 1

```
G90
F100.
G31.10 Z1=10. Q101 P100 // set Z1 to move to 10 by F100, skip source is C101, set
deceleration time to 100 ms
G31.10 Z2=20. Q102 P100 // set Z2 to move to 20 by F100, skip source is C102, set
deceleration time to 100 ms
G31.11                  // execute multi-axis multi-signal skip function
M30
```

The value of #1608 when all skip source are triggered.

There are axes: X, Y, Z1, Z2, Z3, Z4， the corresponding Pr21~ and Pr321~ are listed below。

When all skip source are triggered, the value of each bit of #1608 is：

bit 0： 0。(supports G31 only)
bit 1~18：

Command	**G31.10 & G31.11**					
Axis corresponding axis card port number(Pr21~)	Pr21	Pr22	Pr23	Pr24	Pr25	Pr26
Axis name(Pr321~)	X	Y	Z_1	Z_2	Z_3	Z_4
bit	1	2	3	4	5	6
value	0	0	1	1	0	0

Therefore, the value of #1608 is $2^3+2^4=24$。

Example 2:each axis moves and skips with the settings of G31.10.(the second feedrate of G31.10 refers the one of the first G31.10)

example 2

```
G90
G31.10 Z1=10. F100. Q101 P100   // set Z1 to move to 10 by F100, skip source is C101,
set deceleration time to 100 ms
G31.10 Z2=20. Q102 P100         // set Z2 to move to 20 by F100, skip source is C102,
set deceleration time to 100 ms
G31.11                          // execute multi-axis multi-signal skip function
M30
```

The value of #1608 when all skip source are triggered.

There are axes: X, Y, Z1, Z2, Z3, Z4， the corresponding Pr21~ and Pr321~ are listed below。

When all skip source are triggered, the value of each bit of #1608 is $2^3+2^4=24$。

Command	**G31.10 & G31.11**					
Axis corresponding axis card port number(Pr21~)	Pr21	Pr22	Pr23	Pr24	Pr25	Pr26
Axis name(Pr321~)	X	Y	Z_1	Z_2	Z_3	Z_4

Example 3: use multi-axis multi-signal skip function with virtual axis setting

When using G10 L801, all axes are considered as the same axis. Therefore, all axes apply the same skip function setting.

example 3

```
G90
G10 L801 P300 Q0        // cancel virtual axis Z
G10 L801 P300 Q301      // virtual axis Z corresponding to Z1
G10 L801 P300 Q302      // virtual axis Z corresponding to Z2
G10 L801 P300 Q303      // virtual axis Z corresponding to Z3

G31.10 Z10. F100 Q101   // set virtual axis Z to move to 10 by F100. i.e., Z1, Z2 and
Z3 move to 10 by resultant velocity F100
                        // skip source is C101 and there is no deceleration time
G31.11                  // execute multi-axis multi-signal skip function
M30
```

The value of #1608 when all skip source are triggered.

There are axes: X, Y, Z1, Z2, Z3, Z4， the corresponding Pr21~ and Pr321~ are listed below。

When all skip source are triggered, the value of each bit of #1608 is $2^3+2^4+2^5=56$。

Command	**G31.10 & G31.11**					
Axis corresponding axis card port number(Pr21~)	Pr21	Pr22	Pr23	Pr24	Pr25	Pr26
Axis name(Pr321~)	X	Y	Z_1	Z_2	Z_3	Z_4

Example 4: Z1 and Z2 move simultaneously, Z3 moves independently.

example 4

```
G90
G31.10 Z1=10. Z2=20. F100 Q101  // Z1 and Z2 move to 10 and 20 respectively, the
resultant velocity is 100
                                // skip source is C101 and there is no deceleration
time
G31.10 Z3=30. Q103              // set Z3 to move to 30 by F100, skip source is C101
and there is no deceleration time
G31.11                          // execute multi-axis multi-signal skip function
M30
```

The value of #1608 when all skip source are triggered.

There are axes: X, Y, Z1, Z2, Z3, Z4， the corresponding Pr21~ and Pr321~ are listed below。

When all skip source are triggered, the value of each bit of #1608 is 2^3+2^4+2^5=56。

Command	**G31.10 & G31.11**					
Axis corresponding axis card port number(Pr21~)	Pr21	Pr22	Pr23	Pr24	Pr25	Pr26
Axis name(Pr321~)	X	Y	Z_1	Z_2	Z_3	Z_4

2.25 G33- Thread cutting (C-Type)

2.25.1 Command Form

(1) Straight thread cutting: G33 Z(W)_Q___ (F___ or E___)

(2) Tapered thread cutting: G33 X(U)_Z(W)_Q___ (F___ or E___)

(3) Endface thread cutting: G33 X(U)_Q___ (F___ or E___)

X , Z: Specified position (absolute mode)

U , W: Specified position (incremental mode)

F: Pitch in longitudinal direction (G70: inch, G71: mm)

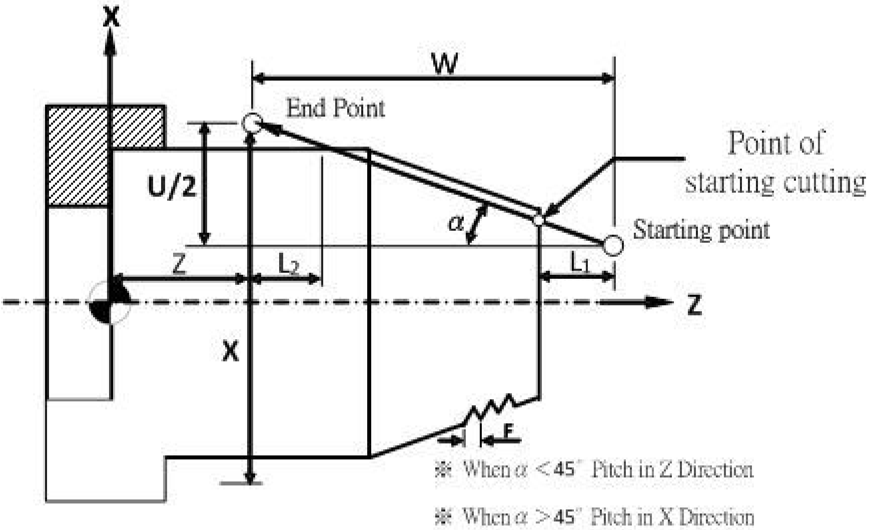
E: Unit pitch number in longitudinal direction (number of pitches per inch)

Q: Angle offset at thread start. To let the tool start cutting at the same angle of a rotating workpiece, usually used in multiple-thread cutting. For single-thread cutting, ignore the Q argument and apply default value Q= 0°(range：0.001~360.000∘)

2.25.2 Description

G33 command executes endface thread cutting, taper threading, and straight thread cutting by synchronously rotating workpiece and feeding tool to create constant pitch thread.

Illustration



※ When α <45° Pitch in Z Direction

※ When α >45° Pitch in X Direction

※ When α =45° Allowed in Z or X direction

2.25.3 Notification

【Note 1】If the converted feedrate is greater than Max. cutting feedrate, the pitch become inconsistent and not the originally specified value.

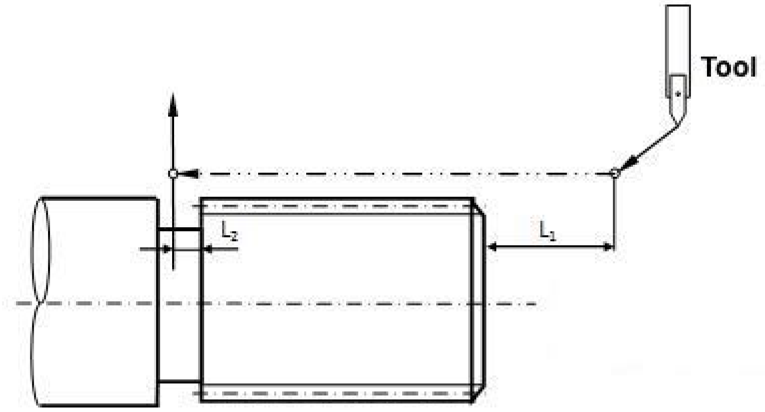
1. Tilt thread cutting command and spiral thread cutting command are unavailable in constant surface speed mode.
2. The spindle speed should be fixed from coarse cutting to fine cutting.
3. The thread will be damaged if dwell is used in thread cutting, avoid dwell in this command. If "FeedHold" button pressed under when thread cutting, the cutting will be terminated (not in G33 mode) and will hold at the end of next block.
4. In the beginning of thread cutting, adjusting cutting feed rate over max feed rate will trigger alarm【Note 1】.
5. In the thread cutting, it is possible that the adjusting cutting feed rate exceeds the max cutting feed rate for keeping the constant pitch.
6. The limitation of spindle RPM is as below:

   $$1 \leqq R \leqq \frac{\text{Max feedrate}}{\text{Thread feed}}$$

   R: spindle rotate speed (rpm)
   Thread feed (F): mm or inch
   Feedrate: mm/min or inch/min
7. Around the start and end of thread cutting point, incorrect pitch length could occur due to servo lag. Therefore the command thread length should be additional (L_1 ,L_2) plus the wanted thread length.
   ※L_1 , L_2 formula are as below:

$$L_1 \approx \frac{S \times P}{400}$$

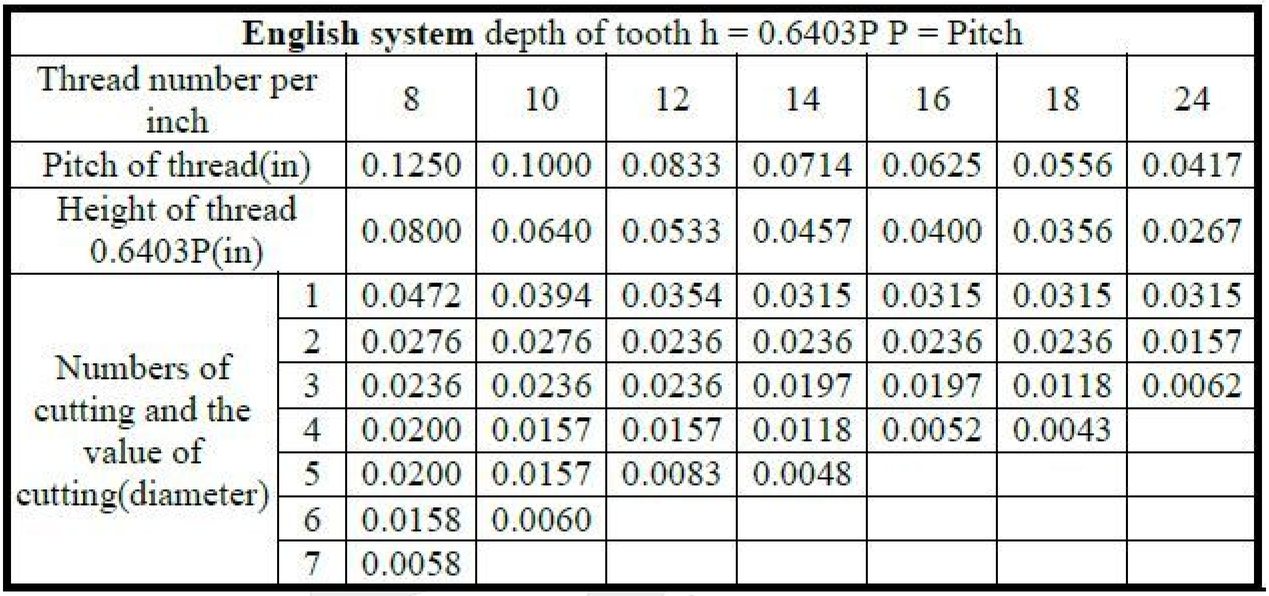
$$L_2 \approx \frac{S \times P}{1800}$$



L1: the minimum distance that cutting tool accelerate from halt state to specified speed of threading.
L2: the minimum distance that cutting tool decelerate from specified speed of threading to halt state.

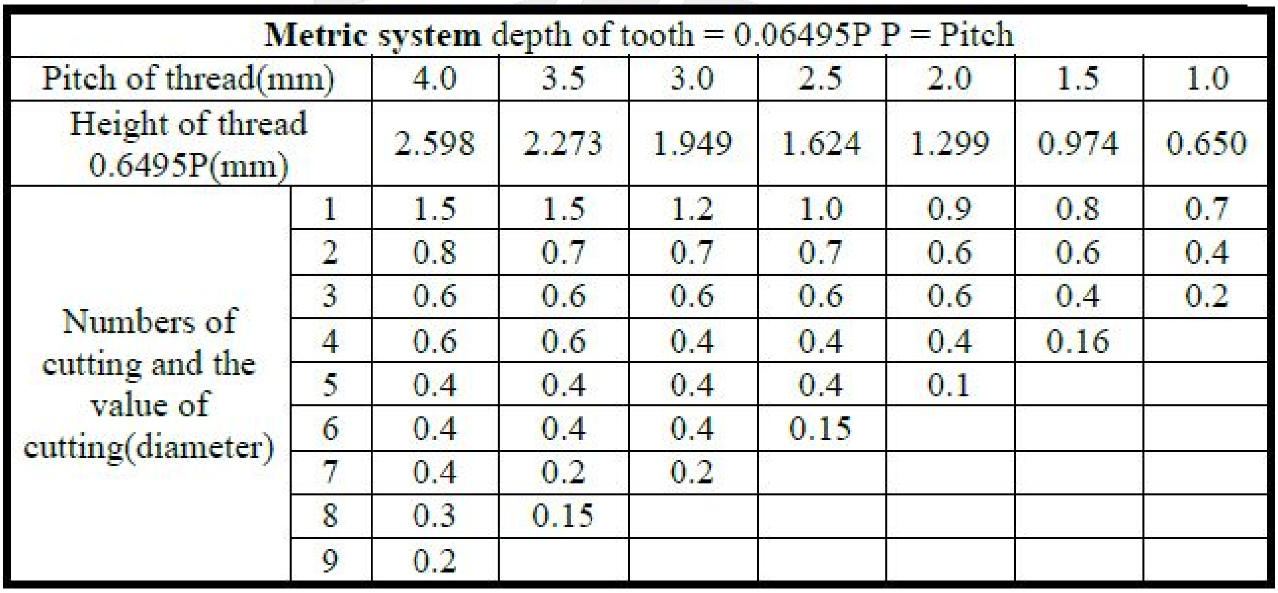
8. The external speed control is effective during the thread cutting, but the feedrate of external speed control cannot synchronize with spindle revolution.
9. In non-synchronous feed (G98) mode, the thread cutting command will become synchronous feed mode.
10. During the thread cutting, manual adjustment of spindle RPM is also effective. Manually adjust the RPM during thread cutting can cause incorrect thread cutting due to servo lag.
11. If thread cutting command executed with tool compensation enabled, tool compensation will be temporarily canceled.
12. When G33 executing, thread cutting will be canceled if system changed to other automatic modes (MDI) and system terminates program after the block finishes.
13. When G33 executing, thread cutting will be canceled if system changed to manual mode (Jog/InJog/MPG), and system terminates program after the block finishes.

Reference Table of Tool Feed of Thread Cutting:



English system depth of tooth h = 0.6403P P = Pitch

Thread number per inch		8	10	12	14	16	18	24
Pitch of thread(in)		0.1250	0.1000	0.0833	0.0714	0.0625	0.0556	0.0417
Height of thread 0.6403P(in)		0.0800	0.0640	0.0533	0.0457	0.0400	0.0356	0.0267
Numbers of cutting and the value of cutting(diameter)	1	0.0472	0.0394	0.0354	0.0315	0.0315	0.0315	0.0315
	2	0.0276	0.0276	0.0236	0.0236	0.0236	0.0236	0.0157
	3	0.0236	0.0236	0.0236	0.0197	0.0197	0.0118	0.0062
	4	0.0200	0.0157	0.0157	0.0118	0.0052	0.0043	
	5	0.0200	0.0157	0.0083	0.0048			
	6	0.0158	0.0060					
	7	0.0058						

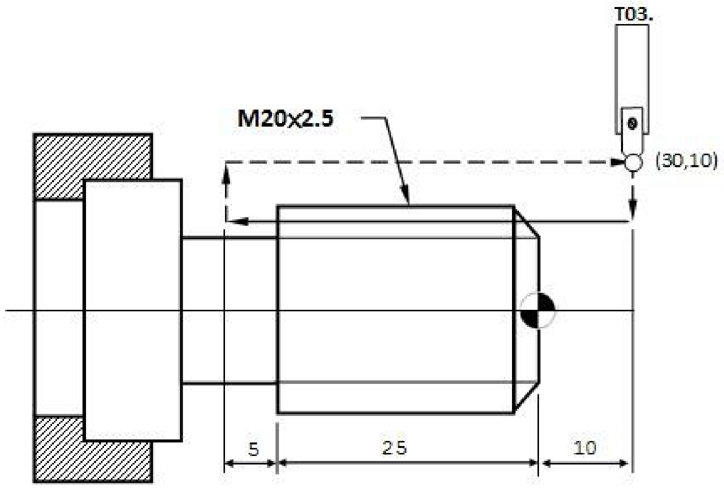


Metric system depth of tooth = 0.06495P P = Pitch

Pitch of thread(mm)		4.0	3.5	3.0	2.5	2.0	1.5	1.0
Height of thread 0.6495P(mm)		2.598	2.273	1.949	1.624	1.299	0.974	0.650
Numbers of cutting and the value of cutting(diameter)	1	1.5	1.5	1.2	1.0	0.9	0.8	0.7
	2	0.8	0.7	0.7	0.7	0.6	0.6	0.4
	3	0.6	0.6	0.6	0.6	0.6	0.4	0.2
	4	0.6	0.6	0.4	0.4	0.4	0.16	
	5	0.4	0.4	0.4	0.4	0.1		
	6	0.4	0.4	0.4	0.15			
	7	0.4	0.2	0.2				
	8	0.3	0.15					
	9	0.2						

2.25.4 **Example 1**

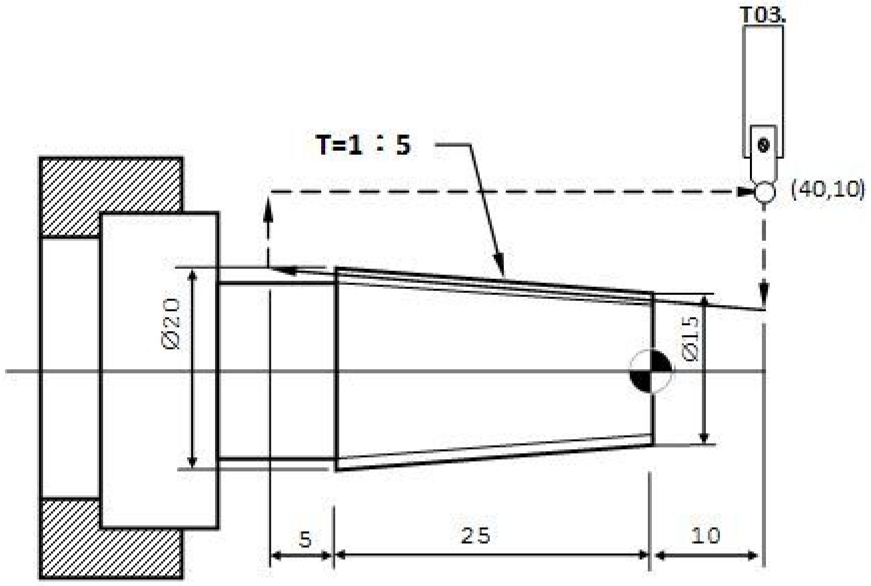
Straight Thread Cutting



```
T03;                    //use tool NO.3
G97 S1000 M03;          //spindle rotate CW 1000 rpm, constant RPM
M08;                    //cutting liquid ON
G00 X30.0 Z10.0;        //positioning to starting point of cutting
X19.0;
G33 Z-30.0 F2.5;        // First cutting 1.0 mm
G00 X30.0;
Z10.0;
X18.3;
G33 Z-30.0 F2.5;        // Second cutting 0.7 mm
G00 X30.0;
Z10.0;
X17.7;
G33 Z-30.0 F2.5;       // Third Sixth cutting 0.6 mm
G00 X30.0;
Z10.0;
X17.3;
G33 Z-30.0 F2.5;        // Fourth Sixth cutting 0.4 mm
G00 X30.0;
Z10.0;
X16.9
G33 Z-30.0 F2.5;         // Fifth Sixth cutting 0.4 mm
G00 X30.0;
Z10.0;
X16.75;
G33 Z-30.0 F2.5;        // Sixth cutting 0.15 mm
G00 X30.0;
Z10.0;
G28 X50.0 Z30.0;         //positioning to specified mid-point, then return to machine zero point
M09;                     //cutting liquid OFF
M05;                     //spindle stops
M30;                    //program ends
```

2.25.5 **Example 2**

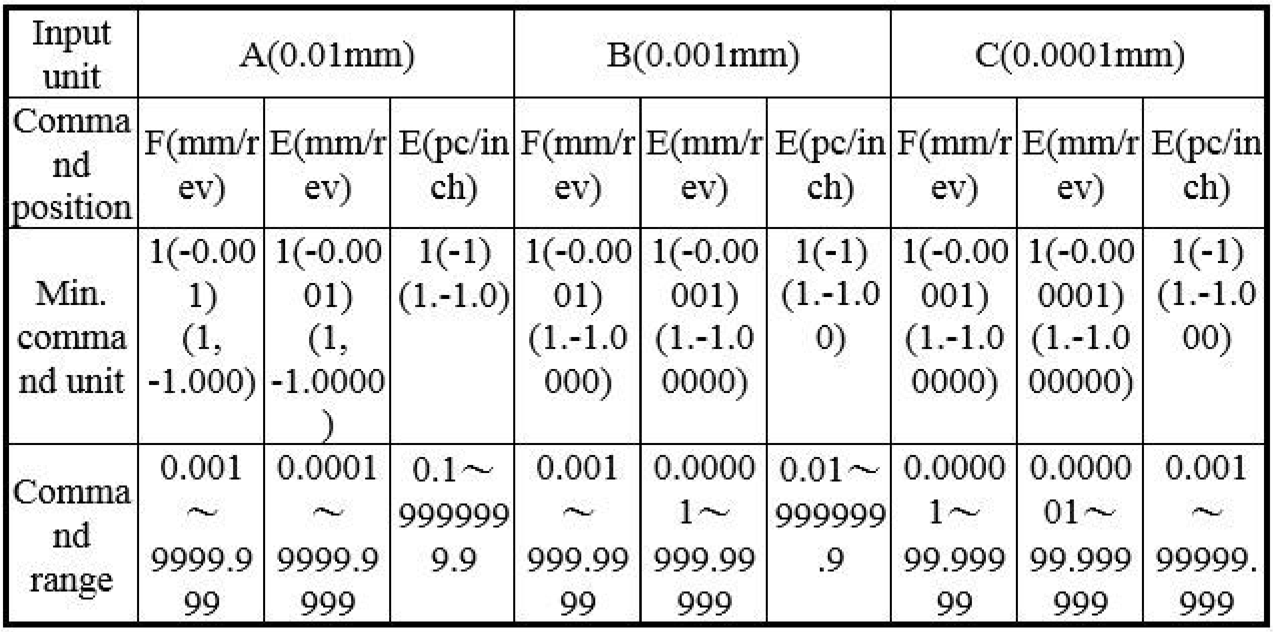
Tapered Thread Cutting, Pitch = 2.5



```
T03;                        //use tool NO.3
G97 S1000 M03;              //spindle rotate CW 1000 rpm, constant RPM
M08;                        //cutting liquid ON
G00 X40.0 Z10.0;            //positioning to starting point of cutting
X12.0;
G33 X20.0 Z-30.0 F2.5;   // First cutting 1.0 mm
G00 X40.0;
Z10.0;
X11.3;
G33 X19.3 Z-30.0 F2.5;   // Second cutting 0.7 mm
G00 X40.0;
Z10.0;
X10.7;
G33 X18.7 Z-30.0 F2.5;    // Third cutting 0.6 mm
G00 X40.0;
Z10.0;
X10.3;
G33 X18.3 Z-30.0 F2.5;   // Fourth cutting 0.4 mm
G00 X40.0;
Z10.0;
X9.9;
G33 X17.9 Z-30.0 F2.5;    // Fifth cutting 0.4 mm
G00 X40.0;
Z10.0;
X9.75;
G33 X17.75 Z-30.0 F2.5;   // Sixth cutting 0.15 mm
G00 X40.0;
Z10.0;
G28 X50.0 Z30.0;  //positioning to specified mid-point, and return to machine zero point
M09;                //cutting liquid OFF
M05;                //spindle stops
M30;                //program ends
```

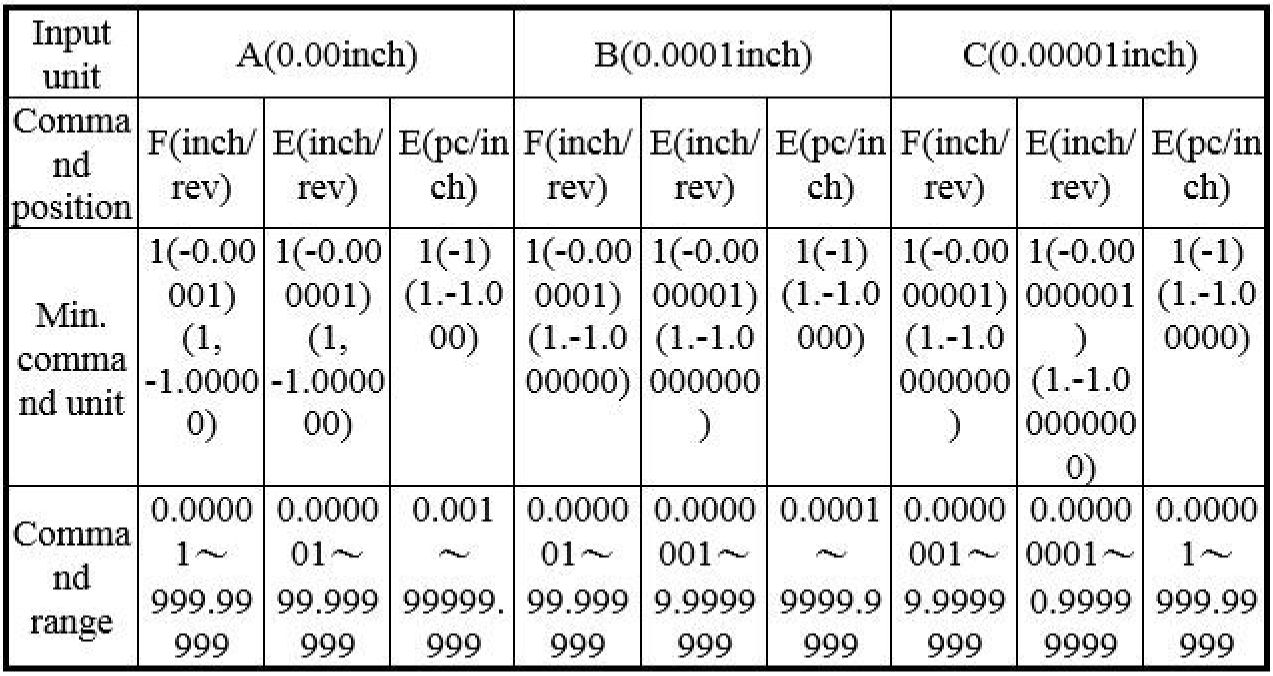
2.25.6 **Appendix**

- Input unit and modal of E ,F value as below table: table 1. Metric system ,table 2. English system



Input unit	A(0.01mm)			B(0.001mm)			C(0.0001mm)		
Command position	F(mm/rev)	E(mm/rev)	E(pc/inch)	F(mm/rev)	E(mm/rev)	E(pc/inch)	F(mm/rev)	E(mm/rev)	E(pc/inch)
Min. command unit	1(-0.001) (1, -1.000)	1(-0.0001) (1, -1.0000)	1(-1) (1.-1.0)	1(-0.0001) (1.-1.0000)	1(-0.00001) (1.-1.00000)	1(-1) (1.-1.00)	1(-0.00001) (1.-1.00000)	1(-0.000001) (1.-1.000000)	1(-1) (1.-1.000)
Command range	0.001～9999.999	0.0001～9999.9999	0.1～9999999.9	0.001～999.9999	0.00001～999.99999	0.01～999999.9	0.00001～99.99999	0.000001～99.999999	0.001～99999.999

Table 1. input by Metric system



Input unit	A(0.00inch)			B(0.0001inch)			C(0.00001inch)		
Command position	F(inch/rev)	E(inch/rev)	E(pc/inch)	F(inch/rev)	E(inch/rev)	E(pc/inch)	F(inch/rev)	E(inch/rev)	E(pc/inch)
Min. command unit	1(-0.00001) (1, -1.00000)	1(-0.000001) (1, -1.000000)	1(-1) (1.-1.000)	1(-0.000001) (1.-1.000000)	1(-0.0000001) (1.-1.0000000)	1(-1) (1.-1.0000)	1(-0.0000001) (1.-1.0000000)	1(-0.00000001) (1.-1.00000000)	1(-1) (1.-1.00000)
Command range	0.00001～999.99999	0.000001～99.999999	0.001～99999.999	0.000001～99.999999	0.0000001～9.9999999	0.0001～9999.9999	0.0000001～9.9999999	0.00000001～0.99999999	0.00001～999.99999

Table 2. input by English system

2.26 G34- Variable Pitch Thread Cutting (C-Type)

2.26.1 **Command Form**

(1) Straight thread cutting: G34 Z(W)__Q__F__K__
(2) Tapered thread cutting: G34 X(U)__Z(W)__Q__F__K__
(3) Endface thread cutting: G34 X(U)__Q__F__K__

X , Z: Specified position (absolute mode)
U , W: Specified position (incremental mode)
F: Pitch in longitudinal direction <- ordinary thread, metric thread

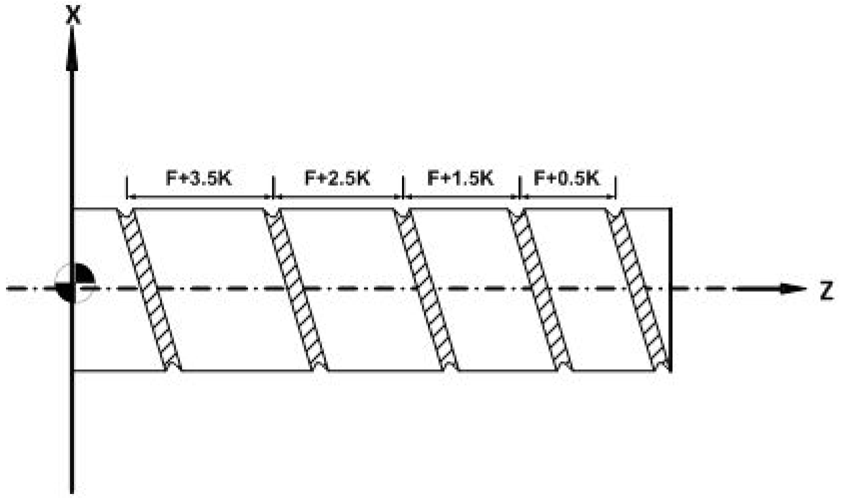
Q: Angle offset at thread start. To let the tool start cutting at the same angle of a rotating workpiece, usually used in multiple-thread cutting. For single-thread cutting, ignore the Q argument and apply **default value Q= 0°**(range: 0.001~360.000∘)

K: Pitch increase or decrease per revolution, can be positive or negative.

2.26.2 **Description**

G33 command executes endface thread cutting, taper threading, and straight thread cutting by synchronously rotating workpiece and feeding tool to create variable pitch thread. (Available in version 10.112.0 and later, unavailable in 9.0 version)

2.26.3 **Illustration**



2.26.4 **Notice**

1. If pitch decrease per revolution causes pitch to become negative, Alarm [Invalid threading pitch] will be triggered. If the calculated thread cutting feedrate is greater than maximum allowable feedrate, the pitch will decrease and trigger alarm [threading block feedrate exceed].
2. Total move distance in one block: [F+(F+Rev*K)]* Rev/2.
3. Other notices are the same with G33.

2.26.5 **Example 1**

```
T03                          // use tool no. 3
G97 S1000 M03         //Spindle rotate CW 1000rpm, and constant RPM.
M08                     //cutting liquid On
G00 X0.0 Z0.0           // G00 move to cutting original point
G34 Z-50.0 F1.0 K0.2    // Variable thread cutting with pitch increase 0.2mm per rev
M09                     //cutting liquid Off
M05                     //Spindle stop
M30                     //finish
```

2.26.6 **Example 2**

```
T03                     //use tool no. 3
G97 S1000 M03           //Spindle rotate CW 1000rpm, and constant RPM.
M08                     //cutting liquid On
G00 X0.0 Z0.0           //Positioning to cutting original point
G33Z16F4                //Threading with fix pitch 4mm
G34W19F4K5.5            //Pitch increase 5.5mm per rev. Pitch changes from 4mm to 15mm.
G33W4F15                //Threading with fix pitch 15mm
G34W18F15K-4            //Pitch decrease 4mm per rev. Pitch changes from 15mm to 9mm.
G33W12F9                //Threading with fix pitch 9mm
M09                     //cutting liquid Off
M05                     //Spindle stop
M30                     //Program finish
```

2.27 G40/G41/G42- Tool Nose Radius Compensation (C-Type)

2.27.1 **Command Form**

G41 X(U)___ Z(W)___
G42 X(U)___ Z(W)___
G40 compensation cancel
X , Z: Specified position (absolute mode)
U , W: Specified position (incremental mode)

2.27.2 **Description**

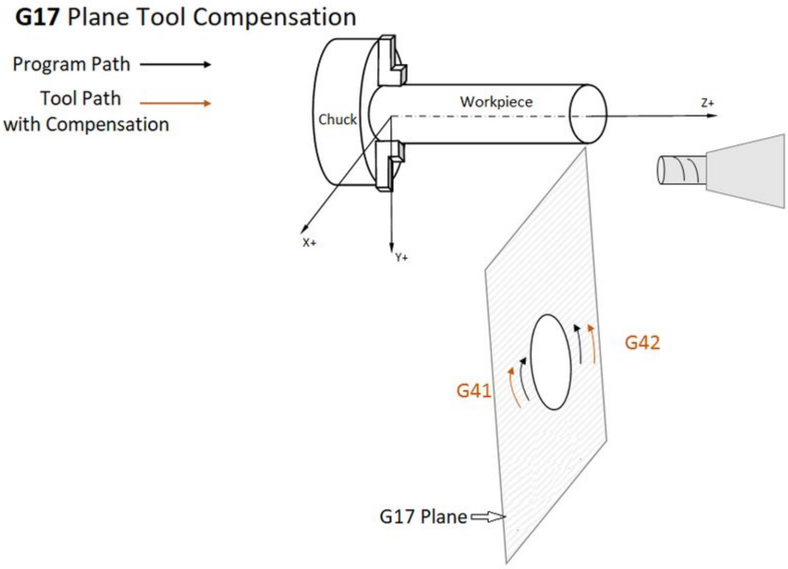
The tip of a Lathe tool is made into a small sphere to increase tip rigidity, expand tool life, decrease stress concentration, improve heat dissipation, and smoothen cutting surface. The sphere is called Tool Nose and its radius is Tool Nose Radius. But when cutting corners and arcs, errors will occur because of spherical tool nose cannot perform the exactly shape. G41 & G42 can calculate error of tool nose radius accurately and compensate it.

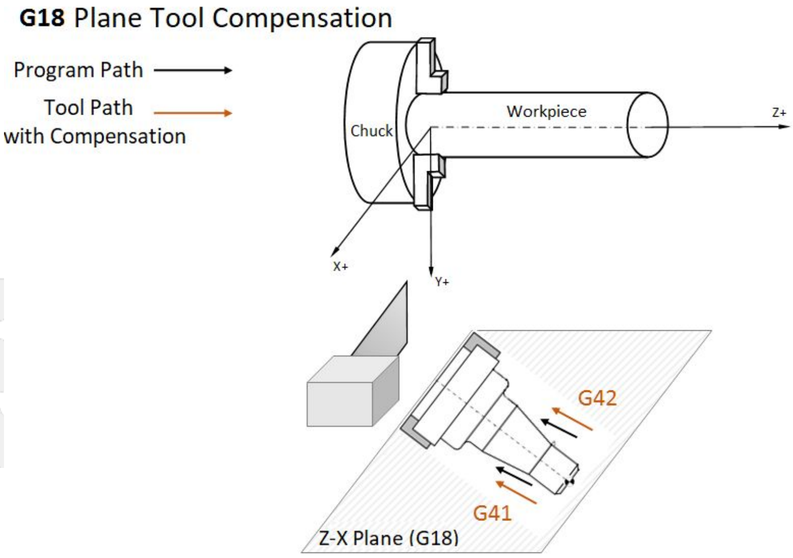
G code	Function	Position of Tool
G40	Disable tool nose compensation	Tool moves along the path of program

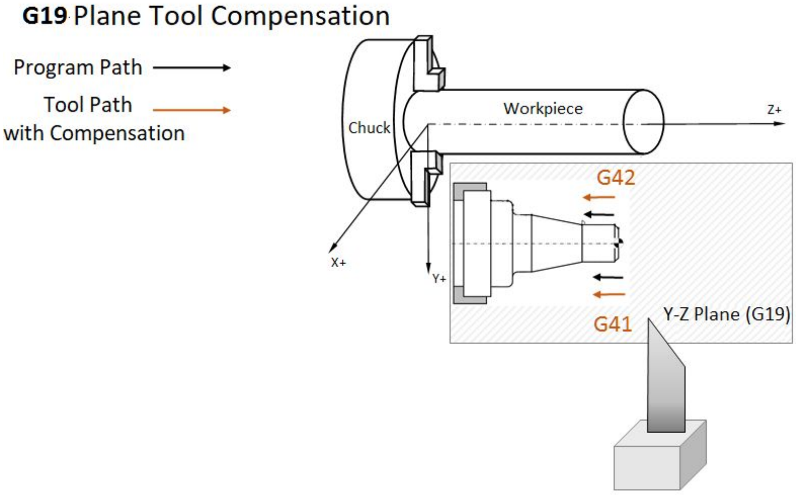
G41	Tool nose compensation left	Tool offsets left a specified value of program path
G42	Tool nose compensation right	Tool offsets right a specified value of program path

Precautions

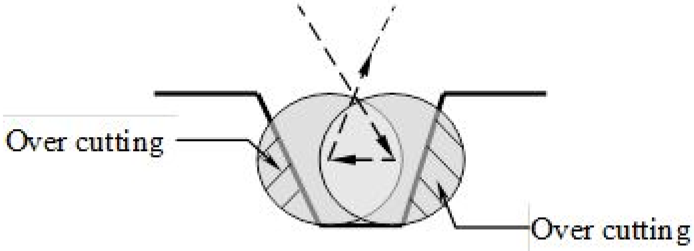
1. When enabling tool radius compensation, left/right compensation direction (G41/G42) is determined by the selected work plane (G17/G18/G19).



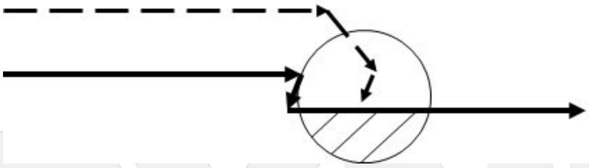




2. The effective block of setup (G41/G42) and cancel (G40) of the cutter radius compensation function is defined by Pr3815 Tool radius compensation mode.
3. The effective block should have displacement on the working plane(G17, G18, G19), and could not be G02, G03 commands.
4. During the grooving process, if the groove width is smaller than twice the cutter radius, alarm [COR-074_excessive cutter radius, path overcut will be issued due to the overcut.] (Only check when Pr3819 is set 1)



5. When the programmed pathe is shorter than the cutter radius, under specific path angles, there might be a over cutting situaltion. If so, alarm [COR-074_excessive cutter radius, path overcut will be issued due to the overcut.] (Only check when Pr3819 is set 1)



6. The G10 L12 and G10 L13 commands cannot be used in the same block as the D code commands.
7. Tool radius compensation relies on the motion block after the G40 block to end the compensation path. If the next line of G40 is not a motion block(i.e. other non-motion blocks in front of the motion block, such as tool change T code, plane switching G17/G18/G19...etc.), the end of compensation path will not be as expected.
8. This feature doesn't support Skip Function(G31) and Multi-Axis Multi-Signal Skip Function(G31.10/G31.11).

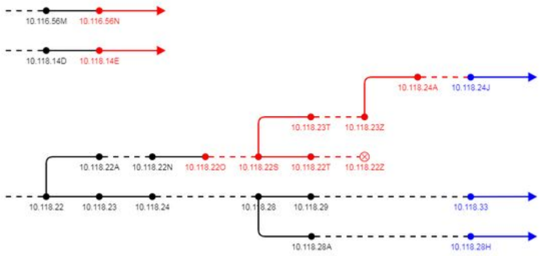
Restrictions

- After tool length compensation is changed[1], it is needed to position the axes on the working plane[3] if users want to execute the tool radius compensation[2]. Otherwise, there will be the following results depending on the version.

1. There is a risk of unexpected axial movement in the following version.
   1. ~ 10.116.56M, 10.118.14D, 10.118.22N
   2. 10.118.28A ~ 10.118.28G
2. The alarm **"COR-355 Need geometric axes positioning after tool length compensation changed"** will happen in the following version.
   1. 10.118.56N ~
   2. 10.118.14E ~
   3. 10.118.22O ~ 10.118.22Z
   4. 10.118.23T ~ 10.118.23Z
   5. 10.118.24A ~ 10.118.24I
3. In 10.118.24J, 10.118.28H ~, there is no unexpected axial movement and the alarm COR-355 does not happen. But in special cases, there are compatibility changes in path compensation results.

- The above descriptions can also be compared to the graphical description in the following diagram.

[1] As long as the commands, H, G43, G44, G43.4, G43.5, G49, T, G10 L1050, G10 L1051 are executed, the tool length compensation is considered to be changed whatever there is effective compensation or not.
[2] Including G41, G42.
[3] Including G90 G00, G90 G01, G92. But G91 (incremental) G00, G01 not included.

Diagram of related versions	**Description**
 10.116.56M 10.116.56N 10.118.14D 10.118.14E 10.118.24A 10.118.24J 10.118.23T 10.118.23Z 10.118.22A 10.118.22N 10.118.22O 10.118.22S 10.118.22T 10.118.22Z 10.118.22 10.118.23 10.118.24 10.118.28 10.118.29 10.118.33 10.118.28A 10.118.28H	**Black lines:** If the specifications are violated, there is a risk of unexpected axial movement. **Red lines:** If the specifications are violated, the alarm COR-355 will happen. **Blue lines:** The problem of unexpected axial movement has been resolved. The alarm COR-355 will not happen. But in special cases, there are compatibility changes in path compensation results.

Related Setting

Parameters: (please refers to parameter manual)

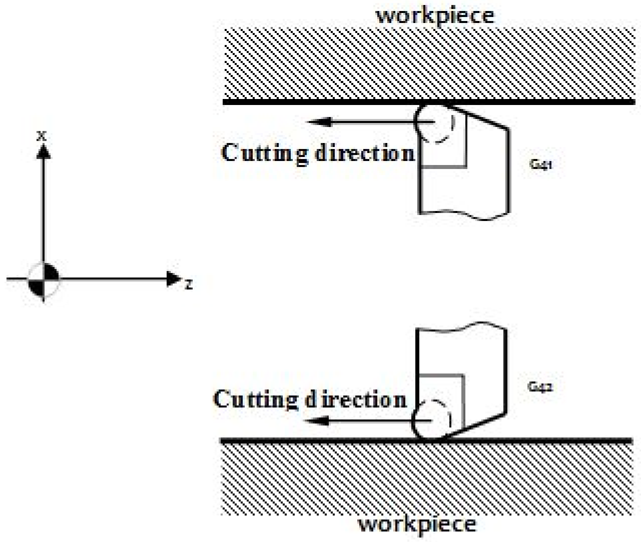
- Pr3814
- Pr3815
- Pr3819

Controller Interface:

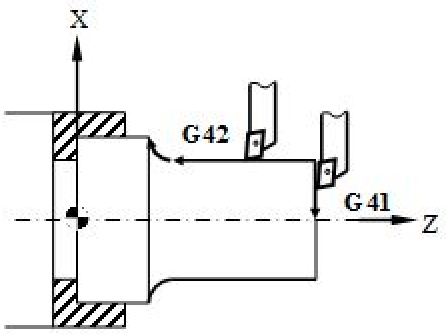
- Tool Nose Setting: set tool nose radius, wearing, direction (ref. : 10/20/200/SUPER Series_Lathe Operation Manual.)

Illustration

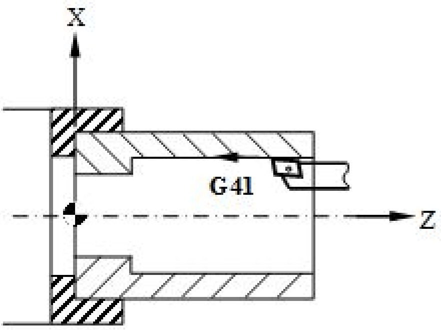
- Relationship between tool feed direction and workpiece, setting of compensation:



- Compensation setting of actually cutting

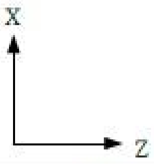


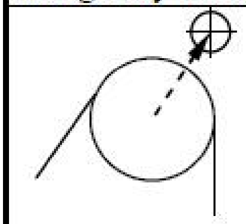
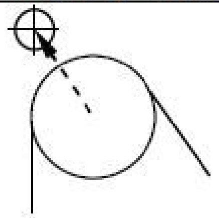
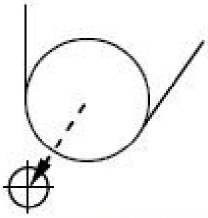
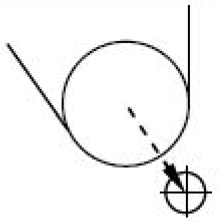
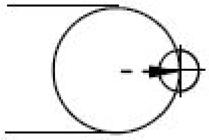
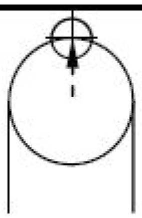
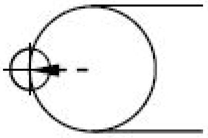
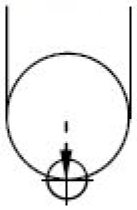
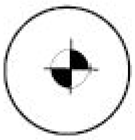
a. cutting outer radius and end surface



b. cutting internal radius

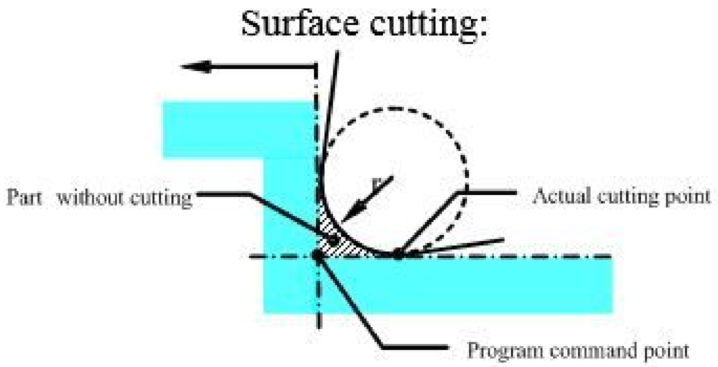
- Imaginary tool nose number setting:



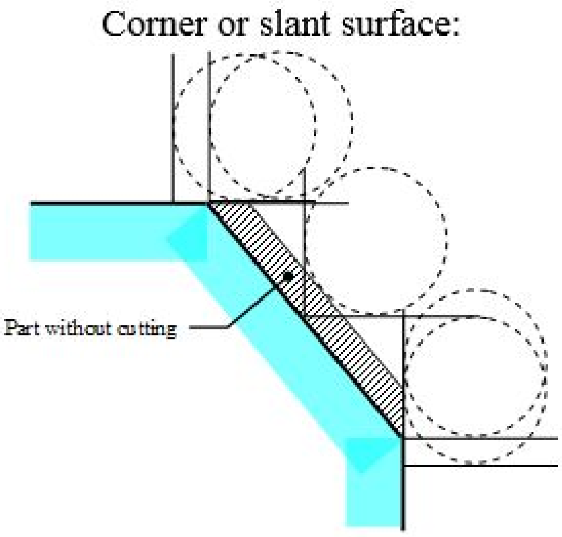
Imaginary tool nose NO.1	Imaginary tool nose NO.2	Imaginary tool nose NO.3
		
Imaginary tool nose NO.4	Imaginary tool nose NO.5	Imaginary tool nose NO.6
		
Imaginary tool nose NO.7	Imaginary tool nose NO.8	Imaginary tool nose NO.9
		

- Without Tool Nose Compensation:

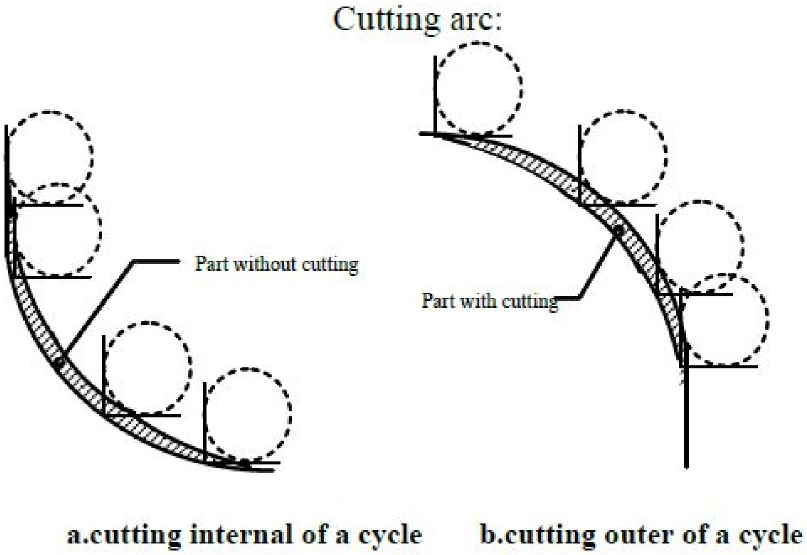
1. End surface cutting:



2. Chamber corner or slant surface:



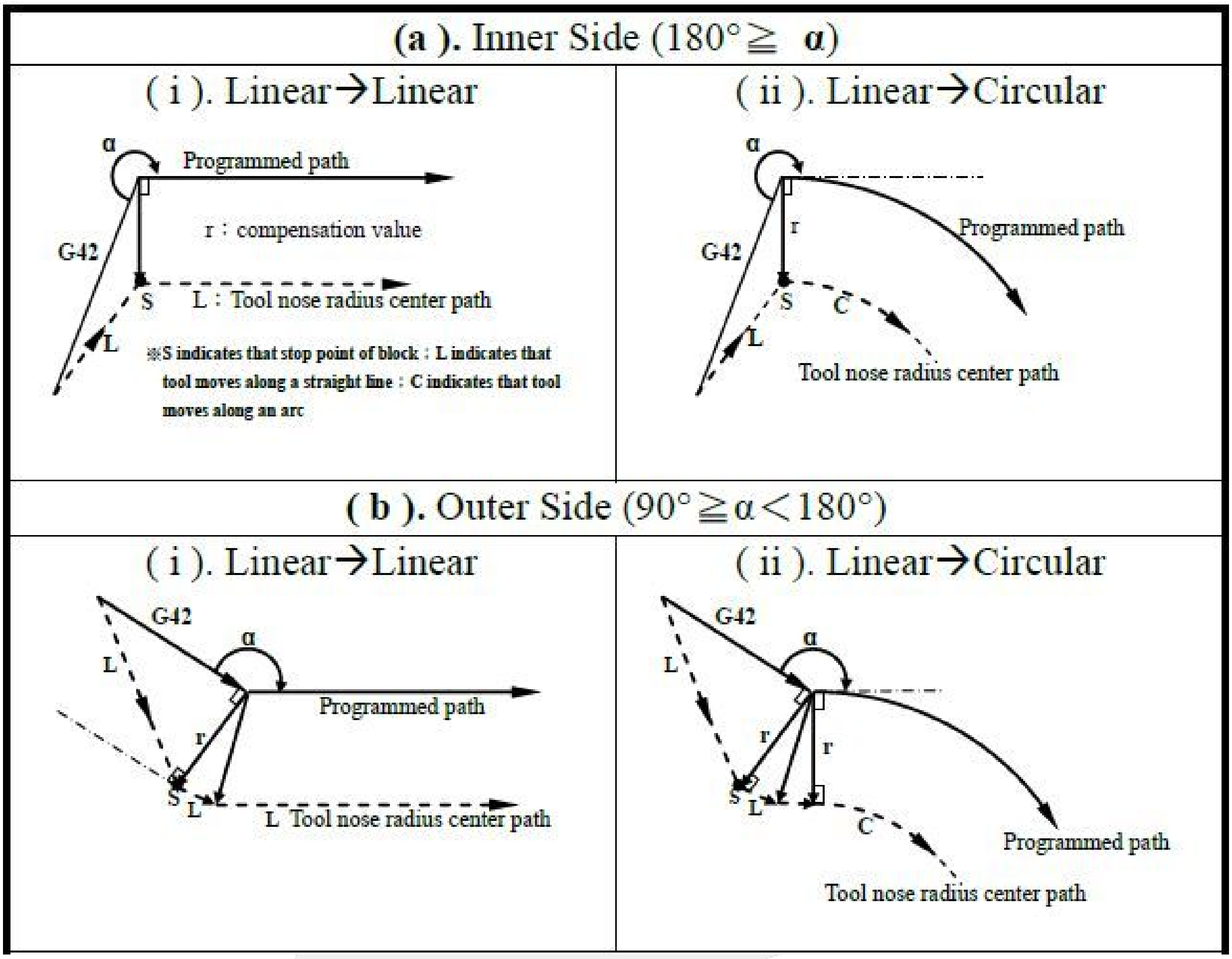
3. Arc Cutting:

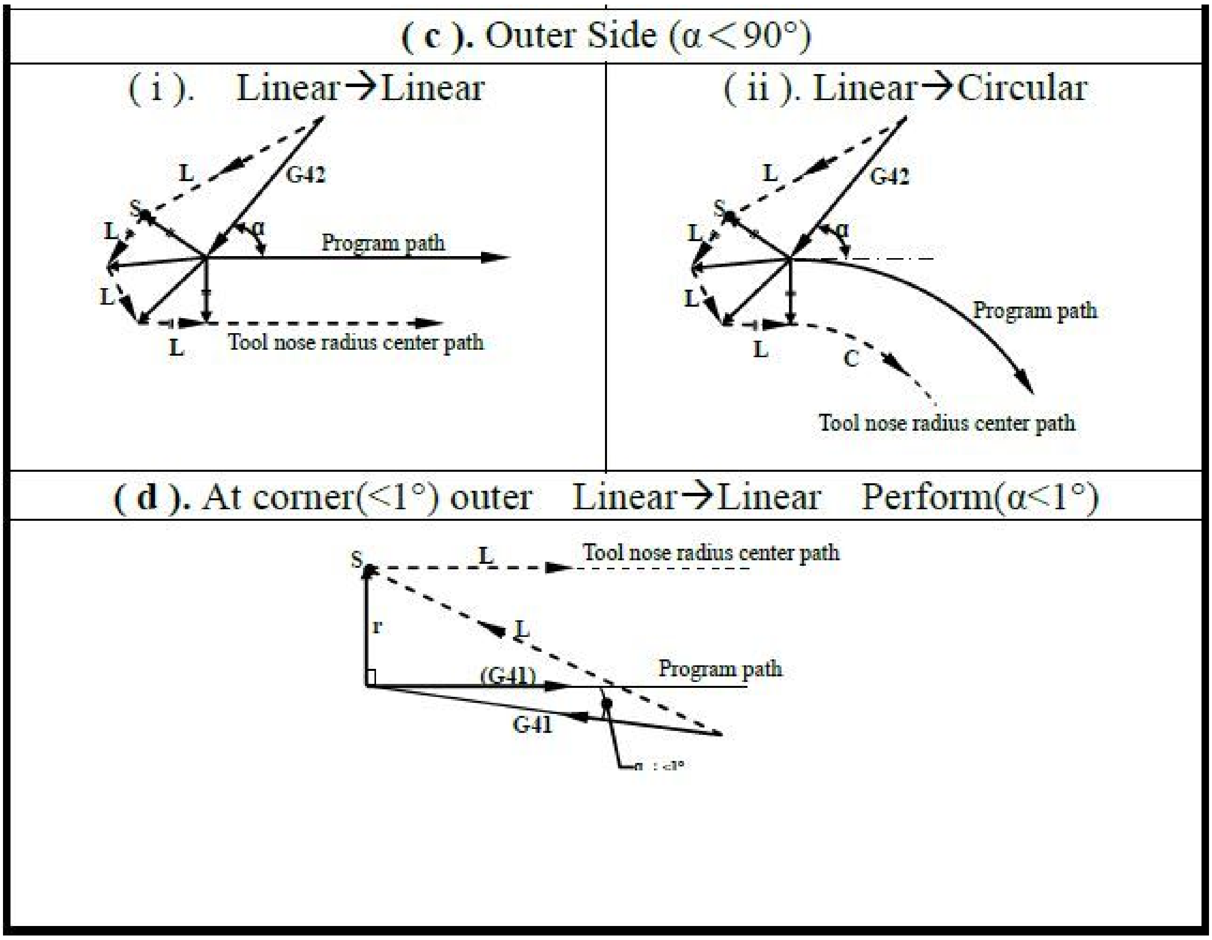


Tool Radius (R) Compensation Detail

- Compensation Starts
  When a block which satisfies all the following conditions is executed, the system enters the compensation mode. Control during this operation is called compensation start.

