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

Contents

Preface

Dear user: We are honored and grateful for your selection of the product!

This user manual describes in detail the programming, operation, installation and connection matters of the SZGH880T/SZGH1080T lathe CNC.

In order to ensure the safe, normal and effective operation, please be sure to read this user manual carefully before installing and using the product.

Improper operation will cause accidents. Only qualified personnel are allowed to operate this system.

Special note: The system power supply installed on (inside) the chassis is a dedicated power supply provided only for the CNC system manufactured by our company. It is prohibited to use this power supply for other purposes. Otherwise, it will be extremely dangerous!

Precautions

■ Transportation and storage

1. Product packaging boxes should not be stacked more than six layers;

2. Do not climb, stand or place heavy objects on the packaging boxes;

3. Do not use the cables connected to the product to drag or move the product;

4. It is strictly forbidden to collide with or scratch the panel and display;

5. The packaging boxes should be kept away from 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 the list to confirm whether all parts are complete or damaged;

4. If there is any discrepancy in the product model, missing accessories or transportation damage, please contact our company in time.

■ Wiring

1. Personnel participating in wiring and inspection must be professionals with corresponding capabilities;

2. The product must be reliably grounded, and the grounding resistance should be less than 0.1Ω. The ground line mustn't be replaced by neutral line;
3. The wiring must be correct and firm to avoid product failure or unexpected consequences;

4. The surge absorption diode connected to the product must be connected in the specified direction, or it will damage the product;
5. The power supply of the product must be cut off before plugging or unplugging the plug or opening the chassis.

■ Maintenance

1. The power supply must be cut off before maintenance or replacement of components;

2. When a short circuit or overload occurs, the fault should be checked and restart is not allowed until the fault is eliminated;

3. The product shouldn’t be frequently turned on and off. If it needs to be re-powered after a power outage, the interval should be at least 1 min.

We try to explain various contents as much as possible in this manual, but due to too many possibilities involved, it is impossible to explain all the operations that can or cannot be performed one by one. Therefore, the contents not specially explained in this manual can be considered as unusable.

Before installing, connecting, programming and operating this product, you must read this product manual and the machine tool manufacturer's instruction manual in detail, and strictly follow the requirements of the manual and instruction manual to perform related operations, or it may cause damage to the product and machine tool, scrapping of workpieces or even personal injury.

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

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

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

Part 1 Programming Instructions

Introduction to technical specifications, product models, instruction codes and program formats.

Part 2 Operating Instructions

Introduction to the operation and use of SZGH880T/SZGH1080TCNC.

Part 3 Installation and Connection

Introduction to the installation, connection and setting methods of SZGH880T/SZGH1080TCNC.

Appendix

Introduction to the standard parameters, alarm information of SZGH880T/SZGH1080TCNC.

Safety Responsibilities of Manufacturer

- The manufacturer shall be responsible for the dangers that have been eliminated and/or controlled in the design and structure of the provided CNC system and the supplied accessories.

- The manufacturer shall be responsible for the safety of the provided CNC system and the supplied accessories.

- The manufacturer shall be responsible for the information and suggestions about using the product provided to the user.

Safety Responsibilities of Users

- The user shall be familiar with and master the content of safe operation through learning and training on safe operation of the CNC system.

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

- The user shall be responsible for the danger caused by operating, adjusting, maintaining, installing and storing the product without following the provisions of the manual.

* This manual shall be kept by the end user. Thank you for your support for our company and for your use of our products!

Part 1

Programming Instructions

Chapter 1 Programming Basics

1.1 Introduction to SZGH880T/SZGH1080T

1.1.1 Product Overview

The system can control 6 feed axes (including C axis) and 2 analog spindles, with 1ms high-speed interpolation and 0.1μm control accuracy, significantly improving the efficiency, accuracy and surface quality of parts processing.

* The six axis controls, that is, X, Z, Y, A, B, C, can be re-named and their axis type can be defined by the user

*Users can opt for 1ms interpolation cycle, and control accuracy（either 1μm or 0.1μm)

* Maximum speed 60m/min

* Adapting servo spindle can realize continuous positioning of the spindle, rigid tapping, and rigid thread processing

* With built-in multiple PLC programs, the currently running PLC program can be selected

* Statement-based macro code programming, and macro program call with parameters are supported

* Metric/imperial system programming, with automatic tool setting, automatic chamfering, tool life management functions is provided

*Display in Chinese or English is selectable by setting the parameters

* The USB interface provides support for file operation, system configuration and software upgrade on USB flash drive

* 2 electrical circuits with analog voltage output ranging from -10V ~ 10V support dual spindle control

* 1 electrical circuit handwheel input, which supports handheld unit

* 48-point general input/36-point general output

1.1.2 Technical Specifications

Number of controlled axes

* Number of controlled axes: 6 (X, Z, Y, A, B, C)

* Number of linked axes: 6
Feed axis function

* Least input increment: 0.001mm (0.0001 inch) and 0.0001mm (0.00001 inch) optional

* Least command increment: 0.001mm (0.0001 inch) and 0.0001mm (0.00001 inch) optional

* Maximum stroke: ±99999999× least command increment

* Rapid moving speed: up to 60m/min

* Rapid rate: 4 levels of real-time adjustment at F0, 25%, 50% and 100% respectively

* Feed rate: Sixteen levels of real-time adjustment ranging from 0 to 150%

* Interpolation mode: linear interpolation, circular interpolation (three-point circular interpolation is supported), thread interpolation, rigid tapping

* Automatic chamfering function

Thread function

* Ordinary thread (following the spindle)/rigid thread

* Single-start/multi-start straight thread, tapered thread and end face thread, equal pitch thread and variable pitch thread measured in metric/imperial system

* Thread back-off length, angle and speed characteristics can be set

* Thread pitch: 0.01mm ~ 500mm or 0.06 thread/inch ~ 2540 threads /inch

Acceleration and deceleration function

* Cutting feed: front acceleration/deceleration linear type, front acceleration/deceleration S type, rear acceleration/deceleration linear type, rear acceleration/deceleration exponential type

* Quick movement: front acceleration/deceleration linear type, front acceleration/deceleration S type, rear acceleration/deceleration linear type, rear acceleration/deceleration exponential type

* Thread cutting: linear type and exponential type optional

* The starting speed, end speed and acceleration/deceleration time of acceleration/deceleration are set by parameters

Spindle function

* 2 electrical circuits with analog voltage output ranging from -10V ~ 10V support dual spindle control

* With 1 electrical circuit of spindle encoder feedback, spindle encoder line number can be set (100p/r ~ 5000p/r)

* Transmission ratio of encoder to spindle: (1 ~ 255) : (1 ~ 255)

* Spindle speed: It can be given by S code or PLC signal with speed ranging from 0r/min ~ 9999r/min

* Spindle ratio: 8 levels of real-time adjustment from 50% to 120%

* Spindle constant linear speed control

* Rigid tapping

Tool functions

* Tool length compensation

* Tool nose radius compensation (C type)

* Tool wear compensation

* Tool life management

* Tool setting method: fixed point, trial cutting, reference point, and automatic tool setting

* Tool offset execution method: coordinate modification, tool movement

Precision compensation

* Backlash compensation

* Memory pitch error compensation

PLC functions

* Two-level PLC program, up to 4700 steps, refresh cycle of the first level program is 8ms

* PLC program communication download

* PLC warning and PLC alarm are available

* Support multiple PLC programs (up to 20), currently running PLC program can be selected

* Basic I/O: 36 inputs/36 outputs

Human-machine interface

* 8.0-inch widescreen LCD, resolution 800×600

* Display in Chinese, English and other languages

* 2D tool trajectory display

* Real-time clock

Operation management

* Operation mode: editing, automatic, input, machine tool homing, handwheel/single step, manual, program homing

* Multi-level operation authority management

* Alarm log

Program editing

* Program capacity: 240MB, 1000 programs (including subroutines, macro programs)

* Editing function: program/program segment/word search, modification, deletion, copy, paste

* Program format: ISO code, macro code programming, support relative coordinates, absolute coordinates and mixed coordinates programming are available

* Program call: macro program call with parameters, including 4 subroutine nesting, is available

Communication functions

* USB: file operation and direct file processing on USB flash drive, support PLC program, system software USB upgrade

Safety functions

* Emergency stop

* Hardware travel limit

* Software travel check

* Data backup and recovery

G code table

Code	Function	Code	Function
G00	Fast positioning	G31	Jump function
G01	Linear interpolation	G32	Equal pitch thread cutting
G02	Clockwise circular interpolation	G32.1	Rigid thread cutting
G03	Counterclockwise circular interpolation	G33	Z axis tapping cycle
G04	Pause, pre-stop	G34	Variable pitch thread cutting
G05	Three-point circular interpolation	G40	Cancel tool nose radius compensation
G6.2	Clockwise elliptical interpolation	G41	Tool nose radius left compensation
G6.3	Counterclockwise elliptical interpolation	G42	Tool nose radius right compensation
G7.1	Cylindrical interpolation	G50	Floating workpiece coordinate system
G7.2	Clockwise parabola interpolation	G52	Local workpiece coordinate system
G7.3	Counterclockwise parabola interpolation	G54-G59	Set workpiece coordinate system
G8.2	Clockwise hyperbolic interpolation	G65	Macro code non-modal call
G8.3	Counterclockwise hyperbolic interpolation	G71	Axial roughing cycle
G10	Wear compensation	G72	Radial roughing cycle
G12.1	Start polar coordinate interpolation	G73	Closed cutting cycle
G13.1	Cancel polar coordinate interpolation	G70	Finishing cycle
G12.2	Clockwise involute interpolation	G74	Axial grooving cycle
G13.2	Counterclockwise involute interpolation	G75	Radial grooving cycle
G12.3	Clockwise exponential interpolation	G76	Multiple thread cutting cycle
G13.3	Counterclockwise exponential interpolation	G78	T-thread cutting cycle
G15	Cancel polar coordinate instruction	G80	Cancel rigid tapping state
G16	Polar coordinate instruction	G84	Axial rigid tapping
G17	Plane selection code	G88	Radial rigid tapping

G18	Plane selection code	G90	Axial cutting cycle
G19	Plane selection code	G92	Thread cutting cycle
G20	Imperial unit selection	G94	Radial cutting cycle
G21	Metric unit selection	G96	Constant linear speed control
G24	Start turning	G97	Cancel constant linear speed control
G25	Cancel turning	G98	Feed per minute
G28	Automatically return to machine tool origin	G99	Feed per revolution
G30	Return to the 2nd, 3rd, and 4th reference points of the machine tool	G104	Chip breaking function

1.1.3 Adaptability to Climate and Environment

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

Item	Working climate conditions	Storage and transportation climate conditions
Ambient temperature	0℃~ 45℃	-40℃~ +70℃
Relative humidity	≤ 90% (non-condensing)	≤ 95% (40℃)
Atmospheric pressure	86 kPa ~ 106 kPa	86 kPa ~ 106 kPa
Altitude	≤ 1000m	≤ 1000m

1.1.4 Power Adaptability

Operate normally under the following AC input power conditions. Voltage change: within the range of (0.85 ~ 1.1) × rated AC input voltage (AC220V); frequency change: 49Hz ~ 51Hz continuous change.

1.1.5 Protection

Protection level is not less than IP20.

1.2 Programming Basics

1.2.1 Workpiece Coordinate System and Program Origin

The workpiece coordinate system is a rectangular coordinate system set according to the part drawing. After the part is clamped on the machine tool, use G50/G54~G59 to set the absolute coordinates of the current position of the tool according to the size of the workpiece, and establish the workpiece coordinate system in the 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 end of the part. Once the workpiece coordinate system is established, it will remain valid until it is replaced by a new workpiece coordinate system.

The current position of the workpiece coordinate system set with G50 is called the program origin, and it returns to this position after executing the program homing operation.

Note: If the workpiece coordinate system is not set with G50 after power-on, do not execute the program homing operation, or an alarm will be generated.

1.2.2 Absolute Coordinate Programming and Relative Coordinate Programming

When writing a program, it is necessary to give the coordinate value of the trajectory end point or target position. According to the type of programming coordinate value, it can be divided into three programming methods: absolute coordinate programming, relative coordinate programming and mixed coordinate programming.

Programming with the absolute coordinate values of the X and Z axes (expressed by X and Z) is called absolute coordinate programming; programming with the relative displacement of the X and Z axes (expressed by U and W) is called relative coordinate programming;

It is allowed to use absolute programming coordinate values and relative displacement programming for the X and Z axes in the same program segment, which is called mixed coordinate programming.

1.2.3 Diameter Programming and Radius Programming

According to whether the X-axis coordinate value is input as a diameter value or a radius value during programming, it can be divided into: diameter programming and radius programming.

Diameter programming: When the system parameter N0001 is 0, the programming value of the X-axis in the program is input as a diameter value, and the coordinate of the X-axis is displayed as a diameter value.

Radius programming: When the system parameter N0001 is 1, the programming value of the X-axis in the program is input as a radius value, and the coordinate of the X-axis is displayed as a radius value.

1.2.4 General Structure of Program

Program name	O0001
Comment line “//”	//Comment
	N0010 G50 X0 Z0
Program segment selection character “/”	/N0020 G01 X100 F500
	N0030 X200

	N0040 M30
Program end character	%

Program name

A maximum of 1000 programs can be stored. In order to identify and distinguish each program, each program has a unique program name (program names are not allowed to be repeated). The program name is located at the beginning of the program starting with the letter O, which is followed by four digits.

Chapter 2 MST Code

2.1 M Code (Auxiliary Function)

M code consists of code address M followed by 1~2 digits or 4 digits. It is used to control the flow of program execution or output M code to PLC.

M98 and M99 are processed independently by NC and do not output M code to PLC.

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

M98 and M99 are used as program call codes, and M02 and M30 are used as program end codes. The PLC program cannot change the meaning of the above codes. Other M codes are output to PLC, and the code functions are defined by the PLC program. Please refer to the manual of the machine tool manufacturer.

There can only be one M code in a program segment. When two or more M codes appear in the program segment, the CNC will alarm.

Table 2-1 M code list for controlling the execution of program

Code	Function
M02	Program end
M30	Program end
M97	Local loop
M98	Subroutine call
M99	Return from subroutine; if M99 is used to end the main program (i.e. the current program is not called by other programs), the program will be executed repeatedly
M35	Automatic repetitive feeding function
M80, M81	Forced signal output
M82, M83	Wait for external signal
M103	Y axis counterclockwise rotation
M104	Y axis clockwise rotation
*M105	Y axis stop
M110	Y axis disabled
M111	Y axis enabled

2.1.1 Program End M02

Code format: M02 or M2

Code function: In automatic mode, execute M02 code, after other codes of the current program segment are executed, automatic operation ends, and the number of processed pieces increases by 1. Tool nose radius compensation cancels, and the cursor returns to the beginning of the program (whether to return to the beginning of the program is determined by parameters).

2.1.2 End of Program Operation M30

Code format: M30

Code function: In automatic mode, execute M30 code, after other codes of the current program segment are executed, automatic operation ends, and the number of processed pieces increases by 1. Tool nose radius compensation cancels, and the cursor returns to the beginning of the program (whether to return to the beginning of the program is determined by parameters).

2.1.3 Local Loop M97

Code format: M97 P Q L

Code description: P: starting number N;

Q: ending number N;
L: number of calls;

Code function: Execute the code between P and Q in a loop, and after executing L times, execute the code after M97. M97 is usually placed after Q.

2.1.4 Subroutine Call M98

Code format: M98 P。。 □□□□

The subroutine number to be called (0000 ~ 9999). When the number of calls is not entered, the leading 0 of the subroutine number can be omitted; when the number of calls is entered, the subroutine number must be 4 digits; the number of calls (1-9999) can be omitted when calling once.

Code function: In automatic mode, when executing the M98 code, after the other codes of the current program segment are executed, the CNC calls and executes the subroutine specified by P, and the subroutine can be executed up to 9999 times. M98 code is invalid when running under MDI.

2.1.5 Return from Subroutine M99

Code format: M99 P_

Code description: P: Returns the program segment number (0000 ~ 9999) to be executed in the main program, and the leading 0 can be omitted.

Code function: After the other codes of the current program segment (in the subroutine) are executed, return to the program segment specified by P in the main program to continue execution. When P is not entered, return to the next program segment after the M98 code that calls the current subroutine in the main program to continue execution. If M99 is used to end the main program (i.e. the current program is not called by other programs), the current program will be executed repeatedly. M99 code is invalid when running under MDI.

2.1.6 Automatic Repeated Feeding Function M35

Instruction format: M35 K_ I_ J_ P_ L_

Code description: K: feeding output port;

I: in-position signal input port, a signed number, the sign is used to indicate the effective level, <0 low level is effective, >0 high level is effective, and it cannot be 0;

J: The maximum waiting time for the in-position signal, unit: seconds, during this period, the port specified by K remains open, and when the waiting time is up, the output port specified by K is closed;

P: The delay between two feedings, that is, the delay time from closing the feeding output port to opening it again, unit: seconds, may have a decimal point, during this period, the port specified by K remains closed;

L: The number of repeated feeding executions, when less than 0, it is treated as 1.

Function description: M35 is suitable for automatic feeding process. When the feeding is stuck, it can automatically return and feed again to improve the success rate of feeding and processing efficiency.

This function is used for feeding control. When feeding fails, it will automatically retry L times.

Action process:

(1) Feeding output port is opened;

(2) Check whether the input of port I is valid. If it is valid, execute (6); otherwise when the time specified by J is reached, execute the next step, otherwise execute (2);

(3) Feeding output port is closed, and the number of retries is increased by 1;

(4) If the number of retries is greater than or equal to L, an alarm is issued; otherwise the next step is executed;

(5) Wait for the time specified by P, execute (1);

(6) Close the feeding output and end.

2.1.7 Forced Signal Output M80 and M81

Code format: M80 K J;

M81 K J;

Code description: K: port number, defined in the output port definition interface;
J: delay time, unit ms;

Code function: M80: Force a port to output high level, and after delaying J, restore to the original state;
M81: Force a port to output low level, and after delaying J, restore to the original state;

2.1.8 Wait for External Signal M82 and M83

Code format: M82 L J;
M83 L J;

Code description: L: Port number, defined in the input port definition interface;
J: Waiting timeout, if there is no J, wait all the time, unit ms;

Code function: M82: Wait until the input port is high level;
M83: Wait until the input port is low level;

2.1.9 Y Axis Counterclockwise, Clockwise and Stop Control M103, M104 and M105

Code format: M103;
M104;
M105;

Code function: M103: Counterclockwise rotation;
M104: Clockwise rotation;
M105: Stop;

2.1.10 Whether Y Axis Enabled M110 and M111

Code format: M110;
M111;

Code function: M110: Y axis is not enabled;
M111: Y axis is enabled;

2.1.11 M Codes Defined by Standard PLC Ladder Diagram

Except for the above codes (M02, M30, M98, M99), other M codes are defined by PLC. The following are M codes defined by standard PLC. SZGH880T/SZGH1080T series lathe CNC is used for machine tool control. Please refer to the instructions of the machine tool manufacturer for the function, meaning, control sequence and logic of M codes.

M codes defined by standard PLC ladder diagram

Code	Function	

M00	Program pause	
M01	Program selection stop	
M03	Spindle forward	
M04	Spindle reverse	
M05	Spindle stop	
M08	Cooling on	
M09	Cooling off	
M10	Tailstock in	
M11	Tailstock out	
M12	Chuck clamping	Chuck not associated with spindle: common parameters, chuck classification, K12.0 changed to 0, K12.1 changed to 1
M13	Chuck release	
M14	Spindle position control	
M15	Spindle speed control	
M20	Spindle clamping	
M21	Spindle release	
M32	Lubrication on	
M33	Lubrication off	
M41, M42, M43, M44	Spindle automatic gear shifting	
M50	Cancel spindle orientation	
M51	Spindle orientation	
M63	Second spindle clockwise rotation	Open the second spindle: PLC parameter, K17.7 changed to 1
M64	Second spindle counterclockwise rotation	
M65	Second spindle stop	

Code	Function	Remarks
M00	Program pause	
M01	Program selection stop	
M03	Spindle counterclockwise rotation	Function interlock, state hold
M04	Spindle clockwise rotation	
*M05	Spindle stop	
M08	Coolant on	Function interlock, state hold
*M09	Coolant off	
M10	Tailstock in	Function interlock, state hold

M11	Tailstock out	
M12	Chuck clamping	Function interlock, state hold
M13	Chuck release	
M14	Spindle position control	Function interlock, state hold
*M15	Spindle speed control	
M20	Spindle clamping	Function interlock, state hold
*M21	Spindle release	
M24	Second spindle position control	Function interlock, state hold
*M25	Second spindle speed control	
M32	Lubrication on	Function interlock, state hold
*M33	Lubrication off	
M63	Second spindle counterclockwise rotation	Function interlock, state hold
M64	Second spindle clockwise rotation	
*M65	Second spindle stop	
*M41, M42, M43, M44	Spindle automatic gear shifting	Function interlock, state hold

Note: The codes marked with "*" defined by the standard PLC are valid when powered on.

2.1.12 Program Stop M00

Code format: M00 or M0

Code function: After executing the M00 code, the program stops and the word "Pause" is displayed. After pressing the cycle start key, the program continues to run.

2.1.13 Program Selection Stop M01

Code format: M01 or M1

Code function: Valid in automatic and input modes. Press the selection stop button 

to make the stop button indicator light up, which indicates that it has entered the selected stop state. At this time, after executing the M01 code, the program stops running and the word "Pause" is displayed. After pressing the cycle start key, the program continues to run. If the program selection stop switch is not turned on, the program will not pause even if the M01 code is run.

2.1.14 Counterclockwise Rotation, Clockwise Rotation 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 rotation;
M05: Spindle stop.

Note: For the control timing and logic of M03, M04 and M05 defined by standard PLC, please refer to Chapter 3 *Installation and Connection* of this user manual.

2.1.15 Cooling Pump Control M08, M09

Code format: M08 or M8;
M09 or M9;

Code function: M08: Cooling pump on;
M09: Cooling pump off.

Note: For the control timing and logic of M08 and M09 defined by standard PLC, please refer to Chapter 3 *Installation and Connection* of this user manual.

2.1.16 Tailstock Control M10, M11

Code format: M10;
M11;

Code function: M10: Tailstock forward;
M11: Tailstock backward.

Note: For the control timing and logic of M10 and M11 defined by standard PLC, please refer to Chapter 3 *Installation and Connection* of this user manual.

2.1.17 Chuck Control M12, M13

Code format: M12;
M13;

Code function: M12: Chuck clamping;
M13: Chuck releasing.

Note: For the control timing and logic of M12 and M13 defined by standard PLC, please refer to Chapter 3 *Installation and Connection* of this user manual.

2.1.18 Spindle Position/Speed Control Switch 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 mode to speed control mode.

Note: For the control timing and logic of M14 and M15 defined by standard PLC, please refer to Chapter 3 *Installation and Connection* of this user manual.

2.1.19 Spindle Clamping/Releasing Control M20, M21

Code format: M20;
M21;

Code function: M20: Spindle clamping;
M21: Spindle releasing.

Note: For the control timing and logic of M20 and M21 defined by standard PLC, please refer to Chapter 3 *Installation and Connection* of this user manual.

2.1.20 Second Spindle Position/Speed Control Switch M24, M25

Code format: M24;
M25;

Code function: M24: Second spindle switches from speed control mode to position control mode;
M25: Second spindle switches from position control mode to speed control mode.

Note: For the control timing and logic of M24 and M25 defined by standard PLC, please refer to Chapter 3 *Installation and Connection* of this user manual.

2.1.21 Lubricating Fluid Control M32, M33

Code format: M32;
M33;

Code function: M32: Lubricating pump on;
M33: Lubricating pump off.

Note: For the control timing and logic of M32 and M33 defined by standard PLC, please refer to Chapter 3 *Installation and Connection* of this user manual.

2.1.22 Spindle Automatic Gear Shifting M41, M42, M43, M44

Code format: M4n; (n=1, 2, 3, 4)

Code function: When M4n is executed, the spindle shifts to the n^{th} gear

Note: For the control timing and logic of M41, M42, M43 and M44 defined by standard PLC, please refer to Chapter 3 Installation and Connection of this user manual.

2.1.23 Second Spindle Counterclockwise Rotation, Clockwise Rotation and Spindle Stop Control M63, M64 and M65

Code format: M63;
M64;
M65;

Code function: M63: Counterclockwise rotation;
M64: Clockwise rotation;
M65: Spindle stop.

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

Note 2: This function is effective only when the second spindle function is valid.

2.1.24 Synchronous Subroutine Call M70

Purpose: To solve the problem that the user does not know how to program PLC and needs the CNC system to perform backend action control.

Instruction: M70 P_

Description: P: program call number, range: 0~8, P0 is the default when P is not specified

P0	O9940
P1	O9941
P2	O9942
P3	O9943
...	...
P8	O9948

The codes that can be used in the subroutine are: G04, M35, M80, M81, M82, and M83.

Note: The last M70 action was not completed. When M70 is executed again, it is determined by system parameter 286:
0: Wait for the last M70 to be executed;
1: Alarm;
2: The current M70 action is queued, and the main program continues to execute

1. Synchronous instruction

There are 5 synchronous instructions: G4.5, G4.6, G4.7, G4.8, G4.9

Function: Through this instruction, the main program can be synchronized with the synchronous subroutine

```
O0001
G0 X0 Z0
M70 P0 // Starts the background program
G1 x100 z100
x5 Z-5
G4.5            // Synchronization
M30
```

```
O9940
M80 K1   // Output from port 1
G4 X1    // Delay 1 second
M81 K1   // Port 1 is closed
M80 K2   // Output from port 2
G4 X1    // Delay 1 second
M81 K2   // Port 2 is closed
G4.5     // Synchronization
M99
```

2. Reset and call subroutine

Press the reset key to call the O9949 subroutine.

Modify system parameter 285: 0-do not call; 1-manually press reset to call; 2-automatically run the program. Press reset after stopping the program first, and then call.

3. Backend instructions

(1) Backend linear interpolation G201 X(U)_ Y(V)_ Z(W)_ A(E)_ B_ C(H)_ F_

(2) Wait for the backend to end; Get the coordinates of each axis and cancel the backend mark G204

(3) End the backend program; Just set the speed to 0, the backend axis does not exit the backend task G205

(4) Adjust the backend speed G206 F_

(5) Fixed-length reciprocating control; G207 X(U)_ Y(V)_ Z(W)_ A(E)_ B_ C(H)_ F_ P_(0: Any position; 1: Origin position; 2: Specified position, macro variables 950-955, corresponding to XZYABC)

(6) Delay G209 J_ Delay time, unit: seconds;

(7) Force port switch G210 K_ I_

Description K: Port number

I: (0: Close 1: Open)

(8) Wait for input port G211 I_ J_

Description I: Port number; greater than 0: waiting for a signal; less than 0: waiting for no signal

J: waiting time, unit: milliseconds

(9) Automatic feeding G212 K_ I_ Q_ J_ P_ L_

Description K: Feeding output port; after feeding, K > 0: keep output, K < 0: turn off output

I: in-position signal input port (signed number, the sign is used to indicate the effective level)

Q: in-position signal holding time, seconds

J: Maximum waiting time for in-position signal, seconds

P: Delay between two feedings, seconds

L: number of repeated feedings

2.2 Spindle Functions

S code is used to control the spindle speed. There are two ways for the system to control the spindle speed: spindle speed switch control method: S □□ (2-digit code value) code is processed by PLC, PLC outputs switch signal to the machine tool to achieve step-by-step change of spindle speed.

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

2.2.1 Spindle Speed Switch Control

When parameter N0105 is set to 0, the spindle speed is controlled by switch quantity. A program segment can only have one S code. When two or more S codes appear in the program segment, the CNC will alarm.

When the S code and the code word for executing the move function are in the same segment, the execution order is defined by the PLC program. For details, please refer to the manual of the machine tool manufacturer.

When the spindle speed is controlled by switch quantity, the lathe CNC is used for machine tool control. The timing and logic of the S code execution should be based on the instructions of the machine tool manufacturer. The following is the S code defined by the standard PLC for reference only.

Code format: S_

Code description: S: 00~04 (leading zero can be omitted): 1~4 gear spindle speed switch control.
When CNC is reset, the output status of S01, S02, S03, and S04 remains unchanged.

When CNC is powered on, the output of S1~S4 is invalid. When any code among S01, S02, S03, and S04 is executed, the corresponding S signal output is valid and maintained, and the output of the other 3 S signals is canceled. When the S00 code is executed, the output of S1~S4 is canceled, and only one of S1~S4 is valid at the same time.

2.2.2 Spindle Speed Analog Voltage Control

When parameter N0105 is set to 1, the spindle speed is analog voltage controlled.

Code format: S_

Code description: S: 0000 ~ 9999 (leading 0 can be omitted): spindle speed analog voltage control

Code function: Set the spindle speed, CNC outputs 0V ~ 10V analog voltage to control the spindle servo or inverter, and realizes the stepless speed change of the spindle. The S code value is not memorized when power is off, and is set to 0 when power is on.

When the spindle speed analog voltage control function is valid, there are two ways to input the spindle speed: S code sets the fixed spindle speed (r/min), and the spindle speed remains constant when the S code value does not change, which is called constant speed control (G97 mode); S code sets the tangential speed of the tool relative to the outer circle of the workpiece (m/min), which is called constant linear speed control (G96 mode). In the constant linear speed control mode, the spindle speed during cutting feed changes with the absolute value of the absolute coordinate value of the X axis of the programmed trajectory.

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 current spindle gear's maximum spindle speed (output analog voltage is 10V) (corresponding to parameter N0112), and then output to the spindle servo or inverter to control the actual spindle speed to be consistent with the required speed.

When the CNC is powered on, the analog voltage output is 0V. After executing the S code, the output analog voltage value remains unchanged (unless it is in the cutting feed state of constant linear speed control and the absolute value of the absolute coordinate value of the X axis changes). After executing S0, the analog voltage output is 0V. When the CNC is reset or stopped urgently, the analog voltage output remains unchanged.

2.2.3 Constant Linear Speed Control G96, Constant Rotation Speed Control G97

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

Code function: constant linear speed control is effective, given cutting linear speed (m/min), constant speed control is canceled. G96 is a modal G code. If the current mode is G96, there is no need to enter G96.

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

Code function: cancelling constant linear speed control, making constant speed control effective, and setting spindle speed (r/min). G97 is a modal G code. If the current mode is G97, there is no need to enter G97.

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

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

G96 and G97 are modal code words of the same group, and only one can be valid. G97 is the initial state code word. When the CNC is powered on, G97 is valid by default. When a lathe is turning a workpiece, the workpiece usually rotates with the spindle axis as the center line. The cutting point of the tool cutting the workpiece can be regarded as a circular motion around the spindle axis. The instantaneous rate in the tangent direction of the circle is called the cutting linear speed (usually referred to as linear speed). Workpieces of different materials and tools of different materials require different linear speeds.

The constant linear speed control function is only effective when the spindle speed analog voltage control function is effective. In constant linear speed control, the spindle speed changes with the absolute value of the absolute coordinate value of the X-axis of the programmed trajectory (ignoring tool length compensation). The absolute value of the absolute coordinate value of the X-axis increases, the spindle speed decreases, the absolute value of the absolute coordinate value of the X-axis decreases, and the spindle speed increases, so that the cutting linear speed remains at the S code value. Cutting workpieces using the constant linear speed control function can keep the surface finish of workpieces with varying diameters consistent.

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

Spindle speed: r/min

|X|: Absolute value of X-axis absolute coordinate value (diameter value), mm

π ≈ 3.14

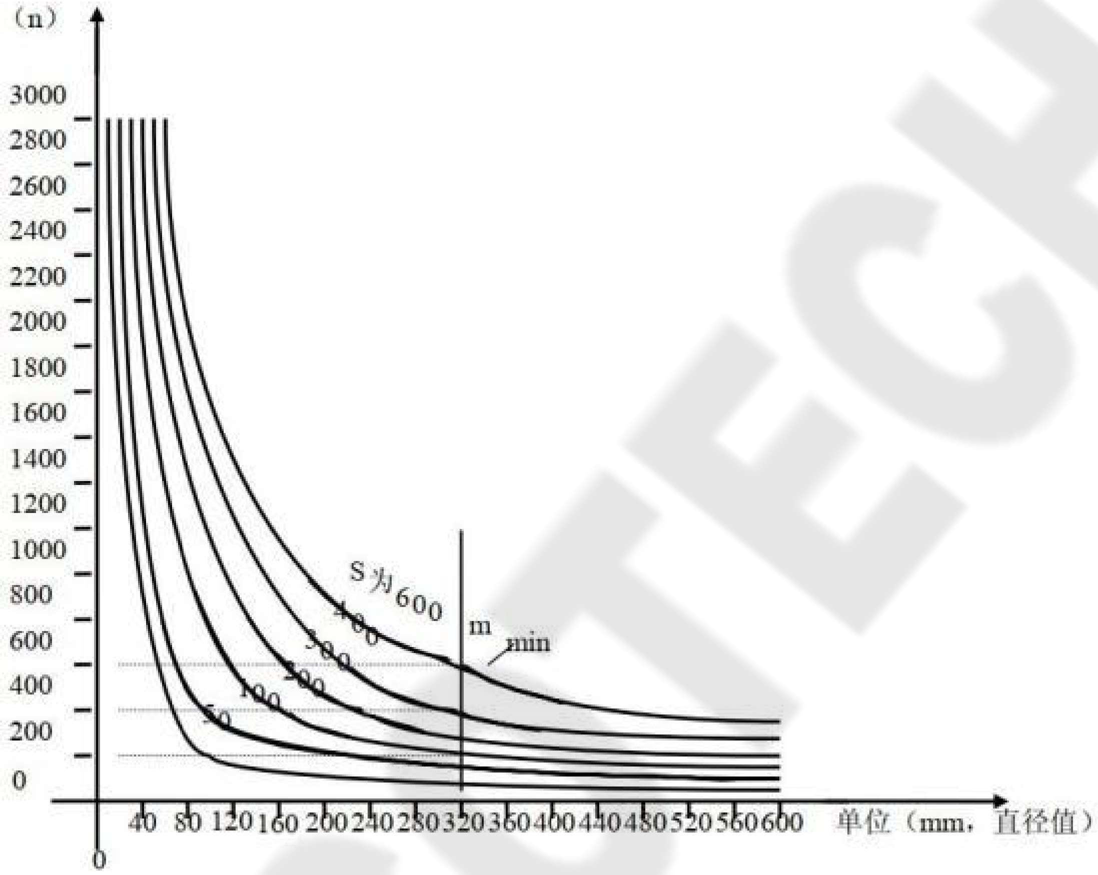


Figure 2-4

When constant linear speed control is used, the spindle speed is changed only during the cutting feed (interpolation) process as the absolute value of the X-axis absolute coordinate value of the programmed trajectory changes. For G00 quick movement, since no actual cutting is performed, the spindle speed remains unchanged during the execution of G00. The spindle speed at this time is calculated according to the linear speed at the end position of the program segment.

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

When constant linear speed control is effective, G50 S_ can limit the maximum spindle speed (r/min). When the spindle speed calculated according to the linear speed and the X-axis coordinate value is higher than the spindle maximum speed limit value set by G50 S_, the actual spindle speed is the spindle maximum speed limit value. When the CNC is powered on, the maximum spindle speed limit value is not set and the maximum spindle speed limit function is invalid. The maximum speed limit value defined by G50 S_ is retained before re-assignment. The maximum speed limit function is valid in G96 state. In G97 state, the maximum spindle speed set by G50 S_ does not have a limiting effect, but the maximum spindle speed limit value is still retained.

Special note: when parameter N0124 (the minimum spindle speed in constant linear speed control) is set to 0, if G50 S0 is executed, the spindle speed will be limited to 0r/min (the spindle will not rotate) in constant linear speed control.

CNC parameter N0124 is the lower limit of the spindle speed in constant linear speed control. When the spindle speed calculated by the linear speed and the X-axis coordinate value is lower than this value, the actual spindle speed is limited to the lower limit of the spindle speed.

Example:

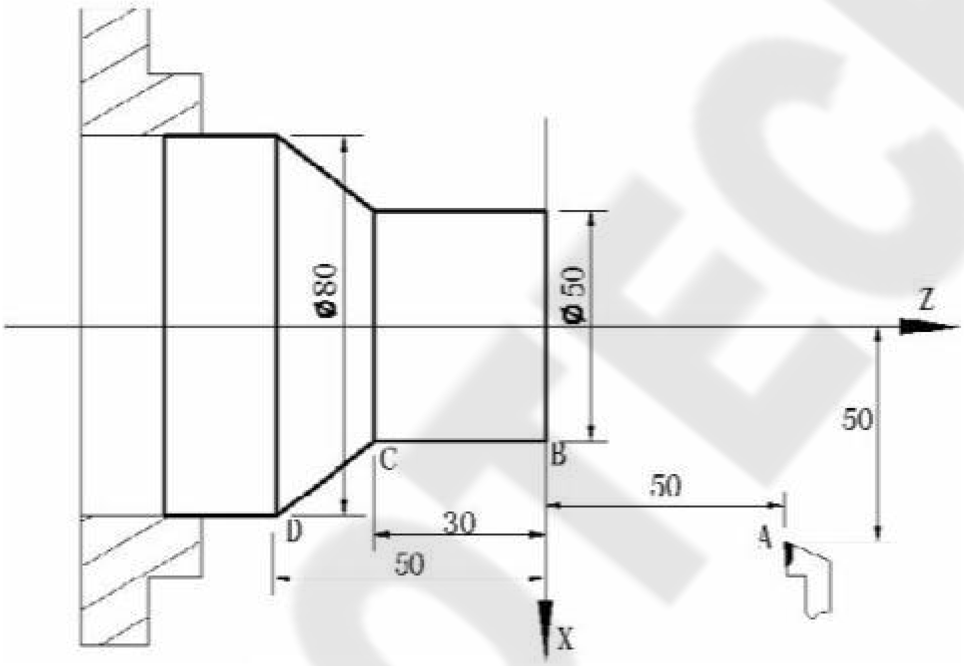


Figure 2-5

O0001;	(Program name)
N0010 M3 G96 S300;	(Spindle rotation, constant linear speed control is effective, linear speed is 300m/min)
N0020 G0 X100 Z50;	(Quickly move to point A, spindle speed is 955r/min during the movement)
N0030 G0 X50 Z0;	(Quickly move to point B, spindle speed is 1910r/min during the movement)
N0040 G1 W-30 F200;	(Cutting from point B to point C, spindle speed is constant at 1910r/min during cutting)
N0050 X80 W-20 F150;	(Cutting from point C to point D, spindle speed changes linearly from 1910r/min to 1194r/min)
N0060 G0 X100 Z50;	(Quickly return to point A, spindle speed is 955r/min during the movement)
N0110 M30;	(The program ends, spindle and coolant are turned off)
N0120 %	

Note 1: In G96 state, the instructed S value is maintained even in the G97 state. When returning to G96 state, its value is restored;

For example: G96 S50; (cutting linear speed 50m/min)

G97 S1000; (spindle speed 1000r/min)

G96 X3000; (cutting linear speed 50m/min)

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

Note 3: When cutting threads, the constant linear speed control function is also valid, but in order to ensure the thread processing accuracy, do not use constant linear speed control when cutting threads, and cut threads in G97 state;

Note 4: When changing from G96 state to G97 state, if there is no S code (r/min) in the G97 program segment, the last speed of G96 state is used as the S code of G97 state, that is, the spindle speed remains unchanged at this time;

Note 5: When the spindle speed calculated by the cutting linear speed is higher than the maximum speed of the current spindle gear (CNC parameter N0112) during constant linear speed control, the spindle speed is limited to the maximum speed of the current spindle gear.

2.2.4 Spindle Override

When the spindle speed analog voltage control mode is valid, the actual spindle speed can be adjusted with the spindle override. The actual speed after the spindle override is limited by the maximum speed of the current spindle gear. In the constant linear speed control mode, it is also limited by the minimum spindle speed limit value and the maximum spindle speed limit value.

NC provides 8 levels of spindle override (50%~120%, each level changes by 10%). The actual number of levels and adjustment methods of the spindle override are defined by the PLC ladder diagram. When using, the instructions of the machine tool manufacturer shall prevail. The following is the functional description of the SZGH880T/SZGH1080T series standard PLC ladder diagram for reference only.

The SZGH880T/SZGH1080T series standard PLC ladder diagram defines a total of 8 levels of spindle override. The actual spindle speed can be adjusted in real time within the range of 50%~120% of the instruction speed using the spindle override adjustment key, and the spindle override is memorized when the power is off.

2.2.5 Multi-spindle Control Function

The SZGH880T/SZGH1080T series can control up to two analog spindles. An S code is used to instruct any of these spindles. The selection of which spindle is determined by the signal from the PLC, and each has a gear shift function.

Since the SZGH880T/SZGH1080T series has only one spindle encoder interface, the second spindle has no encoder feedback and the spindle speed is not displayed.

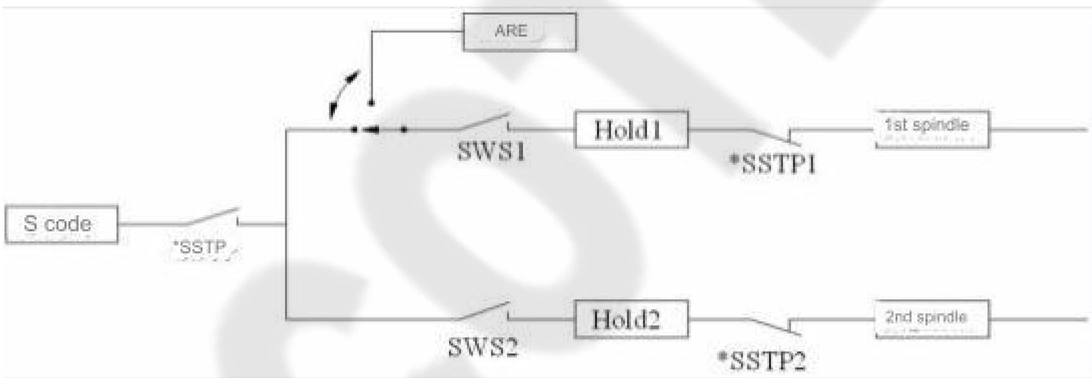
The S code is sent as a speed instruction to the spindle selected by the spindle selection signal (SWS1, SWS2<G25＃0, G25＃1>), and each spindle rotates at the specified speed. If a spindle does not receive a spindle selection signal, it will continue to rotate at the previous speed. This allows each spindle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

There are several forms of spindle control:

● Multi-spindle control type A

When the first spindle is selected by the SWS1 signal, the SIND signal is used to determine whether the spindle analog voltage is controlled by the PLC or CNC, and the R01I to R12I signals are used to set the spindle analog voltage. These signals do not affect the second spindle.

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

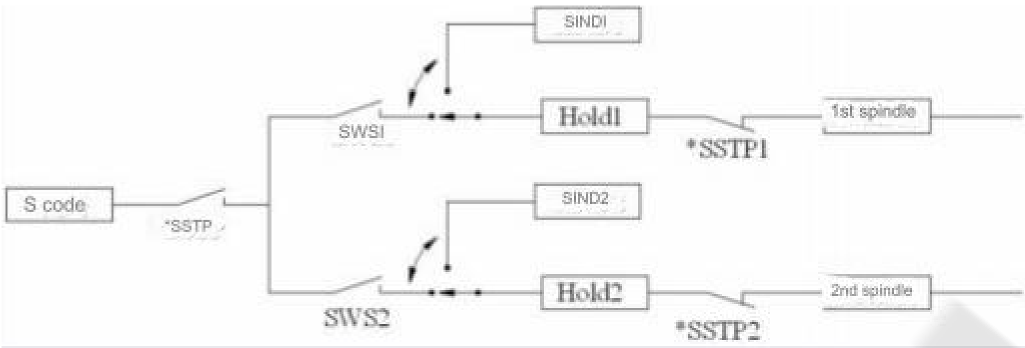


●Multi-spindle control type B

Each spindle has an independent SIND signal.

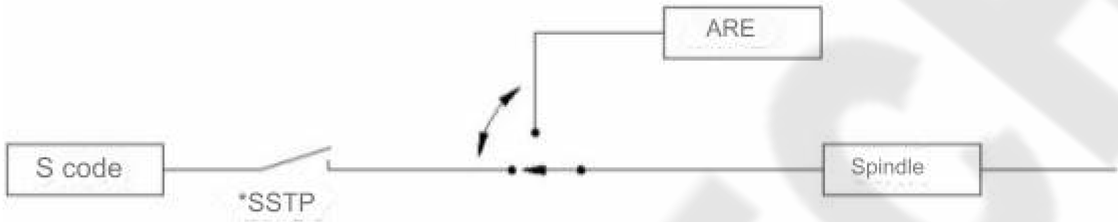
When the spindle selection signal and the SIND signal of the first or second spindle are set to "1", the SIND signal determines whether each spindle is controlled by the PLC or CNC.

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



●Multi-spindle control function is invalid

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



2.2.6 Cs Contour Control Function

The situation of controlling the spindle speed is called spindle rotation control (rotating the spindle through speed instructions), and the situation of controlling the spindle position is called spindle contour control (rotating the spindle through movement instructions). The function of contour control for this spindle is the Cs contour control function. The spindle works as a servo feed axis, rotates and locates through position movement instructions, and can interpolate with other feed axes to process contour curves.

2.3 Tool Function

2.3.1 Tool Control

The tool functions (T code) of the controller are the following two: automatic tool change and tool offset execution. The control logic of automatic tool change is handled by the PLC ladder diagram, and the execution of tool offset is handled by the NC.

Code format:

T □□ ○○

Tool offset number (00-32, leading 0 cannot be omitted)

Target tool number (01-32, leading 0 cannot be omitted)

Code function: Automatically change the tool holder to the tool position of the target tool number, and execute the tool offset according to the tool offset number of the code. The tool offset number can be the same as the tool number or different, that is, one tool can correspond to multiple offset numbers. After executing the tool offset, execute T □□ 00 again, the CNC will reverse the offset according to the current tool offset, and the

CNC will change from the executed tool offset state to the uncompensated state. This process is called canceling tool offset. When power is on, the tool number and tool offset number displayed by the T code are the states before power-off. There can only be one T code in a program segment. When two or more T codes appear in the program segment, the CNC will alarm.

Before machining, the position offset data (called tool offset) of each tool is obtained through tool setting operation. After the T code is executed during program running, the tool offset is automatically executed. In this way, when editing the program, each tool is written according to the size of the part drawing, and the position relationship between each tool in the machine tool coordinate system does not need to be considered. If the machining size deviates due to tool wear, the tool offset can be modified according to the size deviation.

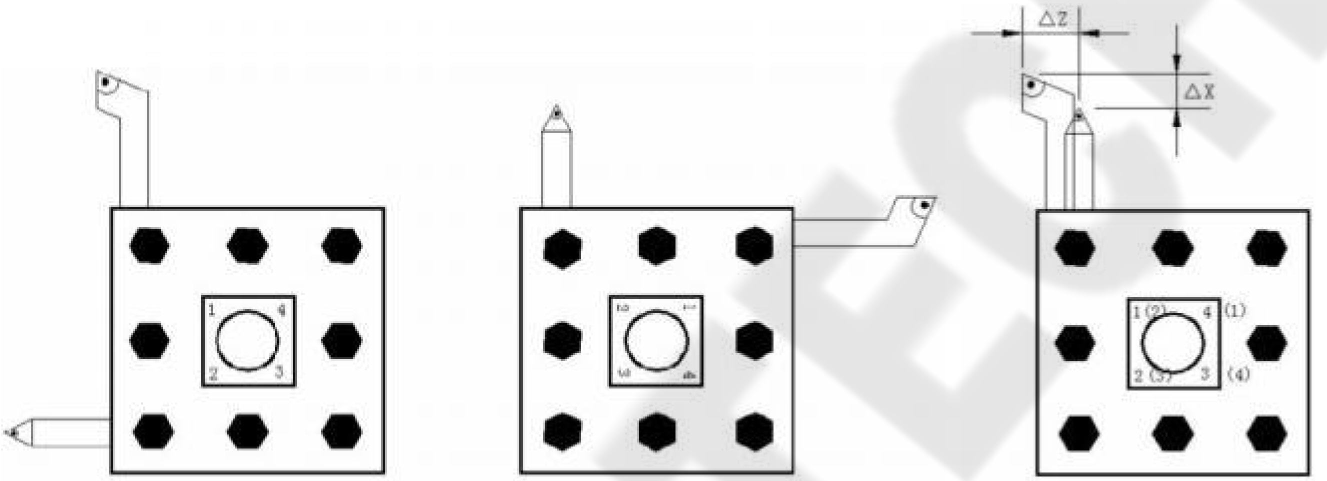


Figure 2-4 Tool offset

The tool offset is related to the programming trajectory. The offset corresponding to the tool offset number in the T code is added or subtracted by the compensation amount at the end of each program segment. Whether the X-axis tool offset is expressed in diameter value or radius value is set by system parameter N0001. The meaning of the X-axis tool offset value being expressed in diameter value/radius value is that when the tool length compensation value changes, the outer diameter of the workpiece changes in diameter value/radius value.

When the T code and the code for executing the moving function are in the same program segment and the tool offset is executed in the coordinate modification mode, the moving code and the T code are executed simultaneously. When the tool is changed, the current tool offset is superimposed on the coordinate movement value of the moving code and executed together. The moving speed is determined by the moving code as cutting feed or rapid moving speed.

