In this user manual, we will try to describe various matters related to the operation of the system. Due to space limitations and specific use of the product, it is not possible to describe in detail all the operations that do not have to be done and/or cannot be done in the system. Therefore, any operation not specifically specified in this user manual is considered "impossible" or "not allowed".

preface

Dear Customer:

For the products you choose, the company is deeply honored and grateful!

This user manual details the programming, operation, and installation and connection of lathe CNC. In order to ensure the safe, normal and effective operation of the product, please be sure to carefully read this user manual before installing and using the product.

Security Warnings

⚠Improper operation will cause accidents, and personnel with appropriate qualifications must be required to operate the system.

Special note: The system power supply installed on the chassis (inside) is a dedicated power supply provided only for the numerical control system manufactured by the company. Users are prohibited from using this power supply for other purposes. Otherwise, there will be great danger!

Precautions

■ Transportation and storage

1. The stacking of product packaging boxes should not exceed six layers;
2. Do not climb, stand or place heavy objects on the product box;
3. Do not use the cable connected to the product to drag or move the product;
4. Collision and scratching of the panel and display screen are strictly prohibited;
5. The product packaging box should avoid moisture, exposure to the sun and rain.

■ Unpacking inspection

1. After opening the package, please confirm whether it is the product you purchased;
2. Check whether the product is damaged during transportation;
3. Check whether the components are complete and whether there is damage to the list;
4. If there is a product model discrepancy, missing accessories or transportation damage, please contact the company in time.

■Wiring

1. The personnel participating in the wiring and inspection must be professionals with corresponding abilities;
2. The product must be reliably grounded, the grounding resistance should be less than 0.1Ω, and the neutral wire (zero line) cannot be used instead of the ground wire;
3. The wiring must be correct and firm, so as not to lead to product failure or unintended consequences;
4. The surge absorption diode connected to the product must be connected in the prescribed direction, otherwise the product will be damaged;

5. Before plugging in or unplugging the plug or opening the product chassis, you must cut off the power to the product.

■Overhaul

1. The power supply must be cut off before overhauling or replacing components;

2. When short circuit or overload, the fault should be checked, and the restart can be restarted after troubleshooting;

3. Do not frequently power off the product, if you must re-energize after the power failure, the interval is at least 1min.

statement

due to

The manual describes as many different elements as possible, but because of the possibilities involved to describe all the operations that can or cannot be performed, the contents of this manual that are not specifically described may be considered unusable.

warn

programming, and operating

Before installing and connecting, programming and operating this product, you must read the manual of this product and the instruction manual of the machine tool manufacturer in detail, and strictly follow the requirements of the manual and the manual to carry out related operations, otherwise it may lead to damage to the product and machine tool, scrapping of the workpiece and even personal injury.

note

described in this manual are for purposes only

The product functions and technical indicators (such as accuracy, speed, etc.) described in this manual are only for this product, the CNC machine tool of this product is installed, the actual functional configuration and technical performance are determined by the design of the machine tool manufacturer, and the functional configuration and technical indicators of the CNC machine tool are subject to the instruction manual of the machine tool manufacturer;

Although the system is equipped with a standard machine tool operation panel, the functions of the buttons of the standard machine tool panel are defined by the PLC program (ladder diagram). Please note that the functions of the keys in the machine panel of this manual are described for standard PLC procedures!

***The contents of this manual are subject to change without notice**

First programming instructions

Describes technical specifications, product models, directive codes, and program formats.

Safety Responsibility

Manufacturer's responsibility for safety

- The manufacturer shall be responsible for the design and structural elimination and/or control of the CNC system provided and the accessories supplied with the accompanying supply.
- The manufacturer shall be responsible for the safety of the CNC system provided and the accessories supplied with the accompanying supply.
- The manufacturer shall be responsible for the information and suggestions for the use of the user.

User's Security Responsibility

——Users should learn and train the safe operation of the numerical control system, and be familiar with and master the content of safe operation.

-- The user shall be responsible for the safety and danger caused by adding, changing or modifying the original NUMERICAL control system and accessories.

- The user shall be responsible for the hazards caused by the failure to operate, adjust, maintain, install and store the product in accordance with the provisions of the manual.

***This manual is for the collection of end users. Sincerely thank you** for your **friendly support when using our products!**

directory

The first one

Programming instructions

Chapter 1 Programming Basics

1.1 Lathe Introduction

1.1.1 Product Introduction

The system adopts 32-bit high-performance CPU and ultra-large-scale programmable device FPGAs, real-time control and hardware interpolation technology to ensure the high efficiency of the system μ m-level accuracy, and the editable PLC makes the logic control function more flexible and powerful.

The system 1ms high-speed interpolation, 0.1μm control accuracy, significantly improve the efficiency, accuracy and surface quality of parts processing

* 1ms interpolation cycle, control accuracy 1μm, 0.1μm optional
* Maximum speed 60m/min (24m/min at 0.1μm).
* Adapted to the servo spindle can realize continuous spindle positioning, rigid tapping, rigid threading
* Built-in multi-PLC program, the currently running PLC program can be selected
* Support statement-based macro code programming, support macro program calls with parameters
* Support metric/ imperial programming, with automatic tool pairing, automatic chamfering, tool life management functions
* Support Chinese, English display, selected by parameters
* Usb interface, support USB stick file operation, system configuration and software upgrade
* 1 handwheel input, support handheld unit

1.1.2 Technical Specifications

Feed shaft function

* Minimum input increments: 0.001mm (0.0001inch) and 0.0001mm (0.00001inch) are optional
* Minimum instruction increments: 0.001mm (0.0001inch) and 0.0001mm (0.00001inch) are optional
* Maximum Stroke: ± 99999999 × Minimum instruction increment

* Fast movement speed: up to 60m/min
* Fast magnification: F0, 25%, 50%, 100% a total of four levels of real-time trimming
* Feed rate: 0 ~ 150% A total of 16 levels of real-time correction
* Interpolation mode: straight line interpolation, circular arc interpolation (support three-point circular interpolation), thread interpolation, rigid tapping
* Auto chamfer function

Thread function

* Normal thread (follow spindle) / rigid thread
* Single/multi-head metric straight threads, tapered threads and face threads, equal pitch threads and variable pitch threads
* Thread ejection length, angle and speed characteristics can be set
* Thread pitch: 0.01mm ~ 500mm or 0.06 teeth / inch ~ 2540 teeth / inch

Acceleration and deceleration function

* Cutting feed: front acceleration and deceleration linear type, front acceleration and deceleration S type, rear acceleration and deceleration linear type, rear acceleration and deceleration exponential type
* Fast movement: front acceleration and deceleration linear type, front acceleration and deceleration S type, rear acceleration and deceleration linear type, rear acceleration and deceleration exponential type
* Thread cutting: linear, exponential optional
* The start speed, stop speed and acceleration and deceleration time of acceleration and deceleration are set by the spindle function by parameters
* 1 channel spindle encoder feedback, spindle encoder line number can be set (100p/r ~ 5000p/r).
* The transmission ratio of the encoder to the spindle: (1 ~ 255): (1 ~ 255).
* Spindle speed: can be given by S code or PLC signal, the speed range is 0r/min ~ 9999r/min
* Spindle magnification: 50% ~ 120% A total of 8 levels of real-time adjustment
* Spindle constant linear speed control
* Rigid tapping

Tool function

* Tool length compensation
* Tip radius compensation (type C).
* Tool wear compensation
* Tool life management
* Knife pairing method: fixed point knife, trial cutting knife, back reference point knife, automatic knife
* Tool bias execution mode: modify the coordinate mode and tool movement mode

Accuracy compensation

* Backlash compensation
* Memory-type pitch error compensation

PLC function

* Two-stage PLC program, up to 4700 steps, level 1 program refresh cycle 8ms

* PLC program communication download
* Support PLC warning and PLC alarm
* Supports multiple PLC programs (up to 20), selectable currently running PLC programs

Hmi

* 8. 4-inch widescreen LCD with a resolution of 800×480
* Displayed Chinese, English and other languages
* 2D tool trajectory display
* Real-time clock

Operations management

* Operation mode: edit, automatic, input, machine machine zero, hand wheel / step, manual, program zero
* Multi-level operation permission management
* Alarm logs

Program editing

* Program capacity: 56MB, 400 programs (including subroutines, macro programs).
* Editing function: program / program segment / word retrieval, modification, deletion, copy, paste
* Program format: ISO code, support statement macro code programming, support relative coordinates, absolute coordinates and mixed coordinate programming
* Program invocation: Support macro program calls with parameters, 4-level subprogram nested communication functions
* RS232: Two-way transmission of parts programs, parameters and other files, support PLC program, system software serial port upgrade
* USB: U disk file operation, U disk file direct processing, support PLC program, system software U disk upgrade

Security features

* Emergency stop
* Hardware stroke limit
* Software trip check
* Data backup and restore

G code table

code	function	code	function
G00	Quick positioning	G50	Floating workpiece coordinate system
G01	Straight line interpolation	G54-G59	Set the workpiece coordinate system
G02	Clockwise arc interpolation	G65	Non-modal calls to macro code
G03	Counterclockwise arc interpolation	G71	Axial rough car cycle

G04	Time out	G72	Radial rough car cycle
G17	XY plane selection	G73	Closed cutting cycles
G18	XZ plane selection	G70	Finishing cycle
G19	ZY plane selection	G74	Axial grooving cycle
G10	The data entry method is valid	G75	Radial grooving cycle
G11	Cancels the data entry method	G76	Multiple thread cutting cycles
G20	Imperial unit selection	G80	The rigid tapping state is canceled
G21	Metric unit selection	G84	Axial rigid tapping cycle
G28	Return to machine zero	G88	Radial rigid tapping cycle
G30	Return to reference points 2, 3, and 4 of the machine	G90	Axial cutting cycle
G31	Jump function	G92	Thread cutting cycle
G32	Equal pitch thread cutting	G94	Radial cutting cycle
G33	Z-axis tapping cycle	G96	Constant line speed control
G34	Variable pitch thread cutting	G97	Cancel constant line speed control
G40	Eliminate tip radius compensation	G98	Feed per minute
G41	Tip radius left compensated	G99	Feeds per turn
G42	Tip radius right compensation		

1.1.3 Climate and environmental adaptability

The environmental conditions for storage, transportation and operation are as follows:

project	**Working climatic conditions**	**Storage and transport climatic conditions**
Ambient temperature	0℃～45℃	-40℃～+70℃
relative humidity	≤ 90% (non-condensing).	≤95%（40℃）
Atmospheric pressure	86KPa～106 KPa	86KPa～106 KPa
altitude	≤1000m	≤1000m

1.1.4 Power adaptability

It can operate normally under the following AC input power supply conditions. Voltage variation: within the range of (0.85 ~ 1.1) × rated AC input voltage (AC220V); Frequency change: 49Hz ~ 51Hz continuous change.

1.1.5 Protection

Protection class is not less than IP20.

1.2 CNC system and CNC machine tool

CNC machine tool is a mechanical and electrical integration product composed of numerical control systems (Numerical Control Systems of machine tools), mechanical, electrical control, hydraulic, pneumatic, lubrication, cooling and other subsystems (components), and the machine tool numerical control system is the control core of cnc machine tools. The machine numerical control system consists of a control device (Computer Numerical Controler abbreviated as CNC), a servo (or stepper) motor drive unit, a servo (or stepper) motor, and so on.

The working principle of CNC machine tool: according to the processing process requirements to write the processing program (hereinafter referred to as the program) and enter the CNC, CNC according to the processing program to the servo (or stepper) motor drive unit to issue a motion control code, servo (or stepper) motor through the mechanical transmission mechanism to complete the feed movement of the machine tool; The logic control codes such as spindle start and stop, tool selection, cooling, lubrication and so on in the program are transmitted by CNC to the machine tool electrical control system, and the machine tool electrical control system completes the control of input and output devices such as buttons, switches, indicators, relays, and contactors. At present, the electrical control of machine tools usually uses a programmable logic device (Programable Logic Controler referred to as PLC), PLC has the advantages of small size, convenient application and high reliability. It can be seen that motion control and logic control are the main control tasks of CNC machine tools.

Lathe CNC has both motion control and logic control functions, which can complete the two-axis motion control of CNC lathes, and also has a built-in PLC function. According to the input and output control requirements of the machine tool, the PLC program (ladder diagram) can be written to achieve the required machine tool electrical control requirements, which facilitates the electrical design of the machine tool and reduces the cost of CNC machine tools.

The software that realizes the CNC control function of the lathe is divided into two modules: system software (hereinafter referred to as NC) and PLC software (hereinafter referred to as PLC), and the NC module completes the control of display, communication, editing, decoding, interpolation, acceleration and deceleration The PLC module completes ladder interpretation, execution, and input and output processing.

Lathe CNC has been loaded with standard PLC programs (except for special orders) at the factory, and when the functions and operating instructions are described below, the instructions involving the control functions of the PLC will be described according to the control logic of the standard PLC program, and the instructions are "standard." PLC Function" to identify. The machine tool manufacturer may modify or rewrite the PLC program, so please refer to the machine tool manufacturer's operating instructions for functions and operations controlled by the PLC.

Figure 1-1

Programming is the process of writing machining programs according to CNC-specific programming codes for the shape and size of parts, machining processes, process parameters, tool parameters and other information. CNC machining is cnc according to the requirements of the machining program, control the machine tool to complete the process of processing parts.

The process flow of CNC machining is shown in Figure 1-2.

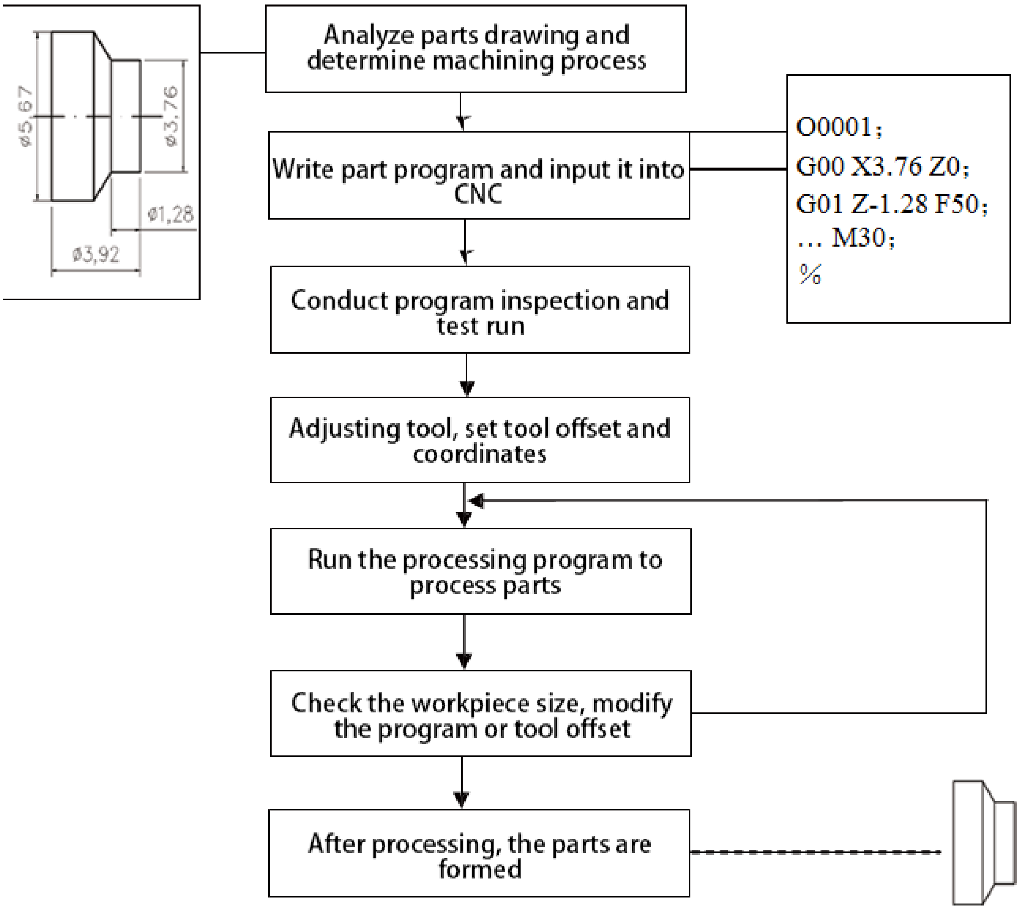


Figure 1-2

1.3 Programming Basics

1.3.1 Axis definition

The figure on the right is a schema

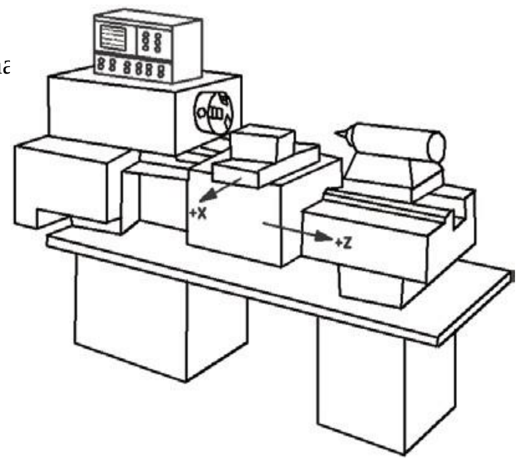
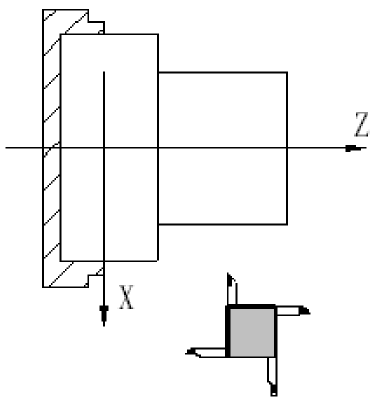


Figure 1-3

The lathe uses a Rectangular coordinate system composed of X axis and Z axis, the X axis is perpendicular to the spindle axis, the Z axis is parallel to the spindle axis, the direction of approaching the workpiece is negative, and the direction of leaving the workpiece is positive.

According to the relative position of the tool holder and the spindle of the machine tool, the CNC lathe has a front tool seat coordinate system and a rear tool seat coordinate system, Figure 1-4 is the coordinate system of the front tool holder, and Figure 1-5 is the coordinate system of the rear tool seat. As can be seen from the figure, the X-axis direction of the front and rear toolholder coordinate systems is exactly opposite, while the Z-axis direction is the same. In later illustrations and examples, the front tool holder coordinate system is used to illustrate the application of programming.



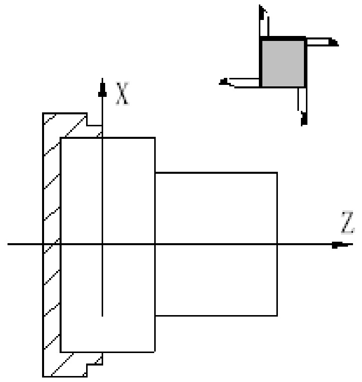


Fig. 1-4 Coordinate System of the Front Toolholder Figure 1-5 The Coordinate System of the Rear Toolholder

1.3.2 Machine coordinate system, machine zero and machine reference point

The machine coordinate system is the reference coordinate system for coordinate calculation by CNC and is the coordinate system inherent in the machine tool. The machine zero point is a fixed point on the machine tool, determined by the zero switch or the zero switch installed on the machine tool. Typically, the zero-return switch is installed at the maximum stroke in the positive direction of the X and Z axes. The machine reference point is the position after the value of the machine zero offset data parameters No 114 and No 115. When the data parameters No 114 and No 115 are set to 0, the machine reference point coincides with the machine zero. The coordinates of the machine tool reference points are the values set by the data parameters No 120 and No 121. Performing the machine machine return to zero, G28 code back to zero operation is the return machine reference point position. After performing the machine tool zero operation and returning to the machine tool reference point, the lathe establishes the machine coordinate system with the values set by No 120 and No 121 as the reference point. Reference points 2, 3, and 4 for machine tools are detailed in Section 3.10 of this article.

Note: If the zero switch is not installed on the lathe, do not perform the machine tool zero operation, otherwise it may cause the movement to exceed the stroke limit and mechanical damage.

1.3.3 Workpiece coordinate system and program zero

The workpiece coordinate system is a Cartesian coordinate system set according to the part drawing, also known as the floating coordinate system. When the part is clamped to the machine tool, the absolute coordinate of the current position of the tool is set with G50 according to the size of the workpiece, and the workpiece coordinate system is established in CNC. Usually the Z axis of the workpiece coordinate system coincides with the spindle axis, and the X axis is located at the beginning or tail end of the part. The workpiece coordinate system is valid once established until it is replaced by a new workpiece coordinate system.

The current position of the workpiece coordinate system set with the G50 is called the program zero point, and it is returned to this position after performing the program zero operation.

Note: If the workpiece coordinate system is not set with **G50** after power-on**, do not perform the operation of returning to program zero, otherwise an alarm will be generated.**

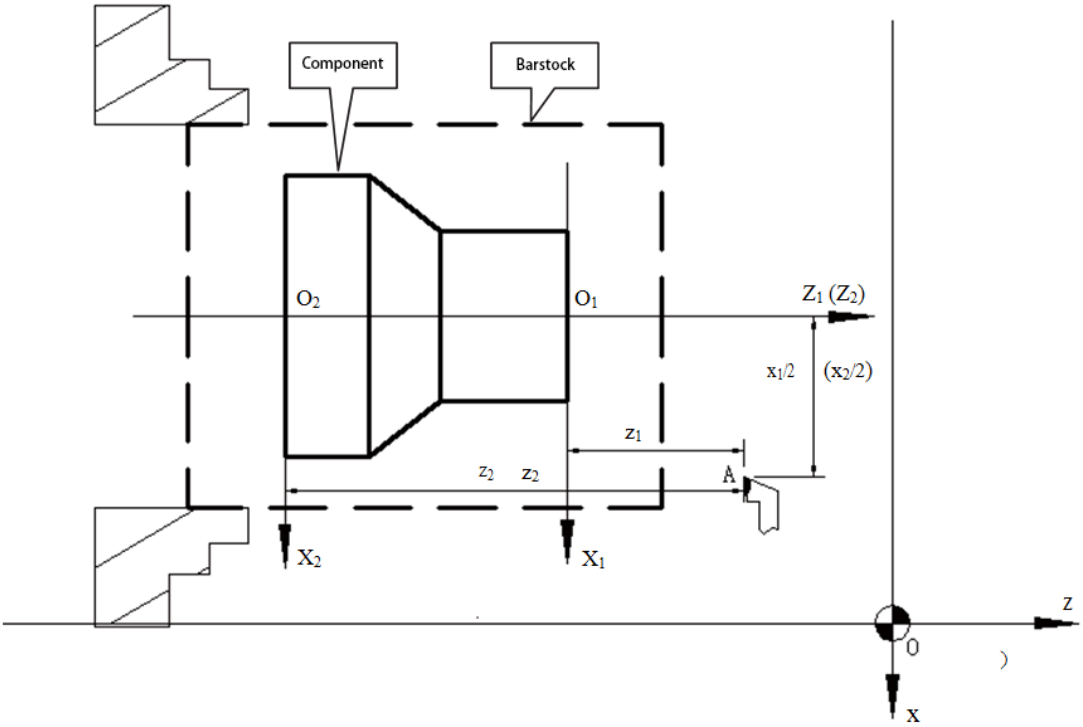


Figure 1-6

In the figure, XOZ is the coordinate system of the machine tool, X_1O_1Z1 is the workpiece coordinate system of the X axis at the end of the workpiece, X_2 O_2Z_2 is the workpiece coordinate system with the X axis at the end of the workpiece, O is the zero point of the machine tool, A is the tip, A The coordinates in the above three-coordinate system are as follows:

The coordinates of point A in the coordinate system of the machine tool are (x,z);

The coordinates of the A point in the X1 O $_1$ Z1 coordinate system are (X1, Z1);

The coordinates of the point A in the X2 O2 Z2 coordinate system are (X2, Z2);

1.3.4 Interpolation functions

Interpolation refers to the simultaneous movement of 2 or more axes, and the trajectory of the motion synthesis conforms to a determined mathematical relationship, constituting a two-dimensional (plane) or three-dimensional (space) contour, this motion control method is also known as contour control. The axis of motion controlled during interpolation is called the linkage axis, and the amount of movement, direction of movement and speed of the linkage axis are controlled simultaneously throughout the movement process to form the required synthetic motion trajectory. Only control the end point of the motion of 1 axis or multiple axes, and do not control the motion trajectory of the motion process, this kind of motion control is called positioning control.

The X and Z axes of the lathe are linkage axes and belong to the 2-axis linkage CNC. The lathe has linear, circular arc and thread interpolation functions.

Straight Line Interpolation: The synthetic motion trajectory of the X and Z axes is a straight line from the start to the end.

Arc interpolation: The synthetic motion trajectory of the X and Z axes is an arc from the start point to the end point with a radius specified by R or a circle center specified by I and K.

Thread interpolation: The angle of spindle rotation determines the amount of movement of the X axis or Z axis or two axes, so that the tool forms a spiral cutting trajectory on the surface of the rotary body workpiece rotating with the spindle to achieve thread turning. In the thread interpolation method, the feed shaft follows the rotational movement of the spindle, the spindle rotates for one week, and the long axis of the thread cutting moves a pitch, and the short axis is interpolated in a straight line with the long axis.

Example:

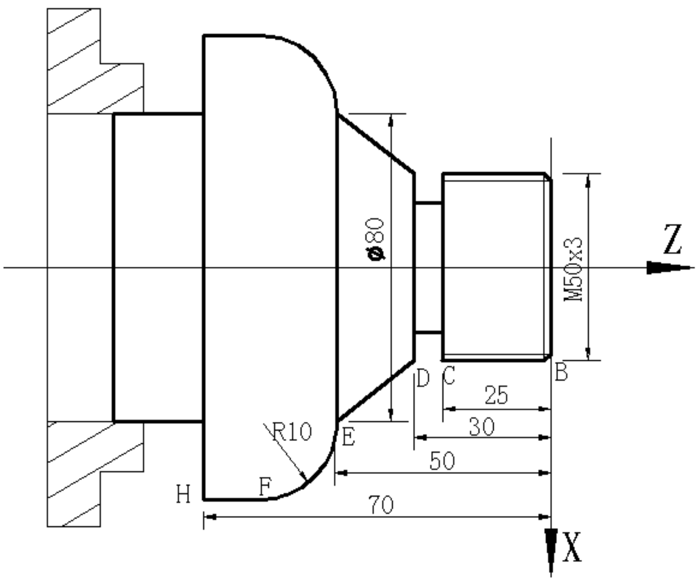


Figure 1-7

...

G32 W-27 F3；　　　　　（B → C；　Thread interpolation).

G1 X50 Z-30 F100；

G1 X80 Z-50；（D → E； Straight line interpolation).
G3 X100 W-10 R10；（E → F； Arc interpolation).
...
M30；

1.3.5 Absolute and relative coordinate programming

When writing a program, the coordinate value of the given trajectory end point or target location is required, which can be divided into three programming methods according to the type of programming coordinate value: absolute coordinate programming, relative coordinate programming and mixed coordinate programming.

Programming the absolute coordinate values of the X and Z axes (denoted by X and Z) is called absolute coordinate programming;

Programming using the relative displacement of the X and Z axes (expressed in U, W) is called relative coordinate programming;

The lathe allows absolute programming of coordinate values and relative displacements in the X and Z axes of the same program segment, respectively, called mixed coordinate programming.

Example: A → B line interpolation.

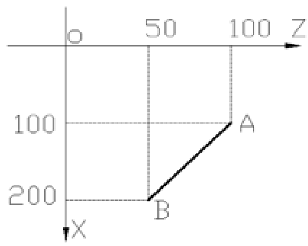


Figure 1-8

Absolute coordinate programming: G01 X200. Z50. ；
Relative coordinate programming: G01 U100. W-50. ；
Hybrid coordinate programming: G01 X200. W-50. ； or G01 U100. Z50. ；

Note: When there are simultaneous instruction addresses X, U, Z, W in a program segment**, the absolute coordinate programming addresses X and Z are valid.**

For example: G50 X10. Z20. ；
G01 X20. W30. U20. Z30. ； The end coordinate of this program segment is (X20, Z30).

1.3.6 Diameter programming and radius programming

According to the programming X axis coordinate value in diameter value or radius value input can be divided into: diameter programming, radius programming.

Diameter programming: When the Bit2 bit of the status parameter NO.001 is 0, the programmed value of the X axis in the program is entered as the diameter value, and the coordinates of the X axis are displayed as the diameter value.

Radius programming: When the Bit2 bit of the status parameter NO.001 is 1, the programmed value of the X axis in the program is entered as the radius value, and the coordinates of the X axis are displayed as the radius value.

The addresses related to the settings for diameter programming or radius programming are as follows:

address	illustrate	Diameter programming	Radius programming
X	X-axis coordinates	Diameter representation	Radius representation
X	G50 sets the X axis coordinates	Diameter representation	Radius representation
U	X-axis moves coordinates	Diameter representation	Radius representation
U	X-axis finishing allowance in G71, G72, G73 codes	Diameter representation	Radius representation
R	The amount of cut-to-finish retract in G74	Diameter representation	Diameter representation

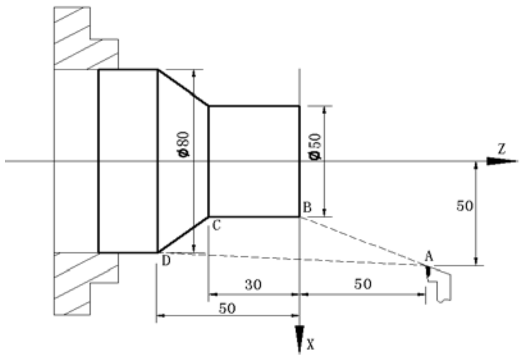
In addition to the addresses listed in the table above, the other addresses, data, such as the radius of the arc, the taper of the G90 and other X-axis programming values are entered as radius values, regardless of the diameter programming or radius programming settings.

Note: Diameter programming is used in the instructions later in this user manual, unless otherwise specified.

1.4 Composition of the Program

In order to complete the automatic machining of parts, users need to write part programs (referred to as programs) in the programming format of CNC. The CNC execution program completes the machine feed movement, spindle start and stop, tool selection, cooling, lubrication and other control, so as to achieve the processing of parts.

Program example:



00001	;	(Program name)
N0005	G0 X100 Z50;	(Quickly locate to point A)
N0010	M12;	(Clamping the workpiece)
N0015	T0101;	(Change the No.1 tool, perform the No.1 tool offset)
N0020	M3 S600;	(Start the spindle, spindle speed 600R /min)
N0025	M8	(Coolant On)
N0030	G1 X50 Z0 F600;	(Close to point B at 600mm/min)
N0040	W-30 F200;	(Cut from point B to point C)
N0050	X80 W-20 F150;	(Cut from point C to point D)
N0060	G0 X100 Z50;	(Quickly return to point A)
N0070	T0100;	(Cancel tool offset)
N0080	M5 S0;	(Stop the spindle)
N0090	M9;	(Coolant Off)
N0100	M13;	(Release the workpiece)
N0110	M30;	(At the end of the program, turn off the main shaft and coolant)
N0120	%	

After performing the above procedure, the tool will go out of the trajectory of A → B → C → D → A.

1.4.1 General structure of the program

A program consists of several lines of program segments that begin with "OXXXX" (program name) and end with a "%" sign. The program segment begins with the program segment number (which can be omitted) and begins with "; "Consists of several code words at the end. The general structure of the program is shown in Figures 1-10.

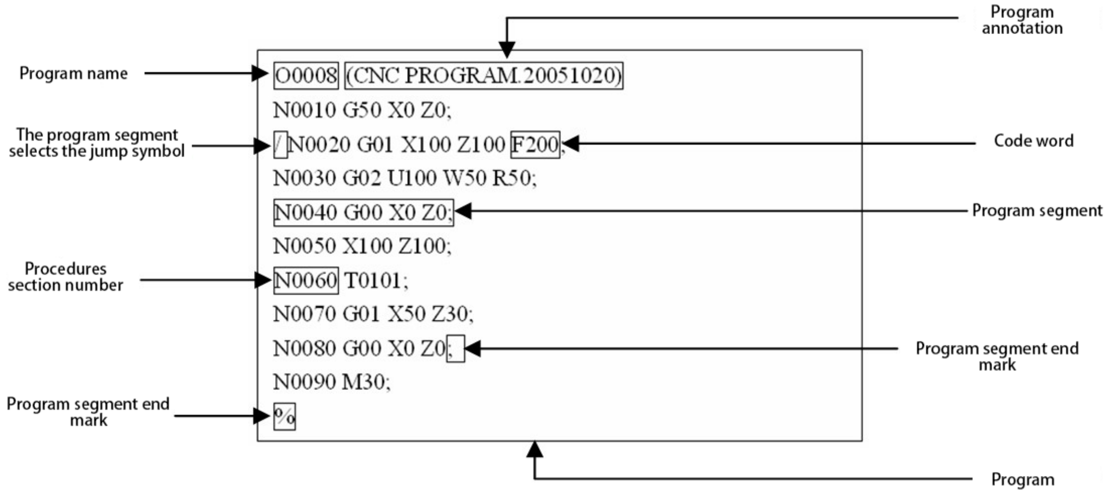
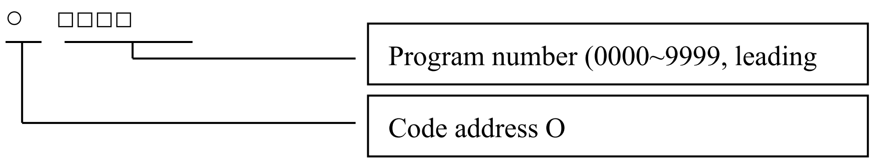


Figure 1-10 The general structure of the program

Program name

The lathe can store up to 400 programs, and in order to distinguish between the programs, each program has a unique program name (the program name is not allowed to be repeated), and the program name at the beginning of the program consists of O and the four digits after it.



Code word

A code word is a basic unit of code used to command a CNC to complete the control function, and the code word consists of an English letter (called a code address) and a numeric value (called a code value, a signed number or an unsigned number). Code addresses dictate the meaning of the code values that follow, and the same code address may have different meanings in the case of different code word combinations. Table 1-2 is a list of all code words for lathes.

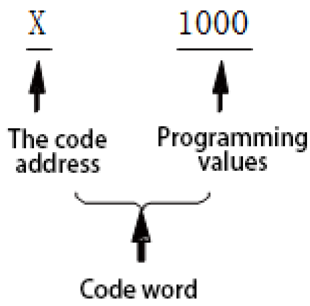


Table 1-2 List of code words

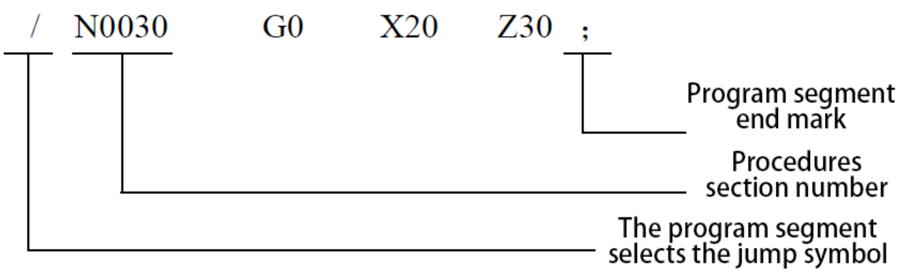
Code address	The range of programmed values	Functional significance	unit
O	0～9999	Program name	
N	0～999999	The program segment number	
G	00～99	Prepare the feature	
X	-999999.9999～999999.9999	X-axis coordinates	Related to the public/imperial system
	0～99999.999（S）	Pause time	
Z	-999999.9999～999999.9999	Z coordinates	Related to the public/imperial system
Y	-999999.9999～999999.9999	Y-axis coordinates	Related to the public/imperial system
U	-999999.9999～999999.9999	X-axis increments	Related to the public/imperial system
	0～99999.999（S）	Pause time	
	-99999.999～99999.999	X-axis finishing allowance in G71, G72, G73 codes	Related to the public/imperial system
	0.001～99.999	Depth of cutting in G71	Related to the public/imperial system
	-999999.9999～999999.9999	X-axis retract distance in G73	Related to the public/imperial system
W	-999999.9999～999999.9999	Z-axis increments	Related to the public/imperial system

	0.001～99.999	Depth of cutting in G72	Related to the public/imperial system
	-999999.9999～999999.9999	X-axis finishing allowance in G71, G72, G73 codes	Related to the public/imperial system
W	-999999.9999～999999.9999	Z-axis retract distance in G73	Related to the public/imperial system
V	-999999.9999～999999.9999	Y-axis increments	Related to the public/imperial system
R	-999999.9999～999999.9999	The radius of the arc	Related to the public/imperial system
R	0～99.999	G71, G72 cyclic retract volume	Related to the public/imperial system
R	1 ~ 9999 (times).	G73 rough car cycle times	
R	0～99.999	G74, G75 cutting retraction tool volume	Related to the public/imperial system
R	0～99999.999	The amount of retracted tool at the end of the cut in the G74 and G75	Related to the public/imperial system
R	0～99.999	G76 finishing allowance	Related to the public/imperial system
R	-999999.9999～999999.9999	G90, G92, G94, G96 medium taper	Related to the public/imperial system
I	-999999.9999～999999.9999	The center of the arc is relative to the starting point in the X-axis vector	Related to the public/imperial system
I	0.06~25400 (teeth/inch).	Number of inch thread teeth	
K	-999999.9999～999999.9999	The center of the arc is relative to the starting point in the Z-axis vector	Related to the public/imperial system
F	0～8000（mm/min）	Feed rate per minute	
F	0.0001～500（r/min）	Feed speed per revolution	
F	0.001～500（mm）	Feed speed per revolution	
S	00～9999（r/min）	Spindle speed is determined	
S	00～04	Multi-stop spindle formulation	

T	01～32	Tool function	
M	00～99	Accessibility output, program execution flow	
M	0000～9999	Subroutine calls	
P	0～9999999（0.001S）	Pause time	
P	0000～9999	Invoke the subroutine number	
P	00～99	The number of times the subroutine was invoked	
P	0000～9999999	X-axis cycle movement in G74 and G75	Related to the public/imperial system
P		Thread cutting data in G76	
P	0000～9999	The compound cycle finishing program segment starts with the program segment number	
Q	0000～9999	The end of the compound cycle finishing program segment number	
Q	0000～9999999	The amount of Z-axis cycle movement in G74 and G75	Related to the public/imperial system
Q	1～9999999	The first cut in G76	Related to the public/imperial system
Q	1～9999999	Minimum cutthrough in G76	Related to the public/imperial system
Q	0～360000	The starting angle in G32 refers to the offset angle between the spindle turn signal and the starting point of the thread cutting	
H	01～99	Operators in G65	

Program segments

The program segment consists of several code words, starting with "; "The end is the basic unit by which the CNC program runs. between segments with the character "; "Separate, as used in this manual"; " indicates. An example is as follows:



Several code words can be entered in a program segment, and no code words are allowed and only "; " sign (EOB key) terminator.

In the same program segment, except for N, G, S, T, H, L, etc., other addresses can only appear once. Otherwise, an alarm is generated (the code word is repeated in the same program segment). When N, S, T, H, L codewords are repeatedly entered in the same program segment, the last codeword of the same address is valid. When the same set of G codes is repeatedly entered in the same program segment, the last G code is valid.

The program segment number

The program segment number consists of address N and the next six digits: N000000 ~ N999999, and the leading zero can be omitted. The program segment number should be at the beginning of the program segment, otherwise it is not valid.

The program segment number can not be entered, but the target program segment of the program call and jump must have the program segment number. The order of the program segment numbers may be arbitrary, and its intervals may not be equal, in order to facilitate the search and analysis of the program, it is recommended that the program segment numbers be increased or decreased in programming order.

If you set "Auto Sequence" to "On" on the switch settings page, an incrementing program segment number is automatically generated when the program segment is inserted, and the program segment increment is set by the parameter No 42 .

The program segment selects the skip character

If a program segment is not executed when the program is executed (and you do not want to delete the program segment), insert a "/" in front of the program segment and turn on the program segment hop switch 

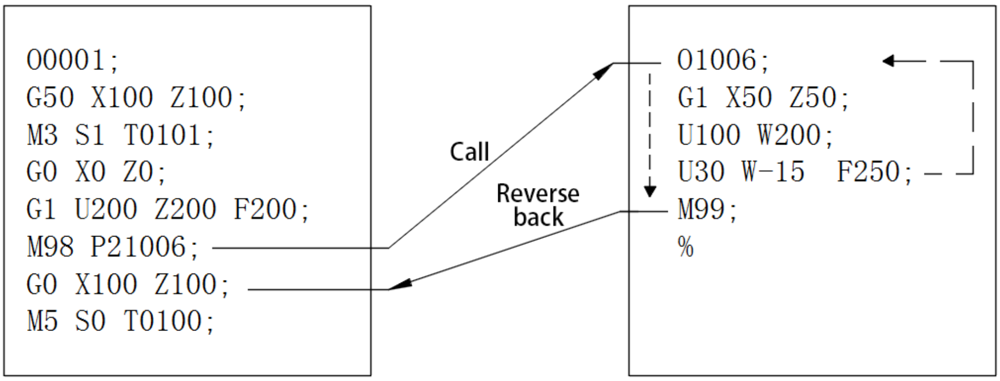
. This segment is skipped and not executed when the program is executed. If the segment hop switch is not turned on, the segment is executed even if there is a "/" in front of the segment.

Program terminator

"%" is the end of the program file, and when communicating with the program, "%" is the end of communication flag. When creating a new program, the CNC automatically inserts "%" at the end of the program.

1.4.2 Main programs and subroutines

In order to simplify programming, when the same or similar processing trajectory and control process need to be used multiple times, the program instructions in this part can be edited to be called by a separate program. The program that calls the program is called the main program, and the program that is called (ending in M99) is called a subroutine. Subroutines occupy the same program capacity and storage space of the system as the main program, and the subroutines must have their own independent program names, and the subroutines can be called by any other main program or run independently. After the subroutine ends, it returns to the main program to continue execution, as shown in the following figure.



Main program   Subroutine

1.5 Operation of the Program

1.5.1 The order in which the program runs

The currently open program must be run in automatic mode, the lathe system cannot have 2 or more programs open at the same time, so only one program can be run at any one time. When you open a program, the cursor is at the beginning of the first program segment and can be moved in the edit mode. In the state of operation stop in the automatic operation mode, the operation of the program is started 

from the program segment where the current cursor is located with a cyclic start signal (the key of the machine panel or an external cyclic start signal), usually in the order in which the program segment is written, until the M02 or M30 code is executed, and the program operation stops. The cursor moves as the program runs, always at the beginning of the line of the current program segment. The order or state of programs running changes in the following situations:

* When the key or emergency stop button is pressed while the program 

is running, the program operation is terminated;

* WHEN the program runs, a CNC alarm or PLC alarm is generated, and the program operation is terminated;

* The operation mode of the program is switched to the input and editing operation mode, the program runs a single-stage stop (after running the current program segment, the program operation is suspended), switch to the automatic operation mode, and then press the 

key or the external loop to start the signal when the signal is connected, start the program operation from the program segment where the current cursor is located;

* The operation mode of the program is switched to other operation modes, and the program operation stops and switches to the automatic operation mode;

* When the program is running and the key is pressed 

or the external pause signal is disconnected, the program is running and paused, and then the key is pressed 

or the external loop starts the signal to turn on, and the program continues to run from the position where it stopped;

* When the single-segment switch is turned on, the program operation is paused after the end of each program segment, and it is necessary to press the 

key again or when the external loop start signal is turned on, and continue to run from the next program segment;

* The program segment selection and jump switch is turned on, and the program segment with "/" in front of the program segment is skipped and not executed;

* When executing G65 jump code, go to the jump target segment to run;

* The program running order of the G70 ~ 73 composite loop code is relatively special, see Chapter 3 "G Code" of this article;

* When executing M98 code, call the corresponding subroutine or macro program to run; When the subprogram or macro program runs, when the M99 code is executed, it returns the next program segment running of the calling program segment in the main program (if the M99 code specifies the returned target program segment number, it is transferred to the target program segment to run);

* When the M99 code is executed in the main program (the program is not run by a call from another program), the first paragraph of the returned program continues to run, and the current program will run repeatedly.

1.5.2 The order in which code words within a program segment are executed

A program segment can have G, X, Z, F, R, M, S, T and other code words, most of the M, S, T code words are interpreted by NC and sent to the PLC for processing, and other code words are directly caused NC processing. M89, M90, M91, M92, M93, M98, M99, M9000~M9999, and r/min, m/min The S code word for a given spindle speed is also processed directly by the NC.

When the G code is in the same program segment as M00, M01, M02, M30, the NC finishes executing G After the code, the M code is executed and the corresponding M signal is sent to the PLC for processing.

When the G code word is in the same program segment as the M98 and M99 code words, the NC executes the G code before executing these M code words (not sent M signal to PLC).

When the G code word is in the same program segment as other M, S, and T code words processed by the PLC, the PLC program (ladder) determines M , S, T code words and G code words are executed at the same time, or M, S is executed after the G code is executed 3. T code word, the order of execution of the relevant code word should be subject to the instructions of the machine tool manufacturer.

The **G**, **M**, **S, T** code words **defined by the standard PLC** program for lathe systems **are executed in the same program segment in the following order**:

M3、M4、M8、M10、M12、M32、M41、M42 , M43, M44, S □□, T □□□□ and G code words are executed at the same time;

M5, M9, M11, M13, M33 execute after executing the G code word;

M00, M01, M02, M30 execute after the execution of the other code in the current sequence segment is complete.

Chapter 2 : MST Code

2.1 M code (accessibility).

The M code consists of the code address M and the 1 to 2 digits or 4 digits thereafter, and is used to control the flow of program execution or to output the M code to the PLC.

M □□□□

Code values (00~99, 9000~9999), leading zero can be omitted.

Code address.

M89, M90, M91, M92, M93, M98, M99, M9000~M9999 are processed independently by the NC and do not output M codes to the PLC 。

M02 and M30 have been defined by NC as the end code of the program, and the M code is also output to the PLC, which can be used by the PLC program for input and output control (off spindle, off cooling, etc.).

M98, M99 as the program call code, M02, M30 as the program end code, PLC program can not change the meaning of the above code. Other M codes are output to the PLC, and the PLC program defines the code functions, please refer to the machine manufacturer's manual.

There can only be one M code in a program segment, and when two or more M codes appear in the program segment, the CNC has an alarm.

Table 2-1 List **of flow M codes that control program execution**

code	function
M00	Unconditional suspension
M01	Select Stop
M02	The program ends
M30	The program ends
M31	The number of processed parts plus 1
M89	Customize the output input (see Debugging Documentation for details).
M90	The program loop ends (refer to the Debugging Documentation for details).
M91	The program loop section begins (see Debugging Documentation for details).
M92	The program segment jumps unconditionally (refer to the Debugging Documentation for details).
M93	The program segment has a conditional jump (for details, refer to the Debugging Documentation).
M98	Call the subroutine

M99	Returned from a subroutine, if M99 is used for the end of the main program (i.e. the current program is not called by another program), the program executes repeatedly

2.1.1 Program endSumage M02

Code format: M02 or M2

Code function: in the automatic mode, the M02 code is executed, after the execution of other codes in the current program segment is completed, the automatic operation ends, the number of processed parts is increased by 1, the radius compensation of the tip is canceled, and the cursor returns to the beginning of the program (whether to return the beginning of the program is determined by the parameters).

2.1.2 Program runs at the end of M30

Code format: M30

Code function: In the automatic mode, the M30 code is executed, after the execution of other codes in the current program segment is completed, the automatic operation ends, the number of machined parts is increased by 1, the radius compensation of the tip is canceled, and the cursor returns to the beginning of the program (whether to return the beginning of the program is determined by the parameters).

When BIT4 of THE CNC status parameter NO.005 is set to 0, the cursor does not return to the beginning of the program; When the CNC status is set to 1 for BIT4 in no.005, the program is executed and the cursor immediately returns to the beginning of the program.

2.1.3 Subroutine calls M98

Code format: (1) M98 P○○ □□□□

The subroutine number (0000~9999) that was called. When the number of calls is not entered, subroutine leading 0 can be omitted; When there are input calls, the subroutine number leading 0 cannot be omitted.

Two digits, the number of calls (1 ~ 99), when the number of calls is 1, you can omit no input; When the number of calls is greater than 99, format (2) is used.

② M98 P○○○○ L□□□□

4 digits, the number of calls (1 ~ 9999), when the number of calls is 1, can be omitted without input, leading 0 can be omitted.

The subroutine number (0000~9999) that was called. Subroutine leading 0 can be omitted.

Code function: In the automatic mode, when the M98 code is executed, after the execution of other codes in the current program segment is completed, the CNC calls to execute the subprogram specified by P, and the subprogram can be executed up to 9999 times. The M98 code does not run under MDI.

2.1.4 Returns M99 from subroutines

Code format: M99 P○○○○○

Returns the program segment number (0000~9999) that the main program will execute, and the leading zero can be omitted

Code function: (in subprograms) After the execution of other codes of the current program segment is completed, the program segment specified by P in the main program is returned to continue execution, and when P is not entered, the next program segment of the M98 code that calls the current subroutine in the main program is returned to continue execution. If M99 is used for the end of the main program (i.e. the current program is not executed by another program call), the current program will be executed repeatedly. The M99 code does not run under MDI.

Example: Figure 2-1 shows the execution path to call a subroutine (there is a P code word in M99). Figure 2-2 shows the execution path that calls the subroutine (no P code word in M99).

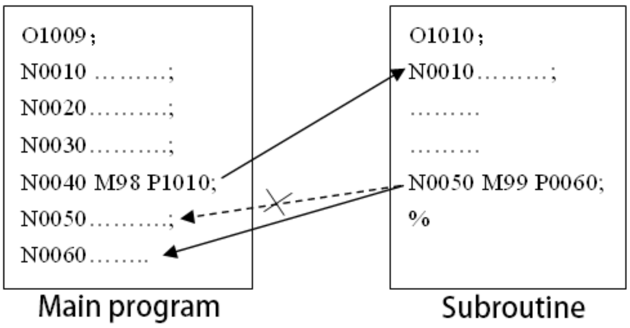


Figure 2-1

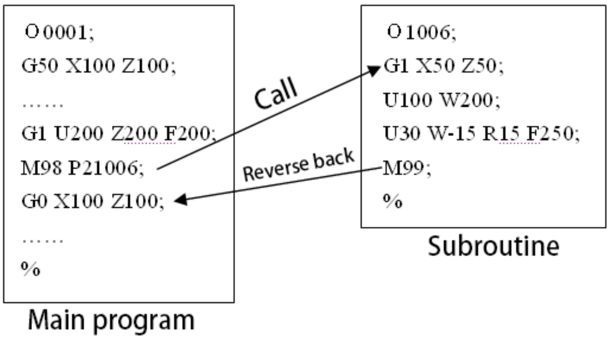


Figure 2-2

Lathe systems can call quadruple subroutines, i.e. other subroutines can be called within subprograms (Figure 2-3).

Main Program Subroutines Subroutines Subroutines Subroutines

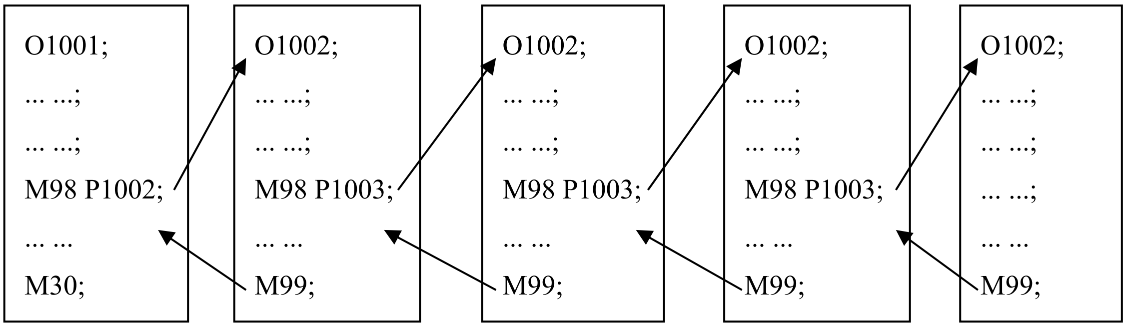


Figure 2-3 Subroutine nesting

Note 1: M99 uses a loop (infinite loop) that can implement the main program when the main program is used;

Note 2: When the M99 P_ is used in **the main program, the main program jumps to P_ at the specified line number N;**

For example:

N10	**N10**
N20	**N20**
N30 M99 P50	**N30**
N40	**N40**
N50	**N50 M99 P30**
N60	**N60**

;

2.1.5 M code defined by a standard PLC ladder

In addition to the above codes (M02, M30, M89, M90, M91, M92, M93, M98, M99), the other M codes are composed of PLC Definition. The following is the M code defined by the standard PLC, the lathe system CNC is used for machine tool control, the function, significance, control timing and logic of the M code are subject to the instructions of the machine tool manufacturer.

The M code defined by the standard PLC ladder

M code	definition	remark
M00	Unconditional suspension	Pause when the program runs to M00
M01	Select Pause to stop	The program runs to M01 and pauses when the system has the option to stop, or run down
M02	The program ends	
M03	The spindle is rotating	Interlocked, power on default M05
M04	Spindle inversion	
*M05	Spindle stops	
M06	retain	
M08	Coolant	Interlocked, power on the default M09
*M09	Turn off the cooling	
M10	Tail seat forward	Interlock
M11	The tailstock is receding	
M12	Chuck clamping	Interlock
M13	The chuck is released	
M14	Spindle positioning toggle position mode	Interlock
*M15	Spindle switching speed mode	
M20	Spindle clamping	Interlock
*M21	Spindle loosened	
M22	retain	Leave the M codes as invalid function codes
M23		
M24		
M25		
M26		
M27		
M28		

M29	Rigid tapping opens	It is possible to switch speed mode disconnect using the M15
M30	The program ends	
M31	The number of processed parts plus 1	
M32	Lubrication output	Interlock
*M33	Lubrication off	
M34	Protective door output	Interlock
*M35	The protective door is closed	
M36	Chip conveyor output	Interlock
*M37	The chip conveyor is turned off	
M38	Work light output	Interlock
*M39	The work light is off	
M40	retain	Leave the M codes as invalid function codes
M41	Spindle shift 1st gear	Interlock
M42	Spindle shift 2nd gear	
M43	Spindle shift 3rd gear	
M44	Spindle shift in 4 gears	
M45	retain	Leave the M codes as invalid function codes
M46		
M47		
M48		
M49		
M50	The 3rd axis power head is forward	Interlock
M51	The 3rd axis power head is reversed	
*M52	The 3rd axis power head stops	
M53	The 4th axis power head is forward	Interlock
M54	The 4th axis power head is reversed	
*M55	The 4th axis power head stops	

M56	retain	Leave the M codes as invalid function codes
M57		
M58		
M59		
M60		
M61		
M62		
M63	The second spindle is forward	Interlocked, power on default M65
M64	The second spindle is reversed	
*M65	The second spindle stops	
M66	retain	Leave the M codes as invalid function codes
M67		
M68		
M69		
M70	Customize the K1 output	Interlock
*M71	Custom K1 off	
M72	Customize the K2 output	Interlock
*M73	Custom K2 off	
M74	Customize the K3 output	Interlock
*M75	Custom K3 off	
M76	Customize the K4 output	Interlock
*M77	Custom K4 off	
M78	**Recovery feed multiplier 100%.**	Interlock
***M79**	**Restore the original feed rate**	
M80	retain	Leave the M codes as invalid function codes
M81		
M82		
M83		

M84		
M85		
M86		
M87		
M88		
M89	Customize the output	Multi-format programming
M90	The program segment ends repeatedly	Interlocked, after using M91 C_ need to use M90 end
M91 C	The program segment starts repeatedly	
M92	Conditional jump	
M93	Unconditional jumps	
M94	retain	Leave the M codes as invalid function codes
M95		
M96		
M97		
M98	Call the subroutine	When the L number of times is not written, it is called 1 time by default
M99	Subroutine ends	When the main program is written, the program is an infinite loop
M9000—M9999		A program that calls O9000-O9999 single time

Note: The code with the label "*" defined by the standard **PLC is valid at power-up; For specific code control logic, refer to debugging documentation.**

2.1.6 Program stopSpeed M00

Code format: M00 or M0

Code function: After executing the M00 code, the program runs to stop, displaying the word "pause", press the loop start key, and the program continues to run.

2.1.7 The program selects Stop M01

Code format: M01 or M1

Code function: in the automatic, input mode is effective, press the select stop button to make the selection stop button 

indicator light, it means that the choice stop state, at this time after the execution of the M01 code, the program runs to stop, displaying the word "pause", press the loop start key, the program continues to run. If the program selection stop switch is not turned on, the program does not pause even if the M01 code is run.

2.1.8 Counterclockwise, clockwise and spindle stop control M03, M04 and M05

Code format: M03 or M3;
M04 or M4;
M05 or M5;
Code function: M03: counterclockwise rotation;
M04: Clockwise;
M05: Spindle stop.

Note: The control timing and logic of M03, M04, and M05 defined by the standard **PLC are detailed in the Debug Documentation or the third installed connection.**

2.1.9 Cooling pump control M08, M09

Code format: M08 or M8;
M09 or M9;
Code function: M08: cooling pump on;
M09: Cooling pump off.

Note: The control timing and logic of M08 and M09 defined by the standard **PLC** are detailed in **the Debug Documentation or the third "Installation Connection".**

2.1.10 Tailstock control M10, M11

Code format: M10;
M11;
Code function: M10: tail seat entry;
M11: Tail seat retreat.

Note: The control timing and logic of M10 and M11 defined by the standard **PLC are detailed in the Debug Documentation or the third "Installation Connection".**

2.1.11 Chuck control M12, M13

Code format: M12;
M13;
Code function: M12: chuck clamping;
M13: The chuck is released.

Note: The control timing and logic of M12 and M13 defined by the standard **PLC are detailed in the Debug Documentation or the third "Installation Connection"**.

2.1.12 Spindle position/speed control switching M14, M15

Code format: M14;
M15;

Code function: M14: Spindle switches from speed control mode to position control mode;

M15: Spindle switches from position control to speed control.

Note: The control timing and logic of M14 and M15 defined by the standard **PLC are detailed in the Debug Documentation or the third "Installation Connection".**

2.1.13 Spindle clamping/loosening controls M20, M21

Code format: M20;

M21；

Code function: M20: spindle clamping;

M21: Spindle loosened.

Note: The control timing and logic of the M20 and M21 defined by the standard **PLC are detailed in the Debug Documentation or the third "Installation Connection"**.

2.1.14 The 2nd spindle position/speed control switches M24, M25

Code format: M24;

M25；

Code function: M24: The 2nd spindle switches from speed control mode to position control mode;

M25: The 2nd spindle switches from position control to speed control.

Note: The control timing and logic of M24 and M25 defined by the standard **PLC are detailed in the Debugging Documentation or the third "Installation Connection" function is not enabled**.

