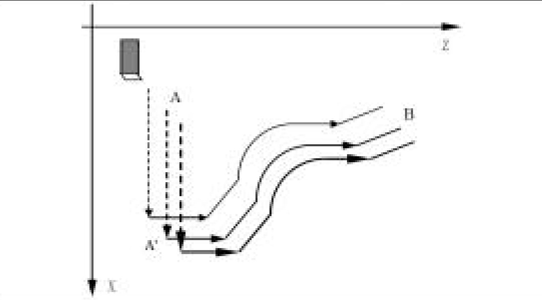
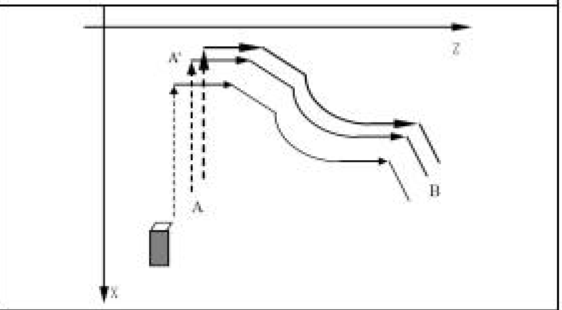
3) Δi<0 Δk<0 , Δu<0 Δw<0 ;	4) Δi>0 Δk<0 , Δu>0 Δw<0 ;
	

3.9.4 Finishing Cycle (G70)

After rough cutting by G71, G72 or G73, the following command permits finishing.

Format: G70 P(<u>ns</u>) Q(<u>nf</u>) ;

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

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

Note:

1 F, S, and T functions specified in the block G71, G72, G73 are not effective but those specified between sequence numbers "ns" and "nf" are effective in G70.

2 When the cycle machining by G70 is terminated, the tool is returned to the start point and the next block is read.

3 In blocks between "ns" and "nf" referred in G70 through G73, the subprogram cannot be called.

3.9.5 Usages of G71,G72,G73 & G70

3.9.5.1 Example of G71&G70

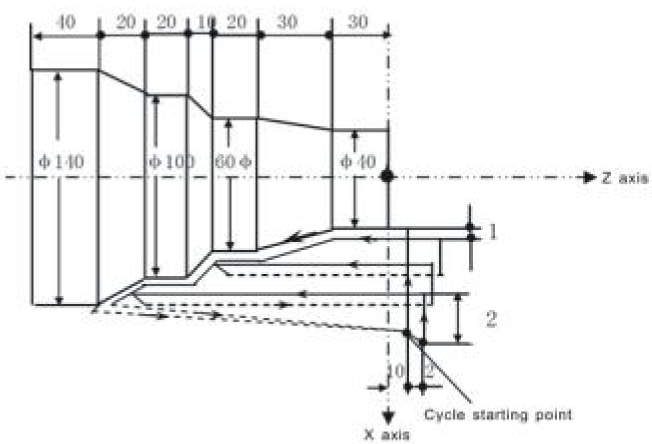


Fig3.9.6 Using of G71&G70

```
N10 M03 S1500;
N20 T0101;
N30 G00 X160 Z10;
N40 G71 U2 R1;
N50 G71 P60 Q120 U2 W1 F100 S2000
N60 G00 X40;
N70 G01 Z-30 F80;
N80 X60 W-30;
N90 W-20;
N100 X100 W-10;
N110 W-20;
N120 X140 W-20;
N130 G70 P60 Q120;
N140 G00 X200 Z50;
N150 T0100 M05;
N160 M30 ;
```

3.9.5.2 Example of G72 & G70

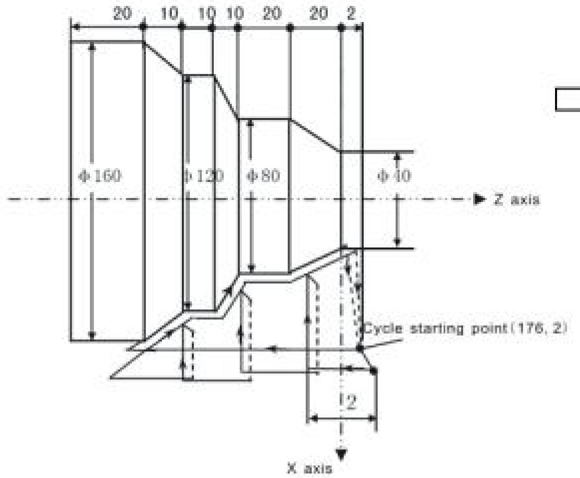


Fig3.9.7 Using of G72&G70

```
N10 M03 S2000;
N20 T0202;
N30 G00 X176 Z2;
N40 G72 W2 R1;
N50 G72 P60 Q120 U2 W1 F100 ;
N60 G00 Z-72;
N70 G01 X160 Z-70 F80;
N80 X120 W10;
N90 W10;
N100 X80 W10;
N110 W20;
N120 X36 W22.08;
N130 G70 P60 Q120;
N140 G00 X200 Z50;
N150 T0200 M05;
N160 M30;
```

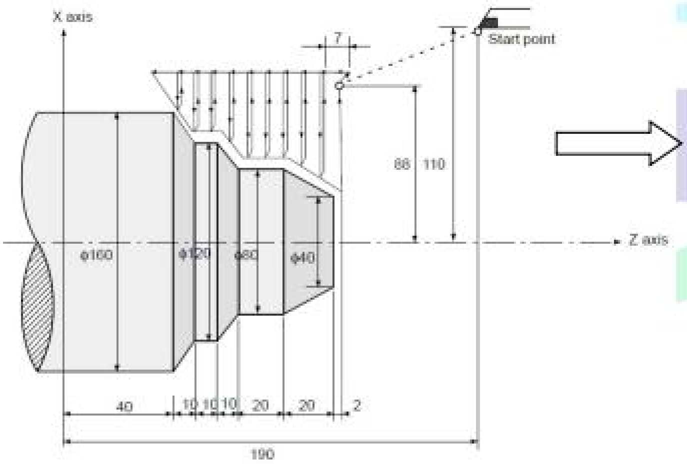


Fig3.9.8 Using of G72&G70

```
(Diameter designation, metric input)
N010 G50 X220.0 Z190.0 ;
N011 G00 X176.0 Z132.0 ;
N012 G72 W7.0 R1.0 ;
N013 G72 P014 Q019 U4.0 W2.0
F0.3 S550 ;
N014 G00 Z58.0 S700 ;
N015 G01 X120.0 W12.0 F0.15 ;
N016 W10.0 ;
N017 X80.0 W10.0 ;
N018 W20.0 ;
N019 X36.0 W22.0 ;
N020 G70 P014 Q019 ;
```

3.9.5.3 Example of G73&G70

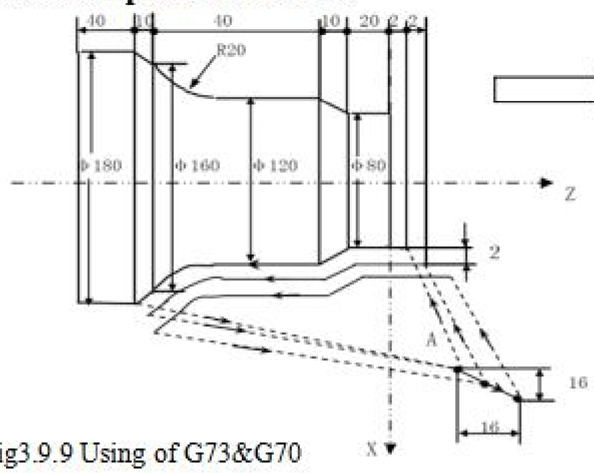


Fig3.9.9 Using of G73&G70

```
N10 M03 S3000;
N20 T0303;
N30 G00 X220 Z40;
N40 G73 U14 W14 R0.010;
N50 G73 P60 Q110 U4   W2 F100;
N60 G00 X80 Z2;
N70 G01 Z-20 F80;
N80 X120 W-10;
N90 W-20;
N100 G02 X160 W-20 R20;
N110 G01 X180 W-10;
N120 G70 P60 Q110;
N130 G00 X250 Z50;
N140 T0300 M05;
N150 M30;
```

(b)

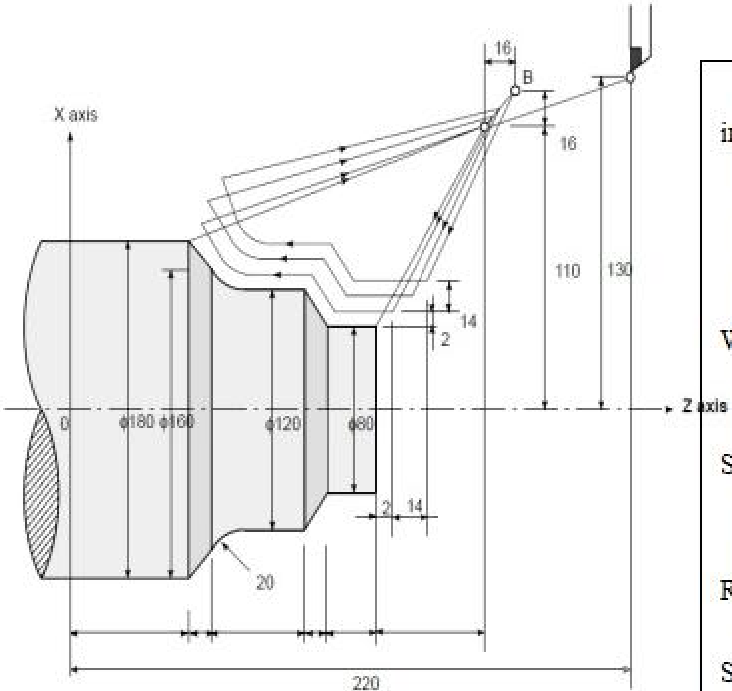


Fig3.9.10 Using of G73 & G70

(Diameter designation, metric input)

N010 G50 X260.0 Z220.0;
N011 G00 X220.0 Z160.0 ;
N012 G73 U14.0 W14.0 R3 ;
N013 G73 P014 Q019 U4.0 W2.0 F0.3 S0180 ;
N014 G00 X80.0 W–40.0 ;
N015 G01 W–20.0 F0.15 S0600 ;
N017 W–20.0 S0400 ;
N018 G02 X160.0 W–20.0 R20.0;
N019 G01 X180.0 W–10.0 S280;
N020 G70 P014 Q019 ;

3.9.6 End Face Peck Drilling Cycle (G74)

The following program generates the cutting path shown in Fig3.9.11. Chip breaking is possible in this cycle as shown below. If X (U) and Pare omitted, operation only in the Z axis results, to be used for drilling.

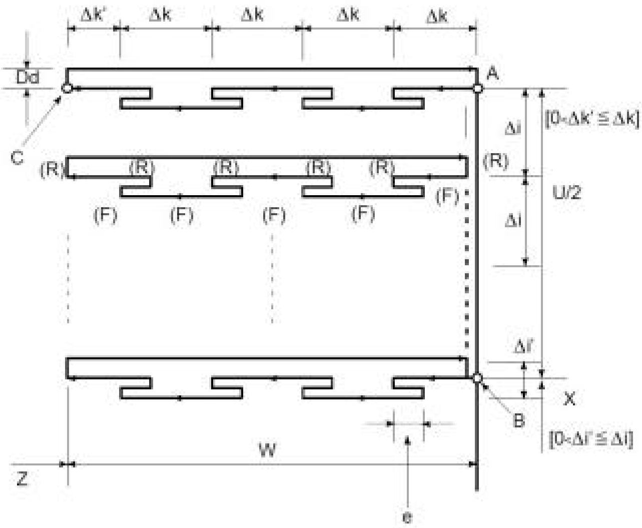


Fig3.9.11 Cutting Path in End Face Peck Drilling Cycle

Format: G74 R(<u>e</u>) ;

G74 X(<u>u</u>) P(<u>△i</u>) Z(<u>w</u>) Q(<u>△k</u>) R(<u>△d</u>) F(<u>f</u>) ;

e: Return amount;

This designation is modal and is not changed until the other value is designated. Also this value can be specified by P10 in User parameter, and the parameter is changed by the program command.

X : X component of point B

U : Incremental amount from A to B

Z : Z component of point C

W : Increment amount from A to C

Δi : Movement amount in X direction (without sign).

Δk : Depth of cut in Z direction (without sign),also can be set by P9 in User parameter. (Unit: um)

Δd : Relief amount of the tool at the cutting bottom. The sign of Δd is always plus(+). However, if address X (U) and Δi are omitted, the relief direction can be specified by the desired sign.

f : Feed rate

NOTE

1. While both e and d are specified by address R, the meanings of them are determined by the present of address X (U). When X(U) is specified, d is used.

2. The cycle machining is performed by G74 command with X(U) specification.

Execution process:(Fig3.9.11)

① Axial (Z axis) cutting feed △k from the starting point of G74, feed in Z negative direction when the coordinates of cutting end point is less than that of starting point in Z direction, otherwise, feed in Z positive direction;

②Axial(Z axis) rapid tool retraction e and its direction is opposite to feeding direction;

③ Cutting feed(Δk+e) again in Z direction, the end point of cutting feed is still in it between startingpoint An of axial cutting cycle and end point of axial tool infeed, cutting feed (Δk+e)again in Z direction and execute ②; after cutting feed (Δk+e)again in Z direction, the end point of cutting feed is on Bn or is not on it between An and Bn cutting feed to Bn in Z direction and then execute ④;

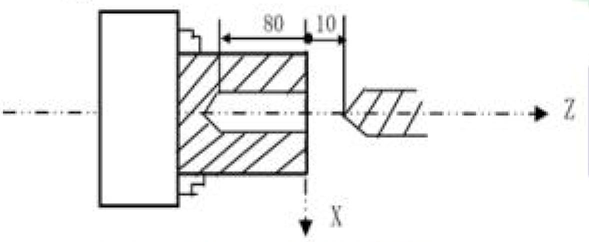
④ Radial(X axis) rapid tool retraction △d（radius value） to Cn , when the coordinates of Bf (cutting end point) is less than that of A (starting point) in X direction, retract tool in X positive, otherwise, retract tool in X negative direction;

⑤ Axial(Z axial) rapid retract tool to Dn, No. n axial cutting cycle is completed. If the current axial cutting cycle is not the last one, execute ⑥ ; if it is the previous one before the last axial cutting cycle, execute ⑦;

⑥ Radial(X axial)rapid tool infeed, and it direction is opposite to ④ retract tool. If the end point of tool infeed is still on it between A and Af (starting point of last axial cutting cycle) after tool infeed(△d+△i) (radius value) in X direction, i.e. n→An+1 and then execute ① (start the next axial cutting cycle); if the end point of tool infeed is not on it between Dn and Af after tool infeed (△d+△i) (radius value) in X direction, rapidly traverse to Af and execute ① to start the first axial cutting cycle;

⑦ Rapidly traverse to return to A in X direction, and G74 is completed.

Example:



N10 G00 X0 Z10;
N20 G74 R2;
N30 G74 Z-80 Q10000
F800;
N40 G00 X50 Z50;
N50 M30;

Fig3.9.12 Example of G74

3.9.7 Outer Diameter/Internal Diameter Drilling Cycle (G75)

The following program generates the cutting path shown in Fig3.9.13.

This is equivalent to G74 except that X is replaced by Z. Chip breaking is possible in this cycle, and grooving in X axis and peck drilling in X axis (in this case, Z, W, and Q are omitted) are possible.

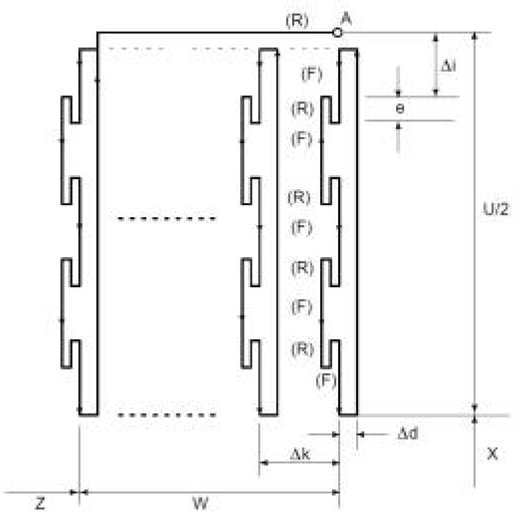


Fig3.9.13 Cutting Path in Outer/Internal Diameter Drilling Cycle

Format: G75 R(e) ;
G75 X(U) P(Δi) Z(w) Q(Δk) R(Δd) F(f) ;

X : X component of point B
U : Incremental amount from A to B
Z : Z component of point C
W : Increment amount from A to C

Δi : Movement amount in X direction (without sign),also can be set by P9 in User parameter. (Unit: um)

Δk : Depth of cut in Z direction (without sign),

Δd : Relief amount of the tool at the cutting bottom. The sign of Δd is always plus(+). However, if address X (U) and Δi are omitted, the relief direction can be specified by the desired sign.

f : Feed rate

Both G74 and G75 are used for grooving and drilling, and permit the tool to relief automatically. Four symmetrical patterns are considered,respectively.

Execution process:(Fig3.9.14)

① Radial (X axis) cutting feed △i from the starting point of radial cutting cycle, feed in X negative direction when the coordinates of cutting end point is less than that of starting point in X direction,otherwise, feed in X positive direction;

② Radial(X axis) rapid tool retraction e and its direction is opposite to the feed direction of ①;

③ Cutting feed(Δk+e) again in X direction, the end point of cutting feed is still in it between starting point An of radial cutting cycle and end point of radial tool infeed, cutting feed (Δ i+e)again in X direction and execute ②; after cutting feed (Δ i+e)again in X direction, the end point of cutting feed is on Bn or is not on it between An and Bn cutting feed to Bn in X direction and then execute ④ ;

④ Axial(Z axis) rapid tool retraction △d（radius value）to Cn , when the coordinates of Bf (cutting end point) is less than that of A (starting point) in Z direction, retract tool in Z positive, otherwise, retract tool in Z negative direction;

⑤ Radial(Z axis) rapid retract tool to Dn, No. n radial cutting cycle is completed. The current radial cutting cycle is not the last one, execute ⑥ ; if it is the previous one before the last radial cutting cycle, execute ⑦;

⑥ Axial(X axis)rapid tool infeed, and it direction is opposite to ④ retract tool. If the end point of tool infeed is still on it between A and Af (starting point of last radial cutting cycle) after tool infeed (△d+△k) (radius value) in Z direction, i.e. Dn→An+1 and then execute ① (start the nextradial cutting cycle); if the end point of tool infeed is not on it between Dn and Af after tool infeed (△ d+ △ k) in Z direction, rapidly traverse to Af and execute ① to start the first radial cuttingcycle;

⑦ Rapidly traverse to return to A in Z direction, and G75 is completed.

Example:

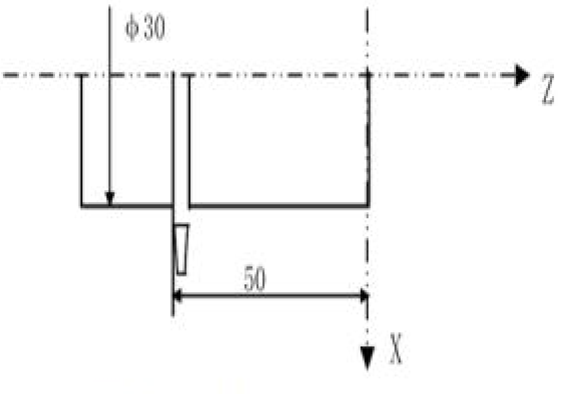


Fig3.9.14 Example of G75

```
N10 M03 S1000;
N20 T0101;
N30 G00 X35 Z-50;
N40 G75 R1;
N50 G75 X-1 P5000 F60;
N60 G00 X100 Z50 M09;
N70 M05;
N80 T0100;
N90 M30;
```

3.9.8 Multiple Thread Cutting Cycle (G76)

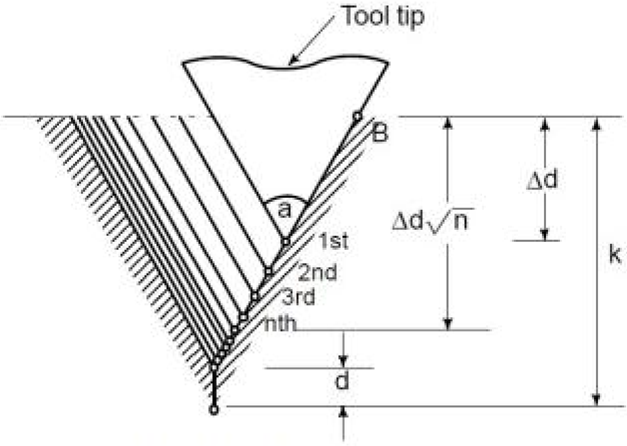


Fig3.9.15 Detailes of Cutting

Format: **G76 P(b)(c)(m)(r)(a) Q(Δdmin) R(d) ;**
G76 X(U)_ Z(W)_ R(i) P(k) Q(Δd) F(I)_ L(L)[or SP];

P actually consists of multiple values which control the thread behavior.

b: 0———Infeed decremently;
1———Equidistant infeed;
2———If the fist feed is too long in digression feed,so divide into two infeed.

c: 0———right enter;
1———left enter;
2———middle enter
3———right and left enter, the first feed is middle.

m: Repetitive count in finishing (1 to 99)

This designation is modal and is not changed until the other value is designated. Also this value can be specified by P11 in User parameter, and the parameterr is changed by the program command.

r: Chamfering amount.