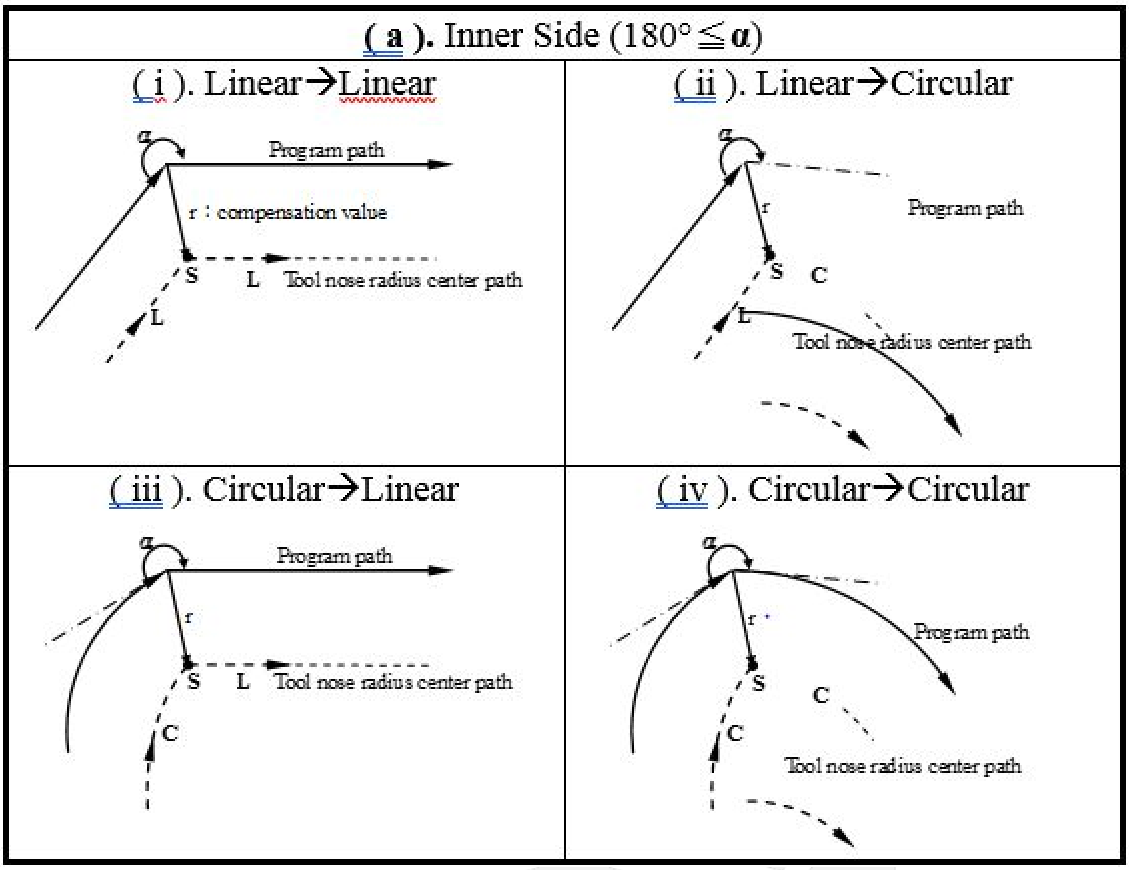
1. A block contains G41 or G42, or system has been set under G41 or G42 mode.
2. The compensation number of tool nose compensation is not "00".
3. X or Z movement specified in the block with non-zero distance.


(a). Inner Side (180°≧ α)
(i). Linear→Linear
α
Programmed path
r：compensation value
G42
S
L：Tool nose radius center path
L
※S indicates that stop point of block；L indicates that tool moves along a straight line；C indicates that tool moves along an arc
(ii). Linear→Circular
α
G42
r
Programmed path
S
C
L
Tool nose radius center path
(b). Outer Side (90°≧α＜180°)
(i). Linear→Linear
G42
α
L
Programmed path
r
S
L
L Tool nose radius center path
(ii). Linear→Circular
G42
α
L
r
r
S
L
C
Programmed path
Tool nose radius center path


(c). Outer Side (α＜90°)
(i). Linear→Linear
L
G42
S
α
L
Program path
L
L
Tool nose radius center path
(ii). Linear→Circular
L
G42
S
α
L
L
Program path
L
C
Tool nose radius center path
(d). At corner(<1°) outer Linear→Linear Perform(α<1°)
S
L
Tool nose radius center path
r
L
(G41)
Program path
G41
α : <1°

- **Compensation mode**

In compensation mode, compensation maintains enabled during rapid positioning, same as straight and circular interpolation. Cannot specify continuous compensation number without giving move block (i.e. dwell or M code), otherwise undercut will occur.



(a). Inner Side (180°≦α)

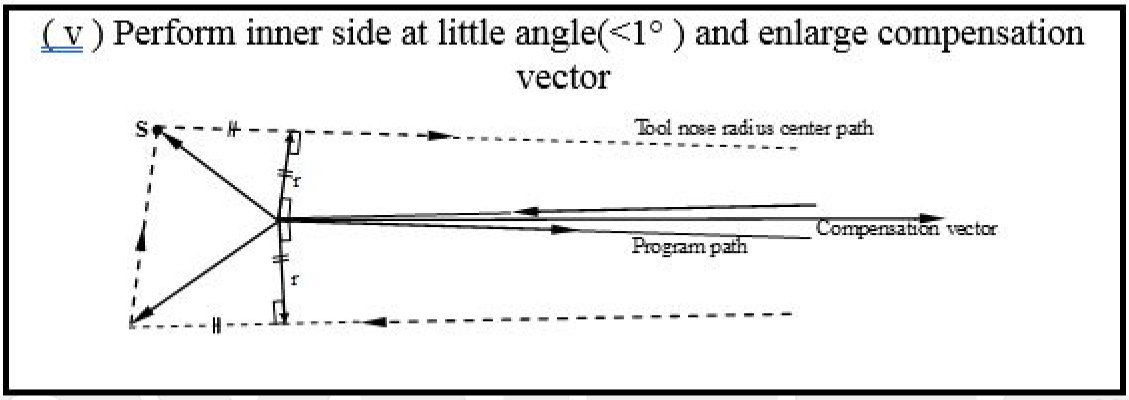
(i). Linear→Linear

(ii). Linear→Circular

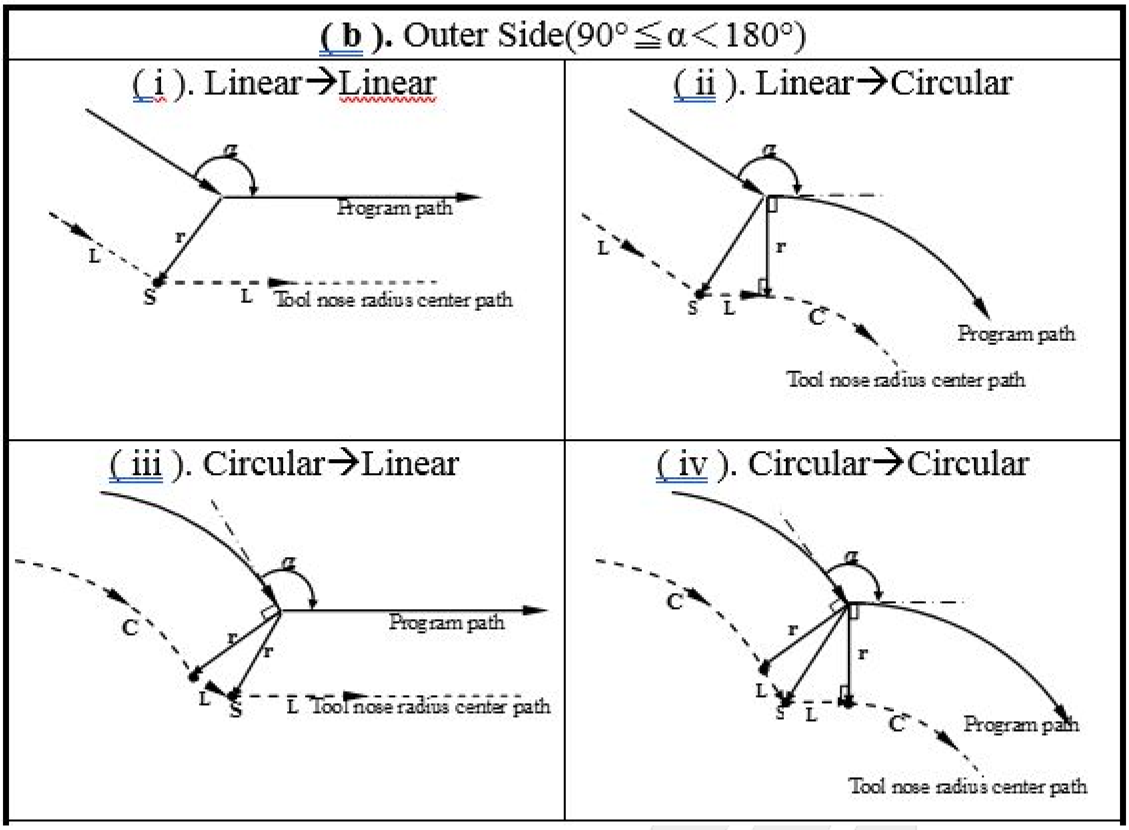
(iii). Circular→Linear

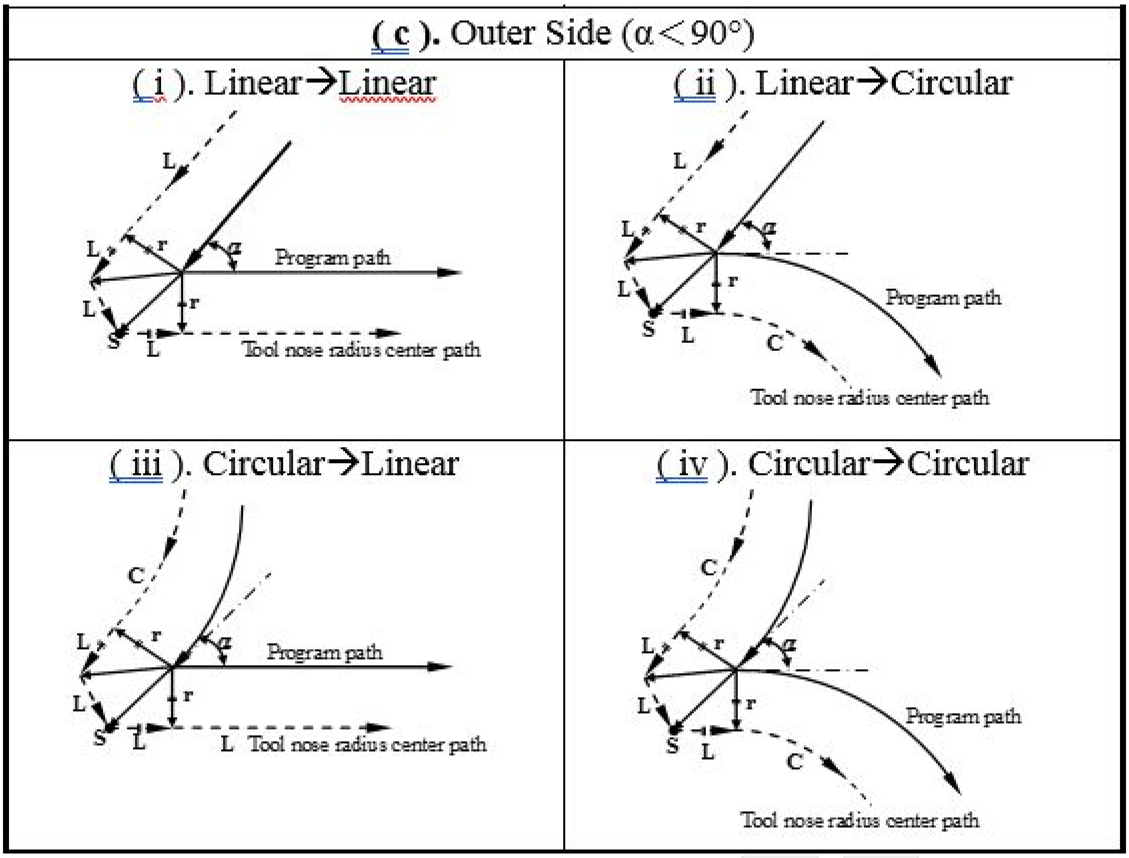
(iv). Circular→Circular

α
Program path
r：compensation value
S
L
Tool nose radius center path
L
α
r
Program path
S
C
L
Tool nose radius center path
α
Program path
r
S
L
Tool nose radius center path
C
α
r
Program path
S
C
C
Tool nose radius center path



(v) Perform inner side at little angle(<1°) and enlarge compensation vector


(b). Outer Side(90°≦α＜180°)
(i). Linear→Linear
α
Program path
r
L
S
L
Tool nose radius center path
(ii). Linear→Circular
α
L
r
S
L
C
Program path
Tool nose radius center path
(iii). Circular→Linear
α
Program path
C
r
r
L
S
L
Tool nose radius center path
(iv). Circular→Circular
α
C
r
r
L
S
L
C
Program path
Tool nose radius center path

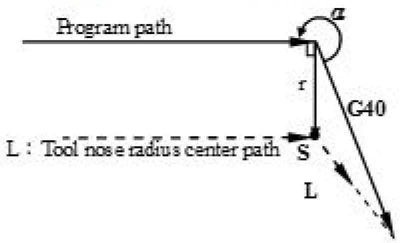
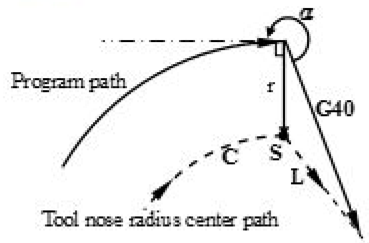
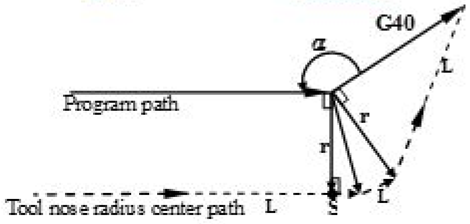
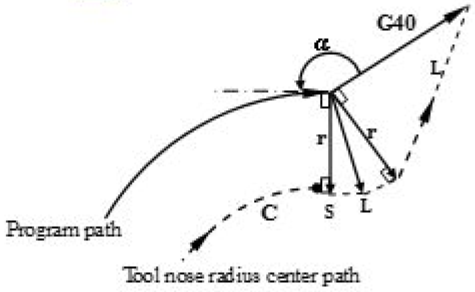


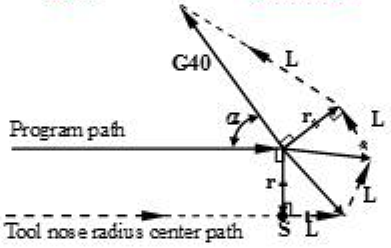
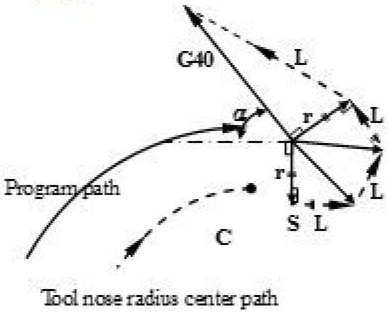
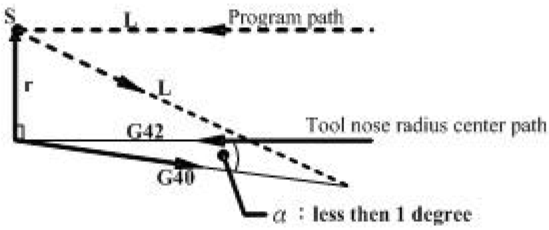
- **Compensation Cancel**

In compensation mode, when block satisfies following conditions, system will enter cancel mode:

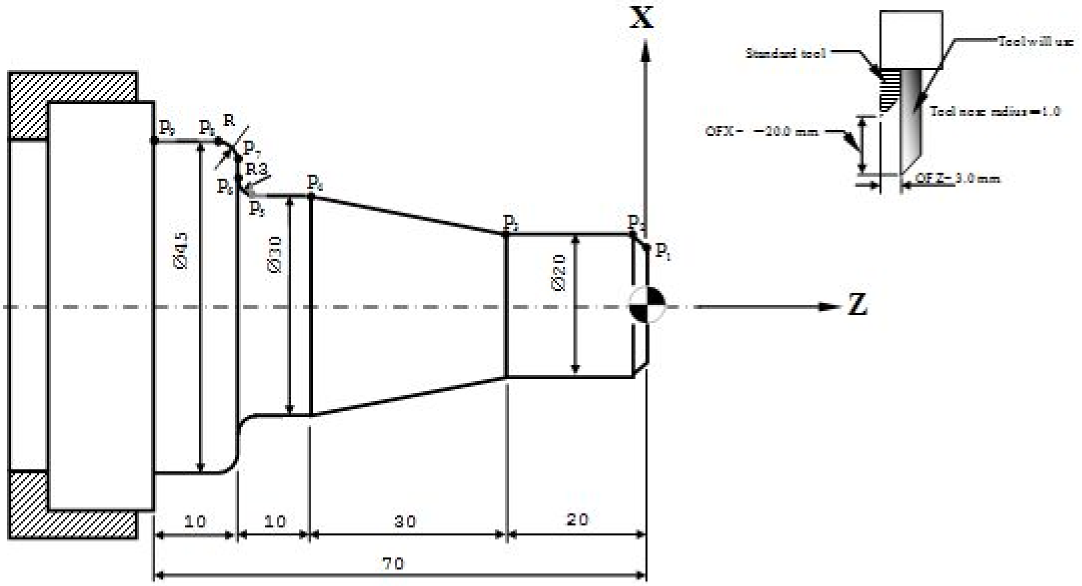
1. Specify G40

2. The number of tool nose radius compensation is specified to “0”

<table>
<tr><th colspan="2">(a). Inner Side (180°≦ α)</th></tr>
<tr><td>(i). Linear→Linear
 </td><td>(ii). Linear→Circular
 </td></tr>
<tr><th colspan="2">(b). Outer Side (90°≦α＜180°)</th></tr>
<tr><td>(i). Linear→Linear
 </td><td>(ii). Linear→Circular
 </td></tr>
</table>

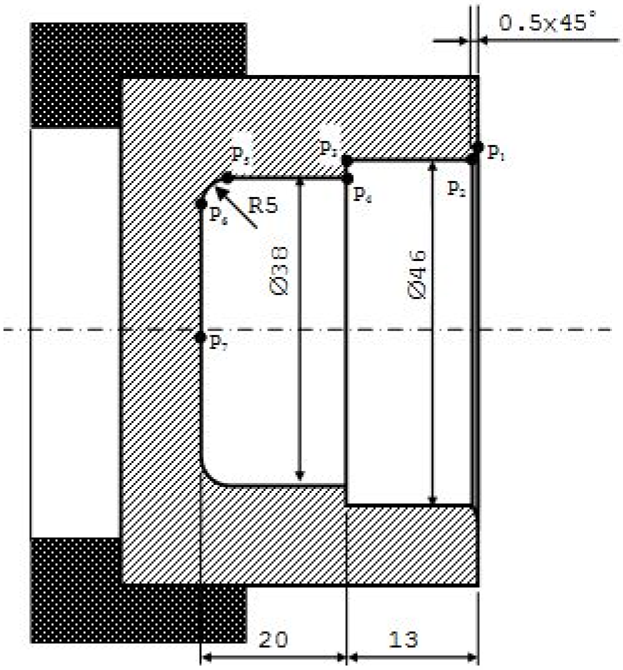
<table>
<tr><th colspan="2">(c). Outer Side (α＜90°)</th></tr>
<tr><td>(i). Linear→Linear
 </td><td>(ii). Linear→Circular
 </td></tr>
<tr><th colspan="2">(d). Tool movement around the outside linear→linear at an acute angle less than 1 degree(α&lt;1°)</th></tr>
<tr><td colspan="2"> </td></tr>
</table>

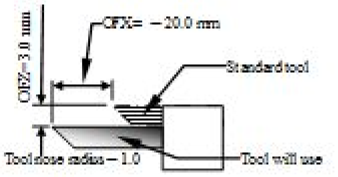
2.27.3 **Example 1**



```
T02                          //use tool NO.2, tool NO.2's nose radius = 1.0mm was set via control interface
G50 S10000                     //max. rotate speed, 10000rpm
G96 S130 M03                   //constant surface speed, spindle rotate, 130 m/min CW
M08                          //cutting liquid ON
G42 X18.0 Z0.0                 //tool compensation right start-up, move to P1
G01 X20.0 Z-2.0 F0.6            //linear interpolation, feedrate 0.6mm/rev, P1--->P2
Z-20.0                       // P2 --->P3
X30.0 Z-50.0                   // P3 --->P4
Z-57.0                       //P4 --->P5
G02 X36.0 Z-60.0 R3.0            // P5 --->P6
G01 X39.0                     // P6 --->P7
G03 X45.0 Z-63.0 R3.0            // P7--->P8
G01 Z-70.0                     // P8 --->P9
X60.0                        //return the tool
G28 X70.0 Z-60.0                //positioning to specified mid-point, then return to machine zero point
M09                          //cutting liquid OFF
M05                          //spindle stops
M30                          //program ends
```

2.27.4 **Example 2**



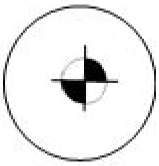


```
T02                     //use tool NO.2, tool NO.2's nose radius = 1.0mm was set via control interface
G50 S1000                //max. rotate speed, 10000rpm
G96 S130 M03              //constant surface speed, spindle rotate 130m/min CW
M08                     //cutting liquid ON
G41 X47.0 Z0.0            //start tool compensation left, move to P1
G01 X46.0 Z-0.5 F0.6       //linear interpolation, feedrate 0.6mm/rev, P1 ---> P2
Z-13.0                  //P2 ---> P3
X38.0                   //P3 ---> P4
Z-28.0                  //P4 ---> P5
G03 X28.0 Z-33.0 R5.0      //circular interpolation CCW, radius 5mm, P5 ---> P6
G01 X-1.0                //linear interpolation
M09                     //cutting liquid OFF
G28 Z20.0                //positioning to specified mid-point, then return to machine zero point
M05                     //spindle stops
M30                     //program ends
```

2.27.5 **Driven Tool Radius Compensation in Lathe**

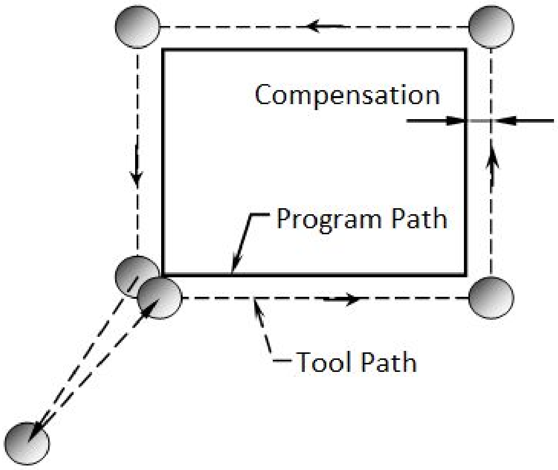
Tool Nose Number Setting

- The imaginary tool nose number of the driven tool is 0 or 9

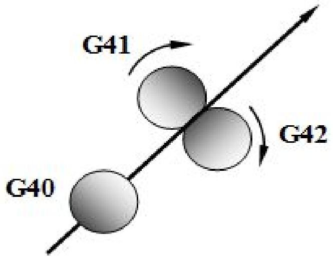


Tool Nose Radius Compensation

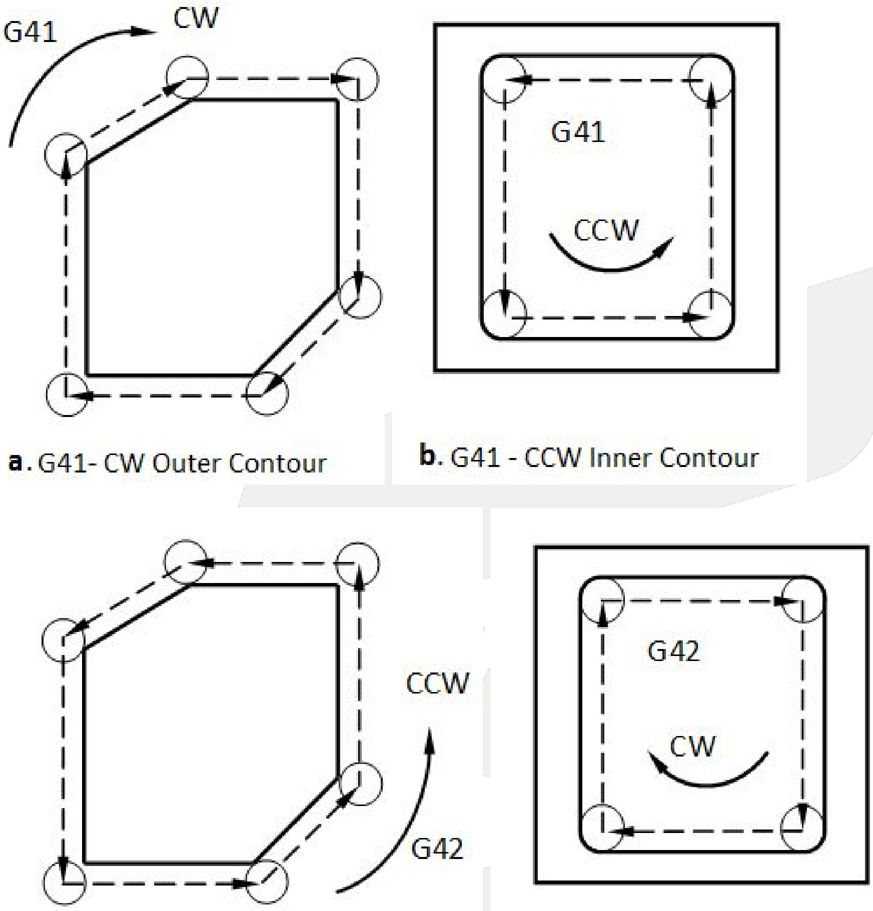
- The selection of the working plane (G17/G18/G19) will affect the compensation of the tool radius. Please select the correct working plane before turning on the compensation.



Define Direction of Tool Radius Compensation



G Code	Positive	Negative
G41	Comp. left	Comp. right
G42	Comp. right	Comp. left



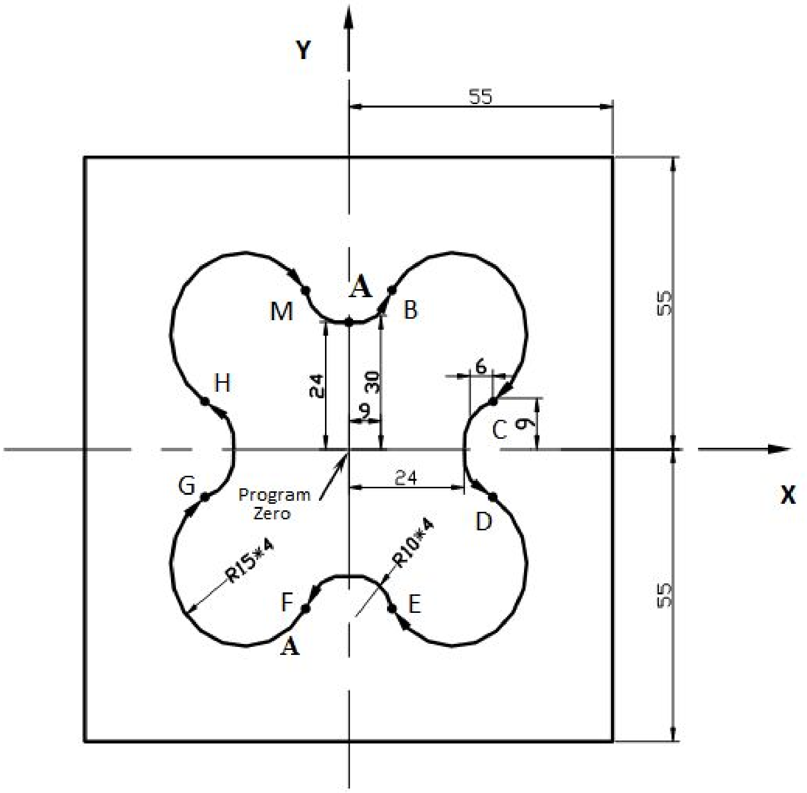
a. G41- CW Outer Contour

b. G41 - CCW Inner Contour

c. G42-CCW Outer Contour

d. G42- CW Inner Contour

2.27.6 **Example 1**



//X axis is diameter positioning and Y axis is radius positioning

```
T0101 S1000 M03; //tool NO.1. start tool NO.1 compensation (with diameter 10mm milling cutter). spindle CW rotate speed , 10000rpm
G00 X0.0 Y0.0 Z10.0; //positioning to program above original point
M08; //cutting liquid ON
G90 G17; // absolute command. switch work plane to XY plane. both XY are diameter axis
G01 Z-10.0 F600; //linear interpolation to the bottom of "flower" pattern, feedrate 600mm/min
G41 Y24.0; //tool left compensation. program original point→A
G03 X18.0 Y30.0 R10.0; //A→B arc cutting CCW
G02 X60.0 Y9.0 R15.0; //B→C arc cutting CW
G03 X60.0 Y-9.0 R10.0; //C→D arc cutting CCW
G02 X18.0 Y-30.0 R15.0; //D->E arc cutting CW
G03 X-18.0 Y-30.0 R10.0; //E→F arc cutting CCW
G02 X-60.0 Y-9.0 R15.0; //F->G arc cutting CW
G03 X-60.0 Y9.0 R10.0; //G→H arc cutting CCW
G02 X-18.0 Y30.0 R15.0; //H->M arc cutting CW
G03 X0.0 Y24.0 R10.0; //M→A arc cutting CCW
G00 Z10.0; //pull up Z axis. Back to machine initial position
G40 X0.0 Y0.0; //cancel tool compensation and back to machine initial position
M09; //cutting liquid OFF
M05; //spindle stops
M30; //program ends
```

2.28 G51/G50- Scaling (C-Type)

2.28.1 **Command Form**

$$X\_\ \ Y\_\ \ Z\_\ \begin{cases} I\_J\_K\_ \\ P\_ \end{cases}$$

X, Y, Z: Coordinate of scaling center and the assigned scaling axis

I, J, K: Scaling factor(applies when the scaling factor of each axis are different)

P: Scaling factor(applies when the scaling factor of each axis are the same)

G50: Disable scaling function

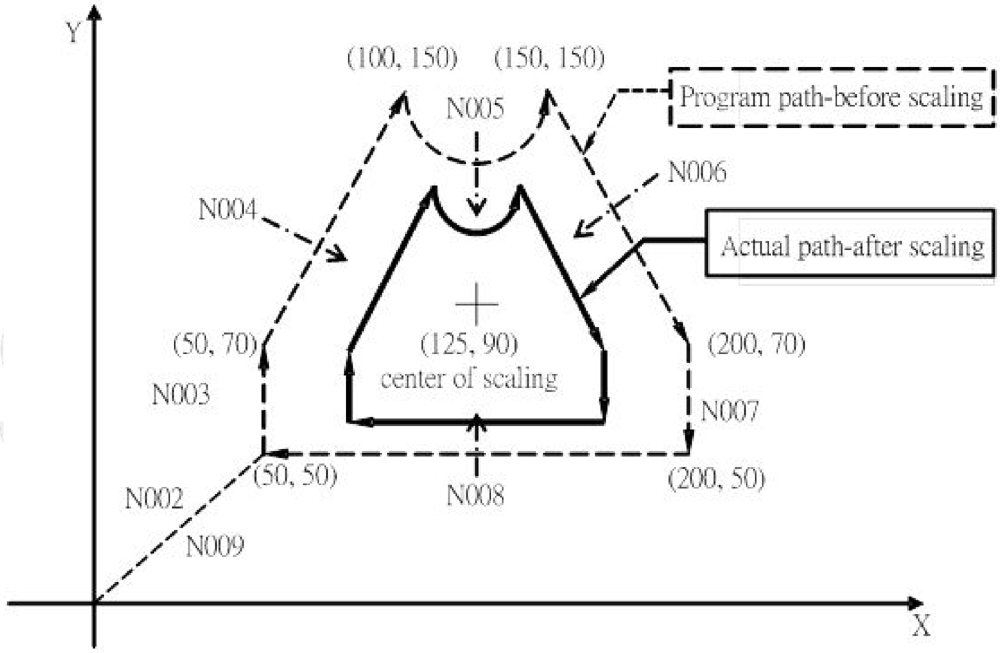
2.28.2 **Description**

G51 command can enlarge or reduce the cutting contour with setup value.

2.28.3 **Note**

1. Do not apply this G code with G51.1.
2. While applying this function to arc interpolation(G02,G03), if the setup scaling factor of each axis is different, the largest one will be set as the magnification value of arc radius.
3. Please use G17, G18, and G19 to set the geometry axis of the working plane first. If the axial address of the G51 command is not the geometry axis, the scaling function will not work.

2.28.4 **Program Example**



Program description:

```
N002 G51 X125.0 Y90.0 P0.5; //decide the scaling center X125,Y90, scaling factor 0.5 and apply to steps N003~N009
Noo3 G00 X50.0 Y50.0;  // rapid orientation
N004 G01 Y70.0 F1000; // linear interpolation, feedrate 1000mm/min
N005  X100.0 Y150.0;
N006 G03 X150.0 I25.0; // arc interpolation, radius 25mm；
N007 G01 X200.0 Y70.0; // linear interpolation
N008 Y50.0;
N009 X50.0;
N010 G00 X0.0 Y0.0; // fast return
N010 G50; // disable scaling function
N011 M30; // program ends
```

2.29 G51.2/G50.2- Polygon cutting (C-Type)

2.29.1 **Command Form**

Enable polygon cutting

G51.2 P_ Q_ [R_] [K_] ;

- P: Base spindle (workpiece axis) rotation speed rate or the number of blades. Default is P=1 (range from integer 1to 999)
- Q: Synchronous spindle (tool axis) rotation speed rate or the number of polygon edges. Default is Q=1(range from integer 1to 999).
- R: Synchronous phase shift (range from 0°to 359.999°).
- K: Synchronous group number 1~3, multiple sets of synchronization, can use up to 3 sets at the same time. When K is not specified, the first set of synchronization is preset. Multiple sets of synchronization function start at version 10.116.24M & 10.116.32.

Disable polygon cutting

G50.2 [K_] ;

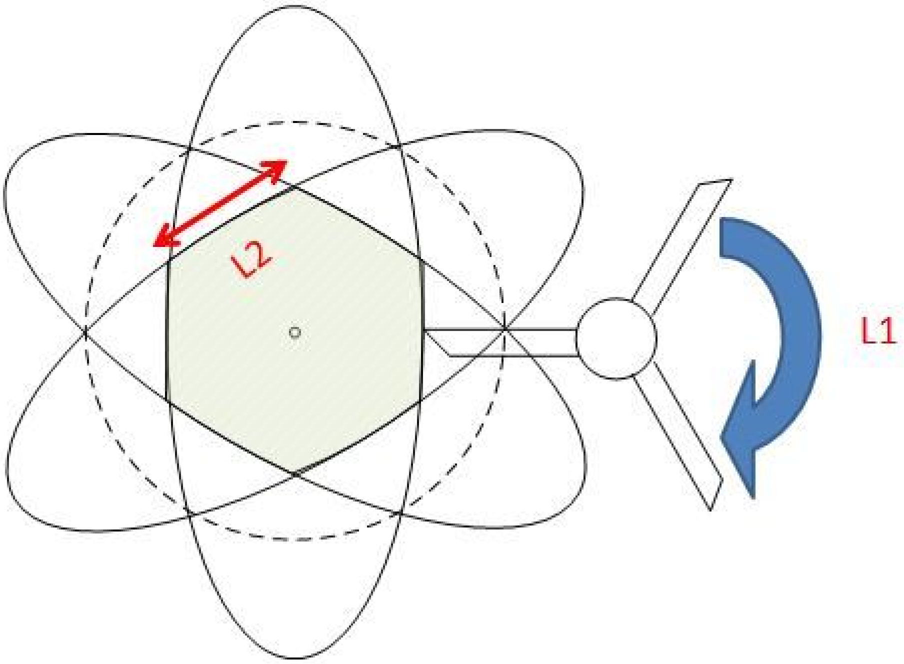
2.29.2 **Description**

1. G51.2 command synchronizes workpiece axis and tool axis at a fixed phase shift and RPM ratio to perform polygon cutting.
2. Synchronous spindle RPM: Base spindle RPM * Q / P
3. Synchronous phase shift: The clockwise angle difference between Synchronous spindle and Base spindle. Phase synchronization will not be enabled if no R argument given.
4. G50.2 disables polygon cutting.
5. G51.2 is available since version 10.113.0, unavailable in earlier versions (9.0 and 10.0).

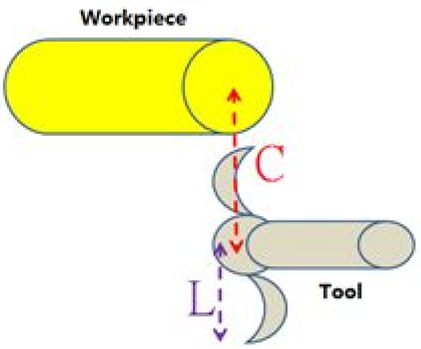
2.29.3 **Note**

1. **Spindle status description:**
   a. When the synchronization complete signal is On, pressing Reset will cancel G114.1 synchronization status (synchronization complete signal Off) after both spindles stop.

   b. When the synchronization complete signal On, G50.2 will cause the system to directly cancel the synchronization status (synchronization completion signal Off).
   c. The Base Spindle cannot use the synchronous function under position control mode (C63), and Synchronous Spindle is not recommended to use the synchronous function under position control mode.
2. **Multiple sets of synchronization:**
   a. The synchronous start command (G114.1) can be repeated (but the K value is not repeatable).
   b. A Base Spindle can have multiple Synchronous Spindle at the same time.
   c. The Synchronous Spindle can no longer be a Base Spindle of other spindles. (COR102)
   d. The value of System Data 45/46 is the angle difference between the Base Spindle and the Synchronous Spindle of the last synchronization command. The set is displayed as the angle difference of the second-to-last set in the same period after sync cancel, and so on.
3. **Precaution when machining:**
   a. When giving synchronous phase shift, the R value equals to the wanted angle difference between tool and workpiece multiply Q and divided by P (see example).
   b. P and Q values must be integers. For non-integer ratio 1:2.5, user shall use equivalent integer ratio instead, e.g. G51.2 P2 Q5.
   c. To ensure that the absolute position of the workpiece is correct, a tool zero teach is necessary when the tool is first installed (see example preparation).
   d. In actual cutting, notice that the perimeter between the blades should be longer than the polygon edge. See the following figure, L1 must be longer than L2 to ensure that the polygon shape is correct.

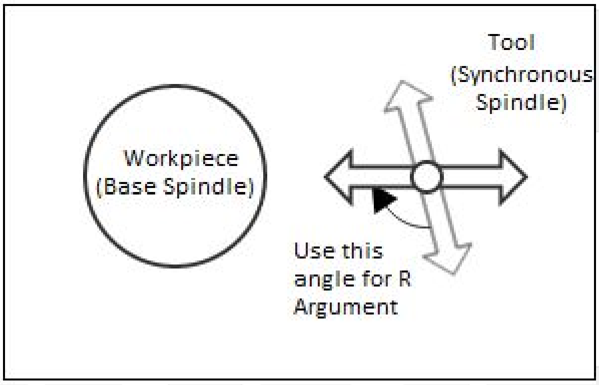


   e. G51.2 is a new processing technique that uses the speed difference between the tool and the workpiece to quickly machine a polygonal workpiece, but the cutting mechanism often causes non-planar workpiece surface due to the change of machining conditions. The following table is used to calculate the resulting flatness of workpiece for reference by the first-line colleagues.

RPM Ratio (i)	**G51.2 Result of Workpiece**			
>2	K>L	K=L	K<L	
	Convex	Flat	Concave	
	Formula: K = C/(i-1)^2			
=2	Convex			
<2	Convex			L=Tool radius; i = RPM Ratio (Q/P); C= Distance from Tool center to Workpiece center

2.29.4 **Example**

- **Preparation**
  To assure the absolute position of the workpiece (predictable absolute angle of the finished workpiece), tool's home position needs to be setup. There are three different ways to setup the tool's home position:
  a. Measure the angle between the tool zero point and tool setting position, phase difference defined via R argument.



  b. Adjust the tool to the tool setting position and set the tool position to the home position.
  c. Adjust the tool to the tool setting position and then perform the main synchronous angle teaching (F4->F4->F3). After this action, the reference angles of the Base and Synchronous axes will be stored in the Registry Table.
  (Note) Tool setting position: Align the tool blade to the direction from tool center to workpiece center (see schematic above). Performing any of the above three tool setting above can ensure the workpiece at 0 degree is aligned (refer to the schematic). To shift the workpiece by an angle, just add (the angle to be shift) x Q /P) to existing angle shift value (R).

- **Sample Command**
  Ex1. Hexagon with 3 flute (blade) tool : G51.2 P3 Q6 (or G51.2 P1Q2)
  Ex2. Pentagon with 2 flute (blade) tool : G51.2 P2 Q5

Example 1

```
S1 = 1000                //Workpiece axis (basic spindle) rotate speed 1000 RPM
M03                         //Workpiece axis (basic spindle) spindle rotate CW
S2 = 500                  //Tool axis (Synchronous spindle) rotate speed 500
                                //RPM
M204                       //Tool axis (Synchronous spindle) spindle rotate CCW
G51.2 P1 Q2 R60     //Tool axis (Synchronous spindle)
                                //synchronizes to 2000RPM and the phase
                                //difference is 30 degree. Cut for quadrangle
M81                         //Reading S62, check the synchronization success.
G01 X50                   //Start cutting
G04 X5
G01 X0                    //return
G50.2;                      // cancel polygon cutting
G51.2 P1 Q3 R180   //Tool axis (Synchronous spindle)
                                 //synchronize to 3000RPM and the
                                 //phase difference is 60 degree. Cut for hexagon.
M81; // reading S62. Check the synchronization success.
G01 X50                   //start cutting
G04 X5
G01 X0                     //return
G50.2                       //cancel polygon cutting
M05                          //Workpiece axis (basic spindle) stop
M205                        //Tool axis (Synchronous spindle) stop
M30                          //program finish
```

Note: After version 10.116.1, the core will automatically wait for synchronization. Non eed for the M code (M81).

Example 2 (Multiple sets of simultaneous use)

Scenarios:

Pr4021 = 1 (K1: 1st spindle)
Pr4022 = 2 (K1: 2nd spindle) // The first and second spindles are synchronized in other processing areas.
Pr4023 = 3 (K2: 3rd spindle)
Pr4024 = 4 (K2: 4th spindle) // The third and fourth spindles clamp the workpiece while rotating
Pr4025 = 3 (K3: 3rd spindle)
Pr4026 = 5 (K3: 5th spindle) // The fifth spindle follows the third spindle for polygon cutting

```
M03 S1000 // spindle 1 CW on
M203 S2=1500 // spindle 2 CW on
M303 S3=2000 // spindle 3 CW on
M403 S4=300 // spindle 4 CW on
M503 S5=100 // spindle 5 CW on
G04 X3. // wait

G114.1 K1 // enable 1st spindle syncronization
G04 X3. // wait
G114.1 R90 K2 // enable 2nd spindle syncronization
G04 X3. // wait
G51.2 P1 Q2 R60 K3 // enable 3rd spindle syncronization
```

```
G04 X3. // wait
S1500 // change spindle target speed
G04 X3. // wait
S500 // change spindle target speed
G04 X3. // wait

G113 K2 // diable 2nd spindle syncronization
G50.2 K3 // diable 3rd spindle syncronization
G113 K1 // diable 1st spindle syncronization
G04 X3. // wait

M05 // stop spindle 1
M205 // stop spindle 2
M305 // stop spindle 3
M405 // stop spindle 4
M505 // stop spindle 5
M30 // end
```

Synchronization error

Since the RPM of the two spindles of G51.2 can be different, the synchronous angle error is not simple subtraction but calculated using the following formula (will be displayed in System Data 45 and 46):
Synchronous angle error = (synchronous axis feedback angle - synchronous axis reference angle) -synchronous ratio* (base axis feedback angle - base axis reference angle) - phase difference

among them:
The synchronous ratio is Q / P, and the reference angle is the setting value of the Registry Table (set by F4->F4->F3)
The phase difference is R, and the feedback angle is the read value from encoder (equivalent to R761~R776)

2.30 G52- Local Coordinate System Setting (C-Type)

2.30.1 Command Form

G52 X__ Y__ Z__ ;

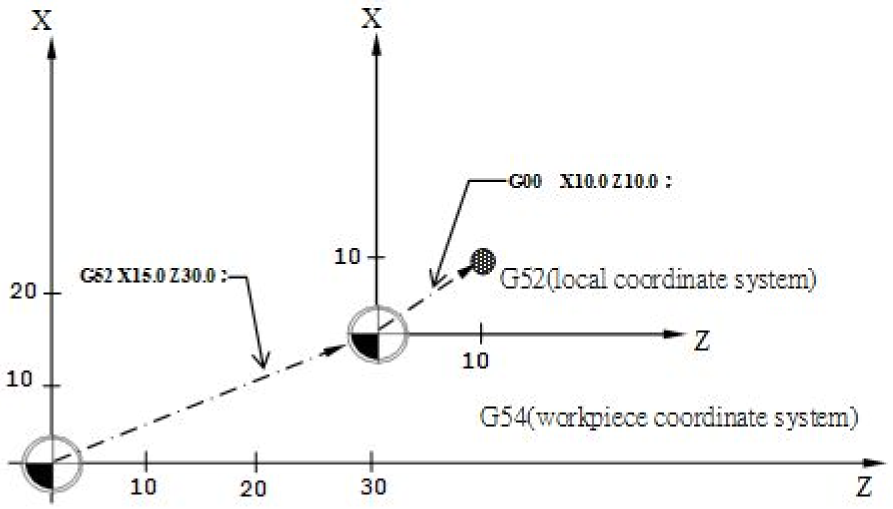
X, Y, Z: Set the local coordinate system

G52 X0.0 Z0.0: Cancel local coordinate

Description

When program needs an additional "sub-coordinate system" besides current a Workpiece Coordinate System (G54~G59.9), this G52 command is used to create the sub-coordinate system called **Local Coordinate System**.

Illustration


X
X
G00 X10.0 Z10.0 ;
10
G52(local coordinate system)
20
G52 X15.0 Z30.0 ;
Z
10
10
G54(workpiece coordinate system)
10
20
30
Z

2.30.2 Example

G54 //specify workpiece coordinate system G54

G52 X15.0 Z30.0 //specify the zero point of local coordinate
//system to X15.0 Z30.0 of workpiece
//coordinate system

G00 X10.0 Z10.0 //positioning to X10.0 Z10.0 of local coordinate

G52 X0.0 Z0.0 //local coordinate system cancel

2.31 G52.1/G52.2-Axis Removal/Axis Borrowing Function(C-Type)

2.31.1 Instruction

G52.1 P_ Q_ R_

P_ Q_ R_ : Remove the axis name corresponding to the axis, the range: 100~999, the axis name refers to the last three codes of Pr321~Pr340 Axis name.

G52.2 P_ Q_ R_ [I_] [J_] [K_]

P_ Q_ R_ : Borrow the axis name corresponding to the axis, the range: 100~999, the axis name refers to the last three codes of Pr321~Pr340 Axis name.

I_ : Waiting for response setting, range: 0~2, if not set, the default value is zero.

0 : Wait at this block of NC program until the axes to be borrowed are all successfully borrowed, then execute the next block of NC program;
1 : If all the specified axes cannot be successfully borrowed, an alarm COR-364 Axis borrowing function failed to borrow will be issued;

- If the J argument is set, the user can judge whether the borrowing is successful by using the # value specified by the J argument in MACRO.
- If the K argument is set, before successfully borrowing all the specified axes, the axis group will wait for the time specified by the K argument in this block of NC program. An alarm will be issued when it cannot be successfully borrowed after the time has elapsed.

2： If there is a axis cannot be borrowed successfully, then do not borrow any axis in this command and no alarm will be issued. Continue to execute the next block of NC program.

- If the J argument is set, the user can judge whether the borrowing is successful by using the # value specified by the J argument in MACRO.
- If the K argument is set, before successfully borrowing all the specified axes, the axis group will wait for the time specified by the K argument in this block of NC program. The next NC block will be executed when it cannot be successfully borrowed after the time has elapsed.

J_： The # variable that stores the borrowing result information, the range: 27~400 (corresponding to #27~#400), if it is not set, the borrowing result will not be returned to any variable.
The meaning of the borrowing result information is:

0: The borrowing of any specified axis in this block failed;
1: The borrowing of all specified axes in this block is successful.

K_: Waiting response delay time (with a decimal point, in seconds; without a decimal point, in milliseconds. Use range: 0.001 to 9999.999 seconds).

2.31.2 Description

When multiple axis groups need to use one axis in turn, the axis can be set as a roaming axis. By using G52.1, G52.2 command to switch the control of roaming axis among axis groups, that ensures the axis will not be controlled by two axis groups at the same time. The above operations also ensure the position synchronization between axis groups.

Setting reference for roaming axis: Pr742 *The rules of the axis group shared axis

2.31.3 Precautions

- About arguments:
    - P, Q, R:
        - A row of G52.1 and G52.2 can be removed or borrowed from at least one and at most three axes, corresponding to the three arguments of P, Q and R respectively.
        - If G52.1, G52.2 do not set P, Q, R arguments, an alarm COR-363 Invalid axis removal/axis borrowing function will be issued.
        - If the axis corresponding to the P, Q, R arguments cannot be found, an alarm COR-363 Invalid axis removal/axis borrowing function will be issued.
        - It is not supported to remove or borrow axis through virtual axis name. Virtual axis reference: G10L800, G10L801.
    - I:
        - The argument I determines the waiting for response of the three axes P, Q and R at the same time.
    - J:
        - After the borrowing instruction ends, the borrowing result will be returned to a # variable, and the variable number is determined by the J argument.
        - For example: G52.2 P100 Q200 R300 I2 J27. Because the third axis cannot be borrowed, no axis is borrowed in a row of G52.2. At the end of this instruction, #27 is filled with 0.

        - If the argument I is set to 0 or not set, storage of the borrowing result is not supported, and the system will ignore the J argument setting.
    - K:
        - If the argument I is set to 0 or not set, the waiting response delay time is not supported, and the system will ignore the K parameter setting.
    - If the argument P, Q, R, I, J are not integers, an alarm COR-146 Single block argument type error will be issued.
    - If the argument is an integer but exceeds the settable range, an alarm COR-363 Invalid axis removal/axis borrowing function will be issued.
- Axis borrowing or removal is only applicable to roaming axes, and an alarm COR-363 Invalid axis removal/axis borrowing function will be issued for borrowing or removing command under general axes.
- When using the axis removal/axis borrowing Function, switch the corresponding axis group to manual mode before executing manual operations on the borrowed axis. If the borrowing function is not used, always switch to manual mode on the axis group with the smallest group number to perform manual operations on the specified axis. It is recommended to enable the axis borrowing function before using manual operations on roaming axes. Supported version—10.118.88J, 10.120.18J, 10.120.32Q, 10.120.42E, 10.120.47, 12.0.4.
- Before the axis borrowing or removing, the coordinate will decelerate to zero first, and then the axis borrowing or removing command will be processed.
- Before the axis borrowing or removing, it must decelerate to zero speed before processing the borrow or removal command. The axis waiting behavior is determined by the setting of parameter I.
- If the argument I is set to 0 for borrowing, the system may get stuck in the "waiting for borrow" state due to the following reasons:
    - The multiple axis groups may wait for the roaming axis that is borrowing by the other coordinate. Solution: Modify the NC program to make sure the other coordinate remove roaming axis first.
    - The axis to be borrowed or removed has not decelerated to zero.
- An attempt is made to execute borrow or removal functions on a non-existent axis, an alarm COR-363 Invalid axis removal/axis borrowing function will be issued.
- If the axis to be removed is not borrowed by the coordinate, an alarm COR-363 Invalid axis removal/axis borrowing function will be issued.
  if the axis to be borrowed has been borrowed by the coordinate, an alarm COR-363 Invalid axis removal/axis borrowing function will be issued.
- The number of axes that can be borrowed by a axis group is limited to the total number of roaming axes belongs to the axis group. Multiple G52.2 command can be issued to borrow all roaming axes.
- For non-CNC main system axis groups, by writing G52.1, G52.2 commands in the sub-program specified by the main program number (ex: R532) of each axis group to achieve the roaming axes removal or borrowing.
- For the PLC Rn sub-program components, by writing G52.1 and G52.2 commands in the designated processing program to achieve the roaming axes removal or borrowing.
- The non-linear kinematic transform related axes are not allow to set as roaming axes. The following conditions will issue an alarm COR-363 Invalid axis removal/axis borrowing function. Include:
    - RTCP(G43.4, G43.5)
    - Tilted working plane machining(G68.2, G68.3 + G53.1, G53.3, G53.6)
    - Polar Coordinates Interpolation(G12.1)
    - General 2D Kinematic Transform(Special Machine)
- It is forbidden to use the break point return in the path with the roaming axis command, because it may move to the unborrowed roaming axis, otherwise an alarm COR-365 Issue a movement command to the unborrowed roaming axis will be issued.
- Before the axis group or PLC Rn subprogram component borrows the roaming axis successfully, it is not allowed to issue a command to the axis, otherwise an alarm COR-365 Issue a movement command to the unborrowed roaming axis will be issued.
- Supported version: 10.118.42R, 10.118.48C, 10.118.50 and later versions.

2.31.4 Example

Example 1: Multi-axis groups execute the same axis in turn

Pr321~Pr324 Axial name = {101, 200, 300, 102}

(There are four axes X1, Y, Z, X2 in the system)

Pr701~Pr704 Axial belonging axis group = {1, 3, 3, 2}

Pr742 The rules of the axis group shared axis = 1

(Y, Z axes are roaming axes, the first and second axis groups can borrow Y, Z axes)

$1	$2

// Initial state: owned axis X1.
G04.1 P1;
G52.2 P200 Q300 I0; // Borrow Y, Z axes.
G04.1 P2;
G52.1 P200 Q300; // Remove Y, Z axes.
G04.1 P3;
#400 := -1;
G52.2 P200 Q300 I2 J400 K5.; // Borrow Y, Z axis, wait for 5 seconds to continue to execute the next block, #400 is written as 0.
#399 := -1;
G52.2 Q300 I2 J399; // Borrow Z axis successfully, #399 is written as 1.
G04.1 P4; // Ensure that $1 M30 will not be executed too early.
M30;
	// Initial state: owned axis X2.
	G04.1 P1; // Synchronize with $1, avoid $2 M99 back to head of the program and continue to execute.
	G04.1 P2; // Make sure to borrow Y, Z axis from $1 first.
	G52.2 P200 I1 K5.; // Borrow Y axis(may not be borrowed at first, wait 5 seconds to borrow successfully).
	G04.1 P3; // Ensure that $2 will borrow the Y axis successfully.
	G04.1 P4;
	M99;

Example 2: Determine the follow-up processing path based on whether the borrowing is successful or not

Pr321~Pr322 Axial name = {100, 200}

(There are X, Y axes in the system)

Pr701~Pr702 Axial belonging axis group = {3, 3}

Pr742 The rules of the axis group shared axis = 1

(X, Y axes are roaming axes, the first and second axis group can borrow X, Y axes)

```
%@MACRO
G10 L1000 P6000 R0;           // R6000=0 cycle state flag(busy)
#27:=#0;                     // Clear the # value of the return value
G52.2 P100 Q200 I2 J27;       // If both X and Y axes can be borrowed, then two axes will be borrowed.
                            // if any one of the X or Y axis cannot be borrowed successfully, then neither axis will be
borrowed. The borrowing result will be returned to #27.
IF ( #27 = 0 ) THEN            // Use #27 to determine the next maching NC program
   G10 L1000 P6000 R999;     // R6000=999 cycle state flag(fail), trigger PLC to change maching NC program
file(not including X, Y axes maching)
   M30;
END_IF
//... Including X, Y axes maching...//
G52.1 P100 Q200;             // Remove X, Y axes
G10 L1000 P6000 R1;          // R6000=0 cycle state flag(finish)
M30;
```

2.32 G53- Machine Coordinate System Positioning (C-Type)

2.32.1 **Command Form**

G53 [P1] X___ Y___ Z___ [F1=_] [Q=_];
X: Move to specified X in machine coordinate.
Y: Move to specified Y in machine coordinate.
Z: Move to specified Z in machine coordinate.
P1： active constant speed command
F1： feedrate mm/min or inch/min
Q： Overlapping distance

Description

Machine Zero Point is the fixed origin set by the machine builder during the CNC machine production, and Machine Coordinate System is fixed to the zero point. When the G53 command and its arguments are specified, the tool moves to the specified position on machine coordinate system.

Additional Remark

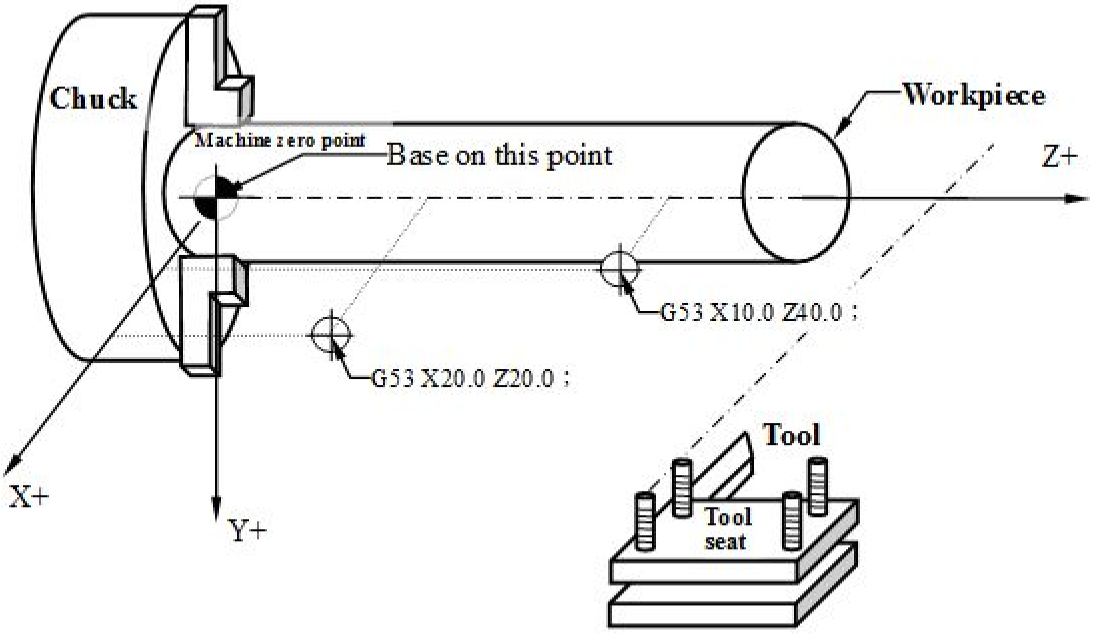
When the axis type (Pr221 ~236) is rotary axis, check SYNTEC CNC Parameters for detail.

Notice

1. G53 command is valid only in the programmed block（returns to program coordinate system if only the coordinate arguments are given in the next block);
2. G53's arguments are valid only in absolute mode (G90) and invalid in the incremental mode (G91).
3. If Pr3809 set 1 and G53 is specified together with the UVW command it won't be take as the XYZ axis inc. command.
4. Prior to specifying G53, cancel related tool radius, length or position compensation.
5. Prior to specifying G53, make sure Machine Zero Point is properly homed.
6. G53 default feedrate is feedrate of G00.