When the T code and the code for executing the moving function are in the same program segment and the tool offset is executed in the tool movement mode, the moving code and the T code are executed separately. The tool change and tool offset are executed first, and then the moving function code is executed. The speed of tool offset execution is the current rapid moving speed.

After executing any of the following operations, the tool offset will be canceled:

1. Executing the T □□ 00 code;

2. Executing the G28 code or manually returning to the machine zero (only canceling the tool offset of the coordinate axis that has returned to the machine zero, and not canceling the tool offset of the other coordinate axis that has not returned to the machine zero).

When the system parameter N0126 (total tool position selection) is not set to 1 (2~32), and the target tool number is not equal to the currently displayed tool number, after the T code is instructed, the control timing and logic of the tool holder are determined by the PLC ladder diagram. When using it, the instructions of the machine tool manufacturer shall prevail.

Chapter 3 G Code

3.1 Overview

The G code consists of the code address G and the subsequent 1~2 code values. It is used to specify the movement mode of the tool relative to the workpiece, coordinate setting and other operations. The G code list is shown in Table 3-1.

G codes are divided into group 00, 01, 02, 03, 05, 06, 07, 16, and 21. Except that group 01 and group 00 codes cannot be in the same segment, G codes of several different groups can be instructed in the same program segment. In principle, two or more G codes of the same group cannot be instructed in the same program segment. If the codes of the same group are instructed in the same segment without alarm, the last G code is valid. G codes of different groups without common parameters (code words) can be in the same program segment, and the functions are valid at the same time and are not related to the order. If G codes other than those in Table 3-1 or G codes of optional functions are used, the system will alarm.

Table 3-1 List of G code words

Instruction word	Group	Function	Remarks
G00	01	Quick movement	Initial G code
G01		Linear interpolation	Modal G code
G02		Circular interpolation (clockwise)	
G03		Circular interpolation (counterclockwise)	
G32		Thread cutting	
G33		Z axis tapping cycle	
G34		Variable pitch thread cutting	
G90		Axial cutting cycle	
G92		Thread cutting cycle	
G94		Radial cutting cycle	
G04	00	Pause, pre-stop	Non-modal G code
G10		Wear compensation	
G12		Stored stroke detection function on	
G13		Stored stroke detection function off	
G27		Return to reference point detection	
G28		Return to first reference point of machine tool	
G29		Automatically return from reference point	
G30		Return to second, third, and fourth reference points of machine tool	
G31		Jump interpolation	

G50		Coordinate system setting	
G65		Macro code	
G70		Finishing cycle	
G71		Axial roughing cycle	
G72		Radial roughing cycle	
G73		Closed cutting cycle	
G74		Axial roughing cycle	
G75		Radial roughing cycle	
G76		Multiple thread cutting cycle	
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 linear velocity on	Modal G code
G97		Constant linear velocity off	Initial G code
G98	03	Feed per minute	Initial G code
G99		Feed per revolution	Modal G code
G40	07	Cancel tool nose radius compensation	Initial G code
G41		Tool nose radius left compensation	Modal G code
G42		Tool nose radius right compensation	
G17	16	XY plane	Modal G code
G18		ZX plane	Initial G code
G19		YZ plane	Modal G code

3.1.1 Modal, Non-modal and Initial State

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

After a G code is executed, the function or state defined by it remains valid until it is changed by other G codes in the same group. This type of G code is called a modal G code. After a modal G code is executed, before the function or state defined by it is changed, the G code does not need to be entered again when the subsequent program segment executes the G code word.

After the G code is executed, the function or state defined by it is valid once. Each time the G code is executed, the G code must be re-entered. This type of G code is called a non-modal G code.

After the system is powered on, the modal G code that is valid without executing its function or state is called the initial state G code. When the G code is not entered after power-on, it 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 characteristic of retaining values after execution. If these code words are already included in the previous program segment, they do not need to be entered when they are needed in the subsequent program segment and have the same value and meaning.

Table 3-2

Programming address	Functional meaning	Initial value at power-on
U	Cutting depth in G71	N0187 parameter value
U	X-axis retraction distance in G73	N0189 parameter value
W	Cutting depth in G72	N0187 parameter value
W	Z-axis retraction distance in G73	N0190 parameter value
R	Cycle retraction amount in G71 and G72	N0188 parameter value
R	Number of roughing cycles in G73	N0191 parameter value
R	Retraction amount after cutting in G74 and G75	N0192 parameter value
R	Finishing allowance in G76	N0196 parameter value
R	Taper in G90, G92, G94 and G76	0
(G98)F	Feed speed (G98)	N0027 parameter value
(G99)F	Revolution feed rate (G99)	0
F	Metric thread pitch (G32, G92, G76)	0
I	Inch thread pitch (G32, G92, G76)	0
S	Spindle rotation speed specification (G97)	0
S	Spindle linear speed specification (G96)	0
S	Spindle rotation speed switch output	0
P	Number of thread cutting finishing operations in G76; Thread cutting thread retraction width in G76 Thread cutting tool nose angle in G76;	N0193 parameter value N0183 parameter value N0194 parameter value
Q	Minimum cutting amount in G76	NO159 parameter value

Note 1: Programming addresses with multiple functions (such as F, which can be used to specify feed per minute, feed per revolution, metric thread pitch, etc.) are only allowed to be omitted when the code word is executed and the same function definition code word is executed again. For example: If G98 F_ is executed, but the G code for thread interpolation is not executed, the pitch must be specified with the F code when metric thread machining is performed;

Note 2: When the address X/U, Z/W is used to give the coordinates of the end point of the program segment, the input can be omitted. When X/U or Z/W is not entered in the program segment, the system takes the current absolute coordinates of the X axis or Z axis as the coordinate value of the end point of the program segment;

Note 3: When using a programming address 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; modal code word G0 is valid)
X20 Z30;	(Quickly move to X20 Z30; modal code word G0 can be omitted)
G1 X50 Z50 F300;	(Linear interpolation to X50 Z50, feed speed 300mm/min; modal code word G1 is valid)
X100;	(Linear interpolation to X100 Z50, feed speed 300mm/min; Z axis coordinate is not entered, take the current coordinate value Z50; F300 is maintained, G01 is a modal code word and can be omitted)
G0 X0 Z0;	(Quickly move to X0 Z0, modal code word G0 is valid)
M30;	

3.1.3 Related Definitions

In the following contents of this manual, the meanings of the relevant words (characters) are as follows unless otherwise specified:

Starting point: the position before the current program segment is executed;

Ending point: the position after the current program segment is executed;

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

U: the difference between the absolute coordinate of the X axis of the end position and that of the start position;

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

W: the difference between the absolute coordinate of the Z axis of the end position and that of the start position;

F: cutting feed rate.

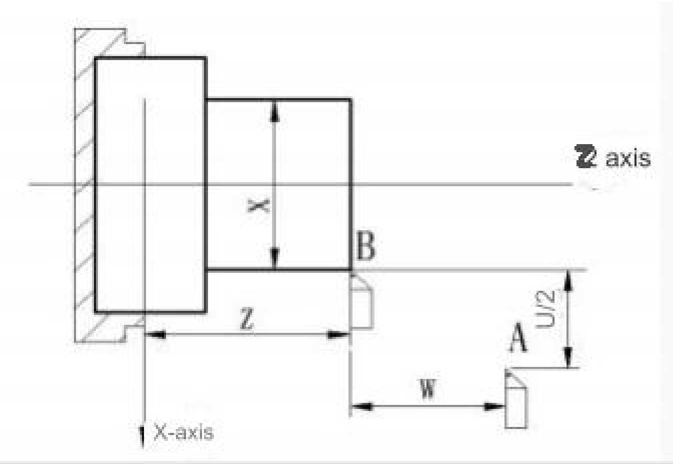
3.2 Quick Positioning G00

Code format: G00 X/U_ Z/W_;

Code function: The X axis and Z axis move from the start point to the end point at their respective rapid moving speeds at the same time, as shown in Figure 3-1. The two axes move at their own independent speeds. The short axis reaches the end point first, and the long axis moves the remaining distance independently. The composite trajectory is not necessarily a straight line.

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

Motion trajectory diagram:



Point C is the middle point of the quick movement from point A to point B

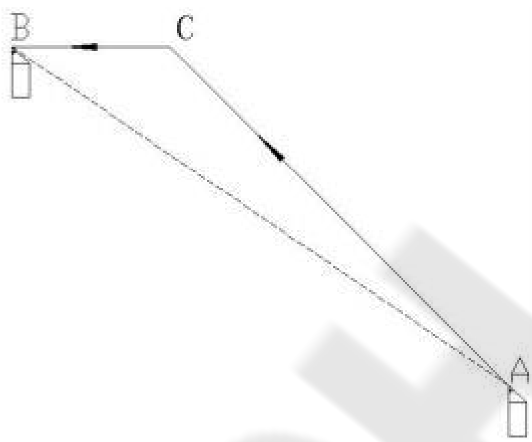


Figure 3-1

The quick movement speed of the X and Z axes is set by the system data parameter N0021 respectively, and the actual movement speed can be adjusted through the quick rate key on the machine panel.

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

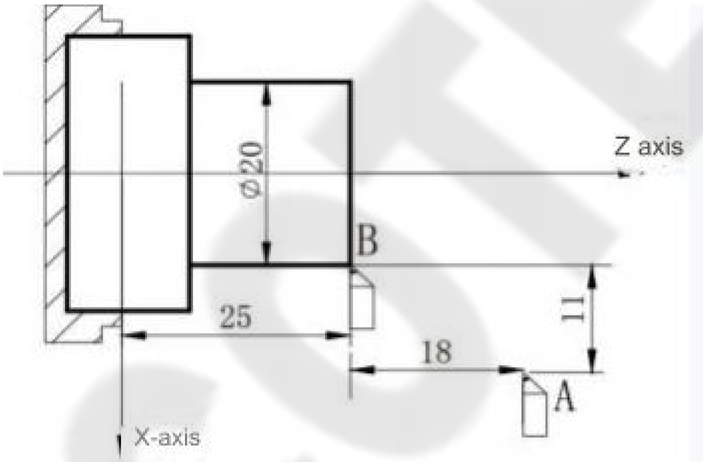


Figure 3-2

G0	X20 Z25;	(Absolute coordinate programming) or
G0	U-22 W-18;	(Relative coordinate programming) or
G0	X20 W-18;	(Mixed coordinate programming) or
G0	U-22 Z25;	(Mixed coordinate programming)

3.3 Linear Interpolation G01

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

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

Code description: G01 is a modal G code;

X/U, Z/W can be omitted one or all of them. When one is omitted, it means that the coordinate values of the start point and the end point of the axis are the same; omitting simultaneously means that the end point and the start point are at the same position.

The F code value is the vector composite speed of the instantaneous speed in the X-axis direction and the Z-axis direction. The actual cutting feed speed is the product of the feed rate and the F code value;

After the F code value is executed, this code value is maintained until a new F code value is executed. When the F code word used by other G codes described later has the same function, it will not be described in detail. The value range is shown in Table 1-10.

Note: In G98 state, the maximum value of F does not exceed the setting value of data parameter NO027 (cutting feed upper limit speed).

Motion trajectory diagram:

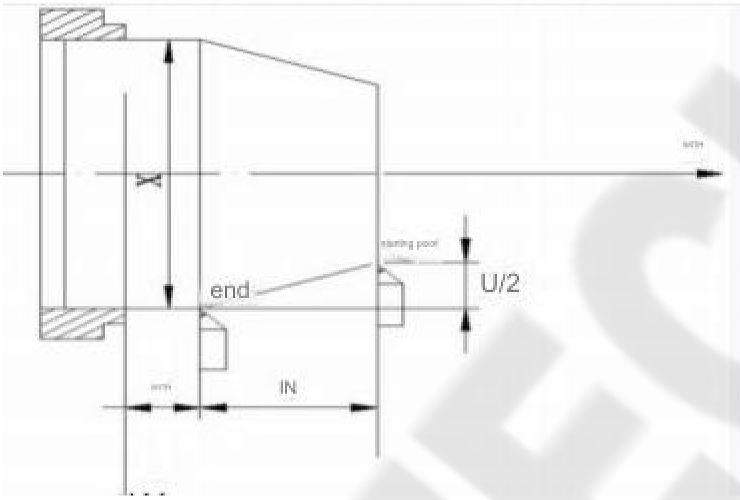
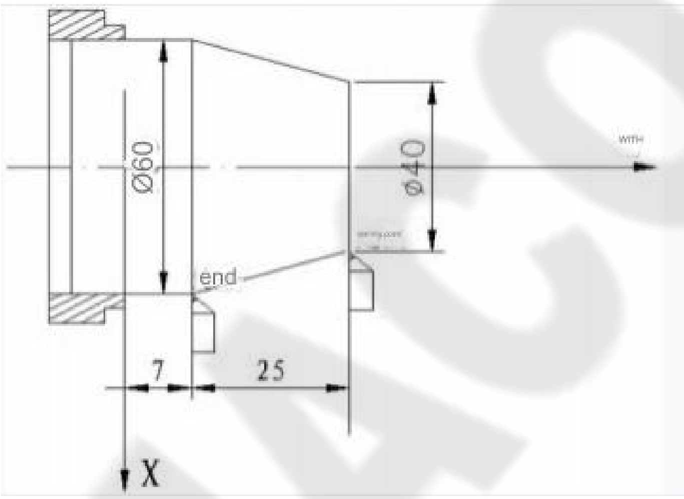


Figure 3-3

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



Program:

G01 X60 Z7 F500; (Absolute value programming)

G01 U20 W-25; (Relative value programming)

G01 X60 W-25; (Mixed programming)

G01 U20 Z7; (Mixed programming)

3.4 Circular (Helical) Interpolation G02, G03

A. Circular interpolation

Code format:

$$\left.\begin{matrix} \text{G02} \\ \\ \text{G03} \end{matrix}\right\} \text{X/U\_} \quad \text{Z/W\_} \left\{\begin{matrix} \text{R} \\ \\ \text{I\_ K\_} \end{matrix}\right.$$

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

Code trajectory diagram:

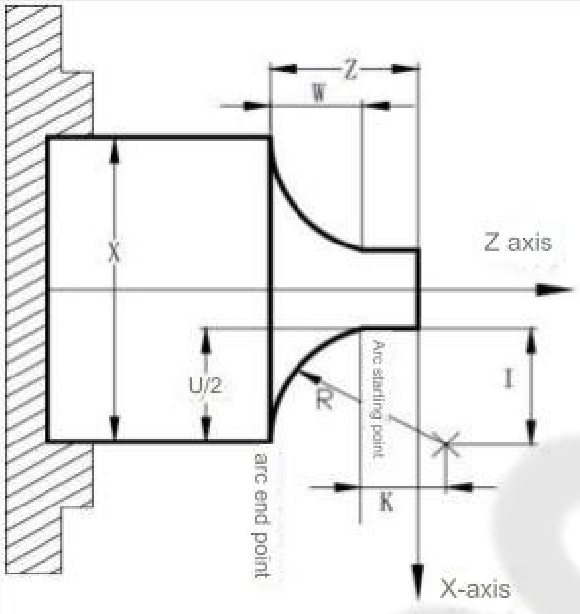


Figure 3-5 G02 trajectory diagram

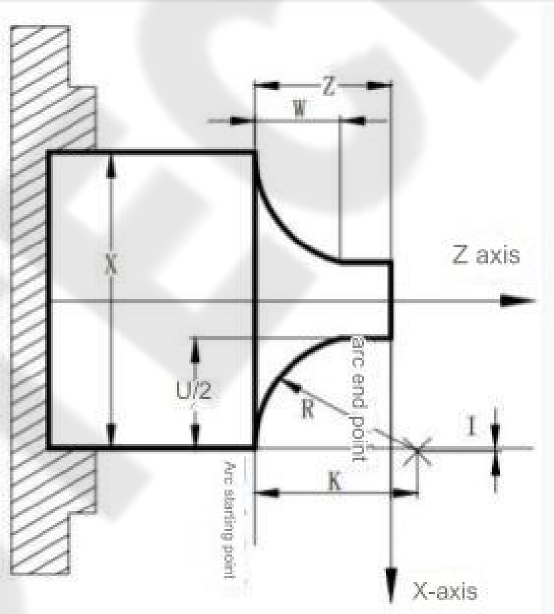


Figure 3-6 G03 trajectory diagram

Code description: G02 and G03 are modal G codes;

R is the arc radius;

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

K is the difference between the center of the circle and the start point of the arc in the Z direction. When the arc center is specified by the addresses I and K, they correspond to the X and Z axes respectively. I and K represent the vector components from the start point of the arc to the center of the circle, which are incremental values; as shown in Figure 3-6-1.

I = Circle center coordinate X - X coordinate of arc start point; K = Circle center coordinate Z - Z coordinate of arc start point;

I and K have signs according to the direction. If the direction of I and K is the same as the direction of X and Z axis, then it takes a positive value; otherwise, it takes a negative value.

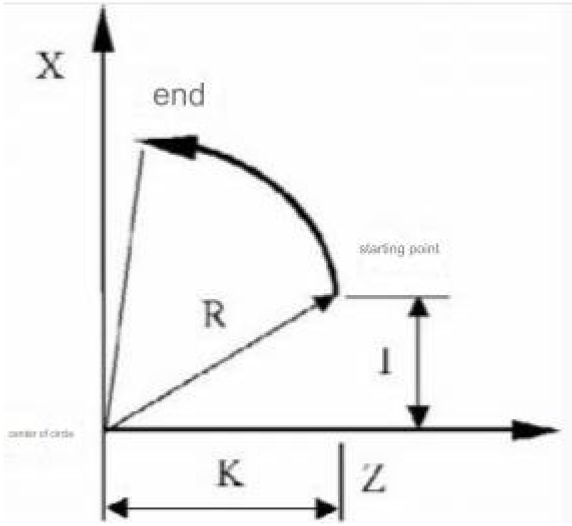
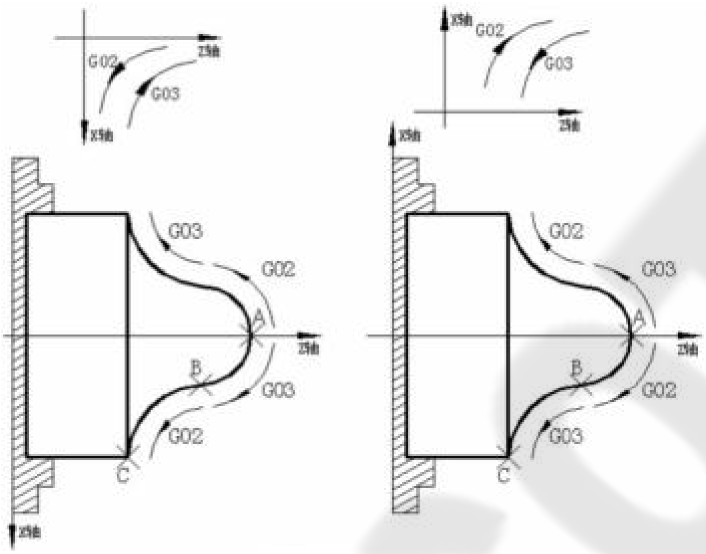


Figure 3-6-1

Arc direction: The direction of the G02/G03 arc is defined in the opposite direction in the front tool holder coordinate system and the rear tool holder coordinate system, as shown in Figure 3-7:



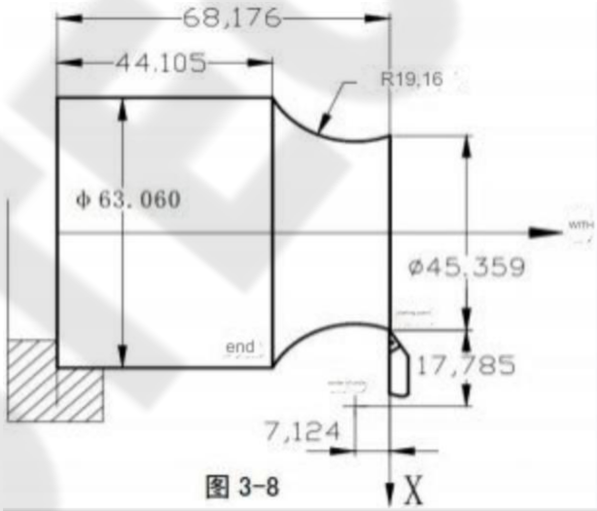


Figure 3-7 Figure 3-8

Notes:

●When I=0 or K=0, it can be omitted; but at least one of the addresses I, K or R must be entered, or the system will generate an alarm;

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

●The R value must be equal to or greater than half of the distance from the start point to the end point. If the end point is not on the arc defined by R, the system will generate an alarm;

●One or all of the addresses X/U, Z/W can be omitted; when one is omitted, it means that the start point and end point of the omitted axis are the same; when omitting simultaneously, it means that the end point and the start point are at the same position. If I and K are used to specify the center of the circle, the trajectory of executing the G02/G03 code is a full circle (360°); when R is used to specify, it means a circle of 0°;

●If I and K are used for programming and the distance from the center of the circle to the end point of the arc is not equal to R (R = $\sqrt{I^2+K^2}$), the system will automatically adjust the center position to ensure that the start and end points of the arc motion are consistent with the specified value. If the distance between the start and end points of the arc is greater than 2R, the system will alarm.

●When R is specified, it is an arc less than 360°, when R is negative, it is an arc greater than 180°, and when R is positive, it is an arc less than or equal to 180°;

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

Program:

G02 X63.060 Z-24.071 R19.16 F300 ; or

G02 U17.701 W-24.071 R19.16 F300 ; or

G02 X63.060 Z-24.071 I17.785 K-7.124 ; or

G02 U17.701 W-24.071 I17.785 K-7.124 F300

G02/G03 code comprehensive programming example:

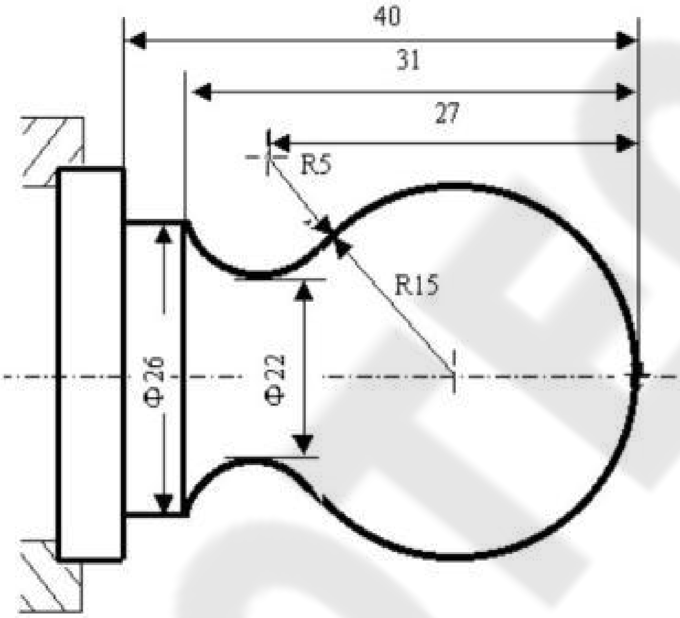


Figure 3-9 Arc programming example

Program: O0001

N001 G0 X40 Z5;	(Quick positioning)
N002 M03 S200;	(Spindle on)
N003 G01 X0 Z0 F900;	(Approaching the workpiece)
N005 G03 U24 W-24 R15;	(Cutting R15 arc segment)
N006 G02 X26 Z-31 R5;	(Cutting R5 arc segment)
N007 G01 Z-40;	(Cutting ф26)
N008 X40 Z5;	(Return to start point)
N009 M30;	(Program end)

B. Helical interpolation

Code format: G02/G03 X/U_ Z/W_ I_K Q_/P_

Code description:

X/U_ Z/W_: Coordinates of the end point of the spiral line;

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

K is the difference between the center of the circle and the start point of the arc in the Z direction.

Q_: pitch between spiral lines;
P_: number of spiral coils;

Arc interpolation for any two axes, only radius R can be used, and the first axis letter in the instruction is the horizontal axis

Example: G2.1 Y_ A_ R_ G3.1 A_ Y_ R_

3.5 Three-point Circular Interpolation G05

Code format: G05 X(U)__ Z(W)__ I__ K__ F__

Code function: If the center and radius of the arc are unknown, but the coordinates of the three points on the arc contour are known, the G5 function can be used; the arc direction is determined by the position of the middle point between the start point and the end point;

Code description: G05 is a modal G code;

I: the relative coordinate value (X direction) of the middle point passed by the arc relative to the start point (radius value, with direction);

K: the relative coordinate value (Z direction, with direction) of the middle point passed by the arc relative to the start point as shown in Figure 3-10:

The value range of X, U, Z, W, I, K is shown in Table 1-2 of 1.4.1, and the unit is mm/inch

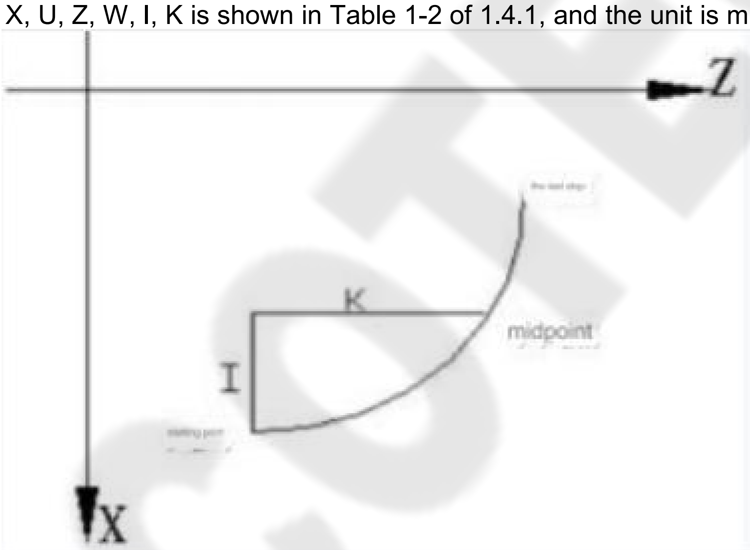


Figure 3-10

Notes:

● Middle point: refers to any point on the arc except the start point and the end point;

● When the three given points are collinear, the system generates an alarm;

● When I is omitted, it is considered that I=0, and when K is omitted, it is considered that K=0; when I and K are omitted at the same time, the system generates an alarm;

● The meaning of I and K is similar to the displacement values I and K of the center coordinates relative to the start point coordinates in the G02/G03 code;

● G05 cannot process a full circle;

Example: (assuming that a semicircle is processed)

Program:

G0 X10 Z10

G05 X30 Z10 I5 K-5

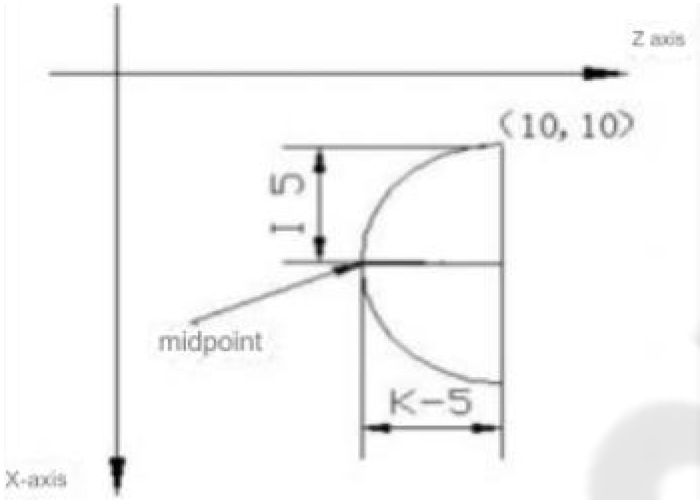


Figure 3-10-1

3.5.1 Ellipse interpolation G6.2, G6.3

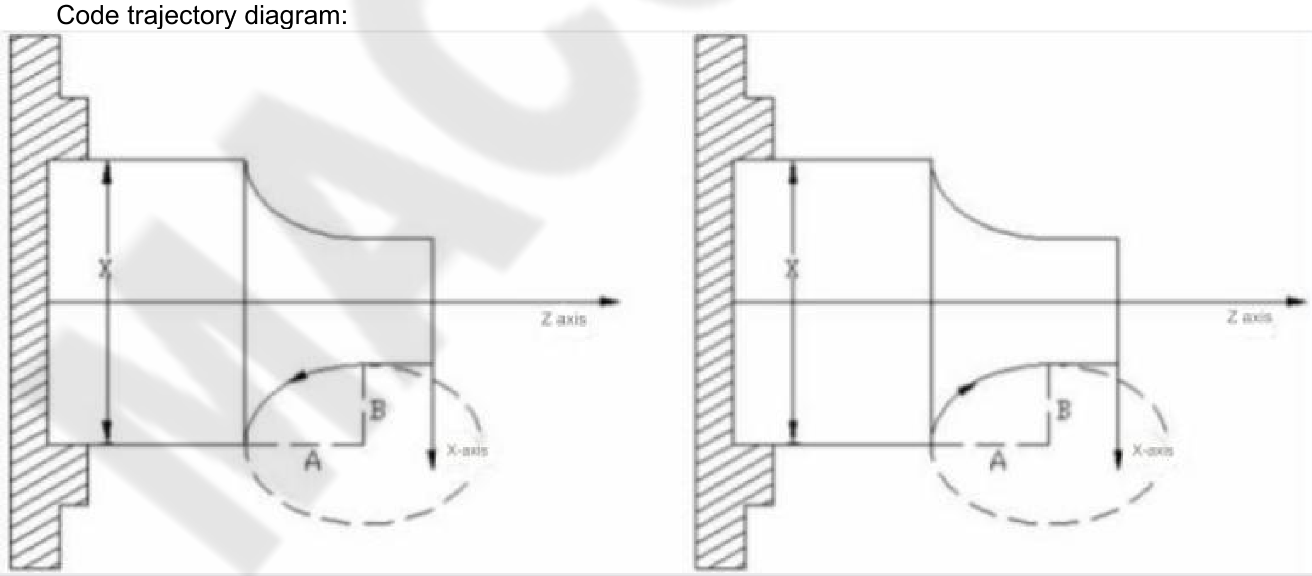
Code format: G6.2 / G6.3 } X(U)_ Z(W)_ A_ B_ Q_

Code function: G6.2\6.3 is a modal G code;

The motion trajectory of the G6.2 code is a clockwise (rear tool holder coordinate system)/counterclockwise (front tool holder coordinate system) ellipse from the start point to the end point.

The motion trajectory of the G6.3 code is a counterclockwise (rear tool holder coordinate system)/clockwise (front tool holder coordinate system) ellipse from the start point to the end point.

Code trajectory diagram:



G6.2 code trajectory diagram G6.3 code trajectory diagram

Code description:

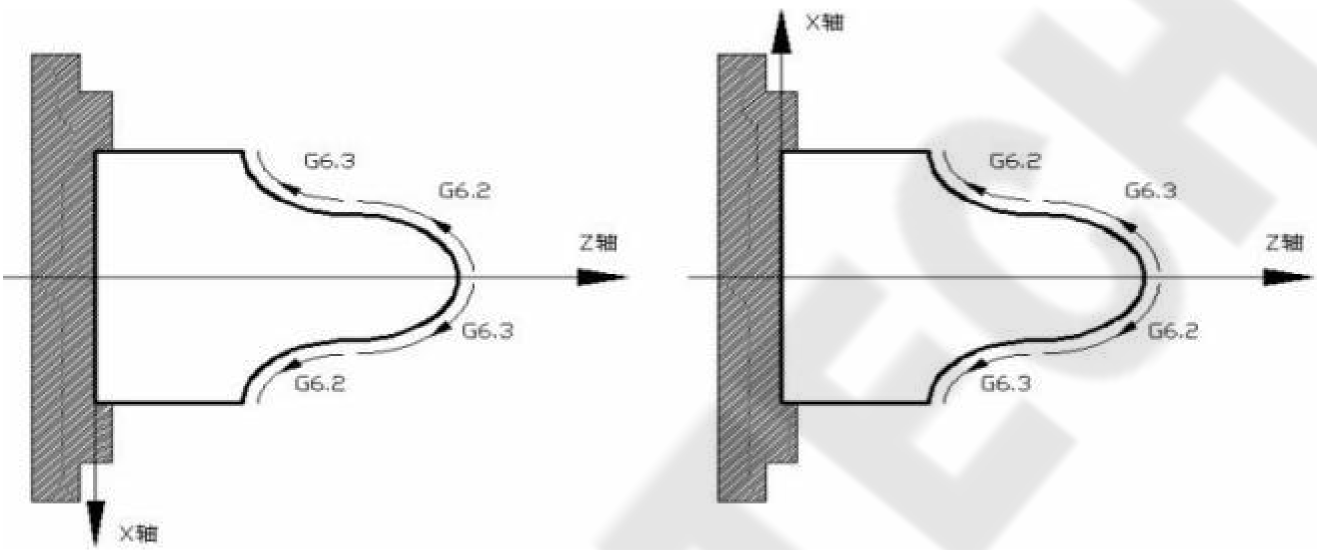
A: Major semi-axis length of ellipse (0<A≤ 99999999×least input increment, unsigned, unit: mm/inch);

B: Minor semi-axis length of ellipse (0<B≤ 99999999×least input increment, unsigned, unit: mm/inch);

Q: The angle between the major axis of the ellipse and the Z axis of the coordinate system

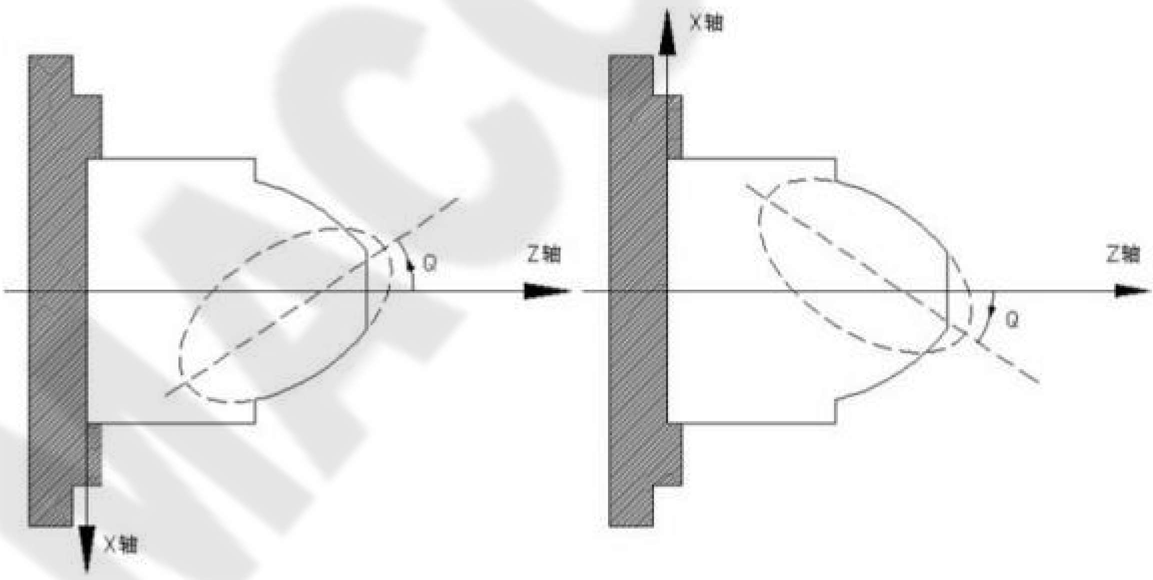
(counterclockwise 0-99999999, unit: 0.001 degree, unsigned, angle modulo 180).

Ellipse direction: The definition of G6.2/G6.3 direction is opposite in the front tool holder coordinate system and the rear tool holder coordinate system.



Front tool holder coordinate system Rear tool holder coordinate system

Q value description: Q value refers to the angle when the positive direction of the Z axis rotates clockwise to coincide with the major axis of the ellipse when looking down at the XZ plane from the positive direction of the Y axis in the right-handed rectangular Cartesian coordinate system, as shown in the figure below:



Front tool holder coordinate system Rear tool holder coordinate system

Notes:

● A and B are non-modal parameters. If not entered, the default value is 0. When A = 0 or B = 0, the system generates an alarm; when A = B, it is processed as an arc (G02/G03);

● Q value is a non-modal parameter and must be specified each time when it is used. When omitted, the default value is 0 degree, and the major axis is parallel to or coincides with the Z axis;

The unit of Q is 0.001 degree. If the angle with the Z axis is 180 degree, enter Q180000 in the program. If Q180 or Q180.0 is entered, it is considered to be 0.18 degree;

● If the distance between the programmed start and end points is greater than the major axis length, the system will generate an alarm;

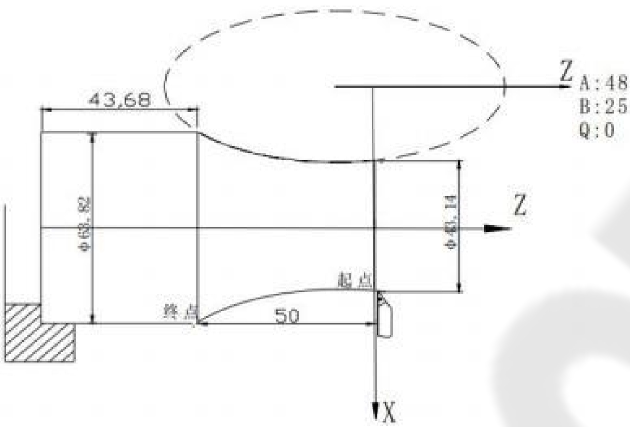
● One or all of the addresses X(U) and Z(W) can be omitted; when one is omitted, it means that the omitted start and end points of the axis are the same; omitting simultaneously means that the end point and the start point are at the same position and will not be processed;

● Only ellipses less than 180° (including 180°) are processed;

● G6.2 and G6.3 codes can be used in compound cycles G70-G73, and the notes are the same as G02 and G03;

● G6.2 and G6.3 codes can be used in C tool compensation, and the notes are the same as G02 and G03;

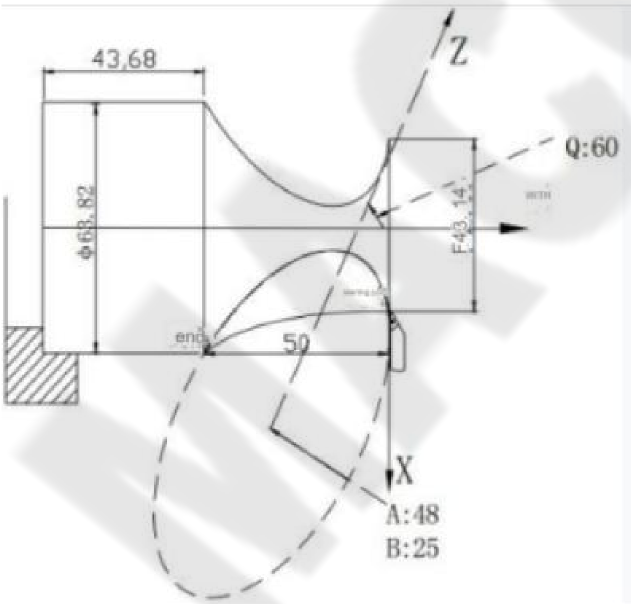
Example: Cutting an ellipse from a diameter of Φ43.14 to Φ63.82



Program:

G6.2 X63.82 Z-50.0 A48 B25 Q0; or
G6.2 U20.68 W-50.0 A48 B25;

Example: Cutting an ellipse from a diameter of Φ43.14 to Φ63.82



Program:

G6.2 X63.82 Z-50.0 A48 B25 Q60000;
Or

G6.2 U20.68 W-50.0 A48 B25 Q60000;

G6.2/G6.3 code comprehensive programming example

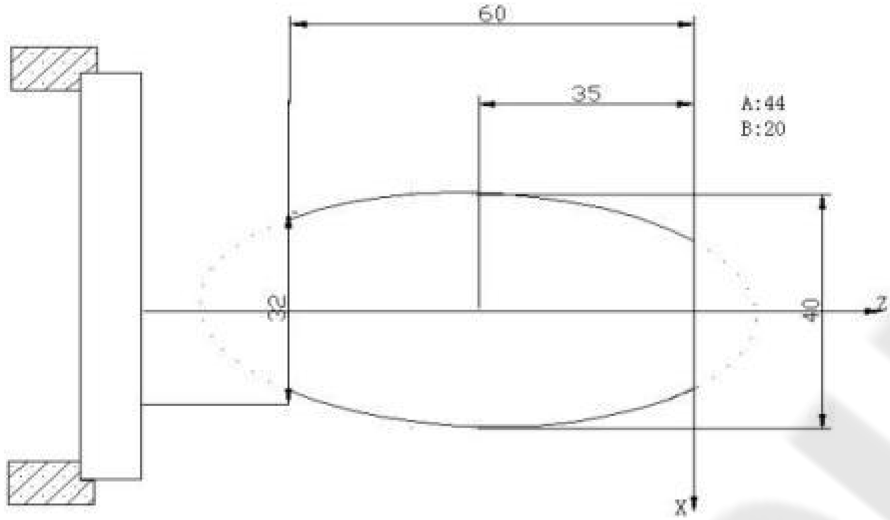


Figure 3-11

Program: O0001

N001 G0 X60 Z5; (Quick positioning)

N002 M03 S200; (Spindle on)

N003 G01 X24.24 Z0 F100; (Approaching the workpiece)

N005 G6.3 X32 W-60 A44 B20; (Cutting A44 B20 ellipse segment)

N006 G01 Z-79;

N007 G0 X60

N008 Z5; (Return to start point)

N009 M05; (Spindle off)

N010 M30; (Program end)

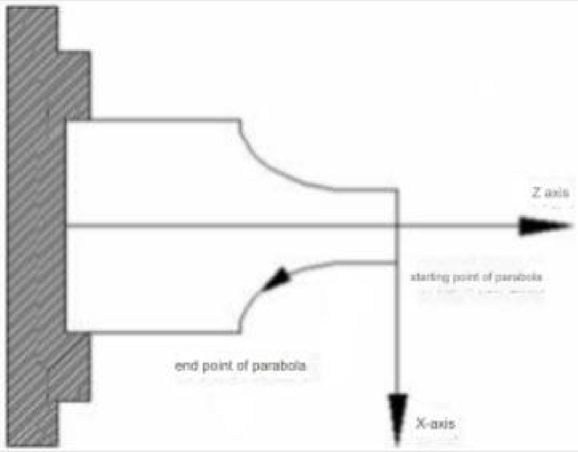
3.5.2 Parabola Interpolation G7.2, G7.3

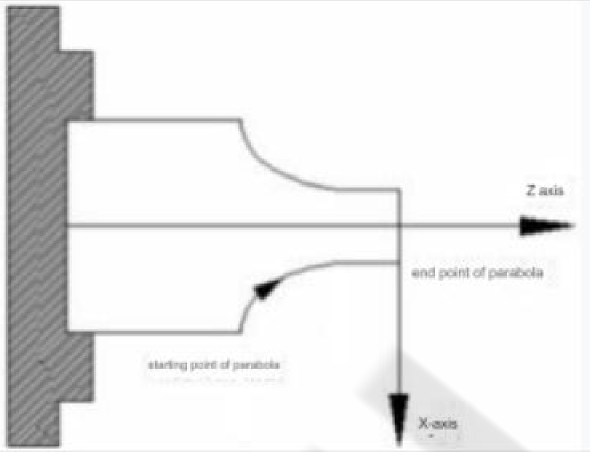
Code format: G7.2 / G7.3 X(U)__ Z(W)__ P__ Q__

Code function: G7.2 The motion trajectory of the code is a clockwise (rear tool holder coordinate system)/counterclockwise (front tool holder coordinate system) parabola from the start point to the end point;

G7.3 The motion trajectory of the code is a counterclockwise (rear tool holder coordinate system)/clockwise (front tool holder coordinate system) parabola from the start point to the end point.

Code trajectory diagram:





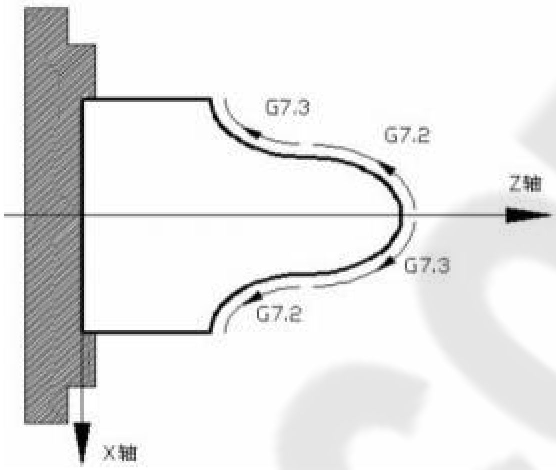
G7.2 code trajectory diagram G7.3 code trajectory diagram

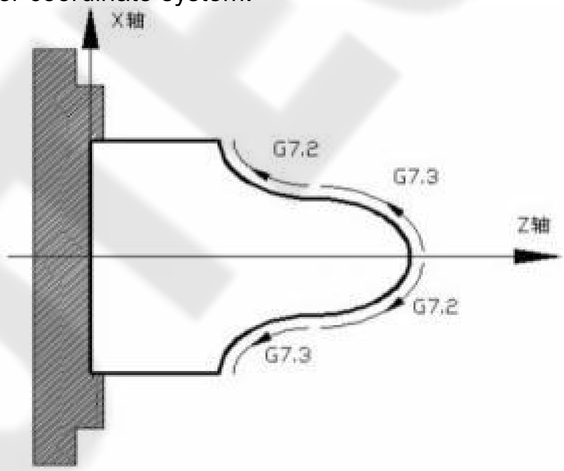
Code description: G7.2 and G7.3 are modal G codes;

P is the P value in the parabola standard equation Y2=2PX, with a value range of 1 ~ 99999999 (unit: least input increment, unsigned);

Q is the angle between the parabola symmetry axis and the Z axis, with a value range of 0 ~ 99999999 (unit: 0.001 degree, unsigned).

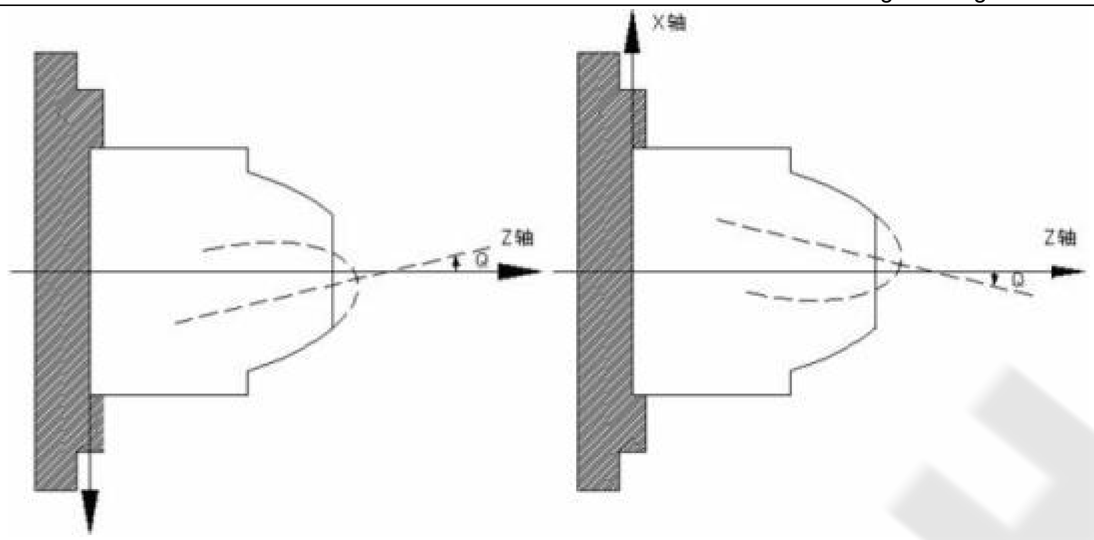
Parabola direction: The direction definition of G7.2/G7.3 interpolation is opposite in the front tool holder coordinate system and the rear tool holder coordinate system.