When the thread lead is expressed by L, the value of L can be set from 0.0L to 9.9L in 0.1L increment (2–digit number from 00 to 90). This designation is modal and is not changed until the other value is designated. Also this value can be specified by P12 in User parameter, and the parameter is changed by the program command.

a: Angle of tool tip

One of six kinds of angle, 80°, 60°, 55°, 30°, 29°, and 0°, can be selected,and specified by 2–digit number. This designation is modal and is not changed until the other value is designated. Also this value can be specified by P13 in User Parameter, and the parameter is changed by the program command.

b ,c,m, r, and a are specified by address P at the same time.

Example: When b=2, c=3, m=1, r=1.2K, a=60°, specify as shown below (K is lead of thread).

$P\,\underset{b}{\underline{2}}\,\underset{c}{\underline{3}}\,\underset{m}{\underline{01}}\,\underset{r}{\underline{12}}\,\underset{a}{\underline{60}}$, coding instructe is : P23011260

Q(Δdmin): Minimum cutting depth. (specified by the radius value)

When the cutting depth of one cycle operation ($\Delta d^* \sqrt{n} - \Delta d^* \sqrt{n-1}$) becomes smaller than this limit, the cutting depth is clamped at this value. This designation is modal and is not changed until the other value is designated. Also this value can be specified by P14 in User Parameter, and the parameter is changed by the program command. Unit: um.

R(d): Finishing allowance.

This designation is modal and is not changed until the other value is designated. Also this value can be specified by P15 in User parameter, and the parameter is changed by the program command.

X: Absolute coordinates (unit: mm) of thread end point in X direction;

U: Different value (unit: mm) of absolute coordinates between thread end point and starting point in X direction;

Z: Absolute coordinates (unit: mm) of thread end point in Z direction;

W: Different value (unit: mm) of absolute coordinates between thread end point and starting point in Z direction;

R(i): Difference of thread radius.

If i = 0, ordinary straight thread cutting can be made.

P(k): Height of thread. This value is specified by the radius value.

F: metric thread pitch. (same as G32).0.1~500.000mm.

I: Thread teeth per inch for inch thread. 0.1~99 teeth.

L: multiple thread head numbers.

SP: starting angle: 0-360°, unit is 0.001 degree. No specify means 0 degree.

△d: Depth of cut in 1st cut (radius value) .Or infeed times.

First cut amount (with G32 threading) in microns; or feed times.

P24 in User parameter is set for meaning of Q (△d) of G76. [P24 = 8 ,times of roughing infeed]. When P24 = 8, Q (△d) means times that needed to complete the roughing cycle, the default is 1; otherwise Q (△d) means that depth of cut in 1st cut. Q (△d) , times of infeed, also there are modes of equidistant infeed and decremently infeed.

The infeed amount and infeed times of roughing in all case are as follows:

1) b = 0, P24 ≠ 8, every infeed depth: $\nabla d\sqrt{n}$;

2) b = 0, P24 = 8, each infeed depth is: the same way as a) according to △d calculated as: $\frac{(k-d)}{\sqrt{\nabla d}}\sqrt{n}$, infeed times is △d;

3) b=1, P24≠8, amount of each infeed:△d,times of roughing infeed is (k-d)/△d;

4) b=1, P24=8,the amount of each infeed: (k-d)/△d,roughing feed times for△d;

The thread cutting cycle as shown in Fig3.9.16 is programmed by G76 command.

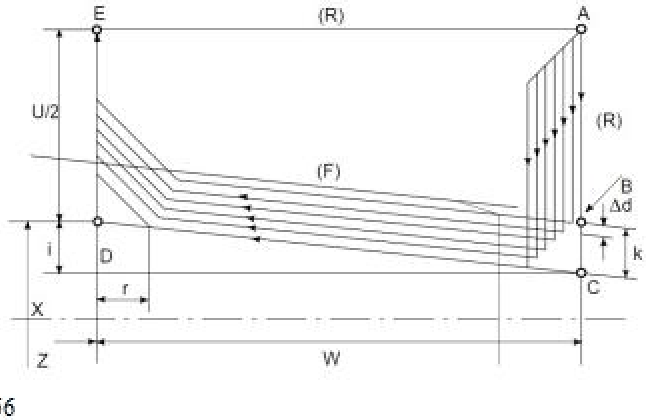


Fig3.9.16 Cutting path in multiple thread cut

Execution process:

① The tool rapidly traverses to B(1st),and the thread cutting depth is △d. The tool only traverses in X direction when a=0; the tool traverses in X and Z direction and its direction is the same that of A→D when a≠0；

② The tool cuts threads paralleling with C→D to the intersection of D→E (r≠0: thread run-out)；

③ The tool rapidly traverses to E point in X direction；

④ The tool rapidly traverses to A point in Z direction and the single roughing cycle is completed；

⑤ The tool rapidly traverses again to tool infeed to B(nth) (is the roughing times), the cutting depth is the bigger value of ($\sqrt{n}\times\triangle$d),($\sqrt{n-1}\times\triangle$d+△dmin) and execute ② if the cutting depth is less than（k-d）; if the cutting depth is more than or equal to（k-d）, the tool infeeds（k-d）to B((n+1)th) ,and then,execute ⑥ to complete the last thread roughing；

⑥ The tool cuts threads paralleling with C→D to the intersection of D→E (r≠0: thread run-out)；

⑦ The tool rapidly traverses to E point in X direction；

⑧ The tool rapidly traverses to A point in Z direction and the thread roughing cycle is completed to execute the finishing；

⑨ After the tool rapidly traverses to B(the cutting depth is k and the cutting travel is d), execute the thread finishing, at last the tool returns to A point and so the thread finishing cycle is completed；

⑩ If the finishing cycle times is less than m, execute ⑨ to perform the finishing cycle, the thread cutting depth is k and the cutting travel is 0; if the finishing cycle times is equal to m, G76 compound thread machining cycle is completed.

Thread Cutting Cycle Retract:

When feed hold is applied during threading in the multiple thread cutting cycle (G76), the tool quickly retracts in the same way as in chamfering performed at the end of the thread cutting cycle. The tool goes back to the start point of the cycle. When cycle start is triggered, the multiple thread cutting cycle resumes.

Note: 1. The meanings of data specified by address P, Q, and R determined by the presence of X(U) and X (W).

2. The cycle machining is performed by G76 command with X (U) and Z (W) specification. By using this cycle , one edge cutting is performed and the load on the tool tip is reduced.Making the cutting depth d for the first path, and dn for the nth path, cutting amount per one cycle is held constant.Four symmetrical patterns are considered corresponding to the sign of each address. The internal thread cutting is available. In the above figure, the feed rate between C and D is specified by address F, and in the other path, at rapid traverse. The sign of incrementaldimensions for the above figure is as follows:

U, W : minus (determined by the direction of the tool path AC and CD.)

R(i) : minus (determined by the direction of the tool path AC.)

P(k) : plus (always)

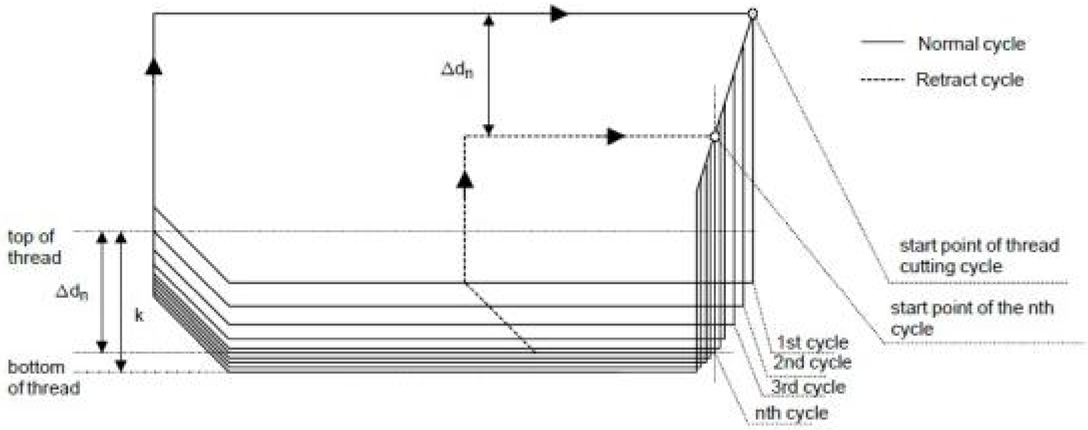
Q(△d) : plus (always)

3. Notes on thread cutting are the same as those on G32 thread cutting and G92 thread cutting cycle.

4. The designation of chamfering is also effective for G92 thread cutting cycle.

5. The tool returns to the cycle start point (cutting depth dn) as soon as the feed hold status isentered

during thread cutting. (dn : cutting depth in nth cut)



6. If start point of thread cutting cycle is close to a workpiece, tool may interfere with the workpiece during retract cycle because of passing along the retract cycle route described at Note 5. Therefore start point of thread cutting cycle must be at least k(height of thread) away from the top of thread.

7. pay attention to cut thread, use G32 to cut is the same as using G92.

8. specify the chamfering amount of thread, it's also effective to the G92 thread cutting circle.

9. when matching step motor, because of the acceleration or deceleration the thread in tail will be inhomogeneous. So should choose the linear acceleration or deceleration to control X axis with G00 to back tail fast.

Example: Multiple Repetitive Cycle G76

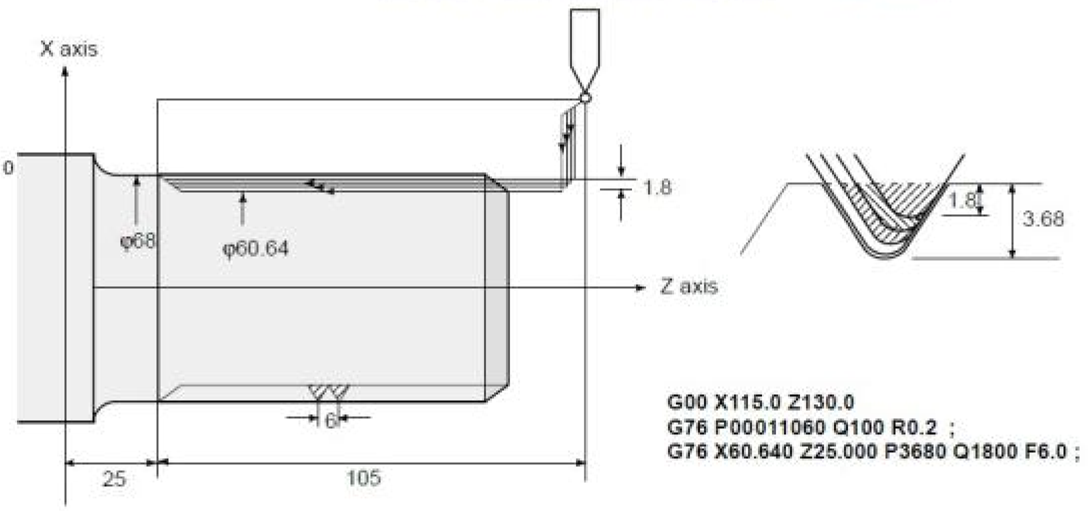


Fig3.9.16 Example of G76

3.9.9 Notes On Multiple Repetitive Cycle (G70 ~ G76)

1. In the blocks where the multiple repetitive cycle are commanded, the addresses P, Q, X, Z, U, W, and R should be specified correctly for each block.

2. In the block which is specified by address P of G71, G72 or G73, G00 or G01 group should be commanded.

3. In MDI mode, G70, G71, G72, or G73 cannot be commanded.G74, G75, and G76 can be commanded in MDI mode.

4. In the blocks in which G70, G71, G72, or G73 are commanded and between the sequence number specified by P and Q, M98 (subprogram call) and M99 (subprogram end) cannot be commanded.

5. In the blocks between the sequence number specified by P and Q, the following commands cannot be specified.

- ❖ One shot G code except for G04 (dwell)
- ❖ 01 group G code except for G00, G01, G02, and G03
- ❖ 06 group G code
- ❖ M98 / M99

6. While a multiple repetitive cycle (G70~G76) is being executed, it is possible to stop the cycle and to perform manual operation. But, when the cycle operation is restarted, the tool should be returned to the position where the cycle operation is stopped. If the cycle operation is restarted without returning to the stop position, the movement in manual operation is added to the absolute value, and the tool path is shifted by the movement amount in manual operation.

7. When G70, G71, G72, or G73 is executed, the sequence number specified by address P and Q should not be specified twice or more in the same program.

8. The blocks between the sequence number specified by P and Q on the multiple repetitive cycle must not be programmed by using "Direct Drawing Dimensions Programming" or "Chamfering/Corner R".

9. G74, G75, and G76 also do not support the input of a decimal point for P or Q. The least input increments are used as the units in which the amount of travel and depth of cut are specified.

10. When #1 = 2500 is executed using a custom macro, 2500.000 is assigned to #1. In such a case, P#1 is equivalent to P2500.

11. Tool nose radius compensation cannot be applied to G71, G72, G73, G74, G75, or G76.

12. The multiple repetitive cycle cannot be executing during DNC operation.

13. Interruption type custom macro cannot be executed during executing the multiple repetitive cycle.

3.10 Skip Function(G31,G311)

Linear interpolation can be commanded by specifying axial move following the G31 command, like G01. If an external skip signal is input during the execution of this command, execution of the command is interrupted and the next block is executed.

The skip function is used when the end of machining is not programmed but specified with a signal from the machine, for example, in grinding. It is used also for measuring the dimensions of a workpiece.

For details of how to use this function, refer to the manual supplied by the machine tool builder.

Format: G31 IP_ P_ ;

G311 IP_ P_ ;

G31&G311 are One-shot G code.(It is effective only in the block in which it is specified).

IP_: coordinate value ;

P_ : Specify jumping line number & detecting if input point is valid;

P(a)(b)(c)

a: Jumping line number specfied by " N**" ;when missed, stop running current line, and jump to next block and run;

b: 10 or 20 ; 10 means that when input point is valid, skip to specfied line, when input point is invalid, don't skip, keep on running or alarm hint; 20 means that when input point is invalid, skip to specfied line ; when input point is valid, don't skip , keep on running or alarm hint;

c: Specify detecting input point. address of input point, X00~X39

P 56 10 24 } when Input point X24 is valid, stop running current
a b c liune, jump to N56 block and run.

Difference between G31 & G311: When system don't detect signal of specified input point, G31 don't hint alarm & keep on running program; G311 will hint that don't input is valid & stop running, after Press "Enter", it will go on run program ;

The coordinate values when the skip signal is turned on can be used in a custom macro because they are stored in the custom macro system variable as follows:

#5021 X axis coordinate value

#5022 Third axis (Cs axis) coordinate value

#5023 Z axis coordinate value

#5024 4th axis (A axis) coordinate value

Example: G31 X50 Z100 F100 P**33**1022 ;if X22 is valid then go to **N33**(line no.).

G311 X50 Z100 F100 P2021 ;if X21 is invalid then go to next line.Valid-Alarm.

3.11 Block Cycle (G22,G800)

G22 is a program loop instruction, G800 is the end of the cycle instruction. Both must be paired for parts machining process requires repeated occasions. L is the number of cycles, ranging from 1-99999. Cycle instructions can be nested.

Format:

G22 L_

.

. } **Block Cycle**

.

G800 ;End

Example:

```
N0000  M03 M08
N0001  G0   X200 Z200
N0002  G01 W-100 F300
N0003  G22  L6              ; 6 times cycle
N0004  G01  U-22  F100
```

```
N0005   W－11   U6
N0006   W-30
N0007   W-10 U5
N0008   G0   U10
N0009   W51
N0010   G800                ;loop end
N0011   G26
N0012   M30
```

3.12 Return to Starting Point (G26,G261~G264)

These instructions are used for return back to the starting point of the program. Starting point is coordinate position of N0000 block. The returning speed is same to G00 speed.

Format:

G26 ; All Feeding axes return to starting point.
G261 ; X-axis return to starting point
G262 ; Y(C)-axis return to starting point
G263 ; Z-axis return to starting point
G264 ; A-axis reutm to starting point

3.13 Save Current Position (G25)

G25 is used for memory current coordinate position of all axes(XZYA), save current position as specified point.

Format: G25 ; Save current coordinate

3.14 Return to Specified Position (G61,G611~G614)

These instructions are used for return to point specified by G25.

G61 ; all axes return to specified point ;
G611 ; X-axis returns to specified point ;
G612 ; Y(C)-axis returns to specified point ;
G613 ; Z-axis returns to specified point ;
G614 ; A-axis returns to specified point ;

Note: If user don't use G25 to save current position, these instructions will return to starting point as G26.

Example:

```
N0000   G0 X20   Z80          ;
N0001   G01 U5 W－16 F200     ;
N0002   W－100                ;
N0003   G00 U10               ;
N0004   Z80                   ;
N0005   G25                   ; save current positon (X35,Z80)
N0006   G01 U10 W-30          ;
N0007   G0 X100 Z200          ;
N0008   G61                   ; return to (X35,Z80)
N0009   M2                    ; End of program
```

3.15 Return to Reference Position (G28)

The reference position is a fixed position on a machine tool to which the tool can easily be moved by the reference position return function.For example, the reference position is used as a position at which tools are automatically changed.

If there is machine zero point(hardware switch),it is also reference point; when user set float zero point(software switch) as home and system will take float zero point as reference point.

Reference Position offset of axes to home can be set by parameters.

X-axis offset: P32 in Axis parameters , unit: 0.01mm.

Z-axis offset: P33 in Axis parameters, unit: 0.01mm.

C-axis offset: P116 in Axis parameters, unit: 0.01mm.

A-axis offset: P214 in Axis parameters, unit: 0.01mm.

User could return reference point in Manual (Press "Return" key) or in Auto (Use G28 instruction).

Tools are automatically moved to the reference position via an intermediate position along a specified axis.

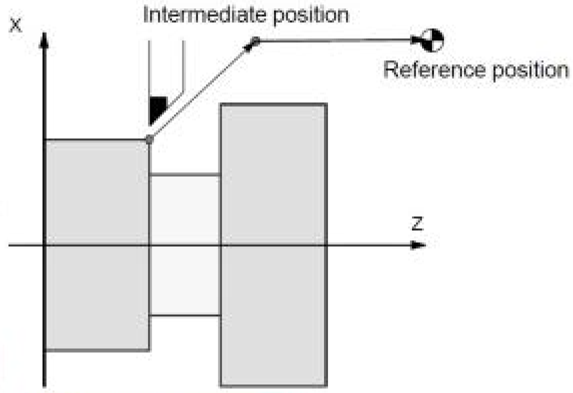


Fig3.15.1 Reference Position Return

Format: G28 IP_;

IP_:Command specifying the intermediate position of all feeding axes (Absolute/incremental command)

G281 ; Only X-axis return to reference position

G282 ; Only Y(C)-axis return to reference position,

G283 ; Only Z-axis return to reference position,

G284 ; Only A-axis return to reference position,

Note: 1. When use G282 instruction, Y(C)-axis return reference position,the system only detect Y0 (PIN2 of CN3 plug),don't detect Z0 signal of encoder.