7. Constant feedrate is legally with both P1 and F1 argument are determined. If P1 argument is missing, system wouldn't refer F1 argument as moving speed.
8. G70/G71 supported unit of F1 can be mm/min or inch/min.
9. Unit of F1 can be mm/min, but always remains G94. G93/G95 is invalid.
10. F1 argument support version are 10.118.41P, 10.118.48A, 10.118.49 and after.
11. When using the G53 command for axial overlapping, both the front and rear blocks are G53 commands.
12. The last G53 must have no Q parameter when at the end of the continuous G53 overlapping.
13. Q argument support version are 10.118.41P, 10.118.48A, 10.118.49 and after.

Example



G53 X20.0 Z20.0 //move to specified position in machine coordinate

G53 X10.0 Z40.0 //move to specified position in machine coordinate

2.33 G54…G59.9 - Workpiece Coordinate System (C-Type)

2.33.1 Command Form

```
G54 X___ Y___ Z___
G55 X___ Y___ Z___
G56 X___ Y___ Z___
G57 X___ Y___ Z___
G58 X___ Y___ Z___
G59 X___ Y___ Z___
G59.1 X___ Y___ Z___
G59.2  X___ Y___ Z___

…
G59.9  X___ Y___ Z___
```

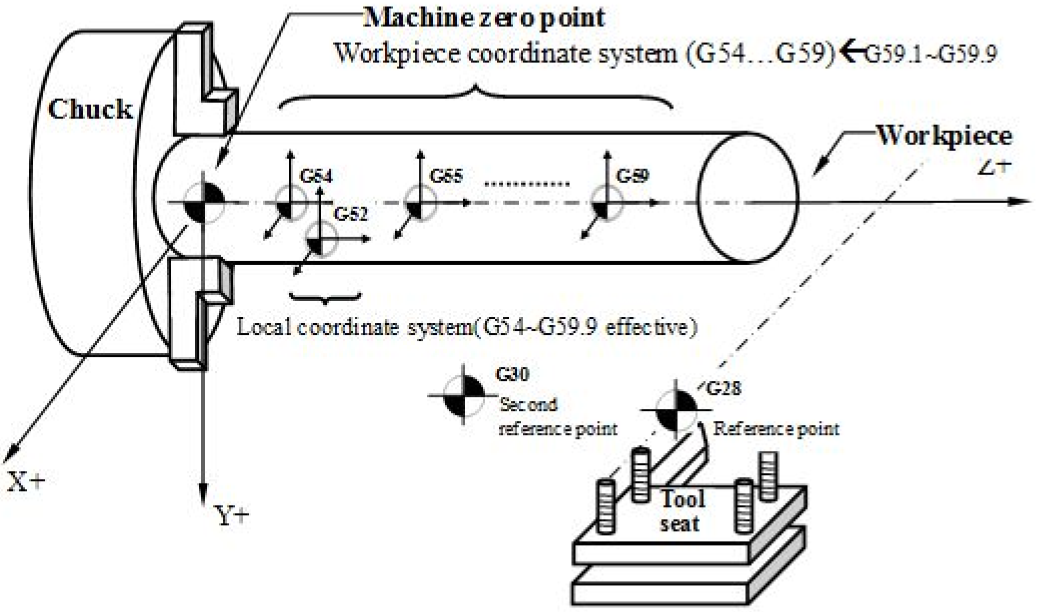
G54: First workpiece coordinate system
…
G59: Sixth workpiece coordinate system
G59.1: Seventh workpiece coordinate system
…
G59.9: 15th workpiece coordinate system

X,Y,Z: Move to specified position in workpiece coordinate system which has been set

Description

When operating the lathe, we may repeat performing the same process in different positions on same workpiece. By specifying a G code from G54 to G59 and G59.1 to G59.9 (total 15 workpiece coordinate systems), same program can repeat at different positions. The function can be set by Pr3229 "Disable workpiece coordinate system (0: enable, 1: disable)".

2.33.2 **Illustration**



2.34 G54.4: Enable/Disable Workpiece Coordinate Rotation Functionality (C-Type)

2.34.1 **Command Form**

G54.4 P_;

P: Workpiece Coordinate System Rotation Function Switch, 0: Disable **current workpiece coordinate** rotation, 1: Enable **current workpiece coordinate** rotation

2.34.2 Description:

G54.4 is used to enhance the operability of the workpiece coordinate system (G54) for users. It temporarily disables the rotation function of the workpiece coordinate system in the NC file, thereby improving customization flexibility and convenience.

2.34.3 Example

As an example, use the macro during tool alignment to toggle the rotation function of the workpiece coordinate system using **G54.4 P_**:

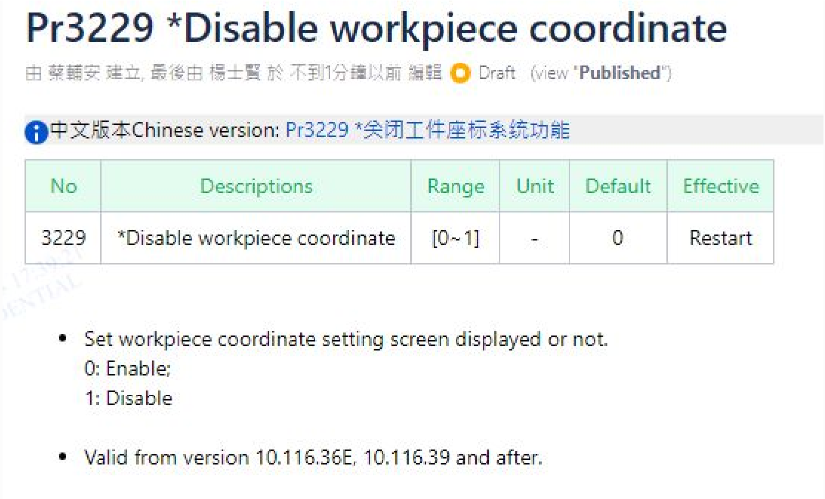
Tool Alignment Macro

```
BackupModal();                      //BackUp modal state
#100 = #1849;                       //Back up current state of G54 rotation

G54.4 P0;                           //Close G54 rotation
G91;                                //Plan for tool alignment program with incremental mode
MeasureWorkpieceTiltAngle();        //Composed of multiple G31 segments, measure the rotation amount of the coordinate system.
G54RotationSetting();               //The measured rotation amount of the coordinate system is filled into the corresponding # value.

G54.4 P1;                           //Enable G54 rotation
CenteringProgram();                 //The center point of the workpiece is measured by multiple G31 segments.
G54TransitionSetting();             //Fill the coordinates measured by the program into the origin of the workpiece coordinate system.

G54.4 P#100;                        //Recover G54 rotation state
RecoverModal();                     //Recover other modal state
```

2.34.4 Notice

Pr3229 and Workpiece Coordinate Rotation Function Enabled State

Pr3229 Illustration



Pr3229 *Disable workpiece coordinate

由 蔡輔安 建立, 最後由 楊士賢 於 不到1分鐘以前 編輯 Draft (view "Published")

中文版本Chinese version: Pr3229 *关闭工件座标系统功能

No	Descriptions	Range	Unit	Default	Effective
3229	*Disable workpiece coordinate	[0~1]	-	0	Restart

- Set workpiece coordinate setting screen displayed or not.
  0: Enable;
  1: Disable
- Valid from version 10.116.36E, 10.116.39 and after.

Pr3229 can disable functionality of workpiece coordinate system， when Pr3229 = 1， G54.4 has no effect。

	Pr3229 = 0	**Pr3229 = 1**
G54.4 = 0	X	X
G54.4 = 1	O	X

O: workpiece coordinate rotation function On
X: workpiece coordinate rotation function Off

Warning

If the workpiece coordinate system rotation is enabled, the function will be deactivated when Pr3229 = 1.

Default Enabled State of Workpiece Coordinate Rotation Function

※ ***When the system boots up or the workpiece coordinate system function is enabled (Pr3229 = 1), if Pr3839 is set to 1****, the workpiece coordinate system rotation function will be automatically activated.*

Pr3839 Default Enabled Mode of Workpiece Coordinate Rotation Function

Check Enabled state of G54 Rotation

Can check if the workpiece coordinate system rotation is enabled by using #1849.

G54.4 Related Alarms

- If P is not specified for G54.4, the system will raise an alarm 【COR-023 Semantic error】

- If the argument range for G54.4 P_ is not 0 or 1, the system will raise an alarm 【COR-064 Invalid P address】

2.35 G61/G62/G63/G64- Cutting Mode (C-Type)

2.35.1 Command Form

G61; // exact stop examination mode

G62; // curved surface cutting mode

G63; // tapping mode

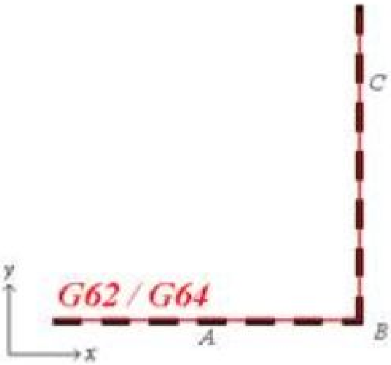
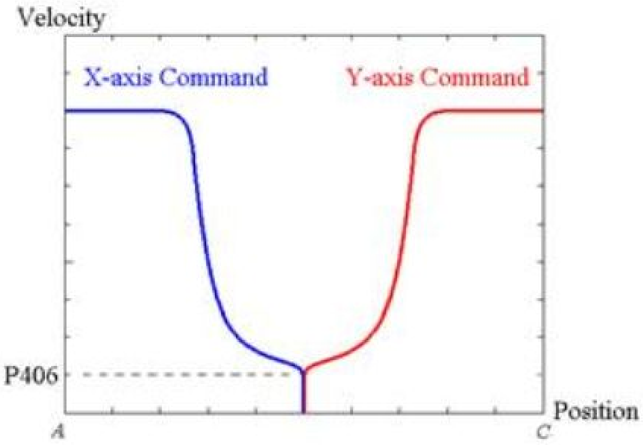
G64; // curved surface cutting mode

2.35.2 Description

The differences between each mode are listed below, the default mode is G64 cutting mode. After execute a certain mode, it'll be effective till another mode is assigned.

Command name	G code	Valid region	Description
Exact stop	G09	Only effective in block with G09.	When tool decelerates at the end of contour, The precision error occurs at the corner when the tool direction turns. G09 is used to control the precision error.
Exact stop mode	G61	Enable after execute G61, disable til G62, G63 or G64	The tool decelerates at the end of cutting contour, when it arrives at the end point, a feedback signal is sent to make sure the position is in the set range. The next contour will be executed after confirmation.
Curved Surface Cutting Mode	G62	Enable after execute G62, disable til G61, G63 or G64	Applicable to curved surface cutting. The tool does not decelerate at the end of contour(refer to the speed command curve shown below) and continues to execute next contour. Able to carry the P argument and select a high speed high accuracy parameter. (Note 2)
Tapping mode	G63	Enable after execute G63, disable til G61, G62 or G64	Applicable to tapping. The synchronous between spindle and feeding axis is determined by the ratio of spindle rotation speed S and feedrate F. During tapping, feed overrode and feed hold cannot be adjusted.
Cutting mode	G64	Enable after execute G64, disable til G61, G62 or G63	Applicable to curved surface cutting. The tool does not decelerate at the end of path (refer to the speed command curve shown below) and continues to execute next path. Able to carry the P argument and select a high speed high accuracy parameter. (Note 2)

Figure： The action of G62/G64 while cutting over the corner.

G Code	Cutting Path	Speed Command Curve
G62/G64		

Explanation：

G62/G64 corner speed control mode will slow the speed down to the corner speed with Pr406 while interpolation over the corner, so there will be no command contour error at the corner. For the machining requires repeating cutting process such as mold machining, this mode provides better corner precision and reappearance.
For the corner, the vibration caused by the JERK of speed command can be improved by Pr404, set Pr404 to 10~20 can make effective improvements.

2.35.3 **Note**

1. G62 / G64 mode are more suitable for mold machining.
2. G62 Pn/G64 Pn, n = 0 ~ 9, it's able to choose a high speed high accuracy parameter, n = 21 ~ 23, it's able to choose high speed high accuracy parameter for product. Valid version is 10.118.78B, 10.118.81.
3. For multiple high speed high accuracy parameter sets, the command later one will overwrite the former one. It'll return to the default parameter (P0) after reset.
    a. Before version 10.118.42T, 10.118.48E, 10.118.52, the default parameter (P0) is Standard Parameter.
    b. After version 10.118.42T, 10.118.48E, 10.118.52, the default parameter (P0) is Pr3835 Initial Machining Condition.
4. Using G62/G64 P_ or G62/G64 R_ to choose invalid precision channel, post COR-103 alarm.
5. The G63 mode requires the use of M03/M04 to specify the tapping rotation direction. M03 indicates right-hand tapping, while M04 indicates left-hand tapping.

2.36 G65- Single Marco Call (C-Type)

2.36.1 **Command Form**

G65 P__ L__;

P: Call program number ;
L: Repeat time;

2.36.2 **Description**

When the macro command is called, the P__ specified program number called for execution, and L__ specifies that repeats times of G65 (only repeat G65 block).

2.36.3 **Example**

```
G65 P10 L20 X10.0 A10.0 Q10.0;
//Call and repeat subprogram O0010 20 times, and take X, A, and Q arguments into the subprogram execution.
//That is, the values of the arguments #24, #1, #17 can be used in the subroutine for operation.
//The use of arguments is not limited to XYZ, as long as it meets the macro rules.
```

2.37 G66/G67- Call/Cancel Modal Marco Program (C-Type)

Command Form

G66 P__L__   Modal Marco **call**
G67   Modal Marco **cancel**
P: Program number to call
L: Repeat time (1 by default)

2.37.1 **Description**

When macro command (G66) is executed, the P___ number subroutine is called and L__ specifies the number of G65 repeat time. And after the a movement block finishes, the subprogram G66 called will be executed again. The mode is canceled until system executes a G67 block.
(if there is a variable operation in the called subroutine, notice that the variable has a pre-solution).

Example

```
G91;
G66 P10 L2 X10.0 Y10.0;     //call O0010 two times and input the value
                            //X10.0 Y10.0 into the program for calculation
X20.0;                        //move X axis to 20.0, then execute
                             //G66 P10 L2 X10.0 Y10.0
Y20.0                         //move Y axis to 20.0, then execute
                             //G66 P10 L2 X10.0 Y10.0
G67                           //cancel the model marco mode
```

2.38 G68/G69- Enable/Disable Turret Mirror Function (C-Type)

2.38.1 **Command Form**

G68 Enable X mirror function
G69 Disable mirror function

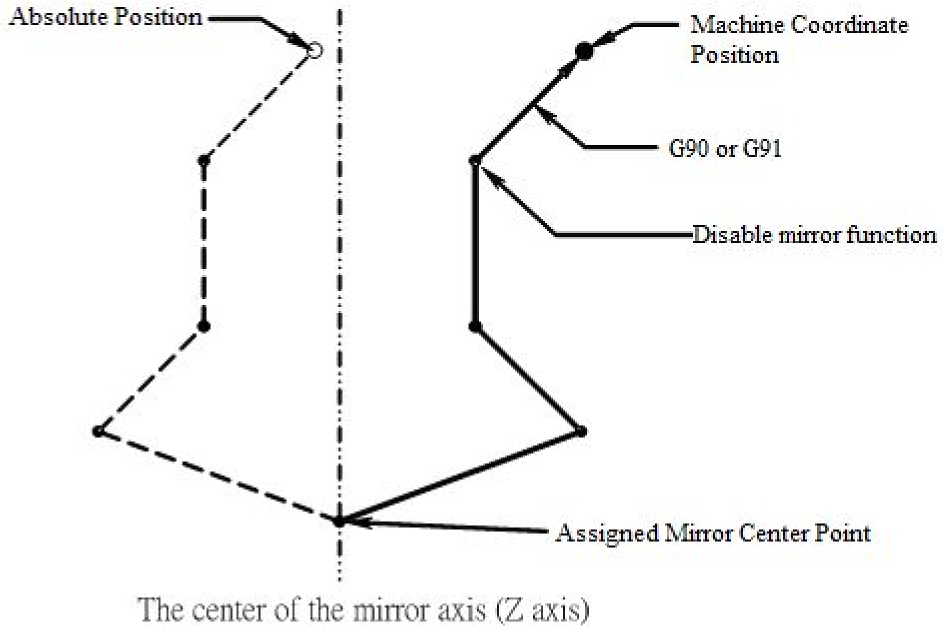
2.38.2 **Description**

In double turret lathe, G68 command can mirror the X coordinate referencing X0 for the convenience of double turrets so there is no need to consider turret moving direction while programming.

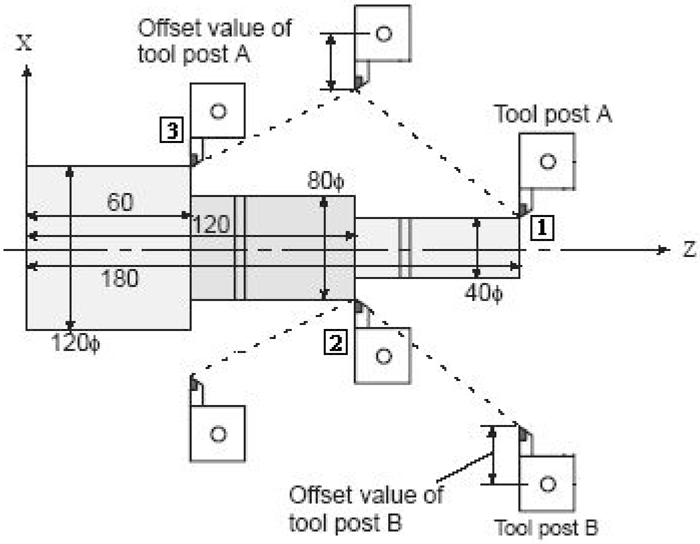
1. Arc interpolation, tool nose radius compensation, and coordinate rotation are all opposite.
2. The command is used in local coordinate, the center of the mirror moves when the counter reset or the working coordinate is changed.
3. When executing Reference Point Return (G28, G30) with mirror function enabled, the movement from start to mid-point is mirrored, from mid-point to reference point is not mirrored.
4. When execute the Return From Reference Point (G29) with mirror function enabled, the mid-point is mirrored.

2.38.3 **Precaution**

When disabling mirror function out of the center of the mirror, the absolute position cannot match the machine position. As figure shown below, this situation lasts until a absolute position [positioning of G90] or machine zero point return G28 and G30 are executed. If re-assign the center of mirror under this position without movement, the mirror center may be assigned to unexpected position. Must disable mirror function at the center of the mirror or do a G90 movement after disabling mirror function.


Absolute Position
Machine Coordinate Position
G90 or G91
Disable mirror function
Assigned Mirror Center Point
The center of the mirror axis (Z axis)

2.38.4 **Example**



```
T0101 //turret 1
G01 Z180. X40. //position-1
Z120.
T0202 //turret 2
G68 //enable X-axis mirror image
G01 Z120. X80. //position 2
Z60.
T0101 //turret 1
G69 //disable X-axis mirror image
G01 Z60. X120. //position 3
M99
```

2.39 G70/G71- Imperial/Metric Unit Selection (C-Type)

2.39.1 **Command Form**

G70;

G71;

2.39.2 **Description**

G70: Imperial unit setting

G71: Metric unit setting

After unit is switched, the workpiece coordinate offset, tool compensation table, system parameters, and reference point position are the same, system will take care of conversion automatically.

After unit conversion, the following operating units will change:

1. Display coordinates, feedrate units
2. Incremental Jog unit

3. MPG unit

2.39.3 **Precaution**

The rotary axis does not have unit change.

When the movement command is linear and rotary axis combined, the linear axis command is multiplied by 25.4 after converting to Inch, so the proportion of the rotary axis at the combined speed is greatly reduced. When converting to metric, the proportion of the linear movement will reduce significantly. Please pay more attention.

2.40 G72~G78- Complex Canned Cycle (C-Type)

2.40.1 **Description**

The workpiece materials of turning machines are generally cylindrical and need multiple rough cuts and one fine cut when machining certain size or shape.

Therefore, Syntec CNC has a group of canned cycles to automatically generate a series of tool paths for rough and fine cutting to reduce the workload of programming.

These commands automatically generate tool paths for workpieces that include linear, arc, and taper geometry contour.

X axis only supports setting as diameter axis, and Z axis only supports setting as radius axis.

2.41 G72- Fine Cutting Cycle (C-Type)

2.41.1 **Command Form**

G72 P(ns) Q(nf) ;
ns: Starting block number of the cutting cycle
nf: Ending block number of the cutting cycle

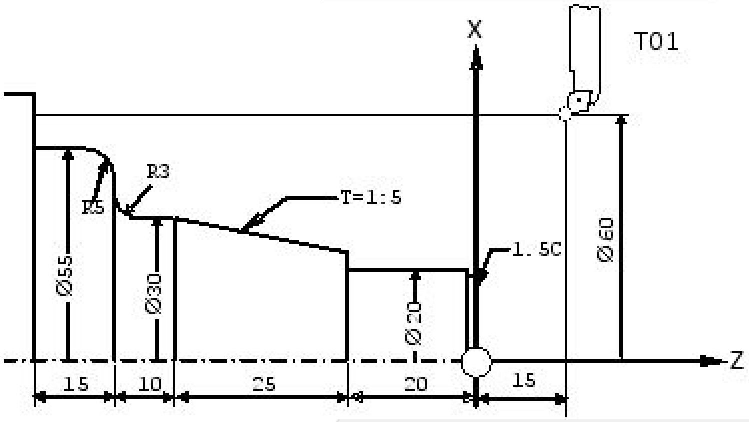
2.41.2 **Description**

1. Definition: Initial point, cycle path, retract path
    a. Initial point = the block position before turning cycle G code (G72).
    b. Cycle path = The path formed by the blocks between start number (P_) and end number (Q_).
    c. Retract path = the path from the end of the cycle path to the initial point.
2. G72 command is a fine turning cycle, also known as a contour turning cycle. This command must be used with a rough cutting cycle in the previous block.
3. In general, the fine turning cycle in the program is connected after the rough cutting cycle, and its execution range includes only from start block number"**P(ns)**" to the ending block number "**Q(nf)**".
4. After G71 / G72 / G73 cycle rough cutting, a fine cutting G72 must be performed to achieve the final required size.
5. Retracting path: G00 XZ is synchronized.
6. X axis only supports setting as diameter axis, and Z axis only supports setting as radius axis.

2.41.3 **Precaution**

1. The F, S, and T functions specified in G71, G72, and G73 blocks are invalid, but are valid between the G72 block numbers "ns"->"nf".
2. When the cycle of the G72 is finished, the tool returns to the starting point and reads the next block.
3. The subroutine cannot be called out in any single block between "ns" and "nf" used by G72 to G73.
4. It is recommended that the same contour of rough (G71/G72/G73) and fine turning be carried out continuously. In other word, avoid executing G72 after performing multiple rough turning. If it is needed to be used separately, please be careful the correctness of G72 start point location.
5. When the [G72 cutting contour] and its [retraction path] form a sharp contour, the path after enabling the tool compensation (G41/G42) may not be as expected. See program example 4.
6. The sequence numbers nf must be existing, otherwise the 【COR-010 G66 macro program call level is too deep】 alarm will be issued.
7. G72 command cannot be used with other G code within one single block.

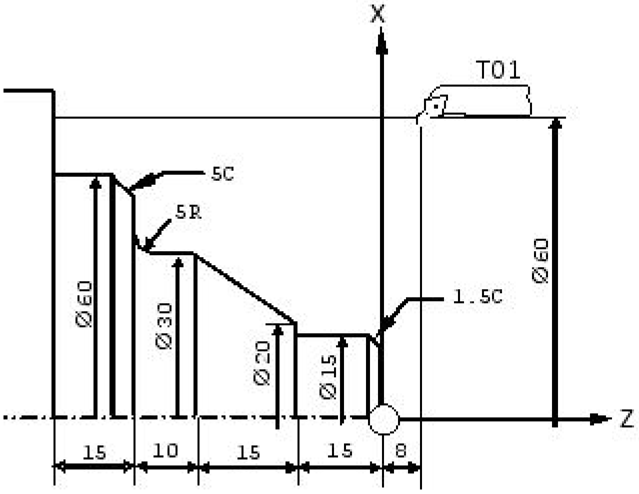
2.41.4 **Example 1**



```
T01; //Use tool No.1
G50 S5000; //maximum speed 5000 rpm
G96 S130 M03; //constant surface speed, surface speed 130 m/min,
//spindle rotates CW
G00 X60.0 Z15.0; //positioning to initial point
M08; //cutting liquid ON
G71 U2.0 R1.0; //X-axis cutting depth 2.0 mm, retract 1.0 mm
G71 P01 Q02 U0.8 W0.1 F0.3;
//Perform a horizontal (outer diameter) rough turning cycle with the block number N01→N02,
//The reserved X-axis distance for fine cut is 0.8 mm.
//The reserved Z axis distance for fine cut is 0.1mm, feed rate 0.3 mm/rev


N01G00 X17.0;
G01 Z0.0;
X20.0 Z-1.5;
Z-20.0;
X25.0;
X30.0 Z-45.0;//the contour to be cut

Z-52.0;
G02 X36.0 Z-55.0 R3.0;
G01 X45.0;
```

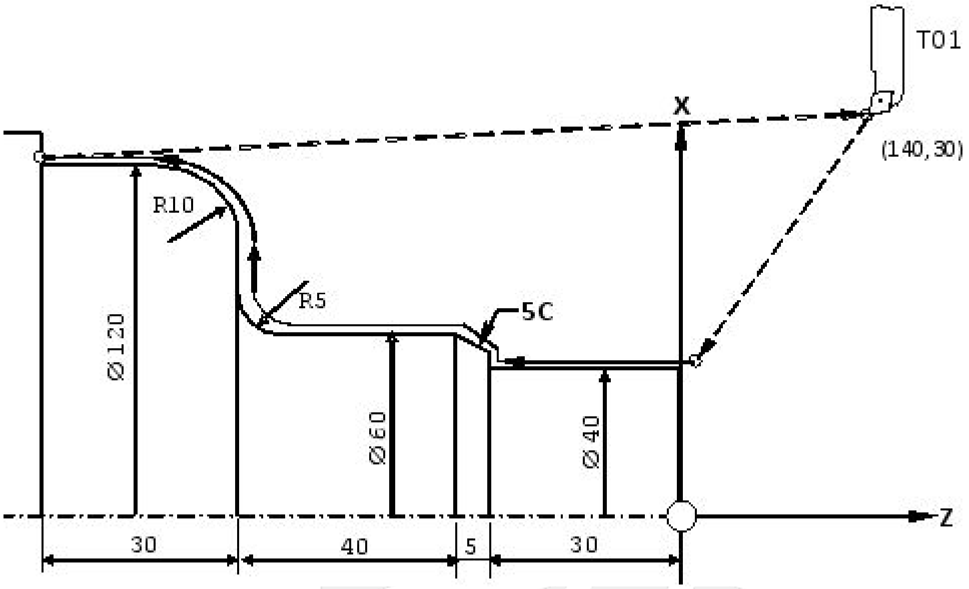
```
G03 X55.0 Z-60.0 R5.0;
N02G01 Z-70.0;
G72 P01 Q02; //execute the fine turning cycle with the block number N01->N02
M09; //cutting liquid OFF
M28 X60.0 Z20.0; //tool moves to specified mid-point then return to machine initial point
M05; //spindle stop
M30; //program end
```

2.41.5 **Example 2**



```
T01; //Use tool No.1
G50 S5000; //maximum speed 5000 rpm
G96 S130 M03; //constant surface speed, surface speed 130 m/min, spindle rotates CW
G00 X60.0 Z8.0;//positioning to initial point
M08; //cutting liquid ON
G72 W3.0 R1.0;//Z-axis cutting depth 3.0 mm, retract 1.0 mm
G72 P01 Q02 U0.8 W0.2 F0.6;
//Perform radial (end face) rough turning cycle with the block number N01→N02
//The reserved X distance for fine cut is 0.8 mm. The reserved Z distance for fine cut is 0.2mm.
//feed rate 0.6 mm/rev
N01G00 Z-55.0;
G01 X60.0;
Z-45.0;
X50.0 Z-40.0;
X40.0;
G03 X30.0 Z-35.0 R5.0; //the contour to be cut
G01 Z-30.0;
X20.0 Z-15.0;
X15.0;
Z-1.5;
N02X12.0 Z0.0;
G72 P01 Q02; //execute the fine turning cycle with the block number N01->N02
M09; //cutting liquid OFF
G28 X60.0 Z10.0; //tool moves to specified mid-point then return to machine initial point
M05; //spindle stop
M30; //program end
```

2.41.6 **Example 3**

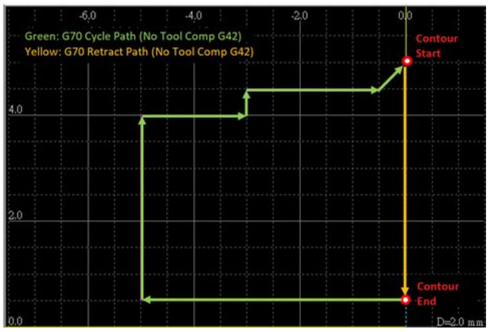
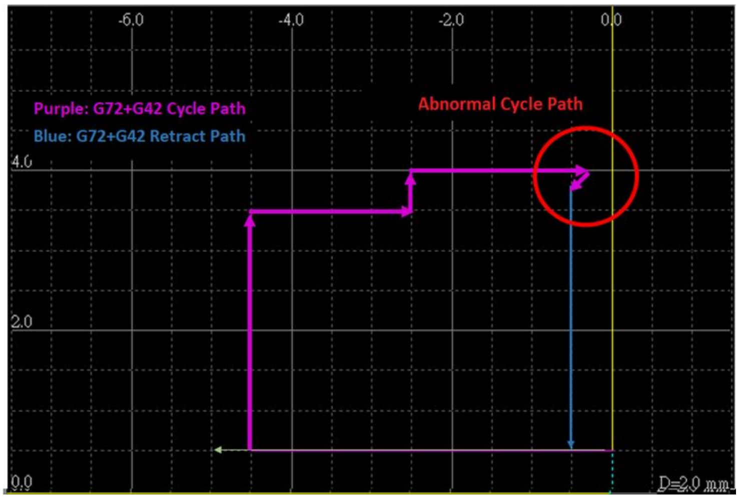


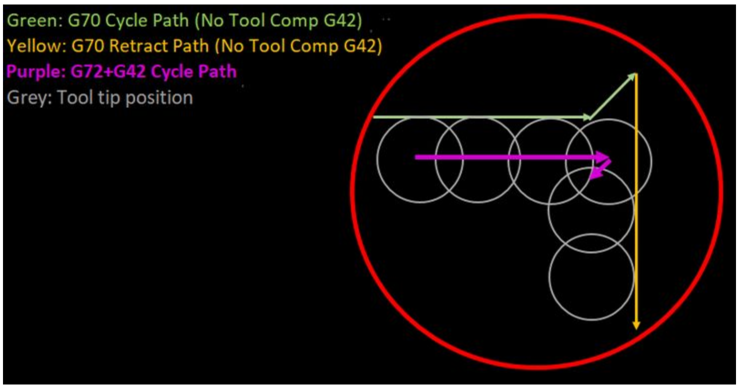
```
T01; //Use tool No.1
G50 S5000; //Maximum speed 5000 rpm
G96 S130 M03; //constant surface speed, surface speed 130 m/min
//spindle rotates CW
G00 X140.0 Z30.0; //positioning to initial point
M08; //cutting liquid ON
G73 U15.0 W15.0 R3.0;//X axial cutting amount 15.0mm, Z axial cutting amount
// 15.0 mm, cutting for three times
G73 P01 Q02 U0.8 W0.2 F0.3;
//perform the forming contour rough turning cycle, the serial number of the block is N01->N02,
//The reserved X distance for fine cut is 0.8 mm. The reserved Z distance for fine cut is 0.2mm.
//feed rate 0.3 mm/rev
N01G00 X40.0 Z5.0;//the contour to be cut
G01 Z-30.0;
X50.0;
X60.0 Z-35.0;
Z-70.0;
G02 X70.0 Z-75.0 R5.0;
G01 X100.0 ;
G03 X120.0 Z-85.0 R10.0;
N02G01 Z-105.0;
G72 P01 Q02; //perform the contour fine turning cycle, the number of the block is N01->N02
M09; //cutting liquid OFF
G28 X140.0 Z30.0; //tool moves to specified mid-point them return to machine initial point
M05; //spindle stop
M30; //program end
```

2.41.7 Example 4

[Cutting of G72] and its [retract path] form a sharp contour and cause an abnormal path after tool compensation.

Without G42	**With G42（path is not as expected）**

Setting
No interpolation
 	
Figure simulation result
 -6.0
-4.0
-2.0
0.0
4.0
2.0
0.0
Green: G70 Cycle Path (No Tool Comp G42)
Yellow: G70 Retract Path (No Tool Comp G42)
Contour Start
Contour End
D=2.0 mm
 	
Processing command
T01;
G97M3S60;
G50S1200;
 	
Z0.0;
X1.0;
 	
//execute fine turning（No interpolation）
G01;
G72P3Q4;
 	
//program end
M30;
 	
// fine turning contour
N03Z-5.0;
X8.0F2000;
Z-3.0;
X9.0;
Z-0.5;
N04X10.0W0.5;
	Setting
	Open right interpolation, tool nose radius 1.0
	
	Figure simulation result
	 -6.0
	-4.0
	-2.0
	0.0
	4.0
	2.0
	0.0
	Purple: G72+G42 Cycle Path
	Blue: G72+G42 Retract Path
	Abnormal Cycle Path
	D=2.0 mm
	
	Program
	
	T01;
	G97M3S60;
	G50S1200;
	
	Z0.0;
	X1.0;
	
	//Enable tool compensation and execute fine turning
	G01G42;
	G72P3Q4;
	
	//Program end
	M30;
	
	// Fine turning contour
	N03Z-5.0;
	X8.0F2000;
	Z-3.0;
	X9.0;
	Z-0.5;
	N04X10.0W0.5;

Without G42	With G42（path is not as expected）
	Description: The section where the turning is problematic, the G72 original turning path, the retracting path, and the tool position are shown as follows. Because the contours of [final cut of fine cutting] and [retract path] are too narrow for the tool tip, therefore the tool path is not as expected. 

Green: G70 Cycle Path (No Tool Comp G42)
Yellow: G70 Retract Path (No Tool Comp G42)
Purple: G72+G42 Cycle Path
Grey: Tool tip position

2.42 G73- Lateral (Outer-Surface) Rough Turning Cycle (C-Type)

2.42.1 Command Form

G73 UΔd Re H__ ;

G73 Pns Qnf UΔu WΔw F__ S__ T__ D__ ;

Nns ;

⋮

Nnf ;

Δd: Each cutting depth in the X direction, can be preset by system parameter Pr4013.
e: The retraction distance, can be preset by system parameter Pr4012.
ns: Sequence number of starting block in finishing cycle
nf: Sequence number of ending block in finishing cycle
Δu: The reserved amount of X-axis (outer surface) direction
Δw: The reserved amount of Z-axis (length) direction
F: Feed rate
T: Tool number
S: Spindle RPM setting
H: Machining method: 0 selects TYPE I method; 1 selects TYPE II method. System will judge the processing method by itself of no H sepcified.
D: Chip breaking switch. → Enable function : Set D to 1 / Disable function : Set D to others than 1, effective version: 10.118.70F

2.42.2 **Description**

The G73 command is a horizontal (outer surface) rough turning cycle, it can be used with tool compensation (G41/ G42) to machine the workpiece to a target contour with some reserve for finishing.

This cutting cycle needs a defined block range including the workpiece contour path, cutting depth of rough turning, and the reserved depth for fine turning.

Notice

This page is mainly for Pr4014=1, outer-face machining circumstances.

Pr4014 is used to set the rough turning mode. For other modes, see Pr4014 setting rough turning cycle mode.

First, introduce the definition and use it with the figures below for command detail.

Form: nouns definition table

#	Noun	Symbol	Description
1	Initial Point	A	1. One block before the rough turning cycle 2. The rough turning cycle G code (A-Type G73/G72 & C-Type G73/G74)
2	Reserved Point	C	1. To reserve a distance for fine turning, this point is offset a distance (Δw, Δu/2) from the initial point (A) 2. If no reserved distance (Δw, Δu/2) is specified, Reserved point (C) equals to Initial point (A).
3	Turning start point	D	1. The position starts to execute rough turning cycle.
4	Fine turning contour path	A' → B	1. The path formed by blocks between the start number (ns) and the end number (nf) is the final fine turning contour.
5	Fine turning start point	A'	1. Start position block number (ns).
6	Fine turning end point	B	1. End position block number (nf)
7	Turning end point	E	1. The end point of the rough turning cycle. 2. If no specified reserve distance (Δw, Δu/2) specified, the rough turning end point (E) equals to fine turning end point (B).
8	Retraction path	E → C	1. The path from the Turning end point (E) to the Reserved point (C).

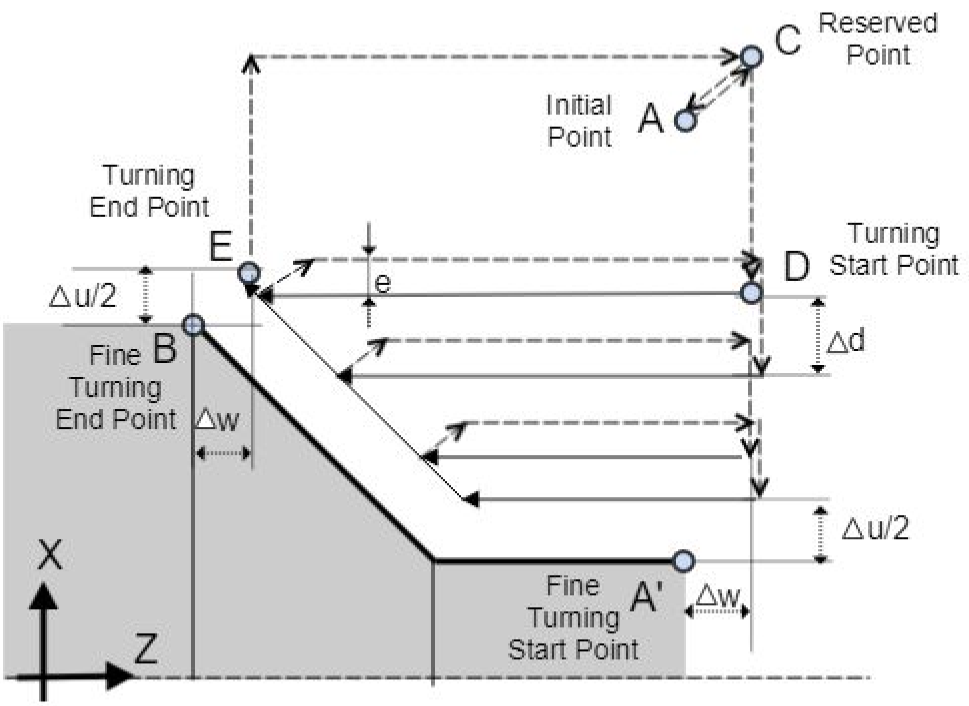
TYPE I

Restriction: Usually used to start machining from the end face. The contour **can only has one groove**. **Excluding the initial program segment,** the direction of monotonic increase (or decrease) along the X-axis is allowed to **change only once,** and each block must be uni-directional increase/decrease in Z axis. That is, the next block needs to be increased/decreased from the previous block.

Example: The followings are the path diagrams of two types of **TYPE I** fine turning contours.

Illustration: Type I Example 1

> ℹ In this example, the contour **has one groove**, and the X-axis maintains monotonic increasing behavior.



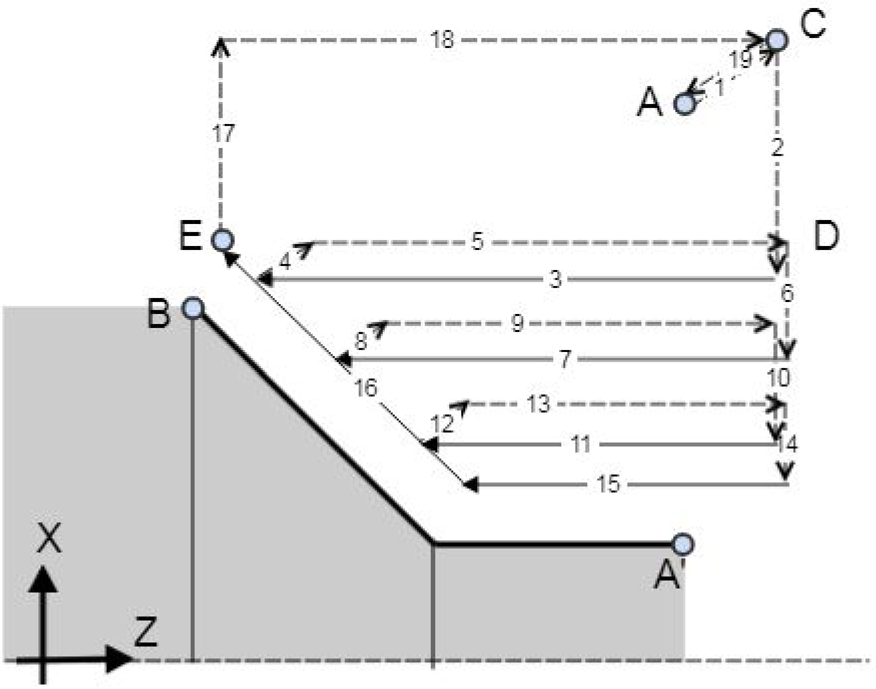
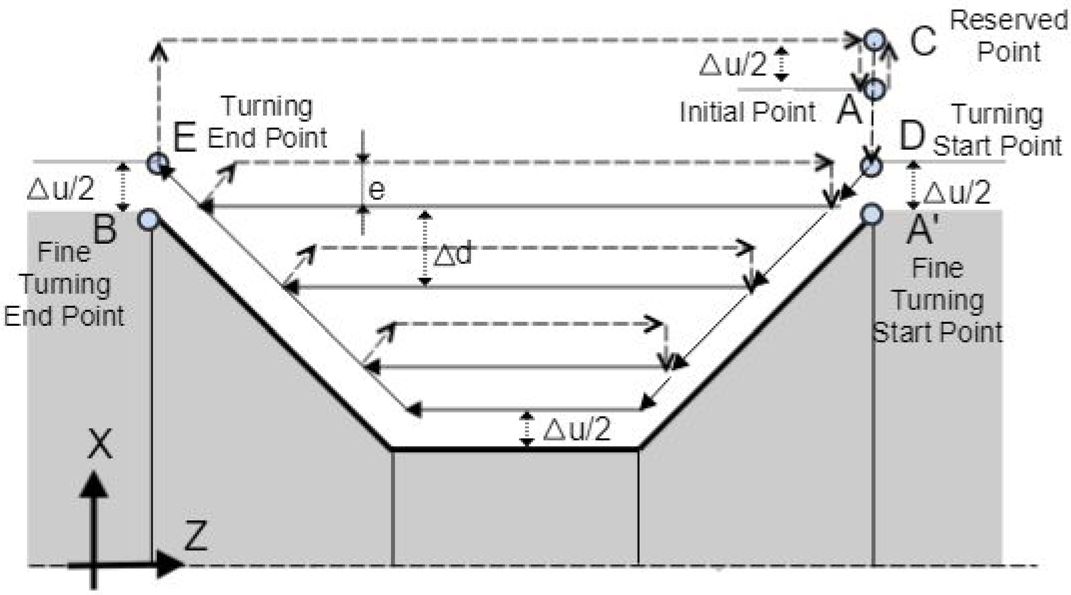
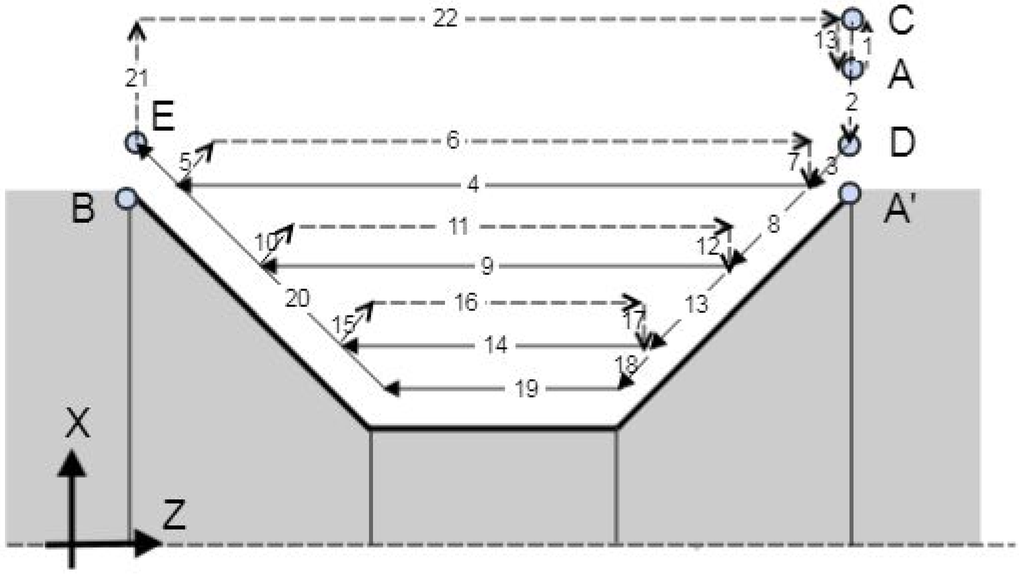


Illustration: Type I Example 2

In this example, the contour **has one groove**, and the monotonic increasing and decreasing direction of the X-axis **changes once** during the process.





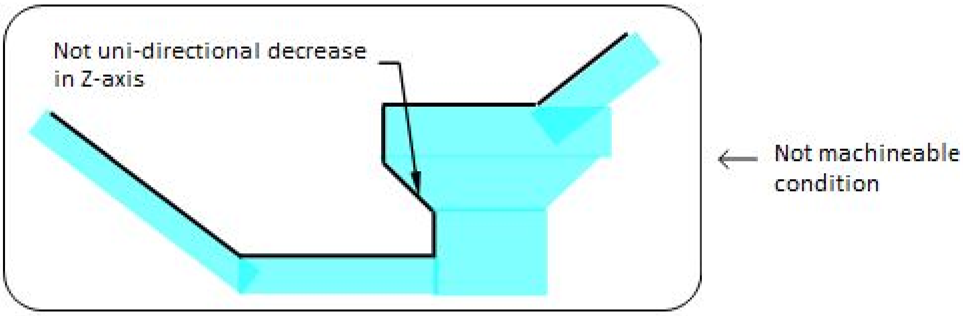
Action Description

1. Before version 10.118.43(inclusive) : X axis only supports setting as a diameter axis, and Z axis only supports setting as a radius axis. After version 10.118.44 (inclusive) : X axis and Z axis support settings are diameter / radius axis.
2. Before the cycle, the tool should rapid position (G00) to the Initial point (A).
3. After executing the G73 command, the tool moves to Reserved point (C) referencing fine turning reserved distance (Δw, Δu/2).
4. The tool then rapid position (G00) in the X axis to the Turning start point (D) and starts feeding to the contour.
    a. Turning start point (D) is set by Pr4014, please see Pr4014 Set Rough Cycle Mode.
    b. The illustration shows that the Pr4014 is set to 0, and Fine turning end point (B) is used as the Tuning start point (D).
5. After retracting in 45° direction until e distance reached in the X-axis, tool rapid position in Z-axis to the next cutting start point.
6. Then tool moves Δd in X-axis and continues to the next repeat cut.
7. Repeat 4. & 5. until the end of the last cut, the tool will cut along the rough contour path one more time.
    a. Rough cutting path = fine cutting path (A' → B) + fine cutting reserve distance (Δw, Δu/2)
8. When the Rough turning is completed, tool will return to the Reserved point (C) from the Turning end point (E).
    a. The retract path (E → C) is set by Pr4014, please see Pr4014 Set Rough Cycle Mode.
    b. The illustration shows when Pr4014 is set to 0, X-axis retracts first, and then Z-axis retracts.
9. Finally, the tool moves from Reserved point (C) to Initial point (A) with G00 XZ two-axis motion.

TYPE II

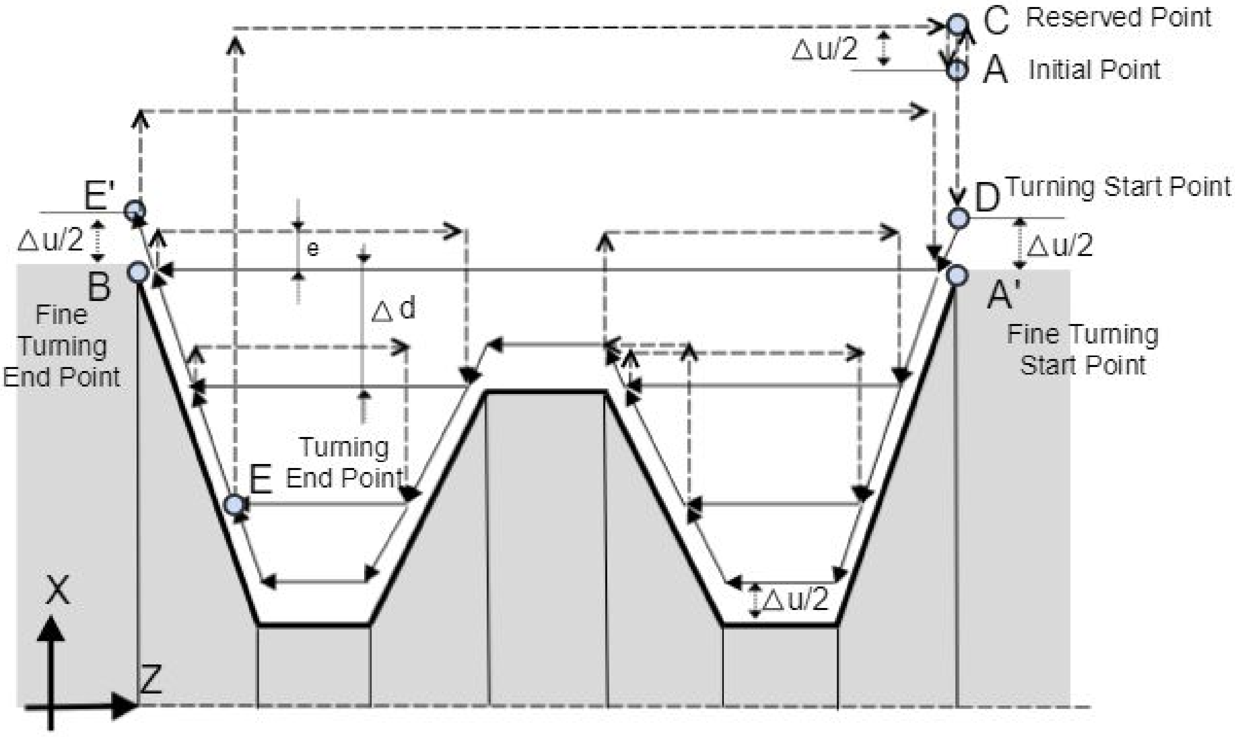
Restriction: Usually used for machining from the middle of the workpiece material. The contour **can have multiple grooves**. Which means in **TYPE II**, only the Z axis must meet the condition of uni-directional increase/decrease.

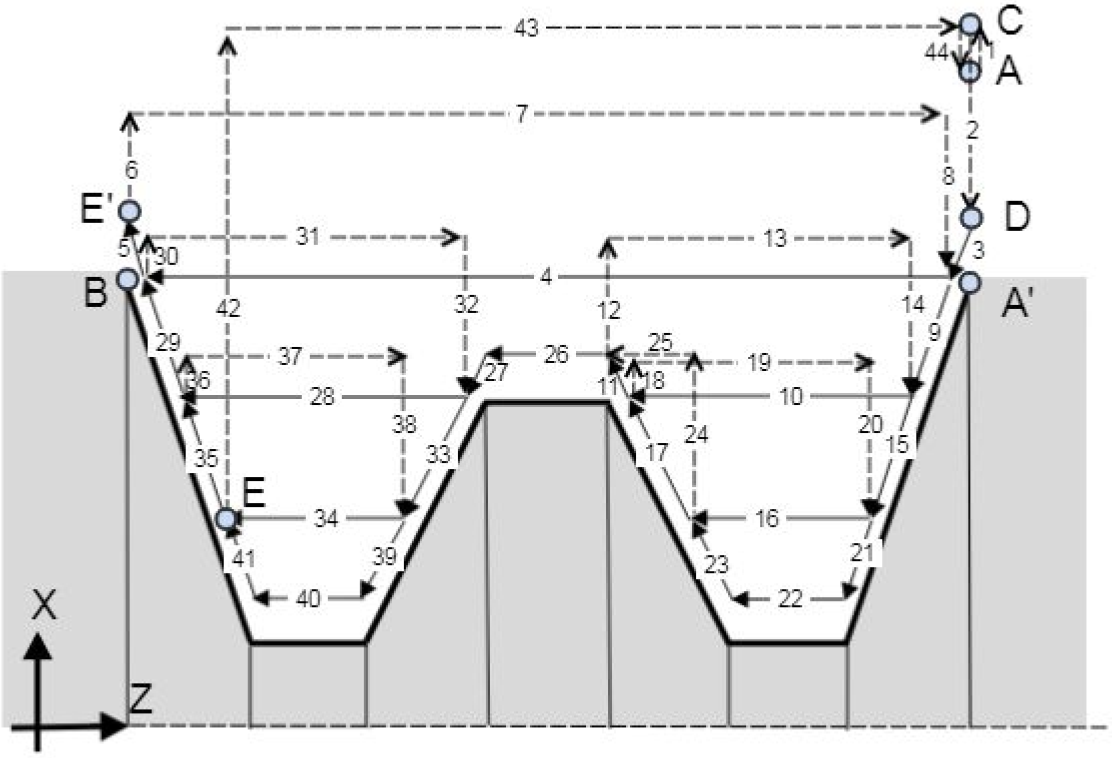
Illustration: Not machinable condition

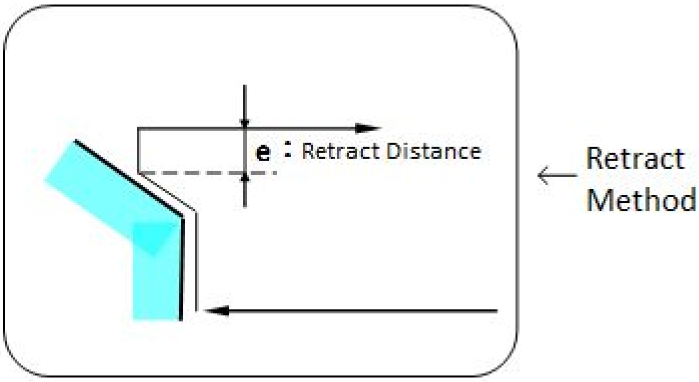

Not uni-directional decrease in Z-axis
Not machineable condition

Example: The following are the paths and diagrams of two types of **TYPE II** fine turning contours.

Illustration: Type II Example



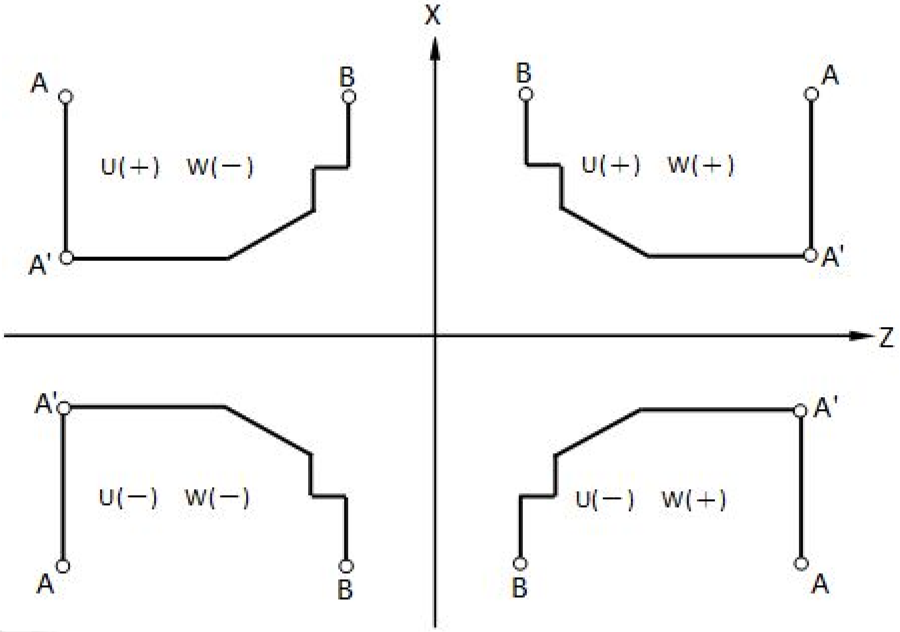




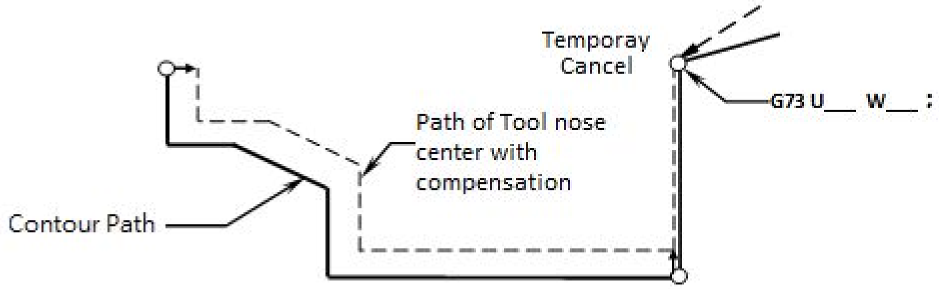
2.42.3 **Precaution**

1. Before the turning cycle starts, the path check will be performed in the following two cases in order to avoid interference.
    a. Case 1: Pr4014 is 0. Case 2: Pr4014 is 1, and TYPE II is executed.
    b. For details, see Pr4014 Set Rough Cycle Mode.
2. From settings above, the path check condition is: If the Initial point (A) is within the Fine Turing path (A' → B), the system triggers [MAR-005 Turning starting point lower than the path]. Check the path to ensure the initial point (A) is outside the Fine turning path (A' → B) when the alarm pops.
3. When given wrong H value, the system triggers [MAR-018 G73/G74 H value input error]. Please use reasonable H value or give no H value for system to judge automatically.