Front tool holder coordinate system Rear tool holder coordinate system

Q value description: Q value refers to the angle when the positive direction of the Z axis rotates clockwise to coincide with the parabola’s symmetry axis when looking down at the XZ plane from the positive direction of the Y axis in the right-hand rectangular Cartesian coordinate system, as shown in the figure below:



Front tool holder coordinate system Rear tool holder coordinate system

Notes:

● P value cannot be zero or omitted, otherwise an alarm will be generated;

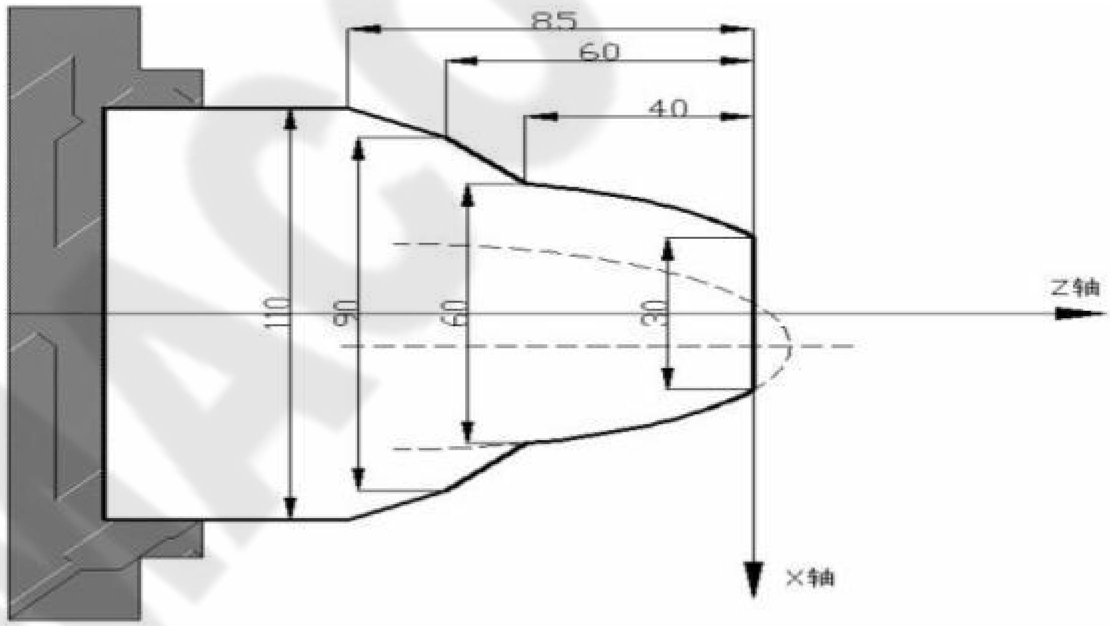
● P value does not contain a sign. If a negative value is entered, its absolute value is taken;

● Q value can be omitted. When Q value is omitted, the parabola's symmetry axis is parallel to or coincides with the Z axis, and Q does not contain a sign;

● When the straight line where the start point and the end point are located is parallel to the parabola's symmetry axis, an alarm will be generated;

● G7.2 and G7.3 codes can be used in compound cycles G70-G73 and C tool compensation. The notes are the same as G02 and G03;

Example: If the parabola's P=10mm (the system’s minimum increment is 0.0001mm), its symmetry axis and Z axes are parallel, and the machining dimension diagram of the part is shown in the figure. The finishing program can be compiled as follows:



Program: O0001(O0001)

G00 X120 Z100 T0101 M03 S800;

```
G00 X10 Z10;
G00 X0;

G01 Z0 F120 M08;
X30;

G7.3 X60 Z-40 P100000 Q180000;
G01 X90 Z-60;

X110 Z-85;
X120;
M09;

G00 X120 Z100 M05 S0;
M30;
```

3.5.3 Polar Coordinate Interpolation G12.1, G13.1

Code format: G12.1---- Start polar coordinate interpolation mode --- (1)

G98
G01 X_ C_
G04 X_
G41/G42 G1 X_ C_
G6.2/G6.3 X_ C_ A_ B_ Q_
G7.2/G7.3 X_ C_ A_ B_ Q_
G02/G03 X_ C_ R_
G40 G1 X_ C_
G65/G66/G67

-- (2) Available instructions

G13.1---- Cancel polar coordinate interpolation mode --- (3)

Code function: Polar coordinate interpolation is a contour control that converts programming instructions in the Cartesian coordinate system into linear axis movement (tool movement) and rotary axis movement (workpiece rotation). It is effective for front cutting in turning and camshaft grinding.

Code description: G12.1 and G13.1 are non-modal G codes

Linear axis: X axis or Z axis Y axis or 4th axis or 5th axis

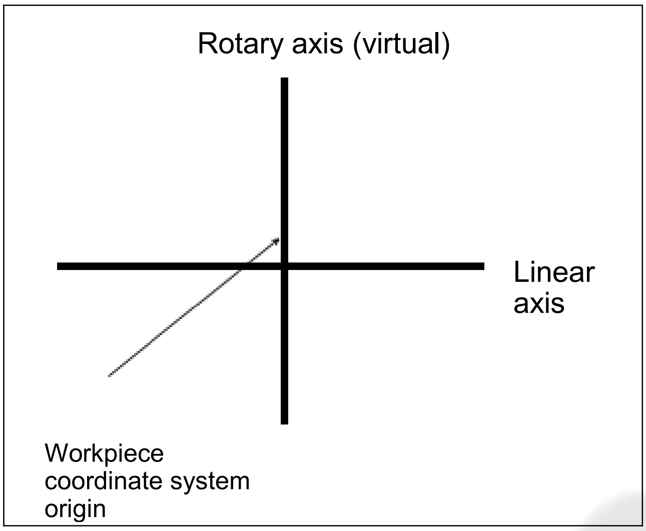
Rotary axis: Axis other than feed axis (Y axis or 4th axis or 5th axis)

Before starting polar coordinates, the linear axis and rotary axis must be set in parameter

N0002 in advance.

The following takes linear axis X and rotary axis 5 as examples

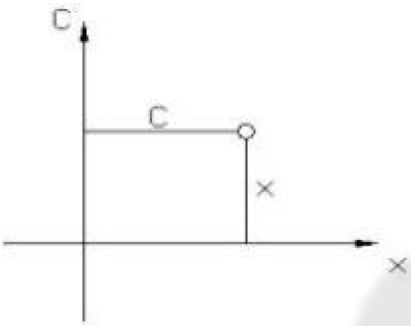
Polar coordinate interpolation plane: G12.1 starts polar coordinate interpolation mode and selects a polar coordinate interpolation plane (as shown below), and polar coordinates are completed in this plane.



Polar coordinate interpolation plane

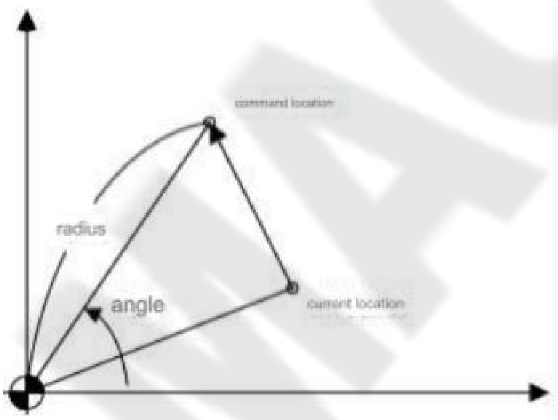
Note: After G12.1, the previous plane is canceled. Entering the polar coordinate interpolation plane, the plane is canceled after using G13.1, and the previous plane is restored; after resetting, the polar coordinate interpolation is canceled, the plane is restored to the previous plane, and the cursor returns to the beginning of the program.

Programming format: 1. Program rectangular coordinates in the polar coordinate interpolation plane, as shown below:



X: linear axis distance, unit mm/inch;
C: distance on the rotary axis, unit: mm/inch;
Linear axis can be programmed in diameter, and rotary axis can only be programmed in radius;

2. Program polar coordinates in the polar coordinate interpolation plane, as shown in the figure below:



X: length from the current tool to the origin, unit: mm/inch;
C: angle of the current rotary axis, unit: deg;
Use G16 to indicate that the currently programmed coordinates are polar coordinates, and use G15 to cancel.
If G16 is not programmed, the program defaults to rectangular coordinates.
G16/G15 are only valid in polar coordinate interpolation.

Both linear and rotary axes are programmed in radius. As shown below:

●Length compensation: There is no length compensation for the rotary axis, and the length offset should be instructed before becoming G12.1 mode. The length offset cannot be changed in the polar coordinate interpolation mode.

●Tool nose radius compensation: The tool nose direction is 0.

●Machine tool movement: The linear axis and the rotary axis are perpendicular.

● Circular interpolation in the interpolation plane: The address for the arc radius instructed for circular interpolation in the polar coordinate interpolation plane depends on the first axis (linear axis) in the interpolation plane.

• When the linear axis is the X axis or its parallel axis, use I and J in the Xp-Yp plane

• When the linear axis is the Y axis or its parallel axis, use J and K in the Yp-Zp plane

• When the linear axis is the Z axis or its parallel axis, use K and I in the Zp-Xp plane. The arc radius can also be instructed with R.

● Command speed: Tangential speed in the polar coordinate plane.

When the tool moves near the center of the workpiece, the speed component of the C axis becomes larger and exceeds the maximum cutting feed rate of the C axis. The F value in the program should be derived from the following formula:

L: The distance between the tool center and the workpiece center when the tool center is closest to the workpiece center (mm)

R: The maximum cutting feed rate of the C axis (deg/min)

It can be obtained: $F<L\times R\times\pi/180$ (mm/min)

Therefore, it is not recommended to process the workpiece near the pole, because in some cases, the feed rate is required to change rapidly to prevent the rotary axis from overloading.

If the tool is exactly at the pole, do not select the polar coordinate interpolation function.

●Movement along the axis in the non-polar coordinate interpolation plane in polar coordinate interpolation mode: The tool can move normally along these axes regardless of polar coordinate interpolation, but the non -polar coordinate axis is considered invalid in the arc, ellipse or parabola instruction.

●Coordinate display: After executing G12.1, the absolute coordinates, machine tool coordinates, and relative coordinates all display the actual position of the tool, and the remaining distance is displayed according to the rectangular coordinates in the polar coordinate interpolation plane, and the coordinates after executing G13.1 or pressing reset display the coordinates in the current system plane.

Notes:

1. G12.1 and G13.1 are group 21 codes. G12.1, G13.1, G16 and G15 should be placed in a separate line.

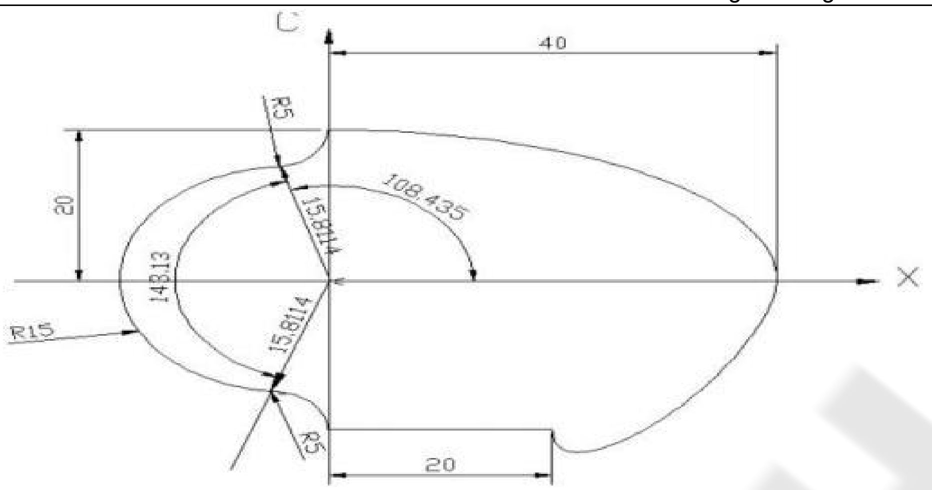
2. Tool change is not allowed between G12.1 and G13.1. Tool change and positioning after tool change must be placed before G12.1.

3. Polar coordinate interpolation cannot be started in the middle of C tool compensation or G99 state, or an alarm will be issued.

4. When G12.1 is commanded, the tool position of polar coordinate interpolation starts from angle 0.

5. The angle Q in the ellipse and parabola instructions is the angle with the Z axis in G18, the angle with the Y axis in G19 plane, and the angle with the X axis in G17 plane.

Example:



00001 (00001)
T0101
G0 X80 C0 W0
G12.1
G6.3 X0 C20 A40 B20 F1000

G2 X-10 C15 R5
G3 X-10 C-15 R15

} can be replaced by

G16-------indicates that the following programming is length and angle programming
G2 X15.8114 C108.435 R5

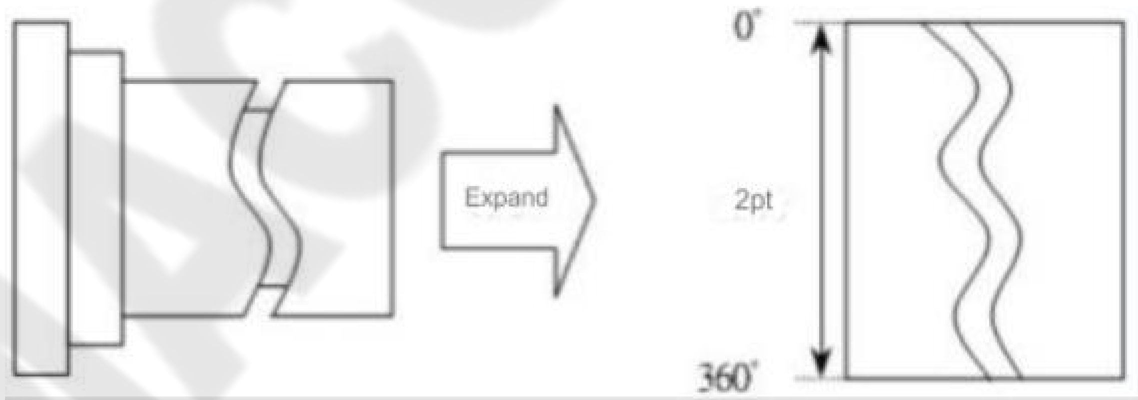
G3 X15.8114 C251.565 R15

G15-------indicates that the above programming mode is canceled and the following is rectangular coordinate programming

G2 X0 C-20 R5
G1 X40 C-20
G7.3 X80 C0 P10000 Q0
G13.1
M30

3.5.4 Cylindrical Interpolation G7.1

Code function: The movement of the rotary axis specified by the angle is internally converted into the distance of the linear axis along the outer surface so that linear interpolation or circular interpolation can be completed together with other axes. After the interpolation is completed, this distance is converted into the movement of the rotary axis. Cylindrical interpolation is programmed with the unfolded surface of the cylinder (as shown below).



Code format: G07.1 Cc; (cylindrical interpolation mode start/cancel)

Cc is the cylinder radius value;

Radius value ≠ 0: Cylindrical interpolation mode starts
Radius value = 0: Cylindrical interpolation mode cancels

(1) The coordinate instruction for the interval from the start of cylindrical interpolation mode to the cancellation is the cylindrical coordinate system;

G07.1 Cxxxx (cylindrical radius value); ……cylindrical interpolation starts

……;

……; ………………the coordinate instruction in this interval is the cylindrical coordinate system

……;

G07.1 C0; ………………cylindrical interpolation cancels

(2) G7.1 is a non-modal code;

(3) At power-on and reset, it is in cylindrical interpolation cancel mode;

(4) The rotation axis executes the program according to the angle. The scrolling function of the rotation axis in the cylindrical interpolation mode will be automatically invalid. When the interpolation range is greater than one circle, the programming instruction value must be greater than 3600;

(5) Coordinate values can be absolute or incremental;

(6) Tool nose radius compensation G41 and G42 can be performed and the tool nose direction is considered to be 0;

(7) Feed rate F is the tangent speed on the cylindrical unfolded surface, in mm/min or inch/min;

Code restrictions:

(1) In cylindrical interpolation mode, linear interpolation G1, circular interpolation G2 and G3 (arc radius can only be specified by R, in mm or inch), ellipse interpolation G6.2 and G6.3, parabola interpolation G7.2 and G7.3 can be realized;

(2) G00 positioning operation is not allowed in cylindrical interpolation mode;

(3) Before entering cylindrical interpolation mode, the ongoing tool radius compensation mode should be canceled, and tool compensation should be started and ended in cylindrical interpolation mode;

(4) Auxiliary function T can’t be used in cylindrical interpolation mode;

(5) The cylindrical interpolation feed rate can only be specified by G98 feed in steps;

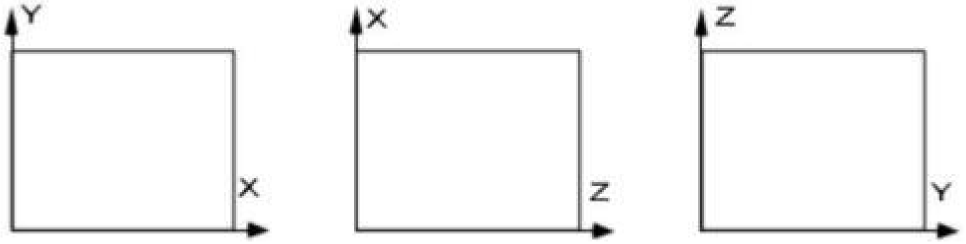
(6) G50 can’t be used to set the workpiece coordinate system in cylindrical interpolation mode;

(7) Only the rotation axis and linear axis of the current cylinder can be specified in cylindrical interpolation;

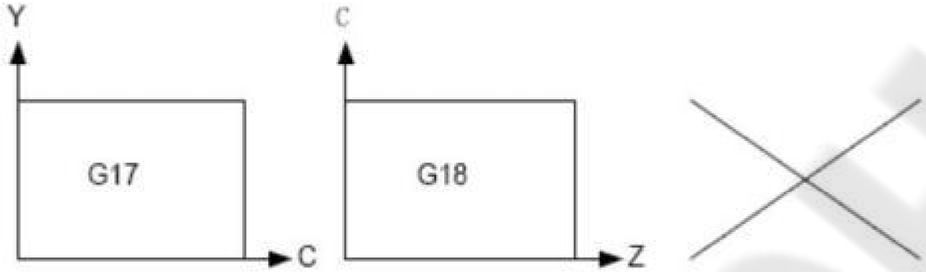
Plane selection: Before starting cylindrical interpolation, the plane where the interpolation is located should be selected first. One axis in the plane will be the linear axis in cylindrical interpolation, and the other axis will be the linear axis corresponding to the expansion of the rotation axis in cylindrical interpolation (see the figure below).

Basic coordinate system:

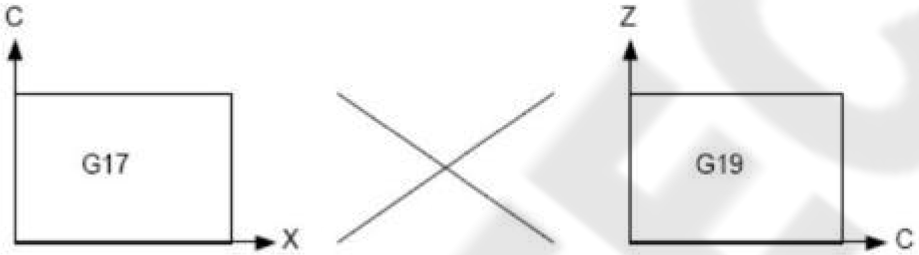
G17 G18 G19



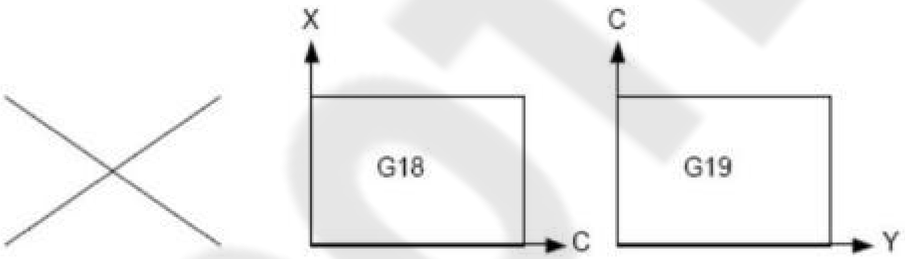
When the rotation axis in cylindrical interpolation is set to the X axis or an axis parallel to the X axis:



When the rotation axis in cylindrical interpolation is set to the Y axis or an axis parallel to the Y axis:



When the rotation axis in cylindrical interpolation is set to the Z axis or an axis parallel to the Z axis:



Related parameters:

Only one rotary axis can be specified in cylindrical interpolation. The rotary axis can be either a basic axis or an axis parallel to the basic axis. The axis names of the three additional axes can be set by parameter NO.225 (Y: 89, A: 65, B: 66, C: 67), and the axis attributes are set by data parameter NO.230 (see the table below).

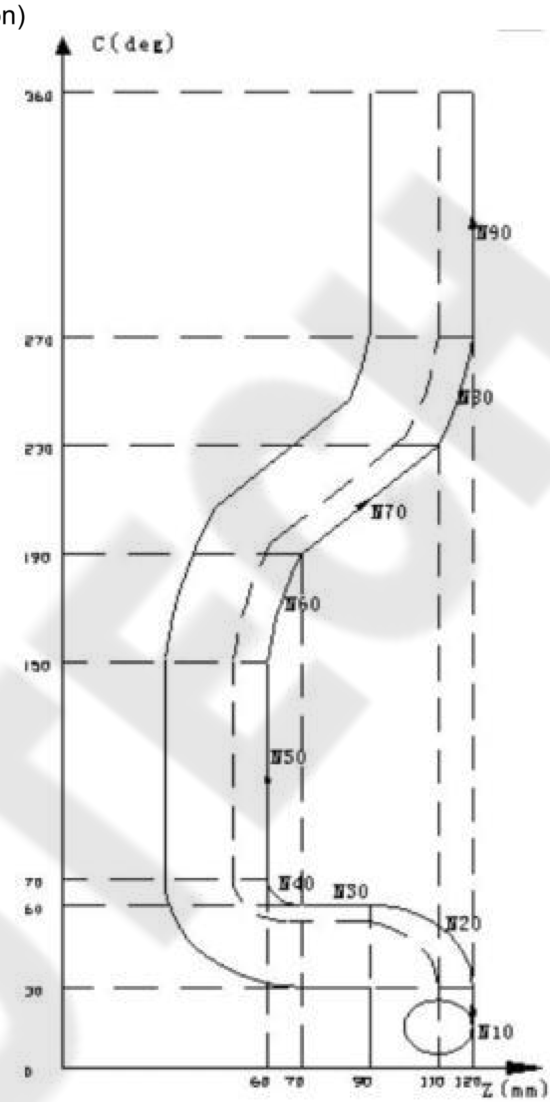
Setting value	Meaning
0	Neither the basic three axes nor the parallel axis
1	X axis of the basic three axes
2	Y axis of the basic three axes
3	Z axis of the basic three axes
5	The parallel axis of X axis
6	The parallel axis of Y axis
7	The parallel axis of Z axis

Example:

First set parameter NO.224 to 5, select cylindrical interpolation under the G18 plane, set the rotation axis used for cylindrical interpolation to the 5th axis, set the axis name of the 5th axis to C (data parameter NO.225), and set the attribute of the 5th axis to the parallel axis of the X axis (data parameter NO.230). The radius of the cylinder is 57.299mm, and the trajectory unfolded by the cylinder is shown in the figure below:

O0071 (example of cylindrical interpolation G7.1 application)

```
G18;
G98;
G00 X150 Z105 C0;

G01 X114.598 Z105 F200;
G07.1 C57.299;

G41 G01 Z120;
N10 G01 C30.0;

N20 G03 Z90 C60 R30;
N30 G01 Z70;

N40 G02 Z60 C70 R10;
N50 G01 C150;

N60 G02 Z70 C190 R75;
N70 G01 Z110 C230;

N80 G03 Z120 C270 R75;
N90 G01 C360;

G40 G01 Z105;
G07.1 C0;
M30;
```



3.5.5 Automatic Tool Compensation G10

G10 P_Tool offset number U_ W_

3.5.6 Plane Selection Code G17 ~ G19

Code format:

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

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

Code description: G17, G18, G19 are modal G codes. In the program segment without instructions, the plane does not change.

Notes:

● When selecting G17 or G19 planes, the basic axis Y must be set first;

● Plane switching cannot be performed in C tool compensation state;

● G71 ~ G76, G90, G92, and G94 can only be used in the G18 plane;

● The plane selection code can be shared with other groups of G codes;

● The movement instruction is independent of plane selection;

● Regarding the processing of diameter or radius programming: Since there is only one parameter N0001 to select diameter or radius programming and it is only valid for the X axis, when using G2, G3 and other instructions, the Z axis and Y axis can only use radius programming, and the X axis is selected by parameters;

● The tool nose direction of C tool compensation under G17 and G19 planes is 0.

3.5.6 Cutting G24

G24 P_number of tools (negative number means reverse direction) Q_number of cutters R_synchronous phase difference (0-359.999)

G25 End cutting)

3.6 Chamfering Function

The chamfering function is to insert a straight line or arc between two contours so that the tool can smoothly transition from one contour to another. It has two chamfering functions: straight line and arc.

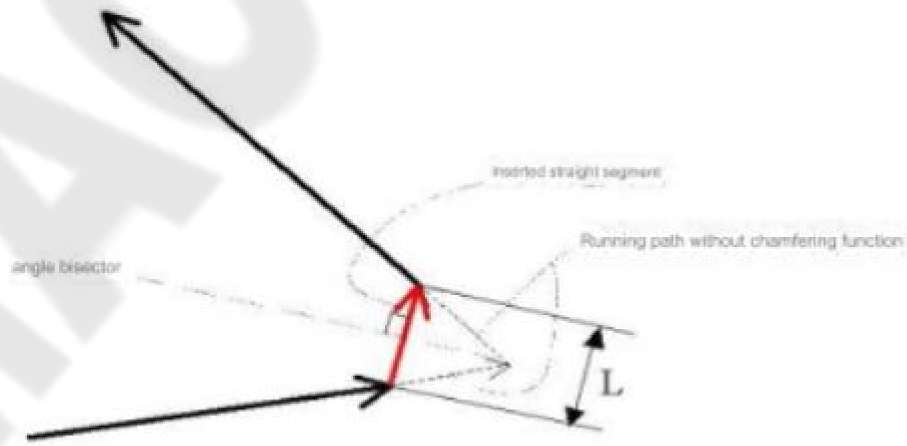
3.6.1 Straight Line Chamfering

Straight line chamfering: insert a straight line between straight line contours, between arc contours, and between straight line contour and arc contour. The code address of straight line chamfering is L. The length of the chamfering line is specified by L. The value range is 0-1000mm. If the value specified by L exceeds the range, the L code can be ignored. Straight line chamfering must be used in G01, G02 or G03 code segments.

A. Straight line to straight line

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

Code function: insert a straight line segment between two straight line interpolation code segments.



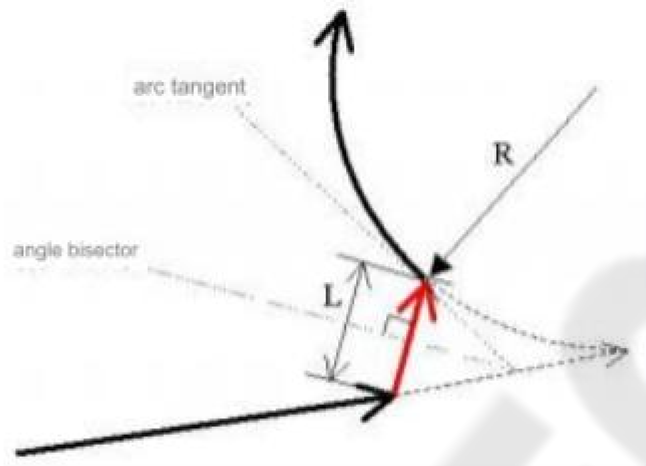
B. Straight line to arc

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 function: insert a straight line segment between straight line and arc interpolation codes.



C. Arc to arc

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

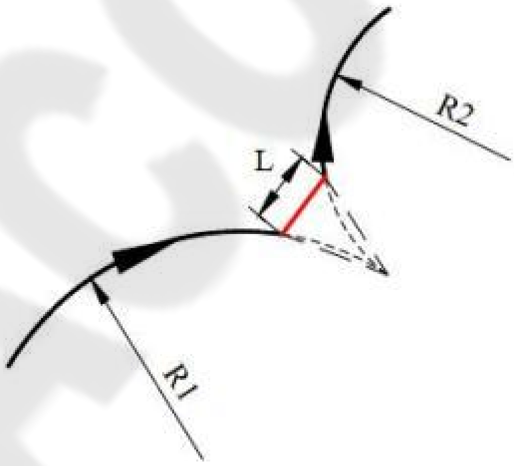
G02/G03 X/U_ Z/W_ R_;

Or

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

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

Code function: insert a straight line segment between two circular interpolation codes.



D. Arc to straight line

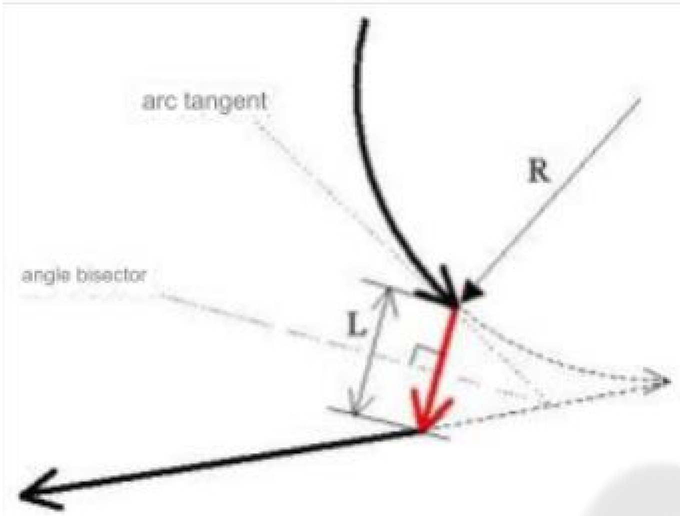
Code format: G02/G03 X/U_ Z/W_ R_ L_;
G01 X/U_ Z/W_;

Or

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

G01 X/U_ Z/W_;

Code function: insert a straight line segment between circular and straight line interpolation codes.



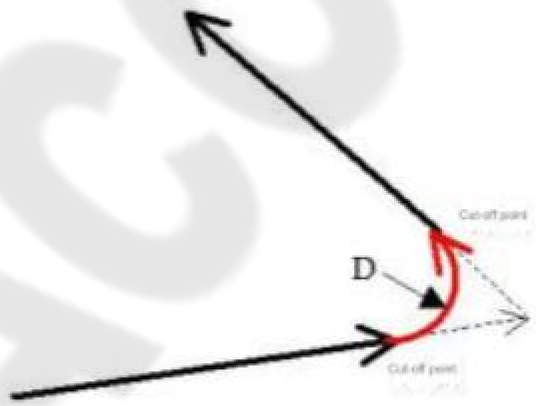
3.6.2 Arc Chamfering

Arc chamfering: insert an arc between straight line contours, between arc contours, or between straight line contour and arc contour, and make a tangent transition between the arc and the contour line. The code address of the arc chamfering is D. The radius of the chamfering arc is specified by D, and the value range is 0~1000mm. If the value specified by D exceeds the range, the D code can be ignored. Arc chamfering must be used in the G01, G02 or G03 code segment.

A. Straight line to straight line

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

Code function: insert an arc between two straight line interpolation segments, and the inserted arc segment is tangent to the two straight lines. The radius value is specified by D.



B. Straight line to arc

Code format: G01 X/U_Z/W_ D_;

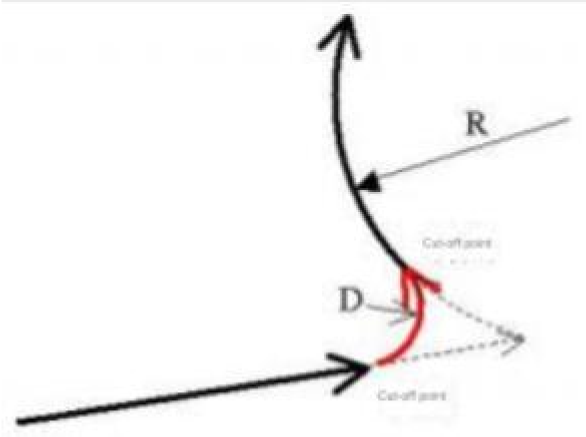
G02/G03 X/U_Z/W_ R_;

Or

G01 X/U_ Z/W_ D_;

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

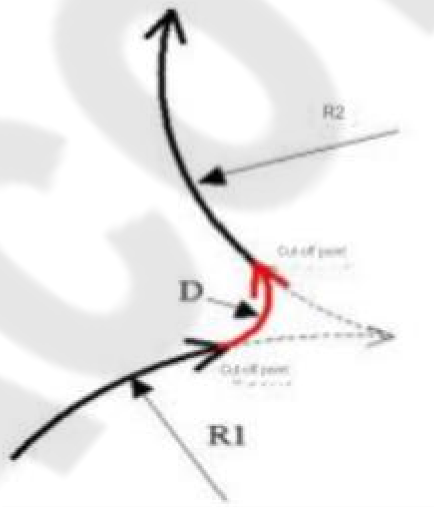
Code function: Insert an arc at the intersection of a straight line and an arc. The inserted arc segment is tangent to both the straight line and the arc. The radius value is specified by D.



C. Arc to 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_;

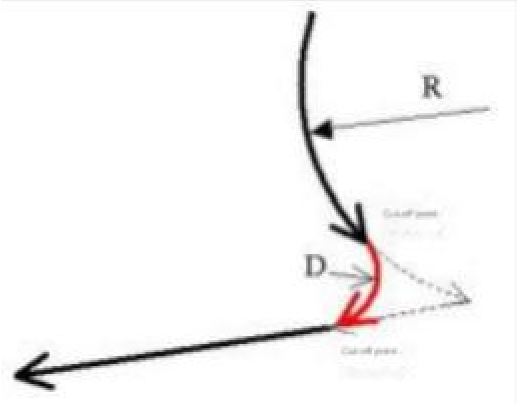
Code function: Insert an arc between two arcs. The inserted arc segment is tangent to both arcs. The radius value is specified by D.



D. Arc to 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 an arc at the intersection of an arc and a straight line. The inserted arc is tangent to both the arc and the straight line. The radius value is specified by D.



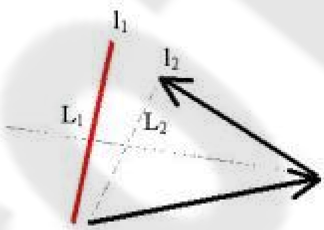
3.6.3 Special Cases

In the following situations, the chamfering function is invalid or an alarm is issued.
1) When chamfering a straight line

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

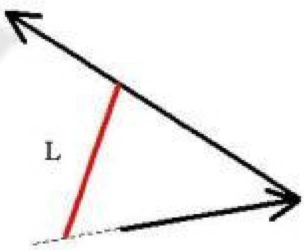
B. The length of the chamfering straight line is too long, and the CNC generates an alarm.

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



C. A certain straight line (arc) is too short, and an alarm is issued

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

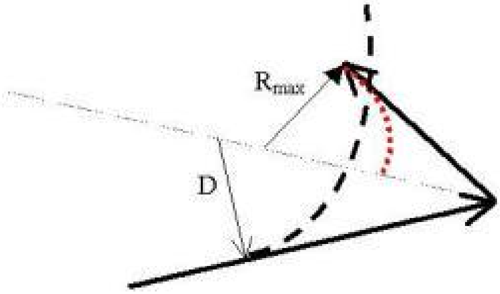


2) When chamfering an arc

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

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

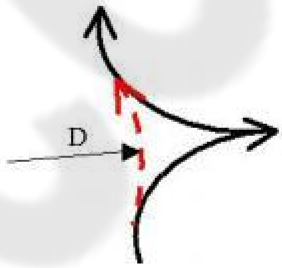
As shown in the figure below, the radius of the chamfered arc is D, and the maximum arc radius of the two straight lines is Rmax. If Rmax is less than D, the CNC will generate an alarm.



C. When the straight line is tangent to the arc, or the arc is tangent to the straight line, the arc chamfering function is invalid.

D. When the arcs are tangent to each other, the arc chamfering function is invalid;

However, if the arcs are tangent like the following figure, the arc chamfering function is valid.



3.7 Pause Code G04

Code format: G04 P_; or
G04 X_; or
G04 U_; or
G04;

Code function: Each axis stops moving, does not change the current G code mode and the data and status that are maintained, and executes the next program segment after a given delay time.

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

The G04 delay time is specified by the code word P_, X_ or U_;
The P value range is 0 ~ 99999 (unit: ms).

The range of X and U codes is 0 ~ 9999.999 × the least input increment (unit: s).

Notes:

● When P, X, and U are not input, it means accurate stop between program segments.

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

3.8 Mechanical Origin (Machine Tool Origin) Function

3.8.1 Machine Tool First Reference Point G28

Code format: G28 X/U_ Z/W_;

Code function: Starting from the start point, reach the intermediate point position specified by X/U and Z/W at a quick moving speed and then return to the machine tool origin.
Code description: G28 is a non-modal G code;

X, Z: Absolute coordinates of the intermediate point position;

U, W: The difference between the X-axis absolute coordinates of the intermediate point position and the start point position.

Code address X/U, Z/W can be omitted one or all of them; see the following table for details:

Table 3-4

Instruction	Function
G28 X/U	X axis returns to machine tool origin, Z and Y axes remain in original position
G28 Z/W	Z axis returns to machine tool origin, X and Y axes remain in original position
G28	Remain in original position and continue to execute the next program segment
G28 X/U Z/W	X and Z axes return to machine tool origin at the same time

Code action process (as shown in Figure 3-12):

(1) Quick positioning from the current position to the middle point (point A → point B);
(2) Quick positioning from the middle point to the reference point (point B → point R);

(3) If the machine tool is not locked, the home light will be on when the reference point return is completed.

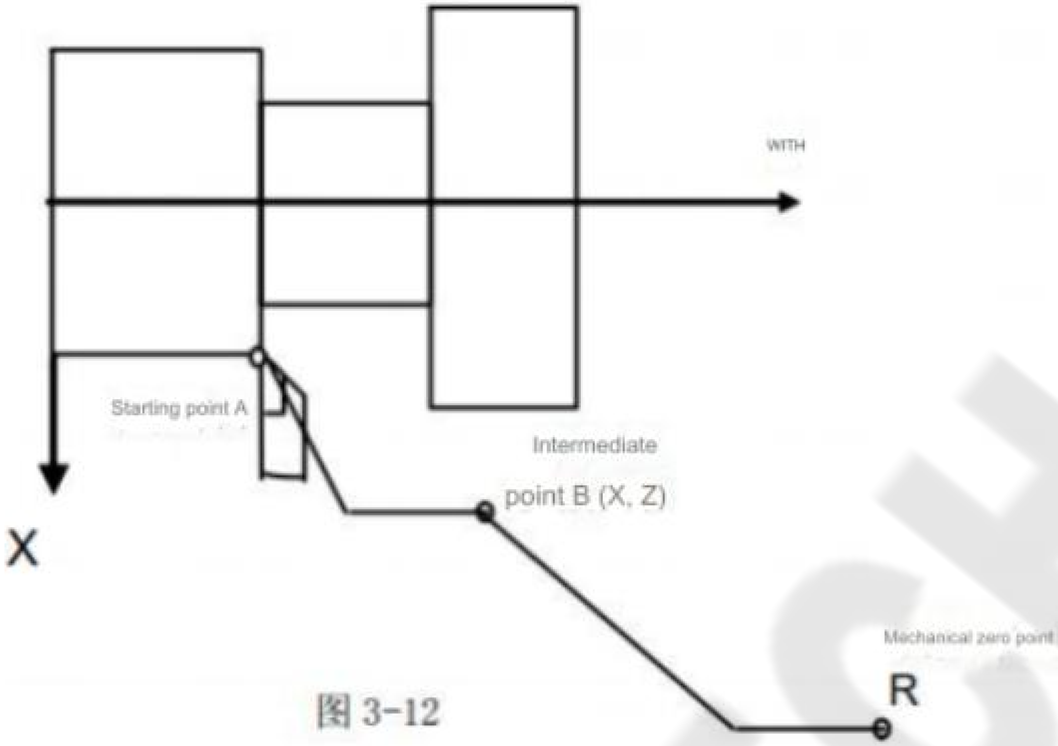


Figure 3-12

Note 1: The process of manual machine tool homing is the same as that of executing the G28 code. The deceleration signal and the one-turn signal must be detected each time;

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

Note 3: After executing the G28 code to return to the machine origin, the system cancels the tool length compensation;

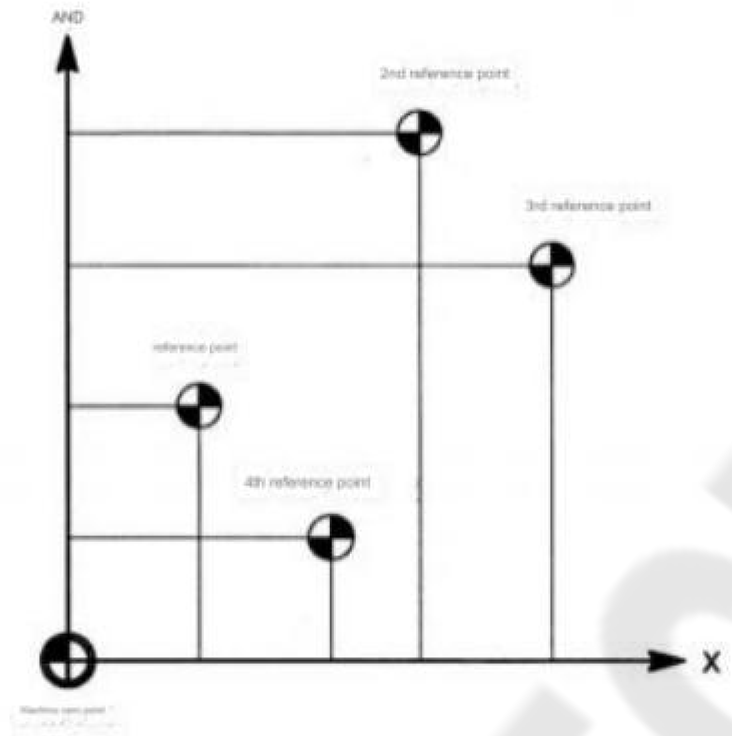
Note 4: If the machine tool is not equipped with an origin switch, the G28 code and the operation of returning to the machine tool origin must not be executed.

3.8.2 Machine Tool 2nd, 3rd, 4th Reference Point G30

The machine tool origin is a fixed point on the machine tool, which is determined by the origin switch or home switch installed on the machine tool. The coordinates of the machine reference point are the values set by parameter N0090.

The SZGH880T/SZGH1080T series has the function of the 2nd, 3rd, and 4th reference points of the machine tool. The machine tool coordinates of the X and Z axes of the 2nd, 3rd, and 4th reference points of the machine tool can be set respectively by using parameters N0091~N0095.

The relationship between the machine tool origin, the machine tool reference point, and the 2nd, 3rd, and 4th reference points of the machine tool in the machine tool coordinate system is shown in the figure below.



Code format:

G30 P2 X/U_ Z/W_;

G30 P3 X/U_ Z/W_;

G30 P4 X/U_ Z/W_;

Code function: Starting from the start point, move to the intermediate point position specified by X/U and Z/W at a quick moving speed and then return to the 2nd, 3rd, and 4th reference points of the machine tool. When returning to the 2nd reference point of the machine tool, the code address P2 can be omitted.

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

X: Absolute coordinate of the X axis of the middle point;
U: Relative coordinate of the X axis of the middle point;

Z: Absolute coordinate of the Z axis of the middle point;

W: Relative coordinate of the Z axis of the middle point.

Code address X/U, Z/W can be omitted one or all of them; see the following table for details:

Instruction	Function
G30 Pn X/U_	X axis returns to the nth reference point of the machine tool, Z axis remains in original position
G30 Pn Z/W_	Z axis returns to the nth reference point of the machine tool, X axis remains in original position
G30	Both axes remain in original position and continue to execute the next program segment
G30 Pn X/U_ Z/W_	X and Z axes return to the nth reference point of the machine tool at the same time

Note 1: In the table, n takes the value 2, 3 or 4;

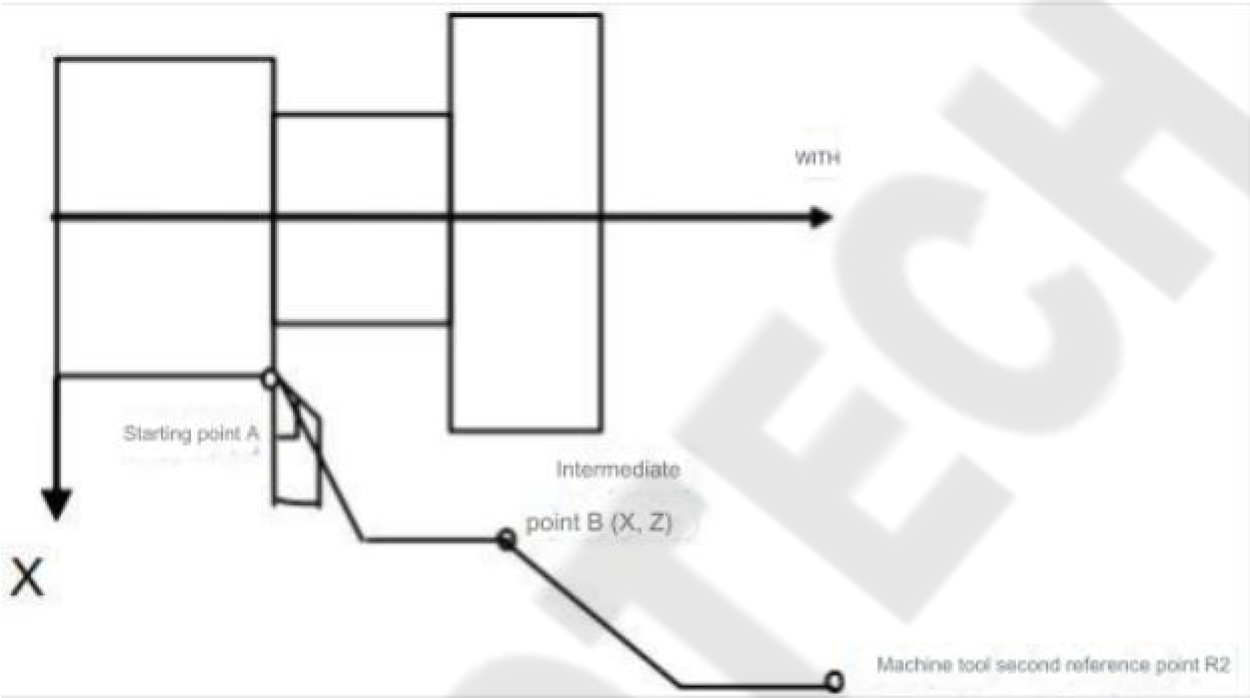
Note 2: It is not necessary to detect the deceleration and origin signals during the process of returning to the 2nd, 3rd, and 4th reference points of the machine tool.

Code execution process (as shown in the figure below, using the second reference point of the machine tool as an example):

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

(2) Position from the middle point to the second reference point set by parameter N0091 (point B → point R2) at the speed set by parameter N0021;

(3) If the machine tool is not locked, when returning to the reference point, the Bit0 and Bit1 of the reference point position return completion signal ZP21 are high.



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

Note 2: 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 lines;

Note 3: After executing the G30 code to return to the machine reference point 2, 3, and 4, the system cancels the tool length compensation;

Note 4: If the machine tool is not equipped with an origin switch, the G30 code must not be executed to return to the machine reference point 2, 3, and 4;

Note 5: Return to the machine reference point 2, 3, and 4, and do not set the workpiece coordinate system.

3.9 Jump Interpolation G31

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

Code function: During the execution of this code, if an external jump signal (X3.1) is input, the execution of this code is interrupted and the next program segment is executed instead. This function can be used for dynamic measurement of workpiece dimensions (such as grinding machines), tool setting measurement, etc.

Code Description: Non-modal G code (00 group);

The address format is consistent with G01 code, and the use is similar;

The tool nose radius compensation needs to be canceled before using this code;

To ensure the stop position accuracy, the feed rate should not be set too high.

a. The line of the subsequent segment when the jump occurs:

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

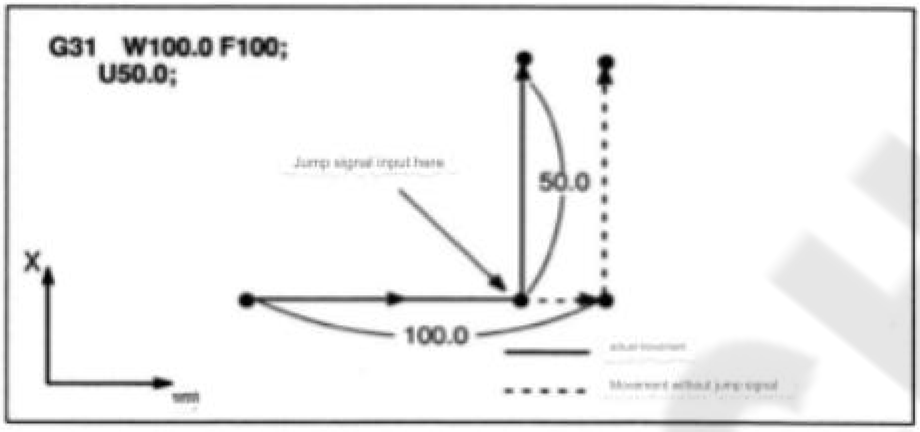


Figure 3-13

2. The next program segment of G31 is absolute coordinate programming of 1 axis, see Figure 3-14

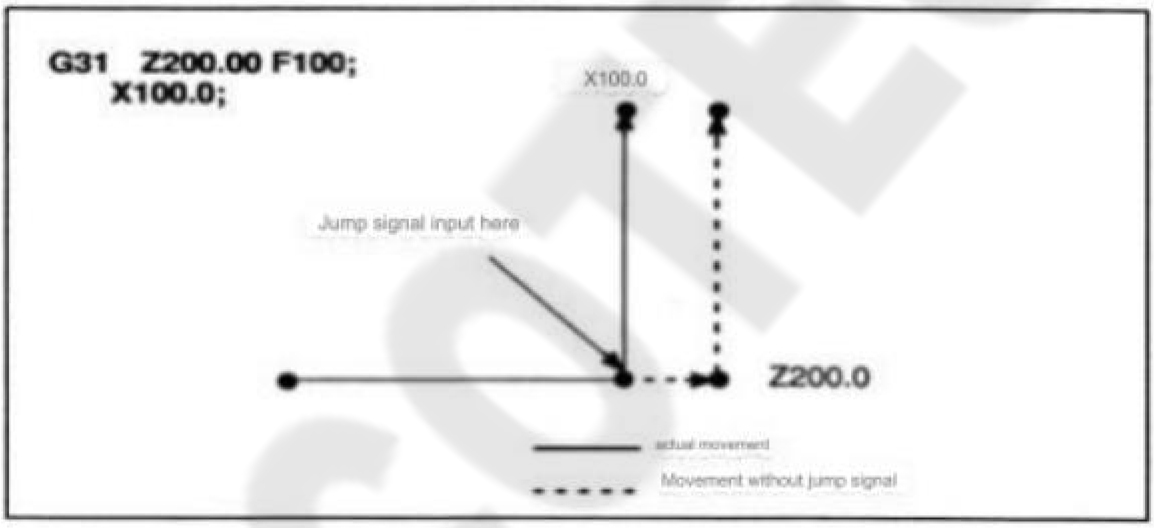


Figure 3-14

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

Program: G31 Z200 F100

G01 X100 Z300

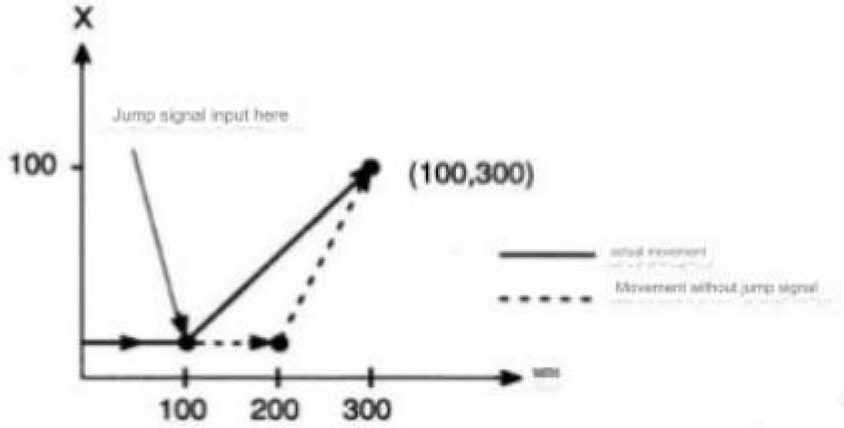


Figure 3-15

b. Signals related to the G31 jump code:
Jump signal:

SKIP: G6.6

Type: Input signal

Function: G6.6 signal ends 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, Y, 4th, 5th respectively). Also, the motion code of the program segment is ended at the same time.

