Century Star Turning CNC System

Programming Guide

V3.5
April, 2015

Wuhan Huazhong Numerical Control Co., Ltd

Preface

Organization of documentation

1. General
2. Preparatory Function
3. Interpolation Function
4. Feed Function
5. Coordinate System
6. Spindle Speed Function
7. Tool Function
8. Miscellaneous Function
9. Functions to Simplify Programming
10. Comprehensive Programming Example
11. Custom Macro

Applicability

This Programming Guide is applicable to the following CNC system:

HNC-18iT/19iT v4.0

HNC-18xp/T

HNC-180xp/T

HNC-19xp/T

HNC-21TD/22TD v05.62.07.10

Internet Address

http://www.huazhongcnc.com/

Table of Contents

1 General

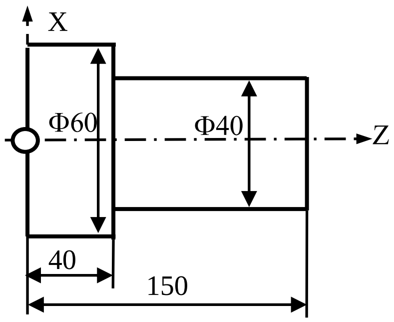
This chapter is to introduce the basic concepts in Computerized Numerical Control (CNC) system: HNC-21T/22T, HNC-18iT/19iT, HNC-18xp/T, HNC-180xp/T, HNC-19xp/T.

1.1 CNC Programming

To operate CNC machine tool, the first step is to understand the part drawing and produce a program manual script. The procedure for machining a part is as follows (Figure 1.1):

1) Read drawing
2) Produce the program manual script
3) Input the program manual script by using the machine control panel
4) Manufacture a part

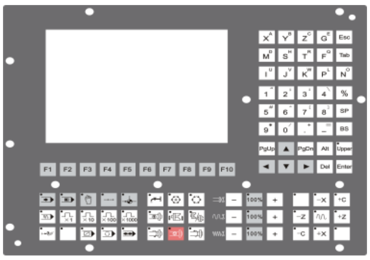
1. Read drawing



2. Produce the program manual script

N1 T0106
N2 M03 S460
N3 G00 X90Z20
N4 G00 X31Z3
N5 G01 Z-50 F100
N6 G00 X36
N7 Z3
…

3. Input the program manual script



4. Manufacture a part

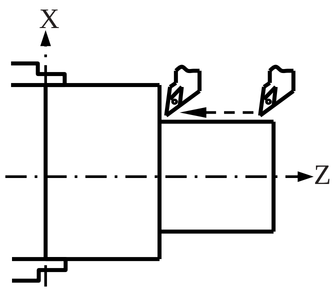


Figure 1.1 The workflow of operation of CNC machine tool

1.2 Interpolation

Interpolation refers to an operation in which the machine tool moves along the workpiece parts. There are five methods of interpolation: linear, circular, helical, parabolic, and cubic. Most CNC machine can provide linear interpolation and circular interpolation. The other three methods of interpolation (helical, parabolic, and cubic interpolation) are usually used to manufacture the complex shapes, such as aerospace parts. In this manual, linear and circular interpolation are introduced.

1.2.1 Linear Interpolation

There are two kinds of linear interpolation:

1) Tool movement along a straight line (Figure 1.2).

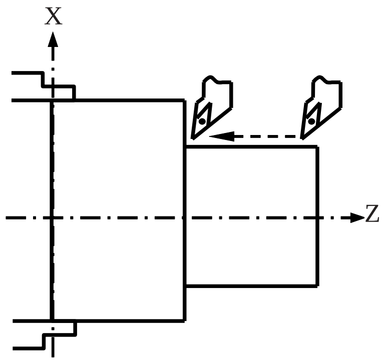


Figure 1.2 Linear Interpolation (1)

2) Tool movement along the taper line

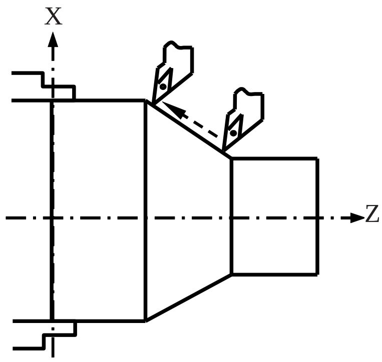


Figure 1.3 Linear Interpolation (2)

1.2.2 Circular Interpolation

Figure 1.4 shows a tool movement along an arc.

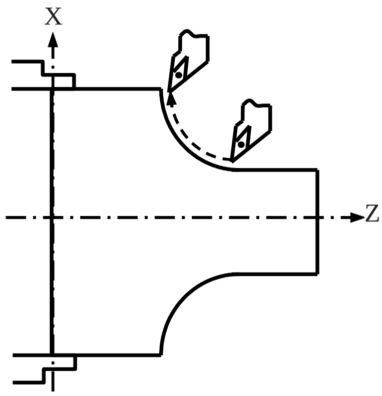


Figure 1.4 Circular Interpolation

Note:

In this manual, it is assumed that tools are moved against workpieces.

1.2.3 Thread Cutting

There are several kinds of threads: cylindrical, taper or face threads. To cut threads on a workpiece, the tool is moved with spindle rotation synchronously.

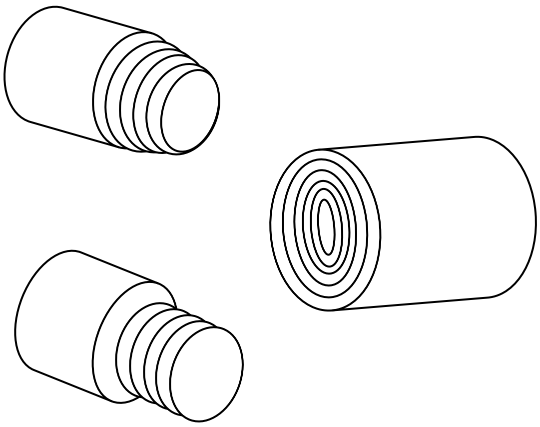


Figure 1.5 Thread Cutting

1.3 Feed Function

- Feed refers to an operation in which the tool moves at a specified speed to cut a workpiece.
- Feedrate refers to a specified speed, and numeric is used to specified the feedrate.
- Feed function refers to an operation to control the feedrate.

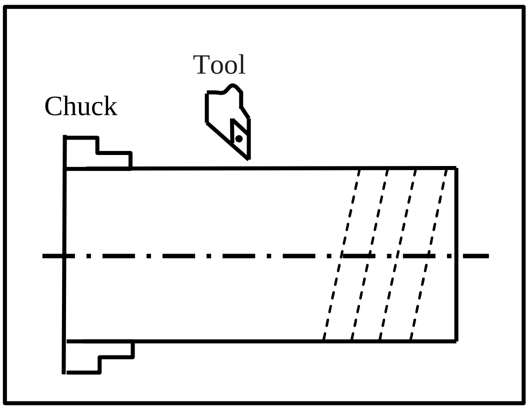


Figure 1.6 Feed Function

For example:

F2.0 //feed the tool 2mm, while the workpiece makes one turn

1.4 Coordinate System

1.4.1 Reference Point

Reference point is a fixed position on CNC machine tool, which is determined by cams and measuring system. Generally, it is used when the tool is required to exchange or the coordinate system is required to set.

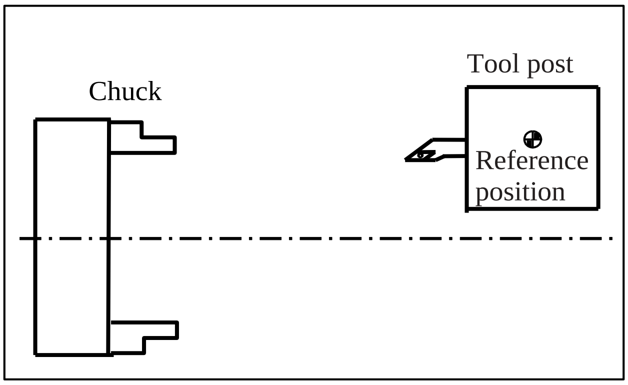


Figure 1.7 Reference Point

There are two ways to move to the reference point:

- Manual reference position return: The tool is moved to the reference point by operating the button on the machine control panel. It is only used when the machine is turned on.
- Automatic reference position return: It is used after the manual reference position return has been used. In this manual, this would be introduced.

1.4.2 Machine Coordinate System

The coordinate system is set on a CNC machine tool. Figure 1.8 is a machine coordinate system of turning machine, and shows the direction of axes:

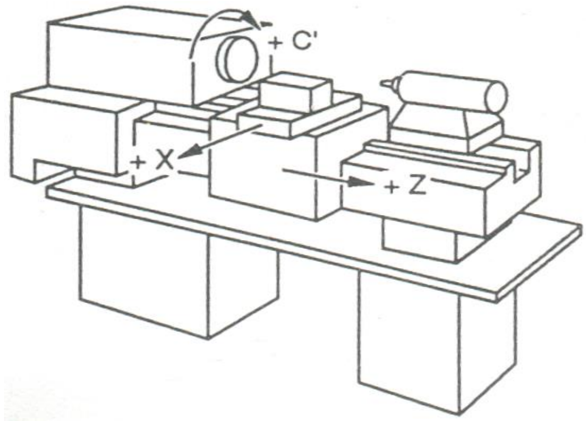


Figure 1.8 Machine Coordinate System

In general, three basic linear coordinate axes of motion are X, Y, Z. Moreover, X, Y, Z axis of rotation is named as A, B, C correspondently. Due to different types of turning machine, the axis direction can be decided by following the rule – “three finger rule” of the right hand.

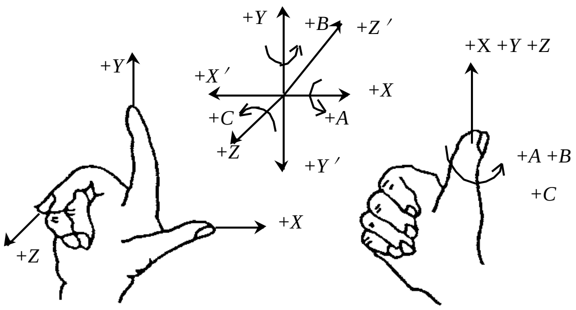


Figure 1.9 “three finger rule”

- The thumb points the X axis. X axis controls the cross motion of the cutting tool. “+X” means that the tool is away from the spindle centerline
- The index points the Y axis. Y axis is usually a virtual axis.
- The middle finger points the Z axis. Z axis controls the motion of the cutting tool. “+Z” means that the tool is away from the spindle.

1.4.3 Workpiece Coordinate System

The coordinate system is set on a workpiece. The data in the NC program is from the workpiece coordinate system.

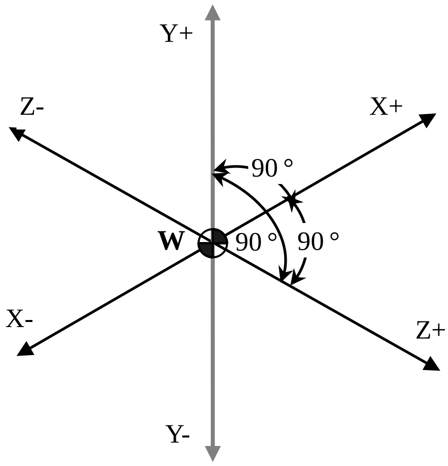


Figure 1.10 Workpiece Coordinate System

Example: Those four points can be defined on workpiece coordinate system:

P1 corresponds to X25 Z-7.5

P2 corresponds to X40 Z-15

P3 corresponds to X40 Z-25

P4 corresponds to X60 Z-35

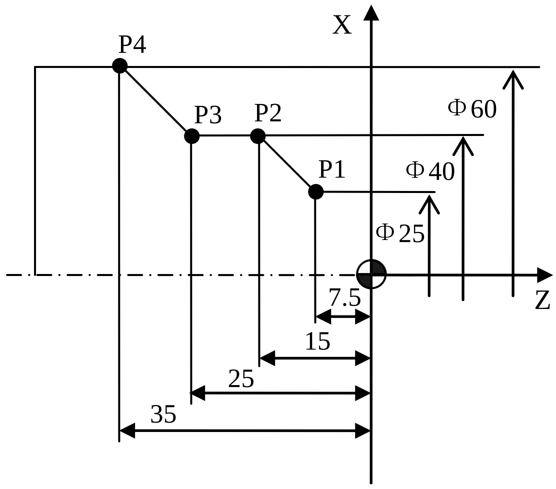


Figure 1.11 Example of defining points on workpiece coordinate system

1.4.4 Setting Two Coordinate Systems at the Same Position

There are two methods used to define two coordinate systems at the same position.

1) The coordinate zero point is set at chuck face

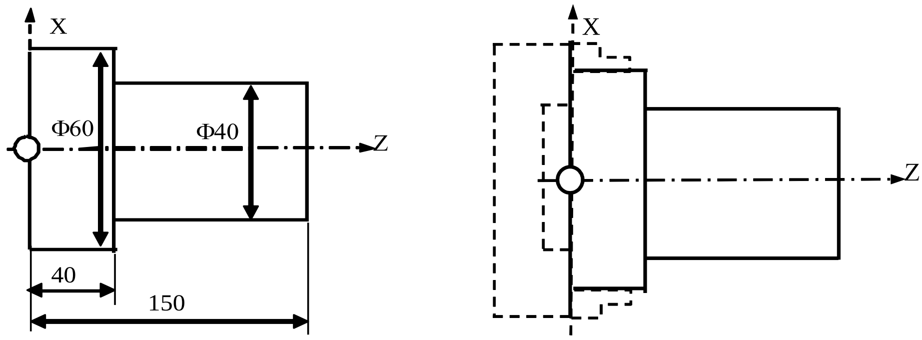


Figure 1.12 The coordinate zero point set at chuck face

2) The coordinate zero point is set at the end face of workpiece

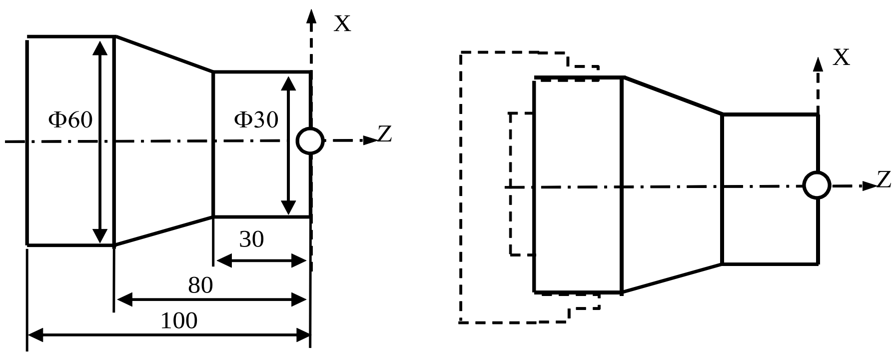


Figure 1.13 The coordinate zero point set at the end face of workpiece

1.4.5 Absolute Commands

The absolute dimension describes a point at “the distance from zero point of the coordinate system”.

Example: These four point in absolute dimensions are the following:

P1 corresponds to X25 Z-7.5

P2 corresponds to X40 Z-15

P3 corresponds to X40 Z-25

P4 corresponds to X60 Z-35

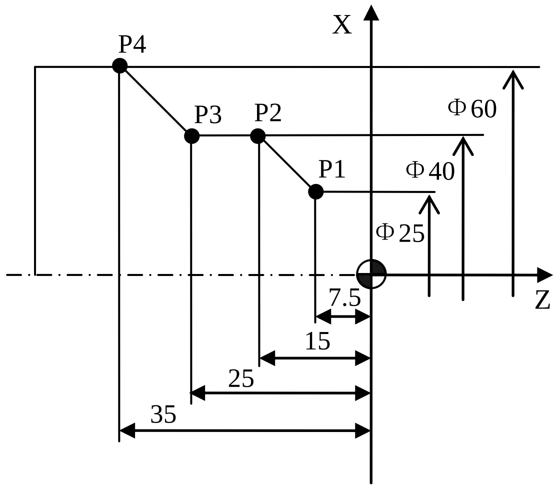


Figure 1.14 Absolute Dimension

1.4.6 Incremental Commands

The incremental dimension describes a distance from the previous tool position to the next tool position.

Example: These four point in incremental dimensions are the following:

P1 corresponds to X25 Z-7.5 //with reference to the zero point

P2 corresponds to X15 Z-7.5 //with reference to P1

P3 corresponds to Z-10 //with reference to P2

P4 corresponds to X20 Z-10 //with reference to P3

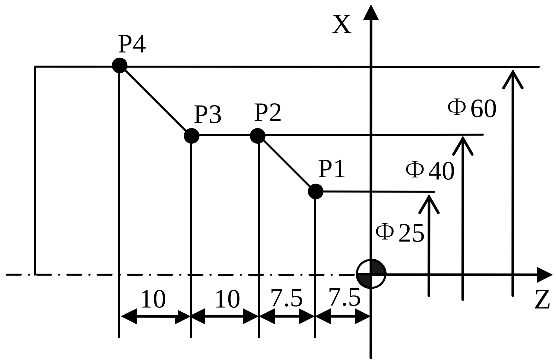


Figure 1.15 Incremental Dimension

1.4.7 Diameter/Radius Programming

The coordinate dimension on X axis can be set in diameter or radius. It should be noted that diameter programming or radius programming should be applied independently on each machine.

Example: Describe the points by diameter programming.

A corresponds to X30 Z80

B corresponds to X40 Z60

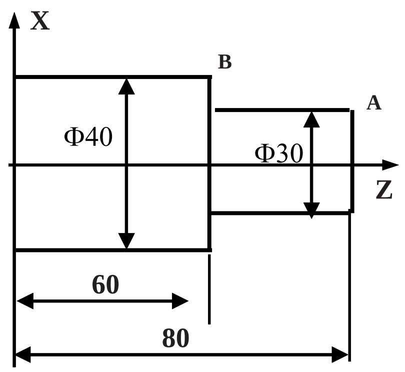


Figure 1.16 Diameter Programming

Example: Describe the points by radius programming.

A corresponds to X15 Z80

B corresponds to X20 Z60

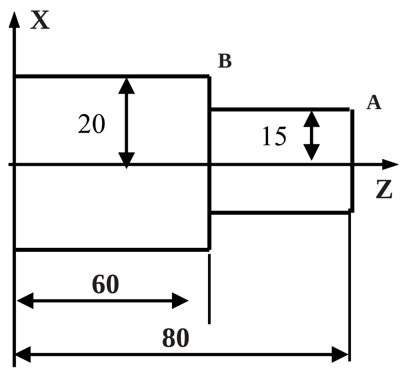


Figure 1.17 Radius Programming

1.5 Spindle Speed Function

Spindle speed function (s) is to control spindle rotation speed. The numerical value of following S refers to spindle speed, and the unit of spindle speed is r/min.

The constant surface speed control refers to the specified cutting speed. The unit is m/min (G96 starts constant surface speed, G97 constant surface speed is cancelled, and G46 setting the limit of spindle speed).

S is modal command, i.e. Function S is effective until the another spindle speed is set.

The spindle speed can be set by the spindle override switch on NC control board.

1.6 Tool Function

1.6.1 Tool Selection

It is necessary to select a suitable tool when drilling, tapping, boring or the like is performed. As it is shown in Figure 1.18, a number is assigned to each tool. Then this number is used in the program to specify that the corresponding tool is selected.

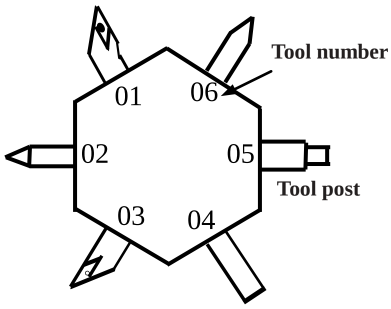


Figure 1.18 Tool Selection

1.6.2 Tool Offset

When writing a program, the operator just use the workpiece dimensions according to the dimensions in the part drawing. The tool nose radius center, the tool direction of the turning tool, and the tool length are not taken into account. However, when machining a workpiece, the tool path is affected by the tool geometry.

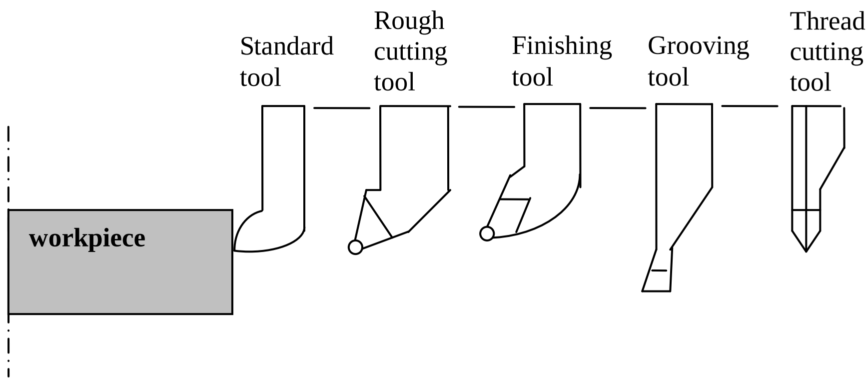


Figure 1.19 Tool Offset

- Tool Length Compensation

There are two kind of ways to specify the value of tool length compensation.

- Absolute value of tool length compensation (the distance between tool tip and machine reference point)
- Incremental value of tool length compensation (the distance between tool tip and the standard tool)

As it is shown in Figure 1.20, L1 is the tool length on X axis. L2 is the tool length on Z axis. It should be noted that the tool wear values on X axis or Z axis are also contained in the tool length compensation.

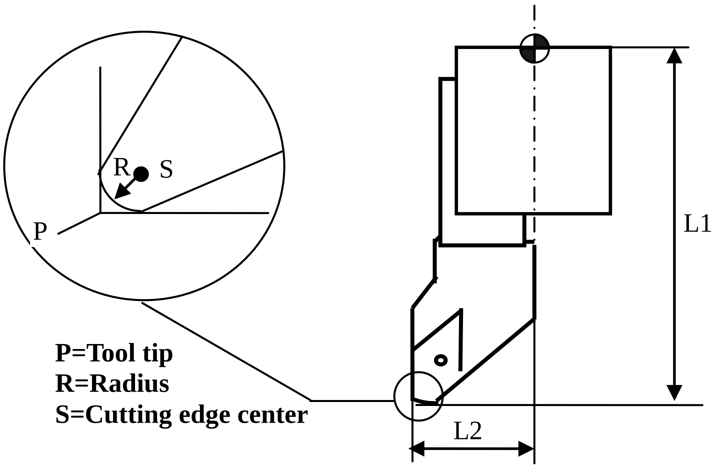


Figure 1.20 Tool Length Compensation

- Tool Radius Compensation

Figure 1.21 shows the imaginary tool nose as a start position when writing a program.

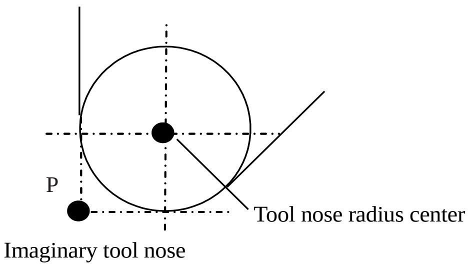


Figure 1.21 The imaginary tool nose

The direction of imaginary tool nose is determined by the tool direction during cutting. Figure 1.22 and Figure 1.23 show the relation between the tool and the imaginary tool tip.

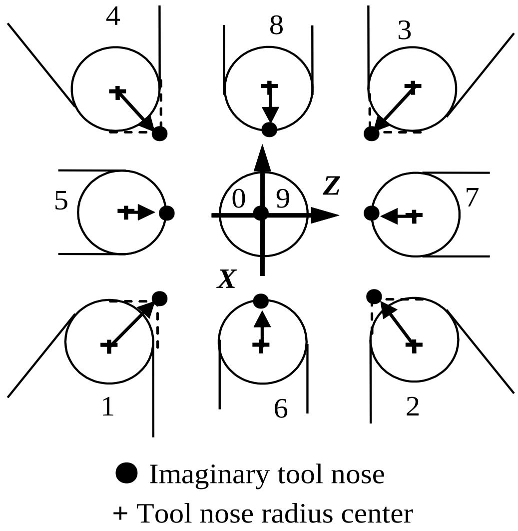


Figure 1.22 The direction of imaginary tool nose (1)

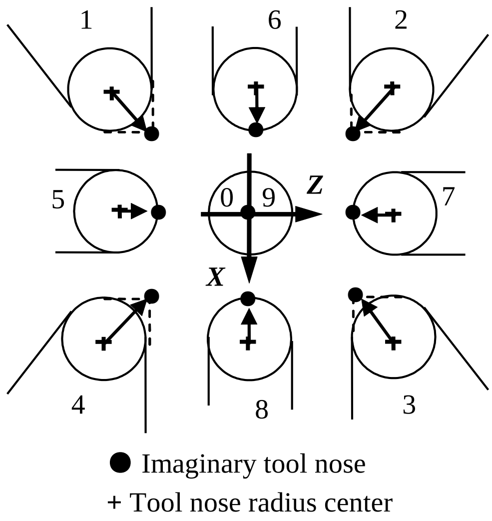


Figure 1.23 The direction of imaginary tool nose (2)

1.7 Miscellaneous Function

Miscellaneous function refers to the operation to control the spindle, feed, and coolant. In general, it is specified by an M code.

When a move command and M code are specified in the same block, there are two ways to execute these commands:

1) Pre-M function

   M command is executed before the completion of move command

2) Post-M function

   M command is executed after the completion of move command.

The sequence of the execution depends on the specification of the machine tool builder.

1.8 Program Configuration

1.8.1 Structure of an NC Program

As it is shown in Figure 1.24, an NC program consists of a sequence of NC **blocks**. Each block is one of machining steps. **Commands** in each block are the instruction.

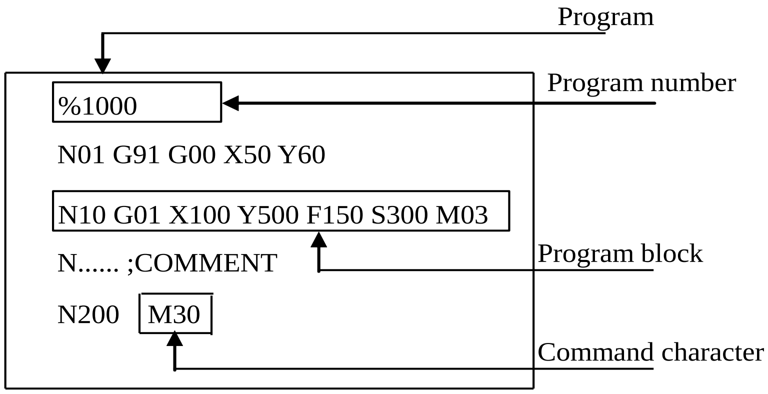


Figure 1.24 Structure of an NC Program

- Format of **program name**

The program name must be specified in the format OXXXX (X could be letters or numbers).

- Format of **program number**

The program number should be started with %XXXX or OXXXX (X could be numbers only).

- Format of **blocks**

A block starts with the program block number.