2. When use M800, C-axis return to zero position(Z0) of encoder.please check details of M800 in M code chapter.

3. When the G28 insruction is specified when manual return to the reference position has not been performed after the power has been turned on, the movement from the intermediate point is the same as in manual return to the reference position.In this case, the tool moves in the direction for reference position return specified by P28 in Axis parameter. Therefore the specified intermediate position must be a position to which reference position return is possible.

4. When use G28 instruction to return to reference position, if just specify intermediate position of some axes, which can return to reference position, the others axes that don't be specified cannot return to reference position.

5. Before run these codes, reference position must be set well.

6. After return to reference positoin, system will cancel tool compensation automatically.

3.16 Coordinate System

By teaching the CNC a desired tool position, the tool can be moved to the position. Such a tool position is represented by coordinates in a coordinate system. Coordinates are specified using program axes.

When two program axes, the X – axis and Z – axis, are used, coordinates are specified as follows:

X_Z_

This command is referred to as a dimension word.

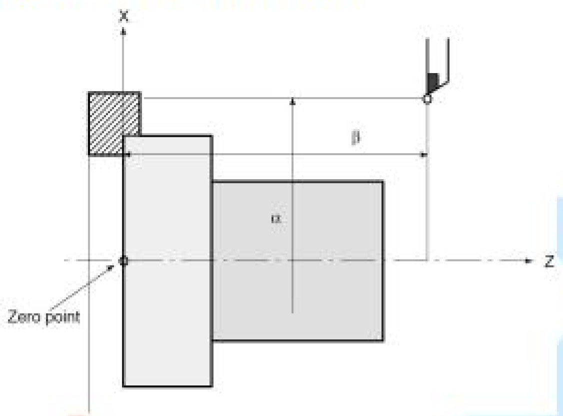


Fig3.16.1 Tool Position Specified by XαZβ

Coordinates are specified in one of following three coordinate systems:

(1) Machine coordinate system

(2) Workpiece coordinate system

(3) Local coordinate system

The number of the axes of a coordinate system varies from one machine to another. So, in this manual, a dimension word is represented as **IP_**.

3.16.1 Machine Coordiante System (G53)

The point that is specific to a machine and serves as the reference of the machine is referred to as the machine zero point. A machine tool builder sets a machine zero point for each machine.

A coordinate system with a machine zero point set as its origin is referred to as a machine coordinate system.

A machine coordinate system is set by performing manual reference position return after power–on (see Chapter5.5.3). A machine coordinate system, once set, remains unchanged until the power is turned off.

G53 IP_ ;

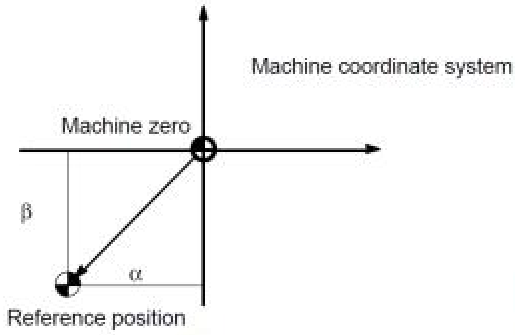
IP_ : absolute dimension word

When a position has been specified as a set of machine coordinates, the tool moves to that position by means of rapid traverse. G53, used for selecting the machine coordinate system, is a one–shot G code. Any commands based on the selected machine coordinate system are thus effective only in the block containing G53. The G53 command must be specified using absolute values. If incremental values are specified, the G53 command is ignored. When the tool is to be moved to a machine–specific position such as a tool change position, program the movement in a machine coordinate system based on G53.

Note: 1. When the G53 command is specified, cancel the tool nose radius compensation and tool offset.

2. Since the machine coordinate system must be set before the G53 command is specified, at least one manual reference position return or automatic reference position return by the G28 command must be performed after the power is turned on. This is not necessary when an absolute–position detector is attached.

When manual reference position return is performed after power–on, a machine coordinate system is set so that the reference position is at the coordinate values of (α, β) set by P32 & P33 in Axis parameter .



3.16.2 Workpiece Coordinate System

A coordinate system used for machining a workpiece is referred to as a workpiece coordinate system. A workpiece coordinate system is to be set with the NC beforehand **(setting a workpiece coordinate system)**.

A machining program sets a workpiece coordinate system **(selecting a workpiece coordinate system)**.

A set workpiece coordinate system can be changed by shifting its origin **(changing a workpiece coordinate system)**.

A workpiece coordinate system can be set using one of three methods:

(1)Method using G50

A workpiece coordinate system is set by specifying a value after G50 in the program.

(2) Automatic setting

A workpiece coordinate system is automatically set when manual reference position return is performed .

(3) Mothod of using G54 to G59

Make settings on the MDI panel to preset six workpiece coordinate systems. Then, use program commands G54 toG59 to select which workpiece coordinate system to use.

When an absolute command is used, a workpiece coordinate system must be established in any of the ways described above.

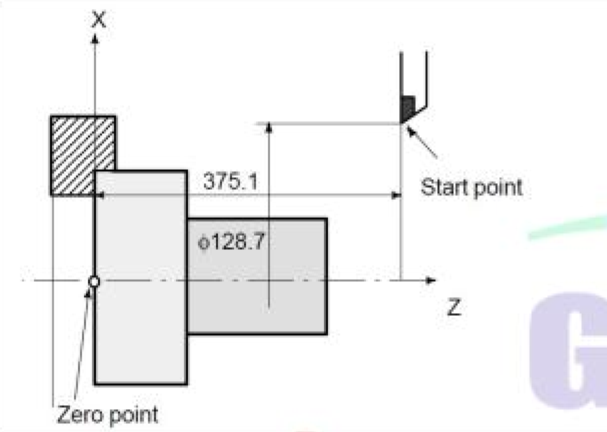
3.16.3 Setting a Workpiece Coordinate System(G50)

Format: G50 IP_ ;

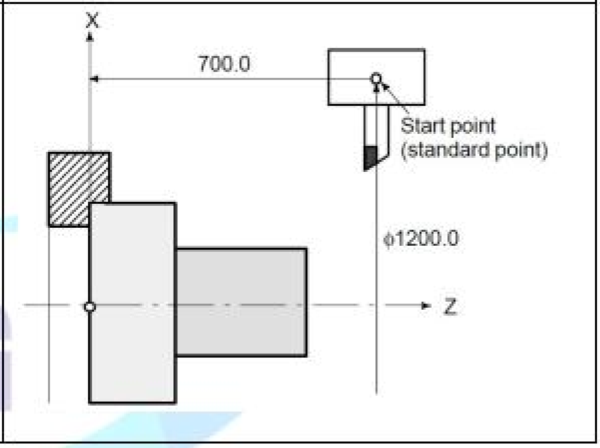
A workpiece coordinate system is set so that a point on the tool, such as the tool tip, is at specified coordinates. If IP is an incremental command value, the work coordinate system is defined so that the current tool position coincides with the result of adding the specified incremental value to the coordinates of the previous tool position. If a coordinate system is set using G50 during offset, a coordinate system in which the position before offset matches the position specified in G50 is set. And after workpiece coordinate system set well, absolute coordiante position of following commands are based on this coordinate system. Setting solution of G50 workpiece coordiante system(See Chapter of coordiante system set)

Note: In the status of compensation, if use G50 to set the workpiece coordinate system, position before make tool comepensation is based on G50 workpiece coordinate system. Usually cancel the tool compensation first of all before starting program. After return to reference positoin, system will cancel tool compensation automatically.

Example1:
Setting the coordinate system by the G50X128.7Z375.1; command (Diameter designation)



Example2: Base Point
Setting the coordinate system by the G50X1200.0Z700.0; command (Diameter designation)



3.16.4 Selecting a Workpiece Coordinate System

The user can choose from set workpiece coordinate systems as described below.

(1)G50 or automatic workpiece coordinate system setting

Once a workpiece coordinate system is selected, absolute commands work with the workpiece coordinate system.

(2)Choosing from six workpiece coordinate systems set in the MDI

By specifying a G code from G54 to G59, one of the workpiece coordinate systems 1 to 6 can be selected.

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

Workpiece coordinate system 1 to 6 are established after reference position return after the power is turned on.

G55 G00 X100.0 Z40.0 ;

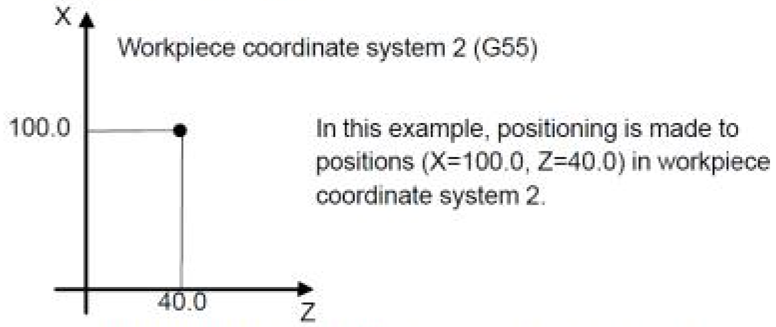


Fig3.16.2 Example of Workpiece Coordinate System

3.16.5 Changing Workpiece Coordinate System

The six workpiece coordinate systems specified with G54 to G59 can be changed by changing an external workpiece zero point offset value or workpiece zero point offset value.

Three methods are available to change an external workpiece zero point offset value or workpiece zero point offset value.

(1) Inputting from the MDI panel (see III－11.4.9)

(2) Programming by G50

(3) Using the external data input function

An external workpiece origin offset can be changed by using a signal input to the CNC,also alter coordinate system in Coor parameter.

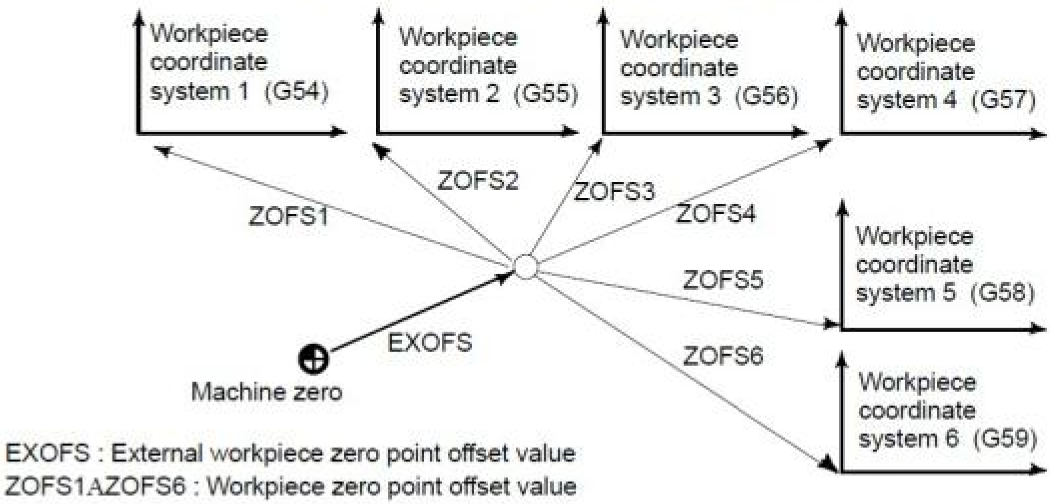


Fig3.16.3 Workpiece Zero Point Offset

Format: Changing by G50 : G50 IP_ ;

By specifying G50IP_;, a workpiece coordinate system (selected with a code from G54 to G59) is shifted to set a new workpiece coordinate system so that the current tool position matches the specified coordinates (IP_).

If IP is an incremental command value, the work coordinate system is defined so that the current tool position coincides with the result of adding the specified incremental value to the coordinates of the previous tool position. (Coordinate system shift)

Then, the amount of coordinate system shift is added to all the workpiece zero point offset values. This means that all the workpiece coordinate systems are shifted by the same amount.

Examples:

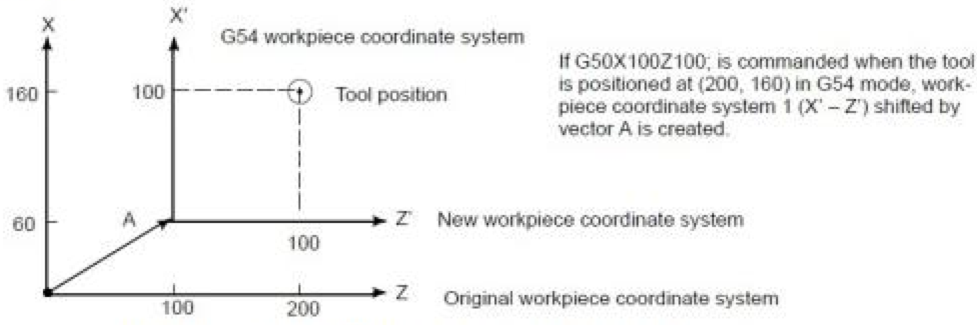


Fig3.16.4 Example1 of Workpiece Coordiante System

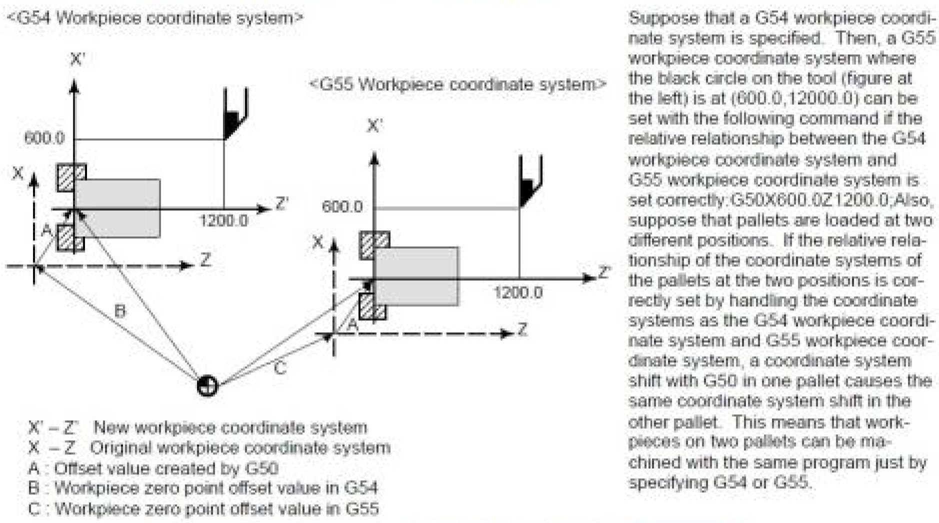
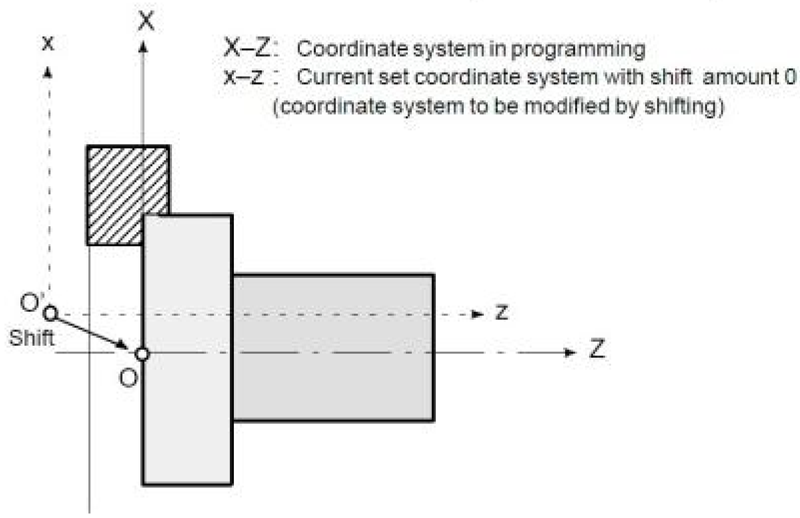


Fig3.16.5 Example2 of Workpiece Coordinate System

3.16.6 Workpiece Coordinate System Shift

When the coordinate system actually set by the G50 command or the automatic system setting deviates from the programmed work system, the set coordinate system can be shifted.

Set the desired shift amount in the work coordinate system shift memory.



Set the shift amount from O' to O in the work coordinate system shift memory.

Fig3.16.6 Workpiece Coordinate System Shift

3.16.7 Local Coordinate System

When a program is created in a workpiece coordinate system, a child workpiece coordinate system may be set for easier programming. Such a child coordinate system is referred to as a local coordinate system.

Format: G52 IP_; Setting the local coordinate system

G52 IP 0 ; Canceling of the local coordinate system

IP_: Origin of the local coordinate system

By specifying G52IP_;, a local coordinate system can be set in all the workpiece coordinate systems (G54 to G59). The origin of each local coordinate system is set at the position specified by IP_ in the workpiece coordinate system.

Once a local coordinate system is established, the coordinates in the local coordinate system are used in an axis shift command. The local coordinate system can be changed by specifying the G52 command with the zero point of a new local coordinate system in the workpiece coordinate system.

To cancel the local coordinate system and specify the coordinate value in the workpiece coordinate system, match the zero point of the local coordinate system with that of the workpiece coordinate system.

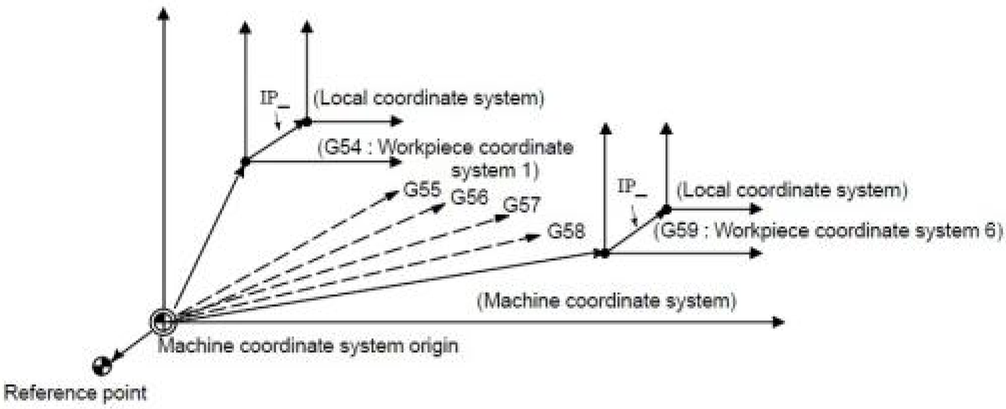


Fig3.16.7 Setting the Local Coordinate System

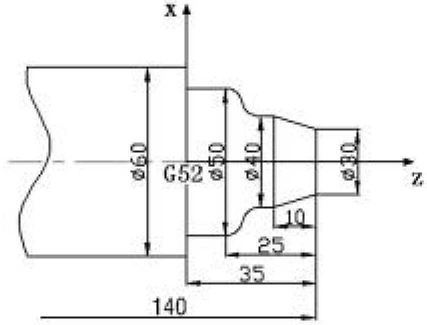
Note:1. The local coordinate system setting does not change the workpiece and machine coordinate systems.

2. When G50 is used to define a work coordinate system, if coordinates are not specified for all axes of a local coordinate system, the local coordinate system remains unchanged. If coordinates are specified for any axis of a local coordinate system, the local coordinate system is canceled.

3. G52 cancels the offset temporarily in tool nose radius compensation.

4. Command a move command immediately after the G52 block in the absolute mode.

Example1:



```
N1 G00 X60 Z20
N2 G52 X0 Z-236
N3 T0101
N4 M03 S800 M08
N5 G01 Z35 F100
N6 X-1
N7 X70
N8 G71 U2 R1
N8 G71 P10 Q15 U0.5 W0.5 F150
N10 G01 X30
N11 X40 Z25
N12 Z20
N13 G02 X50 Z15 R5
N14 G03 X60 Z10 R5
N15 G01 Z0
N16 G00 X70
N17 G52 X0 Z0
N18 M05
N19 M30
```

Example2:Processing 5pcs same workpieces on one workpiece.

```
G22 L5
X50 Z60; workpiece
......
G52 W-50; each Z-axis relative offset 50mm
G800
G52 X0 Z0; cancel the local coordinate system
G0 X100 Z100
M02
```


3.17 Constant Surface Speed Control (G96/G97)

Specify the surface speed (relative speed between the tool and workpiece) following S. The spindle is rotated so that the surface speed is constant regardless of the position of the tool.