Operation: When the jump signal becomes "1", the CNC processes as follows:

When the program segment is executing the jump code G31, the CNC stores the current absolute coordinate position of each axis. CNC stops the movement of G31 code and starts the execution of the next program segment. 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 met.

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

G31

First usage: waiting for external X3.1

Second usage: G31 I_ K_

I: The port number of the input port, a signed number. Greater than 0: indicates that the waiting port is valid; less than 0: indicates that the waiting port is invalid

K: The trajectory is completed, no signal is detected, if an alarm is required, add K1. If an alarm is not required, do not use K

3.10 Floating Workpiece Coordinate System Setting 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 uses the current position as the program origin, and returns to this position when executing the program homing operation. After the floating workpiece coordinate system is established, the absolute coordinate programming inputs the coordinate value according to this coordinate system until G50 is executed again to establish a new workpiece coordinate system.

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

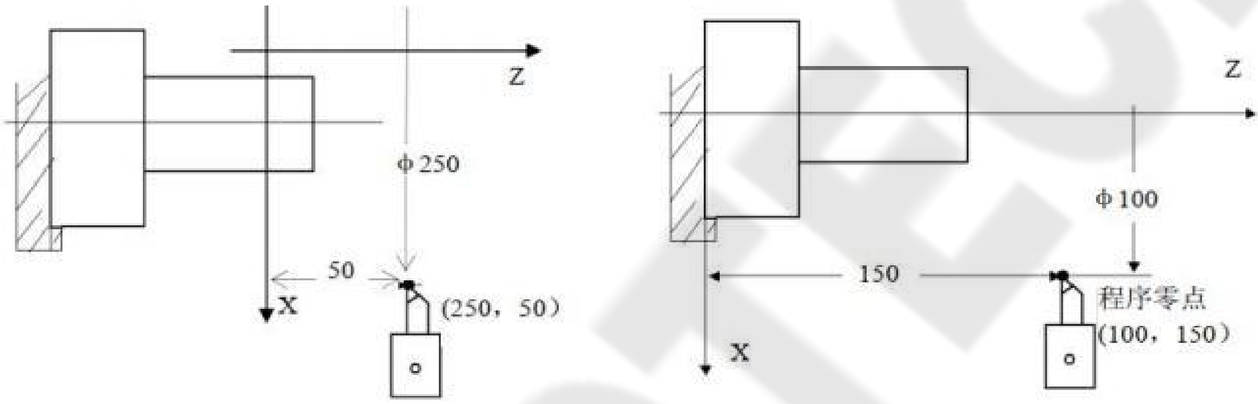
X: the new X-axis absolute coordinate of the current position;

U: the difference between the new X-axis absolute coordinate of the current position and the absolute coordinate before the code is executed;

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

W: the difference between the new Z-axis absolute coordinate of the current position and the absolute coordinate before the code is executed;

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



Before setting coordinate system with G50 After setting coordinate system with G50

Figure 3-16

As shown in Figure 3-16, after executing the code segment "G50 X100 Z150;", the workpiece coordinate system shown in the figure is established, and the point (X100 Z150) is set as the program origin.

3.11 Mirror G51.1, G50.1

G51.1 X_Z_ set programmable mirroring; example: G51.1 X0 // X=0 position, start mirroring as mirror axis

G50.1 X_Z_ cancel programmable mirroring; example: G50.1 X0 // end mirroring

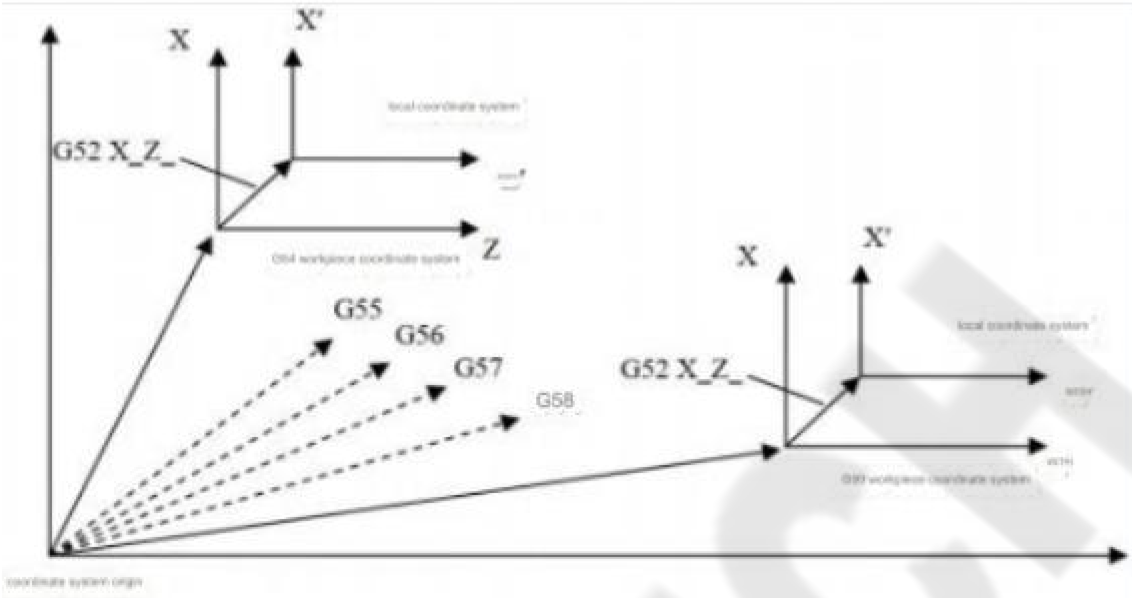
3.12 Local Coordinate System G52

When programming on the workpiece coordinate system, another sub-workpiece coordinate system can be created in the workpiece coordinate system for convenience. Such a sub-coordinate system is called a local coordinate system.

Code format: G52 X(U)_ Z(W)_

Code function: Using the G52 instruction, the local coordinate system can be set in all workpiece coordinate systems (G54~G59). The origin of each local coordinate system becomes the position specified by

X(U)_ Z(W)_ in each workpiece coordinate system. The corresponding relationship with the workpiece coordinate system is as follows:



Code description: G52 code is 00 group G code, which is a non-modal code. X(U) Z(W) is the coordinate position of the origin of the specified local coordinate system in the current workpiece coordinate system. The results of execution are the same when the absolute code or relative code is specified.

Notes:

● The setting of the local coordinate system does not change the workpiece coordinate system and the machine tool coordinate system.

● When executing G52, the tool nose radius compensation will be temporarily canceled.

● After G52 is specified, the local coordinate system remains valid before the next G52 instruction is specified. Also, no movement occurs when the G52 instruction is specified.

● To cancel the local coordinate system, the origin of the local coordinate system should be consistent with the origin of the workpiece coordinate system, that is, the instruction G52 X0 Z0 or G52 U0 W0.

● When the workpiece coordinate system is set with the G50 instruction, the local coordinate system of all workpiece coordinate systems of the specified axis is canceled. If the coordinate values of all axes are not specified, the local coordinate system of the axis without the specified coordinate value will not be canceled.

● Whether to cancel the local coordinate system at reset, mechanical homing, or at the end of program execution can be set by parameters.

Example:

N1 G28 X0 Z0;

N2 G55 G00 X50 Z50;
N3 G52 X100 Z100;

N4 G00 X50 Z50;

N5 G01 Z100 F100;
N6 X100;

N7 G52 X0 Z0;
N8 G00 X0 Z0;

N9 M30;

In the N3 segment, the local coordinate system is established according to the G55 workpiece coordinate system and is canceled in the N7 segment.

3.13 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 allows setting 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 system 3

G57 Workpiece coordinate system 4

G58 Workpiece coordinate system 5

G59 Workpiece coordinate system 6

3. When different workpiece coordinate systems are called in the program segment, the axis that is instructed to move will be positioned to the coordinate point in the new workpiece coordinate system; the axis that is not instructed to move will jump to the corresponding coordinate value in the new workpiece coordinate system, and the actual machine tool position will not change.

Example: The machine tool coordinate corresponding to the origin of the coordinate system of G54 is (20, 20)
The machine tool coordinate corresponding to the origin of the coordinate system of G55 is (30, 30)

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

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

4.

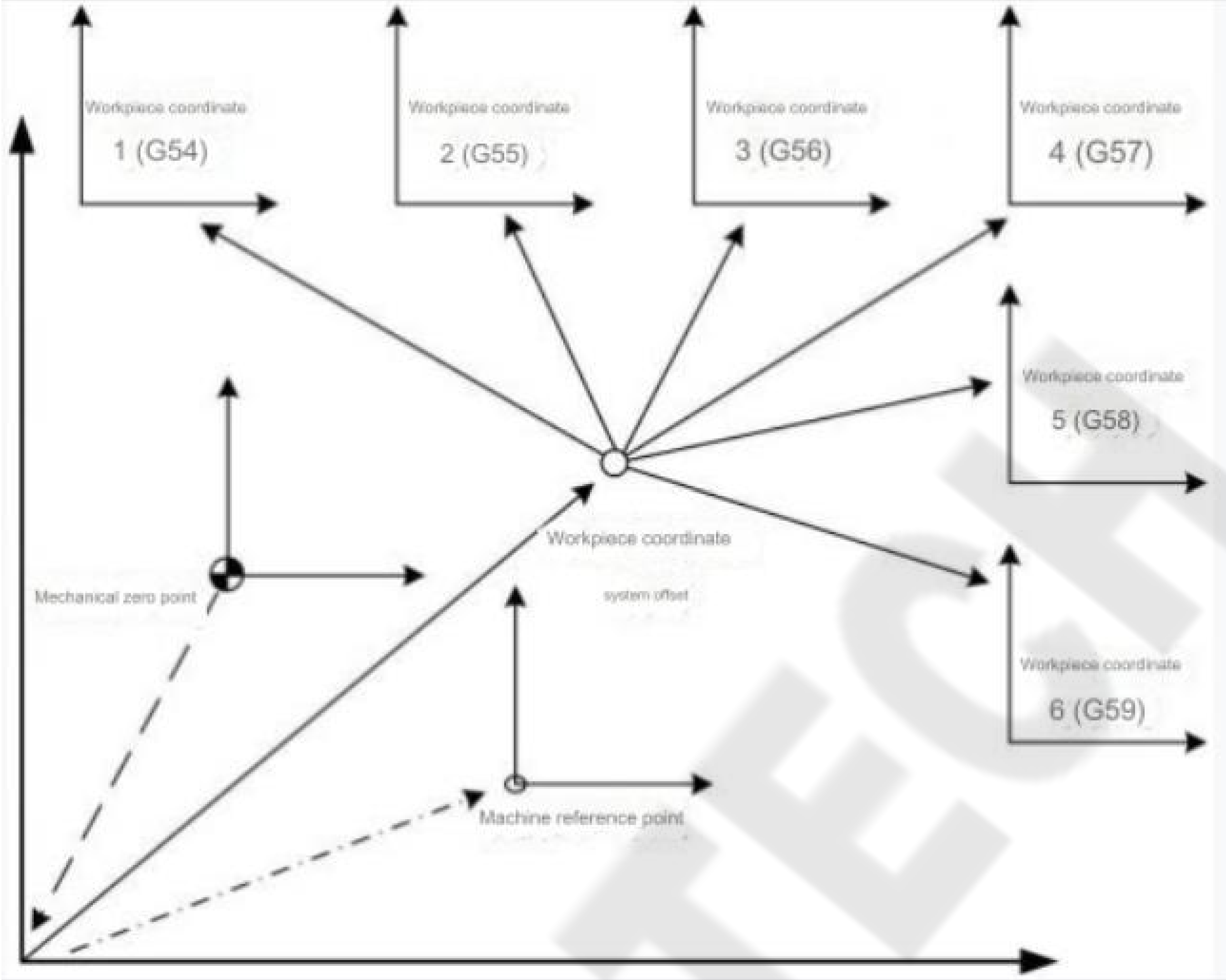


Figure 3.11.2

As shown in the figure above, after the machine tool is turned on, return to the mechanical origin by manual homing, and the machine tool coordinate system is established by the mechanical origin, thereby generating the machine tool reference point and determining the workpiece coordinate system. The value corresponding to the external workpiece origin offset data parameter N0096 is the overall offset of the six workpiece coordinate systems. The origin of the six workpiece coordinate systems can be specified by inputting the coordinate offset in the input mode or setting parameters N0090~N0095. These six workpiece coordinate systems are set according to the distance from the mechanical origin to the origin of each 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 to execute, it quickly locates to the position of the workpiece coordinate system G55 (X=100, Z=20). When the N20 program segment starts to execute, it quickly locates to 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).

3.14 G68 YAC axis is rotary axis tapping

1. Y axis is the rotary axis tapping

(1) XY dual-axis interpolation tapping, Y axis is the rotary axis, and X axis is the feed axis.

Instruction: G68 X(U)_ F_ Q_ P_ R_ J_ K

Description: Modal G code

X(U)_: X-axis target position;

F_: Pitch, F>0 right-hand thread, F<0 left-hand thread, unit: mm;

Q_: Y-axis speed, unit: rpm

P_: After the tapping is in place, delay, then execute reverse exit, unit: second

R_: Indicates the absolute coordinate value of point R

J_: Cutting depth of each cutting feed, unit: mm;

K_: Number of repetitions

Parameter 199: Set to 0, standard mode, return to point R each time); set to 1, high-speed mode, retract d each time, parameter 206 sets the retract amount d.

(2) YZ dual-axis interpolation tapping, Y axis is the rotation axis, Z axis is the feed axis

Instruction: G69 Z(W)_ F_ Q_ P_ R_ J_ K_

Description: Modal G code

Z(W)_: Z axis target position;

F_: Pitch, F>0 right-hand thread, F<0 left-hand thread, unit: mm;

Q_: Y-axis speed, unit: rpm

P_: After the tapping is in place, delay, then execute reverse exit, unit: second

R_: Indicates the absolute coordinate value of point R

J_: Cutting depth of each cutting feed, unit: mm;

K_: Number of repetitions

Parameter 199: Set to 0, standard mode, return to point R each time); set to 1, high-speed mode, retract d each time, parameter 206 sets the retract amount d.

(3) XZAB and Y axis interpolation tapping, Y axis is the rotation axis, XZAB axis is the feed axis;

Instruction: G68.2 X(U)_ Z(W)_ A(E)_ B_ F_ Q_ P_

Description: Non-modal G code

X(U)_: X-axis target position;

Z(W)_: Z axis target position;

A(E)_: A axis target position;

B_: B axis target position;

F_: Pitch, F>0 right-hand thread, F<0 left-hand thread, unit: mm;

Q_: Y-axis speed, unit: rpm

P_: After the tapping is in place, delay, then execute reverse exit, unit: second

2. A axis is the rotation axis tapping

(1) XA dual axis interpolation tapping, A axis is the rotation axis, X axis is the feed axis;

Instruction: G68.1 X(U)_ F_ Q_ P_ R_ J_ K_

Description: Modal G code

X(U)_: X-axis target position;

F_: Pitch, F>0 right-hand thread, F<0 left-hand thread, unit: mm;

Q_: Y-axis speed, unit: rpm

P_: After the tapping is in place, delay, then execute reverse exit, unit: second

R_: Indicates the absolute coordinate value of point R

J_: Cutting depth of each cutting feed, unit: mm;

K_: Number of repetitions

Parameter 199: Set to 0, standard mode, return to point R each time); set to 1, high-speed mode, retract d each time, parameter 206 sets the retract amount d.

(2) ZA dual-axis interpolation tapping, A axis is the rotation axis, Z axis is the feed axis;

Instruction: G69.1 Z(W)_ F_ Q_ P_ R_ J_ K_

Description: Modal G code

Z(W)_: Z axis target position;

F_: Pitch, F>0 right-hand thread, F<0 left-hand thread, unit: mm;

Q_: Y-axis speed, unit: rpm

P_: After the tapping is in place, delay, then execute reverse exit, unit: second

R_: Indicates the absolute coordinate value of point R

J_: Cutting depth of each cutting feed, unit: mm;

K_: Number of repetitions

Parameter 199: Set to 0, standard mode, return to point R each time); set to 1, high-speed mode, retract d each time, parameter 206 sets the retract amount d.

(3) XZYB and A axis interpolation tapping, A axis is the rotary axis, XZYB axis is the feed axis;
Instruction: G69.2 X(U)_ Z(W)_ Y(V)_ B_ F_ Q_ P_
Description: Non-modal G code
X(U)_: X-axis target position;
Z(W)_: Z axis target position;
Y(V)_: Y axis target position;
B_: B axis target position;
F_: Pitch, F>0 right-hand thread, F<0 left-hand thread, unit: mm;
Q_: Y-axis speed, unit: rpm
P_: After the tapping is in place, delay, then execute reverse exit, unit: second

3. C axis is rotary axis tapping

(1) End face rigid tapping G84 X(U)_ C(H)_ Z(W)_ P_ Q_ F(I)_ K_

(2) Side rigid tapping G88 Z(W)_ C(H)_ X(U)_ P_ F(I)_ K_
Description: Modal G code
(X, C): tapping hole position; ------G84
Z: tapping hole bottom position; ------G84
(Z, C): tapping hole position; ------G88
X: tapping hole bottom position; ------G88
P: tapping to hole bottom pause time, unit: milliseconds
Q_: each feed amount, unit: mm
F(I): thread lead, F(I) > 0 right-hand tapping, F(I) < 0 left-hand tapping
K: tapping repetition times
Example:
(Parameter 198 is changed to 1)
M29 S100
G84 W-3 F1

3.15 Fixed Cycle Code

In order to simplify programming, the SZGH880T/SZGH1080T series provides a single machining cycle G code that uses only one program segment to complete quick movement positioning, linear/thread cutting, and finally 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 function section.

3.15.1 Axial Cutting Cycle G90

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

G90 X/U_ Z/W_ R_ F_; (conical cutting)

Code function: Starting from the cutting point, radial (X axis) feed and axial (Z axis or X and Z axis at the same time) cutting are performed to realize cylindrical or conical surface cutting cycle.

Code description: G90 is a modal code;

Cutting start point: the starting position of linear interpolation (cutting feed);
Cutting end point: the ending position of linear 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 start point;

Z: 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 start 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 signs of R and U are inconsistent, │R│≤│U/2│ is required; when R＝0 or default input, cylindrical cutting is performed, as shown in Figure 3-17, otherwise conical cutting is performed, as shown in Figure 3-18.

Cycle process: ① X axis moves quickly from the start point to the cutting start point;

② Linear interpolation (cutting feed) from the cutting start point to the cutting end point;

③ X axis retracts at the cutting feed speed and returns to the position where the absolute coordinate of the X axis is the same as the start point;

④ Z axis moves quickly back to the start point, and the cycle ends.

Figure 3-17

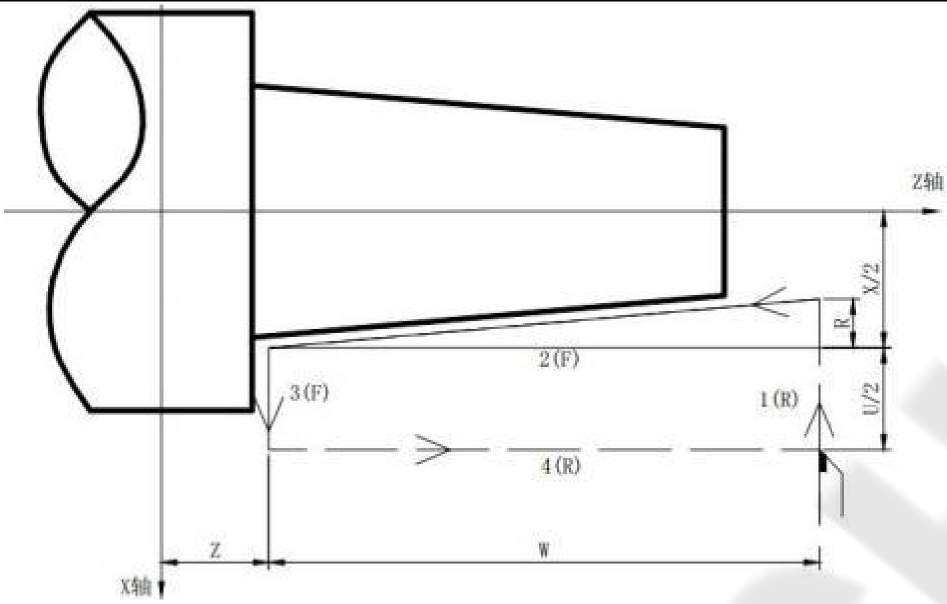
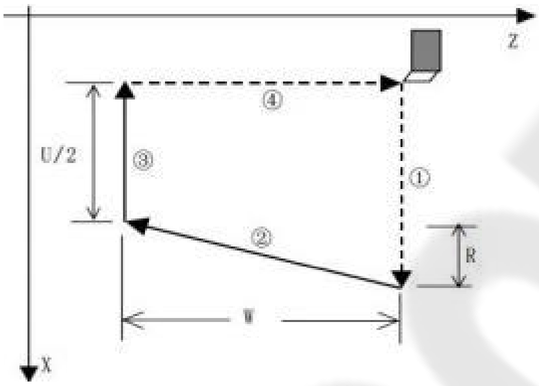


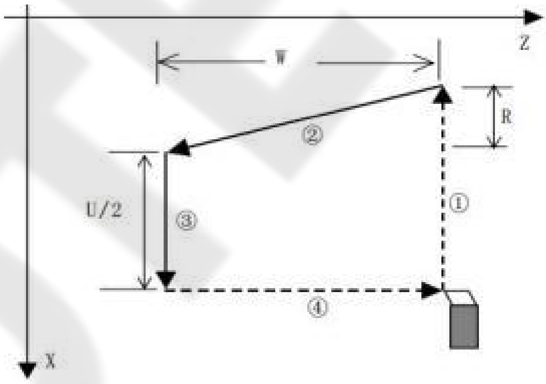
Figure 3-18

Code trajectory: U, W, and R reflect the relative position of the cutting end point and the start point. The tool trajectory of U, W, and R combined when the symbols are different is shown in Figure 3-19.

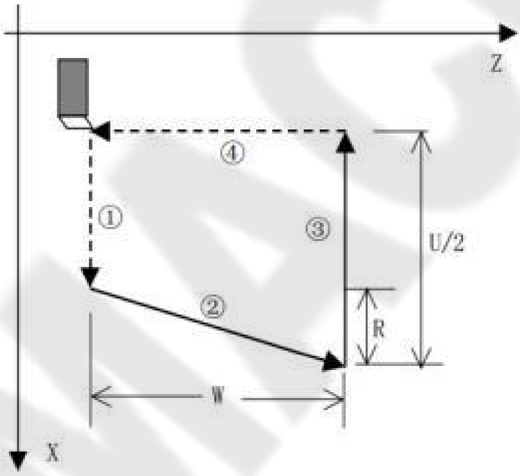
1) U>0, W<0, R>0



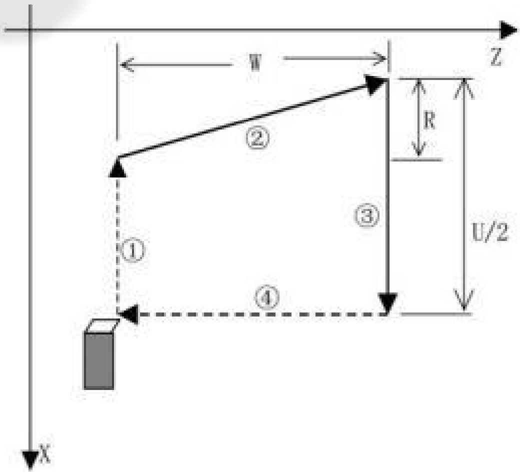
2) U<0, W<0, R<0



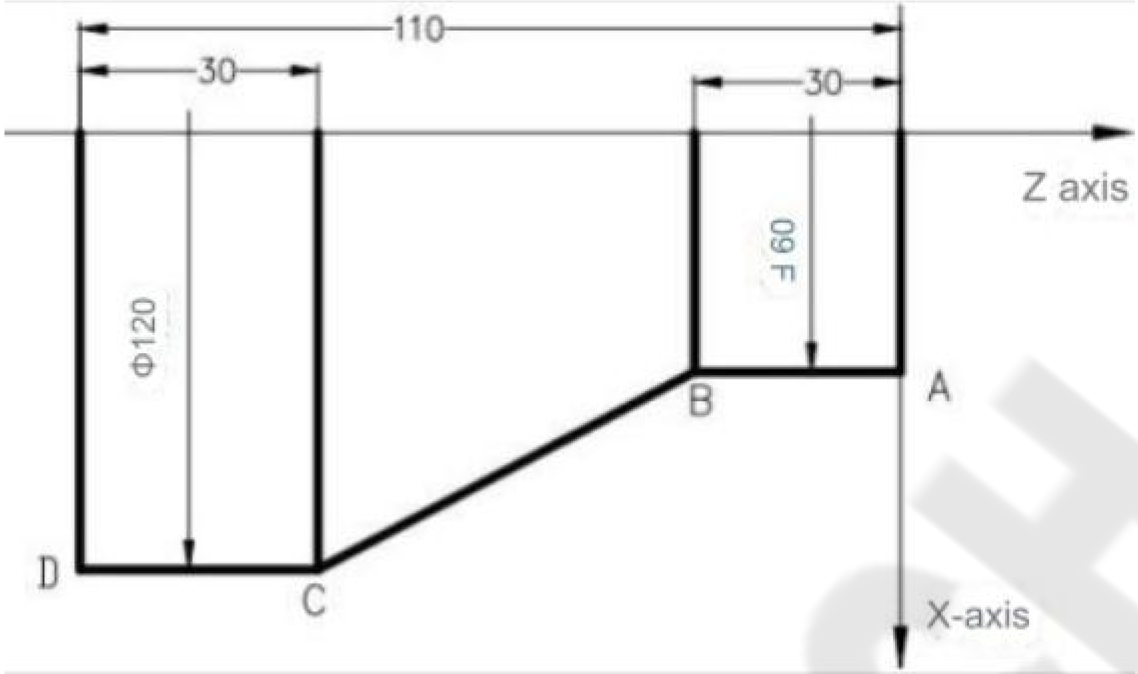
3) U>0, W>0, R<0, │ R │≤│ U/2 │



4) U<0, W>0, R>0, │ R │≤│ U/2 │



Example: Figure 3-20, blank Φ125×110



Program: O0002;

M3 S300 G0 X130 Z3;

G90 X120 Z-110 F200; (A → D, Φ120 cutting)

X110 Z-30;

X100;

X90;

X80;

X70;

X60;

(A→B, Φ60 cutting, six-step feed cycle cutting, 10mm each time)

G0 X120 Z-30;

G90 X120 Z-44 R-7.5 F150;

Z-56 R-15

Z-68 R-22.5
Z-80 R-30

(B→C, taper cutting, four-step feed cycle cutting)

3.15.2 Radial Cutting Cycle G94

Code format: G94 X/U_ Z/W_ F_; (end face cutting)

G94 X/U_ Z/W_ R_ F_; (taper end face cutting)

Code function: Start from the cutting point, axial (Z axis) feed, and radial (X axis or X and Z axis at the same time) cutting, to achieve end face or taper cutting cycle, the start and end points of the code are the same.

Code description: G94 is a modal code;

Cutting start point: the starting position of linear interpolation (cutting feed);
Cutting end point: the ending position of linear interpolation (cutting feed);

X: absolute coordinate of the X axis of the cutting end point, unit: mm/inch;

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

Z: 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 start point;

R: the difference between the absolute coordinate of the Z axis of the cutting start point and the cutting end point. When the signs of R and U are different, $|R| \leq |W|$ is required. Radial linear cutting is shown in Figure 3-21, and radial taper cutting is shown in Figure 3-22.

Cycle process: ① Z axis moves quickly from the start point to the cutting start point;

② Linear interpolation (cutting feed) from the cutting start point to the cutting end point;

③ Z axis retracts at the cutting feed speed (opposite to ①) and returns to the same position of the absolute coordinate of the Z axis as the start point;

④ X axis moves quickly back to the start point, and the cycle ends.

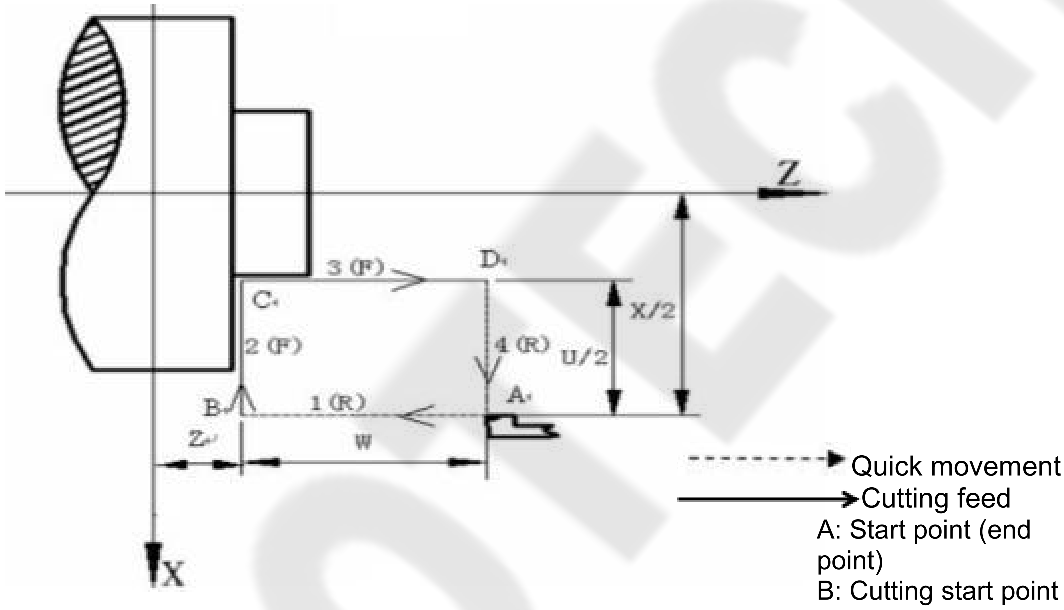


Figure 3-21

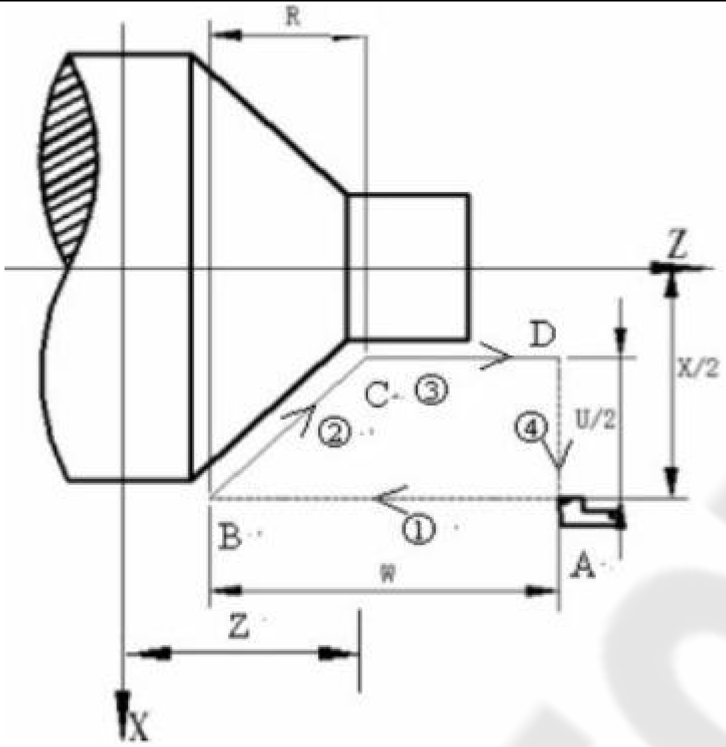
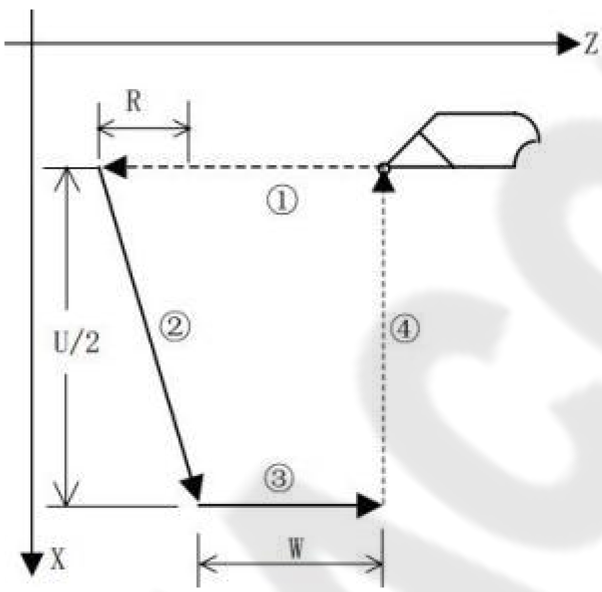


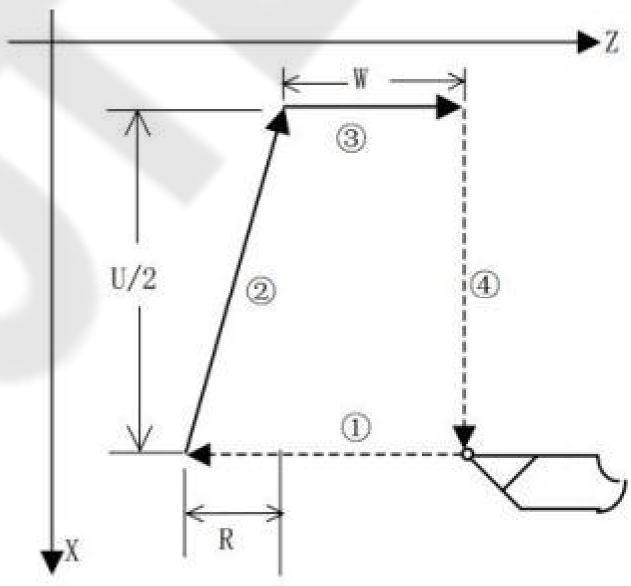
Figure 3-22

Code trajectory: U, W, and R reflect the relative position of the cutting end point and the start point. The tool trajectory of U, W, and R combined when the symbols are different is shown in Figure 3-23:

1) U>0 W<0 R<0

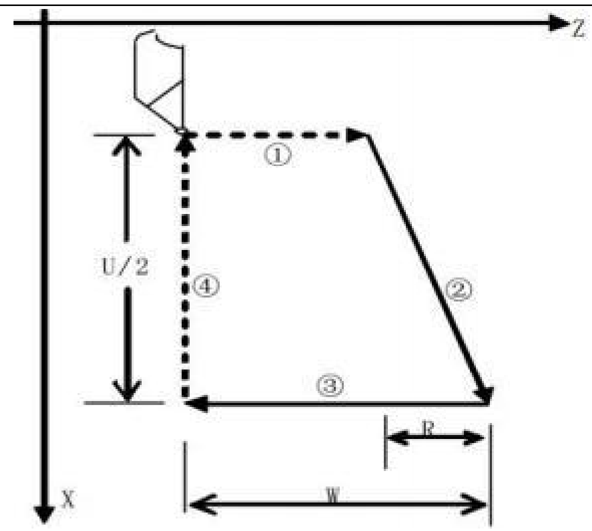


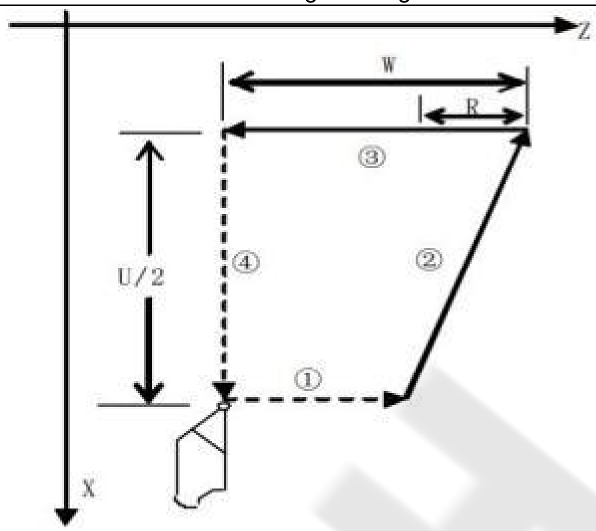
2) U<0 W<0 R<0



3) U>0 W>0 R<0 (|R|≤|W|)

4) U<0 W>0 R<0 (|R|≤|W|)





Example: Figure 3-24, blank Φ125×112

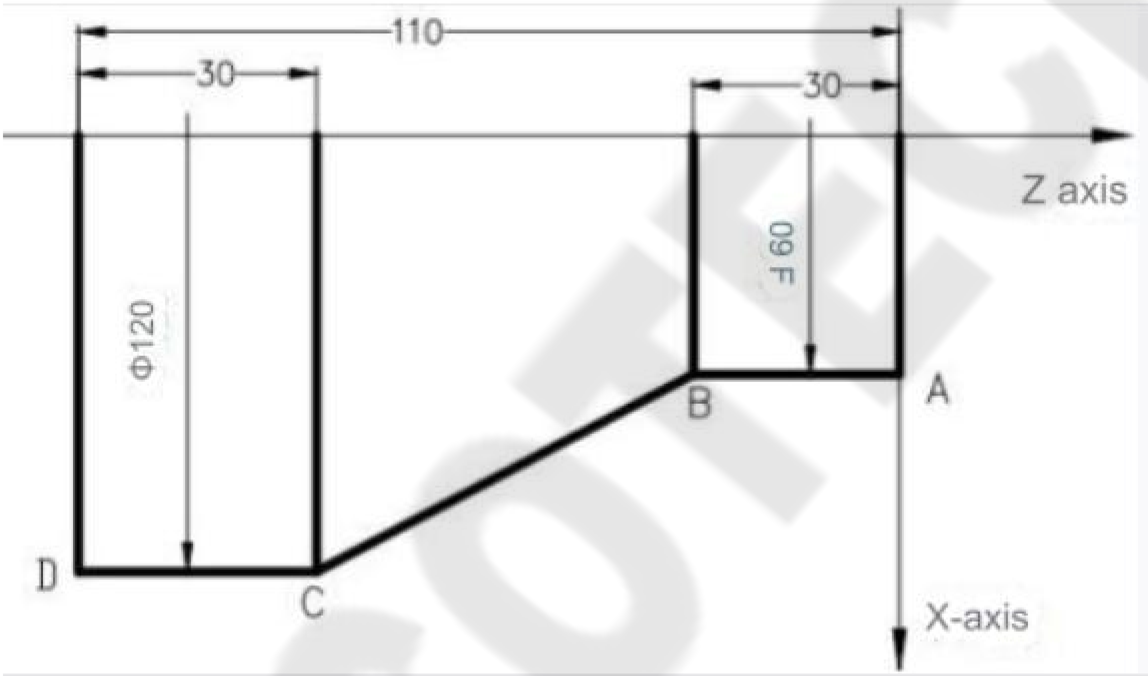


Figure 3-24

Program: O0003;

G00 X130 Z5 M3 S1;
G94 X0 Z0 F200
X120 Z-110 F300;
G00 X120 Z0

End face cutting (outer circle Φ120 cutting)

G94 X108 Z-30 R-10

X96 R-20
X84 R-30
X72 R-40
X60 R-50;

(C→B→A, Φ60 cutting)

M30;

3.15.3 Notes on Fixed Cycle Codes

1) In a fixed cycle code, once X/U, Z/W, and R are executed, the specified values of X/U, Z/W, and R remain valid when no new fixed cycle code is executed to re-assign X/U, Z/W, and R. If a non-modal (00 group) G code other than G04 or G00, G01, G02, G03, or G32 is executed, the specified values of X/U, Z/W, and R are cleared.

2) In the fixed cycle G90 and G94 codes, if single-segment operation is performed, the single-segment stop is performed after the entire fixed cycle is completed.

3.16 Multiple Cycle Codes

The multiple cycle codes of the SZGH880T/SZGH1080T series include: Axial rough turning cycle G71, radial rough turning cycle G72, closed cutting cycle G73, finishing cycle G70, axial grooving multiple cycles G74, radial grooving multiple cycles G75 and multiple thread cutting cycle G76. When the system executes these codes, it automatically calculates the number of cutting times and cutting trajectories based on the programmed trajectory, feed amount, retract amount and other data, and performs multiple feed → cutting → retract → re-feed processing cycles to automatically complete the rough and finish machining of the workpiece blank. The start point and end point of the code are the same.

The G76 multiple thread cutting cycle code is described in the thread function section.

3.16.1 Axial Rough Turning Cycle G71

G71 has two types of rough turning cycles: Type I and Type II

Code format:

G71 U(Δd) R(e) F_ S_ T_; (1)
G71 P(ns) Q(nf) U(Δu) W(Δw) K0/1; (2)

Type I		Type II	
N(ns) G0/G1 X/U. . ;	(3)	N(ns) G0/G1 X(U) Z(W)…;	(3)
. . . . ;		. . . . ;	
. . . . . F;		. . . . . F;	
. . . . . S;		. . . . . S;	
. . . .		. . . .	
N(nf). . . . . ;		N(nf). . . . . ;	

Code function: G71 code is divided into three parts:

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

(2): The program segment interval with given finishing trajectory and program segment of finishing allowance;

(3): Several consecutive program segments defining the finishing trajectory. When executing G71, these program segments are only used to calculate the rough turning trajectory and are not actually executed.

The system automatically calculates the rough machining route based on the finishing trajectory, finishing allowance, feed amount, retract amount and other data, cuts in the direction parallel to the Z axis, and completes the rough machining of the workpiece through multiple feed→cut→retract cutting cycles. The start point and end point of G71 are the same. This code is suitable for rough turning of non-molded blanks (bars).

Related definitions:

Finishing trajectory: The workpiece finishing trajectory given by the part (3) of the code (ns~nf program segments). The start point of the finishing trajectory (i.e. the start point of the ns program segment) is the same as the start point and end point of G71, referred to as point A; the first segment of the finishing trajectory (ns program segment) can only be the quick movement or cutting feed of the X axis, 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 program segment) is referred to as point C. The finishing trajectory is point A→point B→point C.

Rough turning contour: The trajectory of the finishing trajectory after offsetting according to the finishing allowance (Δu, Δw) is the trajectory contour formed by executing G71. Points A, B, and C of the finishing trajectory correspond to points A', B', and C' of the rough turning contour after offsetting. The final continuous cutting trajectory of the G71 code is point B'→point C'.

Δd: Cutting amount of X axis during rough turning, value range 0.001~99.999 (IS_B)/0.0001~99.9999 (IS_C) (unit: mm/inch, radius value), unsigned, the feed direction is determined by the moving direction of the ns program segment. After U(Δd) is executed, the specified value Δd is kept, and the data is converted to the corresponding value and saved in parameter N0187. When U(Δd) is not entered, the value of parameter N0187 is used as the feed amount.

e: Retraction amount of X axis during rough turning, value range 0~99.999 (unit: mm/inch, radius value) under (IS_B)/(IS_C), unsigned, the retraction direction is opposite to the feed direction; after R(e) is executed, the specified value e is kept, and the data is converted to the corresponding value and saved in parameter N0188. When R(e) is not entered, the value of parameter N0188 is used as the retraction amount.

ns: The segment number of the first segment of the fine turning trajectory;

nf: The segment number of the last segment of the fine turning trajectory.

Δu: The finishing allowance of the X axis, value range is -99999.999~99999.999 (IS_B)/-9999.9999~9999.9999 (IS_C) (diameter, unit: mm/inch, signed), the X axis coordinate offset of the rough turning contour relative to the fine turning trajectory, that is: the difference between the absolute X axis coordinates of point A' and point A. When U(Δu) is not entered, the system processes it as Δu=0, that is: the X axis of the rough turning cycle does not leave a finishing allowance.

Δw: Z-axis finishing allowance, value range -99999.999~99999.999 (IS_B)/-9999.9999~9999.9999 (IS_C), (unit: mm/inch, signed), the Z-axis coordinate offset of the rough turning contour relative to the finishing trajectory, that is: the difference between the absolute Z axis coordinates of point A' and point A. When

W (Δw) is not entered, the system processes it as Δw=0, that is: the rough turning cycle Z axis does not leave finishing allowance.

K: When K is not entered or K is not 1, the system does not check the monotonicity of the program except that the Z values of the start and end points of the arc, ellipse or parabola are equal or the arc is greater than 180 degrees; when K=1, the system checks the monotonicity of the program.

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

M, S, T, F: can be specified in the first G71 code or the second G71 code, or in the ns~nf program. In the G71 cycle, the M, S, T, and F functions of the program segment numbers between ns~nf are invalid, and are only valid in the program segment with the G70 fine turning cycle.

Type I:

1) Code execution process: Figure 3-25.

① Quickly move from the start point A to point A’, the X axis moves Δu, and the Z axis moves Δw;

② The X axis moves Δd (feed) from point A’, feeds at the rapid moving speed when the ns program segment is G0, feeds at the cutting feed speed F of G71 when the ns program segment is G1, and the feed direction is consistent with the direction of point A→point B;

③ The Z axis feeds to the rough turning contour, and the feed direction is consistent with the change of the Z axis coordinate from point B→point C;

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

⑤ The Z axis retracts to the position with the same absolute coordinate of the Z axis of point A’ at the rapid moving speed;

⑥ After the X axis feeds again (Δd+e), if the end point of the movement is still in the middle of the connection of point A’→point B’ (not reaching or exceeding point B’), the X axis feeds again (Δd+e), then execute ③; if the end point of the movement reaches point B’ or exceeds the connection of point A’→point B’ after the X axis feeds again (Δd+e), the X axis feeds to point B’, then execute ⑦;

⑦ Cut and feed from point B’ to point C’ along the rough turning contour;

⑧ Quickly move from point C’ to point A, the G71 cycle execution ends, and the program jumps to the next program segment of the nf program segment for execution.