2.1.15 Lubrication control M32, M33

Code format: M32;

M33；

Code function: M32: lubrication pump on;

M33: Lubrication pump off.

Note: The control timing and logic of M32 and M33 defined by the standard **PLC are detailed in the Debugging Documentation or the third "Installation Connection".**

2.1.16 Spindle auto-shift M41, M42, M43, M44

Code format: M4n; (n=1、2、3、4)

Code function: When M4n is executed, the spindle is changed to the nth gear

Note: The control timing and logic of M41, M42, M43, and **M44 defined by** the standard **PLC are detailed in the Debug Documentation or the third Install connection.**

2.1.17 The second spindle is counterclockwise, clockwise and spindle stop controls M63, M64 and M65

Code format: M63;

M64;

M65;

Code function: M63: counterclockwise rotation;

M64: Clockwise;

M65: Spindle stopped.

Note 1: The control timing of **M63, M64, and M65 as defined by** the standard **PLC is the same as that of M03, M04, M05。**

Note 2 to entry: This feature is only effective if the second spindle function is active.

2.2 Spindle function

S code is used to control the speed of the spindle, lathe system has two ways to control the spindle speed: spindle speed switch control mode: S□□ (2 digit code value) Code is processed by the PLC, PLC Outputs a switching signal to the machine tool to achieve a stepwise change in spindle speed.

Spindle speed analog voltage control mode: S □□□□ (4 digit code value) specifies the actual spindle speed, NC outputs 0 ~ 10V analog voltage signal to the spindle servo device or frequency converter, to achieve stepless speed regulation of spindle speed.

2.2.1 Spindle speed switching control

When the BIT4 of the status parameter NO.001 is set to 0, the spindle speed is switched. A program segment can have only one S code, and when two or more S codes appear in the program segment, the CNC has an alarm.

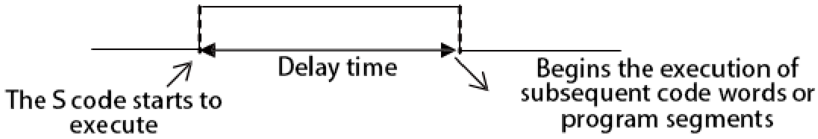
When the S code is in the same segment as the code that performs the mobile function, the order of execution is defined by the PLC program, please refer to the manual of the machine manufacturer.

When the spindle speed switch is controlled, the lathe system CNC is used for machine tool control, and the timing and logic of the S code execution should be subject to the instructions of the machine tool manufacturer. The S codes described below are defined for standard PLCs for lathe systems and are for reference only.

Code format: S □□

00~04 (leading zero can be omitted): 1~4 gear spindle speed switch control

Under the spindle speed switch control mode, after the code signal of the S code is sent to the PLC, the FIN signal is returned after delaying the time set by the data parameter NO.081, which is called the execution time of the S code.



When the CNC resets, the output states of S01, S02, S03, and S04 are unchanged.

When the CNC is powered on, the S1 ~ S4 output is invalid. Execute any of the codes in S01, S02, S03, S04, and the corresponding S signal output is valid and maintained, while canceling the remaining 3 Output of S signals. When executing S00 or S05 codes, the output of S1 ~ S4 is canceled, and only one S1 ~ S4 is valid at the same time.

2.2.2 Spindle speed analog voltage control

When bit4 of status parameter NO.001 is set to 1, the spindle speed is controlled by analog voltage.

Code format: S □□□□

0000~9999 (leading zero can be omitted): spindle

Code function: set the speed of the spindle, CNC output 0V ~ 10V analog voltage control spindle servo or frequency converter, to achieve the spindle's stepless speed change, S code value power failure is not remembered, power-up set 0.

When the spindle speed analog voltage control function is effective, there are 2 ways to input the spindle speed: S code sets the fixed speed of the spindle (r/min), and the spindle speed is constant when the S code value does not change, which is called constant speed control (G97 mode); The S code sets the tangent speed (m/min) of the tool relative to the outer circle of the workpiece, which is called constant linear speed control (G96 mode), and under the constant linear speed control mode, the spindle speed of the cutting feed follows the programming trajectory X axis The absolute value of the absolute coordinate value changes. See section 2.2.3 of this chapter for details.

The CNC has a four-speed spindle mechanical gear function, when executing the S code, the analog voltage value corresponding to the given speed is calculated according to the setting value of the highest spindle speed (output analog voltage of 10V) of the current spindle gear (corresponding to the data parameter NO.037 ~ NO.040). , and then output to the spindle servo or frequency converter, the actual speed of the control spindle is consistent with the required speed.

When the CNC is powered up, the analog voltage output is 0V, and after executing the S code, the output analog voltage value remains unchanged (unless the absolute value of the X axis absolute coordinate value is changed in the constant line speed controlled cutting feed state). After performing S0, the analog voltage output is 0V. When the CNC resets and stops, the analog voltage output remains unchanged.

2.2.3 Constant linear speed control G96, constant speed control G97

Code format: G96 S__; (S0000 ~ S9999, leading zero can be omitted)

Code function: constant line speed control is effective, given the cutting line speed (m/min), cancel the constant speed control. G96 is a modal G code, and if it is currently in the G96 mode, you can enter G96 without entering it.

Code format: G97 S__; (S0000 ~ S9999, leading zero can be omitted)

Code function: cancel the constant linear speed control, the constant speed control is effective, given the spindle speed (r/min). G97 is a modal G code, if currently in G97 mode, you can enter G97 without entering it.

Code format: G50 S__; (S0000 ~ S9999, leading zero can be omitted)

Code function: set the maximum spindle speed limit value (r/min) for constant linear speed control.

G96 and G97 are modal code words of the same group, and only one valid one can be valid. G97 is the initial code word, and the default G97 is valid when the CNC is powered on.

When the lathe turns the workpiece, the workpiece is usually rotated to the central line of the spindle axis, and the cutting point of the tool cutting workpiece can be regarded as a circular motion around the spindle axis, and the instantaneous rate of the circumferential tangent direction is called the cutting line speed (usually referred to as the linear speed). Workpieces of different materials and tools of different materials require different linear speeds.

When the spindle speed analog voltage control function is effective, the constant linear speed control function is effective. In the constant linear speed control, the spindle speed changes with the absolute value of the X axis absolute coordinate value of the programmed trajectory (ignoring the tool length compensation), the absolute value of the X axis absolute coordinate value increases, the spindle speed decreases, the absolute value of the X axis absolute coordinate value decreases, and the spindle speed increases, so that the cutting line speed remains S Code value. Cutting the workpiece using the constant linear speed control function allows the surface finish of the workpiece with changing diameter to be consistent.

Linear speed = spindle speed × | X| ×π÷1000 （m/min）

Spindle speed: r/min

| X| : The absolute value (diameter value) of the absolute coordinate value of the X axis, mm

π ≈ 3.14

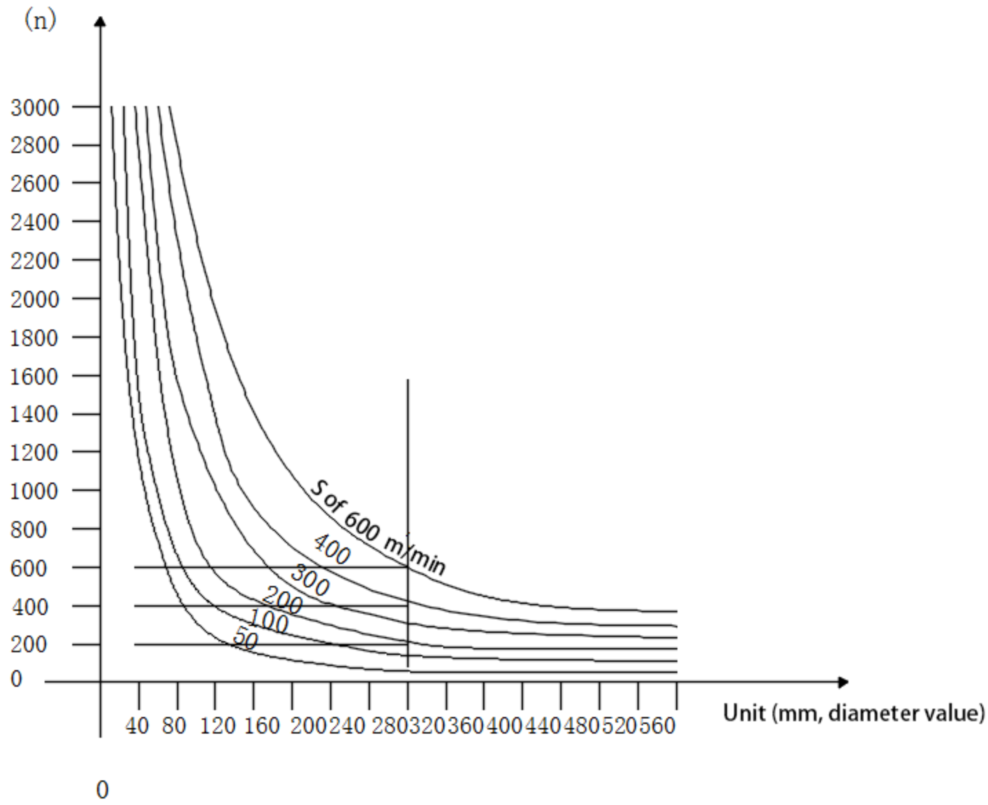


Figure 2-4

Constant linear speed control, only in the cutting feed (interpolation) process with the programming trajectory X axis absolute coordinate value of the absolute value of the change in the spindle speed, for the G00 fast movement, because the actual cutting is not carried out, the spindle speed remains unchanged during the G00 execution process, the spindle speed at this time is calculated according to the linear speed of the end of the program segment.

When controlling constant linear velocity, the Z axis of the workpiece coordinate system must coincide with the spindle axis (workpiece rotation axis), otherwise, the actual linear velocity will be inconsistent with the given linear velocity.

When the constant line speed control is effective, the G50 S__ can limit the maximum spindle speed (r/min), and when the spindle speed calculated by the line speed and the X axis coordinate value is higher than the maximum speed limit value of the limit spindle set by G50 S__, the actual spindle speed is the maximum spindle speed limit value of the main shaft. When the CNC is powered on, the maximum speed limit value of the spindle is not set, and the maximum speed limit function of the spindle is ineffective. The maximum speed limit value defined by the G50 S__ is maintained until it is reassigned, and the maximum speed limit function is valid in the G96 state and the G50 S__ in the G97 state The maximum spindle speed set does not limit, but the spindle maximum speed limit remains.

Special attention needs to be paid: when the parameter NO.043 (minimum speed of spindle when constant linear speed control) is set to 0, if G50 S0 is performed, the spindle speed will be limited to 0r/min when constant linear speed control is performed (The spindle does not rotate).

CNC parameter NO.043 is the lower limit of spindle speed when constant linear speed control, when the spindle speed calculated according to the linear speed and X axis coordinate value is lower than this value, the actual spindle speed limit is limited to the lower limit of the spindle speed.

Example:

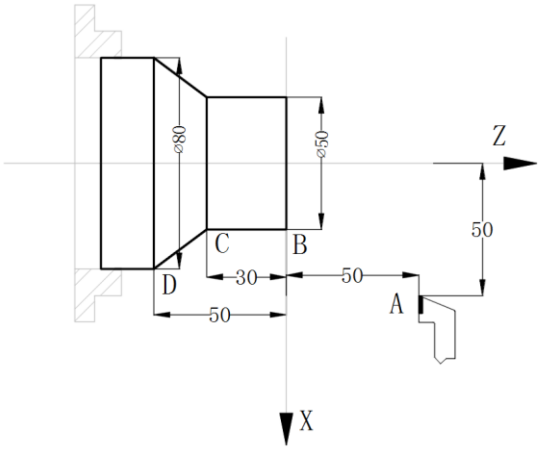


Figure 2-5

O0001; (program name).

N0010 M3 G96 S300; (Counterclockwise rotation, constant line speed control is effective, line speed is 300m/min).

N0020 G0 X100 Z50; (Quickly move to point A, spindle speed 955r/min during movement).

N0030 G0 X50 Z0; (Quickly move to point B, spindle speed 1910r/min during movement).

N0040 G1 W-30 F200; (Cutting from point B to point C, spindle speed 1910r/min in cutting).

N0050 X80 W-20 F150; (Cutting from point C to point D, spindle speed 1910r/min linear change 1194r/min).

N0060 G0 X100 Z50; (Quickly retreat to point A, spindle speed 955r/min during movement).

N0110 M30; (At the end of the program, turn off the spindle, coolant).

N0120 %;

Note 1 to entry: In the G96 state, the S value of the instruction is maintained even in the **G97 state. When returned to the G96 state, its value is restored;**

Example: G96 S50; (Cutting line speed 50m/min).

G97 S1000； (Spindle speed 1000r/min).

G96 X3000； (Cutting line speed 50m/min).

Note 2: When the machine tool is locked (the X and Z axes do not move when the **X** and **Z** axis **movement codes are executed), the constant linear speed control function is still valid;**

Note 3: When thread cutting, although the constant linear speed control function can also be effective, in order to ensure the accuracy of thread processing, do not use constant linear speed control when thread cutting, and thread cutting should be carried out under **G97;**

Note 4: When changing **from the G96 state to the G97 state, if** the **G97 segment does not have an S code (r/min), then G96 The last speed of the state is used as** the S code **for the G97** state**, i.e. the spindle speed does not change at this time;**

Note 5: When the spindle speed calculated from the cutting line speed is higher than the maximum speed of the current spindle gear (CNC parameter NO.037 ~ NO.040), the spindle speed at this time is limited to the maximum speed of the current spindle gear.

2.2.4 Spindle magnification

When the spindle speed analog voltage control mode is effective, the actual speed of the spindle can be modified with the spindle magnification, and the actual speed after the spindle magnification adjustment is limited by the maximum speed of the current gear of the spindle, and is also limited by the minimum spindle speed limit value and the maximum spindle speed limit value under the constant line speed control mode.

NC provides 8 levels of spindle magnification (50% ~ 120%, each stage changes 10%), the actual number of spindle magnification, trimming methods, etc. are defined by the PLC ladder diagram, and the use should be subject to the instructions of the machine tool manufacturer. The following is a functional description of the standard PLC ladder diagram and is for reference only.

Lathe system standard PLC ladder diagram defines a total of 8 levels of spindle magnification, the actual speed of the spindle can be adjusted in real time with the spindle magnification repair key in the range of 50% ~ 120% command speed, spindle magnification power loss memory. The spindle magnification trimming operation is detailed in the manual "Operation Instructions".

2.2.5 Multi-spindle control

The lathe system can control up to two analog spindles, with an S code to instruct any of these spindles, which spindle to choose is determined by the signal from the PLC, and each with a gear shift function.

Since the lathe system has only one spindle encoder interface, there is no encoder feedback on the second spindle and no display of spindle speed.

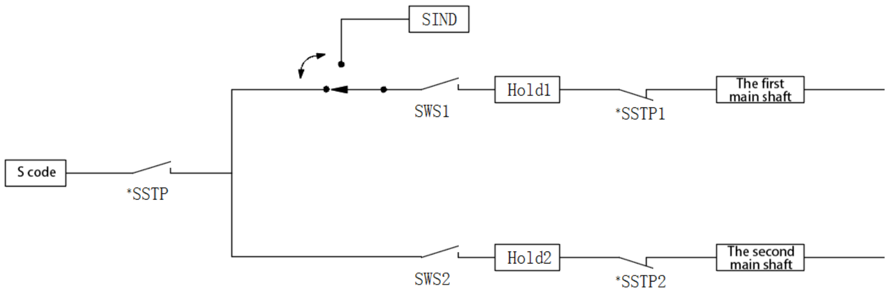
The S code is sent as a speed command to the spindle selected by the spindle selection signals (SWS1, SWS2< G25#0, G25#1), and each spindle rotates at a specified speed. If a spindle does not receive a spindle selection signal, it continues to rotate at the previous speed. This allows each spindle to rotate at different speeds at the same time. Each spindle has its own spindle stop and spindle enable signals.

When the status parameter No.001#4 is set to 1, the multi-spindle control function can be activated by MSEN (No.196#4) for 1. Spindle control has the following forms, which are determined by MSI (No.196#4). Here's how:

*** Multi-spindle control type A**

When the 1st spindle is selected by the SWS1 signal, the SIND signal is used to determine whether the spindle analog voltage is PLC or CNC controlled, R01I to The R12I signal is used to set the spindle analog voltage. These signals do not affect the 2nd spindle.

The block diagram of multi-spindle control method A is as follows:

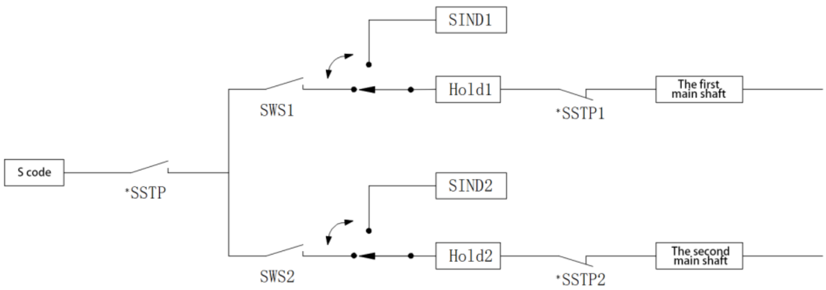


*** Multi-spindle control type B**

Each spindle has an independent SIND signal.

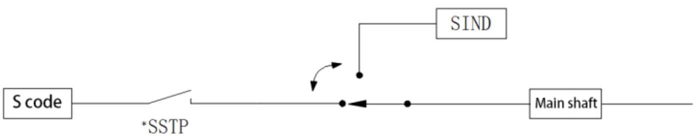
When the spindle selection signal and the SIND signal of the 1st spindle or the 2nd spindle are set to 1, each spindle is determined by the SIND signal PLC or CNC control.

The block diagram of multi-spindle control method B is as follows:



***Multi-spindle control function is not valid**

When the multi-spindle control is invalid, the control method is shown in the following figure



2.2.6 CS profile control function

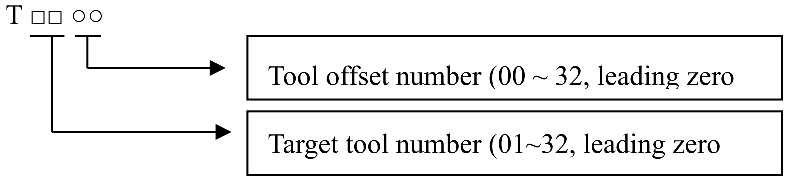
The situation of controlling the spindle speed is called spindle rotation control (rotating the spindle through the speed command), and the situation in which the position of the spindle is controlled is called the spindle profile control (rotating the spindle by moving the command). The function of profile control of this spindle is the CS profile control function. The spindle works as a servo feed shaft, rotates and positions by position movement instructions, and can be interpolated with the feed shaft. The contour curve is machined.

2.3 Tool functions

2.3.1 Tool control

The tool function (T code) of the lathe system serves two purposes: automatic tool change and tool offset. The control logic for the automatic tool change is handled by the PLC ladder diagram and the execution of the tool bias is handled by NC.

Code format:



Code function: Automatic tool holder change to the target tool number position and perform tool offset according to the tool offset number of the code. The tool offset number can be the same as the tool number, or it can be different, that is, a tool can correspond to multiple offset numbers. After performing the tool bias, and then performing T □□ 00, the CNC will reverse offset according to the current tool bias, and the CNC will change from the executed tool bias state to the uncompensated state, a process called removing the tool bias. At power-up, the tool number and tool offset number displayed by the T code are the state before the power is lost.

There can be only one T code in a program segment, and when two or more T codes appear in the program segment, the CNC generates an alarm. The position offset data of each tool (called tool bias or tool bias) is obtained by the tool operation before machining, and the tool bias is automatically performed after the T code is executed during the program operation. In this way, when editing the program, each tool is written according to the paper size of the part, and the position relationship between each tool in the coordinate system of the machine tool is not considered. If the machining size deviates due to tool wear, the tool bias can be modified according to the size deviation.

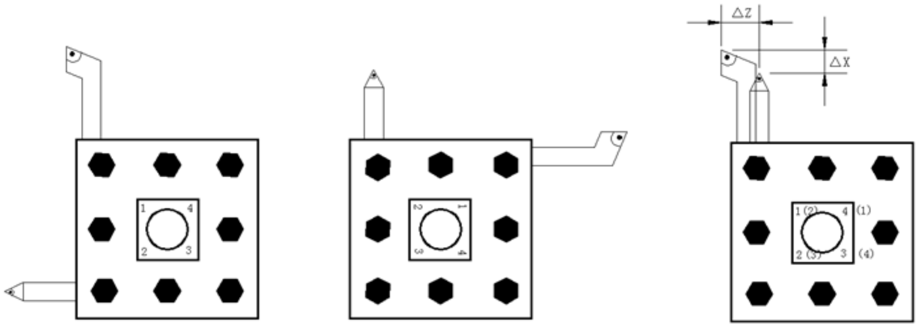


Figure 2-4 Tool offset

Tool bias is for programming trajectories, and the offset corresponding to the tool offset number in the T code is added or subtracted at the end of each program segment. The X-axis tool bias is represented by the Bit4 bit of the status parameter NO.004 using a diameter value or a radius value. The meaning of the tool offset value for the X axis using the diameter value/radius value refers to the fact that when the tool length compensation value changes, the workpiece outer diameter changes in diameter value/radius value.

Example: If the bit4 bit of the status parameter NO.004 is 0, if the tool length compensation value of the X axis is changed by 10 mm, the diameter value of the outer diameter of the workpiece is changed by 10 mm; If the bit4 bit of the status parameter NO.004 is 1, if the tool length compensation value of the X axis is changed by 10 mm, the diameter value of the workpiece outer diameter is changed 20mm。 Figure 2-5 is the process of establishing, executing and canceling the tool bias for the mobile mode.

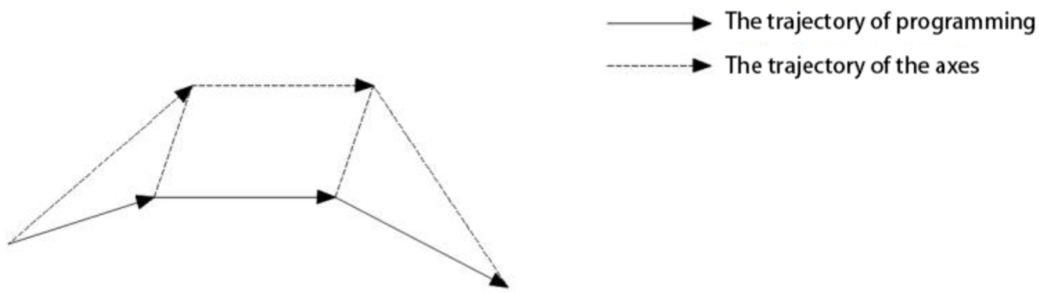


Fig. 2-5 Tool length compensation is established, executed and canceled

G01 X100 Z100 T0101; (Program 1, start of tool offset)

G01 W150; (Program 2, tool offset state)

G01 U150 W100 T0100; (Program paragraph 3, remove tool offset, execute tool).

There are two ways to perform tool bias, set by bit4 bit of status parameter NO.003:

When Bit4 = 0 , tool offset is performed in the form of tool movement;

When Bit4 = 1 , tool offset is performed by modifying coordinates;

Example:

Table 2-4

Tool offset	X	Z
00	0.000	0.000
01	0.000	0.000
02	12.000	-23.000
03	24.560	13.452

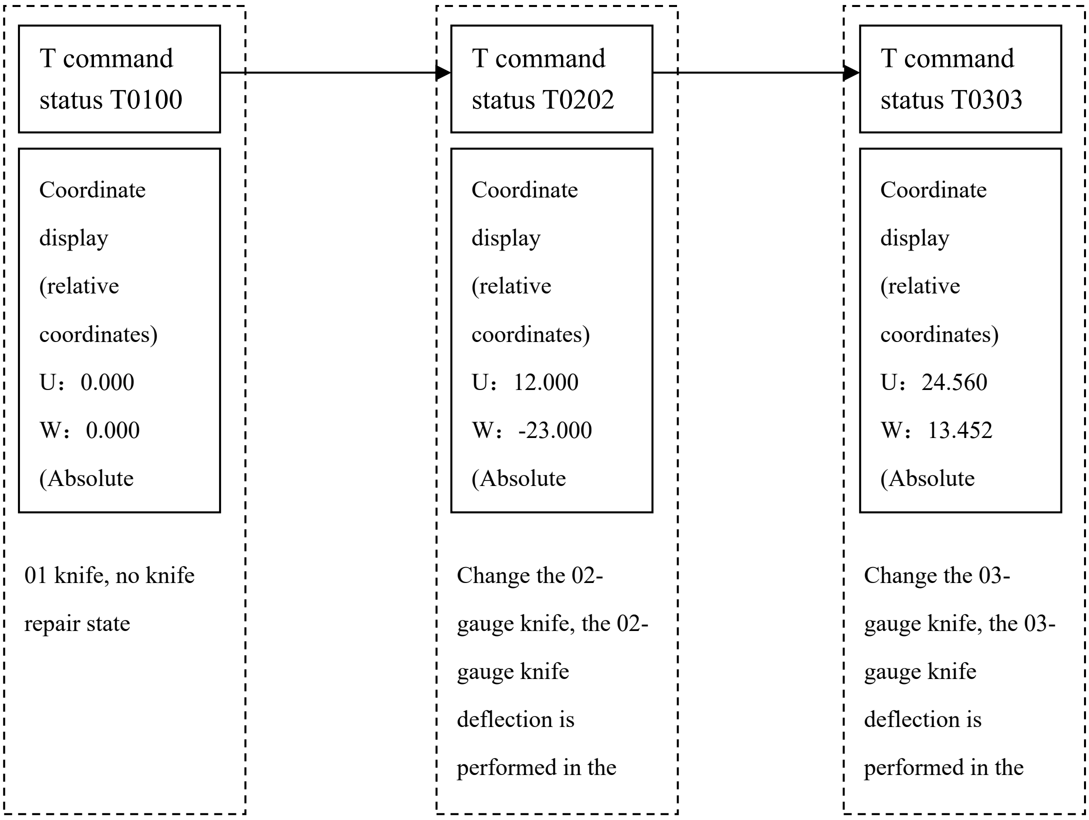


Fig. 2-6 Tool shifting is performed in the tool movement mode

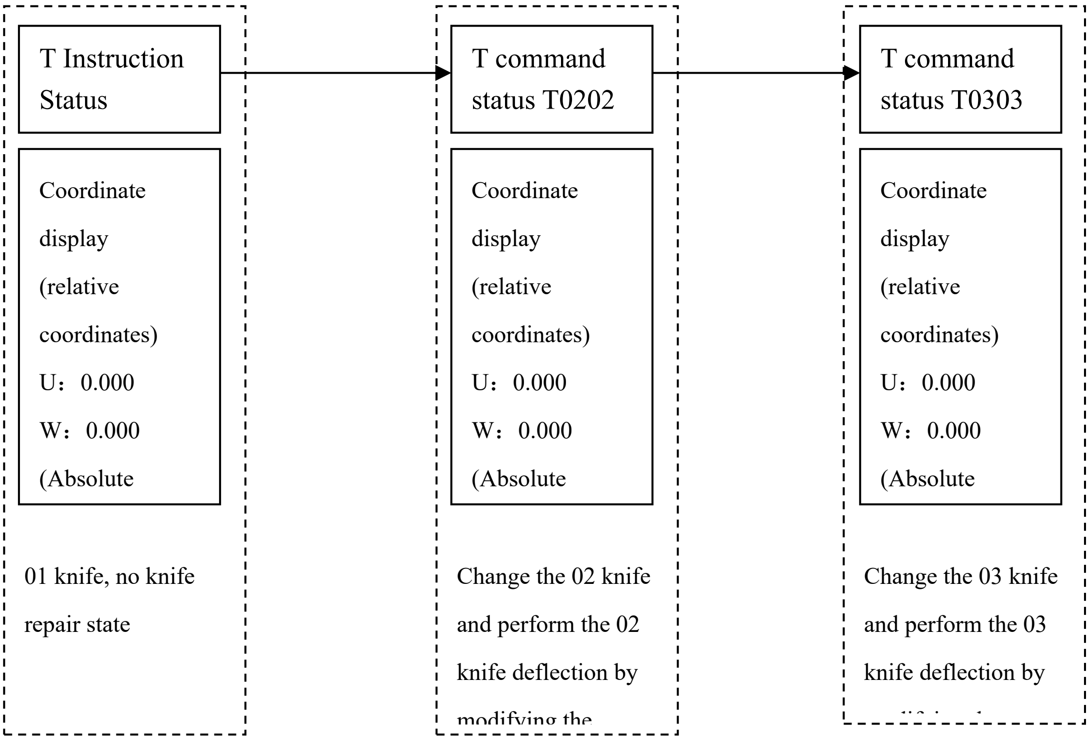


Fig. 2-7 Modify the coordinate mode to perform tool bias

In the input and automatic mode, the process of performing tool bias in a separate T code (not in the same program segment as the move code) is related to the BIT4 setting of the status parameter NO.003 (as shown in Fig. 2-6, Fig. 2-7).

When the T code and the code that performs the move function are executed in the same program section and the tool bias is performed in the way of modifying coordinates, the move code and the T code are executed at the same time, and the current tool bias is superimposed into the coordinate movement value of the move code at the same time as the tool change, and the movement speed is determined by the move code whether it is the cutting feed or the fast movement speed.

When the T code and the code that performs the move function are executed in the same program segment and the tool bias is performed in the way of tool movement, the move code and the T code are executed separately, first performing the tool change and tool bias, and then executing the code of the move function, and the speed at which the tool bias is executed is the current fast movement speed.

Tool bias is canceled when any of the following operations are performed:

1. Executed the T □□ 00 code;

2. The G28 code or manual return to the machine zero point is executed (only the tool bias of the coordinate axis that has returned to the zero point of the machine tool is canceled, and the other coordinate axis that has not returned to the zero point of the machine tool does not cancel the tool bias).

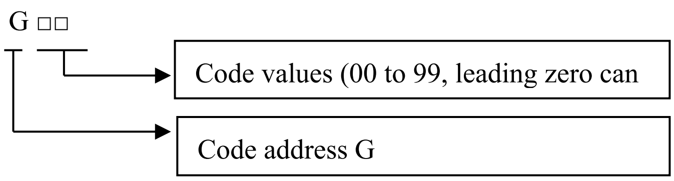
When the data parameter NO.084 (Total Tool Bit Selection) is not set to 1 (2~32), and the target tool number is not equal to the current display tool number, after the T code is instructed, the control timing and logic of the tool holder are controlled by the PLC The ladder diagram determines that the use should be subject to the instructions of the machine manufacturer. Lathe system standard PLC ladder diagram is defined as forward tool selection, reverse rotation locking, tool position signal direct input tool change mode, tool change timing logic refer to the "Debugging Document" or the third "Installation Connection Article".

When using a row tool holder (without an automatic tool holder installed), the data parameter NO.084 (Total Tool Bit Selection) should be set to 1, and different tool numbers are achieved by performing different tool biases, such as: T0101, T0102, T0103 。

Chapter 3 G Code

3.1 Overview

The G code is composed of the code address G and the 1 to 2 digit code values thereafter, which are used to specify the movement direction of the tool relative to the workpiece, coordinate setting and other operations, and the G code list is shown in Table 3-1.



G codes are divided into 00, 01, 02, 03, 05, 06, 07, 16, 21 groups. In addition to the 01 and 00 sets of code can not be shared segments, the same program segment can instruct several different groups of G code, in principle, can not instruct two or more of the same set of G code in the same program segment, If the same set of codes is instructed not to alarm in the same paragraph, the last G code is valid. Different sets of G codes without common parameters (code words) can be in the same program segment, with both functions valid and independent of sequencing. If a G code other than Table 3-1 or a G code with an optional function is used, an alarm occurs.

Table 3-1 List of G code words

Instruction word	Constituencies	function	remark
G00	01	Move fast	The original G code
G01		Straight line interpolation	Modal G code
G02		Arc interpolation (clockwise)	
G03		Arc interpolation (counterclockwise)	
G32		Thread cutting	
G33		Z-axis flexible tapping cycle (not enabled)	
G34		Variable pitch thread cutting	
G90		Axial cutting cycle	

G92		Thread cutting cycle	
G94		Radial cutting rings	
G04	00	Time out	Non-modal G code
G10		The data entry method is valid	
G11		Cancels the data entry method	
G12		The storage stroke detection function is turned on	
G13		The storage stroke detection function is disconnected	
G27		Returns to reference point detection	
G28		Return to the machine reference point 1	
G29		Automatically returns from the reference point	
G30		Return to the machine reference points 2, 3, and 4	
G31		Jump interpolation	
Instruction word	**Constituencies**	**function**	**remark**
G50	00	Coordinate system settings	Non-modal G code
G65		Macro code	
G70		Finishing cycle	
G71		Axial roughing cycle	
G72		Radial roughing cycle	
G73		Closed roughing cycles	
G74		Axial grooving multiple cycles	
G75		Radial groove multiple cycles	
G76		Multiple thread cutting cycles	
G54	05	Workpiece coordinate system 1	Modal G code
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	

G20	06	Imperial unit selection	Modal G code
G21		Metric unit selection	
G96	02	Constant line speed on	Modal G code
G97		Constant line speed off	Initial G code
G98	03	Feed per minute	Initial G code
G99		Feeds per turn	Modal G code
G40	07	Eliminate tip radius compensation	Initial G code
G41		Tip radius left compensated	Modal G code
G42		Tip radius right compensation	
G17	16	XY plane	Modal G code
G18		XZ plane	Initial G code
G19		ZY plane	Modal G code

3.1.1 Modal, non-modal and primary

G codes are divided into 00, 01, 02, 03, 05, 06, 07, 16, 21 groups.

After the G code is executed, its defined function or state remains valid until it is changed by other G codes in the same group, which are called modal G codes. After the modal G code is executed, its defined function or state is changed before, subsequent program segments execute the G code word, there is no need to enter the G code again.

After the G code is executed, its defined function or state is valid at one time, and each time the G code is executed, the G code word must be re-entered, which is called a non-modal G code.

After a system is powered on, a modal G code that is valid without performing its function or state is called an initial G code. When the G code is not entered after power-on, the G code is executed according to the initial state G code.

3.1.2 Omitted input of code words

To simplify programming, the code words listed in Table 3-2 have the characteristics of value retention after execution, if these code words are already included in the previous program segments, they need to be used in subsequent program segments and have the same value and the same meaning, and they can not be entered.

Table 3-2

Programmatic address	Functional significance	The initial value at power-up
U	Depth of cutting in G71	NO.51 parameter value
U	X-axis retract distance in G73	NO.53 Parameter value
W	Depth of cutting in G72	NO.51 parameter value
W	Z-axis retract distance in G73	NO.54 Parameter value

R	G71, G72 cyclic retract volume	NO.52 parameter value
R	The number of rough car cycles in G73	NO.55 parameter value
R	The amount of retracted tool after cutting in G74 and G75	NO.56 parameter value
R	Finishing allowance in G76	NO.60 parameter value
R	G90, G92, G94, G76 medium taper	0
（G98）F	Feed rate per minute (G98)	NO.30 parameter value
（G99）F	Feed rate per revolution (G99).	0
F	Metric thread pitch (G32, G92, G76).	0
I	Inch thread pitch (G32, G92, G76).	0
S	Spindle speed setting (G97).	0
S	Spindle linear speed formulation (G96).	0
S	Spindle speed switching output	0
P	Number of thread cutting finishes in G76	NO.57 Parameter value
P	G76 thread cutting retract width	NO.19 parameter value
P	G76 thread cutting tool tip angle	NO.58 parameter value
Q	Minimum cutthrough in G76	NO.59 parameter value

Note 1 to entry is allowed for programming addresses with multiple functions (e.g. F, which can be used to give a per-minute feed, per-turn feed, metric thread pitch, etc.) only allows the omitting of the input after the code word has been executed and the same function is executed again. For example, G98 F is executed, **the G** code **of thread interpolation is not executed**, and the pitch must be specified with **the F** code **when performing metric thread machining;**

Note 2: The Omitting of input is allowed when **the address X/U, Z/W is used for the coordinates of the end of the program segment,** and when **X/U or Z/W** is not entered in the program segment**, the system takes the current X axis or The absolute coordinates of the Z axis are used as the coordinate values of the end point of the program segment;**

Note 3 to entry: When using programming addresses not listed in Table **3-2, the corresponding code word must be entered and the input cannot be omitted.**

Example 1:
O0001；
G0 X100 Z100； (Quickly move to X100 Z100; The modal code word G0 is valid).

X20 Z30； (Quickly move to X20 Z30; The modal code word G0 can be omitted from the input).

G1 X50 Z50 F300； (Straight line interpolation to X50 Z50, feed speed 300mm/min; The modal code word G1 is valid).

X100； (Straight line insertion to X100 Z50, feed speed 300mm/min; If you do not enter Z coordinates, take the current coordinate value Z50; F300 holds, G01 is the modal code word can be omitted input).

G0 X0 Z0； (Quickly move to X0 Z0, modal code word G0 is valid).

M30；

Example 2:

O0002；

G0 X50 Z5； (Quickly move to X50 Z5).

G04 X4； (4 seconds delay).

G04 X5； (Delay again by 5 seconds, G04 is a non-modal G code word and must be entered again).

M30；

Example 3 (power-up first run):

O0003；

G98 F500 G01 X100 Z100； (G98 Feed per minute, feed speed of 500mm/min).

G92 X50 W-20 F2； (For thread cutting, f-number for pitch must be entered).

G99 G01 U10 F0.01； (G99 For each feed, the F value is re-entered).

G00 X80 Z50 M30；

3.1.3 Definitions

The meaning of the words (words) in the following parts of this user manual is as follows:

Starting point: the position before the current program segment runs;

End Point: The position after the execution of the current program segment has ended;

X: The absolute coordinate of the X axis of the end position;

U: The difference between the absolute coordinates of the X axis of the end point position and the start point position;

Z: The absolute coordinate of the Z axis of the end position;

W: The difference between the absolute coordinates of the Z axis of the end point position and the start point position;

F: Cutting feed speed.

3.2 Quickly locate G00

Code format: G00 X/U_ Z/W_ ;

Code function: The X-axis and Z-axis simultaneously move from the starting point to the end point at their respective fast moving speeds, as shown in Figure 3-1. The two axes move at their own independent speed, the short axis reaches the end point first, and the long axis moves independently for the remaining distance, and its synthetic trajectory is not necessarily a straight line.

Code description: G00 is the initial value of the 01 group G code; X/U, Z/W can omit one or all of them, and when one is omitted, it means that the start and end coordinate values of the axis are the same; Simultaneous omitting means that the end point and the start point are in the same position, X and U, Z and W when in the same program segment X, Z Valid, U, W is invalid.

Motion trajectory diagram:

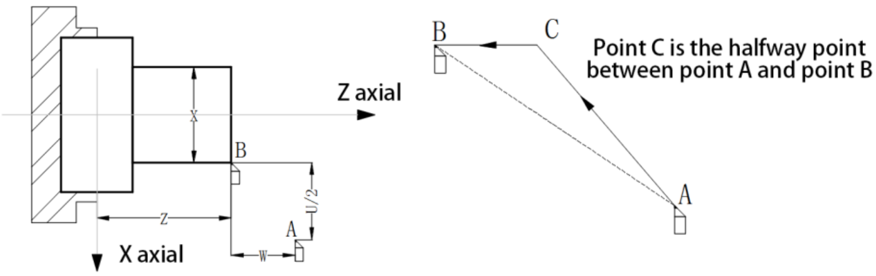


Figure 3-1

The fast movement speed of the X and Z axes is set by the system data parameters NO.022 and NO.023 respectively, and the actual movement speed can be adjusted by the fast magnification key of the machine panel.

Example: The tool moves quickly from point A to point B. Figure 3-2

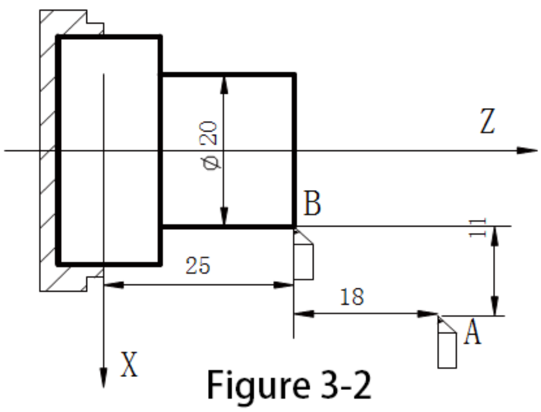


Figure 3-2

G0 X20 Z25; (Absolute coordinate programming) or
G0 U-22 W-18; (Programming relative coordinates) or
G0 X20 W-18; (Mixed Coordinate Programming) or
G0 U-22 Z25; (Mixed Coordinate Programming).

3.3 Straight line interpolation G01

Code format: G01 X/U_ Z/W_ F_;

Code function: The motion trajectory is a straight line from the start to the end. The trajectory is shown in Figure 3-3.

Code description: G01 is the modal G code;

X/U, Z/W can omit one or all of them, and when one is omitted, it means that the start and end coordinate values of the axis are the same; Omitting at the same time means that the end point and the start point are in the same location.

The F code value is the vector synthesis speed of the instantaneous velocity in the X axis direction and the Z axis direction, and the actual cutting feed rate is the product of the feed rate and the F code value;

After the F code value is executed, the code value remains until the new F code value is executed. The F-code word functions used by the other G codes described below are the same and will not be described in detail. The range of values is shown in Table 1-10.

Note: In the G98 state, the maximum value of F does not exceed the value set for data parameter NO027 (Cutting Feed Upper Limit Speed).

Motion trajectory diagram:

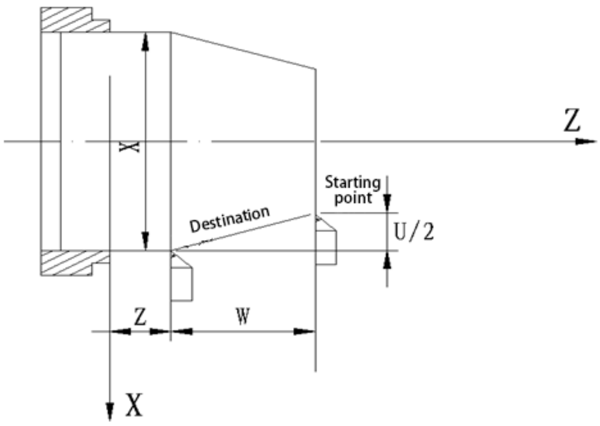


Figure 3-3

Example: Program code for cutting from diameter Φ40 to Φ60, Fig. 3-4

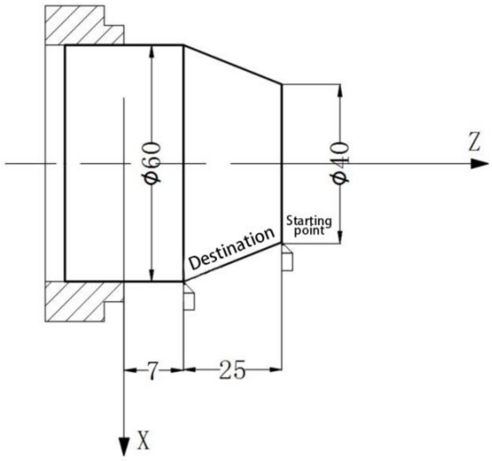


Figure 3-4

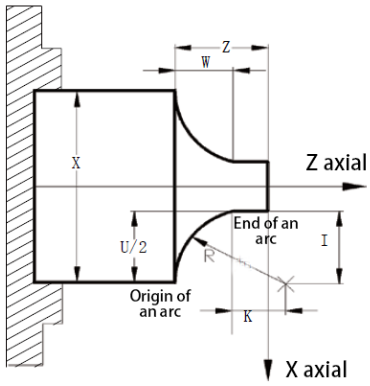
Procedure:
G01 X60 Z7 F500; (Absolute programming) or
G01 U20 W-25; (Relative value programming) or
G01 X60 W-25; (Mixed Programming) or
G01 U20 Z7; (Hybrid Programming)

3.4 Arc interpolation G02, G03

Code format: G02 / G03 X/U_ Z/W_ R_ / I_ K_

Code function: G02 code motion trajectory is clockwise (rear tool holder coordinate system) / counterclockwise (front tool seat coordinate system) arc from start to finish point, the trajectory is shown in Figure 3-5 as shown. The G03 code trajectory is a counterclockwise (rear toolholder coordinate system) / clockwise (front toolholder coordinate system) arc from start to finish, as shown in the figure 3-6 shown.

Code trace diagram:



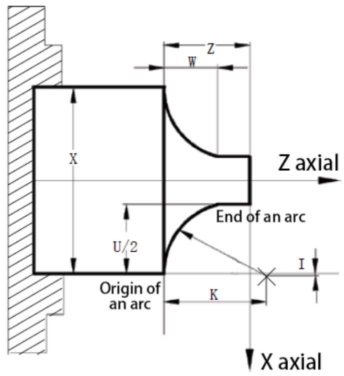


Fig. 3-5 G02 Trajectory Diagram Fig. 3-6 G03 Trajectory Diagram

Code description: G02, G03 are modal G codes;

R is the arc radius;

I is the difference between the center of the circle and the beginning of the arc in the X direction, expressed in radius;

K is the difference between the center of the circle and the beginning of the arc in the Z direction. When the center of the arc is specified with the addresses I and K, which correspond to X, Z axis I and K respectively, represent the vector components from the beginning of the arc to the center of the circle, which is the increment value; This is shown in Figure 3-6-1.

I = Center Coordinate X - X coordinate of the start point of the arc;   K = center coordinate Z - Z coordinate of the start point of the arc;

I and K are symbolized according to the direction, and the I and K directions are the same as the X and Z axis directions, which are positive; Otherwise, take a negative value.

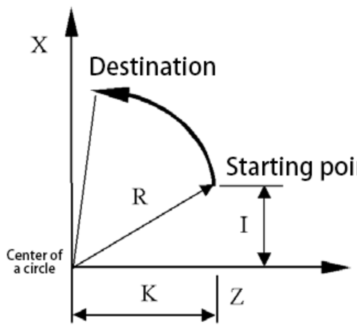


Figure 3-6-1

Arc direction: G02/ G03 The direction of the arc is defined, and the front and rear tool seat coordinate systems are opposite, see Figure 3-7:

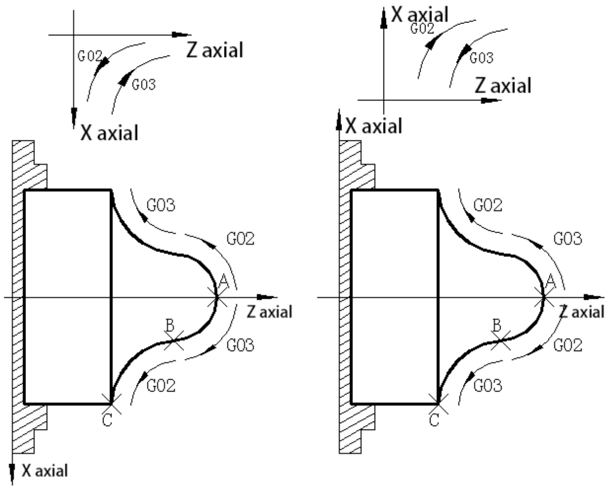


Figure 3-7

Notes:

* Can be omitted when I=0 or K=0; However, the address I, K or R must be entered at least one, otherwise the system generates an alarm;

* When I, K and R are entered at the same time, R is valid, I and K are invalid;

* The R value must be equal to or greater than half of the start to the end point, if the end point is not on the arc defined with R, the system will generate an alarm;

* Address X/U, Z/W can omit one or all; When one is omitted, it means that the start and end points of the omitted axis are consistent; At the same time, the omission indicates that the end point and the starting point are the same position, if the center of the circle is specified with I and K, the trajectory of the G02/G03 code is full circle (360°); When specified with R, a circle representing 0°;

* If you program with I and K values, if the distance from the center of the circle to the end of the arc is not equal to R(R = $I^2 + K^2$), The system automatically adjusts the center position of the circle to ensure that the start and end points of the arc movement are consistent with the specified values, and if the distance between the beginning and end points of the arc is greater than 2R, the system alarms.

* R is specified as an arc less than 360°, an arc greater than 180° when R is negative, and an arc less than or equal to 180 degrees when R is positive;

Example: Arc program code for cutting from diameter Φ45.25 to Φ63.06, Fig. 3-8

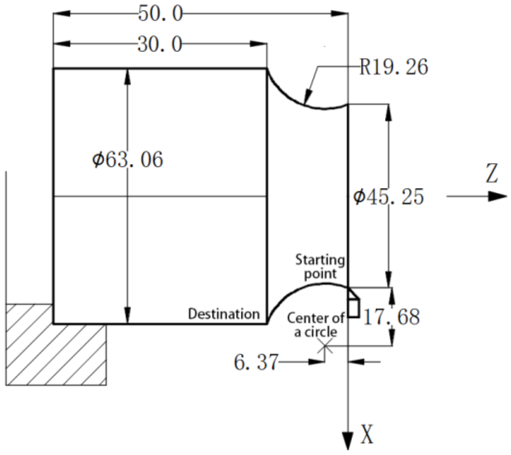


Figure 3-8

Procedure:
G02 X63.06 Z-20.0 R19.26 F300; or
G02 U17.81 W-20.0 R19.26 F300; or
G02 X63.06 Z-20.0 I17.68 K-6.37; or
G02 U17.81 W-20.0 I17.68 K-6.37;

G02/G03 Code Synthesis Programming Example:

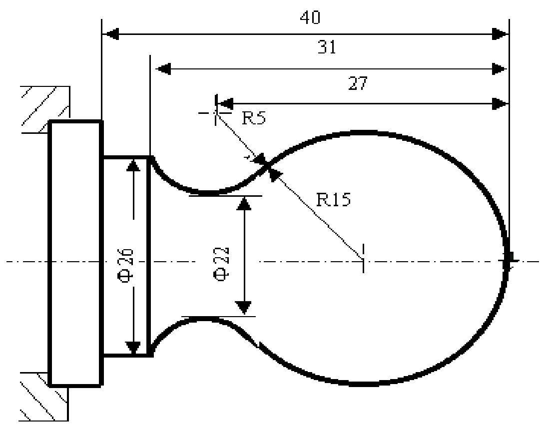


Figure 3-9 An example of arc programming

Procedure: O0001
N001 G0 X40 Z5； (Quick Positioning).
N002 M03 S200； (Spindle open).
N003 G01 X0 Z0 F900； (near the workpiece).
N005 G03 U24 W-24 R15； (Cutting R15 circular arc segment).
N006 G02 X26 Z-31 R5； (Cutting R5 circular arc segments).

N007 G01 Z-40；　　　　(Cut ф26).
N008 X40 Z5；　　　　(Return to Starting Point).
N009 M30；　　　　　(End of procedure).

3.5 Plane selection codes G17 ~ G19

Code format:

G17...... XY plane
G18...... ZX plane
G19...... YZ plane

Code function: Use the G code to select the plane of arc interpolation or the plane of tool radius compensation

Code description: G17, G18, G19 is a modal G code, in the program segment without instructions, the plane does not change.

Notes:

* When selecting the G17 and G19 planes, the basic axis Y should be set first;
* C Plane switching cannot be performed under knife repair;
* G71 ~ G76, G90, G92, G94 can only be used in the G18 plane;
* Plane selection code can be co-segmented with other group G codes;
* The move instruction is independent of plane selection;
* About the handling of diameter or radius programming: Since there is currently only one bit parameter No1.2 to choose whether to program the diameter or radius and is only valid for the X axis, the Z axis is with when using instructions such as G2, G3, etc The Y axis can only be programmed with a radius, while the X axis is selected by parameters;
* The tip direction of the C tool patch under the G17 and G19 planes is 0.

3.6 Chamfer function

The chamfer function inserts a straight line or arc between two profiles, allowing the tool to transition from one profile to another relatively smoothly. It has two chamfering functions: straight line and circular arc.

3.6.1 Straight line chamfering

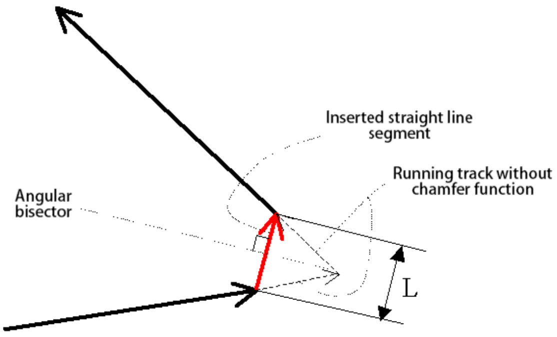
Straight line chamfer: Insert a straight line between line profiles, between arc profiles, between line profiles and arc profiles. The code address of the line chamfer is L, the length of the chamfer line is specified with L, the value range is 0 ~ 1000mm, if the value specified by L exceeds the range, the L code is ignored. Straight line chamfers must be used in the G01, G02, or G03 code snippets.

A. Straight line to straight line

Code format: G01 X/U_ Z/W_ L_ ;

G01 X/U_ Z/W_ ;

Code function: Inserts a straight line segment in two straight line interpolation code snippets.



B. Straight line connected to arc

Code format: G01 X/U_ Z/W_ L_;

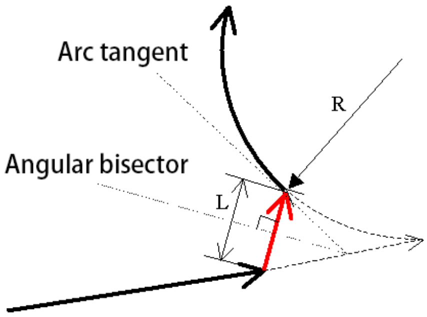
G02/G03 X/U_ Z/W_ R_;

or

G01 X/U_ Z/W_ L_;

G02/G03 X/U_ Z/W_ I_ K_;

Code feature: Inserts a straight line segment between line and arc interpolation code.



C. The arc is connected to the arc

Code format: G02/G03 X/U_ Z/W_ R_ L_;

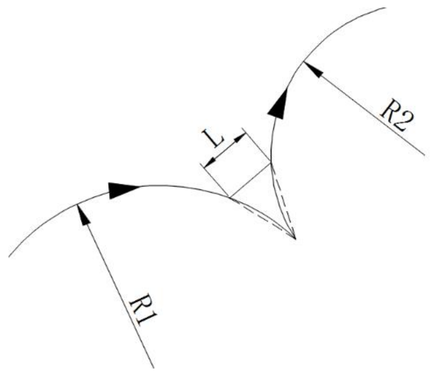
G02/G03 X/U_ Z/W_ R_;

or

G02/G03 X/U_ Z/W_ I_ K_ L_;

G02/G03 X/U_ Z/W_ I_ K_;

Code function: Inserts a straight line segment between two arc interpolation codes.



D. The arc is connected to a straight line

Code format: G02/G03 X/U_ Z/W_ R_ L_;

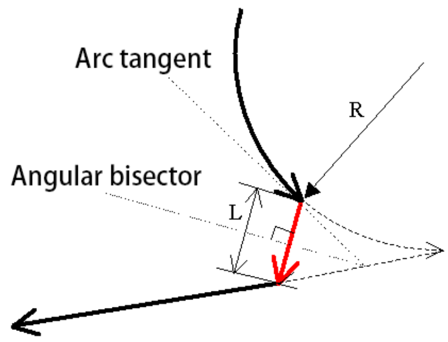
G01 X/U_ Z/W_;

or

G02/G03 X/U_ Z/W_ I_ K_ L_;

G01 X/U_ Z/W_;

Code function: Inserts a straight line segment between arc and line interpolation code.



3.6.2 Arc chamfering

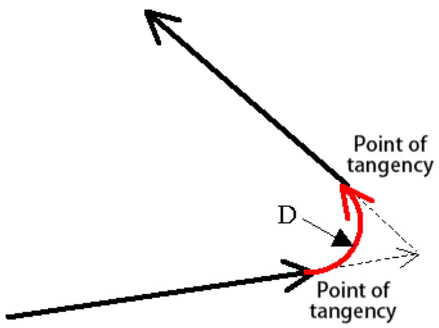
Arc chamfer: Insert an arc between straight contours, between arc profiles, between straight lines and arc profiles, and make tangent transitions between arcs and contour lines. The code address of the arc chamfer is D, the radius of the chamfer arc is specified with D, the value range is 0 ~ 1000mm, if the value specified by D exceeds the range, the D code is ignored. Arc chamfering must be used in the G01, G02, or G03 code snippet.

A. Straight line to straight line

Code format: G01 X/U_ Z/W_ D_;

G01 X/U_ Z/W_;

Code function: Insert an arc in the interpolation segment of two straight lines, the inserted arc segment is tangent to the two straight lines, and the radius value is specified in D.



B. Straight line connected to arc

Code format: G01 X/U_ Z/W_ D_;

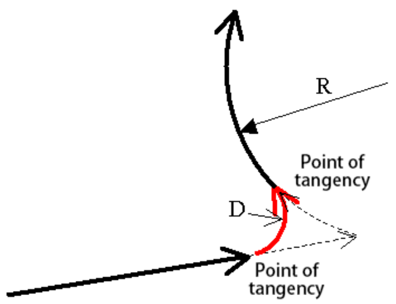
G02/G03 X/U_ Z/W_ R_;

or

G01 X/U_ Z/W_ D_;

G02/G03 X/U_ Z/W_ I_ K_;

Code function: insert a circular arc at the junction of the straight line and the arc, insert the circular arc segment and the straight line and the arc are tangent, and the radius value is specified with D.



C. The arc is connected to the arc

Code format: G02/G03 X/U_ Z/W_ R_ D_;

G02/G03 X/U_ Z/W_ R_;

or

G02/G03 X/U_ Z/W_ R_ D_;

G02/G03 X/U_ Z/W_ I_ K_;

or

G02/G03 X/U_ Z/W_ I_ K_ D_;

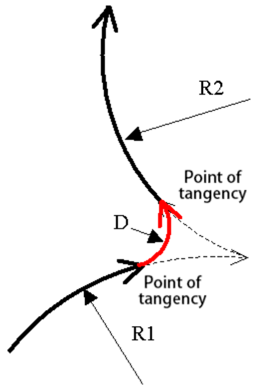
G02/G03 X/U_ Z/W_ I_ K_;

or

G02/G03 X/U_ Z/W_ I_ K_ D_;

G02/G03 X/U_ Z/W_ R_;

Code function: Insert a circular arc between two arcs, the inserted arc segment is tangent to both arcs, and the radius value is specified in D.