Format: G96 SOOOO ; Constant surface speed control command

↑ Surface Speed (m/min or feet/min)

G97 SOOOO ; Constant surface speed control cancel command

↑ Spindle Speed (min^{-1})

G50 S_ ; The maximum spindle speed (min^{-1}) follows S.

G96 (constant surface speed control command) is a modal G code. After a G96 command is specified, the program enters the constant surface speed control mode (G96 mode) and specified S values are assumed as a surface speed. A G96 command must specify the axis along which constant surface speed control is applied. A G97 command cancels the G96 mode.When constant surface speed control is applied, a spindle speed higher than the value specified in G50S_; (maximum spindle speed) is clamped at the maximum spindle speed. When the power is turned on, the maximum spindle speed is not yet set and the speed is not clamped. S (surface speed) commands in the G96 mode are assumed as S = 0 (the surface speed is 0) until M03 (rotating the spindle in the positive direction) or M04 (rotating the spindle in the negative direction) appears in the program.

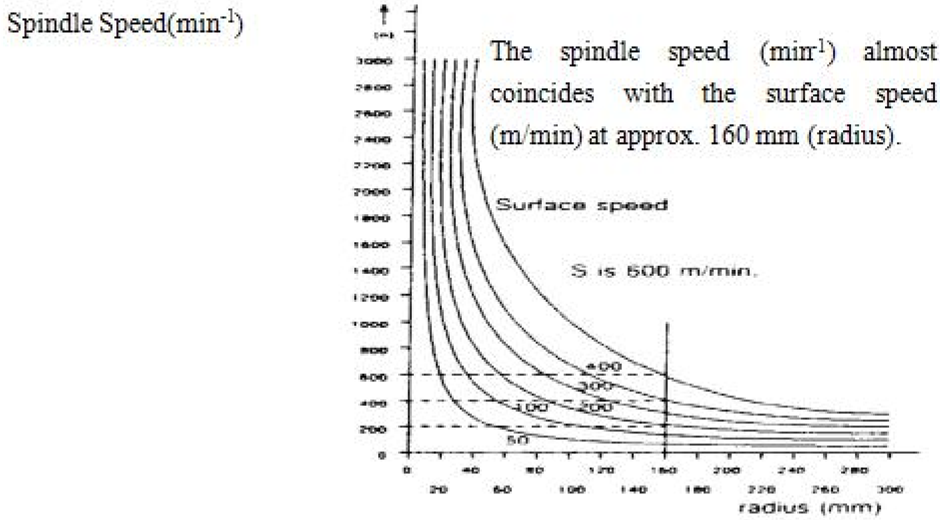


Fig3.17.1 Relation between workpiece radius, spindle speed and surface speed

To execute the constant surface speed control, it is necessary to set a workpiece coordinate system , Z axis, (axis to which the constant surface speed control applies) becomes zero.

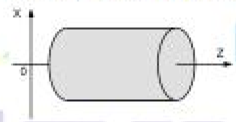
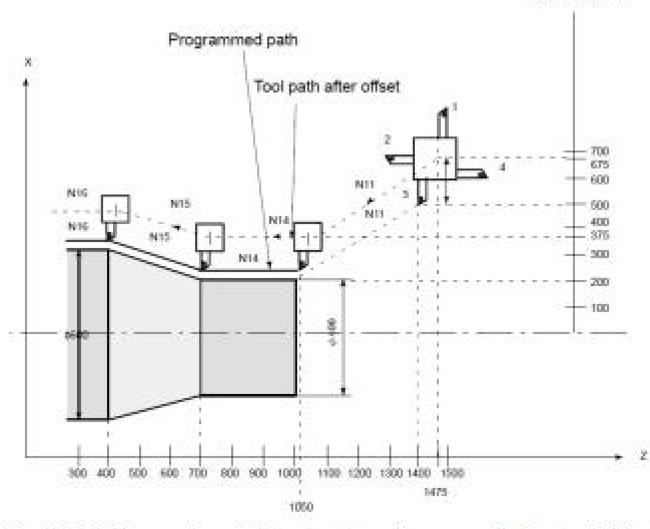


Fig3.17.2 Example of the workpiece coordinate system for constant surface speed control

The constant surface speed control is also effective during threading. Accordingly, it is recommended that the constant surface speed control be invalidated with G97 command before starting the scroll threading and taper threading, because the response problem in the servo system may not be considered when the spindle speed changes.

In a rapid traverse block specified by G00, the constant surface speed control is not made by calculating the surface speed to a transient change of the tool position, but is made by calculating the surface speed based on the position at the end point of the rapid traverse block, on the condition that cutting is not executed at rapid traverse.



Example1:

N8 G00 X1000.0Z1400.0 ;
N9 T33;
N11 X400.0Z1050.0;
N12 G50S3000 ; (Designation of max. spindle speed)
N13 G96S200 ; (Surface speed 200m/min)
N14 G01 Z 700.0F1000 ;
N15 X600.0Z 400.0;
N16 Z … ;

Fig3.17.3 Example of Constant surface speed Control for G00

The CNC calculates the spindle speed which is proportional to the specified surface speed at the position of the programmed coordinate value on the X axis. This is not the value calculated according to the X axis coordinate after offset when offset is valid. At the end point N15 in the example above, the speed at 600 dia. (Which is not the turret center but the tool nose) is 200 m/min. If X axis coordinate value is negative, the CNC uses the absolute value.

Press "absolute coordinate" to change the spindle speed in the status of constant linear speed G96.

Note: 1. use constant linear speed, the spindle must be able to change speed automatically. (such as: Servo Spindle,Inverter Spindle).Max SP-speed is set by parameter.

2. Min SP-speed in condition of G96 can be set by P35 in Speed parameter.

3. Spindle override don't work when processing with constant surface speed.

Example2:

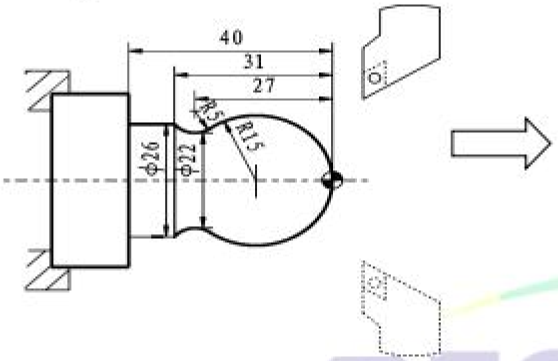


Fig3.17.4 Example2

```
N1 T0102 X40 Z5
N2 M03 S400
N3 G96 S80
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 G97 S300
N11 M30
```

3.18 Cutting Feed (G98,G99)

Feedrate of linear interpolation (G01), circular interpolation (G02, G03), etc. are commanded with numbers after the F code.

In cutting feed, the next block is executed so that the feedrate change from the previous block is minimized.

Two modes of specification are available:

1. Feed per minute (G98)

After F, specify the amount of feed of the tool per minute.

2. Feed per revolution (G99)

After F, specify the amount of feed of the tool per spindle revolution.

Format: G98 ; Feed per minute instruction

F_; Feedrate command (mm/min or inch/min)

G99 ; Feed per revolution instruction

F_ ; Feedrate command (mm/rev or inch/rev)

Cutting feed is controlled so that the tangential feedrate is always set at a specified feedrate.

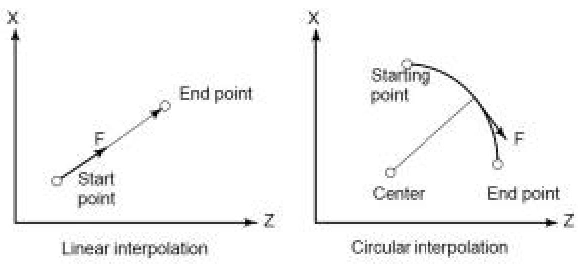


Fig3.18.1 Tangential feedrate (F)

After specifying G98 (in the feed per minute mode), the amount of feed of the tool per minute is to be directly specified by setting a number after F. G98 is a modal code. Once a G98 is specified, it is valid until G99 (feed per revolution) is specified. At power – on, the feed per revolution mode is set.

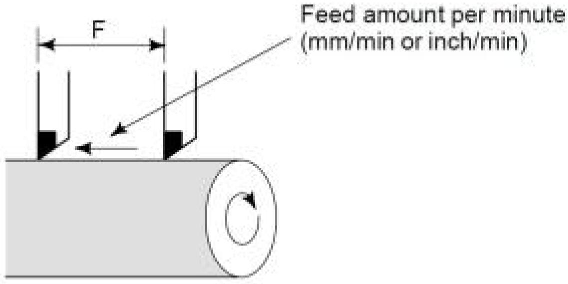


Fig3.18.2 Feed Per Minute

After specifying G99 (in the feed per revolution mode), the amount of feed of the tool per spindle revolution is to be directly specified by setting a number after F. G99 is a modal code. Once a G99 is specified, it is valid until G98 (feed per minute) is specified.

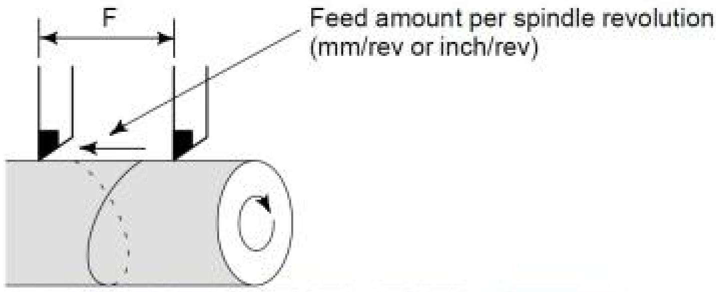


Fig3.18.2 Feed Per Revolution

Note: 1. When using G99 instruction, Spindle must configured with encoder,otherwise system will stay on.

2.When the speed of the spindle is low, feedrate fluctuation may occur. The slower the spindle rotates, the more frequently feedrate fluctuation occurs.

3.19 Pole Coordinate Interpolation (G15/G16)

Polar coordinate interpolation is a function that exercises contour control in converting a command programmed in a Cartesian coordinate system to the movement of a linear axis (movement of a tool) and the movement of a rotary axis (rotation of a workpiece). This method is useful in cutting a front surface and grinding a cam shaft on a lathe.

Polar coordinates input directive allows radius and angle in polar coordinates, the angle of the positive Z direction is counterclockwise turned, while the negative direction is a clockwise turn. Radius with absolute command value instruction (Z), the angle with absolute command (X).

G16 IP_ ; Starts polar coordinate interpolation mode (enables polar coordinate interpolation)

... ... } Specify linear or circular interpolation using coordinates in a Cartesian coordinate system consisting of a linear axis and rotary axis (virtual axis).

G15 ; Polar coordinate interpolation mode is cancelled (for not performing polar coordinate interpolation)

Explanations:

1. Z specify zero as the origin of the polar coordinate system from the point of measuring the radius of the workpiece coordinate system.

2. IP_: Plane selection & home of polar coordinate system

Z_ : Radius of polar coordinate system

X_ : Angle of polar coordinate system

3. Z set to zero as the origin of the polar coordinate system workpiece coordinate system:

By absolute programming instructions a specified radius (distance between the zero point and programming). Zero set workpiece coordinate system for the origin of the polar coordinate system

G16 starts the polar coordinate interpolation mode and selects a polar coordinate interpolation plane(Fig3.19.1).Polar coordinate interpolation is performed on this plane.

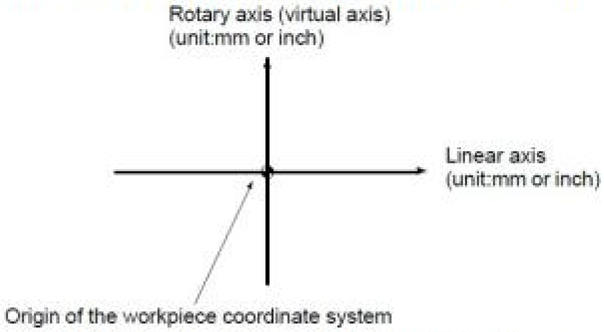


Fig3.19.1 Polar Coordinate Interpolation Plane

When the power is turned on or the system is reset, polar coordinate interpolation is canceled (G15).

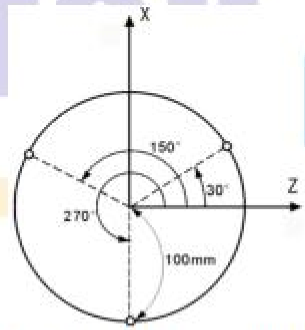


Fig3.19.2 Example of Polar Coordiante System

Example: N1 G16 X0 Z0 ; setup polar coordinate

N2 G01 X30.0 Z100.0 F200.0

N3 X150.0

N4 X270.0

N5 G15 ; cancel polar coordiante system

N6 M02

3.20 Absolute and Incremental Programming (G990,G991)

There are two ways to command travels of the tool; the absolute command, and the incremental command. In the absolute command, coordinate value of the end position is programmed; in the incremental command, move distance of the position itself is programmed. G90 and G91 are used to command absolute or incremental command, respectively.

Absolute programming or incremental programming is used depending on the command used. See following tables.

G code System	A	B or C
Command method	Address word	G990,G991

G code system A

	Absolute command	**Increment command**
X axis move command	X	U
Z axis move command	Z	W
C axis move command	C	V

G code system B or C

Absolute command	**G990 IP_ ;**
Incremental command	**G991 IP_ ;**

Example: Tool movement from point P to point Q (diameter programming is used for the X-axis)

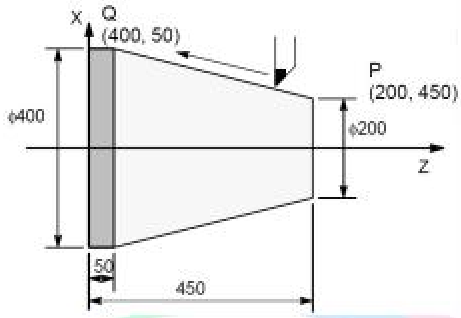


Fig3.20.1 Example of Absolute and Incremental Programming

	G code system A	G code system B or C
Absolute command	X400.0 Z50.0	G90 X400.0 Z50.0
Incremental command	U200.0 W-400.0	G91 X200.0 Z-400.0

Note: 1. Absolute and incremental commands can be used together in a block. In the above example, the following command can be specified : X400.0 W–400.0 ;

2. When both X and U or W and Z are used together in a block, the one specified later is valid.

3. Incremental commands cannot be used when names of the axes are A and B during G code system A is selected.

3.21 Inch/Metric Conversion (G20/G21)

Either inch or metric input can be selected by G code.

Format: G20 ; Inch input

G21 ; mm input

This G code must be specified in an independent block before setting the coordinate system at the beginning of the program. After the G code for inch/metric conversion is specified, the unit of input data is switched to the least inch or metric input increment of increment system IS－B or IS－C. The unit of data input for degrees remains unchanged.

The unit systems for the following values are changed after inch/metric conversion:

- **Feedrate commanded by F code**
- **Positional command**
- **Work zero point offset value**
- **Tool compensation value**
- **Unit of scale for manual pulse generator**
- **Movement distance in incremental feed**
- **Some parameters**

When the power is turned on, the G code is the same as that held before the power was turned off.

Note:

1. G20 and G21 must not be switched during a program.

2. Movement from the intermediate point is the same as that for manual reference position return. The direction in which the tool moves from the intermediate point is the same as the reference position return direction.

3. When the least input increment and the least command increment systems are different, the maximum error is half of the least command increment. This error is not accumulated.

4. The inch and metric input can also be switched using setting of data setting.

3.22 Dwell (G04)

By specifying a dwell, the execution of the next block is delayed by the specified time.

Format: G04 P_ ; or G04 X_; or G04 U_;

P_: Specify a time (decimal point not permitted) , unit: ms (millisecond)

X_: Specify a time (decimal point permitted) , unit: s (second)

U_: Specify a time (decimal point permitted) , unit: s (second)

Example 1:

G04 X1; delay 1s.

G04 P1000; delay 1s.

G04 U1; delay 1s.

Special application:G04 can be accurate stop instruction, such as processing corner kinds of workpiece, it appears over cutting sometimes, if use G04 instruction around the corner, it will clear the over cutting.

Example 2:

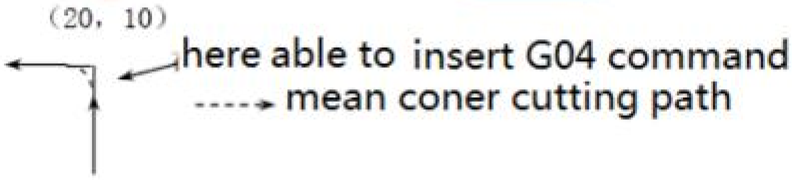


Fig3.22.1 Application of Dwell(G04)

Program: ……

N150 G01 X20 Z10 F100;

N160 G04 P150; （Clear the over cutting）

N170 G01 W-10;

……

Note: Set P21 in User parameter, also by setting intervals between blocks of G01/G02/G03 to clear the over cutting.

3.23 Positioning/Continous Path Processing(G60/G64)

According to process requirements, user can specify the connection way between program block by G60/G64.

Format: G60 ; Acturate Positioning Processing

G64 ; Continous Path Processing

Both G60 and G64 are modal instructions.

3.24 Workpiece Position and Move Command

In tool nose radius compensation, the position of the workpiece with respect to the tool must be specified.G40, G41, and, G42 are modal.

G code	Workpiece Position	Tool Path
G40	(Cancel)	Moving along the programmed path
G41	Right side	Moving on the left side the programmed path
G42	Left side	Moving on the right side the programmed path

The tool is offset to the opposite side of the workpiece.

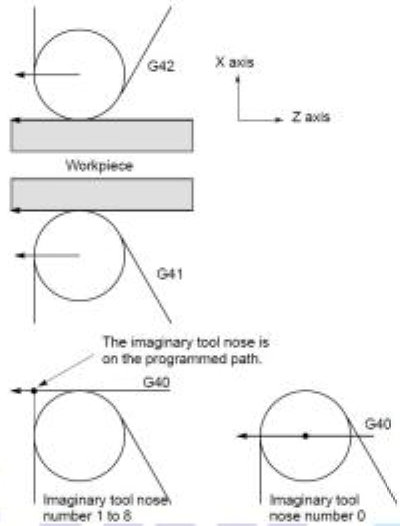


Fig2.24.1 a Workpiece Position of G41/G42

The workpiece position can be changed by setting the coordinate system as shown below.

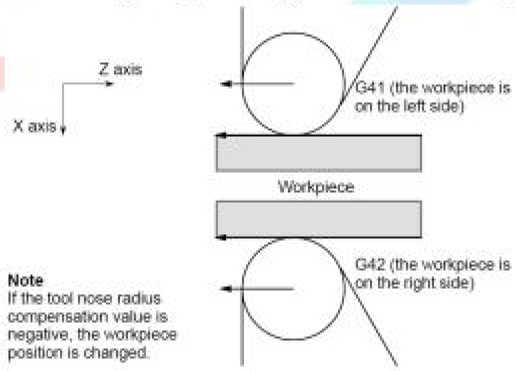


Fig3.24.2 b Workpiece Position of G41/G42

When the tool is moving, the tool nose maintains contact with the workpiece.

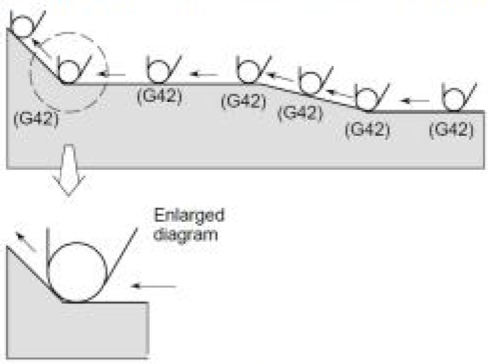


Fig3.24.3 Tool movement when the workpiece position does not change

The workpiece position against the toll changes at the corner of the programmed path as shown in the following figure.

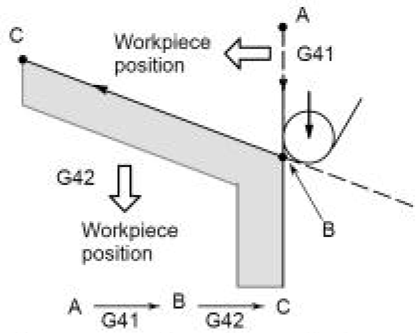


Fig3.24.4 Tool movement when the workpiece position changes

Although the workpiece does not exist on the right side of the programmed path in the above case. the existence of the workpiece is assumed in the movement from A to B. The workpiece position must not be changed in the block next to the start－up block. In the above example, if the block specifying motion from A to B were the start－up block, the tool path would not be the same as the one shown.