4. When the turning cycle command (G73) does not specify H value and the first block has only the X-axis movement, TYPE I will be used.
5. When [ns] and [nf] are not specified, the U specified in the G73 block is the cutting depth Δd, otherwise is the X-axis reserve.
6. The contour path is described by blocks between ns and nf from point A to point A', and then to point B'.If the Z-axis is not uni-directional increase/decrease, the system triggers [MAR-002 turning path X, the Z axis only allows one direction to increase or decrease].
7. The F, S, and T functions in the blocks between ns and nf are invalid. These commands are valid only in the block of Rough turning cycle (G73).
8. The cutting mode G00/ G01 used for each block between ns and nf will be used as the cutting mode when the tool is roughed along this block.
9. The last block in the path, nf, cannot call subroutine. If all the paths are in a subroutine, the following format can be used (see Example 3 for details).
   Nns M98 Pxxxx; // Subroutine of the machining path
   Nnf U0; // Insert an empty block.
10. The sequence numbers nf must be existing, otherwise the 【COR-010 G66 macro program call level is too deep】 alarm will be issued.
11. Direction of reserve distance for fine turning: The direction of reserve distance for fine turning is determined by the shape as shown below. The program for fine turning is A→A'→B.



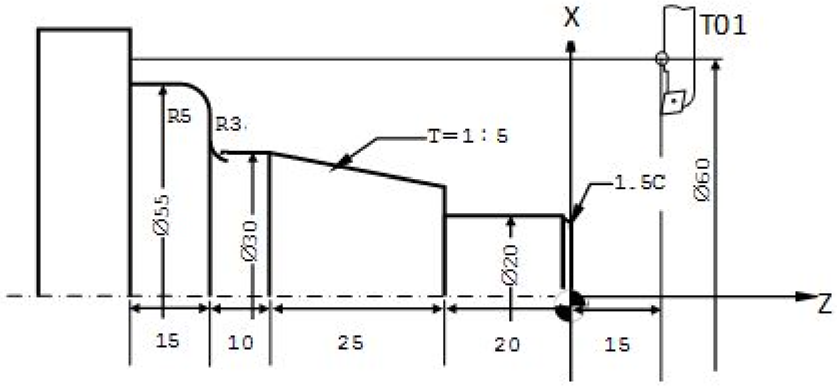
12. The tool radius compensation (G41/G42) needs to be before G73to be valid. The function only compensates the contour path, that is, A → A' → B in the Type I diagram.
13. If a compensation command is included in the G73 command, the compensation will be invalid, but the compensation value will be added to the Reserved distance.
14. Any tool nose compensation command between ns -> nf is invalid. Use the tool nose compensation G41/G42 before G73.
15. G73 Macro identified Initial point (A) point is the position that does not include the tool nose compensation before entering G73. After entering A' point, the compensation will become valid. That is, the tool nose compensation will be temporarily canceled at point A.



2.42.4 **Example**

The following are sample programs for Type I and Type II.
The Type I example program is G73 example 1 in the Manual (tool nose compesnation 2.0, tool nose direction 8)
The Type II sample program is G73 example 2 in the Manual (tool nose compesnation 5.0, tool nose direction 8)
Both sample programs are executed twice, and the results are displayed below each program. One without tool compensation (red) and one with tool compensation (green).

TYPE I



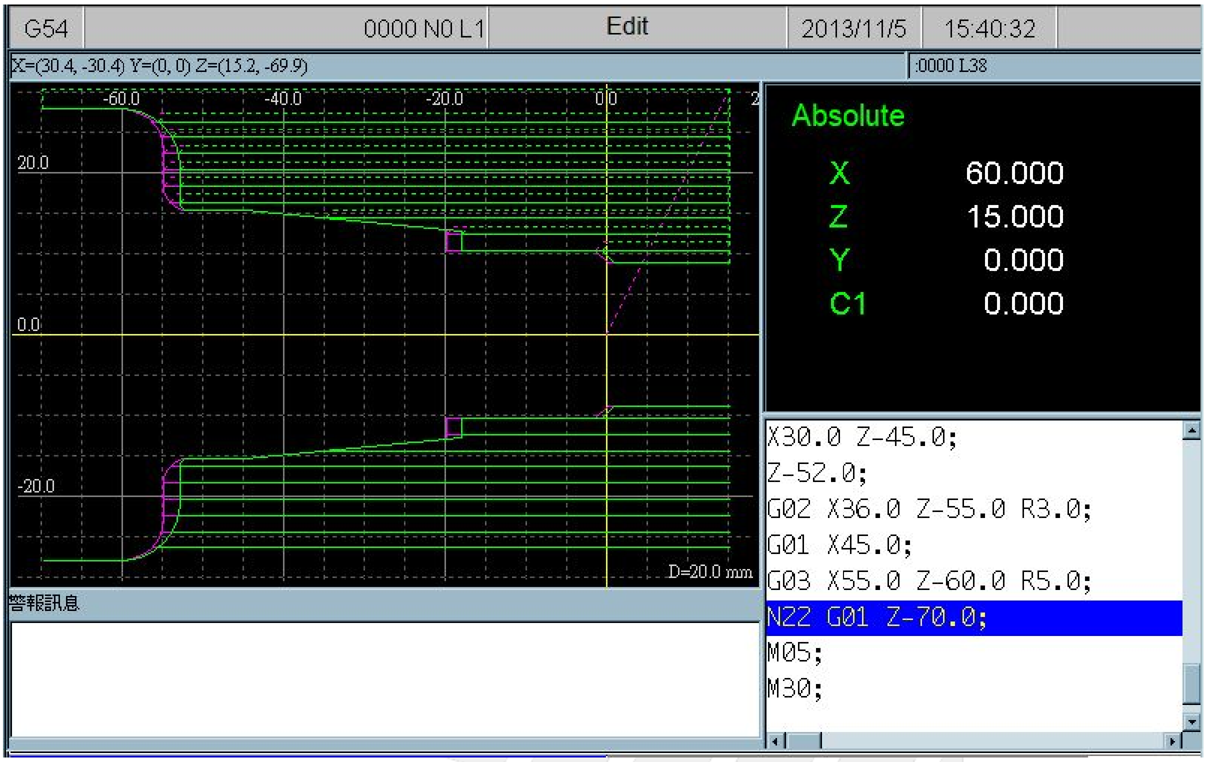
```
% @MACRO
SETDRAW(13);
T01;                //use tool NO. 1
G50 S5000;          //Max. rotate speed 5000 rpm
G96 S130 M03;   //Constant surface speed at 130 m/min, spindle rotate
M08;                //cutting liquid ON
G00 X60.0 Z15.0;  //positioning to start point

G73 U2.0 R1.0 H0;
```

//MACRO syntax must be in C type (=G73), so G73 is changed to G73 in the example.
//cut 2.0 mm in X axis direction, tool returned value 1.0 mm, can have no H.
//Equivalent to G73 U2.0 R1.0
//H value is 0, use TYPE I

```
G73 P01 Q02 U0.8 W0.1 F0.3;

//MACRO syntax must be in C type (=G73), so G73 is changed to G73 in the example.
//Perform radial (end face) rough turning cycle with the block number N01→N02
//The X-axis reserved distance for fine turning is 0.8 mm. The Z axis reserved distance is 0.1mm.
//Feed rate 0.3 mm/rev

N01 G00 X17.0;
//Turning cycle start sequence number

G01 Z0.0;
X20.0 Z-1.5;
Z-20.0;
X25.0;
X30.0 Z-45.0;
Z-52.0;
G02 X36.0 Z-55.0 R3.0;
G01 X45.0;
G03 X55.0 Z-60.0 R5.0;

N02 G01 Z-70.0;
//cutting cycle end sequence number

SETDRAW(0);
G50 S5000;
G96 S130 M03;
G00 X60.0 Z15.0;
```

G42;
// Enable tool compensation before the second cutting

```
SETDRAW(10);
G73 U2.0 R1.0 H0;
G73 P11 Q22 U0.8 W0.1 F0.3;

//MACRO syntax must be in C type (=G73), so G73 is changed to G73 in the example.
N11 G00 X17.0;
G01 Z0.0;
X20.0 Z-1.5;
Z-20.0;
X25.0;
X30.0 Z-45.0;
Z-52.0;
G02 X36.0 Z-55.0 R3.0;
G01 X45.0;
G03 X55.0 Z-60.0 R5.0;
N22 G01 Z-70.0;
G40;    //disable tool compensation
M09;    //cutting liquid OFF
M05;    //spindle stop
M30;    //program end
```



G54 | 0000 N0 L1 | Edit | 2013/11/5 | 15:40:32

X=(30.4, -30.4) Y=(0, 0) Z=(15.2, -69.9)

:0000 L38

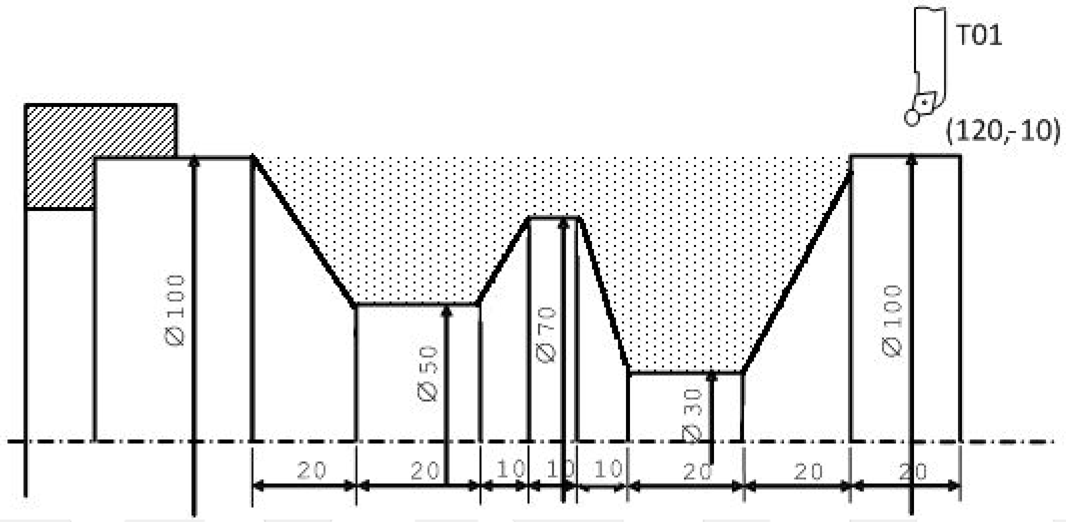
Absolute

X	60.000
Z	15.000
Y	0.000
C1	0.000

```
X30.0 Z-45.0;
Z-52.0;
G02 X36.0 Z-55.0 R3.0;
G01 X45.0;
G03 X55.0 Z-60.0 R5.0;
N22 G01 Z-70.0;
M05;
M30;
```

警報訊息

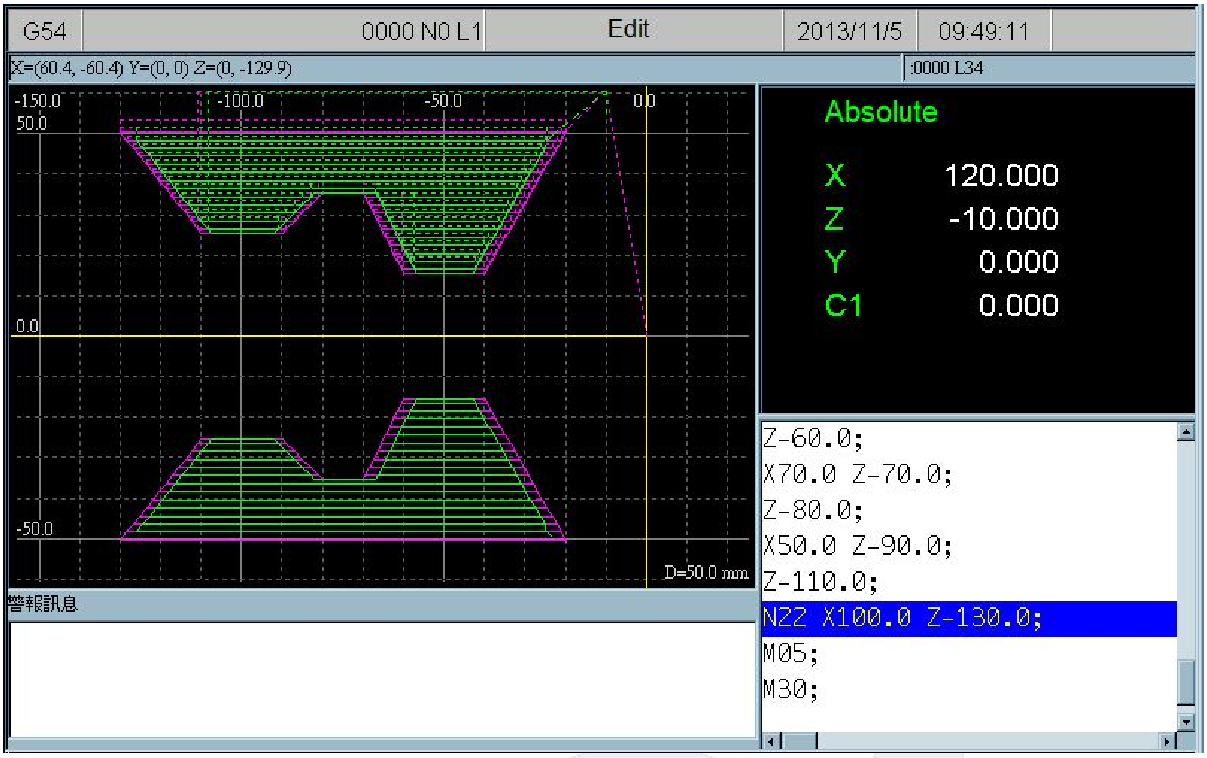
TYPE II



```
% @MACRO
SETDRAW(13);
T01;                    //use tool NO. 1
G50 S5000;              //Max. rotate speed 5000rpm
G96 S130 M03;           //constant surface speed at 130 m/min
M08;                    //cutting liquid ON
G00 X120.0 Z-10.0;      //positioning to start point
```

```
G73 U2.0 R1.0 H1;
//MACRO syntax must be in C type (=G73), so G73 is changed to G73 in the example.
//cut 2.0 mm in X axis direction, tool retracts 1.0 mm, can have no H.
//H value is 0, use TYPE II
//Equivalent to G73 U2.0 R1.0

G73 P01 Q02 U0.8 W0.1 F0.3;
//MACRO syntax must be in C type (=G73), so G73 is changed to G73 in the example.
//perform radial (end face) rough turning cycle with the block number N01→N02
//reserved X-axis distance for fine turning is 0.8 mm. The reserved Z axis distance is 0.1mm.
//feed rate 0.3 mm/rev

N01 G00 X101.0 Z-20.0;
//cutting cycle start block number

G01 X100.0;
X30.0 Z-40.0;
Z-60.0;
X70.0 Z-70.0;
Z-80.0;
X50.0 Z-90.0;
Z-110.0;

N02 X100.0 Z-130.0;
//cutting cycle end blocke number
SETDRAW(0);
G50 S5000;
G96 S130 M03;
G00 X120.0 Z-10.0;

G42;
// Enable tool compensation before the second cutting

SETDRAW(10);
G73 U2.0 R1.0 H1;
G73 P11 Q22 U0.8 W0.1 F0.3;
//MACRO syntax must be in C type (=G73), so G73 is changed to G73 in the example.
N11 G00 X101.0 Z-20.0;
G01 X100.0;
X30.0 Z-40.0;
Z-60.0;
X70.0 Z-70.0;
Z-80.0;
X50.0 Z-90.0;
Z-110.0;
N22 X100.0 Z-130.0;
G40;    //close toolr compensation
M09;    //cutting liquid OFF
```

```
M05;    //spindle stop
M30;    //program end
```

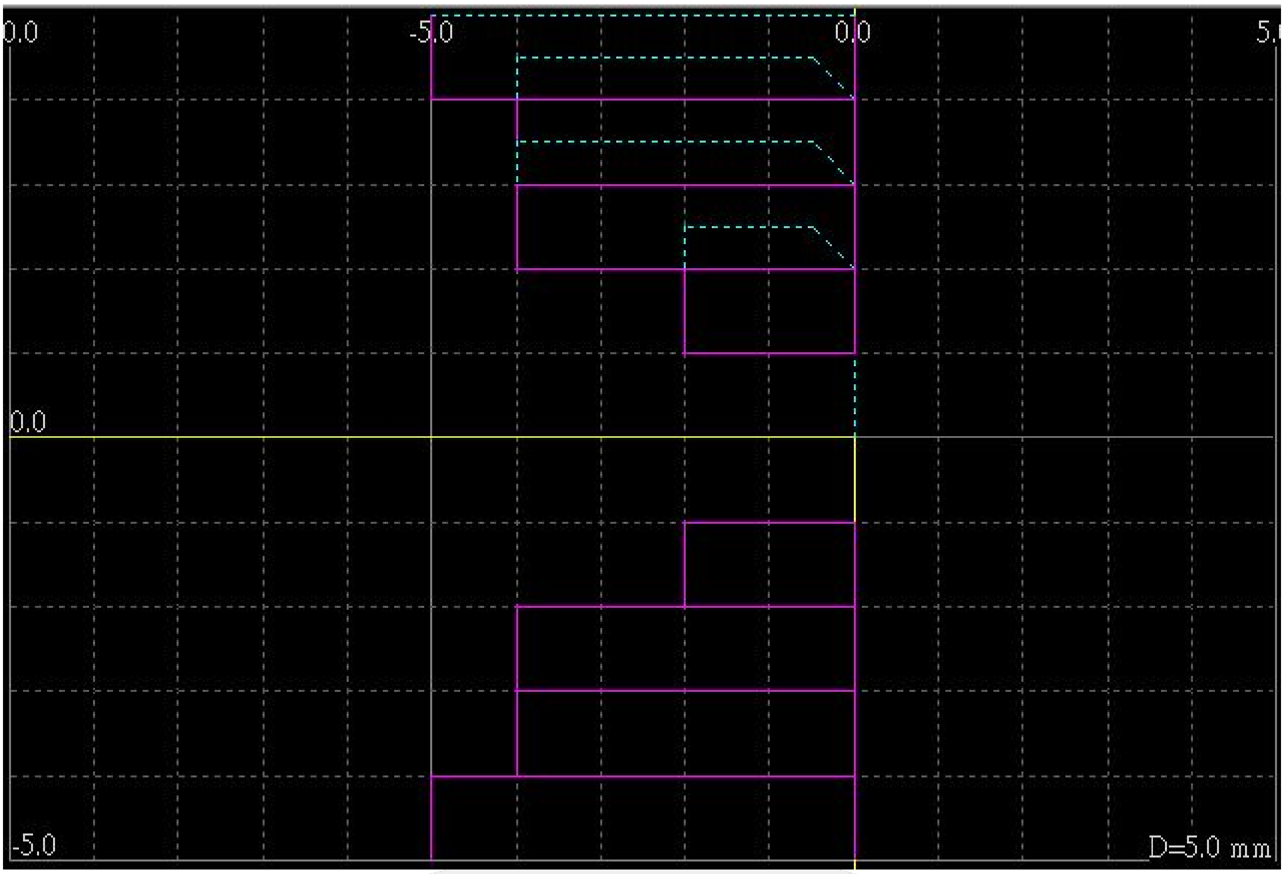


Absolute

X	120.000
Z	-10.000
Y	0.000
C1	0.000

```
Z-60.0;
X70.0 Z-70.0;
Z-80.0;
X50.0 Z-90.0;
Z-110.0;
N22 X100.0 Z-130.0;
M05;
M30;
```

Example 3: G73 calls all paths with a subroutine



Main program	Subroutine O0073

T0202G99
G97M3S60
G50S1200
 	
G0X10.0
G0Z-5.0
 	
// Stock Removal
G73U1.0R0.5
G73P3Q4F5000
 	
// Cutting profile of removal region
N3 M98 P71
N4 U0
 	
M30
	// Cutting profile of removal region
	G00Z-5.0;
	G01X8.0F2000;
	Z-4.0;
	X4.0;
	Z-2.0;
	X2.0;
	Z0.0;
	
	M99;

2.43 G74- Radial (End-Face) Rough Facing Cycle (C-Type)

2.43.1 **Command Form**

G74 W Δd R e H___;
G74 P (ns) Q (nf) UΔu WΔw F___ S___ T___ D___;

Δd: Each cutting depth in the X direction, can be preset by system parameter Pr4013.
e: The retraction distance, can be preset by system parameter Pr4012.
ns: Sequence number of starting block in finishing cycle
nf: Sequence number of ending block in finishing cycle
Δu: The reserved amount of X-axis (outer surface) direction
Δw: The reserved amount of Z-axis (length) direction
F: Feed rate
T: Tool number
S: Spindle RPM setting
H: Machining method: 0 selects TYPE I method; 1 selects TYPE II method. System will judge the processing method by itself of no H sepcified.
D: Chip breaking switch. → Enable function : Set D to 1 / Disable function : Set D to others than 1, effective version: 10.118.70E

2.43.2 **Description**

The G74 command is a radial (end face) rough turning cycle. When workpiece is thick and short, that is, the radial direction cutting is larger than the lateral direction, G74 is suitable.

Notice

This page is mainly for Pr4014=1, outer-face machining circumstances.

Pr4014 is used to set the rough turning mode. For other modes, see Pr4014 setting rough turning cycle mode.

In addition, first introduce the definition of nouns, and match the following illustrations to facilitate specification.

Form: nouns definition table

#	Noun	Symbol	Description
1	Initial point	A	1. One block before the rough turning cycle 2. The rough turning cycle G code (A-Type G71/G74 & C-Type G73/G74)
2	Reserved point	C	1. To reserve a distance for fine turning, this point is offset a distance (Δw, Δu/2) from the initial point (A) 2. If no reserved distance (Δw, Δu/2) is specified, Reserved point (C) equals to Initial point (A).

#	Noun	Symbol	Description
3	Turning start point	D	1. The position starts to execute rough turning cycle.
4	Fine turning contour path	A' → B	1. The path formed by blocks between the start number (ns) and the end number (nf) is the final fine turning contour.
5	Fine turning start point	A'	1. Start position block number (ns).
6	Fine turning end point	B	1. End position block number (nf).
7	Turning end point	E	1. The end point of the rough turning cycle. 2. If no specified reserve distance (Δw, Δu/2) specified, the rough turning end point (E) equals to fine turning end point (B).
8	Retraction path	E → C	1. The path from the Turning end point (E) to the Reserved point (C).

TYPE I

Restriction: Usually used to start machining from the end face, the first block must be uni-directional increase/decrease in Z axis, and each block must be uni-directional increase/decrease in X axis. That is, the next block needs to be increased/decreased from the previous block.

Example: The followings are the path diagrams of two types of **TYPE I** fine turning contours.

Illustration: Type I Example 1

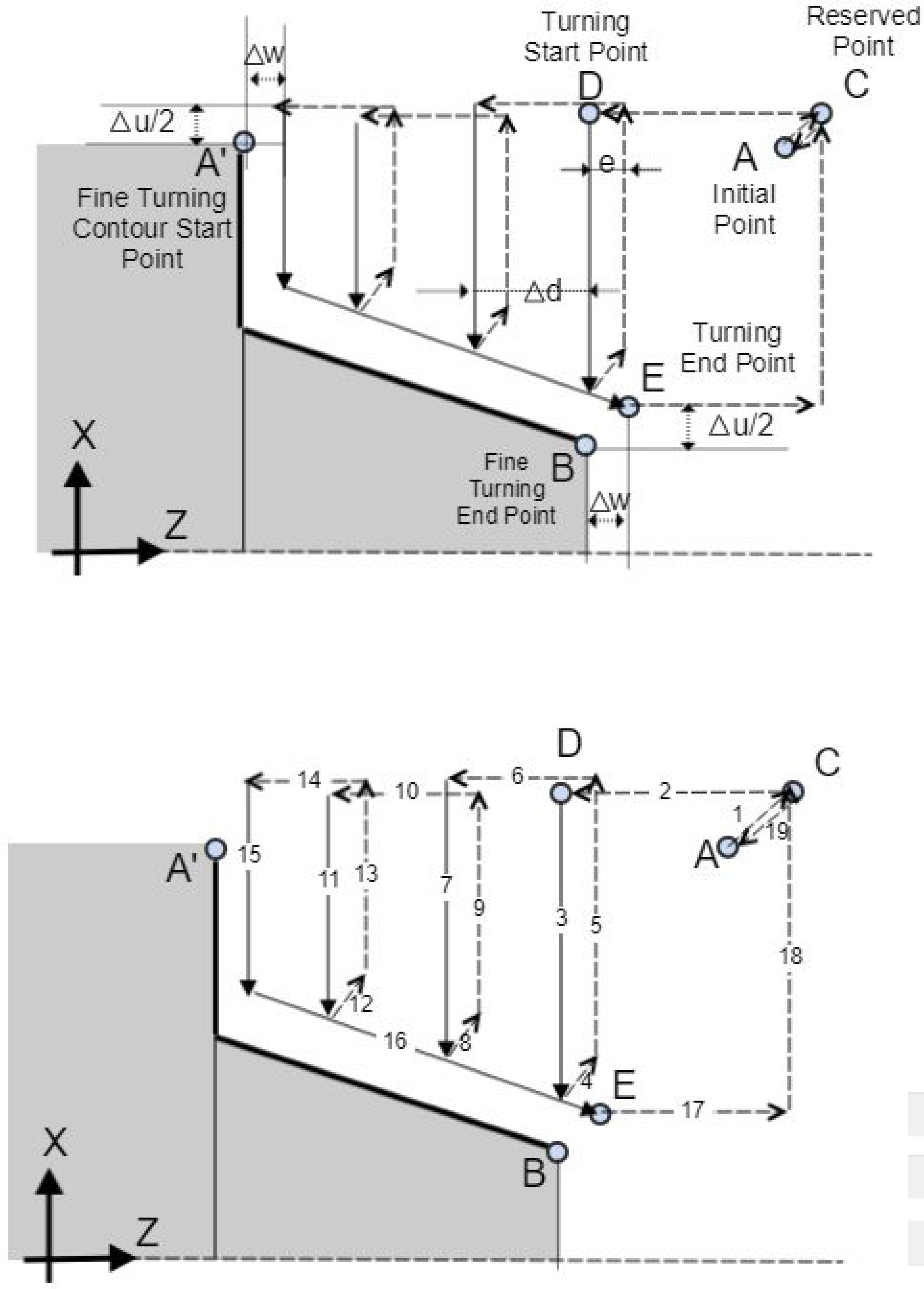
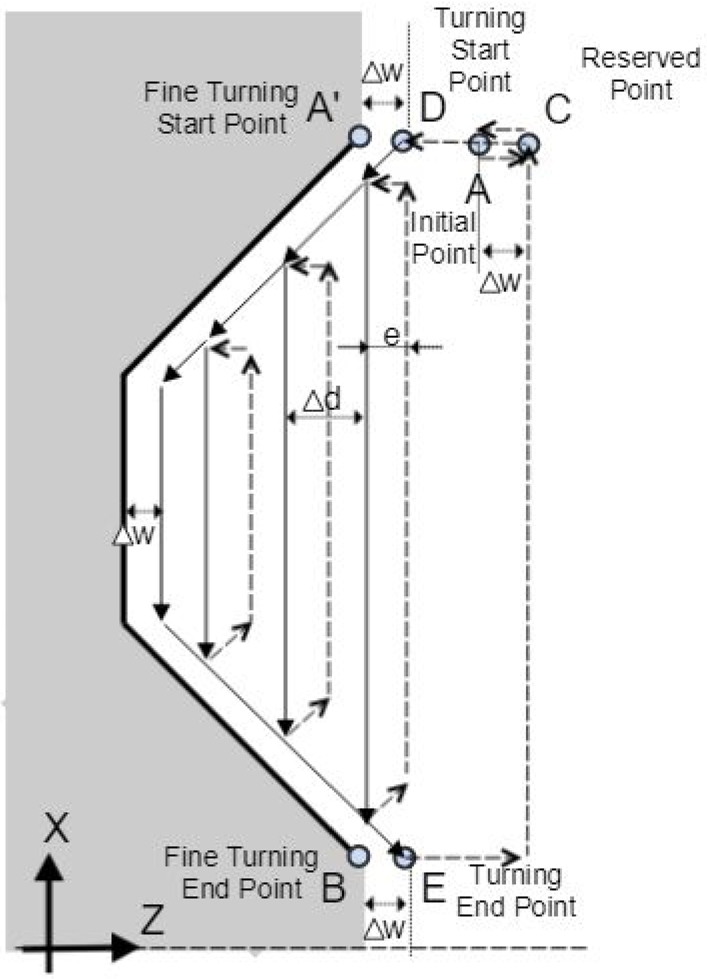
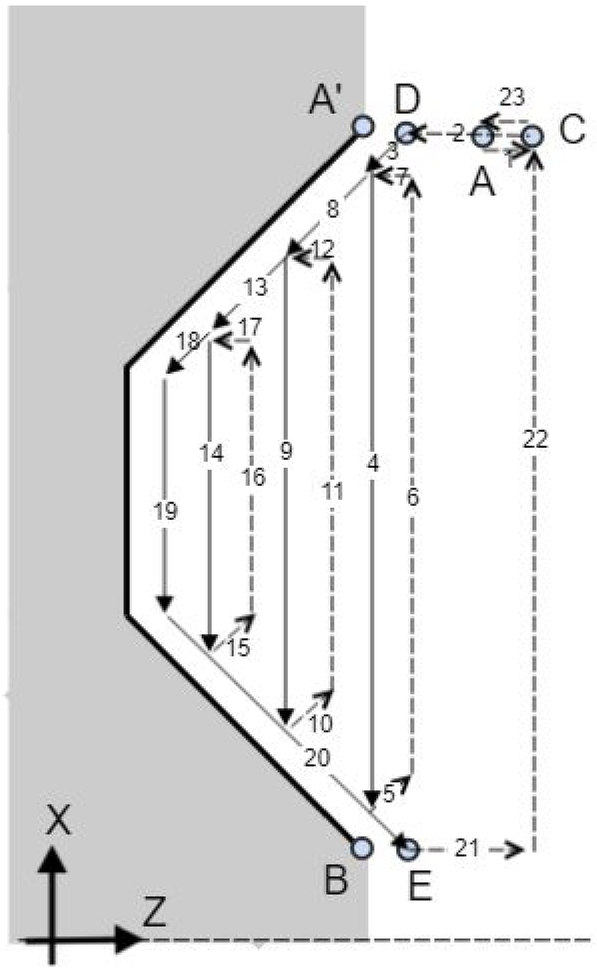


Illustration: Type I Example 2


Turning
Start
Point
Reserved
Point
ΔW
Fine Turning
Start Point
A'
D
C
A
Initial
Point
ΔW
e
Δd
ΔW
X
Fine Turning
End Point
B
E
Turning
End Point
ΔW
Z



Action description

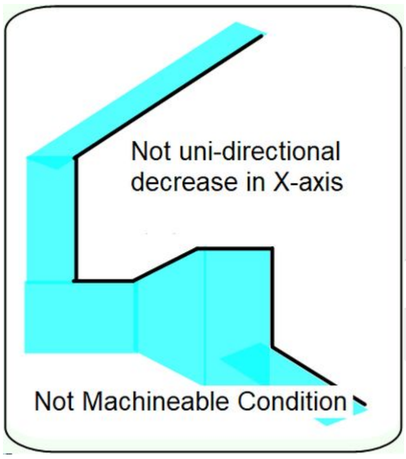
1. Before version 10.118.43(inclusive) : X axis only supports setting as a diameter axis, and Z axis only supports setting as a radius axis. After version 10.118.44 (inclusive) : X axis and Z axis support settings are diameter / radius axis.
2. Before the cycle, the tool should rapid position (G00) to the Initial point (A).
3. After executing the G74 command, the tool moves to Reserved point (C) referencing fine turning reserved distance (Δw, Δu/2).
4. The tool then rapid position (G00) in the Z axis to the Turning start point (D) and starts feeding to the contour.
    a. Turning start point (D) is set by Pr4014, please see Pr4014 Set Rough Cycle Mode.
    b. The illustration shows that the Pr4014 is set to 0, and Fine turning end point (B) is used as the Tuning start point (D).
5. After retracting in 45° direction until e distance reached in the Z-axis, tool rapid position in X-axis to the next cutting start point.
6. Then tool moves Δd in Z-axis and continues to the next repeat cut.
7. Repeat 4. & 5. until the end of the last cut, the tool will cut along the rough contour path one more time.
    a. Rough cutting path = fine cutting path (A' → B) + fine cutting reserve distance (Δw, Δu/2)

a. When the Rough turning is completed, tool will return to the Reserved point (C) from the Turning end point (E).
    i. The retract path (E → C) is set by Pr4014, please see Pr4014 Set Rough Cycle Mode.
    ii. The illustration shows when Pr4014 is set to 0, X-axis retracts first, and then Z-axis retracts.
b. Finally, the tool moves from Reserved point (C) to Initial point (A) with G00 XZ two-axis motion.

TYPE II

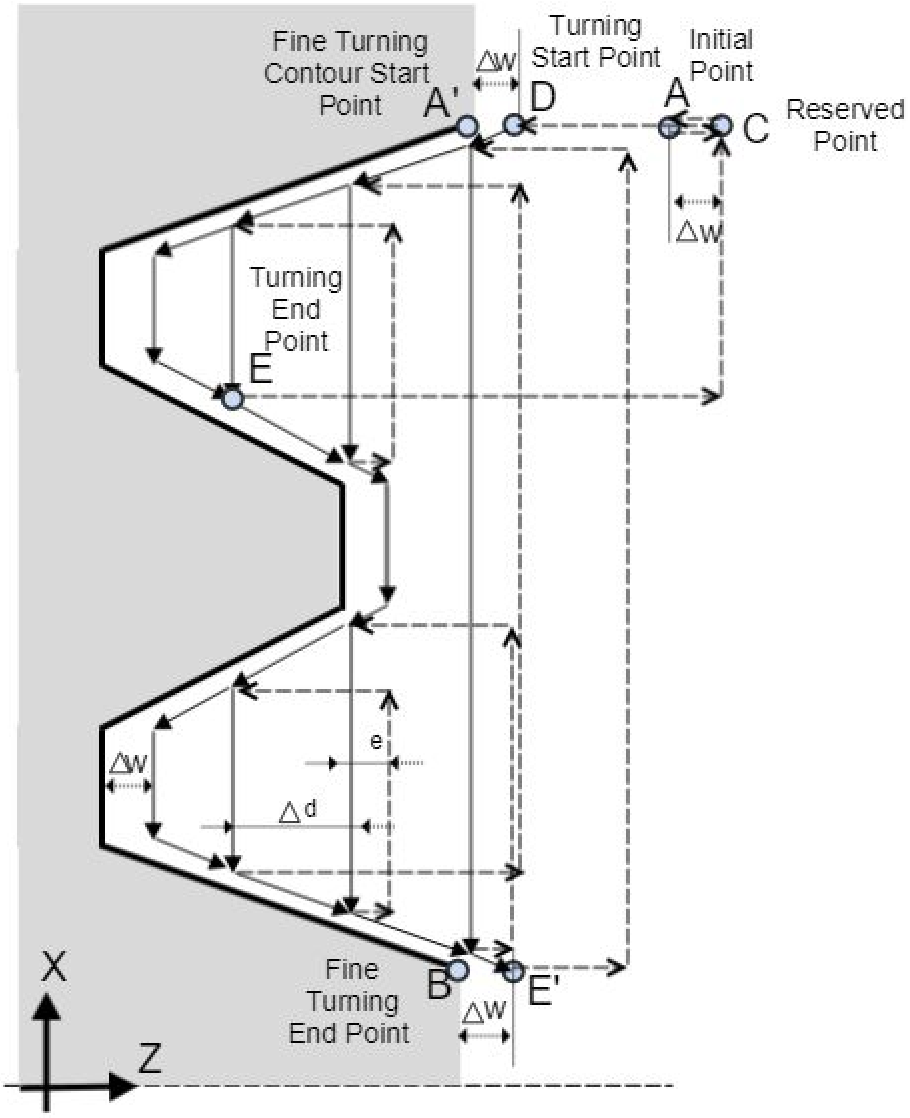
Restriction: Usually used for machining from the middle of the workpiece material. In **TYPE II**, only the X axis must meet the condition of uni-directional increase/decrease.

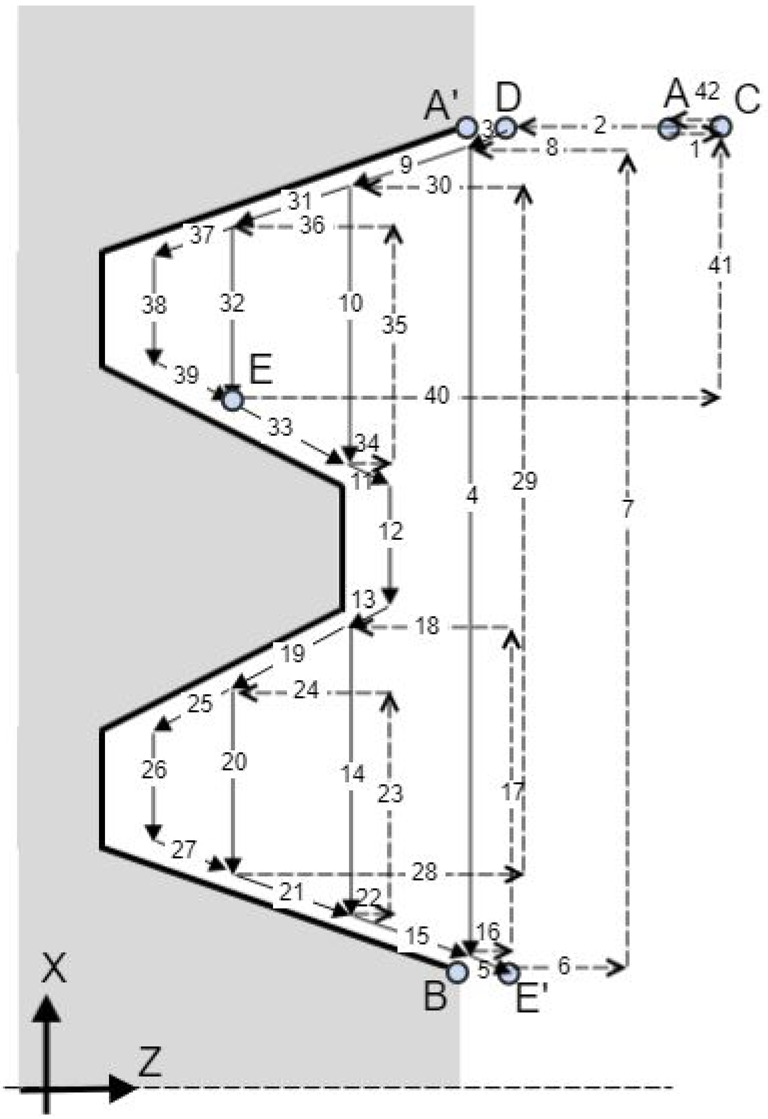
Illustration: Not machinable condition

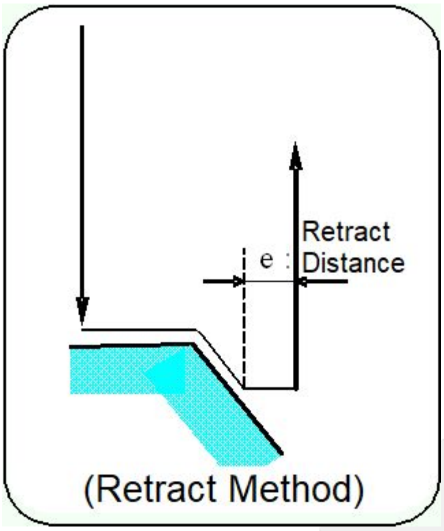


Example: The following are the paths and diagrams of **TYPE II** fine turning contours.

图示：Type II 范例


Fine Turning Contour Start Point
A'
ΔW
Turning Start Point
D
Initial Point
A
C
Reserved Point
ΔW
Turning End Point
E
e
ΔW
Δd
X
Z
Fine Turning End Point
B
ΔW
E'

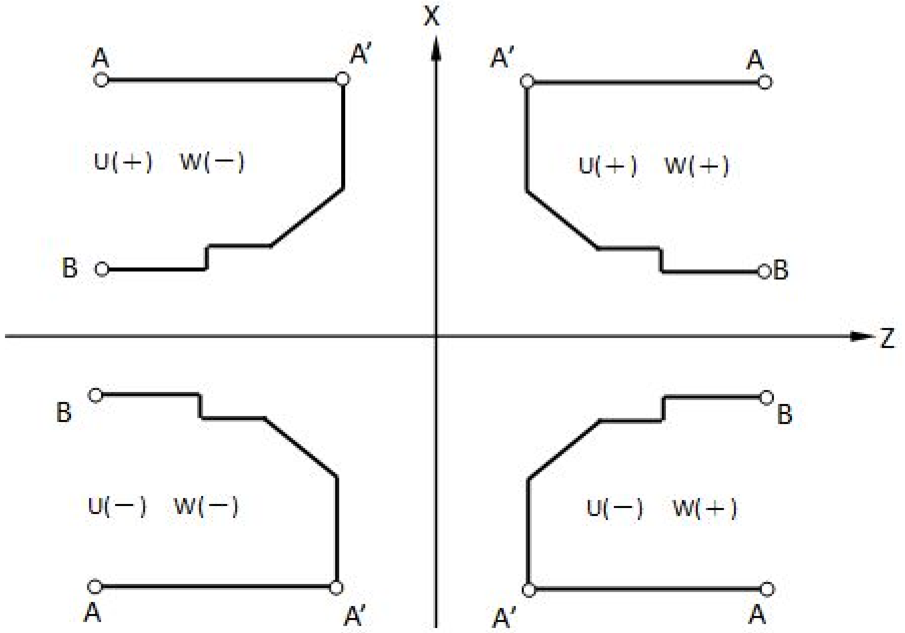

A'
D
3
2
A
42
C
8
1
9
30
31
36
37
41
38
32
10
35
E
39
40
33
34
11
4
29
7
12
13
18
19
24
25
26
20
14
23
17
27
28
21
22
15
16
5
6
B
E'
X
Z



2.43.3 **Precaution**

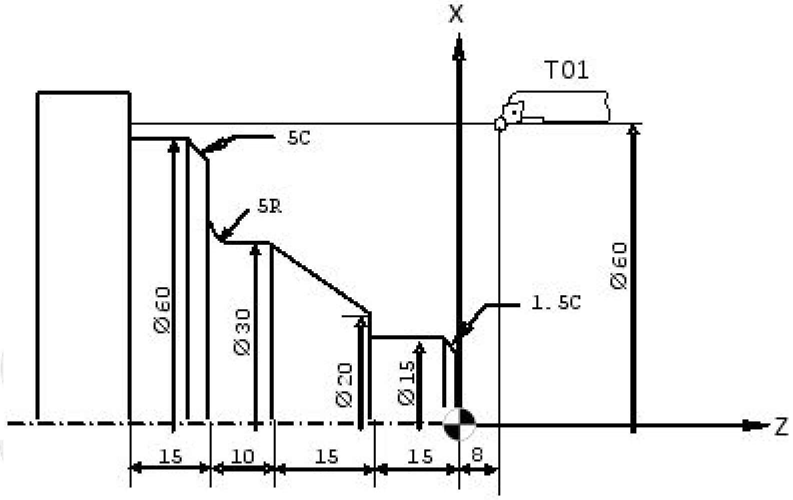
1. Before the turning cycle starts, the path check will be performed in the following two cases in order to avoid interference.
    a. Case 1: Pr4014 is 0. Case 2: Pr4014 is 1, and TYPE II is executed.
    b. For details, see Pr4014 Set Rough Cycle Mode.
2. From settings above, the path check condition is: If Z position of Initial point (A) is within the Fine Turing path (A' → B), the system triggers [MAR-005 Turning starting point lower than the path]. Check the path to ensure the initial point (A) is outside the Fine turning path (A' → B) when the alarm pops.
3. When [ns] and [nf] are not specified, the W specified in the G74 block is the cutting depth Δd, otherwise is the Z-axis reserve.
4. The contour path is described by blocks between ns and nf from point A to point A', and then to point B'. If the X-axis is not uni-directional increase/decrease, the system triggers [MAR-002 turning path X, the Z axis only allows one direction to increase or decrease].
   If the path does not move Z axis first, then system triggers [MAR-003 the first block has no net movement on Z axis].
5. The F, S, and T functions in the blocks between ns and nf are invalid. These commands are valid only in the block of Rough turning cycle (G74).
6. When given wrong H value, the system triggers [MAR-018 G73/G74 H value input error].
7. The cutting mode G00/ G01 used for each block between ns and nf will be used as the cutting mode when the tool is roughed along this block.
8. The last block in the path, nf, cannot call subroutine. If all the paths are in a subroutine, the following format can be used (see Example 3 for details).
   Nns M98 Pxxxx; // Subroutine of the machining path
   Nnf U0; // Insert an empty block.
9. The sequence numbers nf must be existing, otherwise the 【COR-010 G66 macro program call level is too deep】 alarm will be issued.
10. Tool nose compensation (G41/G42) is not supported. Any tool nose compensation in a block with G74 command will be invalid.

11. Direction of reserve distance for fine turning: The direction of reserve distance for fine turning is determined by the shape as shown below. The program for fine turning is A→A'→B.



2.43.4 **Example**

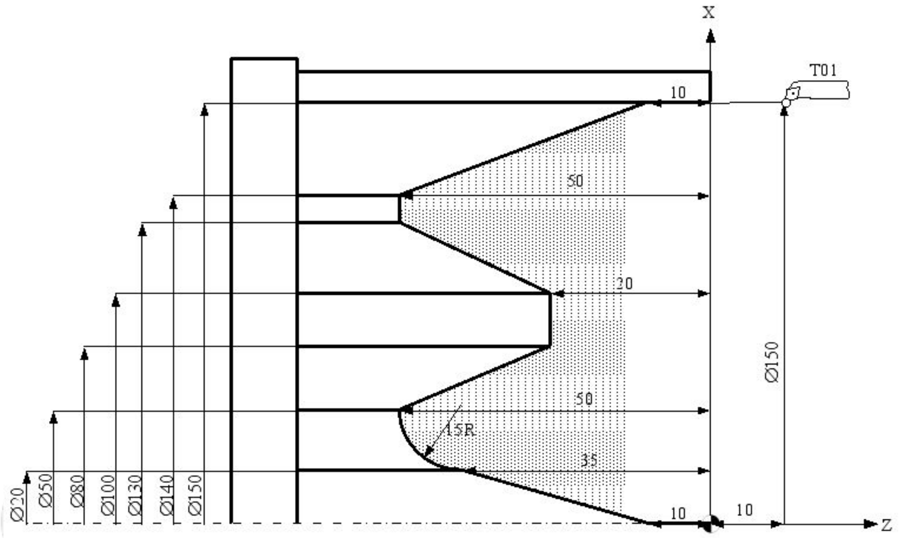
TYPE I



```
T01; //use tool NO. 1
G50 S5000; //Max. rotate speed 5000 rpm
G96 S130 M03; //constant surface speed, surface speed 130 m/min, spindle rotate CW
G00 X60.0 Z8.0;//positioning to start point
M08; //cutting liquid ON
G74 W3.0 R1.0 H0;//cut 3.0 mm in X axis direction, tool retract 1.0 mm
```

```
//can have no H, equivalent to G74 U3.0 R1.0

G74 P01 Q02 U0.8 W0.2 F0.6;
//perform radial (end face) rough turning cycle with the block number N01→N02
//the reserved X-axis distance for fine turning is 0.8 mm. The reserved Z axis distance 0.2mm.
//feed rate 0.6 mm/rev
N01G00 Z-55.0;//the contour to be cut
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;
N02X12.0 Z0.0;
M09; //cutting liquid OFF
G28 X60.0 Z10.0;//tool moves to specified mid-point
M05; //spindle stop
M30; //program end
//note: When H is 1, it will be cut in TYPE II mode.
```

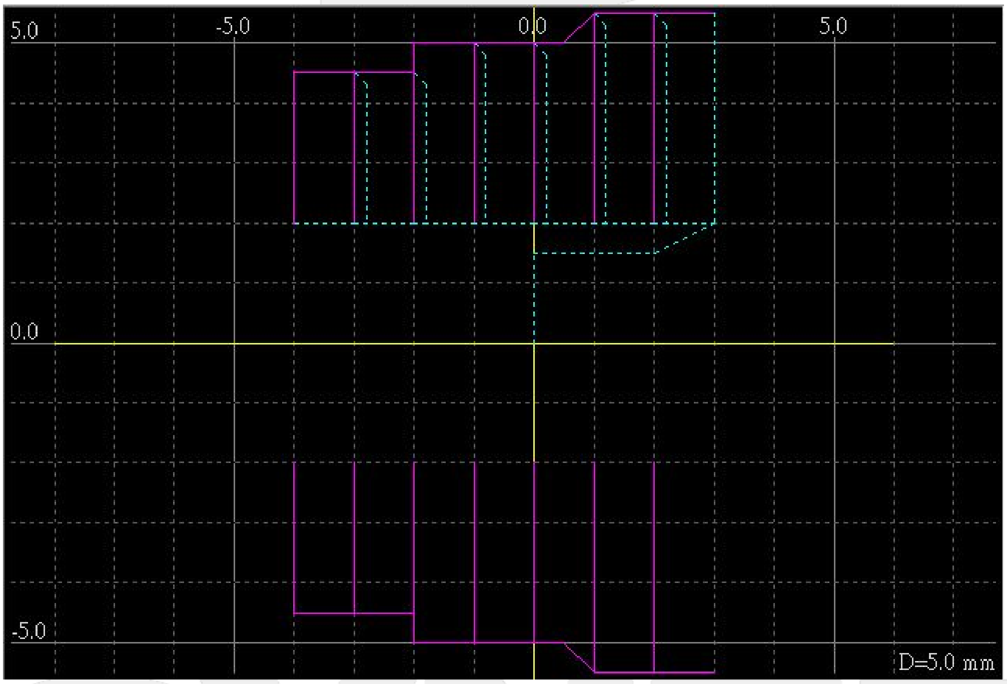
TYPE II



```
T01; //use tool NO. 1
G50 S5000; //Max. rotate speed 5000 rpm
G96 S130 M03; //constant surface speed, surface speed 130 m/min, spindle rotate CW
M08; //cutting liquid ON
G00 X150.0 Z10.0;//positioning to start point
G74 W2.0 R1.0 H1; //cut 2.0 mm in Z axis direction, tool returned value 1.0 mm
//can have no H, equivalent to G74 U3.0 R1.0
G74 P01 Q02 U0.8 W0.1 F0.6;
//perform radial (end face) rough turning cycle with the block number N01→N02

//the reserved X-axis precision car is 0.8 mm. The reserved Z axis precision car is 0.1mm.
```

```
//feed rate 0.6 mm/rev
N01G00 X150.0 Z0.0; //the contour to be cut
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;
N02X0.0 Z0.0;
M09; //cutting liquid OFF
M05; //spindle stop
M30; //Program end
```

Example 3: G74 calls the cutting profile with a subroutine



Main program	Subroutine O0074

T0202G99
G97M3S60
G50S1200
G0X3.0
G0Z2.0
// Stock Removal
G74W1R0.2H0
G74P3Q4 U1 W1 F5000
 	
// Cutting profile of removal region
N3 M98 P72
N4 U0
 	
M30
	// Cutting profile of removal region
	G00Z-5.0;
	G01X8.0F2000;
	Z-3.0;
	X9.0;
	Z-0.5;
	X10.0W0.5;
	M99;

2.44 G75- Contour Rough Turning Cycle (C-Type)

Command Form

G75 UΔi WΔk R d ;
G75 P (ns) Q (nf) UΔu WΔw F___ S___ T___;

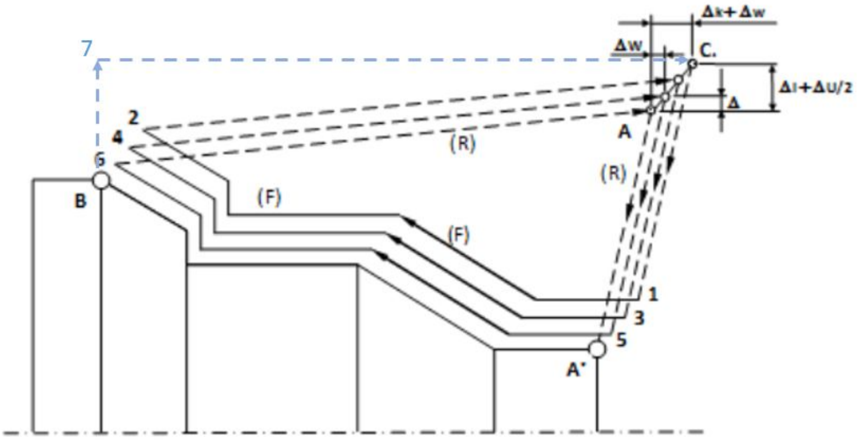
Δi: depth of each cut in X direction (radial), it can be specified by the Pr4015.
ΔK: depth of each cut in Z direction(lateral), it can be specified by the Pr4016.
d: The number of cuts in the cycle, can be contourset by Pr4017.
ns: block number of the cycle start
nf: block number of the cycle end
Δu: reserved distance for fine turning in X direction (radial)
Δw: reserved distance for fine turning in Z direction (lateral)
F: feedrate **T**: tool number
S: spindle rotate speed

2.44.1 Description

1. The G75 command is a contour rough contour turning cycle, suitable for cast or forged workpiece which is only slightly larger than the fine turning contour. Using G71 and G70 commands will cause many unnecessary cutting paths resulting in wasted time. Therefore, G75 contour rough turning cycle can cut with a required number of times along the existing contour of the workpiece, with appropriate distance and depth in each cut.
2. **The last block in the path, nf, cannot call subroutine. If all the paths are in a subroutine, the following format can be used (see Example 2 for details).**
   Nns M98 Pxxxx; // Subroutine of the machining path
   Nnf U0; // Insert an empty block.
3. The sequence numbers nf must be existing, otherwise the 【COR-010 G66 macro program call level is too deep】 alarm will be issued.
4. Support tool compensation G41/G42, but the tool compensation command needs to be under G75 or the tool compensation command is invalid (see example 3 for details)

5. Before version 10.118.43(inclusive) : X axis only supports setting as a diameter axis, and Z axis only supports setting as a radius axis. After version 10.118.44 (inclusive) : X axis and Z axis support settings are diameter / radius axis.