D. The arc is connected to a straight line

Code format: G02/G03 X/U_ Z/W_ R_ D_;

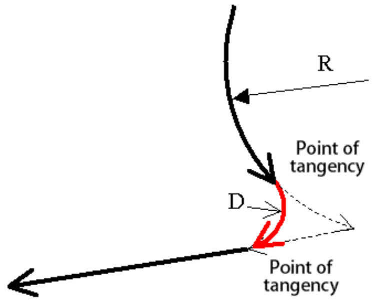
G01 X/U_ Z/W_;

or

G02/G03 X/U_ Z/W_ I_ K_ D_;

G01 X/U_ Z/W_;

Code function: Insert a circular arc at the junction of the arc and the straight line, insert the arc segment and the arc and the straight line are tangent, and the radius value is specified with D.



3.6.3 Special Circumstances

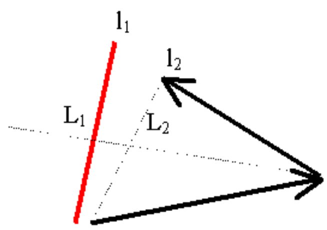
When in the following situation, the chamfer function is invalid or alarmed.

1) When the line is chamfered

A. When two interpolated straight line segments are on the same line, the chamfer function is invalid.

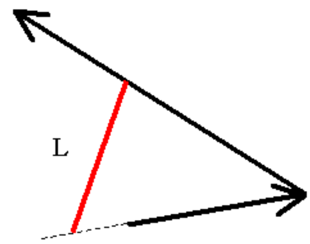
B. The length of the chamfer line is too long, and the CNC generates an alarm.

As shown in the following figure, l1 is a chamfered line with a length of L1; l2 is the third side of a triangle formed by the connection of two interpolated straight lines, the length is L2, and when L1 is greater than L2, the CNC generates an alarm.



C. A straight line (arc) is too short, alarm

As shown in the figure below, the length of the chamfered line is L, and after calculation, the other end of the chamfered line is not on the interpolated line (on the extension line of the interpolated line), and the CNC generates an alarm.

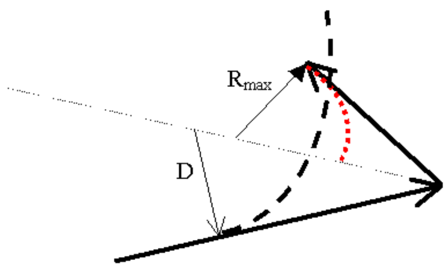


2) When the arc is chamfered

A. When two interpolated straight line segments are on the same line, the arc chamfer function is invalid.

B. The radius of the chamfer arc is too large, and the CNC generates an alarm.

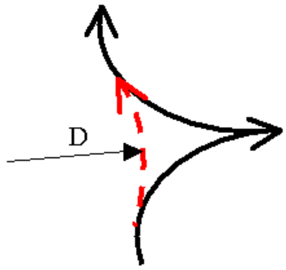
As shown in the figure below, the chamfer arc radius is D, the maximum arc radius of two straight lines tangent is Rmax, Rmax is less than D, and the CNC generates an alarm.



C. When a straight line is tangent to an arc, and an arc is tangent to a straight line, the arc chamfer function is invalid.

D. When the arc is tangent to the arc, the arc chamfer function is invalid;

However, if it is tangential like the following figure, the arc chamfer function is effective.



3.7 Pause code G04

Code format: G04 P; Or

G04 X ; Or

G04 U ; Or

G04;

Code function: Each axis movement stops, does not change the current G code mode and holds the data, state, delay a given time, and then execute the next program segment.

Code description: G04 is a non-modal G code;

G04 The delay time is specified by the code word P_, X_, or U_;

The P-value ranges from 0 to 99999 (in ms).

X and U codes range from 0 to 9999.999× minimum input increment (unit: s).

Notes:

* When P, X, U is not entered, it means that the program segment is accurately stopped.

* P, X, U cannot be in the same program segment.

3.8 Mechanical zero point (machine zero point) function

3.8.1 Machine tool first reference point G28

Code format: G28 X/U Z/W;

Code function: Starting from the starting point, moving fast to the midpoint position specified by X/U, Z/W, and then returning to the machine zero.

Code description: G28 is a non-modal G code;

X, Z: The absolute coordinates of the midpoint position;

U, W: The difference between the absolute coordinates of the X axis of the midpoint position and the starting point position.

Code addresses X/U, Z/W can omit one or all of them, as detailed in the table below:

Table 3-4

directives	function
G28 X/U_	The X axis returns to the zero point of the machine, and the Z and Y remain in their original positions
G28 Z/W_	The Z axis returns to the zero point of the machine, and the X and Y remain in their original positions
G28	Leave in place and continue to the next line of the program segment
G28 X/U_ Z/W_	The X and Z axes return to the machine zero point at the same time

Code action process (Figure 3-12):

(1) Quickly locate from the current position to the midpoint position (point A→ point B);

(2) Quickly locate from the midpoint to the reference point (B point → R point);

(3) If the machine is not locked, when the reference point is returned, the zero light is turned on.

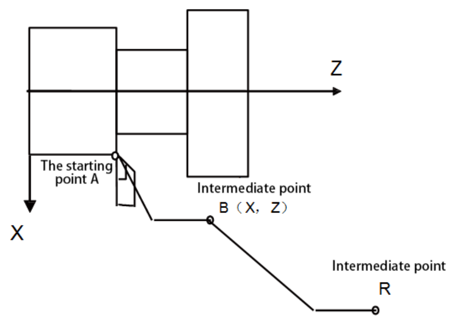


Figure 3-12

Note 1: The manual return to the machine zero point is consistent with the process of executing the G28 code back to the machine zero point, and the deceleration signal and the one-turn signal must be detected each time;

Note 2: From point **A →** point **B and point B → point R, the two axes move at their own independent fast speeds, so their trajectories are not necessarily straight;**

Note 3: After performing the G28 code back to the zero point operation of the machine tool, the system cancels the tool length compensation;

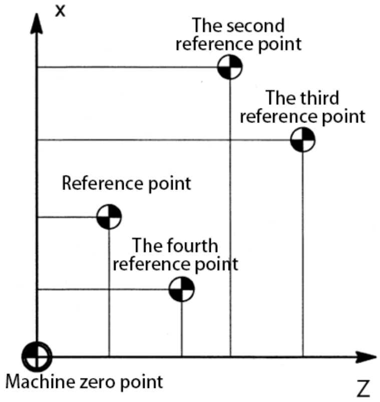
NOTE 4 If the machine does not have a zero switch installed, the G28 code and the operation of returning the zero point of the machine must not be performed.

3.8.2 Machine tool reference points 2, 3 and 4 G30

The zero point of the machine tool is a fixed point on the machine tool, which is determined by the zero switch or the zero switch installed on the machine tool. The coordinates of the machine tool reference point are the data parameter NO. 120、NO. 121 Set the value.

The lathe system has the function of the reference point of the 2nd, 3rd and 4th machine tools, and the data parameters NO.122~NO.127 can be set up for the machine tools no. 2 and 3 respectively , 4Cell coordinates of the X and Z axes of the reference point.

The relationship between the machine zero point, the machine tool reference point, and the machine tool reference points 2, 3, and 4 in the coordinate system of the machine tool is shown in the following figure.



Code format:

G30 P2 X/U Z/W ；

G30 P3 X/U Z/W ；

G30 P4 X/U Z/W ；

Code function: From the starting point, move at a fast movement speed to the midpoint position specified by X/U, Z/W and then return to the 2nd, 3rd, and 4th reference points of the machine. When returning to the machine tool reference point 2, the code address P2 can be omitted.

Code description: G30 is a non-modal G code;

X: The absolute coordinate of the X axis of the middle point;

U: The relative coordinates of the X axis of the intermediate point;
Z: The absolute coordinate of the Z axis of the middle point;
W: The relative coordinate of the Z axis of the middle point.

Code addresses X/U, Z/W can omit one or all of them, as detailed in the table below:

directives	function
G30 Pn X/U_	The X axis returns to the nth reference point of the machine, and the Z axis remains in place
G30 Pn Z/W_	The Z axis returns to the nth reference point of the machine, and the X axis remains in place
G30	The two axes remain in place and proceed to the next program segment
G30 Pn X/U_ Z/W_	The X and Z axes return to the nth reference point of the machine at the same time

Note 1: The **value** of **n in the table is 2, 3**, **or 4;**

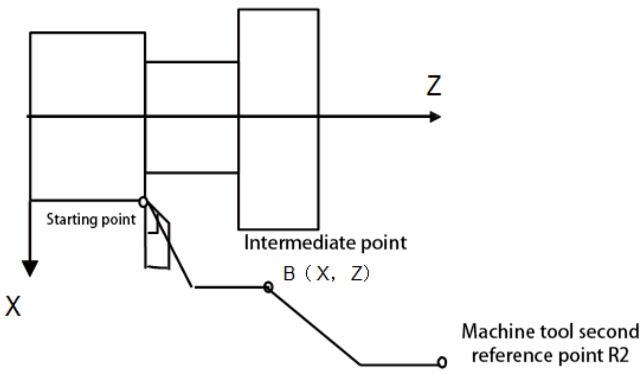
Note **2**: There is no need to detect deceleration **and** zero signals during **the return to the 2nd**, **3rd**, and **4th reference points of the machine.**

The code executes the action process (as shown in the figure below, as explained back to the machine tool reference point 2):

(1) Quickly position from the current position to the midpoint position of the specified axis (point A→ point B);

(2) Positioning from the midpoint to the 2nd reference point (point B→ R2) set by data parameters NO.122 and NO.123 at the speed set by data parameters NO.113 ;

(3) If the machine is not locked, when the reference point is returned, the reference point position returns to the end signal ZP21 Bit0 bit, Bit1 bit is high.



Note 1: After manually returning to the machine reference point or executing the G28 code back to the machine reference point, the function of returning to the machine tool reference point **2, 3, and 4** can be used**;**

Note 2: In the process from point **A** → point **B and** point **B → point R2, the two axes move at their own independent speeds, so their trajectories are not necessarily straight;**

Note 3: After executing the **G30 code back to reference points 2,** 3**, and 4 of the machine, the system cancels the tool length compensation;**

Note **4: If the machine tool is not installed with a zero switch, the G30 code must not be performed to return the machine tool** to reference **points** 2**, 3, and 4;**

Note 5: Returns to the 2nd, 3rd, and 4th reference points of the machine without setting the workpiece coordinate system.

3.9 Jump plug G31

Code format: G31 X/U_ Z/W_ F_;

Code function: During the execution of the code, if an external jump signal is entered (X_ the specific signal of the input signal refers to the Debugging Documentation), the execution of the code is interrupted and the next program segment is executed. This function can be used for dynamic measurement of workpiece dimensions (e.g. grinding machines), tool counter measurements, etc.

Code description: non-modal G code (00 groups);

Consistent with the G01 code address format, the use is similar;

Tip radius compensation needs to be revoked before using this code;

In order to ensure the accuracy of the stop position, the feed speed should not be set too large.

a. Line for subsequent paragraphs when the jump occurs:

1. The next program segment of G31 is incremental coordinate programming, as shown in Figure 3-13

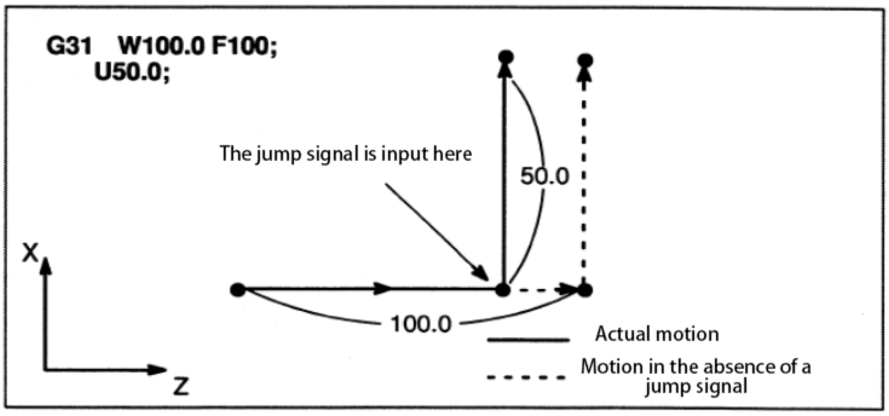


Figure 3-13

2. The next program segment of the G31 is the absolute coordinate programming of 1 axis, as shown in Figures 3-14

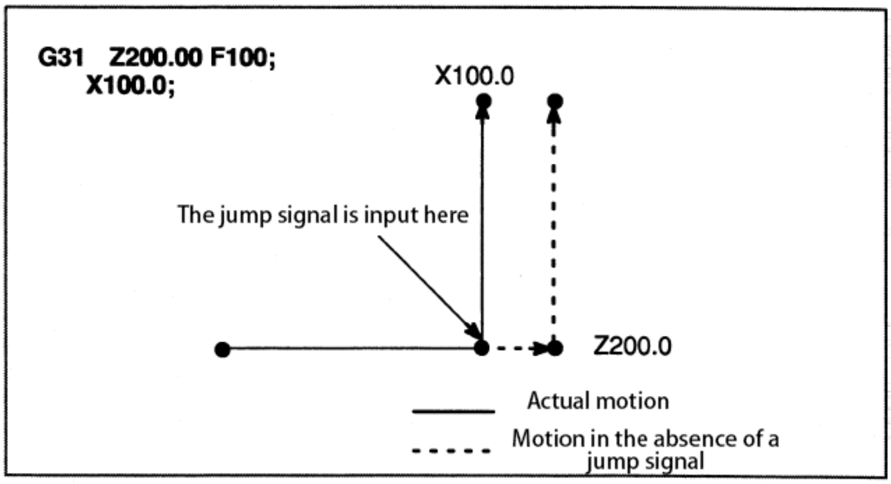


Figure 3-14

3． The next program segment of the G31 is the absolute coordinate programming of the 2 axes, as shown in Figure 3-15

Program: G31 Z200 F100

G01 X100 Z300

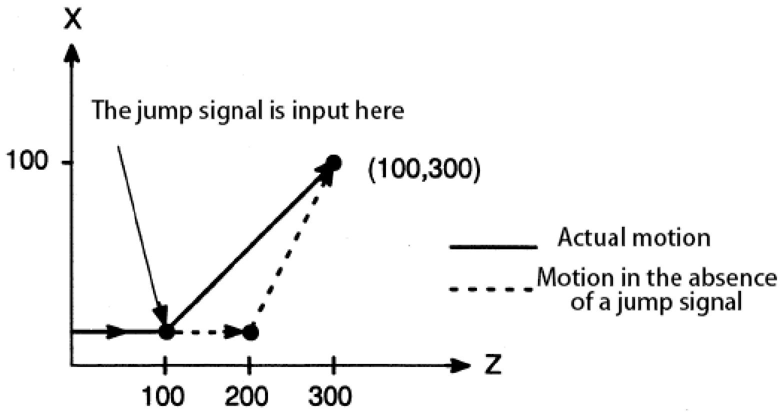


Figure 3-15

b. Signals related to G31 jump codes:

Jump Signal:

SKIP：G6.6

Type: Input signal

Function: G6.6 signal end jump cutting. That is, in a program segment containing G31, the absolute coordinate position where the jump signal becomes "1" is stored in the user macro variable (#5011 ~ #5015 corresponds to X, Z, respectively ，Y，4th，5th)。 Also, end the motion code of the program segment at the same time.

Operation: When the jump signal changes to "1", the CNC processing is as follows:

When the program segment is executing the jump code G31, the CNC stores the current absolute coordinate position of each axis. The CNC stops the movement of the G31 code and begins execution of the next program segment, and the jump signal detects not its rising edge, but its state. Therefore, if the jump signal is "1", it is considered that its jump condition is immediately satisfied.

Note: In order to ensure the accuracy of the stop position, the feed speed of the G31 should be as low as possible.

3.10 The floating workpiece coordinate system is set to G50

Code format: G50 X/U_　Z/W_ ;

Code function: set the absolute coordinates of the current position, and establish a floating workpiece coordinate system in the system by setting the absolute coordinates of the current position.

After executing this code, the system takes the current position as the program zero point, and returns this position when performing the operation back to the program zero point. After the floating workpiece coordinate system is established, the absolute coordinate programming enters coordinate values according to this coordinate system until the G50 is executed again to establish a new workpiece coordinate system.

Code description: G50 is a non-modal G code;

X: The absolute coordinate of the new X axis of the current position;

U: The difference between the absolute coordinates of the new X axis in the current position and the absolute coordinates before the code is executed;

Z: The absolute coordinate of the new Z axis of the current position;

W: The difference between the absolute coordinates of the new Z axis at the current location and the absolute coordinates before the code is executed;

In the G50 code, if X/U and Z/W are not entered, the current coordinate value is not changed, and the current point coordinate value is set to the program zero point (the program zero point is not set when the G50 SXXXX　is not set).

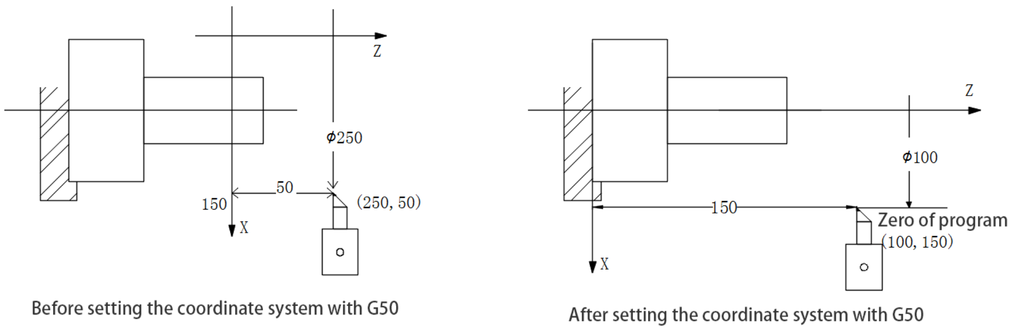


Figure 3-16

As shown in Figure 3-16, when the code snippet "G50 X100 Z150; After that, the workpiece coordinate system as shown is established and the (X100Z150) point is set to the program zero point.

3.11 Workpiece coordinate system G54 ~ G59

Code format: G54 ~ G59

Code function: Specify the current workpiece coordinate system, and select the workpiece coordinate system by specifying the workpiece coordinate system G code in the program.

Code Description:

1. No instruction parameters.

2. The system itself can set six workpiece coordinate systems, and any one of them can be selected by the instructions G54 ~ G59

G54 ---------------- workpiece coordinate system 1
G55 ---------------- workpiece coordinate system 2
G56 ---------------- workpiece coordinate system3
G57 ---------------- workpiece coordinate system 4
G58 ---------------- workpiece coordinate system5
G59 ---------------- Workpiece Coordinate System6

3. When different workpiece coordinate systems are called in the program segment, the axis that is moved by the instruction will be positioned to the coordinate point under the new workpiece coordinate system; For axes that do not have instructions to move, the coordinates will jump to the corresponding coordinate values under the new workpiece coordinate system, and the actual machine position will not change.

Example: The coordinate system origin of the G54 corresponds to the coordinate coordinates of the machine tool (20, 20).

The coordinate system origin of the G55 corresponds to the coordinates of the machine tool (30, 30).

When executing the program sequentially, the absolute coordinates of the end point and the coordinates of the machine are displayed as follows:

Table 3.11.1

procedure	Absolute coordinates	Machine coordinates
G0 G54 X50 Z50	50,50	70,70
G55 X100	100,40	130,70
X120 Z80	120，80	150,110

4.

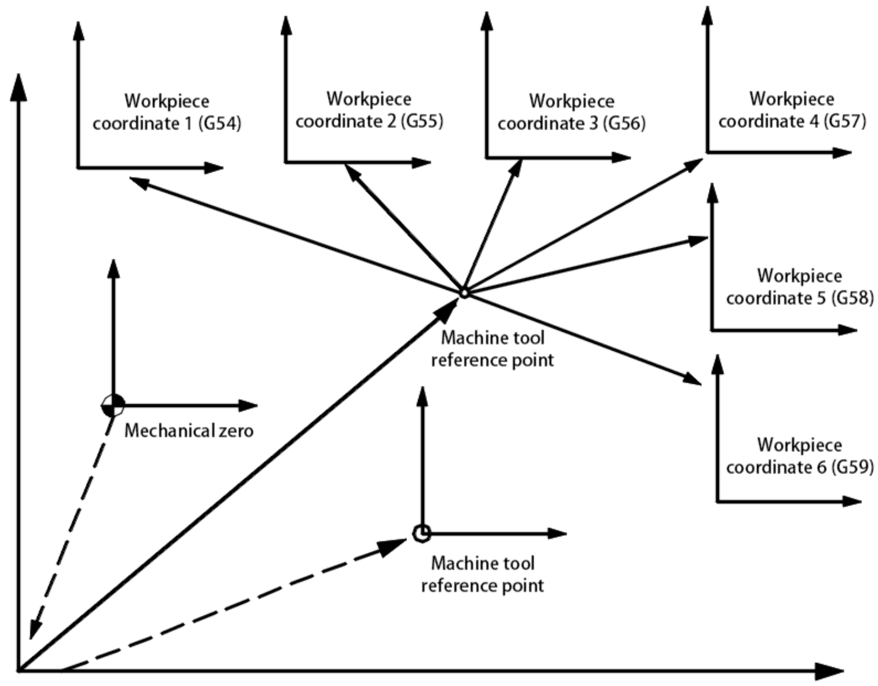


Figure 3.11.2

As shown in the figure above, after the machine tool is started, the machine tool manually returns to the mechanical zero point, and the machine tool coordinate system is established by the mechanical zero point, which generates the machine tool reference point and determines the workpiece coordinate system. The external workpiece origin offset data parameter P270 ~ 274 corresponds to the overall offset of the 6 workpiece coordinate systems. The input or setting data parameters P128 ~ P139 under the coordinate bias can be entered, and the origin of 6 workpiece coordinate systems can be specified by P275 ~ 292. The six workpiece coordinate systems are set according to the distance from the mechanical zero point to the zero point of the respective coordinate system.

Example: N10 G55 G90 G00 X100 Z20;

N20 G56 X80.5 Z25.5；

In the above example, when the N10 program segment starts executing, it is quickly positioned to the position of the workpiece coordinate system G55 (X=100, Z=20). When the N20 program segment starts to execute, quickly locate the position of the workpiece coordinate system G56, and the absolute coordinate value automatically becomes the coordinate value under the G56 workpiece coordinate system (X=80.5, Z=25.5 ）。

3.12 Fixed Loop Codes

To simplify programming, the lathe system provides G-codes for a single machining cycle with only one program segment for fast moving positioning, straight line/thread cutting, and finally a quick movement back to the starting point:

G90: Axial cutting cycle; G92: Thread cutting cycle; G94: Radial cutting cycle

The G92 Thread Cutting Fixed Cycle Code is described in the Thread Functions section.

3.12.1 Axial cutting cycle G90

Code format: G90 X/U_ Z/W_ F_ ; (Cylindrical cutting).

G90 X/U_ Z/W_ R_ F_ ； (Cone cutting).

Code function: starting from the cutting point, radial (X-axis) tool feed, axial (Z-axis or X- and Z-axis simultaneous) cutting, Enables cylindrical or cone cutting cycles.

Code description: G90 is a modal code;

Cutting starting point: the starting position of straight line interpolation (cutting feed);

Cutting end point: the end position of the straight line interpolation (cutting feed);

X: Absolute coordinate of the X axis of the cutting end point;

U: The difference between the absolute coordinates of the X axis of the cutting end point and the starting point;

Z: Absolute coordinates of the Z axis of the cutting end point;

W: The difference between the absolute coordinates of the Z axis of the cutting end point and the starting point;

R: The difference between the absolute coordinates of the X axis of the cutting start point and the cutting end point (radius value), with direction, when the R and U signs are inconsistent, the requirement $|R| \leq |U/2|$； When R = 0 or the default input, cylindrical cutting is performed, as shown in Fig. 3-17, otherwise cone cutting is performed, as shown in Fig. 3-18.

Cycle process: (1) The X-axis moves quickly from the starting point to the cutting starting point;

(2) Straight line interpolation from the starting point of the cutting (cutting feed) to the cutting end;

(3) The X-axis retracts at the cutting feed rate and returns to the same absolute coordinate of the X-axis as the starting point;

(4) The Z axis moves quickly back to the starting point, and the loop ends

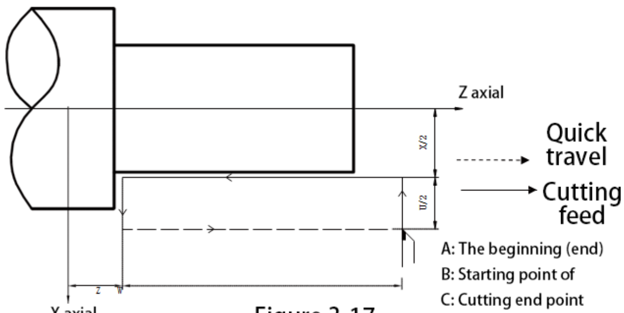


Figure 3-17

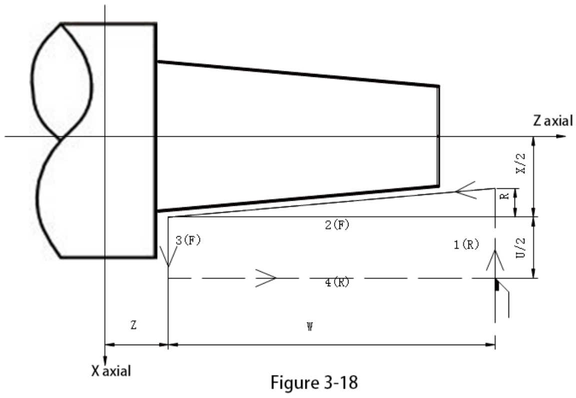


Figure 3-18

Code trajectory: U, W, R reflect the relative position of the cutting end point and the starting point, and the tool trajectory of U, W, R combined at different symbols, as shown in the figure

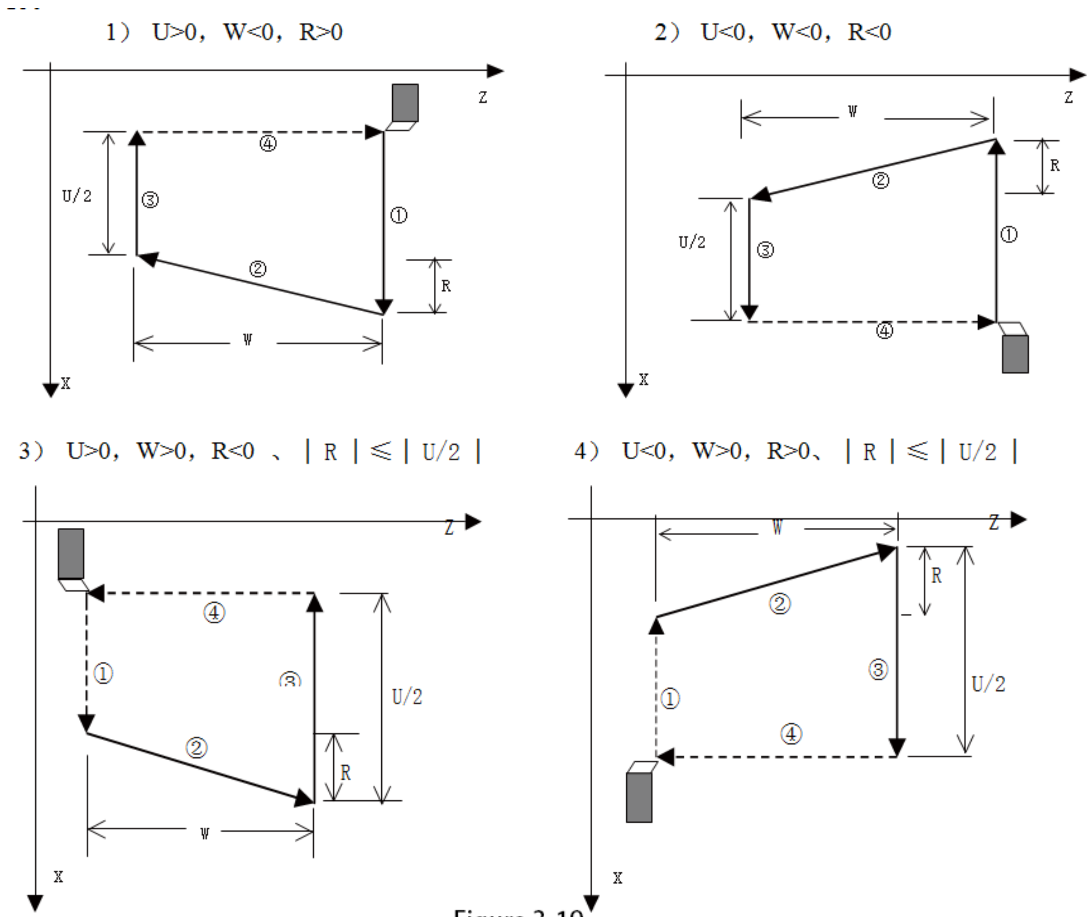


Figure 3-19

Example: Fig. 3-20, blank Φ125×110

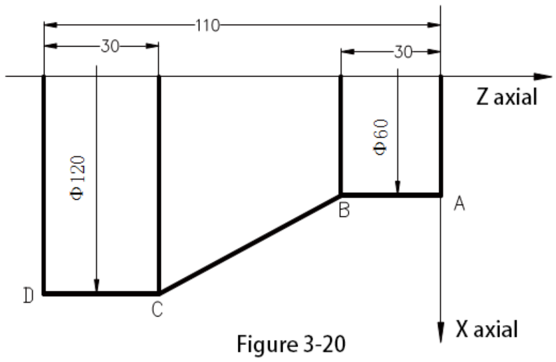


Figure 3-20

Procedure: O0002;
M3 S300 G0 X130 Z3;
G90 X120 Z-110 F200; (A→D, ∅ 120 cutting).
X110 Z-30;
X100;
X90;
X80;
X70;
X60;
(A→B, ∅60 cuts, divided into six feed cycles, each time to 10 mm).
G0 X120 Z-30;
G90 X120 Z-44 R7.5 F150;
Z-56 R-15;
Z-68 R-22.5;
Z-80 R-30;
(B→C, taper cutting, cut in four feed cycles).
M30;

3.12.2 Radial cutting cycle G94

Code format: G94 X/U_ Z/W_ F_ ; (Face cutting).
G94 X/U_ Z/W_ R_ F_ ； (Taper face cutting).

Code function: starting from the cutting point, axial (Z axis) feed, radial (X axis or X, Z axis simultaneous) cutting, to achieve end face or cone cutting cycle, The start and end points of the code are the same.

Code description: G94 is a modal code;
Cutting starting point: the starting position of straight line interpolation (cutting feed);
Cutting end point: the end position of linear interpolation (cutting feed);
X: Absolute coordinate of X axis of the cutting end point, unit: mm/inch;
U: The difference between the absolute coordinates of the cutting end point and the X axis of the starting point;
Z: The absolute coordinate of the Z axis of the cutting end point;
W: The difference between the absolute coordinates of the Z axis of the cutting end point and the starting point Z axis;
R: The difference between the absolute coordinate of the Z axis of the cutting start point and the cutting end point, when the symbols of R and U are different, the requirement is $|R| \le |W|$, radial straight line cutting as shown in Figure 3-21, Radial taper cutting is shown in Figure 3-22.

Cycle process: (1) The Z axis moves quickly from the starting point to the cutting starting point;
(2) Straight line interpolation from the starting point of the cutting (cutting feed) to the cutting end;

(3) The Z axis retracts at the cutting feed rate (opposite to the (1) direction) and returns to the Z axis where the absolute coordinates of the Z axis are the same as the starting point;

(4) The X axis moves quickly back to the starting point, and the loop ends.

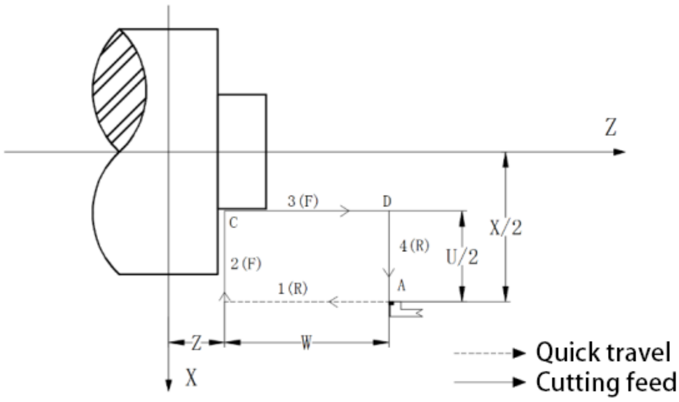


Figure 3-2-1

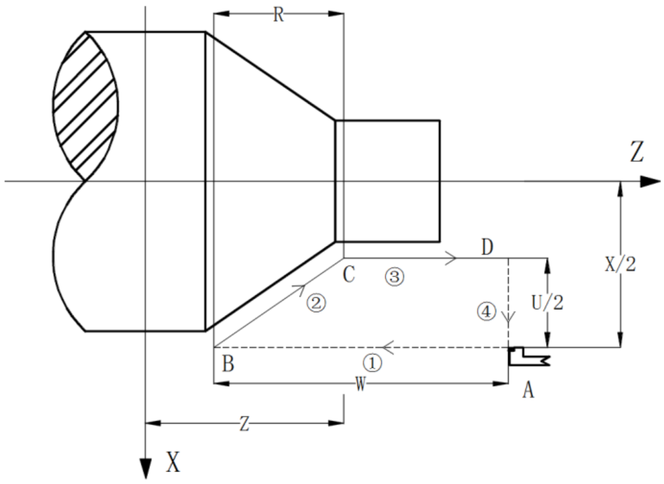


Figure 3-22

Code trajectory: U, W, R reflect the relative position of the cutting end point and the starting point, and the tool trajectory of U, W, R combined at different symbols, as shown in the figure 3-23：

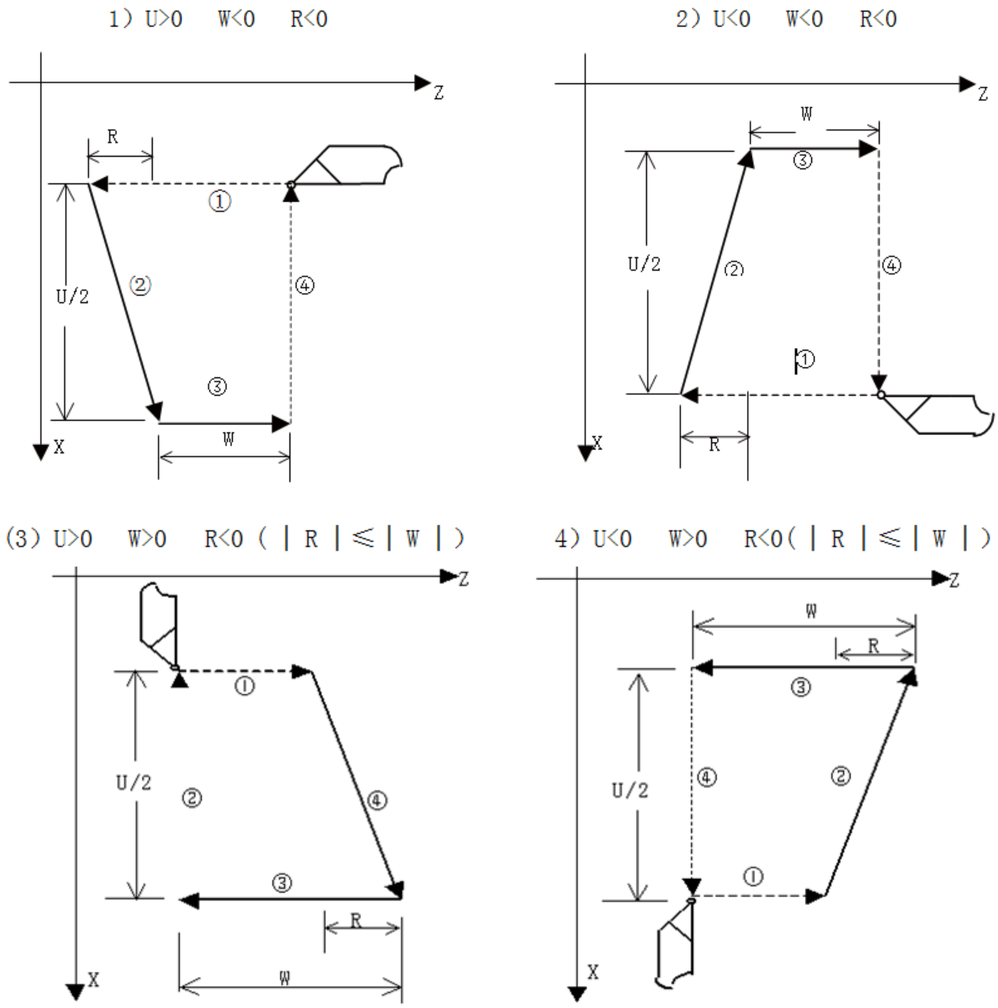


Figure 3-23

Example: Fig. 3-24, Blank Φ125×112

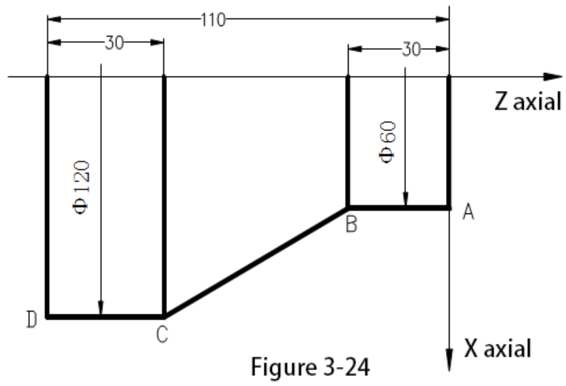


Figure 3-24

Procedure: O0003;
G00 X130 Z5 M3 S1;
G94 X0 Z0 F200;
X120 Z-110 F300; End face cutting (outer circle ∅ 120 cutting).
G00 X120 Z0;
G94 X108 Z-30 R-10;
X96 R-20;
X84 R-30; (C→ B→A, ∅60 cutting).
X72 R-40;
X60 R-50;
M30;

3.12.3 Considerations for Fixed Loop Codes

1) In the fixed loop code, X/U, Z/W, R, once executed, the new fixed loop code is re-given X/U, Z/W, R without executing When, the specified values for X/U, Z/W, R remain valid. If non-modal (group 00) G codes other than G04 or G00, G01, G02 are executed , G03, G32, the specified values for X/U, Z/W, R are cleared.

2) In the fixed loop G90 and G94 codes, if the single segment is running, the single segment stops after executing the entire fixed loop.

3.13 Multiple Loop Codes

The multiple cycle codes for lathe systems include: axial rough cycle G71, radial rough cycle G72, closed cutting cycle G 73, precision machining cycle G70, axial groove multi-cycle G74 3. Radial groove multi-cycle G75 and multi-thread cutting cycle G76. When the system executes this code

According to the programming trajectory, the amount of infeed, the amount of retracted and other data automatically calculate the number of cutting times and the cutting trajectory, carry out multiple inlet → cutting → the retraction → the re-entry of the processing cycle, automatically complete the roughing and finishing of the workpiece blank, the starting point and end point of the code are the same.

The G76 Multi-Thread Cutting Cycle Code is described in the Thread Functions section.

3.13.1 Axial rough car cycle G71

Code format: G71 U(△d)R(e) F_ S_ T_;　　　　　(1)

G71 P（ns）Q（nf）U（△u）W（△w）K0/1; （2）

N（ns） G0/G1 X/U..;
... ... F_ ;
... ... S_ ;　　　　　　　　　　（3）
... ... ;
... ... ;
N（nf） ... ...;

Code features: The G71 code is divided into three sections:

(1): The program segment of the cutting amount, the amount of retracting and cutting speed, the spindle speed and the tool function when the rough turning is given;

(2): Given the program segment interval that defines the refined car trajectory, the program segment of the refined car margin;

(3): Define several consecutive program segments of the fine car trajectory, when executing G71 , these program segments are only used to calculate the trajectory of the rough car, and are not actually executed.

The system automatically calculates the roughing route according to the data such as fine turn trajectory, fine turning margin, feed amount, and withdrawal amount, cuts in a direction parallel to the Z axis, and completes the roughing of the workpiece through the cutting cycle of multiple tool entry → cutting → retracting. The start and end points of the G71 are the same. This code applies to molded rough cars of non-formed blanks (bar stock).

Related Definitions:

Finishing trajectory: The finishing trajectory of the workpiece given by part (3) of the code (ns ~ nf program segment), the starting point of the finishing trajectory (that is, the starting point of the ns program segment) and The starting and ending points of G71 are the same, referred to as point A; The first stage of the finishing trajectory (ns program segment) can only be the rapid movement of the X axis or the cutting feed, and the end point of the ns program segment is referred to as point B; The end point of the finishing trajectory (the end point of the nf sequence segment) is referred to as point C. The tracks are point A→ point B→ point C.

Rough car profile: The trajectory formed by the G71 is formed by the trajectory profile formed by the execution of G71 after the trajectory is offset by the lean car margin (Δ u, Δw). The A, B, and C points of the finishing trajectory correspond to the A', B', and C' points of the rough car contour after they are offset. The final continuous cutting trajectory of the G71 code is B' point → C' point.

Δd: The amount of X-axis cut at rough turning, in the range of 0.001 (IS_B)/0.0001 (IS_C) ~ 99.999 (Units: mm/inch, radius value), unsigned, the direction of the knife is determined by the direction of movement of the ns program segment. After U(Δd) is executed, specify the value Δd hold and convert this data to the corresponding value saved in the data parameter NO.051. When U(Δd) is not entered, the value of the data parameter NO.051 is used as the feed amount.

e: The amount of retracted X-axis in rough car, value range 0 ~ 99.999 (unit: mm/inch, radius value), unsigned, The retract direction is opposite to the inlet direction, after R(e) is executed, the specified value e is maintained, and the conversion of this data to the corresponding value is saved in the data parameter NO.052. When R(e) is not entered, the value of the data parameter NO.052 is used as the retract amount.

ns: The program segment number of the first program segment of the fine car trajectory.

nf: The program segment number of the last program segment of the fine car trajectory.

Δu: Finishing allowance for the X axis, value range -99999.999 ~ 99999.999 (diameter/ radius specified), signed, The X-axis coordinate offset of the rough car profile relative to the fine car trajectory, that is, the difference between the absolute coordinates of the X axis of the A' point and the A point. When U(Δu) is not entered, the system presses Δu=0, that is, the rough cycle X axis leaves no finishing margin.

Δw: Finishing allowance of the Z axis, value range -99999.999 ~ 99999.999, signed, the thick car outline is offset from the Z axis coordinates of the finished car trajectory, that is: A The difference between the point and the absolute coordinate of the Z axis of point A. When W(Δw) is not entered, the system presses Δw=0, that is, the rough car circulation Z axis leaves no finishing margin.

K: When K is not entered or K is not 1, the system does not check the monotonicity of the program; When K=1, the system checks the monotonicity of the program.

F: Cutting feed speed; S: Spindle speed; T: tool number, tool offset number.

M, S, T, F: Can be in the first G71 code or the second G71 code, or in ns ~ nf Specified in the program (except for the T directive). In the G71 loop, the M, S, F functions of the program segment number between ns ~ nf are invalid, only if they are present The G70 is only valid in the program segment of the fine car cycle.

1) **Code execution process**: Figure 3-25.

(1) Quickly move from the starting point A point to the A' point, the X axis moves Δu, the Z axis moves Δw;

(2) Move Δd (incoming knife) from the X axis of point A', and when the ns program segment is G0, enter the tool at a fast moving speed, ns When the program segment is G1, the cutting feed speed F of G71 is fed, and the direction of the feed is consistent with the direction of point A → point B;

(3) Z-axis cutting feed to the rough car profile, the feed direction is consistent with the change of B point → C point Z axis coordinates;

(4) The X axis and Z axis retract the knife according to the cutting feed speed e (45 ° straight line), and the direction of the knife withdrawal is opposite to the direction of the knife entry of each axis;

(5) The Z axis retreats at a fast moving speed to the same position as the absolute coordinates of the Z axis at point A';

(6) If the X axis is re-entered (Δd+e), the end point of the movement is still in the middle of the connection between point A' → B' point (Failing to reach or exceed the B' point), the X axis is fed again (Δd + e) and then executed (3); If the X axis is re-entered (Δd+e), the end point of the

movement reaches the B' point or exceeds the A' point→ B' point is online, X axis is fed to B' point, and then executed (7);

(7) Cutting feed from point B' to point C' along the contour of the rough turn;

(8) Quickly move from the C' point to the A point, the G71 loop execution ends, and the program jumps to the next program segment of the nf program segment for execution.

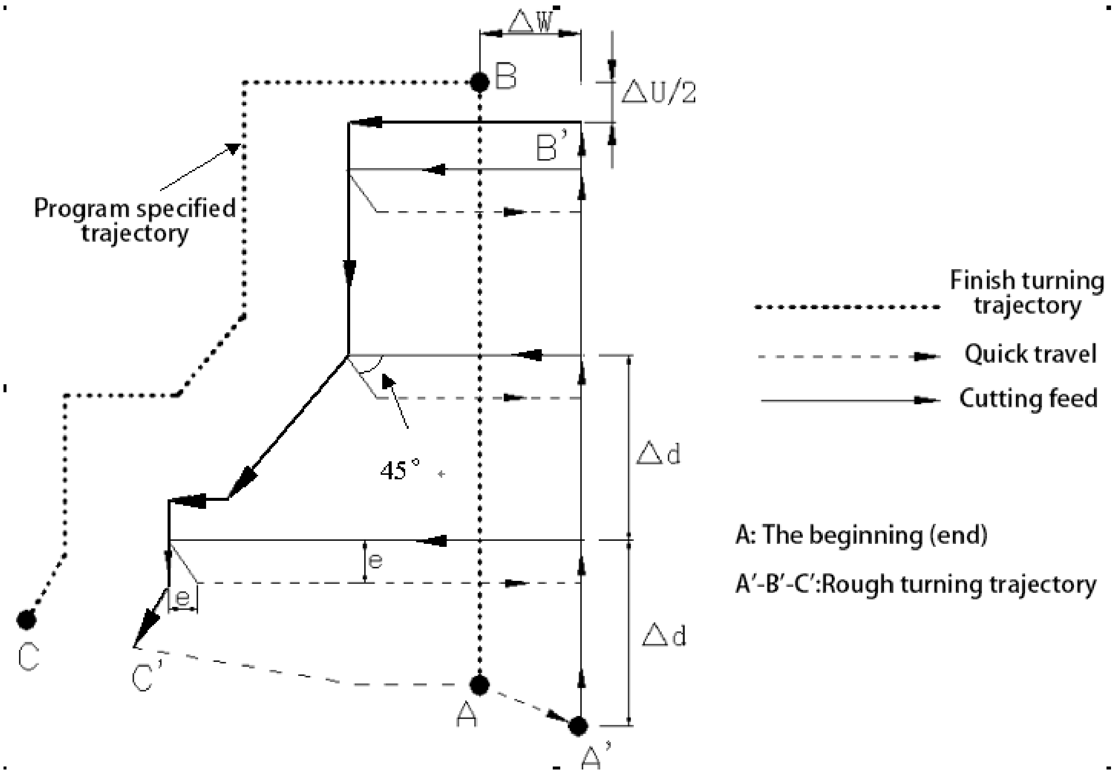


Figure 3-25 G71 code loop trajectory

2) Coordinate offset direction when leaving the balance of the fine car:

Δu, Δw reflect the coordinate offset and entry direction when finishing, and there are four different combinations of symbols according to Δ u and Δw, see Figure 3-26, and B → in Figure C is the fine car trajectory, B'→ C' is the thick car contour, and A is the starting point.

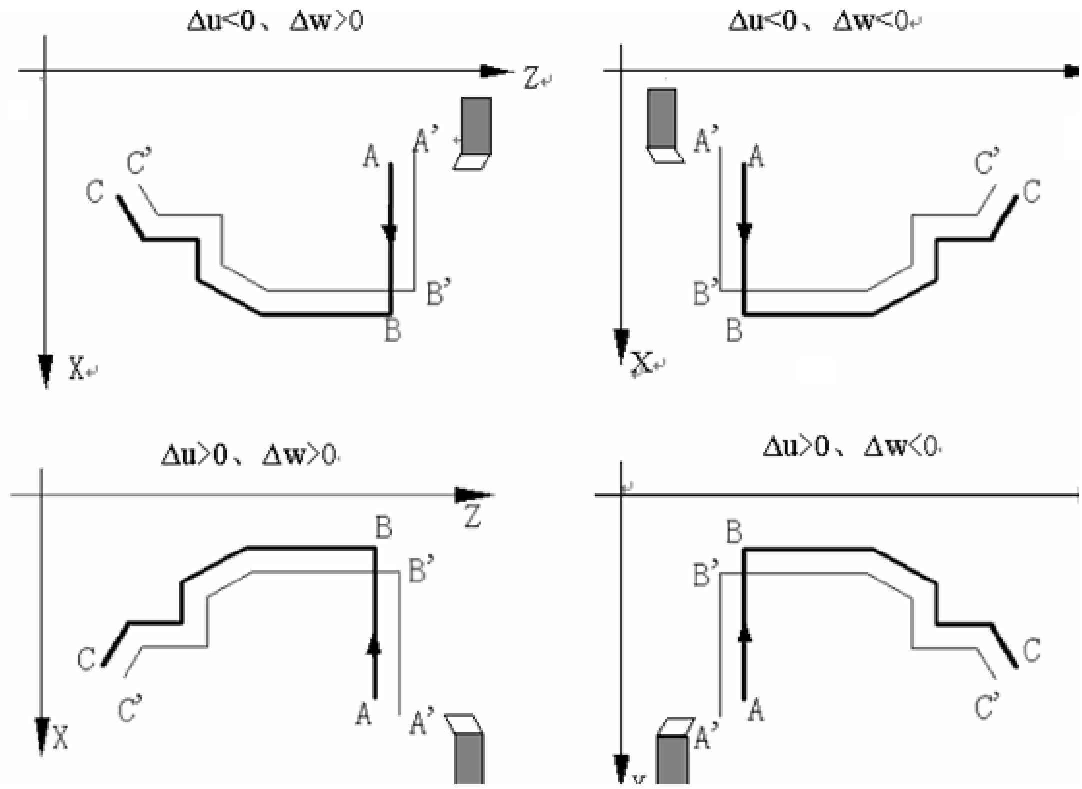


Figure 3-26

Notes:

● The ns program segment can only be G00, G01 code.

● Fine car trajectory (ns ~ nf program segment), the size of the X axis, Z axis must be monotonical change (has been increasing or decreasing all the time). 。

● The ns ~ nf segment must be written immediately after the G71 segment.

● When executing G71, the ns ~ nf program segment is only used to calculate the rough car contour, and the program segment is not executed. The F, S, T code in the ns ~ nf program segment is invalid when executing the G71 loop; When performing the G70 finishing cycle, the F,S code in the ns ~ nf segment is valid.

● ns ~ nf program segment, can only have G function: G00, G01, G02, G03, G04、

G96, G97, G98, G99, G40, G41, G42 codes; There can be no subroutine call code (such as M98/M99).

● G96, G97, G98, G99, G40, G41, G42 codes are executing Invalid in the G71 cycle, valid when performing the G70 finishing cycle.

● During G71 code execution, the automatic operation can be paused and manually moved.

● Perform a single-segment operation, and the program pauses after running the end of the current trajectory.

● △d, △u are all specified with the same address U, and the distinction is based on whether the program segment has specified P, Q code.

● The G71 code cannot be executed in the input mode, otherwise an alarm will be generated.

When compound loop code needs to be used multiple times in the same program, ns~nf does not allow the same program segment number.

● The retract point should be as high or low as possible to avoid the retracting tool touching the workpiece.

Example: Figure 3-27

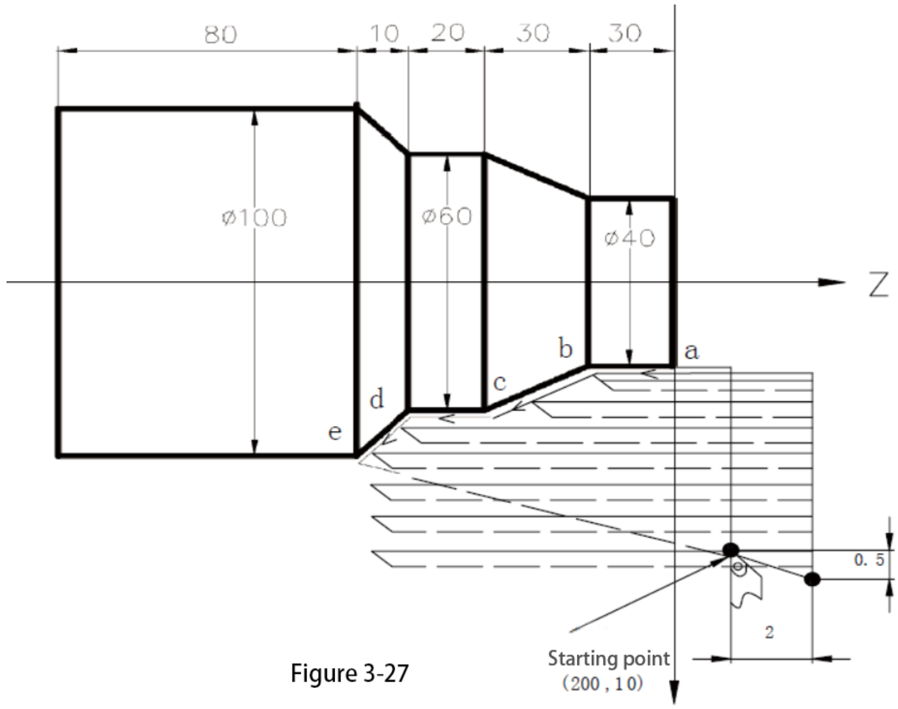


Figure 3-27

Procedure: O0004;

G00 X200 Z10 M3 S800; (Counterclockwise rotation, speed 800r/min)

G71 U2 R1 F200; (Each cut depth is 4mm, retractable 2mm, [diameter]).

G71 P80 Q120 U1 W2; (For a---e rough turning, allowance X direction 1mm, Z direction 2mm).

N80 G00 X40 S1200 ; (Positioning).

G01 Z-30 F100; (a → b)

X60 W-30; (b → c)

W-20; (c → d)

N120 X100 W-10; (d → e)

Finishing route a → b → c → d → e program segments

G70 P80 Q120; (to a---e finishing processing).

M30; (End of procedure).

3.13.2 Radial rough car cycle G72

Code format: G72 W(△d)R(e)F_ S_ T_; （1）

G72 P（ns）Q（nf）U（△u）W（△w）K0/1; （2）

N（ns） G0/G1 X/U..;
... ... F_ ;
... ... S_ ;
... ... ;
... ... ;
N（nf） ... ...;

（3）

Code features: The G72 code is divided into three sections:

(1): The program segment of the cutting amount, the amount of retracting and cutting speed, the spindle speed and the tool function when the rough turning is given;

(2): Given the program segment interval that defines the refined car trajectory, the program segment of the refined car margin;

(3): Define several consecutive program segments of the fine car trajectory, when the G72 is executed, these program segments are only used to calculate the trajectory of the rough car, and are not actually executed.

The system automatically calculates the roughing route according to the data such as fine turn trajectory, fine turning margin, tool feed amount, and withdrawal amount, cuts in a direction parallel to the X axis, and completes the roughing of the workpiece through the cutting cycle of multiple tool entry → cutting → retracting, and the starting and ending points of G72 are the same. This code applies to molded rough cars of non-formed blanks (bar stock).

Related Definitions:

Finishing trajectory: The finishing trajectory of the workpiece given by part (3) of the code (ns ~ nf program segment), the starting point of the finishing trajectory (that is, the starting point of the ns program segment) and G72 has the same starting and ending points, referred to as point A; The first stage of the finishing trajectory (ns program segment) can only be the fast movement of the Z axis or the cutting feed, and the end point of the ns program segment is referred to as the B point; The end point of the finishing trajectory (the end point of the nf program segment) is referred to as point C. The tracks are point A→ point B→ point C.

Rough car profile: The trajectory of the fine car is offset by the balance of fine car (Δu, Δw), which is the trajectory profile formed by the execution of G72. The A, B, and C points of the finishing trajectory correspond to the A', B', and C' points of the rough car contour after they are offset. The final continuous cutting trajectory of the G72 code is B' point → C' point.

Δd: The amount of cutting of the Z-axis during rough turning, in the range of 0.001 (IS_B)/0.0001 (IS_C) ~ 99.999 (Units: mm/inch), unsigned, the direction of the knife is determined by the direction of movement of the ns program segment. After W(Δd) is executed, the specified value Δd is maintained, and the conversion of this data into the corresponding value is saved in the data parameter NO.051. When W(Δd) is not entered, the value of the data parameter NO.051 is used as the feed amount.

e: The amount of retracted in the Z axis when rough turning, the value range is 0 ~ 99.999 (unit: mm/inch), unsigned, the direction of retracting the knife is opposite to the direction of the knife, after R(e) is executed, the value is specified e Hold and convert that data to the corresponding value saved in data parameter NO.052. When R(e) is not entered, the value of the data parameter NO.052 is used as the retract amount.

ns: The program segment number of the first program segment of the fine car trajectory.

nf: The program segment number of the last program segment of the fine car trajectory.

Δu: The finishing allowance left by the X-axis during rough turning, the value range is -99999.999 ~ 99999.999 (The X-axis coordinate offset of the rough car profile relative to the finishing trajectory, that is: A 'Difference between the point and the absolute coordinates of the X axis of point A, diameter/radius specified, signed).

Δw: The finishing allowance left by the Z axis during the rough car, the value range is -99999.999 ~ 99999.999 (The contour of the rough car is offset from the Z axis coordinate of the finished car trajectory, that is: A 'The difference between the point and the absolute coordinates of the Z axis of point A, signed) .

K: When K is not entered or K is not 1, the system does not check the monotonicity of the program; When K=1, the system checks the monotonicity of the program.

F: Cutting feed speed;

S: Spindle speed;

T: tool number, tool offset number.

M, S, T, F: Can be in the first G72 code or the second G72 code, or in ns ~ nf Specified in the program (except for the T directive). In the G72 loop, the M, S, F functions of the program segment number between ns ~ nf are invalid, only in the presence The G70 is only valid in the program segment of the fine car cycle.

Code execution: Figure 3-28.

1. Quickly move from the starting point A to the A' point, the X axis moves Δu, the Z axis moves Δw;

2. Move Δd(incoming) from the Z axis of A' point, and the ns program segment is G0 when the fast moving speed is entered, ns The program segment is G1 according to the G72 cutting feed speed F into the tool, the direction of the inlet is consistent with the direction of point A → point B;

3. X-axis cutting feed to the rough car profile, the feed direction is consistent with the B point → C point X axis coordinate change;

4. The X axis and the Z axis retract the knife according to the cutting feed speed e (45 ° straight line), and the direction of the knife withdrawal is opposite to the direction of the knife feed of each axis;

5. The X axis retreats at a fast moving speed to the same position as the absolute coordinates of the A' point Z axis;

6. If the Z axis is re-entered (Δd+e), the end point of the movement is still in the middle of the connection between point A' → B' point (Failing to reach or exceed the B' point), the Z axis is again fed (Δd+e) and then executed (3); If the Z axis is re-entered (Δd+e), the end point of the

movement reaches the B' point or exceeds the A' point → B' point is connected, Z axis is fed to B' point, and then executed (7);

7. Cut the feed from the B' point along the rough car contour to the C' point;

8. Quickly move from point C' to point A, the G72 loop execution ends, and the program jumps to the next program segment of the nf segment for execution.