The block in which the mode changes to G41 or G42 from G40 is called the start－up block.

G40 _ ;

G41 _ ; (Start-up block)

Transient tool movements for offset are performed in the start–up block. In the block after the start-up block, the tool nose center is positioned Vertically to the programmed path of that block at the start position.

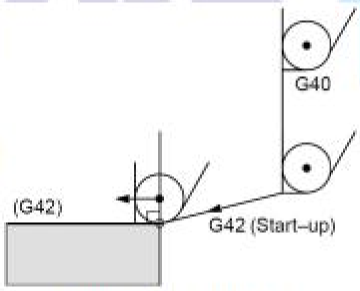


Fig3.24.5 Start-Up

The block in which the mode changes to G40 from G41 or G42 is called the offset cancel block.

G41 _ ;

G40 _ ; (Offset cancel block)

The tool nose center moves to a position vertical to the programmed path in the block before the cancel block. The tool is positioned at the end position in the offset cancel block (G40) as shown below.

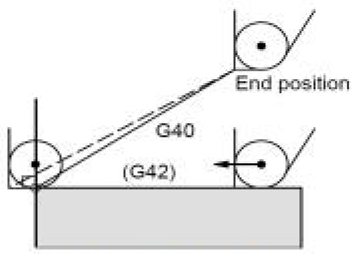


Fig3.24.6 Offset Cancel

When is specified again in G41/G42 mode , the tool nose center is positioned vertical to the programmed path of the preceding block at the end position of the preceding block.

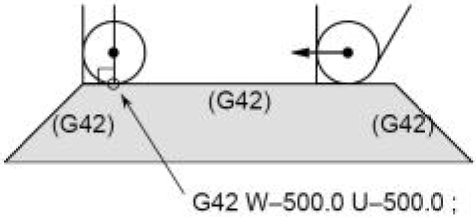


Fig3.24.7 Specification of G41/G42 in G41/G42 mode

In the block that first specifies G41/G42, the above positioning of the tool nose center is not performed.

When you wish to retract the tool in the direction specified by X(U) and Z(W) cancelling the tool nose radius compensation at the end of machining the first block in the figure below, specify the following :

G40 X(U) _ Z(W) _ I _ K _ ;

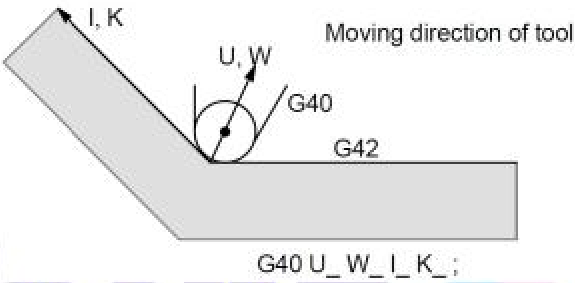


Fig3.24.8 Tool movement when the moving direction of the tool in a block

The workpiece position specified by addresses I and K is the same as that in the preceding block. If I and/or K is specified with G40 in the cancel mode, the I and/or K is ignored.

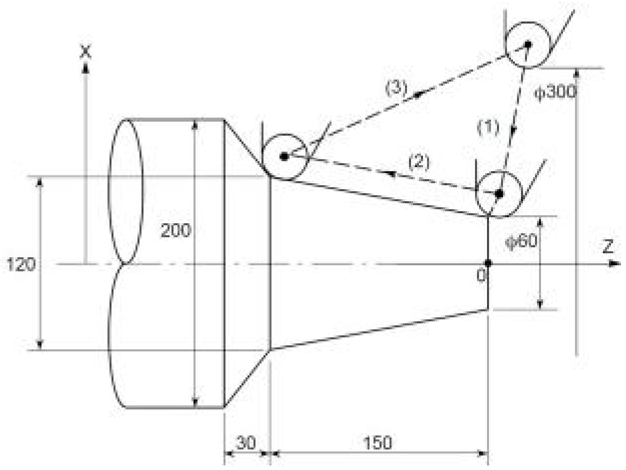
G40 X_ Z_ I_ K_;	Tool nose radius compensation
G40 G02 X_ Z_ I_ K_;	Circular interpolation

G40 G01 X_ Z_ ;

G40 G01 X_ Z_ I_ K_ ; Offset cancel mode (I and k are ineffective.)

The numeral s followed I and K should always be specified as radius values.

Example



(G40 mode)
1.G42 G00 X60.0 ;
2.G01 X120.0 W–150.0 F10 ;
3.G40 G00 X300.0 W150.0 I40.0 K–30.0 ;

Fig3.24.9 Example of Tool Nose Radius Compensation

Tool nose radius compensation with G90 (outer diameter/internal diameter cutting cycle) or G94 (end face turning cycle) is as follows, :

1. Motion for imaginary tool nose numbers

For each path in the cycle, the tool nose center path is generally parallel to the programmed path.

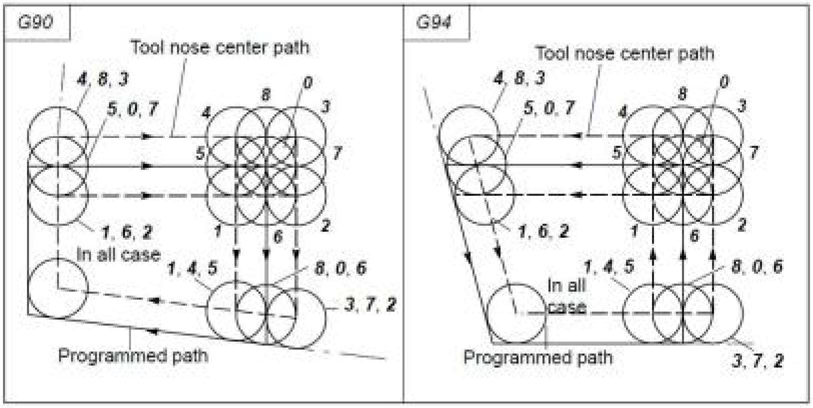


Fig3.24.10 Motion for imaginary tool nose numbers

2. Direction of the offset

The offset direction is indicated in the figure below regardless of the G41/G42 mode.

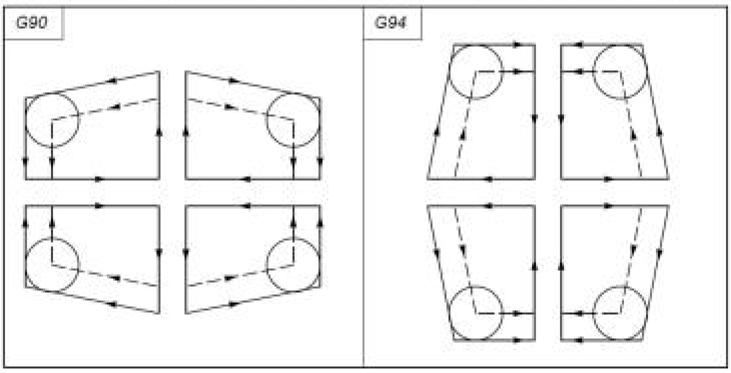
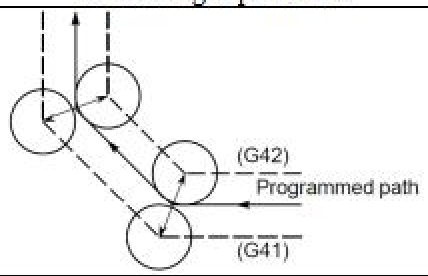
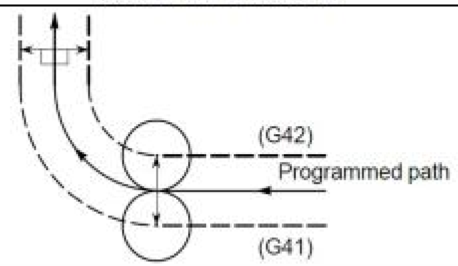


Fig3.24.11 Direction of the offset

Movement after compensation is shown below.

Tool nose radius compensation when chamfering is performed	Tool nose radius compensation when a corner arc is inserted
	

3.25 Details of Tool Nose Radius Compensatation (G40/G41/G42)

The tool nose radius center offset vector is a two dimensional vector equal to the offset value specified in a T code, and the is calculated in the CNC.

Its dimension changes block by block according to tool movement. This offset vector (simply called vector herein after) is internally crated by the control unit as required for proper offsetting and to calculate a tool path with exact offset (by tool nose radius) from the programmed path.

This vector is deleted by resetting.

The vector always accompanies the tool as the tool advances.

Proper understanding of vector is essential to accurate programming.

Read the description below on how vectors are created carefully.

G40, G41 or G42 is used to delete or generate vectors.

These codes are used together with G00, G02, or G32 to specify a mode for tool motion (Offsetting).

G code	Function	Workpiece Position
G40	Tool nose radius compensation cancel	Neither
G41	Left offset along tool path	Left
G42	Right offset along tool path	Right

G41 and G42 specify an off mode, while G40 specifies cancellation of the offset.

The system enters the cancel mode immediately after the power is turned on, when the RESET button on the MDI is pushed or a program is forced to end by executing M02 or M30. (the system may not enter the cancel mode depending on the machine tool.) In the cancel mode, the vector is set to zero, and the path of the center of tool nose coincides with the programmed, path. A program must end in cancel mode. If it ends in the offset mode, the tool cannot be positioned at the end point, and the tool stops at a location the vector length away from the end point.

Note: 1. G40/G41/G42 are modal codes, can cancel each other.

2. G41/G42 without parameters, the compensation number (on behalf of the tool tip radius compensation corresponding values) specified by the T code. Its tip arc offset number and tool offset number corresponding compensation.Establish and canceled .

3. Tool nose radius compensation with G00 or G01 instruction only, not the G02 or G03.Nose radius compensation tool compensation interface "radius compensation", the definition of the turning radius; imaginary tool nose number defines the direction of the tip.

The imaginary tool nose tip number defines the positional relationship between the tool and cutter tip arc center point, which is from 0-9 ten directions, as shown above.

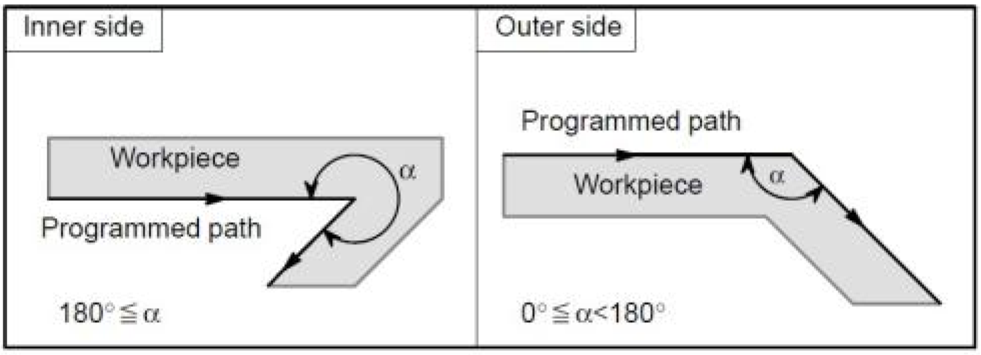
Note: For the definition of the edge position code "Posit tip" See Chapter 6.8 .

3.26 Tool Nose Radius Compensation of Offset C

C means the system calculates the tool trajectory of radius compensation according to the last program line and the next program line.

1) Inner side and outer side

When an angle of intersection created by tool paths specified with move commands for two blocks is over 180°, it is referred to as "inner side." When the angle is between 0° and 180° , it is referred to as "outer side."

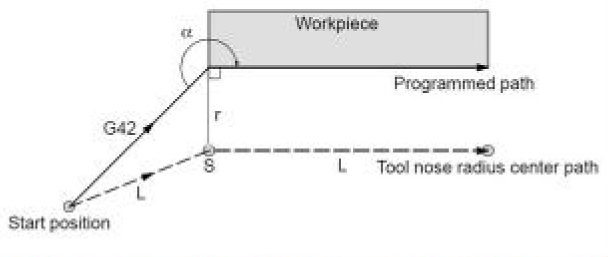


2) Tool Movement in Start-up

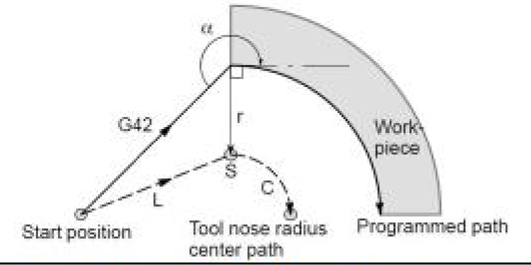
When the offset cancel mode is changed to offset mode, the tool moves as illustrated below (start-up):

(a) Tool movement around an inner side of a corner (α≥180°)

Linear → Linear

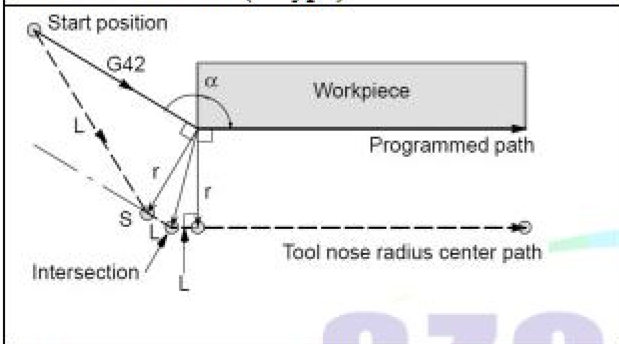


Linear → Circular

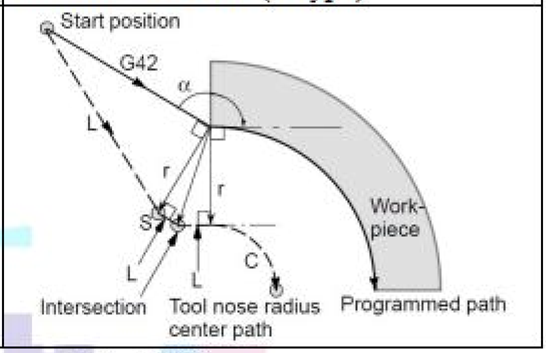


(b) Tool movement around the outside of a corner at obtuse angle (90≤α<180)

Linear → Linear (B type)

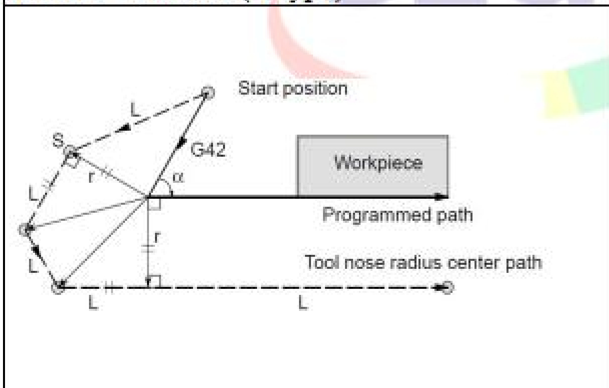


Linear → Circular (B type)

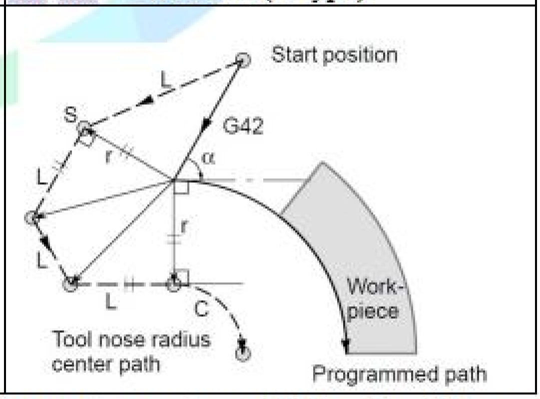


(c)Tool movement around the outside of an acute angle (α<90°)

Linear → Linear (B type)



Linear → Circular (B type)



d)Tool movement around the outside linear→linear at an acute angle less than 1 degree (α <1°)

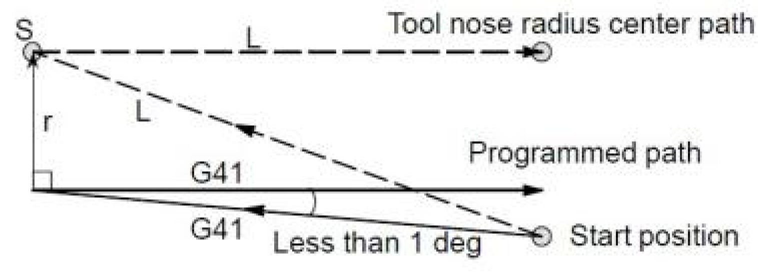
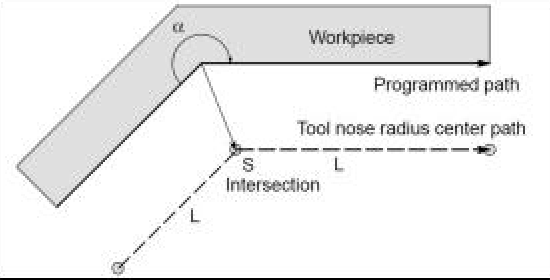
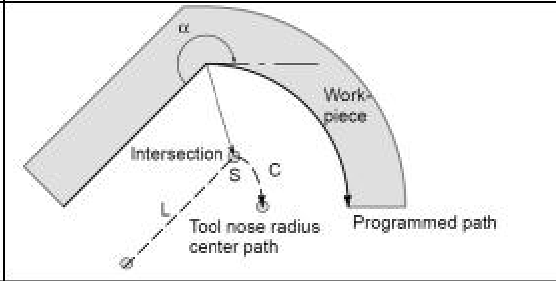
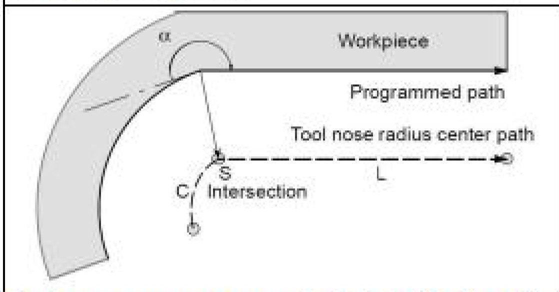
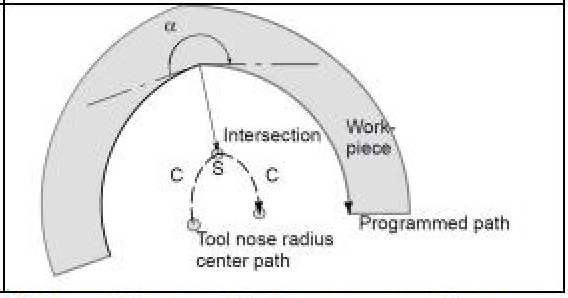


Fig3.26.1 Tool Movement at angle less than 1°

3) Tool Movement in Offset Mode

In the offset mode, the tool moves as illustrated below:

(a)Tool movement around the inside of a corner (180°≤α)

Linear → Linear	Linear → Circular
	
Circular → Linear	**Circular → Circular**
	

(b)Tool movement around the inside (α<1°) with an abnormally long vector, linear → linear

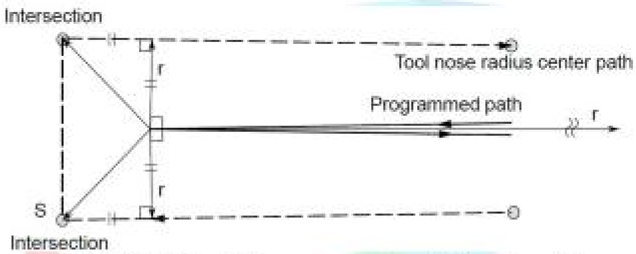
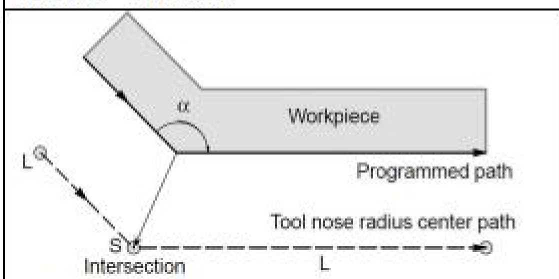
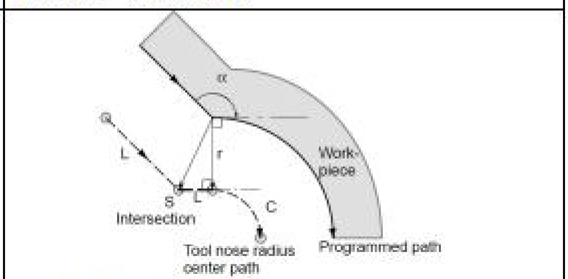
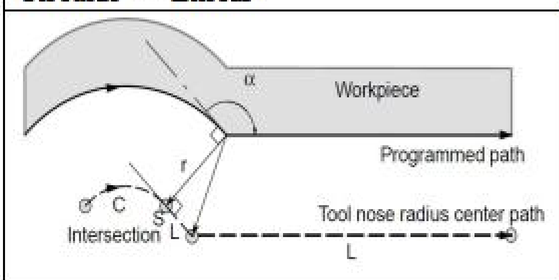
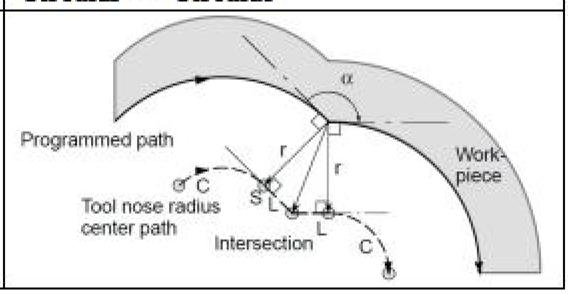


Fig3.26.2 Tool Movement at angle less than 1°

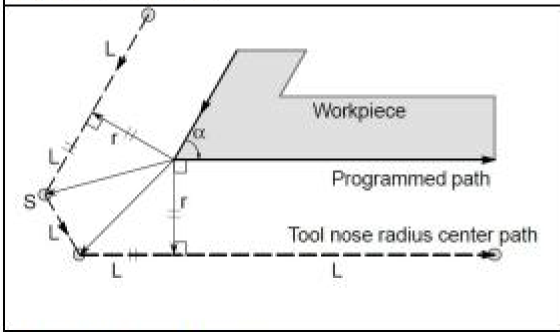
Also in case of arc to straight line, straight line to arc and arc to arc, the reader should infer in the same procedure.