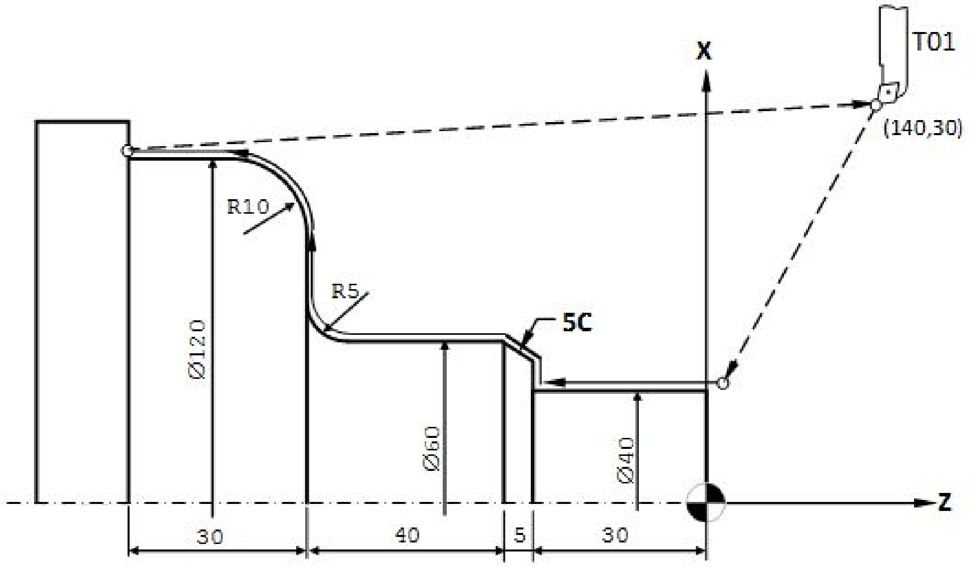
Action Description



1. `Position the tool at point A (starting point) before cycle start`
2. After executing the G75 command, the tool will offset a reserved distance for fine turning (X axis is ΔU/2, Z axis is ΔW) plus the cutting amount (X axis is Δi, Z axis is ΔW). Offset to point C;
3. The tool will cut along path A->A'->B, and complete the cycle according to the cut depth and the number of cuts;
4. At the end of the last cycle, the tool will automatically return to point A for the next cycle turning.
5. When rough turning cycle completes, the tool will first return to the point C from the end point of the contour (retract X first and then Z), then return to point A from point C (return in both XZ directions)

2.44.2 **Example**

Example 1

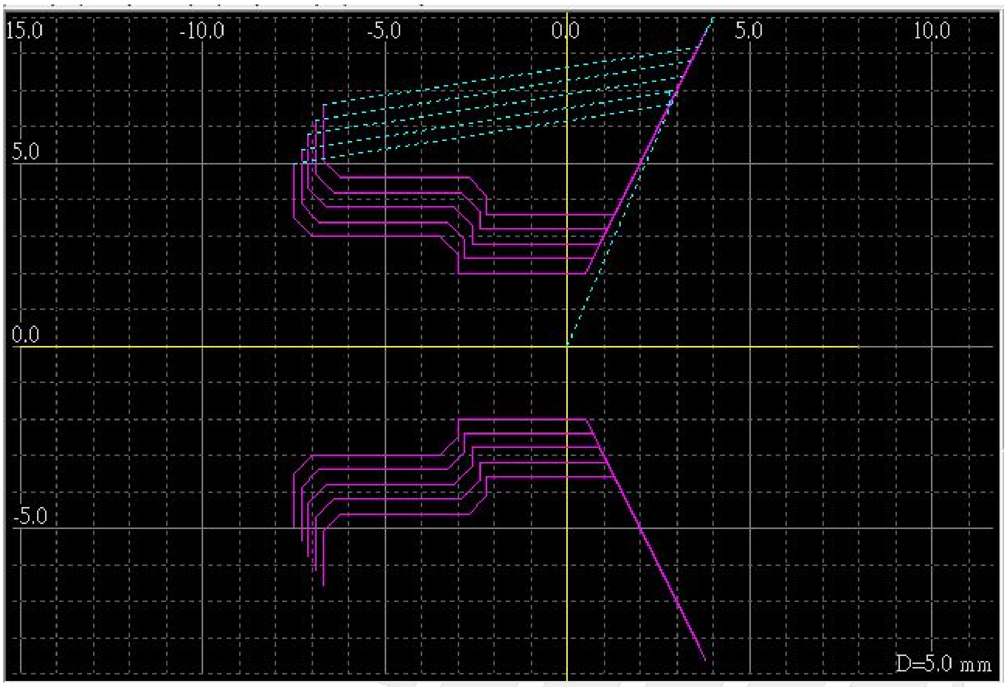


T01; //use tool NO.1
G50 S5000; //max. rotate speed 5000 rpm
G96 S130 M03; //constant surface speed, surface speed 130 m/min, spindle rotate CW
G00 X140.0 Z30.0; //position to start point
M08; //cutting liquid ON

G75 U15.0 W3.0 R3.0;
//depth of cutting in X direction is 15.0 mm, depth of cutting in Z direction is 3.0 mm, cut for three times

G75 P01 Q02 U0.8 W0.2 F0.3;
//perform radial (end face) rough turning cycle with the block number N01→N02
//The reserved X-axis precision car is 0.8 mm. The reserved Z distance for fine turning is 0.2mm feedrate 0.3 mm/rev
N01G00 X40.0 Z5.0;//the contour to be cut
G01 Z-30.0;
X50.0;
X60.0 Z-35.0;
Z-70.0;
G02 X70.0 Z-75.0 R5.0;
G01 X100.0 ;
G03 X120.0 Z-85.0 R10.0;
N02G01 Z-105.0;
M09; //cutting liquid OFF
G28 X140.0 Z30.0; //tool position to specified mid-point then return to machine zero
M05; //spindle stop
M30; //program end

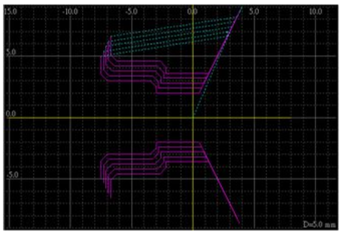
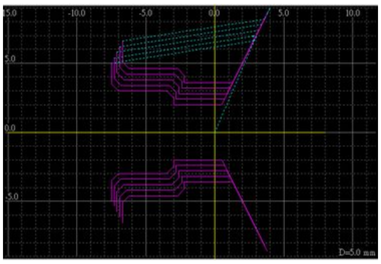
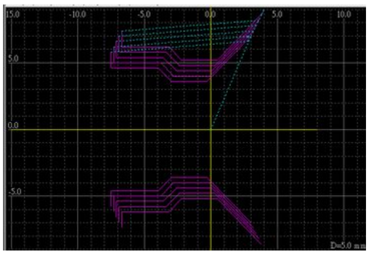
Example 2： G75 calls the cutting profile with a subroutine



Main program	Subroutine O0075

T0202 G99
G97 M3 S60
G50 S1200
G0 X14.0 Z3.0
// Stock Removal
G75 U2.0 W1.0 R5
G75 P1 Q2
// Cutting profile
N01 M98 P73
N02 U0
M30
	// Cutting profile
	G01 X4.0 Z0.5 ;
	Z-3.0 ;
	X5.0 ;
	X6.0 Z-3.50 ;
	Z-7.0 ;
	X7.0 Z-7.50 R0.5 ;
	X10.0;

Example 3： G75 with tool compensation

	Without tool compensation	G42 in G75 contour (invalid tool compensation)	G42 before G75(valid tool compensation)
Simulation Path	 15.0 -10.0 -5.0 0.0 5.0 10.0 5.0 0.0 -5.0 D=5.0 mm	 15.0 -10.0 -5.0 0.0 5.0 10.0 5.0 0.0 -5.0 D=5.0 mm	 15.0 -10.0 -5.0 0.0 5.0 10.0 5.0 0.0 -5.0 D=5.0 mm

Main Program	T0202 G99
	G97 M3 S60
	G50 S1200
	G0 X14.0 Z3.0
	
	// Stock Removal
	G75 U2.0 W1.0 R5
	G75 P1 Q2
	
	// Cutting profile
	N01 G01 X4.0 Z0.5 ;
	Z-3.0 ;
	X5.0 ;
	X6.0 Z-3.50 ;
	Z-7.0 ;
	X7.0 Z-7.50 R0.5 ;
	N02 X10.0;
	
	M30
		T0202 G99
		G97 M3 S60
		G50 S1200
		G0 X14.0 Z3.0
		
		// Stock Removal
		G75 U2.0 W1.0 R5
		G75 P1 Q2
		
		// Cutting profile
		//tool compensation in G75 contour blocks
		N01 **G42** G01 X4.0 Z0.5 ;
		Z-3.0 ;
		X5.0 ;
		X6.0 Z-3.50 ;
		Z-7.0 ;
		X7.0 Z-7.50 R0.5 ;
		N02 X10.0;
		
		//Disable tool compensation
		G40
		M30
			T0202 G99
			G97 M3 S60
			G50 S1200
			G0 X14.0 Z3.0
			
			// Stock Removal
			G42 // tool comp. before G75, tool nose radius 0.5, direction 0
			G75 U2.0 W1.0 R5
			G75 P1 Q2
			
			// Cutting profile
			N01 G01 X4.0 Z0.5 ;
			Z-3.0 ;
			X5.0 ;
			X6.0 Z-3.50 ;
			Z-7.0 ;
			X7.0 Z-7.50 R0.5 ;
			N02 X10.0;
			
			//Disable tool compensation
			G40
			M30

2.45 G76- End Face (Z-Axis) Peck Drilling Cycle (C-Type)

2.45.1 **Command Form**

G76 R e ;
G76 X(U) Z(W) PΔi QΔk R Δd F ;

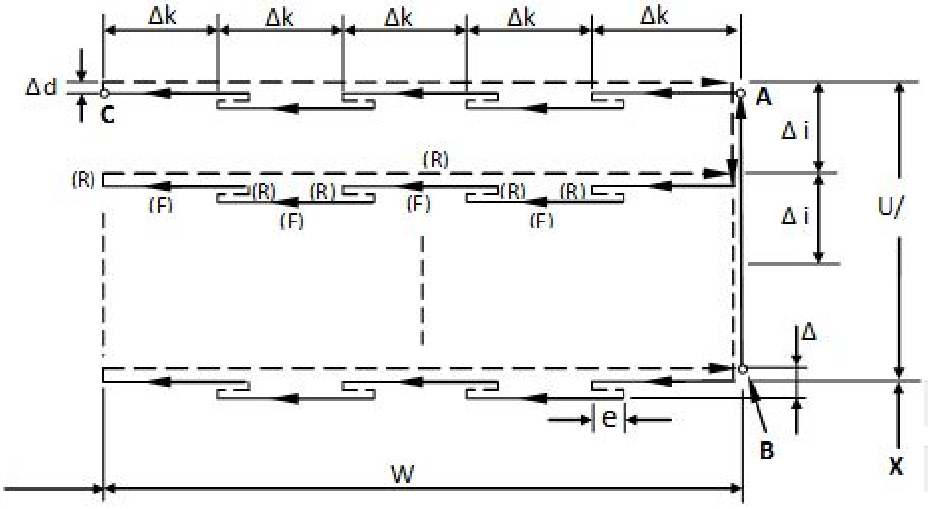
e: Retract distance (retract in Z cutting Δk) <- can be set by in Z cutting Δk) <- can be set by Pr4011
X: X position of B point (diameter positioning)
Z: Z position of C point
U: Incremental value (diameter) from point A to point B
W: Incremental value from point A to point C
Δi: The distance of each X axis cut (in a positive, radius value)
Δk: Cutting depth of each Z axis cut (positive value)
Δd: The retract distance of X axis when cutting to the end point (this value is zero if retract along original path)
F: Feedrate

Description

The G76 command is an end face (Z-axis) peck drilling for cutting grooves or holes on end face;. When executing, every cut in Z-axis with distance Δk has a retract distance e. The command can not only be used for end-face grooving and perimeter intermittent cutting, but also for deep hole drilling.

X axis only supports setting as diameter axis, and Z axis only supports setting as radius axis.

Action description



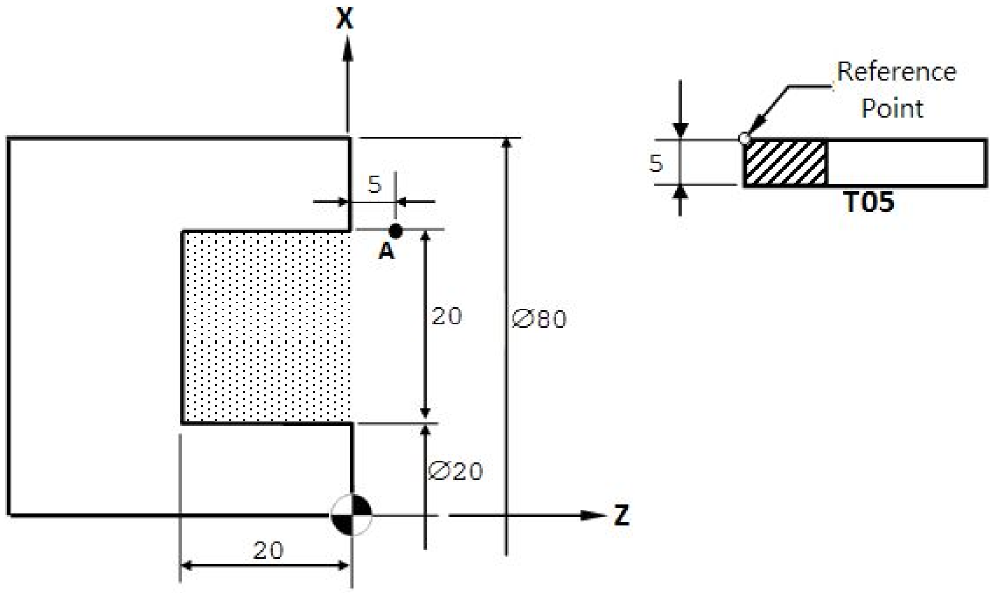
1. Positioning tool to point A (start point) before cycle.
2. When executing G76, the tool will start peck style cutting from point A to point C. Each feed Δk followed by a retract distance e. At point C, retract distance Δd in X axis and then quickly move to the start point.
3. Then the tool moves in X direction by Δi distance and continues the same cycle until reaches the machining end point B. Then tool will automatically return from point B to point A and wait for the next cycle.

2.45.2 **Precaution**

1. When e and Δd are assigned by R argument with X or Z, R is the X-axis retract distance. R is usually positive, but If the starting and end point (point A to point B) are the same, then R should use sign to indicate retract direction.
2. If G76 is followed by the R argument without X or Z, this indicates the retract distance in Z direction. This is a modal G code which will keep enabled until the program end.
3. If no QΔk is set, the peck movement is cancelled and tool goes to Z axis endpoint directly.
4. If the starting and end point (point A to point B) are different but RΔd is negative, then triggers alarm [MAR28 peck type turning retract direction conflict].
5. If there is RΔd, then the tool will retract with Δd at the end of each peck cycle including the first peck cycle. If the first peck cycle has a risk of interference in X, add two lines of program can avoid this. As shown in Example 2.

Example

Example 1:

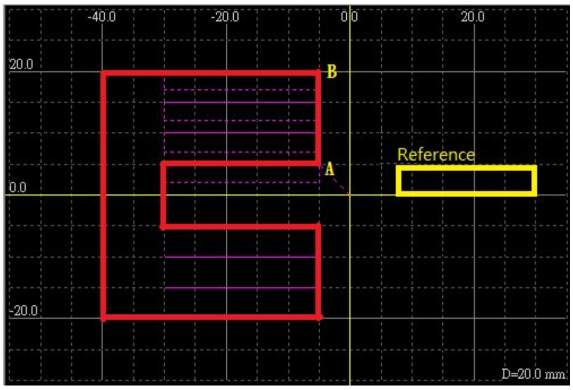
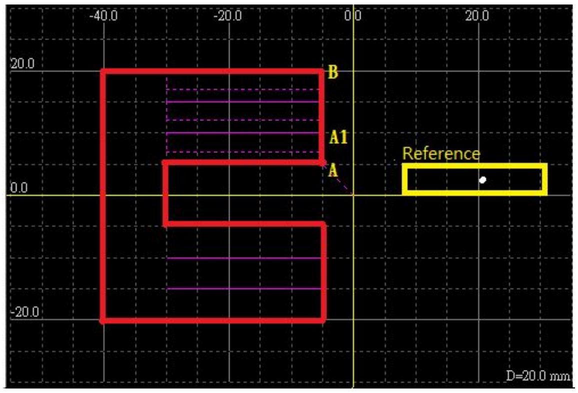


T05; //use tool NO. 5
G50 S1000; //max rotate speed 1000 rpm
G96 S100 M03; //constant surface speed at 100 m/min, spindle rotate CW
M08; // open cutting fluid
G00 X60.0 Z5.0;// positioning to point A
G76 R1.0;
G76 X20.0 Z-20.0 P5.0 Q8.0 F0.1;
//start end face peck drilling cycle, depth of cutting in Z 8.0 mm, and retract 1.0 mm.
//X axis moves 5.0 mm after each peck cycle, and feedrate is 0.1 mm/rev
M09; //cutting liquid liquid
G28 X100.0 Z30.0;//positioning to specified mid-point, then return to machine zero
M05; //spindle stops
M30; //program ends

Example 2： R_ Retraction

	Retract X in First Peck Cycle	No Retract X in First Peck Cycle

Example Program	G00 X10.0 Z-5.0; //positioning to point A
	G76 R1.0;
	G76 X40.0 Z-30.0 P5.0 Q8.0 R3. F0.1;
	//Cycle start Z cut depth is 8.0 mm and retract 1.0 mm. After one cycle, the X retract 3.0mm and moves 5.0 mm to next peck cycle,feed rate 0.1mm/rev.
		G00 X10.0 Z-5.0; //positioning to point A
		G76 R1.0;
		G76 X10.0 Z-30.0 P5.0 Q8.0 F0.1;
		//Cycle start Z cut depth is 8.0 mm and retract 1.0 mm. After one cycle, the X axis returns to point A.
		G00 X20.0 Z-5.0; //positioning to point A1
		G76 R1.0;
		G76 X40.0 Z-30.0 P5.0 Q8.0 R3. F0.1;
		//Cycle start Z cut depth is 8.0 mm and retract 1.0 mm. After one cycle, the X retract 3.0mm and moves 5.0 mm to next peck cycle,feed rate 0.1mm/rev.

Path Diagram	 -40.0 -20.0 0.0 20.0 20.0 0.0 -20.0 B A Reference D=20.0 mm	 -40.0 -20.0 0.0 20.0 20.0 0.0 -20.0 B A1 A Reference D=20.0 mm

2.46 G77- Lateral (X-Axis) Peck Grooving Cycle (C-Type)

2.46.1 **Command Form**

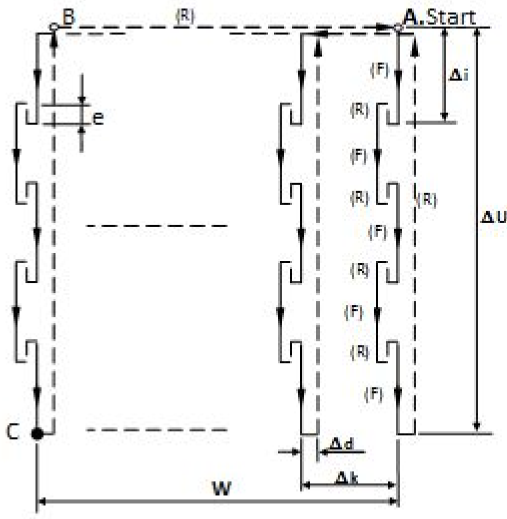
G77 R e ;
G77 X(U)___ Z(W)___ PΔi QΔk RΔd F___ ;
e: Retract distance (retract in X cutting Δi) <- can be set by system Pr4011
X: X position of C point (diameter positioning)
Z: Z position of C point
U: Incremental value (diameter) from point B to point C
W: Incremental value from point A to point B
Δi: The moving length of each X axis cut (in a positive, radius value)
Δk: Cutting depth of each Z axis cut (positive value)
Δd: The retract distance of Z axis when cutting to the end point (this value is zero if retract along original path)
F: Feedrate

2.46.2 **Description**

The G77 command is a lateral (X-axis) peck cutting cycle. This command can be used for X-axis grooving or drilling in the X-axis. For example, a groove is cut on the outer face to facilitate threading retraction and avoid incomplete threading. In addition, lathe often needs parting tool to cut the workpiece from the chuck, the G77 command is needed..

X axis only supports setting as diameter axis, and Z axis only supports setting as radius axis.

Action description



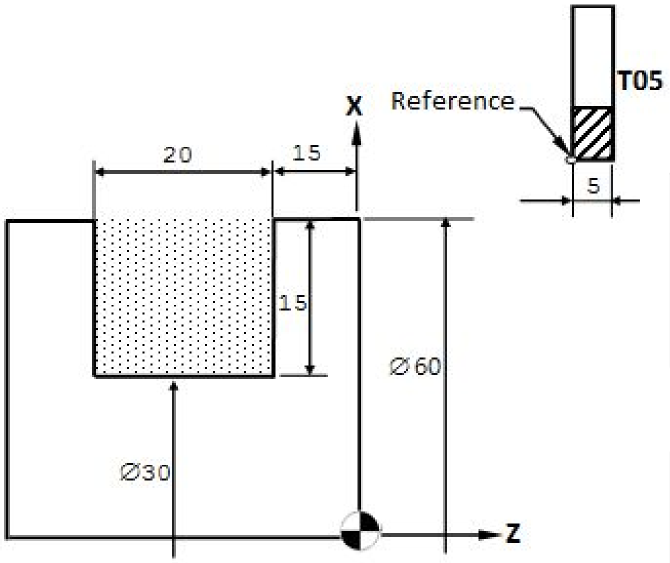
1. Positioning to **point A** (start point) before cycle starts
2. When executing G77, the tool starts peck type cutting from point A to target X position. Each feed Δi followed by a retract distance e. At target X position, retract a distance Δd in Z direction and then quickly retract to start point.
3. Then the tool moves along Z direction by Δk distance and continues the same cycle until reaches the machining endpoint B. The tool will automatically return from point B to point A and wait for the next cycle.

2.46.3 **Precaution**

1. When e and Δd are assigned by R argument with X or Z, R is the Z-axis retract distance. R is usually positive, but If the starting and end point (point A to point B) are the same, then R should use sign to indicate retract direction.
2. If G77 is followed by the R argument without X or Z, this indicates the retract distance in X direction. This is a modal G code which will keep enabled until the program end.
3. If no Q**Δk** is set, the peck movement is cancelled and tool goes to X axis endpoint directly.
4. If the starting and endpoint (point A to point B) are different but RΔd is negative, then triggers alarm [MAR28 peck type turning retract direction conflict].
5. If there is RΔd, then the tool will retract with Δd at the end of each peck cycle including the first peck cycle. If the first peck cycle has a risk of interference in Z, add two lines of program can avoid this. As shown in Example 2.

2.46.4 **Example**

Example 1:

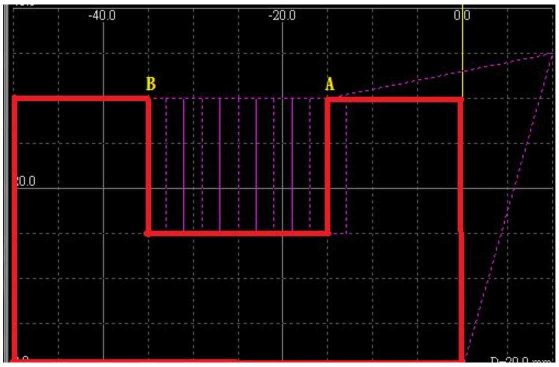
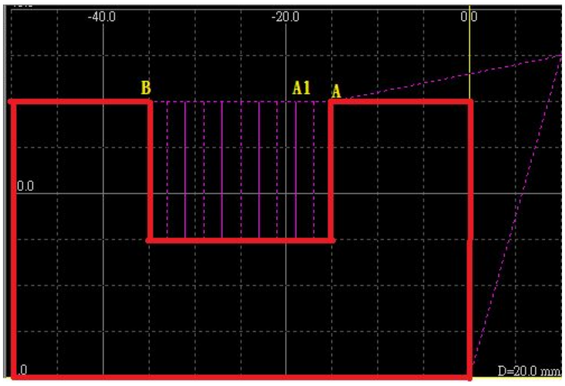


T05; //use tool NO. 5
G50 S1000; //max. rotate speed 1000 rpm
G96 S100 M03; //constant surface speed at 100 m/min, spindle rotate CW
M08; //cutting liquid ON
G00 X70.0 Z20.0;//positioning to workpiece
X60.0 Z-15.0; //positioning to start point
G77 R1.0;
G77 X30.0 Z-35.0 P8.0 Q4.0R0.0 F0.15;
//Execute lateral peck cutting cycle, depth of cutting in X 8.0 mm, and retract 1.0 mm.
//Z axis moves 4.0 mm after each peck cycle, and feed rate is 0.15 mm/rev
M09; //cutting liquid OFF
G28 X80.0 Z50.0;//positioning to specified mid-point, then return to machine zero
M05; //spindle stops
M30; //program ends

Example 2： R_ Retraction

	Retract Z in First Peck Cycle	No Retract Z in First Peck Cycle

Suggested Program	G00 X60.0 Z-15.0; //positioning to point A
	G77 R1.0;
	G77 X30.0 Z-35.0 P8.0 Q4.0 R3.0 F0.15;
	//Cycle start X cut depth is 8.0 mm and retract 1.0 mm. After one cycle, the Z retract 3.0mm and moves 4.0 mm to next peck cycle, feed rate 0.15mm/rev.
		G00 X60.0 Z-15.0; //positioning to point A
		G77 R1.0;
		G77 X30.0 Z-15.0 P8.0 Q4.0 F0.15;
		///Cycle start X cut depth is 8.0 mm and retract 1.0 mm. After one cycle, the Z axis returns to point A. Feed rate 0.15 mm/rev.
		G00 X60.0 Z-19.0; //positioning to point A1
		G77 R1.0;
		G77 X30.0 Z-35.0 P8.0 Q4.0 R2.0 F0.15;
		//Cycle start X cut depth is 8.0 mm and retract 1.0 mm. After one cycle, the Z retract 3.0mm and moves 4.0 mm to next peck cycle, feed rate 0.15mm/rev.

Graphic Simulation		

2.47 G78- Multiple Thread Cutting Cycle (C-Type)

中文文档：G78-复合型螺纹切削固定循环(C-Type)

Command Form

G78 P m r a Q Δdmin R d ;

G78 X(U)___ Z(W)___ R Δi P(Δk) Q(Δd) H___ (F__ or E__) Z1=__ D__;

m: repetition count in finishing (1~99), specified by system parameter #4044.

r: chamfering amount. When the pitch is set by L, the set value can be from 0.0L to 9.9L, the unit is 0.1L (double digit 00 to 99), specified by system parameter #4043.

a: angle of tool tip, angle from 80^0, 60^0, 55^0, 30^0, 29^0 and 0^0 can be specified or specified by system parameter #4042.

Δdmin: minimum cutting depth , specified by system parameter #4045 d: finishing allowance $(\Delta d\sqrt{n} - \Delta d\sqrt{n-1}) <$ Δdmin, specified by system parameter #4041

X(U): X coordinate in end point(bottom of tooth)

Z(W): Z coordinate in end point(bottom of tooth)

Δi: difference of thread radius

Δk: height of thread

Δd: depth of cut in first cycle

F: lead of thread in metric system (unit: mm/tooth)

E: lead of thread in imperial system (unit: tooth/inch)

H: numbers of thread (ex: H3 is three thread type cutting. Multiple thread F function is the distance neighbor thread)

Z1: Thread modification feature, set the tool tip Z axis absolute coordinate when reach into effective thread wall (with explanation down below)

D : The switch of chip removal function. → **Enable function : Set D to 1** / **Disable function : Set D to others than 1**

2.47.1 **Description**

Instructions for use

1. The G78 complex threading cycle automatically creates multiple cutting paths for threading.
2. for running G78 correctly, there is two lines in it,
    a. First line is not necessary(will use Pr4042~4045 if empty).
    b. Second line is necessary.
3. With given the parameters, the controller calculates the number of times required to cut the thread, the depth of each cut, and the starting point for each cut.
4. Starting from 10.114.50, the G78 also provide the re-threading function.
5. X axis only supports setting as diameter axis, and Z axis only supports setting as radius axis.
6. Incremental infeed method (using parameter 4052) supports current and after Controller version : 10.118.28E

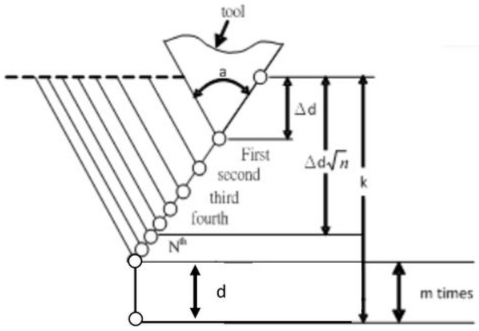
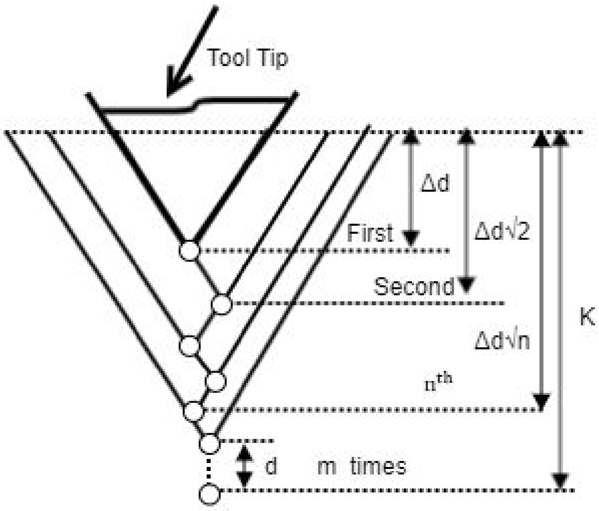
Comparison of Threading G Codes

1. G33(thread cutting): A 4-block sequence of commands is needed to finish one thread cutting, thus the programming of thread cutting in G33 is inconvenient and time-consuming.
2. G21(thread cutting cycle): A “single” cycle command of thread cutting. we can use one block of command to finish thread cutting, but it also need to repeating thread cutting many times so the program is also too long.
3. G78(multiple thread cutting cycle): Byusing only one commandG78finishes all needed cycle in thread cutting.Therefore G78 much simplifies shortens the procedure of programming.

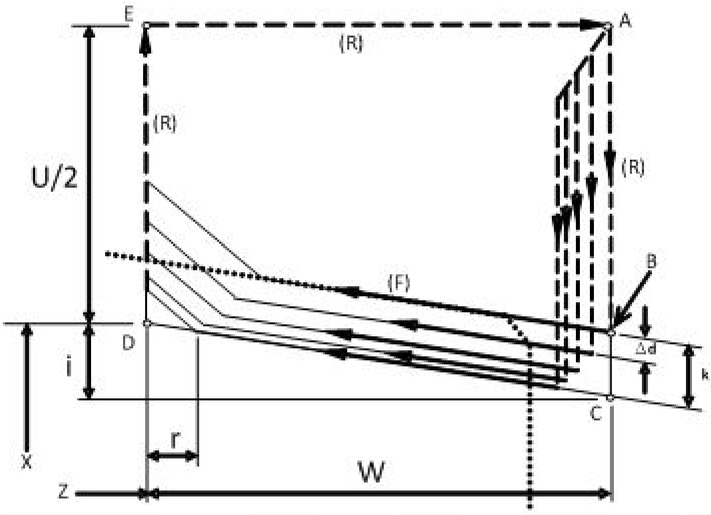
G78 Action description

Infeed methods:

Infeed methods can be chosen by setting parameter 4052 =0 or 1.

Flank Infeed (Pr4052 = 0)	**Incremental Infeed (Pr4052 = 1)**
 tool a Δd First second third fourth Nth Δd√n k d m times The way of feed in thread cutting and the depths of each cut: d: finishing allowance m: finishing times	 Tool Tip Δd First Δd√2 Second Δd√n nth K d m times The way of feed in thread cutting and the depths of each cut: d: finishing allowance m: finishing times

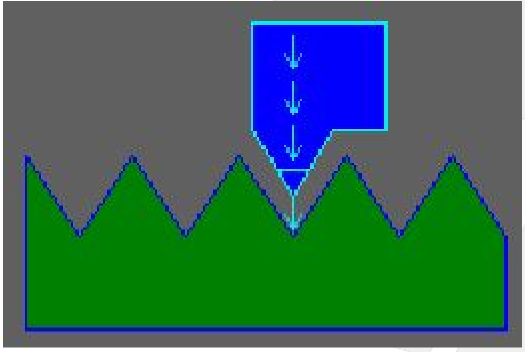
Cutting path



1. Positioning to **point A(start point)** by rapid traverse before cycle starts.
2. When executing G78, the tool will move along A→ B→ E→ A, to finish the rough threading according to the infeed setting.
3. After the rough turning is completed, base on to the value of the reserved distance and the number of fine turning, a constant-area cutting is performed to complete the fine threading.
4. Tool stops at point A after the last cut ends and wait for the next cycle of cutting.

Re-threading Function

- Threads may be worn or deformed for a long time and some can only be repaired instead of replaced. However, these workpieces have been removed from the spindle chuck, how to process again? Re-threading is a function that developed for this situation. As long as the workpiece to be repaired is re-clamped on the spindle, the thread can be re-cut along the old thread.
- The point of re-threading is actually the speed matching between Z-axis and the spindle. The trimming function is achieved by using this speed correlation, even if the workpiece has been removed from the lathe, it can be repeated twice or more times.
- To use the re-threading function, please follow the steps below:
    - Clip the workpiece to be repaired on the spindle chuck
    - Perform spindle positioning to return the spindle to the index
    - Maintain the spindle angle and move X and Z axes to let the tool tip to reach into any effective thread wall as close as possible.



    - Record the Z-axis absolute coordinate position and enter the Z1 argument
    - Move threading tool to a safe position
    - Perform G78 re-threading cycle

2.47.2 **Precaution**

1. From version 10.114.56E/10.116.0E/10.116.5 (included), the spindle override of entire thread-cutting cycle is locked at the value of the start of cycle, i.e., the spindle override button is in vain during thread-cutting cycle.
2. Before version 10.114.56E/10.116.0E/10.116.5, during thread-cutting cycle, the spindle override is locked at 100% when cutting and resume to setting of control panel while retracting. Therefore, one apply thread-cutting cycle with a spindle override that is not equal to 100% will find the spindle is under a frequent acceleration and deceleration situation.
3. Because the system will read the spindle positioning angle during the execution of the action, if Pr1621~the spindle axis number is not specified correctly, 【COR-016 Illegal variable access】 will be issued.
4. Chip removal function
    a. Syntec ring-type encoder and controller version 10.118.28E(and newer one) are required, then use D command to enable this function.
    b. Disable chip removal function when processing fine turning.
    c. Support only when reaching conditions below:
        i. Thread mode :
            1. Normal thread mode
            2. Fast thread mode
            3. Extremely fast thread
        ii. Machining direction :
            1. Parallel thread
            2. Taper thread
        iii. Thread number :
            1. One-start thread

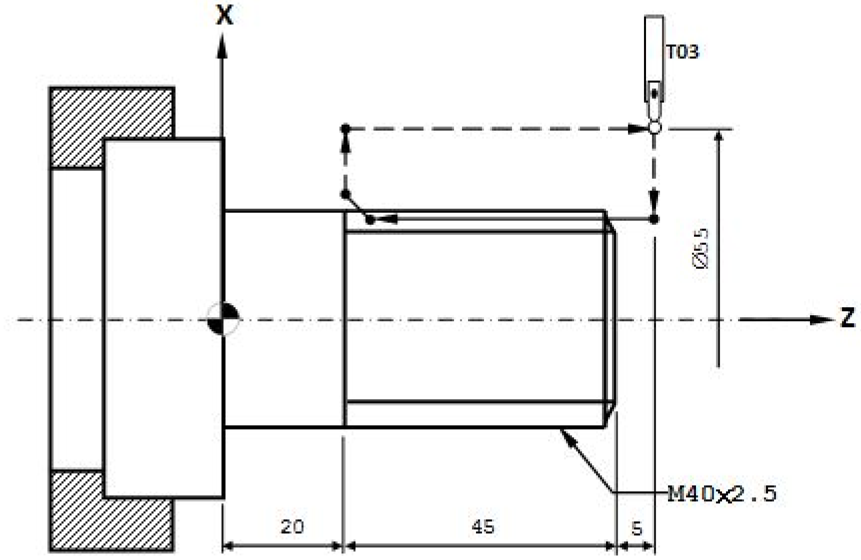
iv. Cutting mode :

1. One side cutting mode

d. Not support only when reaching one of conditions below: multi-start thread, two side cutting mode

2.47.3 **Example 1**

Compare with example one of G21

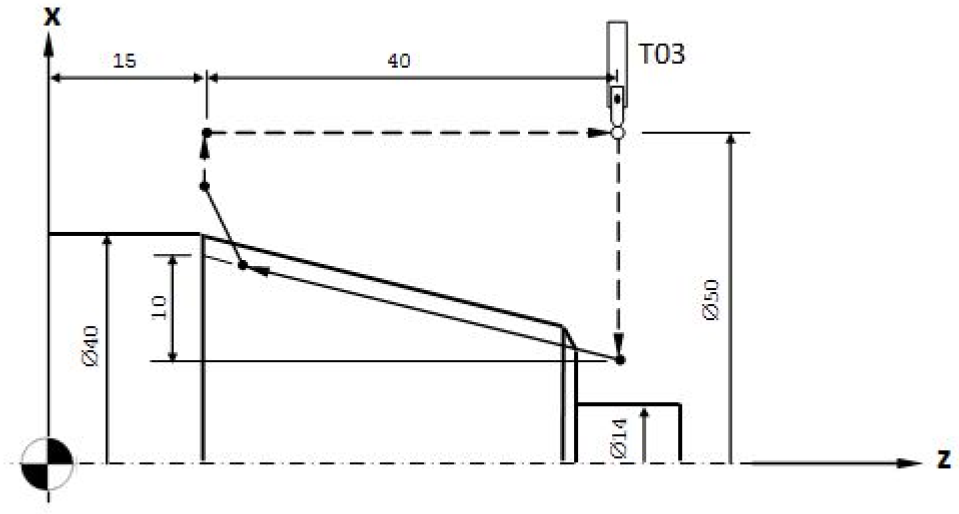


```
T03                //use tool NO. 3
G97 S600 M03     //constant rotate speed, 600 rpm CW
G00 X50.0 Z70.0 //positioning to the start point of cycle
M08                //cutting liquid ON
G78 P011060 Q0.15 R0.02//execute multiple repetitive cycle,
                         //finishing cutting once, escaping amount=Lead,
                         //angle of tooth 60°, Min. depth of cutting 0.15
                         //mm, finishing allowance 0.02 mm
G78 X36.75 Z20.0 R0.0 P1.624 Q1.0 H3 F2.5//difference radius
                         //of multiple thread cutting cycle is 0 mm, depth
                         //of thread 1.624 mm, first cutting value is 1.0
                         //mm, pitch is 2.5 and lead of thread is 7.5 mm, three tooth thread
                         //cutting
G28 X60.0 Z75.0 //positioning to specified mid-point and return to
                         //machine zero point
M09               //cutting liquid OFF
M05               //spindle stops
```

```
M30             //program ends
```

2.47.4 **Example 2**

compare with example two of G21, single tooth type, Pitch = 2.5 mm



```
T03             //use tool NO. 3
G97 S600 M03    //constant rotate speed, 600 rpm CW
G00 X50.0 Z55.0 //positioning to start point of cycle
M08             //cutting liquid ON
G78 P011060 Q0.15 R0.02//execute multiple repetitive cycle,
                    //finishing cutting once, escaping amount=Lead,
                    //angle of tooth 60°, Min. depth of cutting 0.15
                    //mm, finishing allowance 0.02 mm
G78 X36.75 Z15.0 R-10.0 P1.624 Q1.0 F2.5//difference radius of
                    //multiple thread cutting cycle is 10.0 mm, depth of
                    //thread 1.624 mm, first cutting value is 1.0 mm,
                    //lead of thread 2.5 mm, single tooth thread cutting
G28 X60.0 Z70.0 //positioning to specified mid-point and then return
                    //to machine zero point
M09             //cutting liquid OFF
M05             //spindle stops
M30             //program ends
```

-

-

-

2.48 G78.2- Complex Mid-Section Threading Cycle (C-Type)

中文文件：G78.2-复合型螺纹切削中段进刀固定循环(C-Type)

2.48.1 **Command Form**

G78.2 P m r a Q Δdmin R d I___ K___;

G78.2 X(U)___ Z(W)___ R Δi P Δk Q Δd H__ (F__ or E__) D__;

m: Number of fine turning(1~99), which can be set by Pr4044

r: Chamfer retract distance. When the pitch is set by L, the set value can be from 0.0L to 9.9L, the unit is 0.1L (double digit 00 to 99). Can also be set by Pr4043.

a: The angle of the tool tip can be selected from 80, 60, 55, 30, 29, 0, ...etc, or can be set by Pr4042.

Δdmin: Minimum depth of cut $(\Delta d\sqrt{n}-\Delta d\sqrt{n-1})<$ Δdmin which can be set by Pr4045.

d: The reserved distance for fine turning, can also be set by Pr4041.

I: Oblique feed height, which can be set by parameter Pr4047

K: Oblique feed length, which can be set by parameter Pr4046

X(U): X-axis endpoint (root)

Z(W): Z-axis endpoint (root)

Δi: Thread radius difference

Δk: Thread height

Δd: Depth of first cut

F: Metric thread pitch (mm/thread)

E: Thread per inch (thread/inch, English style)

H: Number of starts (Ex: H3 three-start threading, F refers to pitch under multi-start)

D : The switch of chip removal function. → **Enable function : Set D to 1** / **Disable function : Set D to others than 1**

2.48.2 **Description**

Instructions for use

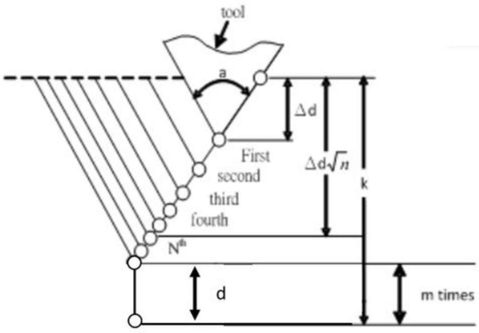
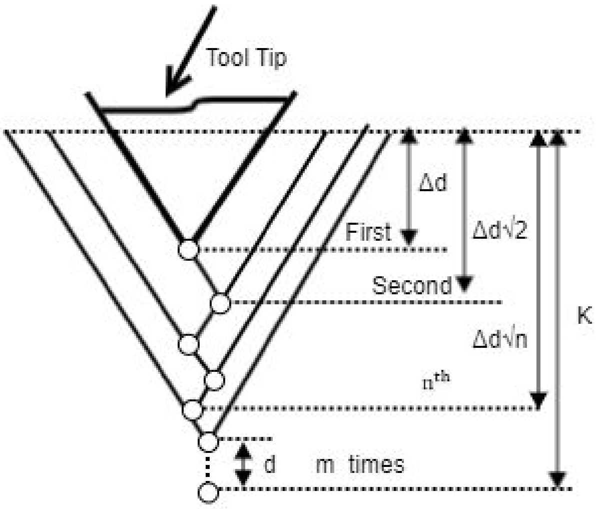
1. The G78.2 complex threading cycle automatically creates multiple cutting paths for threading.
2. For running G78.2 correctly, there is two lines in it,
    a. First line is not necessary(will use Pr4042~4045 if empty).
    b. Second line is necessary.
3. With given the parameters, the controller calculates the number of times required to cut the thread, the depth of each cut, and the starting point for each cut.
4. G78.2 differs from G78 in that it provides the feed setting for threading; when the thread begins at the middle of workpiece and tool have no clearance to accelerate, G78.2 can be used to avoid damage existing thread.

5. G78.2 supports straight and tapered threading, but the re-threading function is not supported.
6. X axis only supports setting as diameter axis, and Z axis only supports setting as radius axis.
7. Incremental infeed method (using parameter 4052) supports current and after Controller version : 10.118.28E

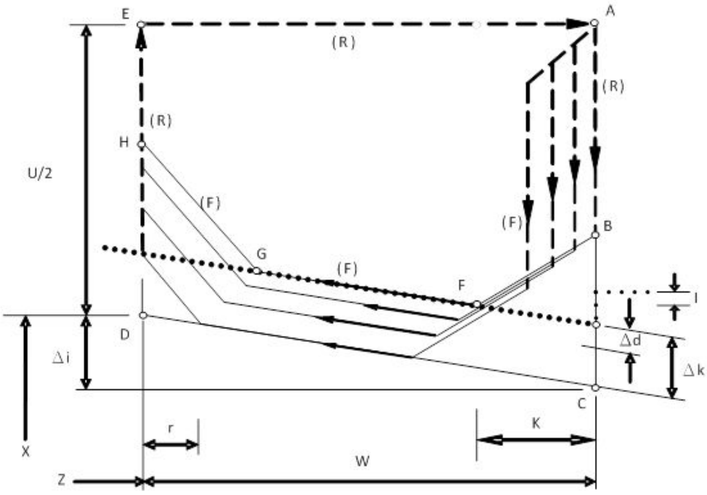
G78.2 Action Description

Infeed methods:

Infeed methods can be chosen by setting parameter 4052 =0 or 1.

Flank Infeed (Pr4052 = 0)	**Incremental Infeed (Pr4052 = 1)**
 tool a Δd First second third fourth N^th Δd√n k d m times The way of feed in thread cutting and the depths of each cut: d: finishing allowance m: finishing times	 Tool Tip Δd First Δd√2 Second Δd√n n^th K d m times The way of feed in thread cutting and the depths of each cut: d: finishing allowance m: finishing times

Cutting path:

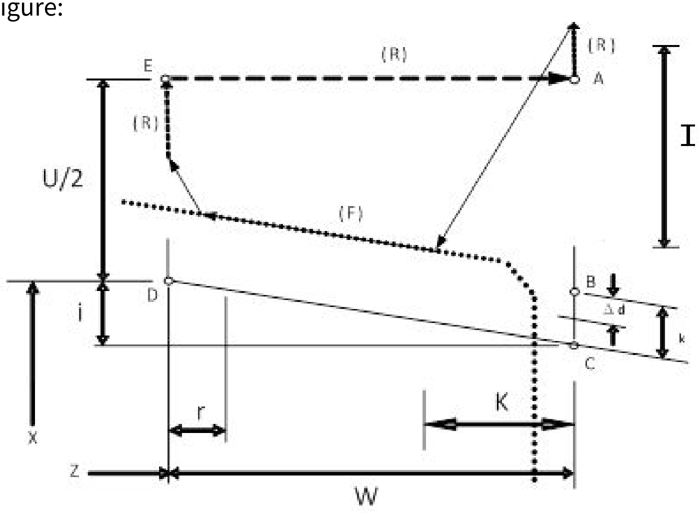


1. Position the tool to point A (starting point) before cycle starts.
2. Rapid position from point A to point B (oblique infeed point), and then feed the tool obliquely to point F (workpiece surface) to begin the turning cycle.
3. When executing G78.2, the tool will move along A→B→F→G→H→E→A to finish the rough threading according to the infeed setting.
4. After the rough turning is completed, base on to the value of the reserved distance and the number of fine turning, a constant-area cutting is performed to complete the fine threading.
5. Tool stops at point A after the last cut ends and wait for the next cycle of cutting.

2.48.3 **Precaution**

1. After 10.114.56E/10.116.0E/10.116.5 (include), the spindle override during threading cycle is locked to the setting before entering the cycle. That is, the override knob in the threading cycle is invalid until the cycle ends.
2. Moreover, before 10.114.56E/10.116.0E/10.116.5, the spindle override is locked to 100% during the infeed and resume knob setting when retracting. So if threading under non-100% spindle override, the spindle will frequently accelerate and decelerate.
3. When using G78.2 command, the oblique feed height I and the length K must be correctly set. If the arguments are not given, Pr4046 and Pr4047 will be used. If both Pr4046 and Pr4047 are 0, MACRO alarm 19 "The thread feed has no specified length or height" will be triggered.
4. If two G78.2 commands are used to cut two consecutive threads, besides setting the oblique feed height I and length K, the second command must meet the following conditions:
   a. Z-axis feed point position must be equal to a positive integer multiple of the pitch
   b. The distance between the last revolution of thread one and the first revolution of thread two must be a positive integer multiple of the pitch (refer to Example 3).
5. If the distance of the infeed plus the retract exceeds the total movement of the Z axis, the MARCO alarm 20 "Thread feed/retraction chamfer distance exceeds total Z axis movement" will be triggered.
6. If the infeed position in the X-axis direction is higher than the starting point, MARCO alarm 21 "Threading X-axis feed position higher than the starting point" will be triggered to avoid interference. See the following

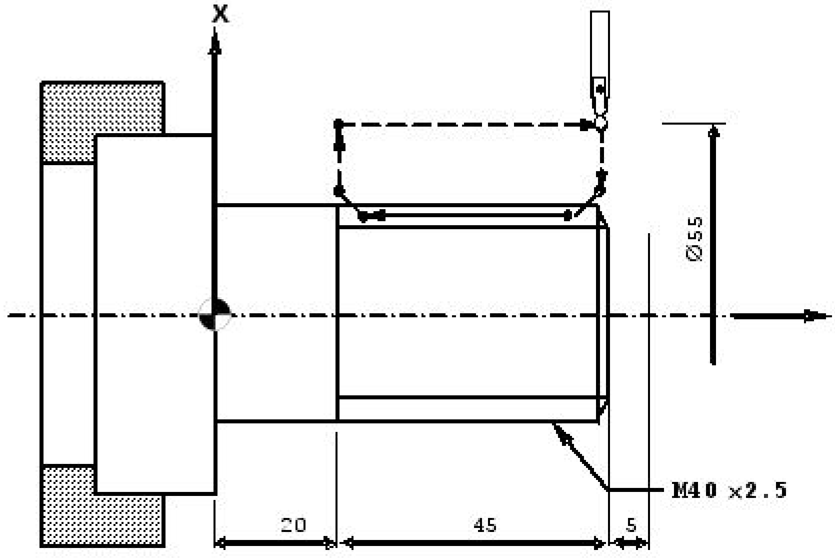
figure:



7. After the 10.118.12D (include), the end retract refers to the PR4018 retracting angle under ultra high-speed threading, the finish quality is better than the high-speed threading and the general threading.
8. Chip removal function
    a. Syntec ring-type encoder and controller version 10.118.28E(and newer one) are required, then use D command to enable this function.
    b. Disable chip removal function when processing fine turning.
    c. Support only when reaching conditions below:
        i. Thread mode :
            1. Normal thread mode
            2. Fast thread mode
            3. Extremely fast thread
        ii. Machining direction :
            1. Parallel thread
            2. Taper thread
        iii. Thread number :
            1. One-start thread
        iv. Cutting mode :
            1. One side cutting mode
    d. Not support only when reaching one of conditions below: multi-start thread, two side cutting mode

2.48.4 Example 1

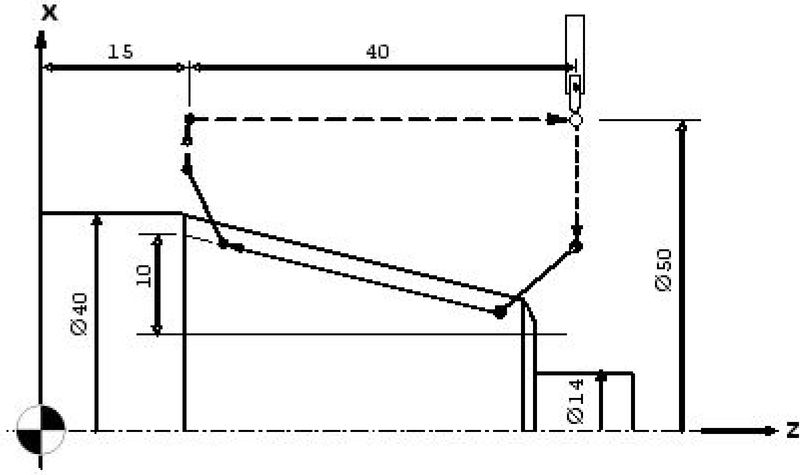
Compare with **G92 (Threading Cycle)** example 1, three-start thread.



```
T03; //use tool NO.3
G97 S600 M03;//constant RPM at 600 rpm CW
G00 X50.0 Z70.0;//positioning to start point
M08; //cutting liquid ON

G78.2 P011060 Q0.15 R0.02 I2.0 K1.0;
//complex mid-section threading cycle, the number of fine turning is 1, retract distance=pitch, oblique feed height 2 mm, oblique feed length 1 mm, thread angle is 60 ° , min cutting depth is 0.15 mm, reserved distance of fine turning 0.02 mm.

G78.2 X36.75 Z20.0 R0.0 P1.624 Q1.0 H3 F2.5;
//complex mid-section threading cycle, with a radius difference 0 mm, thread depth of 1.624 mm, first tool feed of 1.0 mm, thread lead (pitch) is 2.5 mm, three start threading

G28 X60.0 Z75.0;//positioning to specified mid-point, then return to machine zero
M09; //cutting liquid OFF
M05; //spindle stops
M30; //program ends
```

2.48.5 **Example 2**

Compare with **G92 (Threading Cycle)** example 2, single start thread, pitch = 2.5 mm



T03; //use tool NO.3
G97 S600 M03;//constant surface speed, rotate 600 rpm CW
G00 X50.0 Z55.0;//positioning to start point
M08; //cutting liquid ON

G78.2 P011060 Q0.15 R0.02 I2.0 K1.0;
//complex mid-section threading cycle, the number of fine turning is 1, retract distance=pitch, oblique feed height 2 mm, oblique feed length 1 mm, thread angle is 60 ° , min cutting depth is 0.15 mm, reserved distance of fine turning 0.02 mm.

G78.2 X36.75 Z15.0 R-10.0 P1.624 Q1.0 F2.5;
//complex mid-section threading cycle, with a radius difference 10 mm, thread depth of 1.624 mm, first tool feed of 1.0 mm, thread lead (pitch) is 2.5 mm, single start threading

G28 X60.0 Z75.0;//positioning to specified mid-point, then return to machine zero
M09; //cutting liquid OFF
M05; //spindle stops
M30; //program ends

Example 3

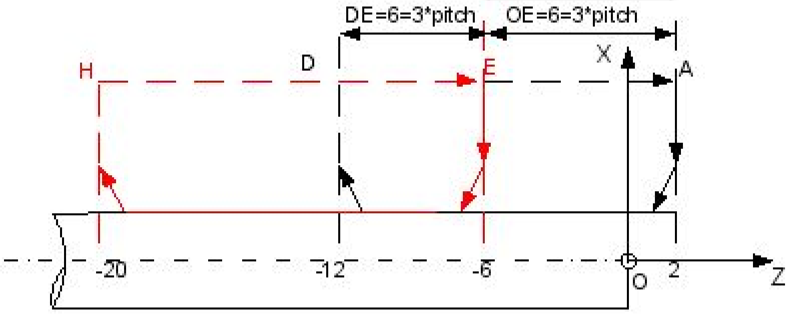
For round bar of 20mm long, two G78.2 are used to thread two segments (pitch 2mm, angle 60).
The 1st segment is from Z2 to Z-12, the 2nd segment is from Z-6 to Z-20.

1. Program:
   T0404 //use tool NO.4
   M03 S1500 //spindle rotates 1500 rpm CW
   M98 H11 //call the subroutine, starting with N11
   M98 H12 //call the subroutine, starting with N12
   M30

   N11
   G0X50. Y0. //positioning to start point
   Z2. //position to 1st segment infeed point Z2

```
G78.2 P010560 Q0.1 R0.02;
//complex mid-section threading cycle, the number of fine turning is 1, retract distance=0.5 of the pitch, oblique feed height 1.732 mm, oblique feed length 0.5 mm, thread angle is 60 ° , min cutting depth is 0.1 mm, reserved distance of fine turning 0.02 mm.
G78.2 X14.85 Z-12. P0.6 Q0.2 F2.0; //threading cycle of 1st segment, retraction point Z-12.0
//complex mid-section threading cycle, thread depth of 1.624 mm, first tool feed is 0.2 mm, thread pitch is 2.0 mm, single start threading
M99

N12
G0X50. Y0. //positioning to start point
Z-6. //position to 2nd segment infeed point
Z-6.0
G78.2 P010560 Q0.1 R0.02;
//complex mid-section threading cycle, the number of fine turning is 1, retract distance=0.5 of the pitch, oblique feed height 1.732 mm, oblique feed length 0.5 mm, thread angle is 60 ° , min cutting depth is 0.1 mm, reserved distance of fine turning 0.02 mm.
G78.2 X14.85 Z-20. P0.6 Q0.2 F2.0; //threading cycle of 2nd segment, retraction point Z-20.0
//compound thread cutting mid-stage feed fixed cycle with thread depth is 1.624 mm, first tool feed is 0.2 mm, thread pitch is 2.0 mm, single start threading
M99
```

2. Parameter setting
   Pr4018= 60 //tooth cutter angle
   Pr4046= 1000 //0.5pitch= 1mm（Unit: LIU）
   Pr4043= 5 //0.5pitch（Unit: 0.1pitch）
   Pr4047=1732 //Pr4046*tan60（Unit: LIU）



2.48.6 **Appendix**

Thread Definition

Pitch and Lead

Start

Start means the number of grooves in a screw.

Pitch

Pitch is the distance between screw grooves and is commonly used with inch sized products and specified as threads per inch.

Lead

Lead is the linear travel the nut makes per one screw revolution and is how ball screws are typically specified.

The pitch and lead are equal with single start screws. For multiple start screws, the lead is the pitch multiplied by the number of starts.

2.49 G80~G89- Canned Cycle For Drilling (C-Type)

2.49.1 Description

The canned cycle for drilling simplifies the programs generally contain several blocks of movement into a single block G command.

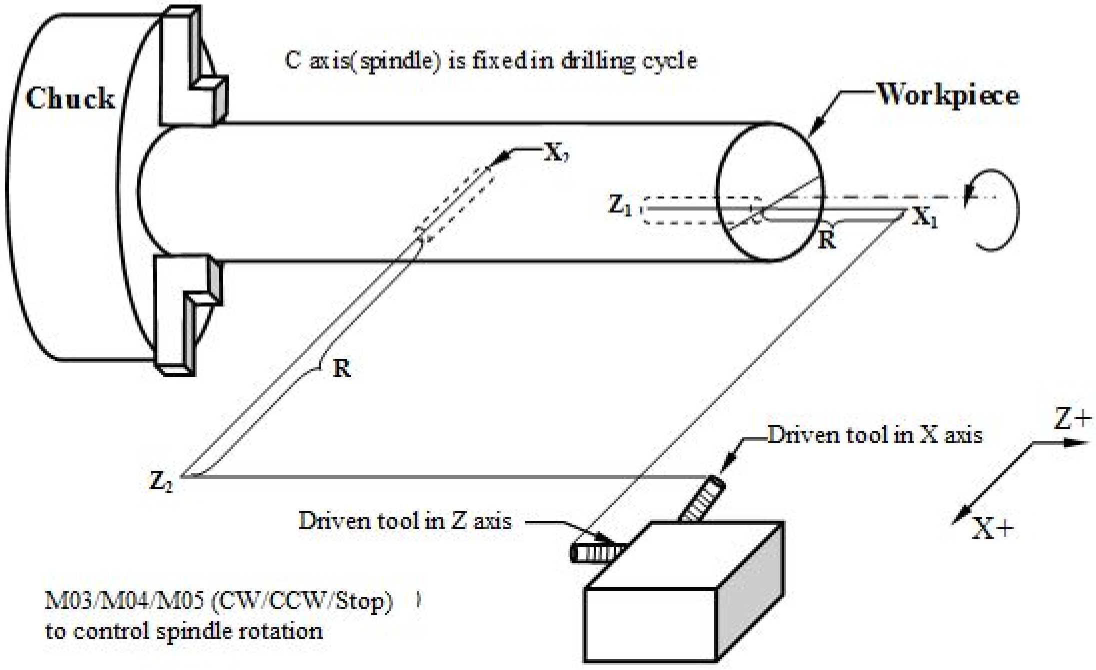
Table of Canned Cycle

G Code	Drilling Axis	Hole Bottom Action	Retract Move	Application
G80	--	--	--	Disable cycle
G83	Z	Dwell	Rapid Positioning	Drilling cycle

G84	Z	Spindle reverse (CCW)	Feed	Tapping cycle
G85	Z	Dwell	Feed	Boring cycle
G87	X/Y	Dwell	Rapid Positioning	Drilling cycle
G88	X/Y	Spindle reverse (CCW)	Feed	Tapping cycle
G89	X/Y	Dwell	Feed	Boring cycle

Note 1: Use M04 command to reverse the spindle.
Note 2: With Q argument or not decides G83/G87 cutting feeds **continuously** or **intermittently**.

Drilling Cycle Diagram

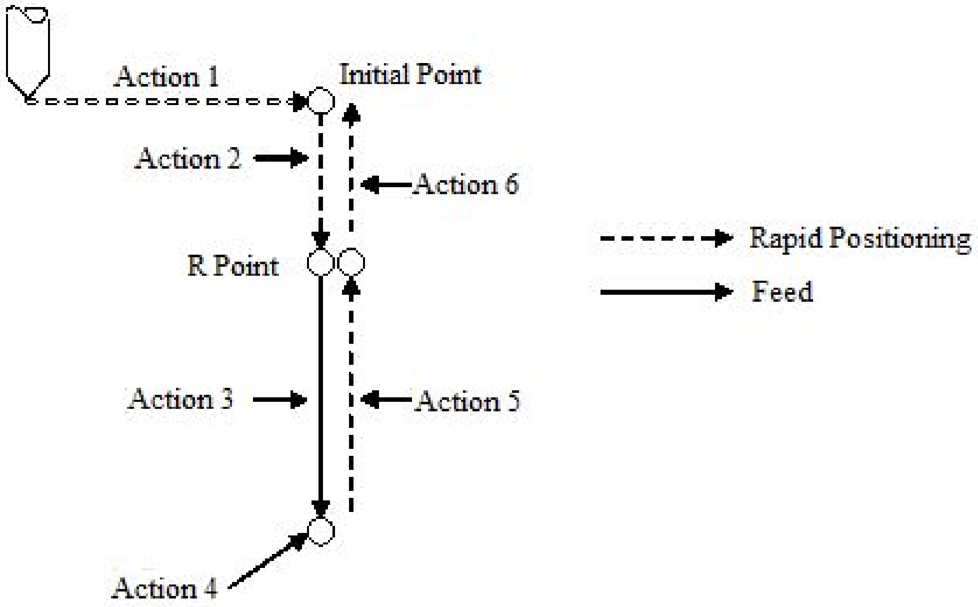

C axis(spindle) is fixed in drilling cycle
Chuck
Workpiece
X2
Z1
X1
R
R
Z2
Driven tool in X axis
Z+
X+
Driven tool in Z axis
M03/M04/M05 (CW/CCW/Stop)
to control spindle rotation

※The difference between G83/G87, G84/G88, G85/G89 is the direction of drilling in X/Y axis

In general, the drilling cycle consists of the following six operation sequences:

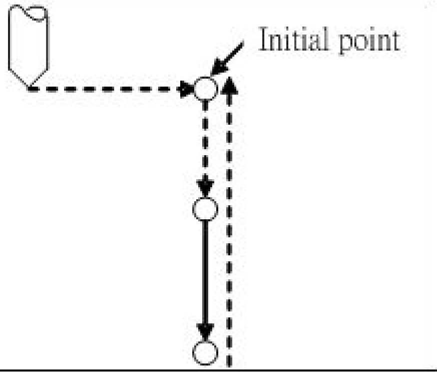
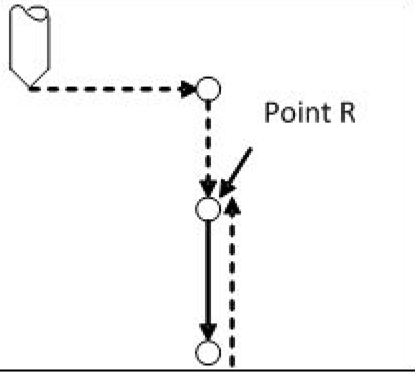
Action 1	Rapid positioning of X(Z), Y and C axis
Action 2	Rapid positioning up to point R
Action 3	Hole machining
Action 4	Action at the bottom of the hole
Action 5	Retract to point R
Action 6	Rapid positioning to the initial point

Note: 10.114.50 (included) provides Pr4019 to determine whether to enable Y-axis positioning function (factory preset off)



※ Regarding the retract movement, the lathe A Type will be returned to the initial level. (Please refer to the figure below)

Note that G98/G99 in the lathe A Type is the feed unit setting, not the R(retract) point.