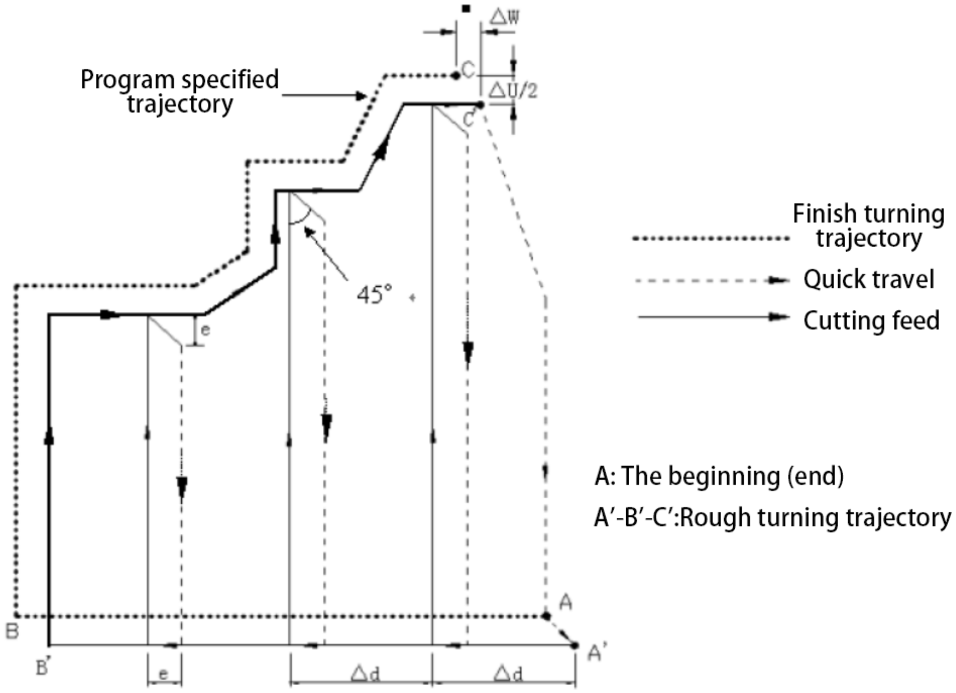


Figure 3-28

Code Description:

● The ns ~ nf segment must be written immediately after the G72 program.

●When executing G72, the ns ~ nf program segment is only used to calculate the rough car contour, and the program segment is not executed. The F, S, M code in the ns ~ nf segment is invalid when executing the G72 loop. When performing the G70 finishing cycle, the F, S, M code in the ns ~ nf segment is valid.

● The ns program segment can only be G00 and G01 codes without X/U code words, otherwise the alarm is alarmed.

●Precision trajectory (ns ~ nf program segment), the size of the X axis and Z axis must be monotonous (always increasing or decreasing all the time). 。

● ns ~ nf program segment, can only have G function: G00, G01, G02, G03, G04、G96、G97、G98、G99、G40、 G41, G42 code; There can be no subroutine call code (such as M98/M99).

● G96, G97, G98, G99, G40, G41, G42 codes are executing Invalid in the G72 cycle, valid when performing the G70 finishing cycle.

● During G72 code execution, the automatic run can be paused and moved manually.

● Perform a single-stage operation, and the program pauses after running the end point of the current trajectory.

●△ d, △ w are all specified with the same address W, which is distinguished by whether the program segment has specified P, Q code words.

● When you need to use compound loop code multiple times in the same program, ns ~ nf does not allow the same program segment number.

●The G72 code cannot be executed in the input mode, otherwise an alarm will be generated.

●The retract point should be as high or low as possible to avoid the retracting tool touching the workpiece.

Coordinate offset direction when leaving the balance of the fine car:

Δu, Δw reflect the coordinate offset and entry direction when finishing, and there are four different combinations of symbols according to Δ u and Δw, see Figure 3-29, Figure: B → C is the finished car trajectory, B'→ C' is the thick car contour, and A is the starting point.

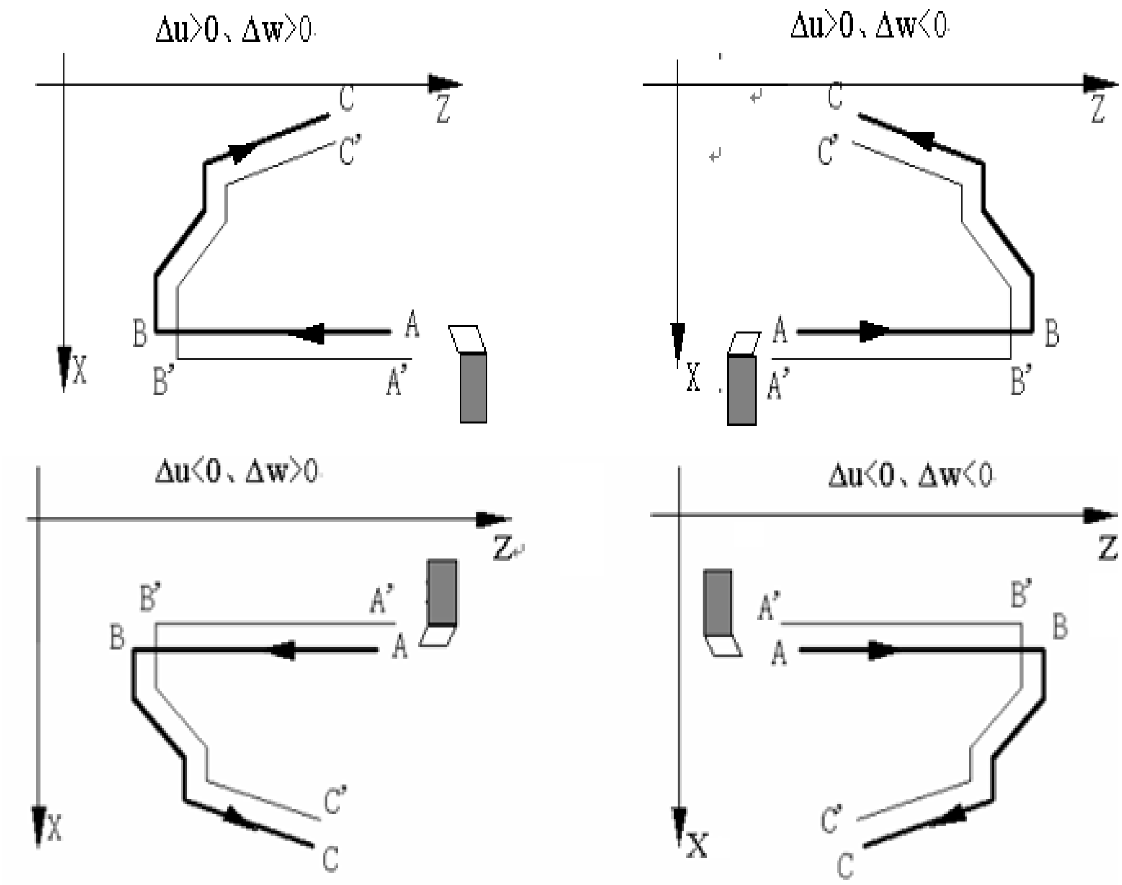


Figure 3-29

Example: Figure 3-30

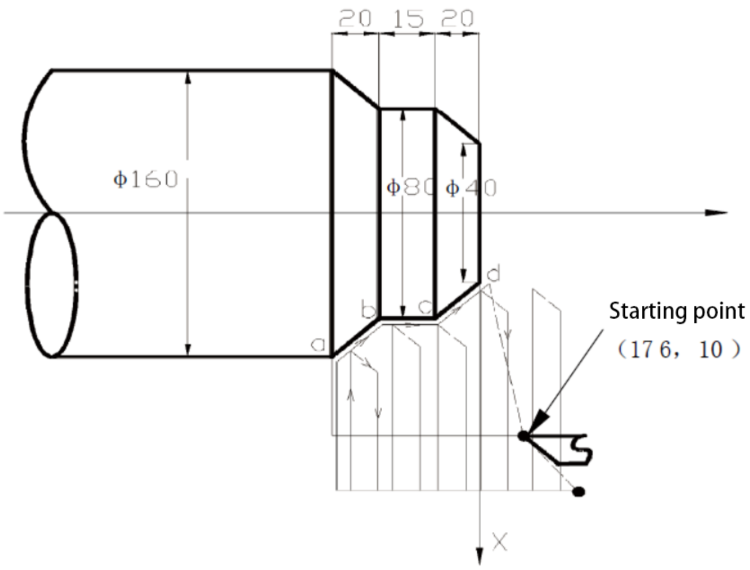


Figure 3-30

Procedure: O0005 ;

G00 X176 Z10 M03 S500 (Change the No. 2 knife, perform the No. 2 deflection, counterclockwise rotation, speed 500).

G72 W2.0 R0.5 F300； (The amount of knife feed is 2mm, the amount of retracting knife is 0.5mm).

G72 P10 Q20 U0.2 W0. 1 ； (For a - - d rough car, X stay 0 .2 m m, Z stay.)
0.1mm headroom).

N10 G00 Z-55 S800； (Fast Move).
G01 X160 F120； (Knife feed to point a).
X80 W20； (Processing a—b)
W15； (Processing b—c).
N20 X40 W20； (Processing c—d).

Finishing route program segment

G70 P010 Q020； (Finishing a-d).
M30；

3.13.3 Closed cutting cycle G73

Code format: G73 U(△i) W(△k)R(d) F_ S_ T_; （1）

G73 P（ns）Q（nf）U（△u）W（△w）; （2）

N（ns） G0/G1 X/U..;
... ... F_ ;
... ... S_ ; （3）
... ... ;
... ... ;
N（nf） ... ...;

Code features: The G73 code is divided into three sections:

(1) The program segment of the given tool withdrawal amount, the number of cutting times and the cutting speed, the spindle speed, and the tool function;

(2) Given the program segment interval that defines the refined car trajectory and the program segment of the refined car margin;

(3) Define several consecutive program segments of the fine car trajectory, which are used only to calculate the trajectory of the rough car when the G73 is executed

It is not actually executed.

The system automatically calculates the rough car offset, the single tool feed volume and the rough car trajectory according to the data such as the amount of fine turning, the amount of retracted tools, the number of cutting times, etc. The trajectory of each cutting is the offset of the fine car trajectory, the cutting trajectory is gradually close to the fine car trajectory, and the last cutting trajectory is the fine car trajectory offset by the fine car margin. The G73 starts and ends the same way, and this code applies to rough cars that form rough blanks. The G73 code is non-modal, and the code trace is shown in Figure 3-31.

Related Definitions:

Finishing trajectory: The finishing trajectory of the workpiece given by part (3) of the code (ns ~ nf program segment), the starting point of the finishing trajectory (that is, the starting point of the ns sequence segment). The same as the starting and ending points of G73, referred to as point A; The end point of the first segment of the finishing trajectory (ns program segment) is referred to as point B; The end point of the finishing trajectory (the end point of the nf program segment) is referred to as point C. The track is point A→ point B→ point C.

Rough car trajectory: A set of offset trajectories of the fine car trajectory, the number of rough tracks is the same as the number of cuttings. After the coordinate offset, the A, B, and C points of the fine car trajectory correspond to the An, Bn, and Cn points (n For the number of cuts, the first cut is represented as A1, B1, C1 points, and the last is represented as Ad, Bd, Cd points). The coordinate offset of the first cut relative to the fine turning trajectory is (Δi×2+Δu, Δw+Δk) (programmed by diameter), the coordinate offset of the last cut relative to the fine turning trajectory is (Δu, Δw), and the coordinate offset of each cut relative to the previous cutting trajectory is:

$$\left(-\frac{\triangle i \times 2}{1000 \times \mathrm{d} - 1},\ -\frac{\triangle k}{1000 \times d - 1} \right)$$

Δi: X shaft rough car retract volume, value range -99999.999 ~ 99999.999 (unit: mm, radius value), Δi Equal to the X axis coordinate offset (radius value) of point A1 relative to point Ad, X when rough The total amount of cutting (radius value) of the axis is equal to | Δi| , the X-axis is cut in the opposite direction to the symbol for Δ i: Δi > 0, cutting in the negative direction of the X-axis when rough turning. Δi Specifies that the value is persisted after execution and converts that data to the corresponding value is saved in the data parameter NO.053. When U(Δi) is not entered, the value of the data parameter NO.053 is used as the X-axis rough retract quantity.

Δk: Z-axis rough car retracting amount, the value range is -99999.999 ~ 99999.999 (unit: mm), Δk is equal The Z-axis coordinate offset of point A1 relative to Ad point, and the total amount of cut of the Z axis at rough turning is equal to | Δk| , the direction of cutting of the Z axis is opposite to the symbol of Δ k: Δk > 0, cutting in the negative direction of the Z axis when rough turning. Δk Specifies that the value is persisted after execution and converts this data to the corresponding value saved in the data parameter NO.054. When W(Δk) is not entered, the value of the data parameter NO.054 is used as the amount of rough tool retract for the Z-axis.

d: the number of cuts, the value range of 1 ~ 9999 (unit: times), R5 means that 5 times the cut completes the closed cutting cycle. R(d) specifies that the value is persisted after execution and modifies the value of the data parameter NO.055 to d (unit: times). When R(d) is not entered, the value of the data parameter NO.055 is used as the number of cuts. If the number of cuts is 1, the system will complete the closed cutting cycle by 2 cuts.

ns: The program segment number of the first program segment of the fine car trajectory.

nf: The program segment number of the last program segment of the fine car trajectory.

Δu: X axis finishing allowance, value range -99999.999 ~ 99999.999 (unit: mm, diameter / radius specified), the X axis coordinate of the last rough car trajectory relative to the fine car trajectory is offset, that is, the difference between the A1 point and the absolute coordinate of the X axis of the A point. Δu > 0, and the last rough trajectory is offset in the positive direction of the X axis relative to the finished car trajectory. When U(Δu) is not entered, the system presses Δu=0, that is, the rough cycle X axis leaves no finishing allowance.

Δw: Finishing allowance of the Z axis, value range -99999.999 ~ 99999.999 (unit: mm), the last rough car trajectory relative to the finishing trajectory Z Axis coordinate offset, i.e. the difference between the absolute coordinates of the A1 point and the Z axis of point A. Δw > 0, and the last rough trajectory is offset from the fine car trajectory in the positive direction of the Z axis. When W(Δw) is not entered, the system presses Δw=0, that is, the rough cycle Z axis leaves no finishing margin.

F: Cutting feed speed;

S: Spindle speed;

T: tool number, tool offset number.

M, S, T, F: The code word can be in the first G73 code or the second G73 code, or in ns ~ Except for the T directive specified in the nf program). In the G73 loop, the M, S, F functions of the program segment number between ns ~ nf are invalid, only in existence The G70 is only valid in the program segment of the fine car cycle.

Code execution: Figure 3-31.

(1) A → A1: Fast movement;

(2) The first rough car, A1 → B1 → C1:

A1 → B1:The ns program segment is G0 at the fast moving speed, and the ns program segment is G1 at the cutting feed rate specified by G73;

B1 → C1: Cutting feed.

(3) C1 → A2: Fast movement;

(4) The second rough car, A2 → B2 → C2:

A2 → B2:The ns program segment is the fast moving speed when G0, and the ns program segment is G1 at the cutting feed speed specified by G73;

B2 → C2: Cutting feed.

(5) C2 → A3: Fast movement;

....

Nth rough car, An → Bn → Cn:

An → Bn:the ns segment is the fast moving speed when G0, and the ns segment is the cutting feed speed specified by G73 when the NS segment is G1;

Bn → Cn: Cutting Feed.

Cn → An+1: Fast Moving;

....

The last rough car, Ad → Bd → Cd:

Ad → Bd:ns the cutting feed rate specified by G73 when the ns segment is G0 and G73 when the ns segment is G1;

Bd → Cd: Cutting feed.

Cd → A: Quickly move to the starting point;

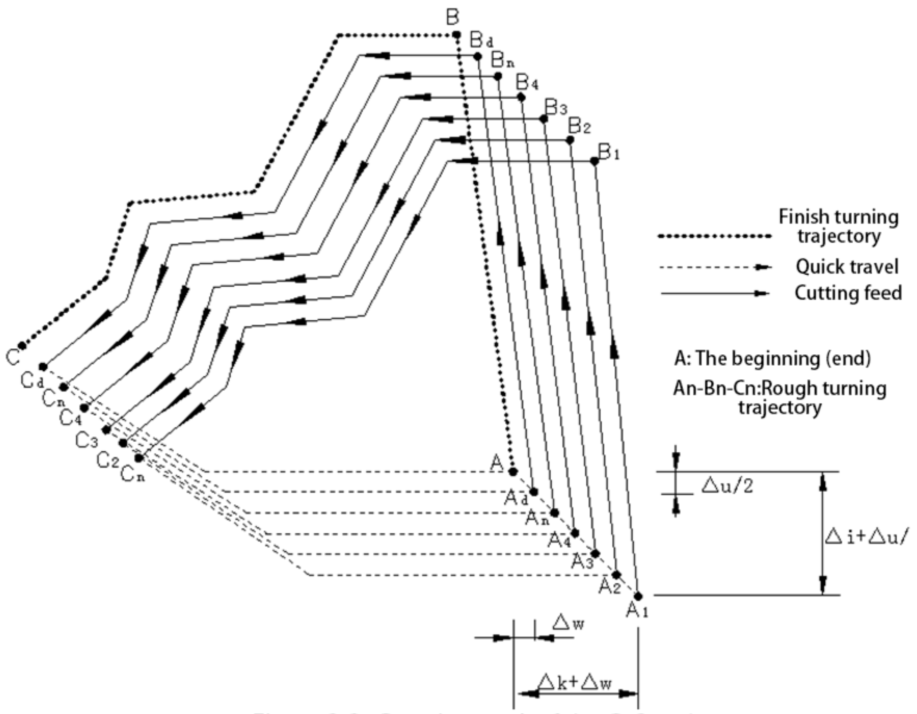


Figure 3-31 Running track of the G73 code

Code Description:

● The ns ~ nf segment must be written immediately after the G73 segment.

● When executing G73, the ns ~ nf program segment is only used to calculate the rough car contour, and the program segment is not executed. The F, S, M code in the ns ~ nf program segment is invalid when executing G73. When performing the G70 finishing cycle, the F, S, M code in the ns ~ nf segment is valid.

● The ns program segment can only be G00, G01 code.

● In the ns ~ nf program segment, there can only be the following G functions: G00, G01, G02, G03 、G04、G96、G97、G98、G99、G40 , G41, G42 code; The following M functions cannot be used: subroutine call code (such as M98/M99).

● G96, G97, G98, G99, G40, G41, G42 codes are executing Invalid in the G73 cycle, valid when performing the G70 finishing cycle.

● During G73 code execution, the auto run can be paused and moved manually.

● Perform a single-segment operation, and the program pauses after running the end of the current trajectory.

● Δi, △ u are all specified with the same address U, Δk, Δw are all specified with the same address W, the distinction is based on whether the program segment is specified or not P, Q code word.

● The G73 code cannot be executed in the input mode, otherwise an alarm will be generated.

● When you need to use the composite loop code multiple times in the same program, ns ~ nf does not allow the same program segment number.

● The retract point should be as high or low as possible to avoid the retracting tool touching the workpiece.

Coordinate offset direction when leaving the balance of the fine car:

Δi and Δk reflect the coordinate offset and entry direction of rough car time, and Δu and Δw reflect the coordinate offset and entry direction when finishing; Δi, Δk, Δu, Δw can have a variety of combinations, in general, Δi and Δu symbols are consistent, Δk is the same as the symbol of Δ w, and there are four common combinations, see Figure 3-32, in the figure: A is the starting point, B → C is the workpiece profile, B'→ C' is the rough car profile, and B'''→ C' is the finished car trajectory.

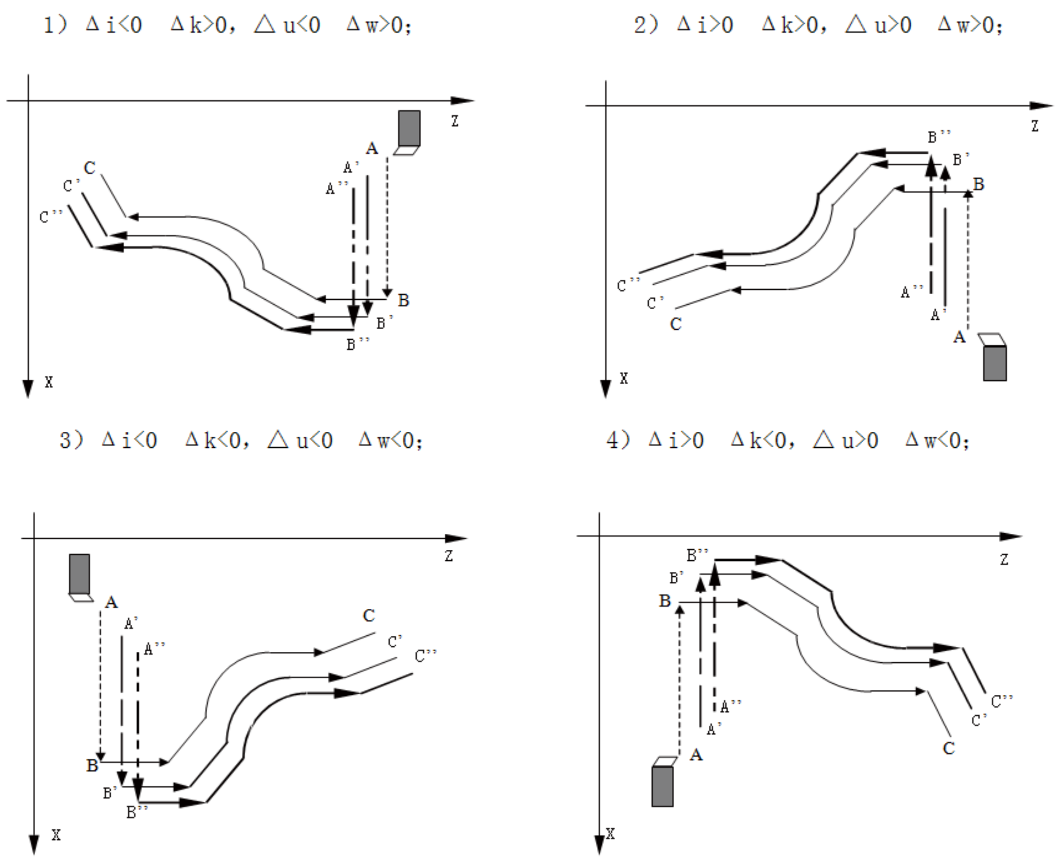


Figure 3-32

Example: Figure 3-33

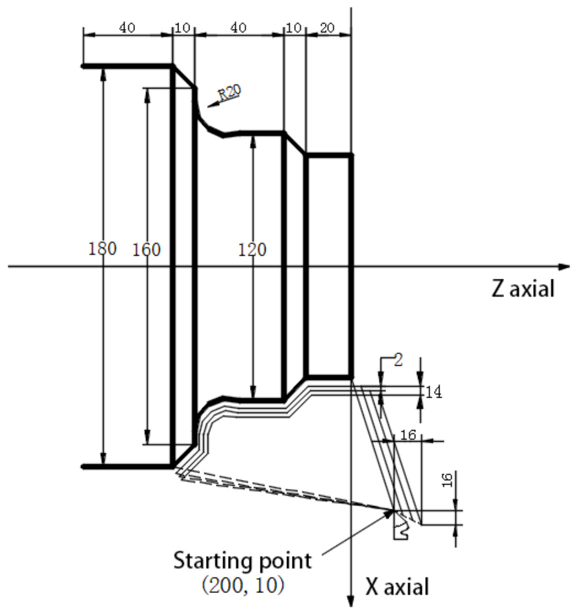


Figure 3-33

Procedure:

O0006；

G99 G00 X200 Z10 M03 S500； (Specify each turn feed, locate the starting point, start the spindle).

G73 U1.0 W1.0 R3； (X-axis retract 2mm, Z-axis retract 1mm).

G73 P14 Q19 U0.5 W0.3 F0.3； (Rough car, X-axis left 0.5mm, Z-axis left 0.3mm fine car margin)

N14 G00 X80 Z0；
G01 W-20 F0.15 S600；
X120 W-10； Finishing shape program segments
W-20；
G02 X160 W-20 R20；
N19 G01 X180 W-10；

G70 P14 Q19 M30； (Finishing).

3.13.4 Finishing cycle G70

Code format: G70 P(ns)Q(nf);;

Code function: The tool is finished from the starting point along the workpiece finishing trajectory given by the ns ~ nf program segment. After roughing on the G71, G72 or G73, the finishing turn is carried out with the G70 code, and the finishing allowance is cut in a single pass. At the end of the G70 cycle, the tool returns to the starting point and executes the next program segment after the G70 program segment.

where: ns: the program segment number of the first program segment of the fine car trajectory;

nf: the program segment number of the last program segment of the fine car trajectory;

The G70 code trace is determined by the programming trace of the program segment between ns ~ nf. The relative position relationship of ns and nf in the G70 ~ G73 program segment is as follows:

... ...;
G71/G72/G73...;
N（ns）... ...;
... ...;
... ... F_; Finishing route program segment group
... ... S_;
N（nf）... ...;
... ...;
G70 P（ns）Q（nf）;
... ...

Code Description:

● G70 must be written after the ns ~ nf segment.

● When performing the G70 finishing cycle, the F, S, M code in the ns ~ nf program segment is valid.

● G96, G97, G98, G99, G40, G41, G42 codes are executing G70 is effective during finishing cycles.

● During G70 code execution, the auto run can be paused and moved manually.

● Perform a single-stage operation, and the program pauses after running the end point of the current trajectory.

● The G70 code cannot be executed in the input mode, otherwise an alarm will be generated.

● When you need to use the composite loop code multiple times in the same program, ns ~ nf does not allow the same program segment number.

● The retract point should be as high or low as possible to avoid the retracting tool touching the workpiece.

3.13.5 Axial grooving multi-cycle G74

Code format: G74 R(e);;

G74 X/U_ Z/W_ P(Δi) Q(Δk) R(Δd) F ;

Code function: radial (X axis) feed cycle Composite axial intermittent cutting cycle: feed from the starting point axial (Z axis), refeed, refeed... Until the cutting reaches the same position as the Z axis coordinate of the cutting end point, then radial retract and axial retreat to the same position as the Z axis coordinate of the starting point, and complete an axial cutting cycle; After the radial re-entry, the next axial cutting cycle is carried out; After cutting to the end of the cut, return to the start point (the start and end points of the G74 are the same), and the axial groove compound cycle is completed. The G74's radial and axial approach directions are determined by the relative positions of the cutting end points X/U, Z/W and the starting point, and this code is used to machine annular grooves or central deep holes in the end face of the workpiece, and the axial intermittent cutting plays the role of chip breaking and timely chip removal.

Related Definitions:

Axial cutting cycle start point: The position where the axial feed starts at the beginning of each axial cutting cycle, expressed as An(n=1,2,3...).), The Z coordinate of An is the same as the starting point A, and the difference between The X coordinate of An and An-1 is Δi。 The start point A1 of the first axial cutting cycle is the same as the starting point A, and the X of the starting point (expressed as Af) of the starting point of the last axial cutting cycle The axis coordinates are the same as the cut end point.

Axial feed end point: The end point of each axial cut cycle, expressed as Bn(n=1,2,3...).), bn has the same Z axis coordinate as the cut end point, Bn has the same X axis coordinate as An, and the last axial feed end point (Expressed as Bf) and the cut end point is the same point;

Radial retract end point: After each axial cutting cycle reaches the axial approach end point, the end point position of the radial retract (retract amount is Δd) is expressed as Cn(n=1,2,3...).), the Z coordinate of Cn is the same as the end point of cutting, and the difference between cn and the An X axis coordinate is Δd; Axial cutting cycle end point: The end point position of the axial retract from the radial retract end point, expressed as Dn(n=1,2,3...).),, the Z coordinate of the Dn is the same as the starting point, and the X coordinate of the Dn is the same as that of cn (The difference from the X-axis coordinates is Δd) ;

Cut end: X/U Z/W specified position, last axial feed end Bf.

R(e): The amount of axial retract after each axial (Z axis) inlet, the value range is 0 ~ 99.999 (unit: mm), unsigned. The specified value remains valid after R(e) execution and the conversion of this data to the corresponding value is saved in the data parameter NO.056. When R(e) is not entered, the value of the data parameter NO.056 is used as the axial retract amount.

X: The absolute coordinate value of the X axis of the cut end Bf (in mm).

U**:** The difference in mm) between the absolute coordinates of the X axis of the cut end Bf and the start point A.

Z: The absolute coordinate value of the Z axis of the cut end Bf (in mm).

W: The difference in absolute coordinates of the Z axis between the cut end point Bf and the start point A (in mm).

P(Δi): Radial of a single axial cutting cycle(X 轴) The amount of cutting, the range of values0 ＜ Δi ≤ 9999999(Unit:0.001mmdiameter/ The radius is specified)。

Q(Δk): Axial(Z 轴) When cutting,Z The amount of input of the shaft intermittently feeding, the value range0 ＜ Δk ≤ 9999999(Unit:0.001mm)。

R(Δd): The amount of radial (X-axis) retract after cutting to the end of axial cutting, with a value range of 0 ~ 99999.999 (Unit: mm, diameter / radius specified), when R(Δ d) is omitted, The system defaults to 0 radial (X-axis) after the axial cut end point. When the X/U and P(Δi) code words are omitted, the knife is retracted in the positive direction by default.

Code execution process: Figure 3-34.

(1) From the axial cutting cycle starting point An axial (Z axis) Cutting feed △ k, the cutting end point Z axis coordinate is less than the starting point Z axis coordinate, the direction The Z axis is fed negatively, and vice versa, the Z axis is fed positively;

(2) Axial (Z axis) fast movement retract e, retract direction is opposite to (1) feed direction;

(3) If the Z axis cuts the feed again (Δk+e), the feed end point is still between the axial cutting cycle start Point An and the axial feed end point Bn, Z The shaft cuts the feed again (Δk+e) and then executes (2); If the Z axis cuts the feed again (Δk+e), the feed end point reaches bn or is not with An Between bns, the Z-axis cutting feeds to the Bn point, and then performs (4);

(4) Radial (X axis) Quickly move the retracted tool △ d (radius value) to the Cn point, Bf point (When the X coordinate of the cut end point is smaller than the X coordinate of the A point (start point), the X coordinate is oriented to X The shaft is forward to retract the knife, and vice versa to the X axis to the negative return;

(5) Axial (Z axis) Quickly move the retract to the Dn point, and the nth axial cutting cycle ends. If the current is not the last axial cutting cycle, execute (6); If the current is the last axial cutting cycle, execute (7);

(6) Radial (X axis) Move the knife fast, and the direction of the knife is opposite to the direction of (4) the return of the knife. If the X axis is fed (△ d + △ i) (radius value), the end point of the inlet is still with point A Between the Af points (the starting point of the last axial cutting cycle), the X axis moves quickly through the tool feed (△ d + △ i) (Radius Value), i.e. Dn → An+1, then execute (1) (Start next time axial cutting cycle); If the X-axis feed (△ d + △ i) (radius value) is followed, the end point of the tool feed reaches Af Point or not between Dn and Af point, the X axis quickly moves to the Af point, and then executes (1), starting the last axial cutting cycle;

(7) The X axis moves quickly back to the starting point A, and the G74 code execution ends.

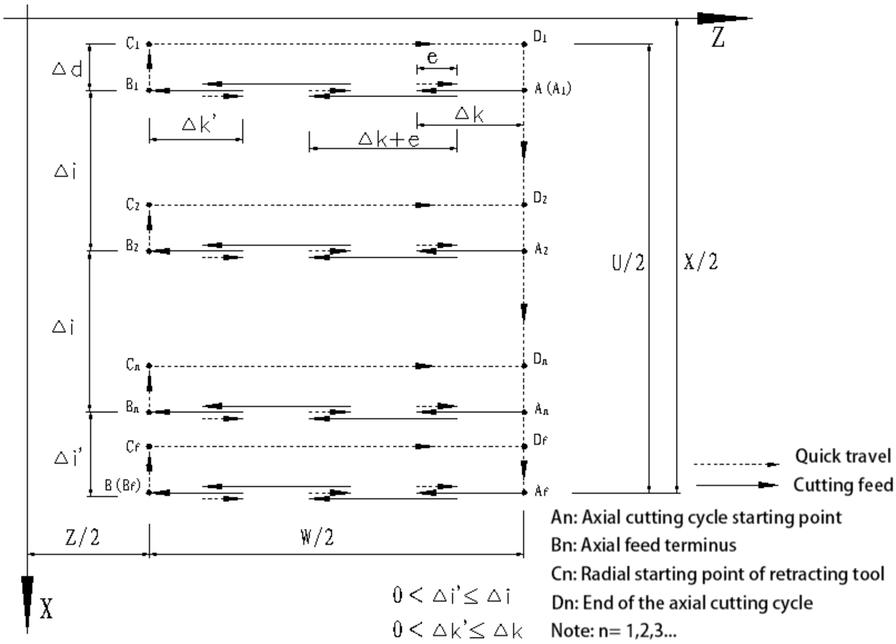


Fig. 3-34 G74 trajectory diagram

Code Description:

● The circular action is carried out by the G74 program segment containing Z/W and P(Δk), if only "G74 R(e) is executed ； "Program segment, system alarm;

● Δd and e are both specified with the same address R, the difference being whether there is Z/W and P(Δk) in the program segment Code word;

● During G74 code execution, the auto run can be paused and moved manually.

● Perform a single-segment operation, and the program pauses after running the end of the current trajectory.

● When performing blind hole cutting, the R(Δd) code word must be omitted, because there is no retract distance from cutting to the axial cutting end point.

Example: Figures 3-35

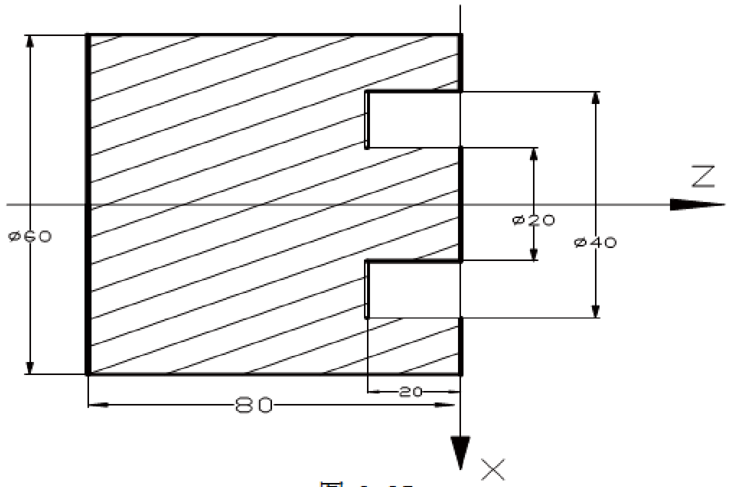


图 3-35

Program (assuming the grooving knife width is 4mm and the minimum increment of the system is 0.001mm):

O0007 ；

G0 X36 Z5 M3 S500；　(Start the spindle, position to the starting point of machining, X direction plus tool width).

G74 R0.5；　(Processing cycle).

G74 X20 Z-20 P3000 Q5000 F50；　(Z axis each time into the knife 5mm, retract 0.5mm, feed to the end (Z-20), quickly return to the starting point (Z5), X axis into the tool.) 3mm, cycle the above steps to continue running).

M30；　(End of procedure).

3.13.6 Radial groove multi-cycle G75

Code format: G75 R(e);;

G75 X/U_　Z/W_　P(Δi) Q(Δk) R(Δd) F ；

Code function: axial (Z axis) feed cycle Composite radial intermittent cutting cycle: from the starting point radial (X axis) feed, back,re-feed... Until the cutting reaches the same position as the X axis coordinate of the cutting end point, and then the axial retract and radial retreat to the same position as the X axis coordinate of the starting point, a radial cutting cycle is completed; After the axial re-entry, the next radial cutting cycle is carried out; After cutting to the end of the cut, return to the start point (the start and end points of the G75　are the same), and the radial groove compound cycle is completed. The G75's axial and radial approach directions are determined by the position of the cutting end point X/UZ/W　relative to the starting point, this code is used for machining radial annular grooves or cylindrical surfaces, radial intermittent cutting plays the role of chip breaking, timely chip evacuation.

Related Definitions:

Radial cutting cycle start: The position where each radial cutting cycle begins the radial feed, expressed as An(n=1,2,3...).), the X coordinate of An is the same as the starting point A, and An is the same as the Z of An-1 The difference between axis coordinates is Δk. The starting point of the first radial cutting cycle A1 is the same as the starting point A, and the Z of the starting point of the last radial cutting cycle (denoted as Af). The axis coordinates are the same as the cut end point.

End point of the approach: The end position of the radial feed of each radial cut cycle, expressed as Bn(n=1,2,3...). Bn has the same X axis coordinate as the cut end point, Bn has the same Z axis coordinate as An, and the last radial approach end point (Expressed as Bf) and the cut end point is the same point;

Axial retract end point: After each radial cutting cycle reaches the radial approach end point, the end point position of the axial retract (retract amount of Δd) is expressed as Cn(n=1,2,3...).), the X axis coordinate of Cn is the same as the end point of cutting, and the difference between cn and The Z axis coordinate is Δd;

Radial cut cycle end: The end point of the radial retract from the axial retract end point, expressed as Dn (n=1,2,3...).), the X coordinate of the Dn is the same as the starting point, and the Z coordinate of the Dn is the same as the Cn (and The difference between An Z coordinates is Δd);

Cut end point: X/U Z/W specified position, last radial approach end point Bf.

R(e): The amount of radial retract after each radial (X axis) inlet, the value range is 0 ~ 99.999 (unit: mm, radius value),, unsigned. The specified value remains valid after R(e) execution and the conversion of this data to the corresponding value is saved in the data parameter NO.056. When R(e) is not entered, the value of the system parameter NO.056 is used as the radial retract amount.

X: The absolute coordinate value of the X axis of the cut end Bf (in mm).

U: The difference between the absolute coordinates of the X axis of the cut end Bf and the starting point A (in mm).

Z: The absolute coordinate value of the Z axis of the cut end Bf (in mm).

W: The difference between the absolute coordinates of the Z axis of the cut end Bf and the starting point A (in mm).

P(Δi): Radial(X 轴) When entering the knife,X The amount of input of the shaft intermittently feeding, the value range0 ＜ Δi ≤ 9999999(Unit:0.001mmdiameter/ The radius is specified)。

Q(Δk): The axial direction of a single radial cutting cycle(Z 轴) The amount of knife fed, the range of values0 ＜ Δk ≤ 9999999(Unit:0.001mm)。

R(Δd): The amount of axial (Z-axis) retract after cutting to the radial cutting end point, the value range is 0 ~ 99999.999 (unit: mm)。

When R(Δd) is omitted, the system defaults to 0 in the axial (Z axis) after the radial cutting end point.

Omit Z/W and Q(<u>Δk</u>) and retract the knife in the positive direction by default.

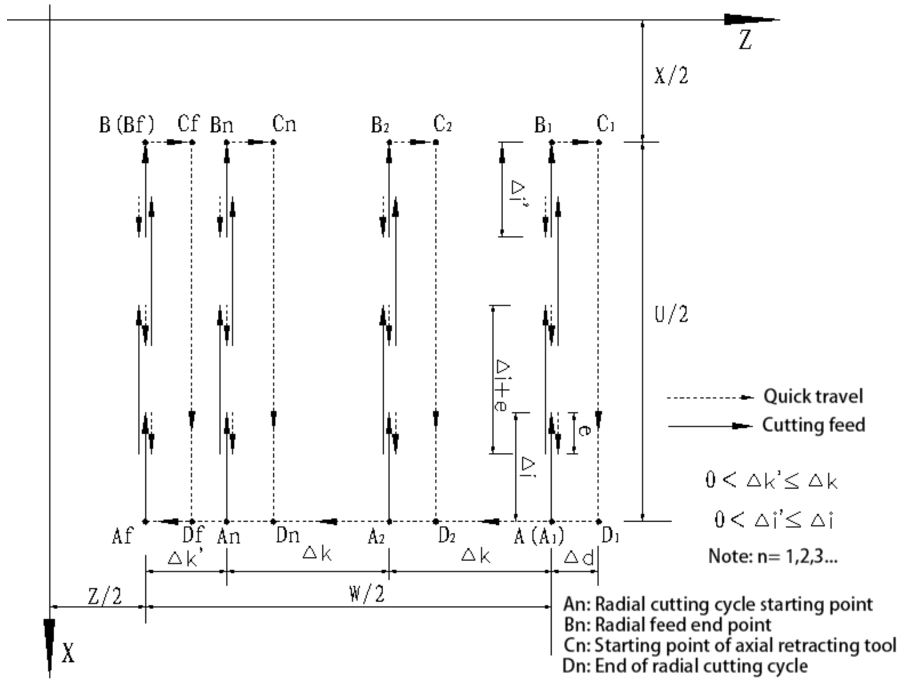


Figure 3-36 G75 trajectory diagram

Code execution: Figure 3-36

(1) Cut feed from the beginning of the radial cutting cycle An radial (X axis) △ i, when the X axis coordinate of the cutting end point is smaller than the X axis coordinate of the starting point, the direction The X axis is fed negatively, and vice versa is fed positively to the X axis;

(2) Radial (X axis) fast movement retract e, retract direction opposite to (1) feed direction;

(3) If the X axis cuts the feed again (Δi + e), the feed end point is still between the radial cutting cycle start point An and the radial approach end point Bn, X The shaft cuts the feed again (Δi+e) and then executes (2); If the X axis cuts the feed again (Δi+e), the feed end reaches the Bn point or is not with An Between bns, the X-axis cutting feeds to the Bn point, which is then performed (4);

(4) Axial (Z axis) Quickly move the retract △ d to the Cn point, the Bf point (cutting end point). When the Z axis coordinate is less than the Z axis coordinate of the A point (starting point), the knife is retracted in the positive direction of the Z axis, and vice versa to Z Axial negative retract;

(5) Radial (X-axis) Quickly move the retract to the Dn point, and the nth radial cutting cycle ends. If the current is not the last radial cutting cycle, execute (6); If the current is the last radial cutting cycle, execute (7);

(6) Axial (Z axis) Move the knife fast, and the direction of the knife is the opposite of (4) the direction of the retract. If the Z axis is fed (△ d + △ k), the end point is still at point A and point Af (the start of the last radial cutting cycle) between the Z axis moves quickly (△ d + △ k) , i.e. Dn → An+1 and then executes (1) (to start the next radial cutting cycle); If the Z axis is followed by a tool feed (△ d + △ k), the end point of the tool feed reaches the Af point or is not

in Dn Between the Af point, the Z axis moves quickly to the Af point, and then executes (1), starting the last radial cutting cycle;

(7) Z axis moves quickly back to the starting point A, and G75 code execution ends.

Code Description:

● The circular action is carried out by the G75 program segment containing X/U and P(Δi), if only "G75 R(e)" is executed; "Program segment, system alarm;

● Δd and e are both specified with the same address R, the difference being whether X/U and P(Δi) are present in the program segment Code word;

● During G75 code execution, the automatic operation can be paused and manually moved;

● Perform a single-segment operation, and the program pauses after running the end of the current trajectory.

● When performing a grooving cycle, the R(Δd) code word must be omitted, because there is no retract distance from cutting to the radial cutting end point.

Example: Figures 3-37

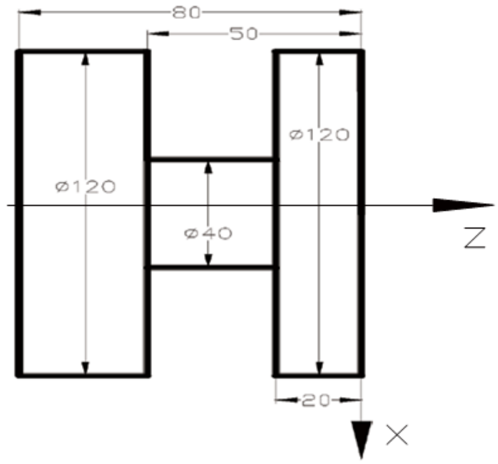


Figure 3-37 Cutting diagram of G75 code

Procedure (assuming the width of the grooving knife is 4mm, the minimum increment of the system is 0.001mm) :

O0008 ;

G00 X150 Z50 M3 S500; (Start spindle, set speed 500)

G0 X125 Z-24; (Positioned to the starting point of machining, Z direction plus tool width).

G75 R0.5 F150; (Processing cycle).

G75 X40 Z-50 P6000 Q3000; (X-axis each tool feed 6mm, retract 0.5mm, feed to finish (X40), quickly return to the starting point (X125), Z-axis feed.) 3mm, cycle the above steps to continue running).

G0 X150 Z50; (Return to the processing start point).

M30; (End of procedure).

3.14 Thread cutting codes

The lathe system has a variety of thread cutting functions for machining single-head, multi-head, variable-lead threads and tapping cycles (F units are inches / guide for imperial input, F units are millimeters / lead for metric inputs,). I Specifies that the number of teeth per inch of thread is independent of metric), the length and angle of the thread ejection are variable, and the multi-cycle thread cutting can be cut unilaterally, protecting the tool and improving the surface finish. Thread features include: Continuous Thread Cutting Code G32, Variable Pitch Thread Cutting Code G34, Flexible Tapping Cycle Cutting Code G33 (not currently enabled), Thread Cycle Cutting Code G92, thread multi-cycle cutting code G76.

The machine tool using the thread cutting function must be equipped with a spindle encoder, and the number of spindle encoder lines is set by parameter NO.070, and the ratio of spindle to encoder is set by parameters NO.110 and NO.111. When cutting threads, the system receives a signal from the spindle encoder to move the X axis or Z axis and start thread processing, so as long as the spindle speed is not changed, it can be cut multiple times in rough and fine cars to complete the processing of the same thread.

The lathe system has a variety of thread cutting functions can be used to machine threads without retract slots, but because there is an acceleration and deceleration process in the X axis and Z axis at the beginning and end of thread cutting, the pitch error at this time is large, so it is still necessary to leave the distance between the actual thread starting point and end of the thread introduction length and the retract.

Under the condition that the thread pitch is determined, the movement speed of the X axis and Z axis during thread cutting is determined by the spindle speed and has nothing to do with the cutting feed rate magnification. When the spindle magnification control changes to 100% when the spindle speed changes, the pitch will be error due to the acceleration and deceleration of the X axis and the Z axis, so do not adjust the spindle speed when thread cutting, let alone stop the spindle, and the spindle stop will cause damage to the tool and workpiece.

3.14.1 Equal pitch thread cutting code G32

Code format: G32 X/U_ Z/W_ F(I)_ J_ K_ Q_

Code function: the motion trajectory of the tool is a straight line from the start point to the end point; Displacement from the start point to the end point (X axis by radius value) The larger coordinate axis is called the long axis, the other coordinate axis is called the short axis, the spindle moves a lead each turn of the long axis during the movement, the short axis and the long axis are inserted in a straight line, when the tool cuts the workpiece, a spiral groove with equal pitch is formed on the surface of the workpiece to achieve the machining of the equal pitch thread. F, I code words are used for a given thread pitch, and the G32 code can be executed to machine straight threads with equal pitches, taper threads and end face threads, and continuous multi-segment threading.

Code description: G32 is a modal G code;

The lead of the thread refers to the displacement of the spindle by a long axis (the displacement of the X axis is based on the radius value);

When the X coordinate value of the start and end points are the same (do not enter X or U), straight thread cutting is performed;

End face thread cutting is performed when the Z coordinate values of the start and end points are the same (without entering Z or W);

The X and Z coordinate values of the start and end points are not the same, and the taper thread cutting is performed.

Related Definitions:

F: Specify the thread lead, the amount of movement of the major axis to rotate the long axis, and the value range0 ＜ F ≤ 500mm(Imperial input is0 ～50inch)，F The specified value remains valid after execution until the given thread pitch is performed againFCode word.

I: Specifies the number of teeth per inch of thread, which is the number of teeth threaded on the length of 1 inch (25.4mm) in the direction of the major axis, and can also be understood as the number of turns of the spindle rotation when the major axis moves 1 inch (25.4mm). The value range is 0.06 to 25400 teeth/inch, and the I specified value remains valid after execution until the I code word for the given thread pitch is executed again. Both metric inputs and imperial inputs represent the number of teeth per inch of thread.

J: The amount of movement in the short axis direction when the thread is tailed (the amount of tailing), with a positive and negative direction; If the minor axis is the X axis, the value is specified for the radius; The J-value is a non-modal parameter.

K: The length in the direction of the long axis when the thread is retracted. If the major axis is the X axis, the value is specified as the radius; Without direction; The K value is a nonmodal parameter.

Q: The starting angle refers to the offset angle between the spindle turn signal and the starting point of the thread cutting. The value range is 0 ~ 360000 (unit: 0.001 degrees).

The Q value is a non-modal parameter that must be specified for each use and is considered 0 degrees if not specified.

Q Usage rules:

1. If you do not specify Q, the default is 0 degrees of the starting angle;

2. For continuous thread cutting, in addition to the Q of the first segment, the Q specified in the subsequent thread cutting section is invalid, and even if Q is defined, it is ignored;

3. The total number of multi-head threads formed by the definition of the indexing of the starting angle does not exceed 65535 heads.

4, the unit of Q is 0.001 degrees, if the signal with the spindle is offset by 180 degrees, the program needs to enter Q180000, if the input is Q180 or Q180.0 , all are considered to be 0.18 degrees.

Judgment method of major axis and short axis: Figure 3-38.

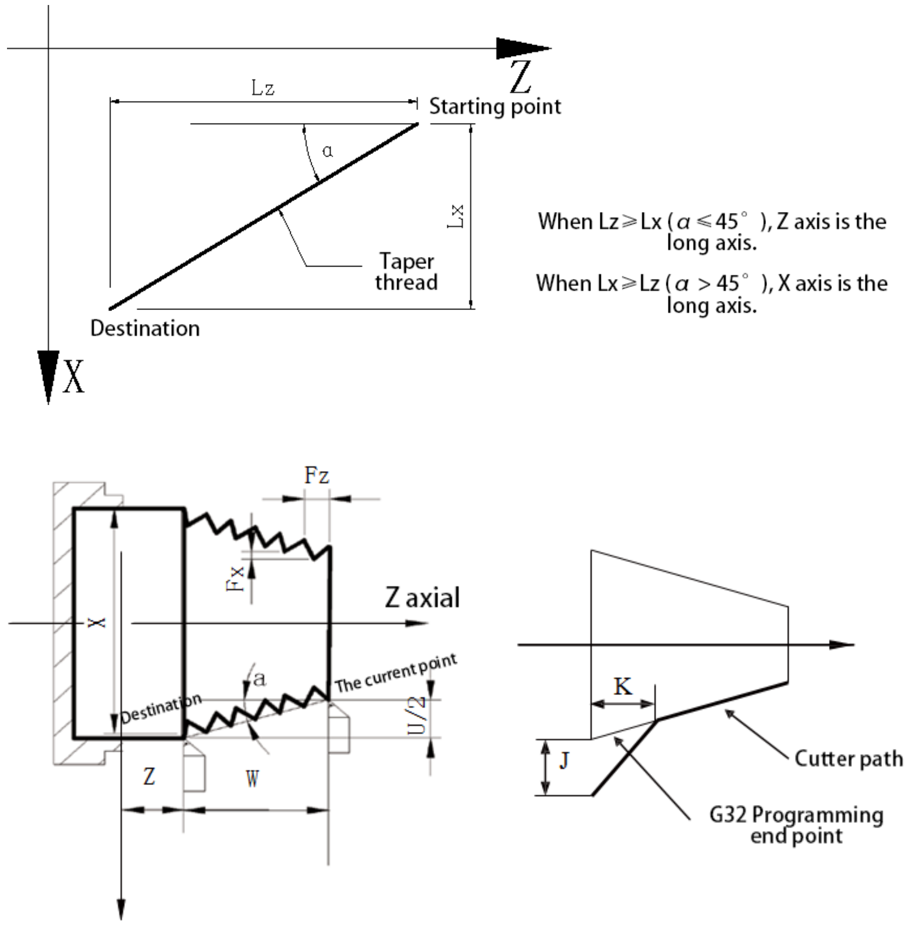


Figure 3-38 G32 trajectory chart

Notes:

● When J or J, K is omitted, there is no backlash; When K is omitted, press K=J to back the tail;

● When J=0 or J=0, K=0, there is no ejection;

● J ≠ 0, when K=0, press J=K to back the tail;

● J=0, K ≠ 0, no egress;

● The current program segment is thread cutting, the next program segment is also thread cutting, at the beginning of the next program segment cutting does not detect a turn signal of the spindle position encoder, directly start the threading, this function can achieve continuous threading.

● After performing the feed hold operation, the system displays "pause", thread cutting does not stop, and the movement is stopped until the current program segment is executed;

In the case of continuous threading, the thread cutting program segment is performed before the movement is stopped, and the program operation is suspended.

● In a single section of operation, the current program segment stops moving, and if it is continuous threading, the thread cutting program segment is completed before the movement is stopped.

● When the system is reset, emergency stop or drive alarm, the thread cutting speed stops.

Example: Thread pitch: 2mm. δ1 = 3mm, δ2 = 2mm, total depth of cut 2mm, cut in two passes.

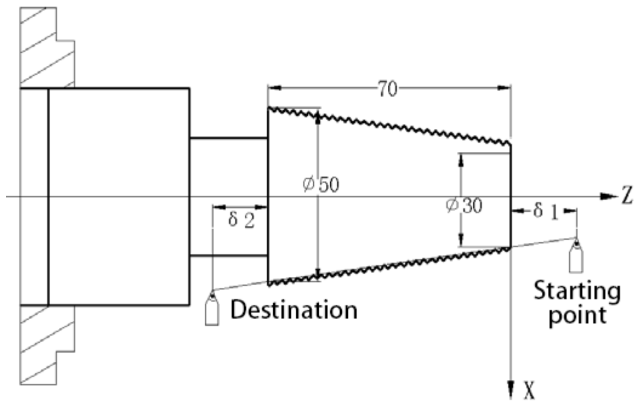


Figure 3-39

Procedure:
O0009;
G00 X28 Z3; (First cut into 1mm).
G32 X51 W-75 F2.0; (First cutting of taper thread).
G00 X55; (Tool exit).
W75; (Z axis back to start point).
X27; (Second re-entry knife 0.5mm).
G32 X50 W-75 F2.0; (Second cut of taper thread).
G00 X55; (Tool exit).
W75; (Z axis back to start point).
M30;

3.14.2 Variable pitch thread cutting code G34

Code format: G34 X/U_ Z/W_ F(I)_ J_ K_ R_ ;

Code function: The motion trajectory of the tool is a straight line from the starting point of the X and Z axes to the end point specified by the program segment. Displacement from start to finish point (X axis by radius value) The larger coordinate axis is called the long axis, the other coordinate axis is called the short axis, the spindle moves a lead per turn of the main axis during the movement, and the pitch of the spindle moves every turn is to increase the specified value or reduce the specified value, forming a spiral groove on the surface of the workpiece to achieve the machining of the variable pitch thread. When cutting, the amount of retract can be set. F, I code words are used to specify the pitch of the thread, and the G34 code can be executed to process straight threads, taper threads and end face threads with metric or imperial variable pitch.

Code description: G34 is a modal G code;

The meaning of X/U, Z/W, J, K is consistent with G32;

F: Specify the lead, the value range is 0 ~ 500mm;

I: Specify the number of teeth per inch thread, the value range is 0.06 ~ 25400 teeth / inch;

R: the increment or decrement value of the pitch per spindle, R=F1-F2, R with direction;

When F1 > F2 , the pitch decreases when R is negative;

When F1 < F2, the pitch increases when R is positive (Figure 3-40);

Range of R values: ±0.001 ~ ±500.000 mm / pitch per pitch (metric thread); ±0.060 ~ ±25400 teeth per inch (inch thread). An alarm is generated when the R value exceeds the above range value and the pitch exceeds the permissible value or the pitch is negative due to the increase/decrease of R.

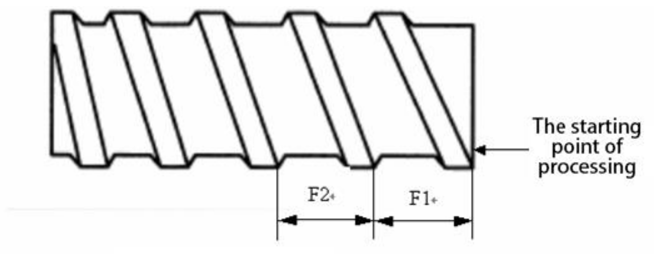


Figure 3-40

Notes:

● The precautions are the same as G32 thread cutting.

Example: The first pitch of the starting point is 4mm, and the increment value of the spindle pitch per turn is 0.2.

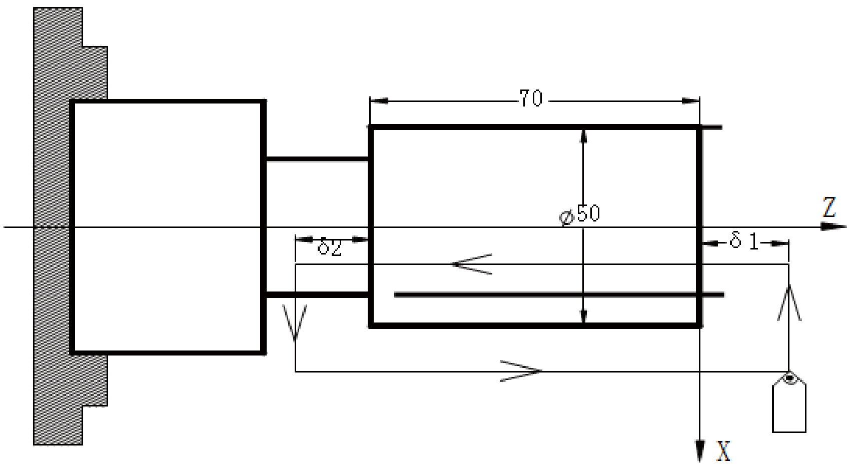


Figure 3-42 Machining variable pitch threads

Procedure:

O0010;

G00 X60 Z1 M03 S500;

G00 X48；
G34 W-78 F3.8 J5 K2 R0.2;
N30 M30；

3.14.3 Z-axis flexible tapping cycle G33

Code format: G33 Z/W_ F(I)_ L_;

Code function: The motion trajectory of the tool is from the beginning to the end, and then from the end point back to the starting point. During the movement, the spindle moves a pitch of one pitch per turn of the Z axis, which is always consistent with the pitch of the tap, forming a spiral groove in the inner hole of the workpiece, which can complete the threading of the inner hole at one time.