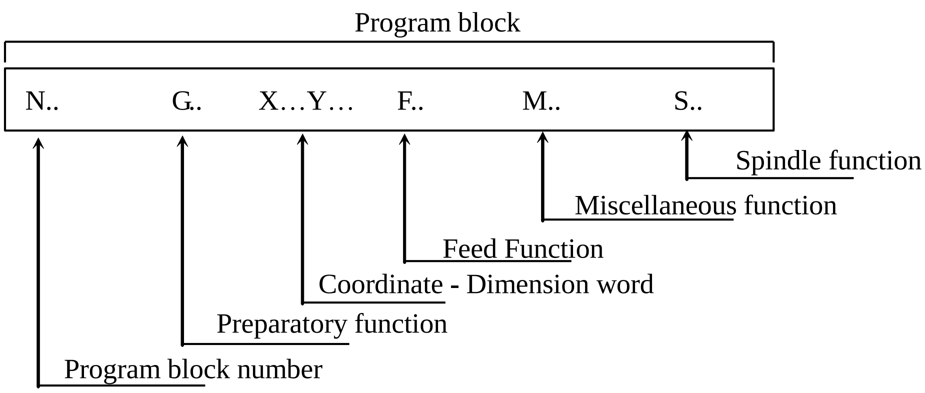


Figure 1.25 Structure of Block

- Format of **end of program**

The last block should contain M02 or M03 to indicate the end of program.

- Format of **Comments**

All information after the ";" is regarded as comments.

All information between "( )" is regarded as comments.

1.8.2 Main Program and Subprogram

There are two type of program: main program and subprogram. The CNC operates according to the main program. When a execution command of subprogram is at the execution line of the main program, the subprogram is called. When the execution of subprogram is finished, the system returns control to the main program.

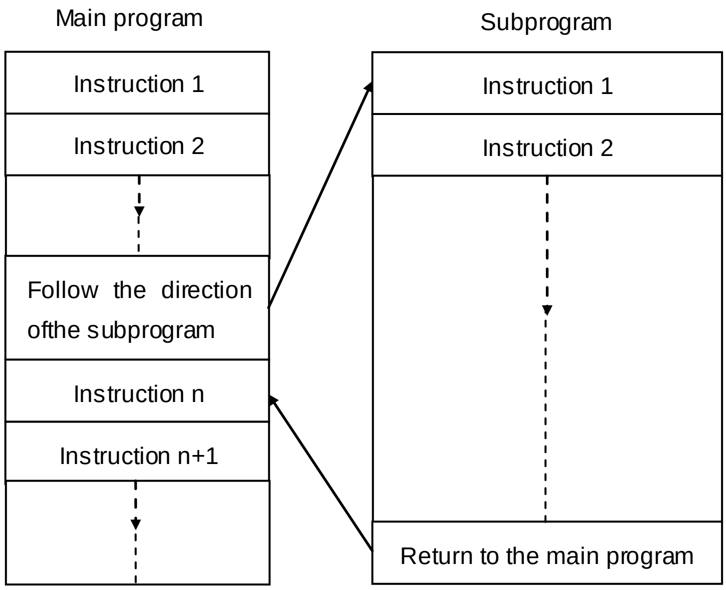


Figure 1.26 Main program and subprogram

Note:

Main program and its subprogram must be written in a same file with a different program codes.

2 Preparatory Function (G code)

There are two types of G code: one-shot G code, and modal G code.

Table 2-1 Type of G code

Type	**Meaning**
One-shot G code	The G code is only effective in the block in which it is specified
Modal G code	The G code is effective until another G code is specified.

Example: G01 and G00 are modal G codes.

```
G00X_
   Z_    G00 is effective in this range
   X_
G01Z_
```

2.1 G code List

The following table is the list of G code in HNC system.

Table 2-2 G code list

G code	Group	Function
G00 ◣G01 G02 G03	01	Positioning (Rapid traverse) Linear interpolation (Cutting feed) Circular interpolation CW Circular interpolation CCW
G04	00	Dwell
G20 ◣G21	08	Input in inch Input in mm
G28 G29	00	Reference point return Auto return from reference point
G32 G34	01	Thread cutting with constant lead Tapping
◣G36 G37	17	Diameter programming Radius programming
◣G40 G41 G42	09	Tool nose radius compensation cancel Tool nose radius compensation on the left Tool nose radius compensation on the right
G46	16	Setting the limit of spindle speed
◣G50 G51	04	Canceling the workpiece's origin movement Moving the origin of workpiece coordinate system
G53	00	Selecting a machine coordinate system
◣G54 G55 G56 G57 G58 G59	11	Setting a workpiece coordinate system

G71	06	Stock Removal in Turning
G72		Stock Removal in Facing
G73		Pattern repeating
G74		Front drilling cycle
G75		Side drilling cycle
G76		Multiple thread cutting cycle
G80		Internal diameter/Outer diameter cutting cycle
G81		End face turning cycle
G82		Thread cutting cycle
◣G90	13	Absolute programming
G91		Incremental programming
G92	00	Setting a coordinate system
◣G94	14	Feedrate per minute
G95		Feedrate per revolution
G96	16	Constant cutting speed starts
◣G97		Constant cutting speed is cancelled

Explanation:

1) G codes in 00 group are one-shot G code, while the other groups are modal G code.
2) ◣ means that it is default setting.

3 Interpolation Functions

This chapter would introduce:

1) Positioning Command (G00)
2) Linear Interpolation (G01)
3) Circular Interpolation (G02, G03)
4) Chamfering and Rounding (G01, G02, G03)
5) Thread Cutting with Constant Lead (G32)
6) Tapping (G34)
7) Direct Drawing Dimension Programming (G01)

3.1 Positioning (G00)

Programming

G00 X(U)… Z(W)…

Explanation of the parameters

X, Z Coordinate value of the end point in the absolute command

U, W Coordinate value of the end point in the incremental command

Function

The tool is moved at the highest possible speed (rapid traverse). If the rapid traverse movement is required to execute simultaneously on several axes, the rapid traverse speed is decided by the axis which takes the most time. The operator can use this function to position the tool rapidly, to travel around the workpiece, or to approach the tool change position.

Example

Move tool from P1 (45, 90) to P2 (10, 20) at the rapid traverse speed.

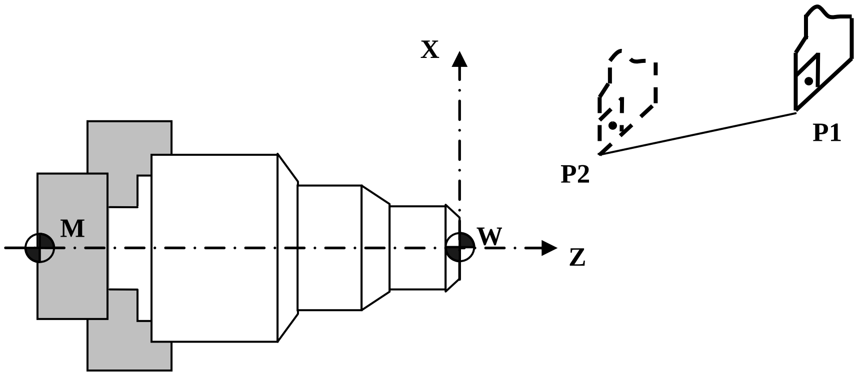


Figure 3.1 Positioning (Rapid Traverse)

Absolute programming:

G00 X10 Z20

Incremental programming:

G00 U30 W70

3.2 Linear Interpolation (G01)

Programming

G01 X(U)… Z(W)… F…

Explanation of the parameters

X, Z	Coordinate value of the end point in the absolute command
U, W	Coordinate value of the end point in the incremental command
F	Feedrate. It is effective until a new value is specified.

Function

The tool is moved along the straight line at the specified feedrate.

Example 1

Use G01 command to rough machining and finish machining the simple cylinder part.

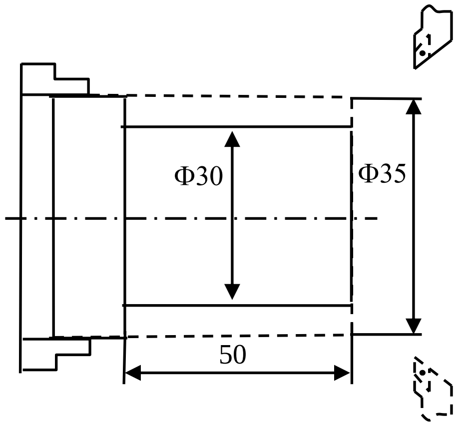


Figure 3.2 Linear Interpolation – Example 1

%3306（Absolute command）	%3306（Incremental command）
N1 T0106	N1 T0101
N2 M03 S460	N2 M03 S460
N3 G00 X90Z20	N3 G00 X90Z20
N4 G00 X31Z3	N4 G00 X31Z3
N5 G01 Z-50 F100	N5 G01 W-53 F100
N6 G00 X36	N6 G00 U5
N7 Z3	N7 W53
N8 X30	N8 U-6
N9 G01 Z-50 F80	N9 G01 Z-50 F80
N10 G00 X36	N10 G00 X36
N11 X90 Z20	N11 X90 Z20
N12 M05	N12 M05
N13 M30	N13 M30

Example 2

Use G01 command to rough machining and finish machining simple conical part.

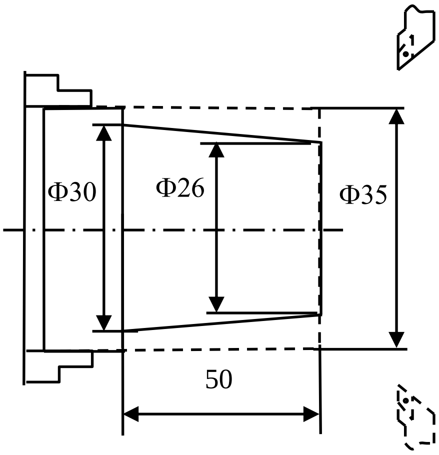


Figure 3.3 Linear Interpolation – Example 2

%3307

N1 T0101

N2 M03 S460

N3 G00 X100Z40

N4 G00 X26.6 Z5

N5 G01 X31 Z-50 F100

N6 G00 X36

N7 X100 Z40

N8 T0202

N9 G00 X25.6 Z5

N10 G01 X30 Z-50 F80

N11 G00 X36

N12 X100 Z40

N13 M05

N14 M30

Example 3

Use G01 command to rough machining and finish machining the part.

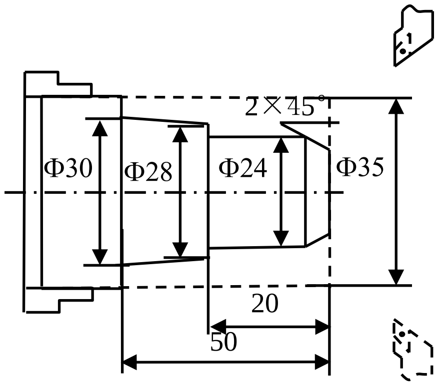


Figure 3.4 Linear Interpolation – Example 3

```
%3308
N1 T0101
N2 M03 S450
N3 G00 X100 Z40
N4 G00 X31 Z3
N5 G01 Z-50 F100
N6 G00 X36
N7 Z3
N8 X25
N9 G01 Z-20 F100
N10 G00 X36
N11 Z3
N12 X15
N13 G01 U14 W-7 F100
N14 G00 X36
N15 X100 Z40
N16 T0202
N17 G00 X100Z40
N18 G00 X14 Z3
N19 G01 X24 Z-2 F80
N20 Z-20
N21 X28
N22 X30 Z-50
N23 G00 X36
N24 X80 Z10
N24 M05
N25 M30
```

3.3 Circulation Interpolation (G02, G03)

Programming

$$\begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X(U)\_Z(W)\_ \begin{Bmatrix} I\_K\_ \\ R\_ \end{Bmatrix} F\_$$

Explanation of the parameters

G02 a circular path in clockwise direction (CW)

G03 a circular path in counterclockwise direction (CCW)

X, Z Coordinate values of the circle end point in absolute command

U, W Coordinate values of the circle end point with reference to the circle starting point in incremental command.

I, K Coordinate values of the circle center point with reference to the circle starting point in incremental command.

R Circle radius. R is valid when I, K, R are all specified in this command.

F Feedrate

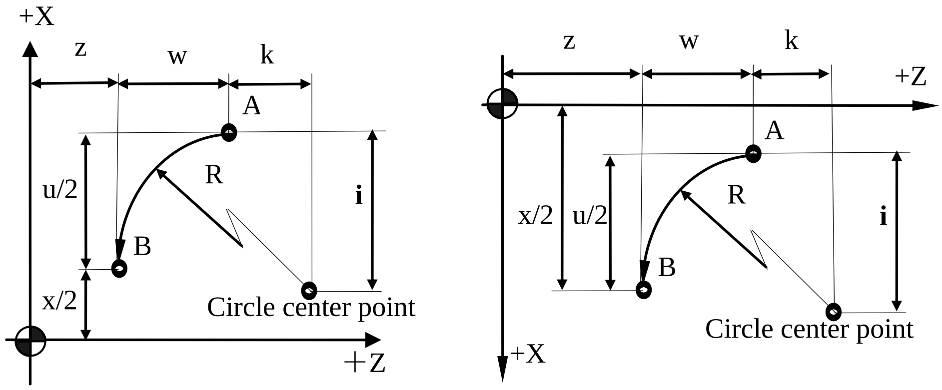


Figure 3.5 Description of G02/G03 parameter

G02 and G03 are defined when the working plane is specified. Figure 3.6 shows the direction of circular interpolation.

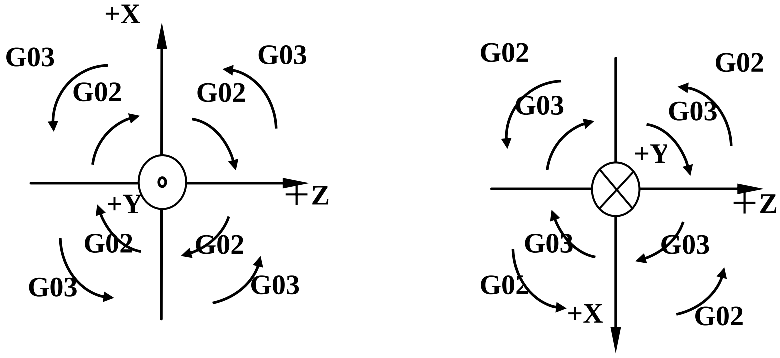


Figure 3.6 Direction of Circular Interpolation

Function

The tool is moved along a full circle or arcs.

Example 1

Use the circular interpolation command to program

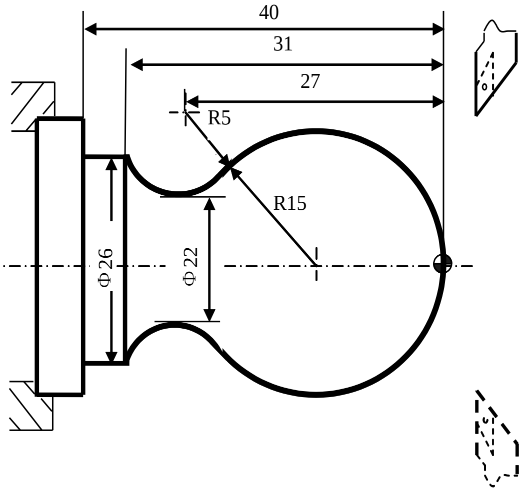


Figure 3.7 Circular Interpolation – Example 1

%3309

N1 T0101

N2 G00 X40 Z5

N3 M03 S400

N4 G00 X0

N5 G01 Z0 F60

N6 G03 U24 W-24 R15

N7 G02 X26 Z-31 R5

N8 G01 Z-40

N9 X40 Z5

N10 M30

Example 2

Use the circular interpolation command to program

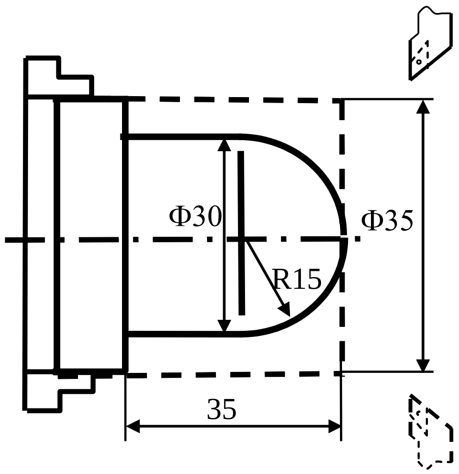


Figure 3.8 Circular Interpolation – Example 2

%3310（Absolute programming）

N1 T0101

N2 M03 S460

N3 G00 X90Z20

N4 G00 X0 Z3

N5 G01 Z0 F100

N6 G03 X30 Z-15 R15

N7 G01 Z-35

N8 X36

N9 G00 X90 Z20

N10 M05

N11 M30

%3310（Incremental programming）

N1 T0101

N2 M03 S460

N3 G00 X90Z20

N4 G00 U-90 W-17

N5 G01 W-3 F100

N6 G03 U30 W-15 R15

N7 G01 W-20

N8 X36

N9 G00 X90 Z20

N10 M05

N11 M30

Example 3

Use the circular interpolation command to program.

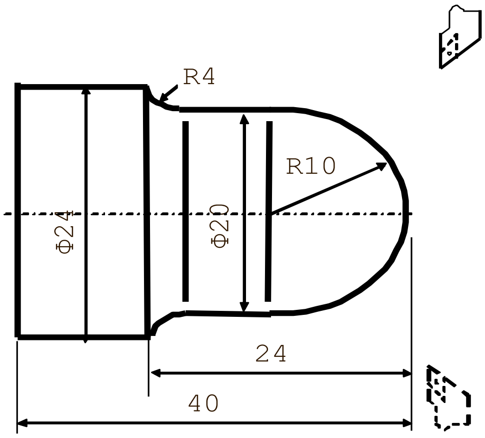


Figure 3.9 Circular Interpolation – Example 3

```
%3311
N1 T0101
N2 M03 S460
N3 G00 X100 Z40
N4 G00 X0 Z3
N5 G01 Z0 F100
N6 G03 X20 Z-10 R10
N7 G01 Z-20
N8 G02 X24 Z-24 R4
N9 G01 Z-40
N10 G00 X30
N11 X100 Z40
N12 M05
N13 M30
```

Example 4

Use the circular interpolation command to program

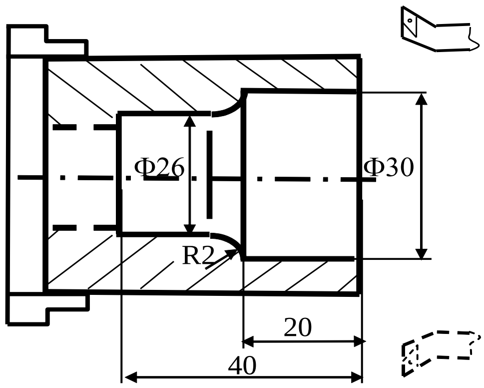


Figure 3.10 Circular Interpolation – Example 4

```
%3312
N1 T0101
N2 M03 S460
N3 G00 X80 Z10
N4 G00 X30 Z3
N5 G01 Z-20 F100
N6 G02 X26 Z-22 R2
N7 G01 Z-40
N8 G00 X24
N9 Z3
N10 X80 Z10
N11 M05
N12 M30
```


3.4 Chamfering and Rounding (G01, G02, G03)

Note: These commands cannot be used in thread cutting.

3.4.1 Chamfering (G01)

Programming

G01 X(U)_ Z(W)_ C_

Explanation of the parameters

X, Z Coordinate values of the intersection (point G) in absolute command

U, W Coordinate values of the intersection (point G) in incremental command

C Width of chamfer in original direction of movement (c)

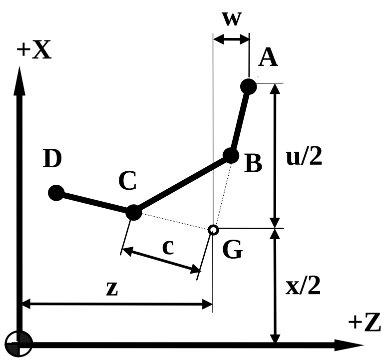


Figure 3.11 Chamfering (G01)

Function

A chamfer can be inserted between two blocks which intersect at a right angle (point A→B→C).

Note: The length of GA should be more than the length of GB

3.4.2 Rounding (G01)

Programming

G01 X(U)_ Z(W)_ R_

Explanation of the parameters

X, Z Coordinate values of the intersection (point G) in absolute command

U, W Coordinate values of the intersection (pint G) in incremental command

R Radius of the rounding (r)

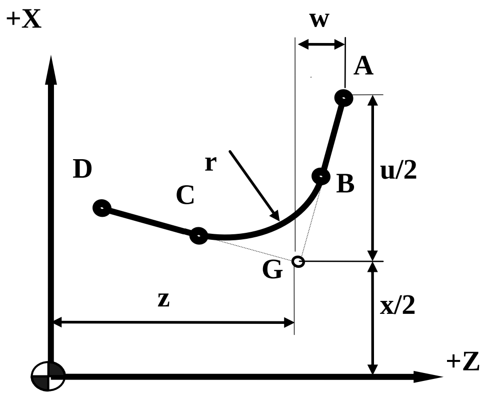


Figure 3.12 Rounding (G01)

Function

A corner can be inserted between two blocks which intersect at a right angle (point A→B→C).

Note: The length of GA should be more than the length of GB

Example

Use the chamfering and rounding command (G01):

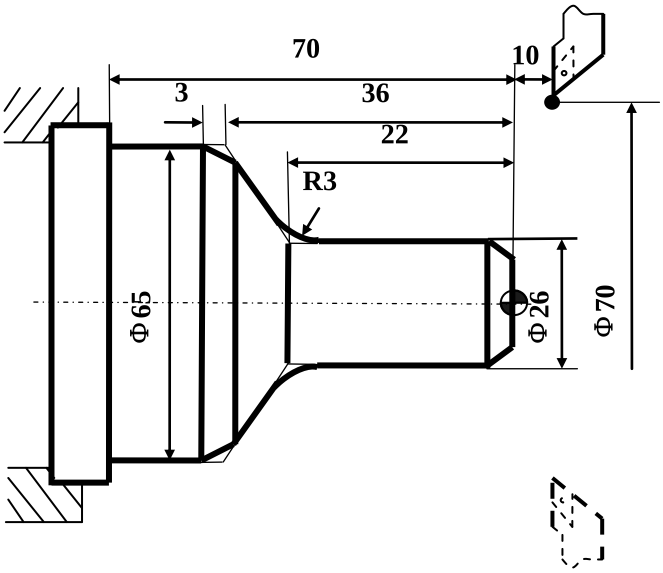


Figure 3.13 Chamfering and Rounding (G01) - Example

%3314

N1 M03 S460

N2 G00 U-70 W-10

N3 G01 U26 C3 F100

N4 W-22 R3

N5 U39 W-14 C3

N6 W-34

N7 G00 U5 W80

N8 M30

3.4.3 Chamfering (G02, G03)

Programming

$$\begin{Bmatrix} G02 \\ G03 \end{Bmatrix} X(U)\_\ Z(W)\_\ R\ \_\ \ RL=\_$$

Explanation of the parameters

X, Z Coordinate values of the intersection (point G) in absolute command

U, W Coordinate values of the intersection (point G) with reference to the circle starting point (point A) in incremental command

R Circle Radius (r)

RL= Width of chamfer in original direction of movement (RL)

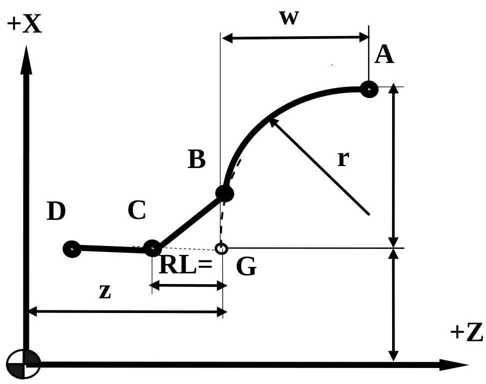


Figure 3.14 Chamfering (G02/G03)

Function

A chamfer can be inserted between two blocks which intersect at a right angle (point A→B→C).

Note: RL must be capitalized letters.

3.4.4 Rounding (G02, G03)

Programming

$$\begin{Bmatrix} \text{G02} \\ \text{G03} \end{Bmatrix} \text{X}(U)\_\ \text{Z}(W)\_\ \text{R}\ \_\ \ \text{RC} = \_$$

Explanation of the parameters

X, Z Coordinate values of the intersection (point G) in absolute command

U, W Coordinate values of the intersection (point G) with reference to the circle starting point (point A) in incremental command

R Circle radius (r)

RC Radius of rounding (rc)

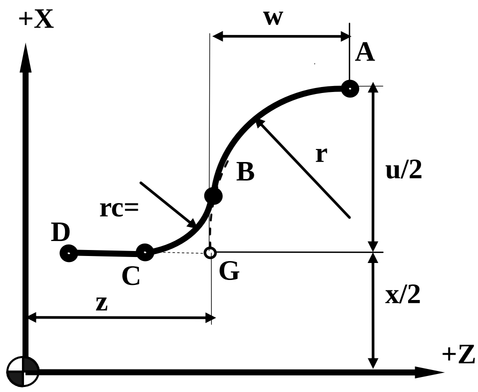


Figure 3.15 Rounding (G02/G03)

Function

A corner can be inserted between two blocks which intersect at a right angle (point A→B→C).

Note: RC must be capitalized letters.

Example

Use the chamfering and rounding command (G02/G03):

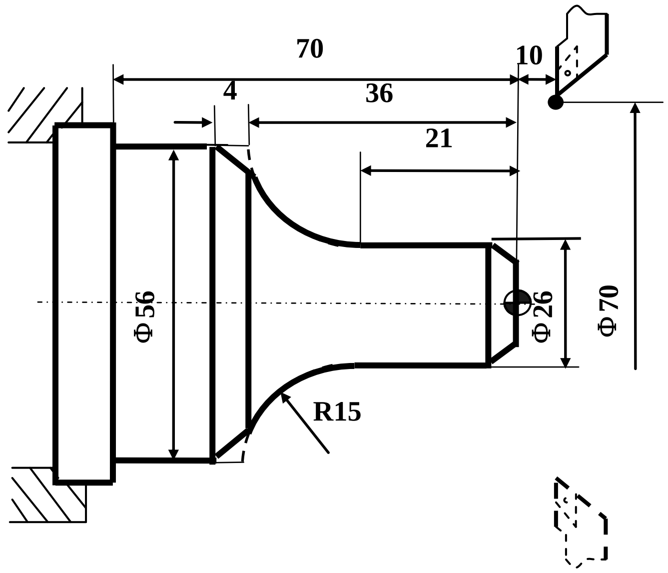


Figure 3.16 Chamfering and Rounding (G02/G03) - Example

%3315

N1 T0101

N2 G00 X70 Z10 M03 S460

N3 G00 X0 Z4

N4 G01 W-4 F100

N5 X26 C3

N6 Z-21

N7 G02 U30 W-15 R15 RL=4

N8 G01 Z-70

N9 G00 U10

N10 X70 Z10

N11 M30

3.5 Thread Cutting with Constant Lead (G32)

Programming

G32 X(U)__Z(W)__R__E__P__F/I__Q__

Explanation of the parameters

X, Z Coordinate values of end point in absolute command

U, W Coordinate values of end point with reference to the starting point in incremental command

F Thread lead i.e. the feed of tool with reference to the tool at one spindle revolution

I Thread lead at inch measurement. Unit: threads/inch

R, E Retraction amount of thread cutting. R is the retraction amount on axis Z. E is the retraction amount on axis X. They all use the incremental command in absolute or incremental programming. The positive R or E means the positive retraction on axis Z or X. The negative R or E means the negative retraction on axis Z of X. The retraction slot can be ignored when using R or E. When there is no R or E, it means that the retraction function is not validated. In general, R is set as two times value of thread lead, and E is set as the thread height.