G98	G99
	

2.50 G83/G87- Front/Side Drilling Cycle (C-Type)

Command Form

G83 X(U)/Y(V)__ C(H)__ Z(W)__ R__ Q__ P__ F__ K__ M__ I__ J__ D__;
G83.1 X(U)/Y(V)__ C(H)__ Z(W)__ R__ Q__ P__ F__ K__ M__ I__ J__;
G83.2 X(U)/Y(V)__ C(H)__ Z(W)__ R__ Q__ P__ F__ K__ M__ I__ J__;
G83.3 X(U/Y(V))__ C(H)__ Z(W)__ R__ Q__ P__ F__ K__ M__ I__ J__;
or
G87 Z(W)__ C(H)__ X(U)/Y(V)__ R__ Q__ P__ F__ K__ M__ I__ J__ D__;
G87.1 Z(W)__ C(H)__ X(U)/Y(V)__ R__ Q__ P__ F__ K__ M__ I__ J__;

G87.2 Z(W)__ C(H)__ X(U)/Y(V)__ R__ Q__ P__ F__ K__ M__ I__ J__;
G87.3 Z(W)__ C(H)__ X(U)/Y(V)__ R__ Q__ P__ F__ K__ M__ I__ J__;

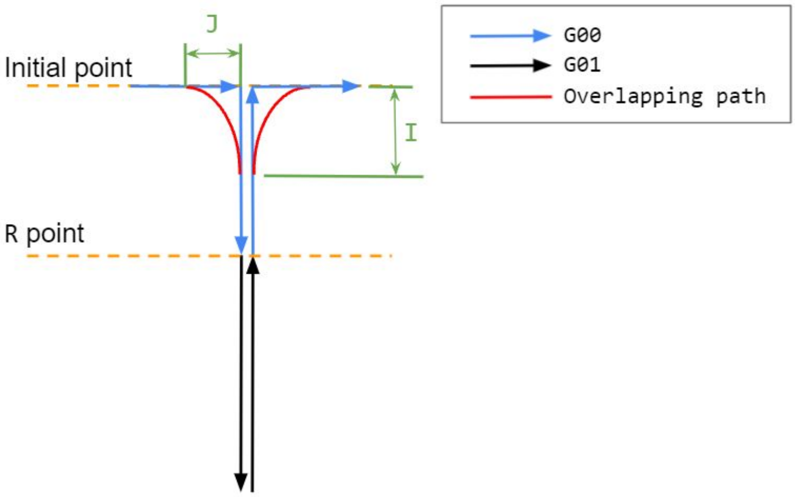
X(U)/Y(V)__C__ or Z(W)__C__ : Hole position

- To use the Y(V) argument, Pr4019 needs to be set to 1.

Z(W)__C__ or X(U)/Y(V)__C__ : Absolute position of the hole bottom (incremental distance from R point to the hole bottom)

- To use the Y(V) argument, Pr4019 needs to be set to 0.

R : The distance from the initial level to point R level (The sign is invalid, regardless of tapping axis is diameter/radius positioning, R argument is always in radius)
Q : Depth of cut for each cutting feed (The sign is invalid)
P : Dwell time at the bottom of the hole (With decimal points unit: seconds; no decimal point, unit refer to Pr17 and Pr3241)
F : Cutting federate
K : Number of repetitions
M: C-axis clamps (Clamp) M Code, Clamp Code plus 1 is C-axis release code (Unclamp Code)
I : Overlapping distance of the drilling axis (When the drilling finishes, it is the overlapping distance from exit the hole along the drilling axis to the next G00.). It is valid when Pr4008(setting for drilling/tapping mode) value is 1. Unit: IU. effective version: 10.116.54.
J : Overlapping distance of the positioning axis (After the previous drilling finishes, it is the overlapping distance from G00 positioning to the current position.). It is valid when Pr4008(setting for drilling/tapping mode) value is 1. Unit: IU. effective version: 10.116.54.



D: Drilling chip breaking switch. → Enable function : Set D to 1 / Disable function : Set D to others than 1

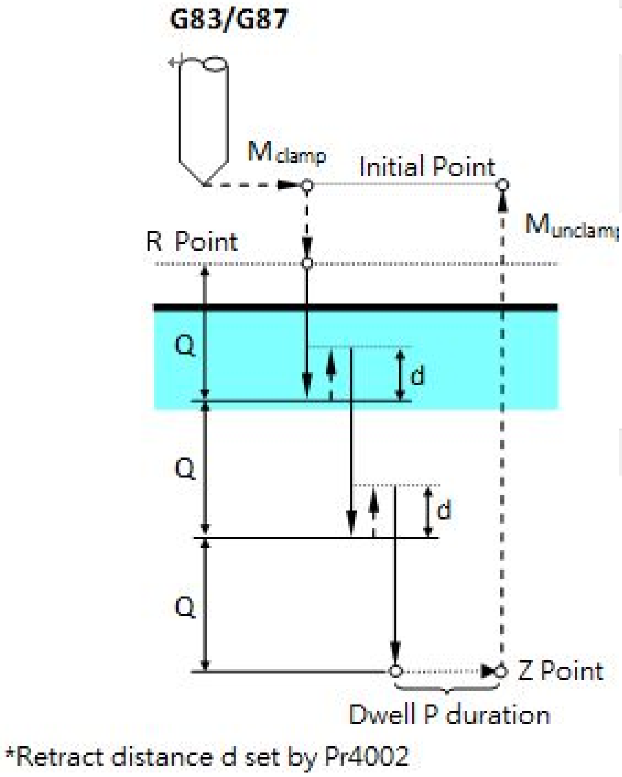
Description

1. The G83/G87 command is a **front/side drilling cycle** for the drilling in CNC lathes. The spinning driven tool drills on front/side of a workpiece fixed by the spindle. The peck drilling corresponding to G83/G87 is specified by user parameter Pr4001. For the need to use different types in the same program, the peck drilling type can be overridden by the following G codes:
   - G83.1/G87.1 High-speed Deep Hole Drilling Cycle
   - G83.2/G87.2 Front/side Deep Hole Peck Drilling Cycle
   - G83.3/G87.3 Front/side General Deep Hole Drilling Cycle
2. In order to enable overlapping, set Pr4008 to 1, otherwise I and J arguments have no effect.
3. Overlapping will only be valid above R point.

4. Overlapping means when the program has two consecutive blocks of drilling G codes or the consecutive blocks of drilling G code and G00, the second block will start when the first block remains a certain distance from the finish. The distance is called "overlapping distance", see I & J in the figure above.
5. Overlapping is suitable for continuous drilling cycles. It does not need I/J argument in every row of the program but 'overlaps' by a constant distance. In Example 5, The overlapping distance of the drilling axis and the positioning axis of each drilling command is 2 and 3 respectively. And the overlapping distance becomes zero when the disable command G80 is executed.
6. If Pr4008 is set to 1 but the I and J arguments are not given, the system will use argument R as the overlapping distance.
7. If all the below conditions are satisfied, The rapid drilling mode will be triggered.
    a. Argument Q isn't issued.
    b. Serial spindle or not serial spindle but Pr1791~ equals to 3.
    c. Argument P isn't issued.
    d. Fast tapping option is supportive.
8. The machine action is smoother when moving reversely along the Z axis at the bottom of the hole.
9. Not able to activate Feedhold, Reset during the drilling process, Feedhold and Reset will be activated after returning to the initial point or R point.
10. Do not modify the G01 feedrate or activate MPG simulation during the drilling process of multiple holes.
11. The drilling chip breaking function is only applicable to TYPE 3 drilling cycle, in this mode, if D argument exists, the drilling chip breaking will be activated whatever the value of P and Q arguments. (Even if both P and Q arguments are null, the rapid drilling mode will not be triggered).

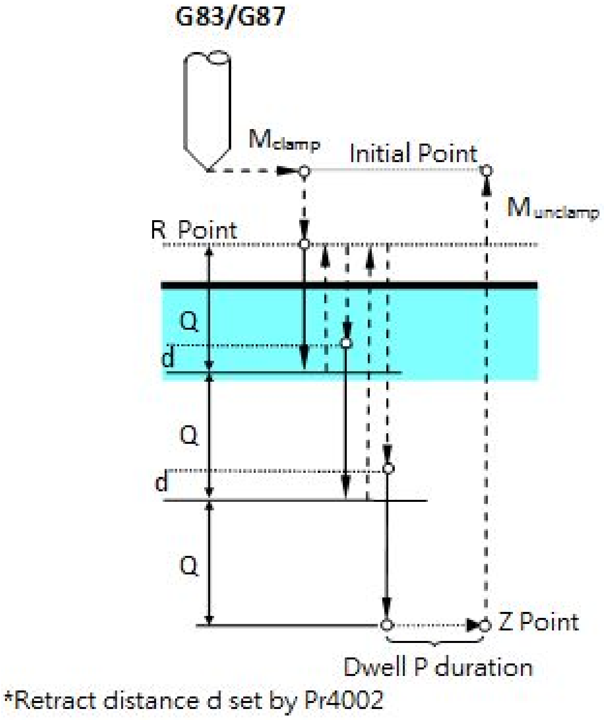
2.50.1 TYPE I

High-speed Deep Hole Drilling Cycle (Pr4001= 1 or G83.1/G87.1)



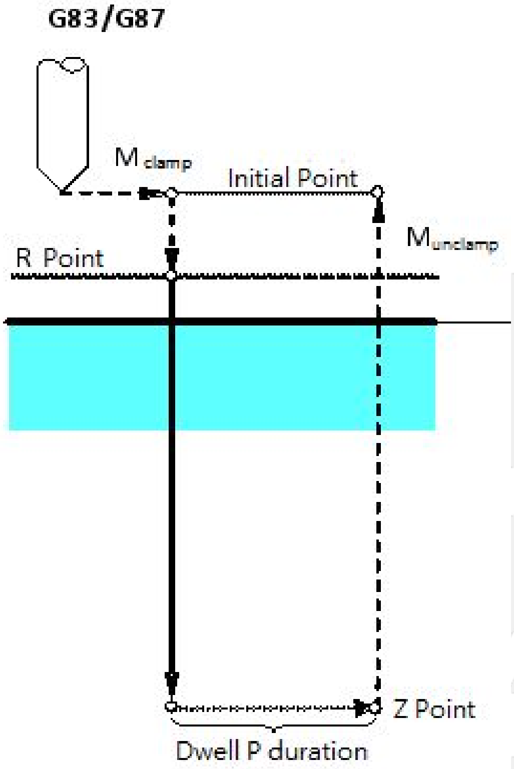
2.50.2 TYPE II

General Deep Hole Drilling Cycle (Pr4001=0 or G83.3 /G87.3)



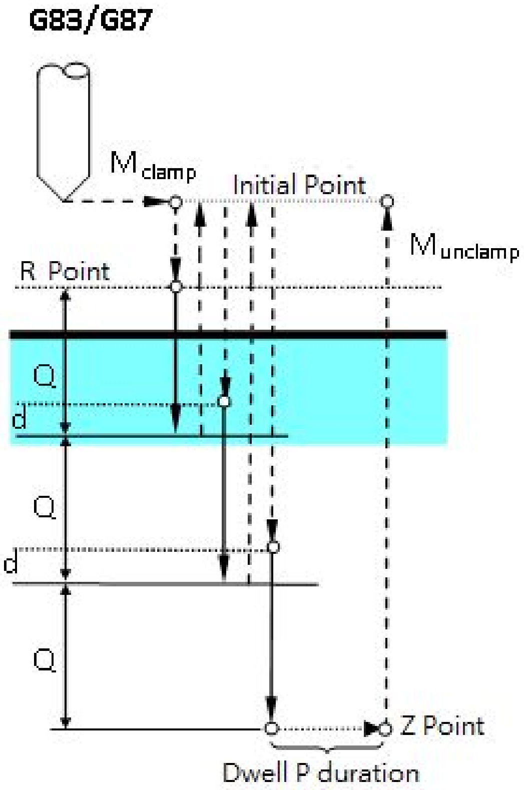
2.50.3 TYPE III

Drilling Without Q



2.50.4 Type IV

Deep Hole Peck Drilling Cycle (Pr4001=2 or G83.2/G87.2)



2.50.5 **Precaution**

1. When the R point plane is set lower than the hole bottom plane such as R value is greater than the distance from the hole bottom plane to the initial point, the system triggers [MAR- 011 drilling (boring) hole cycle feed plane R is lower than the bottom plane]
2. If G83/G87 command does not specify the hole bottom plane (Z/X position), the system triggers [MAR-012 drilling (boring) cycle without specifying the bottom plane of the hole].
3. Use G83/G87 command with feed distance Q=0, the system will issue [MAR-013 peck drilling cycle without specifying the feed amount Q]
4. If the machine can use the Y axis as one of the sides, Y (V) axial commands in G83/G87 can be used in the X (U) axis format.
5. Because the system will read the spindle positioning angle during the execution of the action, if Pr1621~the spindle axis number is not specified correctly, 【COR-016 Illegal variable access】 will be issued.

2.50.6 **Example**

- Example 1:

Assume M31 is the Clamp command of C axis, M32 is Unclamp command of C axis.
M03 S1000; //spindle rotates 1000 rpm
G00 X50.0; //position to starting point
G83 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31;

```
// first drilling at 0° of C axis
C90.0 M31; // second drilling at 90° of C axis
C180.0 M31; // third drilling at 180° of C axis
G80; //disable cycle
M02; //program ends
```

- Example 2: (Front-side High-speed Deep HoleDrilling)

Assume M31 is the Clamp command of C axis, M32 is Unclamp command of C axis.

```
M03 S1000; //spindle rotates 1000 rpm
G00 X50.0; //position to starting point
G83.1 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31;
// first drilling at angle 0° of C axis
C90.0 M31; // second drilling at 90° of C axis
C180.0 M31; // third drilling at 180° of C axis
G80; //disable cycle
M02; //program ends
```

- Example 3: (Front Deep Hole Peck Drilling)

Assume M31 is the Clamp command of C axis, M32 is Unclamp command of C axis.

```
M03 S1000; //spindle rotates 1000 rpm
G00 X50.0; //position to starting point
G83.2 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31;
// first drilling at angle 0° of C axis
C90.0 M31; // second drilling at 90° of C axis
C180.0 M31; // third drilling at 180° of C axis
G80; //disable cycle
M02; //program ends
```

- Example 4: (Front General Deep Hole Peck Drilling)

Assume M31 is the Clamp command of C axis, M32 is Unclamp command of C axis.

```
M03 S1000; //spindle rotates 1000 rpm
G00 X50.0; //position to starting point
G83.3 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31;
// first drilling at angle 0° of C axis
C90.0 M31; // second drilling at 90° of C axis
C180.0 M31; // third drilling at 180° of C axis
G80; //disable cycle
M02; //program ends
```

- Example 5: (Front Continuous Drilling + Pr4008 set to 1)

Assume M31 is the Clamp command of C axis, M32 is Unclamp command of C axis.

```
M03 S1000; //spindle rotates 1000 rpm
G00 X50.0; //position to starting point
G83 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31 I2. J3.;
// first drilling at angle 0° of C axis, and the overlapping distance of the drilling axis is 2 mm, the overlapping distance of the positioning axis is 3 mm.
C90.0 M31;
// second drilling at 90° of C axis, and follow I, J argument settings
C180.0 M31;
// third drilling at 180° of C axis, and follow I, J argument settings
G80; //disable cycle and clear the I, J argument settings
M02; //program ends
```

- Example 6: (XY Positioning, Front High-speed Deep Hole Drilling +Pr4019 =1)

Assume M31 is the Clamp command of C axis, M32 is Unclamp command of C axis.

```
M03 S1000; //spindle rotates 1000 rpm
G00 X50.0 Y20.0; //position to starting point
G83.1 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31;
// first drilling at angle 0° of C axis
C90.0 M31; // second drilling at 90° of C axis
C180.0 M31; // third drilling at 180° of C axis
G80; //disable cycle
M02; //program ends
```

- Example 7: (Y axis drilling and Z axis position+Pr4019=0)

Assume M31 is the Clamp command of C axis, M32 is Unclamp command of C axis.

```
M03 S1000; //spindle rotates 1000 rpm
G00 Z50.0 ; //position to starting point
G87.1 Y-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31;
// first drilling at angle 0° of C axis
C90.0 M31; // second drilling at 90° of C axis
C180.0 M31; // third drilling at 180° of C axis
G80; //disable cycle
M02; //program ends
```

2.51 G84/G88- Front/Side Tapping Cycle (C-Type)

2.51.1 **Description**

G84 X(U)/Y(V)_ C(H)_ Z(W)_ R_ P_ (F_ or E__) K_ M_ Q_ I_ J_;

G84.1 X(U)/Y(V)_ C(H)_ Z(W)_ R_ P_ (F_ or E__) K_ M_ Q_ I_ J_;

G84.2 X(U)/Y(V)_ C(H)_ Z(W)_ R_ P_ (F_ or E__) K_ M_ Q_ I_ J_;

or

G88 Z(W)_ C(H)_ X(U)/Y(V)_ R_ P_ (F_ or E__) K_ M_ Q_ I_ J_;

G88.1 Z(W)_ C(H)_ X(U)/Y(V)_ R_ P_ (F_ or E__) K_ M_ Q_ I_ J_;

G88.2 Z(W)_ C(H)_ X(U)/Y(V)_ R_ P_ (F_ or E__) K_ M_ Q_ I_ J_;

X(U)/Y(V)_C_or Z(W)_C_: Hole position data

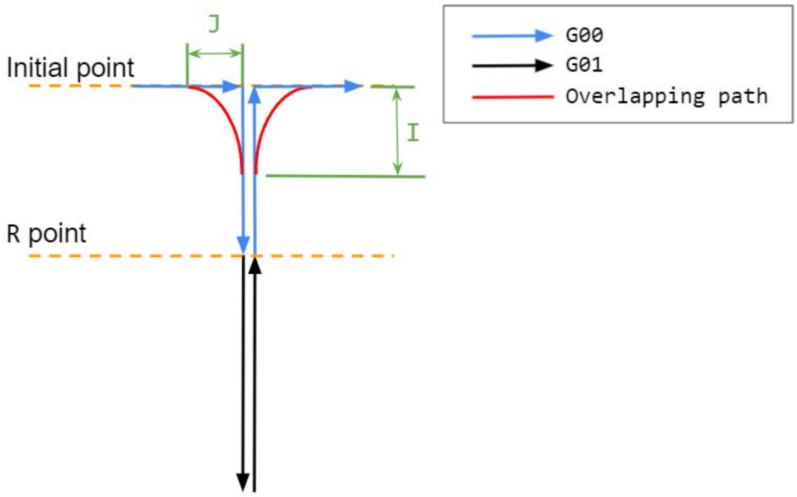
- To use the Y(V) argument, Pr4019 needs to be set to 1.

Z(W)_C_or X(U)/Y(V)_C_: Absolute position of the hole bottom (incremental distance from R point to the hole bottom)

- To use the Y(V) argument, Pr4019 needs to be set to 0.

R: The distance from the initial level to point R level (The sign is invalid, regardless of tapping axis is diameter/radius positioning, R argument is always in radius)
P: Dwell time at the bottom of the hole (With decimal points unit: seconds; no decimal point, unit refer to Pr17 and Pr3241)
F: Cutting feedrate (mm/rev). Equivalent to the pitch of the metric tooth.
E: Thread per inch (if F and E both used, E argument will be ignored), only available in ver 10.116.16B, 10.116.18,

10.117.19, and after.
K: Number of repetitions
M: C-axis clamps (Clamp) M Code, Clamp Code plus 1 is C-axis release code (Unclamp Code)
Q: Peck tapping feed distance (incremental & positive value, no Q or Q is larger than the distance from R to the bottom means "Don't use peck tapping")
I: Overlapping distance of the tapping axis (When the tapping finishes, it is the overlapping distance from exit the hole along the tapping axis to the next G00.). It is valid when Pr4008(setting for drilling/tapping mode) value is 1. Unit: IU. effective version: 10.116.54.
J: Overlapping distance of the positioning axis (After the previous tapping finishes, it is the overlapping distance from G00 positioning to the current position.). It is valid when Pr4008(setting for drilling/tapping mode) value is 1. Unit: IU. effective version: 10.116.54.



2.51.2 **Description**

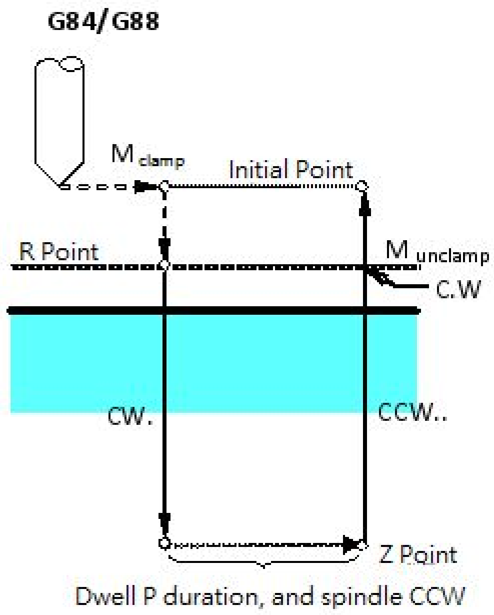
1. The G84/G88 command is **front Z / side X direction tapping cycle** for the tapping in CNC lathes. The spinning driven tool taps on front/side of a workpiece fixed by the spindle. The peck tapping types corresponding to G84/G88 is specified by Pr4004. For the need to use different types in the same program, the peck tapping type can be overridden by the following G codes:
    - G84.1/G88.1 front/side high speed peck tapping
    - G84.2/G88.2 front/side general peck tapping
2. In order to enable overlapping, set Pr4008 to 1, otherwise I and J arguments have no effect.
3. Overlapping will only be valid above R point.
4. Overlapping means when the program has two consecutive blocks of tapping G codes or the consecutive blocks of tapping G code and G00, the second block will start when the first block remains a certain distance from the finish. The distance is called "overlapping distance", see I & J in the figure above.
5. Overlapping is suitable for continuous tapping cycles. It does not need I/J argument in every row of the program but 'overlaps' by a constant distance. In Example 5, The overlapping distance of the tapping axis and the positioning axis of each tapping command is 2 and 3 respectively. And the overlapping distance becomes zero when the disable command G80 is executed.
6. If Pr4008 is set to 1 but the I and J arguments are not given, the system will use argument R as the overlapping distance.
7. The speed of retracting can be increased (tapping tool and Z-axis synchronously accelerate) up to three times of the tapping feed speed. Set by Pr4006, preset with the same feed speed.
8. The function of spindle positioning before tapping can be enabled by Pr4007. When enabled, the spindle will always perform positioning before each tapping; therefore the same hole can be repeatedly tapped without

thread failure. Valid version starts from 10.116.14 and only available for Serial Bus Spindles. In addition, the spindle positioning angle can be defined by the spindle origin offset (Pr1771~Pr1780).

9. If all below conditions are satisfied, The fast tapping mode will be triggered.
   a. Argument Q isn't issued or Q is larger than the distance from R to the bottom.
   b. Serial spindle or not serial spindle but Pr1791~ equals to 3.
   c. Argument P isn't issued.
   d. Fast tapping option is supportive.

TYPE I

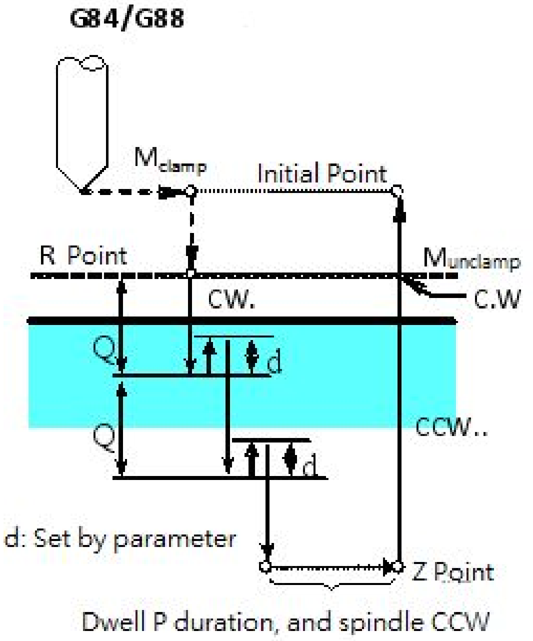
No Q Argument



1. The tapping axis reaches the tapping position, C axis clamping
2. Z axis reach to point R by G00 (R must be incremental value)
3. Position the spindle (If Pr4007=0, this action can be ignored.)
4. Start tapping, pitch is the specified F_ value
5. Until Z axis reachs the specified Z depth of G84 (Z absolute / W incremental)
6. Tapping tool stops
7. Dwell P seconds (with decimal point, unit: 1 s, without decimal point, unit: 0.001 s)
8. Tapping tool rotates CCW (internal command M04)
9. Retract to point R by the feedrate of tapping.
10. C axis unclamping
11. Dwell several second (dwell time set at Pr4003. Default value is 0 second)
12. Tapping tool rotates CW (M03)
13. Return to the initial point.

TYPE II

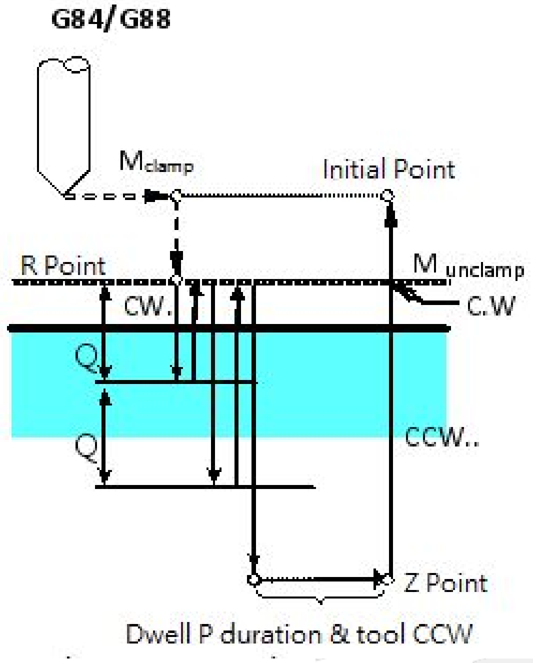
High Speed Peck Tapping (Pr4004 = 1 or G84.1/G88.1)



1. The tool first moves to the specified (X, C) point with G00
2. C axis clamping
3. The tapping axis reaches point R by G00.
4. Position the spindle (If Pr4007=0, this action can be ignored.)
5. Tap down with to a depth of one cut Q relative to the current depth with G01.
6. Tapping tool stops and rotates CCW, tool retracts d distance with G01 (Pr4005 is set).
7. Tapping tool stops and rotates CW again, and then tap down with G01 to the depth of one cut Q relative to current depth.
8. Tapping tool stops and rotates CCW, tool retracts d distance with G01 (Pr4005 is set).
9. Repeat the above tapping action until reaching the bottom of the hole Z point.
10. Pause P seconds at the bottom of the hole and then CCW
11. Retract to program P point with G01
12. C axis unclamping
13. Dwell several seconds and then rotate CW (Dwell time set at Pr4003. Default value is 0 second)
14. Move to starting point with G00.

TYPE III

General Peck Tapping (Pr4004 = 0 or G84.2/G88.2)



1. The tool first moves to the specified (X, C) point with G00
2. C axis clamps
3. The tapping axis reaches point R by G00.
4. Position the spindle(If Pr4007=0, this action can be ignored.)
5. Tap down with to a depth of one cut Q relative to the current depth with G01.
6. Tapping tool stops and rotates CCW, then rises to R point above workpiece with G01.
7. Tapping tool stops and rotates CW again, and then tap down with G01 to the depth of one cut Q relative to current hole depth.
8. Tapping tool stops and rotates CCW, then rises to R point above workpiece with G01.
9. Repeat the above tapping action until reaching the bottom of the hole Z point.
10. Pause P seconds at the bottom of the hole and then CCW
11. Retract to program P point with G01
12. C axis unclamping
13. Dwell several seconds and then rotate CW (Dwell time set at Pr4003. Default value is 0 second)
14. Move to starting point with G00

2.51.3 **Precaution**

1. Depending on the spindle type, the CNC will use different modes for tapping. Each tapping mode requires a different initial state of the spindle before tapping.

Tapping Mode	**Spindle Type**	**Need Spindle Spinning Before Tapping**
Tracking Tapping	Pr.1791=1	Yes. Spindle 1 needs C64 or C65 to be ON first.

<table>
<tr><th>Tapping Mode</th><th>Spindle Type</th><th>Need Spindle Spinning Before Tapping</th></tr>
<tr><td>Synchronous Tapping</td><td><table>
<tr><th></th><th>Spindle type</th></tr>
<tr><td>Pulse Spindle</td><td>Pr.1791=3</td></tr>
<tr><td>Syntec Serial Bus Spindle</td><td>Pr.1791=4 or Pr.1791=2</td></tr>
<tr><td>Yaskawa Serial Bus Spindle</td><td>Pr.1791=3</td></tr>
<tr><td>Other Serial Bus Spindles</td><td>Depends, recommend Pr1791=3</td></tr>
</table></td><td>No. The spindle can start tapping from zero speed without first rotating CW or CCW.</td></tr>
<tr><td>Inverter VF Tapping</td><td>Pr.1791=0</td><td>Yes. The PLC needs to enable output that starts the inverter CW or CCW.</td></tr>
</table>

2. When G84/G88 end, the spindle resumes to the state before tapping. For example, if the spindle was rotating CW before tapping, the spindle automatically resumes to rotating CW after tapping.
3. G84/G88 can be disabled with G80, and the mode will also be disabled when encountering G00, G01, G02, G03, and other canned cycle G codes. Otherwise, the mode will maintain (modal G code). At this time, if the M code is used for the spindle clamp/unclamp, give C axis command will lead to a sequence action: unclamp→ C axis move→ clamp→ tapping. Give X and Y axis move commands will not affect spindle state: X or Y axis movement→ tapping. (According to Pr4019 Y axis positioning control)
4. G84/G88 can do left-handed thread tapping. Use M4 after G84/G88 will begin left-handed tapping. Note that this M4 will not be processed by PLC (PLC will not receive M4 signal), but PLC will process the G84/G88 arguments. Therefore, regardless of the M code of the spindle CCW, M4 is used for the left-handed tapping. EX: Main Spindle uses M4 for CCW, and the Secondary Spindlespindle uses M104 for CCW. However, both Main & SEcondary Spindles use G84Z_F_M4 for left-handed tapping.
5. Before G84/G88 tapping, the machining spindle must be specified (R791). R791=2 means the second spindle is used for machining. Default is the first spindle if not specified.
6. If Feedhold or Reset is pressed during tapping, the tapping action will still complete and stop at point R.
7. When the R point is set lower than the hole bottom, such as R value is greater than the distance from the hole bottom plane to the initial point plane, the system triggers [MAR- 011 drilling (boring) hole cycle feed plane R is lower than the bottom plane]
8. If the G85/G89 command does not specify the bottom plane of the hole (Z/X coordinate), the system triggers [MAR-012 drilling (boring) hole cycle without specifying the bottom plane of the hole].
9. If the machine can use the Y axis as a side, the X (U) axis in the G84/G88 command format can be changed to Y (V) axis.
10. Before using the driven tool tapping, please switch the machining spindle of the system to the driven tool. The switching M code of the main spindle, refer to the manual from machine builder.
11. Before switching the machining spindle (R791~), if the current spindle is in the tapping state, it must be canceled first with G80 to avoid unexpected actions.

12. Because the system will read the spindle positioning angle during the execution of the action, if Pr1621~the spindle axis number is not specified correctly, 【COR-016 Illegal variable access】 will be issued.
13. Since the tapping cycle uses the standard M03/M04/M05 for spindle rotation direction control, if there is a custom M03/M04/M05, please include the M03/M04/M05 command within the custom MACRO. In this case, it will be treated as the standard M03/M04/M05 to ensure the correct operation of the tapping cycle.

2.51.4 **Example**

- Example 1: (General Tapping)

Assume M31 is Clamp command of C axis; M32 is Unclamp command of C axis

```
M03 S500;          //spindle rotates CW with 500rpm/
G00 X50.0;         //position to starting point
G84 Z-40.0 C0.0 R-5.0 P10.0 F0.5 M31; //first hole tapping of C axis at 0°
C90.0 M31;         //second hole tapping of C axis at 90°
C180.0 M31;        //third hole tapping of C axis at 180°
G80 M05;           //disable tapping mode, spindle stops
M02;               //program ends
```

- Example 2: (Front High-speed Peck Tapping)

Assume M31 is Clamp command of C axis; M32 is Unclamp command of C axis

```
M03 S500;          //spindle rotates CW with 500rpm
G00 X50.0;     //position to starting point
G84.1 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31; //first hole tapping of C axis at 0°
C90.0 M31;     //second hole tapping of C axis at 90°
C180.0 M31;        //third hole tapping of C axis at 180°
G80 M05;           //disable tapping mode, spindle stops
M02;               //program ends
```

- Example 3: (Front General Peck Tapping)

Assume M31 is Clamp command of C axis; M32 is Unclamp command of C axis

```
M03 S500;          //spindle rotates CW with 500rpm
G00 X50.0;         //position to starting point
G84.2 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31; //first hole tapping of C axis at 0°
C90.0 M31;         //second hole tapping of C axis at 90°
C180.0 M31;        //third hole tapping of C axis at 180°
G80 M05;          //disable tapping mode, spindle stops
M02;              //program ends
```

- Example 4: (General Left-Handed Tapping)

```
M03 S500;          //spindle rotates CW with 500rpm
G00 X50.0;         //position to starting point
G84 Z-40.0 C0.0 R-5.0 P10.0 F0.5 M4; //first hole tapping of C axis at 0°
                  // Regardless of spindle number, always use M4 for left-handed tapping.
C90.0;            //second hole tapping of C axis at 90°
C180.0;           //third hole tapping of C axis at 180°
G80 M05;           //disable tapping mode, spindle stops
M02;               //program ends
```

- Example 5: (Universal tapping+Pr4008=1)

Assume M31 is Clamp command of C axis; M32 is Unclamp command of C axis

```
M03 S500;          //spindle rotates CW with 500rpm
G00 X50.0;        //position to starting point
```

```
G84 Z-40.0 C0.0 R-5.0 P10.0 F0.5 M31 I2. J3.;
//first hole tapping of C axis at 0°, and the overlapping distance of the tapping axis is 2 mm, the overlapping
distance of the positioning axis is 3 mm.
C90.0 M31;
//second hole tapping of C axis at 90°, inherit I, J argument set value
C180.0 M31;
//third hole tapping of C axis at 180°, inherit I, J argument set value
G80 M05;               //cancel tapping mode, spindle stops and clear the I, J argument settings
M02;                   //program ends
```

- Example 6: (XY-axis Positioning, Z-axis Front High-speed Peck Tapping+Pr4019 =1)

Assume M31 is Clamp command of C axis; M32 is Unclamp command of C axis

```
M03 S500; //spindle rotates CW with 500rpm
G00 X50.0 Y20.0; //position to starting point
G84.1 Z-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31; //first hole tapping of C axis at 0°
C90.0 M31; //second hole tapping of C axis at 90°
C180.0 M31; //third hole tapping of C axis at 180°
G80 M05; //disable tapping mode, spindle stops
M02; //program ends
```

- Example 7: (Y-axis High-speed Peck Tapping, Z-axis Positioning+Pr4019=0)

Assume M31 is Clamp command of C axis; M32 is Unclamp command of C axis

```
M03 S500; //spindle rotates CW with 500rpm
G00 Z50.0; //position to starting point
G88.1 Y-40.0 C0.0 R-5.0 P10.0 Q500 F0.5 M31; //first hole tapping of C axis at 0°
C90.0 M31; //second hole tapping of C axis at 90°
C180.0 M31; //third hole tapping of C axis at 180°
G80 M05; //disable tapping mode, spindle stops
M02; //program ends
```

2.52 G85/G89 - Front/Side Boring Cycle (C-Type)

2.52.1 Command Form

G85 X(U)/Y(V)__ C(H)__ Z(W)__ R__ P__ F__ K__ M__Q__D__I__J__;

or

G89 Z(W)__ C(H)__ X(U)/Y(V)__ R__ P__ F__ K__ M__Q__D__I__J__;

X(U)/Y(V)__C__ or Z(W)__C__: Hole coordinate

- To use the Y(V) argument, Pr4019 needs to be set to 1.

Z(W)__ C__ or X(U)/Y(V)__C__: Absolute position of the bottom of the hole (incremental value from the R point to the bottom of the hole)

- To use the Y(V) argument, Pr4019 needs to be set to 0.

R: The distance from the initial point to the point R (sign is invalid, regardless boring axis is diameter/radius positioning; use radius value for R)

P: Dwell time at the bottom of the hole (unit with decimal in second; unit without decimal refer to Pr17 and Pr3241)

F: Feedrate

K: Number of repetitions

M: M code for C axis clamping. Unclamp code is Clamp Code +1.

Q : Tool displacement at the bottom (need to be positive)

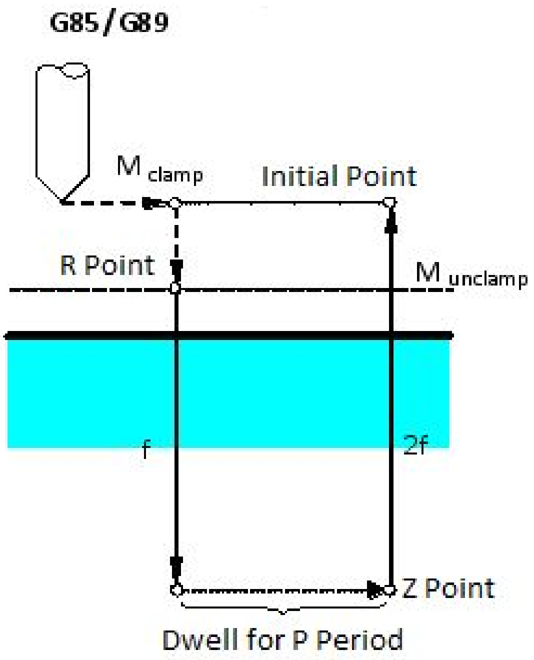
D : Tool OSS direction

I : Milling Tool CW M code, and Milling Tool stop M code equal to **I**+2

J : Milling Tool Position M code

2.52.2 Description

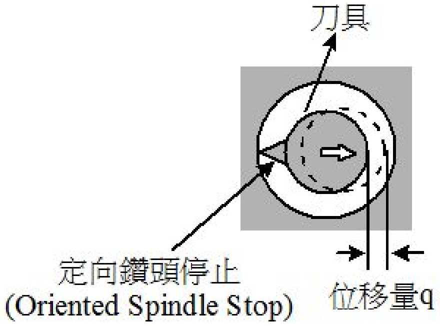
G85/G89 command is **Front/Side Boring Cycle**, used in the boring in the CNC lathe. The rotating tool performs front/side boring cycle on a clamped & fixed workpiece.



2.52.3 Precaution

1. When the R point is set lower than the hole bottom, such as R value is greater than the distance from the hole bottom plane to the initial point plane, the system triggers [MAR- 011 drilling (boring) hole cycle feed plane R is lower than the bottom plane]
2. If the G85/G89 command does not specify the bottom plane of the hole (Z/X coordinate), the system triggers [MAR-012 drilling (boring) hole cycle without specifying the bottom plane of the hole].
3. If the machine can use the Y axis as a side, the X (U) axis in the G85/G89 command format can be changed to Y (V) axis.
4. Because the system will read the spindle positioning angle during the execution of the action, if Pr1621~the spindle axis number is not specified correctly, 【COR-016 Illegal variable access】 will be issued.
5. Fine boring function:
    a. **Q, D, I, J all need to input at the begin of the Cycle**
    b. The Q value is a modal value required in the G85/G89 cycle, and it must be specified with great care.

c. **OSS（Oriented Spindle Stop）Direction**



D	OSS direction
0	+X
1	-X
2	+Y
3	-Y
4	+Z
5	-Z

2.52.4 **Example**

- Example 1: Front (end-face) boring cycle

```
Assume M31 is Clamp C axis ; M32 is Unclamp C axis
S1000 M03; //spindle rotates CW with 1000 rpm
G00 X50.0 Y20.0; //position to starting point
G85 Z-40.0 C0.0 R-5.0 P100 F0.5 M31;
//first hole drilling of C axis at 0°
C90.0 M31; //second hole drilling of C axis at 90°
C180.0 M31; //third hole drilling of C axis at 180°
G80; //cancel cycle
M02; //program ends
```

- Example 2: side boring cycle

```
Assume M31 is Clamp C axis ; M32 is Unclamp C axis
S1000 M03; //spindle rotates CW with 1000 rpm
G00 Z50.0; //position to starting point
G89 X-40.0 C0.0 R-5.0 P100 F0.5 M31;
//first hole drilling of C axis at 0°
C90.0 M31; //second hole drilling of C axis at 90°
```

```
C180.0 M31; //third hole drilling of C axis at 180°
G80; //cancel cycle
M02; //program ends
```

- Example 3: side Y axis boring cycle

```
Assume M31 is Clamp C axis ; M32 is Unclamp C axis
S1000 M03; //spindle rotates CW with 1000 rpm
G00 Z50.0; //position to starting point
G89 Y-40.0 C0.0 R-5.0 P100 F0.5 M31;
//first hole drilling of C axis at 0°
C90.0 M31; //second hole drilling of C axis at 90°
C180.0 M31; //third hole drilling of C axis at 180°
G80; //cancel cycle
M02; //program ends
```

2.53 G92- Coordinate System Setting/ Spindle Max RPM Limit (C-Type)

2.53.1 Command Form

G92 X__ Z__;
or
G92 S__;
X, Z: Set the position of the basic coordinate system (G92) in the program coordinate system;
S: Effective Spindle RPM;

2.53.2 Description

The G92 command has two functions: 1. Coordinate system setting or 2. Effective Spindle maximum RPM limit.

Any suitable position can be defined as the working coordinate system zero point. That is, take the relative position of the tool and the mechanical zero point, and use G92 to set the zero point of another new sub-coordinate system.

The tool starts machining from this point after setting, and the absolute position references this coordinate system.

This command can also be used to offset coordinate system. If the old coordinate is (X, Z), the new coordinate is (X + ΔU, Z + ΔW).

When using the G96 (constant surface speed) command, this command is used to limit the maximum spindle RPM to prevent the spindle RPM too high due to the small effective diameter of the workpiece.

2.53.3 Precaution

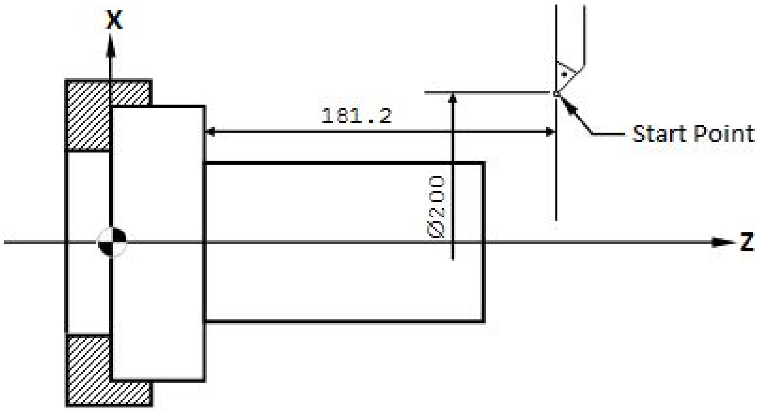
- Syntax S_ **is not applied for the spindle ID**, it is applied to the effective spindle of corresponding axis path. Therefore, do not use syntax such as "S1=_".
- The two functions of this G code are far different. Be careful when programming to avoid the coordinate system being misplaced and causing unexpected movements.
- When this G code is used to set coordinate system, its retention mode can be determined by Pr413. For details, please refer to the parameter manual.
- When the machine is turned on, the speed will not be limited if the maximum spindle speed limit G92 S_ has not been commanded.
- The spindle maximum speed limit G92 S_ will not be canceled by other commands except the system reset.
    a. 10.118.28D, 10.118.31 and earlier versions: When G97 is commanded after G96, G92 S_ spindle maximum speed limit will be canceled at the same time.

b. 10.118.28E, 10.118.32 ~ 10.118.30K, 10.118.32K, 10.118.40O, 10.118.41O, 10.118.48: G97 will not cancel G92 S_ spindle maximum speed limit.
c. 10.118.30L, 10.118.32L, 10.118.40P, 10.118.41P, 10.118.48A and later versions: When G97 is commanded after G96, G92 S_ spindle maximum speed limit will be canceled at the same time.

- G92 S0 can be used to stop the spindle.
- If the set maximum speed is lower than Pr1731~, the upper limit of the spindle speed will be the value set in Pr1731~.

2.53.4 **Example**

Coordinate setting



Specified method: G92 X200.0 Z181.2;
//tool starts from specified initial position to execute program

2.54 G92.1- Reset Absolute Coordinate System (C-type)

2.54.1 **Command Form**

G92.1 X_ Y_ Z_ I_ J_ K_ R_

X, Y, Z： specified the coordinate as the zero point of program coordinate system;

I： take X axis as the rotation center and rotates the YZ plane.

J： take Y axis as the rotation center and rotates the XZ plane.

K： take Z axis as the rotation center and rotates the XY plane.

R： the rotation angle of the coordinate

2.54.2 **Description**

G92.1 is similar to G92, both are used to build new coordinate systems. This command sets the specified point (assigned by the command) of current coordinate system as the program zero point of the new sub-coordinate system.

After setup, the tool will start the machining at the specified point and all the absolute commands will be computed with this coordinate system.

Comparing between G92 and G92.1

Command	Description
G92 X20. Y15. Z20.	Set the current position as the **X20. Y15. Z20 of new coordinate system**
G92.1 X20.Y15. Z20.	Set the **X20. Y15. Z20. of current coordinate system** as the zero point of new coordinate system

2.54.3 Note

1. The Machine Coordinate of controller is obtained from the formula： Machine Coordinate = Workpiece Coordinate (G54~) + Program Coordinate + G92.1 Offset + External Offset + MPG Offset + Tool Length Compensation.
2. G92.1 Offset is equal to the argument X_, Y_, Z_ in command of G92.1.
3. Argument I_, J_, K_ in G92.1 = The axes of G92.1 coordinate system's rotation center.
4. Argument R in G92.1 = G92.1 coordinate system's rotation angle.
5. The axial macro-variables of G92.1 coordinate system's offset are #1901~1918.
6. The macro-variable of G92.1 coordinate system's rotation angle is #1930. The default value is 0.
7. The axial macro-variables of G92.1 coordinate system's rotation center are #1931~#1933. The default values are 0, 0, 1.
8. #1930~#1933 will be affected by Pr413
   a. When Pr413 is set to 0, #1930~#1933 will be restored to default value after CNC reset or reboot.
   b. When Pr413 is set to 1, #1930~#1933 will be restored to default value after CNC reboot. However, these system variables will maintain the value of user's setting after CNC reset.
   c. When Pr413 is set to 2, #1930~#1933 will maintain the value of user's setting after CNC reset or reboot.
9. Please do not use G92 and G92.1 at the same time.

Program Example

Example 1, comparing between G92 and G92.1 (no external offset, tool length, tool wear compensation)

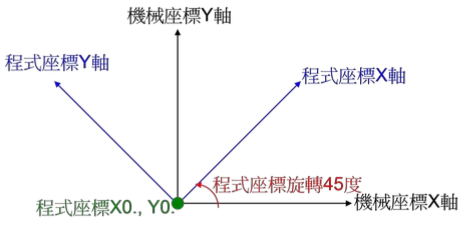
G92	G92.1

N1 G90 X10. Y10.
//machine coordinate X10. Y10.
//program coordinate X10. Y10
//#1901 #1902 coordinate X0. Y0.
N2 G92 X20. Y20.
//machine coordinate X10. Y10.
//program coordinate X20. Y20.
//#1901 #1902 coordinate X-10. Y-10.
N3 X50.
//machine coordinate X40. Y10.
//program coordinate X50. Y20.
//#1901 #1902 coordinate X-10. Y-10.
N4 M30
	N1 G90 X10. Y10.
	//machine coordinate X10. Y10.
	//program coordinate X10. Y10.
	//#1901 #1902 coordinate X0. Y0.
	N2 G92.1 X20. Y20.
	//machine coordinate X10. Y10.
	//program coordinate X-10. Y-10.
	//#1901 #1902 coordinate X20. Y20.
	N3 X50.
	//machine coordinate X70. Y10.
	//program coordinate X50. Y-10.
	//#1901 #1902 coordinate X20. Y20.
	N4 M30

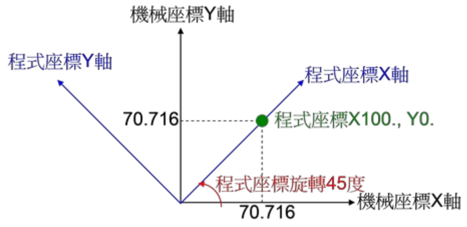
Example 2

Program Content	Figure

N1 G90 G0 X0. Y0.
//machine coordinate X0. Y0.
//program coordinate X0. Y0.
//#1901 #1902 coordinate X0. Y0.	

N2 G92.1 X0. Y0. K1. R45.
//machine coordinate X0. Y0.
//program coordinate X0. Y0.
//#1901 #1902 coordinate X0. Y0.
//XY plane rotates 45° with rotation center Z axis on program coordinate system, #1930 = 45°
	

Program Content	Figure

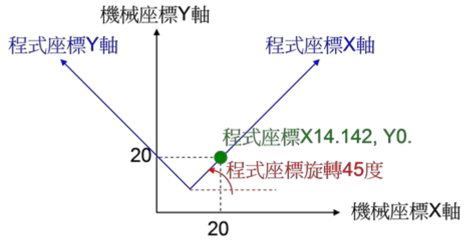
N3 G01 X100.
//machine coordinate X70.711 Y70.711
//program coordinate X100.000 Y0.000
//#1901 #1902 coordinate X0.000 Y0.000
	 機械座標Y軸
	程式座標Y軸
	程式座標X軸
	70.716
	程式座標X100., Y0.
	程式座標旋轉45度
	機械座標X軸
	70.716

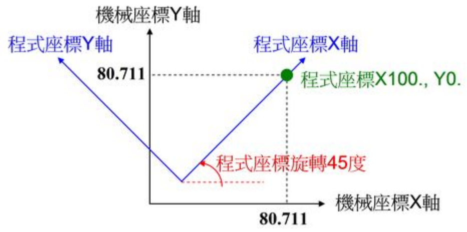
N4 M30	

Example 3

Program Content	Figure

N1 G90 G0 X20. Y20.
//machine coordinate X20. Y20.
//program coordinate X20. Y20.
//#1901 #1902 coordinate X0. Y0.	

N2 G92.1X10. Y10.K1.R45.
//machine coordinate X20. Y20.
//program coordinate X14.142 Y0.
//#1901 #1902 coordinate X10. Y10.
//XY plane rotates 45° with rotation center Z axis on program coordinate system, #1930 = 45°
	 機械座標Y軸
	程式座標Y軸
	程式座標X軸
	程式座標X14.142, Y0.
	20
	程式座標旋轉45度
	機械座標X軸
	20

N3 G01 X100. machine coordinate X80.711 Y80.711 program coordinate X100. Y0. #1901 #1902 coordinate X10. Y10.	 機械座標Y軸 程式座標Y軸 程式座標X軸 80.711 程式座標X100., Y0. 程式座標旋轉45度 機械座標X軸 80.711
N4 M30	

2.55 G93- Inverse Time Feed (C-type)

2.55.1 **Command Form**

G93;

G01…F_;

G02…F_;

G03…F_;

2.55.2 **Description**

This command is a feedrate mode command. It is used to specify the current format for the feedrate. The mode remains once specified in the program, and will not be canceled until G98/G99 is specified.

This mode only affects the feed rate of G01, G02, G03.

In G93 mode, F only affects the feedrate of its own block, so each block needs to have its own F argument. Otherwise system will trigger Cor 85: "F argument in G93 mode is wrong" alarm.

G01 block in G93 mode, its feedrate is defined as: F * block length

G02 / G03 block In G93 mode, its feedrate is defined as: F * block radius

2.55.3 **Example**

```
G21
G93
G01 X10. F1          // feed rate in this block 1*10 = 10 mm/min
G02 X20. R5. F3      // feed rate in this block 3*5 = 15 mm/min
G03 X0 R10. F5       // feed rate in this block 5*10 = 50 mm/min
M30
```

2.56 G94/G95- Feed Unit Setup

2.56.1 **Command Form**

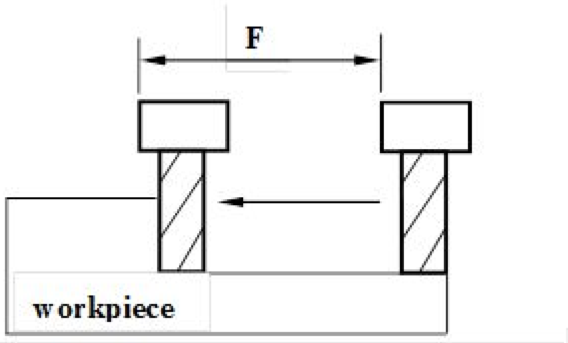
G94 F_;

G95 F_;

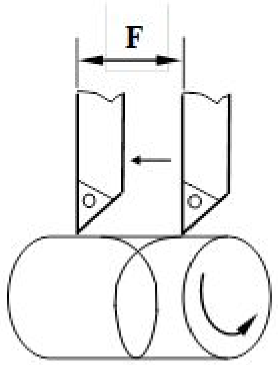
2.56.2 **Description**

This command set up the feed unit of F_function (tool moving distance per unit time/per revolution). G94 is feed per minute, unit: mm/min, inch/min; G95 is feed per revolution, unit: mm/rev, inch/rev.

2.56.3 **Figure**



G94. feed per minute(mm/min or inch/min)



G95. feed per revolution(mm/rev or inch/rev)

2.57 G96/G97- Constant Surface Speed Control (C-Type)

2.57.1 **Command Form**

G96 S__ [P__]; constant linear surface speed control: ON. Specifies an axis name or a geometry axis as the reference axis for calculation.

S : Constant surface cutting speed

P : Reference axis for calculation in constant surface speed control

- P1 ~ P3 : Selects the reference axis by specifying a geometry axis.
    - P1 = Geometry X-axis
    - P2 = Geometry Y-axis
    - P3 = Geometry Z-axis
- P100 ~ P999 : Selects the reference axis by specifying an axis name.

G97 S_; constant linear surface speed control: OFF

S : Spindle speed command

2.57.2 **Description**

G96 command specifies the surface speed of the contact point between tool and workpiece; G97 cancels G96 command, and is also the function to set spindle speed. To control the surface speed while the diameter of the workpiece varies, G96 can be used to specify the constant surface speed. If a constant spindle rotate speed is fixed regardless the diameter of workpiece, G97 can be used to control spindle speed.

The surface speed S of G96 follows the formula shown below:

$V = \frac{\pi DN}{1000}$ (metric) or $V = \frac{\pi DN}{12}$ (imperial)

V: Surface speed, use G96 to specify a value. (unit: m/min or ft/min)
D: Effective diameter of workpiece. (unit: mm or inch)
N: spindle rotate speed, specified by G97. (unit: RPM)

2.57.3 **Example**

- Constant surface speed:

```
G50 S2000         //limit max. rotate speed of spindle by G50
G96 S130 M03     //cutting speed maintains to be130m/min
```

Notice:
G50 often be used with G96 to limit max. rotate speed of spindle. From the above example, spindle rotate speed of the workpiece with 10mm diameter is:

$$N = \frac{1000\times130}{\pi\times10} = 4140\text{rpm}$$

The max. rotate speed of spindle is limited to be no more than 2000 rpm by G50. Chuck failure due to the excessive centrifugal force caused by spindle over speeding is prevented. Therefore G50 is sometimes used with G96.

- Constant rotate speed

```
G97 S1300 M03    //spindle rotate speed maintains 1300 rev/min
```

2.57.4 **Precaution**

1. Starting from versions 10.120.42H, 10.120.49, 12.0.6 and later, the G96 function allows the reference axis for constant surface speed control to be specified through the P_ parameter.
2. When G96 is enabled, if the reference axis type is rotary axis, alarm **[COR-383 G96 reference axis selection error]** will be triggered.
3. When G96 is enabled, if the **P_** parameter specifies an axis name as the reference axis for constant surface speed control but the axis name does not exist or the axis does not belong to the coordinate, alarm **[COR-383 G96 reference axis selection error]** will be triggered.

4. When G96 is enabled, if the **P_** parameter is not an axis name and the **P_** value is less than 0 or greater than 3, alarm **[COR-383 G96 reference axis selection error]** will be triggered.
5. After G96 is enabled, the command G96 P_ can be reissued to change the reference axis. The system will then calculate the constant surface speed based on the newly specified reference axis.
6. After G96 is enabled, if the user redefines the geometry plane axis by using **G17**, or if parameters are set to associate geometry axes with rotary axes and the kinematic chain function is enabled, G96 will still reference the original geometry axis for surface speed calculation. The user must reissue G96 for the new geometry axis to be used as the reference axis.
7. 10.118.28D, 10.118.31 and earlier versions: When G97 is commanded after G96, G92 S_ spindle maximum speed limit will be canceled at the same time.
8. 10.118.28E, 10.118.32 ~ 10.118.30K, 10.118.32K, 10.118.40O, 10.118.41O, 10.118.48: G97 will not cancel G92 S_ spindle maximum speed limit.
9. 10.118.30L, 10.118.32L, 10.118.40P, 10.118.41P, 10.118.48A and later versions: When G97 is commanded after G96, G92 S_ spindle maximum speed limit will be canceled at the same time.
10. When G96 is commanded, the surface speed of spindle will be kept constant in G01、G02/G03. However, the surface speed will be calculated by end position of G00/G53 block, and it will not be kept in constant.