Code description: G33 is the modal G code;

Z/W: When you do not enter Z or W , the Z coordinate values of the start and end points are the same, and no thread cutting is performed;

F: Thread lead, the value range is shown in Table 1-2;

I: The number of teeth per inch thread, the value range table see 1-2;

Right-handedness is positive and left-handedness is negative;

L_ the number of processing heads, the range is 1-99, and the default is 1 head when L is omitted;

Cyclic process:

① Z-axis attack (no need to specify spindle opening before G33 code);
② After reaching the end point of the Z-axis coordinate specified by the program, the M05 signal is output;
③ Detection after the spindle has stopped completely;
④ Clockwise rotation signal output (opposite to the direction in which the original spindle rotates);
⑤ Z-axis retract to the starting point;
⑥ M05 signal output, spindle stop;
⑦ If it is a multi-head thread, repeat steps (1) to (6).

Description: This format will be a one-time punch, no layered drilling.

Note 1: Alarm when editing Z(W): Tapping cycle does not specify the bottom plane of the hole;

Note 2: The spindle and the Z axis are linked to tapping, and the spindle feedback speed is read to control the Z axis speed; Before the G33 instruction is executed, the internal execution of the M05 spindle stop is performed, the output spindle direction is judged according to the positive or negative of F/I, the output M 05 stop spindle is completed after the tapping is completed, and the Z-axis inlet is similar to the car thread waiting for the encoder to turn the signal before entering the tool.

Example of procedure: Fig . 3-43, thread M10×1.5

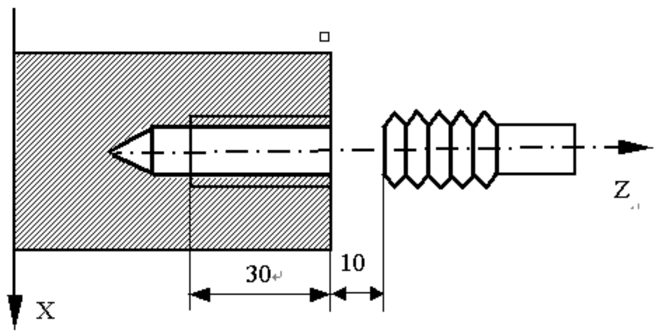


Figure 3-43

Procedure: O0011;

G00 Z90 X0; (Positioning)

G33 Z50 F1.5; (Tapping cycle, if the current F value is positive for the right-handed tapping, F-value is negative for the left-handed tapping).)

M03; (Start the spindle again).

G00 X60 Z100; (Continue processing).

M30; (End of procedure).

Note 1: The spindle rotation direction should be determined according to the rotation direction of the tap before tapping, and the spindle will stop rotating after the tapping is completed, and the spindle needs to be restarted if the processing needs to continue.

Note 2: This code is flexible tapping, after the spindle stop signal is effective, the spindle will also have a certain deceleration time to stop rotating, at this time the Z axis will still follow the rotation of the spindle and feed, until the spindle is completely stopped, so the actual processing of the thread's bottom hole position should be slightly deeper than the actual required position, the specific length exceeded according to the tapping time spindle speed and spindle brake device and determined.

Note 3: The movement speed of the **Z** axis **during tapping cutting is determined by the spindle speed and pitch, independent of the cutting feed rate multiplier.**

Note 4: Running in a single program segment or performing a feed-hold operation, the system displays "Pause", the tapping cycle does not stop, and the movement does not stop until the tapping is complete and the starting point is returned.

Note 5: When the system is reset, stops, or drives an alarm, the tapping cut speed stops.

3.14.3.1 Rigid/flexible tapping cycle G84 /G88

Code format: G84 Z/W_ F(I) ; (End tapping).

Code function: In the rigid mode, the control of the spindle motor is a servo motor that can achieve high-speed and high-precision tapping. It can be guaranteed that the starting position of the tapping is consistent without changing the starting point. That is, the tapping instruction is repeated multiple times in one position, and the thread wire will not be randomly buckled and rotten.

Code description: G84 is a modal G code;

Z/W: When you do not enter Z or W , the Z coordinate values of the start and end points are the same, and no thread cutting is performed;

F: Thread lead, the value range is shown in Table 1-2

I: The number of teeth per inch thread, the value range table see 1-2;

When tapping is being performed, the feed speed multiplier and spindle magnification are considered to be 100%.

Rigid mode:

You can specify a rigid mode in any of the following ways:

(1) Specify M29 S***** before the tapping instruction section

(2) Specify M29 S***** in the program segment containing tapping instructions

When specified in the same program segment as the G84 instruction and the M instruction, the M code is executed at the same time as the first hole positioning action, and then the system processes the next drilling action.

Shaft switching: The fixed loop must be canceled before switching the tapping shaft. If the tapping shaft is changed in the rigid mode, the alarm is raised.

If the S and Axis Movement instructions are specified between the M29 and G84, the system alarms. If M29 is specified in a tapping cycle, the system alarms.

In the per-minute feed method, the thread lead is expressed in terms of feed speed / spindle speed.

Z-axis feed speed = spindle speed * Thread lead.

Procedure:

G80;

M29 S500; (Turn on tapping mode).

G84 Z50 F1.5 ; (tapping cycle).

G80;

M30;

Code format: G88 X/U_ F(I)_; (side tapping cycle).

Code function: In the rigid mode, the control of the spindle motor is a servo motor that can achieve high-speed and high-precision tapping. It can be guaranteed that the starting position of the tapping is consistent without changing the starting point. That is, the tapping instruction is repeated multiple times in one position, and the thread wire will not be randomly buckled and rotten.

Code description: G88 is a modal G code;

X/U: When no X or U is entered, the X coordinate values of the start and end points are the same, and no thread cutting is performed;

F: Thread lead, the value range is shown in Table 1-2;

I: The number of teeth per inch thread, the value range table see 1-2;

Procedure:
G80;
M29 S500; (Turn on tapping mode).
G88 X50 F1.5 ; (tapping cycle)
G80;
M30;

Limit:

F: If the specified F value exceeds the upper limit of the cutting feed speed, an alarm is issued.

S: If the speed is higher than the maximum speed of the specified grade, it is alarmed.

Cancel: You cannot specify 01 set of G codes (G00 to G03) in the same program segment.

Tool offset: Tool radius bias is ignored during fixed cyclic positioning.

Program restart: The program restart is invalid during rigid tapping.

3.14.4 Thread cutting cycle G92

Code format: G92 X/U_ Z/W_ F_ J_ K_ L; (Metric straight thread cutting cycle).
G92 X/U_ Z/W_ I_ J_ K_ L ; (Inch straight thread cutting cycle).
G92 X/U_ Z/W_ R_ F_ J_ K_ L ; (Metric taper thread cutting cycle).
G92 X/U_ Z/W_ R_ I_ J_ K_ L ; (Inch taper thread cutting cycle).

Code function: From the starting point of cutting, radial (X-axis) tooling, axial (Z-axis or X, Z-axis simultaneous) cutting, to achieve equal pitch of straight thread , taper thread cutting cycle. Perform the G92 code, where there is a thread ejection process at the unended threading: at a fixed length from the end of the thread cutting (called the ejection length of the thread), while the Z axis continues to interpolate the thread, X The axis accelerates out exponentially or linearly (set by the parameter 175#5) in the retract direction, and after the Z axis reaches the end of the cutting, the X axis retracts at a fast moving speed, as shown in the figure 3-44 shown.

Code description: G92 is a modal G code;

Cutting start point: the starting position of thread interpolation;

Cutting end point: the end position of thread interpolation;

X: Absolute coordinate of X axis of the cutting end point;

U: The difference between the absolute coordinates of the cutting end point and the X axis of the starting point;

Z: The absolute coordinate of the Z axis of the cutting end point;

W: The difference between the absolute coordinates of the Z axis of the cutting end point and the starting point Z axis;

R: The difference between the absolute coordinates of the X axis of the cut start and the cut end point (radius value), when R does not match the sign of U Requires $|R| \le |U/2|$;

F: Thread lead, value range0 $< F \le 500$ mm，F Specify that the value is persisted after execution, and the input can be omitted;

I: The number of thread teeth per inch, the value range is 0.06 ~ 25400 teeth / inch, I specified value is maintained after execution, the input can be omitted;

J: The amount of movement in the short axis direction when the thread is tailed, the value range is 0 ~ 99999.999 (unit: mm), without direction (the tailing direction is automatically determined according to the starting point position of the program). , modal parameter, if the minor axis is the X axis, the value is specified as the radius;

K: The length in the direction of the long axis when the thread is tailed, the value range is 0 ~ 99999.999 (unit: mm). Without direction, modal parameters, such as the long axis is the X axis, this value is specified for the radius;

L: The number of heads of a multi-headed thread, the range of this value is: 1 ~ 99, modal parameters. (Defaults to single-head thread when L is omitted).

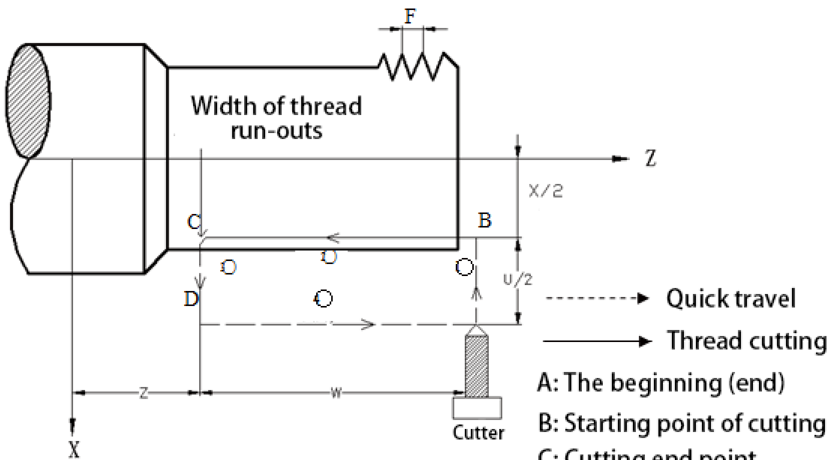


Figure 3-44

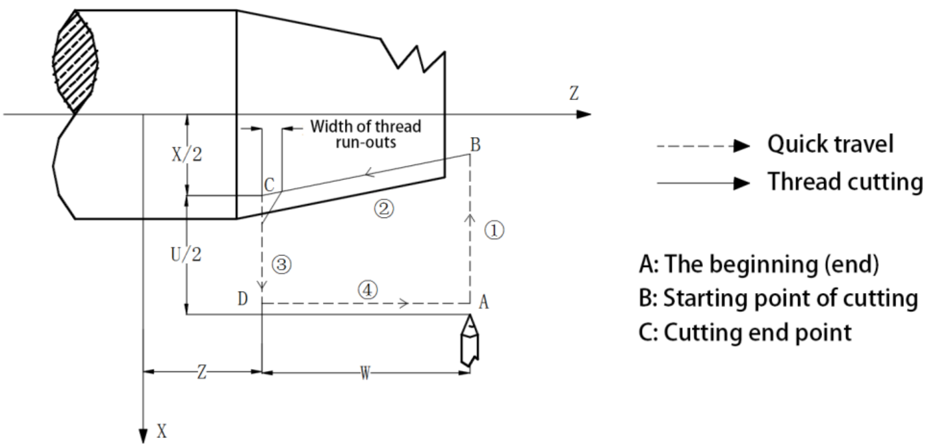


Figure 3-45

The G92 code can complete the machining of one thread in multiple passes, but it cannot achieve the machining of 2 consecutive threads, nor can it machine the end face thread. The definition of the G92 code thread pitch is the same as that of the G32, which refers to the displacement of the spindle by one turn of the long axis (the displacement of the X axis is based on the radius value).

The pitch of the taper thread refers to the displacement of the spindle by a long axis (X axis displacement by radius value), and the absolute value of the difference between the Z axis coordinates of point B and point C is greater than that of the X axis (Radius value) When the absolute value of the coordinate difference, the Z axis is the long axis; Conversely, the X axis is the long axis.

Cycle process: straight thread as shown in Fig. 3-44, taper thread as shown in Fig. 3-45.

(1) The X axis moves quickly from the starting point to the cutting starting point;

(2) Thread insertion from the cutting starting point to the cutting end point;

(3) The X axis retracts the knife at a fast moving speed (opposite to the (1) direction) and returns to the same absolute coordinate of the X axis as the starting point;

(4) The Z axis moves quickly back to the starting point, and the loop ends.

Notes:

● When J and K are omitted, the parameter setting value of NO.19 is faded;

● When J is omitted, the major axis direction is tailed by K, and the minor axis direction is tailed by the parameter setting value of NO.19;

● When K is omitted, press J=K to back the tail;

● When J=0 or J=0, K=0, there is no ejection;

● J ≠ 0, when K=0, press K=J to back the tail;

● J=0, K ≠ 0, no egress;

● After the feed hold operation is performed during the thread cutting process, the system still performs thread cutting, and after the thread cutting is completed, the "pause" is displayed and the program operation is suspended;

● After performing a single-program operation during the thread cutting process, the operation stops after returning to the starting point (a thread cutting cycle operation is completed);

● When J and K enter negative values, they are treated as positive values;

● When the system is reset, emergency stop or drive alarm, the thread cutting speed stops.

Code trajectory: U, W, R reflects the relative position of the thread cutting end point and the starting point, and the tool trajectory and the ejection direction are shown in the same symbol as shown in figure:

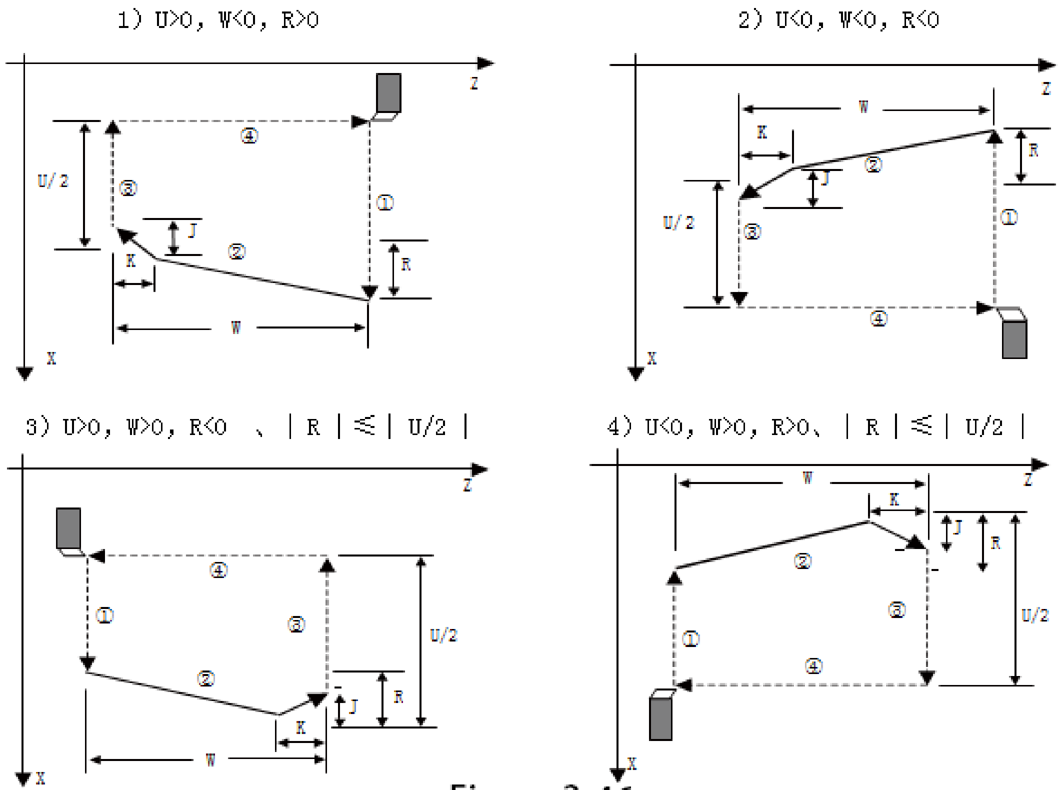


Figure 3-46

Example: Figure 3-47

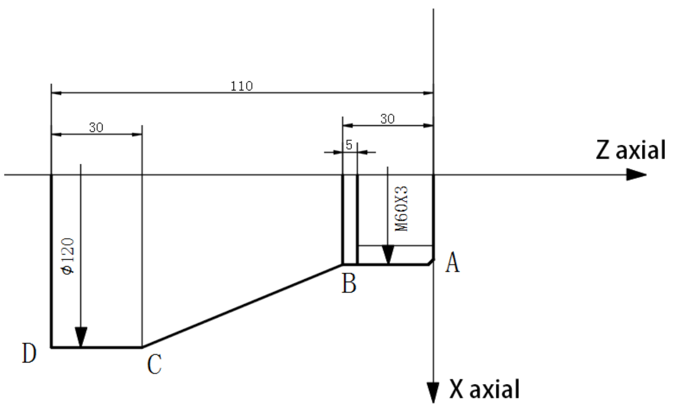


Figure 3-47

Procedure:

O0012 ；

M3 S300 G0 X150 Z50 T0101； (Thread knife).

G0 X65 Z5； (Quick Positioning).

G92 X58.7 Z-28 F3 J3 K1； (Machining threads, cut in 4 knives, the first time into the knife 1.3mm).

X57.7； (Second knife feed 1mm).

X57； (Third knife feed 0.7mm).

X56.9； (Fourth knife feed 0.1mm).

M30；

3.14.5 Multi-thread cutting cycle G76

Code format: G76 P(m)(r)(a) Q(△dmin) R(d)；

G76 X/U_ Z/W_ R(i) P(k) Q(△d) F(I) _ ；

Code function: through multiple thread coarse turn, thread fine car to complete the specified tooth height (total depth of cut) of the thread processing, if the defined thread angle is not 0 °, the entry point of the thread thick car from the top of the thread tooth gradually moved to the bottom of the thread tooth, so that the angle between the adjacent two threads is the specified thread angle. The G76 code can machine straight threads and taper threads with thread ejectors, which can achieve one-sided cutting of the cutting edge thread, and the amount of knife eaten is gradually reduced, which is conducive to protecting the tool and improving the thread accuracy. G76 codes cannot machine face threads. The processing trajectory is shown in Figure 3-48(a).

Related Definitions:

Start point (end point): the position before and after the operation of the program segment, expressed as point A;

Thread End: The end point of a thread cut defined by X/U Z/W, expressed as point D. If there is a thread ejection, the cutting end point is the long axis direction of the thread cutting end, and the position after the short axis direction is rewinding.

Thread start point: The difference between the Z axis absolute coordinate and the A point A and the X axis absolute coordinate and the X axis absolute coordinate of the D point is i (Thread taper, radius value), represented as a C-point. If the defined thread angle is not 0°, the C point cannot be reached when cutting;

Thread cut depth reference point : The difference between the absolute coordinate of the Z axis and the absolute coordinate of the A axis is k (the total cutting depth and radius value of the thread), represented as a B-point. The thread cut depth of point B is 0, which is the reference point for the system to calculate the depth of each thread cutting;

Thread cutting depth: The cutting depth of each thread cutting cycle. The inverse extension of each thread cutting trajectory intersects the line BC, and the difference between the point and the absolute coordinates of the X axis of point B (unsigned, radius value) is the thread cut depth. The thread cut depth of each rough car is $\sqrt{n} \times \Delta d$, n is the current number of rough car cycles, and △d is the thread cut depth of the first rough car;

Thread cutting amount: the difference between the thread cutting depth and the previous thread cutting depth: $(\sqrt{n} - \sqrt{n-1}) \times \Delta d$;

Retract end point: After the end of thread cutting in each thread rough turning cycle and fine turning cycle, the end position of radial (X axis) retracted tool is expressed as E point;

Thread entry point: The point at which thread cutting actually begins in each thread rough turning cycle and fine turning cycle is expressed as bn point (n is the number of cutting cycles), B1 is the first thread rough car entry point, and Bf is the last thread rough car entry point. Be is the threaded entry point. The displacement of the Bn point relative to the X and Z axes of point B conforms to the formula:

$$\tan\frac{a}{2}=\frac{|Z-axis\ displacement|}{|X-axis\ displacement|}\quad ;\qquad a\text{：Angle of thread}$$

X: Absolute coordinate of the X axis of the thread end point;

U: The difference between the absolute coordinate of the end point of the thread and the X axis of the starting point;

Z: The absolute coordinate value of the Z axis of the thread end point;

W: The difference between the absolute coordinates of the Z axis of the thread end point and the starting point;

P(m):Thread finishing number 00 ~ 99 (unit: times), m specifies that the value remains valid after execution, and puts the system data parameter NO.057 Modify the value of m. When m is not entered, the value of the system data parameter NO.057 is used as the number of finishing times. In thread finishing, the amount of cutting per feed is equal to the amount of cutting of the thread turn.

P(r):Thread ejection length 00 ~ 99 (unit: 0.1×L, L is the thread pitch) , r Specifies that the value remains valid after execution, and modifies the value of the system data parameter NO.019 to r. When r is not entered, the value of the system data parameter NO.019 is used as the thread ejection width. Thread ejection function can realize threading without retract slot, and the thread egress width defined by system parameter NO.019 is valid for G92 and G76 codes;

P(a):The angle between the two adjacent threads, the value range is 00~ 99, the unit: degree (°),, a the specified value remains valid after execution, And modify the value of the system data parameter NO.058 to a. When a is not entered, the value of the system data parameter NO.058 is used as the angle of the thread tooth. The angle of the actual thread is determined by the tool angle, so a should be the same as the tool angle;

Q(△dmin): The minimum amount of cutting when the thread is rough, and the value range is00 ～ 99999(Unit:0.001mm , the radius value)。 while($(\sqrt{n}-\sqrt{n-1})\times\Delta d$ ＜ △dmin time 以△dmin As the cutting volume of this rough turn, Namely: This time the thread cut depth is （$\sqrt{n-1}\times\Delta d+\Delta dmin$ ）。 Set up△dmin It is to avoid the amount of rough turning cuts being too small and the number of rough turns too many due to the decreasing amount of thread rough turning.Q(△dmin) After execution, specify a value△dmin Keep the parameters valid and put the system data NO.059 Modify the value of△dmin。 Not entered Q(△dmin) when the system data parameter is used NO.059 value as the minimum amount of cutting;

R(d): The amount of cutting of the thread fine turning, the value range is 00 ~ 99.999, (unit: mm/inch, unsigned, radius value). , the radius value is equal to the difference between the X-axis absolute coordinate of the threaded car entry point Be and the last thread rough entry point Bf. After R(d) is executed, the specified value d remains valid and the value of the system data parameter NO.060 is modified to d×1000 (IS_B)/d× 1000(IS_C)。 When R(d) is not entered, the value of the system data parameter NO.060 is used as the thread fine turning cutting amount;

R(i):Thread taper, the difference between the absolute coordinate of the X axis of the thread start point and the thread end point, the value range is -99999.999 ~ 99999.999 (unit: mm/ inch, radius value). When R(i) is not entered, the system is treated as R(i)=0 (straight thread);

P(k):Thread tooth height, total thread cutting depth, value range from 1 to 99999999× minimum input increment 0.001 (radius value , unsigned). When P(k) is not entered, the system alarms;

Q(△d):First thread cutting depth, value range from 1 ~ 99999999× minimum input increment 0.001 (Radius value, unsigned). When △ d is not entered, the system alarms;

F: Thread lead, The value range is0 ＜ F ≤ 500 mm;

I: The number of threaded teeth per inch, the value range is 0.06 ~ 25400 teeth / inch;

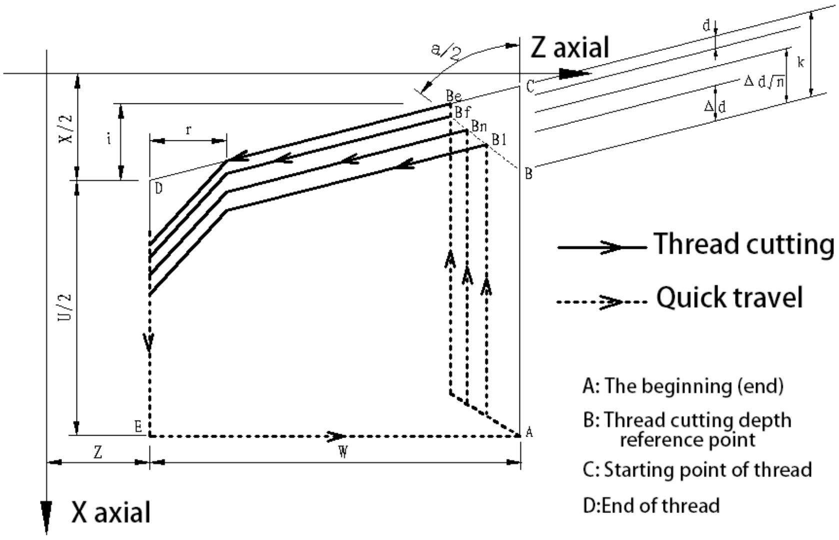


Figure 3-48(a).

Details of the entry method are shown in Figures 3-48 (b):

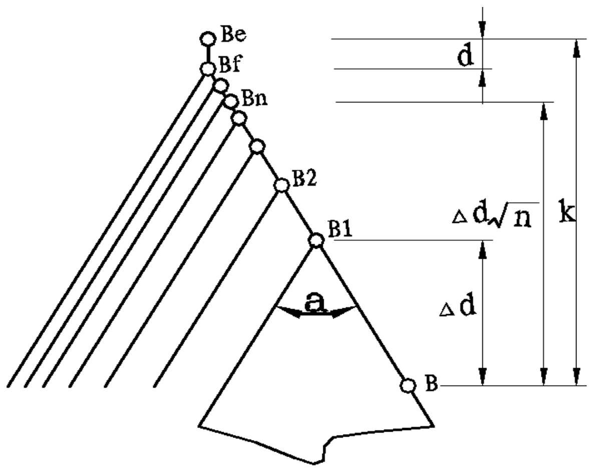


Figure 3-48 (b).

Thread pitch refers to the displacement of the spindle turning a long axis (X axis displacement by radius value), and the absolute value of the difference between the Z axis coordinates of the C point and the D point is greater than the absolute value of the X axis coordinate difference (Radius value, equal to the absolute value of i), the Z axis is the long axis; Conversely, the X axis is the long axis.

Code Execution Procedure:

(1) Quickly move from the starting point to B1 with a thread cut depth of △d. If a = 0, only move the X axis; If a ≠ 0, the X and Z axes move at the same time, moving in the same direction as A → D;

(2) Thread cut in a direction parallel to C→ D to intersect D → E (r ≠ 0 with egress process) ；

(3) The X axis moves quickly to point E;

(4) The Z axis is quickly moved to point A, and a single rough car cycle is completed;

(5) Move the knife into Bn again quickly (n is the number of rough turns), and the depth of cut takes $\sqrt{n} \times \Delta d$the larger of the larger of (), (), if the depth of cut is less than (k-d).$\sqrt{n-1} \times \Delta d + \Delta dmin$, turn (2) execution; If the depth of cut is greater than or equal to (k-d), press the depth of cut (k-d) into the Bf point and turn (6) to perform the last threaded rough turn;

(6) Thread cut in a direction parallel to C → D to intersect D → E (r ≠ 0 with egressing process) ；

(7) The X axis moves quickly to point E;

(8) The Z axis is quickly moved to point A, and the thread coarse car cycle is completed, and the thread fine car is started;

(9) After quickly moving to the Be point (thread cutting depth is k, cutting amount is d), thread finishing is performed, and finally return to point A to complete a thread finishing cycle;

(10) If the number of fine turning cycles is less than m, turn (9) to carry out the next fine turning cycle, the thread cutting depth is still k, and the cutting amount is 0; If the number of finishing cycles is equal to m, the G76 composite threading cycle ends.

Notes:

●After the feed hold operation is performed during the thread cutting process, the system still performs thread cutting, and after the thread cutting is completed, the "pause" is displayed, and the program operation is suspended;

● Perform a single-program operation during the thread cutting process, and stop after returning to the starting point (a thread cutting cycle operation is completed);

● When the system is reset, emergency stop or drive alarm, the thread cutting speed stops;

● G76 P(m)(r)(a) Q(△dmin) R(d) can omit all or omit part of the code address, Omitted addresses run according to the parameter setpoint;

● m, r, a with the same code address P entered once, m, r, a are all omitted, press the parameter NO.57, 19, 58 setpoint operation; Address P takes a value when 1 or 2 digits are entered; Address P takes r and a when entering 3 or 4 digits;

● The symbols of U and W determine the direction of A → C → D → E, R(i) The sign of C determines the direction of C → D. There are four combinations of U and W symbols, corresponding to four processing trajectories, see Figure 3-46.

Example: Fig. 3-49 with threads M68×6.

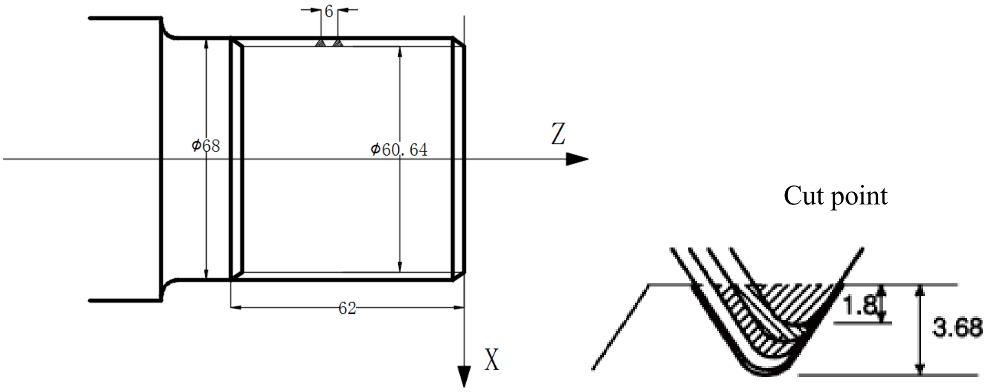


Figure 3-49

Procedure: O0013;

G50 X100 Z50 M3 S300； (Set the workpiece coordinate system to start the spindle and specify the speed).

G00 X80 Z10； (Quickly move to the starting point of processing).

G76 P020560 Q150 R0.1； (Number of finishing repetitions 2, chamfer width 0.5mm, tool angle 60°, minimum depth of entry 0.15, finishing allowance.) 0.1)

G76 X60.64 Z-62 P3680 Q1800 F6； (Thread tooth height 3.68, first thread cutting depth 1.8).

G00 X100 Z50； (Return to the program starting point).

M30； (End of procedure).

3.15 Constant linear speed control G96, constant speed control G97

Detailed instructions are provided in Section 2.2.3 of this Article.

3.16 G98 per minute and G99 per transfer

Code format: G98 F__; (Leading zero can be omitted, given feed rate per minute).

Code function: Given the cutting feed rate in mm/min, G98 is the modal G code, if it is currently in the G98 mode, G98 can be entered without entering it.

Code format: G99 F__;

Code function: given the cutting feed rate in millimeters per revolution, G99 is the modal G code. If you are currently in the G99 mode, you can enter G99 without entering it. When the CNC executes the G99 F__, it combines the F-code value (mm/r) with the current spindle speed (r/min) product as the code feed rate to control the actual cutting feed rate, when the spindle speed changes, the actual cutting feed rate changes. Using the G99 F__ the cutting feed per revolution of a given spindle, a uniform cutting pattern can be formed on the surface of the workpiece. For machining in the G99 mode, the machine must be fitted with a spindle encoder.

G98 and G99 are modal G codes of the same group, and only one valid one can be valid. G98 is the initial G code, and the default G98 is valid when the CNC is powered on. Conversion formula for feed per turn to feed per minute:

Fm = Fr×S

where: Fm: feed per minute (mm/min);

Fr: Feed per revolution (mm/r);

S: Spindle speed (r/min).

When the CNC is powered on, the feed speed is the value set by the system data parameter NO.076. After performing F0, the feed speed is 0. When the CNC resets and stops, the F-value remains unchanged.

Note: In the G99 mode, when the spindle speed is less than 1r/min, the cutting feed speed will be uneven; When the spindle speed fluctuates, there is a following error in the actual cutting feed speed. In order to ensure the quality of processing, it is recommended that the spindle speed selected during machining should not be lower than the minimum speed of the spindle servo or inverter output effective torque.

Related parameters:

CNC parameter N0.027: upper limit of cutting feed rate;

CNC parameter N0.029: Exponential acceleration and deceleration time constant during cutting feed and manual feed;

CNC parameter N0.030: Start (stop) speed at cutting feed.

3.17 Macro Code

The lathe system provides macro code similar to the high-level language, and the user macro code can realize variable assignment, arithmetic operation, logical judgment and condition transfer, which is conducive to the preparation of processing programs for special parts, reducing the cumbersome numerical calculations performed during manual programming, and simplifying the user program.

3.17.1 Macro Variables

● Representation of variables

Variables are specified with the symbol "#"+ variable number;

Format: # i(i=100,102,103,......);

Examples: #105, #109, #125.

● The type of the variable

Variables can be divided into four types based on the variable number.

The variable number	The type of variable	function
#0	Empty variable	The variable is always empty and no value can be assigned to the variable
#1 ～ #50	Local variables	Local variables can only be used to store data in macro programs, for example, the result of an operation. When powered down, local variables are initialized to null. When a macro program is called, an argument assigns a value to a local variable.
#100 ～ #199 #500 ～ #999	Public variables	Public variables have the same meaning in different macro programs. When powered off, the variable #100 ~ #199 is initialized to empty, and the value of the variable #500 ~ #999 is saved, even if the power is lost.
#1000 ～ #5235	System variables	System variables

● A reference to a variable

Replace the value after the address with a variable.

Format: < address > +“#I"or < address > +“－ #I", indicating that the variable is "#I"Value or put variable"#IThe negative value of the value of "" is used as the address value.

Example: F#103... When #103=15, it functions the same as the F15 code;

Z-#110... When #110=250, it functions the same as Z-250 code;

Note 1: Addresses O and N cannot reference variables. As O#100, N#120 is an illegal reference;

Note 2: If the maximum code value specified by the address is exceeded, it cannot be used; Example: When **#150 = 120, M#150 exceeds the maximum code value.**

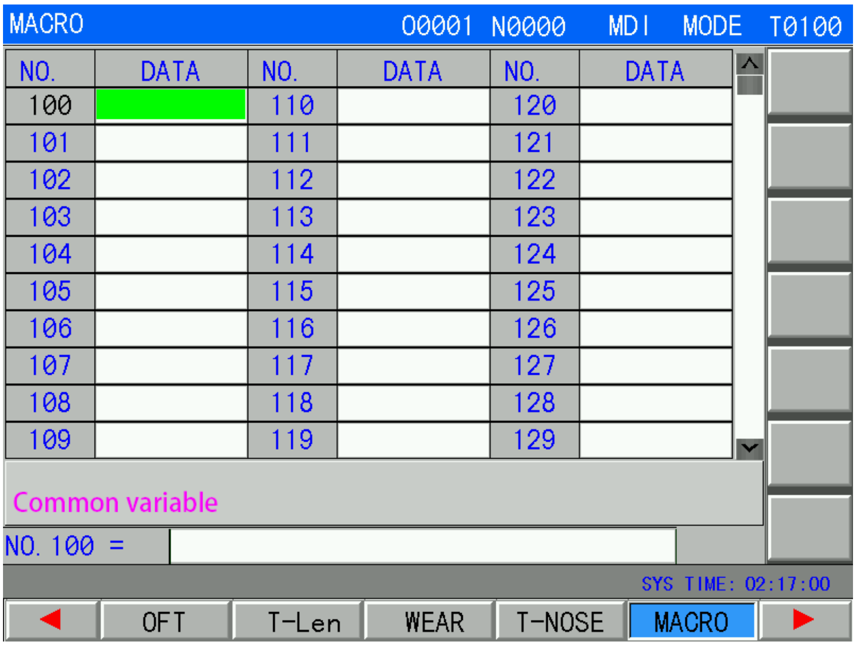
● Empty variable

When the value of a variable is undefined, the variable is empty, and variable #0 is always an empty variable, which cannot be written, only read.

When referencing an undefined variable (an empty variable), the address itself is also ignored.

When #1 = < empty >	When #1 = 0
The G00 X100 Z#1 is equivalent to the G00 X100	The G00 X100 Z#1 is equivalent to the G00 X100 Z0

● Variable display (different system models have different display interfaces).



（1） On the Macro Variables page, when a variable appears blank, it means that the variable is empty, that is, it is not defined.

（2） The values of public variables (#100 ~ #199, #500 ~ #999) are displayed on the Macro Variables page, or you can directly enter data to assign values to public variables under the Change page.

（3） The values of local variables (#1 to #50) and system variables are not displayed, and if you need to view the values of a local variable or system variable, you can display it by assigning it to a public variable.

● System variables - are as follows:

1） Interface input signals #1000 - #1047 (bitwise read PMC input signal).

2） Interface output signals #1100 - #1047 (bitwise writes out signals to pmC).

3） X-axis length compensation values #1500 - #1531 (radius value, readable and writable).

4） Z-axis length compensation values #1600 - #1631 (read-write).

5） Y-axis length compensation value #1700 - #1731 (read-write).

6） Tip radius compensation value #1800 - #1831 (readable and writable).

7） X-axis wear compensation values #1900 - #1931 (radius value, readable and writable).

8） Z-axis wear compensation values #2000 - #2031 (read-write).

9） Y-axis wear compensation values #2100 - #2131 (read-write).

10） Tip radius wear compensation value #2200 - #2231 (read-write).

11） Alarm #3000

12） User Datasheet #3500 - #3755 (read-only, not write)

13） Modal Information #4000 - #4030 (read-only, not write)

14） Location #5001 - #5030 (read-only, not write)

System variables are detailed

(1) Interface signal: CNC only operates on G and F signals, as to whether there is a corresponding I/O number corresponding to it depends on the specific PLC definition.

The variable number	function
#1000 ～#1015	Corresponds to the signal status of the system G54.0 ~ G54.7, G55.0 ~ G55.7
#1032	Corresponds to the signal state of the system G54 and G55 bytes
#1100 ～#1115	Corresponds to the signal status of the system F54.0 ~ F54.7, F55.0 ~ F55.7
#1132	Corresponds to the signal state of the system F54 and F55 bytes
#1133	Corresponds to the signal state of the system F56, F57, F58, F59 four bytes

(2) Tool compensation system variables:

Compensation number	Bias compensation value					Wear compensation value			
	X-axis	Z-axis	Y-axis	radius		X-axis	Z-axis	Y-axis	radius
1	#1500	#1600	#1700	#1800		#1900	#2000	#2100	#2200
…	…	…	…	…		…	…	…	…
…	…	…	…	…		…	…	…	…
32	#1531	#1631	#1731	#1831		#1931	#2031	#2131	#2231

(3) System modal information variables

The variable number	function	
#4001	G00、G01、G02、G03、G32、G33、G34、G80 、G84、G88、G90、G92、G94	Group 1
#4002	G96、G97	Group 2
#4003	G98、G99	Group 3
#4005	G54、G55、G56、G57、G58、G59	Group 5
#4006	G20、G21	Group 6
#4007	G40、G41、G42	Group 7
#4016	G17、G18、G19	Group 16
#4020	F code	
#4021	M code	
#4022	The sequence number	
#4023	The program number	
#4024	S code	
#4025	T code	

(4) System variables of coordinate location information:

The variable number	Location signal	coordinate system	Tool compensation value	Read operation during exercise
#5000 ～ #5005	End of the program segment	Workpiece coordinate system	Not included	OK
#5006 ～ #5010	Current position (machine coordinates).	Machine coordinate system	contain	No
#5011 ～ #5015	The current position (absolute coordinates).	Workpiece coordinate system	contain	No

Note: The position information listed in the table above corresponds sequentially to the X axis, Z axis, Y axis, 4th axis, and 5th axis, respectively, for example: #5001 represents X Position information for the axis, #5002 for the Z axis, #5003 for the position information for the Y axis, and #5004 for the first Position information of the 4th axis, #5005 indicates the position information of the 5th axis;

(5) Workpiece zero offset and workpiece coordinate system:
Base offset: #5201 ~ #5205
G54：#5206 ～ #5210
G55：#5211 ～ #5215
G56：#5216 ～ #5220
G57：#5221 ～ #5225
G58：#5226 ～ #5230
G59：#5231 ～ #5235

(6) Number of processed parts and number of required parts:

Number of machined parts and number of required machined parts	
The variable number	Function (can be read and written)
#3901	Number of machined parts (number of completes) [maximum value of 99999, automatic zeroing when the maximum value is exceeded].
#3902	The number of parts to be machined (target quantity) [maximum9999].

When the value of #3901 is changed, the number of machined parts displayed in the position interface also changes, and the number of machined parts recorded in the same number parameter 365 also changes; When the value of parameter 365 is changed, the number of machined parts displayed in the position interface also changes, as does the value of the system variable #3901.

When the value of #3902 is changed, the number of required pieces displayed in the position interface also changes, and the number of demand pieces in the same number parameter 366 changes accordingly; When the value of parameter 366 is changed, the number of required pieces displayed in the position interface also changes, as does the value of the system variable #3902.

● Local variables

Correspondence between addresses and local variables:

Argument address	Local variable number	Argument address	Local variable number	Argument address	Local variable number
A	#1	E	#8	U	#21
B	#2	F	#9	V	#22
C	#3	M	#13	W	#23
I	#4	Q	#17	X	#24
J	#5	R	#18	Y	#25
K	#6	S	#19	Z	#26
D	#7	T	#20		

3.17.2 Arithmetic commands and transfer commands G65

General code format: G65 H(m) P(#I) Q(#J) R(#K);

where m: indicates the operation command or transfer command function.

#I: The name of the variable that stores the result of the operation.

#J: The name of the variable on which the operation is performed is 1, which can be a constant.

#K: The name of the variable to which the operation is performed is 2, which can be a constant.

Code meaning: #I = #J □ #K

Operator sign, determined by H(m).

Example: P#100 Q#101 R#102... .. #100 = #101 □ #102；

P#100 Q#101 R15... . #100 = #101 □ 15；

P#100 Q-100 R#102... .. #100 = -100 □ #102；

Note: Variables cannot be taken with "#" when they are constants;

Macro operations (jump) table

Code format	function	definition
G65 H01 P#I Q#J	Assignment operations	#I =#J ； Assign the value of variable #j to variable #i
G65 H02 P#I Q#J R#K	Decimal addition	#I = #J + #K
G65 H03 P#I Q#J R#K	Decimal subtraction	#I = #J - #K
G65 H04 P#I Q#J R#K	Decimal multiplication	#I = #J × #K
G65 H05 P#I Q#J R#K	Decimal division operation	#I = #J ÷ #K
G65 H11 P#I Q#J R#K	Binary addition (or operation)	#I = #J OR #K
G65 H12 P#I Q#J R#K	Binary multiplication (and operation)	#I = #J AND #K
G65 H13 P#I Q#J R#K	Binary XOR	#I = #J XOR #K
G65 H21 P#I Q#J	Decimal squared	#I = $\sqrt{\#J}$
G65 H22 P#I Q#J	The decimal takes the absolute value	#I = $\|\#J\|$
G65 H23 P#I Q#J R#K	The decimal remainder	The ÷ remainder of #I = (#J #K).
G65 H24 P#I Q#J	Decimal becomes binary	#I = BIN (#J)
G65 H25 P#I Q#J	Binary becomes decimal	#I = BCD (#J)
G65 H26 P#I Q#J R#K	Decimal multiplication and division	#I = #I × #J ÷ #K
G65 H27 P#I Q#J R#K	Compound square root	#I = $\sqrt{\#J^2 + \#K^2}$
G65 H31 P#I Q#J R#K	sine	#I = #J × SIN(#K)
G65 H32 P#I Q#J R#K	cosine	#I = #J × COS(#K)
G65 H33 P#I Q#J R#K	Cut the strings	#I = #J × TAN(#K)
G65 H34 P#I Q#J R#K	Anyway cut	#I = #J × ATAN(#K)
G65 H80 Pn	Unconditional transfer	Jump to segment n
G65 H81 Pn Q#J R#K	Condition Transfer 1	If #J = #K , jump to segment n, otherwise executed sequentially
G65 H82 Pn Q#J R#K	Condition Transfer 2	If #J ≠ #K , jump to segment n, otherwise executed sequentially
G65 H83 Pn Q#J R#K	Condition Transfer 3	If #J > #K , jump to segment n, otherwise executed sequentially
G65 H84 Pn Q#J R#K	Condition transfer 4	If #J < #K , jump to segment n, otherwise executed sequentially
G65 H85 Pn Q#J R#K	Condition Transfer 5	If #J ≥ #K , jump to segment n, otherwise executed sequentially

G65 H86 Pn Q#J R#K	Condition Transfer 6	If #J ≤ #K , jump to segment n, otherwise executed sequentially
G65 H99 Pn	Generates a user alarm	Generates (3000+n) user alarms

1. **Operation command**

1) Assignment of variables: # I = # J
G65 H01 P#I Q#J
G65 H01 P# 101 Q1005 ； (#101 = 1005)
G65 H01 P#101 Q#110； (#101 = #110)
G65 H01 P#101 Q-#102； (#101 = -#102)

2) Decimal addition: # I = # J+# K
G65 H02 P#I Q#J R#K
G65 H02 P#101 Q#102 R15 ； (#101 = #102+15)

3) Decimal subtraction: # I = # J - # K
G65 H03 P#I Q#J R# K
G65 H03 P#101 Q#102 R#103 ； (#101 = #102 — #103)

4) Decimal multiplication: #I = #J× #K
G65 H04 P#I Q#J R#K
G65 H04 P#101 Q#102 R#103 ； (#101 = #102×#103)

5) Decimal division: #I = #J÷ #K
G65 H05 P#I Q#J R#K
(Example) G65 H05 P#101 Q#102 R#103 ； (#101 = #102÷#103)

6) Binary logic plus (or):#I = #J.OR.#K
G65 H11 P#I Q#J R#K
G65 H11 P#101 Q#102 R#103 ； (#101 = #102.OR. #103)

7) Binary logic multiplication (and):#I = #J.AND. # K
G65 H12 P#I Q#J R#K
G65 H12 P# 101 Q#102 R#103 ； (#101 = #102.AND.#103)

8) Binary XOR: # I = # J.XOR. # K
G65 H13 P#I Q#J R#K
(Example) G65 H13 P#101 Q#102 R#103 ； (#101 = #102.XOR. #103)

9) Decimal squared: #I = $\sqrt{\#j}$
G65 H21 P#I Q#J

(Example) G65 H21 P#101 Q#102 ; (#101 =$\sqrt{\#102}$)

10) Decimal takes the absolute value: #I = | #J |
G65 H22 P#I Q#J
(Example) G65 H22 P#101 Q#102 ; (#101 = | #102 |)

11) Decimal remainder: #I = # J - TRUNC(#J/#K) ×# K, TRUNC: Rounds the fractional part
G65 H23 P#I Q#J R#K
G65 H23 P#101 Q#102 R#103 ; (#101 = #102- TRUNC (#102/#103)×#103)

12) Decimal to binary: # I = BIN (# J).
G65 H24 P#I Q#J
G65 H24 P#101 Q#102 ; (#101 = BIN(#102))
13) Binary to decimal: # I = BCD (# J).
G65 H25 P#I Q#J
(Example) G65 H25 P#101 Q#102 ; (#101 = BCD(#102))

14) Decimal multiplication and division: #I = (# I× # J) ÷ # K
G65 H26 P#I Q#J R# k
(Example) G65 H26 P#101 Q#102 R#103 ; (#101 =(# 101×# 102)÷# 103)

15) Compound square root: #I =$\sqrt{\#j^2 + \#k^2}$
G65 H27 P#I Q#J R#K
(Example) G65 H27 P#101 Q#102 R#103 ; (#101 =$\sqrt{\#102^2 + \#103^2}$)

16) Sine: # I = # J•SIN(# K) (unit: degrees).
G65 H31 P#I Q#J R#K
G65 H31 P#101 Q#102 R#103 ; (#101 = #102•SIN(#103))

17) Cosine: # I = # J•COS(# K) (unit: degrees).
G65 H32 P#I Q#J R# k
(Example) G65 H32 P#101 Q#102 R#103 ; (#101 =#102•COS(#103))

18) Tangent: # I = # J•TAN(# K) (unit: degrees).
G65 H33 P#I Q#J R# K
G65 H33 P#101 Q#102 R#103 ; (#101 = #102•TAN(#103))

19) Anyway tangent: # I = ATAN(# J /# K) (unit: degrees).
G65 H34 P#I Q#J R# k
G65 H34 P#101 Q#102 R#103 ; (#101 =ATAN(#102/#103))

2. Transfer command

1) Unconditional transfer

G65 H80 Pn； n: Sequence number

G65 H80 P120 ； (Go to N120 Segment).

2) Conditional transfer 1 #J.EQ.# K(=).

G65 H81 Pn Q#J R# K； n: Sequence number

G65 H81 P1000 Q#201 R#202 ；

When #101 = #102, go to the N1000 section, and when #101 ≠ #102, execute sequentially.

3) Conditional Transfer 2 #J.NE.#K (≠).

G65 H82 Pn Q#J R# K； n: Sequence number

G65 H82 P1000 Q#101 R#102 ；

When #101 ≠ #102, go to the N1000 segment, and when #101 = #102, the program executes sequentially.

4) Conditional Transfer 3 #J.GT.#K (>).

G65 H83 Pn Q#J R# K； n: Sequence number

G65 H83 P1000 Q#101 R#102 ；

When #101 > #102, go to the N1000 segment, and when #101 ≤ #102, the program executes sequentially.

5) Conditional transfer 4 #J.LT.# K (＜)

G65 H84 Pn Q#J R# K； n: Sequence number

G65 H84 P1000 Q#101 R#102 ；

当# 101 ＜ #102 when, go to N1000 Program segment, when#101 ≥ #102 when the program is executed sequentially.

6) Conditional transfer 5 #J.GE.#K (≥).

G65 H85 Pn Q#J R# K； n: Sequence number

G65 H85 P1000 Q#101 R#102 ；

当# 101 ≤#102 when, go to N1000 Program segment, when#101 ＜ #102 when executed sequentially.

7) Conditional Transfer 6 #J.LE.# K (≤).

G65 H86 Pn Q#J R# K； n: Sequence number

G65 H86 P1000 Q#101 R#102 ；

When #101 ≤ #102, go to the N1000 section and execute sequentially when #101 > #102.

8) A P/S alarm has occurred

G65 H99 Pi； i: Alarm number +3000

(Example) G65 H99 P15 ；

P/S alarm 3015 occurred

Note: You can specify the sequence number with variables. Such as: G65 H81 P#100 Q#101 R#102; When the condition is met, the program moves to the program segment of the sequence number specified by **#100.**

3.17.3 Macro program call code

The difference between a user macro program call (G65) and a subroutine call (M98) is as follows:

1. With G65, you can specify independent variable data and transmit it to a macro program, but the M98 does not have this function.

2, with G65 can change the level of local variables, with M98 can not.

3. G65 The code is only allowed to appear before the code word N and the P or H code word is to appear immediately after it.

Nonmodal call (G65).

Code format: G65 P_ L_ < Argument>_; The macro program specified in address P is called, and the argument (data) is passed to the user macro body.

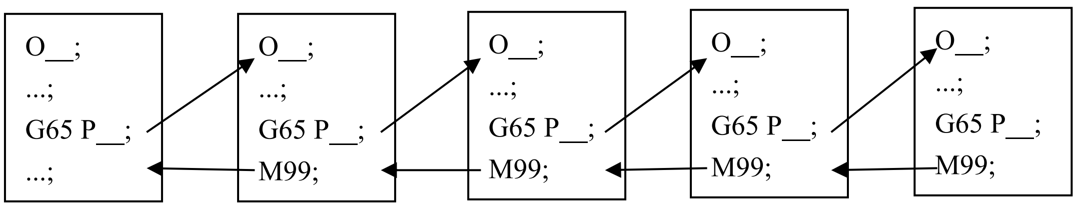
Code description: P_ The macro program number that was called

L_ is called (omits default to 1, you can specify the number of repetitions from 1 to 9999).

< arguments> data that is transferred to the macro program, whose values are assigned to the corresponding local variables.

Nested calls: G65 calls can have four levels of nesting.

Main Program Macro Program (Level 1) Macro Program (Level 2) Macro Program (Level 3) Macro Program (Level 4).



(0 level) (1 level) (2 levels) (3 levels) (4 levels)

#01 ... #33	#01 ... #33	#01 ... #33	#01 ... #33	#01 ... #33

Local variables

Specifying arguments: Use letters other than G, L, O, N, P , each letter can only be specified once, and the last specified is valid if repeatedly specified.

List of argument addresses and corresponding variable numbers for mode **I**

Argument address	Local variable number	Argument address	Local variable number	Argument address	Local variable number
A	#1	E	#8	U	#21
B	#2	F	#9	V	#22
C	#3	M	#13	W	#23
I	#4	Q	#17	X	#24
J	#5	R	#18	Y	#25
K	#6	S	#19	Z	#26
D	#7	T	#20		

Note: Addresses that do not need to be specified can be omitted, and local variables corresponding to omitted addresses will be assigned < empty >.

3.18 Metric Conversions

3.18.1 Function Overview

The input and output units of the CNC system are available in two units: metric units, millimeters (mm) and imperial units, inches (inch).

The parameters related to the metric system in the lathe system have the following status parameters:

NO.001 # 0 (INI): Enter incremental unit selection
0: Metric input (G21).
1: Imperial input (G20).

This parameter corresponds exactly to the function code G20/G21. That is, when the G20/G21 is executed in the program, this parameter also changes; When this parameter is modified, the G20/G21 modes change accordingly.

NO.003 # 0 (OIM):When the metric input mode is converted, whether the tool compensation value and wear value are automatically converted:

0: No automatic conversion (only one decimal place is moved).

1: Perform automatic conversion

NO.004 # 0 (SCW): Metric machine, Imperial machine selection (minimum output increment selection).

0: Metric machine output (0.001mm).

1: Inch machine output (0.0001inch).

3.18.2 Function codes G20/G21

Code format: G20; (inch input).

G21； (mm input).

The G code must be programmed at the beginning of the program and specified as a separate program segment.

3.18.3 Precautions

（1）. NO.001 #0(INI) Input increment unit change

(1). After the input incremental unit change (imperial/metric input) conversion, change the unit system of the following values (i.e., mm<>inch; mm/min<>inch/min):

- Feed speed (mm/min<>inch/min), thread lead (mm <>inch), specified by F code

—Position code (mm<>inch)

—Tool compensation value (mm<>inch)

— Scale unit of handwheel (mm<>inch)

— Distance of movement in incremental feed (mm<>inch)

— Some data parameters, including NO.45~NO.48, NO.56, NO.59, NO.60, NO.114~NO.116, NO.120~NO.131, NO.139, No.140, No.154; its units are 0.001mm (IS-B) when it is a metric input (G21), and its units are 0.0001inch (IS-B) when it is an imperial input (G20). For example, the same parameter NO.45 is set to 100, and the meaning represented when the input method is metric G21 is 100mm; The meaning represented when the input is imperial G20 is 100inch.

(2) After the input incremental unit change (imperial/metric input) conversion, the machine coordinates will be automatically converted:

（2）. NO.004# 0 (SCW) Output code units changed

SCW=0 indicates that the system's minimum code increment is output by metric output (0.001mm) SCW=1 when the system's minimum code increment is output by imperial (0.0001inch) The meaning of some data parameters changes when changing the output control bit parameter SCW:

(1) Speed parameters: metric
Machine: mm/min inch
Machine tool: 0.1 inch/min
For example: speed setpoint 3800, metric machines represent 3800 mm/min, and imperial machines represent 380 inch/min. These velocity parameters are: No.22, No.23, No.27, No.28~No.31, No.32, No.33, No.41, No.107, No.113, No.134;
(2) Position (length) Parameters: Metric
Machine: 0.001 mm inch
Machine tool: 0.0001 inch

For example: the setpoint is 100, the metric machine represents 0.1 mm, and the imperial machine represents 0.01 inch. These parameters are: No.34, No.35, No.37~ No.40, No.45~ No.48, No.102~ No.104, No.136~No.138 and all the pitch error compensation parameters;

Note 1: When the minimum input increment and the minimum instruction increment unit are different, the maximum error is half of the minimum instruction increment. This error does not accumulate.
Note 2: In the above instructions, the current system increment is IS-B.

Chapter 4 Tip Radius Compensation (G41, G42).

4.1 Application of tip radius compensation

4.1.1 Overview

Part machining procedures are generally prepared according to the part drawings at a certain point of the tool (usually with a hypothetical tip, as shown in point A of Fig. 4-1). However, in the actual machining of the turning tool, due to process or other requirements, the tip is often not a hypothetical point, but a circular arc. When cutting, there is a deviation between the actual cutting point and the cutting point in the ideal state, which will cause overcut or undercut, affecting the accuracy of the part. Therefore, the tip radius compensation is carried out in machining to improve the accuracy of the part.

Figure 4-1 Cutters

The method of offsetting the trajectory of the part shape by a tool tip radius is the B-type tool compensation method, which is simple, but after the execution of one program segment is completed, the motion trajectory of the next program segment is processed, so there will be over-cutting at the intersection of the two programs. In order to solve the above problems and eliminate errors, it is necessary to establish a C-type tool compensation method. The C-type tool compensation method is not executed immediately when reading into one program segment, but is read into the next program segment, and the corresponding motion trajectory (transfer vector) is calculated according to the connection of the intersection of the two program segments. Thanks to the reading of two program segments for pre-processing, the C-type tool compensation allows for more precise compensation on the contour. This is shown in Figure 4-2.

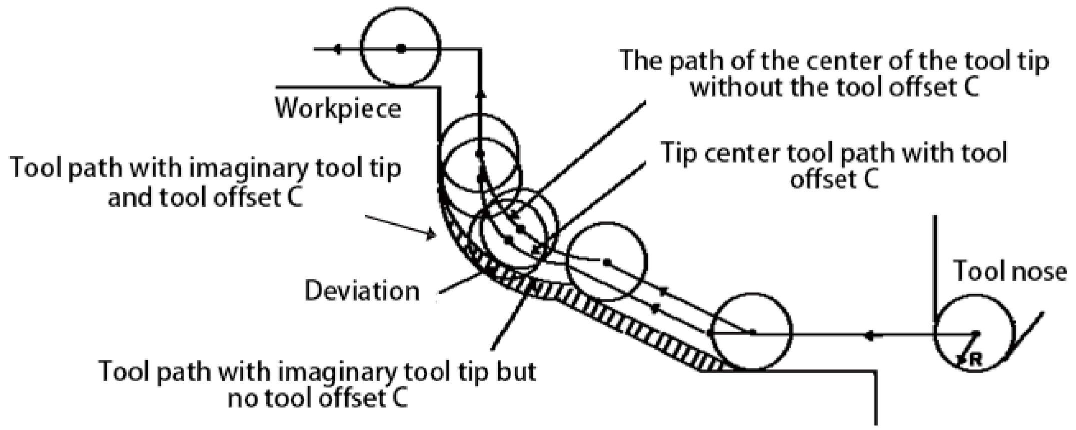


Figure 4-2

4.1.2 Imaginary tip direction

The imaginary tip setting is because it is generally difficult to set the center of the tip radius at the starting position, as shown in Figure 4-3; It is easier to imagine that the tip of the knife is set at the starting position, as shown in Figure 4-4; Tip radius can be programmed without regard to. Fig. 4-5 and 4-6 when programmed with a tip center and programmed with a hypothet tip, respectively, compared with the tool trajectory diagram using tip radius compensation without tip radius compensation.