P Spindle angle of thread cutting start point at the spindle reference pulses

Q

1) Acceleration constant of thread cutting retraction. When it is set to zero, the acceleartion is maximum. The more this value is, the acceleration time is longer, and the retraction is longer. Q must be set to zero or more than zero.
2) When there is no Q, the set acceleration constant on each axis is used in the retraction.
3) R and E must be set when the retraction function is required.
4) The retraction ratio of minor axis:major axis should not be more than “20“.
5) Q is one-shot G code.

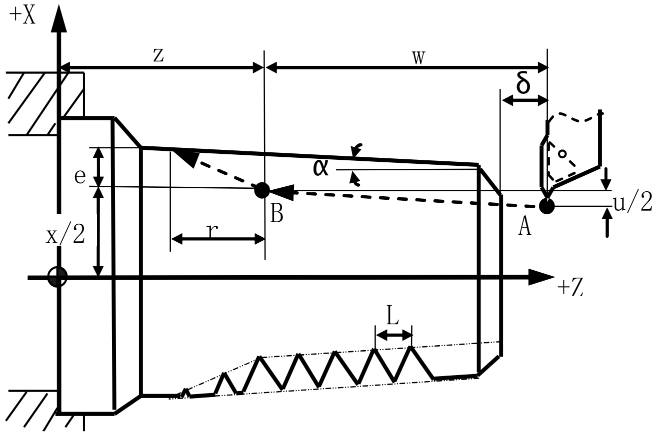


Figure 3.17 Thread Cutting with Constant Lead (G32)

Function

Cylindrical thread, taper thread and face thread can be machined with G32.

Thread cutting is form turning and the feed is much. If the tool intensity is low, it is required to feed cutting at serveral times. The following table is the general feed times and amount of thread cutting.

Table 3-1 feed times and amount of thread cutting

Thread in metric measurement								
Lead		1.0	1.5	2	2.5	3	3.5	4
Threads (radius)		0.649	0.974	1.299	1.624	1.949	2.273	2.598
feed times and amount (diameter)	Once	0.7	0.8	0.9	1.0	1.2	1.5	1.5
	Twice	0.4	0.6	0.6	0.7	0.7	0.7	0.8
	Three	0.2	0.4	0.6	0.6	0.6	0.6	0.6
	Four		0.16	0.4	0.4	0.4	0.6	0.6
	Five			0.1	0.4	0.4	0.4	0.4
	Six				0.15	0.4	0.4	0.4
	Seven					0.2	0.2	0.4
	Eight						0.15	0.3
	Nine							0.2
Thread in inch measurement								
Threads/in		24	18	16	14	12	10	8
Threads (radius)		0.678	0.904	1.016	1.162	1.355	1.626	2.033
feed times and amount (diameter)	Once	0.8	0.8	0.8	0.8	0.9	1.0	1.2
	Twice	0.4	0.6	0.6	0.6	0.6	0.7	0.7
	Three	0.16	0.3	0.5	0.5	0.6	0.6	0.6
	Four		0.11	0.14	0.3	0.4	0.4	0.5
	Five				0.13	0.21	0.4	0.5
	Six						0.16	0.4
	Seven							0.17

Note:

1) The spindle speed should remain constant during rough cutting and finish cutting.
2) The feed hold function is ineffective during the thread cutting. Even though the “feed hold” button is pressed, it is effective until the thread cutting is done.
3) It is not recommended to use the constant surface speed control during the thread cutting.
4) Allowant amount must be specified to avoid the error.

Example

Given that F=1.5mm, δ=1.5mm, δ'=1mm, cutting for four times and each cutting depth is separately: 0.8mm, 0.6 mm, 0.4mm, 0.16mm. It is diameter programming.

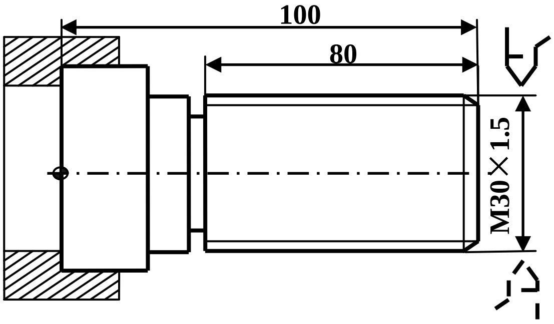


Figure 3.18 Thread Cutting – Example

```
%3316
N1 T0101
N2 G00 X50 Z120
N3 M03 S300
N4 G00 X29.2 Z101.5
N5 G32 Z19 F1.5
N6 G00 X40
N7 Z101.5
N8 X28.6
N9 G32 Z19 F1.5
N10 G00 X40
N11 Z101.5
N12 X28.2
N13 G32 Z19 F1.5
N14 G00 X40
N15 Z101.5
N16 U-11.96
N17 G32 W-82.5 F1.5
N18 G00 X40
N19 X50 Z120
N20 M05
N21 M30
```


3.6 Tapping (G34)

Programming

G34 K_ F_ P_

Explanation of the parameters

K The distance from the starting point to the bottom of the hole

F Thread lead

P Dwell time at the bottom of a hole

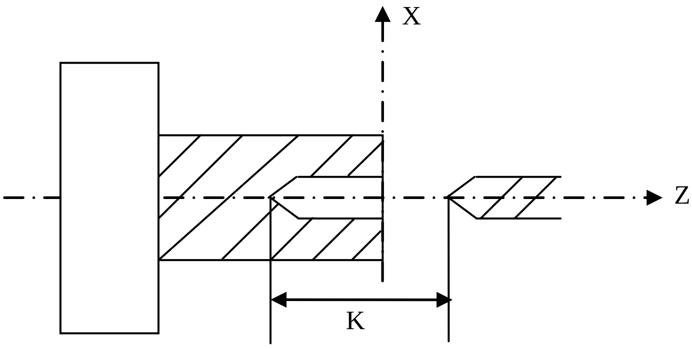


Figure 3.19 Rigid Tapping

Function

With this command, the operator can rigid tap a thread.

In general, there is overshoot of the tap at the bottom of the thread during the spindle-braking portion of the tapping cycle. It can be set by PMC parameters (Table 3-1) to eliminate the overshoot errors.

Table 3-2 PMC parameters

CNC system	PMC parameters	
HNC 18/19i	#0062	Maximum spindle speed during tapping
	#0063	Minimum spindle speed during tapping
	#0064	Dwelled unit for tapping
	#0065	Optional dwelled unit for tapping
HNC 21/22	#0017	Maximum spindle speed during tapping
	#0018	Minimum spindle speed during tapping
	#0019	Dwelled unit for tapping
	#0030	Optional dwelled unit for tapping

Optional dwelled unit for tapping is only effective when “dwelled unit for tapping” is assigned to “0”. Moreover, it is not necessary to restart the system.

The following formular is to calculate the dwelled unit (X):

D = (S * S / C) * X / 10000 = L * 360 / F

- D dwelled amount
- S spindle speed
- C Transmission gear ratio
- X dwelled unit
- L overshoot error
- F thread lead

Since the workpiece is chucked on the spindle, the spindle decceleration time of turning machine is more than a milling machine’s. The quicker the spindle rotates, the quicker the feedrate on Z axis is, and then the more time the decceleration time takes. Thus, the spindle speed should be set accoording to the thread length.

Example

The following is a tested data for tapping when the thread lead is 1.25mm.

```
%0034
T0101
S100
G90G1X0Z0F500
G34K-10F1.25P2
S200
G90G1X0Z0F500
G34K-10F1.25P2
S300
G90G1X0Z0F500
G34K-10F1.25P2
S400
G90G1X0Z0F500
G34K-20F1.25P2
S500
G90G1X0Z0F500
G34K-30F1.25P3
S600
G90G1X0Z0F500
G34K-40F1.25P3
S700
G90G1X0Z0F500
G34K-50F1.25P3
S800
G90G1X0Z0F500
G34K-50F1.25P2
S1000
G90G1X0Z0F500
G34K-60F1.25P3
M30
```


3.7 Direct Drawing Dimension Programming (G01)

Angles of straight lines, chamfering value, corner rounding values, and other dimensional values on machining drawings can be programmed by directly inputting these values. In addition, the chamfering and corner rounding can be inserted between straight lines having an optional angle. It is called direct drawing dimension programming.

This programming is only valid in turning system G01.

3.7.1 Instruct a line

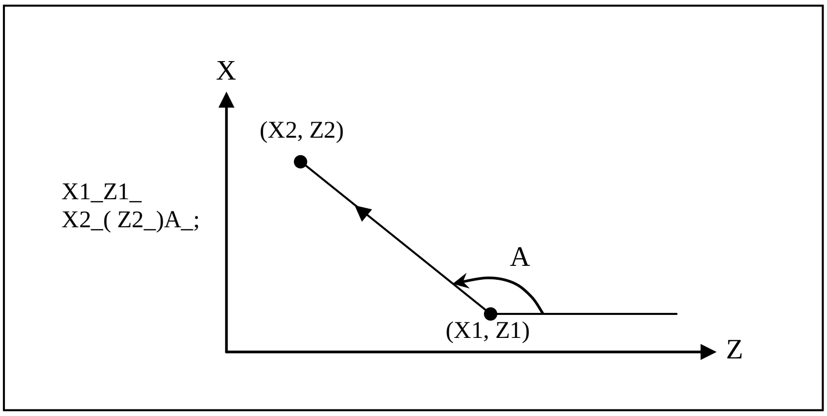
Programming

G01 X_Z_A_

Explanation of the parameters

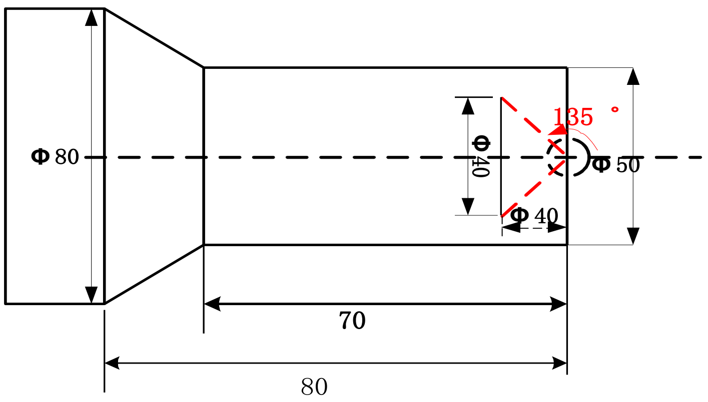
X_Z_: Line location address;

A_: Angle between linear motion direction and the positive Z-axis direction. Counter clockwise is positive, while clockwise is negative. Unit: degree.



Note: the target position only needs to specify a movement value in one direction. E.g.:Z50a45 or X100a45

Example



%3324

N1 T0101

N2 M03S400

N3 G00 X100Z40

N4 G00 X0Z0

N5 G01X40A135

N6 G00 X100Z40

N7 M30

3.7.2 Rounding

Programming

G01 X_Z_R_

G01 X_Z_

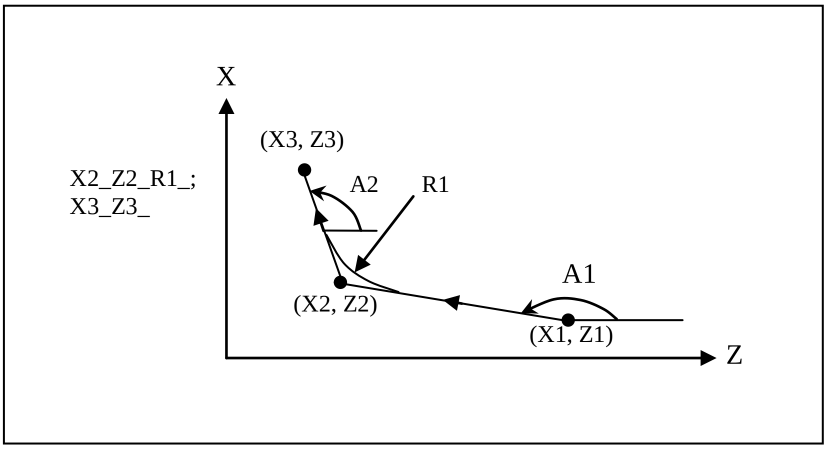
Explanation of the parameters

X_/Z_: Line location address;

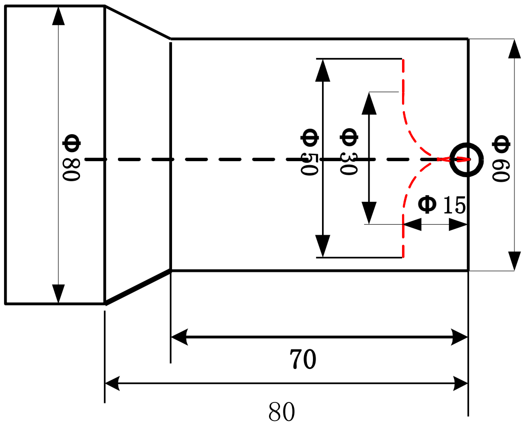
R_: Rounding radius;

Function

An arc is inserted between two linear interpolations. This arc is tangent to the two lines.



Example



%3325

N1 T0101

N2 M03 S400

N3 G00 X100 Z40

N4 X0 Z0

N5 G01 X0 Z-15 R15

N6 G01X50 Z-15

N7 G00X100 Z40

N8 M30

3.7.3 Chamfering

Programming

G01 X_Z_C_

G01 X_Z_

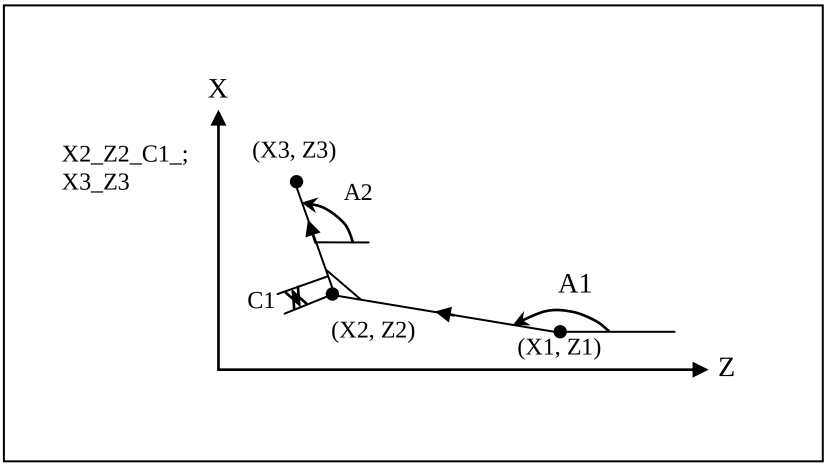
Explanation of the parameters

X_Z_: Line location address;

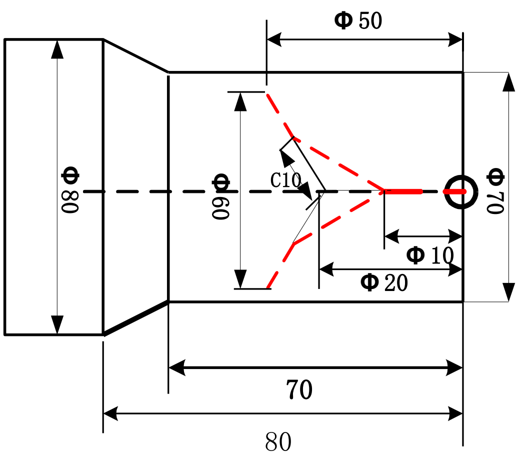
C_: Chamfer edge length;

Function

Chamfering is inserted between two linear interpolations.



Example



%3326

N1 T0101

N2 M03 S400

N3 G00 X100 Z40

N4 X0 Z0

N5 G01 X0 Z-20 C10

N6 G01X60Z-50

N7 G00X100 Z40

N8 M30

3.7.4 Continuous Rounding

Programming

G01 X_Z_R_

G01 X_Z_R_

G01 X_Z_

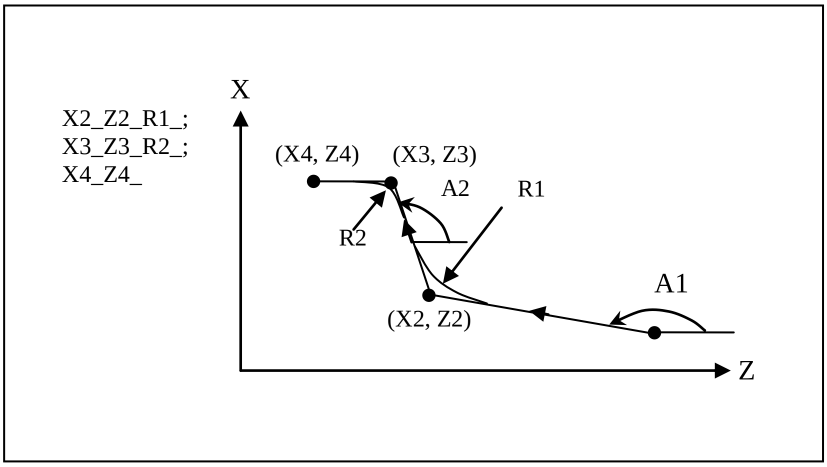
Explanation of the parameters

X_/Z_: Line location address;

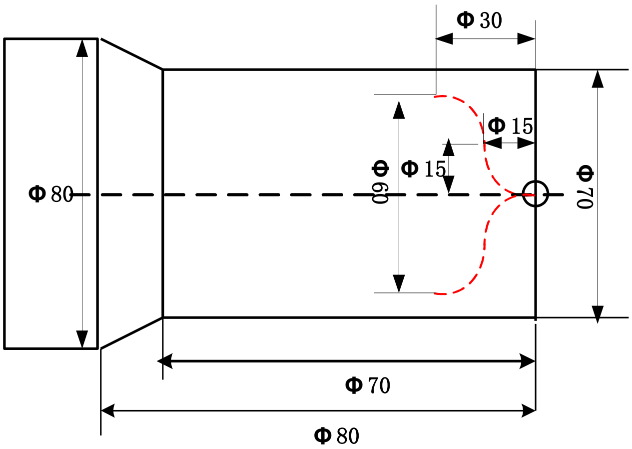
R_: Rounding radius;

Function

Circular Interpolation is continuously inserted between two linear interpolations.



Example



%3327

N1 T0101

N2 M03 S400
N3 G00 X100 Z40
N4 X0 Z0
N5 G01 X0 Z-15 R15 ;the first rounding
N6 G01 X60 Z-15 R15 ;the second rounding
N7 G01 X60 Z-30
N8 G00X100 Z40
N9 M30

3.7.5 Continuous Chamfering

Programming

G01 X_Z_C_

G01 X_Z_C_

G01 X_Z_

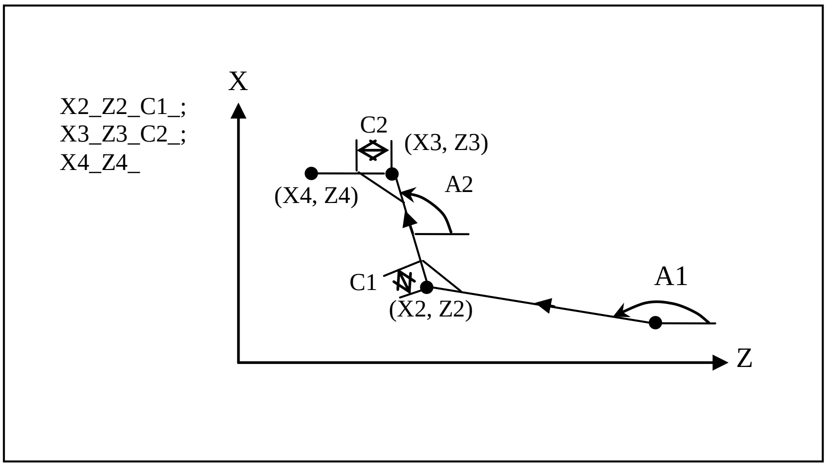
Explanation of the parameters

X_/Z_: Line location address;

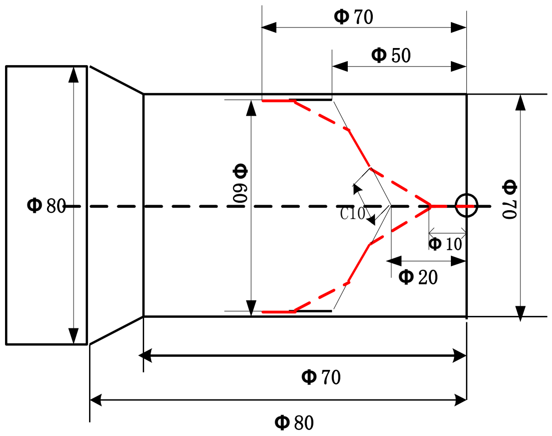
C_: Chamfer edge length;

Function

Chamfering is continuous inserted between two linear interpolation.



Example



%3328

N1 T0101

N2 M03 S400

N3 G00 X100 Z40

N4 X0 Z0

N5 G01 X0 Z-20C10 ;the first chamfering

N6 G01 X60 Z-50C10 ;the second chamfering

N7 G01 X60 Z-70

N8 G00X100 Z40

N9 M30

3.7.6 Rounding then Chamfering

Programming

G01 X_Z_R_

G01 X_Z_C_

G01 X_Z_

Explanation of the parameters

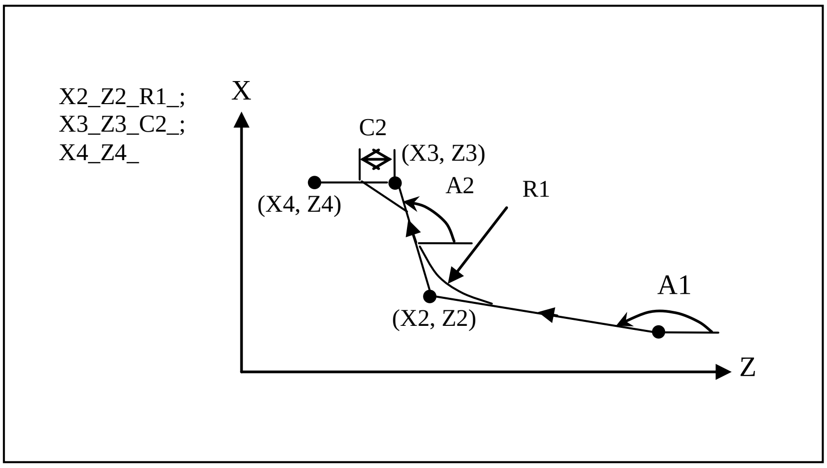
X_/Z_: Line location address;

R_: Rounding radius;

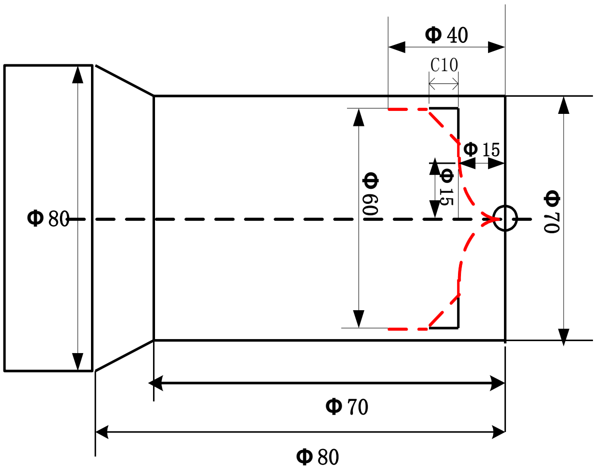
C_: Chamfer edge length;

Function

Round and Chamfering are inserted between two linear interpolation.



Example



%3329

N1 T0101

N2 M03 S400

N3 G00 X100 Z40

N4 X0 Z0

N5 G01 X0 Z-15 R15 ;rounding

N6 G01 X60 Z-15C10 ;chamfering

N7 G01 X60 Z-40

N8 G00X100 Z40

N9 M30

3.7.7 Chamfering then Rounding

Programming

G01 X_Z_C_

G01 X_Z_R_

G01 X_Z_

Explanation of the parameters

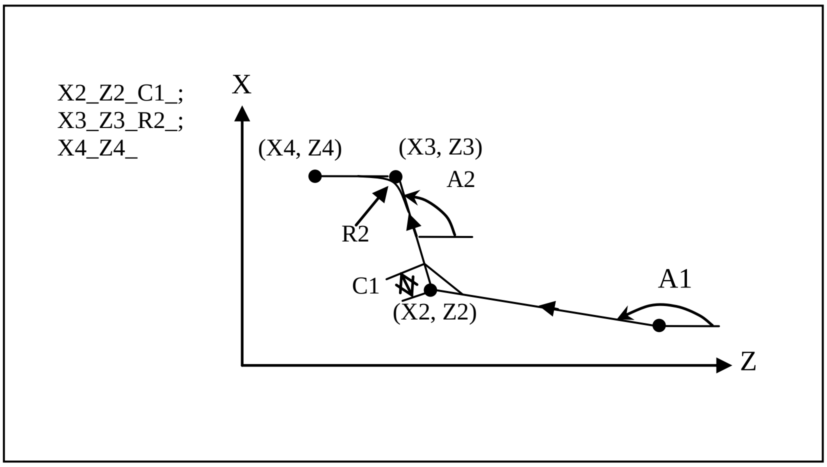
X_/Z_: Line location address;

R_: Rounding radius;

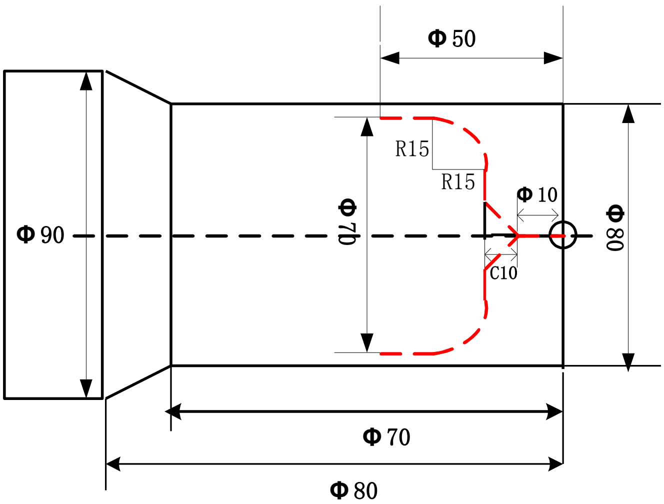
C_: Chamfer edge length;

Function

Round and Chamfering are inserted between two linear interpolation.



Example



```
%3330
N1   T0101
N2   M03 S400
N3   G00 X100 Z40
N4   X0 Z0
N5   G01 X0 Z-20 C10     ;Chamfering
N6   G01 X70 Z-20 R15    ;Rounding
N7   G01 X70 Z-50
N8   G00 X100Z40
N8   M30
```


4 Feed Function

There are two kinds of feed functions:

1. Rapid Traverse

   The tool is moved at the rapid traverse speed set in CNC.

2. Cutting Feed

   The tool is moved at the programmed cutting feedrate.

Moreover, this chapter would introduce “Dwell”.