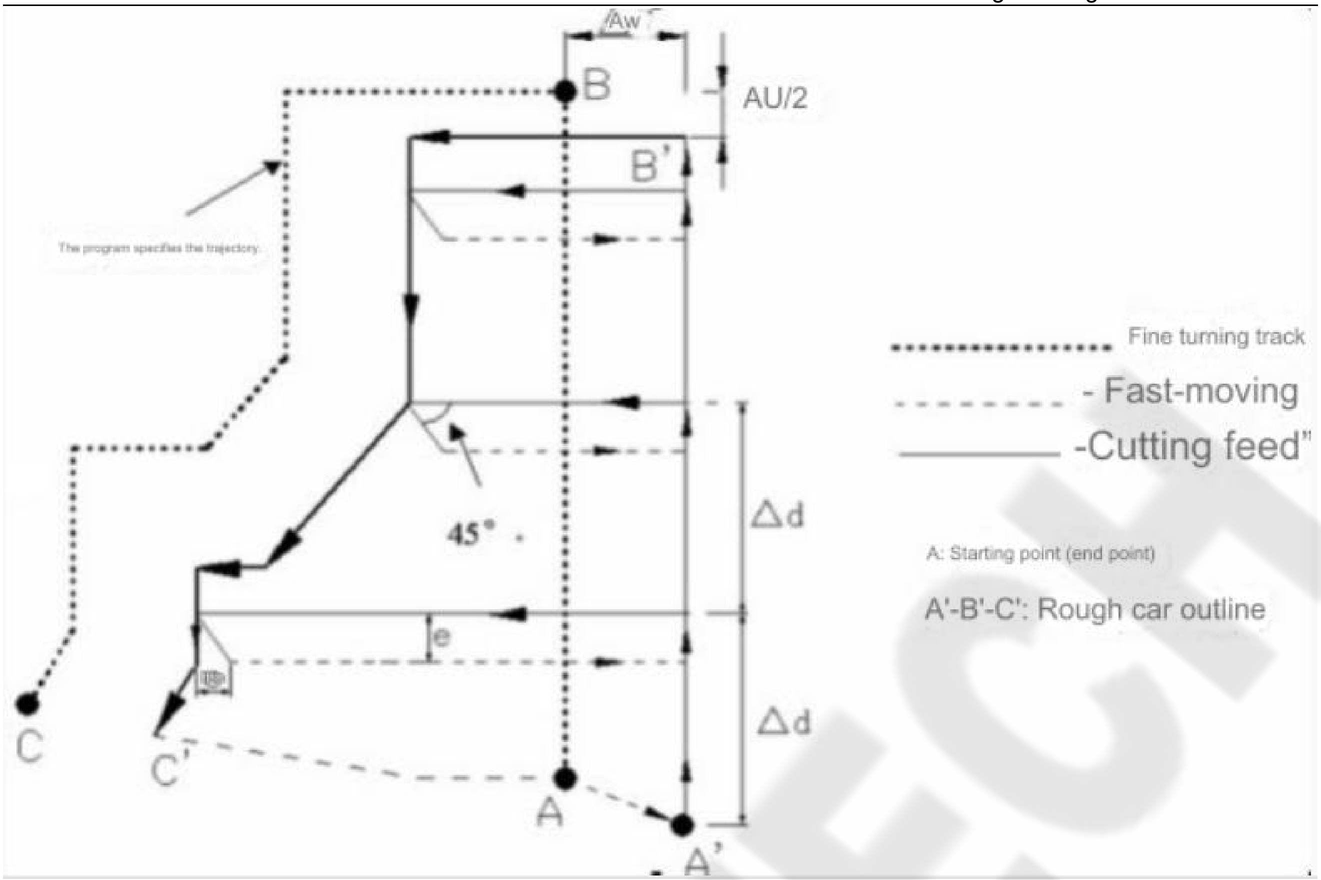


Figure 3-25 G71 code cycle trajectory

2) Coordinate offset direction when leaving fine turning allowance:

Δu, Δw reflect the coordinate offset and cutting direction during fine turning. There are four different combinations according to the signs of Δu and Δw, as shown in Figure 3-26. In the figure, B→C is the fine turning trajectory, B'→C' is the rough turning contour, and A is the start point.

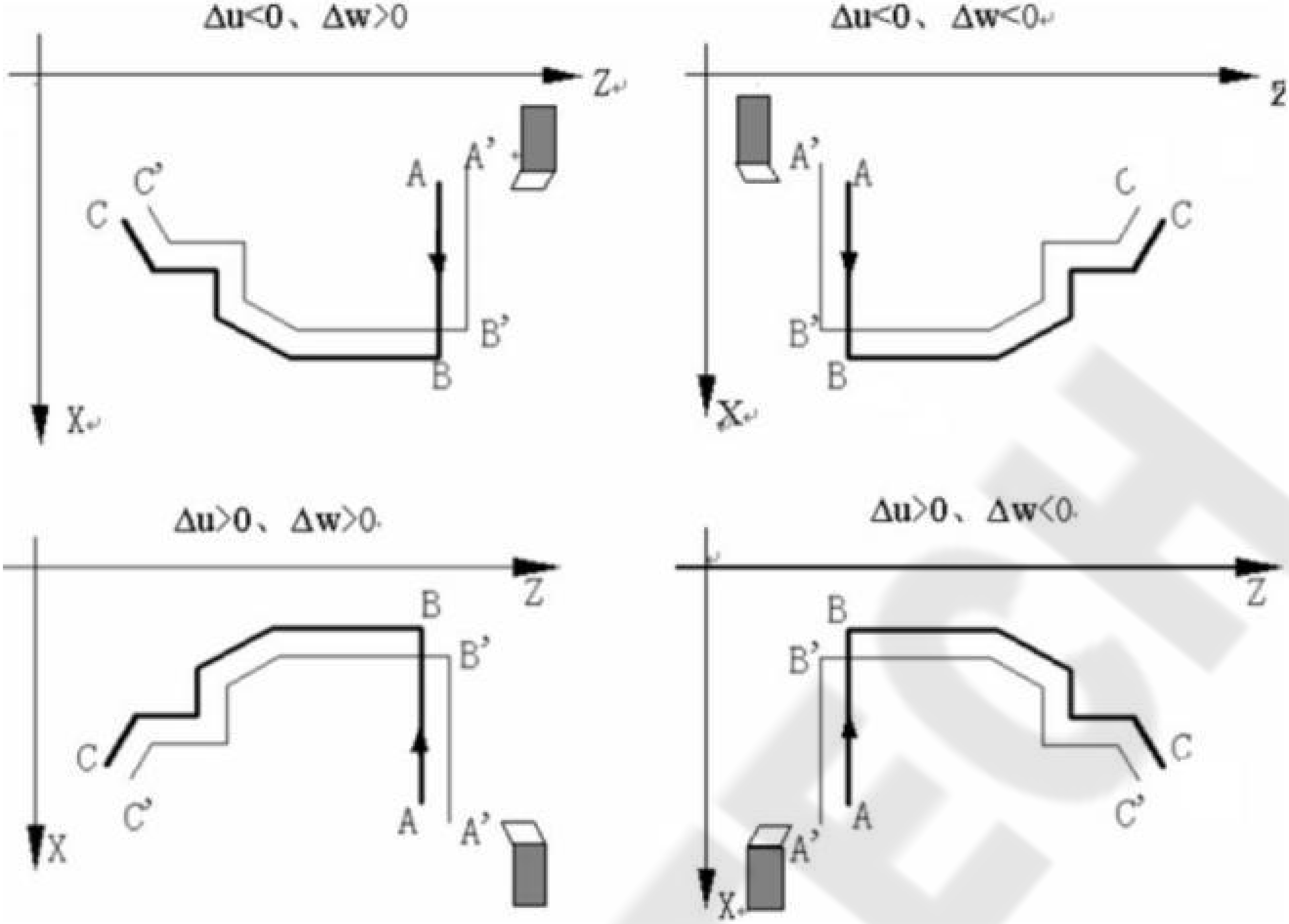


Figure 3-26

Type II:

Type II is different from Type I as follows:

1) Related definitions: 1 more parameter than Type I

J: When J is not input or J is not 1, the system will not run along the rough turning contour again; when J=1, the system will run along the rough turning contour again

2) The contour along the X axis does not have to be monotonically increasing or monotonically decreasing, and there can be a maximum of 10 grooves, as shown below.

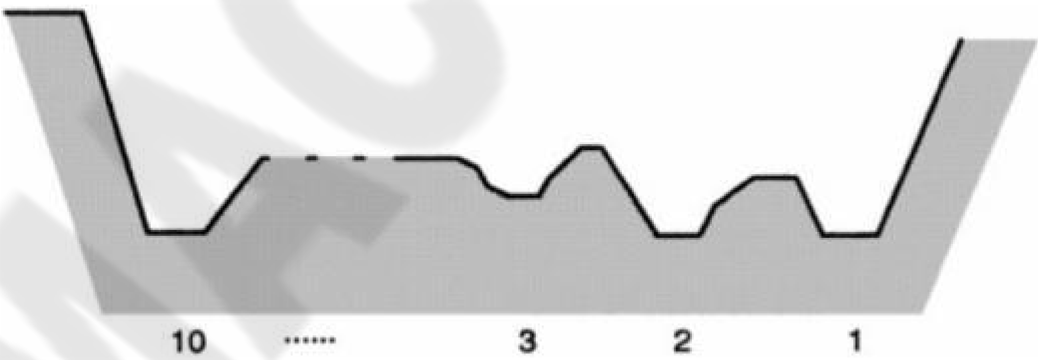


Figure 3-26-1 (Type II)

However, the contour along the Z axis must be monotonically increasing or decreasing, and the following contour cannot be processed:

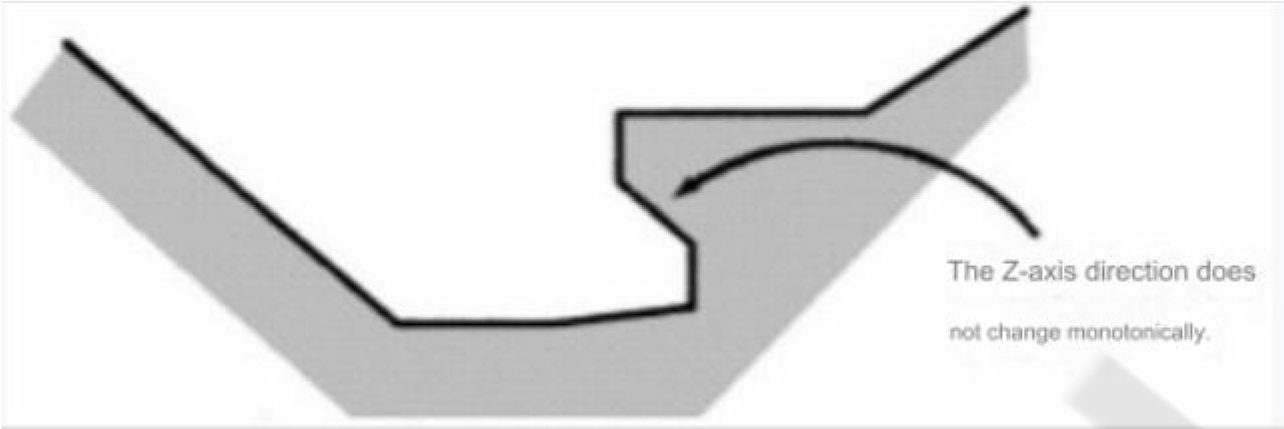


Figure 3-26-2 (Type II)

3) The first cut does not have to be vertical: If the shape along the Z axis is monotonically changing, it can be processed. The schematic diagram is as follows:

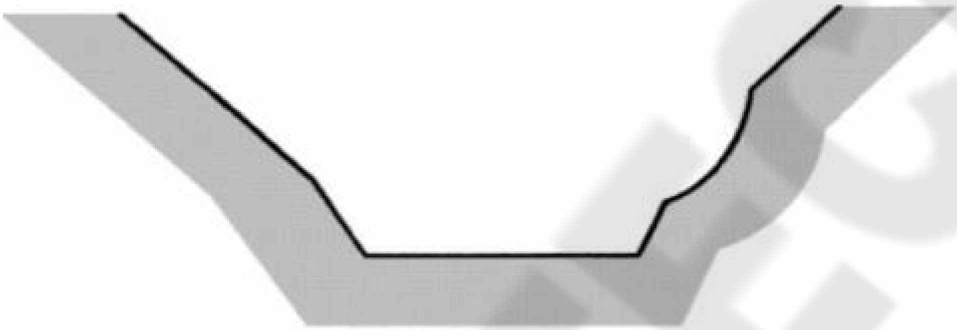


Figure 3-26-3 (Type II)

4) After turning, the tool should be retracted. The retraction amount is specified by the R (e) parameter or the setting value of parameter N0188. The schematic diagram is as follows:

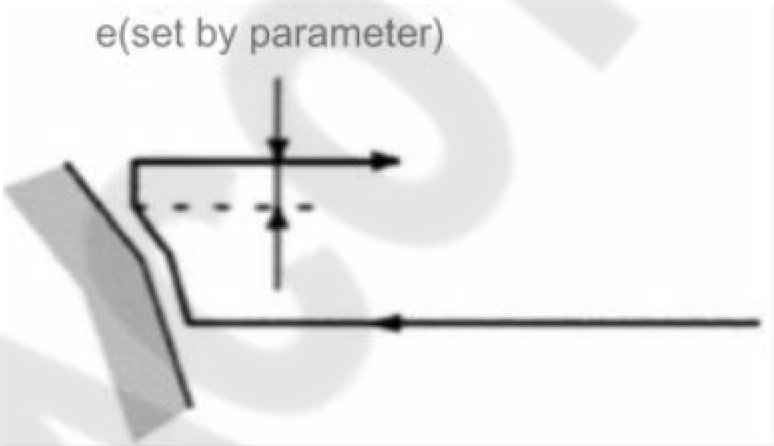
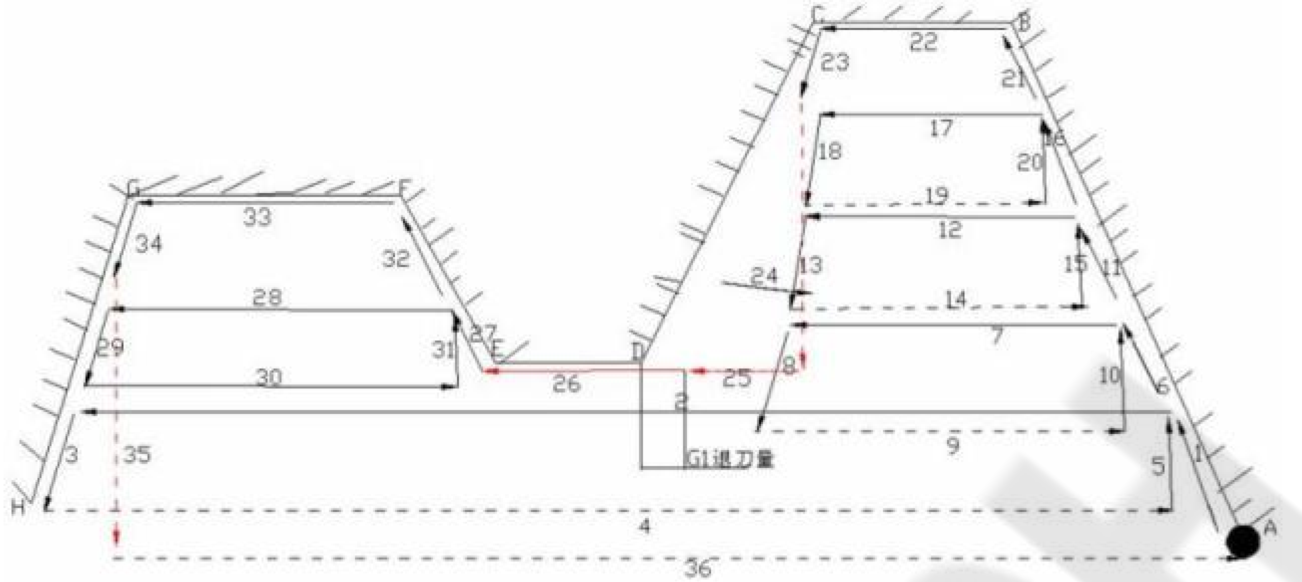


Figure 3-26-4 (Type II)

5) Code execution process: rough turning trajectory A->H



Notes:

- The ns program segment can only be G00 and G01 codes. If it is type II, the X (U) and Z (W) axes must be specified. When the Z axis does not move, W0 must also be specified.
- For type II, the finishing allowance can only be specified in the X direction. If the finishing allowance in the Z direction is specified, the entire processing trajectory will be offset. If specified, it is best to specify it as 0.
- For type II, when the current groove is cut and the next groove is to be cut, the tool is left with a retraction distance to allow the tool to approach the workpiece at the speed of G1 (labels 25 and 26). If the retraction amount is 0 or the remaining distance is less than the retraction amount, the system approaches the workpiece at G1.
- For the parts that are not marked as type I or type II, they are common to both.
- For the fine turning trajectory (ns~nf program segment), the Z axis size must be monotonically changed (always increasing or decreasing), and the X axis size must also be monotonically changed in type I, but not required for type II.
- The ns~nf program segment must be written immediately after the G71 program segment. If it is written before the G71 program segment, the system automatically searches for the ns~nf program segment and executes it. After execution, the next program of the G71 program segment is executed in sequence, which will cause the G71 program segment to be executed repeatedly.
- When executing G71, the ns~nf program segment is only used to calculate the rough turning contour, and the program segment is not executed. The F, S, and T codes in the ns~nf program segment are invalid when executing the G71 cycle and valid when executing the G70 finishing cycle.
- In the ns~nf program segment, there can only be G functions: G00, G01, G02, G03, G04, G05, G6.2, G6.3, G7.2, G7.3, G96, G97, G98, G99, G40, G41, and G42 codes; there cannot be subroutine call codes (such as M98/M99).
- The G96, G97, G98, G99, G40, G41, G42, and G04 codes are invalid when executing the G71 cycle, but are valid when executing the G70 finishing cycle.
- During the execution of G71 code, you can stop automatic operation and move manually, but to execute G71 cycle again, you must return to the position before manual movement. If you continue to execute without returning, the subsequent running trajectory will be misplaced.

● Execute feed hold and single program segment operations, and the program will pause after running to the end point of the current trajectory.

● Δd and Δu are both specified by the same address U, and the distinction is based on whether the program segment specifies P and Q codes.

● G71 code cannot be executed in input mode, or an alarm will be generated.

● When compound cycle codes need to be used multiple times in the same program, ns~nf are not allowed to have the same program segment number.

● The retraction point should be as high or low as possible to avoid retraction hitting the workpiece.

Example: Figure 3-27 (Type I)

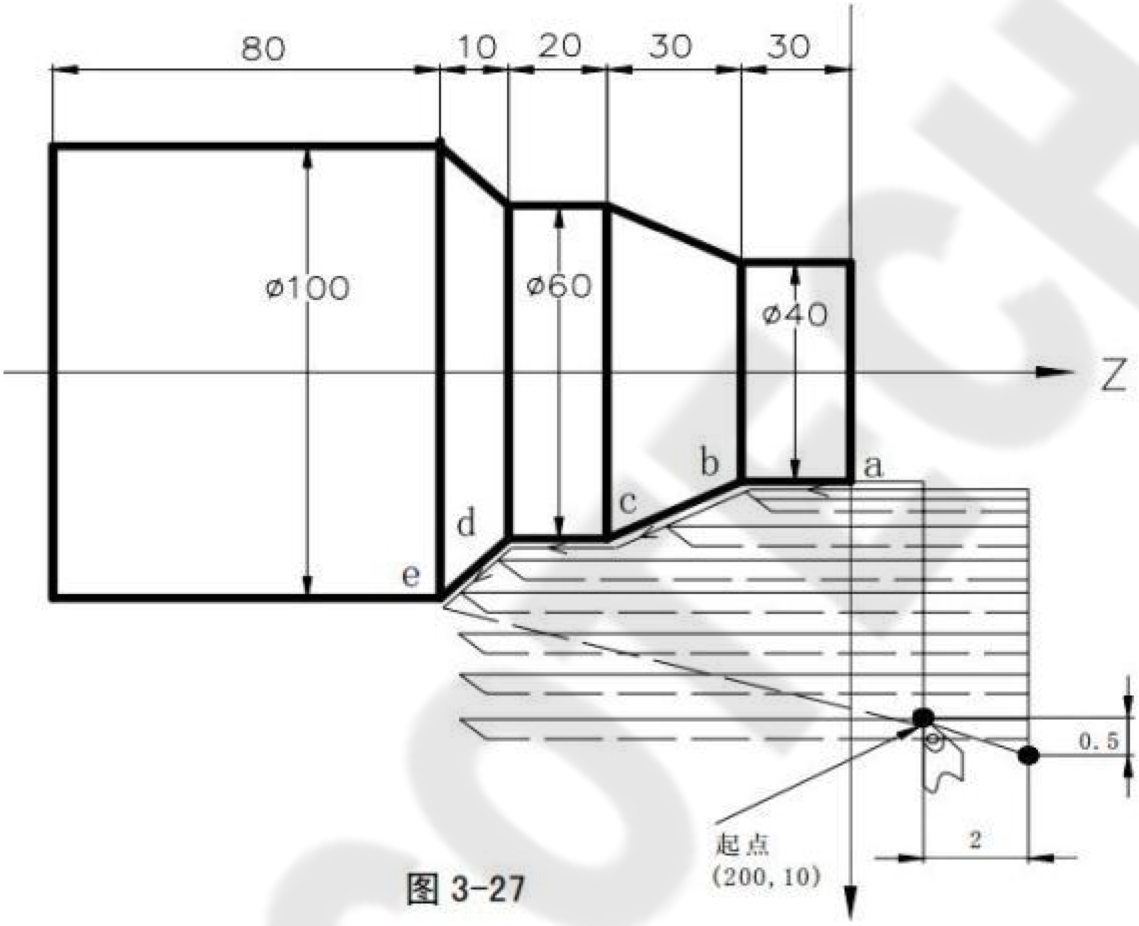


Figure 3-27

Program: O0004;

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

G71 U2 R1 F200; (4mm cutting depth each time, 2mm tool retraction, [diameter])

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

N80 G00 X40 S1200;	(Positioning)	Finishing route a→b→c→d→e program segment
G01 Z-30 F100;	(a → b)	
X60 W-30;	(b → c)	
W-20;	(c → d)	
N120 X100 W-10;	(d → e)	

G70 P80 Q120; (fine turning of a---e)

M30; (end of program)

3.16.2 Radial Rough Turning Cycle G72

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

G72 P(ns) Q(nf) U(Δu) W(Δw) K0/1; (2)

N_ (ns) . . . . . ;

. . . . ;

. . . . F;

. . . . S; (3)

. . . . ;

.

N_ (nf). . . . . ;

Code function: G72 code is divided into three parts:

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

(2): The program segment interval with given finishing trajectory and program segment of finishing allowance;

(3): Several consecutive program segments defining the finishing trajectory. When executing G72, these program segments are only used to calculate the rough turning trajectory and are not actually executed.

The system automatically calculates the rough machining route based on the finishing trajectory, finishing allowance, feed amount, retract amount and other data, cuts in the direction parallel to the X axis, and completes the rough machining of the workpiece through multiple feed→cut→retract cutting cycles. The start point and end point of G72 are the same. This code is suitable for rough turning of non-molded blanks (bars).

Related definitions:

Finishing trajectory: The workpiece finishing trajectory given by the part (3) of the code (ns~nf program segments). The start point of the finishing trajectory (i.e. the start point of the ns program segment) is the same as the start point and end point of G72, referred to as point A; the first segment of the finishing trajectory (ns program segment) can only be the quick movement or cutting feed of the Z axis, 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 program segment) is referred to as point C. The finishing trajectory is point A→point B→point C.

Rough turning contour: The trajectory of the finishing trajectory after offsetting according to the finishing allowance (Δu, Δw) is the trajectory contour formed by executing G72. Points A, B, and C of the finishing trajectory correspond to points A’, B’, and C’ of the rough turning contour after offsetting. The final continuous cutting trajectory of the G72 code is point B’→point C’.

Δd: The cutting amount of the Z axis during rough turning, value range is 0.001 (IS_B)/0.0001 (IS_C) ~ 99.999 (unit: mm/inch), unsigned, the feed direction is determined by the moving direction of the ns program segment. After W(Δd) is executed, the specified value Δd is retained, and the data is converted to the corresponding value and saved in parameter N0187. When W(Δd) is not entered, the value of parameter N0187 is used as the feed amount.

e: The retract amount of the Z axis during rough turning, the value range is 0~99.999 (unit: mm/inch), unsigned, and the retract direction is opposite to the feed direction. After R(e) is executed, the specified value e is retained, and the data is converted to the corresponding value and saved in data parameter N0188. When R(e) is not entered, the value of parameter N0188 is used as the retraction amount.

ns: The segment number of the first segment of the fine turning trajectory.

nf: The segment number of the last segment of the fine turning trajectory.

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

Δw: The finishing allowance left on the Z axis during rough turning, value range is -99999.999 ~ 99999.999 (the Z-axis coordinate offset of the rough turning contour relative to the fine turning trajectory, that is: the difference between the absolute Z-axis coordinates of point A' and 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 rate;

S: Spindle speed; T: Tool number, tool offset number.

M, S, T, F: can be specified in the first G72 code or the second G72 code, or in the ns~nf program (except T instruction). In the G72 cycle, the M, S, and F functions of the program segment numbers between ns and nf are invalid and are only valid in the program segment with the G70 fine turning cycle.

Code execution process: Figure 3-28.

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

2. The Z axis moves Δd (feed) from point A', feeds at the rapid moving speed when the ns program segment is G0, feeds at the cutting feed speed F of G72 when the ns program segment is G1, and the feed direction is consistent with the direction of point A→point B;

3. The X axis feeds to the rough turning contour, and the feed direction is consistent with the change of the X axis coordinate from point B→point C;

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

5. The X axis retracts to the position with the same absolute coordinate of the Z axis of point A' at the rapid moving speed;

6. After the Z axis feeds again (Δd+e), if the end point of the movement is still in the middle of the connection of point A'→point B' (not reaching or exceeding point B'), the Z axis feeds again (Δd+e), then

execute ③; if the end point of the movement reaches point B' or exceeds the connection of point A'→point B' after the Z axis feeds again (Δd+e), the Z axis feeds to point B', then execute ⑦;

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

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

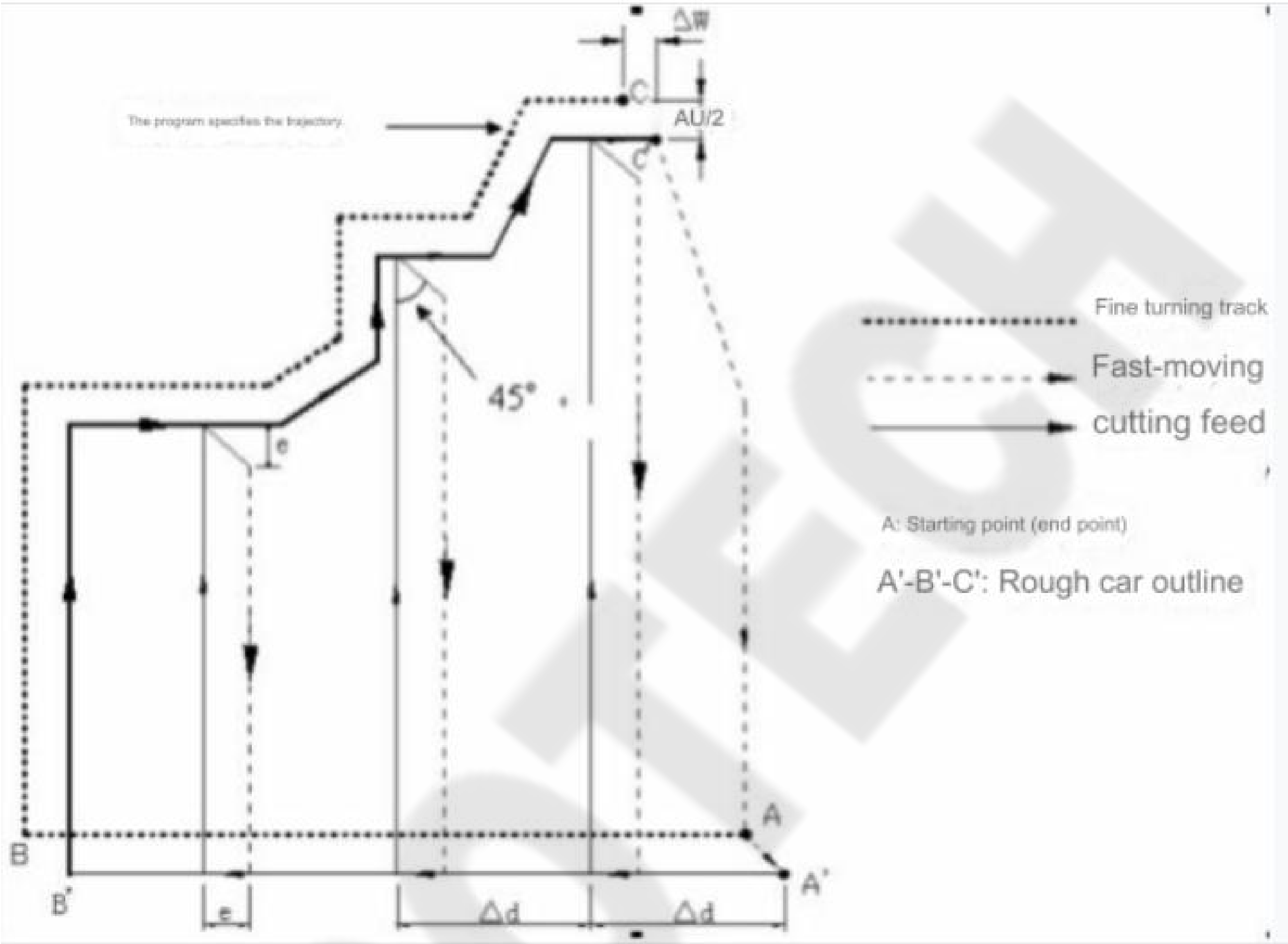


Figure 3-28

Code description:

● The ns~nf program segment must be written immediately after the G72 program. If it is written before the G72 program segment, the system automatically searches for the ns~nf program segment and executes it. After execution, the next program of the nf program segment is executed in sequence, which will cause the ns~nf program segment to be executed repeatedly.

● When executing G72, the ns~nf program segment is only used to calculate the rough turning contour, and the program segment is not executed. The F, S, and T codes in the ns~nf program segment are invalid when executing the G72 cycle and valid when executing the G70 finishing cycle When executing the G70 finishing cycle, the F, S, and T codes in the ns~nf program segment are valid.

● The ns program segment can only be G00 and G01 codes without X(U) code words, otherwise an alarm will be issued.

● For the fine turning trajectory (ns~nf program segment), the size of X axis and Z axis must be monotonically changed (always increasing or decreasing).

● In the ns~nf program segment, there can only be G functions: G00, G01, G02, G03, G04, G05, G6.2, G6.3, G7.2, G7.3, G96, G97, G98, G99, G40, G41, and G42 codes; there cannot be subroutine call codes (such as M98/M99).

● The G96, G97, G98, G99, G40, G41, and G42 codes are invalid when executing the G72 cycle, but are valid when executing the G70 finishing cycle.

● During the execution of G72 code, you can stop automatic operation and move manually, but to execute G72 cycle again, you must return to the position before manual movement. If you continue to execute without returning, the subsequent running trajectory will be misplaced.

● Execute feed hold and single program segment operations, and the program will pause after running to the end point of the current trajectory.

● Δd and Δw are both specified by the same address W, and the distinction is based on whether the program segment specifies P and Q code words.

● When compound cycle codes need to be used multiple times in the same program, ns~nf are not allowed to have the same program segment number.

● G72 code cannot be executed in input mode, or an alarm will be generated.

● The retraction point should be as high or low as possible to avoid retraction hitting the workpiece.

Coordinate offset direction when leaving fine turning allowance:

Δu, Δw reflect the coordinate offset and cutting direction during fine turning. There are four different combinations according to the signs of Δu and Δw, as shown in Figure 3-29. In the figure, B→C is the fine turning trajectory, B'→C' is the rough turning contour, and A is the start point.

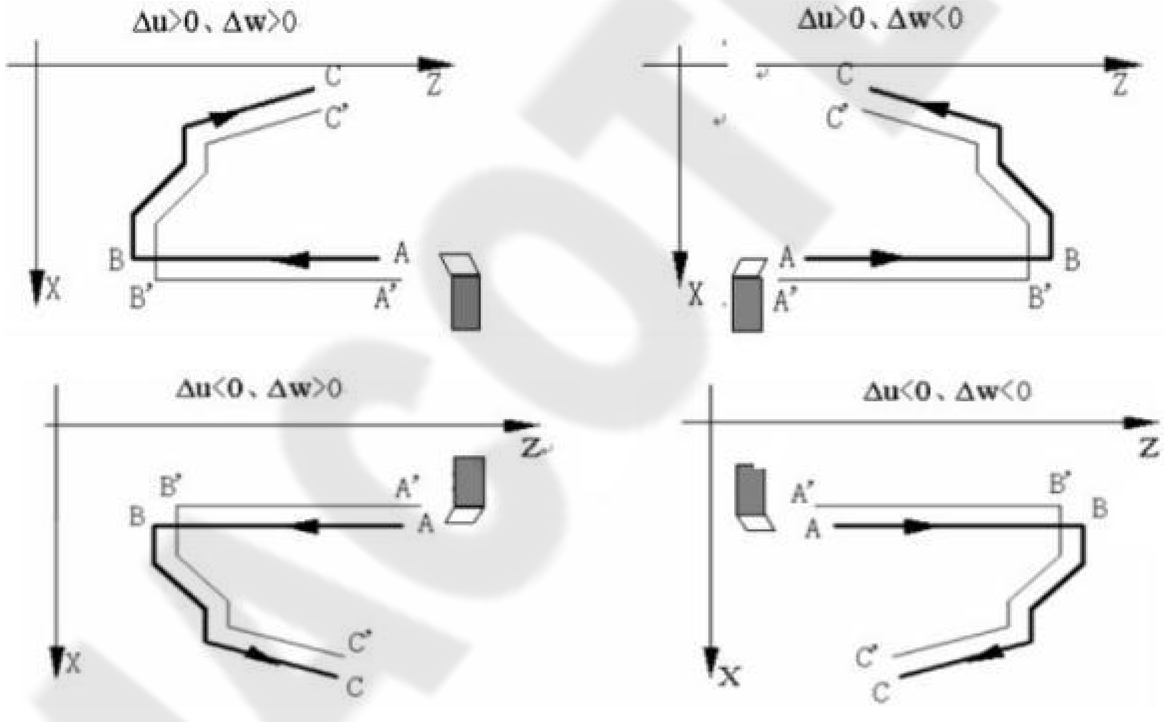


Figure 3-29

Example: Figure 3-30

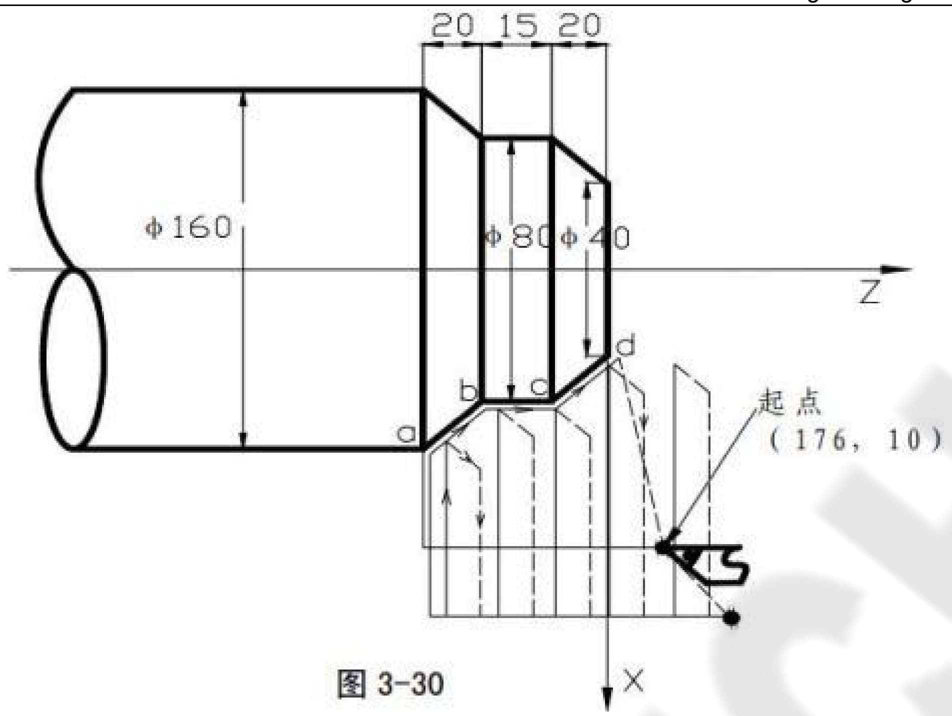


Figure 3-30

Program: O0005;

G00 X176 Z10 M03 S500	(Change tool 2#, execute tool 2# offset, counterclockwise, speed 500)
G72 W2.0 R0.5 F300;	(Infeed 2mm, retract 0.5mm)
G72 P10 Q20 U0.2 W0.1;	(Roughing a--d, 0.2mm allowance for X, 0.1mm allowance for Z)
N10 G00 Z-55 S800;	(Quick movement)
G01 X160 F120;	(Infeed to point a)
X80 W20;	(Processing a-b)
W15;	(Processing b-c)
N20 X40 W20 ;	(Processing c-d)
G70 P010 Q020 M30;	(Finishing a-d)

Finishing route program segment (N10 … N20)

3.16.3 Closed Cutting Cycle G73

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

G73 P(ns) Q(nf) U(Δu) W(Δw); (2)

N_ (ns) . . . . . ;
. . . . ;
. . . . F;
. . . . S;
. . . . ;
N_ (nf). . . . . ; (3)

Code function: G73 code is divided into three parts:

(1) A program segment that gives the amount of tool retraction, number of cuts and cutting speed, spindle speed, and tool function;

(2) The program segment interval with given finishing trajectory and program segment of finishing allowance;

(3) Several consecutive program segments defining the finishing trajectory. When executing G73, these program segments are only used to calculate the rough turning trajectory and are not actually executed.

The system automatically calculates the rough turning offset, the single feed amount of rough turning, and the rough turning trajectory based on data such as the fine turning allowance, tool retraction, and number of cuts. The trajectory of each cutting is the offset of the fine turning trajectory, and the cutting trajectory gradually approaches the fine turning trajectory. The last cutting trajectory is the fine turning trajectory offset by the fine-turning allowance. The start point and end point of G73 are the same. This code is suitable for rough turning of formed blanks. The G73 code is a non-modal code, and the code trajectory is shown in Figure 3-31.

Related definitions:

Fine turning trajectory: The workpiece finishing trajectory given by part (3) of the code (ns~nf program segments). The start point of the finishing trajectory (i.e. the start point of the ns program segment) is the same as the start point and end point 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 finishing trajectory is point A→point B→point C.

Rough turning trajectory: A set of offset trajectories of the fine turning trajectory, and the number of rough turning trajectories is the same as the number of cutting times. After the coordinate offset, the A, B, and C points of the fine turning trajectory correspond to the An, Bn, and Cn points of the rough turning trajectory respectively (n is the number of cutting times, the first cutting is represented by points A1, B1, and C1, and the last cutting is represented by points Ad, Bd, and Cd). The coordinate offset of the first cutting relative to the fine turning trajectory is (Δ i×2+ Δu, Δw+ Δk) (expressed by diameter programming), the coordinate offset of the last cutting relative to the fine turning trajectory is (Δu, Δw), and the coordinate offset of each cutting relative to the previous cutting trajectory is: $\left(-\frac{\Delta i \text{根} 2}{d \quad 1}, -\frac{\Delta k}{d-1}\right)$

Δi: X-axis rough turning retraction, value range -99999.999 ~ 99999.999 (unit: mm, radius value), Δi is equal to the X-axis coordinate offset (radius value) of point A1 relative to point Ad, the total cutting amount (radius value) of the X-axis during rough turning is equal to |Δi|, and the cutting direction of the X-axis is opposite to the sign of Δi: Δi >0, cutting in the negative direction of the X-axis during rough turning. The Δi specified value is retained after execution, and the data is converted into the corresponding value and saved in parameter N0189. When U (Δi) is not entered, the value of parameter N0189 is used as the X-axis rough turning retraction.

Δk: Z-axis rough turning retraction, value range is -99999.999 ~ 99999.999 (unit: mm), Δk is equal to the Z-axis coordinate offset of point A1 relative to point Ad, the total cutting amount of the Z-axis during rough turning is equal to |Δk|, and the cutting direction of the Z-axis is opposite to the sign of Δk: Δk> 0, cutting in the negative direction of the Z-axis during rough turning. The Δk specified value is retained after execution, and the data is converted to the corresponding value and saved in parameter N0190. When W(Δk) is not entered, the value of parameter N0190 is used as the Z-axis rough turning retraction.

d: The number of cutting times, the value range is 1 ~ 9999 (unit: times), and R5 means that the closed cutting cycle completes in 5 cuttings. The R(d) specified value is retained after execution, and the value of data parameter NO.055 is changed to d (unit: times). When R(d) is not entered, the value of parameter N0191 is used as the number of cuts. If the number of cuts is 1, the system will complete the closed cutting cycle with 2 cuttings.

ns: The segment number of the first segment of the fine turning trajectory.

nf: The segment number of the last segment of the fine turning trajectory.

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

Δw: Z-axis finishing allowance, the value range -99999.999 ~ 99999.999 (unit: mm), the Z-axis coordinate offset of the last rough turning trajectory relative to the fine turning trajectory, that is: the difference between the Z-axis absolute coordinates of point A1 and point A. Δw > 0, the last rough turning trajectory is offset in the positive direction of the Z-axis relative to the fine turning trajectory. When W (Δw) is not entered, the system processes it as Δw=0, that is: the rough turning cycle Z-axis does not leave the finishing allowance.

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

M, S, T, F: Code words can be specified in the first G73 code or the second G73 code, or in the ns~nf program (except T instruction). In the G73 cycle, the M, S, and F functions of the program segment numbers between ns and nf are invalid and are only valid in the program segment with the G70 fine turning cycle.

Code execution process: See Figure 3-31.

① A → A1: Quick movement;

② The first rough turning, A1 → B1 → C1:

A1 → B1: When the ns program segment is G0, the quick movement speed is used; when the ns program segment is G1, the cutting feed speed specified by G73 is used;

B1 → C1: Cutting feed.

③ C1 → A2: Quick movement;

④ The second rough turning, A2 → B2 → C2:

A2 → B2: When the ns program segment is G0, the quick movement speed is used; when the ns program segment is G1, the cutting feed speed specified by G73 is used;

B2 → C2: Cutting feed.

⑤ C2 → A3: Quick movement;

…………

The nth rough turning, An → Bn → Cn:

An → Bn: When the ns program segment is G0, the quick movement speed is used; when the ns program segment is G1, the cutting feed speed specified by G73 is used;

Bn → Cn: Cutting feed.

Cn → An+1: Quick movement;

…………

The last rough turning, Ad → Bd → Cd:

Ad → Bd: When the ns program segment is G0, the quick movement speed is used; when the ns program segment is G1, the cutting feed speed specified by G73 is used;

Bd → Cd: Cutting feed.

Cd → A: Quick movement to the starting point;

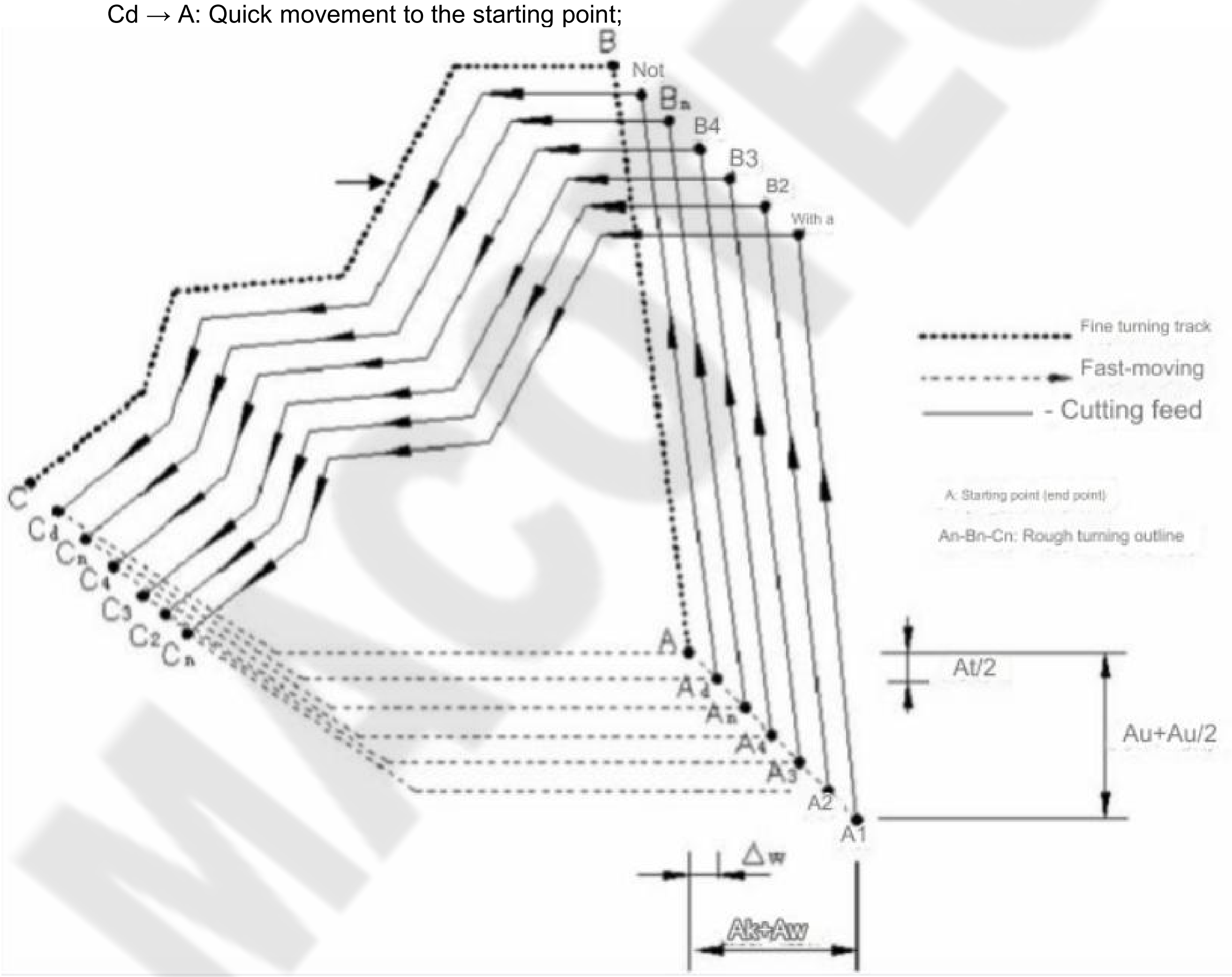


Figure 3-31 G73 code running trajectory

Code description:

● The ns~nf program segment must be written immediately after the G73 program segment. If the ns~nf program segment is written before the G73 program segment, the system can automatically search for the ns~nf program segment and execute it. After execution, the next program of the nf program segment is executed in sequence, which will cause the ns~nf program segment to be repeatedly executed.

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

● The ns program segment can only be G00 and G01 codes.

● In the ns~nf program segment, there can only be the following G functions: G00, G01, G02, G03, G04, G05, G6.2, G6.3, G7.2, G7.3, G96, G97, G98, G99, G40, G41, G42 codes; there cannot be the following M function: subroutine call codes (such as M98/M99).

● The G96, G97, G98, G99, G40, G41, and G42 codes are invalid when executing the G73 cycle, but are valid when executing the G70 finishing cycle.

● During the execution of the G73 code, automatic operation can be stopped and moved manually, but to execute the G73 cycle again, you must return to the position before manual movement. If you continue to execute without returning, the subsequent running trajectory will be misplaced.

● Execute feed hold and single program segment operations, and the program will pause after running to the end point of the current trajectory.

● Δi and Δu are specified by the same address U, and Δk and Δw are specified by the same address W. The distinction is based on whether the program segment specifies P and Q code words.

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

● Macro programs can be written in G73

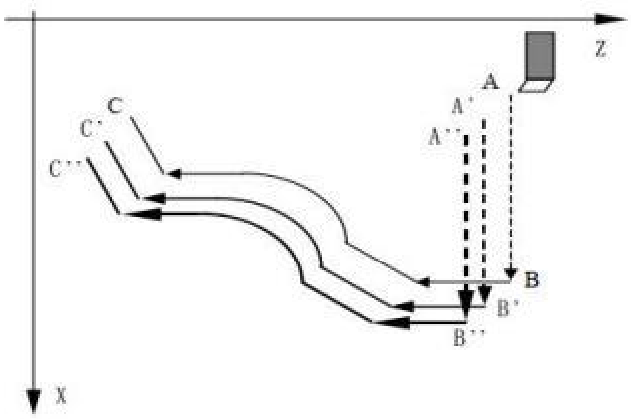
● When compound cycle codes need to be used multiple times in the same program, ns~nf are not allowed to have the same program segment number.

● The retraction point should be as high or low as possible to avoid retraction hitting the workpiece.

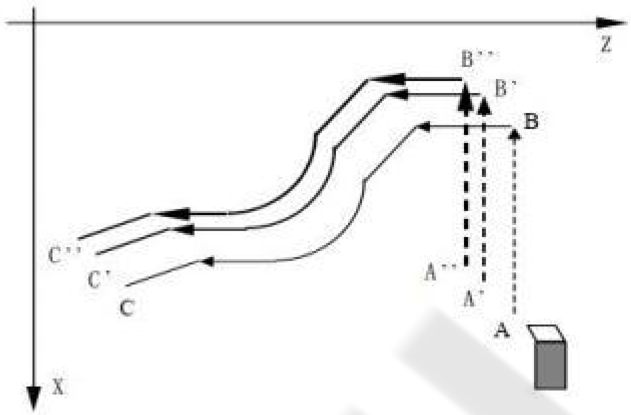
Coordinate offset direction when leaving fine turning allowance:

Δi and Δk reflect the coordinate offset and cutting direction during rough turning, and Δu and Δw reflect the coordinate offset and cutting direction during fine turning. Δi, Δk, Δu, and Δw can have multiple combinations. In general, the signs of Δi and Δu are consistent, and the signs of Δk and Δw are consistent. There are four commonly used combinations, as shown in Figure 3-32. In the figure: A is the starting point, B→C is the workpiece contour, B’→C’ is the rough turning contour, and B”→C” is the fine turning trajectory.