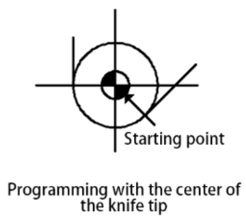


Figure 4-3

If tip radius compensation is not used, the tip center trajectory will be the same as the programmed trajectory

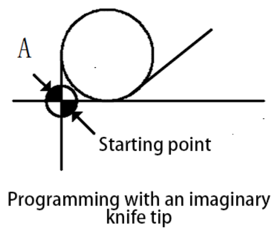


Figure 4-4

If tip radius compensation is used, precision cutting is achieved

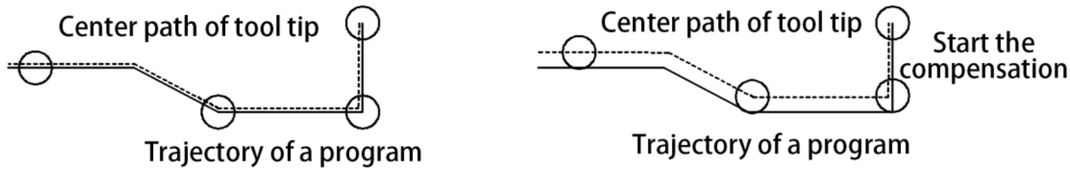


Figure 4-5

Without tip radius compensation, the hypothet tip trajectory will be the same as the programmed trajectory Using tip radius compensation, precision cutting will be achieved

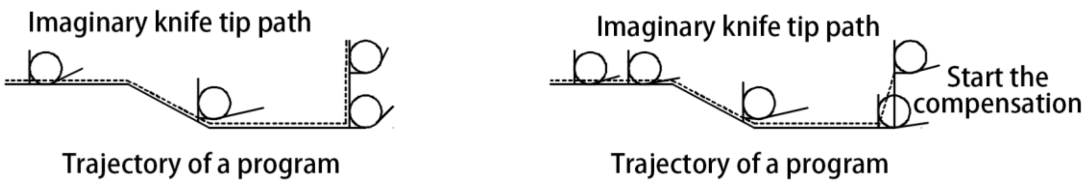


Figure 4-6

The tool trajectory programmed with a hypothetical tip is imagined to be a point during the programming process, and the actual cutting edge cannot be an ideal point due to process requirements or other reasons. This machining error, which is caused by the fact that the cutting edge is not an ideal point but a circular arc, can be eliminated by the tool tip arc radius compensation function. In actual machining, the imaginary tip point has a different position relationship with the center point of the tip arc, so it is necessary to correctly establish the tip direction of the imaginary tip (that is, which position of the tool is the tool to the tool point).

Looking from the center of the tip to the direction of the imaginary tip, the imaginary tip is determined by the direction of the tool in the cutting. The hypothetical tip has 10 (T0 ~ T9) settings, expressing the position relationship in 9 directions. It is important to note that even the same tip direction number is different in different coordinate systems (rear and front toolholder coordinate systems), as shown in the following figure. The relationship between the tip of the knife and the starting point is illustrated in the figure, and the end of the arrow is the imaginary tip; The case of the rear tool holder coordinate system T1 ~ T8, as shown in Figure 4-7; The case of the front tool holder coordinate system T1 ~ T8, as shown in Figure 4-8. T0 and T9 are the cases when the tip center is consistent with the starting point, as shown in Figure 4-9.

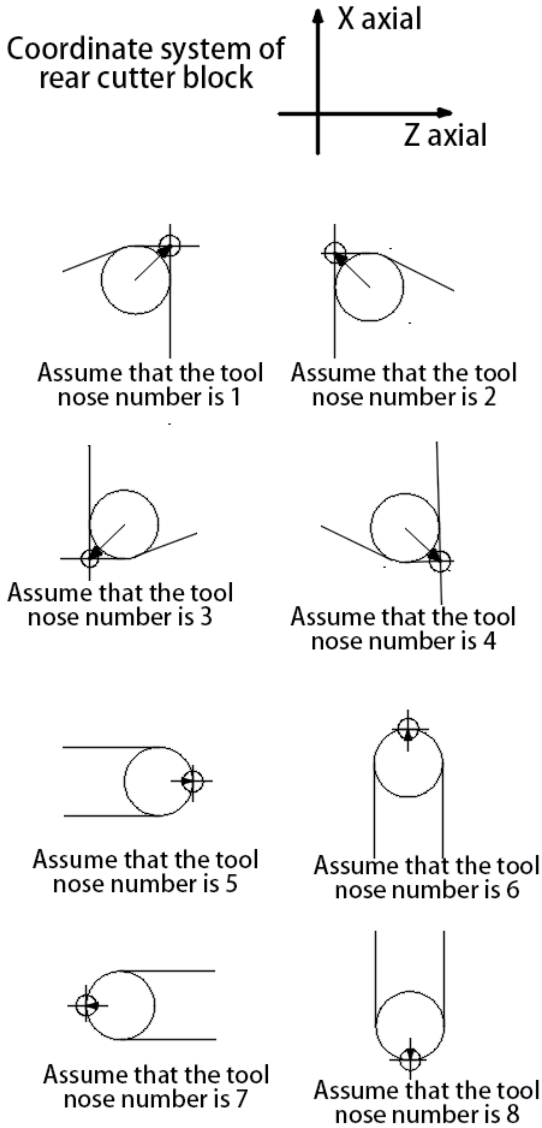


Fig. 4-7 Hypothetical tip number in the rear tool base coordinate system

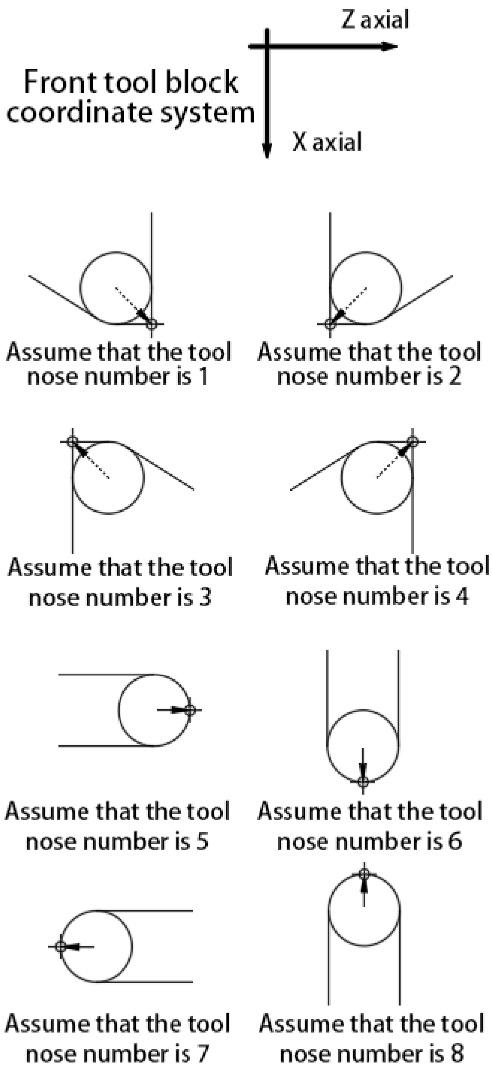


Figure 4-8 Hypothetical tip number in the front toolholder coordinate system

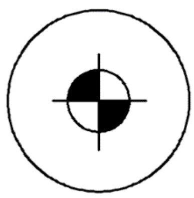


Figure 4-9 The center of the tip is consistent with the starting point

4.1.3 Setting of compensation values

The imaginary tip number and tip radius value for each knife must be pre-set before the C tool complement is applied. The tip radius compensation value is set under the bias page (see Table 4-1), R is the tip radius compensation value, and T is the imaginary tip number.

Table 4-1 CNC Tip Radius Compensation Values are displayed on the page

serial number	X	Z	R	T
00	0.000	0.000	0.000	0
01	0.020	0.030	0.020	2
02	1.020	20.123	0.180	3
...	...	...	...	...
32	0.050	0.380	0.300	6

Note: The X-oriented tool offset value can be specified with a diameter or radius value, set by the ORC of **bit4 bits of** the parameter **No.004, and the offset value is expressed as a radius at ORC = 1.** The offset value at **ORC = 0 is expressed as a diameter.**

When performing the knife operation, special attention should be paid to the imaginary tip of Tn(n=0 ~ 9) when the imaginary tip of Tn(n=0 ~ 9) is selected, and the point of the knife must also be the imaginary tip point of Tn(n=0 ~ 9). Figure 4-10 shows the different tool pairing methods when selecting T0 and T3 tip points in the rear toolholder coordinate system, with the center of the tool holder as the standard point, the same tool, from the standard point to the center of the tip radius (the hypothetical tip is T0). when) is not the same as the offset value from the standard point to the imaginary tip (when the imaginary tip is T3). It is much easier to measure the distance from the standard point to the imaginary tip than it is to measure the distance from the standard point to the center of the tip radius, so the tool offset value is usually set in terms of the distance from the standard point to the imaginary tip (i.e. the T3 tip direction is usually chosen).

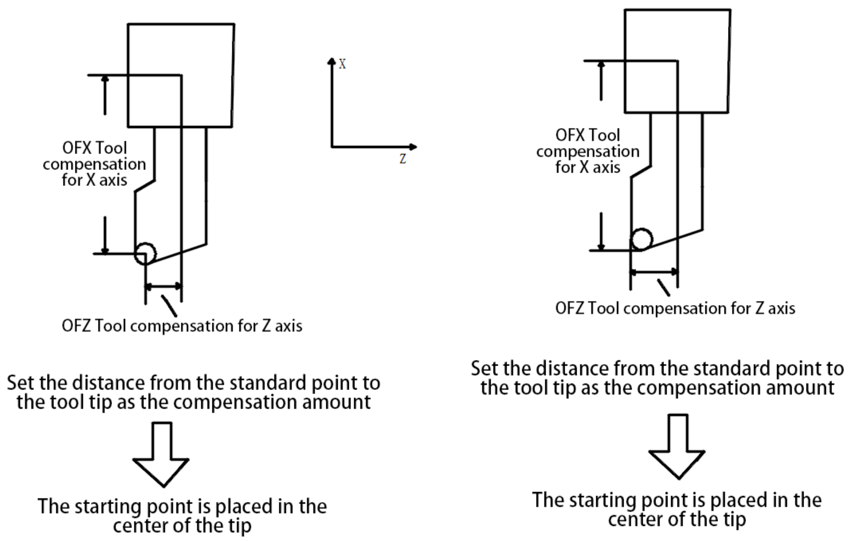


Figure 4-10 Tool offset values with the center of the tool holder as the reference point

4.1.4 Code Format

```
G40 ┐   ┌ G00 ┐
G41 ├ ─ ┤     ├ X_ Z_ T_
G42 ┘   └ G01 ┘
```

code	Function description	remark
G40	Eliminate tip radius compensation	
G41	In the rear tool holder coordinate system, G41 specifies the left knife complement, and the front tool holder coordinate system G41 specifies the right knife complement	See Figures 4-11 and 4-12 for details
G42	In the rear tool holder coordinate system, G42 specifies the right knife complement, and the front tool seat coordinate system G42 specifies the left knife complement	

4.1.5 Direction of Compensation

To apply tip radius compensation, the direction of compensation must be determined according to the position of the tip relative to the workpiece, such as 4-11, 4-12.

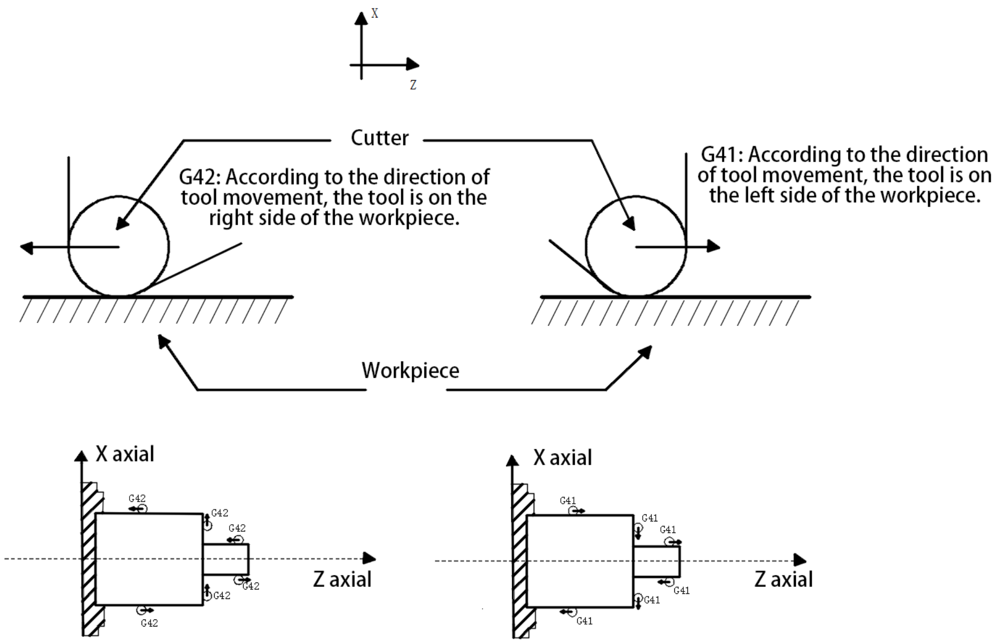


Fig. 4-11 Rear tool holder coordinate system compensation direction

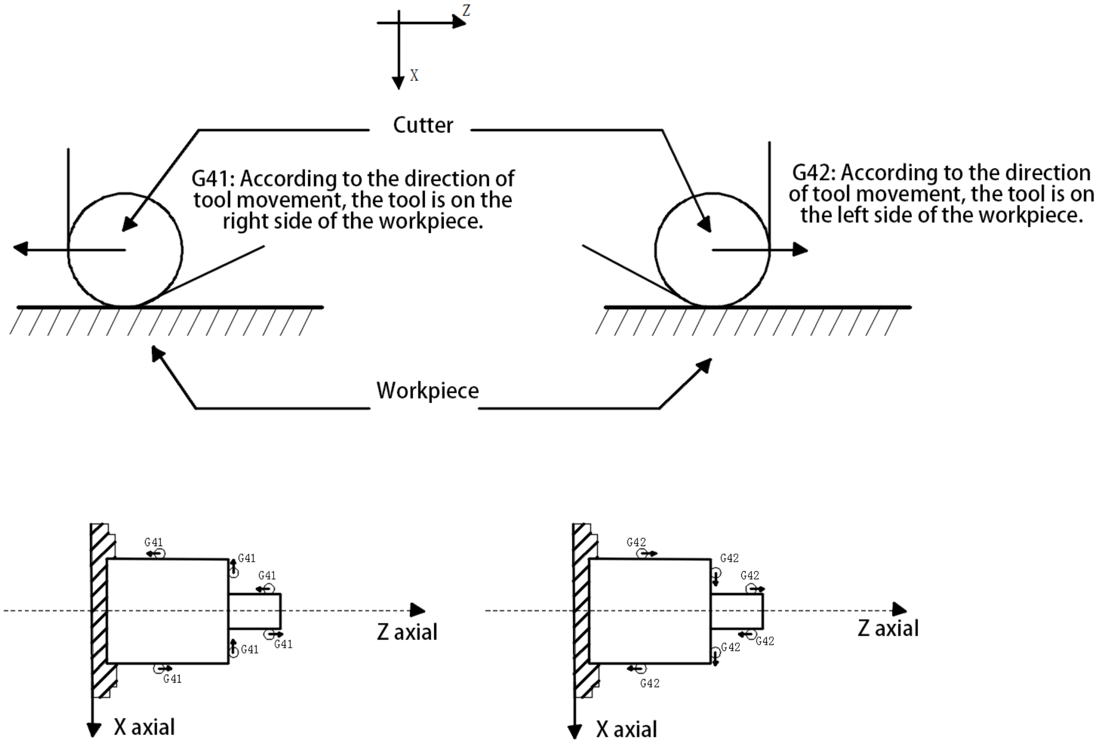


Fig. 4-12 The coordinate system of the front tool holder compensates for the direction

4.1.6 Precautions

* The initial state of the CNC is in the tip radius compensation cancellation mode, after executing the G41 or G42 code, the CNC begins to establish the tip radius compensation offset mode. At the beginning of the compensation, the CNC reads 2 program segments in advance, and when one program segment is executed, the next program segment is stored in the tip radius compensation buffer memory. In single-segment operation, two program segments are read in and stopped after the first program segment is executed at the end of the program segment. On consecutive executions, two program segments are read in advance, so the program segments that are being executed in the CNC and the two program segments that follow.

* In tip radius compensation, when processing 2 or more program segments without a move code (e.g. accessibility, pause, etc.), the tip center moves to the end of the previous program segment and is perpendicular to the position of the program path of the previous program segment.

* Knife patch C setup and tool patch C undo cannot be performed under Input Mode (MDI).

* Tip radius R value can not be entered negative value, otherwise the running trajectory error.

* The establishment and undoing of the tip radius compensation can only be with G00 or G01 codes, not arc codes (G02 or G03). If specified, an alarm is generated.

* After pressing the RESET (Reset) key or executing M30, the CNC will cancel the tool complement C compensation mode.

* G40 de-bias mode must be specified before the end of the program. Otherwise, the tool trajectory deviates from a tip radius value when executed again.

* Tip radius compensation is used in the main program and subprogram, and before the subroutine is called (i.e., before M98 is executed), the CNC must establish the tool complement C again in the subroutine in the compensatory cancellation mode.

*G71, G72, G73, G74, G75, G76 codes do not perform tip radius compensation, temporarily undo the compensation mode.

*The G90 and G94 codes perform tip radius compensation, and both G41 and G42 are offset by one tip radius (according to the imaginary tip 0 number) for cutting.

4.1.7 Application Examples

Machining the part shown in Fig. 4-13 in the front toolholder coordinate system. Use a tool number of T0101, a tip radius of R = 2, and a hypothet tip number of T = 3.

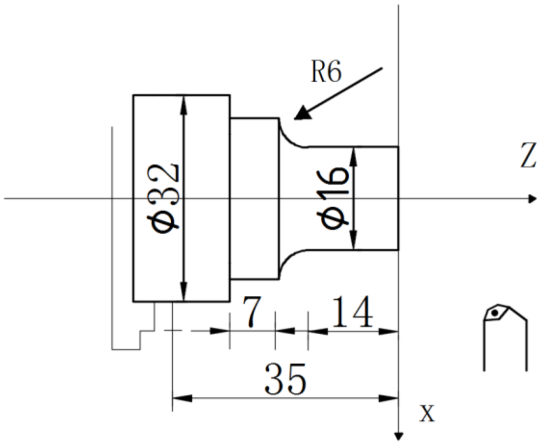


Figure 4-13

In the offset cancellation mode, the knife is paired, after the knife is completed, usually the Z axis should offset a tip radius value, and the direction of the offset is related to the hypothetical tip direction and the knife point, otherwise it will overcut a tip radius value when starting the knife.

Under the Knife Bias Settings page, set the tip radius R and the imaginary tip direction:

Table 4-2

serial number	X	Z	R	T
00	0.000	0.000	0.000	0
01	...	...	2	3
...	...	...	...	...
08	...	...	...	...

Procedure:

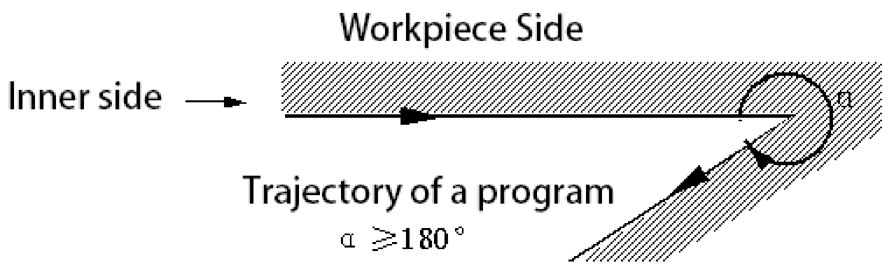
G00 X100 Z50 M3 T0101 S600; (Positioning, opening spindle, tool changing and performing tool repair).

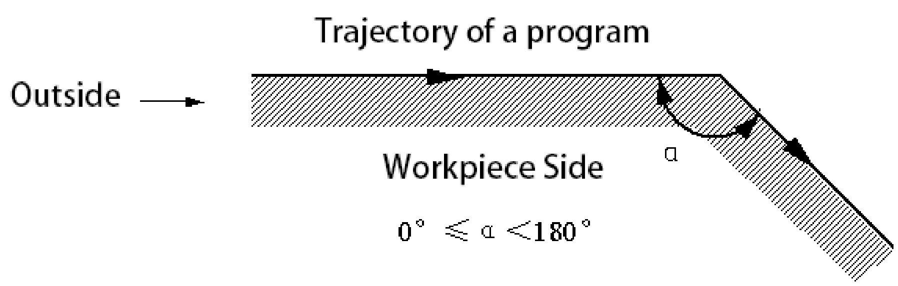
```
G42 G00 X0 Z3;           (Establish tip radius compensation).
G01 Z0 F300;             (Cutting begins).
X16;
Z-14 F200;
G02 X28 W-6 R6;
G01 W-7;
X32;
Z-35;
G40 G00 X90 Z40;         (Remove tip radius compensation).
G00 X100 Z50 T0100;
M30;
```

4.2 Tip radius compensation offset trajectory description

4.2.1 Inside and Outside Concepts

Two terms 'inside' and 'outside' will be used in the following instructions. The angle between the intersection points of two mobile program segments is greater than or equal to 180°, which is called 'inside'; when the angle between the intersection points of two moving program segments is between 0 and 180°, it is called 'outside'.





4.2.2 Tool movement at the time of tool start

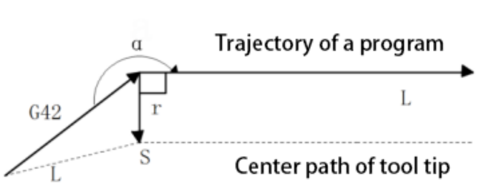
There are 3 steps to achieve tip radius compensation: knife patch establishment, knife patch implementation, and knife patch undo.

The movement of the tool from the offset cancellation method to the beginning of the G41 or G42 code is called tool patch setup (also known as tool start).

Note: The S, L, C noted in the figure below**, unless otherwise noted, have the following meanings: S - single-segment stop point; L - straight line; C - Arc.**

There are two types of toolpath actions at the start or cancellation of compensation, type A and type B, selected by the status parameter 172.3.

(a) Move along the inside of the corner (α ≥ 180°).



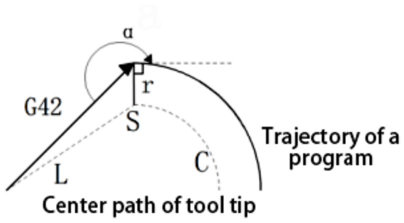
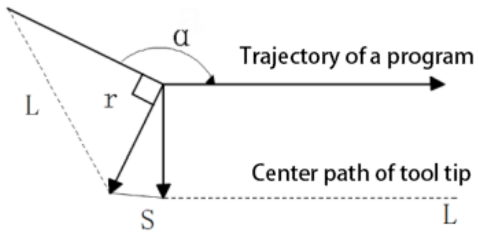


Fig. 4-14a Straight line - straight line (inner knife start) Figure 4-14b straight line - arc (inside knife start).

(b) Move along the outside of the corner where the corner is obtuse (180°> α ≥ 90°).



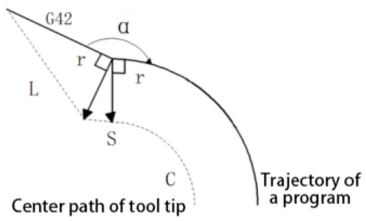
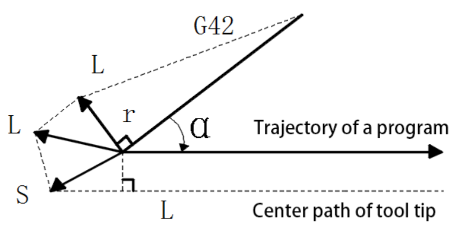


Fig. 4-15a Straight line - straight line (outer knife start) Fig. 4-15b Straight line - Arc (outside knife start)

（cMove along the outside of the corner where the sharp angle is (α ＜ 90°）



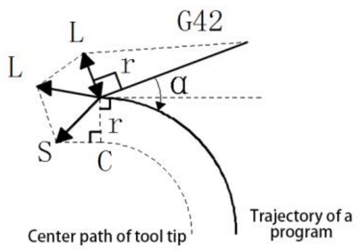


Fig. 4-16a Straight line - straight line (outer knife start) Fig. 4-16b Straight line - Arc (outside knife start)

(d) Move along the outside of an acute angle with an angle of less than 1 degree, and the straight line → straight line. （α ≦ 1°）

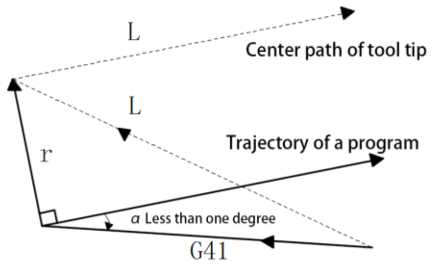


Fig. 4-17 Straight line - straight line (corner less than 1 degree, outside the knife).

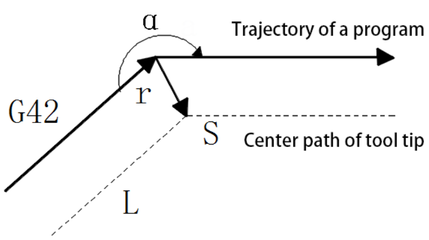
4.2.3 Tool movement in bias mode

After the tip radius compensation is established, the offset method is called before the tip radius compensation is canceled.

*** Offset trajectory in compensation mode that does not change the direction of compensation**

(a) Move along the inside of the corner (α ≥ 180°).

1) Straight line - straight line 2) Straight line - circular arc



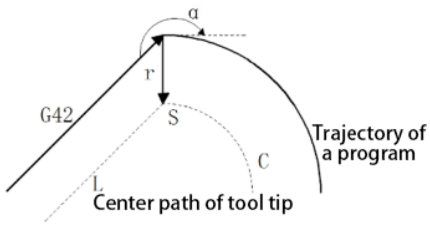
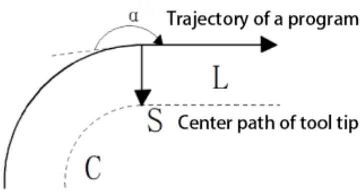


Fig. 4-18a Straight line - straight line (inner movement) Fig. 4-18b Straight line - Arc (inner movement)

3) Arc - straight line 4) Arc - arc



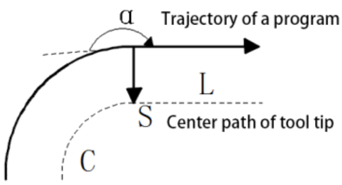
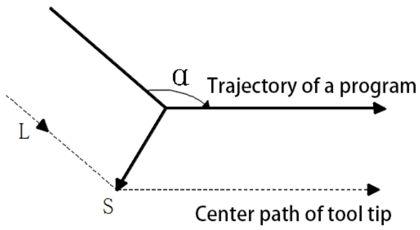


Fig. 4-18c Arc - Straight line (inner movement) Fig. 4-18d Arc - Circular arc (inner movement).

(b) Move along the outside of the corner where the corner is obtuse (180°> α ≥ 90°).

1) Straight line - straight line 2) Straight line - circular arc



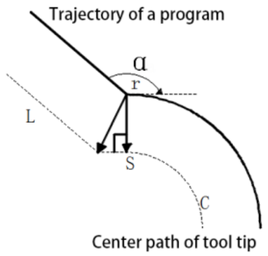
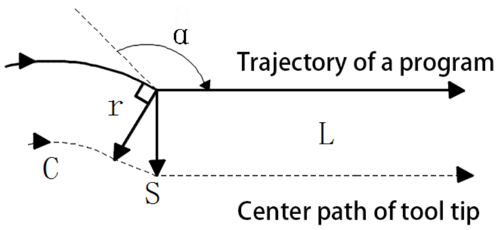


Fig. 4-19a Straight line - straight line (obtuse angle, lateral movement) Fig. 4-19b Straight line - Arc (obtuse angle, lateral movement)

3) Arc - straight line 4) Arc - arc



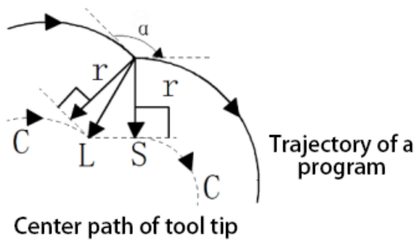
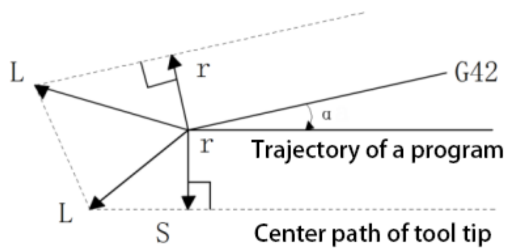


Fig. 4-19c Arc - straight line (obtuse angle, lateral movement) Fig. 4-19d Arc - circular arc (obtuse angle, lateral movement).

（cMove along the outside of the corner where the sharp angle is (α ＜ 90°）

1) Straight line - straight line 2) Straight line - circular arc



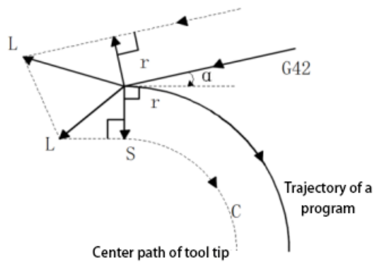
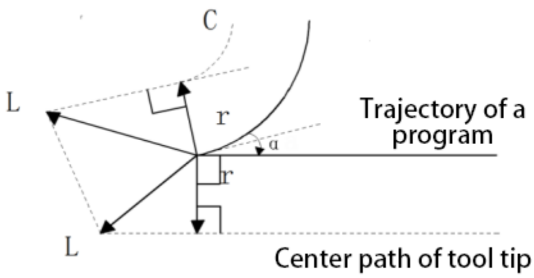


Fig. 4-20a Straight line - straight line (sharp angle, outer movement) Fig. 4-20b Straight line - Arc (sharp angle, outer movement)

3) Arc - straight line 4) Arc - arc



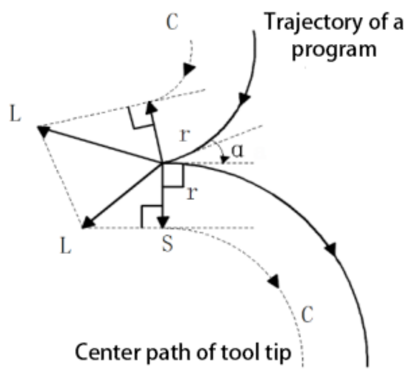


Fig. 4-20c Straight line - straight line (sharp angle, outside movement) Fig. 4-20d straight line - arc (sharp angle, outer movement)

5) Less than 1 degree of internal processing and compensation vector amplification

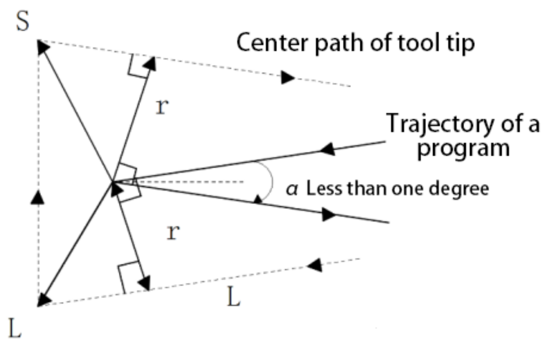


Fig. 4-20e Straight line - straight line (corner less than 1 degree, inner movement)

(d) Special circumstances

1) When there is no intersection

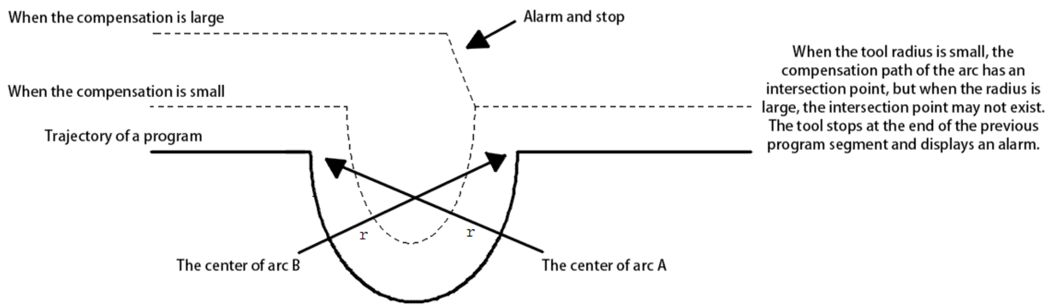


Fig. 4-21 Special case - the trajectory after bias has no intersection points

2) The center of the arc matches the start or end point

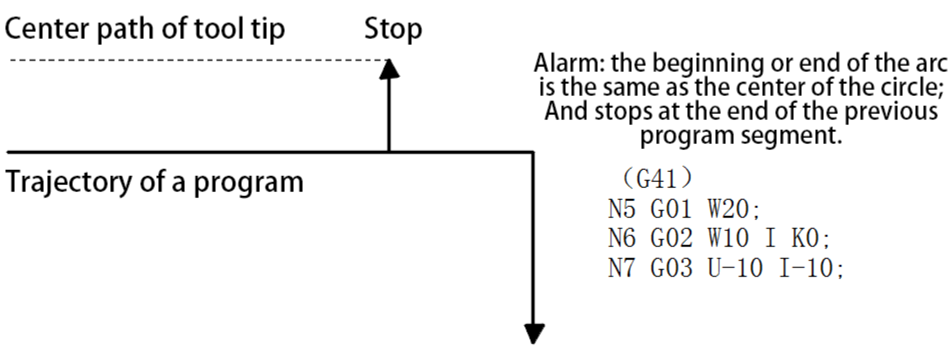


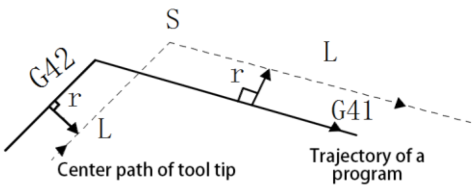
Fig. 4-22 The center of the arc is consistent with the start or end point

*** Offset trajectory that changes the direction of compensation in compensation mode**

Compensation quantity number / G code	+	-
G41	Left side compensation	Right side compensation
G42	Right side compensation	Left side compensation

In special cases, the direction of compensation can be changed in the compensation mode. However, it is not possible to change the starting program segment and its subsequent program segments. When compensating for a change in direction, there is no concept of the inside and outside sides of the full condition. The following compensation amounts are assumed to be positive.

1) Straight line - straight line 2) Straight line - circular arc



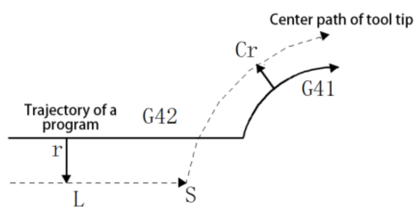
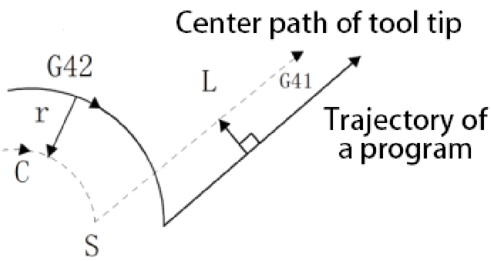


Fig. 4-23 Straight line - straight line (change compensation direction) Fig. 4-24 Straight line - Arc (change compensation direction)

3) Arc - straight line 4) Arc - arc



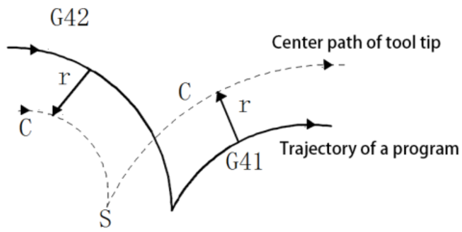
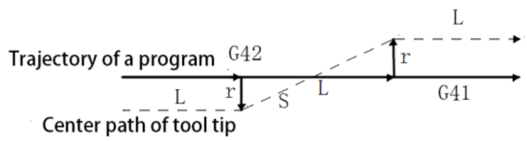


Fig. 4-25 Arc - straight line (change compensation direction) Fig. 4-26 Arc - arc (change compensation direction)

5) If the compensation is performed normally, but there is no intersection

When using G41 and G42 to change the bias direction of program segment A to program B, if the intersection of the bias path is not needed, a vector of vertical and program B is made at the beginning of program segment B.

1） Straight line - straight line



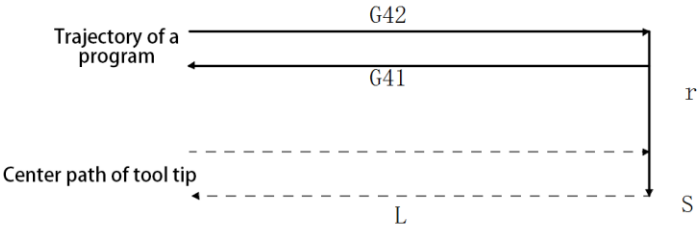


Fig. 4-27a Straight line - straight line, no intersection point (change compensation direction)

2） Straight line - arc

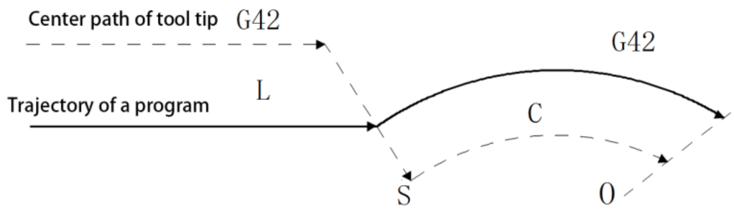


Fig. 4-27b Straight line - arc, no intersection point (change compensation direction).

3) Circular arc - circular arc

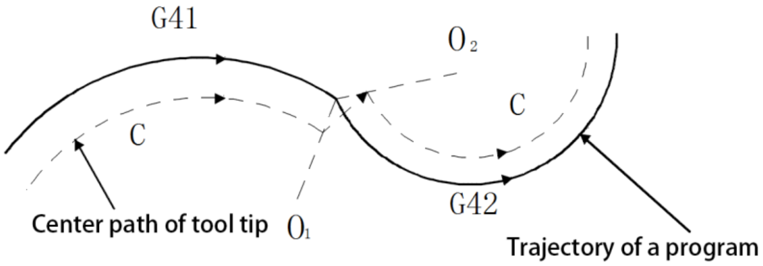


Fig. 4-27c Arc - circular arc, no intersection point (change compensation direction).

4.2.4 Tool movement in bias cancellation mode

In compensation mode, when the program segment meets any of the following conditions, the CNC enters the compensation cancellation mode, and the action of this program segment is called compensation cancellation.

1. The G40 code is used in the program;

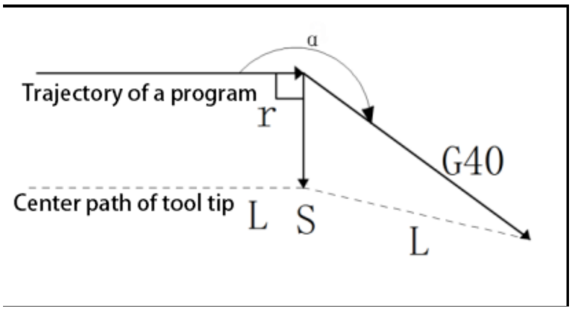
2. The M30 code was executed.

Arc codes (G02 and G03) are not available when C tool patch is canceled. If the command arc generates an alarm (N0.34) and the operation stops.

In compensatory cancellation mode, control the execution of the program segment and the program segment in the tip radius compensation buffer register. At this point, if the Single Segment switch is On, it stops after executing one program segment. Press the start button again to execute the next program segment without reading the next one.

（a） Move along the inside of the corner (α ≥ 180°)

1) Straight line - straight line 2) Straight line - circular arc



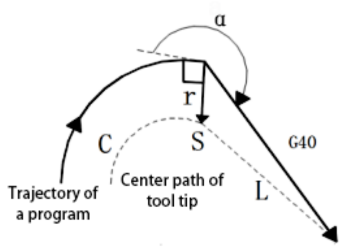
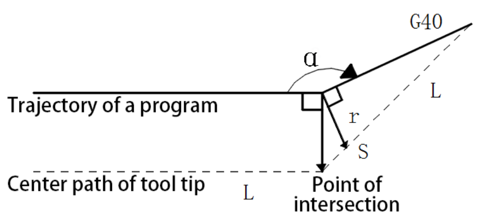


Fig. 4-28a Straight line - straight line (inside, cancel offset) Fig. 4-28b Arc - straight line (inside, cancel offset)

（b） Move along the outside of the corner with obtuse angle (180°> α ≥ 90°)

1) Straight line - straight line 2) Arc - straight line



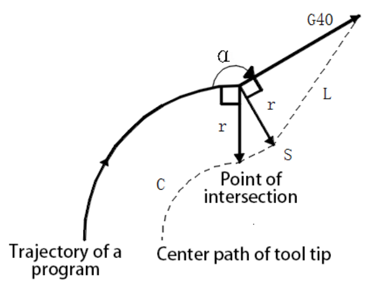
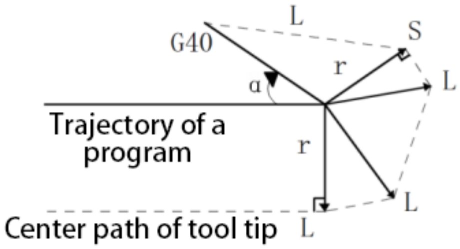


Fig. 4-29a Straight line - straight line (obtuse angle, outside, cancel offset) Fig. 4-29b Arc - straight line (obtuse angle, outside, cancel offset)

（c） Move along the outside of the sharp corner (α < 90°)

1) Straight line - straight line 2) Arc - straight line



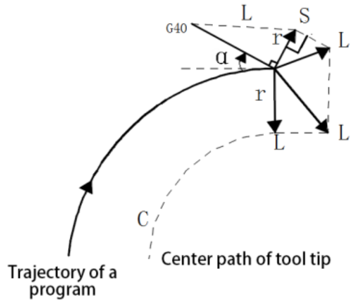


Fig. 4-30a Straight line - straight line (sharp angle, outside, cancel offset) Fig. 4-30b Arc - straight line (sharp angle, outside, cancel offset)

（d） Move along the outside of the sharp angle where the corner is less than 1 degree; Straight lines → straight lines. （α ＜ 1°）

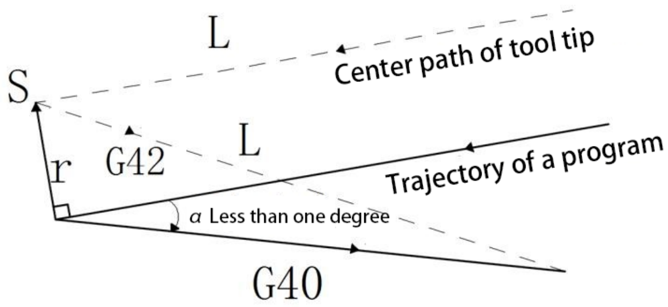


Fig. 4-31 Straight line - straight line (angle less than 1 degree, outside, cancel offset).

4.2.5 Tool interference check

Tool transition cutting is called "interference", and interference checks the tool transition in advance, even if the transition cut does not occur. But not all tool interference can be detected.

(1) Basic conditions for interference

1) The toolpath direction is different from the program path direction. (The angle between paths is between 90 degrees and 270 degrees).

2) In arc machining, in addition to the above conditions, the angle between the start and end points of the tool center path is very different from the angle between the start and end points of the program path (more than 180 degrees).

Example: Linear machining

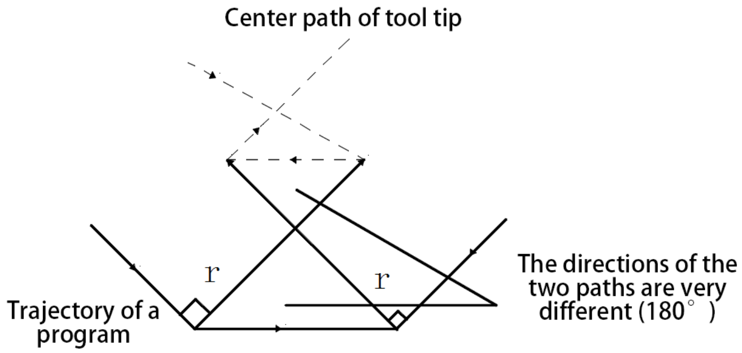


Fig. 4-32a Process interference (1)

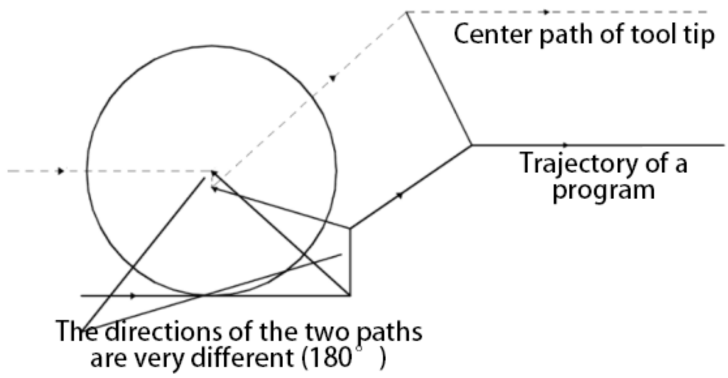


Fig. 4-32b Processing interference (2).

(2) There is no actual interference, and it is treated as interference.

1) The groove depth is less than the compensation amount

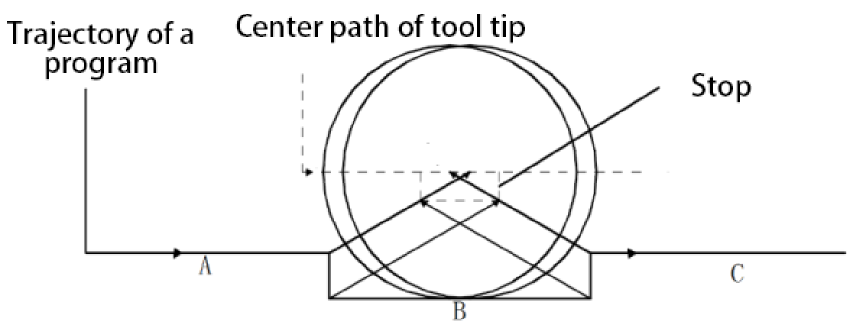


Fig. 4-33 Special case of interference treatment (1).

There is actually no interference, but in program segment B the direction of the program is opposite to the path compensated by the radius of the tip, the tool stops and displays an alarm.

2) The depth of the groove is less than the compensation amount

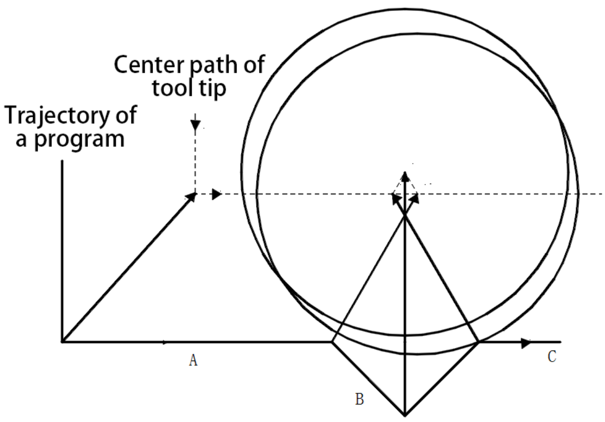


Fig. 4-34 Special cases for interference treatment (2).

There is actually no interference, but in program segment B the direction of the program is opposite to the path compensated by the radius of the tip, the tool stops and displays an alarm.

4.2.6 Temporarily cancel the code of the compensation vector

In the compensation mode, if the G50, G71 ~ G76 code is specified, the compensation vector is temporarily canceled, and after the code is executed, the compensation vector is automatically restored. The compensation temporary cancellation at this time is different from the compensation cancellation mode, and the tool moves directly from the intersection point to the command point of the compensation vector cancellation. When compensation mode is restored, the tool moves directly to the intersection point again.

***Coordinate system setting G50 code**

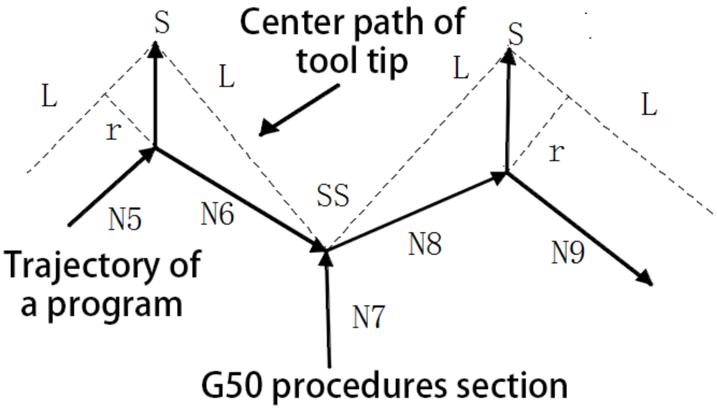


Fig. 4-35 G50 temporarily cancels the compensation vector

Note: SS represents the point where the tool stops twice in single-segment mode.

***G28 automatically returns to the reference point**

In compensation mode, if G28 is instructed, compensation is canceled at the midpoint and the compensation mode automatically resumes after the reference point is returned.

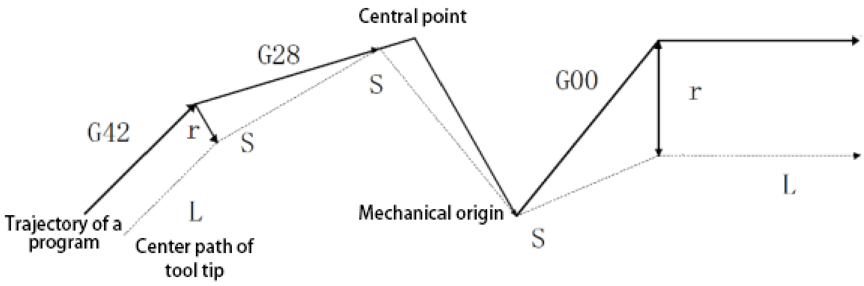


Fig. 4-36 G28 temporarily cancels the compensation vector

***G71 ~ G75 compound cycle; G76, G92 thread cutting**

When executing G71 ~ G76 fixed loop code; G92 thread cutting code, in the process of cycling, do not perform the tip radius compensation, temporarily cancel the tip radius compensation, in the later program segment there are G00, G01 code, CNC will automatically restore the compensation mode.

***G32, G33, G34 and other thread cutting**

Can not be operated in tip radius compensation mode, if running will alarm No. 131 "... Instructions cannot be used in C knife repair".

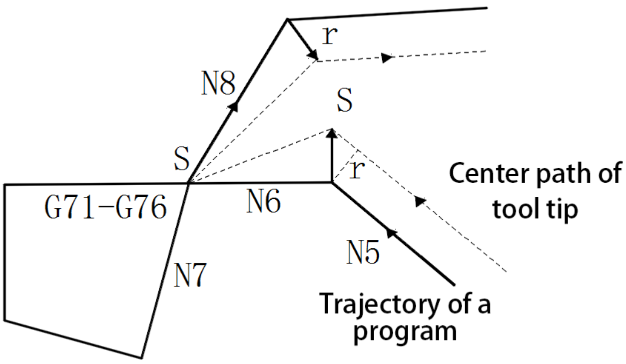


Fig. 4-37 G71~G76 temporarily cancels the compensation vector

***G90, G94 code**

How G90 or G94 codes perform tip radius compensation:

A. The original tip radius compensation will be undone when positioned at the beginning of the cycle;

B. Establish the previous C compensation before the start of cutting, and the trajectory (1) below will establish the original radius compensation mode;

C. The trajectories (2) and (3) of the following figures are the cutting with radius compensation;

D. The trajectory (4) below will undo the radius compensation and return to the beginning of the loop; There are G00 and G01 codes in the later program segments, and the CNC will automatically restore the compensation mode;

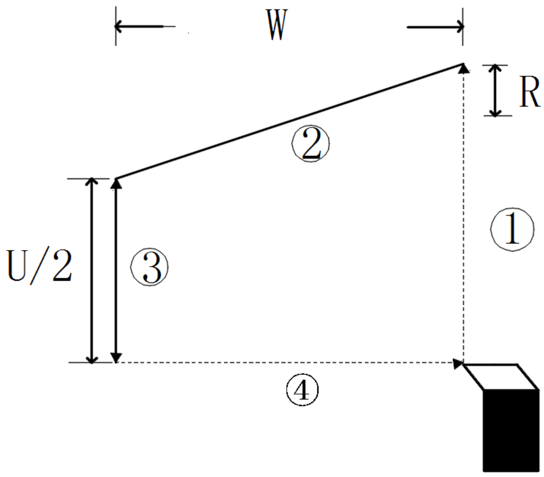


Fig. 4-38 Bias direction of G90 tip radius compensation

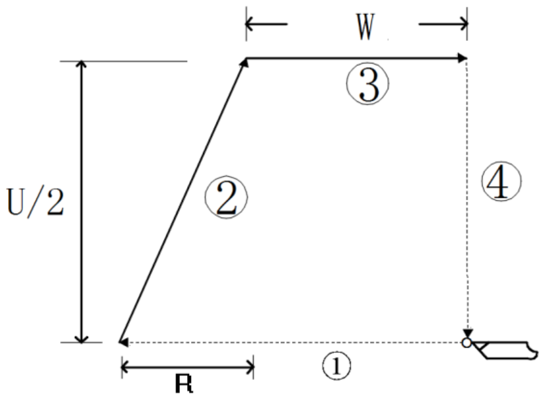


Fig. 4-3The bias direction of the **9 G94 tip radius compensation**

4.2.7 Special Circumstances

* When the inner corner machining is less than the radius of the tip, the inside bias of the tool will cause excessive cutting. After the start of the previous program segment or the movement of the corner, the tool movement stops and an alarm is displayed (P/S41). However, if the 'Single Program Segment' switch is ON, the tool stops at the end of the previous program segment.

* When machining a concave shape smaller than the diameter of the tip, an overcut will occur when the tip radius compensation causes the tip center to form a direction opposite to the program path. At this point, after the start of the previous program segment or after the corner movement, the tool movement stops and an alarm is displayed.

* When machining a step smaller than the tip radius When the program contains a step smaller than the tip radius and the step is in turn an arc, the center path of the tool may form a direction of motion opposite to the program path. The first vector is automatically ignored and the line moves directly to the end of the second vector. When the program is single-segmented, the program stops at this point, and if it is not in single-segment mode, the loop continues. If the steps are straight, the compensation is performed correctly without generating an alarm. (However, the uncut part remains)

*When subroutines are included in G code, the CNC must be in compensated cancellation mode before the subroutines are called (i.e., before M98 is executed). Once the subroutine is entered, the bias can be started, but the compensatory cancellation mode must be in place before returning to the main program (i.e., before M99 is executed). Otherwise, an alarm will appear.

* When the amount of compensation is changed

(a) Usually when canceling the mode tool change, the value of the compensation amount is changed. If the amount of compensation is changed in the compensation mode, the new amount of compensation is valid only after the tool change.

(b) The positive and negative amount of compensation and the center path of the tip, if the compensation amount is negative (-), the G41 and G42 are procedurally exchanged with each other.

If the center of the tool moves along the outside of the workpiece, it will move along the inside and vice versa.

The following example shows. Generally, the amount of compensation when making a program is (+). When the toolpath is made in (a), if the compensation amount is negative (-), the center of the tool moves as (b), and vice versa. Also note that when the offset quantity symbol changes, the tip offset direction also changes, but the hypothetical tip direction does not change. So do not arbitrarily change the sign of the bias quantity.

* When the end point of the programmed arc is not on the arc, the tool movement stops and displays the alarm message "The end of the arc is not on the arc".