4.1 Rapid Traverse (G00)

Positioning command (G00) is to move the tool at the rapid traverse speed (the highest possible speed).

This rapid traverse speed can be controlled by the machine control panel. For more detailed information, please refer to turning operation manual.

4.2 Cutting Feed (G94, G95)

Programming

G94 [F_]

G95 [F_]

Explanation of the parameters

G94 feedrate per minute.

On linear axis, the unit of feedrate is mm/min, or in/min.

On rational axis, the unit of feedrate is degree/min.

G95 feedrate per revolution

The unit of feedrate is mm/rev, or in/rev.

Note:

1) G94 is the default setting
2) G95 is only used when there is spindle encoder.

Function

The feedrate can be set by G94 or G95.

4.3 Dwell (G04)

Programming

G04 P_

Explanation of the parameters

P dwell time (specified in seconds)

Function

It can be used to interrupt machining to get the smooth surface. It can be used to control the groove cutting, drilling, and turning path.

5 Coordinate System

This chapter would introduce:

1) Reference Position Return (G28)
2) Auto Return from Reference Position (G29)
3) Setting a Workpiece Coordinate System (G92)
4) Selecting a Machine Coordinat System (G53)
5) Selecting a Workpiece Coordinate System (G54~G59)
6) Origin of a Workpiece Coordinate System (G51, G50)
7) Absolute and Incremental Programming (G90, G91)
8) Diameter and Radius Programming (G36, G37)
9) Inch/Metric Conversion (G20, G21)
10) Changing Coordinate and Tool Offset (G10)

5.1 Reference Position Return (G28)

Programming

G28 X(U)_ Z(W)_

Explanation of the parameters

X, Z Coordinate values of the intermediate point in absolute command

U,W Coordinate values of the intermediate point with reference to the starting point in incremental command

Function

The tool is moved to the intermediate point rapidly, and then returned to the reference point.

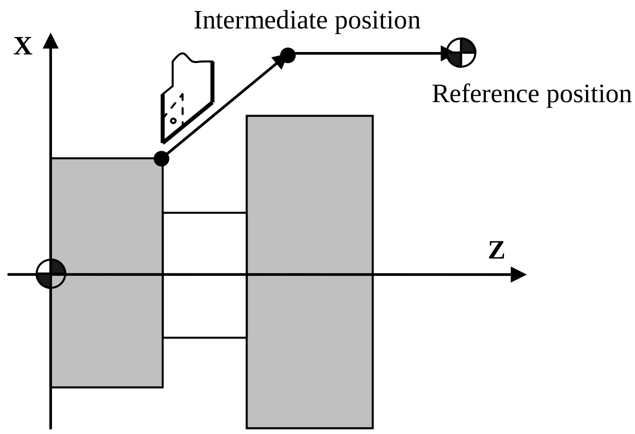


Figure 5.1 Reference Position Return

Note:

1) In general, G28 is used to change tools or cancel the mechanical error. Tool radius compensation and tool length compensation should be cancelled when G28 is executed.
2) G28 can not only make the tool move to the reference point, but also can save the intermediate position to be used in G29.
3) When the power is on and manual reference position return is not available, G28 is same as the maunaul reference position return. The direction of this reference position return (G28) is set by the axis parameter – reference approach direction.
4) G28 is one-shot G code.

5.2 Auto Return from Reference Position (G29)

Programming

G29 X(U)_ Z(W)_

Explanation of the parameters

X, Z Coordinate value of the end point in absolute command

U, W Coordinate value of the end point in incremental command

Function

The tool is moved rapidly from the intermediate point defined in G28 to the end point. Thus, G29 is generally used after G28 is defined.

Note:

G29 is one-shot G code.

Example

Use G28, G29 command to program the track shown in. It moves from the starting point A to the intermediate point B, and then returns to the reference point R. At last, it moves from the reference point R to the end point C through the intermediate point B.

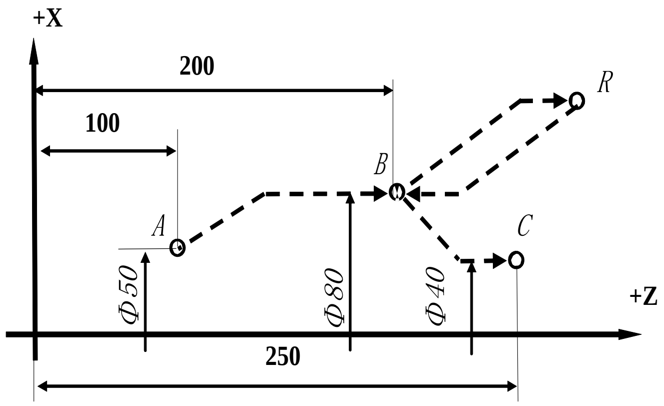


Figure 5.2 Reference Position – Example

%3317

N1 T0101

N2 G00 X50 Z100

N3 G28 X80 Z200

N4 G29 X40 Z250

N5 G00 X50Z100

N6 M30

5.3 Setting a Workpiece Coordinate System (G92)

Programming

G92 X_ Z_

Explanation of the parameters

X, Z   Coordinate values of the tool position in the workpiece coordinate system.

Functions

G92 can set a workpiece coordinate system based on the current tool position (X_ Z_).

Example

Use G92 to set a workpiece coordinate system.

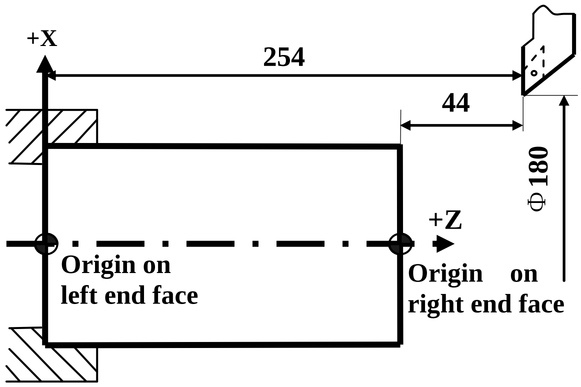


Figure 5.3 Setting a Coordinate System – Example

If the origin is set on the left end face,

G92 X180 Z254

If the origin is set on the right end face

G92 X180 Z44

5.4 Selecting a Machine Cooridinate System (G53)

Programming

G53 X_Z_

Explanation of the parameters

X, Z Absoulte coordinate values of a point in the machine coordinate system.

Function

A machine coordinate system is selected, and the tool moves to the position at the rapid traverse speed.

Note:

1) Absolute values must be specified in G53. The incremental values would be ignored by G53.
2) G53 is one-shot G code.

5.5 Selecting a Workpiece Coordinate System (G54~G59)

Programming

$$\begin{Bmatrix} G54 \\ G55 \\ G56 \\ G57 \\ G58 \\ G59 \end{Bmatrix} \text{X\_ Z\_}$$

Explanation of the parameters

X, Z Coordinate values of the point in absolute command

Function

There are six workpiece coordinate system to be selected. If one coordinate system is selected, the tool is moved to a specified point.

Note:

1) The workpiece coordinate system must be set before these commands (G54~G59) are used. The workpiece coordinate system can be set by using the MDI panel. For detailed information, please refer to the turning operation manual.
2) Reference position must be returned before these commands (G54~G59) are executed.
3) G54 is the default setting.

Example

Select one of workpiece coordinate system, and the tool path is Current point→A→B.

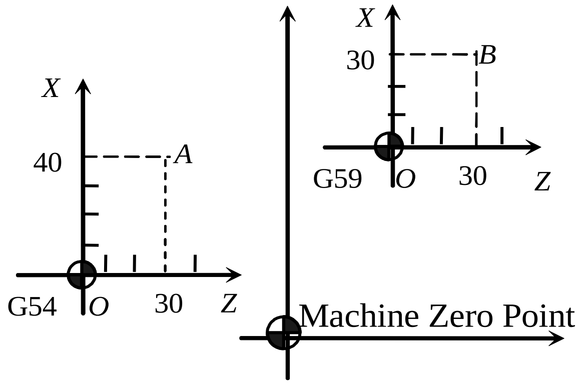


Figure 5.4 Workpiece Coordinate System – Example

%3303

N01 G54 G00 G90 X40 Z30

N02 G59

N03 G00 X30 Z30

N04 M30

5.6 Origin of a Workpiece Coordinate System (G51, G50)

Programming

G51 U_ W_

G50

Explanation of the parameters

G51 can move the origin of workpiece coordinate system.

U, W Coordinate values of the position in incremental command

G50 can cancel the movement.

Function

The origin of workpiece coordinate system can be moved.

Note:

1) G51 is only effective when T command or G54~G59 is defined in the program.
2) G50 is only effective when T command or G54~G59 is defined in the program.

Example

```
%1234
G51 U30 W10
M98 P1111 L4
G50
T0101
G01 X30 Z14
M30
```

```
%1111
T0101
G01 X32 Z25
G01 X34.444 Z99.123
M99
```


5.7 Absolute and Incremental Programming (G90, G91)

Programming

G90 X_ Z_

G91 U_W_

Explanation of the parameters

G90 Absolute programming

X, Z Coordinate values on X axis and Z axis in the coordinate system

G91 Incremental programming

U, W Coordinate values with reference to the previous position in the coordinate system

Function

The tool is moved to the specified position.

Example

Move the tool from point 1 to point 2 through point 3, and then return to the current point.

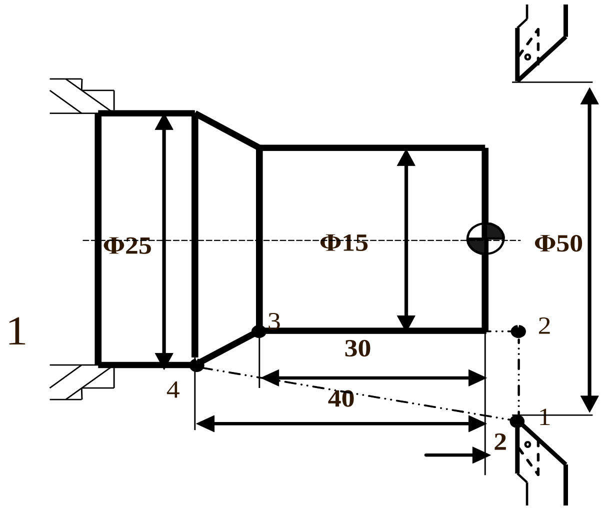


Figure 5.5 Absolute and Incremental Programming – Example

Absolute Programming	Incremental Programming	Absolute and Incremental
%0001 N 1 T0101 N 2 M03 S460 N3 G90 G00 X50 Z2 N4 G01 X15 N 5 Z-30 N 6 X25 Z-40 N 7 X50 Z2 N 8 M30	%0001 N 1 M03 S460 N 2 G91 G01 X-35 N 3 Z-32 N 4 X10 Z-10 N 5 X25 Z42 N 6 M30	%0001 N 1 T0101 N 2 M03 S460 N 3 G00 X50 Z2 N 4 G01 X15 N 5 Z-30 N 6 U10 Z-40 N 7 X50 W42 N 8 M30

5.8 Diameter and Radius Programming (G36, G37)

Programming

G36

G37

Explanation of the parameters

G36 Diameter programming

G37 Radius programming

Function

The coordinate value on X axis is specified in two ways: diameter or radius. It allows to program the dimension straight from the drawing without conversion.

Note:

1) In all the examples of this book, we always use diameter programming if the radius programming is not specified.
2) If the machine parameter is set to diameter programming, then diameter programming is the default setting. However, G36 and G37 can be used to exchange. The system shows the diameter value.
3) If the system parameter is set to radius programming, then radius programming is the default setting. However, G36 and G37 can be used to exchange. The system shows the radius value.

Example

Use Diameter programming and Radius programming for the same path

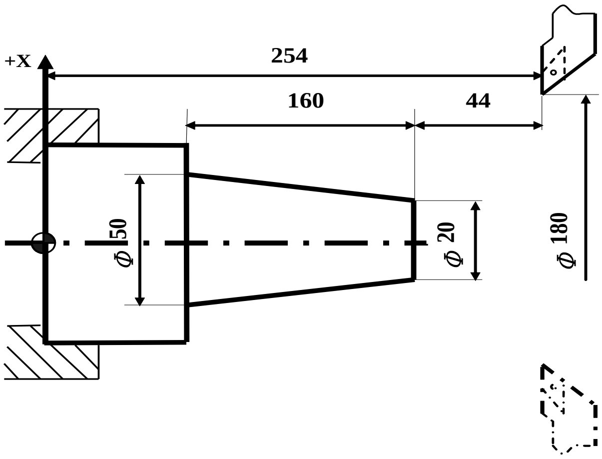


Figure 5.6 Diameter and Radius Programming – Example

Diameter Programming	Radius Programming	Compound Programming
%3304 N1 G92 X180 Z254 N2 M03 S460 N3 G01 X20 W-44 N4 U30 Z50 N5 G00 X180 Z254 N6 M30	%3314 N1 G37 M03 S460 N2 G54 G00 X90 Z254 N3 G01 X10 W-44 N4 U15 Z50 N5 G00 X90 Z254 N6 M30	%3314 N1 T0101 N2 M03 S460 N3 G37G00 X90 Z254 N4 G01 X10 W-44 N5 G36 U30 Z50 N6 G00 X180 Z254 N7 M30

5.9 Inch/Metric Conversion (G20, G21)

Programming

G20

G21

Explanation of the parameters

G20: Inch input

G21: Metric input

The units of linear axis and circular axis are shown in the following table

Table 5-1. Unit of Linear axis and Circular axis

	Linear axis	Circular axis
Inch system (G20)	Inch	Degree
Metric system (G21)	Mm	Degree

Function

Depending on the part drawing, the workpiece geometries can be programmed in metric measures or inches.

5.10 Changing Coordinate and Tool Offset (Programmable Data Input) (G10)

Programming

G10P__X__Z__I__K__R__Q__

G10P__X__Y__Z__

Explanation of the parameters

1. Command type is set by P.

   P53: modify the machine coordinate system

   P54~P59: modify G54~G59, for example, P54 is to modify G54

   P92: modify the current workpiece coordinate system

   P101~P132: the modified tool number, for example, P101 corresponds to T01.

2. Modifying coordinates (P53, P54~P59, P92)

   X, Y, Z: the value and origin of coordinates. P53 is used to modify the current position of machine coordinate system. P54~P59 and P92 are used to modify the origin of the coordinates.

   When G90 is used, the value and origin of coordinates are directly assigned to the specified coordinates.

   When G91 is used, the coordinate value and origin are assigned to the specified coordinates in an incremental way.

3. Modifying tool offset

   X: the tool offset in the X-direction

   Z: the tool offset in the Z-direction

   U: the tool offset in the X-direction (incremental way)

   W: the tool offset in the X-direction (incremental way)

   I: the tool wear-out in the X-direction

   K: the tool wear-out in the Z-direction

   R: the tool radius. It is used to set the current tool radius

   Q: the direction of tool tip. The range is 0~8. The other values are invalid.

   According to G90/G91, the data of parameters X, Z, I, K, R could be absolute or incremental.

   When G90 is used, the data of parameters X, Z, I, K, R is directly set to the tool parameters.

When G91 is used, the data of parameters X, Z, I, K, R is set to the tool parameters in an incremental way.

For example:

G91 G10 P101 X40 Z10

G90 G10 P101 X40 G91 Z10

Function

It is used to modify the coordinate system, the tool offset and compensation.

6 Spindle Speed Function

Spindle function controls the spindle speed (S), the unit of spindle speed is r/min. Spindle speed is the cutting speed when it is at the constant speed, the unit of speed is m/min.

S is modal G code command; it is only available when the spindle is adjustable. Spindle speed programmed by S code can be adjusted by overrides on the machine control panel.

This chapter would introduce

1) Limit of spindle speed (G46)
2) Constant surface cutting control (G96, G97).

6.1 Limit of Spindle Speed (G46)

Programming

G46 X_ P_

Explanation of the parameters

X The minimum speed of the spindle when using constant surface speed（r/min）

P The maximum speed of the spindle when using constant surface speed（r/min）

Function

G46 command can set the minimum of spindle speed, and the maximum of spindle speed.

Note:

It can only used with G96 (constant surface speed control command).

6.2 Constant Surface Speed Control (G96, G97)

Programming

G96 S

G97 S

Explanation of the parameters

G96	activate the constant surface speed
S	surface speed (m/min)

G97	deactivate the constant surface speed
S	spindle speed (r/min)

Function

G96 and G97 commands are to control the constant surface speed.

Note:

1) The spindle speed must be controlled automatically when the constant surface cutting command is executed.
2) The maximum of spindle speed can be set by the axis parameter.

Example

Use the constant surface control command

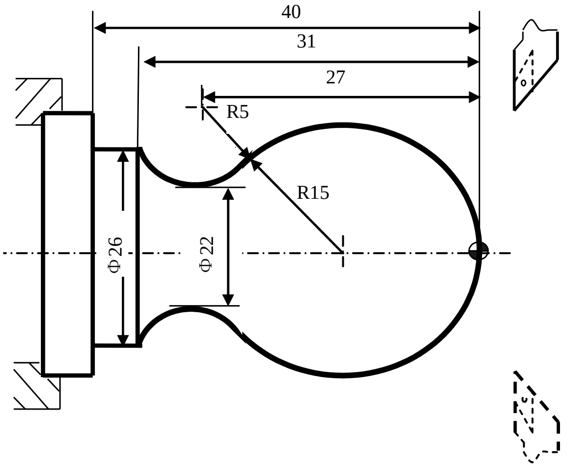


Figure 6.1 Constant Surface Control – Example

%3318

N1 T0101

N2 G00 X40 Z5

N3 M03 S460

N4 G96 S80

N5 G46 X400 P900

N5 G00 X0

N6 G01 Z0 F60

N7 G03 U24 W-24 R15

N8 G02 X26 Z-31 R5

N9 G01 Z-40

N10 X40 Z5

N11 G97 S300

N12 M30

7 Tool Compensation Function

There are two types of tool compensation. One is geometry compensation and the other is radius compensation. The tool geometry compensation is categorized as tool offset compensation and tool wear compensation. The tool offset compensation is categorized as absolute tool offset compensation and relative tool offset compensation.

Statement: T code is used in the tool geometry compensation (the sum of offset and wear compensation). G40, G41, and G42 are set for tool radius compensation.

This chapter would introduce:

1) Tool offset and Tool wear-out compensation (T code)
2) Tool radius compensation (G40, G41, G42)

7.1 Tool Offset and Tool Wear Compensation

The trajectory of the turning machine programming is the tool nose movement trajectory. But actually, the geometry size and installation position of different cutting tool is varied, the cutter point relative to the tool center position is also different. Therefore, it needs to measure the tool nose position in order to compensate for the tool offset during the machining process. So there is no need to consider tool shape and install position causing the position consistency of tool tip to simplify programming. There are two types of tool offset compensation.

7.1.1 Tool Offset

1. Absolute compensation mode

As it is shown in Figure7.1, the absolute offset means the workpiece origin relative to the directed distance of the tool nose position on the cutter frame, when the machine returns to workpiece origin. When executing tool offset compensation, tools use this value to set the coordinates. Therefore, although the cutter frame is on the machine origin, the distance of tool position relative to workpiece origin is different caused by the different sizes of the cutters. These set coordinates coincide with the workpiece coordinates (programming).

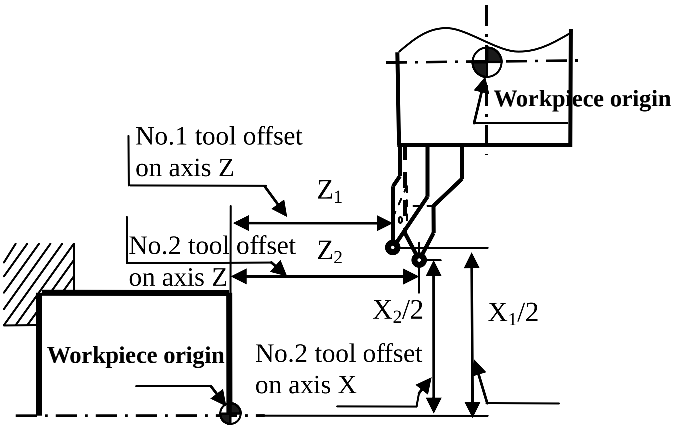


Figure 7.1 Absolute tool offset compensation

As it is shown in Figure7.2, when the machine is reached the machine origin, the value of machine coordinate system is zero, and the point on the cutter frame is regarded as the ideal point. Thus, when the tool is aligned, it is considered that the machine origin is on the cutter position. The system can automatically calculate the distance of workpiece origin relative to the tool position by the input trial diameter and length. The procedure is as followed:

1) Press “Tool Offset” function key;
2) Input the workpiece coordinates value of the tool on axis Z in the trial face cutting of the workpiece. Input 0 if the workpiece origin is set at the front face of workpiece (no movement on axis Z before setting zero). The system would automatically calculate the distance of the workpiece origin relative to the tool position on axis Z.
3) Input the workpiece coordinates value of the tool on axis X in the trial cylindrical surface cutting of the workpiece. It is the diameter of workpiece after trial cutting (no movement on axis X before setting zero). The system would automatically calculate the distance of the workpiece origin relative to the tool position on axis X.
4) Change a tool and use another tool to repeat the above steps 2~3. The absolute tool offset of this tool can be get, and automatically input to the table of tool offset.

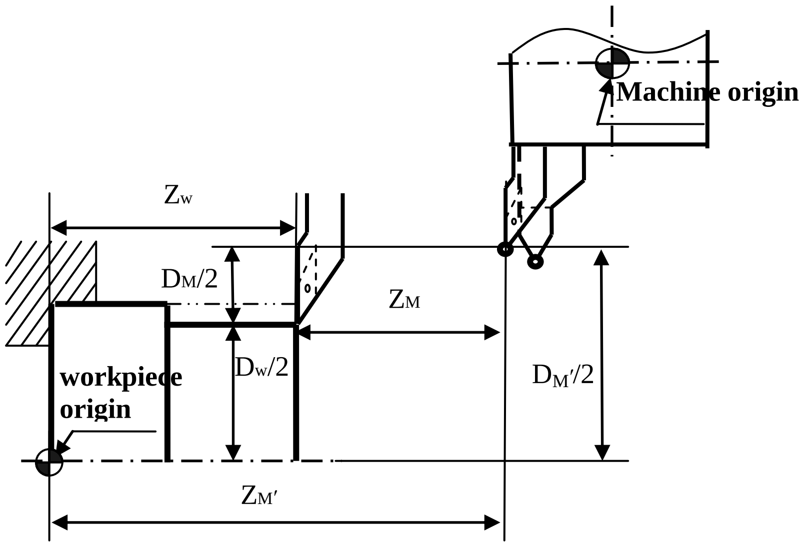


Figure 7.2 Setting the absolute tool offset compensation

2. Relative compensation mode

As it is shown in Figure7.3, one tool is set as a standard tool when aligning tool, and the coordinates is set based on the position A of this tool tip. When the other tools are at the machining position, the position B of tool tip relative to position A would arise the offset, and the original coordinates would not be applicable. Thus, the offset △x and △z are used, and the tool tip is moved from position B to A. The compensation is implemented by controlling the movement of machine carriage in this system.

Standard offset is the directed distance of the workpiece origin relative to the standard tool position on the cutter frame.

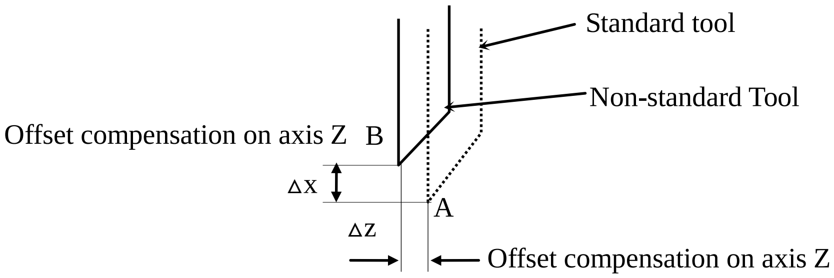


Figure 7.3 Incremental tool offset compensation

If there is tool setting gauge, the following steps are for the measurement of relative tool offset:

1) Move the standard tool to the cross center of tool setting gauge.
2) Press MDI function key, and set the current position of tool as the relative origin.
3) Change a tool and move the other tool to the cross center of tool setting gauge. The shown value is the offset relative to the standard tool.

If there is no tool setting gauge, the following steps are for the measurement of relative tool offset:

1) Press “MDI” function key, and set the current position of axis Z as the relative origin in the trial face cutting of the workpiece. (no movement on axis Z before setting zero).
2) Press “MDI” function key, and set the current position of axis X as the relative origin in the trial cylindrical surface cutting of the workpiece (no movement on axis X before setting zero). The standard tool has set a reference point on the workpiece. When the standard tool is at the reference point, it is the position of the relative origin.
3) Change a tool and move the another tool to the reference point of workpiece. The shown value is the offset relative to the standard tool.

The system would automatically calculate the distance of the workpiece origin relative to the tool position, and compare with the standard tool’s value to get the tool offset relative to the standard tool’s, when the cutter frame is at the machine origin. The procedures are as followed:

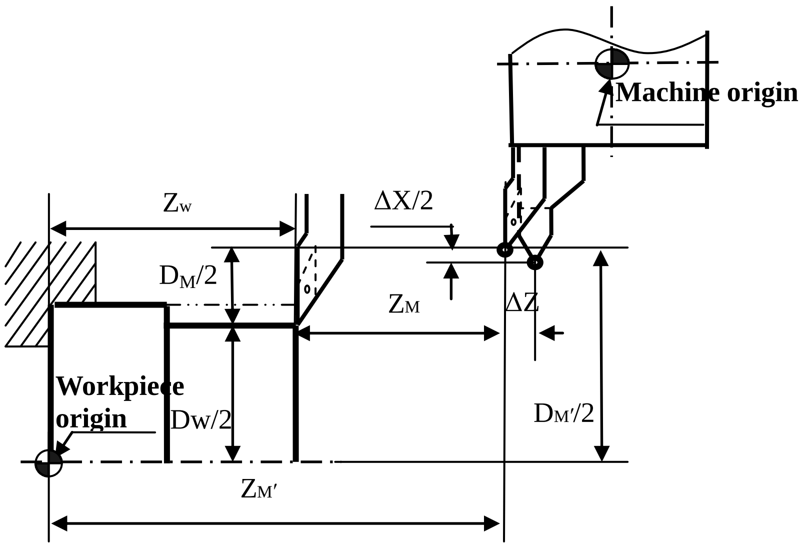


Figure 7.4 Setting the incremental tool offset compensation

1) Press “MDI->Tool Offset” function key;
2) Use the tool to cut the front face of workpiece, and input the coordinates value of workpiece coordinates on axis Z, i.e. the length of workpiece. If the workpiece origin is set at the front face of workpiece, input “0” (no movement on axis Z before setting zero). The system would automatically calculate the distance of workpiece origin relative to the tool position of standard tool, i.e. the standard tool offset on axis Z.
3) Input the coordinate value of workpiece coordinates on axis X in the trial cylindrical surface cutting of the workpiece, i.e. the diameter of workpiece (no movement on axis X before setting zero). The system would automatically calculate the distance of workpiece origin relative to the tool position of standard tool, i.e. the standard tool offset on axis Z.
4) Press “Tool Offset->Standard Tool” to set the standard tool offset as the reference.
5) Change a tool and repeat the steps 2~3 to get the tool offset relative to the standard’s and automatically input to the table of tool offset.