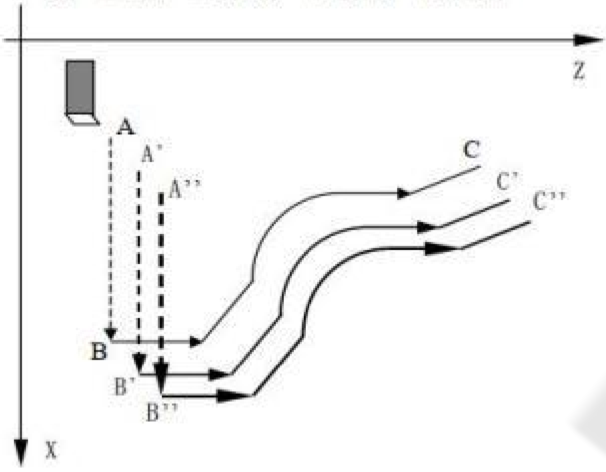
1) $\Delta i<0$ $\Delta k>0$, $\Delta u<0$ $\Delta w>0$;



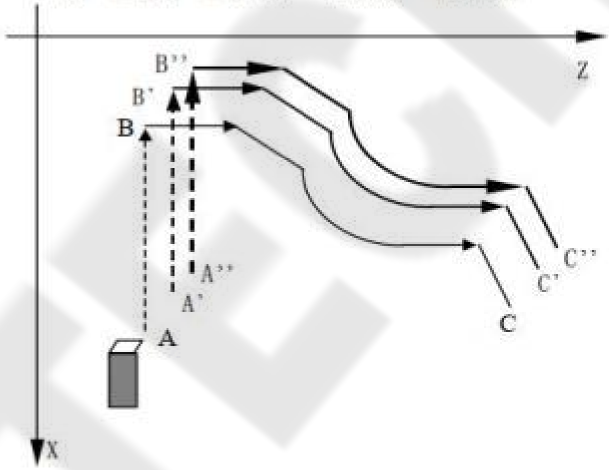
2) $\Delta i>0$ $\Delta k>0$, $\Delta u>0$ $\Delta w>0$;



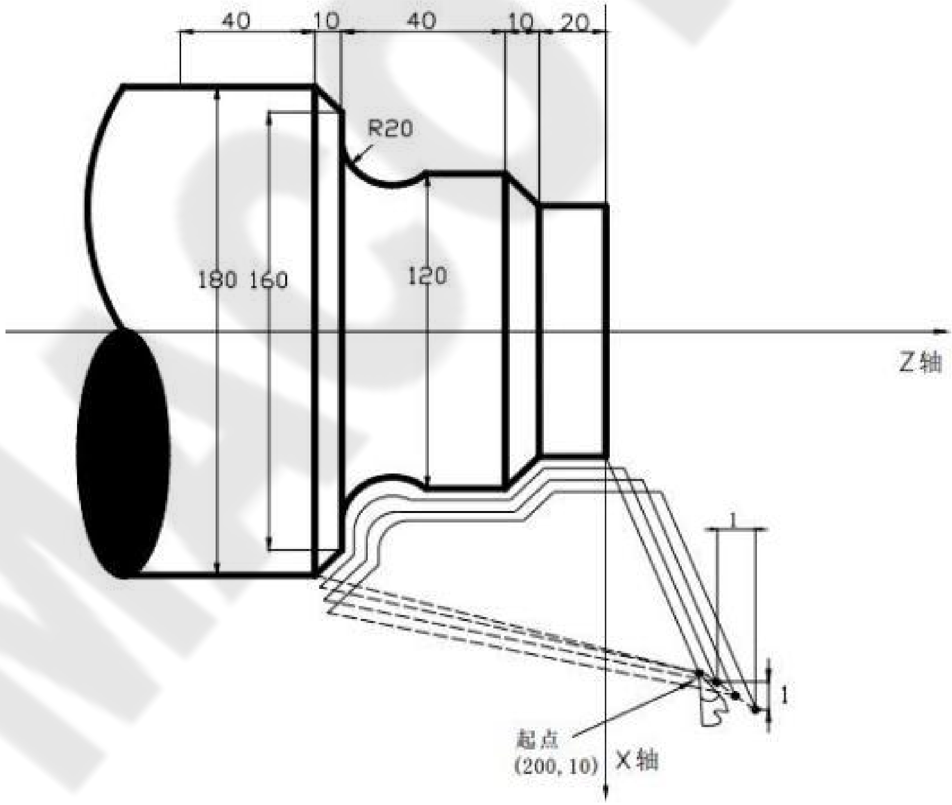
3) $\Delta i<0$ $\Delta k<0$, $\Delta u<0$ $\Delta w<0$;



4) $\Delta i>0$ $\Delta k<0$, $\Delta u>0$ $\Delta w<0$;



Example: Figure 3-33



Program:

```
00006;
G99 G00 X200 Z10 M03 S500;        (Specify feed per revolution, position starting point, start 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 turning, 0.5mm for X-axis, 0.3mm for Z-axis finishing turning allowance)
N14 G00 X80 Z0 ;
G01 W-20 F0.15 S600;
X120 W-10 ;                       Finishing shape program segment
W-20 ;
G02 X160 W-20 R20 ;
G02 X160 W-20 R20 ;
G70 P14 Q19 M30; (finishing)
```

3.16.4 Finishing Cycle G70

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

Code function: The tool performs finishing from the starting position along the workpiece finishing trajectory given by the ns~nf program segment. After roughing with G71, G72 or G73, use the G70 code for finishing, and complete the cutting of the finishing allowance in a single pass. At the end of the G70 cycle, the tool returns to the start point and executes the next program segment after the G70 program segment.

Among them: ns: The segment number of the first segment of the fine turning trajectory;

nf: The segment number of the last segment of the fine turning trajectory;

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

```
. . . . . . . . .
G71/G72/G73 ······;
N_ (ns) . . . . . .
. . . . . . . . .
      • F
      • S                         Finishing route program segment group
      •
      •
N_ (nf)······
     . . .
G70 P(ns) Q(nf);
     . . .
```

Code description:

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

● When executing the G70 finishing cycle, the F, S, and T codes in the ns~nf program segment are valid.

● G96, G97, G98, G99, G40, G41, and G42 codes are valid when executing the G70 finishing cycle.

● During the execution of the G70 code, automatic operation can be stopped and moved manually, but to execute the G70 cycle again, you must return to the position before manual movement. If you continue to execute without returning, the subsequent running trajectory will be misplaced.

● Execute single program segment operations, and the program will pause after running to the end point of the current trajectory.

● G70 code cannot be executed in input mode, or an alarm will be generated.

● When compound cycle codes need to be used multiple times in the same program, ns~nf are not allowed to have the same program segment number.

● The retraction point should be as high or low as possible to avoid retraction hitting the workpiece.

3.16.5 Axial Grooving Multiple Cycles G74

Code format: G74 R(e);

G74 X(U)_ Z(W)_ P(Δ i) Q(Δk) R(Δd) F_;

Code meaning: Radial (X-axis) feed cycle compound axial intermittent cutting cycle: feed, retract, and feed again from the start point axially (Z-axis) until the cutting reaches the same position as the Z-axis coordinate of the cutting end point, and then retract radially and retract axially to the same position as the Z-axis coordinate of the start point, completing an axial cutting cycle; after radial feed again, carry out the next axial cutting cycle; after cutting to the cutting end point, return to the start point (the start point and end point of G74 are the same), and complete the axial grooving compound cycle. The radial feed and axial feed directions of G74 are determined by the relative positions of the cutting end point X(U), Z(W) and the start point. This code is used to process annular grooves or center deep holes on the end face of the workpiece. Axial intermittent cutting plays a role in chip breaking and timely chip removal.

Related definitions:

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

End point of axial feed: The end point of axial feed in each axial cutting cycle, expressed as Bn (n=1,2,3……), the Z-axis coordinate of Bn is the same as the cutting end point, the X-axis coordinate of Bn is the same as An, and the end point of the last axial feed (expressed as Bf) is the same as the cutting end point;

End point of radial retraction: The end point of radial retraction (retraction amount is Δd) after each axial cutting cycle reaches the end point of axial feed, expressed as Cn (n=1,2,3……), the Z-axis coordinate of Cn is the same as the cutting end point, and the difference between Cn and An X-axis coordinates is Δd;

End point of axial cutting cycle: The end point of axial retraction from the radial retraction end point, expressed as Dn (n=1,2,3……), the Z-axis coordinate of Dn is the same as the starting point, and the X-axis coordinate of Dn is the same as Cn (the difference with An X-axis coordinate is Δd);

End point of cutting: X(U)_ Z(W)_ specified position, the end point of the last axial feed is Bf.

R(e): Axial retraction amount after each axial (Z-axis) feed, the value range 0~99.999 (IS-B) / 0~99.9999 (IS-C) (unit: mm), unsigned. The specified value remains valid after R(e) is executed, and the data is converted into the corresponding value and saved in parameter N0192. When R(e) is not entered, the value of parameter N0192 is used as the axial retraction amount.

X: X-axis absolute coordinate value of the cutting end point Bf.

U: The difference between the X-axis absolute coordinates of the cutting end point Bf and the start point A.

Z: The absolute coordinate value of the Z-axis of the cutting end point Bf.

W: The difference between the Z-axis absolute coordinates of the cutting end point Bf and the start point A.

P(Δi): The radial (X-axis) cutting amount of a single axial cutting cycle, the value range 0 < Δ i ≤ 9999999 (IS_B) / 0 < Δ i ≤ 99999999 (IS_C) (unit: least input increment, diameter value, unsigned).

Q(Δk): The feed amount of Z-axis intermittent feed during axial (Z-axis) cutting, the value range 0 < Δk ≤ 9999999 (IS_B) / 0 < Δk ≤ 99999999 (IS_C) (unit: least input increment, unsigned).

R(Δd): The radial (X-axis) retraction amount after cutting to the axial cutting end point, the value range 0 ~ 99999999 × least input increment (unit: mm/inch, diameter value, unsigned). When R(Δd) is omitted, the system defaults to 0 radial (X-axis) retraction amount after the axial cutting end point. When the X(U) and P(Δ i) code words are omitted, the tool is retracted in the positive direction by default.

Code execution process: As shown in Figure 3-34.

①The axial (Z-axis) cutting feed Δk from the axial cutting cycle start point An. When the Z-axis coordinate of the cutting end point is less than the Z-axis coordinate of the start point, feed to the negative direction of the Z-axis, and otherwise, feed to the positive direction of the Z axis;

② Quickly retract the tool in the axial (Z-axis) by e, and the retraction direction is opposite to the feeding direction of ①;

③ If the Z-axis cuts and feeds again (Δk+e), and the feed end point is still between the axial cutting cycle start point An and the axial feed end point Bn, the Z-axis cuts and feeds again (Δk+e), and then executes ②; if the feed end point reaches point Bn or is not between An and Bn after the Z-axis cuts and feeds again (Δk+e), the Z-axis cuts and feeds to point Bn, and then executes ④;

④ Quickly retract the tool in the radial (X-axis) by Δd to point Cn. When the X-axis coordinate of point Bf (cutting end point) is less than the X-axis coordinate of point A (starting point), retract the tool to the positive direction of the X-axis, and otherwise, retract to the negative direction of the X-axis;

⑤ Quickly retract to Dn point in axial (Z-axis) direction, 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 Quickly feed in radial (X-axis) direction, the feed direction is opposite to the retract direction of ④. If after X-axis feed (ɑʌʌ d+ɑʌʌ i), the feed end point is still between point A and point Af (the start point of the last axial cutting cycle), X-axis quickly feeds (ɑʌʌ d+ɑʌʌ i), that is: Dn→An+1, and then execute ① (start the next axial cutting cycle); if after X-axis feed (ɑʌʌ d+ɑʌʌ i), the feed end point reaches point Af or is not between points Dn and Af, X-axis quickly moves to point Af, then executes ①, and starts the last axial cutting cycle;

⑦ X-axis quickly returns to start point A, and G74 code execution ends.

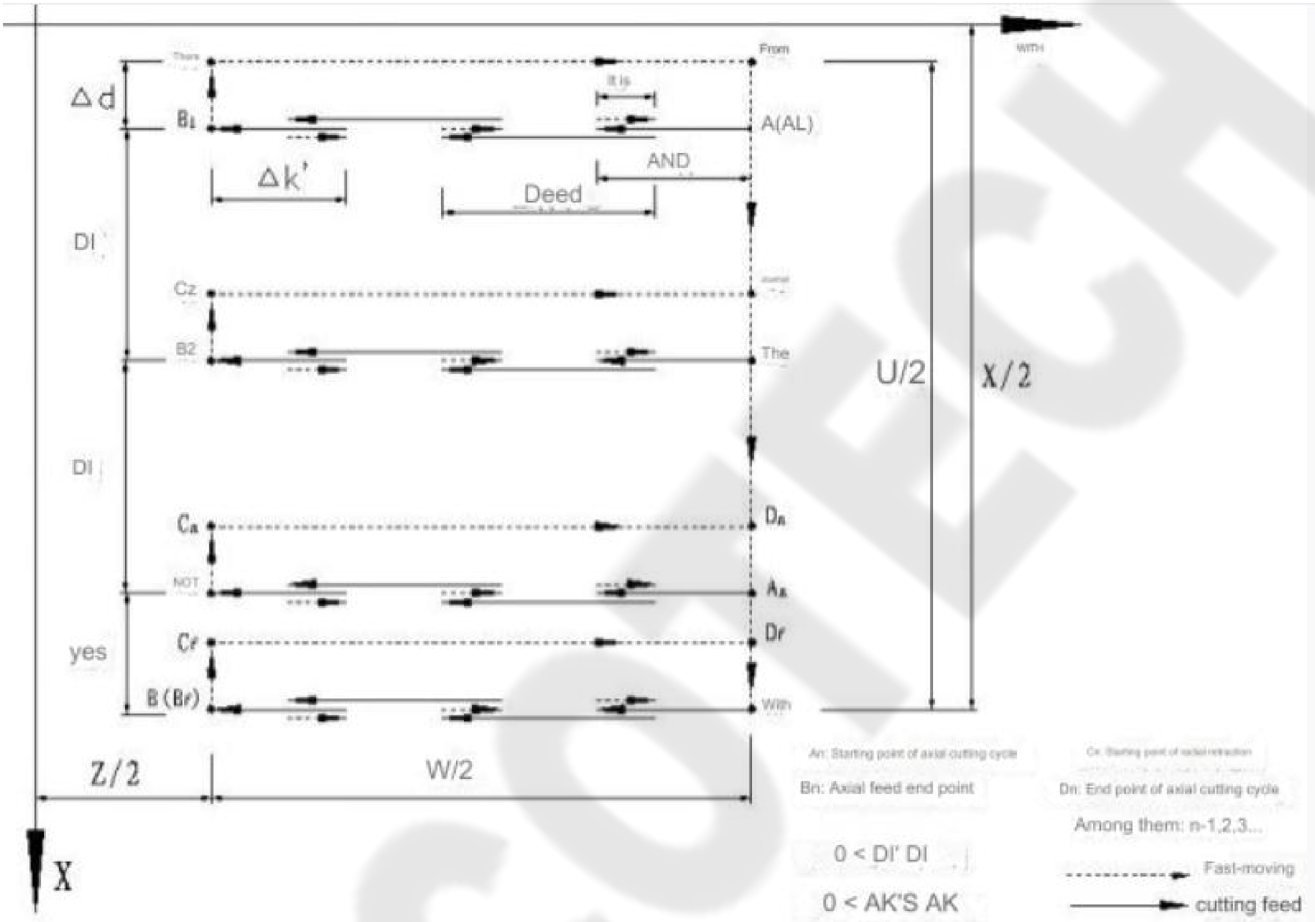


Figure 3-34 G74 trajectory diagram

Code description:

● The cycle action is performed by the G74 program segment containing Z(W) and Q(Δk). If only the "G74 R(e);" program segment is executed, the cycle action will not be performed;

● Δd and e are both specified by the same address R, and the difference is based on whether there are Z(W) and Q(Δk) code words in the program segment;

● During the execution of the G74 code, automatic operation can be stopped and moved manually, but to execute the G74 cycle again, you must return to the position before manual movement. If you continue to execute without returning, the subsequent running trajectory will be misplaced.

● Execute single program segment operations, and the program will pause after running to the end point of the current trajectory.

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

Example: Figure 3-35

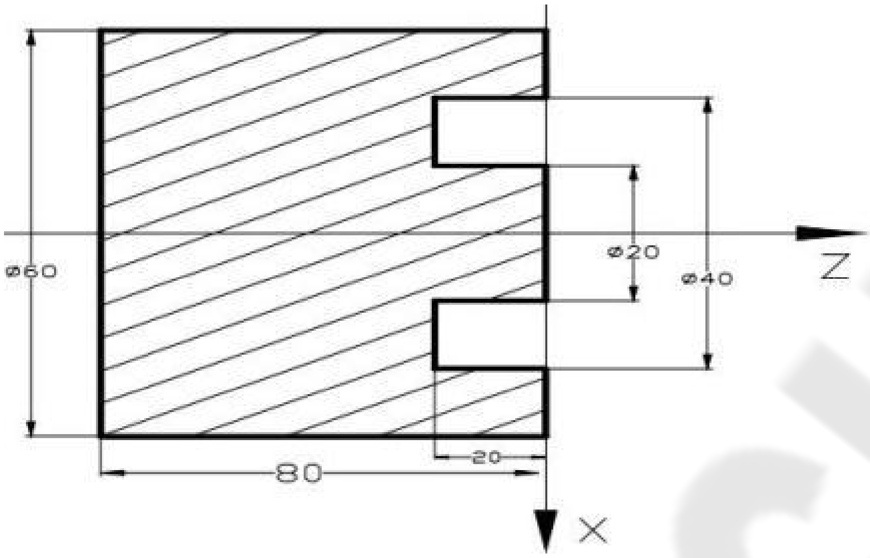


Figure 3-35

Program (Assuming the grooving tool width is 4mm and the system's minimum increment is 0.001mm):

O0007;

G0 X36 Z5 M3 S500;	(Start the spindle, position it to the machining start point, and add the tool width to the X direction)
G74 R0.5;	(Machining cycle)
G74 X20 Z-20 P3000 Q5000 F50;	(Z-axis feeds 5mm each time, retracts 0.5mm, and quickly returns to the start point (Z5) after feeding to the end point (Z-20), and the X-axis feeds 3mm. Repeat the above steps to continue running)
M30;	(Program ends)

3.16.6 Radial Grooving Multiple Cycles G75

Code format: G75 R(e);

G75 X(U)_ Z(W)_ P(Δ i) Q(Δk) R(Δd) F_;

Code meaning: Axial (Z-axis) feed cycle compound radial intermittent cutting cycle: feed, retract, and feed again from the start point radially (X-axis) until the cutting reaches the same position as the X-axis coordinate of the cutting end point, and then axially retract and radially retract to the same position as the X-axis coordinate of the start point, completing a radial cutting cycle; after axially feeding again, carry out the next radial cutting cycle; after cutting to the cutting end point, return to the start point (the start point and end point of G75 are the same), and complete the radial grooving compound. The axial feed and radial feed directions of G75 are determined by the relative position of the cutting end point X(U)Z(W) and the start point. This code is used to process radial annular grooves or cylindrical surfaces. Radial intermittent cutting plays a role in chip breaking and timely chip removal.

Related definitions:

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

End point of radial feed: The end point of radial feed in each radial cutting cycle, expressed as Bn (n=1,2,3……), the X-axis coordinate of Bn is the same as the cutting end point, the Z-axis coordinate of Bn is the same as An, and the end point of the last radial feed (expressed as Bf) is the same as the cutting end point;

End point of axial retraction: The end point of axial retraction (retraction amount is Δd) after each radial cutting cycle reaches the end point of radial feed, expressed as Cn (n=1,2,3……), the X-axis coordinate of Cn is the same as the cutting end point, and the difference between Cn and An Z-axis coordinates is Δd;

End point of radial cutting cycle: The end point of radial retraction from the axial retraction end point, expressed as Dn (n=1,2,3……), the X-axis coordinate of Dn is the same as the start point, and the Z-axis coordinate of Dn is the same as Cn (the difference with An Z-axis coordinate is Δd);

End point of cutting: The position specified by X(U) Z(W), the end point of the last radial feed is Bf.

R(e): Radial retraction amount after each radial (X-axis) feed, the value range 0 ~ 99.999 (IS-B) / 0 ~ 99.9999 (IS-C) (unit: mm, radius value), unsigned. The specified value remains valid after R(e) is executed, and the data is converted into the corresponding value and saved in parameter N0192. When R(e) is not entered, the value of parameter N0192 is used as the radial retraction amount.

X: X-axis absolute coordinate value of the cutting end point Bf.

U: The difference between the X-axis absolute coordinates of the cutting end point Bf and the start point A.

Z: The absolute coordinate value of the Z-axis of the cutting end point Bf.

W: The difference between the Z-axis absolute coordinates of the cutting end point Bf and the start point A.

P(Δi): The amount of intermittent X-axis feeding during radial (X-axis) feeding, the value range is 0 < Δi ≤ 9999999 (IS_B) / 0 < Δk ≤ 99999999 (IS_C) (unit: least input increment, unsigned).

Q(Δk): The amount of axial (Z-axis) feeding in a single radial cutting cycle, the value range is 0 < Δk ≤ 9999999 (IS_B) / 0 < Δk ≤ 99999999 (IS_C) (unit: least input increment, unsigned).

R(Δd): The amount of tool retraction in the axial direction (Z-axis) after cutting to the radial cutting end point, the value range is 0 ~ 99999999 × least input increment (unit: mm/inch, unsigned).

When R(Δd) is omitted, the system defaults the amount of tool retraction in the axial direction (Z-axis) to 0 after the radial cutting end point.

When Z(W) and Q(Δk) are omitted, the tool retraction is in the positive direction by default.

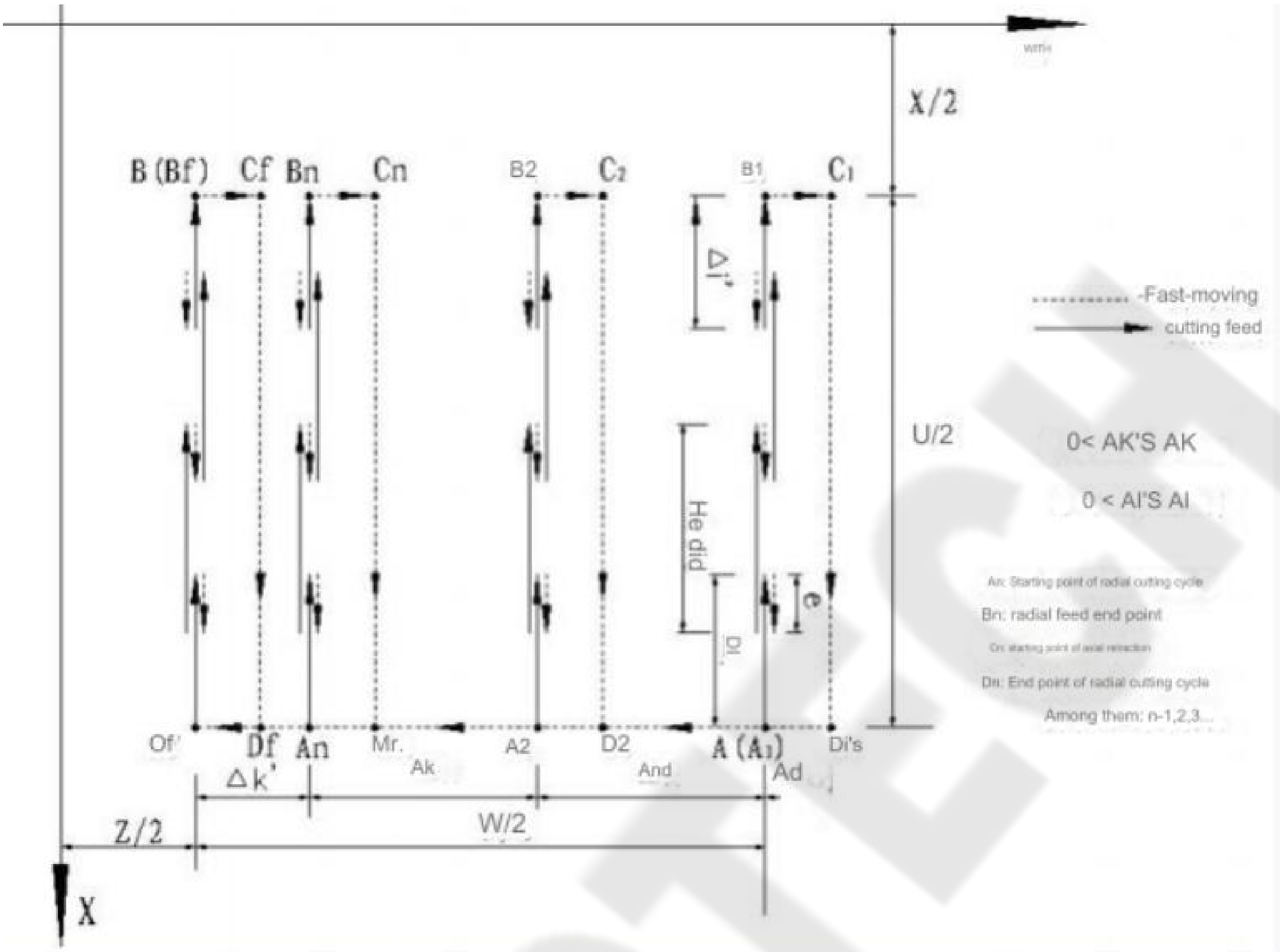


Figure 3-36 G75 trajectory diagram

Code execution process: Figure 3-36

① Radial (X-axis) cutting feed Δi from the radial cutting cycle start point An. When the X-axis coordinate of the cutting end point is less than the X-axis coordinate of the start point, feed to the negative direction of the X-axis, and otherwise, feed to the positive direction of the X-axis;

② Quickly retract the tool in the radial (X-axis) by e, and the retraction direction is opposite to the feeding direction of ①;

③ If the X-axis cuts and feeds again (Δ i+e), the feed end point is still between the radial cutting cycle start point An and the radial feed end point Bn, the X-axis cuts and feeds again (Δ i+e), and then executes ②; if the feed end point reaches point Bn or is not between An and Bn after the X-axis cuts and feeds again (Δ i+e), the X-axis cuts and feeds to point Bn, and then executes ④;

④ Quickly retract the tool in the axial (Z-axis) by Δd to point Cn, when the Z-axis coordinate of point Bf (cutting end point) is less than the Z-axis coordinate of point A (starting point), retract the tool in the positive direction of the Z-axis, and otherwise, retract in the negative direction of the Z axis;

⑤ Quickly retract to Dn point in radial (X-axis) direction, 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 Quickly feed in axial (Z-axis) direction, and the feed direction is opposite to the retract direction of ④. If after Z-axis feed (Δd+Δk), the feed end point is still between point A and point Af (the start point of the last radial cutting cycle), Z-axis quickly feeds (Δd+Δk), that is: Dn→An+1, and then execute ① (start the next radial cutting cycle); if after Z-axis feed (Δd+ Δk), the feed end point reaches point Af or is not between points Dn and Af, Z-axis quickly moves to point Af, then executes ①, and starts the last radial cutting cycle;

⑦ Z-axis quickly returns to start point A, and G75 code execution ends.

Code description:

● The cycle action is performed by the G75 program segment containing X(U) and P(Δi). If only the "G75 R(e);" program segment is executed, the cycle action will not be performed;

● Δd and e are both specified by the same address R, and the difference is based on whether there are X(U) and P(Δi) code words in the program segment;

● During the execution of the G75 code, automatic operation can be stopped and moved manually, but to execute the G75 cycle again, you must return to the position before manual movement. If you continue to execute without returning, the subsequent running trajectory will be misplaced;

● Execute single program segment operations, and the program will pause after running to the end point of the current trajectory.

● When performing a grooving cycle, the R(Δd) code word must be omitted because there is no retraction distance when cutting to the radial cutting end point. Example: Figure 3-37

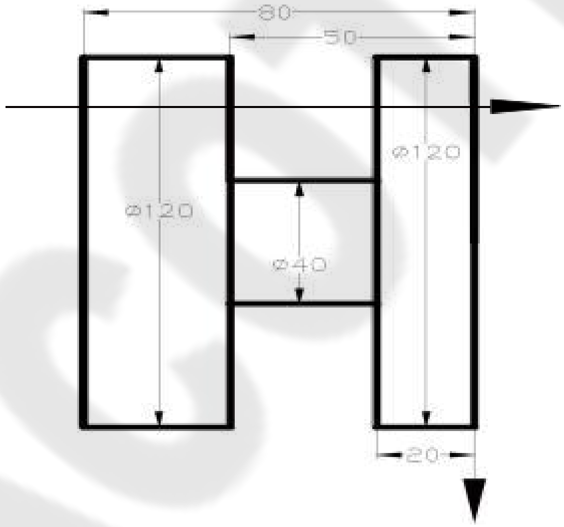


Figure 3-37 G75 code cutting diagram

Program (Assuming the width of the grooving tool is 4mm, and the least increment of the system is 0.001mm):

O0008 ;

G00 X150 Z50 M3 S500;	(Start the spindle and set the speed to 500)
G0 X125 Z-24;	(Position it to the processing start point and add the tool width in the Z direction)
G75 R0.5 F150;	(Machining cycle)

G75 X40 Z-50 P6000 Q3000;	(The X-axis feeds 6mm each time, retracts 0.5mm, feeds to the end point (X40), quickly returns to the start point (X125), and the Z-axis feeds 3mm. Repeat the above steps to continue running)
G0 X150 Z50;	(Return to the processing start point)
M30;	(Program end)

3.16.7 G83 and G83.1 are end drilling cycles, G87 and G87.1 are side drilling cycles

(1) G83 Z(W)_ R_ P_ Q_ F_ K_

G87 X(U)_ R_ P_ Q_ F_ K_

R_: Indicates the absolute coordinate value of point R;

P_: Dwell time at the bottom of the hole, modal data, unit: ms, no dwell when not specified or 0;

Q_: Cutting depth of each cutting feed;

F_: Cutting feed rate;

K_: Number of repetitions;

(2) For high-speed drilling, system parameter 206 sets the retraction distance each time

G83.1 Z(W)_ R_ P_ Q_ F_ K_

G87.1 X(U)_ R_ P_ Q_ F_ K_

3.17 Thread Cutting Code

It has a variety of thread cutting functions, and can process single-start head, multi-start head, variable lead threads and tapping cycles (F unit is inch/lead when input in the inch system; F unit is mm/lead when input in the metric system. I specifies the number of threads per inch and has nothing to do with the metric system). The thread back-off length and angle are variable. Multiple-cycle thread cutting can be cut on one side to protect the tool and improve the surface finish. Thread functions include continuous thread cutting code G32, variable pitch thread cutting code G34, tapping cycle cutting code G33, thread cycle cutting code G92, and thread multiple cycle cutting code G76.

The machine tool that uses the thread cutting function must be equipped with a spindle encoder. The spindle encoder line number is set by parameter N0110, and the transmission ratio between the spindle and the encoder is set by parameters N0113 and N0114. When cutting threads, the system will move the X-axis or Z-axis and start thread processing only after receiving a signal from the spindle encoder. Therefore, as long as the spindle speed is not changed, the same thread can be processed by multiple rough turning and fine turning.

The various thread cutting functions can be used to process threads without undercut grooves. However, since the X-axis and Z-axis have acceleration and deceleration processes at the beginning and end of thread cutting, and the pitch error is large at this time, so it is still necessary to leave the thread lead-in length and the distance of undercut at the actual thread start point and end.

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 speed ratio. Spindle ratio control is effective during thread cutting. When the spindle speed changes, pitch error will occur due to the acceleration and deceleration of the X-axis and Z-axis. Therefore, do not adjust the spindle speed during thread cutting, let alone stop the spindle, or it will cause damage to the tool and workpiece.

3.17.1 Equal Pitch Thread Cutting Code G32

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

Code function: The tool's motion trajectory is a straight line from the start point to the end point; the coordinate axis with the larger displacement (X-axis according to the radius value) from the start point to the end point is called the major axis, and the other coordinate axis is called the minor axis. During the motion process, the major axis moves one lead for each rotation of the spindle, and the minor axis and the major axis are inear interpolation. When the tool cuts the workpiece, a spiral groove with equal pitch is formed on the surface of the workpiece to achieve the processing of equal pitch threads. The F and I code words are used to give the pitch of the thread. Executing the G32 code can process straight threads, tapered threads, end threads, and continuous multi-segment threads with equal pitch.

Code description: G32 is a modal G code;

The lead of the thread refers to the displacement of the major axis when the spindle rotates one circle (the displacement of the X-axis is based on the radius value);

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

When the Z coordinate values of the start point and the end point are the same (do not enter Z or W), end face thread cutting is performed;

When the X and Z coordinate values of the start point and the end point are different, tapered thread cutting is performed.

Related definitions:

F: Specify the thread lead, which is the displacement of the major axis when the spindle rotates one circle. The value range is 0 < F ≤ 500mm (0 ~ 50 inch for inch input). The F specified value remains valid after execution until the F code word with a given thread pitch is executed again.

I: Specify the number of threads per inch, which is the number of threads on the length of 1 inch (25.4mm) in the major axis direction. It can also be understood as the number of revolutions of the spindle when the major axis moves 1 inch (25.4mm). The value range is 0.06 ~ 25400 threads/inch. I specified value remains valid after execution until I code word of the given thread pitch is executed again. Both metric input and inch input represent the number of threads per inch.

J: The movement amount in the minor axis direction when the thread backs off (back-off amount), with positive and negative directions; if the minor axis is the X-axis, the value is specified by the radius; the J value is a non-modal parameter.

K: The length of the thread in the major axis direction when the thread back-off. If the major axis is the X-axis, the value is specified by the radius; without direction; the K value is a non-modal parameter.

Q: Starting angle, which refers to the offset angle between the one-turn signal of the spindle and the start point of thread cutting. The value range is 0 ~ 360000 (unit: 0.001 degree). The Q value is a non-modal parameter and must be specified each time it is used. If it is not specified, it is considered to be 0 degrees.

Q usage rules:

1. If Q is not specified, the default starting angle is 0 degrees;

2. For continuous thread cutting, except for the first segment Q, the Q specified in the subsequent thread cutting segments is invalid, and even if Q is defined, it is ignored;

3. The total number of multi-start threads formed by the starting angle definition index does not exceed 65535.

4. The unit of Q is 0.001 degrees. If it is offset by 180 degrees from the spindle one-turn signal, Q180000 must be entered in the program. If Q180 or Q180.0 is entered, it is considered to be 0.18 degrees.

The judgment method of the major axis and minor axis: Figure 3-38.

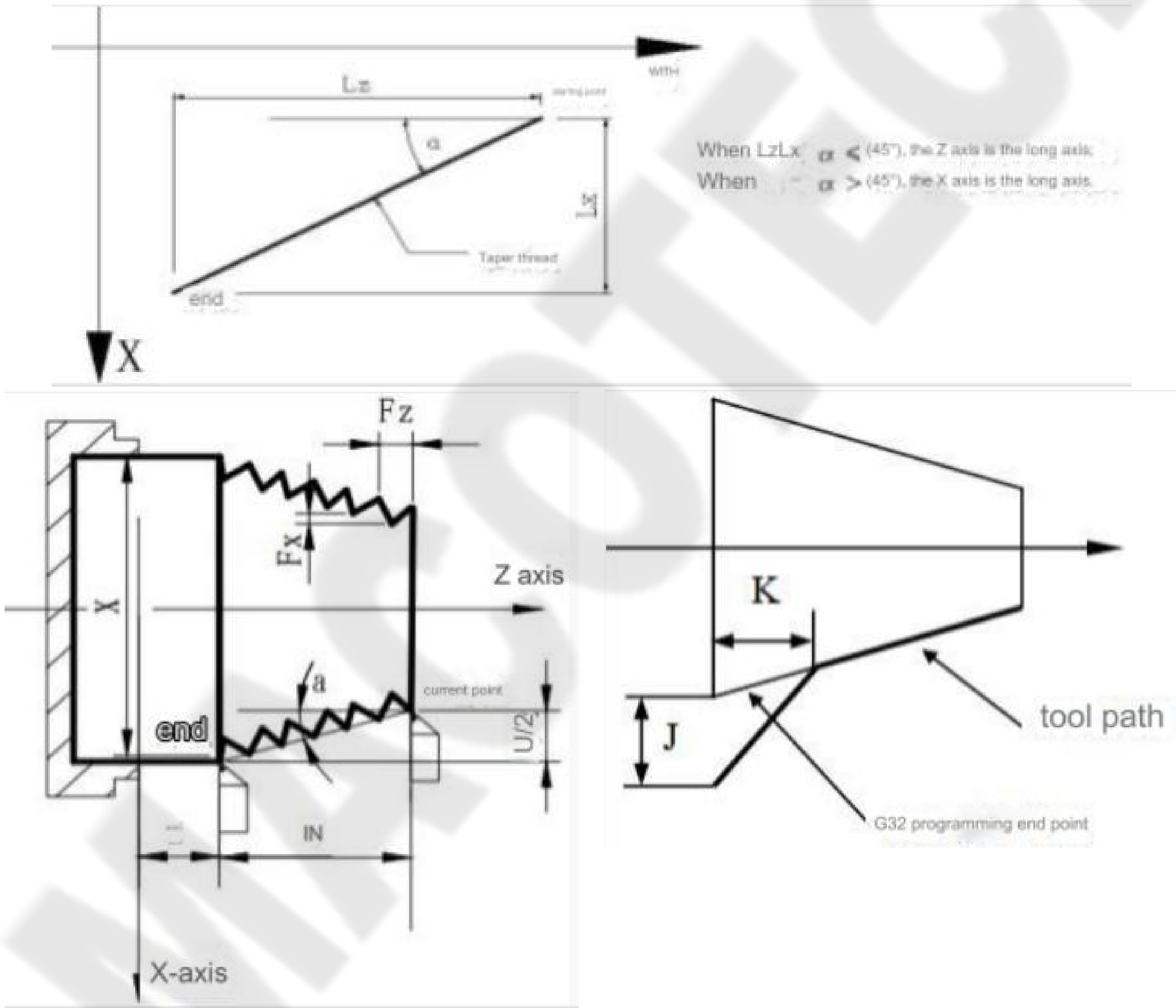


Figure 3-38 G32 trajectory diagram

Notes:

● When J is omitted or J and K are omitted, there is no back-off; when K is omitted, back-off by K=J;

● When J=0 or J=0 and K=0, there is no back-off;

● When J≠0 and K=0, back-off by J=K;

● When J=0 and K≠0, there is no back-off;

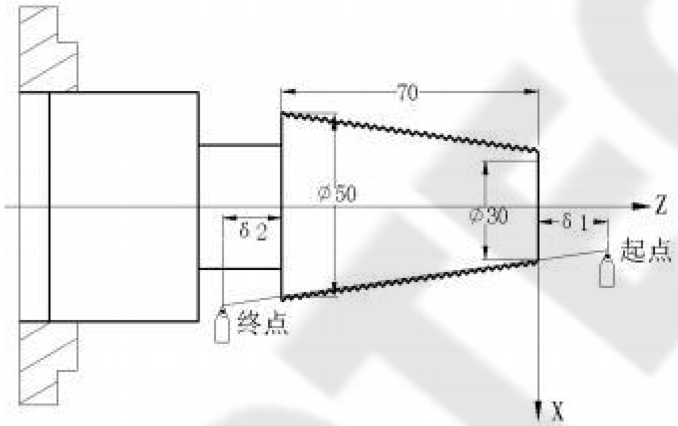
● The current program segment is thread cutting, and the next program segment is also thread cutting. At the beginning of the next program segment cutting, thread processing starts directly without detecting the one-turn signal of the spindle position encoder. This function can realize continuous thread processing.

● After executing the feed hold operation, the system displays "Pause", and thread cutting does not stop until the current program segment is executed; if it is continuous thread processing, it stops after the thread cutting program segment is executed, and the program operation is paused.

● In single-segment operation, it stops after the current program segment is executed. If it is continuous thread processing, it stops after the thread cutting program segment is executed.

● When the system is reset, during an emergency stop, or its drive alarm occurs, the thread cutting decelerates and stops.

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



Program:
O0009;

G00 X28 Z3;	(First cut 1mm)
G32 X51 W-75 F2.0;	(First taper thread cutting)
G00 X55;	(Tool exit)
W75;	(Z-axis returns to start point)
X27;	(Second feed 0.5mm)
G32 X50 W-75 F2.0;	(Second taper thread cutting)
G00 X55;	(Tool exit)
W75;	(Z-axis returns to start point)
M30;	

3.17.2 Rigid Thread Cutting Code G32.1

Code format: G32.1 X(U)__ Z(W)__ C(H)__F(I)__ S__;

Code function: Traditional thread interpolation uses the number of pulses fed back by the position encoder installed on the spindle to calculate the current movement of the feed axis to achieve the thread interpolation method in which the feed axis follows the spindle. Its disadvantage is that the thread lead error at the acceleration/deceleration point is large. In the rigid thread interpolation method, the spindle motor works like a servo motor, and the thread interpolation is performed by the interpolation between the feed axis and the spindle, thereby obtaining a thread with higher precision.

Code description: G32.1: G code for rigid thread interpolation;

C: Starting angle of thread interpolation;

(X, Z): End point coordinates of thread interpolation;

F(I): Lead of thread, F(I) > 0 right-hand thread, F(I) < 0 left-hand thread;

S: Spindle speed;

The schematic diagram of the instruction execution trajectory is as follows:

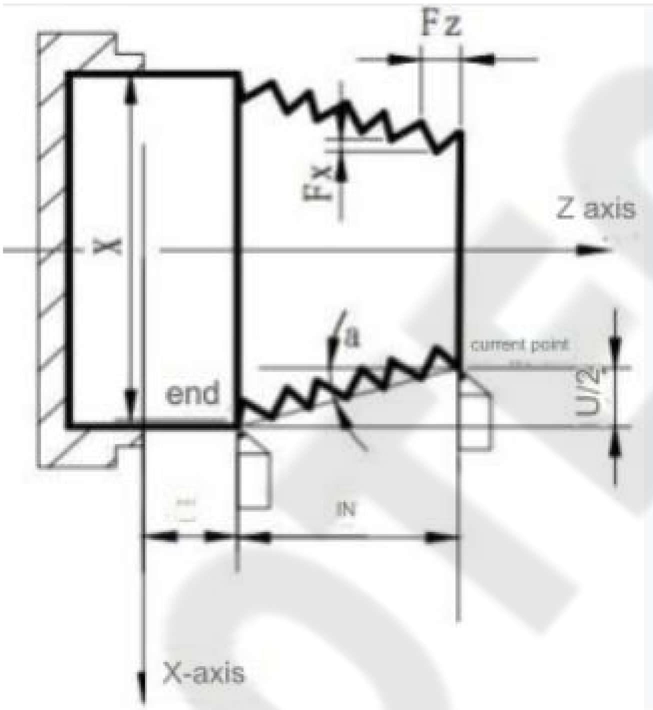


Figure 3-40 Schematic diagram of the G32.1 instruction trajectory

Description:

1) G32.1 is a 01 group G instruction;

2) When there is a C-axis movement instruction in the program segment, position the spindle first to the starting position of the C-axis before turning the thread;

3) When repeated processing is required, the starting positions of the X, Z, and C axes must be the same each time;

4) The G32.1 instruction does not specify thread back-off. At the end point of the thread, the feed axis and the spindle stop moving at the same time;

5) The input range of the address value programmed in the G32.1 instruction is the same as that of the ordinary thread turning instruction (G32);

6) When the G32.1 instruction is executed, the pause signal is temporarily ineffective and the spindle ratio is fixed at 100%.

Notes:

● The spindle must work in position control mode;

● In the case of the multi-spindle control function, G32.1 is only allowed to be used between the first spindle and the feed axis. The method of use and related parameter settings are the same as rigid tapping;

● When executing the G32.1 instruction, the CNC system will not detect whether the current spindle is in position control mode or speed control mode. Therefore, when using this instruction, please set the servo control axis of the first spindle to the Cs axis working mode to ensure safety;

● The absolute coordinate of the C-axis should be set to the cycle mode to avoid coordinate value overflow;

● When the spindle control mode is switched from speed control mode to position mode, please execute the spindle return reference point operation or use the G50 instruction to set the current C-axis starting position;

Example: Assume M14: the spindle is switched to position control mode; M15: the spindle is switched to speed control mode;

The thread is right-handed, the lead is 2mm, the spindle speed is 500 rpm during thread cutting, and the thread cutting length is 20mm. The programming is as follows:

```
O0132 (0132);
G00 X100 Z100;          // Position to a safe position to change the tool
T0101;                  // Change the thread tool (Assuming that 01# is the thread tool)
G00 X25 Z2;             // Position to the start point of the thread (Assuming that it is the last cut to form)
M14;                    // Switch the spindle from speed control to position control (The position is 0
                        degrees after the switch is completed)
G50 C0;                 //Set the origin of the rotary axis (very important as it is related to the starting
                        angle of the following thread turning)
G32.1 Z-20 F2 S500 M08;     //Thread turning, the speed of the spindle and feed axis are both 0
                            when reaching the end point
G00 X30;                //Retract the tool
X24.5 Z2 C0;            //Return to the start point of the thread and prepare for repeated processing
G32.1 Z-20 F2 S500;     //Repeat processing
. . . . ;               //Allow to repeat processing
G00 X100;               //Retract the tool
Z100;                   //Return to the tool change position
M15;                    //Switch the spindle from position control to speed control mode
. . . . ;               //Allow to perform the second process
. . . . ;
M30;                    //Program ends
```

3.17.3 Variable Pitch Thread Cutting Code G34

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

Code function: The tool's motion trajectory is a straight line from the starting position of the X and Z axes to the end position specified by the program segment. The coordinate axis with the larger displacement from the start point to the end point (X-axis according to the radius value) is called the major axis, and the other coordinate axis is called the minor axis. During the movement, the major axis moves one lead for each rotation of the spindle, and the pitch of the spindle moves for each rotation of the spindle is continuously increased or decreased by the specified value, forming a variable pitch spiral groove on the surface of the workpiece to achieve variable pitch thread processing. When cutting, the tool retraction amount can be set. The F and I code words are used to specify the pitch of the thread respectively. Executing the G34 code can process metric or inch variable pitch straight threads, tapered threads and end threads.

Code description: G34 is a modal G code;

The meanings of X/U, Z/W, J, and K are the same as those of G32;
F: Specify the lead, and the value range is 0 ~ 500mm;

I: Specify the number of threads per inch, and the value range is 0.06 ~ 25400 threads per inch;
R: The incremental or decremental value of the pitch per spindle revolution, R=F1-F2, R has a direction;

When F1 > F2, the pitch decreases when R is a negative value;
When F1 < F2, the pitch increases when R is a positive value (as shown in Figure 3-40);

R value range: ±0.001 ~ ±500.000 mm/pitch (metric thread); ±0.060 ~ ±25400 threads/inch (inch thread). When the R value exceeds the above range and the pitch exceeds the allowable value or the pitch becomes negative due to the increase/decrease of R, an alarm is generated.

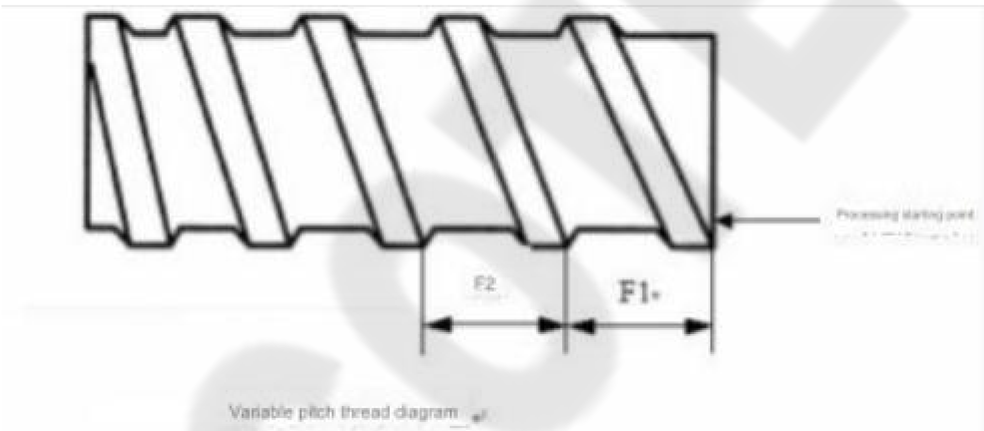


Figure 3-40

Notes:

●Notes are the same as G32 thread cutting.

Example: The first pitch at the start point is 4mm, and the incremental value of the pitch per spindle revolution is 0.2.

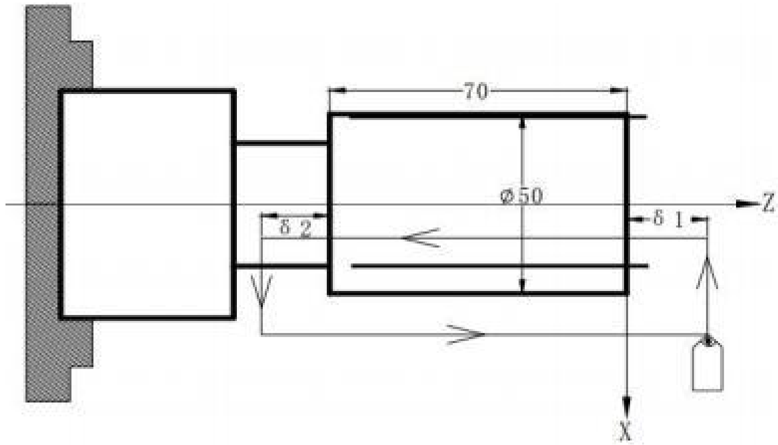


Figure 3-42 Variable thread processing

Program:
O0010;

G00 X60 Z1 M03 S500;

G00 X48;

G34 W-78 F3.8 J5 K2 R0.2;

N30 M30;

3.17.4 Z-axis Tapping Cycle G33

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

Code function: The tool movement trajectory is from the start point to the end point, and then from the end point back to the start point. During the movement, the Z-axis moves one pitch for each rotation of the spindle, which is always consistent with the pitch of the tap, forming a spiral groove in the inner hole of the workpiece, and the thread processing of the inner hole can be completed in one cutting.