Appendix I

1、CNC alarm

Alarm number content

0000 modifies the parameter that the power supply must be cut off once
0001 Failed to open file
0002 The data entered is out of range
0003 The program number copied or renamed exists
0004 Address not found
0005 There is no data after the address
0006 Illegal use of minus signs
0007 Illegal use of decimal points
0008 The program file is too large to load completely
0009 Enter the illegal address
0010 incorrect G code

0011 No feed speed instruction
0012 Insufficient disk space
0013 The number of program files has reached its upper limit
0014 can not command G95, spindle is not supported
0015 instructs too many axes
0016 The current pitch error compensation point is out of range
0017 does not have permission to modify
0018 does not allow modification
0019 The zoom function is not enabled
0020 exceeds radius tolerance

0021 instructed the illegal planar shaft
0022 In the arc, R and IJK are all 0
0023 Arc Interpolation is specified simultaneously in IJK and R
0024 Helical interpolation rotation angle is 0
0025 G12 cannot be in the same paragraph as other G directives
0026 System does not support file formats
0027 The length knife complement instruction cannot be in the same paragraph as G92
0028 Illegal plane selection
0029 illegal bias value
0030 Illegal Compensation Number

0031 G10 instructed illegal P

0032 Illegal compensation value in G10
0033 Knife patch C or no intersection in chamfer
0034 Arc command can not establish or cancel the tool complement
0035 The M99 directive did not cancel the C tool patch before the instruction
0036 can not command G31
0037 in the knife patch C can not change the plane
0038 Interference in the arc program segment
0039 Tool tip positioning error in knife patch C
0040 Tool Patch C changes the coordinate system of the workpiece during execution
0041 There is interference in knife patch C
0042 More than ten non-moving commands in tool complement C
0043 Insufficient permissions
0044 Does not allow instructions G27~G30 in a fixed loop
0045 Address Address Q is not discovered or the Q value is 0 (G73/G83).
0046 Illegal reference point returns instruction
0047 Mechanical return to zero is required before executing this instruction
0048 The Z plane should be higher than the R plane
0049 The Z plane should be lower than the R plane
0050 should move the position when changing the fixed circulation mode

0051 Incorrectly moves after chamfering or chamfers values are too large
0052 The mirroring function cannot be used for the slot milling fixed cycle
0053 Too many address directives
0054 DNC transmits an error
0055 Incorrect move value in chamfer or invert R
0056 M99 cannot be in the same segment as the macro program directive
0057 Failed to write file , must power off reboot
0058 No endpoint found
0059 No program number found
0060 No sequence number found

0061 The X axis is not at the reference point
0062 The Z axis is not at the reference point
0063 The Y axis is not at the reference point
0064 The 4TH axis is not at the reference point
0065 The TH5 axis is not at the reference point
0066 The fixed loop must be canceled before G10 is executed
0067 G10 does not support the setting format
0068 The parameter switch is not turned on
0069 Processing operation requires closing the USB flash drive operation interface

0070 Insufficient memory capacity Insufficient memory

0071 No data ends found
0072 Too many programs
0073 The program number is already in use
0074 Illegal Program Number
0075 protection
0076 does not define address P
0077 Subroutine nesting error
0078 No program number found
0079 System usage time expires
0080 The data entered is unreasonable

0082 The H code was instructed in G37
0083 G37 Illegal Axis Directive
0084 The button has a timeout or short circuit
0085 Communication error
0087 The X-axis reference point returns unfinished
0088 The Z axis reference point returns unfinished
0089 The Y-axis reference point returns unfinished
0090 4TH Axis Reference Point Returns Unfinished

0091 The TH5 axis reference point returns unfinished
0092 is not the axis at the reference point
0094 does not allow type P (coordinate).
0095 TypeP is not allowed (EXT OFS CHG).
0096 Type P Not Allowed (WRK OFS CHG).
0097 Type P is not allowed (auto-execute).
0098 G28 was found in sequential return
0099 MDI is not allowed after retrieval
0100 parameter writes are valid

0101 Power off memory data is confused , make sure the position is correct
0110 Position data exceeds the allowable range Please return to zero
0111 Compute data overflow
0112 is divided by zero
0113 Incorrect instruction
0114 Macro program malformation
0115 Illegal variable
0116 Write a protected variable
0118 Curly brace nesting error
0119 M00~M02, M06, M98, M99, M30 cannot be in the same segment as other M instructions

0122 Quad Macromodal - Call
0123 Macros cannot be used in DNC
0124 The procedure ended illegally
0125 Macro program malformation
0126 Number of illegal loops
0127 NC and macros are in the same program segment
0128 The sequence number of the illegal macro
0129 Illegal argument address
0130 Illegal shaft operation

0131 Too many external alarm messages
0132 No alarm number found
0133 Axis instructions are not supported by the system
0134 The system cannot use rigid tapping when the number of control axes is greater than 3 axes
0135 Illegal Angle Directive
0136 Illegal Axis Directive
0139 can not change the PLC control axis

0142 Illegality ratio rate
0143 Zoom motion data overflow
0144 Illegal plane selection
0148 Illegal data settings
0149 G10L3 format error
0150 Illegal tool set number

0151 No tool set number found
0152 Tool data cannot be stored
0153 The C knife patch was not cancelled before the tool change
0154 Unused life group tools
0155 Illegal T code in M06
0156 No P/L instructions found
0157 Too many tool sets
0158 Illegal tool life data
0159 Tool data setting is not completed
0160 Arcs in polar coordinate mode can only be programmed using R

0161 The directive cannot be executed in polar coordinate mode
0162 uses the G70~G76 instruction in the input mode
0163 The instruction cannot be executed in the rotation mode
0164 The instruction cannot be executed in zoom mode
0165 Please specify this directive within a separate program segment
0166 Back to reference point did not specify an axis

0167 The midpoint coordinates are too large

0168 The minimum pause time for the bottom of the hole should be less than the maximum pause time for the bottom of the hole

0170 Tool radius compensation is not canceled when entering or exiting the subroutine

0172 In the program segment that calls the subroutine , P is not an integer or P is less than 0

0173 The number of subroutine calls should be less than 9999

0175 The fixed loop can only be executed in the G17 plane

0176 The spindle speed is not specified

0177 Does not support spindle orientation

0178 The spindle speed was not specified before the start of the fixed cycle

0181 Illegal M code

0182 Illegal S code

0183 Illegal T code

0184 The selected tool is out of range

0185 L is too small or L is undefined

0186 L is too large

0187 The tool radius is too large

0188 U is too big

0189 The U value is less than the tool radius

0190 V is too small or V is undefined

0191 W is too small or W is undefined

0192 Q is too small or Q is undefined

0193 I is undefined or I is 0

0194 J is undefined or J is 0

0195 D is undefined or D is 0

0198 Illegal axis selection

0199 Macro directive undefined

0200 Illegal S Mode Directive

0201 The feed rate was not found in the rigid tapping

0202 position LSI overflow

0203 The program is not right in rigid tapping

0204 Illegal shaft operation

0205 Rigid mode DI signal off

0206 can not change the plane (rigid tapping).

0207 tapping data is not correct

0208 The instruction cannot be executed in the G10 mode

0212 Illegal plane selection
0224 Returns the reference point

0231 G10 illegal format in L50 or L51
0232 The instruction has too many spiral interpolation shafts
0233 Equipment busy
0235 Record ended
0236 Program restart parameter error
0237 has no decimal point
0238 Address duplicate error
0239 Parameter 0
0240 G41/G42 is not allowed in MDI mode

0241 Handwheel pulse abnormality
0243 Spindle pulse abnormality
0244 Thread processing speed exceeds the upper limit
0245 Spindle speed fluctuations exceed the limit value during thread machining
0251 emergency stop alarm
0255 The thread segment cannot specify the spindle speed
0256 Thread lead out of range
0257 The T directive is used in the program segment of the G71~G73 directive
0258 Address P or Q specifies M98, M99, or M30 in the two program segments specified
0259 In the G71/G72 instruction, the address Z(W)/X(U) is instructed in the P program segment
0260 axis name repeat , please modify the parameters NO.225~227

0261 Tool offset number exceeds the valid range (0~32).
0262 The tool number is not within the range set by data parameter No.084
0263 Tool life management , tool group number out of range (1~32).
0264 T instruction cannot be executed in the C tool patch, please undo the C knife patch
0265 G70~G76, G90, G92, G94, etc. can only be used in the G18 plane
0266 Cannot execute plane conversion instructions G17~G19
0267 G11 or G13.1 is missing from the program
0268 Tool life management , there are no tools in the current tool set
0269 Tool life management, the current tool set is not defined
0270 The life of all tools in the same group has been reached
0271 Tool life management function is invalid , do not use the G10 L3 instruction
0272 G11 cannot be programmed before G10
0273 G33 The amount of movement in the X direction when tapping is not 0
0274 The number of thread indexing heads is greater than 65535 heads

0275 In G90, the absolute value of R in the G92 instruction is greater than the absolute value of U/2

0276 The absolute value of R in the G94 directive is greater than the absolute value of W

0277 The G70~G73 directive contains more than 31 finishing program segments

0278 Ns and Nf sequence of finishing program segments in the G70~G73 instruction are incorrect

0279 The loop segment number Ns or Nf in the G70~G73 directive does not exist

0280 The G70~G73 directive did not enter the loop start loop termination segment number

0281 Subroutine was called in the G70~G73 loop

0282 G70~G73 There is no instruction G00 or G01 in the beginning of the loop

0283 The prohibited G directive was used in the beginning of the G70~G73 loop

0284 G70~G73 The prohibited G directive is used in the loop termination segment

0285 Uses the G70~G73 directive in the input mode

0286 In the G71~G72 cyclic finishing program segment, the coordinate changes are not monotonous

0287 The single-shot feed volume in the G71 or G72 is out of the allowable range

0288 The single retract volume in the G71 or G72 is out of the allowable range

0289 The first paragraph of the G71 directive instructs Z or W

0290 The first paragraph of the G72 instruction gives instructions to X or U

0291 The total cutting volume of the G73 is beyond the allowable range

0292 The number of cycles for G73 is less than 1 or greater than 9999

0293 The single retract R(e) in the G74 or G75 is out of the allowable range

0294 In G74 or G75 , the amount of retract when cutting to the end point is negative

0295 The amount of a single cut in the X or Z direction in the G74 or G75 exceeds the allowable range

0296 The value of Z was not entered in the G74 directive

0297 The value of Q in the G74 directive is 0 or not entered

0298 The value of X was not entered in the G75 directive

0299 The value of P in the G75 directive is 0 or not entered

0300 G76 When machining taper threads, the starting point is between the thread start point and the thread end point

0301 The minimum cutth in the G76 directive is out of the allowable range
0302 G76 Finishing allowance exceeds the allowable range
0303 G76 Teeth height is less than finishing allowance or less than 0
0304 G76 The number of cycles exceeds the allowable range
0305 G76 The width of the thread chamfer is beyond the allowable range
0306 The tip angle in the G76 instruction is out of the allowable range
0307 The G76 instruction has an X or Z axis movement of 0
0308 The G76 directive does not specify a thread tooth height p-value
0309 The G76 directive does not specify a Q value for the first depth of cut or a Q value of 0
0310 The cycle starts in a closed area formed by the start and end points of the finishing trajectory

0311 Pitch less than 0 occurs during thread cutting
0312 The G76 instruction in the tooth height is less than the X axis movement
0320 Additional axis command without chamfer function
0321 Uses the WHILE, END directive in the input mode
0322 Macro statement format specified error
0323 In the macro statement DO, the END label is not 1,2,3
0324 The macro statement in DO, END format specified incorrectly
0325 Parentheses in macro statements do not match or format specified incorrectly

0326 The divisor in the macro statement cannot be 0
0327 The arctangent SPECIFIED INAN format error in the macro statement
0328 The opposition number of LNs in the macro statement is 0 or less than 0

0329 The opening square in the macro statement cannot be negative
0330 The result of the tangent TAN in the macro statement is infinity

0331 The operands of the ASIN or ACOS in the macro statement are out of the range -1 to 1
0332 Macro variable number or variable value in macro statement is illegal (Error).

0451 X-axis drive alarm
0452 Z-axis drive alarm
0453 Y-axis drive alarm
0454 4TH Axis Drive Alarm
0455 TH5 Axis Drive Alarm
0456 Spindle Drive Alarm

0500 Soft Limit Overrun : -X
0501 Soft Limit Overrun : +X

0502 Soft Limit Overrun :-Z
0503 Soft Limit Overrun : +Z
0504 Soft Limit Overrun : -Y
0505 Soft Limit Overreach : +Y
0506 Soft Limit Overrun : -4TH
0507 Soft Limit Overrange : +4TH
0508 Soft Limit Overrun : -Th5
0509 Soft Limit Overrun : +Th5

0510 Hard Limit Overrange : -X
0511 Hard Limit Overrange : +X
0512 Hard Limit Overrange :-Z
0513 Hard Limit Overrange : +Z
0514 Hard Limit Overreach : -Y
0515 Hard Limit Overrange : +Y
0516 Hard Limit Overrange : -4TH
0517 Hard Limit Overrun : +4TH
0518 Hard Limit Overreach : -Th5
0519 Hard Limit Overreach : +Th5

0740 Rigid Tapping Alarm : Out-of-tolerance
0741 Rigid Tapping Alarm : Out-of-tolerance
0742 Rigid Tapping Alarm : LSI Overflow

0751 The detected the first spindle alarm (AL-XX).
0754 spindle abnormal torque alarm

1001 The address of the relay or coil is not set
1002 The function instruction for the input code does not exist
1003 The feature directive COM/COME is not used correctly
1004 User ladder exceeds the maximum allowable number of rows or steps
1005 The feature directive END1 or END2 is not used correctly
1006 There is an illegal output in the network
1007 Hardware failure or system interrupt error causes the PLC to fail to communicate
1008 The function directive is not properly connected
1009 The network horizontal line is not connected
1010 A power outage while editing a ladder causes loss in the editing network

1011 Address data was not entered correctly
1012 The input symbol is undefined or the input address is out of range
1013 specifies an illegal character or data out of range
1014 CTR address duplicates

1015 Feature Directive JMP/LBL is not handled correctly or exceeds capacity
1016 The network structure is incomplete
1017 Appears Network Structures That Are Not Currently Supported
1019 TMR address duplicates
1020 Parameter is missing from the function directive

1021 The PLC performs a timeout and the system automatically stops the PLC
1022 The function directive name is missing
1023 The address or constant of the function directive parameter is out of range

1024 There are unnecessary relays or coils
1025 Function Instructions Are Not Output Correctly
1026 The number of network connection lines is out of support
1027 The same output address is used in another place
1028 Ladder File Format Error
1029 The ladder file in use is missing
1030 There is an incorrect vertical line in the network

1031 The user data area is full , reduce the COD instruction datasheet capacity
1032 The first stage of the ladder diagram is too large to be executed in time
1033 SFT instruction exceeded maximum allowable usage
1034 The functional directive DIFU/DIFD is not used correctly
1035 The currently opened ladder file conversion was unsuccessful
1036 PLC abnormal stop alarm
1037 The open ladder diagram of is inconsistent with the data parameter setting ladder diagram
1039 Instructions or networks are not executable
1040 The function directive CALL/SP/SPE is not used correctly

1041 The horizontal conduction line is connected in parallel with the node network

1042 The PLC system parameter file is not loaded

2、PLC alarm

A000.0 Tool change time is too long

Alarm reason: When performing the tool change, the knife number was not found in the total time of the tool change no. 78;

Processing: Press [Reset Key] to cancel the alarm and check whether the knife position signal is normal.

A000.1 When the tool change is complete, the tool holder is not in place to alarm

Alarm reason: When the tool change is completed, the tool holder feedback signal is inconsistent with the tool number of the tool change instruction;

Process: Press [Reset Key] to cancel the alarm and change the knife again.

A000.2 Tool change did not complete alarm

Alarm reason: there is an alarm or reset during the tool change;

Process: Press [Reset Key] twice to cancel the alarm and change the tool again.

A000.3 The tool holder lock signal was not received

Alarm reason: when performing tool change locking, the lock signal was not detected within the specified time; (Ordinary knife holder time reference: No. 83).

Process: Press [Reset Key] to cancel the alarm and change the knife again.

A000.5 Before the system was powered off, the tool change error occurred

Alarm reason: when the tool change is not completed normally, the system is powered off;

Process: Press [Reset Key] to cancel the alarm and change the knife again.

A000.6 The pre-indexing of the AK31 tool change was not in place

Alarm reason: Select AK31 tool holder, perform the tool change, and the pre-indexing is not detected in place to release within 4 seconds;

Processing: Press [Reset Key] to cancel the alarm, check the pre-indexing sensor, and change the tool again.

A000.7 Turret locking is not loosened

Alarm reason: Select Liuxin hydraulic tool holder, perform the tool change, and the Sensor F signal is not detected within 4 seconds;

Process: Press [Reset Key] to cancel the alarm, check the Sensor F sensor, and change the tool again.

A001.0 Tailstock function is invalid and M10 and M11 codes cannot be executed

Alarm reason: When the tailstock function is not turned on, the M10/M11 code is executed;

Process: Press [Reset Key] to cancel the alarm.

A001.1 The spindle is rotating, and the tail shaft cannot be retracted

Alarm reason: the tailstock function and the spindle function are interlocked, and the tail shaft is reversed when the spindle is rotated;

Process: Press [Reset Key] to cancel the alarm.

A001.3 Tailstock advance is not detected and the spindle cannot be rotated

Alarm reason: the tailstock function and the spindle function are interlocked, and the tail axis is not forward to perform spindle rotation;

Process: Press [Reset Key] to cancel the alarm.

A001.4 A or B can have up to 8 knives

Alarm reason: Ordinary knife holder supports up to 8 knives, and alarm when the total number of knife positions of No. 84 is set to greater than 8;

Processing: Set the correct tool level.

A001.5 End of service life of the tool

Alarm reason: the service life of the tool in the tool set is reached;

Process: Press [Reset Key] to cancel the alarm.

A001.6 The opening time of the external loop is too long (connecting the normal opening point) K20.0

Alarm reason: the external loop starts a direct pass;

Processing: Check that the wiring is correct and the input signal level selection is correct K20.0.

A001.7 External pause open time is too long (normal closure point) K20.1

Alarm reason: external suspension in a direct pass;

Processing: Check that the wiring is correct and the input signal level selection is correct K20.1.

The A002.0 guard door is not closed and does not allow automatic operation

Alarm reason: in automatic mode, the protective door is not closed to start the program;

Processing: Close the guard door, if the guard door is closed, check whether the guard door signal input is normal.

A002.2 The number of processed parts completed the alarm

Alarm reason: alarm when the number of processed parts reaches the set number of parts;

Processing: Press [Reset Key] to cancel the alarm, and at the same time, you need to know the number of processed parts.

A002.3 When the spindle is rotating, the chuck must not be released

Alarm reason: chuck function and spindle function interlock, when the spindle rotates the chuck release;

Process: Press [Reset Key] to cancel the alarm.

A002.4 When the spindle is rotating, the clamping in place signal is invalid

Alarm reason: chuck function and spindle function interlock, detect chuck in place signal; When the spindle is rotating, no chuck clamping in place signal is received

Process: Press [Reset Key] to cancel the alarm.

A002.5 When the chuck clamp in place signal is invalid, the spindle must not be activated

Alarm reason: chuck function and spindle function interlock, detect chuck in place signal; Did not receive the chuck clamping in place signal, rotate the spindle;

Process: Press [Reset Key] to cancel the alarm.

A002.6 The chuck is released and the spindle must not be started

Alarm reason: chuck function and spindle function interlock; The chuck is not clamped, the spindle is rotated;

Process: Press [Reset Key] to cancel the alarm.

A003.0 Chuck function is invalid and the M12/M13 code cannot be executed

Alarm reason: when the chuck function is not turned on, the M12/M13 code is executed;

Process: Press [Reset Key] to cancel the alarm.

A003.1 Chuck clamp/release in place signal was not detected

Alarm reason: When detecting the chuck loosening clamping signal, when performing chuck release or clamping, the arrival signal was not detected within T043 time

Process: Press [Reset Key] to cancel the alarm.

A003.2 The target knife number was not found in Liuxin Turret

Alarm reason: After the turret is rotated, the alarm is called when the knife number is not found within 8 seconds

Process: Press [Reset Key] to cancel the alarm.

A003.3 Liuxin Turret did not receive a cutterhead stop and lock start signal SensorE

Alarm reason: After the turret rotates, the SensorE signal is not signaled within 8 seconds;

Process: Press [Reset Key] to cancel the alarm.

A003.7 The total number of common tool holders is greater than 4, and the external feed rate (address multiplexing) cannot be connected

Alarm reason: Ordinary tool holder T05-T08 signal and external feed rate address multiplexing

Processing: Note the selection of use

A004.1 The spindle analog voltage control state is not currently available and the spindle jog function cannot be performed

Alarm reason: When the spindle is controlled by the spindle switch, the spindle jog function is used

Process: Press [Reset Key] to cancel the alarm.

A004.2 M03, M04 code specified error

Alarm reason: The opposite instruction was executed in the forward or reverse rotation of the 1st spindle

Process: Press [Reset Key] to cancel the alarm.

A004.3 M63, M64 code specified error

Cause of alarm: The opposite instruction was executed in the forward or reverse rotation of the 2nd spindle

Process: Press [Reset Key] to cancel the alarm.

A004.4 Spindle shifting time is too long

Alarm reason: When executing the M41-M44 shift instruction, the shift was not completed within the T045 time

Process: Press [Reset Key] to cancel the alarm.

A004.5 Spindle speed/position switching time is too long

Alarm reason: when the spindle switches position or switches the speed mode, the switching completion signal is not received within 8 seconds;

Process: Press [Reset Key] to cancel the alarm.

A004.6 Spindle auto shift function is not turned on and the M41-M44 cannot be performed

Alarm reason: When the spindle automatic shift function is invalid, the M41-M44 shift instruction is executed;

Process: Press [Reset Key] to cancel the alarm.

A004.7 The AK31 pre-indexing did not loosen at the start of the tool change

Alarm reason: Select AK31 tool holder, perform the tool change, and still sense it when the pre-indexing is in place at the beginning;

Processing: Press [Reset Key] to cancel the alarm, check the pre-indexing sensor, and change the tool again.

A005.3 Clamping of the spindle is not allowed when the spindle is rotating or feeding

Alarm reason: spindle loosening state, spindle clamping when selecting the first spindle; The first spindle rotates, or the 5th axis (C axis) is moved when the first spindle position mode is moved; When the second spindle is selected for spindle clamping; The second spindle rotates, or the second spindle position mode moves the 4th axis (A axis) to perform the spindle clamping alarm;

Process: Press [Reset Key] to cancel the alarm.

A005.4 The spindle is clamped and does not allow rotation or feed

Alarm reason: spindle clamping state, spindle clamping when the first spindle is selected; First spindle rotation, or movement of axis 5 (C axis) alarm when the first spindle position mode ; When the second spindle is selected for spindle clamping; Alarm when the second spindle is rotating, or moving the 4th axis (A axis) when the second spindle is in position mode;

Process: Press [Reset Key] to cancel the alarm.

A006.0 The three-digit switch is in a paused state and the program cannot start

Alarm reason: When the three-digit switch function is valid, the three-digit switch starts the program in the program pause state

Process: Press [Reset Key] to cancel the alarm.

A006.2 M50, M51 code specified error

Alarm reason: The opposite command was executed in the forward or reverse rotation of the Y-axis power head

Process: Press [Reset Key] to cancel the alarm.

A006.3 M53, M54 code specified error

Alarm reason: The opposite command was executed in the forward or reverse rotation of the 4-axis power head

Process: Press [Reset Key] to cancel the alarm.

A006.4 The three-position switch is in the paused state and must not start the spindle

Alarm reason: When the three-digit switch function is effective, the three-position switch starts the spindle rotation in the spindle stop state

Process: Press [Reset Key] to cancel the alarm.

A006.5 Three-position switch 1\2 is on at the same time, and the wiring is wrong

Alarm reason: When the three-digit switch function is effective, the three-digit switch 1\2 simultaneously turns on the alarm

Processing: Confirm that the wiring is correct.

A008.3 The power head cannot be rotated at the same time as the spindle

Alarm reason: When the spindle and the power head cannot be rotated at the same time, the spindle and the power head are rotated together

Process: Press [Reset Key] to cancel the alarm.

A008.4 The Y 4-axis power head cannot be rotated at the same time

Alarm reason: Alarm when the Y and 4-axis power heads rotate at the same time

Process: Press [Reset Key] to cancel the alarm.

A020.0 Guard door not closed warning

Alarm reason: The guard door was not closed when the guard door function was opened

Handling: Close the protective door

A020.1 Insufficient lubricating oil warning

Alarm reason: insufficient lubricating oil

Treatment: Add lubricating oil

A020.2 Insufficient barometric pressure warning

Alarm reason: insufficient air pressure (warns that the program will pause when the air pressure is insufficient)

Treatment: Wait for sufficient air pressure and the warning is canceled

A020.3 Knife holder not locking warning

Alarm reason: The system did not receive the tool holder lock signal without tool change. (This warning is invalid when using a row knife)

Processing: Manual tool change or tool change in input mode to complete the tool change

Appendix II Supplemental Notes to Macro Programs

(2versions above 021) have

1. Statement macro code (macro B).

1.1 Arithmetic and logical operations

Arithmetic and logical operations

function	Expression structure	remark
Define or assign a value	#i=#j	
addition subtraction multiplication division	#i=#j+#k #i=#j-#k #i=#j*#k #i=#j/#k	
or and XOR	#i=#j OR #K #i=#j AND #K #i=#j XOR #K	Logical operations are performed as binary numbers one-bit
square root absolute value rounding Rounding on Round down Napierian logarithm Exponential function	#i=SQRT[#j] #i=ABS[#j] #i=ROUND[#j] #i=FUP[#j] #i=FIX[#j] #i=LN[#j] #i=EXP[#j]	
sine arcsine cosine inverse cosine tangent Anyway cut	#i=SIN[#j] #i=ASIN[#j] #i=COS[#j] #i=ACOS[#j] #i=TAN[#j] #i=ATAN[#j]/[#k]	The units of the angle are specified in degrees, such as 90°30' in 90.5 degrees
Switch from BCD to BIN Switch from BIN to BCD	#i=BIN[#j] #i=BCD[#j]	Used for conversion with PMC signals

Instructions

1、Angle units

The angular units of the functions SIN, COS, ASIN, ACOS, TAN and ATAN are degrees (°). For example, 90 ° 30 ′ should be expressed as 90.5 ° (degrees).

2、Arcsine #i=ASIN[#j]

ｉ、The resulting output range is as follows,

When the no180#7 NAT bit is set to 1, 90°~ 270°;

When the no180#7 NAT bit is set to 0, -90°~ 90°;

ｉｉ、When #j is out of the range of -1 to 1, a P/S alarm is issued.

ｉｉｉ、Constants can replace the variable #j.

3、Inverse #i=ACOS[#j]

ｉ、The result output range is from 180° to 0°;

ｉｉ、P/S alarm is issued when #j is out of the range of -1 to 1;

ｉｉｉ、Constants can override the variable #j.

4、Anyway cut #i=ATAN[#j]/[#k]

Specify the length of the two sides and separate them with a slash '/'.

ｉ、The value range is as follows,

When the no180#7 NAT bit is set to 1: 90°~270°;

[e.g. # 1=225°; when #1=ATAN[-1]/[-1] is specified;

When the no180#7 NAT bit is set to 0- -90°~90°;

[e.g. # 1=45.0° when #1=ATAN[-1]/[-1] is specified;

ｉｉ、Constants can replace the variable #j.

5、Natural logarithm #i=LN[#j]

a constant can be substituted for the variable #j.

6、Exponential function #i=EXP[#j]

ｉ、Constants can replace the variable #j.

7、ROUND rounding function

When the arithmetic or logical operation code IF or WHILE contains a ROUND function, the ROUND function is rounded at the 1st decimal place.

For example:

When #1=ROUND[#2] is executed, at which point #2=1.2345 is executed, and the value of variable 1 is 1.0.

8、Round up and round down

When CNC processes numerical operations, if the absolute value of the integer generated after the operation is greater than the absolute value of the original number, it is called rounding; If it is less than the absolute value of the original number, it is called down-rounding. Care should be taken with regard to the handling of negative numbers.

For example:

Suppose #1=1.2 and #2=-1.2

When #3=FUP[#1] is executed, 2.0 is assigned to #3.

When #3=FIX[#1] is executed, 1.0 is assigned to #3.

When #3=FUP[#2] is executed, -2.0 is assigned to #3.

When #3=FIX[#2] is executed, -1.0 is assigned to #3.

1.2 Transfer and Circulation

In the program, the use of GOTO statements and IF statements can change the flow direction of the control. There are three transfer and loop operations available.

1、GOTO statement (unconditional transfer).

2、Conditional control IF statements.

3、WHILE loop statement.

1）**Unconditional transfer (GOTO statement)**

Transfer to the program segment with the sequential number n. Alarm when specifying a sequence number other than 1 to 99999, you can use an expression to specify the sequence number.

Format: GOTOn; n: Sequence number (1~99999).

Example: GOTO1;

GOTO#101;

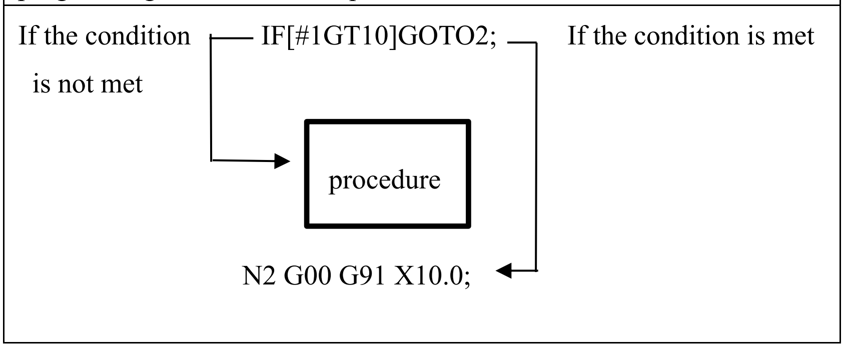
2） **Conditional control (IF statement)**

GOTO format: IF [conditional expression] GOTOn;

If the specified conditional expression holds, the program segment is transferred to the sequence number n; If the specified conditional expression does not hold, the next program segment is executed sequentially.

Example:

If the value of variable #1 is greater than 10, transfer to the program segment with the sequence number N2.



THEN format: IF[conditional expression]THEN< macro program statement >;

If the conditional expression holds, only one statement can be executed after the SEE statement.

Example: IF[#1EQ#2]THEN#3=0;

If the value of #1 is equal to the value of #2, 0 is assigned to variable #3; if not equal, the assignment statement after SEE is executed down.

Conditional expression: A conditional expression must include a conditional operator, which can be flanked by a variable, constant, or expression, and enclosed by parentheses '["]'.

Conditional operators: This system can use the conditional operators listed in the following table.

Conditional operators	meaning
EQ or ==	Equal to (=)
NE or <>	Not equal to (≠).
GT or >	Greater than (>)
GE or >=	Greater than or equal to (≥)
LT or <	Less than (<)
LE or <=	Less than or equal to (≤)

For example: IF[3<>2]GOTO2; The meaning is: if 3 is not equal to 2, then jump to the N2 program segment;

IF[#101>=7.22]THEN#101=SIN30； The meaning is: if #101 is greater than or equal to 7.22, the assignment operation after SEE is performed. That is, the sinusoidal value of 30 degrees is assigned to the variable #101.

Typical procedure: The following program calculates the sum of integers 1 to 10.

```
O9500
#1=0; ...... and initialized to 0
#2=1; ...... The initial value of the added number is 1
N1 IF[#2 GT 10]GOTO2;  ...... Transfer to N2 when the added
number is greater than 10
#1=#1+#2; ...... Calculate the sum of two numbers
#2=#2+1; ...... Is added plus 1
GOTO1;  ...... Unconditional jump to program segment N1
N2 M30;  ...... The program ends
```

3）**Loop (WHILE statement)**

Specify a conditional expression after WHILE, and when the specified condition is true, execute the program segment between DO and END; Otherwise, jump to the program segment after END.

Example:

WHILE [conditional expression] DO m (m=1,2,3)."

Condition not met Conditional satisfies (true).

(result is false) [Procedure].

END m

Description: When the specified condition is true, execute the program segment between DO and END; Otherwise, the program segment after END is executed instead. The label after DO and the label after END should be the same, and the label value can be 1, 2 or 3. If you use a value other than 1, 2, 3, you will be alarmed.

Nested: DO, the label (1~3) in the END loop can be used as many times as needed. However, an alarm is generated when there is a cross-repeat loop in the program.

1. Labels (1 to 3) can be used as many times as required

WHILE[...] DO1

procedu

END1;

⋮

WHILE[...] DO1;

proced

END1;

2. **The range of DO cannot be crossed**

WHILE[...] DO1;

procedu

WHILE[...] DO2;

⋮

END1

proced

END2;

3. **DO loops can be nested 3 layers**

WHILE[...] DO1;

⋮

WHILE[...] DO2;

⋮

WHILE[...] DO3;

END3;

END2;

END1;

4. **Controls can go to the outside of the loop**

WHILE[...] DO1;

IF[...] GOTOn;

END1;

Nn;

5. **Transfers cannot enter the circulating zone**

IF[...] GOTOn;

WHILE[...] DO1;

Nn;

END1;

1.3 Macro alarm

Format: ERR_;

Range 0 to 200;

Alarm no. 3000+n;

For example: ERR01 or ERR1;

The system alarm number is 3001.

1.4 Special instructions (permutation of variable numbers with variables).

Format: "#"+"9"+permutation variable number

Examples: #100=205, #205=500;

The X#9100 and X500 codes function identically;

The X-#9100 and X-500 codes function identically;

It can be understood as #9_ _ _, followed by 3 digits as a common variable number, ranging from 100 to 199,5 00～999；

#9100 represents the macro variable number represented by the value within #100;

For example, #100=510, #510=101;

Write #9100 at this point, #9100=#510=101;

Appendix III: Custom alarm content writing (2021 and above versions have it).

Parameter: Bit parameter 180.4 Read program macro variable/alarm comment (0: Invalid 1: Valid)

Macro A alarm

G65H99P_;//[ERR_, the system displays the alarm content] (within 25 words).

Range 0-200;

Macro B alarm

ERR_;//[ERR_, the system displays the alarm content] (within 25 words).

Range 0-200;

For example:

ERR1;//[ERR1, quantity arrived];

The system displays the macro program alarm 3001 number arrived;

Note: At the same time, you can write the value of the system macro variable #3000 for alarm, the range is also 0-200, and the alarm number is also consistent 3000-3200; The writing method has G65 H01 P#3000 Q_, or it can be #3000= _;

For example: G65 H01 P#3000 Q0;

#3000=0;

G65H99P0;

ERR0; These 4 compilations have the same effect.

Modify the system to read the program alarm comments, and then display the alarm content and report out in the system file WARN2;

Program macro variable annotations can be read and saved in system files MacroVar;

Stipulate the range of writable alarms;

0650-0657； M89K0-K7 detection

3000-3200； Macro program alarm p0-P200

；

Write M89K0-K7 Alarm Specification Format:

M89 K0-K7 J_ ;//[ERR650-657,M89K0-K7 Detection Alarm];

M89K0-K7 detection alarm; Alarm display content, while recording in the file WARN, and then when there is the same alarm number if no alarm content is written, directly read the stored alarm internal display.

For example:

M89K3J5000;//[ERR653, M89K3 top material not in place].

or

[ERR653, M89K3 top material is not in place].

M89K3J5000;

As long as the program reads [ERR653, M89K3 top material is not in place], the alarm content is saved in the system file 653 alarm;

When the next time there is no write alarm display, the same code is instructed, and the same alarm display is written before;

For example:

M89K3J5000;//

At this point, the alarm will still show that the M89K3 top material is not in place;

Write a macro program alarm specification format

Macro A:

G65H99P_;//[ERR_, alarm display content];

Macro B:

ERR_;//[ERR_, alarm display content];

Alarm display content, while logged in the file WARN2, and then there is the same alarm number when there is no write alarm content, directly read the stored alarm internal display.

For example:

G65H99P0;//[ERR0, processing quantity arrived].

or

[ERR0, processing quantity arrives].

G65H99P0;

As long as the program reads [ERR0, the processing quantity arrives], the alarm content is saved in the system file 3000 alarm;

When the next time there is no write alarm display, the same code is instructed, and the same alarm display is written before;

For example:

G65H99P0;

or

ERR0;

At this point, the alarm will still display the macro alarm 3000 processing quantity arrived;

The program reads macro variable comments saved in the system file MacroVar:

Modifiable comments:

#100-#199;

#500-#899; (#900-#999 Reserved System for internal use);

Modify the annotation specification format;

//[#--, display content] (within 20 characters).

//[#--, show content].

The display will be saved in **MacroVar and can be viewed in public variables;**

Macro variable annotations are always saved without changing the same macro variable annotation.

Appendix 4: Automatic chip breaking function (software version 20201016 or more).

Note: When using the chip breaking function, the cutting acceleration and deceleration must be a straight line type of pre-acceleration and deceleration;

Bit parameter 168.2=0/168.3=0; The acceleration and deceleration time was changed to the acceleration and deceleration number parameter 313;

The larger the number parameter 313, the faster the acceleration and deceleration; The smaller the hour, the smaller the acceleration and deceleration; Default 100

1. Function description

Reduce the need for chip removal during roughing, and improve the quality of the processing surface, adding chip breaking function.

Second, the relevant parameters

The newly added parameters are as follows:

213 Chip breaker mode (G04.9) default chip breakage length

[data type] floating-point

[Default] 1.0

[Data description] Chip break mode, the default chip break length. Units of millimeters

Default chip break pause time for 214 chip breaking mode (G04.9).

[Data Type] font

[Default] 100

[Data description] Chip breaking mode, the default chip break pause time. In milliseconds

3. Detailed description of functions

When the chip breaking function is turned on, after interpolating a certain distance (chip break length), the interpolation shaft will pause for a period of time (chip break pause time). Since the spindle does not stop rotating at this time, chip breaking can be carried out.

The format in which chip breaking is enabled is:

G4.9 P [Chip Break Pause Time] Q [Chip Break Length] H [Chip Break Speed];

H [Chip Breaking Speed] Chip breaker at default speed F5 when not **entered.**

To enable chip breaking with the following format, you can use parameters as default values:

G4.9 P [Chip Break Pause Time]; Chip breakage length using parameter 213

G4.9 Q [Chip Breaking Length] ; Chip break suspension time using parameter 214

G4.9 P0 ; Use both parameters as the default value

A PS alarm is triggered when chip breaks are paused for a period of time, or when chip breakage length is less than or equal to 0.

When the chip breaking function is active, when the chip breaking function is turned on again by command, the current accumulated interpolation motion length is reset, and the new data is used for chip breaking in the subsequent motion segment.

The format for turning off chip breaking is:

G4.9 ;

Example: O0001;

T0101;

M3S1200;

M08;

G0X45;

G0Z3;

G4.9 P100 Q1;/Roughing opens chip breaking function, chip breaking length 1mm, chip breaking time 100ms

G71U1R0.5;

G71P10Q20U0.2;

N10G0X0;

G1Z0;

G03X4.09Z-6.55R3.59;

G02X17.45Z-22.72R19.93;

G03X19.07Z-34.46R7.03;

G03X22.78Z-50.31R8.41;

N20G01Z-56;

G4.9;//Finishing closes chip breaking function

M3S1500;

G70P10Q20;

G0Z60;

M30;

%

Appendix V: Polygonal cutting function (only available in the version with turning function).

The car side spindle control can be selected from analog drive control and pulse volume position mode control.

Analog control is used for the selection of spindles using frequency converter control; For spindles using servo spindles, it is recommended to use pulse position mode control (if the spindle inverter control is selected, the analog quantity is selected, the spindle does not need to switch position mode), and the spindle switches position mode at the beginning of the car.

Add parameters:

Position parameter 178.2 Feed shaft when the simulated spindle stops (0: No 1: Yes) stops the position before starting Change to 1 when the car stops after the cutterhead stops at the starting position;

Bit reference 178 3 car side spindle control (0: analog 1: pulse).

Position parameter 178.7 Square axle selection (0:3TH axis 1:4TH axis) Default 0 3TH axis follows the car side

Position reference 179.1 Square spindle selection (0: 5th axis 1: 4th axis).

Bit reference 179 2 Vehicle type selection (0: Encoder Car 1: Pulse Ratio Car) Select 0 Encoder Vehicle Must Have Encoder Feedback Speed, Otherwise Alarm; When selecting 1, it is possible to drive without an encoder, but the change angle must be stopped, and then the angle of rotation is rotated through the spindle position mode (the spindle needs to switch position mode);

The square axle must be set to a rotating shaft when following the shaft selection;

At the same time, the gear ratio emitted by the car side spindle pulse is shared with the gear ratio setting parameters issued by the power head.

Code format:

G68 J_ L_ S_ R_ ; The car code opens the car function

J_ (spindle) number of car aspects (how many sides of the car); When J is positive, the spindle is forward, and when it is negative, the spindle is reversed;

L_ (square axle) number of car blades; When L is positive, the square axle is rotating, and when it is negative, the square axle is reversed;

S_ Spindle speed; After the G68 code ends, the spindle rotates with the S spindle

R_ The angle of the vehicle side (the angle G 68 can be switched in the middleR_ of the car side, Note: the switching angle is invalid when the pulse ratio is 179.2 = 1).

G69 ; The car side code ends, the S speed pulse stops (the code must wait for the pulse to be issued to stop the completion).

Default G69 status when the system is powered on; Restore the G69 state when the system emergency stop alarm is reset;

The car side follower shaft can be set to a 3TH axis or a 4th axis;

System number parameter 345 3TH shaft front acceleration and deceleration constant (start 3TH shaft power head acceleration and deceleration. M50/M51/M52）

System number reference 346 4TH shaft front acceleration and deceleration constant (start 4TH shaft power head acceleration and deceleration M53/M54/M55).

If the spindle does not need to position the car, you can directly use the G68 to switch the position mode, and do not need to switch the position mode after the M14 positioning.

Encoder side 179.2 =0:

Analog vehicle control programming format:

M3S400； //Turn on spindle speed

G68J2L1R0;//Open car mode (spindle does not switch position mode).

....

G68 R_ ;//Angle can be switched halfway

....

G69;//Off car mode

Pulse position mode vehicle control programming format:

M5； Close the spindle

(M14);// Here, the spindle positioning switching position simulation can be added, so that the spindle enters the position mode in the same position in advance, or it can be omitted

G68J2L1S400R0;//Turn on car mode (spindle switching position mode).

... .

G68 R_ ;//Angle can be switched halfway

... .

G69;//Off car mode

Pulse ratio car side 179.2 = 1:

Pulse volume position mode vehicle control programming format:

M5； Close the spindle

M14;//Here you can add spindle positioning switching position simulation, so that the spindle enters the position mode in the same position in advance, and can be omitted without the need for a fixed position

G68J2L1S400;//Turn on car mode (spindle switching position mode).

... .

G69;//Off car mode

G0 C_;//Offset Angle, which requires stop mode when the offset angle is required

G68J2L1S400;//Turn on car mode (spindle switching position mode).

... .

G69;//Off car mode

Appendix VI: Description of G71/G72 Type II (version 2021 or above) has it

The programming format of the G 71/G72 Type II description is consistent with the G71/G72 Type I programming format;

The system adds two parameters to control the type II state;

Bit parameter 177.5 G71/G72 whether to use type II algorithm (0: yes 1: no);

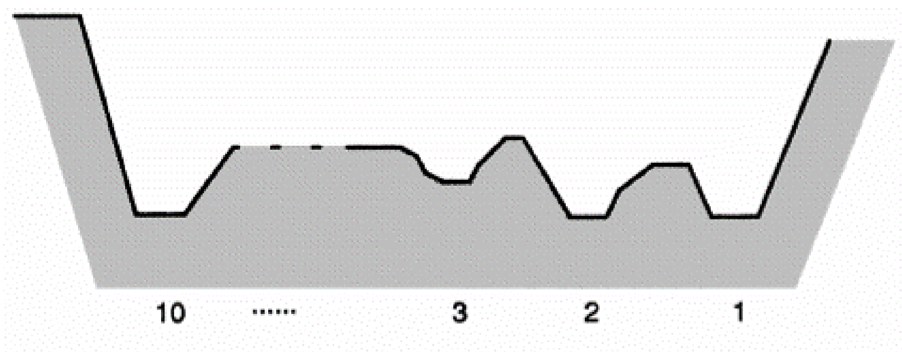
Note: When the G71/G72Type II algorithm is not used, the Type II workpiece program cannot be edited, otherwise the machining tool will be incorrect.

Bit parameter 177.6 G71/G72 type II. algorithm whether to run roughing profiles (0: no 1: yes);;

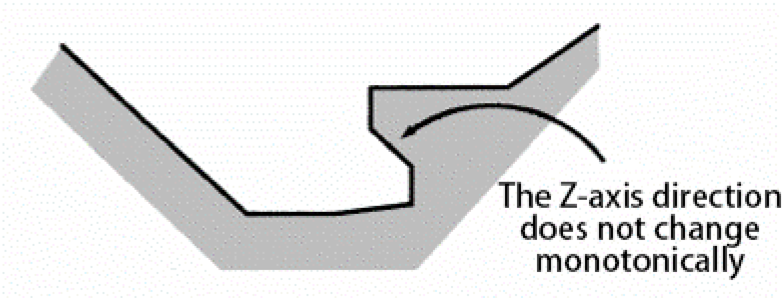
Note: Whether the G71/G72 type II algorithm runs roughing profiles (0:No 1:Yes);; When no is selected, do not walk along the contour again after roughing is completed; Select Yes to walk along the profile again after roughing is complete.

Type G71 II is different from Type I: the differences are as follows

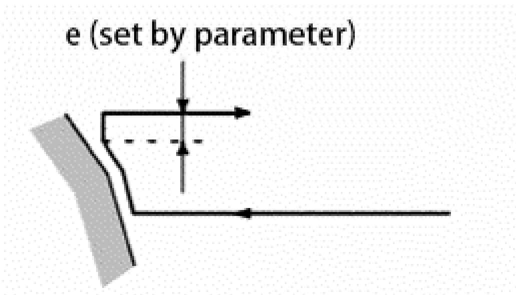
1) The X-axis profile does not have to be monotonically increasing or monotonically decreasing, and up to 10 grooves, illustrated as follows:



2) The contours along the Z axis must be monotonically increasing or monotonically decreasing, and the following contours cannot be machined:

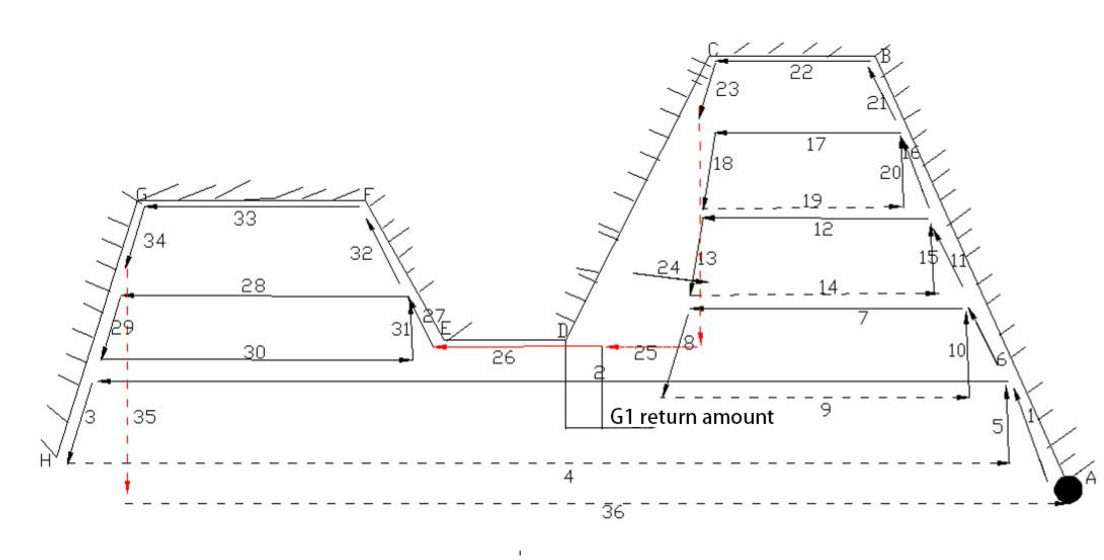


3) After turning, the tool should be retracted, and the amount of retracted should be specified by the R(e) parameter or the parameter number 52, which is shown as follows:



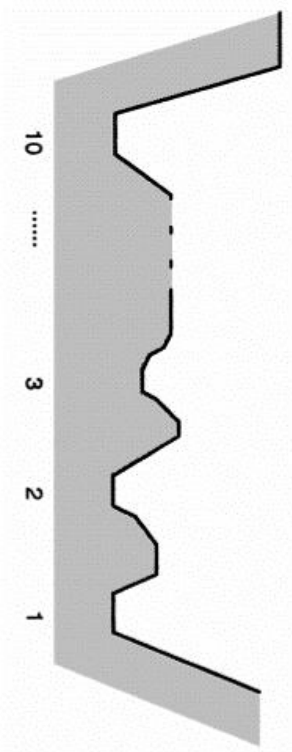
4) The finishing allowance can only specify the X direction, if the Z direction is specified, the entire processing trajectory will be offset, if the specification is preferably 0;

5) Code execution process: Rough car trajectory A-H

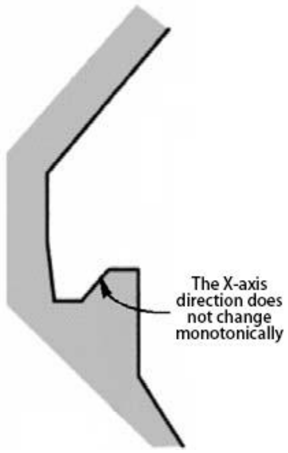


G72 Type II is different from Type I: the differences are as follows

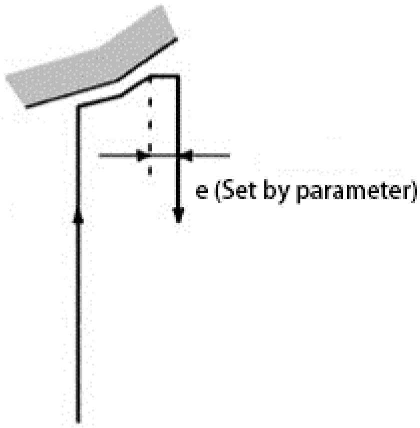
1) The profile of the Z-axis does not have to be monotonically increasing or monotonically decreasing, and up to 10 grooves, as follows:



2) The contours along the Z axis must be monotonically increasing or monotonically decreasing, and the following contours cannot be machined:

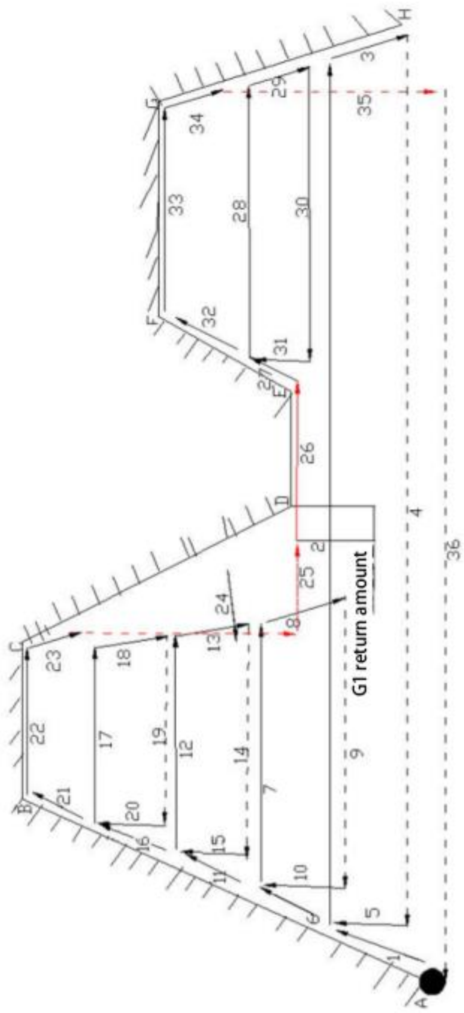


3) After turning, the tool should be retracted, and the amount of retracted should be specified by the R(e) parameter or the parameter number 52, which is shown as follows:



4) The finishing allowance can only specify the Z direction, if the X direction is specified, the entire processing trajectory will be offset, if the specification is preferably 0;

5) Code execution process: Rough car trajectory A-H



Appendix VII: Positive/side punching cycles (available in versions above 2 021).

1. G83/G87 positive/side drilling cycle

1.1 Front drilling cycle G83:

Instruction format 1: G83 X(U)_ Z(W)_ R_ P_ F_ K_ M_;

X_ hole absolute value position (U_ hole for the starting point X axis of the incremental value position);

Z_ absolute value position at the bottom of the hole (W_ position of the incremental value at the bottom of the hole for the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ Feed speed;

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Description: This format will be a one-time punch, no layered drilling.

Parameter 216 [G83/G87 Drilling Clearance] Default 2

Number parameter 217 [spindle clamping M code] default 20

Bit parameter 172.5 [Drilling/boring/rigid tapping/cycling spindle clamping automatic (0: No 1: Yes) is valid] Default 0

Note 1: Alarm when editing R is lower than the bottom of the hole position: the R point position is lower than the bottom position;

Note 2: Alarm when editing Z(W): The bottom plane of the hole is not specified in the drilling/boring cycle;

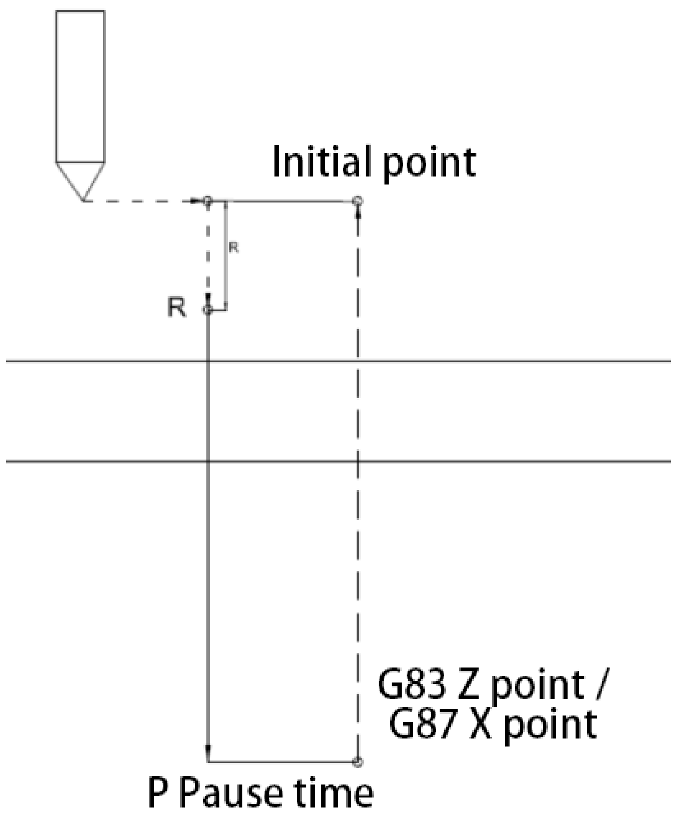
Note 3: When **the G83/G87/G85/G89 drilling cycle CS axis is automatically released (edited with the M20 command),** the CS axis command in the hole punching instruction G83/G87 is not moved. The spindle clamping instruction needs to wait for the instruction to complete before it is output; Instruction CS axis movement command needs to automatically release the spindle clamping,

G83 C100 Z-20 R5 F100 M20; // After M21 is executed, after C100 is completed, M20 is executed;

C140 M20; // First after the execution of M21 is completed, after the execution of C100 is completed, the execution of M20;

G80;

Note 4: When the open bit parameter 172.5=1 [drilling /boring/rigid tapping/cycling spindle clamping automatic (0:no 1:yes) is valid], when the C axis needs to be clamped and loosened in the loop code, the M 20 code can be controlled without editing the spindle clamping and loosening, The system automatically joins the system to execute the release codeM21 before **executing the C coordinates, waiting for the completion of the C coordinates, and executes the M20 clamping code after the coordinates are completed; If the parameter is also edited with M20** open**, M20 shall prevail.**



Instruction format 2: G83 X(U)_ Z(W)_ R_ P_ F_ K_ M_ Q_;

X_ hole absolute value position (U_ hole for the starting point X axis of the incremental value position);

Z_ absolute value position at the bottom of the hole (W_ position of the incremental value at the bottom of the hole for the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

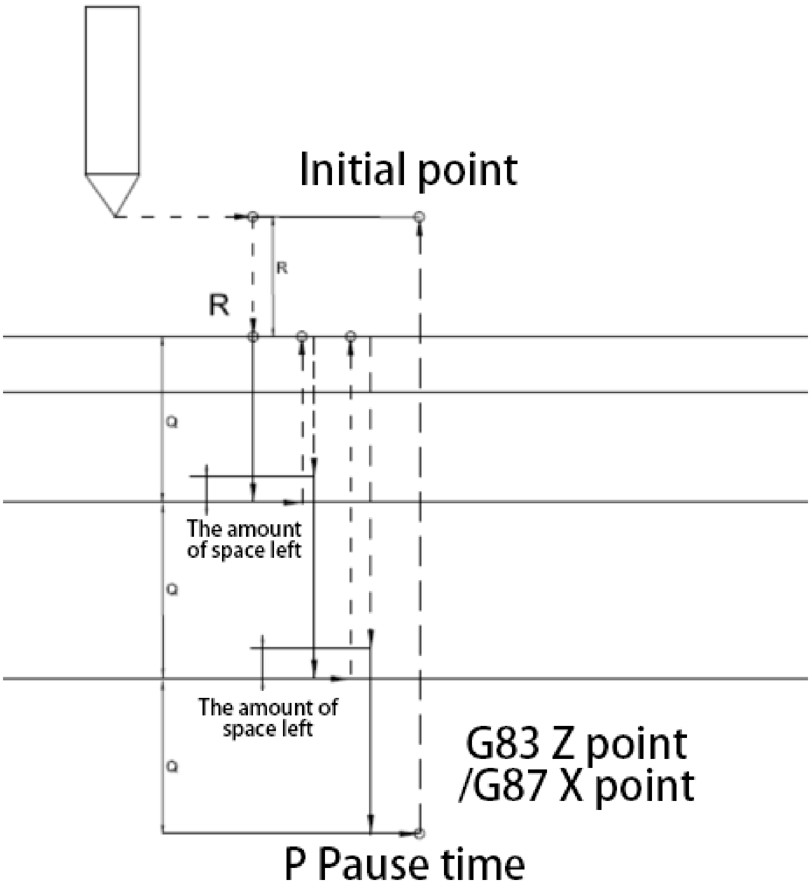
F_ Feed speed;

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Q_ depth of each cutting (regardless of positive and negative signs);

Description: This format will be divided into drill holes, return to the R point after each feed Q_, and then quickly feed to the blank position for drilling feed.



Instruction format 3: G83 X(U)_ Z(W)_ R_ P_ F_ K_ M_ Q_ D_;

X_ hole absolute value position (U_ hole for the starting point X axis of the incremental value position);

Z_ absolute value position at the bottom of the hole (W_ position of the incremental value at the bottom of the hole for the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ Feed speed;

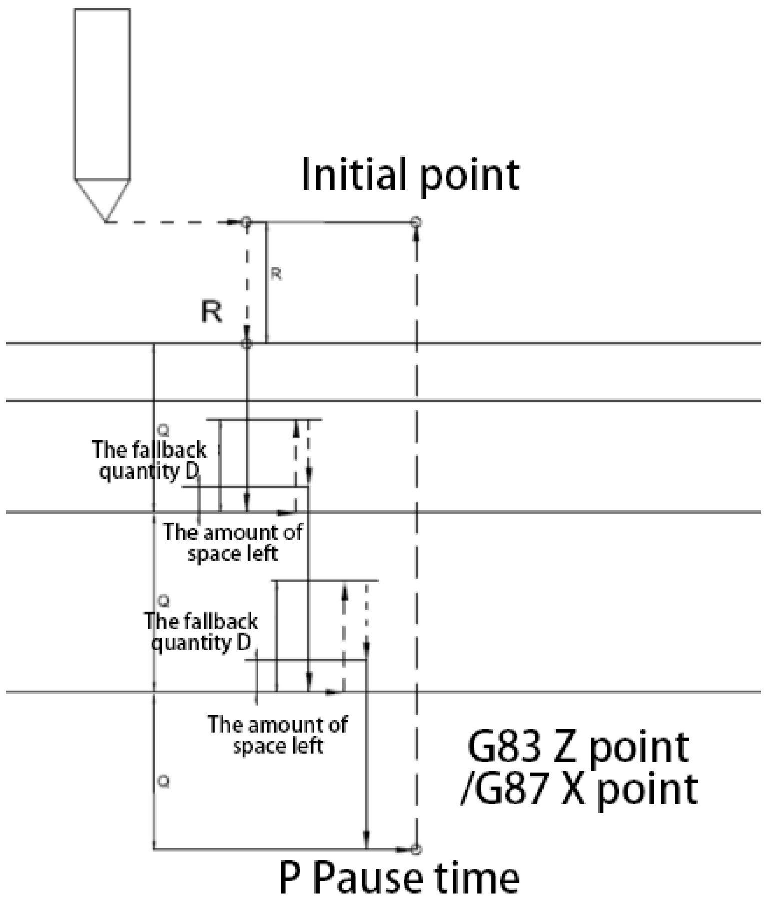
K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Q_ depth of each cutting (regardless of positive and negative signs);

D_ each fallback height (regardless of positive and negative signs);

Description: This format will be divided into drill holes, each feed Q_ after the fallback height, and then quickly feed to the blank position for drill feed.