7.1.2 Tool Wear-out

There would be size error after the tool is used for a long time. Thus, the compensation is required. This compensation and the tool offset compensation are saved in the same address of register. The wear-out compensation is only valid for the corresponding tool’s (including the standard tool).

Programming

T XX XX

Explanation of the parameters

XX Tool number (two digits). The number of tool depends on manufacture's configuration.

XX Tool offset number (two digits). It corresponds to the specific compensation value. "00" means the compensation is 0, i.e. compensation cancelled. The tool compensation function validated or cancelled is implemented by controlling cutter carriage. The tool offset number is the address number of register of tool offset compensation. This register saves the data of offset compensation and wear-out compensation of X-axis and Z-axis.

The offset number and tool number can be same or not, i.e. one tool can corresponds to several tool offset number. As it is shown in Figure7.5, if there is compensation on axis X and Z at the tool path relative to the programming path (compensation vector is composed of the compensation on axis X and Z), the end point of tool path is the position of end point of program plus or minus the compensation (compensation vector) assigned by T code.

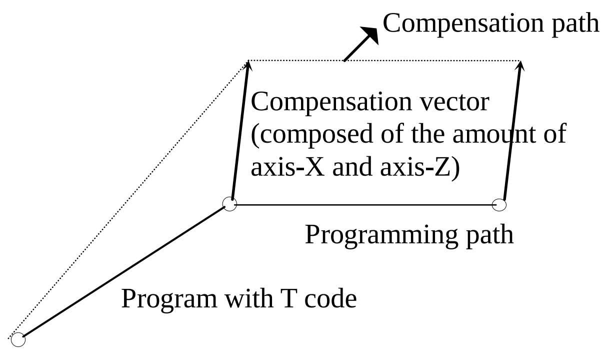
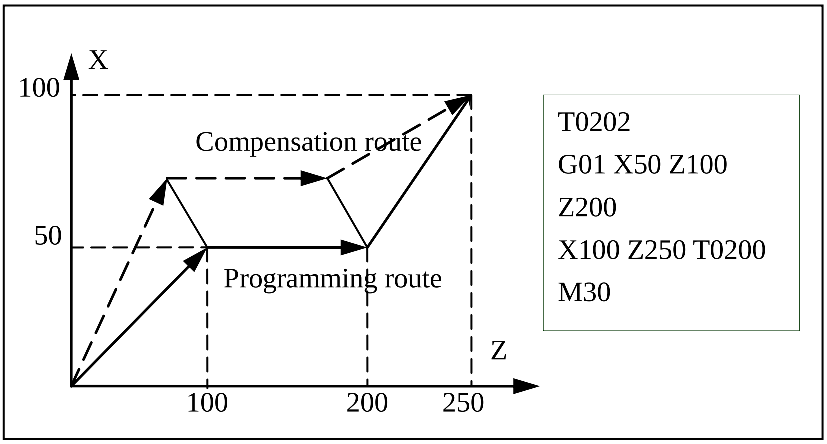


Figure 7.5 Tool wear-out compensation

Example

As it is shown in the following figure, set the tool wear-out compensation, and then cancel it.


X
100
50
Compensation route
Programming route
Z
100
200
250
T0202
G01 X50 Z100
Z200
X100 Z250 T0200
M30

7.2 Tool Radius Compensation (G40, G41, G42)

Programming

$$\begin{Bmatrix} G40 \\ G41 \\ G42 \end{Bmatrix} \begin{Bmatrix} G00 \\ G01 \end{Bmatrix} \text{X\_ Z\_}$$

Explanation of the parameters

Tool nose radius compensation is specifies by G41, G42 and G40 or tool nose radius compensation number specified by T code to add or cancel radius compensation.

G40 Tool nose radius compensation cancels

G41 Left cutter compensation (on the left of tool move direction) (Figure 7.6).

G42 Right cutter compensation (on the right of tool move direction) (Figure 7.6).

X, Z Coordinate values of the end point. It is the point where the tool radius compensation is activated or deactivated.

Note: G40, G41 and G42 are modal G-code.

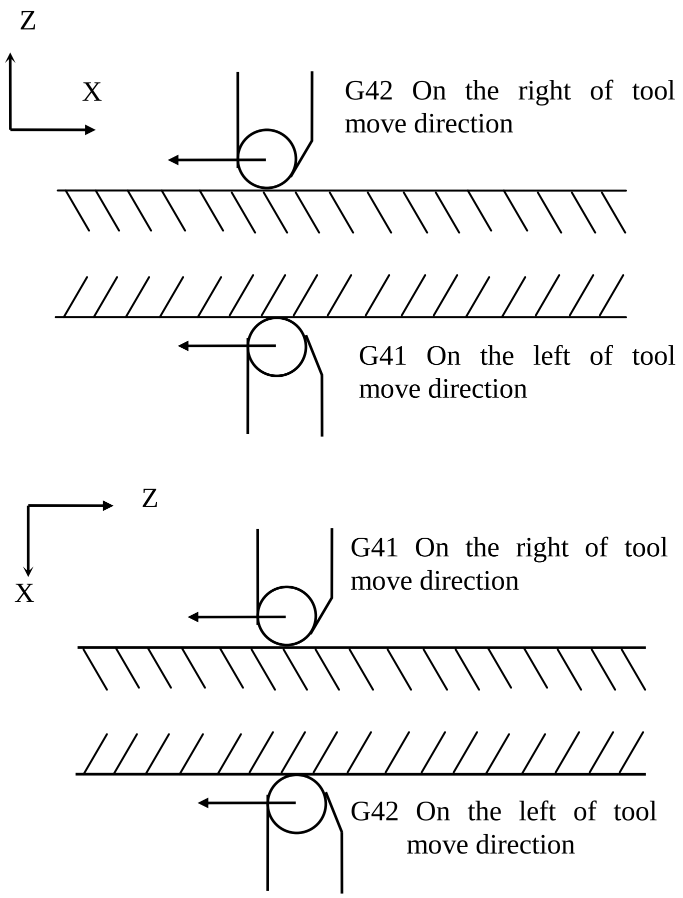


Figure 7.6 Tool Radius Compensation

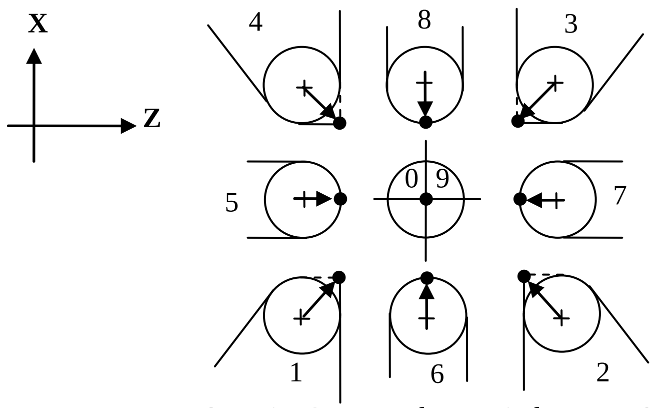
Function

NC program is generally aimed at cutting tool on a point at which the cutter location, according to the contour of the workpiece size preparation. Turning tool cutter locus ideal state of is the imagined cutter-tip A or arc center point O. Actually, tool nose is an arc rather than a point. When the tool moves along an arc, it causes an error which can be cancelled by tool nose radius compensation.

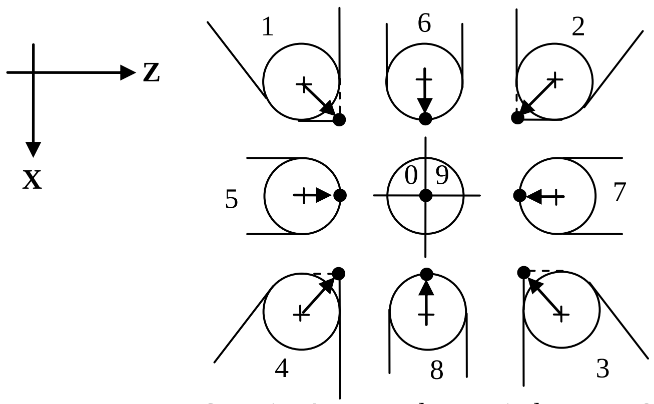
Note:

1) When G41/G42 without parameters, the compensation number representing the tool nose radius compensation is assigned by T code. This corresponds to the tool offset number.
2) G40, G41, and G42 must be used with G00 or G01. G02 or G03 cannot be used.

In the register of tool radius compensation, it defines the direction of tool nose and tool radius. The direction number of turning tool nose gives an identity of the position relationship between CL point and tool nose circle center. There are 10 directions from 0 to 9.



● CL point A， + Tool nose circle center O



● CL point A， + Tool nose circle center O

Figure 7.7 Position code of Tool Nose

Example

Use the tool radius compensation, and program for the part shown in Figure 7.2

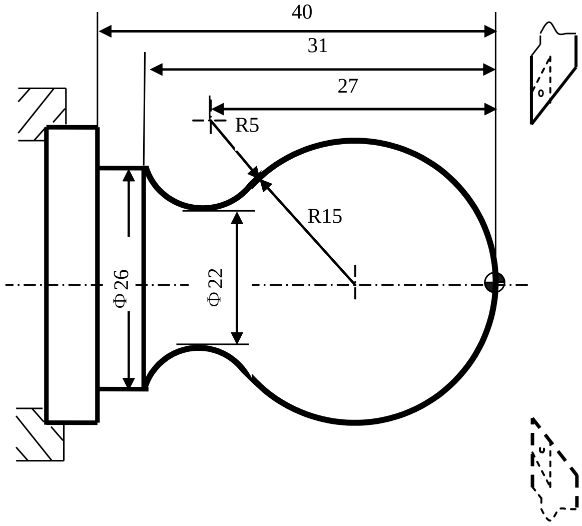


Figure 7.8 Tool Radius Compensation

%3323

N1 T0101 (Change NO.1knife to define its coordinate)

N2 M03 S400 (Spindle:400r/min CW)

N3 G00 X40 Z5 (To the position of program start)

N4 G00 X0 (The tool moves to workpiece center)

N5 G01 G42 Z0 F60 (Add tool radius compensation, and be close to workpiece)

N6 G03 U24 W-24 R15 (Machine R15 circular block)

N7 G02 X26 Z-31 R5 (Machine R5 circular block)

N8 G01 Z-40 (MachineΦ26 excircle)

N9 G00 X30 (Withdraw from the machined surface)

N10 G40 X40 Z5 (Cancel radius compensation, and return to the program start)

N11 M30 (Spindle stop, the main program end and reset)

8 Miscellaneous Function

As it is mentioned in Chapter 1.8, there are two ways of execution when a move command and M code are specified in the same block.

1) Pre-M function

   M command is executed before the completion of move command.

2) Post-M function

   M command is executed after the completion of move command

There are two types of M code: one-shot M code, and modal M code.

Table 8-1 Type of M code

Type	**Meaning**
One-shot M code	The M code is only effective in the block in which it is specified
Modal M code	The M code is effective until another M code is specified.

8.1 M code List

The following is a list of M command.

Table 8-2 M code List

CNC M-function	Type of Mode	Function	Pre/Post-M function
M00	One-shot	Program stop	Post-M function
M01	One-shot	Optional stop	Post-M function
M02	One-shot	End of program	Post-M function
M30	One-shot	End of program with return to the beginning of program	Post-M function
M98	One-shot	Calling of subprogram	Post-M function
M99	One-shot	End of subprogram	Post-M function
PLC M-function	**Type of Mode**	**Function**	**Pre/Post-M function**
M03	Modal	Spindle forward rotation	Pre-M function
M04	Modal	Spindle reverse rotation	Pre-M function
M05	Modal	◣Spindle stop	Post-M function
M07	Modal	Number1 Coolant on	Pre-M function
M08	Modal	Number2 Coolant on	Pre-M function
M09	Modal	◣Coolant off	Post-M function

◣: default setting

8.2 CNC M-Function

8.2.1 Program Stop (M00)

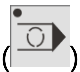
M00 is one-shot M function, and it is post-M function.

The program can be stopped, so that the operator could measure the tool and the part, adjust part and change speed manually, and so on.

When the program is stopped, the spindle is stopped and the coolant is off. All of the current modal information remains unchanged. Resuming program could be executed by pushing “Cycle Run” button on the machine control panel.

8.2.2 Optional Stop (M01)

M01 is one-shot M function, and it is post-M function.

Similarly to M00, M01 can also stop the program. All of the modal information is maintained. The difference between M00 and M01 is that the operator must press M01 button () on the machine control panel. Otherwise, the program would not be stopped even if there is M01 code in the program.

8.2.3 End of Program (M02)

M02 is one-shot M function, and it is post-M function.

When M02 is executed, spindle, feed and coolant are all stopped. It is usually at the end of the last program block. To restart the program, press “Cycle Run” button on the operational panel.

8.2.4 End of Program with return to the beginning of program (M30)

M30 is one-shot M function, and it is post-M function.

Similarly to M02, M30 can also stop the program. The difference is that M30 returns control to the beginning of program. To restart the program, press “Cycle Run” button on the operational panel.

8.2.5 Counting (M64)

The system can calculate the number of machining workpiece at the end of program with M30. It can also calculate the accumulation of machining workpiece by executing M64. It is required to set the parameter->machine parameters->the judgment of counting the workpiece (0: M30, 1:M64).

8.2.6 User-defined Input and Output (M90, M91)

CNC system provides M90 (user-defined input) and system variable #1190 to control the execution of G-code, according to PLC execution. It also provides M91 (user-defined output) and system variable #1190 to control PLC execution by the G-code execution. Those two commands are related to PLC running condition, and must be used with PLC.

Example 1: When PLC input signal X0.4 is valid (high level), one part of program is executed. Otherwise, the other part of program is executed.

The code should be added in the function PLC1 of PLC program:

```
If(bit(X[0],4))
    *ch_user_in(0)=1;    //it can be set to any values if necessary, i.e. #1190=1
else
    *ch_user_in(0)=0;    //#1190=0
```

The example of G-code is as followed:

```
…
…
M90     //user-defined input, system would get the value of #1190 according to PLC execution
If  #1190  EQ  1    //if PLC input signal X0.4 is valid, this part of program is run
    …
    …
else        //if PLC input signal X0.4 is not valid, this part of program is run
    …
    …
endif
```

Example 2: If the first part of G-code is executed, PLC output signal Y0.4 is valid (high level). If the second part of G-code is executed, PLC output signal Y0.4 is not valid (low level).

The example of G-code is as followed:


```
If
    …
    …
    #1191=1    //the first part of program, it can be set to any values if necessary
else
    …
    …
    #1191=0    //the second part of program, it can be set to any values
endif
M91    //user-defined output, the value of #1191 is assigned to *ch_user_out(0)
```

The code should be added in the function PLC1 of PLC program:

```
If(*ch_user_out(0)==1)     //if the first part of program is run
    Y[0]|=0x10;            //Y0.4=1, output signal Y0.4 is valid (high level)
else
    Y[0]&=~0x10;           //if the second part of program is run, Y0.4=0
```

8.2.7 Saving Macro (M94)

M94 is to save macro (#1300~#1399) to the files.

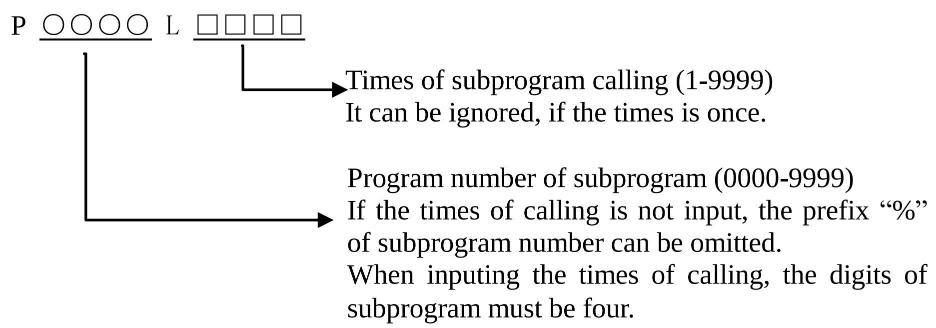
8.2.8 Subprogram Control (M98, M99)

- Calling a Subprogram (M98)

  M98 P_ L_

  P   program number of the subprogram

  L   repeated times of subprogram



In Auto mode, CNC would call the subprogram specified by the parameter P after the rest of code of program is run, when M98 is executed. The maximum times of subprogram calling is 9999. M98 is not valid in MDI mode.

Note: Calling subprogram can be used with parameters. Blank space is not allowed at the beginning of the subprogram.

- End of Subprogram (M99)

M99 indicates the end of subprogram and returns control to the main program. It is not valid in MDI mode.

Example

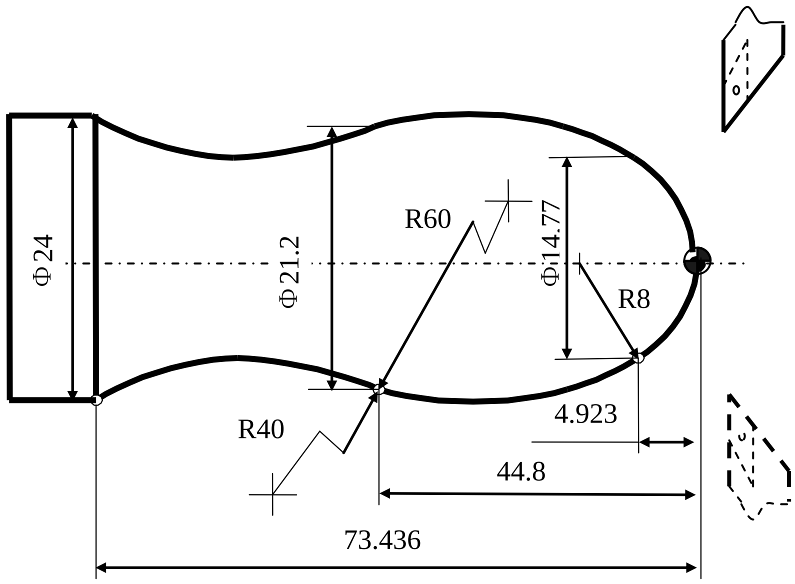


Figure 8.1 Subprogram Control - Example

%3111	(main program name)
N1 T0101	(Tool No.1)
N2 G92 X32 Z1	(coordinates setting, the position of aligning tool)
N3 G00 Z0 M03 S46	(move to the start of subprogram, spindle CW)
N4 M98 P0003 L5	(subprogram call, repeated times: 5)
N5 G36 G00 X32 Z1	(return to the position of aligning tool)
N6 M05	(spindle stop)
N7 M30	(main program ends and resets)
%0003	(subprogram name)
N1 G37 G01 U-12 F100	(radius programming, move to the start of cutting)
N2 G03 U7.385 W-4.923 R8	(R8 arc)
N3 U3.215 W-39.877 R60	(R60 arc)
N4 G02 U1.4 W-28.636 R40	(R40 arc)
N5 G00 U4	(leave the cutting surface)
N6 W73.436	(return to the beginning of cycle of axis-Z)
N7 G01 U-5 F100	(set the cutting amount of cycles)
N8 M99	(subprogram ends, return to the main program)

8.3 PLC M Function

8.3.1 Spindle Control (M03, M04, M05)

M03 starts spindle to rotate CW at the set speed set in the program.

M04 starts spindle to rotate CCW at the set speed in the program.

M05 stops spindle.

M03, M04 are modal M code, and they are pre-M function. M05 is modal M code, and it is post-M function. M05 is the default setting.

8.3.2 Coolant Control (M07, M08, M09)

M07, M08 can turn on the coolant.

M09 can turn off the coolant.

M07 and M08 are modal M code, and they are pre-M function. M09 is one-shot M code, and it is post-M function. Moreover, M09 is the default setting.

9 Functions to Simplify Programming

This chapter would introduce:

1) Canned Cycle

   Internal diameter/ Outer diameter cutting cycle (G80)

   End face turning cycle (G81)

   Thread cutting cycle (G82)

   End face peck drilling cycle (G74)

   Outer diameter grooving cycle (G75)

2) Multiple Repetitive Cycle

   Stock Removal in Turning (G71)

   Stock Removal in Facing (G72)

   Pattern Repeating (G73)

   Multiple Thread Cutting Cycle (G76)

9.1 Canned Cycles

To simplify programming, the canned cycle command can execute the specific operation using one G code, instead of several separated G commands in the program.

9.1.1 Internal Diameter/Outer Diameter Cutting Cycle (G80)

- **Straight Cutting Cycle**

Programming

G80 X(U)_ Z(W)_ F_

Explanation of the parameters

X, Z	Coordinate values of end point (point C) in absolute command

U, W	Coordinate values of end point (point C) with reference to the initial point (point A) in incremental command

F	Feedrate

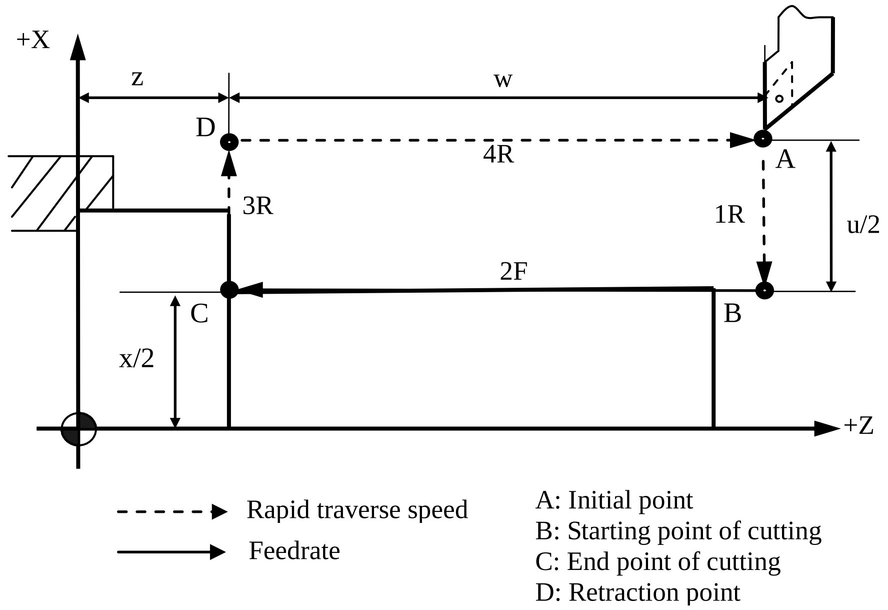


Figure 9.1 Straight cutting cycle (G80)

Function

This command can implement the straight cutting. The machining path is A→B→C→D→A.

- **Taper Cutting Cycle**

Programming

G80 X(U)_ Z(W)_ I_ F_

Explanation of the parameters

X, Z Coordinate values of end point (point C) in absolute command

U, W Coordinate values of end point (point C) with reference to the initial point (point A) in incremental command

I The radius difference between starting point B and end point C. It is negative, if the radius of point B is less than the radius of point C. Otherwise, it is positive.

F Feedrate

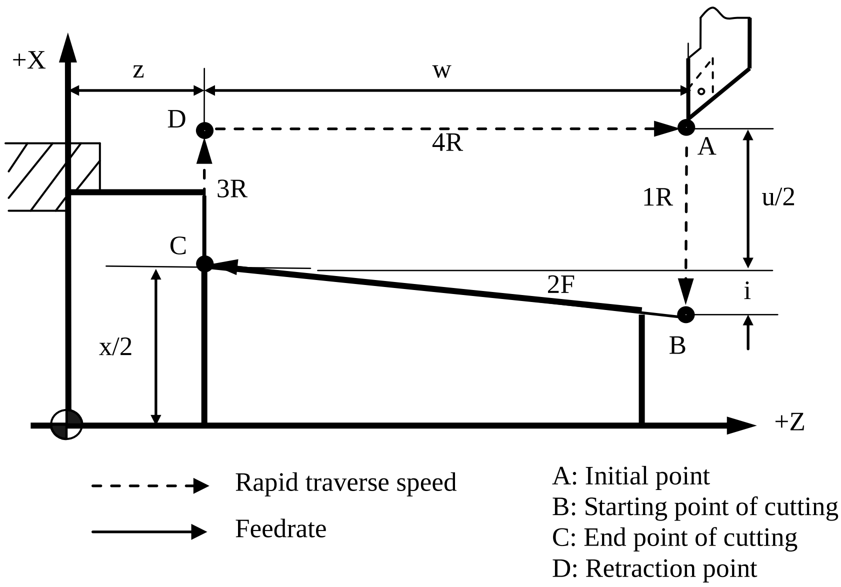


Figure 9.2 Taper Cutting Cycle (G80)

Function

This command can implement the taper cutting. The machining path is A→B→C→D→A.

Example 1

Use G80 command to machine the cylindrical part in two steps – rough machining and finish machining.

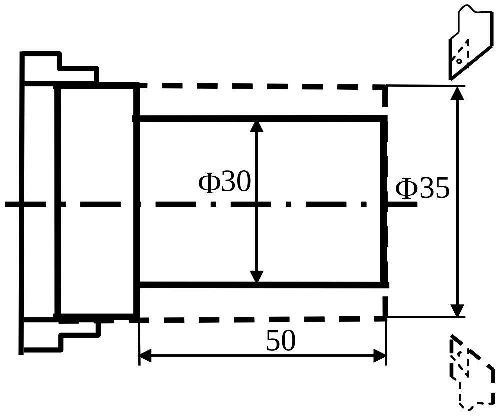


Figure 9.3 Internal Diameter/Outer Diameter Cutting Cycle – Example 1

%3320

N1 T0101

N2 M03 S460

N3 G00 X90Z20

N4 X40 Z3

N5 G80 X31 Z-50 F100

N6 G80 X30 Z-50 F80

N7 G00X90 Z20

N8 M30

Example 2

Use G80 command to machine the tapered part in two steps – rough machining and finish machining.

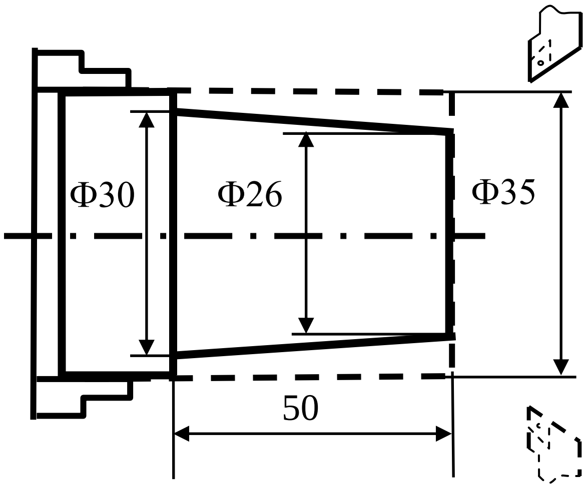


Figure 9.4 Internal Diameter/Outer Diameter Cutting Cycle – Example 2

```
%3321
N1 T0101
N2 G00 X100Z40 M03 S460
N3 G00 X40 Z5
N4 G80 X31 Z-50 I-2.2 F100
N5 G00 X100 Z40
N6 T0202
N7 G00 X40 Z5
N8 G80 X30 Z-50 I-2.2 F80
N9 G00 X100 Z40
N10 M05
N11 M30
```


Example 3

Use G80 command to machine the tapered part in two steps – rough machining and finish machining.