Code description: G33 is a modal G code;

Z/W: When Z or W is not entered, 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 1-9;

I: The number of threads per inch, the value range is shown in Table 1-9;

L: The number of threads for multi-start threads, the value range is 1 ~ 99, and the default is 1 when L is omitted.

Cycle process:

① Z-axis feed tapping (spindle opening must be specified before G33 code);

② After reaching the programmed Z-axis coordinate end point, M05 signal output;

③ After the spindle is completely stopped;

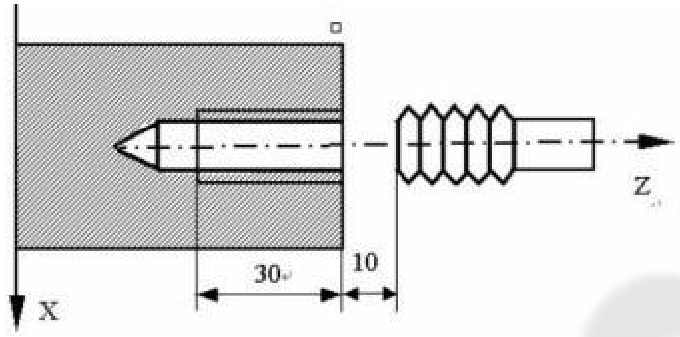
④ Clockwise signal output (opposite to the original spindle rotation direction);

⑤ Z-axis retracts to the start point;

⑥ M05 signal output, spindle stops;

⑦ If it is a multi-start thread, repeat steps ①~⑥.

Program example: Figure 3-43, thread M10×1.5



Program:

O0011 ;

G00 Z90 X0 M03; start spindle

G33 Z50 F1.5; tapping cycle

M03 restart spindle

G00 X60 Z100; continue processing

M30

Note 1: Before tapping, the spindle rotation direction should be determined according to the rotation direction of the tap. After tapping, the spindle will stop rotating. To continue processing, the spindle needs to be restarted.

Note 2: This code is for flexible tapping. After the spindle stop signal is valid, the spindle will still have a certain deceleration time before stopping. At this time, the Z-axis will still feed following the rotation of the spindle until the spindle stops completely. Therefore, the bottom hole position of the thread should be slightly deeper than the actual required position during actual processing. The specific excess length is determined by the spindle speed and the spindle brake device during tapping.

Note 3: The moving speed of the Z-axis during tapping is determined by the spindle speed and pitch, and has nothing to do with the cutting feed rate.

Note 4: In single-program segment operation or feed hold operation, the system displays "Pause", and the tapping cycle does not stop until the tapping is completed and returns to the start point.

Note 5: When the system is reset, during an emergency stop or its drive alarm occurs, the tapping cutting decelerates and stops.

3.17.5 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: Starting from the cutting start point, perform radial (X axis) feed and axial (Z axis or X and Z axis simultaneously) cutting to realize straight thread and taper thread cutting cycle with equal pitch. When executing G92 code, there is a thread back-off process at the end of thread machining: at a fixed length from the end point of thread cutting (called thread retraction length), while the Z axis continues to perform thread interpolation, the X axis accelerates to retract along the retraction direction exponentially or linearly (set by parameters). After the Z axis reaches the cutting end point, the X axis retracts at a quick moving speed, as shown in Figure 3-44.

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 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 start point;

Z: 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 start point;

R: the difference between the absolute coordinate of the X axis of the cutting start point and the cutting end point (radius value). When the signs of R and U are inconsistent, it is required to be |R| ≤ |U/2|;

F: thread lead, the value range is 0 < F ≤ 500 mm, F is retained after the specified value is executed, and the input can be omitted;

I: number of threads per inch, the value range is 0.06~25400 threads/inch, I is retained after the specified value is executed, and the input can be omitted;

J: the movement amount in the minor axis direction when the thread backs off, the value range is 0~99999.999 (unit: mm), without direction (the direction of thread back-off is automatically determined according to the program starting position), modal parameter; if the minor axis is the X axis, the value is specified by the radius;

K: The length of the thread in the major axis direction when the thread backs off, the value range is 0 ~ 99999.999 (unit: mm). Without direction, modal parameter; if the major axis is the X axis, the value is specified by the radius;

L: The number of threads of multi-start threads, the value range is: 1 ~ 99, modal parameter. (When L is omitted, the default is single-start thread)

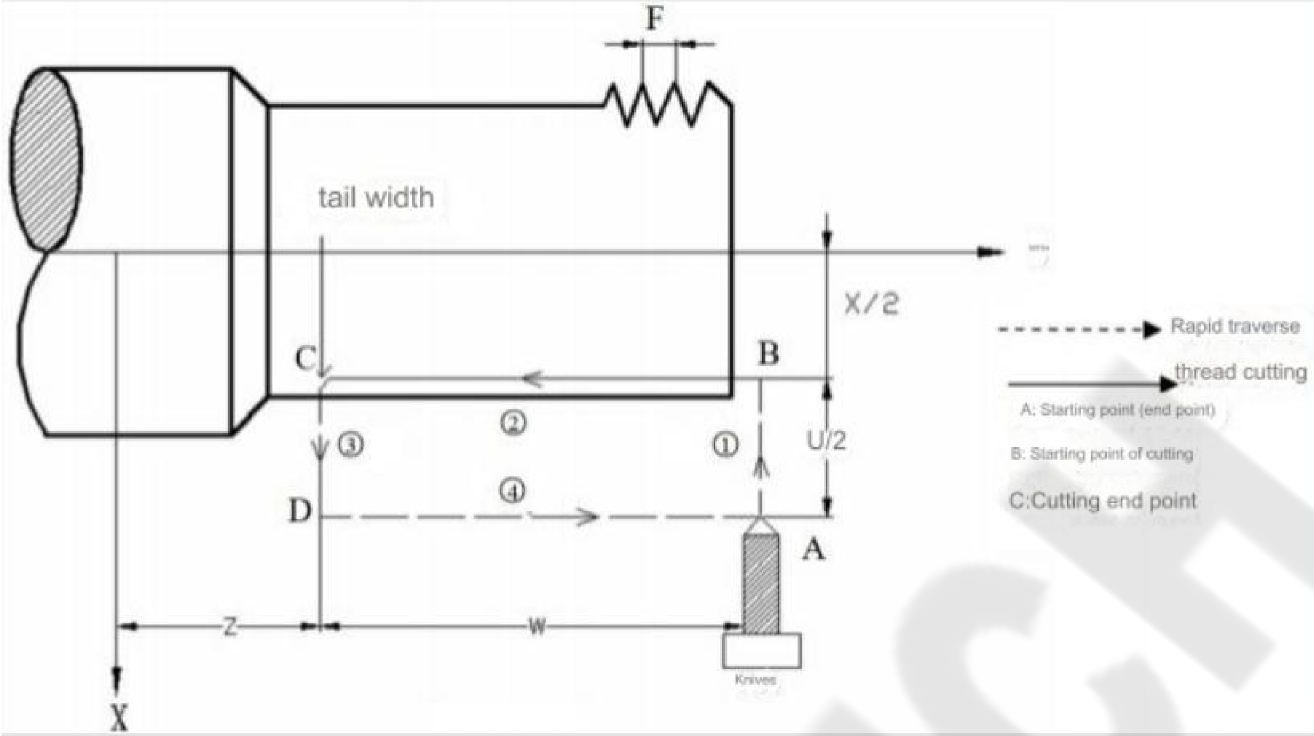


Figure 3-44

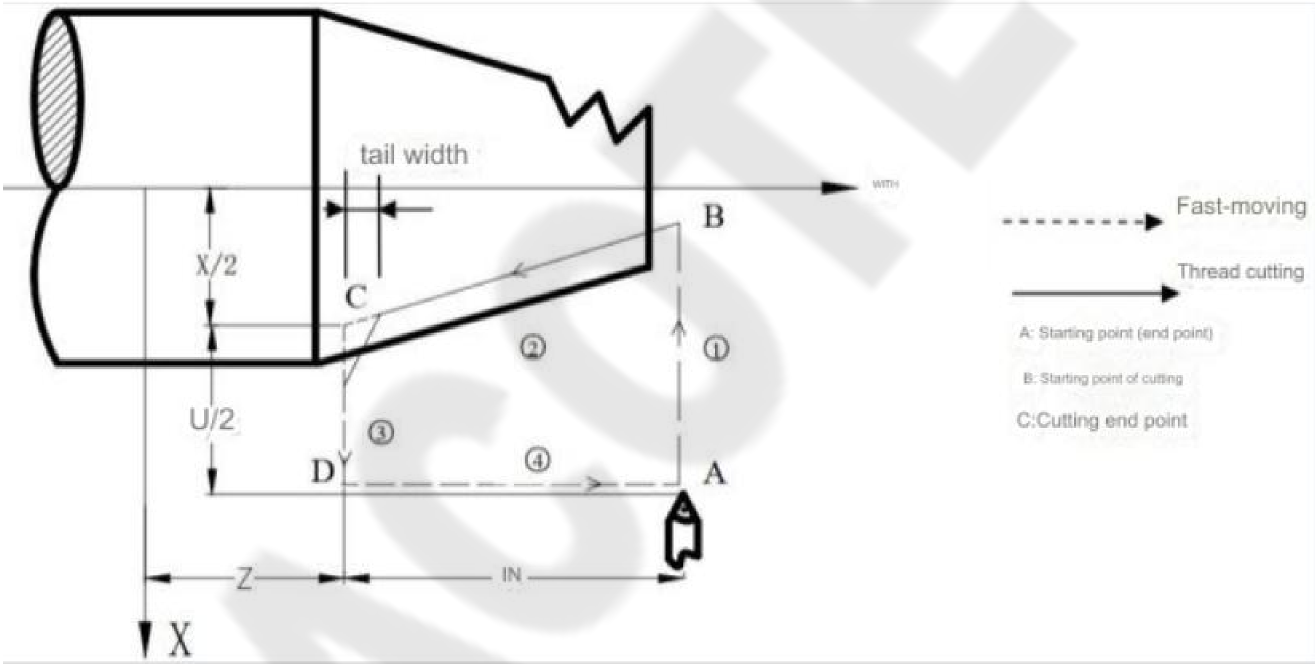


Figure 3-45

G92 code can complete the processing of a thread in multiple feeds, but it cannot realize the processing of two consecutive threads, nor can it process end face threads. The definition of thread pitch in G92 code is consistent with G32. The pitch refers to the displacement of the major axis when the spindle rotates one circle (the displacement of the X axis is based on the radius value).

The pitch of a tapered thread refers to the displacement of the major axis when the spindle rotates one circle (the displacement of the X axis is based on the radius value). When the absolute value of the Z axis coordinate difference between point B and point C is greater than the absolute value of the X axis coordinate difference (radius value), the Z axis is the major axis; otherwise, the X axis is the major axis.

Cycle process: straight thread is shown in Figure 3-44, and tapered thread is shown in Figure 3-45.
① The X axis moves quickly from the start point to the cutting start point;

② Thread interpolation from the cutting start point to the cutting end point;

③ The X axis retracts at a rapid moving speed (opposite to ①) and returns to the same absolute coordinate of the X axis as the starting point;
④ Z axis moves quickly back to the start point, and the cycle ends.

Notes:

● When J and K are omitted, the thread backs off according to the setting value of parameter N0183;

● When J is omitted, the thread backs off according to K in the ma axis direction and according to the setting value of parameter N0183 in the minor axis direction;

● When K is omitted, backs off by J=K;

● When J=0 or J=0 and K=0, there is no back-off;

● When J ≠ 0, K=0, backs off by K=J;

● When J=0, K ≠ 0, there is no back-off;

● After the feed hold operation is performed during thread cutting, the system still performs thread cutting. After the thread cutting is completed, "Pause" is displayed and the program operation is paused;

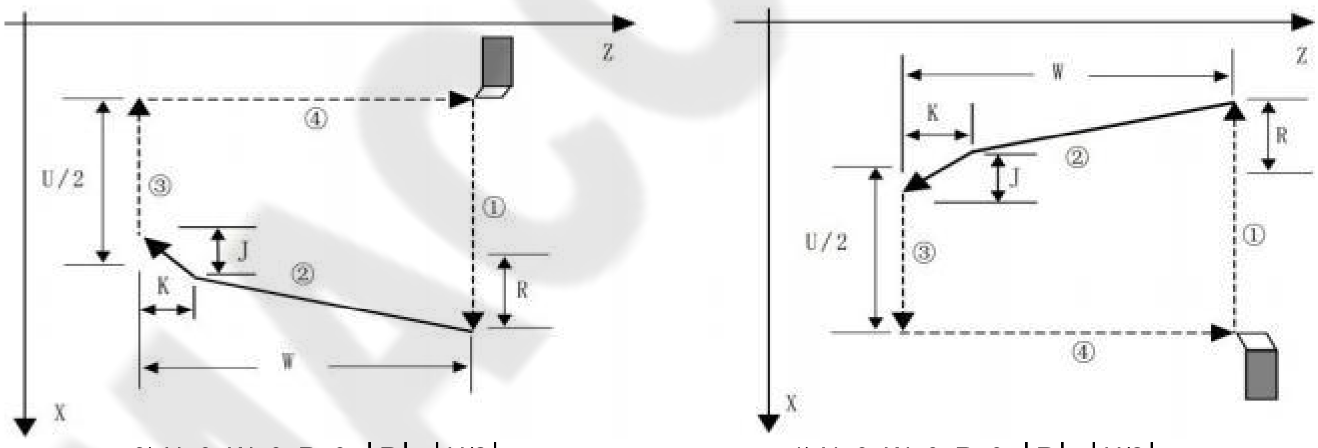
● After the single-program segment operation is performed during thread cutting, the operation stops after returning to the start point (one thread cutting cycle action is completed);

● When J and K are input with negative values, they are processed as positive values;

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

Code trajectory: U, W, R reflect the relative position of the thread cutting end point and the start point. When the symbols are different, the tool trajectory and the direction of the back-off are as shown in the figure:

1) U>0, W<0, R>0 2) U<0, W<0, R<0



3) U>0, W>0, R<0, │R│≤│U/2│ 4) U<0, W>0, R>0, │R│≤│U/2│

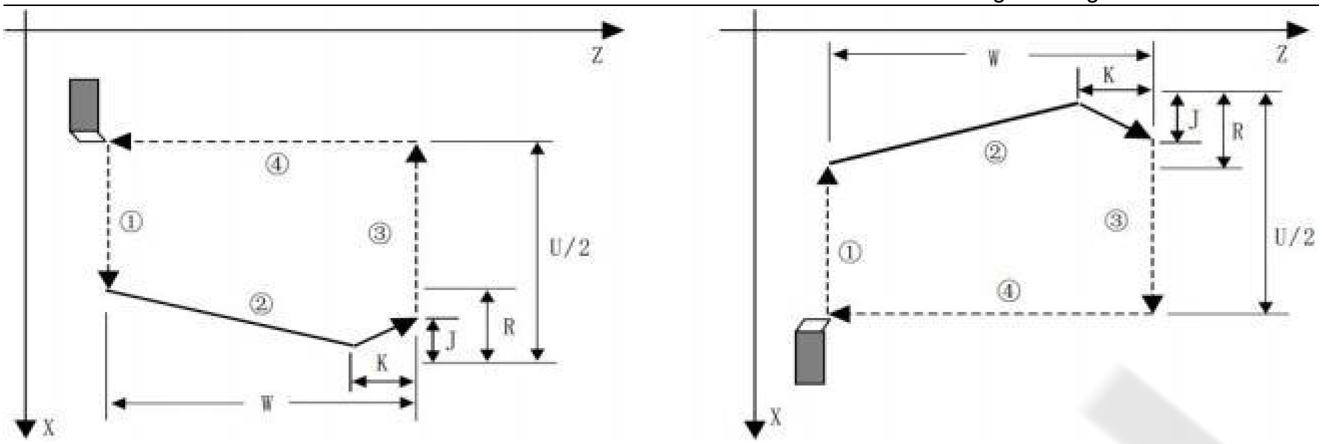


Figure 3-46

Example: Figure 3-47

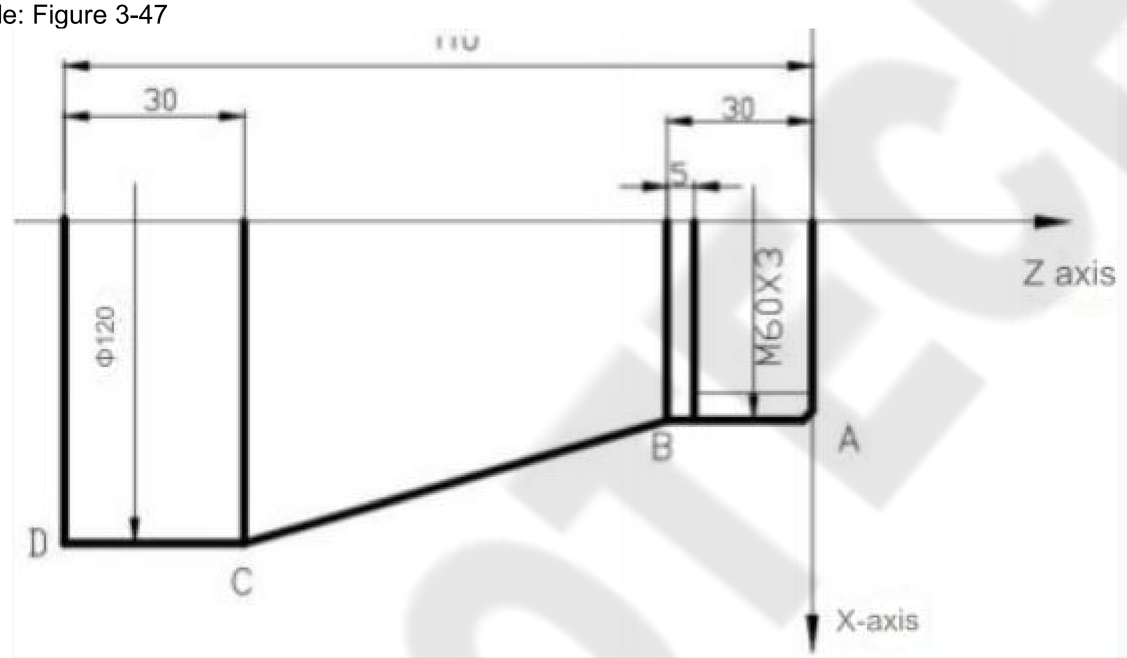


Figure 3-47

Program:
00012 ;

M3 S300 G0 X150 Z50 T0101;	(Thread cutter)
G0 X65 Z5;	(Quick positioning)
G92 X58.7 Z-28 F3 J3 K1;	(Thread processing, cut in 4 times, first feed is 1.3mm)
X57.7;	(Second feed is 1mm)
X57;	(Third feed is 0.7mm)
X56.9;	(Fourth feed is 0.1mm)
M30;	

Thread infeed

G92 X(U)_ Z(W)_ R_ F_ J_ K_ L_ P_

J: Minor axis back-off

K: Major axis back-off

L: Number of threads for multiple threads

P: X-axis infeed distance, mm, radius value

3.17.6 Multiple 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: Complete thread processing of the specified tooth height (total cutting depth) through multiple thread roughing and thread finishing. If the defined thread angle is not 0 °, the entry point of the thread roughing is gradually moved from the thread crest to the thread bottom, so that the angle between the two adjacent threads is the specified thread angle. G76 code can process straight threads and tapered threads with thread back-off, and can realize single-side blade thread cutting, and the cutting amount is gradually reduced, which is beneficial to protect the tool and improve the thread accuracy. G76 code cannot process end face threads. The machining trajectory is shown in Figure 3-48(a).

Related definitions:

Start point (end point): the position before and at the end of the program segment, represented by point A;

Thread end point: the thread cutting end point defined by X/U_ Z/W_, represented by point D. If there is thread back-off, the major axis direction of the cutting end point is the thread cutting end point, and the minor axis direction is the position after back-off.

Thread start point: the absolute coordinate of the Z axis is the same as point A, and the difference between the absolute coordinate of the X axis and the absolute coordinate of the X axis of point D is i (thread taper, radius value), which is expressed as point C. If the defined thread angle is not 0°, point C cannot be reached during cutting;

Thread cutting depth reference point: the absolute coordinate of the Z axis is the same as point A, and the difference between the absolute coordinate of the X axis and the absolute coordinate of the X axis of point C is k (total cutting depth of the thread, radius value), which is expressed as point B. The thread cutting depth of point B is 0, which is the reference point for the system to calculate the thread cutting depth for each time;

Thread cutting depth: the cutting depth of each thread cutting cycle. The intersection of the reverse extension line of each thread cutting trajectory and the straight line BC, the difference between this point and the absolute coordinate of the X axis of point B (unsigned, radius value) is the thread cutting depth. The thread cutting depth of each rough turning is $\sqrt{n}$×△d, n is the current number of rough turning cycles, △d is the thread cutting depth of the first rough turning;

Thread cutting amount: the difference between the current thread cutting depth and the previous thread cutting depth: ($\sqrt{n}$-$\sqrt{n-1}$) ×△d;

End point of tool retraction: the end point position of radial (X axis) tool retraction after the thread cutting is completed in each thread rough turning cycle and fine turning cycle, which is expressed as point E;

Thread entry point: the point where thread cutting actually starts in each thread rough turning cycle and fine turning cycle, expressed as point Bn (n is the number of cutting cycles), B1 is the first thread rough turning entry point, Bf is the last thread rough turning entry point, and Be is the thread fine turning entry point. The displacement of point Bn relative to point B on the X and Z axes conforms to the formula:

$$tg\frac{a}{2} = \frac{\text{Z-axis offset}}{\text{X-axis offset}}$$, a: thread angle

X: absolute coordinate of the X axis of the thread end point;

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

P(m): number of thread finishing operations 00~99 (unit: times), m specifies a value that remains valid after execution, and changes the value of system parameter N0193 to m. When m is not entered, the value of system parameter N0193 is used as the number of finishing operations. During thread finishing, the cutting amount of each feed is equal to the cutting amount of thread finishing.

P(r): thread back-off length 00~99 (unit: 0.1×L, L is the thread pitch), the specified value of r remains valid after execution, and the value of system parameter N0183 is changed to r. When r is not entered, the value of system parameter N0183 is used as the thread back-off width. The thread back-off function can realize thread processing without back-off groove. The thread back-off width defined by system parameter N0183 is valid for G92 and G76 codes;

P(a): the angle between two adjacent threads, the value range is 00~99, unit: degree (°), the specified value of a remains valid after execution, and the value of system parameter N0194 is changed to a. When a is not entered, the value of system parameter N0194 is used as the angle of the thread. The actual thread angle is determined by the tool angle, so a should be the same as the tool angle;

Q(△dmin): The minimum cutting amount during thread roughing, the value range is 00~99999 (unit: 0.001mm, radius value). When $\sqrt{n} - \sqrt{n-1}$ ×△d < △dmin, △dmin is used as the cutting amount for current roughing, that is: current thread cutting depth is ($\sqrt{n-1}$ ×△d +△dmin). △dmin is set to avoid too small roughing cutting amount and too many roughing times due to the decreasing thread roughing cutting amount. After Q(△dmin) is executed, the specified value △dmin remains valid, and the value of system parameter N0195 is changed to △dmin. When Q(△dmin) is not entered, the value of system parameter N0195 is used as the minimum cutting amount;

R(d): The cutting amount of thread finishing, 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 coordinates of the thread finishing entry point Be and the last thread roughing entry point Bf. After R(d) is executed, the specified value d remains valid, and the value of system parameter N0196 is modified to d×1000(IS_B)/d×1000(IS_C). When R(d) is not entered, the value of system parameter N0196 is used as the thread finishing cutting amount;

R(i): Thread taper, the difference between the X-axis absolute coordinates 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 processes it as R(i)=0 (straight thread);

P(k): thread height, total thread cutting depth, the value range is 1~99999999×least input increment (radius value, unsigned). When P(k) is not entered, the system alarms;

Q(△d): first thread cutting depth, the value range is 1~99999999×least input increment (radius value, unsigned). When △d is not entered, the system alarms;

F: thread lead, the value range is 0<F≤500 mm;

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

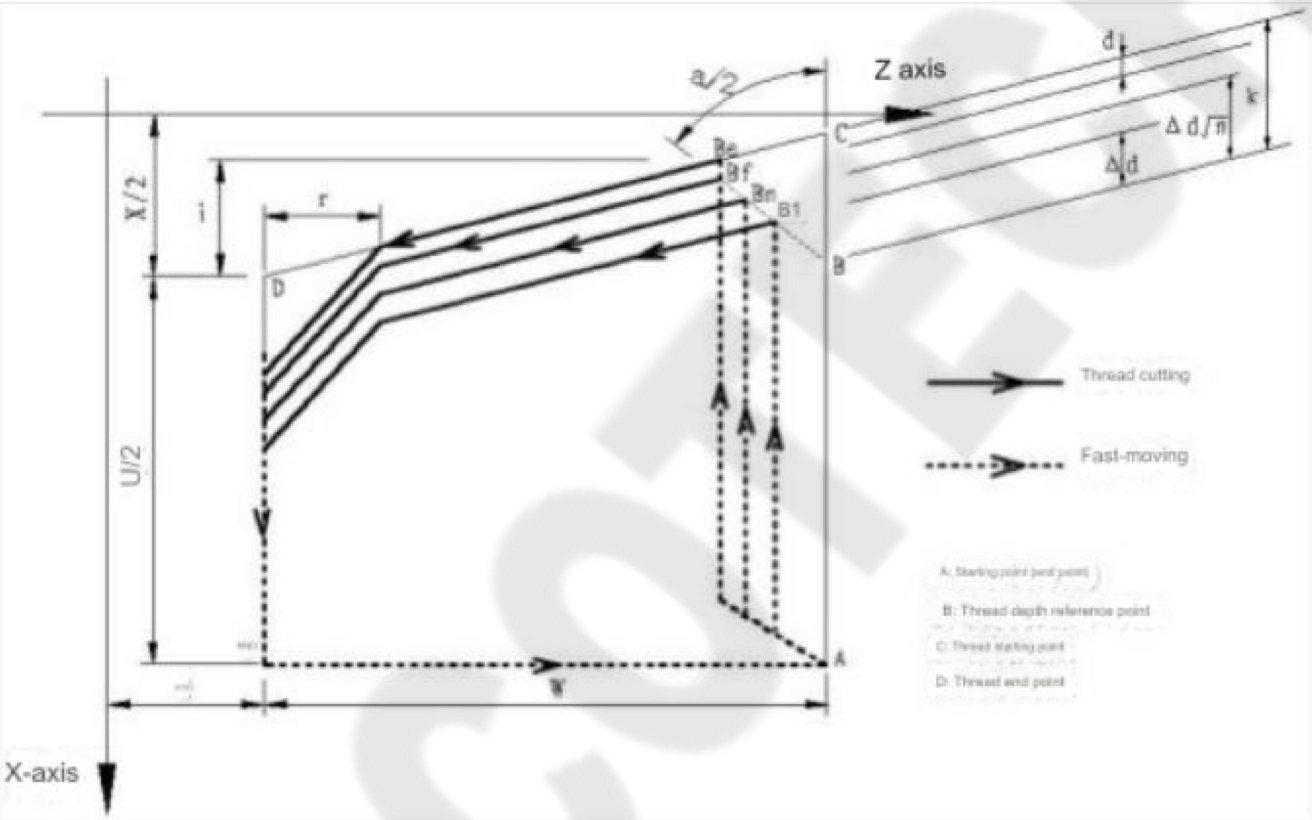


Figure 3-48 (a)

For details of the cutting method, see Figure 3-48 (b):

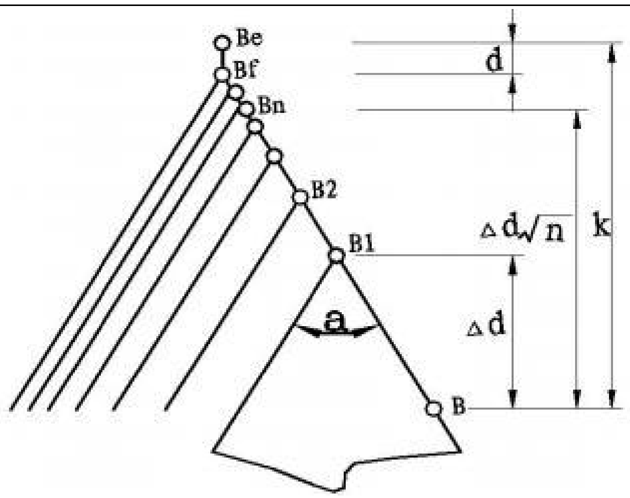


Figure 3-48 (b)

The thread pitch refers to the displacement of the major axis when the spindle rotates one circle (the displacement of the X axis is based on the radius value). When the absolute value of the Z axis coordinate difference between point C and point D 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 major axis; otherwise, the X axis is the major axis.

Code execution process:

① Quickly move from the start point to B1, and the thread cutting depth is △d. If a=0, only the X axis moves; if a ≠ 0, the X axis and Z axis move simultaneously, and the moving direction is the same as that of A → D;

② Cut the thread in the direction parallel to C → D to the intersection with D → E (there is a back-off process when r ≠ 0);

③ The X axis moves quickly to point E;

④ The Z axis moves quickly to point A, and a single rough turning cycle is completed;

⑤ Move quickly again to feed to Bn (n is the number of rough turning times), and the cutting depth is the larger value of ($\sqrt{n}$ ×△d) and ($\sqrt{n-1}$ ×△d +△dmin). If the cutting depth is less than (k-d), go to execute ②; if the cutting depth is greater than or equal to (k-d), feed to point Bf according to the cutting depth (k-d), and go to ⑥ to execute the last thread rough turning;

⑥ Cut the thread in the direction parallel to C → D to the intersection with D → E (there is a back-off process when r ≠ 0);

⑦ The X axis moves quickly to point E;

⑧ The Z axis moves quickly to point A, the thread rough turning cycle is completed, and the thread finishing cycle begins;

⑨ After moving quickly to point Be (thread cutting depth is k, cutting amount is d), perform thread finishing, and finally return to point A to complete a thread finishing cycle;

⑩ If the number of finishing cycles is less than m, turn to ⑨ for the next finishing 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 compound thread processing cycle ends.

Notes:

● After the feed hold operation is performed during thread cutting, the system still performs thread cutting. After the thread cutting is completed, "Pause" is displayed and the program operation is paused;

● After the single-program segment operation is performed during thread cutting, the operation stops after returning to the start point (one thread cutting cycle action is completed);

● When the system is reset, emergency stop or drive alarm occurs, thread cutting decelerates and stops;

● G76 P(m)(r)(a) Q(△dmin) R(d) can omit all or part of the code address, and the omitted address runs according to the parameter setting value;

● m, r, and a are input once with the same code address P. When m, r, and a are all omitted, it runs according to the setting values of parameters N0193, N0183, and N0194; when address P is input with 1 or 2 digits, the value is a; when address P is input with 3 or 4 digits, the value is r and a;

● The signs of U and W determine the direction of A → C → D → E, and the sign of R(i) determines the direction of C → D. There are four combinations of U and W symbols, corresponding to four processing trajectories, as shown in Figure 3-46.

Example: Figure 3-49, the thread is M68×6.

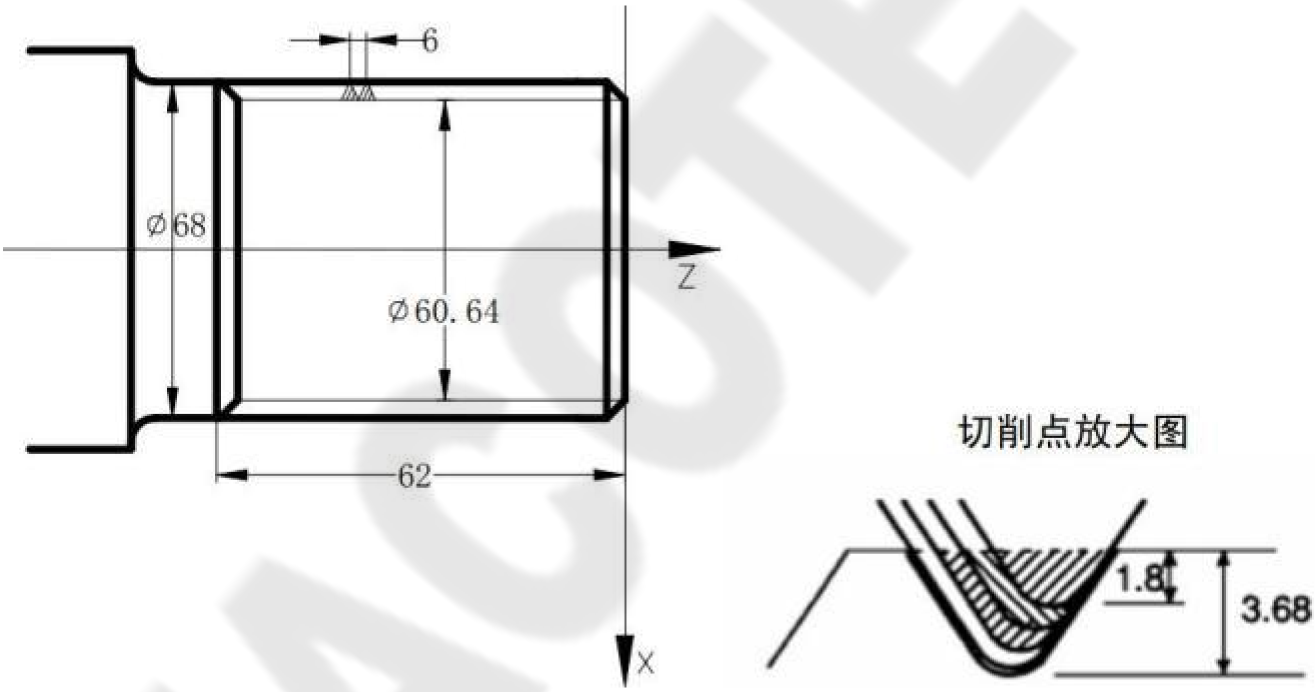


Figure 3-49

Program: O0013;

G50 X100 Z5 M3 S300; (Set the workpiece coordinate system, start the spindle, and specify the speed)

G00 X80 Z10; (Quickly move to the processing start point)

G76 P020560 Q150 R0.1; (Finishing repetitions 2, chamfer width 0.5mm, tool angle 60°, minimum cutting depth 0.15, finishing allowance 0.1)

G76 X60.64 Z-62 P3680 Q1800 F6; (Thread height 3.68, first thread cutting depth 1.8)

G00 X100 Z50; (Return to the program start point)

M30; (Program end)

3.17.7 T-type Thread Cutting Cycle G78

Code format: G78 P_ R_ Q_ L_

G78 X/U_ Z/W_ R_ F(I)_ J_ K_ P_

Code function: Use left and right tools to turn threads in layers. Three tools for each layer, first turn the middle, then turn the left and right tools on both sides.

Related definitions:

P: thread angle

R: feed amount per time

Q: tool width

L: thread bottom width

X: absolute coordinate of the X axis of the thread end point;

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

R: thread taper

F: lead

3.18 Constant Linear Speed Control G96, Constant Speed Control G97

For detailed description, please refer to Section 2.2.3 of this part.

3.19 Feed per Minute G98, Feed per Revolution G99

Code format: G98 F_; (leading zero can be omitted, feed per minute is given)

Code function: give cutting feed rate in mm/min, G98 is modal G code, if the current mode is G98, G98 can be omitted.

Code format: G99 F_;

Code function: give cutting feed rate in mm/rev, G99 is modal G code. If the current mode is G99, G99 can be omitted. When CNC executes G99 F_, the product of F code value (mm/rev) and current spindle speed (r/min) is used as code feed rate to control actual cutting feed rate. When spindle speed changes, actual cutting feed rate changes accordingly. Using G99 F_ to give cutting feed per spindle revolution can form uniform cutting lines on the workpiece surface. To process in G99 mode, the machine tool must be equipped with a spindle encoder.

G98 and G99 are modal G codes of the same group, and only one can be valid. G98 is the initial G code, and G98 is valid by default when the CNC is powered on. The conversion formula for feed per revolution and feed per minute: $Fm = Fr \times S$

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 system parameter N0027. After executing F0, the feed speed is 0. When the CNC is reset or stops urgently, the F value remains unchanged.

Note: In G99 mode, when the spindle speed is lower than 1r/min, the cutting feed speed will be uneven; when the spindle speed fluctuates, the actual cutting feed speed will have a following error. In order to ensure the processing quality, it is recommended that the spindle speed selected during processing should not be lower than the minimum speed of the spindle servo or inverter output effective torque.

Related parameters:

CNC parameter N0013: upper limit of cutting feed rate;

CNC parameter N0029: exponential acceleration and deceleration time constant during cutting feed and manual feed;
CNC parameter N0014: starting (ending) speed during cutting feed.

3.20 Chip Breaking Function G104

Instruction format: G104 K_ L_ chip breaking instruction start

G104 chip breaking instruction end

Where:

K: dwell time
L: chip breaking distance

Description:

1. The chip breaking instruction is only used for linear and circular processing instructions G01, G02, G03, and is invalid for other instructions. After turning, be sure to turn off the chip breaking processing function. Chip breaking state is turned off after resetting or program end.

2. The spindle must be in speed mode and cannot be in position mode of servo spindle.

3. Attention for parameter K: The smaller the setting, the shorter the dwell time is. However, if the setting is too small, the feed axis speed may not reach 0 due to acceleration and deceleration problems, resulting in no chip breaking function. However, the thickness of the chip strips still varies.

4. The shorter the parameter L, the more frequent the stops, so the relative processing time will be much longer. The longer L, the longer the chip strips will be.

5. Lathe function.
Note:

1. When one of K and L is 0, the function cannot be turned on. K cannot be negative, an alarm will be triggered, and L will take the absolute value if it is negative.
2. The unit of K is seconds, and the unit of L is mm.

3. The chip breaking function is only valid for G01, G02, and G03. G01 calculates the straight-line distance between two points, G02 and G03 calculate the arc length. When the distance set by L is reached, the current F multiplier is changed to 0 and the timing starts. When the time set by K is reached, the F multiplier is restored to the previous value, and the distance is recalculated, compared with L again, and repeated.

4. Executing G104 alone will turn off the function. The function will also be turned off when the program ends and is reset.

Turn on chip breaking: G104 L_chip breaking distance K_dwell time, in seconds P_number of chip breaking circles; (when P and K exist at the same time, K is ignored)

Turn off chip breaking: G104

3.21 Planing G105

G105 P_ K_ I_ U_ Q_

P: feed amount each time for deep planning and tool width for plane planing, unsigned

K: tool lift amount after the Z axis advances to the position, signed

I: tool lift amount after the X axis advances to the position, signed

U: width, signed

Q: 0-deep planing, 1-plane planing

G106 //cancel planing mode

3.22 Set G150 and Cancel Follower Axis G151

Set follower axis G150 P_active axis Q_follower axis I_(follower axis divided by the main axis ratio); (1:X 2:Z 3:Y 4:A 5:B 6:C)

Cancel follower axis G151 P_active axis (1:X 2:Z 3:Y 4:A 5:B 6:C); If P does not exist, cancel all follower axes

3.23. Torque Detection G152

G152 P_

P: Maximum load. If the load needs to exceed a period of time before alarming, the lower 3 bits are the load, and the upper bit is the overload time, in milliseconds

Torque limit skip. When the specified torque is reached, the axis movement is interrupted and the next program segment is entered.

3.24 Torque Limit Skip G160

G160 XYZ_ P_ Q_ F_

XYZ_: Axis number, only one axis can be specified, and the specified axis movement amount

P: Torque limit value, unit: percentage

Q: Following error, unit: mm

F: Feed speed

3.25 Macro Code

Macro codes similar to high-level languages are provided. User macro codes can realize variable assignment, arithmetic operations, logical judgment and conditional transfer, which is conducive to compiling processing programs for special parts, reducing tedious numerical calculations during manual programming, and streamlining user programs.

3.25.1 Macro Variables

●Variable representation

Variables are specified using the symbol “#” + variable number;

Format: #i (i=100, 102, 103, ...); examples: #105, #109, #125.

● Variable type

Variables can be divided into four types according to the variable number.

Variable number	Variable type	Function
#0	Null variable	This variable is always null and no value can be assigned to it.
#1 ～ #50	Local variables	Local variables can only be used to store data in macro programs, such as calculation results. When power is off, local variables are initialized to empty. When calling a macro program, the independent variable assigns a value to the local variable.
#100 ～ #199 #500 ～ #999	Public variables	Public variables have the same meaning in different macro programs. When the power is off, variables #100 ~ #199 are initialized to null, and the values of variables #500 ~ #999 are saved and will not be lost even if the power is off.
#1000 ～ #5235	System variables	System variables

● Variable reference

The value after replacing the address with the variable.

Format: <Address> +“#I” or <Address> +“-#I”, which means to use the value of variable “#I” or the negative value of variable “#I” as the address value.

Example: F#103… when #103=15, it has the same function as F15 code;

Z-#110… when #110=250, it has the same function as Z-250 code;

Note 1: Addresses O, G and N cannot reference variables. For example, O#100, G#101, and N#120 are illegal references;

Note 2: If the maximum code value specified by the address is exceeded, it cannot be used; For example: When #150 = 120, M#150 exceeds the maximum code value.

● Null variable

When the variable value is undefined, the variable is null. Variable #0 is always a null variable. It cannot be written but can only be read.

When referencing an undefined variable (null variable), the address itself is also ignored.

When #1=<null>	When #1=0
G00 X100 Z#1 is equivalent to G00 X100	G00 X100 Z#1 is equivalent to G00 X100 Z0

● Display of variables

(1) In the macro variable page, when the variable is blank, it means that the variable is a null variable, that is, it has not been defined.

(2) The values of public variables (#100 ~ #199, #500 ~ #999) are displayed in the macro variable page. You can also directly enter data to assign values to public variables in this page.

(3) The values of local variables (#1 ~ #50) and system variables can't be displayed. If you need to view the value of a local variable or system variable, you can display it by assigning it to a public variable.

● System variables - as follows:

1) Interface input signal	#1000 --- #1047	(Read the signal input by PMC bit by bit)
2) Interface output signal	#1100 --- #1147	(Write the signal output to PMC bit by bit)
3) X-axis length compensation value	#1500 --- #1531	(Radius value, readable and writable)
4) Z-axis length compensation value	#1600 --- #1631	(Readable and writable)
5) Y-axis length compensation value	#1700 --- #1731	(Readable and writable)
6) Tool radius compensation value	#1800 --- #1831	(Readable and writable)
7) X-axis wear compensation value	#1900 --- #1931	(Radius value, readable and writable)
8) Z-axis wear compensation value	#2000 --- #2031	(Readable and writable)
9) Y-axis wear compensation value	#2100 --- #2131	(Readable and writable)
10) Radius wear compensation value	#2200 --- #2231	(Readable and writable)
11) Alarm	#3000	
12) User data table	#3500 --- #3755	(Read only)
13) Modal information	#4000 --- #4030	(Read only)
14) Position information	#5001 --- #5030	(Read only)

Detailed description of system variables

(1) Interface signal: CNC only operates on G and F signals. Whether there is a corresponding I/O number depends on the specific PLC definition.

Variable number	Function
#1000 ~ #1015	Corresponding to G54.0 ~ G54.7, G55.0 ~ G55.7 signal status of the system
#1032	Corresponding to G54, G55 two-byte signal status of the system
#1100 ~ #1115	Corresponding to F54.0 ~ F54.7, F55.0 ~ F55.7 signal status of the system
#1132	Corresponding to F54, F55 two-byte signal status of the system
#1133	Corresponding to F56, F57, F58, F59 four-byte signal status of the system

(2) Tool compensation system variables:

Compensation number	Offset compensation value				Wear compensation value			
	X axis	Z axis	Y axis	Radius	X axis	Z axis	Y axis	Radius
1 ··· 32	#1500 … #1531	#1600 … #1631	#1700 … #1731	#1800 … #1831	#1900 … #1931	#2000 … #2031	#2100 … #2131	#2200 … #2231

(3) System modal information variable

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
#4120	F code	
#4121	M code	
#4122	Sequence number	
#4123	Program number	

#4124	S code
#4125	T code

(4) System variables of coordinate position information:

Variable number	Position signal	Coordinate system	Tool compensation value	Read operation during motion
#5001 ～ #5005	End point of program segment	Workpiece coordinate system	Not included	Yes
#5006 ～ #5010	Current position (machine coordinates)	Machine coordinate system	Included	No
#5011 ～ #5015	Current position (absolute coordinates)	Workpiece coordinate system		

Note: The position information listed in the above table corresponds to the X axis, Z axis, Y axis, 4th axis, and 5th axis in order. For example: #5001 indicates the position information of the X axis, #5002 indicates the position information of the Z axis, #5003 indicates the position information of the Y axis, #5004 indicates the position information of the 4th axis, and #5005 indicates the position information of the 5th axis;

(5) Workpiece origin 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

●The correspondence between local variable addresses and local variables:

Independent variable address	Local variable number	Independent variable address	Local variable number	Independent variable 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		

3.25.2 Operation Instruction and Transfer Instruction G65

General code format: G65 H(m) P(#i) Q(#j) R(#k);
Where: m: indicates the operation instruction or transfer instruction function.

#i: the variable name where the operation result is stored.

#j: the variable name 1 to be operated, which can be a constant.

#k: the variable name 2 to be operated, which can be a constant.

Code meaning: #i = #j O #k

Operation symbol, determined by Hm

Example: P#100 Q#101 R#102…..#100 = #101 0 #102;

P#100 Q#101 R15….#100 = #101 0 15;

P#100 Q-100 R#102…..#100 = -100 0 #102;

Description: "#" cannot be used when the variable is a constant;

Macro operation (jump) table

Code format	Function	Definition
G65 H01 P#I Q#J;	Assignment operation	#i = #j; Assign the value of variable #j to variable #i
G65 H02 P#i Q#j R#k;	Decimal addition operation	#i = #j + #k
G65 H03 P#i Q#j R#k;	Decimal subtraction operation	#i = #j - #k
G65 H04 P#i Q#j R#k;	Decimal multiplication operation	#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 exclusive OR	#i = #j XOR #k
G65 H21 P#i Q#j;	Decimal square root	#i= $\sqrt{\# j}$
G65 H22 P#i Q#j;	Decimal absolute value	#i = |#j|
G65 H23 P#i Q#j R#k;	Decimal remainder	Remainder of #i = (#j÷# k)
G65 H24 P#i Q#j;	Decimal to binary	#i = BIN(#j)
G65 H25 P#i Q#j;	Binary to 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;	Tangent	#i = #j×tan (#k)

G65 H34 P#i Q#j R#k;	Inverse tangent	#i = ATAN(#j / #k)
G65 H80 Pn;	Unconditional transfer	Jump to program segment n
G65 H81 Pn Q#j R#k;	Conditional transfer 1	If # j = # k, jump to program segment n, otherwise execute sequentially
G65 H82 Pn Q#j R#k;	Conditional transfer 2	If # j ≠ # k, jump to program segment n, otherwise execute sequentially
G65 H83 Pn Q#j R#k;	Conditional transfer 3	If # j > # k, jump to program segment n, otherwise execute sequentially
G65 H84 Pn Q#j R#k;	Conditional transfer 4	If # j < # k, jump to program segment n, otherwise execute sequentially
G65 H85 Pn Q#j R#k;	Conditional transfer 5	If # j ≥ # k, jump to program segment n, otherwise execute sequentially
G65 H86 Pn Q#j R#k;	Conditional transfer 6	If # j ≤ # k, jump to program segment n, otherwise execute sequentially
G65 H99 Pn;	Generate user alarm	Generate user alarm (3000+n)

1. Operation instruction

1) Variable assignment: #I = #J

G65 H01 P#I Q#J

(Example) 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 operation: #I = #J+#K
(Example) G65 H02 P#101 Q#102 R15; (#101 = #102+15)

3) Decimal subtraction: #I = #J - #K

G65 H03 P#I Q#J R# K

(Example) 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

(Example) 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 addition (OR): #I = #J OR #K

G65 H11 P#I Q#J R#K

(Example) G65 H11 P#101 Q#102 R#103; (#101 = #102.OR. #103)

7) Binary logical multiplication (AND): #I = #J AND #K

G65 H12 P#I Q#J R#K

(Example) G65 H12 P# 101 Q#102 R#103; (#101 = #102.AND.#103)

8) Binary exclusive OR: #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 square root: #I = $\sqrt{\#J}$

G65 H21 P#I Q#J

(Example) G65 H21 P#101 Q#102; (#101 = $\sqrt{\#102}$)

10) Decimal 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: round off the decimal part

G65 H23 P#I Q#J R#K

(Example) 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

(Example) 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 operations: #I = (#I×#J)÷#K
G65 H26 P#I Q#J R# k

(例)G65 H26 P#101 Q#102 R#103； (#101 =(# 101×# 102)÷# 103)

15) Composite 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: degree)

G65 H31 P#I Q#J R#K

(Example) G65 H31 P#101 Q#102 R#103; (#101 = #102•SIN(#103))
17) Cosine: #I = #J•COS(#K) (Unit: degree)

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•TAM(#K) (Unit: degree)

G65 H33 P#I Q#J R#K

(Example) G65 H33 P#101 Q#102 R#103; (#101 = #102•TAM(#103))
19) Inverse tangent: #I = ATAN(#J /#K) (Unit: degree)

G65 H34 P#I Q#J R#k

(Example) G65 H34 P#101 Q#102 R#103; (#101 =ATAN(#102/#103))

2. Transfer instruction

1) Unconditional transfer G65 H80 Pn;
n: Sequence number

(Example) G65 H80 P120; (Transfer to N120 program segment)
2) Conditional transfer 1 #J EQ #K (=)

G65 H81 Pn Q#J R#K; n: Sequence number
(Example) G65 H81 P1000 Q#201 R#202;

When #101 = #102, transfer to N1000 program segment, when #101 ≠ #102, execute sequentially.
3) Conditional transfer 2 #J NE #K (≠)

G65 H82 Pn Q#J R# K; n: sequence number
(Example) G65 H82 P1000 Q#101 R#102;

When #101 ≠ #102, transfer to N1000 program segment. When #101 = #102, the program is executed sequentially.
4) Conditional transfer 3 #J GT #K (>)

G65 H83 Pn Q#J R# K; n: sequence number
(Example) G65 H83 P1000 Q#101 R#102;

When #101 > #102, transfer to N1000 program segment. When #101 ≤ #102, the program is executed sequentially.
5) Conditional transfer 4 #J LT #K (<)

G65 H84 Pn Q#J R# K; n: sequence number
(Example) G65 H84 P1000 Q#101 R#102;

When #101 < #102, transfer to N1000 program segment. When #101 ≥ #102, the program is executed sequentially.
6) Conditional transfer 5 #J GE #K (≥)

G65 H85 Pn Q#J R# K; n: sequence number
(Example) G65 H85 P1000 Q#101 R#102;

When #101 ≤ #102, transfer to N1000 program segment. When #101 < #102, the program is executed sequentially.