1.2 Side drilling cycle G87:

Instruction format 1: G87 X(U)_ Z(W)_ R_ P_ F_ K_ M_;

X_ hole bottom absolute value position (U_ hole bottom for the R point increment value position);

Z_ hole absolute value position (W_ hole for the starting point Z axis of the incremental value position);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ Feed speed;

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Description: This format will be a one-time punch, no layered drilling.

Parameter 216 [G83/G87 Drilling Clearance] Default 2

Number parameter 217 [spindle clamping M code] default 20

Bit parameter 172.5 [Drilling/boring/rigid tapping/cycling spindle clamping automatic (0: No 1: Yes) is valid] Default 0

Note 1: Alarm when editing R is lower than the bottom of the hole position: the R point position is lower than the bottom position;

Note 2: Alarm when editing X(U): Drilling/boring cycle does not specify the bottom plane of the hole;

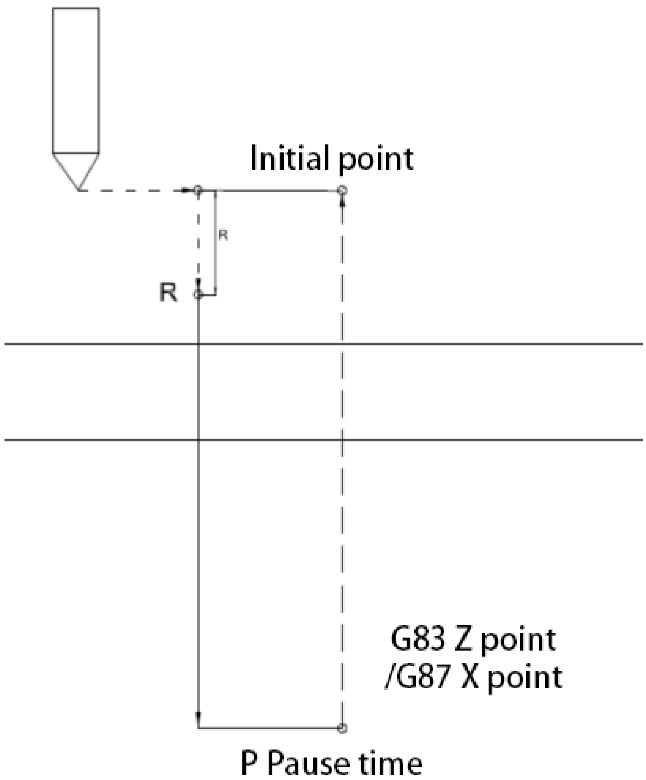
Note 3: When **the G83/G87/G85/G89 drilling cycleC-S axis is automatically released (edit the M20 command),** the CS axis command in the hole punching instruction G83/G87 is not moved. The spindle clamping instruction needs to wait for the instruction to complete before it is output; Instruction CS axis movement instruction needs to automatically loosen the spindle clamping;

G87 C100 X-20 R5 F100 M20; //After M21 is executed, after C100 is completed, M20 is executed;

C140 M20;   // First after the execution of M21 is completed, after the execution of C100 is completed, the execution of M20;

G80;

Note 4: When the open bit parameter 172.5=1 [drilling /boring/rigid tapping/cycling spindle clamping automatic (0:no 1:yes) is valid], when the C axis needs to be clamped and loosened in the loop code, the M 20 code can be controlled without editing the spindle clamping and loosening, The system automatically joins the system to execute the release codeM21 before **executing the C coordinates, waiting for the completion of the C coordinates, and executes the M20 clamping code after the coordinates are completed; If the parameter is also edited with M20** open**, M20 shall prevail.**



Instruction format 2: G87 X(U)_ Z(W)_ R_ P_ F_ K_ M_ Q_;

X_ hole bottom absolute value position (U_ hole bottom for the R point increment value position);

Z_ hole absolute value position (W_ hole for the starting point Z axis of the incremental value position);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

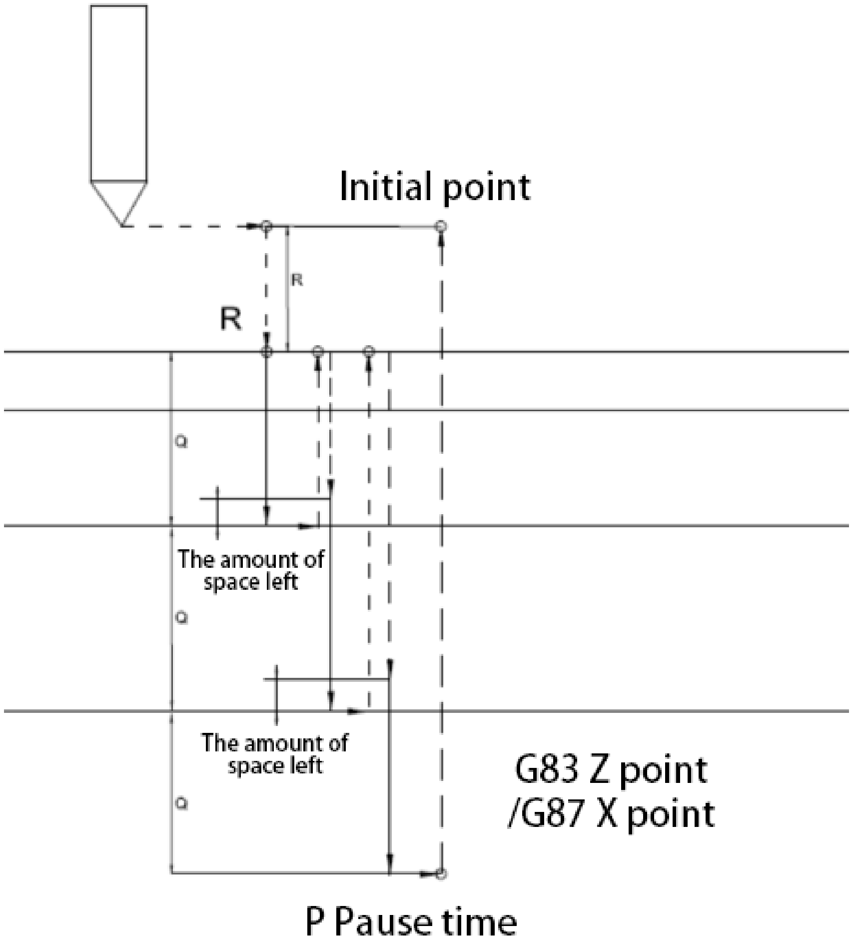
F_ Feed speed;

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Q_ depth of each cutting (regardless of positive and negative signs);

Description: This format will be divided into drill holes, return to the R point after each feed Q_, and then quickly feed to the blank position for drilling feed.



Instruction format 3: G87 X(U)_ Z(W)_ R_ P_ F_ K_ M_ Q_ D_;

X_ hole bottom absolute value position (U_ hole bottom for the R point increment value position);

Z_ hole absolute value position (W_ hole for the starting point Z axis of the incremental value position);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ Feed speed;

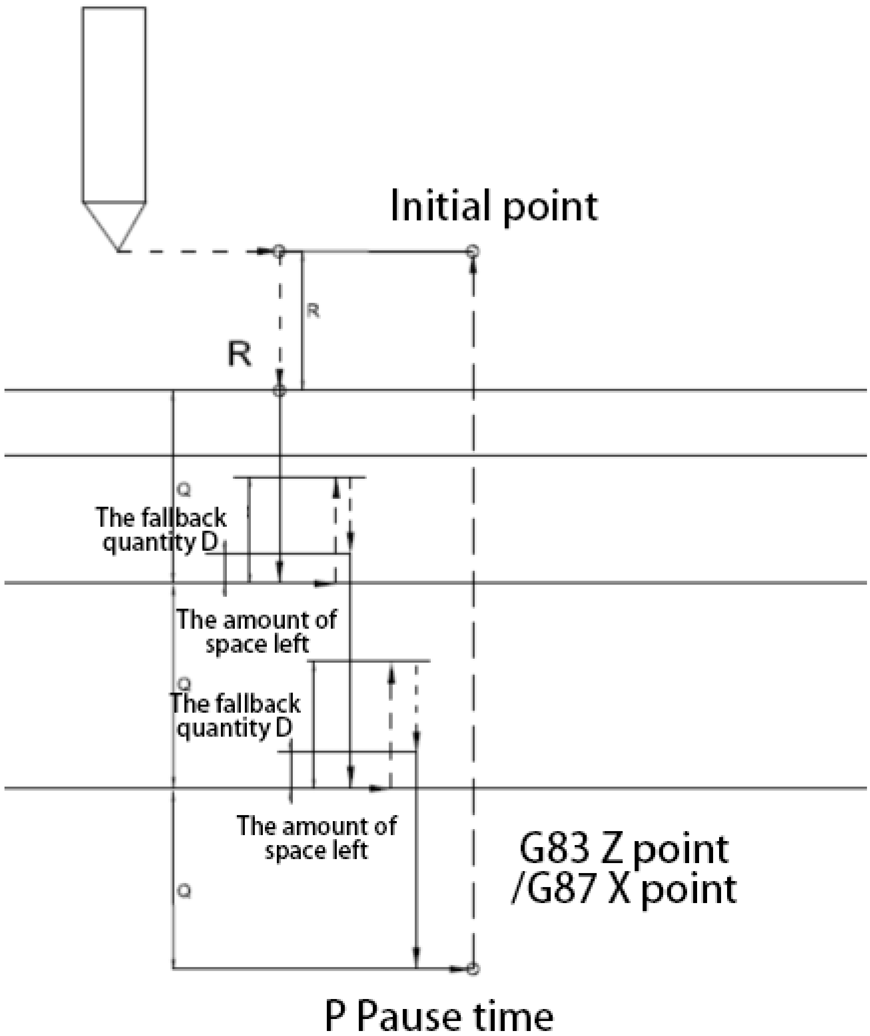
K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Q_ depth of each cutting (regardless of positive and negative signs);

D_ each fallback height (regardless of positive and negative signs);

Description: This format will be divided into drill holes, each feed Q_ after the fallback height, and then quickly feed to the blank position for drill feed.



2. G85/G89 positive/side boring cycle

2.1 Front boring cycle G85:

Instruction format 1: G85 X(U)_ Z(W)_ R_ P_ F_ K_ M_;

X_ hole absolute value position (U_ hole for the starting point X axis of the incremental value position);

Z_ absolute value position at the bottom of the hole (W_ position of the incremental value at the bottom of the hole for the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ Feed speed;

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Note: Boring cycle feed and retract are performed at feed F speed.

Parameter 216 [G83/G87 Drilling Clearance] Default 2

Number parameter 217 [spindle clamping M code] default 20

Bit parameter 172.5 [Drilling/boring/rigid tapping/cycling spindle clamping automatic (0: No 1: Yes) is valid] Default 0

Note 1: Alarm when editing R is lower than the bottom of the hole position: the R point position is lower than the bottom position;

Note 2: Alarm when editing Z(W): The bottom plane of the hole is not specified in the drilling/boring cycle;

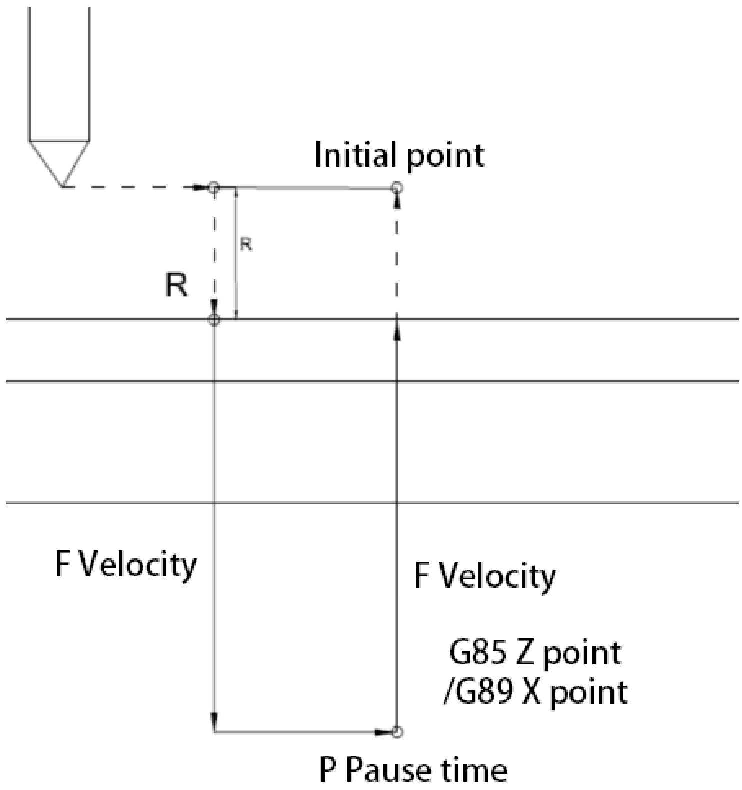
Note 3: When **the G83/G87/G85/G89 drilling cycle CS axis is automatically released (edit the M20 command),** the CS axis command in the hole punching instruction G85/G89 is not moved. The spindle clamping instruction needs to wait for the instruction to complete before it is output; Instruction CS axis movement command needs to automatically release the spindle clamping,

G85 C100 Z-20 R5 F100 M20; // After M21 is performed, after C100 is completed, M20 is performed;

C140 M20; // First after the execution of M21 is completed, after the execution of C100 is completed, the execution of M20;

G80;

Note 4: When the open bit parameter 172.5=1 [drilling /boring/rigid tapping/cycling spindle clamping automatic (0:no 1:yes) is valid], when the C axis needs to be clamped and loosened in the loop code, the M 20 code can be controlled without editing the spindle clamping and loosening, The system automatically joins the system to execute the release codeM21 before **executing the C coordinates, waiting for the completion of the C coordinates, and executes the M20 clamping code after the coordinates are completed; If the parameter is also edited with M20** open**, M20 shall prevail.**



2.2 Side boring cycle G89:

Instruction format 1: G89 X(U)_ Z(W)_ R_ P_ F_ K_ M_;

X_ hole bottom absolute value position (U_ hole bottom for the R point increment value position);

Z_ hole absolute value position (W_ hole for the starting point Z axis of the incremental value position);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ Feed speed;

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Note: Boring cycle feed and retract are performed at feed F speed.

Parameter 216 [G83/G87 Drilling Clearance] Default 2

Number parameter 217 [spindle clamping M code] default 20

Bit parameter 172.5 [Drilling/boring/rigid tapping/cycling spindle clamping automatic (0: No 1: Yes) is valid] Default 0

Note 1: Alarm when editing R is lower than the bottom of the hole position: the R point position is lower than the bottom position;

Note 2: Alarm when editing X(U): Drilling/boring cycle does not specify the bottom plane of the hole;

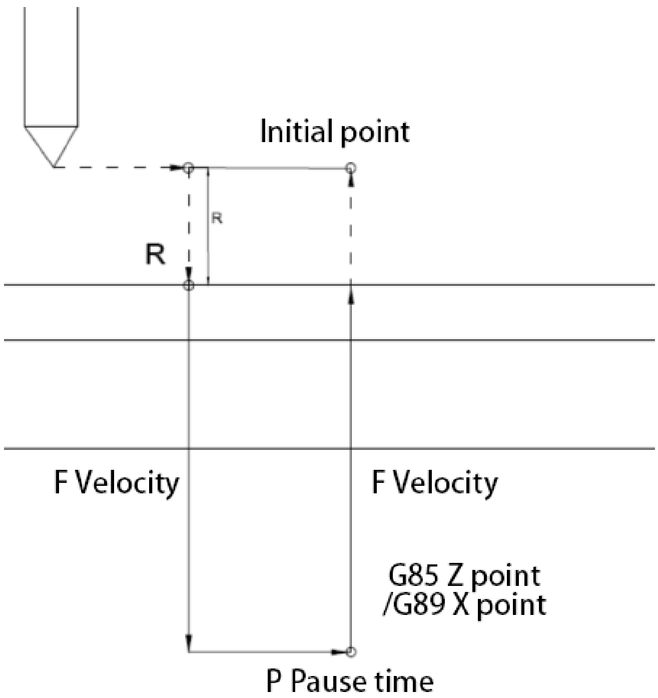
Note 3: When **the G83/G87/G85/G89 drilling cycleC-S axis is automatically released (edit the M20 command),** the CS axis command in the boring instruction G85/G89 is not moved, The spindle clamping instruction needs to wait for the instruction to complete before it is output; Instruction CS axis movement instruction needs to automatically loosen the spindle clamping;

G89 C100 X-20 R5 F100 M20; //First after M21 is executed, after C100 is completed, M20 is executed;

C140 M20; // First after the execution of M21 is completed, after the execution of C100 is completed, the execution of M20;

G80;

Note 4: When the open bit parameter 172.5=1 [drilling /boring/rigid tapping/cycling spindle clamping automatic (0:no 1:yes) is valid], when the C axis needs to be clamped and loosened in the loop code, the M 20 code can be controlled without editing the spindle clamping and loosening, The system automatically joins the system to execute the release codeM21 before **executing the C coordinates, waiting for the completion of the C coordinates, and executes the M20 clamping code after the coordinates are completed; If the parameter is also edited with M20** open**, M20 shall prevail.**



Appendix VIII: Positive/Side G84/G88 Tapping Cycle

(available in versions above 2 021).

1. Front tapping cycle G84:

Front Tapping Loop Command Format 1:
M29 S____ ; M29 open rigid tapping, S_ __ tapping speed // do not writeM29 instructions for flexible tapping
G84 X(U)_ Z(W)_ R_ P_ F/I_ K_ M_;

X_ hole absolute value position (U_ hole for the starting point X axis of the incremental value position);

Z_ absolute value position at the bottom of the hole (W_ position of the incremental value at the bottom of the hole for the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ metric tap pitch, I_ inch tooth count, **right hand is positive, left hand is negative;**

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Description: This format will be machined in one go.

The spindle control mode for bit reference 213.0 tapping is (0: followed by 1: servo).

When 213.0 chooses to follow, whether there is M29 instructions are flexible tapping, controlled by spindle forward and reverse; When selecting the servo, there is an M29 command for rigid tapping, and the tapping pulse port is selected by parameters

Bit reference 209.0 G84 rigid tapping whether to choose the Y axis as the spindle (0:No 1 Yes).

Bit reference 209.1 G84 rigid tapping whether to choose the fourth axis as the spindle (0:No 1 Yes).

Bit reference 209.2 G84 rigid tapping whether to choose the fifth axis as the spindle (0:No 1 yes).

Bit parameter 209.3 G84 Spindle Direction is reversed (0:No 1:Yes).

Bit reference 209.4 G88 rigid tapping whether to choose the Y axis as the spindle (0:No 1 Yes).

Bit reference 209.5 G88 rigid tapping whether to choose the fourth axis as the spindle (0:No 1 yes).

Bit reference 209.6 G88 rigid tapping whether to choose the fifth axis as the spindle (0:No 1 yes).

Bit reference 209.7 G88 spindle direction is reversed (0:no 1:yes).

PLC parameter K014.7 M29 rigid tapping output spindle position mode Y4.2 (0: valid 1: invalid).

Parameter 216 [G83/G87 Drilling Clearance] Default 2

Number parameter 217 [spindle clamping M code] default 20

Bit parameter 172.5 [Drilling/boring/rigid tapping/cycling spindle clamping automatic (0: No 1: Yes) is valid] Default 0

Note 1: Alarm when editing R is lower than the bottom of the hole position: the R point position is lower than the bottom position;

Note 2: Alarm when editing Z(W): Tapping cycle does not specify the bottom plane of the hole;

Note 3: When **the G84/G88 tapping cycle CS axis is automatically released (edited with the M20 command),** the C S axis command in the hole punch instruction G84/G88 is not moved, The spindle clamping instruction needs to wait for the instruction to complete before it is output; Instruction CS axis movement command needs to automatically release the spindle clamping,

M29 S100;//With M29 instructions for rigid tapping

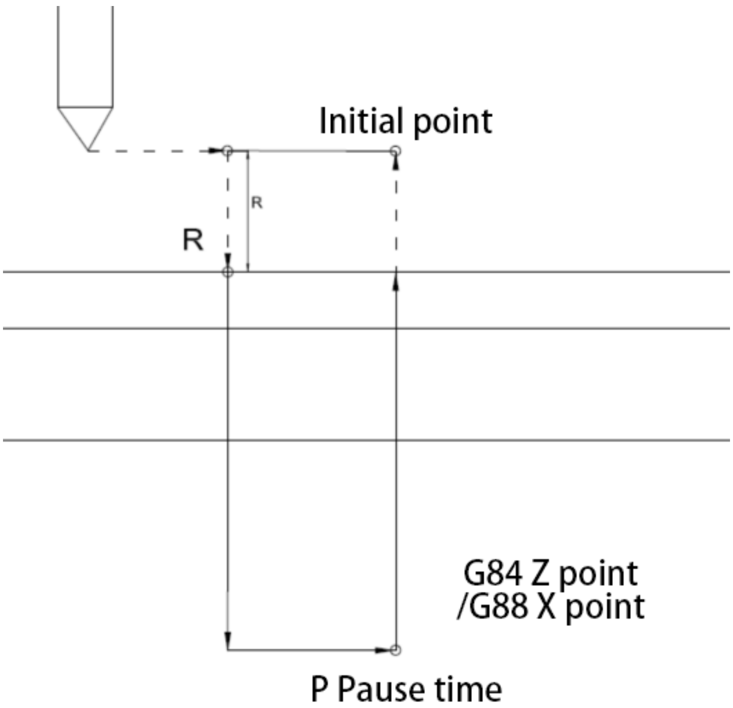
G84 C100 Z-20 R5 F100 M20; //After M21 is executed, after C100 is completed, M20 is executed;

C140 M20; // First after the execution of M21 is completed, after the execution of C100 is completed, the execution of M20;

G80;

Note 4: When the open bit parameter 172.5=1 [drilling /boring/rigid tapping/cycling spindle clamping automatic (0:no 1:yes) is valid], when the C axis needs to be clamped and loosened in the loop code, the M 20 code can be controlled without editing the spindle clamping and loosening, The system automatically joins the system to execute the release codeM21 before **executing the C coordinates, waiting for the completion of the C coordinates, and executes the M20 clamping code after the coordinates are completed; If the parameter is also edited with M20** open**, M20 shall prevail.**

Note 5: When selecting the flexible tapping bit parameter 213.0=0, editing M29 is also flexible tapping without changing the tapping state, G84 Issued spindle forward and reverse control, Z axis and spindle linkage tapping; Read the spindle feedback speed to control the Z axis speed, perform the internal execution M05 spindle stop before the G 84 instruction is executed, judge the direction of the output spindle according to the positive and negative of F/I, and output M05 after tapping Stop the spindle, the Z-axis tool feed is similar to the car thread and wait for the encoder to turn the signal before entering the tool.



Front tapping loop command format 2:

M29 S____ ; M29 opens rigid tapping, S___ tapping speed // do not writeM29 instructions for flexible tapping

G84 X(U)_ Z(W)_ R_ P_ F/I_ K_ M_ Q_;

X_ hole absolute value position (U_ hole for the starting point X axis of the incremental value position);

Z_ absolute value position at the bottom of the hole (W_ position of the incremental value at the bottom of the hole for the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

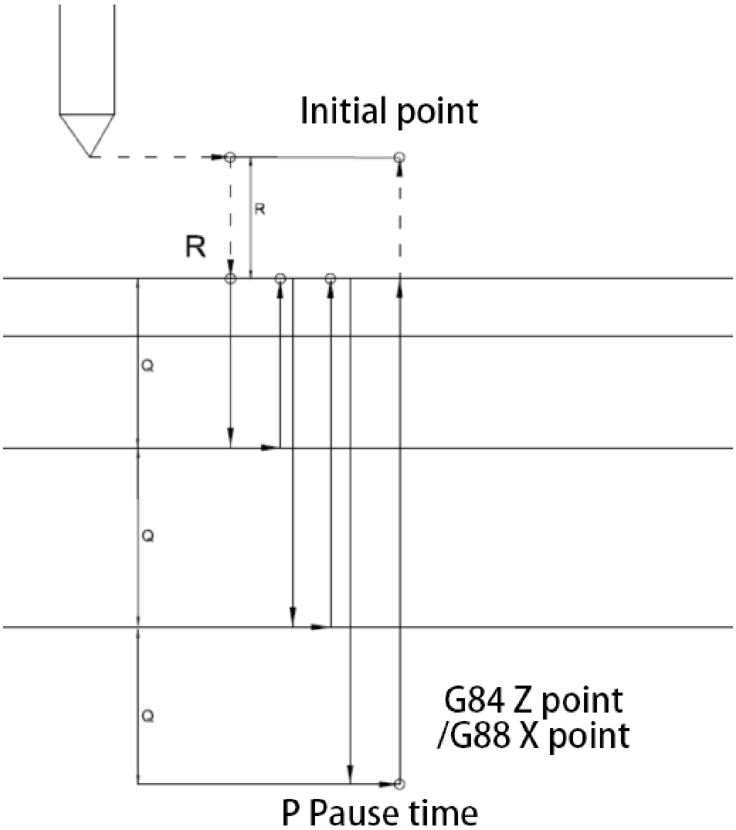
F_ metric tap pitch, I_ inch tooth count, **right hand is positive, left hand is negative;**

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Q_ depth of each cutting (regardless of positive and negative signs);

Description: This format will be layered, rewind to point R after each feed Q_, and then tapped feed.



Front Tapping Loop Command Format 3:

M29 S____ ; M29 opens rigid tapping, S___ tapping speed // do not writeM29 instructions for flexible tapping

G84 X(U)_ Z(W)_ R_ P_ F/I_ K_ M_ Q_ D_;

X_ hole absolute value position (U_ hole for the starting point X axis of the incremental value position);

Z_ absolute value position at the bottom of the hole (W_ position of the incremental value at the bottom of the hole for the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ metric tap pitch, I_ inch tooth count, **right hand is positive, left hand is negative;**

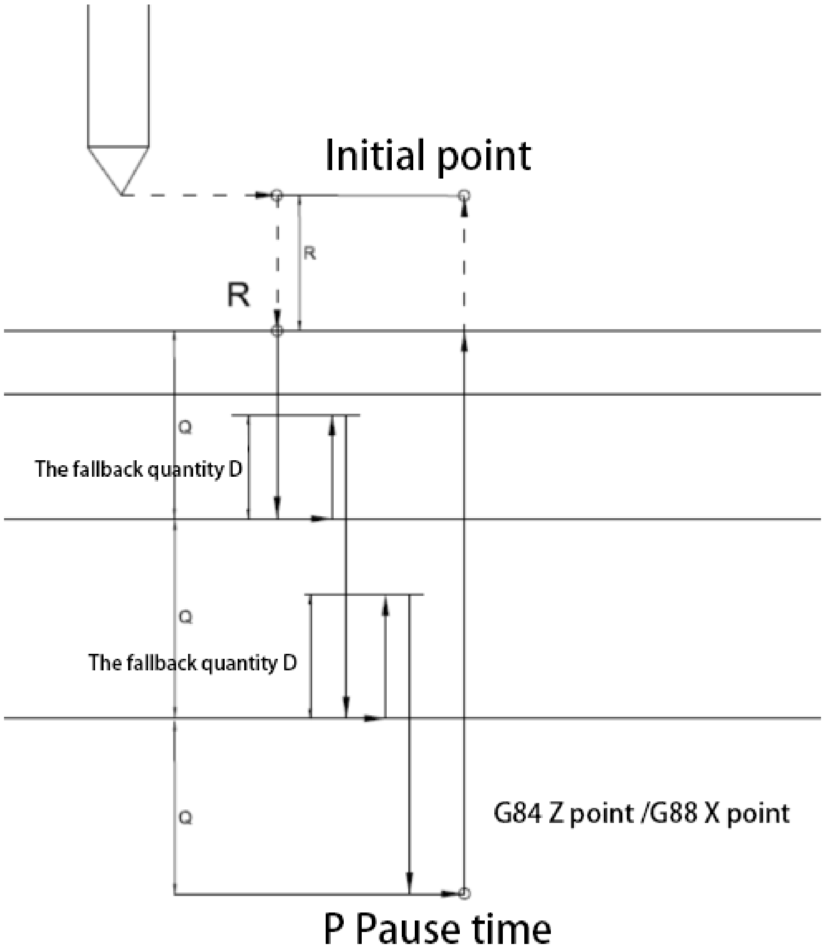
K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Q_ depth of each cutting (regardless of positive and negative signs);

D_ each fallback height (regardless of positive and negative signs);

Description: This format will be layered, each feed Q_ after the fallback height, and then tapped feed.



2. Side tapping cycle G88:

Side tapping loop command format 1:

M29 S____ ; M29 open rigid tapping, S_ __ tapping speed // do not writeM29 instructions for flexible tapping

G88 X(U)_ Z(W)_ R_ P_ F/I_ K_ M_;

Z_ hole absolute value position (W_ hole for the starting point Z axis of the incremental value position);

X_ hole bottom absolute value position (U_ hole bottom for the R point increment value position);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ metric tap pitch, I_ inch tooth count, **right hand is positive, left hand is negative;**

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Description: This format will be machined in one go.

The spindle control mode for bit reference 213.0 tapping is (0: followed by 1: servo).

When 213.0 chooses to follow, whether there is M29 instructions are flexible tapping, controlled by spindle forward and reverse; When selecting the servo, there is an M29 command for rigid tapping, and the tapping pulse port is selected by parameters; NoM29 instruction for flexible tapping.

Bit reference 209.0 G84 rigid tapping whether to choose the Y axis as the spindle (0:No 1 Yes).

Bit reference 209.1 G84 rigid tapping whether to choose the fourth axis as the spindle (0:No 1 Yes).

Bit reference 209.2 G84 rigid tapping whether to choose the fifth axis as the spindle (0:No 1 yes).

Bit parameter 209.3 G84 Spindle Direction is reversed (0:No 1:Yes).

Bit reference 209.4 G88 rigid tapping whether to choose the Y axis as the spindle (0:No 1 Yes).

Bit reference 209.5 G88 rigid tapping whether to choose the fourth axis as the spindle (0:No 1 yes).

Bit reference 209.6 G88 rigid tapping whether to choose the fifth axis as the spindle (0:No 1 yes).

Bit reference 209.7 G88 spindle direction is reversed (0:no 1:yes).

PLC parameter K014.7 M29 rigid tapping output spindle position mode Y4.2 (0: valid 1: invalid).

Parameter 216 [G83/G87 Drilling Clearance] Default 2

Number parameter 217 [spindle clamping M code] default 20

Bit parameter 172.5 [Drilling/boring/rigid tapping/cycling spindle clamping automatic (0: No 1: Yes) is valid] Default 0

Note 1: Alarm when editing R is lower than the bottom of the hole position: the R point position is lower than the bottom position;

Note 2: Alarm when editing X(U): Tapping cycle does not specify the bottom plane of the hole;

Note 3: When **the G84/G88 tapping cycle CS axis is automatically released (edited with the M20 command),** the C S axis command in the hole punch instruction G84/G88 is not moved, The spindle clamping instruction needs to wait for the instruction to complete before it is output; Instruction CS axis movement command needs to automatically release the spindle clamping,

M29 S100;//With M29 instructions for rigid tapping

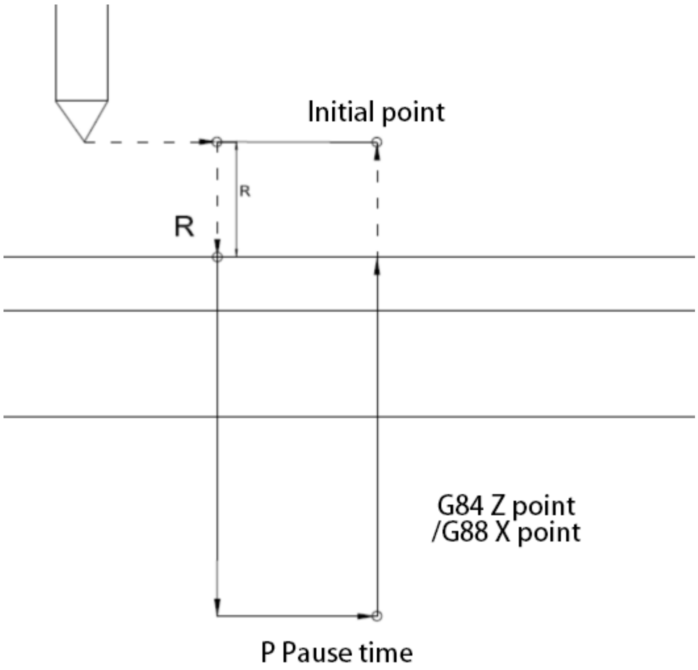
G88 C100 X-20 R5 F1 M20; // After the M21 is executed, after the C100 is completed, the M20 is executed;

C140 M20; After the first execution of M21 is completed, after the execution of C100 is completed, the execution of M20;

G80;

Note 4: When the open bit parameter 172.5=1 [drilling /boring/rigid tapping/cycling spindle clamping automatic (0:no 1:yes) is valid], when the C axis needs to be clamped and loosened in the loop code, the M 20 code can be controlled without editing the spindle clamping and loosening, The system automatically joins the system to execute the release codeM21 before **executing the C coordinates, waiting for the completion of the C coordinates, and executes the M20 clamping code after the coordinates are completed; If the parameter is also edited with M20** open**, M20 shall prevail.**

Note 5: When selecting the flexible tapping bit parameter 213.0 =0, editing M29 is also flexible tapping does not change the tapping state, G88 issues a spindle forward and reverse control, X Shaft and spindle linkage tapping; Read the spindle feedback speed to control the Z axis speed, perform an internal M05 spindle stop before the G 88 instruction is executed, judge the direction of the output spindle according to the positive and negative of F/I, and output M 05 after tapping Stop the spindle, the X-axis tool feed is similar to the car thread and wait for the encoder to turn signal before entering the tool.



Side tapping loop command format 2:

M29 S____ ; M29 open rigid tapping, S_ __ tapping speed // do not writeM29 instructions for flexible tapping

G88 X(U)_ Z(W)_ R_ P_ F_ K_ M_ Q_;

Z_ hole absolute value position (W_ hole for the starting point Z axis increment value position);

X_ absolute value position at the bottom of the hole (U_ hole bottom for the incremental value position of the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

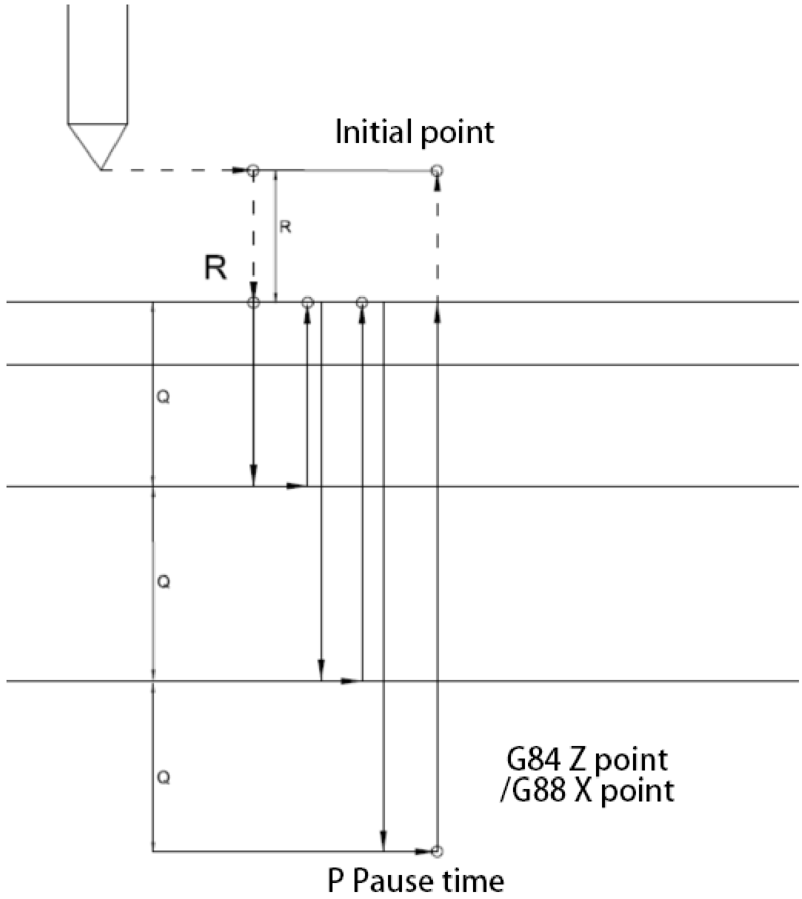
F_ metric tap pitch, I_ inch tooth count, **right hand is positive, left hand is negative;**

K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Q_ depth of each cutting (regardless of positive and negative signs);

Description: This format will be layered, rewind to point R after each feed Q_, and then tapped feed.



Side tapping loop command format 3:

M29 S____ ; M29 open rigid tapping, S_ __ tapping speed // do not writeM29 instructions for flexible tapping

G88 X(U)_ Z(W)_ R_ P_ F_ K_ M_ Q_ D_;

Z_ hole absolute value position (W_ hole for the starting point Z axis increment value position);

X_ absolute value position at the bottom of the hole (U_ hole bottom for the incremental value position of the R point);

R_ the increment value from the starting point to the R point (regardless of positive and negative signs), and the processing starts from the starting point by default without editing;

P_ hole bottom pause time, unit ms, default pause without editing;

F_ metric tap pitch, I_ inch tooth count, **right hand is positive, left hand is negative;**

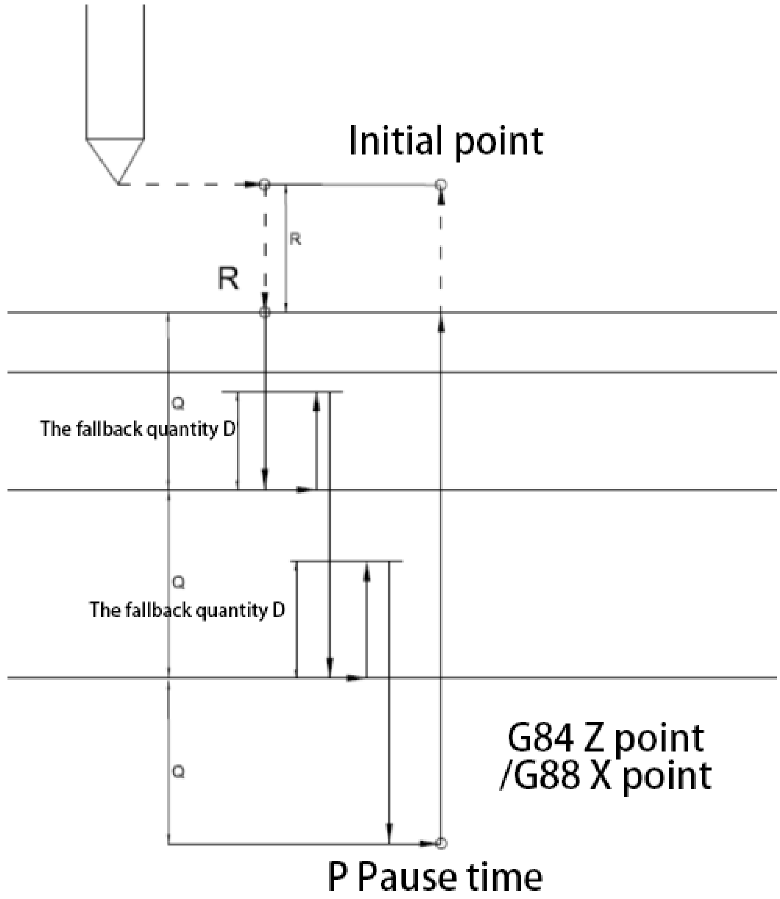
K_ the number of repetitions, the default is 1 time without editing, (only the current hole is valid, and K needs to be added after each coordinate when multiple holes need to be repeatedly machined);

M_ spindle clamping code, default not to clamp the spindle without editing;

Q_ depth of each cutting (regardless of positive and negative signs);

D_ each fallback height (regardless of positive and negative signs);

Description: This format will be layered, each feed Q_ after the fallback height, and then tapped feed.



Appendix IX: Online Modification of Special Instructions for Knife Patching G08/G09 (only available in versions above 2 021).

Programming format:

G08/G09 X(U) Z(W) Y(V) A/B(E) C(H) R(D);

Illustrate:

X/Z/Y/A/B/C absolute value programming, editing data for the current position absolute coordinates into the corresponding knife complement;
U/W/V/E/H/D incremental programming, programming data incrementally entered into the corresponding knife complement;
R(D) is the tip radius compensation: R absolute programming enters the edit data as the tip radius into the knife complement, and D is the incremental programming to enter the edit data increment into the knife complement;
G08 Compensation for the current tool complement;
G09 Compensation 00 No. 0 tool patch;
Corresponding parameters:
Using the G08/G09 instruction requires opening bit parameters 179.3=1 [G10 function is supported from new decoding (0: no 1: yes)];
When you need to modify the 00 gauge tool complement, you need to open the bit parameter 12.6=1 [NO.0 tool complement translation workpiece coordinate system (0: invalid 1: valid)].
Note 1: To compensate for the special case of the 00 gauge tool patch, when the current tool patch is 00, use the G 08 or G09 function, use the absolute programming G80/G09 X_, [X_] data is directly assignedto 0 0 In the case of0 0 0 tool patching of the original data 10, when G 08/G09 X20 is performed, the value of 0 0 0 tool complement becomes 20 , instead of adding 20 on top of10); Using the incremental programming G08/G09 U_, [U_] data increments are added inthe 00 tool complement (00 tool supplement the original data 10, G08/G09 U20 is performed Whenthe number 00 knife complement value becomes 30, it is20 on top of10).
Note 2: After compensatingfor 00 tool complement, all tool complements will increase the 00 compensation, which is similar to the base offset of the coordinate system; For example, if the workpiece is machined witha tool call of 0 1/02/03/04 and the diameter is 0.2 mm smaller, G 09 X/U can be used -0.2 Compensation 0 No. 0 knife patch to achieve compensation for other knife patch effects.
Note 3:0 No. 0 Knife patch compensation is opposite to the direction of other knife patch compensation, and the negative direction is compensated in the case of small size and the positive direction is made up in the case of large size.

G08/G09 Usage Example:

1） Compensation 00 No. 0 tool patch:

```
T0100;
G08 X/U-0.2;
or
T0101;//Current arbitrary tool complement
G09 X/U-0.2;
```

The effect is the same as compensating0 0 knife repair 0.2mm;

2） Incremental compensation for the current tool complement

T0101;//When the tool complement is 01, compensate for 01 tool patch

G08 U0.2;

Compensation 0 No. 1 tool patch, on the basis of the original tool patch data to increase 0.2mm, when the Diameter of the No. 01 tool patch processing workpiece is 0.2 mm smaller, you can use this method to compensate;

3） Absolute compensation for the current tool patch

T0101;//When the tool complement is 01, compensate for 01 tool patch

G08 X20;

Compensate for 0 No. 1 tool patch, take the current position as the position of the counter knife X20,and the value of the 0 No. 1 tool patch is the data of the normal pair X20 [machine coordinates]-[X20]; When T0101 is called again at this location, the absolute coordinate of the current X axis is 20.

Note: After modifying the knife patch, the tool patch needs to be called back to be effective;

For example: T0101;// When the tool complement is 01, compensate for 01 tool complement

G08 U0.2;

At this time, the compensated tool patch does not take effect, and only the compensated tool patch takes effect when the next T0101 instruction is executed.

Appendix 10, rotary axis control mode (only available in version 2 021 or above).

1、 Set the rotation axis parameters:

Bit parameter 187.0 sets the 3TH axis type (0: straight axis 1: rotary axis).

Bit parameter 187.2 sets the Z axis type (0: straight axis 1: rotation axis).

Bit parameter 189.0 sets the 4TH axis type (0: straight axis 1: rotary axis).

Bit parameter 189.2 sets the X-axis type (0: linear axis 1: rotary axis).

Bit parameter 191.0 sets the 5TH axis type (0: straight axis 1: rotary axis).

Note: When setting the X axis as the axis of rotation, you need to program the bit parameter 1.2=1 [(0:diameter 1:radius)], the bit reference 4.4=1 [The tool compensation value is expressed as the (0:diameter 1:radius) value].

2、 Rotary axis type:

The system divides the rotation axis settings into 2 categories, 1) near rotation, 2) rotation by symbol;

Bit reference 188.1 When the 3TH axis is the axis of rotation (0: Near rotation 1: Rotation in the direction of the symbol).

Bit parameter 188.4 When the Z axis is the axis of rotation (0: near rotation 1: rotate in the direction of the symbol).

Bit parameter 190.1 When the 4TH axis is the axis of rotation (0: near rotation 1: rotation in the direction of the symbol).

When the bit parameter 190.4 X axis is the axis of rotation (0: near rotation 1: rotation in the direction of the symbol).

Bit parameter 192.1 When the 5TH axis is the axis of rotation (0: Near rotation 1: Rotation in the direction of the symbol).

1) Near rotation control:

The nearest rotation control mode machine coordinates and absolute coordinates are looped between 0-360 degrees, the relative coordinate coordinates can be set to loop or not loop, the shortest path control is performed during absolute programming, and the incremental amount of movement is performed during incremental programming (the shortest distance is not performed).

N1 G0 C0	The sequence number	Actual position amount	The absolute coordinate value of the end point	Machine coordinates
N2 G0 C300	N2	-60, negative movement	300	300
N3 C100	N3	-160, negative movement	100	100
N4 C360	N4	-100, negative movement	0	0
N5 C-30	N5	-30, negative movement	330	330
N6 C0	N6	30, forward movement	0	0
N7 C360	N7	0, do not move	0	0
N8 C720	N8	0, do not move	0	0
N9 C0	N9	0, do not move	0	0
N10 C360	N10	0, do not move	0	0

N11 C360	N11	0, do not move	0	0
N12 H540	N12	540, forward movement	180	180
N13 H360	N13	360, forward movement	180	180
N14 H-200	N14	-200, negative movement	340	340

2) Select controls by symbol orientation:

According to the symbol direction rotation control mode machine coordinates and absolute coordinates are looped between 0-360 degrees, relative coordinate coordinates can be set to cycle or not loop, according to the symbol rotation can be divided into rotation axis A / B two Types :(Note: A/B types are not valid under the nearest rotation control).

Bit Parameter 187.1 Set 3TH axis as the type when rotating (0: Rotation axis A type 1: Rotation axis Type B).

Position parameter 187.3 Sets the Z axis to the type when rotating (0: axis of rotation A type 1: axis of rotation B type).

Bit parameter 189.1 Sets the 4TH axis to the type of rotation (0: axis of rotation A type 1: axis of rotation B type).

Position parameter 189.3 Sets the X axis to the type of rotation (0: axis of rotation A type 1: axis of rotation B type).

Position parameter 191.1 Sets the 5TH axis to the type of rotation (0: axis of rotation A type 1: axis of rotation B type).

Axis of rotation type A: Performs in a symbolic direction during absolute programming, operating in a manner similar to a linear axis, moving in a positive direction from C 0 to C300, C 300 to C200 when moving negatively, incremental programming performs incremental movement (shortest distance is not performed).

N1 G0 C0	The sequence number	Actual position amount	The absolute coordinate value of the end point	Machine coordinates
N2 G0 C300	N2	300, forward movement	300	300
N3 C100	N3	-200, forward movement	100	100

N4 C360	N4	260, forward movement	0	0
N5 C-30	N5	-30, negative movement	330	330
N6 C0	N6	-330, negative movement	0	0
N7 C360	N7	360, forward movement	0	0
N8 C720	N8	720, forward movement	0	0
N9 C0	N9	0, do not move	0	0
N10 C360	N10	360, forward movement	0	0
N11 C360	N11	360, forward movement	0	0
N12 H540	N12	540, forward movement	180	180
N13 H360	N13	360, forward movement	180	180
N14 H-200	N14	-200, negative movement	340	340

Note: In type A/B, when the coordinates go to a multiple of 3 60 or 360, the next line is followed by 360 or 3 When the value is multiplied by 60, the coordinates are calculated from 0; For example, the above programs C0-C360 (at this time walk a circle 360 degrees) - C720 (at this time from 0 Segments start counting 2 laps).)

Rotary axis type B: Absolute programming is performed in the direction of the symbol, and the operation mode is always moving in the positive direction (moving in the negative direction when there is a negative sign), moving in the positive direction from C0 to C300, Forward movement at C300 to C200 [300-360(0)-200], Incremental movement is performed when incremental programming (shortest distance is not performed).

N1 G0 C0	The sequence number	Actual position amount	The absolute coordinate value of the end point	Machine coordinates

N2 G0 C300	N2	300, forward movement	300	300
N3 C100	N3	160, forward movement	100	100
N4 C360	N4	260, forward movement	0	0
N5 C-30	N5	-30, negative movement	330	330
N6 C0	N6	60, forward movement	0	0
N7 C360	N7	360, forward movement	0	0
N8 C720	N8	720, forward movement	0	0
N9 C0	N9	0, negative movement	0	0
N10 C360	N10	360, forward movement	0	0
N11 C360	N11	360, forward movement	0	0
N12 H540	N12	540, forward movement	180	180
N13 H360	N13	360, forward movement	180	180
N14 H-200	N14	-200, negative movement	340	340

Note: In type A/B, when the coordinates go to a multiple of 3 60 or 360, the next line is followed by 360 or 360 When the multiple value, the coordinates are calculated from 0; For example, the above programs C0-C360 (at this time walk a circle 360 degrees) - C720 (at this time from 0 Segments start counting 2 laps).)

1. Rotate the negative angle of type A/B by symbol **to edit the situation**

Bit parameter 192.6 Rotate negative angle instruction type 0 by sign (00:type A 01:B type 10:C type 11: D type).

Bit Parameter 192.7 Rotate Negative Angle By Symbol Instruction Type 1 (00:Type A 01:B Type 10:Type C 11: D type).

1. Negative angle edits cases where the negative angle does not exceed -360
2. Negative angle editing cases above -360

1) Negative angle type A 192.7=0 , 192.6=0

When not more than -360, go negative direction to the shortest distance;

When it exceeds -360, the negative (linear axis) distance is taken.

N1 G0 C0	The sequence number	The actual amount of movement	Absolute coordinates	Machine coordinates
N2 C-260	N2	-260	100	100
N3 C-260	N3	0	100	100
N4 C-360	N4	-460 (-100-360)	0	0
N5 C-540	N5	-540 (-360-180)	180	180
N6 C-540	N6	-720 (-180-360-180)	180	180
N7 C-260	N7	-80	100	100
N8 C-720	N8	-820 (-100-360-360)	0	0

2) Negative angle type B 192.7=0, 192.6=1

When exceeding or not exceeding -360, go negative (linear axis) distance.

N1 G0 C0	The sequence number	The actual amount of movement	Absolute coordinates	Machine coordinates
N2 C-260	N2	-260	100	100
N3 C-260	N3	-360 (-100-260)	100	100
N4 C-360	N4	-460 (-100-360)	0	0
N5 C-540	N5	-540 (-360-180)	180	180
N6 C-540	N6	-720 (-180-360-180)	180	180
N7 C-260	N7	-440 (-180-260)	100	100
N8 C-720	N8	-820(-100-360-360)	0	0

3) Negative angle type C 192.7=1, 192.6=0

When not more than -360, go negative direction to the shortest distance;

When exceeding -360, go negative for a relatively short distance, continuously the same instruction does not exceed -360 no movement, continuously more than -360 degrees of movement -360 displacement.

N1 G0 C0	The sequence number	The actual amount of movement	Absolute coordinates	Machine coordinates
N2 C-260	N2	-260	100	100
N3 C-260	N3	0	100	100
N4 C-360	N4	-100	0	0
N5 C-540	N5	-180	180	180
N6 C-540	N6	-360 (-180-180)	180	180
N7 C-260	N7	-80	100	100
N8 C-720	N8	-100	0	0

4) Negative angle D type 192.7=1, 192.6=1

When exceeding or not exceeding -360, the shortest distance in the negative direction;

N1 G0 C0	The sequence number	The actual amount of movement	Absolute coordinates	Machine coordinates
N2 C-260	N2	-260	100	100
N3 C-260	N3	0	100	100
N4 C-360	N4	-100	0	0
N5 C-540	N5	-180	180	180
N6 C-540	N6	0	180	180
N7 C-260	N7	-80	100	100
N8 C-720	N8	-100	0	0

Appendix 11: In-program invocation of subroutines functions (2 022.12 subsequent versions have).

Note: On the basis of the original M98P_ call subroutine, the use of calling subroutines within the program is added (the main program is in the same program as the subroutine).

Parameter: Bit parameter 184.6 M98 H_L_ in-program subroutine function (0: invalid 1: valid) [default invalid.]]；

Format: M98 H_ L_;

Description: H_ subroutine start line number;
L_ Number of calls;

When L_ is omitted, it is called once by default
The subroutine ends M99 , and the subroutine returns to execute the next line of the main program;

Note 1: The subroutine must end with M99, otherwise the call will be incorrect;
Note 2: When the subroutine does not end with M99, the subroutine is called only 1 time, and the entire program ends;
Note 3: Subroutines are nested up to 10 levels;

Example:
O0001;
M3S1000;
....
....
M98 D10 L5; //Call the subroutine after N10 5 times
N5 ....
....
M98 D20 L5; //Call N20 follows the subroutine 5 times
N6 ....
N7 ....
......

M30;//End of procedure

N10;//subroutine D10

....

....

M99;// Returns N5 lines after the end of the subroutine

N20;//subroutine D20

....

....

M99;/The subroutine ends returning N6 lines

Appendix XII, M89 Custom Detection/Output

- **Programming Format Description:**

① M89 K_ J_ R_ P_ I_ Q _

K_: Select K0—K7 for the heartbeat

J _ : Wait detection time in milliseconds

R_: Repeat output interval in milliseconds

P_: Number of repetitions of output The range P0 is 1 output, P1 is 2 repetitions, P2 is 3 repetitions, and so on

I _ : Select Open Output Address I0—I7

Q _ : Select The output address Q0—Q7 is selected

Note: The output addresses of I/Q on and off correspond to each other.

M89 output I_, waiting for detection K_, no detection K_ the output I_ in J_ time, turning on the output after R_ time I_ alarm when no K_ detected within the P_ of cycles, and closing the output at the same time;

If K_ is detected within P_ of the number of cycles, the M89 code is completed, and the Q_ output is turned off, and if the Q_ is not filled, it remains I_;

If I_=Q_, the output is turned off when M89 completes I_; If I_ ≠Q_, the output is maintained I_ and the Q_ output is turned off.

② M89 K_ J_

When the M89 does not detect K_ within J_ time, an alarm is generated;

M89 waits for J_ time to detect K_, the M89 code completes;

③ M89 I_

M89 output I_ hold

④ M89 Q _

M89 off output Q _

⑤ M89 I_ J_ Q _

M89 output I_ Elapsed time J_ Turn off output Q_;

If the Q_ is not filled in, the M89 output I_ The elapsed time J_ closes the output;

If elapsed time I_ = Q_J_ the output is turned off;

If the I_ ≠Q_, the I_ output is maintained after the time J_ elapses, and the Q_ output is turned off.

⑥ M89 K_

M89 waits for detection of K_ when the M89 code completes;

⑦ M89 K_ J_ I_

M89 output I_ Wait for detection K_, detect K_ within the time J_, M89 code complete;

If the K_ is not detected within the time J_, the system generates an alarm;

When the J_ or J=0 is not written, wait until the detection K_ signal is detected until the K_ signal is detected to complete the M89 code, and no alarm is generated;

⑧ M89 K_ J_ Q _

M89 closes output Q _ waits for detection K_, K_ detected within the time J_, M89 code completes;

If the K_ is not detected within the time J_, the system generates an alarm;

When the J_ or J=0 is not written, wait until the detection K_ signal is detected until the K_ signal is detected to complete the M89 code, and no alarm is generated;

M89 L_ Signal Detection Minimum Hold Time When the M89 K detection function is turned on, if there is no L programming, the code is completed as soon as there is a signal, and when there is an L programming, it is necessary to continuously detect that the signal is greater than the L editing time to complete the code;

All formats edited with K can use L detection duration.

Note: When K0-K7 is not written after M89**, the M89 code is directly completed;**

The address relationship between the M89K0—K7 heartbeat and input port X

M89I0—I7/Q0—The address relationship between the Q7 output/shutdown signal and the output port Y

Note: The output input address needs to be viewed in the "Debugging Document" M89 corresponding to different system models

Appendix XIII, M90/M91 program segment loops

- Programming Format Description:

M91 C_;//Cycles start with the number of cycles C_
;
;
M90;//End of cyclic program segment

Example:
T0101;
M3 S1000;
G0 X32 Z5;
M91 C5; The looping program segment begins, and the cycle is made 5 times (loop between N10-N20 program segments).
N10 G01 X30 Z0 F500;
G01 Z-20;
G01 X20 Z-40;
G0 X32;
N20 Z0;
M90; //End of loop segment

Appendix 14, M92/M93 Program Jumps

M92 Conditional Jump Command:

- Programming format:

  M92 K_ N_ ;

  M92 L_ N_ ;

  K_ Input high; (Range K0--K7 corresponds to #1000--#1007=1)

  L_ Input low; (Range L0--L7 corresponds to #1000--#1007=0)

  N_ Jump program segment number;

  When the conditions are met, jump to the N_ segment;

Example:

M92 K0 N10;//When #1000=1, the program jumps to the N10 segment; The G65 H81 P10 Q#1000 R1 instruction is the same

G0 X50 ;

N10 G0 X30 ;

G0 X0;

M92 L0 N10;// When #1000=0, the program jumps to the N10 segment G65 H81 P10 Q#1000 R0 instructions are the same

M93 Unconditional Jump Command:

- Programming format:

  M93 N_ ;

  N_ Jump program segment number;

  Jump to N_ when running to M93 N_;

  Example:

  M93 N10;//Jump to N10 segment

  G0 X50;

  N10 G01 X30;

  When running to the M93 N10 segment, the program jumps to the N10 segment without executing the G0 X50.

Note: Enter the address to check the "Debugging Documentation" macro input article corresponding to different system models