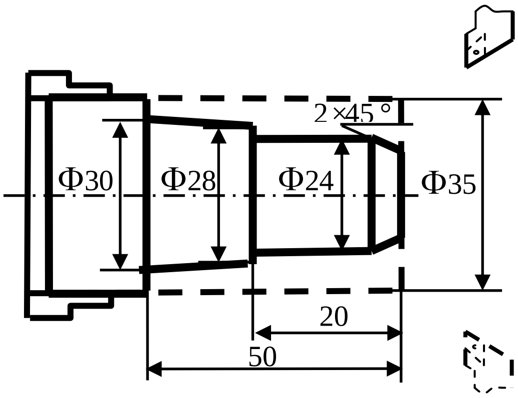


Figure 9.5 Internal Diameter/Outer Diameter Cutting Cycle – Example 3

```
%3322
N1 T0101
N2 M03 S460
N3 G00 X100 Z40
N4 X40 Z3
N5 G80 X31 Z-50 F100
N6 G80 X25 Z-20
N7 G80 X29 Z-4 I-7 F100
N8 G00 X100 Z40
N9 T0202
N10 G00 X100 Z40
N11 G00 X14 Z3
N12 G01 X24 Z-2 F80
N13 Z-20
N14 X28
N15 X30 Z-50
N16 G00 X36
N17 X80 Z10
N18 M05
N19 M30
```

9.1.2 End Face Turning Cycle (G81)

- **Face Cutting Cycle**

Programming

G81 X(U)_ Z(W)_ F_

Explanation of the parameters

X, Z Coordinate values of end point (point C) in absolute command

U, W Coordinate values of end point (point C) with reference to the initial point (point A) in incremental command

F Feedrate

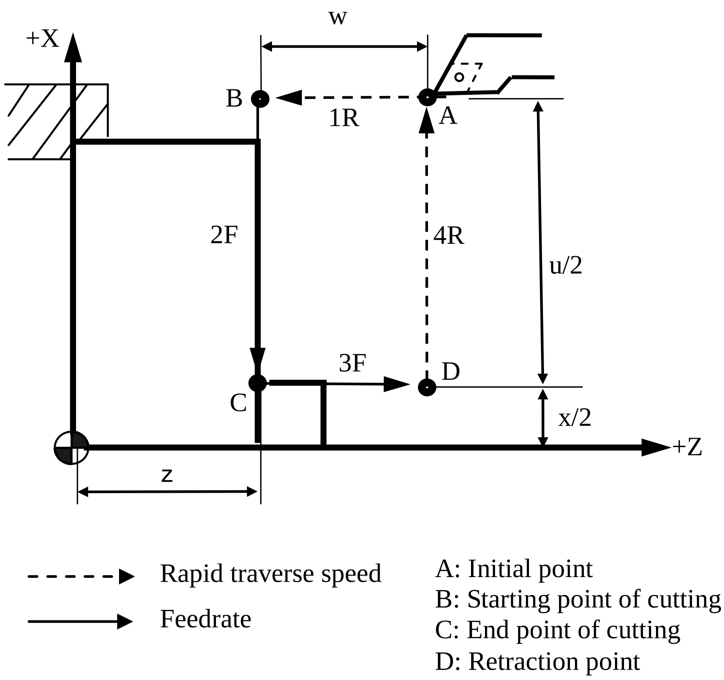


Figure 9.6 Face Cutting Cycle (G81)

Function

This command can implement the end face cutting. The machining path is A→B→C→D→A.

- **Taper Face Cutting Cycle**

Programming

G81 X(U)_ Z(W)_ K_ F_

Explanation of the parameters

X, Z Coordinate values of end point (point C) in absolute command

U, W Coordinate values of end point (point C) with reference to the initial point (point A) in incremental command

K The distance on Z axis of the starting point (point B) with reference to the end point (point C). It is negative, if the value of point C on Z axis is more than point B's. It is positive, if the value of point C on Z axis is less than point B's.

F Feedrate

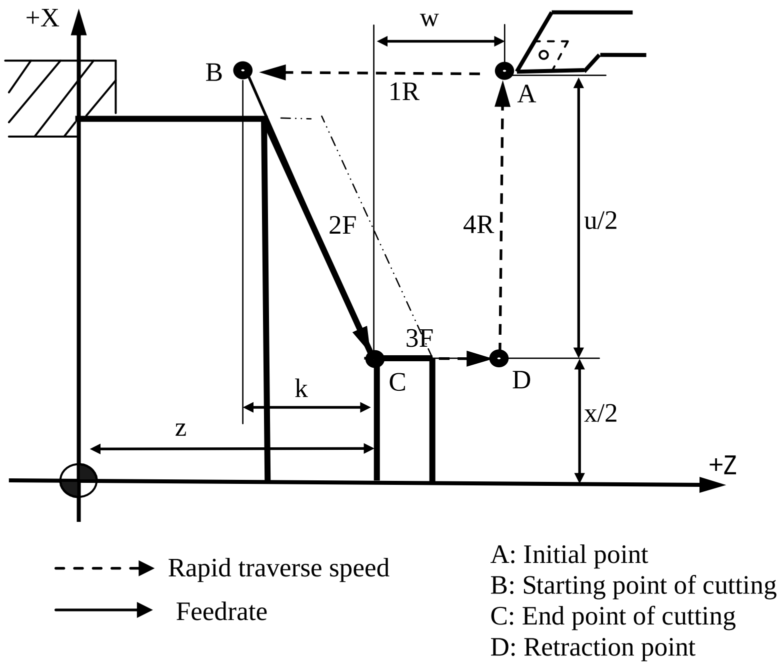


Figure 9.7 Taper Face Cutting Cycle (G81)

Function

This command can implement the taper face cutting. The machining path is A→B→C→D→A.

Example

Use G81 to program. The dashed line stands for the roughcast.

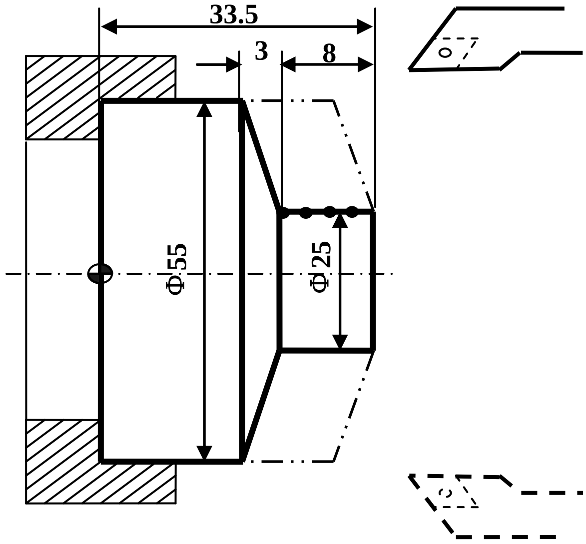


Figure 9.8 End Face Turning Cycle (G81)

%3323

N1 T0101

N2 G00 X60 Z45

N3 M03 S460

N4 G81 X25 Z31.5 K-3.5 F100

N5 X25 Z29.5 K-3.5

N6 X25 Z27.5 K-3.5

N7 X25 Z25.5 K-3.5

N8 M05

N9 M30

9.1.3 Thread Cutting Cycle (G82)

- **Cylindrical Thread Cutting Cycle**

Programming

G82 X(U)_ Z(W)_ R_ E_ C_ P_ F/J_Q_

Explanation of the parameters

X, Z Coordinate values of end point (point C) in absolute command

U, W Coordinate values of end point (point C) with reference to the initial point (point A) in incremental command.

R, E Retraction amount of thread cutting. R and E are vectors. R is the retraction of axis-Z, and E is the retraction of axis-X. R and E can be omitted, and it means that the retraction function is not required.

C The number of thread head. It is single thread when C is 0 or 1.

P Uni-tip thread cutting, it is the spindle turning corner of spindle pulse to start (defaule is 0). Muti-tip thread cutting, it is the spindle turning corner of start points.

F Thread lead per revolution

J Thread lead in inch measurement

Q

1) Acceleration constant of thread cutting retraction. When it is set to zero, the acceleartion is maximum. The more this value is, the acceleration time is longer, and the retraction is longer. Q must be set to zero or more than zero.
2) When there is no Q, the set acceleration constant on each axis is used in the retraction.
3) R and E must be set when the retraction function is required.
4) The retraction ratio of minor axis:major axis should not be more than “20“.
5) Q is one-shot G code.

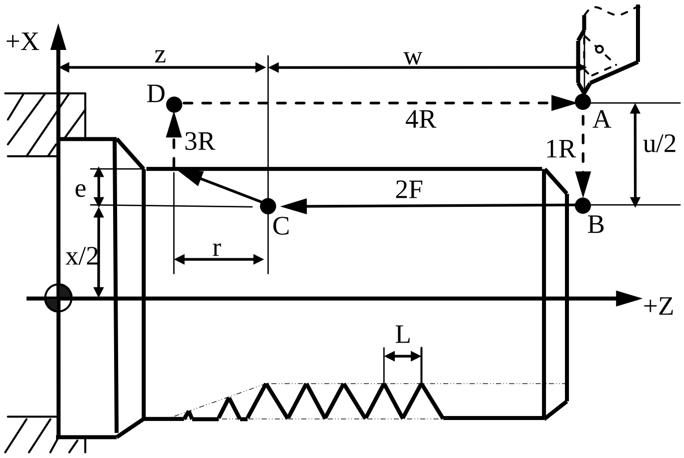


Figure 9.9 Cylindrical Thread Cutting Cycle (G82)

Function

This command can implement the cylindrical thread cutting. The machining path is A→B→C→D→E→A.

Note

This command is same as G32 (Thread cutting with constant lead). This cycle would stop after the whole action is done in the state of feed hold.

- **Taper Thread Cutting Cycle**

Programming

G82 X(U)_ Z(W)_ I_ R_ E_ C_ P_ F(J)_ Q_

Explanation of the parameters

X, Z Coordinate values of end point (point C) in absolute command

U, W Coordinate values of end point (point C) with reference to the initial point (point A) in incremental command

I The radius difference between starting point B and end point C. It is negative, if the radius of point B is less than the radius of point C. Otherwise, it is positive.

R, E Retraction amount of thread cutting. R and E are vectors. R is the retraction of axis-Z, and E is the retraction of axis-X. R and E can be omitted, and it means that the retraction function is not required.

C The number of thread head. It is single thread when C is 0 or 1.

P Uni-tip thread cutting, it is the spindle turning corner of spindle pulse to start (defaule is 0). Muti-tip thread cutting, it is the spindle turning corner of start points.

F Thread lead per revolution

J Thread lead in inch measurement

Q

1) Acceleration constant of thread cutting retraction. When it is set to zero, the acceleartion is maximum. The more this value is, the acceleration time is longer, and the retraction is longer. Q must be set to zero or more than zero.
2) When there is no Q, the set acceleration constant on each axis is used in the retraction.
3) R and E must be set when the retraction function is required.
4) The retraction ratio of minor axis:major axis should not be more than “20“.
5) Q is one-shot G code.

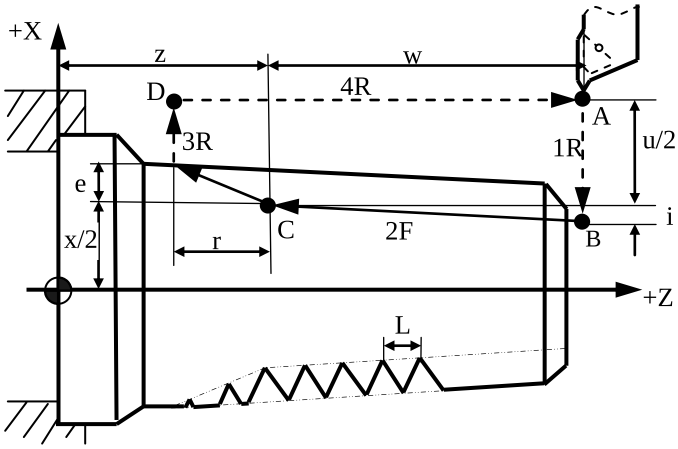


Figure 9.10 Taper Thread Cutting Cycle (G82)

Function

This command can implement the taper thread cutting. The machining path is A→B→C→D→A.

Example

Use G82 command to program. The screw’s pitch is 1.5, and the number of thread head is 2.

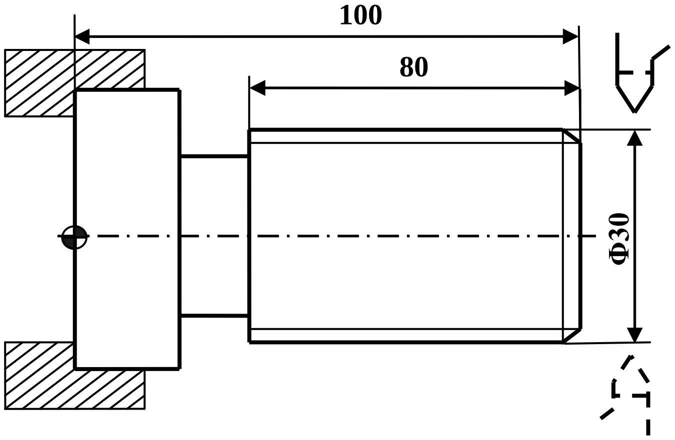


Figure 9.11 Thread Cutting Cycle - Example

%3324	
N1 G54 G00 X35 Z104	(Choose coordiante system G54, to cycle start)
N2 M03 S300	(Spindle CW 300r/min)
N3 G82 X29.2 Z18.5 C2 P180 F3	(1st thread cutting cycle,Depth 0.8mm)
N4 X28.6 Z18.5 C2 P180 F3	(2nd thread cutting cycle, Depth 0.4mm)
N5 X28.2 Z18.5 C2 P180 F3	(3rd thread cutting cycle, Depth 0.4mm)
N6 X28.04 Z18.5 C2 P180 F3	(4th thread cutting cycle, Depth 0.16mm)
N7 M30	(Spindle stop. Main system end and reset)

9.1.4 End Face Peck Drilling Cycle (G74)

Programming

G74 Z(W)_ R(e) Q(△ K) F_

Explanation of the parameters

Z Coordinate value on Z axis of the end point in absolute command

W Coordinate value on Z axis of the end point with reference to the starting point in incremental command

R Retraction amount(e) for each feed. It must be absolute value.

Q Depth of drilling(△ K) for each feed. It must be absolute value.

F Feedrate

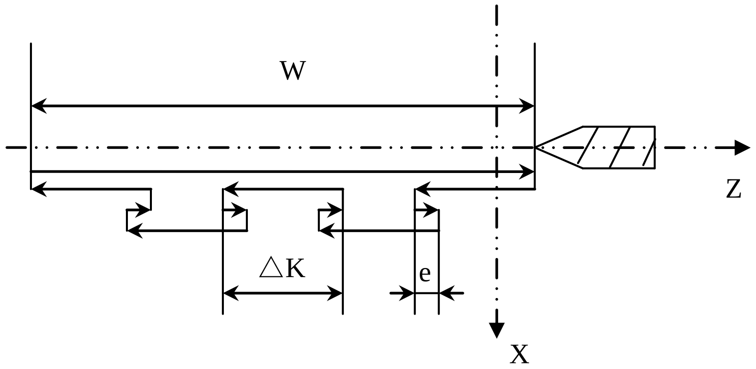


Figure 9.12 End Face Peck Drilling Cycle (G74)

Function

This command can drill a hole on end face.

Example

Use G74 to drill a hole on a workpiece.

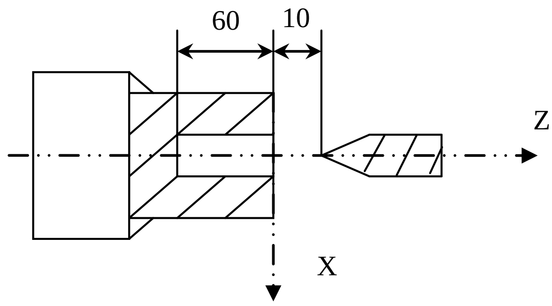


Figure 9.13 End Face Peck Drilling Cycle – Example

%1234

T0101

M03S500

G01 X0 Z10

G74 Z-60R1Q5F1000

M30

9.1.5 Outer Diameter Grooving Cycle (G75)

Programming

G75 X(U)_ R(e) Q(△ K) F_

Explanation of the parameters

X Coordinate value on X axis of the end point in absolute command

U Coordinate value on X axis of the end point with reference to the starting point in incremental command

R Retraction amount(e) for each feed. It must be absolute value.

Q Depth of grooving(△ K) for each feed. It must be absolute value.

F Feedrate

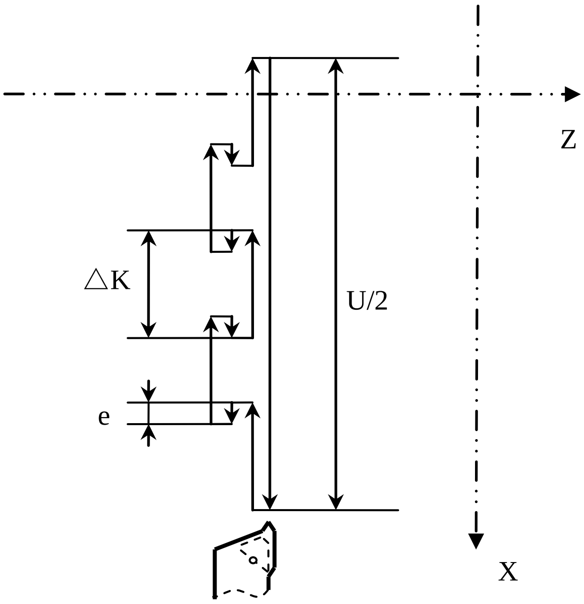


Figure 9.14 Outer Diameter Grooving Cycle (G75)

Function

This command can be used for grooving.

Example

Use G75 to groove a hole on a workpiece.

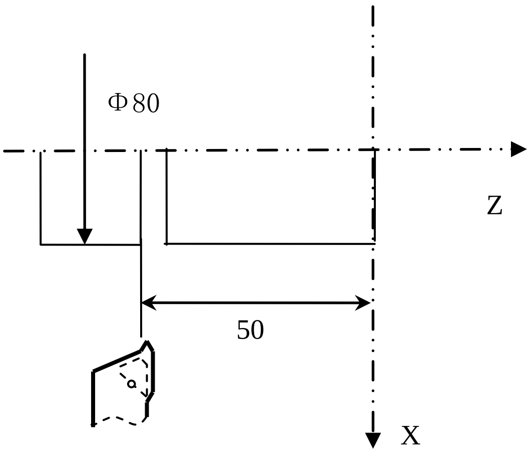


Figure 9.15 Outer Diameter Grooving Cycle - Example

```
%1234
T0101
M03S500
G01 X50 Z50
G75 X10R1Q5F1000
M30
```


9.2 Multiple Repetitive Cycle

Multiple repetitive cycle command can only use one command to finish the rough machining and the finish machining.

Here are some notes for G71, G72 and G73:

1) Blocks, which is specified by address P, have preparatory function G00 or G01 in 01 group, or it will give an alarm;
2) In MDI, multiple repetitive cycle command is forbidden;
3) In multiple repetitive cycle G71,G72,G73, between blocks, whose sequence numbers specified by P and Q, should not contain M98 subprogram call and M99 subprogram return command.

9.2.1 Stock Removal in Turning (G71)

- **Stock Removal in Turning without Groove**

Programming

G71 U(△ d) R(r) P(ns) Q(nf) X(△ x) Z(△ z) F(f) S(s) T(t)

Explanation of the parameters

U(△ d) the cutting depth (radius designation). The cutting direction depends on the direction of AA'.

R(r) Retraction amount

P(ns) Sequence number of the first block for the finishing program.

Q(nf) Sequence number of the last block for the finishing program.

X(△ x) Distance and direction of finishing allowance on X axis

Z(△ z) Distance and direction of finishing allowance on Z axis

F(f), S(s), T(t) F, S, T function are only effective for the rough machining, i.e, it is not effective in the finishing program – between P(ns) and Q(nf).

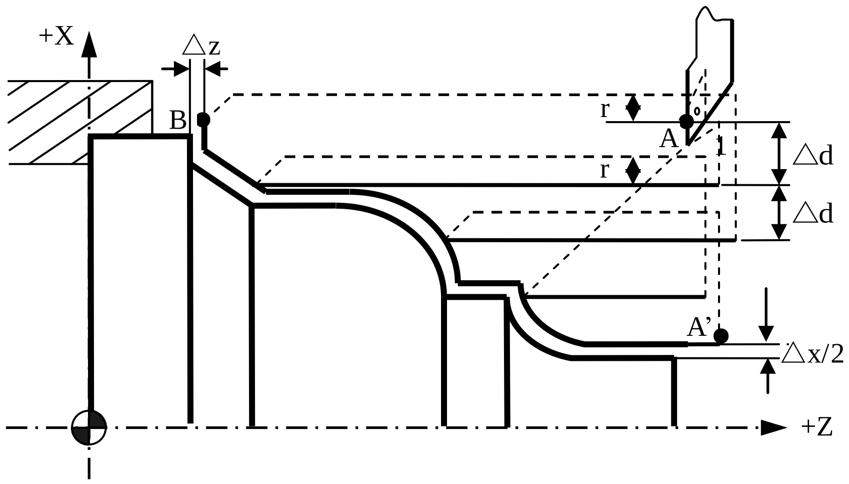


Figure 9.16 Stock Removal in Turning without Groove (G71)

Function

This command can do a stock removal in facing without groove. The machining path is A→A'→B

Note

1) G00 or G01 must be used in the finishing program – between P(ns) and Q(nf). Otherwise, there is an alarm message.
2) G71 cannot be used in MDI mode.
3) G98 and G99 cannot used in the finishing program – between P(ns) and Q(nf).
4) The direction of △ x and △ z is shown in the following figure.

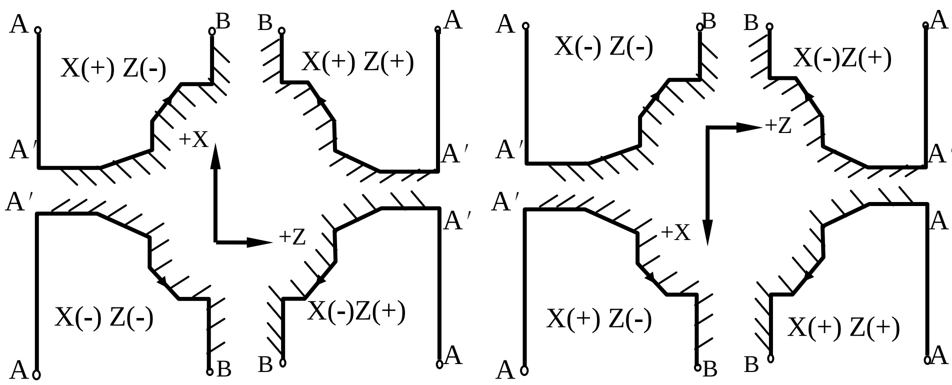


Figure 9.17 Direction of the finishing allowance in G71

Example 1

The initial point A is (46, 3). The depth of cut is 1.5mm (radius designation). The retraction amount is 1mm. The finishing allowance in the X direction is 0.6mm, and the finishing allowance in the Z direction is 0.1mm. The dashed line stands for the original part.

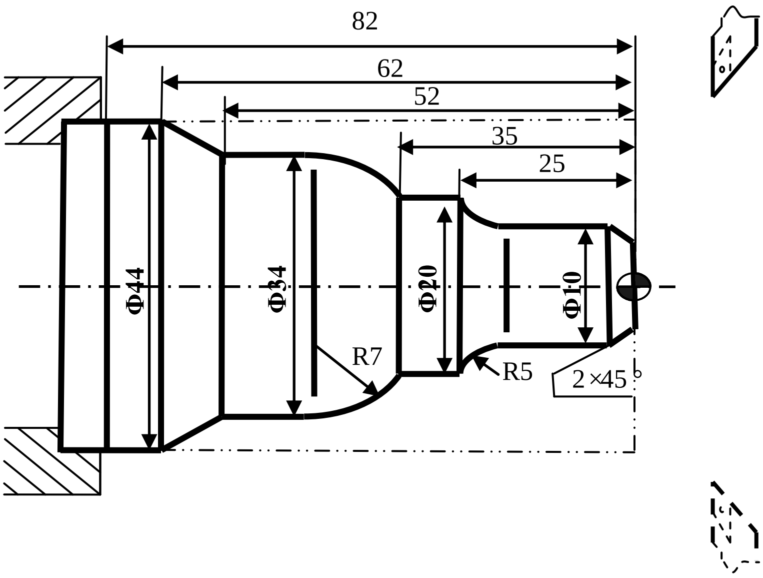


Figure 9.18 Outer Diameter Removal without Groove – Example

```
%3325
T0101
N1 G00 X80 Z80
N2 M03 S400
N3 G01 X46 Z3 F100
N4 G71U1.5R1P5Q13X0.6 Z0.1
N5 G00 X0
N6 G01 X10 Z-2
N7 Z-20
N8 G02 U10 W-5 R5
N9 G01 W-10
N10 G03 U14 W-7 R7
N11 G01 Z-52
N12 U10 W-10
N13 W-20
N14 X50
N15 G00 X80 Z80
N16 M05
N17 M30
```


Example 2

The initial point A is (6, 3). The depth of cut is 1.5mm (radius designation). The retraction amount is 1mm. The finishing allowance in the X direction is 0.6mm, and the finishing allowance in the Z direction is 0.1mm. The dashed line stands for the original part.

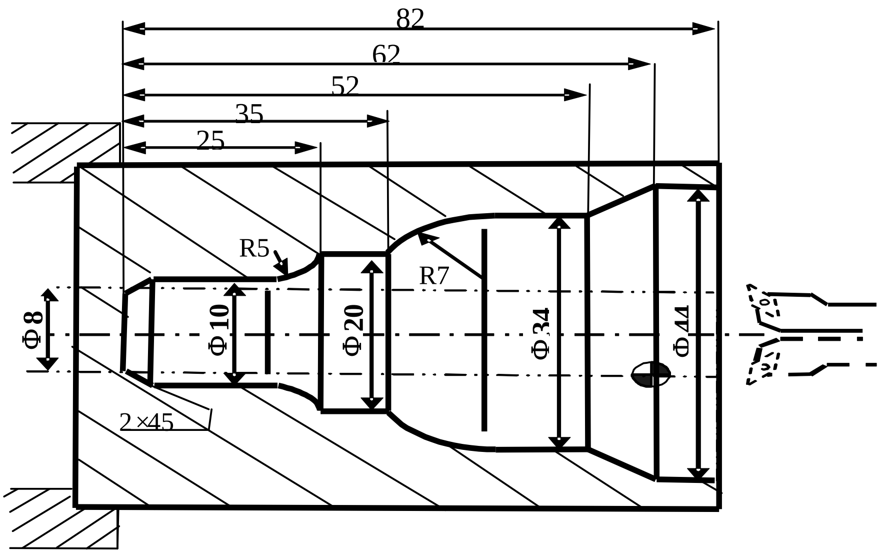


Figure 9.19 Internal Diameter Removal without Groove – Example

```
%3326
N1 T0101
N2 G00 X80 Z80
N3 M03 S400
N4 X6 Z5
G71U1R1P8Q16X-0.6Z0.1 F100
N5 G00 X80 Z80
N6 T0202
N7 G00 G41X6 Z5
N8 G00 X44
N9 G01 Z-20 F80
N10 U-10 W-10
N11 W-10
N12 G03 U-14 W-7 R7
N13 G01 W-10
N14 G02 U-10 W-5 R5
N15 G01 Z-80
N16 U-4 W-2
N17 G40 X4
N18 G00 Z80
N19 X80
N20 M30
```

- **Stock Removal in Turning with Groove**

Programming

G71 U(△ d) R(r) P(ns) Q(nf) E(e) F(f) S(s) T(t)

Explanation of the parameters

U(△ d) the cutting depth (radius designation). The cutting direction depends on the direction of AA'.