(c)Tool movement around the outside corner at an obtuse angle (90≤α<180°)

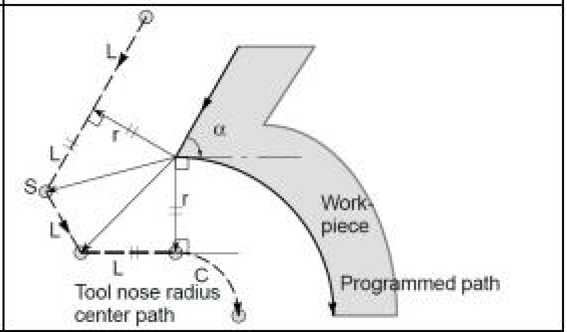
Linear → Linear	Linear → Circular
	
Circular → Linear	**Circular → Circular**
	

(d)Tool movement around the outside corner at an acute angle (α<90º)

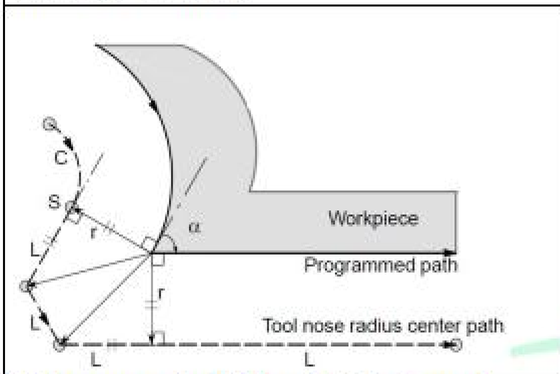
Linear → Linear



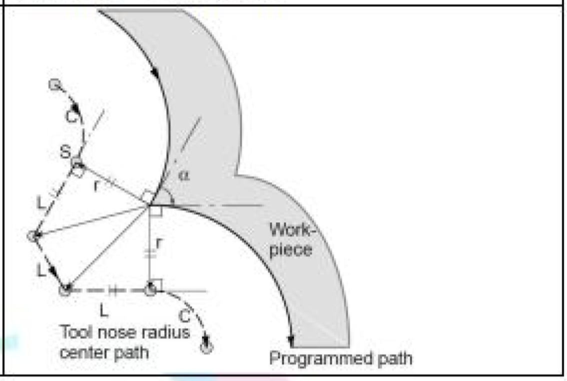
Linear → Circular



Circular → Linear



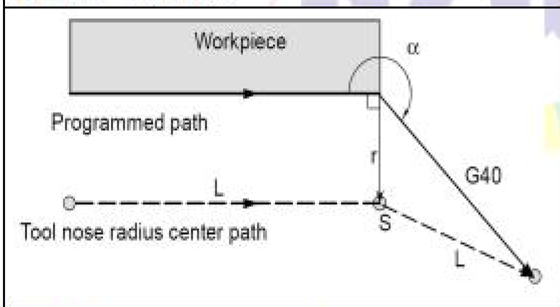
Circular → Circular



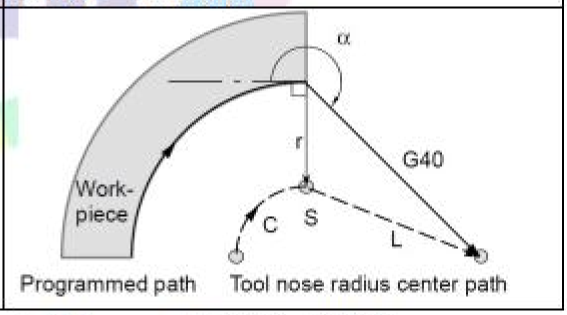
4) Tool Movement in Offset Mode Cancel

(a) Tool movement around an inside corner (180º≤α)

Linear → Linear

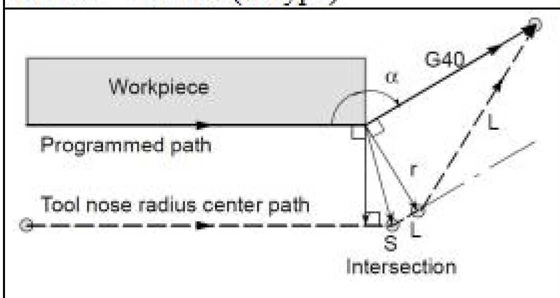


Linear → Circular

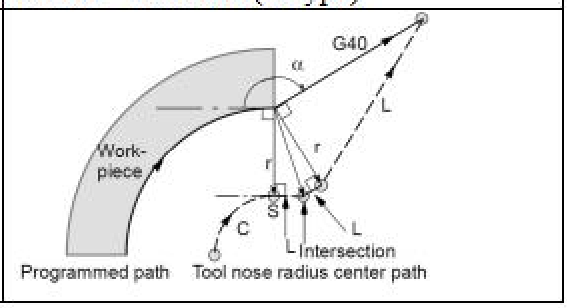


(b) Tool Movement around an outside corner at obtuse angle (90≤α<180º)

Linear → Linear(B type)

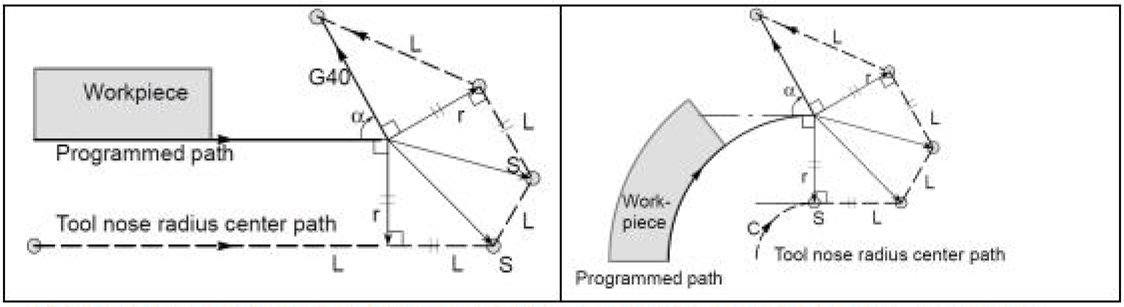


Linear → Circular(B type)



(c) Tool Movement around an outside corner at acute angle (α<90º)

Linear → Linear(B type)	Circular → Linear(B type)



(d) Tool movement around the outside linear→linear at an acute angle less than 1 degree(α＜1º)

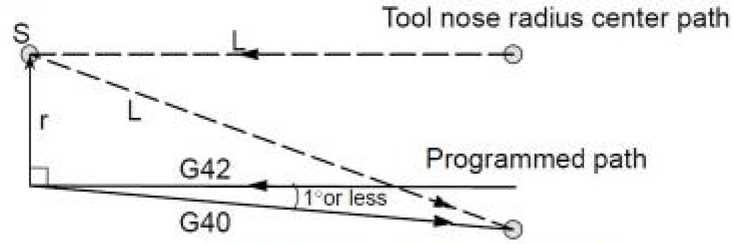


Fig3.26.3 Tool Movement at angle less than 1

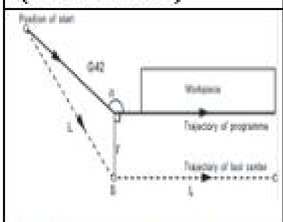
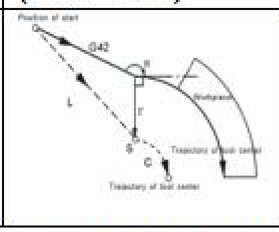
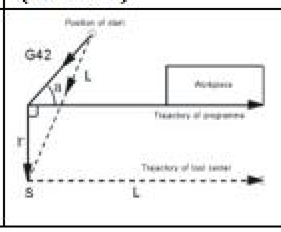
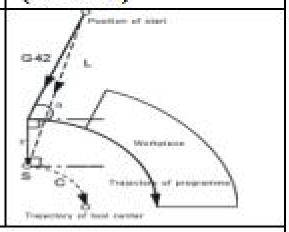
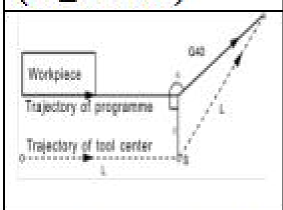
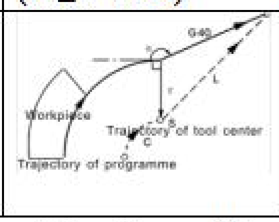
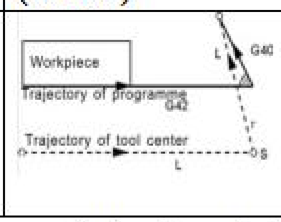
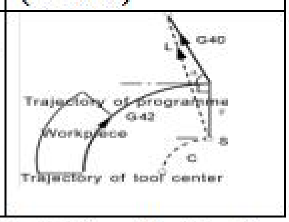
5) Change in the offset direction in the offset mode

The offset direction is decided by G codes (G41 and G42) for tool nose radius and the sign of tool nose radius compensation value as follows.

G code \ Sign of offset value	+	-
G41	Left side offset	Right side offset
G42	Right side offset	Left side offset

The offset direction can be changed in the offset mode. If the offset direction is changed in a block, a vector is generated at the intersection of the tool nose radius center path of that block and the tool nose radius center path of a preceding block. However, the change is not available in the start-up block and the block following it.

6) A Type of Tool Nose Compensation

Tool Movement in Start-up			
(90≤α＜180)	**(90≤α＜180)**	**(α＜90º)**	**(α＜90º)**
			
Tool Movement in Offset Mode Cancel			
(90≤α＜180)	**(90≤α＜180)**	**(α＜90º)**	**(α＜90º)**
			

P41 in Speed is set for A type , B type and other type for Tool Nose Radius Compensation of Offset C.

3.27 Automatical beveling (I) and smoothing(R)

Format:

Automatical Beveling

G01(G00) X(U)_ I_ ;
G01(G00) Z(W)_ } **Automatical Beveling**

G01(G00) Z(W)_ I_;
G01(G00) X(U)_ } **Automatical Beveling**

Automatical Smoothing

G01(G00) X(U)_ R_ ;
G01(G00) Z(W)_ } **Automatical Smoothing**

G01(G00) Z(W)_ R_;
G01(G00) X(U)_ } **Automatical Smoothing**

Note:

1.The address of I and R are specified with radius model. The running distance of this line and the next line must be greater than the length of beveling or radius of smoothing, otherwise the system will decrease the length of beveling or radius of smoothing to minimal running distance of this line and the next line automatically.

2. The two adjacent lines must be 90 degrees.

For example:

```
0 G54 G0 X-50 Y-50 Z20
N1 M03 S500
N2 G01 G42 D01 X0 Y0 F200
N3 G01 Z-5
N4 X100 I4                          ;Beveling4x4
N5 Y40 R6                           ;SmoothingR6
N6 X47 R5                           ;SmoothingR5
N7 Y70 I3                           ;Beveling3x3
N8 X15
N9 X0 Y40
N10 Y0
N11 G0 X-50 Y-50 G40
N12 Z50
N13 M30
```

3.28 3D Space Arc Interpolation G06

When user don't know position of circle center & radius, but know coordinate position of 3 points on arc, now user can use G06 code to processing arc, and direction is decided by middle point between starting point & end point.

Format: G06 X_ Y_ Z_ I_ J_ K_ F_

G06: Modal command

I: Increment Coordiante Value of Middle point relative to starting point in X direction
Radius Designation, with direction;

J: Increment Coordiante Value of Middle point relative to starting point in Y direction
With direction

K: Increment Coordiante Value of Middle point relative to starting point in Z direction
With direction

F: Cutting speed

Note:

1. Middle point is any position point except starting point & end point.

2. System will alarm when three points are at one line.

3. When I,J,K are omitted, default value is I=0, J=0, K=0. But they cannot be omitted all at same time,otherwise system will alarm.

4. The meanings of I,J,K are similar to I,J,K of G02/G03.

5. G06 cannot be used for processing total round.

6. Compute of G06 command is very large, it only can work smoothly on modbus system, at normal system, it would work not smoothly.

Example:

```
G0 X10 Y28 Z10
G06 X30 Y98 Z10 I5 J-6 K-5 F100
    X130 Y198 Z120 I55 J-86 K-65
G0 X0 Z0
M02
```

3.29 Macro program instruction(G65/G66/G67)

3.29.1 Non-Mode Macro Command G65

Format: G65 P_ L_ A_ B_ C_ ......

Non-mode macro command G65 only work at current line , which is different to mode macro command(G66),which always work until macro cancel command(G67)

P_ : Specify name of macro program, E.g: P6000 , name of specified macro program is 6000 .

L_: Set times of call macro program

<A_B_C_... ...> : Argument , which is used for transfer data to macro varible(#**) , Transfering table is as following

Argument	Variable	Argument	Variable	Argument	Variable
A	#0	I	#7	T	#14
B	#1	J	#8	U	#15
C	#2	K	#9	V	#16
D	#3	M	#10	W	#17
E	#4	Q	#11	K	#18
F	#5	R	#12	Y	#19
H	#6	S	#13	Z	#20

Waraning:

1. Macro variables #100-#155&#190-#201 was occupied by system, user cannot use.

2. User cannot use G70,G71,G72,G73,G92,G76 etc loop command on Macro program.

Note: the address G, L, N, Q, P can't be used as user-defined variables.

Example:

Main program:9000	Macro program:8000
G00 X0 Z0	N1 #2=#0+#1
G65 P8000 L1 A5 B6	N2 IF （#2 EQ 10）GOTO 4
G0 X0 Z0	N3 GOO X#2
M30	N4 G00 Z#1
	N5 M99 ;Return

3.29.2 Mode Macro Command G66/G67

G66 is mode macro command , G67 is cancel mode macro command

Format: G66 P_ L_ A_ B_ C_ ......
G67

G66 Mode macro command,which always call macro program until macro cancel command(G67)

P_ : Specify name of macro program, E.g: P7000 , name of specified macro program is 7000 .

L_: Set times of call macro program

<A_B_C_... ...> : Argument , which is used for transfer data to macro varible(#**) , the transfering table is same as above table.

Example:

Main Program : 4000

```
G00 X0 Z0
G66 P6000 L2 A5 B6
A8 B1
A9 B10
G67
M30
Macro Program: 6000
N1 #2=#0+#1
N2 IF (#2 EQ 10) GOTO 4
N3 G00 X#2
N4 G00 Z#1
N5 M99          ; Return
```

3.29.3 Macro Program Instruction

3.29.3.1 Input Instruction: WAT

Waiting for the input port X valid or invalid instruction

Format: WAT+ (-) X

Attention: "+" to means the input is effective;

"-" means the input is invalid;

"X" means the input port X00-X55; see the I/O diagnosis;

3.29.3.2 Output Instruction: OUT

Set the output port Y is valid or invalid instruction

Format: OUT +(-)Y

Attention: "+" means the output is effective;

"-" means the output is invalid;

"Y" means the output port Y00-Y31; see the I/O diagnosis;

3.29.3.3 Assignment Instruction: =

Explanation: used for assignment of a variable

Eg.: #251=890.34   #450=#123

And also it could be mathematical expression , eg.: #440=#234+#470

3.29.3.4 Unconditional Jump: GOTO n

"GOTO n" is the command that for jump to the program line that is specified by sequence number (N**) unconditionally. n is the sequence number.

E.g.: GOTO 5 ; // Jump to N5 program line.

Note: when specified program line , n , is beyond sequence number of N1-N99999, cnc system will hint error.

n , program line,could be macro variable (#**)

E.g.: GOTO #100

3.29.3.5 Conditional Jump

1) IF (Conditional express) GOTO n

If condition is met, execute GOTO n ,jump to N** program line; if the condition isnot met, execute the next segment.

Example:

```
N1 IF(#200 EQ 1) GOTO 20
N10 G00 X0
N20 G00 Z0
```

Explanation: If #200 is equal to 1, system will execute GOTO 20 , jump to N20 , and execute "G00 Z0", if #200 isn't equal to 1, system don't execute operation of "GOTO 20" ,and will execute next segments , "G00 X0",and then execute "G00 Z0".

2) IF (Conditional express) THEN <A Expression>

<B opertional segment>

If condition is met, system execute A expression , and then execute B operational segment ; if condition is not meet, execute the next segment , B operation.

Example: #101=0

N1 IF(#100 EQ 1) THEN #101=1

N2 IF(#101 EQ 1) GOTO 4

N3 G00 X100

N4 G00 Z100

Explanation: If #100 is equal to 1, system will execute "#100=1", and then judge #101 is equal to 1 , jump to N4 & "execute G00 Z100" ; if #100 isn't equal to 1, system will judge #101 also isn't equal to 1 directly , and execute "G00 X100" & "G00 Z100".

NOTE: 1.<A expression> normally is assignment statement.

2. <A expression> after THEN must exist, otherwise system will hint grammatical errors.

Prolongation:

3) IF(conditional express)

<A operational command>

ELSE

<B operational command>

ENDIF

4) IF(conditional express)

<A operational command>

ELIF

<B operational command>

ENDIF

3.29.3.6 Loop Command

Format: (Conditions Initialization)

WHILE (conditional expression) DO n

<A operational segments>

[Alter condition of loop]

END n

<B operational segments>

When conditions are met during WHILE cycle command, execute the operational segments between DO n and END n . Otherwise,when condition isnot met, jump to the program line after END n ,also execute B operational segments.

We can nest for loops by placing one loop within another.

Note: 1.There must have operational codes that are for change condition at operational segments ,which is between Do n & END n. Otherwise system will enter endless loop.

2.Nesting of macro program loop statements of SZGH CNC system is 3 pcs of loops at most . Also n only could be 1 , 2 , 3 .

3.n of "DO n" & "END n" must keep same.