2.58 G114.1/G113- Enable/Disable Spindle Synchronization (C-type)

2.58.1 Command Form

1. Enable spindle synchronization function
   G114.1 [R_] [K_]
   - R: Angle difference (when R is not given, it refers to the speed synchronization, usually used for round bar material)
   - K: Synchronous group number 1~3. Multiple synchronization sets, up to 3 sets can be used simultaneously. When K is not given, default is the first synchronization. The valid version of multiple synchronization sets starts at 10.116.24M, 10.116.32 (include).
2. Disable spindle synchronization function
   G113 [K_]

2.58.2 Description

When two or more spindles are available in a lathe, some two-spindle cooperative movement can achieve special applications. Such as part transfer between two spindles, both spindles must have same rotation speed and their angle shift need to be equal or fixed. This is spindle synchronization.

For details can refer to spindle synchronization and polygon cutting.

2.58.3 Precaution

1. **Spindle Status:**
   a. When signal of spindle synchronization is on and pressing “Reset” button, system will disable G114.1 synchronization state only after two spindles have stopped.
   b. When signal of spindle synchronization is on and G113 is specified to disable spindle synchronization, system will immediately disable synchronization. (synchronization signal Off)
   c. The Base Spindle is prohibited from using the synchronous function in the position control mode (C63), and the Synchronous Spindle is not recommended to use the synchronous function in the position control mode.

2. **Multiple Sets of Synchronization:**
   a. Enable synchronous command (G114.1) can be repeated (but K is not repeatable).

b. A Base spindle (axis) can have multiple synchronous axis at the same time.
c. The synchronous axes can no longer be the base axis of other spindles. (COR102)
d. The value of System Data 45/46 is the angle difference between the base axis and the synchronous axis of the last synchronization command. It displays the angle difference of the second-to-last combination of the same period after the release, and so on.

2.58.4 **Example**

Take Spindle 1 to be Base Spindle and the Spindle 2 to be Synchronous as example. M103 and M203 are commands for “spindle CW”; M105 and M205 are for “spindle stop”; M81 is for “wait until synchronization success”. All M codes above needs to be defined in PLC.

Dual Program Example

$1

```
S1 = 150
M103// spindle 1 CW on.
G04 X0.4// wait spindle speed goal.
G114.1 R0.// enable spindle syncronization.
M81// wait spindle syncrhonization.
S1 = 200// change speed.
G04 X0.4
M105// stop spindle
G113// diable spindle syncronization.
G04.1 P1// wait sync. $2
M30// end.
```

$2

```
S2 = 100
M203// spindle 2 CW on.
G04.1 P1// wait sync. $1
M99// end.
```

Single Program Example

```
G114.1 R0.// enable spindle syncronization.
S1 = 150
M103// spindle 1 CW on.
S2 = 100
M203// spindle 2 CW on.
M81// wait spindle syncrhonization.
M105// stop spindle 1.
G113// diable spindle syncronization.
G04 X1.
M205 // stop spindle2
M30// end.
```

Single Program Example (Static Start)

```
M103 S1 = 0// spindle 1 CW on.
M203 S2 = 0// spindle 2 CW on.
G114.1 R0 // enable spindle syncronization.
M81// wait spindle syncrhonization.
S1 = 150// change spindle target speed.
G04 X3.
M105 // stop spindle 1.
M205 // stop spindle2
G113// diable spindle syncronization.
M30// end.
```

Note: After version10.116.1, system will automatically wait for synchronization finish, no more M code (M81) needed.

Single Program Example (Simultaneous Multiple Sets of Synchronization)

Scenario:
Pr4021 = 1 (K1: spindle 1)
Pr4022 = 2 (K1: spindle 2) // the spindles 1 & 2 are synchronized in other processing areas.
Pr4023 = 3 (K2: spindle 3)
Pr4024 = 4 (K2: spindle 4) // the spindle 3 & 4 clamp the workpiece while rotating
Pr4025 = 3 (K3: spindle 3)
Pr4026 = 5 (K3: spindle 5) // the spindle 5 follows spindle 3 for polygon cutting

```
M03 S1000 // spindle 1 CW on
M203 S2=1500 // spindle 2 CW on
M303 S3=2000 // spindle 3 CW on
M403 S4=300 // spindle 4 CW on
M503 S5=100 // spindle 5 CW on
G04 X3. // wait

G114.1 K1 // enable 1st spindle syncronization
G04 X3. // wait
G114.1 R90 K2 // enable 2nd spindle syncronization
G04 X3. // wait
G51.2 P1 Q2 R60 K3 // enable 3rd spindle syncronization
G04 X3. // wait
S1500 // change spindle target speed
G04 X3. // wait
S500 // change spindle target speed
G04 X3. // wait

G113 K2 // diable 2nd spindle syncronization
G50.2 K3 // diable 3rd spindle syncronization
G113 K1 // diable 1st spindle syncronization
G04 X3. // wait

M05 // stop spindle 1
M205 // stop spindle 2
```

```
M305 // stop spindle 3
M405 // stop spindle 4
M505 // stop spindle 5
M30 // end
```

2.59 G114.3/G113- Enable/Disable Spindle Superimposition Function (C-Type)

2.59.1 **Introduction**

If there are more than two spindles on the machine, and the speed of the synchronous spindle is to be superimposed on the basic spindle, the spindle superimposition function can be used.

To use the two-spindle superimposition function in the case of two spindles tapping: Spindle 1 needs to maintain spinning, and Spindle 2 can be borne by Spindle 1 for tapping. At this time, the speed of spindle 2 = Spindle 2 speed command + Spindle 2 superimposition speed, wherein the superimposition speed of Spindle 2 is the command speed of spindle 1. This is the spindle loading function. We call spindle 1 the basic spindle and spindle 2 the synchronous spindle.

2.59.2 **Operation Method**

Command Form

1. Enable spindle superimposition function
   G114.3
2. Disable spindle loading function
   G113

Superimposition completion signal

1. When the superimposition completion signal is on, it indicates that the Synchronous Spindle is completely borne on the Basic spindle. At this time, the actual speed of Synchronous Spindle is the superimposition speed of Synchronous Spindle + Synchronous Spindle command speed. (The Synchronous Spindle's loading speed = the Basic spindle speed)

User Parameter	Superimposition Completion Signal
Basic Spindle number: Pr4021 Synchronous Spindle number: Pr4022	S60

2.59.3 **Precaution**

1. **Hardware setting:**
   a. Due to the hardware limitation, the adjacent two ports have time difference of 8 us when reading the encoder feedback; and the difference increases when the the two ports are further apart. This phenomenon does not affect the use of the spindle function, but there is a synchronization need for superimposition tapping, the time difference will cause a phase reading error. When using the superimposition tapping function, the two spindles must be connected to adjacent hardware ports

(i.e. P1 & P2 of the same axis card) in order to reduce the phase error caused by time difference, which may leads to the material transfer failure.

2. **Parameter setting:**
   a. The synchronous spindle of the superimposition function must be a **position control servo motor**. The synchronous spindle type in superimposition function only supports Type 1 and Type 3 (Pr1791 to 1796). Type 3 synchronous tapping is recommended.
      - if the spindle type is set incorrectly, alarm COR-93【Spindle synchronization, spindle type error】will show up.

   b. When using the following tapping (spindle Type 1), notice that the resolution of the synchronous spindle must be an integral multiple of the basic spindle. If the encoder is mounted on the motor side, the gear ratio must be converted. That is, the resolution* (no. of the teeth on spindle / no. of teeth on the motor).
   c. When the superimposition function enabled, the direction of the synchronous spindle's superimposition command is determined by Pr1861~Pr1866.
   d. If the spindle superimposition signal On, but the basic and synchronous spindle (Pr4021~4026) do not exist, the alarms will be triggered (COR-091, COR-092).
   e. If the tapping action is performed in the spindle superimposition mode, it is recommended to set the Kp of the two spindles (Pr18x~ of each axis) and the acc/deceleration parameters to be identical or similar, so that the tracking error can be limited.
3. **Process action:**
   a. When the E-Stop is pressed, the spindle rotation and spindle superimposition will be stopped; after releasing E-Stop, the basic and the synchronous spindle will only restore the original S code speed without the superimposition function.
   b. If superimposition completion signal On, pressing Reset will cancel the G114.3 superimposition state.
   c. If superimposition completion signal On and executes G113 to disable superimposition, system wait for spindles to reach target speed then actually disable the loading state.
   d. The spindle superimposition function does not support phase synchronization, only speed superimposition.
   e. After the synchronization is completed, position command is prohibited on the basic spindle to avoid unexpected error.
4. **Alarm list:**
   a. COR091【Spindle synchronization, basic spindle number error】
      Reason: Pr4021~Pr4026 is incorrectly set or the specified spindles do not exist.

   b. COR092【Spindle synchronization, synchronous spindle number error】
      Reason: Pr4021~Pr4026 is incorrectly set or the specified spindles do not exist.

   c. COR093【Spindle synchronization, spindle type error】
      Reason: Pr1791~Pr1796 is incorrectly set

   d. COR094【The spindle tapping command speed over limit during spindle superimposition】
      Reason: The speed of driven tool exceeds the maximum value of the spindle speed.

2.59.4 **Example**

Take Spindle 1 as the basic spindle and Spindle 2 as the synchronous spindle. M103 and M203 are spindle CW, M105 and M205 are spindle stop, and M82 is waiting synchronization completion signal. All M codes above needs to be defined in PLC. If using following tapping, parameters must comply with Precaution 2:

- Parameter setting example 1:

Parameter number	Basic Spindle	Synchronous Spindle

1651~1660 (Pulse number of the spindle motor encoder)	1024	2048
1661~1670 (Spindle feedback multiplier)	4	4
1681~1700 (Spindle first gear screw side teeth / motor side teeth number)	3 (screw side teeth) 4 (motor side teeth)	6 (screw side teeth) 8 (motor side teeth)
1811~1820 (Spindle encoder mounting position)	1	1

Synchronous spindle resolution= 2048*4*(6/8) = 6144
Basic spindle resolution= 1024*4*(3/4) = 3072
(The encoder is mounted on the motor side and requires a gear ratio.)

- Parameter seeing example 2:

Parameter number	Basic spindle	Synchronous spindle
1651~1660 (Pulse number of the spindle motor encoder)	1024	2048
1661~1670 (Spindle feedback multiplier)	4	4
1811~1820 (Spindle encoder mounting position)	0	0

Synchronous spindle resolution= 2048*4 = 8192
Basic spindle resolution= 1024*4 = 2048
(The encoder is mounted on the motor side, need no gear ratio.)

Duel program example

$1

```
G10 L1000 P791 R1// set active spindle.
M103 S1 = 2000// spindle 1 CW on.
G04 X0.4 // wait spindle speed goal.
G114.3// enable spindle superimposition.
M82 // wait spindle superimposition.
G04.1 P1 // wait sync. $2
G04.1 P2 // wait sync. $2
M105 // stop spindle
G113 // disable spindle superimposition.
G04.1 P3 // wait sync. $2
M30
```

$2

```
G10 L1000 P792 R2
// set active spindle.
G04.1 P1 // wait sync. $1
G84 Z-20. R0 F1.// do tapping.
G80// end tapping.
G04.1 P2 // wait sync. $1
G04.1 P3 // wait sync. $1
M99
```

Single program example (superimposition and tapping)

```
G10 L1000 P791 R2// set active spindle.
M103 S1=2000// spindle 1 CW on.
G04 X0.4 // wait spindle speed goal.
G114.3// enable spindle superimposition.
M82 // wait spindle superimposition.
G84 Z-20. R0 F1.// do tapping.
G80// end tapping.
G113 // disable spindle superimposition.
M30
```

Single program example (start from stop)

```
G10 L1000 P791 R2// set active spindle.
M103 S1 = 0 // spindle 1 CW on.
M203 S2 = 0 // spindle 2 CW on.
G114.3 // enable spindle superimposition.
M82 // wait spindle superimposition.
S1 = 2000 // change spindle target speed.
G04 X1.
```

```
G84 Z-20. R0 F1. S1000 // do tapping.
G80// end tapping.
M105 // stop spindle 1.
M205 // stop spindle2
G113 // disable spindle superimposition.
M30 // end.
```

2.59.5 Reference

Device Type	Device	Description
Register	R761~R776	Display corresponding spindle mechanical position, unit 0.001 degree. (current)
S (superimposition completion signal)		S60
Parameter	181~196	Servo loop gain of the axis (Kp)(1/sec)
	1731~1736	Minimum spindle speed
	1791~1796	Spindle type
	1831~1836	Acceleration and deceleration time of the spindle to 1000 RPM(ms)
	1851~1856	Jerk time of acceleration of the spindle to 1000 RPM/Sec (ms)
	1861~1866	Rotate direction, 0: CW， 1: CCW
	4021~4026	Basic/synchronous spindle number(1~6)
Alarm	Cor091	Spindle synchronization, basic spindle number error
	Cor092	Spindle synchronization, synchronous spindle number error
	Cor093	Spindle synchronization, spindle type error

Device Type	Device	Description
	Cor094	The spindle tapping command speed over limit during spindle superimposition
	Cor102	Spindle synchronization, repeat or conflict synchronization or superimposition command
	Cor142	Spindle synchronization, K argument input error

2.60 Spindle Rotate Speed Function: S Code Command (C-Type)

2.60.1 **Command Form**

S__

Description

S code is the spindle speed command, specifying constant revolution per minute or constant surface speed per minute of spindle under mode G96/G97.

2.60.2 **Notes**

When the machining spindle of the axis group is switching between different spindles and the current machining spindle is Spindle 2, to specify Spindle 1 to rotate CW 150 rpm, 'M03 S1=150' should be used in order to avoid the speed given to the Spindle 2 due to insufficient time for spindle switching.

Example

```
G96 S150 M03     //constant surface speed of spindle,150 m/min.
G97 S500 M03     //spindle maintains 500 rev/min
```

2.61 Tool Compensation Function: T Code Command (C-Type)

2.61.1 **Command Form**

T _*_*_ (two-digit form)

T _*_*_ _*_*_ (four-digit form)

2.61.2 **Description**

Tool compensation function, also called T function, is mainly for selecting tools. Machine automatically switch to the the assigned tool number.

Two-digit form: Specify tool number, tool length compensation, and wear compensation.

Four-digit form: The first two digits specify tool number, the other two digits specify tool length and wear compensation.

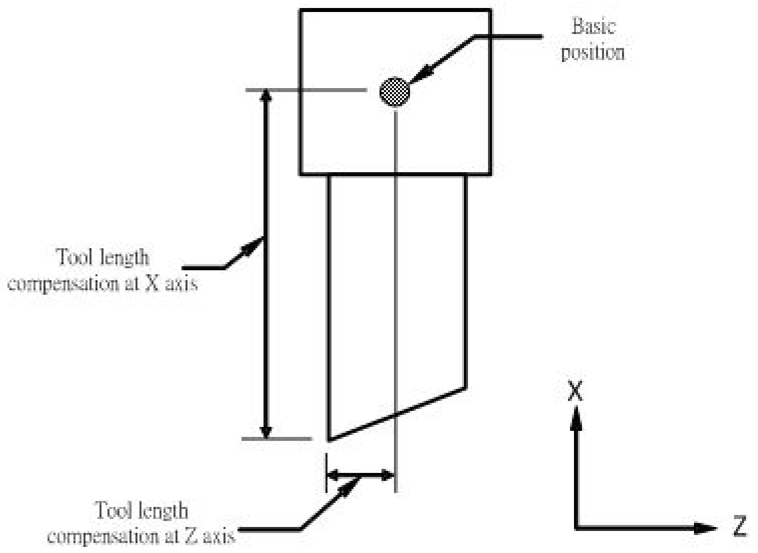
When an user executes T__ command, the compensation value is selected but the no action in the block. The compensation value is not applied until a block with movement is performed.

2.61.3 **Method of Tool Length Compensation**

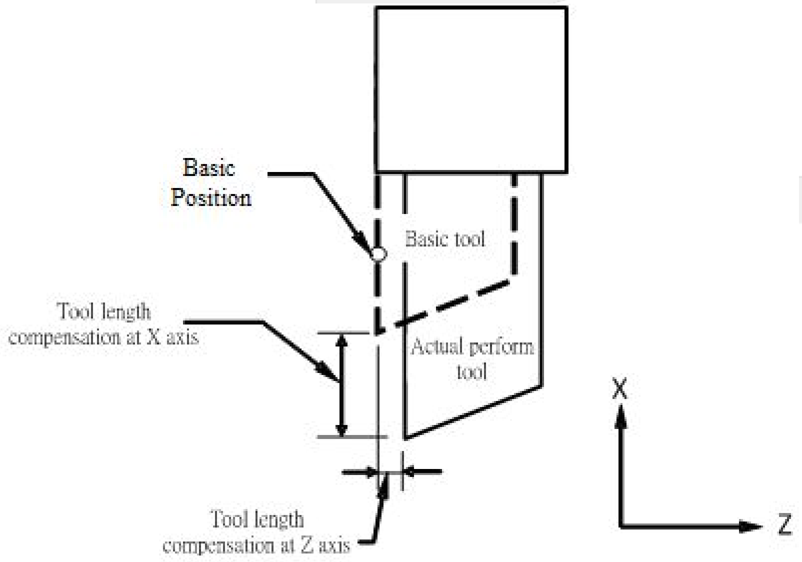
Apply tool length compensation base on the reference point of a tool.

The Basic Position of the program is generally the **center of the tool table** or the **tool nose of the reference tool**:

Center of the tool table


Basic position
Tool length compensation at X axis
Tool length compensation at Z axis
X
Z

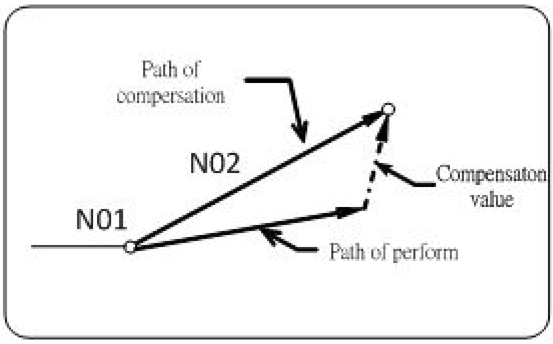
Tool nose of Reference Tool

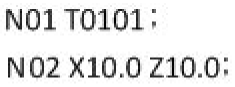

Basic Position
Basic tool
Tool length compensation at X axis
Actual perform tool
Tool length compensation at Z axis
X
Z

2.61.4 **Principle of Tool Length Compensation**

Tool compensation starts

Tool compensation action starts after executing T command and then executing movement command.

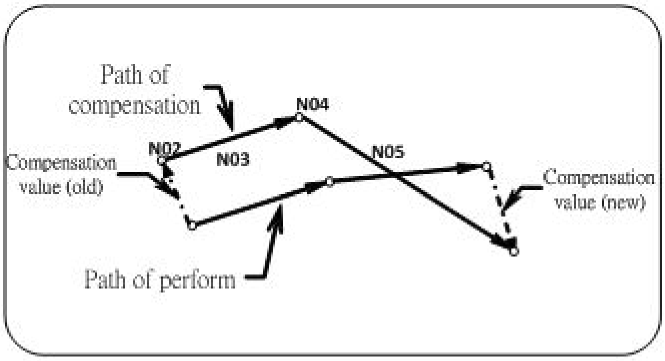


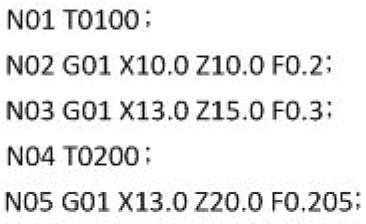


```
N01 T0101；
N02 X10.0 Z10.0;
```

Tool length compensation number change

When tool number changes, the total movement is the new tool compensation plus the program movement.

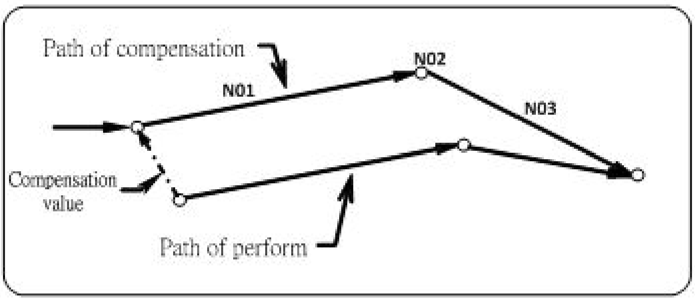


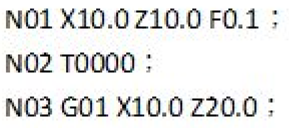


```
N01 T0100；
N02 G01 X10.0 Z10.0 F0.2;
N03 G01 X13.0 Z15.0 F0.3;
N04 T0200；
N05 G01 X13.0 Z20.0 F0.205;
```

Tool length compensation cancel

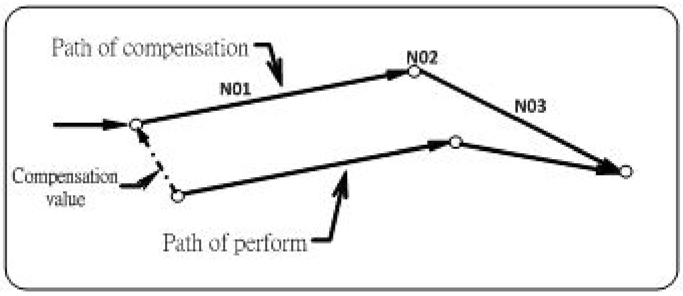
Tool compensation number is 0.
When number of compensation is “0”in T command, compensation cancels.

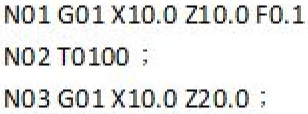




```
N01 X10.0 Z10.0 F0.1 ；
N02 T0000 ；
N03 G01 X10.0 Z20.0 ；
```

Compensation value of command is “0”
When compensation value of the tool number is “0”in T function, the compensation cancels.



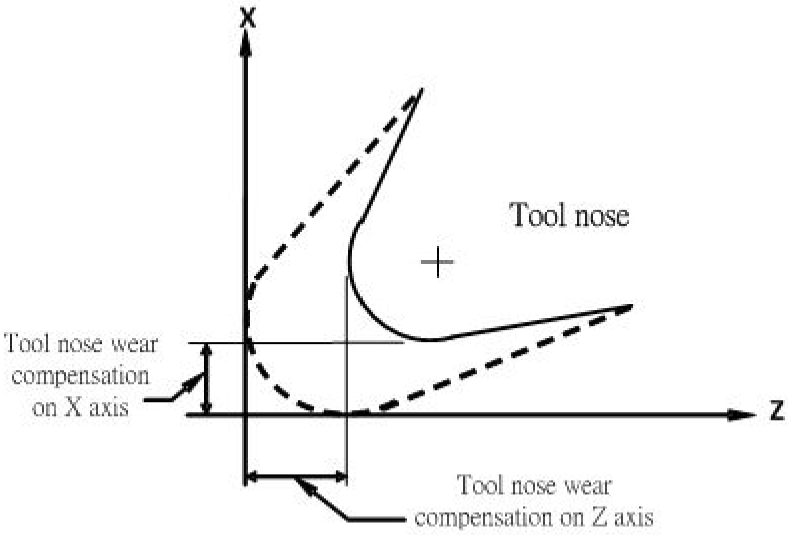


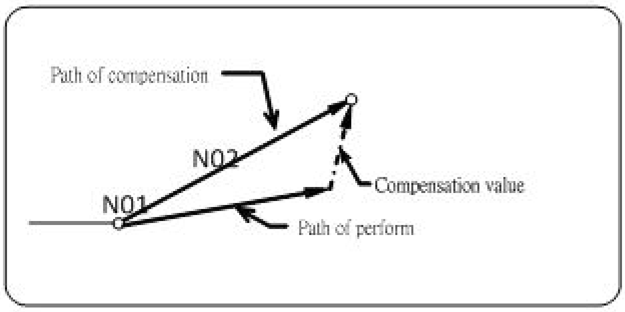
```
N01 G01 X10.0 Z10.0 F0.1
N02 T0100 ；
N03 G01 X10.0 Z20.0 ；
```

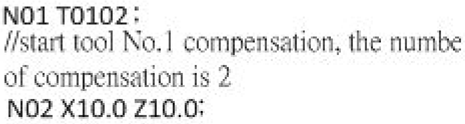
2.61.5 **Tool Nose Wear Compensation**

Tool nose wear compensation value setting

System can perform compensation when tool nose wears, the value will be added into geometric compensation. **Geometric compensation = tool length compensation + wear compensation**. When specifying the number of compensation, geometric compensation will be applied.


X
Tool nose
Tool nose wear compensation on X axis
Z
Tool nose wear compensation on Z axis





```
N01 T0102；
//start tool No.1 compensation, the numbe
of compensation is 2
 N02 X10.0 Z10.0;
```

2.62 Decimal Point Input (C-Type)

2.62.1 Description

The input value with decimal point will be considered as general units such as mm, inch, sec, etc.

If the input value without decimal point will be considered as the minimum units set in the system such as um, ms, etc.

2.62.2 Precaution

When input value is integer, its unit can be modified by Pr3241. Refer to the Parameter Manual for details.

2.62.3 Example

Decimal Point Type:
10**.00** means 10mm

Integer Type:
1000 means 1000um

2.63 Chamfer, Round Corner, Angle Command (C, R, A) (C-Type)

2.63.1 Introduction

In the mechanical drawing, we can input the angle of straight lines, chamfer, round corner (fillet), and other geometry values directly by using the following functions. The system will insert the fillet and chamfer values in the straight lines under enough corner space.

2.63.2 Basic Function- Chamfer C/Round Corner R/Straight Line Angle A

In a continuous blocks of straight line or arc, if ",C__" or ",R__" are attached at the end of a block, the single corner will automatically perform the cutting of the chamfer C and the round corner R.

Or when the only limited conditions are known: the angle between the next block path and the horizontal axis, and the end point coordinate of one of the axial directions (one of the X/Z axes). In this case, use the linear angle function ", A_" to find the full path.

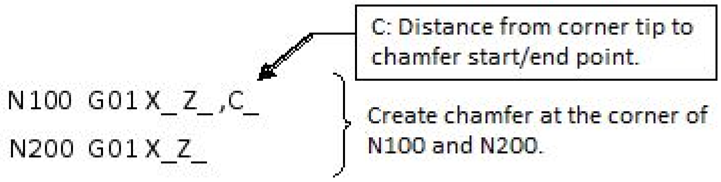
The feedrate of the chamfer C or round corner R can be specified by E__ , and if E__ is not specified, it is preset as the feedrate of the next block.

Compared with the chamfer C/round corner R, the feed rate of the straight line angle A can only be specified by F__ , same as G01.

Chamfer C / round corner R / straight line angle A can be used for absolute or incremental value commands.

2.63.3 Basic Function-Command Form

Chamfer C
In the first block of two continuous blocks (excluding arc), specify “**,C_**” command could execute corner chamfering.



```
N100 G01 X_ Z_ ,C_
N200 G01 X_ Z_
```

C: Distance from corner tip to chamfer start/end point.

Create chamfer at the corner of N100 and N200.

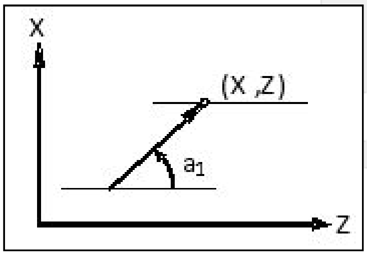
Round Corner R

In the first single block of two continuous blocks (including arc), specify “**,R_**” command could execute round cornering R.

,R ;

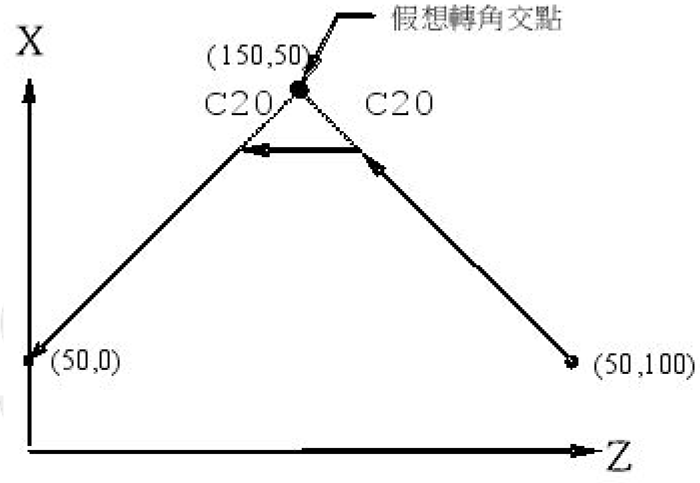
R: Represents the round corner (fillet), the radius of the arc.

Straight Line Angle A
G01 Z_ (X_) ,A_; //specify angle or the coordinate of X axis or Z axis



2.63.4 Basic Function-Example

Chamfer Between Two Straight Line

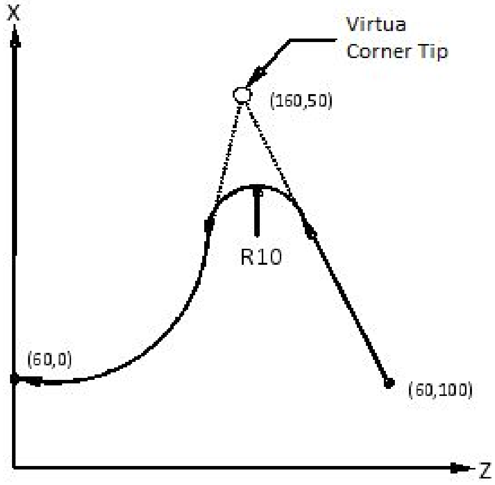


1. Absolute command：
```
G28 X0.0 Z0.0;                  //Return to machine zero through mid-point
G00 X50.0 Z100.0;
G01 X150.0 Z50.0 F0.1 ,C20.0;   //moving path of two single blocks
G01 X50. Z0;                    //Corner cutting with chamfer of C20.0
```

2. Incremental command：

```
G28 X0.0 Z0.0;                  //Return to machine zero through mid-point
G00 U50.0 W100.0;
G01 U100.0 W-50.0 F0.1 ,C20.0; //moving path of two single blocks
G01 U-100.0 W-50.0;             //Corner cutting with chamfer of C20.0
```

Corner Between Straight Line and Arc



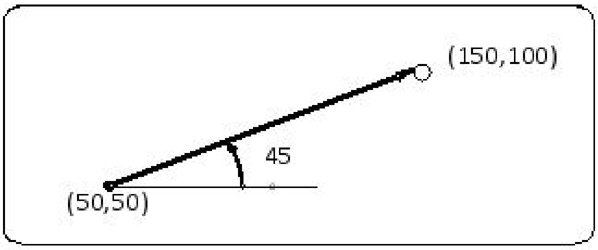
1. Absolute command

```
G28 X0.0 Z0.0;                  //Return to machine zero through mid-point
G00 X60.0 Z100.0;
G01 X160.0 Z50.0 F0.1 ,R10.0; //moving path of two single blocks
G02 X60.0 Z0.0 I0.0 K-50.0;    //Corner cutting with arc angle of R10.0
```

2. Incremental command

```
G28 X0.0 Z0.0;                   //Return to machine zero through mid-point
G00 U60.0 Z100.0;
G01 U100.0 W-50.0 F0.1 ,R10.0;  //moving path of two single blocks
G02 U-100.0 W-50.0 I0.0 K-50.0; //Corner cutting with arc angle of R10.0
```

Straight Line Angle



```
N01 G00 X50.0 Z50.0;       //Rapid position to specified position
N02 G01 Z100.0, A45.0;     //Tool path is 45° from horizontal axis
                           // Path end Z position is 100
```

After executing program→ Path end Z position is 150.

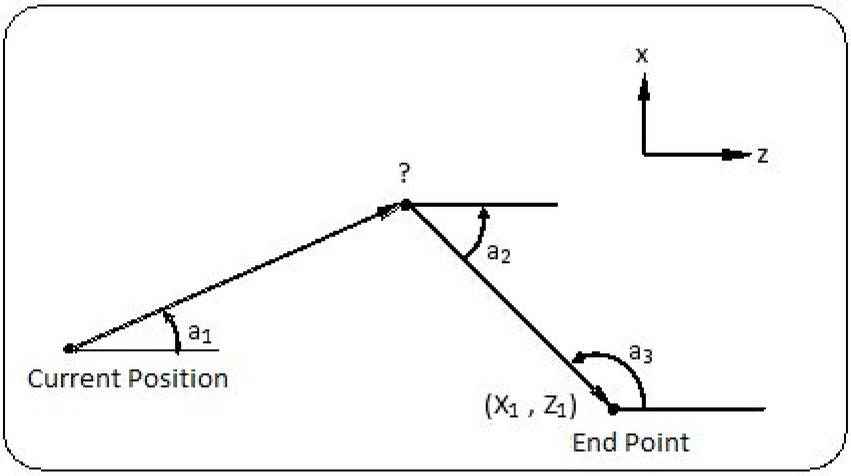
2.63.5 **Advanced Function-Geometric Function Command**

When it is hard to get the intersection point of two lines in continuous linear interpolation, we can use the sloping angle of line 1, the absolute coordinate of line 2, and the sloping angle of line 2 as commands for controller to calculate the end of the first line. Therefore continuous straight line corner function can be executed.

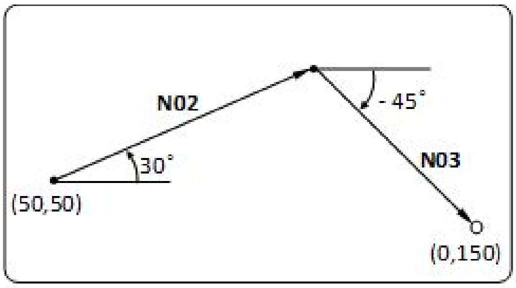
2.63.6 **Advanced Function-Command Form**

Format 1

G01 ,A__ F__; //specify angle
X___ Z___ ,A___; //specified the end coordinate value and the angle of the next block



Format 1 Example



N01 G00 X50.0 Z50.0;   //Rapid position to specified point
N02 G01 ,A30.0 F0.3   // Line 1 is 30° from horizontal axis
N03 X0.0 Z150.0 ,A45.0; // Line 2 is 45° from horizontal axis , and its end point is (0, 150.)

After executing program→ The intersection is (104.904, 97.548)

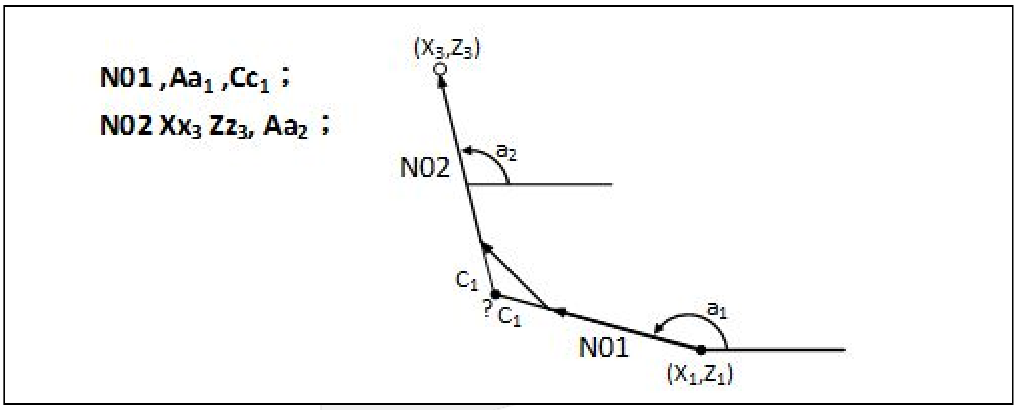
Format 1 Precaution

1. This function is effective only under G01. Any other interpolation or positioning command are ineffective.
2. Angle is between the horizontal axis in selected plane and path direction, the angle is positive for CCW and negative for CW.
3. The sloping angle can be specified in start point side or end point side, NC can automatically determine the angle is specified in which side.

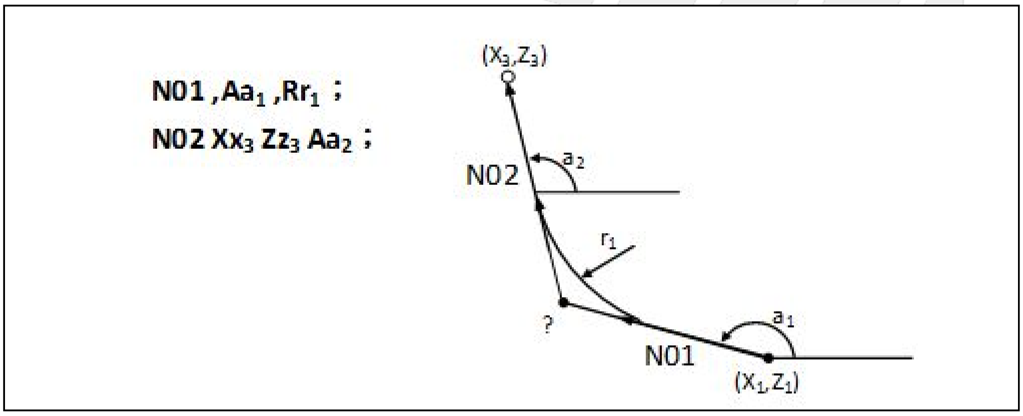
4. To use the second way to specify, we need to specify the end point of the second block to be absolute coordinate.

Format 2

In the first block's angle command, we can specify **Chamfer C** command or **Round Corner R** command.



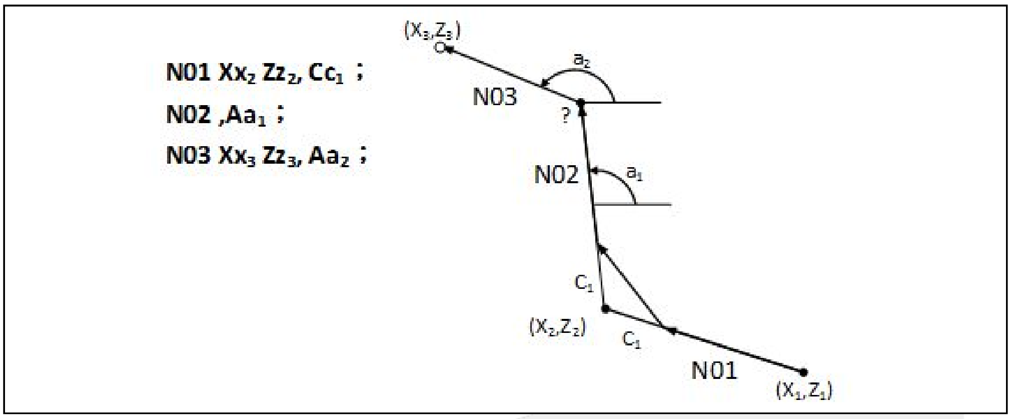
Description:
Given command of Target Position (X$_3$, Z$_3$), angle of two other paths to horizontal direction "a$_1$" and "a$_2$", plus the chamfer of two path intersection "C$_1$". Controller use given values to calculate unknown position of intersection "?", and move tool along the path to reach (X$_3$, Z$_3$).



Description:
Given command of Target Position (X$_3$, Z$_3$), angle of two other paths to horizontal direction "a$_1$" and "a$_2$", plus the round corner of two path intersection "r$_1$". Controller use given values to calculate unknown position of intersection "?", and move tool along the path to reach (X$_3$, Z$_3$).

Format 3

After Chamfer C and round angle R command, linear angle command can be used.

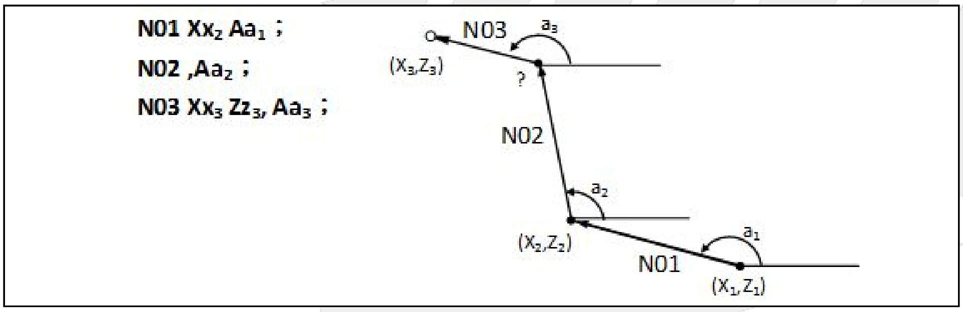


```
N01 Xx2 Zz2, Cc1 ;
N02 ,Aa1 ;
N03 Xx3 Zz3, Aa2 ;
```

Description:
Given command to specified position (X_2, Z_2)→ (X_3, Z_3) , the first two path form chamfer "C1", the angle of two paths to horizontal direction "a_1" & "a_2". Controller use given values to calculate unknown position of intersection "?", and move tool along the three paths to reach (X_3, Z_3).

Format 4

After linear angle command, we can continue to do linear angle command.



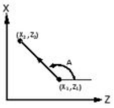
```
N01 Xx2 Aa1 ;
N02 ,Aa2 ;
N03 Xx3 Zz3, Aa3 ;
```

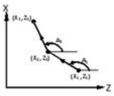
Description:
Given command of first paths X position "X_2", its angle to horizontal direction "a_1", the specified position
(X_3, Z_3), the angle of two paths to horizontal direction "a_1" & "a_2". Controller use given values to calculate unknown position of intersection "?", and move tool along the three paths to reach (X_3, Z_3).

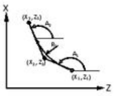
Format 2~4 Precaution

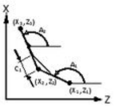
1. Round angle value cannot be inserted in threading area.
2. By directly entering the continuous command in next block according to the drawing size, the end point of previous block is then specified. Block stop cannot be executed in this block, but program pause can be executed in the previous block.
3. Allowance range of angle computing is ±1∘.
   (0). X_ , A_ ; (when the angle is 0∘±1 , 180∘±1 , the alarm will be triggered.)
   (1). Z_ , A_ ; (when the angle is 90∘±1 , 270∘±1 , the alarm will be triggered.)
4. If the angle between two lines is in between of ±1∘, alarm will be triggered when calculating intersection.
5. If the angle between two lines is in between of ±1∘, chamfer and round triggered.)
6. (1). Z_ , A_ ; (when the angle is 90∘+1 , 270∘+1 , the alarm will be triggered.)
7. If the angle between two lines is in between of +1∘, alarm will be triggered when calculating intersection.
8. If the angle between two lines is in between of +1∘, chamfer and round corner will be ignored.

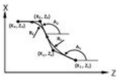
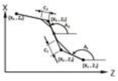
2.63.7 **Advanced Function-Geometric Function Usage Table**

	Command	**Movement**	**Description**
1.	X$_2$_ (Z$_2$)_, A_;		Base on the setting of any coordinate value of X$_2$ (or Z$_2$) and the angle "A" between the moving path and the horizontal axis, controller can calculate the unknown Z$_2$ (or X$_2$). The tool cuts along this path to the specified position (X$_2$,Z$_2$).

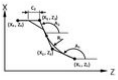
2.	,A$_1$_;
	X$_3$_ Z$_3$_, A$_2$_;
		 X
		(X$_3$,Z$_3$)
		A$_2$
		(X$_2$,Z$_2$)
		A$_1$
		(X$_1$,Z$_1$)
		Z
			Base on the setting of any coordinate value of X$_3$ (or Z$_3$) and the angle "A$_1$" & "A$_2$" between the moving paths and the horizontal axis, controller can calculate the unknown (X$_2$, Z$_2$). The tool cuts along this path to the specified position (X$_3$,Z$_3$).

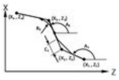
3.	X$_2$_ Z$_2$_,R$_1$_ ;
	X$_3$_ Z$_3$_;
	Or
	,A$_1$_, R$_1$_;
	X$_3$_ Z$_3$_, A$_2$_;
		 X
		(X$_3$,Z$_3$)
		A$_2$
		R$_1$
		(X$_2$,Z$_2$)
		A$_1$
		(X$_1$,Z$_1$)
		Z
			Base on the command to reach specified point (X$_3$,Z$_3$), and the specified angle "A$_1$" & "A$_2$" between each path and horizontal axis, and the intersection round corner "R$_1$", controller can calculate unknown intersection (X$_2$, Z$_2$), and tool will cut to specified point (X$_3$,Z$_3$) along the two paths.

4.	X$_2$_ Z$_2$_,C$_1$_ ;
	X$_3$_ Z$_3$_;
	Or
	,A$_1$_, C$_1$_;
	X$_3$_ Z$_3$_, A$_2$_;
		 X
		(X$_3$,Z$_3$)
		A$_2$
		C$_1$
		(X$_2$,Z$_2$)
		A$_1$
		(X$_1$,Z$_1$)
		Z
			Base on the command to reach specified point (X$_3$,Z$_3$), and the specified angle "A$_1$" & "A$_2$" between each path and horizontal axis, and the intersection chamfer "C$_1$", controller can calculate unknown intersection (X$_2$, Z$_2$), and tool will cut to specified point (X$_3$,Z$_3$) along the two paths.

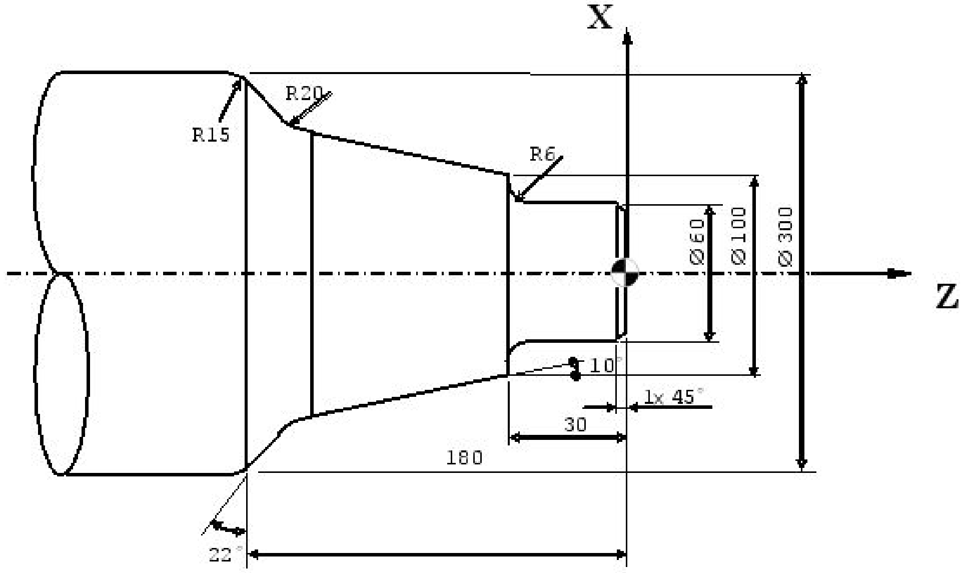
	Command	Movement	Description
5.	X_2_ Z_2_,R_1_; X_3_ Z_3_, R_2_; X_4_ Z_4_; Or ,A_1_, R_1_; X_3 Z_3_, A_2_,R_2_; X_4_ Z_4_;		Base on the command to reach specified point (X_2,Z_2) -> (X_3,Z_3) -> (X_4,Z_4), and the round corner between the first two paths "R_1", and the last two paths "R_2". (Or not the specifying (X_2,Z_2) plus the angle between the two previous moving paths and the horizontal axis "A_1" & "A_2") Controller uses the set value to calculate the angles of the two moving paths from the horizontal axis "A_1" & "A_2", or the unknown intersection (X_2 , Z_2), and the tool cuts along the three paths to arrive the end position (X_4,Z_4).
6.	X_2_ Z_2_,C_1_; X_3_ Z_3_, C_2_; X_4_ Z_4_; Or ,A_1_, C_1_; X_3_ Z_3_, A_2_,C_2_; X_4_ Z_4_;		Base on the command to reach specified point (X_2,Z_2) -> (X_3,Z_3) -> (X_4,Z_4), and the chamfer between the first two paths "C_1", and the last two paths "C_2". (Or not the specifying (X_2,Z_2) plus the angle between the two previous moving paths and the horizontal axis "A_1" & "A_2") Controller uses the set values to calculate the angle of the two paths and the horizontal axis "A_1" & "A_2", or the unknown intersection point (X_2 , Z_2), and the tool cuts along the three paths to arrive end position (X_4,Z_4).

	Command	Movement	Description

7.	X$_2$_ Z$_2$_,R$_1$_;
	X$_3$_ Z$_3$_, C$_2$_;
	X$_4$_ Z$_4$_;
	Or
	,A$_1$_, R$_1$_;
	X$_3$_ Z$_3$_, A$_2$_,C$_2$_;
	X$_4$_ Z$_4$_;
			Base on the command to reach specified point (X$_2$,Z$_2$) -> (X$_3$,Z$_3$) -> (X$_4$,Z$_4$), and the round corner between the first two paths "R$_1$", and the chamfer of last two paths "C$_2$".
			(Or not the specifying (X$_2$,Z$_2$) plus the angle between the two previous moving paths and the horizontal axis "A$_1$" & "A$_2$")
			Controller uses the set values to calculate the angle of the two paths and the horizontal axis "A$_1$" & "A$_2$", or the unknown intersection point (X$_2$, Z$_2$), and the tool cuts along the three paths to arrive end position (X$_4$,Z$_4$).

8.	X$_2$_ Z$_2$_,C$_1$_;
	X$_3$_ Z$_3$_, R$_2$_;
	X$_4$_ Z$_4$_;
	Or
	,A$_1$_, C$_1$_;
	X$_3$_ Z$_3$_, A$_2$_,R$_2$_;
	X$_4$_ Z$_4$_;
			Base on the command to reach specified point (X$_2$,Z$_2$) -> (X$_3$,Z$_3$) -> (X$_4$,Z$_4$), and the round corner between the first two paths "C$_1$", and the chamfer of last two paths "R$_2$".
			(Or not the specifying (X$_2$,Z$_2$) plus the angle between the two previous moving paths and the horizontal axis "A$_1$" & "A$_2$")
			Controller uses the set values to calculate the angle of the two paths and the horizontal axis "A$_1$" & "A$_2$", or the unknown intersection point (X$_2$, Z$_2$), and the tool cuts along the three paths to arrive end position (X$_4$,Z$_4$).

2.63.8 **Example**



(Diameter Positioning in Metric system)

G01 X60.0 A90.0, C1.0 F0.08; //linear interpolation, the angle between the line and horizontal axis is "+90°", and chamfering C1.0 at the next block, feedrate 80mm/rev.

Z-30.0, A180.0, R6.0; //linear interpolation, the angle between the line and horizontal axis is "+180°", and round corner R6.0 at the next block

X100.0, A90.0; //linear interpolation, cutting to specified point, the angle between the straight line and horizontal axis is "+90°"

,A170.0, R20.0; //linear interpolation, the angle between the straight line and horizontal axis is "+170°", and round corner R20.0 at the next block, the end point is specified in the next block

X300.0 Z-180.0, A112.0, R15.0; //linear interpolation, the angle between the straight line and horizontal axis is "+112°", and round corner R15.0 at the next block

Z-230.0, A180.0; //linear interpolation, the angle between the straight line and horizontal axis is "+180°", cutting to specified position.

2.64 Feedrate Function: F Code Command (C-Type)

2.64.1 **Command Form**

F__

2.64.2 **Description**

In cutting mode, the specified movement speed of tool in the program is called feedrate. There are two methods of setting feedrate: feed per minute (G94) and feed per revolution (G95). In G94 mode, **F300** can be directly specified for the tool feedrate of **300 mm/min**; in G95 mode, **F0.5** means **0.5 mm/rev**.

2.64.3 **Notice**

For G95, the F parameter is affected by Pr3241. For G94, the F parameter is not affected by Pr3241.

Pr3241	Pr17	Movement Command	Actual Distance
0	1	G00 F100	1(mm/min)
0	2	G00 F100	0.1(mm/min)
0	3	G00 F100	0.01(mm/min)
1	-	G00 F100	100(mm/min)
-	-	G00 F100. (with decimal point)	100(mm/min)

The least input unit(LIU)

Unit of data	Pr17	Least input unit
mm deg. sec.	1	0.01
	2	0.001
	3	0.0001
inch	1	0.001
	2	0.0001
	3	0.00001

2.64.4 **Example**

G94 G01 X100.0 Y100.0 F300 //linear interpolation, feedrate 300 mm/min

G95 G01 X100.0 Y100.0 F0.5 //linear interpolation, feed rate 0.5 mm/rev

3 M Code Command Description (C-Type)

Auxiliary function is used to control the On and OFF of the machine function. The format is two- digit value after M letter; the M codes and their functions are described as follows:

M Function Table

M Code	Function
M00	Program Pause
M01	Optional Stop
M02	Program Ends and Return to First Line
M03	Spindle Rotates CW
M04	Spindle Rotates CCW
M05	Spindle Stop
M06	Tool Change
M08	Cutting Fluid ON
M09	Cutting Fluid OFF
M10	Chuck Clamp
M11	Chuck Unclamp
M19	Spindle Positioning, to stop and fix spindle at a preset position.
M30	Program Ends and Return to First Line
M96	Interrupting Subprogram Call Function ON（Depend on Pr3600）
M97	Interrupting Subprogram Call Function OFF（Depend on Pr3600）
M98	Call Subprogram
M99	Subprogram Return to Main Program

M198	Call External Subprogram

3.1 M00- Dwell (C-Type)

When M00 is executed by CNC, the spindle stops, the feed dwells, and the cutting liquid is off. The dwell enables an operator to inspect workpiece/tool dimensions, calibrate and make compensation of the workpiec. The “M00 signal button” on the panel is used to control whether a program should be dwelled or not.

3.2 M01- Optional dwell (C-Type)

The function of M01 is similar to M00. M01 is valid only when “optional stop button” turns ON, and the program is therefore dwelled. On the contrary, M01 is invalid while the button turns OFF.

3.3 M02- Program ends (C-Type)

M02 is the same as M30. M02 command is specified at the end of the program. When program executes M02, all actions will stop and execute Reset, and the memory will be reset and return to the beginning state of the program.

3.4 M03- Spindle rotates CW (C-Type)

M03 command spindle to rotate CW. When M03 is used in conjunction with S function, spindle is specified to rotate CW in a given speed. .

3.5 M04- Spindle rotates CCW (C-Type)

M04 command spindle to rotate CCW. Can use with S function, and command the spindle to rotate CCW in set speed.

3.6 M05- Spindle stops (C-Type)

M05 command spindle to stop. When shifting gears or changing the direction of a rotating axis is required, M05 is specified to stop the spindle first.

3.7 M06- Tool exchange (C-Type)

M06 is specified to execute tool exchange. Note that M06 include no tool choosing, thus it must be used in conjunction with T_function.

3.8 M08/M09- Cutting liquid ON/OFF (C-Type)

M08 is specified to turn cutting liquid ON; M09 is specified to turn cutting liquid OFF.

3.9 M19- Spindle locates and stops (C-Type)

3.9.1 This command can locate the spindle at specified corner.

3.10 M30- Program ends (C-Type)

M30 command is specified at the end of the program. When program executes M30, all actions will stop and execute Reset, and the memory will be reset and return to the beginning state of the program.

3.11 M96/M97- Interrupting Subprogram Call Function (C-type)

M96/M97: Interrupting Subprogram Call function

- Command Form
  Take Pr3600 = 96 for example
  a. M96 P_[I_] [Q_] [R_] [L_]: Start interrupting subprogram call function
     i. P argument

Argument Description	Specifies the subprogram number of the call when the interruption triggered.
Argument Unit	-
Argument Range	[1 ~ 9999]
Precaution	a. Subprogram name should start with 'O'. b. P argument cannot have filename extension, and the called subprogram cannot have filename extension either. Ex: When argument P1234 is executed, the subprogram O1234 is called instead of O1234.txt since these two files are treated as different files in kernel.