R(r) Retraction amount

P(ns) Sequence number of the first block for the finishing program.

Q(nf) Sequence number of the last block for the finishing program.

E(e) Distance and direction of finishing allowance on X axis. It is positive when it is outer diameter cutting. It is negative when it is internal diameter cutting.

F(f), S(s), T(t) F, S, T function are only effective for the rough machining, i.e, it is not effective in the finishing program – between P(ns) and Q(nf).

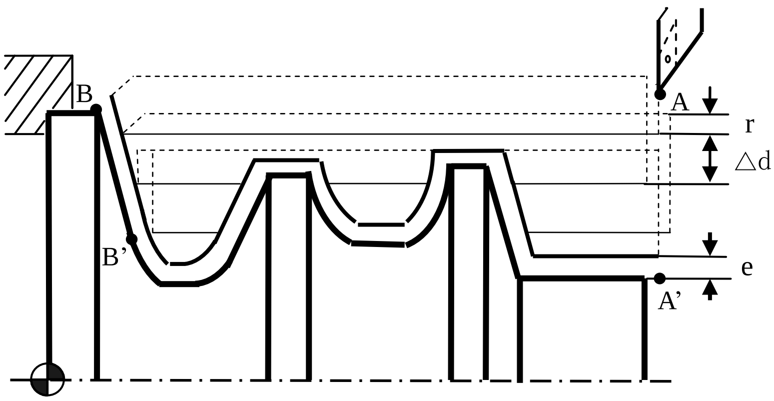


Figure 9.20 Stock Removal in Turning with Groove (G71)

Function

This command can do a stock removal in facing with groove. The machining path A→A'→B'→B.

Example

Use G71 to program.

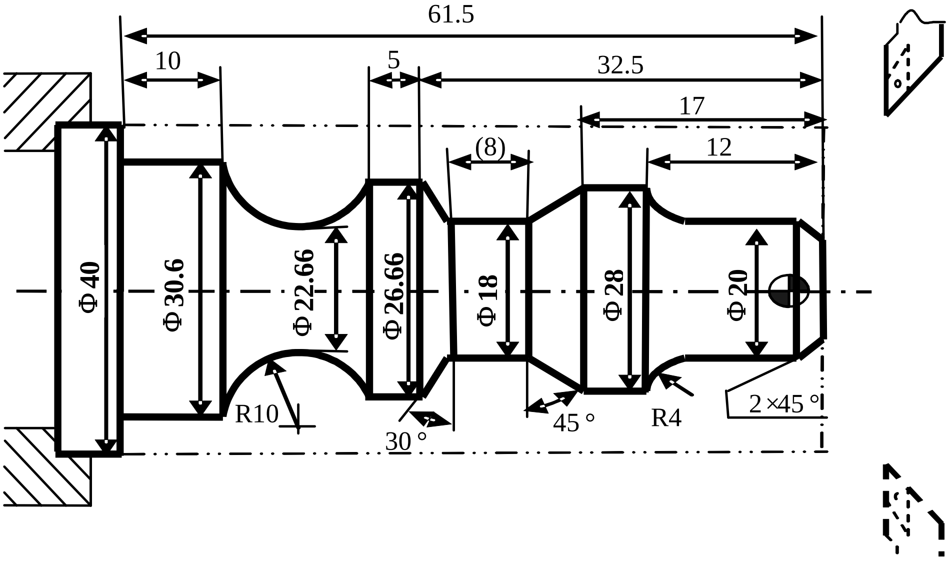


Figure 9.21 Stock Removal in Turning with Groove - Example

```
%3327
N1 T0101
N2 G00 X80 Z100
M03 S400
N3 G00 X42 Z3
N4G71U1R1P8Q19E0.3F100
N5 G00 X80 Z100
N6 T0202
N7 G00 G42 X42 Z3
N8 G00 X10
N9 G01 X20 Z-2 F80
N10 Z-8
N11 G02 X28 Z-12 R4
N12 G01 Z-17
N13 U-10 W-5
N14 W-8
N15 U8.66 W-2.5
N16 Z-37.5
N17 G02 X30.66 W-14 R10
N18 G01 W-10
N19 X40
N20 G00 G40 X80 Z100
N21 M30
```


9.2.2 Stock Removal in Facing (G72)

Programming

G72 W(Δd) R(r) P(ns) Q(nf) X(Δx) Z(Δz) F(f) S(s) T(t)

Explanation of the parameters

W(△ d) the cutting depth (radius designation). The cutting direction depends on the direction of AA'.

R(r) Retraction amount

P(ns) Sequence number of the first block for the finishing program.

Q(nf) Sequence number of the last block for the finishing program.

X(△ x) Distance and direction of finishing allowance on X axis

Z(△ z) Distance and direction of finishing allowance on Z axis

F(f), S(s), T(t) F, S, T function are only effective for the rough machining, i.e, it is not effective in the finishing program – between P(ns) and Q(nf).

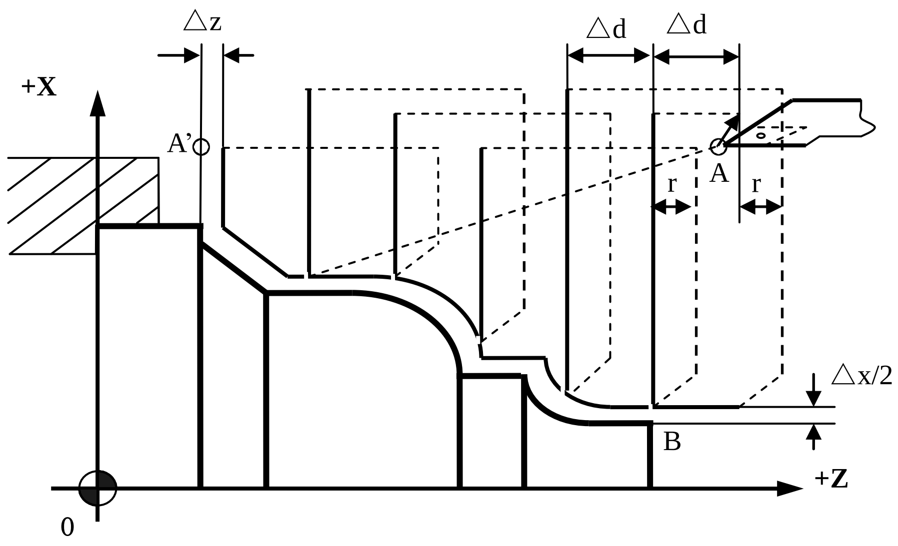


Figure 9.22 Stock Removal in Facing (G72)

Function

This command can do a stock removal in facing. The machining path is A→A'→B

Note

1) G00 or G01 must be used in the finishing program – between P(ns) and Q(nf). Otherwise, there is an alarm message.
2) G72 cannot be used in MDI mode.
3) G98 and G99 cannot used in the finishing program – between P(ns) and Q(nf).
4) The direction of △ x and △ z is shown in the following figure.

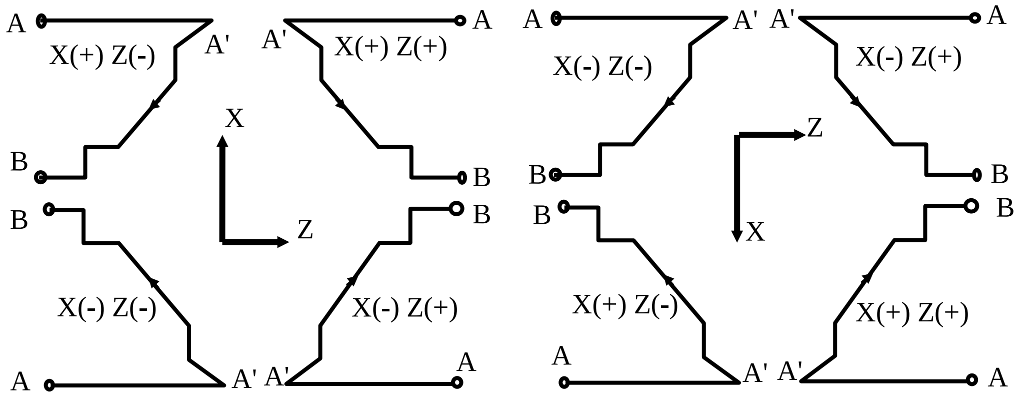


Figure 9.23 Direction of the finishing allowance in G72

Example 1

Use G72 to program. The initial point A is (80, 1). The depth of cutting is 1.2mm. The retraction amount is 1mm. The finishing allowance in the X direction is 0.2mm, and the finishing allowance in the Z direction is 0.5mm. The dashed line stands for the original part.

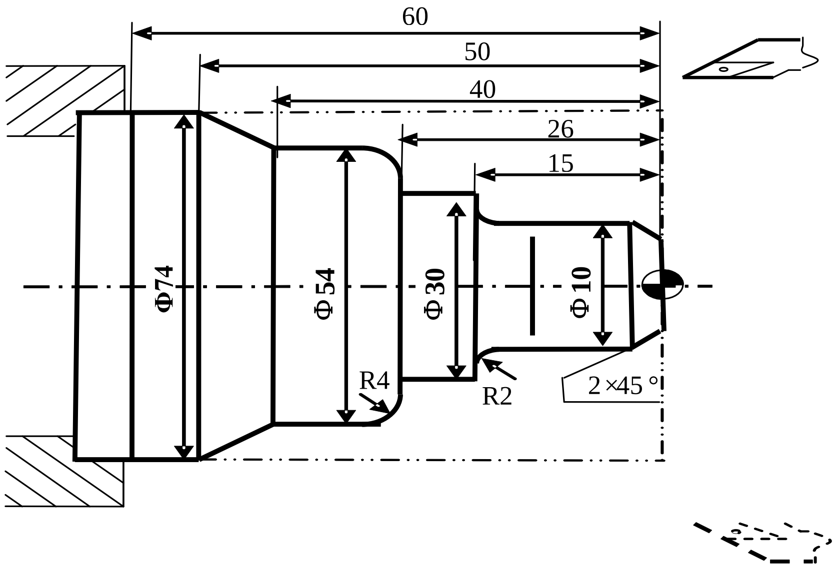


Figure 9.24 Outer Diameter Removal in Facing - Example

```
%3328
N1 T0101
N2 G00 X100 Z80
N3 M03 S400
N4 X80 Z1
N5 G72W1.2R1P8Q17X0.2Z0.5F100
N6 G00 X100 Z80
N7 G42 X80 Z1
N8 G00 Z-53
N9 G01 X54 Z-40 F80
N10 Z-30
N11 G02 U-8 W4 R4
N12 G01 X30
N13 Z-15
N14 U-16
N15 G03 U-4 W2 R2
N16 G01 Z-2
N17 U-6 W3
N18 G00 X50
N19 G40 X100 Z80
N20 M30
```

Example 2

Use G72 to program. The initial point A is (80, 1). The depth of cutting is 1.2mm. The retraction amount is 1mm. The finishing allowance in the X direction is 0.2mm, and the finishing allowance in the Z direction is 0.5mm. The dashed line stands for the original part.

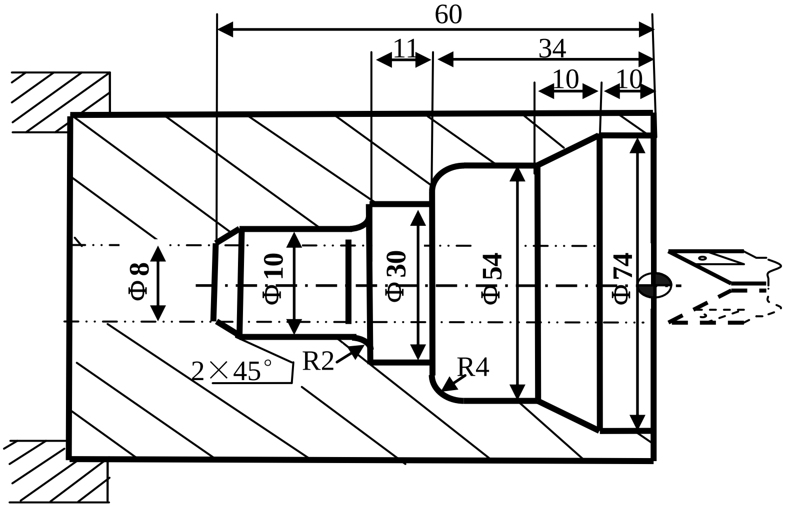


Figure 9.25 Internal Diameter Removal in Facing - Example

```
%3329
N1 T0101
N2 G00 X100 Z80
N3 M03 S400
N4 G00 X6 Z3
N5 G72W1.2R1P5Q15X-0.2Z0.5F100
N6 G00 Z-61
N7 G01 U6 W3 F80
N8 W10
N9 G03 U4 W2 R2
N10 G01 X30
N11 Z-34
N12 X46
N13 G02 U8 W4 R4
N14 G01 Z-20
N15 U20 W10
N16 Z3
N17 G00 X100 Z80
N18 M30
```

9.2.3 Pattern Repeating (G73)

Programming

G73 U(ΔI) W(ΔK) R(r) P(ns) Q(nf) X(Δx) Z(Δz) F(f) S(s) T(t)

Explanation of the parameters

U(Δ I) distance and direction of total roughing allowance in the X direction (radius designation).

W(Δ K) distance and direction of total roughing allowance in the X direction (radius designation)

R(r) Repeated times of cutting

P(ns) Sequence number of the first block for the finishing program.

Q(nf) Sequence number of the last block for the finishing program.

X(Δ x) Distance and direction of finishing allowance on X axis

Z(Δ z) Distance and direction of finishing allowance on Z axis

F(f), S(s), T(t) F, S, T function are only effective for the rough machining, i.e, it is not effective in the finishing program – between P(ns) and Q(nf).

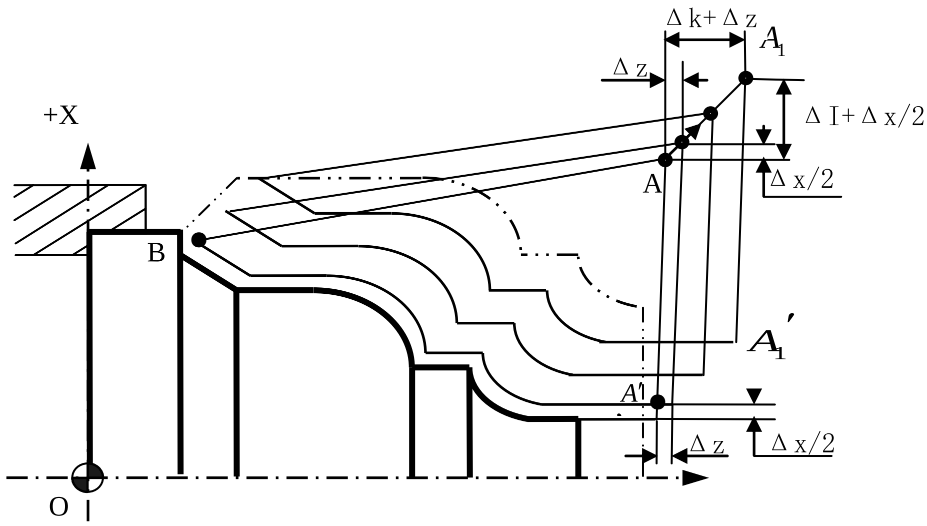


Figure 9.26 Pattern Repeating (G73)

Function

G73 command can cut a wokpiece at a fixed pattern repeatedly.

The machining path is A→A'→B.

Note

1) G00 or G01 must be used in the finishing program – between P(ns) and Q(nf). Otherwise, there is an alarm message.
2) G73 cannot be used in MDI mode.
3) G98 and G99 cannot used in the finishing program – between P(ns) and Q(nf).
4) The depth for each cutting on X axis = △I/r
   The depth for each cutting on Z axis = △K/r
5) The direction of △I and △K, and the direction of △x and △z should be noted.

Example

Use G73 to program. The initial point A is (60, 5). The total roughing allowance on X and Z axis are 3mm, 0.9mm, respectively. The times of rough cutting is 3. The finishing allowance on X and Z axis are 0.6mm, 0.1mm respectively. The dash-dot-line is the part's blank.

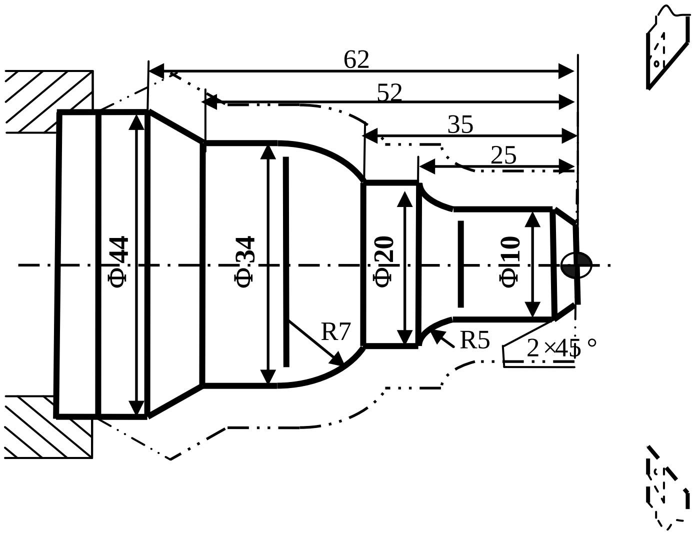


Figure 9.27 Pattern Repeating - Example

```
%3330
N1 T0101
N2 G00 X80 Z80
N3 M03 S400
N4 G00 X60 Z5
N5 G73U3W0.9R3P5Q13X0.6Z0.1F120
N6 G00 X0 Z3
N7 G01 U10 Z-2 F80
N8 Z-20
N9 G02 U10 W-5 R5
N10 G01 Z-35
N11 G03 U14 W-7 R7
N12 G01 Z-52
N13 U10 W-10
N14 U10
N15 G00 X80 Z80
N16 M30
```

9.2.4 Multiple Thread Cutting Cycle (G76)

Programming

G76 C(c) R(r) E(e) A(a) X(x) Z(z) I(i) K(k) U(d) V(Δdmin) Q(Δd) P(p) F(L) O

Explanation of the parameters

C(c) Repetitive count in finishing (1~99), modal value

R(r) Retraction amount on Z axis (00~99), modal value

E(e) Retraction amount on X axis (00~99), modal value

A(a) Angle of tool tip (two-digit number), modal value. It must be more than 10° and less than 80°.

X, Z Coordinate value of end point (point C) in absolute command.

x, z Coordinate value of end point (point C) with reference to the initial point (point A) in incremental command

I(i) Difference of thread radius. If i=0, it is straight thread cutting.

K(k) Height of thread. This value is specified by the radius value on X axis.

U(d) The finishing allowance (radius) (Figure 9.29).

V(Δdmin) The minimum cutting depth (radius). The cutting depth is Δdmin when the cutting depth ($\Delta d\sqrt{n}-\Delta d\sqrt{n-1}$) is less than Δdmin (Figure 9.29).

Q(Δd) Depth of cutting at the first cut (radius)

P(p) spindle angle of spindle reference pulse to the start point of cutting

F(L) Thread lead (same as G32)

O Acceleration constant of thread cutting retraction. When it is set to zero, the acceleartion is maximum. The more this value is, the acceleration time is longer, and the retraction is longer. Q must be set to zero or more than zero. O is modal code.

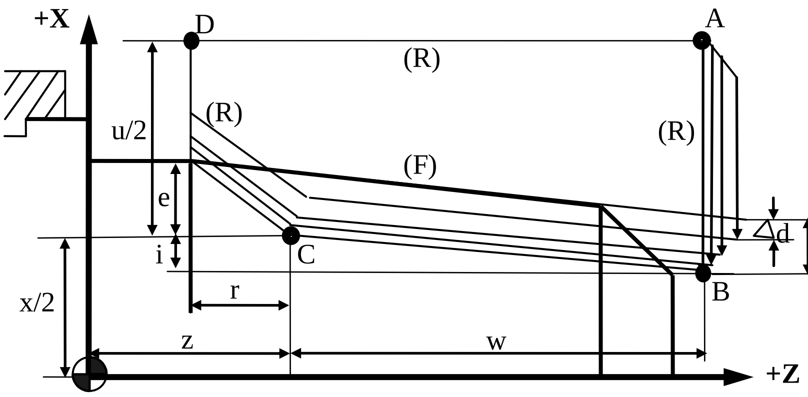


Figure 9.28 Multiple Thread Cutting Cycle (G76)

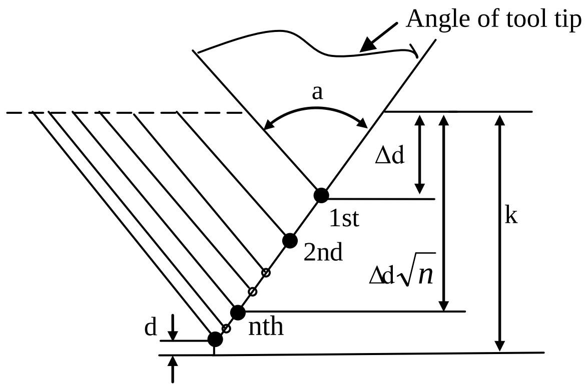


Figure 9.29 The depth of cutting

Function

G76 command can do the multiple thread cutting. The machining path is A→B→C→D.

Note

1) In G76, X(x) and Z(z) realize cycle machining. When incremental programming, note the sign of x and w (determined by the direction of tool path of AC and CD).
2) G76 cycle do singal-side cutting to reduce the stress of tool point. The cutting depth in 1st cut is Δd, the cutting depth in nth cut is $\Delta d\sqrt{n}$. The bite of each cycle is $\Delta d\ (\sqrt{n}-\sqrt{n-1})$.
3) The cutting speed of CD path is specified by feedrate. And the other paths are specified by rapid traverse speed.

Example

Use G76 to program. The thread is ZM60×2. Sizes in bracket is from standards. (tan1.79=0.03125)

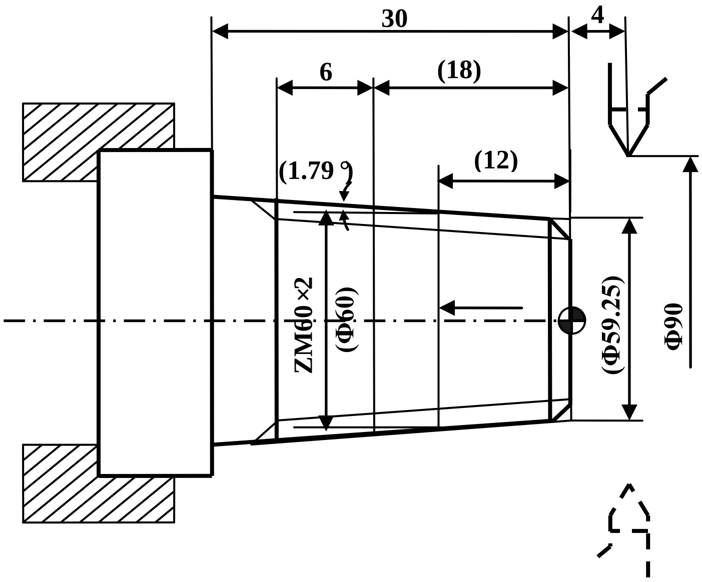


Figure 9.30 Multiple Thread Cutting Cycle - Example

%3331

N1 T0101 ; Change No.1 tool, set its coordiante

N2 G00 X100 Z100 ; To start of program or the position of changing tool

N3 M03 S400 ; Spindle CW 400r/min

N4 G00 X90 Z4 ; To the position of simple cycle start

N5 G80 X61.125 Z-30 I-1.063 F80 ; Machining cone thread surface

N6 G00 X100 Z100 M05 ;To the position of program start or tool change

N7 T0202 ; Change No.2 tool, set its coordiante

N8 M03 S300 ; Spindle CW 300r/min

N9 G00 X90 Z4 ; To the positon of thread cycle start

N10 G76C2R-3E1.3A60X58.15Z-24I-0.875K1.299U0.1V0.1F2

N11 G00 X100 Z100 ; Return to start or change tool position

N12 M05 ; Return to start or change tool position

N13 M30 ; Main program end and reset

10 Comprehensive Programming

10.1 Example 1

Program for the part shown in the figure. The processing condition: material: #45 steel, or aluminum; diameter of the part is Φ54mm, length of the part is 200mm. Tool selection: number 1 face tool is used to machine the part face, number 2 face cylindrical tool is used to rough turning the contour, number 3 face cylindrical tool is used to finish turning the contour, and number 4 cylindrical triple screw is used to machine the thread whose lead is 3mm, pitch is 1mm.

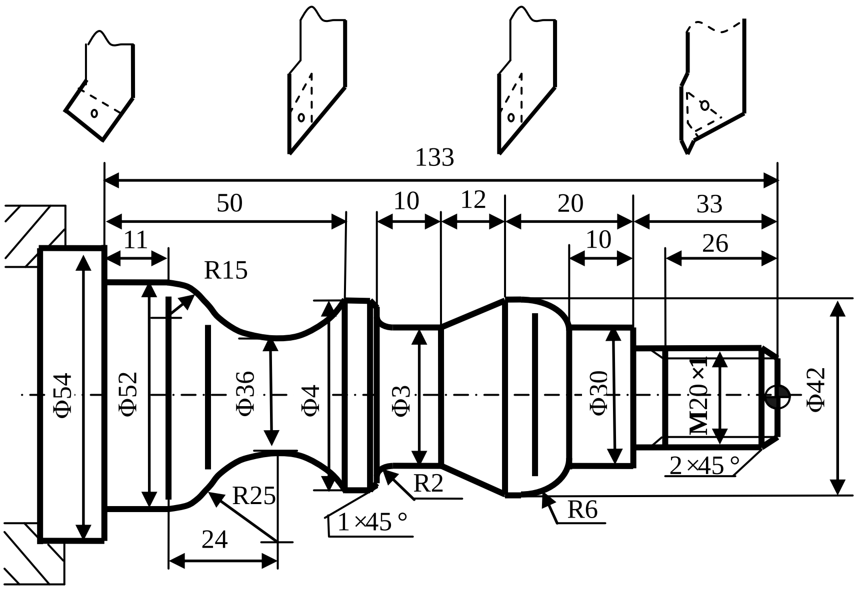


Figure 10.1 Comprehensive Program Example 1

```
%3365
N1 T0101
N2 M03 S500
N3 G00 X100 Z80
N4 G00 X60 Z5
N5 G81 X0 Z1.5 F100
N6 G81 X0 Z0
N7 G00 X100 Z80
N8 T0202
N9 G00 X60 Z3
N10 G80 X52.6 Z-133 F100
N11 G01 X54
N12 G71 U1 R1 P16 Q32 E0.3
```

```
N13 G00 X100 Z80
N14 T0303
N15 G00 G42 X70 Z3
N16 G01 X10 F100
N17 X19.95 Z-2
N18 Z-33
N19 G01 X30
N20 Z-43
N21 G03 X42 Z-49 R6
N22 G01 Z-53
N23 X36 Z-65
N24 Z-73
N25 G02 X40 Z-75 R2
N26 G01 X44
N27 X46 Z-76
N28 Z-84
N29 G02 Z-113 R25
N30 G03 X52 Z-122 R15
N31 G01 Z-133
N32 G01 X54
N33 G00 G40 X100 Z80
N34 M05
N35 T0404
N36 M03 S200
N37 G00 X30 Z5
N38G82X19.3Z-26R-3E1C2P120F3
N39G82X18.9Z-26R-3E1C2P120F3
N40G82X18.7Z-26R-3E1C2P120F3
N41G82X18.7Z-26R-3E1C2P120F3
N42 G76C2R-3E1A60X18.7Z-26 K0.65U0.1V0.1Q0.6P240F3
N43 G00 X100 Z80
N44 M30
```

10.2 Example 2

Program for the part shown in the figure. The processing condition: material: #45 steel, or aluminum; diameter of the part is Φ26mm, length of the part is 70mm. Tool selection: number 1 cylindrical tool is used to rough turning the contour, number 2 cylindrical tool is used to finish turning the contour, number 3 cylindrical thread tool is used to machine the thread. The pitch is 2mm. At last, number 4 parting-off tool is used to cut off the part.