Example:
```
#100=2 #150=5 #200=25
WHILE （#100 LT 3）DO 1
 G00 X100
 WHILE （#150 EQ 5）DO 2
 G00 Y100
 WHILE （#200 GE 20）DO 3
 G00 Z100
 #200=#200-2
 END 3
 #150=#150-1
 END 2
 #100=#100-1
 END 1
```

3.29.4 Operators' meaning

Operator	Sign	Ex.	Operator	Sign	Ex.	Operator	Sign	Ex.
EQ	=	equal	GT	>	greater	LT	<	Less
NE	≠	unequal	GE	≥	G&E	LE	≤	L&E

3.29.5 Arithmetic & Logic Operation

Table:

Function	Format	Attention
Definition	#i = #j	
Addition Subtraction Multiplication Division	#i = #j + #k ; #i = #j - #k ; #i = #j * #k ; #i = #j / #k ;	
Sin Asin Cos Acos Tan Atan	#i = SIN(#j) ; #i = ASIN(#j); #i = COS(#j) ; #i = ACOS(#j); #i = TAN(#j); #i = ATAN(#j);	90.5 degrees means 90 degrees & 30 points
Square root Absolute value Rounding off Round down Round up Natural logarithm Exponential function	#i = SQRT(#j); #i = ABS(#j) ; #i= ROUND(#j); #i = FIX(#j); #i = FUP(#j); #i = LN(#j); #i = EXP(#j);	
Or Exclusive or And	#i = #j OR #k ; #i = #j XOR #k ; #i = #j AND #k ;	Executing with binary system

3.29.6 Local Variable

#0--#20 : local variables only can be used to store data in macro program, such as a result of operation, when power is off, the local variables are initialized to the empty. The argument assignment to the local variable when calling the macro program.

3.29.7 Global Variable

#21--#600 : Their meanings are the same in different macro program.

When power is off, the variable #21--#100 is initialized to zero, the variable #101--#600 data is saved not to loss even if the power is off.

3.29.8 System Variable

#1000-- : the system variables are used to change various data when reading the running CNC. For example, the current position and the compensation of tool.

Special Attention: macro variables #100--#155 and #190--#202 have been used by the system, users can not use.

3.29.9 System Parameter Variable

#1001--#1099 : Value of X-axis length compensation for T1--T99(Unit: um)
#1101--#1199 : Value of D1 radius compensation for T1--T99(Unit: um)
#1201--#1299 : Value of Y(C)-axis length compensation for T1--T99(Unit: um)
#1301--#1399 : Value of D2 radius compensation for T1--T99(Unit: um)
#1401--#1499 : Value of Z-axis length compensation for T1--T99(Unit: um)
#1501--#1599 : Value of D3 radius compensation for T1--T99(Unit: um)
#1601--#1699 : Value of A-axis length compensation for T1--T99(Unit: um)
#1701--#1799 : Value of D4 radius compensation for T1--T99(Unit: um)

3.29.10 I/O variable

#1800: X00-X07 (D0-D7) ; input resistor
#1801: X08-X15 (D0-D7) ; input resistor
#1802: X16-X23 (D0-D7) ; input resistor
#1802: X16-X23 (D0-D7) ; input resistor
#1803: X24-X31 (D0-D70 ; input resistor
#1804: X32-X39 (D0-D7) ; input resistor
#1805: X40-X47 (D0-D7) ; input resistor
#1806: X60-X67 (D0-D7) ; input resistor
#1807: X74-X81 (D0-D7) ; Alarm of driver/Spindle
#1808: Y00-Y15 (D0-D15) ; output resistor
#1809: Y16-Y31 (D0-D15) ; output resistor
#1810: Y32-Y47 (D0-D15) ; output resistor

Waraning:

1. Macro variables #100-#155&#190-#201 was occupied by system, user cannot use.

2. User cannot use G70,G71,G72,G73,G92,G76 etc loop command on Macro program.

Note: the address G, L, N, Q, P can't be used as user-defined variables.

3.29.11 Message Hint Dialog Box

Format: MSG(hint words) or MSG[hint words] ;

Hint words is that user want to hint message on cnc system.

Note: 1. This code can be used on normal NC programs.

2. After hint message, cnc system will pause program automatically.

Format: STAF(hint words) or STAF[hint words];

Hints words is that user want to hint message on cnc system. And CNC system don't pause program automatically.

3.29.12 Build Processing Program Automatically

3.29.12.1 New/Open a program

Format: FILEON(Program) or FILEON[Program]

Example: FILEON(AABBCC) or FILEON[AABBCC]

It means that new or open a program "AABBCC"

3.29.12.2 Close program

Format: FILECE

It means that close current opening program, if without this code, system will close current program after program is finished.

3.29.12.3 Write codes into program

Format: FILEWD(Blocks) or FILEWD[Blocks]

Exmaple: FILEWD(G54G0X0Z0) or FILEWD[G54G0X0Z0]

It means that write a blocks of "G54G0X0Z0" into current openning program.

3.29.12.4 Write current absolute coordiante into program

Format: FILEWC

It means that wirte current abolute coordiante value into program.

Example:

```
G0X0Z0
FILEON[AABBCC]
FILEWD [G54G0X0Z0]
G1X45Z89
FILEWC
G1X99Z76
FILEWC
FILECE
```

After finished this program, system will new a program of "AABBCC" under directory of program, its blocks is :

```
G54G0X0Z0
X45Z89
X99Z76
```

3.30 Complex function for Turning & Milling

CNC lahte system can finish milling processing, solutions is as follow:

1.Parameter sets: Name of 3rd axis should be set to "Y" axis, which is set by P101 in Axis parameter; programming mode is radius programming.

2.Related commands: The Others are same except following commands:

Code	Same functions in milling system	Code	Same functions in milling system
G990	G90 in milling system	G981	G81 in milling system
G991	G91 in milling system	G982	G82 in milling system
G994	G94 in milling system	G983	G83 in milling system
G995	G95 in milling system	G984	G84 in milling system
G998	G98 in milling system	G985	G85 in milling system
G999	G99 in milling system	G986	G86 in milling system
G973	G73 in milling system	G987	G87 in milling system
G974	G74 in milling system	G989	G89 in milling system
G976	G76 in milling system		

Note: CNC lathe system have these functions when CNC system is confiured with 3rd axis.

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

Polar coordinate interpolation is a function that exercises contour control in converting a command programmed in a Cartesian coordinate system to the movement of a linear axis (movement of a tool) and the movement of a rotary axis (rotation of a workpiece). This method is useful in cutting a front surface and grinding a cam shaft on a lathe.

Format:

G12.1 ; Starts polar coordinate interpolation mode (enables polar coordinate interpolation)

Specify linear or circular interpolation using coordinates in a Cartesian coordinate system consisting of a linear axis and rotary axis (virtual axis).

G13.1 ; Polar coordinate interpolation mode is cancelled (for not performing polar coordinate interpolation)

Note: Specify G12.1 and G13.1 in Separate Blocks.

G12.1 starts the polar coordinate interpolation mode and selects a polar coordinate interpolation plane, G17, (Fig3.20). Polar coordinate interpolation is performed on this plane.

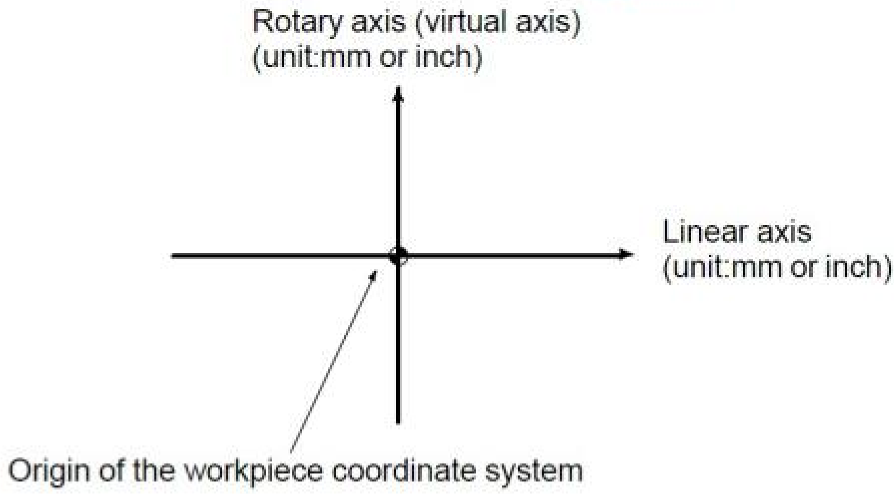


Fig3.20 Polar coordinate interpolation plane(G17)

Performed Steps of Polar coordinate interpolation: Polar coordiante interpolation program, which is based on X axis (Linear axis) & C axis(Rotary axis)

The linear and rotation axes for polar coordinate interpolation must be set in parameters (P102) beforehand.

Note: 1.When the power is turned on or the system is reset, polar coordinate interpolation is canceled (G13.1). G12.1 & G13.1 are mode codes.

2.G codes which can be specified in the polar coordinate interpolation mode

G01:	***linear interpolation***
G02,G03 :	***Circular interpolation***
G04:	***Dwell***

G40,G41,G42: Tool nose radius compensation

(Polar coordinate interpolation is applied to the path after cutter compensation.)

3. When G12.1,start polar coordiante interpolation, system will shift to G17 plane automatically; When running G13.1, cancel polar coordiante interpolation, system will return to G18 plane automatically.

4. Progoramming mode: X-axis: Diameter/Radius programming; C-axis: Radius programming. Even when diameter programming is used for the linear axis (X-axis), radius programming is applied to the rotary axis (C-axis). Programing unit is mm, displaying unit is degree.

5. In the polar coordinate interpolation mode, program commands are specified with Cartesian coordinates on the polar coordinate interpolation plane. The axis address for the rotation axis is used as the axis address for the second axis (virtual axis) in the plane. Whether a diameter or radius is specified for the first axis in the plane is the same as for the rotation axis regardless of the specification for the first axis in the plane.

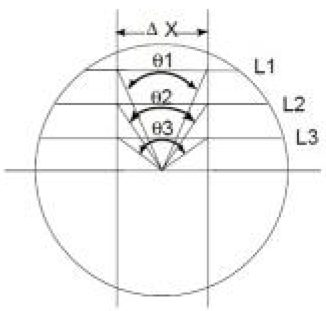
6. The virtual axis is at coordinate 0 immediately after G12.1 is specified. Polar interpolation is started assuming the angle of 0 for the position of the tool when G12.1 is specified.

7. Specify the feedrate as a speed (relative speed between the workpiece and tool) tangential to the polar coordinate interpolation plane (Cartesian coordinate system) using F.

8. The addresses for specifying the radius of an arc for circular interpolation (G02 or G03) in the polar coordinate interpolation plane depend on the first axis in the plane (linear axis). I and J in the Xp −Yp plane when the linear axis is the X −axis or an axis parallel to the X −axis.The radius of an arc can be specified also with an R command.

9.Polar coordinate interpolation converts the tool movement for a figure programmed in a Cartesian coordinate system to the tool movement in the rotation axis (C-axis) and the linear axis (X-axis). When the tool moves closer to the center of the workpiece, the C-axis component of the feedrate becomes larger and may exceed the maximum cutting feedrate for the C-axis (set by P109 in Axis parameter), causing an alarm (see the figure below). To prevent the C-axis component from exceeding the maximum cutting feedrate for the C-axis, reduce the feedrate specified with address F or create a program so that the tool (center of the tool when tool nose radius compensation is applied) does not move close to the center of the workpiece.

WARNING



Consider lines L1, L2, and L3. ΔX is the distance the tool moves per time unit at the feedrate specified with address F in the Cartesian coordinate system. As the tool moves from L1 to L2 to L3, the angle at which the tool moves per time unit corresponding to ΔX in the Cartesian coordinate system increases from θ1 to θ 2 to θ3.

In other words, the C–axis component of the feedrate becomes larger as the tool moves closer to the center of the workpiece. The C component of the feedrate may exceed the maximum cutting feedrate for the C–axis because the tool movement in the Cartesian coordinate system has been converted to the tool movement for the C–axis and the X–axis.

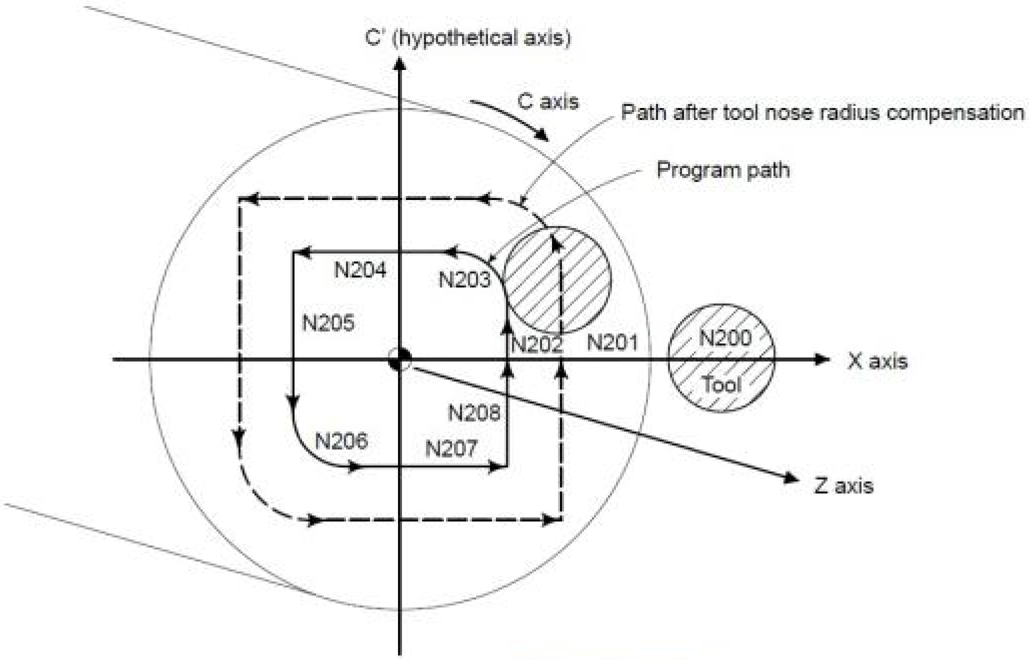
L : Distance (in mm) between the tool center and workpiece center when the tool center is the nearest to the workpiece center

R :Maximum cutting feedrate (deg/min) of the C axis

Then, a speed specifiable with address F in polar coordinate interpolation can be given by the formula below. Specify a speed allowed by the formula. The formula provides a theoretical value; in practice, a value slightly smaller than a theoretical value may need to be used due to a calculation error.

$$F < L \times R \times \frac{\pi}{180} \text{ (mm/min)}$$

Example: Polar Coordinate Interpolation Program Based on X Axis (Linear Axis) and C Axis (Rotary Axis)



X axis is by diameter programming, C axis is by radius programming

N10 T0202
...
N100 G00 X150 C0 Z0 ; Positioning to start position
N200 G12.1 ; Start of polar coordiante interpolation
N201 G42 G01 X40.0 F20 ;
N202 C10 ;
N203 G03 X-20.0 C20.0 R10.0;
N204 G01 X-40.0;
N205 C-10.0;
N206 G03 X-40 C-20 R20 ;
N207 G01 X40 ;
N208 G03 X80 C-20 R20 ;
N209 G01 C0 ;
N210 X150.0 ;

Geometry program (Program based on cartesian coordiantes on X-C' plane)

N300 G13.1; Cancellation of polar coordinate interpolation
N400 Z100.0 C0 ;
...
N500 M30 ;

Chapter 4 M INSTRCUTIONS

4.1 M Function (Auxiliary Function)

M instruction consists of instruction address M and its following 1~2 bit digits, used for controlling the flow of executed program or outputting M instructions to PLC.

M □□

→ Instruction value(the leading zero can be omitted)

→ Instruction address

When address M followed by a number is specified, a code signal and strobe signal are transmitted. These signals are used for turning on/off the power to the machine.

In general, only one M code is valid in a block but up to three M codes can be specified in a block (although some machines may not allow that). The correspondence between M codes and functions is up to the machine tool builder.

All M codes are processed in the machine except for M97 M98, M99,M codes for calling a subprogram , and M codes for calling a custom macro. Refer to the appropriate manual issued by the machine tool builder.

The following M codes have special meanings.

M00, M01, M02, M30, M97, M98, M99 must not be specified together with another M code.

Some M codes other than M00, M01, M02, M30, M97, M98 and M99 cannot be specified together with other M codes; each of those M codes must be specified in a single block.

4.1.1 Program Stop(M00)

Automatic operation is stopped after a block containing M00 is executed.

When the program is stopped, all existing modal information remains unchanged. The automatic operation can be restarted by actuating the cycle operation. This differs with the machine tool builder.

4.1.2 Optional Stop (M01)

Similarly to M00, automatic operation is stopped after a block containing M01 is executed. This code is only effective and program stop when input point M22(PIN5 of CN10 Plug) is valid

4.1.3 End of Program (M02,M30)

This indicates the end of the main program.Automatic operation is stopped and the CNC unit is reset.

4.1.4 Cycle of Program (M20)

Run program cycle , cycle time is set by P18 in User parameter.

4.1.5 Account of Workpiece(M87)

Number of workpieces will add one automatically as P10=0 in Other parameter.

4.1.6 Uncondtional Jump (M97)

M97 P_ , jump to specfied block number that specified by P.P4 said entrance line number with 4 field numbers specified program transfer main program.

Example: M97 P0120 , when excutive this code, CNC will jump to "N0120" block and run.

4.2 Subprogram Configuration

There are two program types, main program and subprogram. Normally, the CNC operates according to the main program. However, when a command calling a subprogram is encountered in the main program, control is passed to the subprogram. When a command specifying a return to the main program is encountered in a subprogram, control is returned to the main program.

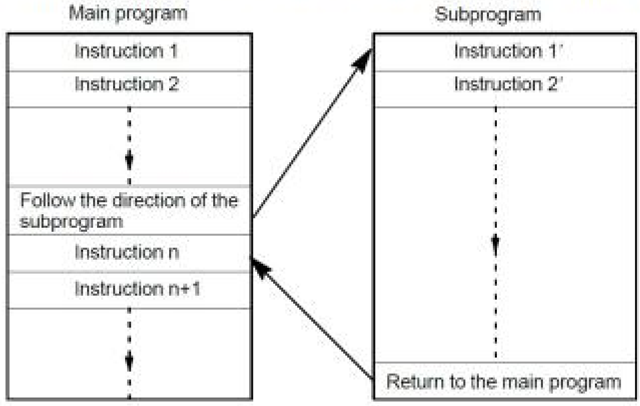


Fig4.2.1 Main program and subprogram

If a program contains a fixed sequence or frequently repeated pattern, such a sequence or pattern can be stored as a subprogram in memory to simplify the program.

A subprogram can be called from the main program.

A called subprogram can also call another subprogram.

4.2.1 Calling of Subprogram (M98)

This code is used to call a subprogram.The code and strobe signals are not sent.

M98 P_ L_ ;

P_: specify address & name of subprogram. Eg.: Psub/1390; sub is a folder.

Subprogram can be hidden files that don't display in program district,First character of these program must be a "HIDEFILE".

Example: "HIDEFILE01", the subprogram in the program area is not displayed, user can use these type commands to call subprogram.

M98 PHIDEFILE01

or **M98 P*01**

If user want to call subprogram in USB-disk, Format is " P[_" or " P]_".

Example: **M98 P[A1234** ; Call A1234 subprogram in USB-disk;

L_: number of times the subprogram is called repeatedly.When no repetition data is specified, the subprogram is called just once.

M98 instruction can be omitted. **Format: PP_** .

Example: **PP[FFDE ;** call "FFDE" subprogram in USB-disk;

Note: 1. There must has a blank before "L_" in this system;

2. Subprogram must be an independent program.

4.2.2 End of Subprogram (M99)

This code indicates the end of a subprogram.M99 execution returns control to the main program.No code or strobe signal is sent.