Input Example	a. Subprogram name O1111, use argument P1111. b. Subprogram name O1213, use argument P1213. c. Subprogram name O0001, use argument P1.

ii. I argument

Argument Description	Interruption signal source
Argument Unit	-
Argument Range	[1 ~ 3]
	1: the interrupted signal is R-bit
	2: the interrupted signal is I-bit
	3: the interrupted signal is A-bit

iii. Q argument

Argument Description	Interrupt signal number
Argument Unit	-
Argument Range	Varies according to I argument.
	I=1(R-bit): [0 ~ 65535][00 ~ 15]
	I=2(I-bit): [0 ~ 511]
	I=3(A-bit): [0 ~ 511]
Input Example	a. Interrupted signal R49, use argument I1 Q4900. b. Interrupted signal R51.1, use argument I1 Q5101. c. Interrupted signal R50.11, use argument I1 Q5011. d. Interrupted signal A350, use argument I3 Q350.

iv. R argument

Argument Description	Trigger method
Argument Unit	-

Argument Range	[0 ~ 1]
	0: Upper-edge triggered
	1: Lower-edge triggered

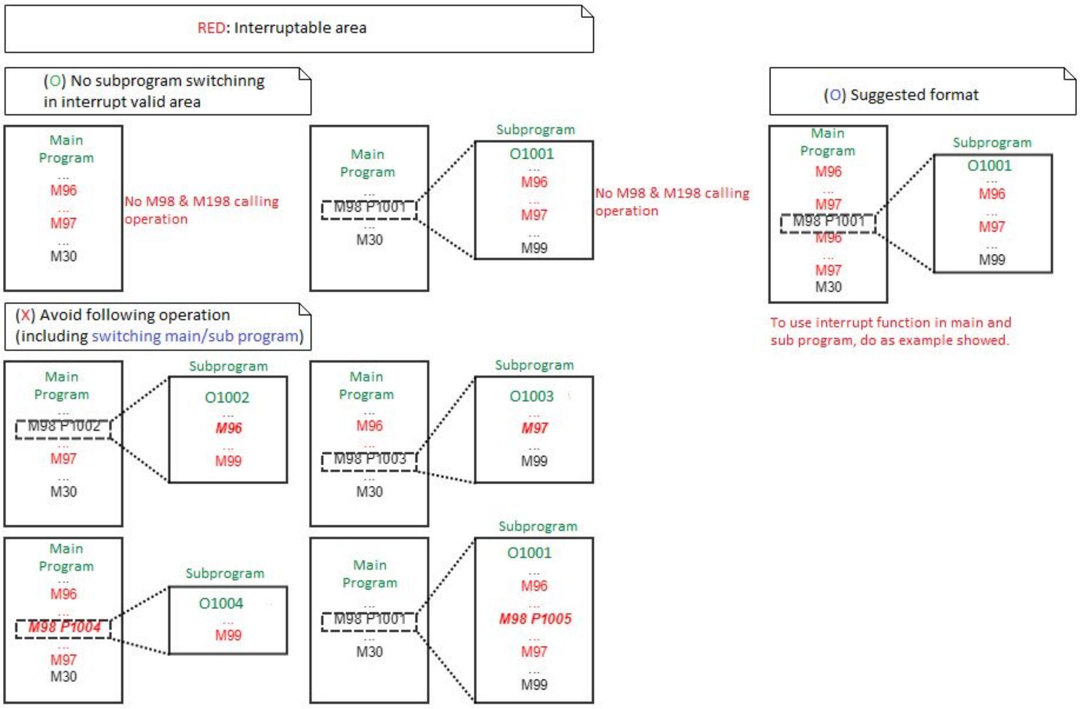
   v. L argument

Argument Description	Signal maintenance time
Argument Unit	ms
Argument Range	[0 ~ 2,147,483,647]

  b. M97: Close interrupting subprogram call function

- Trigger Signal
  a. If the trigger signal source is not specified, then the preset signal is C49. If the trigger signal source is specified, only the specified trigger is valid, and C49 is ignored.
  b. If M96 P__ is used and the trigger source is C49, the executing program stops immediately and calls the interrupting subprogram when this C Bit On.
  c. If the multiple axis groups use M96 P_ I_ Q_ R_ L_ to specify the signal source, each axis group can specify different signal sources, and each axis group only refers to the signal source specified by the respective axis group.
- Description
  a. Return to main program method: Use M99 in the interrupting subprogram, where:
     i. M99 (without argument): Return to the coordinates of the of interruption with G00 and reinterprets it from the interrupted block.
     ii. M99 PXXXX: Return to the specified N-number block to start interpretation (no G00 return action). If the specified return block number (N) does not exist, trigger alarm COR-017.
     iii. M99 QXXXX: Return to the specified row number to start interpretation (no G00 return action). If the specified return number does not exist, trigger alarm COR-018.
  b. The interrupt signal is not supported within the subprogram; triggering in the subprogram may cause wrong interruption return row number problem.
     PS: That is, calling the subprogram with M98 or M198 or returning from a subprogram within the rows between M96 and M97 are not allowed. Please refer to the figure below for how to use it.
     PS: Please note that the above limit refers to the interruption trigger valid area; M98 can be used to

call another subprogram in an interrupting subprogram.



c. The M96 M97 command will block the pre-interpretation so that axes are decelerated to zero.

d. If there are multiple M96 in the program, the interrupt subprogram number is determined by the M96 P argument that is closest to the interrupt signal.

e. If M96 is issued In the program, M97 must be issued to cancel the function before the main program ends, otherwise alarm COR-117 [Interrupt type subprogram does not issue M code] will be triggered.

f. When M96 P_[I_] [Q_] [R_] [L_] is issued in the program, if the I, Q, P, R, L arguments are beyond the specified range, the alarm COR-330 [illegal interrupt signal format] will be triggered.

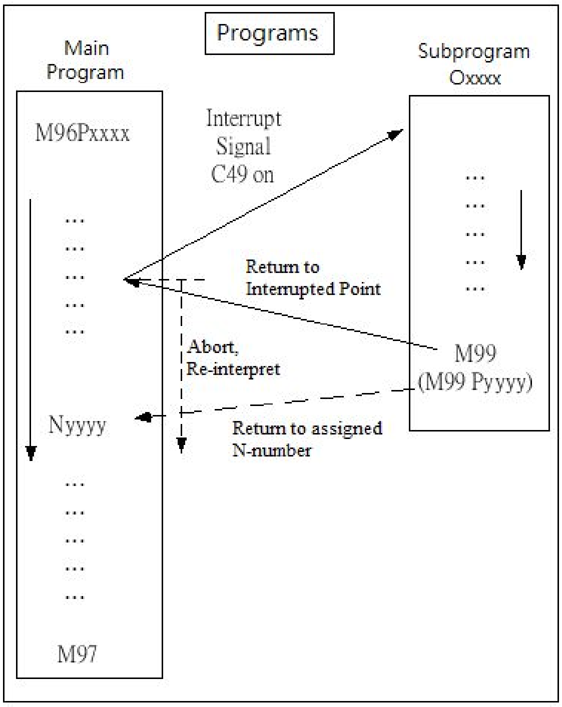
g. G02(G03) I_ J_ K_ and ,A_ ,R_ ,C_, these geometric related functions will be affected by the starting point of the single block. Therefore, if the interruption occurs in the block, an alarm will be triggered or may have different path from the original.

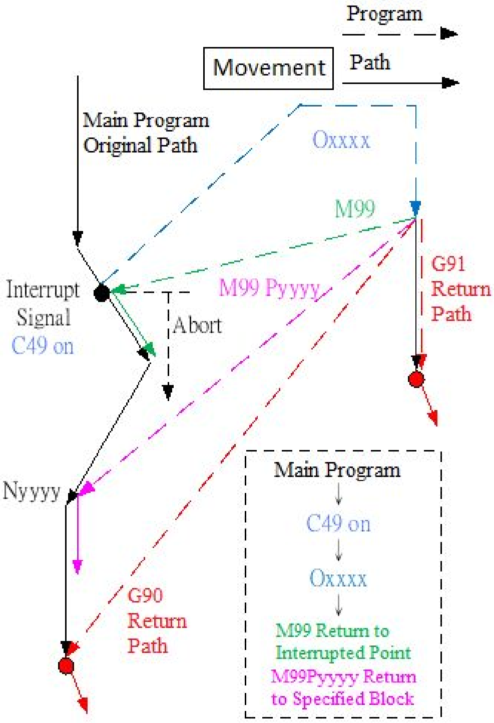
h. The interrupt subprogram inherits the state of the main program interrupt point, including G, S, T, and so on.

    i. S, T, etc, these pre-interpretation blocking commands will correctly inherit the status when entering interruption point.

    ii. G, F, etc., these pre-interpretation commands will inherit the pre-interpreted status, please be careful when programming.
    PS: For example, if the main program receives an interrupt signal in the execution of G00X50., the initial interpolation state is not necessarily G00 when entering to the subprogram.

i. In the figure below, main program's G00 Z100. (starting position 0.) block is interrupted and stopped at the position of Z35. When returning from interrupting subprogram Oxxxx, system returns to interrupt point Z35 and then execute G00 Z100. If using G90 mode to return to interrupted block at Z35, it will move to Z100.; If using G91 mode, it will move to Z135 after returning to interrupted block. (Only C type needs to be notice this description).





Whether M99 returns to the main program interrupted point or M99Pyyyy returns to the main program block Nyyyy, it is both re-interpreted. So if G91 mode is used, need to pay attention to whether the processing path meets the requirements (Only Mill and Lathe C type need to notice this description).

j. The interrupting subprogram function can't work when using followings function
G5: High speed high precision function
G5.1: Path smoothing function
G12.1: Polar coordinate interpolation
G16: Polar coordinate transformation
G41(G42): Tool radius compensation
G51: Scaling function
G51.1: Mirror function
G51.2: Polygon cutting
G114.1: Spindle synchronization
G114.3: Spindle bearing function
When the interrupting function is executed, if the controller is in these modes, the interrupt function (C49, and the specified trigger signal under the command) will not be enabled.
k. The M96 is canceled when Emergency Stop, M30, or Reset are triggered.
l. When executing Feedhold or the Block stop (M00/C40), the M96 trigger will be paused. The signal hold time will be paused and not be cleared until the restart (Cycle Start) and then resumed M96 to interrupt Trigger and continue timing.

- Program example

```
// main program
M96 P1111
G00 X0 Y0 Z0
G01 X10. F500
Y10.
```

```
X0
Y0
M97
M30
// O1111 (interrupting subprogram) simulate the Z-axis ascend to check tool and descend back
%@MACRO
#30 : = #1000; //mode backup: G00/G01/G02/G03
#31 : = #1004; //mode backup: G90/G91
G00 Z100.; // rapif positioning to the Z-axis tool checkpoint
G#30 G#31; // mode restore
M00; // After entering M00, it is allowed to switch to manual mode for axial movement.
M99; // return to interrputed point
```

- Precaution
  - Versions after 10.116.10 support M96/M97 as "interrupting subprogram call function M code".
  -Versions after 10.116.24Y/10.116.36E (included) provide Pr3600 *Define interrupting subprogram call function M code to set the value of subprogram call M code.
  -As mentioned above, when the Pr3600 is set to the same value as the extension M code parameter (Pr3601~) or the part count M code (Pr3804), the OP-020 alarm will be triggered. Please correct accordingly.
  -Versions after 10.118.10 support M96 P_[I_] [Q_] [R_] [L_] command which can specify trigger signals other than C49.
  -After 10.118.12E, 10.118.15, if M96 has specified a trigger signal, the interrupted subprogram only follows the specified trigger signal and ignores C49 signal.

3.12 M98/M99- Subprogram Control (C-Type)

1. M98 P_H_L_ Calling of subprogram, must be used with M99.
   P: the number of subprogram to be called (when P is unspecified, system specifies the program itself, and it is valid only in memory running or MDI mode)
   H: the subprogram sequence number (N) to be called (when H is unspecified, system will execute from the forefront)
   L: repeated times of subprogram.
   - description:
     i. Subprogram is the parameter that includes fixed cutting procedures or repeatedly used parameters. We should prepare it in advance and put it into the memory. We call from the main program when we need to use. Calling subprogram is executed by M98, and it will end by executing M99.
     ii. When running M02 and M30 in the subprogram, system regards it as the end of the subprograms and returns to the main program.
     iii. It's a M code which cannot be registered as Pr3804 part count.
2. M99 P_ subprogram ends
   P: the sequence number of caller program for returning back after subprogram ends. when P argument is unspecified, then after returning to main program, program executes from the next block of M98 or M198 .

3.13 M198- Call External Subprogram Function (C-type)

1. M198: Call external subprogram function, needs to use with M99.
   Command form is M198 P_ H_ L_
   P: The subprogram number to be called (when P is omitted, it is the specified program itself, and can only be used in memory operation or MDI operation mode)
   H: The subprogram serial number to be called (When omitted, start from the front)
   L: Number of repeated executions for the subprogram

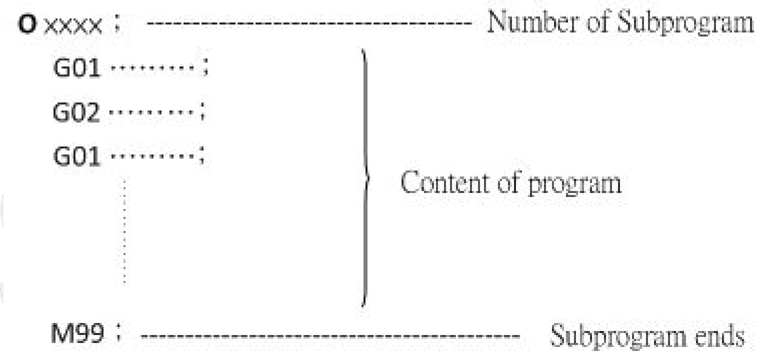
- Description
    1. After interpreting this command, it will force the file to be re-read once, which ensures that the executed sub-file is the latest state when the M198 is interpreted.
    2. The subprogram refers to a fixed machining program or a parameter that is frequently used repeatedly. It is prepared in advance and stored in the memory. When it is needed, it can be called by the main program. The outgoing call of the subprogram is executed by M198, and the end is executed by M99.
    3. If the M02 and M30 commands are executed in the subprogram, it's considered that the subprogram ends and it returns to main program continues executing.。
- Precaution
    1. When the M198 is registered as the M code calling macro function by (parameters 3601~3610), the open file function is invalid.
    2. M198 can only call the subprogram with file type , and issue an alarm COR-52 when there is no P argument.
    3. Since the M198 reopening is a pre-solving operation, if you need to control the pre-solution, you can use WAIT().

2. M99: Back to main program
   Command form is M99 P_
   P: Indicates the execution block number (N) when returning to the main program after the completion of the subprogram. If the P argument does not exist, it means that when returning to the main program, it will continue processing from the next line of M98 or M198.

3.14 Making and Execution of Subprogram (C-Type)

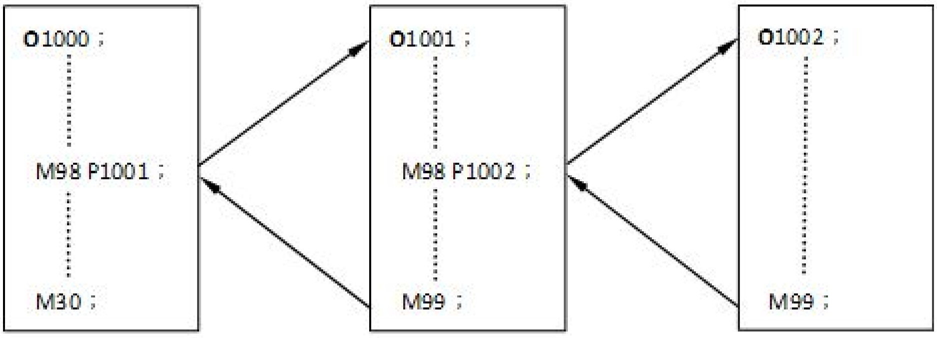
3.14.1 Program Format of General Subprogram (C-Type)

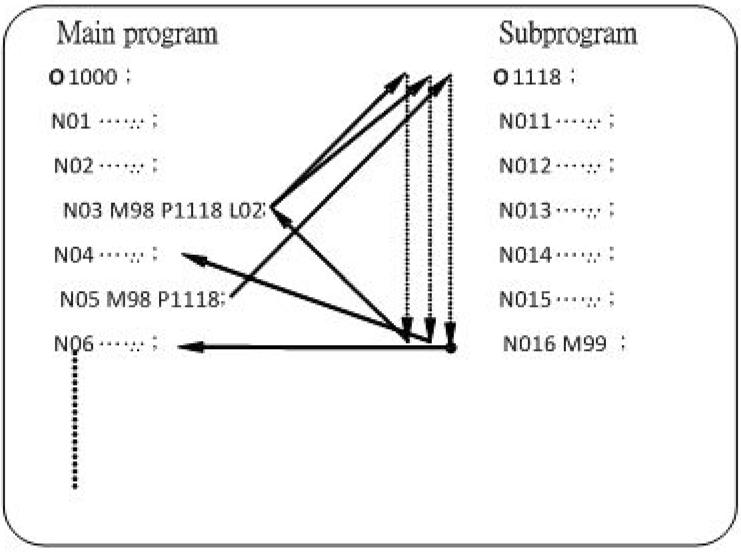
The normal format as below:



```
O xxxx ;
  G01 ·········;
  G02 ·········;
  G01 ·········;
   ⋮
  M99 ;
```

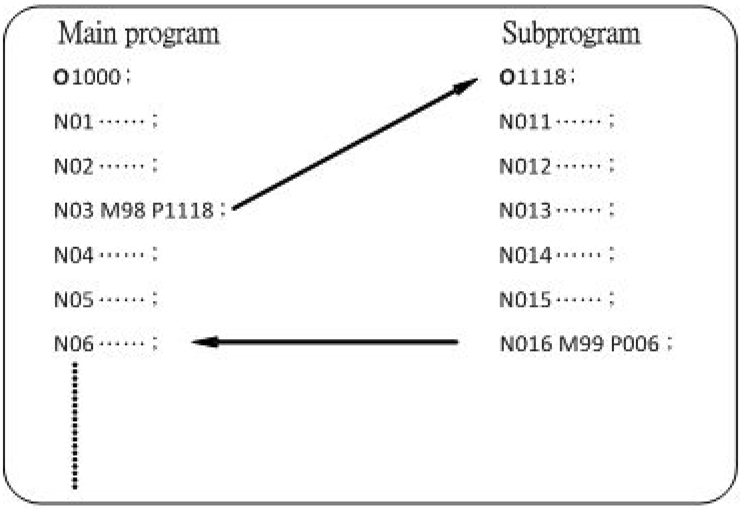
3.14.2 Main program use with Subprogram to call command, and execution sequence (C-Type)



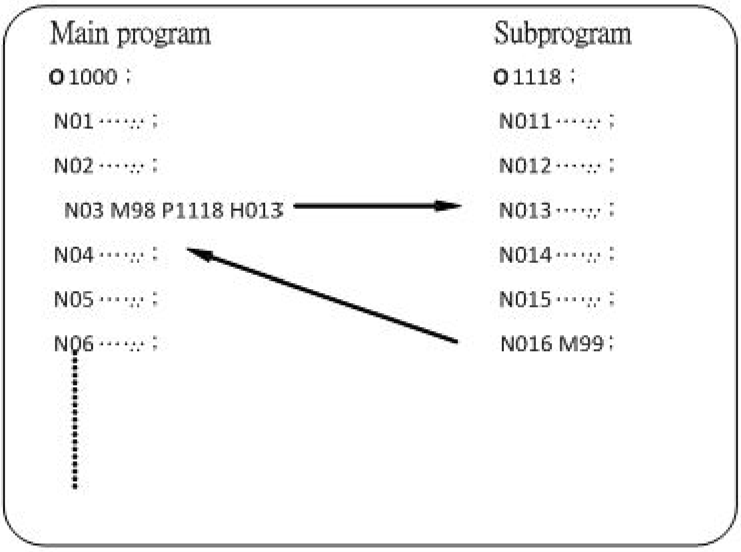


3.15 Special usage of subprogram (C-Type)

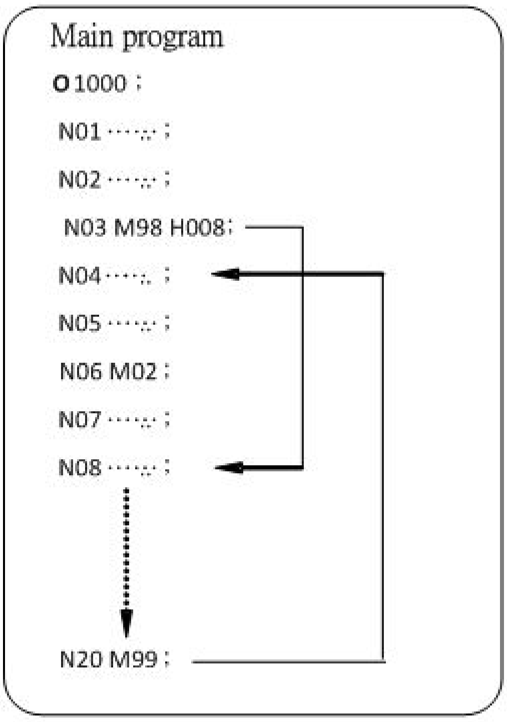
(1) We can execute subprogram by adding **P__** function after M99 in the end of the final block. After finishing this program, it will return to main program, and execute the block which the sequence number specified by P_ function is in.



(2) Subprogram also can execute P_ command with H_ command in M98. The system will execute the subprogram (specified by P_) from the sequence number specified by H_. . The subprogram is therefore versatile. With open only one subprogram to execute muti-purpose function, the system can save more memory space.



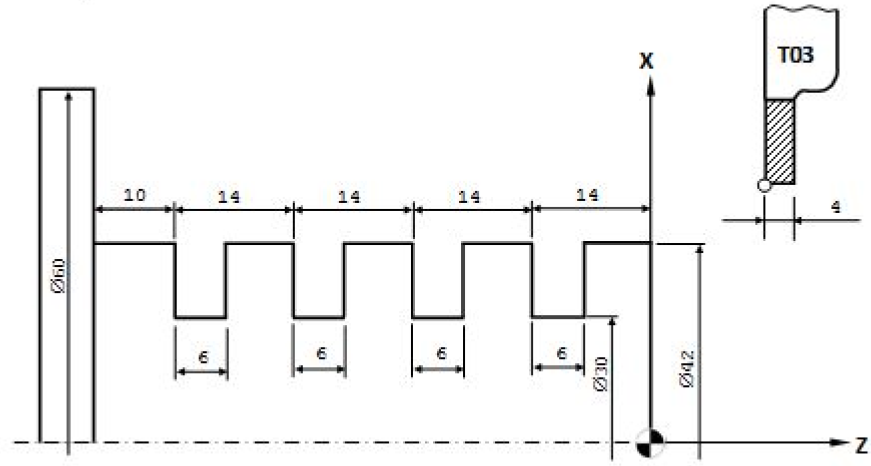
(3) If user leaves P_ command unspecified and only specify H_ command in M98, the system will execute from the sequence number of main program that specified by H_ command. After executing M99, it will return to the next block of M98 and continue to execute the program.



```
Main program
O1000 ;
N01 ·····;
N02 ·····;
N03 M98 H008;
N04 ·····;
N05 ·····;
N06 M02 ;
N07 ·····;
N08 ·····;
N20 M99 ;
```

3.15.1 Example (C-type)

Tank cutting, using “calling of subprogram” to execute repeating machining



(1). First way: P command in block of M98

* Main program.

T03 //use tool NO.3

G97 S710 M03 //constant rotate speed of spindle, 710 rpm CW

M08 //cutting liquid ON

```
G00 X45.0 Z-12.0 //positioning to the above of first tank
M98 P1234 H102 L4      //call the subprogram of sequence number
                        //“O1234”, machining from the block of N102
                        //and repeating 4 times
G28 X80.0 Z80.0 //positioning to specified mid-point and return to
                   //machine zero point
M09               //cutting liquid OFF
M05               //spindle stops
M30
```

* Subprogram.

```
O1234
G00 X45.0 Z-12.0
G01 X30.0 F200  ßStart from this block
               //linear interpolation to the bottom of the tank, feedrate
               //200μm/rev
G00 X45.0 //escaping to start position
     W-2.0        //move2mmtoward negative direction of Z
G01 X30.0 //linear interpolation to the bottom of the tank
G00 X45.0 // escaping to start position
     W-12.0       // move12mmtoward negative direction of Z, and wait
               //for cutting next tank
M99        //return to main program
```

(2). Second way: without executing P_ command in block of M98

* Main program.

```
T03               //use tool NO.3
G97 S710 M03     //constant rotate speed of spindle, 710 rpm CW
M08               //cutting liquid ON
G00 X45.0 Z-12.0 //positioning above the first tank
M98 H0010 L4     //execute from the block of main program sequence
                   //number N0010, and repeat 4 times
G28 X80.0 Z80.0 //positioning to specified mid-point and return to
                   //machine zero point
```

```
M09              //cutting liquid OFF
M05              //spindle stops
M30              //program ends

N0010
G01 X30.0 F200  ßstart with this block after executing M98
                    //linear interpolation to the bottom of the tank,
                    feedrate 200μm/rev
G00 X45.0         //escaping to start point
W-2.0                  // move2mmtoward negative direction of Z
G01 X30.0         //linear interpolation to the bottom of the tank
G00 X45.0         //escaping to start point
W-12.0                 // move12mmtoward negative direction of Z, and
                    //wait for cutting next tank
M99              //return the next block N006 of M98
```

4 Appendix (C-type)

4.1 Description of lathe parameter (C-type)

NO	Explain	Input range	Unit	Description
4001	Drilling mode	[0, 1]		0: high speed 1: normal
4002	Escaping amount of drilling cycle	[0, 999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.
4003	Set Tapping R point pause time	[0,999999999]	ms	
4004	Set Pecking Type	[0,1]	-	
4005	Set the tool retract amount of pecking	[0,999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.
4006	Set the tapping cycle tool retract feedrate	[100,300]	%	
4007	Positioning function before serial spindle tapping	[0,1]	-	

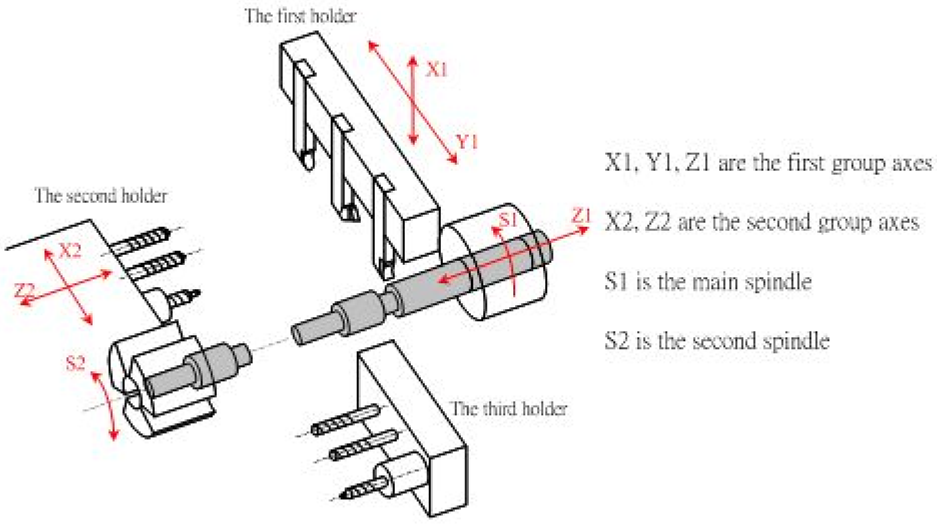
NO	Explain	Input range	Unit	Description
4008	High-speed drilling/ tapping mode	[0,1]	-	
4011	Escaping amount of peck drilling cycle	[0, 999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.
4012	Escaping amount of cutting cycle	[0, 999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.
4013	Cutting value of cutting cycle	[0, 999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.
4014	Rough cutting cycle mode	[0,1]	-	
4015	Cutting value of pattern repeating in X direction	[0, 999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.
4016	Cutting value of pattern repeating in Z direction	[0, 999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.
4017	Number of repeats of pattern repeating	[1, 999]	Number of times	
4018	Chamfer angle of thread cutting G21	[0, 89]	degree	
4019	Whether the Y axis of the drilling, tapping and boring cycle is a positioning command (0: No; 1: Yes)	[0,1]	-	
4020	G12.1 X axis programming (0: radius axis, 1: diameter axis)	[0,1]	-	

NO	Explain	Input range	Unit	Description
4021	*Spindle synchronization function, basic spindle number.	[0,6]	-	
4022	*Spindle synchronization function, synchronous spindle number.	[0,6]	-	
4023	*Second set of spindle synchronization function, basic spindle number.	[0,6]	-	
4024	*Second set of spindle synchronization function, synchronous spindle number	[0,6]	-	
4025	*Third set of spindle synchronization function, basic spindle number.	[0,6]	-	
4026	*Third set of spindle synchronization function, synchronous spindle number	[0,6]	-	
4041	Finishing allowance of threading	[0, 999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.
4042	Thread angle of threading	{0, 29, 30, 55, 60, 80}	Degree	
4043	Chamfering value of threading	[0, 99]	0.1 pitch	
4044	Times of finishing allowance in threading	[0, 99]	Number of times	
4045	Min. cutting value in threading	[0, 999999999]	LIU	LIU is min. input unit, and it will be affected by Metric or Imperial system in use.

NO	Explain	Input range	Unit	Description
4051	Multiple cutting cycle, increasing (decreasing) allowed error range (um)	[0,999999999]		LIU is min. input unit, and it will be affected by Metric or Imperial system in use.

4.2 Lathe dual program instruction (C-type)

To save the time of the processing, the SYNTEC lathe's controllers can drive two programs simultaneously. The two program can drive two pairs of turret to execute linear interpolation and circular interpolation at the same time. The system therefore achieves the most effective lathe status while processing workpieces in external diameter and internal diameter simultaneously.



4.2.1 Dual program related command description (C-type)

$1 → the contents after the instruction in the program is the first group

$2 → the contents after the instruction in the program is the second group

The second group in the program must end with M99.

G04.1 P_ → synchronous instruction, G04.1 P1 inthe first group and one in the second group would wait for each other until synchronization succeeds and go to next section.

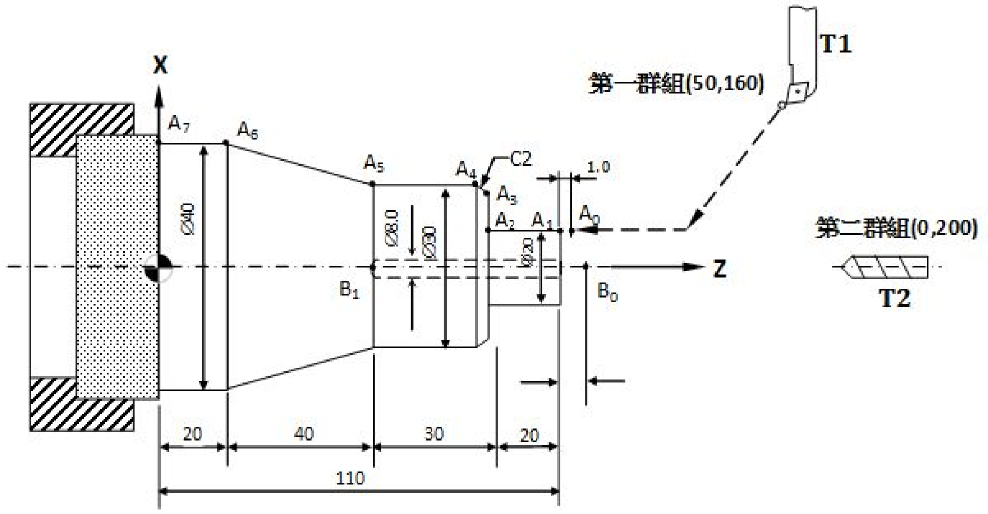
G04.1 P2 waits for each other until synchronization succeeds and go to next section in the same way.

4.2.2 Dual program related M code (C-type)

M_code	The specification

M03	The first main axis rotates in positive direction
M04	The first main axis rotates in negative direction
M05	The first main axis stops
M63	The second main axis rotates in positive direction
M64	The second main axis rotates in negative direction
M65	The second main axis stops
M70	Assign the first main axis to be the main axis of first group.
M71	Assign the second main axis to be the main axis of second group.

4.2.3 Machining program example (C-type)

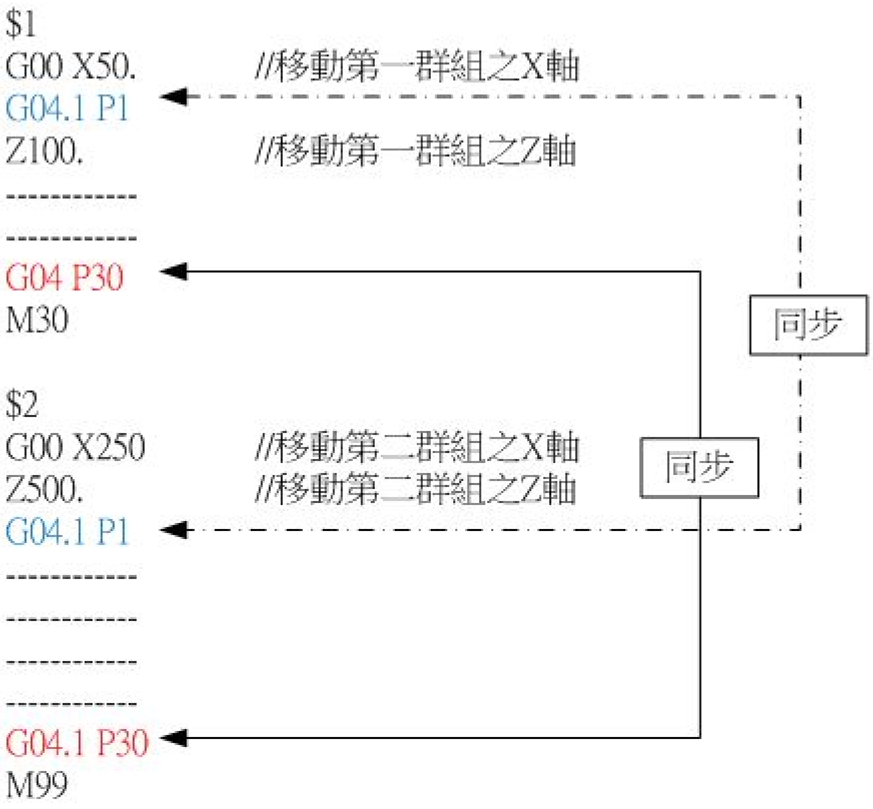


```
$1                              //the first group
G92 X50.0 Z160.0 S10000      //set origin, the highest speed 10000 rpm
T01                          //use knife No.1
G96 S130 M03                   //face speed130m/min, main axis rotates
                                  //in positive direction
M08                          //turn on cutting liquid
G04.1 P1
G00 X20.0 Z111.0        //positioning to A0 rapidly
```

```
G01 Z90.0 F0.6          //linear cutting A0-->A2
    X26.0               //A2-->A3
    X30.0 Z88.0         //A3-->A4
    Z60.0               //A4-->A5
G04.1 P2
    X40.0 Z20.0         //A5-->A6
    Z0.0            //A6-->A7
G00 X50.0           //back knife rapidly
    Z160.0              //return to origin
G04.1 P3
M05 M09             //stop the main axis, turn off cutting liquid
G04.1 P4
M30                 //end program

$2                      //the second group
G04.1 P1
T02                 // use  knife No.2
G04.1 P2
G00 X0 Z120.          //position to B0 rapidly
G01 Z60. F0.5         //move knife B0-->B1
G00 Z120.            //back knife B1-->B0
G04.1 P3
G00 Z200.            //back the knife
G04.1 P4
M99
```

4.2.4 Program editing (C-type)

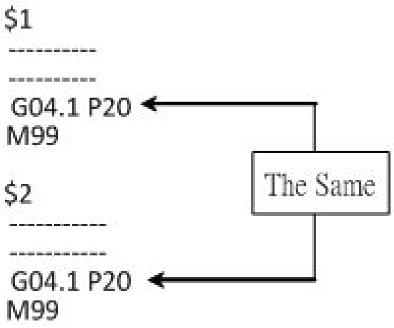
Start a new file and program the processing file according to the following example.



```
$1
G00 X50.        //移動第一群組之X軸
G04.1 P1
Z100.           //移動第一群組之Z軸
------------
------------
G04 P30
M30

$2
G00 X250        //移動第二群組之X軸
Z500.           //移動第二群組之Z軸
G04.1 P1
------------
------------
------------
------------
G04.1 P30
M99
```

4.2.5 Program editing precaution (C-type)

1. The first group of the program must start with $1 and the second one must start with $2.
2. The quantities of G04.1 P_ in the first group must be the same as that in second group and the number after P has to be sequentially assigned in increasing order.
3. Put end command M30 or M02 inthe first group when program ends and M99 must be specified in the last block of the second group.
4. When repeatedly processing several workpieces automatically is required, specify M99 inthe end of the first group program. But notice that in order to enable the synchronization of first and second groups, the same G04.1 P_ code must be specified before M99 of the first and second group.



```
$1
----------
----------
G04.1 P20
M99

$2
-----------
-----------
G04.1 P20
M99
```

5. With the axis set belonged to the second group, G code can only be specified in the second group. With the axis set belonged to the first group, if we specify G code in the second group, commands are ineffective.

6. M code ,S code and T code are all available in the first and second group. Therefore M code ,S code and T code can be properly executed simultaneously in the first and second group.

4.3 Description of Lathe graph assist G code (C-type)

4.3.1 **Description**

Lathe graph assist G code is the special G code specified by inserted cycle in program editing. For example, two lines have to be specified when using G73 command manually. Only a line is to be specified in the special G code which inserted cycle automatically generates. Thus system combines two lines in G73 into special G code G73.1. The following is the instructions of special G code.(Special G code conversational input mode is only available in DOS version)

4.3.2 Assist G code list (C-type)

- G73.1 Stock Removal in Turning
- G74.1 Stock Removal in Facing
- G75.1 Pattern Repeating
- G76.1 End Face (Z axis) Peck Drilling Cycle
- G77.1 Outer Diameter/Internal Diameter Drilling Cycle
- G78.1 Multiple Thread Cutting Cycle

4.3.3 G73.1 Stock Removal in Turning (C-type)

Command Form

G73.1 D(Δd) X(e) P (ns) Q (nf) U(Δu) W(Δw) F___ S___ T___

Δd: depth of cut in X axis direction, default can be specified by the system parameter#4013.

e: escaping amount, specified by the parameter#4012

ns: sequence number of the first block for the program of finishing shape

nf: sequence number of the last block for the program of finishing shape

Δu: distance of finishing allowance in X direction

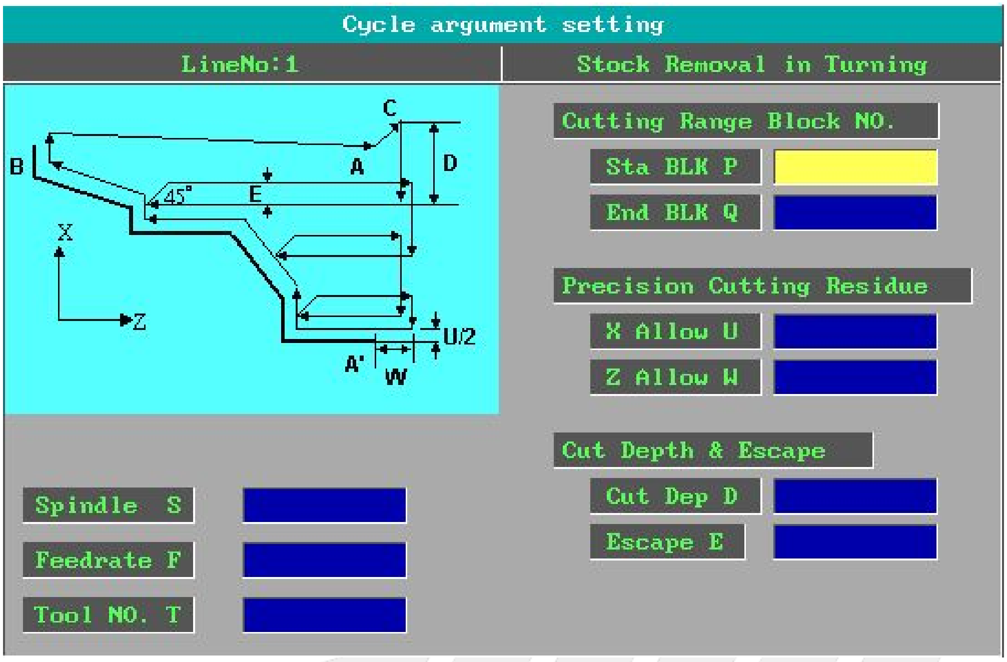
Δw: distance of finishing allowance in Z direction

F: feedrate

T: tool number

S: spindle rotate speed

Description



Cycle argument setting

LineNo:1 | Stock Removal in Turning

Cutting Range Block NO.

Sta BLK P

End BLK Q

Precision Cutting Residue

X Allow U

Z Allow W

Cut Depth & Escape

Cut Dep D

Escape E

Spindle S

Feedrate F

Tool NO. T

4.3.4 G74.1 Stock Removal in Facing (C-type)

Command Form

G74.1 D (d) E (e) P (ns) Q (nf) U(Δu) W(Δw) F___ S___ T___

d: depth of cut in Z axis direction, it can be specified by the parameter#4013 and the parameter is changed by the program command

e: escaping amount, it can be specified by the parameter#4012

ns: sequence number of the first block for the program of finishing shape

nf: sequence number of the last block for the program of finishing shape

Δu: distance of finishing allowance in X direction

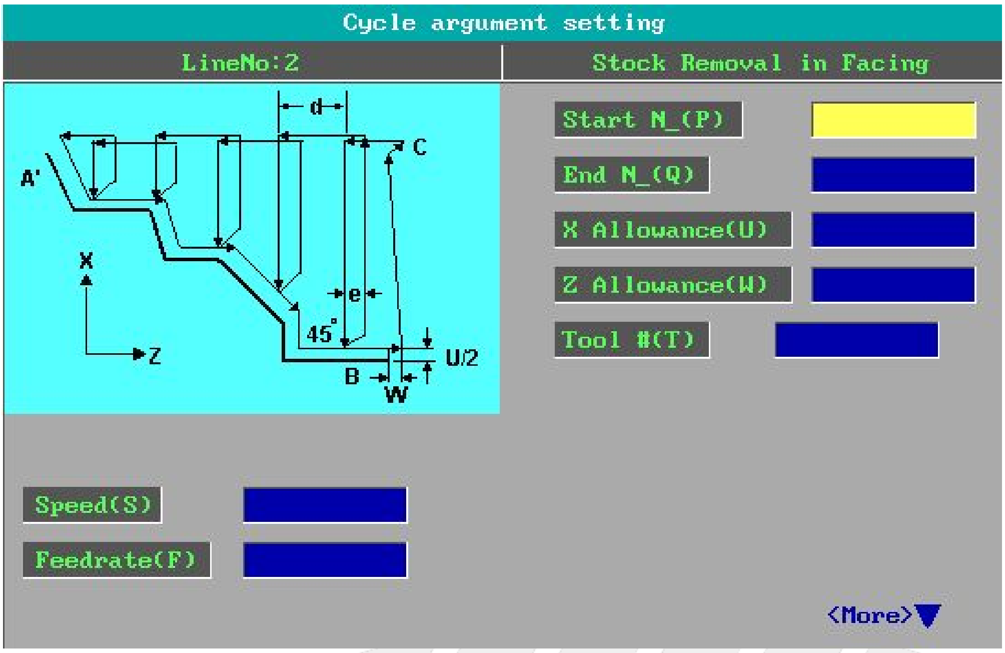
Δw: distance of finishing allowance in Z direction

F: feedrate

T: tool number

S: spindle rotate speed

Description



Cycle argument setting

LineNo:2 | Stock Removal in Facing

Start N_(P)

End N_(Q)

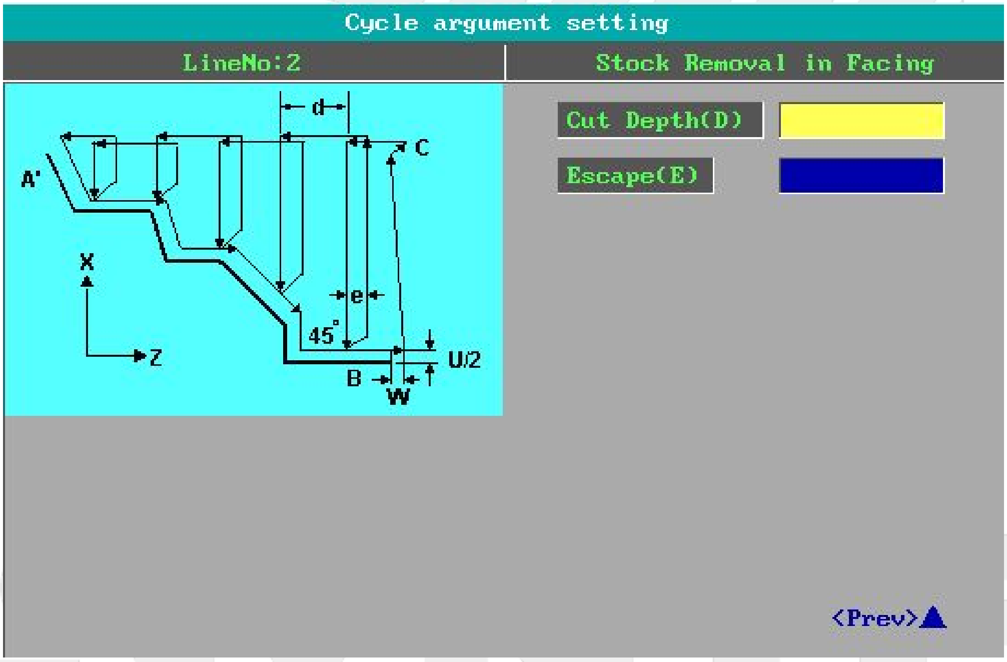
X Allowance(U)

Z Allowance(W)

Tool #(T)

Speed(S)

Feedrate(F)



Cycle argument setting

LineNo:2 | Stock Removal in Facing

Cut Depth(D)

Escape(E)

4.3.5 G75.1: Pattern Repeating (C-type)

Command Form

G75.1 X(Δi) Z(Δk) D(d)_P (ns) Q (nf) U(Δu) W(Δw) F___ S___ T___

Δi: distance and direction of relief in the X axis direction, this value can be specified by the parameter #4015

ΔK: distance and direction of relief in the Z axis direction, this value can be specified by the parameter #4016

d: the number of division, it can be specified by parameter #4017

ns: sequence number of the first block for the program of finishing shape

nf: sequence number of the last block for the program of finishing shape

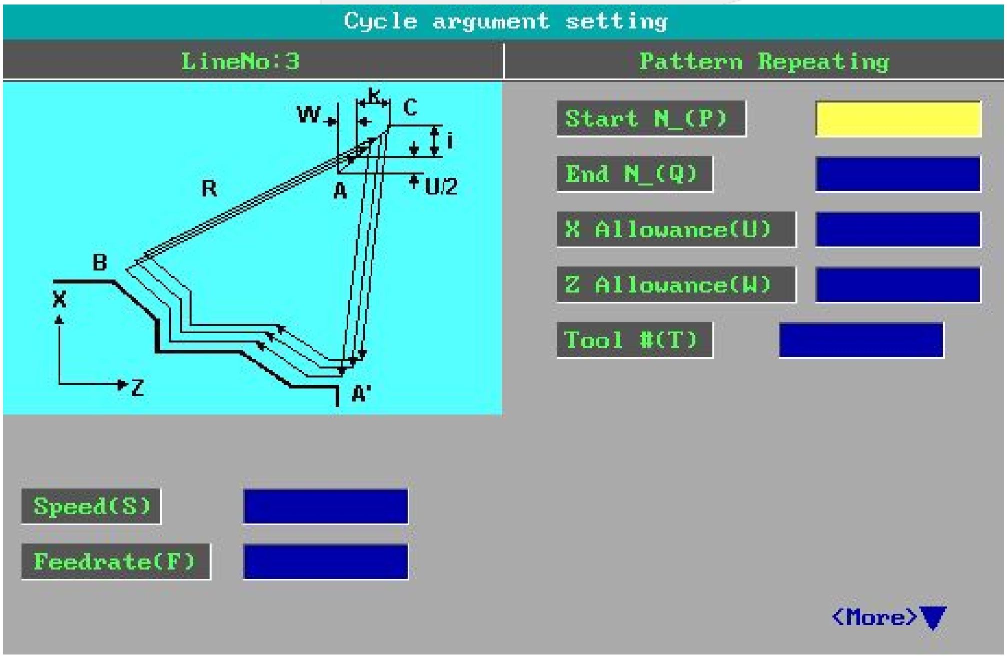
Δu: distance and direction of finishing allowance in X direction

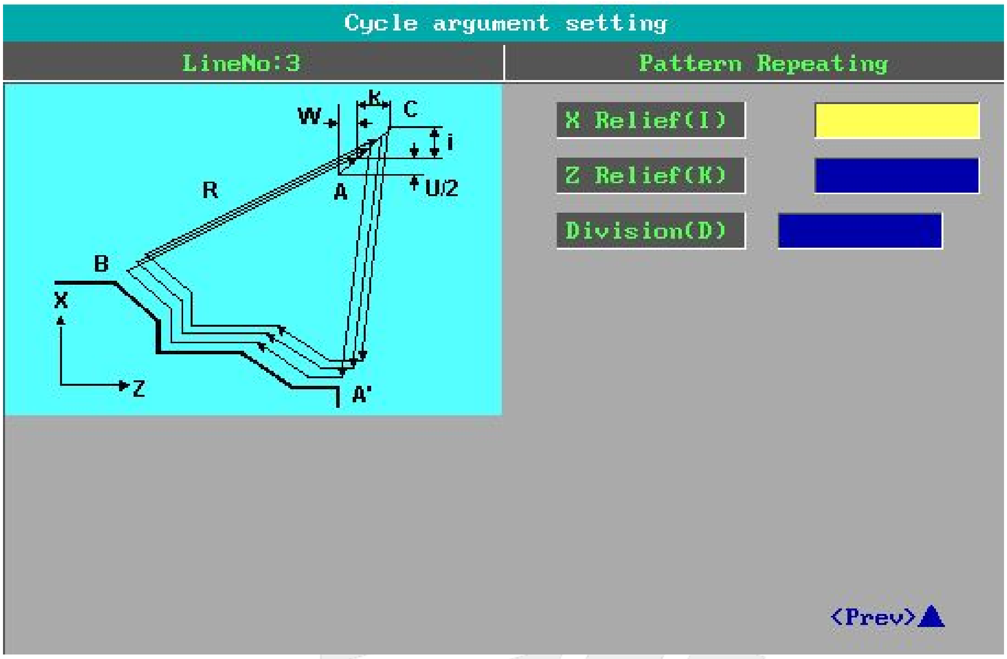
Δw: distance and direction of finishing allowance in X direction

F: feedrate

T: number of the tool
S: spindle rotate speed

Description





Cycle argument setting

LineNo:3

Pattern Repeating

X Relief(I)

Z Relief(K)

Division(D)

<Prev>▲

4.3.6 G76.1: End Face (Z axis) Peck Drilling Cycle (C-type)

Command Form

G76.1 E (e) X(U) Z(W) P(Δi) Q(Δk) R (d) F

e: escaping amount (escaping amount in Z direction when Δk depth is cut) ß it can be specified by parameter #4011

X: X coordinate of point B （diameter）

Z: Z coordinate of point C

U: Incremental amount from A to B （diameter）

W: Incremental amount from A to C

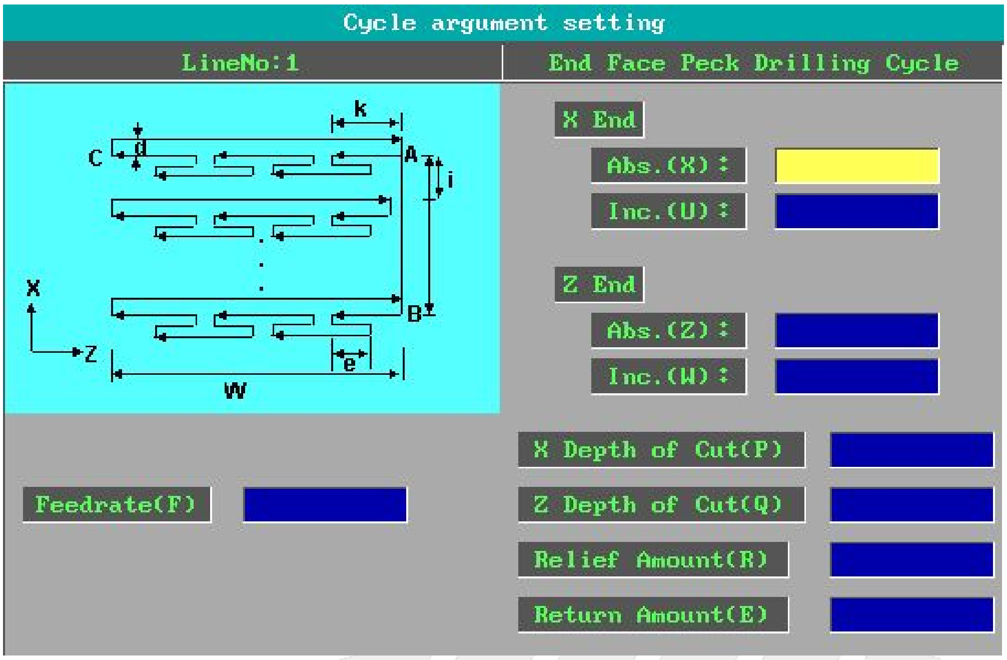
Δi: Movement amount each cut in X direction(displayed by radius, positive)

Δk: Movement amount each cut in Z direction(positive)

Δd: Relief amount of the tool at the cutting bottom. (The value is 0 when it returns in original path)

F: Feed rate

Description



Cycle argument setting

LineNo:1

End Face Peck Drilling Cycle

X End

Abs.(X):

Inc.(U):

Z End

Abs.(Z):

Inc.(W):

X Depth of Cut(P)

Feedrate(F)

Z Depth of Cut(Q)

Relief Amount(R)

Return Amount(E)

4.3.7 G77.1: Outer Diameter/Internal Diameter Drilling Cycle (C-type)

Command Form

G77.1 E (e) X(U)___ Z(W)___ P(Δi) Q(Δk) R(Δd) F___

e: escaping amount(after cutting Δi distance in X axis direction) ßit can be specified by parameter #4011

X: X coordinate of point C(diameter)

Z: Z coordinate of point C

U: increment amount from B to C(diameter)

W: increment amount from A to B

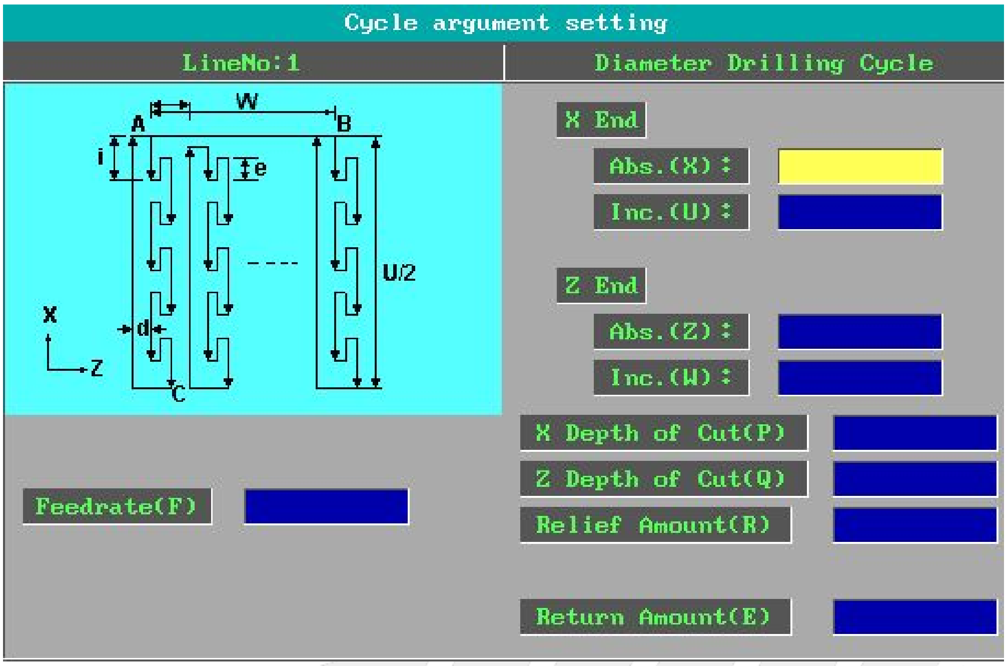
Δi: movement amount in X direction (display by radius, positive)

Δk: depth of cut in Z direction (positive)

Δd: Relief amount of the tool at the cutting bottom. (The value is 0 when it returns in origin path)

F: feedrate

Description



4.3.8 G78.1: Multiple Thread Cutting Cycle (C-type)

Command Form

G78.1 K(m) C(r) A(a) D(Δdmin) B_(d)_ X(U)_ Z(W)_ R (Δi) P (Δk) Q (Δd) (F__ or E__) __

m: repetitive count in finishing, specified by system parameter #4044.

r: chamfering amount, specified by system parameter #4043.

a: angle of tool tip, the angle from 80^0, 60^0, 55^0, 30^0, $29^0$and 0^0 is available. a can also be specified by system parameter #4042.

Q: minimum cutting depth , specified by system parameter #4045d: finishing allowance, specified by system parameter #4041

X(U): X coordinate in end point(bottom of tooth)

Z(W): Z coordinate in end point(bottom of tooth)

Δi: difference of thread radius

Δk: height of thread

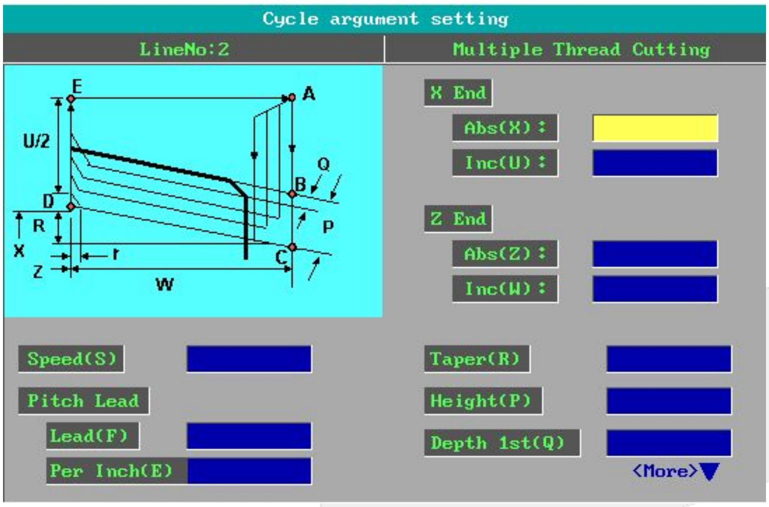
Δd: depth of first cut

F: lead of thread in metric system(unit : mm/tooth)

E: lead of thread in imperial system(unit : tooth/inch)

H: numbers of thread (ex: H3 three thread type cutting, multiple thread F function is for neighbor thread)

Description



Cycle argument setting

LineNo:2 | Multiple Thread Cutting

X End
- Abs(X):
- Inc(U):

Z End
- Abs(Z):
- Inc(W):

Speed(S)

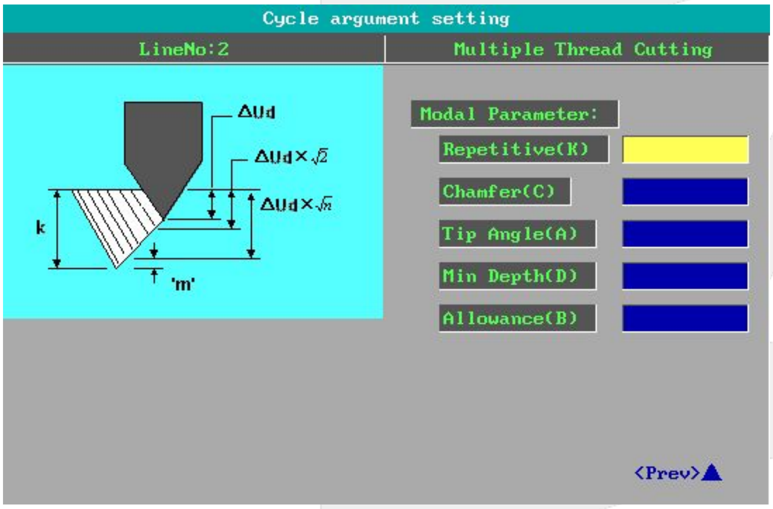
Pitch Lead
- Lead(F)
- Per Inch(E)

Taper(R)

Height(P)

Depth 1st(Q)

<More>▼



Cycle argument setting

LineNo:2 | Multiple Thread Cutting

Modal Parameter:
- Repetitive(K)
- Chamfer(C)
- Tip Angle(A)
- Min Depth(D)
- Allowance(B)

<Prev>▲