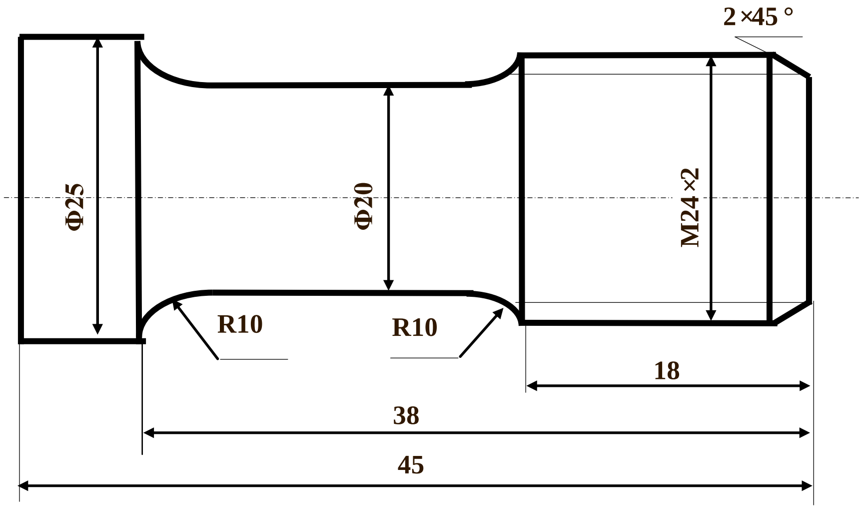


Figure 10.2 Comprehensive Programming Example 2

```
%3368
N1 T0101
N2 M03 S600
N3 G00 X100 Z30
N4 G00 X27 Z3
N5 G71 U1 R1 P9 Q E0.2 F100
N6 G00 X100 Z30
N7 T0202
N8 G00 G41 X27 Z3
N9 G00 X14 Z3
N10 G01 X24 Z-2 F80
N11 Z-18
N12 G02 X20 Z-24 R10
N13 G01 Z-31.39
N14 G02 X25 W-6.61 R10
N15 G01 Z-45
```

```
N16 G00 X30
N17 G40 X100 Z30
N18 T0303
N19 G00 X27 Z3
N20 G82 X23.1 Z-22 F2
N21 G82 X22.5 Z-22 F2
N22 G82 X21.9 Z-22 F2
N23 G82 X21.5 Z-22 F2
N24 G82 X21.4 Z-22 F2
N25 G82 X21.4 Z-22 F2
N26 G00 X100 Z30
N27 T0404
N28 G00 X30 Z-45
N29 G01 X3 F50
N30 G00 X100
N31 Z30
N13 M30
```

10.3 Example 3

Program for the tapered thread ZG2″ shown in the figure. According to the standard, the pitch is 2.309mm(25.4/11), the thread height is 1.479mm. Other sizes are shown in the figure. The depth of cut at each time is separately(diameter designation) 1mm, 0.7 mm, 0.6mm , 0.4mm and 0.26mm, and the angle of tool tip is 55° (tan1.79=0.031).

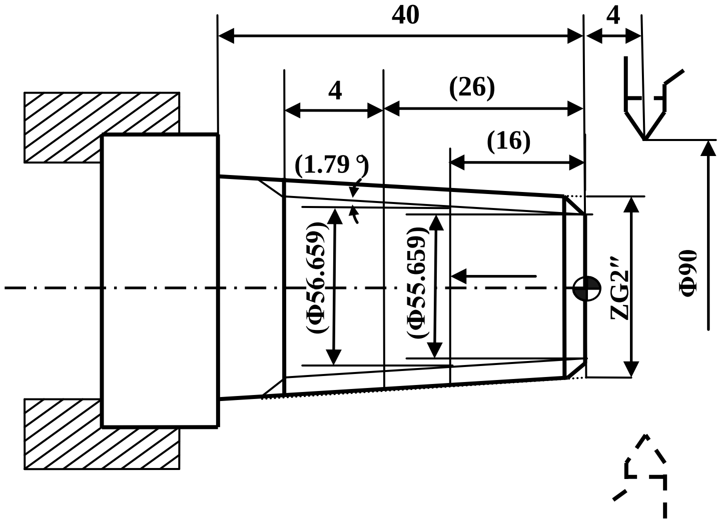


Figure 10.3 Comprehensive Programming Example 3

```
%3366
N1 T0101
N2 M03 S300
N3 G00 X100 Z100
N4 X90 Z4
N5 G80 X61.117 Z-40 I-1.375 F80
N6 G00 X100 Z100
N7 T0202
N8 G00 X90 Z4
N9 G82 X59.494 Z-30 I-1.063 F2.31
N10 G82 X58.794 Z-30 I-1.063 F2.31
N11 G82 X58.194 Z-30 I-1.063 F2.31
N12 G82 X57.794 Z-30 I-1.063 F2.31
N13 G82 X57.534 Z-30 I-1.063 F2.31
N14 G00 X100 Z100
N15 M30
```

10.4 Example 4

Program for the M40×2 inner thread shown in the figure. According to the standard, the pitch is 2.309mm(25.4/11), thread height is 1.299mm. Other sizes are shown in the figure. The depth of cut at each time(diameter designation) is 0.9mm, 0.6mm, 0.6mm, 0.4mm and 0.1mm. The angle of tool tip is 60°.

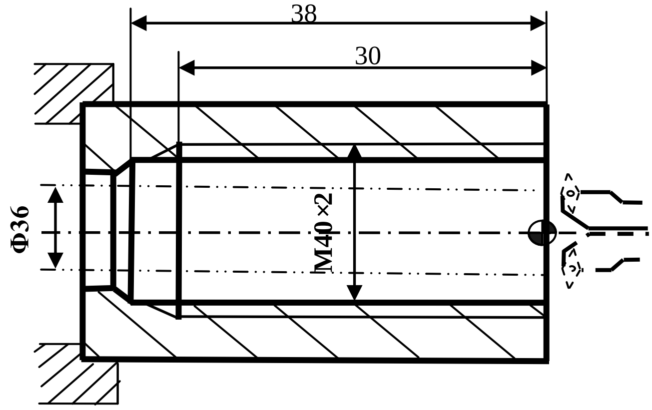


Figure 10.4 Comprehensive Programming Example 4

```
%3367
N1 T0101
N2 M03 S300
N3 G00 X100 Z100
N4 X20 Z4
N5 G80 X37.35 Z-38 F80
N6 G00 X100 Z100
N7 T0202
N8 G00 X20 Z4
N9 G82 X38.25 Z-30 R-4 E-1.3 F2
N10 G82 X38.85 Z-30 R-4 E-1.3 F2
N11 G82 X39.45 Z-30 R-4 E-1.3 F2
N12 G82 X39.85 Z-30 R-4 E-1.3 F2
N13 G82 X39.95 Z-30 R-4 E-1.3 F2
N14 G00 X100 Z100
N15 M30
```

11 Custom Macro

Similarly to subprogram, the custom macro function allows operators to define their own program. The way of calling the custom macro is same as subprogram's.

The difference is that custom macro allows use of variables, arithmetic and logic operations, selection and repetition.

11.1 Variables

Format and Explanation

#_ Variable is composed of a number sign (#) and a number.

Example

#1

#1=#2+100

11.1.1 Type of Variables

There are four types of variables.

Table 11-1 Type of Variables

Variable number	Type of variables	Function
#0~#49	Local variables	They are used in a macro program.
#50~#199	Common variables	They can be shared among different macro programs.
#200~#249	0 layers local variables	
#250~#299	1 layers local variables	
#300~#349	2 layers local variables	
#350~#399	3 layers local variables	
#400~#449	4 layers local variables	
#450~#499	5 layers local variables	
#500~#549	6 layers local variables	
#550~#599	7 layers local variables	
#600~	System variables	They are used to read and write NC data.

Note:

1) The operator can only use the #0~#599 local variables for programming.
2) Variables after #599 can only be used by the system programmer for reference.

11.1.2 System Variables

#1000 “current position X in machine coordinate system”

#1001 “current position Y in machine coordinate system”

#1002 “current position Z in machine coordinate system”

#1003 “current position A in machine coordinate system”

#1004 “current position B in machine coordinate system”

#1005 “current position X in machine coordinate system”

#1006 “current position U in machine coordinate system”

#1007 “current position V in machine coordinate system”

#1008 “current position W in machine coordinate system”

#1009 “diameter programming”

#1010 “position X – machine coordinate system in programming”

#1011 “position Y – machine coordinate system in programming”

#1012 “position Z – machine coordinate system in programming”

#1013 “position A – machine coordinate system in programming”

#1014 “position B – machine coordinate system in programming”

#1015 “position C – machine coordinate system in programming”

#1016 “position U – machine coordinate system in programming”

#1017 “position V – machine coordinate system in programming”

#1018 “position W – machine coordinate system in programming”

#1019 reserved

#1020 “position X – workpiece coordinate system in programming”

#1021 “position Y – workpiece coordinate system in programming”

#1022 “position Z – workpiece coordinate system in programming”

#1023 “position A – workpiece coordinate system in programming”

#1024 “position B – workpiece coordinate system in programming”

#1025 “position C – workpiece coordinate system in programming”

#1026 “position U – workpiece coordinate system in programming”

#1027 “position V – workpiece coordinate system in programming”

#1028 “position W – workpiece coordinate system in programming”

#1029 reserved

#1030 “origin X in workpiece coordinate system”

#1031 “origin Y in workpiece coordinate system”

#1032 “origin Z in workpiece coordinate system”

#1033 “origin A in workpiece coordinate system”

#1034 “origin B in workpiece coordinate system”

#1035 “origin C in workpiece coordinate system”

#1036 “origin U in workpiece coordinate system”

#1037 “origin V in workpiece coordinate system”

#1038 “origin W in workpiece coordinate system”

#1039 “axis of the coordinate system”

#1040 “origin X of G54”

#1041 “origin Y of G54”

#1042 “origin Z of G54”

#1043 “origin A of G54”

#1044 “origin B of G54”

#1045 “origin C of G54”

#1046 “origin U of G54”

#1047 “origin V of G54”

#1048 “origin W of G54”

#1049 reserved

#1050 “origin X of G55”

#1051 “origin Y of G55”

#1052 “origin Z of G55”

#1053 “origin A of G55”

#1054 “origin B of G55”

#1055 “origin C of G55”

#1056 “origin U of G55”

#1057 “origin V of G55”

#1058 “origin W of G55”

#1059 reserved

#1060 “origin X of G56”

#1061 “origin Y of G56”

#1062 “origin Z of G56”

#1063 “origin A of G56”

#1064 “origin B of G56”

#1065 “origin C of G56”

#1066 “origin U of G56”

#1067 “origin V of G56”

#1068 “origin W of G56”

#1069 reserved

#1070 “origin X of G57”

#1071 “origin Y of G57”

#1072 “origin Z of G57”

#1073 “origin A of G57”

#1074 “origin B of G57”

#1075 “origin C of G57”

#1076 “origin U of G57”

#1077 “origin V of G57”

#1078 “origin W of G57”

#1079 reserved

#1080 “origin X of G58”

#1081 “origin Y of G58”

#1082 “origin Z of G58”

#1083 “origin A of G58”

#1084 “origin B of G58”

#1085 “origin C of G58”

#1086 “origin U of G58”

#1087 “origin V of G58”

#1088 “origin W of G58”

#1089 reserved

#1090 “origin X of G59”

#1091 “origin Y of G59”

#1092 “origin Z of G59”

#1093 “origin A of G59”

#1094 “origin B of G59”

#1095 “origin C of G59”

#1096 “origin U of G59”

#1097 “origin V of G59”

#1098 “origin W of G59”

#1099 reserved

#1100 “break point X”

#1101 “break point Y”

#1102 “break point Z”

#1103 “break point A”

#1104 “break point B”

#1105 “break point C”

#1106 “break point U”

#1107 “break point V”

#1108 “break point W”

#1109 “axis of the coordinate system”

#1110 “middle point X of G28”

#1111 “middle point Y of G28”

#1112 “middle point Z of G28”

#1113 “middle point A of G28”

#1114 “middle point B of G28”

#1115 “middle point C of G28”

#1116 “middle point U of G28”

#1117 “middle point V of G28”

#1118 “middle point W of G28”

#1119 “shield of G28”

#1120 “mirror-image position X”

#1121 “mirror-image position Y”

#1122 “mirror-image position Z”

#1123 “mirror-image position A”

#1124 “mirror-image position B”

#1125 “mirror-image position C”

#1126 “mirror-image position U”

#1127 “mirror-image position V”

#1128 “mirror-image position W”

#1129 “shield of mirror image”

#1130 “rotational axis 1”

#1131 “rotational axis 2”

#1132 “rotation angle”

#1133 “shield of rotational axis”

#1134 reserved

#1135 “scale axis 1”

#1136 “scale axis 2”

#1137 “scale axis 3”

#1138 “scaling”

#1139 “shield of scale axis”

#1140 “code 1 of changing a coordinate system”

#1141 “code 2 of changing a coordinate system”

#1142 “code 3 of changing a coordinate system”

#1143 reserved

#1144 “number of tool length compensation”

#1145 “number of tool radius compensation”

#1146 “linear axis 1”

#1147 “linear axis 2”

#1148 “shield of virtual axis”

#1149 “specified feedrate”

#1150 “modal value of G code – 0”

#1151 “modal value of G code – 1”

#1152 “modal value of G code – 2”

#1153 “modal value of G code – 3”

#1154 “modal value of G code – 4”

#1155 “modal value of G code – 5”

#1156 “modal value of G code – 6”

#1157 “modal value of G code – 7”

#1158 “modal value of G code – 8”

#1159 “modal value of G code – 9”

#1160 “modal value of G code – 10”

#1161 “modal value of G code – 11”

#1162 “modal value of G code – 12”

#1163 “modal value of G code – 13”

#1164 “modal value of G code – 14”

#1165 “modal value of G code – 15”

#1166 “modal value of G code – 16”

#1167 “modal value of G code – 17”

#1168 “modal value of G code – 18”

#1169 “modal value of G code – 19”

#1170 “residual CACHE”

#1171 “spare CACHE”

#1172 “residual buffer storage”

#1173 “spare buffer storage”

#1174 reserved

#1175 reserved

#1176 reserved

#1177 reserved

#1178 reserved

#1179 reserved

#1180 reserved

#1181 reserved

#1182 reserved

#1183 reserved

#1184 reserved

#1185 reserved

#1186 reserved

#1187 reserved

#1188 reserved

#1189 reserved

#1190 “customized input”

#1191 “customized output”

#1192 “customized output shield”

#1193 reserved

#1194 reserved

#2000~#2600 data for the repetitive cycle

- #2000 number of contour point
- #2001~#2100 type of contour (0: G00, 1: G01, 2: G02, 3: G03)
- #2101~#2200 contour point X (diameter or radius designation)
- #2201~#2300 contour point Z
- #2301~#2400 contour point R
- #2401~#2500 contour point I
- #2501~#2600 contour point J

11.1.3 Memorable User-defined Variables

CNC provides 100 memorable user-defined variables #1300~#1399. The methods are as followed:

1. Set the initial value of macro variables
2. Add M94 where the macro is saved in the program

11.2 Constant

PI	π, 3.14151926
TRUE	True condition
FALSE	False condition

11.3 Operators and Expression

1) Mathematic operator

+, -, *, /

2) Conditional operator

EQ(=), NE(≠), GT(＞), GE(≥), LT(＜), LE(≤)

3) Logic operator

AND, OR, NOT

4) Function

SIN	Sine
COS	Cosine
TAN	Tangent
ATAN	Arctangent
ATAN2	Arctangent2
ABS	Absolute value
INT	Integer
SIGN	Sign
SQRT	Square root
EXP	Exponential function

5) Expression

The expressions are composed of constants, operators and variables.

Example:

175/SQRT[2] * COS[55 * PI/180];

#3*6 GT 14;

11.4 Assignment

Assignment refers to assign a variable value to a constant or expression.

Format:

Variable=constant or expression

Example

#2 = 175/SQRT[2] * COS[55 * PI/180]

#3 = 124.0

11.5 Selection statement IF, ELSE,ENDIF

Format (i)

IF Conditional expression

…

ELSE

…

ENDIF

Explanation (i)

If the specified conditional expression is satisfied, the statements between IF and ELSE are executed. If the specified conditional expression is not satisfied, the statements between ELSE and ENDIF are executed.

Format (ii)

IF Conditional expression

…

ENDIF

Explanation (ii)

If the specified conditional expression is satisfied, the statements between IF and ENDIF are executed. If the specified conditional expression is not satisfied, the system would proceed to the blocks after ENDIF.

11.6 Repetition Statement WHILE, ENDW

Format

WHILE Conditional expression

…

ENDW

Explanation

When the conditional expression is satisfied, the statements between WHILE and ENDW are executed. If the conditional expression is not satisfied, the system would proceed to the blocks after ENDW.

11.7 Macro Call

The following table shows the local variable and the corresponding system variable when it is macro call.

Table 11-2 Transmission of Macro Call

Local variables	System variables in macro call
#0	A
#1	B
#2	C
#3	D
#4	E
#5	F
#6	G
#7	H
#8	I
#9	J
#10	K
#11	L
#12	M
#13	N
#14	O
#15	P
#16	Q
#17	R
#18	S
#19	T
#20	U
#21	V
#22	W
#23	X
#24	Y
#25	Z
#26	Mode value of Z-plane in canned cycle
#27	Unavailable
#28	Unavailable
#29	Unavailable
#30	Absolute coordinate of 0-axis when subprogram call
#31	Absolute coordinate of 1-axis when subprogram call
#32	Absolute coordinate of 2-axis when subprogram call
#33	Absolute coordinate of 3-axis when subprogram call
#34	Absolute coordinate of 4-axis when subprogram call
#35	Absolute coordinate of 5-axis when subprogram call
#36	Absolute coordinate of 6-axis when subprogram call
#37	Absolute coordinate of 7-axis when subprogram call
#38	Absolute coordinate of 8-axis when subprogram call

Explanation

1) To check whether the variable is defined in the program, the format is as follows:

   AR [#Variable number]

   Return:

   0 – the variable is not defined

   90 – the variable is defined as absolute command G90

   91 – the variable is defined as incremental command G91

2) When it is macro call (subprogram or canned cycle) with G code, the system would copy the system variables (A~Z) to local variables #0-#25 in the macro. Meanwhile, the system can copy the axis position (machine coordinate value in absolute command) of nine channels to local variables #30-#38.
3) When calling a subprogram, the subprogram can modify the system mode.
4) When calling a canned cycle, the canned cycle does not modify the system mode.

11.8 Example

Example 1

Program the parabola B in interval [0, 8] shown in Figure 11.1. The parabola $B = -A^2/2$

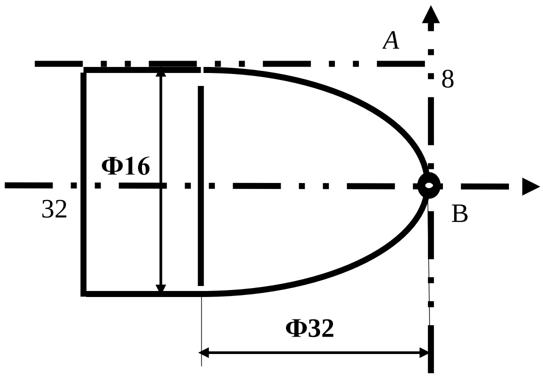


Figure 11.1 Custom Macro – Example 1

```
%3401
N1 T0101
N2 G37
N3 #10=0;
N4 M03 S600
N5 WHILE #10 LE 8
N6 #11=#10*#10/2
N7 G90 G01 X[#10] Z[-#11] F500
N8 #10=#10+0.08
N9 ENDW
N10 G00 Z0 M05
N11 G00 X0
N12 M30
```

Example 2

Program the parabola B in interval [0, 8] shown in Figure 11.2. The parabola $B = -A^2 / 2$

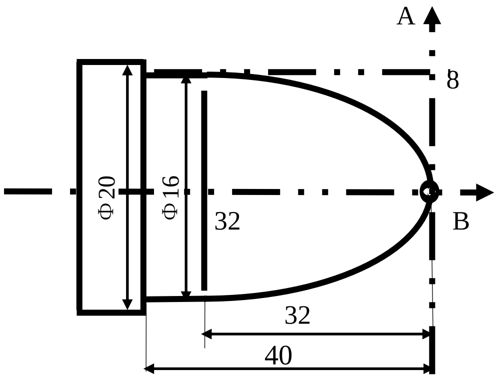


Figure 11.2 Custom Macro Example 2

```
%3402
T0101
G00 X21 Z3
M03 S600
#10=7.5
WHILE #10 GE 0
#11=#10*#10/2
G90 G01 X[2*#10+0.8] F500
Z[-#11+0.05]
U2
Z3
#10=#10-0.6
ENDW
#10=0
WHILE #10 LE 8
#11=#10*#10/2
G90 G01 X[2*#10] Z[-#11] F500
#10=#10+0.08
ENDW
G01 X16 Z-32
Z-40
G00 X20.5 Z3
M05
M30
```

Example 3

Program the parabola B in interval [12, 32] shown in Figure 11.3. The parabola $B = -A^2/2$

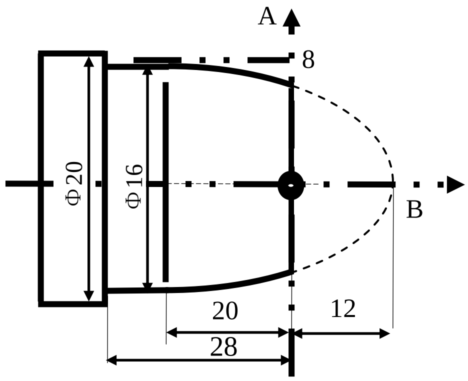


Figure 11.3 Custom Macro Example 3

```
%3403
N1 T0101
N2 G00 X20.5 Z3
N3 #11=12
N4 M03 S600
N5 WHILE   #11 LE 32
N6 #10=SQRT[2*[#11]]
N7 G90 G01 X[2*#10] Z[-[#11-12]] F500
N8 #11=#11+0.05
N9 ENDW
N10 G01 X16 Z-20
N11 Z-28
N12 G00 X20.5 Z3 M05
N13 M30
```

Example 4

Program the parabola B in interval [12, 32] shown in Figure 11.4. The parabola $B = -A^2 / 2$

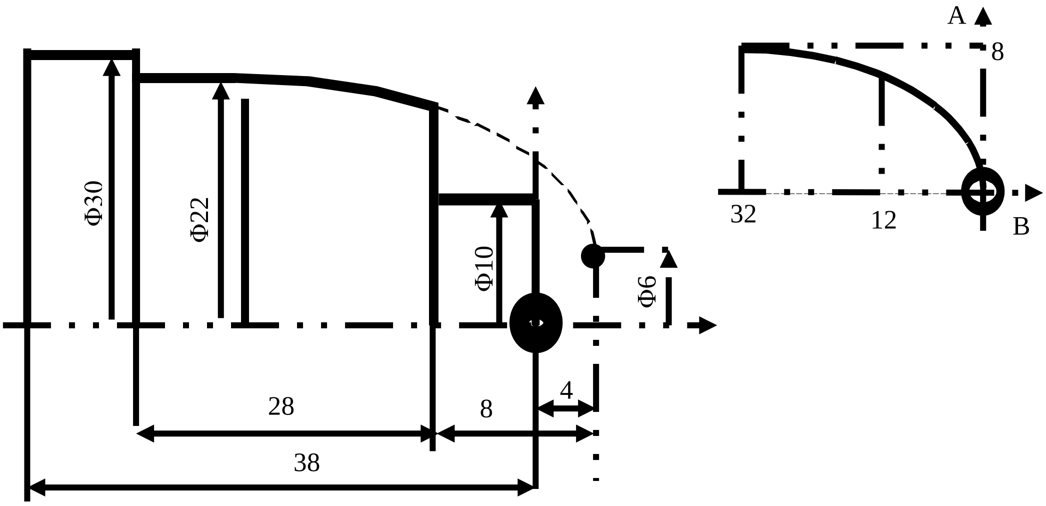


Figure 11.4 Custom Macro Example 4

%3404

N1 T0101

N2 G00 X25 Z3

N3 #11=12

N4 M03 S600

N5 WHILE #11 LE 32

N6 #10=SQRT[2*[#11]]

N7G90G01X[2*#10+6]Z[-[#11-4]]F500

N8 #11=#11+0.06

N9 ENDW

N10 G01 X22 Z-28

N11 Z-36

N12 X30

N13 Z-40

N12 G00 X25 Z3 M05

N13 M30

Example 5

Program the part shown in Figure 11.5.

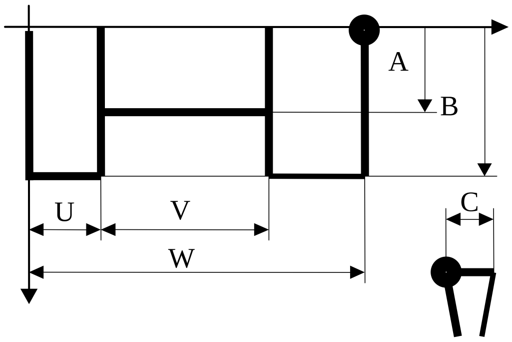


Figure 11.5 Custom Macro Example 5

```
%3405
N1 T0101
N2 G00 X90 Z30
N3 U10 V50 W80 A20 B40 C3 M98 P01(#20=10, #21=50, #22=80, #0=20, #1=40, #2=3)
N4 M30
%01
N1 G00 Z[-#22+#21+#20]
N2 X[#1+5]
N3 #10=#2
N4 WHILE  #10 LE #21
N5 G00 Z[-#22+#21+#20-#10]
N6 G01 X[#0]
N7 G00 X[#1+5]
N8 #10=#10+#2-1
N9 ENDW
N10 G00 Z[-#22+#20]
N11 G01 X[#0]
N12 G00 X[#1+5]
N13 G00 X90 Z30
N14 M99
```