1) M99 in main program is same to M02;

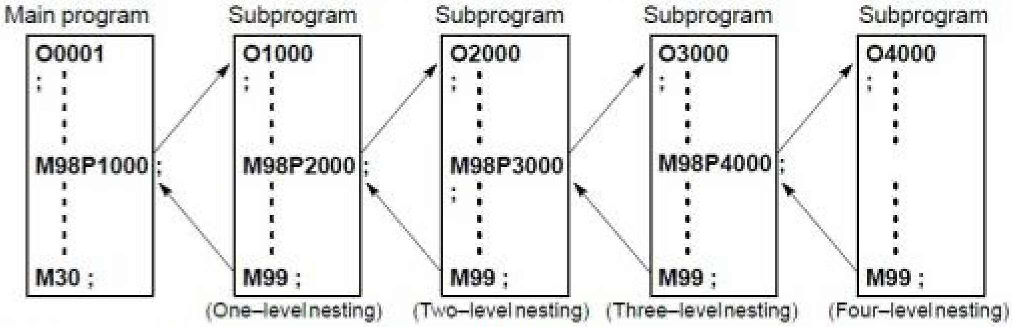
2) M99 with P in main programis same to M97 ;

3) M99 in subprogram return to next block of M98;

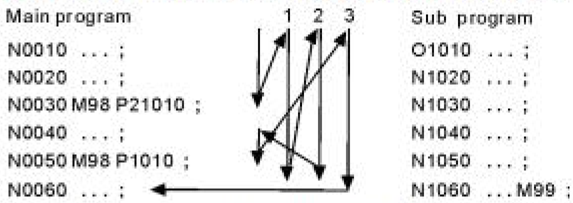
4) M99 with P in the subroutine return to specified block in main program.

When the main program calls a subprogram, it is regarded as a one–level subprogram call.

Thus, subprogram calls can be nested up to four levels as shown below.



Eg.:Execution sequence of subprograms called from a main program.A subprogram can call another subprogram in the same way as a main program calls a subprogram.



A single call command can repeatedly call a subprogram up to 9999 times.

4.3 Standard PLC M Command List

4.3.1 M Output Command List

Function	Code	Introduction	Statement
Spindle	M03	Spindle on CW	Functions interlocked and states reserved
	M04	Spindle on CCW	
	M05	Spindle Stop	
Coolant	M08	Coolant ON	Functions interlocked and states reserved
	M09	Coolnat OFF	
Chuck	M10	Chuck Clamping	Functions interlocked and states reserved
	M11	Chuck Unclamping	
Tailstock	M79	Tailstock Forward	Functions interlocked and states reserved
	M78	Tailstock Backward	
Lubricate	M32	Lubrication ON	Functions interlocked and states reserved
	M33	Lubrication OFF	
Huff	M59	Huff ON	Functions interlocked and states reserved
	M58	Huff OFF	
1	M61	User-define Output 1 ON	Functions interlocked and states reserved
	M60	User-define Output 1 Off	
2	M63	User-define Output 2 ON	Functions interlocked and states reserved
	M62	User-define Output 2 Off	
3	M65	User-define Output 3 ON	Functions interlocked and states reserved
	M64	User-define Output 3 Off	
4	M67	User-define Output 4 ON	Functions interlocked and states reserved
	M66	User-define Output 4 Off	
5	M69	User-define Output 5 ON	Functions interlocked and states reserved
	M68	User-define Output 5 Off	
6	M71	User-define Output 6 ON	Functions interlocked and states reserved
	M70	User-define Output 6 Off	

7	M73	User-define Output 7 ON	Functions interlocked and states reserved
	M72	User-define Output 7 Off	
8	M75	User-define Output 8 ON	Functions interlocked and states reserved
	M74	User-define Output 8 Off	

M75/M74: When CNC system is configurated with C axis , which is used for switch the control mode (Position control Mode & Analog Speed Mode) of servo spindle. When M75 is valid, which is set servo spindle to position control mode,when M03/M04 is valid, turn off M75.

M71/M70: When P20 in Other parameter set to 1, M11 output M71.

M73/M72: When P21 in Other parameter set to 1, M78 output M73.

Note: All M output commands, output 0V effective level.

4.3.1.1 Spindle Control (M03/M04/M05)

M03 is for control CW of spindle , M04 is for control CCW of spindle , M05 is for stop spindle.

Input Point

M03	PIN19_CN3 Plug	CW of Spindle
M04	PIN7_CN3 Plug	CCW of Spindle
M05	PIN20_CN3 Plug	STOP of Spindle

Wiring Diagram

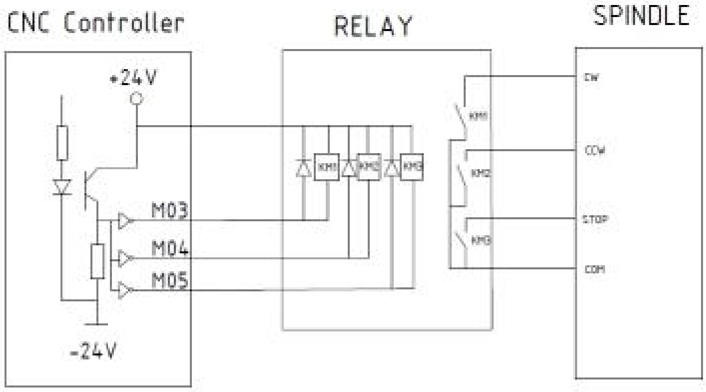


Fig4.3.1 Wiring Diagram for Spindle

According to this wiring diagram , it will consist a control circuit with +24V when system output M03/M04/M05 signal, coil of relay works and NO type switches will be ON , and control related function of spindle.

Note:1. Effective level of all output points is 0V.

2.When the relays and others inductance load, must connected with the reverse diode to absorb the reverse current so as not to damage the system, if use the electromagnetic contactor, then plus resistive and capacitive spark circuit.

Parameter Set

System output dual analog for spindle for control speed of two spindles,2nd analog output 1st analog,which set by D11_P9 in Other parameter.

In Axis parameter

P7 : Set the braking time of spindle, also the hold time of output M05, Unit:10ms. The time less , the braking faster.

P8 : Set the braking signal is long signal 1 or short signal 0.

P9: To set system whether checking spindle feedback signal of spindle position, also the feedback signal is spindle encoder signal. To set the parameter value 1 means check; 0 means not to check.

P10: To set feedback pulse numbers of spindle encoder turn a round, the value: Line number of encoder * 4.

P11: Whether turn on the spindle or not when shifting [1 means on, 0 means off]

P51: The speed of motor when spindle shifting (unit: 1/100rpm)

P52: The direction when spindle shifting (0 means positive, 1means negative)

P53: The stopping time when spindle shifting (unit: 10ms)

P54: Turning time of low speed when spindle shifting (unit:10ms)

P55: Stopping delay time of spindle (Unit:10ms)

In Speed parameter:

P8: To set the speed of spindle at manual condition. Unit: rpm.

P36: To set the max speed of spindle, also the speed of corresponding 10V.

Attention: when spindle system is with gears,this is the speed of first gear.

P37: To set the max speed of spindle(second gear), that's the turning speed of corresponding 10V instruction voltage.Unit: rpm.

P38: To set the max speed of spindle (Third gear), that's the turning speed of corresponding 10V instruction voltage. Unit: rpm.

P39: To set the max speed of spindle (Fourth gear), that's the turning speed of corresponding 10V instruction voltage. Unit: rpm.

P40: To set the highest speed of second spindle, also the speed of corresponding 10V. Unit: r/min

In Other parameter:

P13: To set whether spindle and chuck is interlocking or not: 0 means they are separately; 1 means the spindle only start turning when chuck on. The thumbstall can't be use when the spindle is turning.

Setting parameter is related with the configuration of lathe and user's service condition, but consider for safe, suggest setting 1,also interlocking.

4.3.1.2 Spindle Gear Shifting(M41/M42/M43/M44)

Remark	PIN	Function	Command
S01	PIN10_CN3 Plug	Output 1st gear of spindle	M41
S02	PIN23_CN3 Plug	Output 2nd gear of spindle	M42
S03	PIN11_CN3 Plug	Output 3rd gear of spindle	M43
S04	PIN24_CN3 Plug	Output 4th gear of spindle	M44

M41/M42/M43/M44 output S01/S02/S03/S04 for shifting gear of spindle , and adjust analog voltage to adjust speed of spindle.

P36 in Speed parameter is set for max speed of 1st class spindle;

P37 in Speed parameter is set for max speed of 2nd class spindle;

P38 in Speed parameter is set for max speed of 3rd class spindle;

P39 in Speed parameter is set for max speed of 4th class spindle;

Note:Functions interlocked and states reserved

4.3.1.3 Coolant(M08/M09)

M08:Turn on coolant

M09: Turn off coolant

Remark	PIN	Function
M08	PIN8_CN3 Plug	Turn On/Off coolant

4.3.1.4 Lubricate(M32/M33)

M32: Turn on lubrication

M33: Turn off lubrication

Remark	PIN	Function
M32	PIN9_CN3 Plug	Turn On/Off Lubrication

In Other parameter,

P4 controls the function of lubricate automatically.

P6 is set the interval time of lubrication (Unit: s);

P5 set the time of lubrication ,also holding time of output M32(Unit: 10ms).

4.3.1.5 Chuck(M10/M11)

M10/M11 are for control clamping/unclamping of chuck.

Remark	PIN	Function
M10	PIN21_CN3 Plug	Control Chuck clamping
M71	PIN9_CN10 Plug	Control chuck unclamping(Spare)
M12	PIN11_CN10 Plug	Detect position of clamping_chuck(spare)
M14	PIN24_CN10 Plug	Detect position of unclamping_chuck(Loose)(spare)
M16	PIN12_CN10 Plug	Input point for switch to control chuck(spare)

Chuck of this system control is related with parameter as follows:

In Other parameter:

P2: Type of chuck, (Inner: Chuck to center when M10; Outer: Chuck opening outward when M10). 1 means outer, 0 means inner.

P13: Interlock between Chuck & Rotation_Spindle.0: No interlock, 1: yes.

P15: Detect position of clamping/unclamping of chuck,0:No detect, 1: yes.

M12: input point for detecting position of clamping of chuck;

M14: input point for detecting position of unclamping of chuck.

P20: Type of controlling singal for chuck,0: one output controlling signal for chuck; 1: two output controlling signals for chuck.

M10:output point for controlling clamping of chuck , controlling code is M10;

M71:output point for controlling unclamping of chuck,controlling code is M11.

P22: External Switch(Foot switch) for chuck, reciprocating mode, price one time clamping chuck; press twice time, unclamping chuck. 0: without switch ; 1: with external switch for control chuck; input signal is M16.

P24: Holding time of output M10/M71 of chuck, unit: s. 0: mode type*.

4.3.1.6 Tailstock(M79/M78)

M79: Tailstock forward

M78: Tailstock backward

Remark	PIN	Function
M79	PIN22_CN3 Plug	Tailstock Forward/Backward
M73	PIN22_CN10 Plug	Control chuck unclamping(Spare)
M18	PIN11_CN10 Plug	Detect position of forward_Tailstock(spare)
M28	PIN24_CN10 Plug	Detect position of backward_Tailstock(spare)
M14	PIN24_CN10 Plug	Input point for switch to control Tailstock(spare)

Parameters set for tailstock,in Other parameter:

P16: Detect Position of Forward/Backward of Tailstock; 0: no detect,1:Yes;

M18: input point for detecting position of forward of tailstock;

M28: input point for detecting position of backward of tailstock.

P21: Type of controlling singal for tailstock,0: one output controlling signal for tailstock; 1: two output controlling signals for tailstock.

M79:output point for controlling forward of tailstock , controlling code is M79;

M73:output point for controlling backward of tailstock,controlling code is M78.

P23: External Switch(Foot switch) for tailstock, reciprocating mode, price one time, tailstock forward; press twice time, tailstock backward. 0: without switch ; 1: with external switch for control tailstock; input signal is M14.

P25: Holding time of output M79/M73 of tailstock, unit: s. 0: mode type*.

Note: 1. When user-defined signals M71/M70 , M73/M72 are used for output signal of spindle chuck and Tailstock, it can't be used for other functions.

2. M12,M14,M1,M18,M28 are multi-function codes, only one function when using.

3. *mode type means that once valid,always valid until cancelling/resetting code.

4.3.1.7 Condition Output of Machine Tool(M65/M67/M69)

M65/M67/M69 be used for output condition of machine tool, set by parameters

Remark	PIN	Function
M65	PIN20_CN3 Plug	Output Alarm condition of machine tool
M67	PIN8_CN3 Plug	Output Running condition of machine tool
M69	PIN21_CN3 Plug	Output Pausing condition of machine tool

Parameter sets for condition output ofmachine tool. In Other parameter:

P28: M65/M69 are as outputs for run/halt condition of machine tool,0:no,1:yes.

P29: M67 is as output for alarm condition of machine tool, 0: no, 1:yes.

Note:1.When M65,M67,M69 are used for output condition of machine tool, and then they can't be used for other functions.

2.Valid level of all output points is 0V. When fixed with inductance load, must connected with the reverse diode to protect inner circuit of cnc system.

4.3.2 M Input Command List

No.	Code	Function Introduction	Statement
1	M12	Check M12 is valid	These codes can be used for conditional wait or conditional jump.
	M13	Check M12 is invalid	
2	M14	Check M14 is valid	
	M15	Check M14 is invalid	
3	M16	Check M16 is valid	
	M17	Check M16 is invalid	
4	M18	Check M18 is valid	
	M19	Check M18 is invalid	
5	M28	Check M28 is valid	
	M29	Check M28 is invalid	
6	M22	Check M22 is valid	
	M23	Check M22 is invalid	
7	M24	Check M24 is valid	
	M25	Check M24 is invalid	

Note: All M input commands, input effective level is 0V.

There are two kinds of special application about M input commands,as following,

a) Conditional Wait

Example: M12

When Input point M12 is valid, program goes on run following blocks, if M12 is invalid, system will wait if M12 is valid.

b) Conditional Jump

Example: M14 P0120

When the program running to this block and the system detecting if M14 input signal is valid. When M14 is valid, program will jump to 120th line of program (also N0120 block), otherwise , system will execute the next block.

4.4 Analog Speed of Spindle(S , SS)

SZGH CNC system support dual analog outputs for SP_speed.

Speed of 1st spindle is set by "S**"; Speed of 2nd spindle is set by "SS**".

P36 in Speed parameter is set for max speed of 1st spindle; P40 in Speed parameter is set for max speed of 2nd spindle.

Note: D11_P9=1 in Other parameter is set for output 1st analog voltage to dual analog output(+10V of CN3&CN10) at same time, without function of "SS**".

There are two gears control ways for 1st spindle as following,

(1) Four gears spindle speed electrical control, output four bits code of step speed change, M41-M44 control instructions correspondly output S01-S04 code, with fixed speed. P50/P51/P52/P53/P54 in Axis parameter are set for mode of shifting.

(2) Four gears+Variable speed, M41-M44 instruction control, correspond the output S01-S04 code. P42/P43/P44/P45 in Speed parameter are set for maximum speed of corresponding gear, P50/P51/P52/P53/P54 in Axis parameter are set for mode of shifting.

Variable speed,range is 0-99999, output 0-10V variable-frequency voltage.

Note: Output 10V is corresponding to max speed of spindle.

4.5 T Tool Function Command

Format	Function
Tab	a: Exchanging Tool Number b: Tool compensate number

Note: 1. a=0 ; means that don't exchange tool;

2. b=0; means that don't make tool compensation , display machine coordinate;

3.Range: a= 0~99 ; b=0~99.

Eg1: T0204 ; change to No.2 tool , and make No.4 tool compensation.

T0300 ; change to No.3 tool , and don't make tool compensation.

T0004 ; don't exchange tool , only make No.4 tool compensation.

When machine tool is with electrical turret, each station can fix serval tools,adopt different tool compensation to exchange tools.

Eg2: 4 station turret, there are 2 tools in 3rd station (use No.3 & No.5 tool compensation)

```
N0000   T0101    ; change to No.1 tool, and make no.1 tool compensation
N0001   G0  X30 Z500
N0002   T0303         ; change to 1st tool in 3rd station
N0003   G00  X50
N0004   T0505         ; change to 2nd tool in 3rd station
N0005   M02
```

4.6 User-defined macro instruction(G120-G160,M880-M889)

Every user-defined G code is corresponding to a macro program ProgramGxxx, the M code is corresponding to a macro program of ProgramUser0 --ProgramUser9, the user cannot programme the macro program in NC system, must edit the macro code in the computer, and then copy into the system.

For example, defines the G152 function: the arc model porous drilling cycle. (must copy the macro program ProgramG152 into system).

Format:G152 Xx Yy Zz Rr Ii Aa Bb Hh Ff;

X: The X coordinate with absolute value or incremental value of center to specify.

Y: The Y coordinate with absolute value or incremental value of center to specify.

Z: Hole depth

R: Approaching fast to the point coordinate

F: Cutting feed speed

I: Radius

A: The angle of the first hole

B: Incremental angle specify(CW when negative)

Macro program ProgramG152 as follows:

```
#80=#0
#81=#1
#82=#2
#83=#3
#84=#4
#85=#5
#86=#6
#87=#7
#88=#8
#89=#9
#90=#10
#91=#11
#92=#12
#93=#13
#94=#14
#95=#15
#96=#16
#97=#17
#98=#18
#99=#19
#100=#20
#30=#4003
#31=#4014
G90
IF[#30 EQ 90] GOTO 1
G53
```

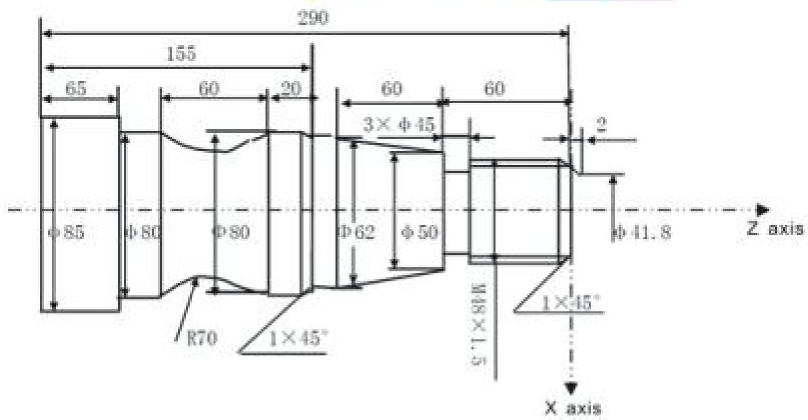
```
#98=#5001+#98
#99=#5002+#99
N1 WHILE[#86 GT 0] DO 1
#35=#98+#87*COS[#80]
#36=#99+#87*SIN[#80]
G81X#35Y#36Z#100R#92F#85
#80=#80+#81
#86=#86-1
END 1
G#30 G#31 G80
M99
```

4.7 Synthetic instance for programming

In the actual programming, must according to the drawing and processing requirement to select the correct install folder mode and suitable tool, and combined with the actual working performance of lathe to select the right cutting allowance, for example:

Example 1：The tool is:

T01 cylindrical cutting tool; T02 cutting groove, tool width 3m; T03 thread tool with 60 degree angle



Program as follows:

N10 M03 S1000; Start spindle
N20 T0101; Choose the first tool and execute the first redeem
N30 G00 X41.8 Z2 M08; Move fast to the cutting point， cutting fluid is on
N40 G01 X48 Z-1 F100; Chamber
N50 Z-60; Fine machining for thread
N60 X50; Tool is backing
N70 X62 W－60; Fine machining in cone
N80 W－15; Fine machining in Φ62mm ex-circle
N90 X78; Tool is backing
N100 X80 W-1; Chamber
N110 W-19; Fine machining in Φ80mm ex-circle
N120 G02 X80 W-60 R70; Fine machining in arc（I63.25 K－30）
N130 G01 Z-225; Fine machining in Φ80mm ex-circle

N140 X85; Tool is backing
N150 Z-290; Fine machining in Φ85mm ex-circle
N160 X90 M09; Tool is backing， cutting fluid is off
N170 G00 X150 Z50; Move fast to the point of changing tool
N180 T0202; Change tool and set the No.2 redeem
N190 M03 S800; Change speed of spindle
N200 G00 X51 Z-60 M08; Move fast to the processing point，use the left point of tool to redeem
N210 G01 X45 F90; Cutting Φ45mm groove
N220 G00 X51; Tool is backing
N230 X150 Z50 M09; Return to the point of backing tool， cutting fluid is off
N240 T0303; Change tool and set the redeem
N250 M03 S1500; Change speed of spindle
N260 G00 X62 Z6 M08； Move fast to the processing point， cutting fluid is on
N270 G92 X47.54 Z－58 F1.5; Cutting thread is cycle
N280 X46.94;
N290 X46.54;
N300 X46.38;
N310 G00 X150 Z50 M09; Return to the point of start cutting, cutting fluid is off
N320 T0300; Cancel redeem
N330 M05; Stop spindle
N340 M30; Program is over