7) Conditional transfer 6 #J LE #K (≤)
G65 H86 Pn Q#J R# K; n: sequence number

(Example) G65 H86 P1000 Q#101 R#102;

When #101 ≤ #102, transfer to N1000 program segment, when #101 > #102, the program is executed sequentially.

8) P/S alarm occurs

G65 H99 Pi; i: alarm number +500

(Example) G65 H99 P15; P/S alarm 515 occurs.

Note: The sequence number can be specified by a variable. For example: G65 H81 P#100 Q#101 R#102; when the condition is met, the program moves to the program segment with the sequence number specified by #100.

3.25.3 Macro Program Call Code

The difference between user macro program call (G65) and subroutine call (M98) is as follows:

1. G65 can specify independent variable data and transfer it to the macro program, while M98 does not have this function.
2. G65 can change the level of local variables, while M98 cannot.
3. G65 code is only allowed to be preceded by code word N and followed by code word P or H.

Non-modal call (G65)

Code format: G65 P_ L_ <independent variable>_; the macro program specified by address P is called, and the independent variable (data) is passed to the user macro program body.

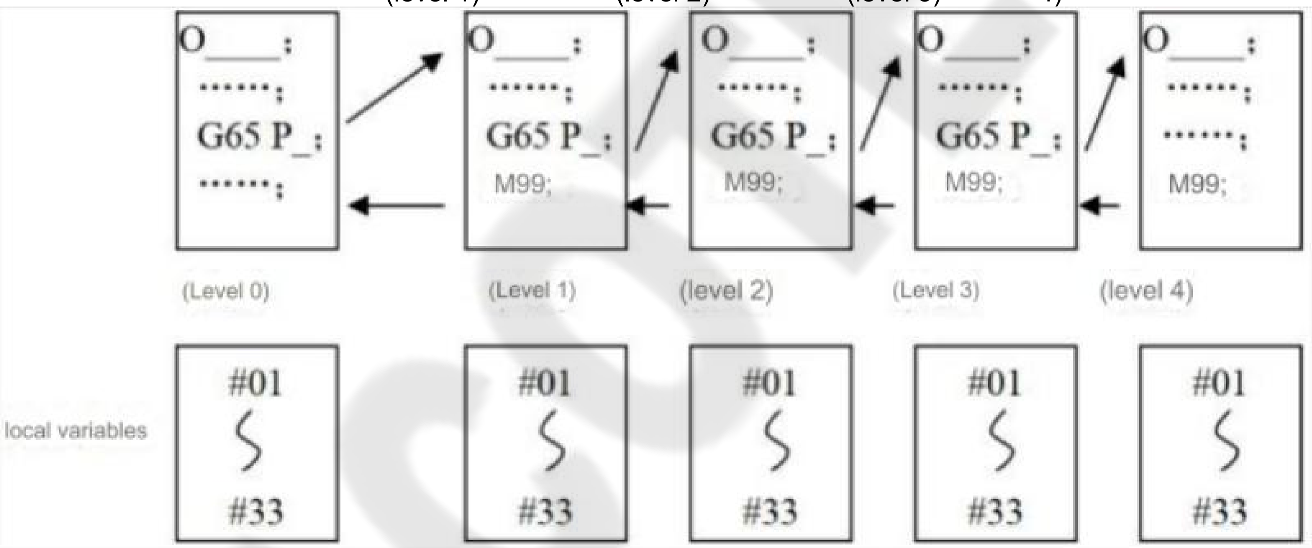
Code description: P_ number of the macro program called

L_ number of times it is called (if omitted, the default is 1, and the number of repetitions can be specified from 1 to 9999)

<Independent variable>_the data transferred to the macro program, and its value is assigned to the corresponding local variable.

Nested call: G65 call can have four levels of nesting.

Main program | Macro program (level 1) | Macro program (level 2) | Macro program (level 3) | Macro program (level 4)



Specification of independent variables: Use letters other than G, L, O, N, P. Each letter can only be specified once. If it is specified repeatedly, the last specified one is valid.

Independent variable address	Local variable number	Independent variable address	Local variable number	Independent variable 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: The address that does not need to be specified can be omitted. The local variable corresponding to the omitted address will be assigned to <null>.

3.25.4 Instructions for Using Macro B Instruction

I. Format and reference:

Variable representation: #I (I=1,2,3...) or #[<formula>]
Use of variables:

1. Specify the variable number or formula after the address word. Format: <address word>#I, <address word>#[<formula>]

2. The variable number can be replaced by a variable, for example: #[#30], if #30=3, then it is #3.

[<mathematical expression>]: The operation formula can be directly described by referencing variables or variable numbers.

Example: 1. X[#1+#2-12] Y[#24+#18*COS[#1]]
2. #20=#500*SIN[#120]

Note: If a mathematical expression is to be followed by an address symbol X/Z, it must be enclosed in [].

II. Arithmetic and logical operations:

The right side of the operator can be a constant, variable, function, or formula, and #j and #k in the formula can also be constants. #j and #k in the expression can be assigned constant values. The variable on the left side of the operator can also be assigned an expression.

Function	Expression format	Remarks
Define or assign	#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 bit by bit according to binary numbers
Square root Absolute value Rounding Round up Round down Natural 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 Arc sine Cosine Inverse cosine	#i = SIN[#j] #i = ASIN[#j] #i = COS[#j] #i = ACOS[#j]	The angle unit is specified in degrees, such as: 90 °30’ is expressed as 90.5 degrees

Tangent Inverse tangent	#i = TAN[#j] #i = ATAN[#i]/ [#j]	
Convert from BCD to BIN Convert from BIN to BCD	#i = BIN[#j] #i = BCD[#j]	Used for PLC signal conversion

Note: This system supports mixed arithmetic operations including brackets, but the brackets in the expression must be square brackets "[]", and special functions such as trigonometric functions must be followed by square brackets "[]".

III. Transfer and loop

In the program, the GOTO statement and IF statement can be used to change the flow of control. There are three types of transfer and loop operations available.

1. GOTO statement (unconditional transfer).

2. Conditional control IF statement.
3. WHILE loop statement.

1) Unconditional transfer (GOTO statement)

Transfer to the program segment with sequence number n. When a sequence number other than 1~99999 is specified, an alarm is triggered. The sequence number can be specified by an expression.

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 is true, transfer to the program segment with sequence number n; if the specified conditional expression is not true, execute the next program segment in sequence.

Example:

If the value of variable #1 is greater than 10, transfer to the program segment with sequence number N2.		
If the condition is not true	IF [#1GT10] GOTO2; Program N2 G00 G91 X10.0;	If the condition is true

THEN format: IF[conditional expression]THEN<macro program statement>;

If the conditional expression is true, execute the statement after THEN, and only one statement can be executed.

Example: IF[#1 EQ #2] THEN #3=0;

If the value of #1 is equal to the value of #2, assign 0 to variable #3; if not equal, the sequence goes down without executing the assignment statement after THEN.

Conditional expression: The conditional expression must include the conditional operator. The conditional operator can be variables, constants or expressions on both sides. The conditional expression must be enclosed in brackets '[' ']'.

Conditional operator: This system can use the conditional operators listed in the following table.

Conditional operator	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 (≤)

Typical program: The following program calculates the sum of integers 1~10.
O9500

#1=0; … …The sum is initialized to 0

#2=1; … …The initial value of the addend is 1

N1 IF[#2 GT 10]GOTO2; … …When the addend is greater than 10, transfer to N2

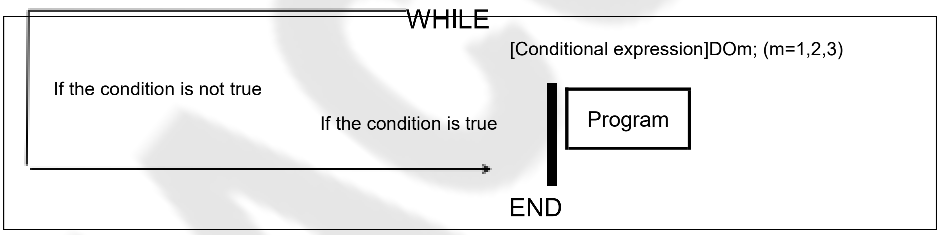
#1= #1+#2; … …Calculate the sum of two numbers
#2= #2+1; … …Add 1 to the addend

GOTO1; … …Unconditionally jump to the program segment N1
N2 M30; … …Program ends

3) Loop (WHILE statement)

Specify a conditional expression after WHILE. When the specified condition is met, execute the program segment from DO to END; otherwise, jump to the program segment after END.

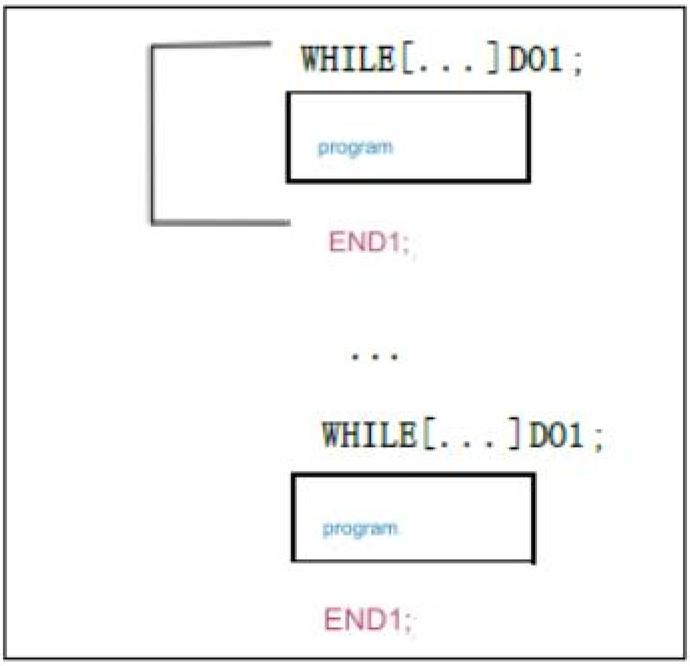
Example:



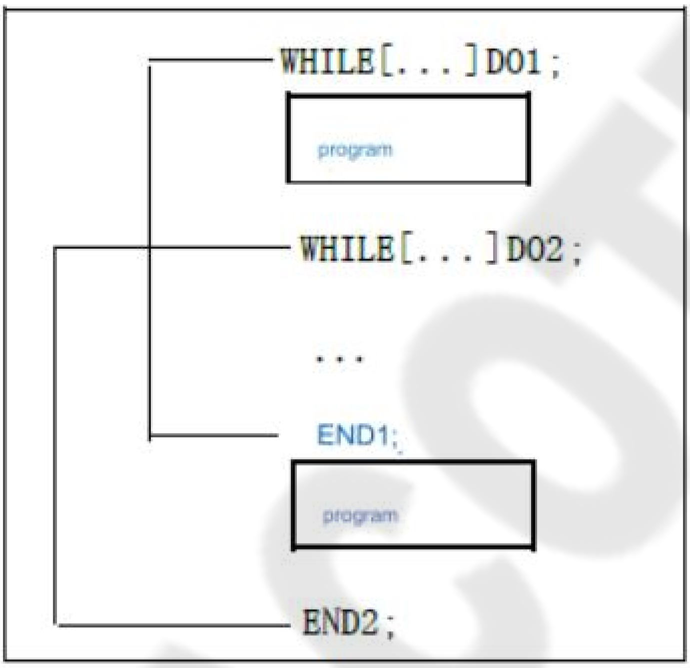
Description: When the specified condition is met, execute the program segment from DO to END; otherwise, execute the program segment after END instead. The label after DO and the label after END must be consistent, and the label value can be 1, 2 or 3. If a value other than 1, 2, 3 is used, an alarm will be issued.

Nesting: The labels (1~3) in the DO, END loop can be used as many times as needed. However, an alarm will be issued when there are cross-repeated loops in the program.

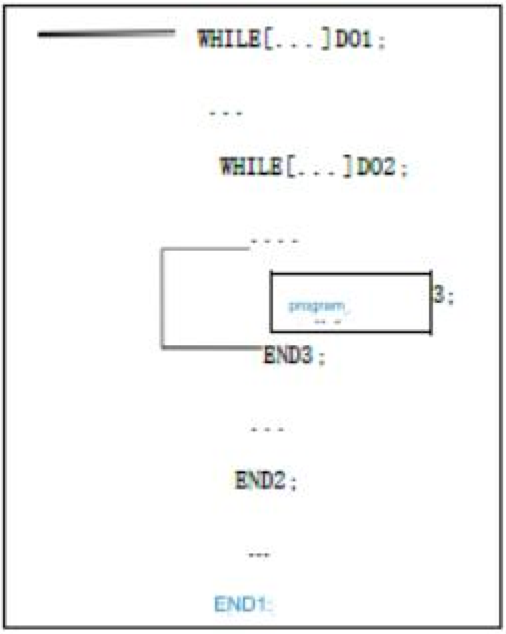
1. The label (1 ~3) can be used multiple times as required



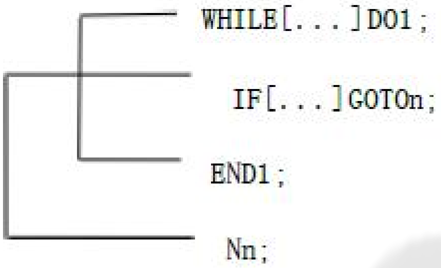
2. The range of DO cannot overlap



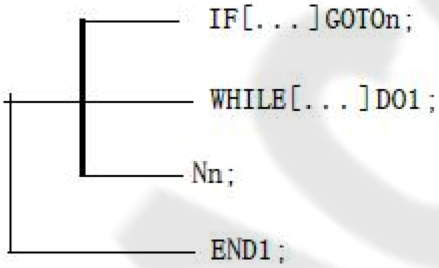
3. DO loops can be nested 3 levels



4. Control can be transferred to the outside of the loop



5. Transfer cannot enter the loop area



3.26 Metric-inch Conversion

3.26.1 Function Overview

The input and output units of the CNC system are divided into two types: metric unit, mm and imperial unit, inch. The parameters related to metric and inch in the system are as follows:

N0134: Input increment unit selection

0: Metric input (G21)

1: Inch input (G20)

This parameter completely corresponds to the function code G20/G21. That is: when G20/G21 is executed in the program, this parameter also changes; when this parameter is modified, the G20/G21 mode also changes accordingly.

N0162: Whether tool compensation value and wear value are automatically converted when switching between metric and inch input modes:

0: No (only move one decimal point)
1: Yes

N0135: Metric machine tool, inch machine tool selection (least output increment selection)

0: Metric machine tool output (0.001mm)

1: Inch machine tool output (0.0001inch)

3.26.2 Function Code G20/G21

Code format: G20; (inch input)

G21; (millimeter input)

This G code must be programmed at the beginning of the program and specified in a separate program segment.

3.26.3 Precaution

(1) N0134 input incremental unit change

① Change the unit system of the following values (i.e.: mm<>inch; mm/min<>inch/min) after the input incremental unit (inch/metric input) conversion:

- Feed rate specified by F code (mm/min<>inch/min), thread lead (mm<>inch)

- Position code (mm<>inch)

- Tool compensation value (mm<>inch)

- Handwheel scale unit (mm<>inch)

- Move distance in incremental feed (mm<>inch)

- Some parameters, including speed and displacement; when it is metric input (G21), the unit is 0.001mm (IS-B), when it is inch input (G20), the unit is 0.0001inch (IS-B). For example: the same parameter N0101 setting value is 100, when the input mode is metric G21, it means 100mm; when the input is inch G20, it means 100inch.

② After the input increment unit is changed (inch/metric input), the machine tool coordinates will be automatically converted:

(2) N0135 output code unit changes

When N0135=0, it means that the system's least code increment is output in metric (0.001mm)

When N0135=1, the system's least code increment is output in inch (0.0001inch)

When changing the output control parameters, the meaning of some parameters will change:

① Speed parameters: Metric machine tools: mm/min

Inch machine tools: 0.1 inch/min

For example: if the speed setting value is 3800, the metric machine tool represents 3800 mm/min, and the inch machine tool represents 380 inch/min.

② Position (length) parameters: Metric machine tool: 0.001 mm

Inch machine tool: 0.0001 inch

For example: if the setting value is 100, it means 0.1 mm for metric machine and 0.01 inch for inch machine.
It also includes all pitch error compensation parameters;

Note 1: When the least input increment and the least command increment are in different units, the maximum error is half of the least instruction increment. This error is not cumulative.
Note 2: In the above description, the current system increment is IS-B.

Chapter 4 Tool Nose Radius Compensation (G41, G42)

4.1 Application of Tool Nose Radius Compensation

4.1.1 Overview

Part machining programs are generally compiled according to part drawings based on a certain point of the tool (usually an imaginary tool nose, as shown in point A in Figure 4-1). However, due to process or other requirements, the tool nose of the turning tool in actual machining is often not an imaginary point, but an arc. During cutting, the position deviation between the actual cutting point and the ideal cutting point will cause overcutting or undercutting, affecting the accuracy of the part. Therefore, tool nose radius compensation is performed during processing to improve the accuracy of the part.

Figure 4-1 Tool

The method of offsetting the trajectory of the part shape by a tool nose radius is the B-type tool compensation method. This method is simple, but the motion trajectory of the next program segment is processed only after the execution of one program segment is completed, so overcutting and other phenomena may occur 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. When the C-type tool compensation method reads a program segment, it does not execute it immediately, but reads the next program segment again, and calculates the corresponding motion trajectory (transfer vector) according to the connection of the intersection of the two program segments. Since two program segments are read for preprocessing, the C-type tool compensation method can perform more accurate compensation on the contour. As shown in Figure 4-2.

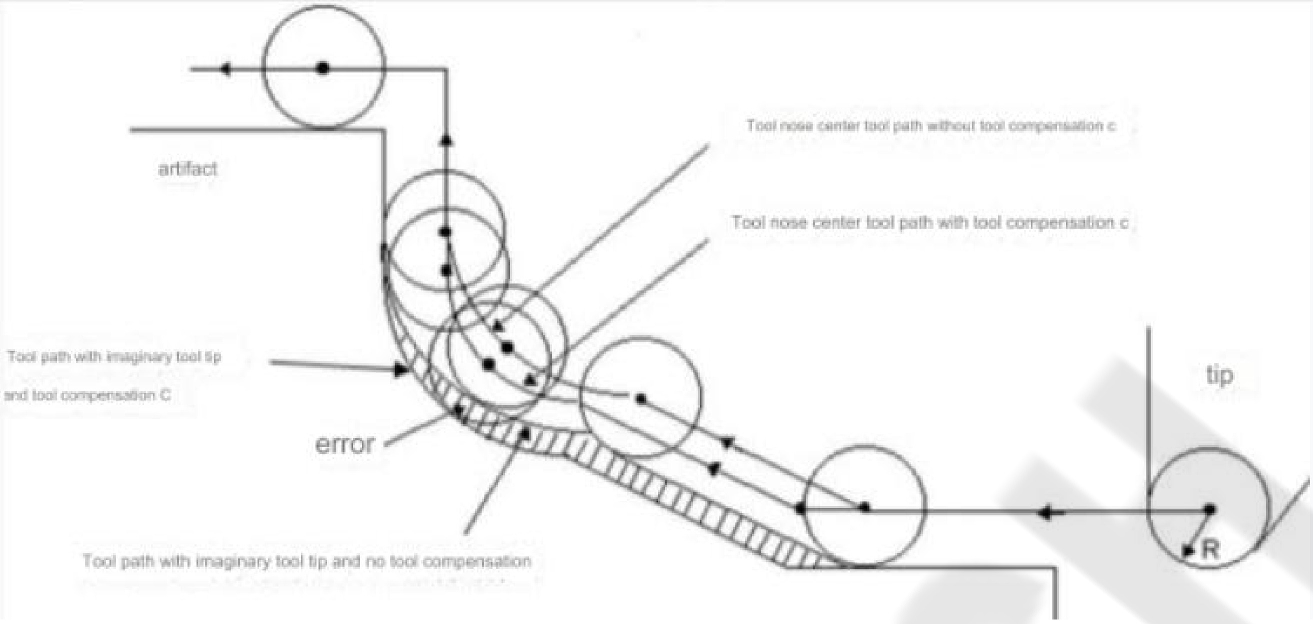
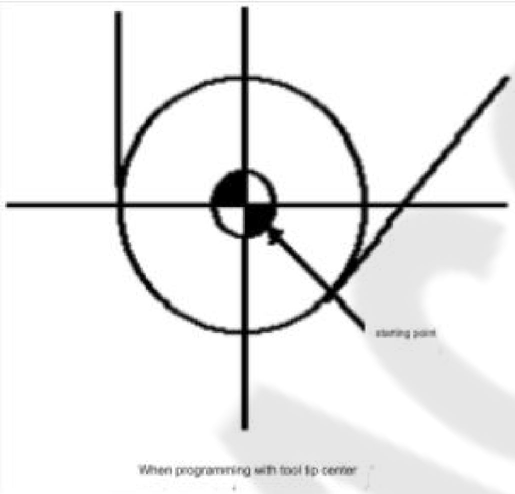


Figure 4-2

4.1.2 Direction of Imaginary Tool Nose

The imaginary tool nose is set because it is generally difficult to set the tool nose radius center at the starting position, as shown in Figure 4-3; it is easier to set the imaginary tool nose at the starting position, as shown in Figure 4-4; the tool nose radius can be ignored during programming. Figures 4-5 and 4-6 are tool trajectory diagrams for programming with tool nose center and imaginary tool nose, respectively, when using tool nose radius compensation and when not using tool nose radius compensation.



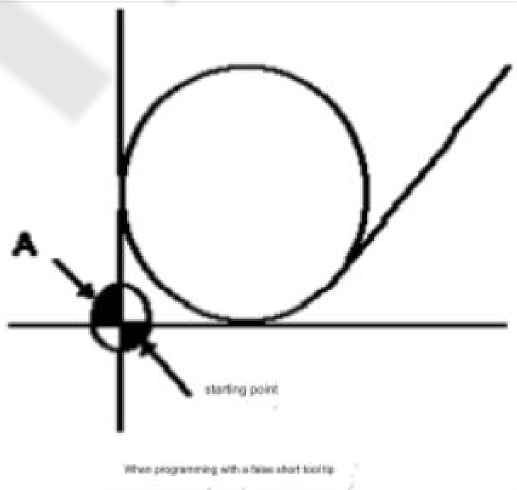


Figure 4-3

If tool nose radius compensation is not used, the tool nose center trajectory will be the same as the programmed trajectory

Figure 4-4

If tool nose radius compensation is used, precise cutting will be achieved

Figure 4-5 Tool trajectory when programming with tool nose center

Without tool nose radius compensation, the imaginary tool nose trajectory will be the same as the programmed trajectory

Using tool nose radius compensation, precise cutting will be achieved

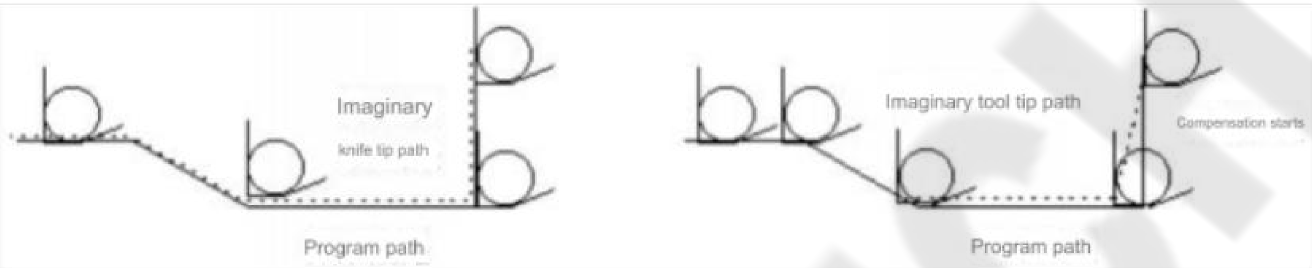


Figure 4-6

When programming with imaginary tool nose, the tool is assumed 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 caused by the cutting edge being not an ideal point but an arc can be eliminated by using the tool nose arc radius compensation function. In actual machining, the imaginary tool nose point and the tool nose arc center point have different positional relationships, so the tool nose direction of the imaginary tool nose (i.e., the tool nose position of the tool setting point) must be correctly established.

From the tool nose center to the imaginary tool nose, the imaginary tool nose number is determined by the direction of the tool in cutting. There are 10 (T0~T9) settings for the imaginary tool nose, which express the positional relationship in 9 directions. It should be noted that even if the same tool nose direction number is expressed in different coordinate systems (the rear tool holder coordinate system and the front tool holder coordinate system), the tool nose direction is different, as shown in the figure below. The figure illustrates the relationship between the tool nose and the start point, with the end point of the arrow being the imaginary tool nose; the situation of the rear tool holder coordinate system T1~T8 is shown in Figure 4-7; the situation of the front tool holder coordinate system T1~T8 is shown in Figure 4-8. T0 and T9 are the situations when the tool nose center is consistent with the start point, as shown in Figure 4-9.

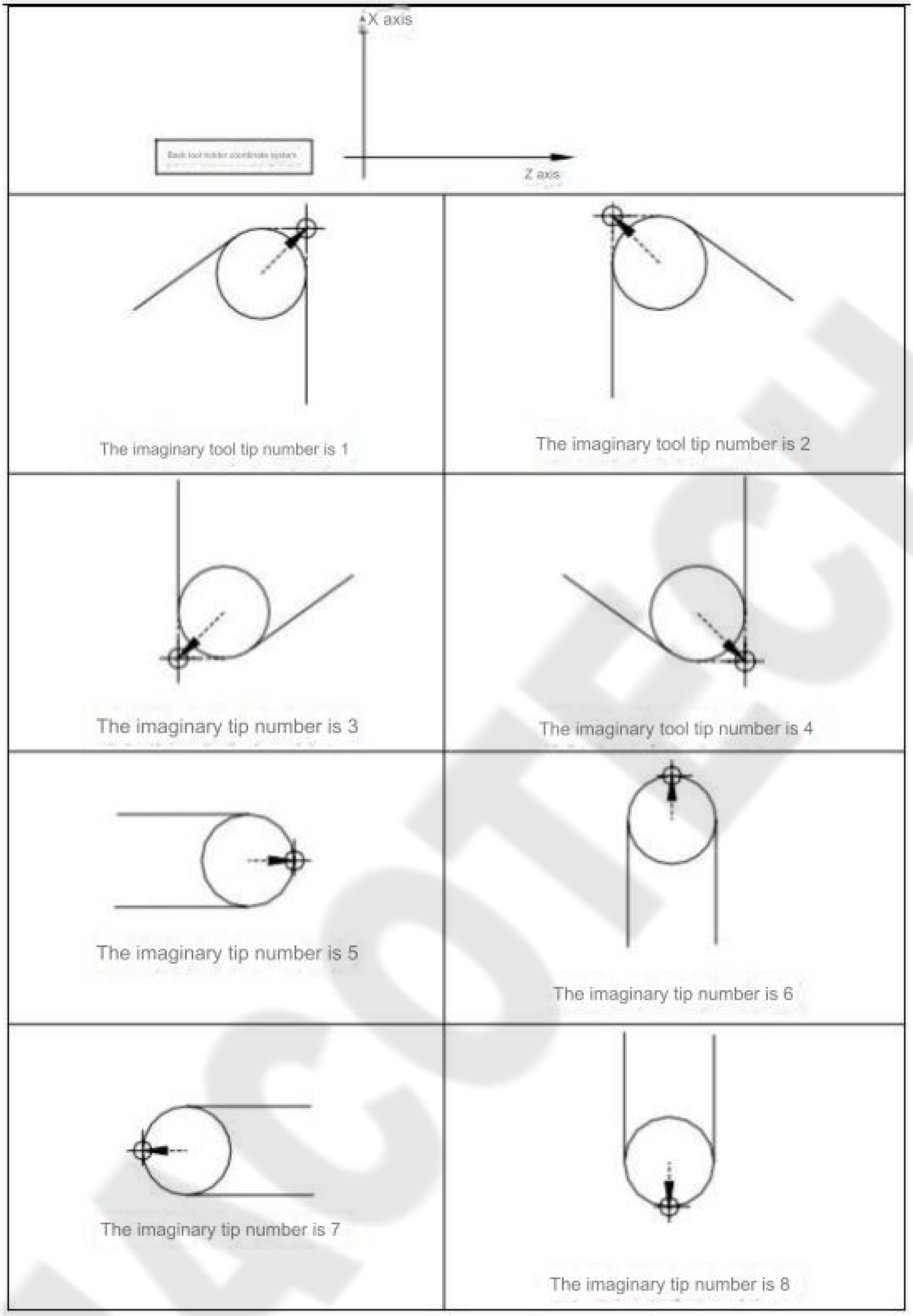


Figure 4-7 Imaginary tool nose number in the rear tool holder coordinate system

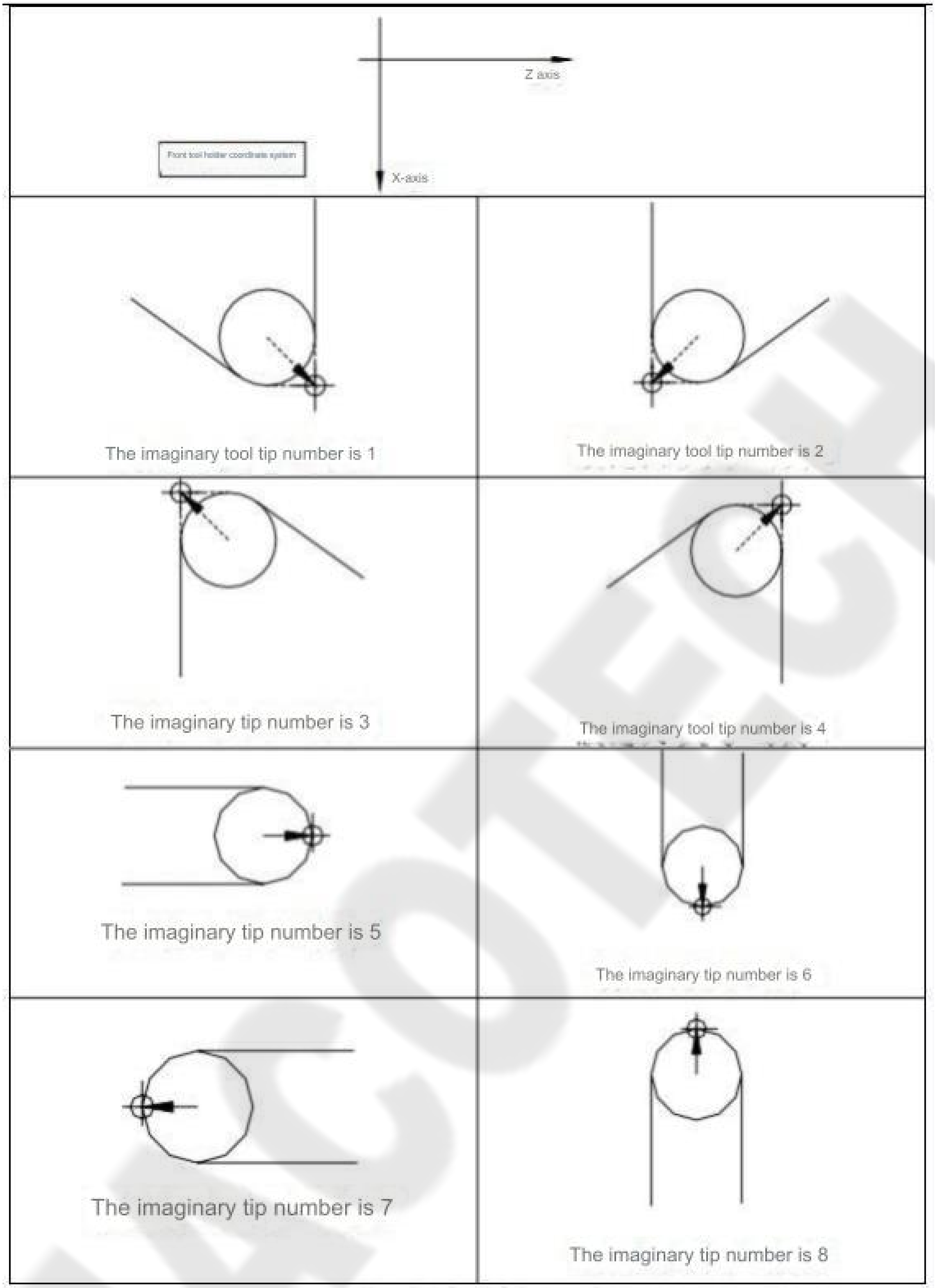


Figure 4-8 Imaginary tool nose number in the front tool holder coordinate system

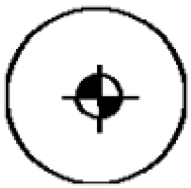


Figure 4-9 Tool nose center is consistent with the start point

4.1.3 Compensation Value Setting

The imaginary tool nose number and tool nose radius value of each tool must be set in advance before applying C tool compensation. The tool nose radius compensation value is set on the offset page (see Table 4-1), R is the tool nose radius compensation value, and T is the imaginary tool nose number.

Table 4-1 CNC tool nose radius compensation value display page

No.	X	Z	R	T
000	0.000	0.000	0.000	0
001	0.020	0.030	0.020	2
002	1.020	20.123	0.180	3
…	…	…	…	…
032	0.050	0.038	0.300	6

When performing the tool setting operation, pay special attention to the fact that when the Tn (n=0~9) imaginary tool nose is selected, the tool setting point must also be the Tn (n=0~9) imaginary tool nose point. As shown in Figure 4-10, different tool setting methods are used when selecting the tool nose points T0 and T3 in the rear tool holder coordinate system. Taking the tool holder center as the standard point, for the same tool, the offset value from the standard point to the tool nose radius center (when the imaginary tool nose is T0) and the offset value from the standard point to the imaginary tool nose (when the imaginary tool nose is T3) are different. It is much easier to measure the distance from the standard point to the imaginary tool nose than to measure the distance from the standard point to the tool nose radius center. Therefore, the tool offset value is usually set based on the distance from the standard point to the imaginary tool nose (that is, the tool nose direction T3 is usually selected).

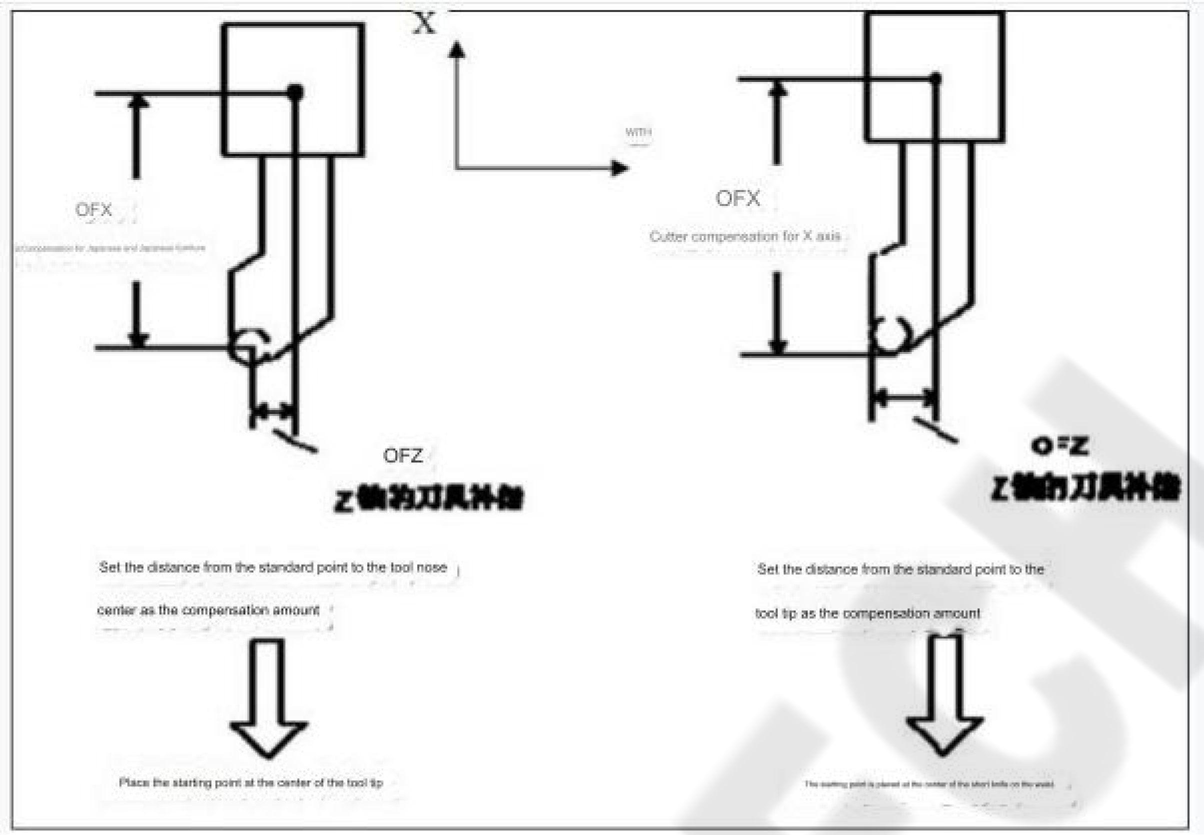


Figure 4-10 Tool offset value with tool holder center as reference point

4.1.4 Code Format

$\left\{\begin{matrix}G40\\G41\\G42\end{matrix}\right\} \left\{G00/G01\right\}$ X_ Z_ T_;

Code	Function description	Remarks
G40	Cancel tool nose radius compensation	See Figure 4-11 and Figure 4-12 for details
G41	G41 specifies left tool compensation in the rear tool holder coordinate system and right tool compensation in the front tool holder coordinate system	
G42	G42 specifies right tool compensation in the rear tool holder coordinate system and left tool compensation in the front tool holder coordinate system	

4.1.5 Compensation Direction

When applying tool nose radius compensation, the direction of compensation must be determined based on the relative position of the tool nose and the workpiece, as shown in Figures 4-11 and 4-12.

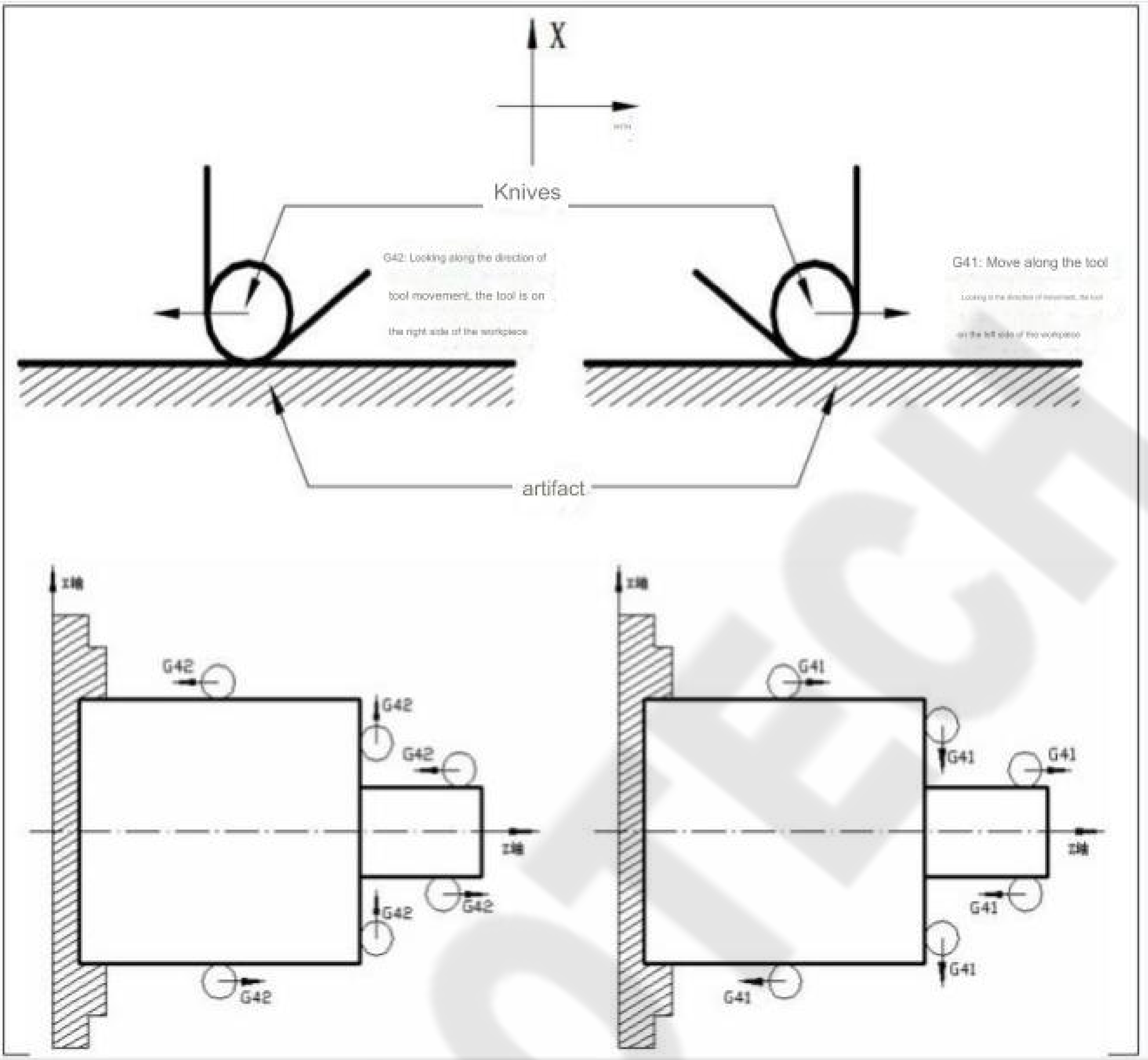


Figure 4-11 Compensation direction of rear tool holder coordinate system

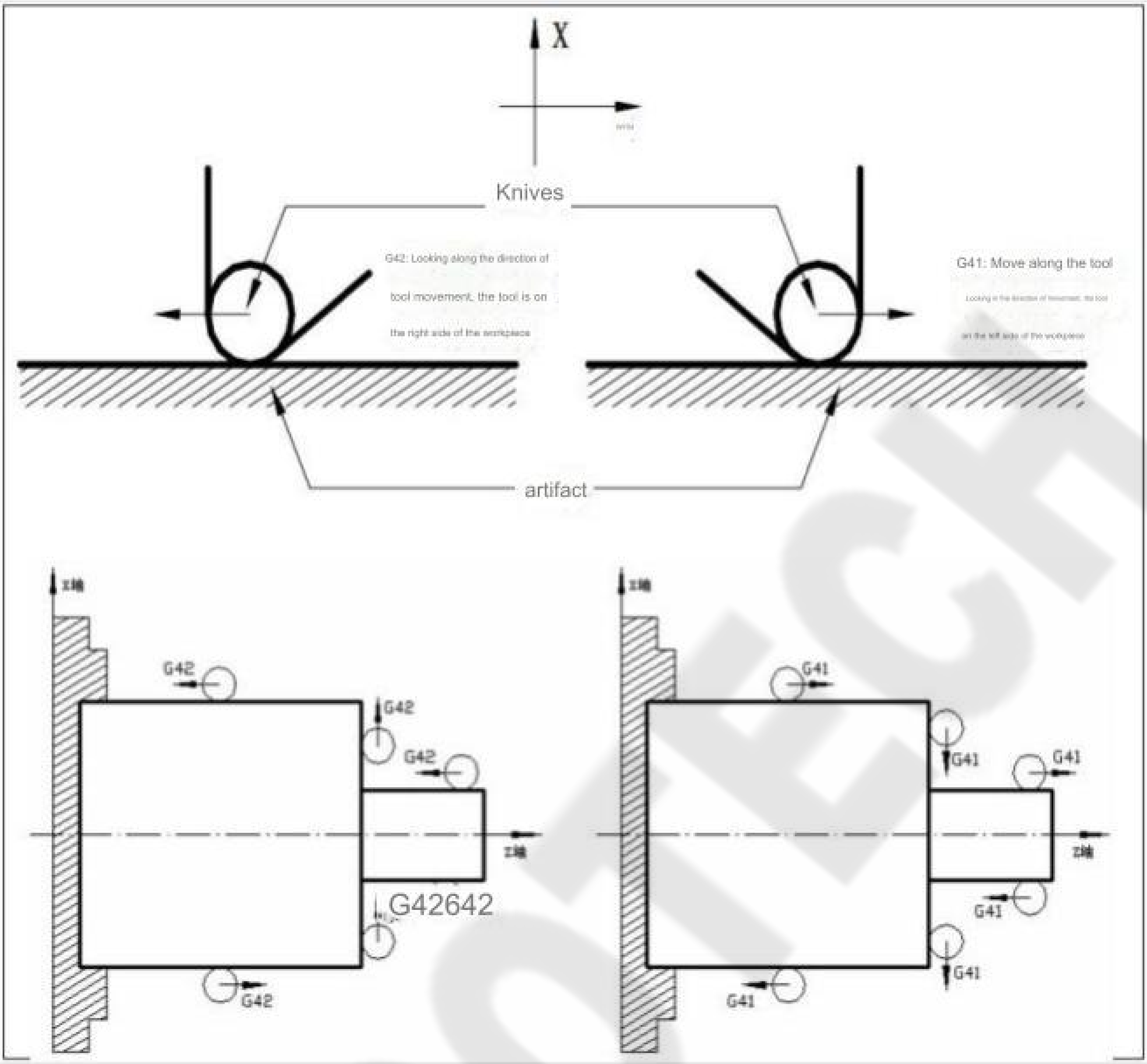


Figure 4-12 Compensation direction of front tool holder coordinate system

4.1.6 Notes

● In the initial state, CNC is in tool nose radius compensation cancellation mode. When executing G41 or G42 code, CNC starts to establish tool nose radius compensation offset mode. At the beginning of compensation, CNC reads 2 program segments in advance. When executing a program segment, the next program segment is stored in the tool nose radius compensation buffer memory. In single-segment operation, two program segments are read in, and the end point of the first program segment is executed and stops. In continuous execution, two program segments are read in advance, so the program segment being executed in CNC and the two subsequent program segments.

● In tool nose radius compensation, when processing two or more program segments without movement codes (such as auxiliary functions, pauses, etc.), the tool nose center will move to the end point of the previous program segment and be perpendicular to the position of the program path of the previous program segment.

● In the input mode (MDI), tool compensation C establishment and tool compensation C cancellation cannot be executed.

● The tool tip radius R-value cannot be negative, otherwise the running trajectory will be wrong.

● The establishment and cancellation of tool tip radius compensation can only be done with G00 or G01 code, not arc code (G02 or G03). If specified, an alarm will be generated.

* After pressing the RESET key or executing M30, the CNC will cancel the tool compensation C compensation mode.

* G40 must be specified before the program ends to cancel the offset mode. Otherwise, the tool trajectory will deviate from a tool tip radius value when it is executed again.

* When using tool tip radius compensation in the main program and subroutine, the CNC must be in compensation cancel mode before calling the subroutine (that is, before executing M98), and establish tool compensation C again in the subroutine.

* The G71, G72, G73, G74, G75, and G76 codes do not execute tool nose radius compensation and temporarily cancel the compensation mode.

*G90 and G94 codes execute tool nose radius compensation. Whether it is G41 or G42, they are all offset by a tool nose radius (according to the imaginary tool nose number 0) for cutting.

4.1.7 Application Example

Process the part shown in Figure 4-13 in the front tool holder coordinate system. The tool number used is T0101, the tool nose radius R = 2, and the imaginary tool nose number T = 3.

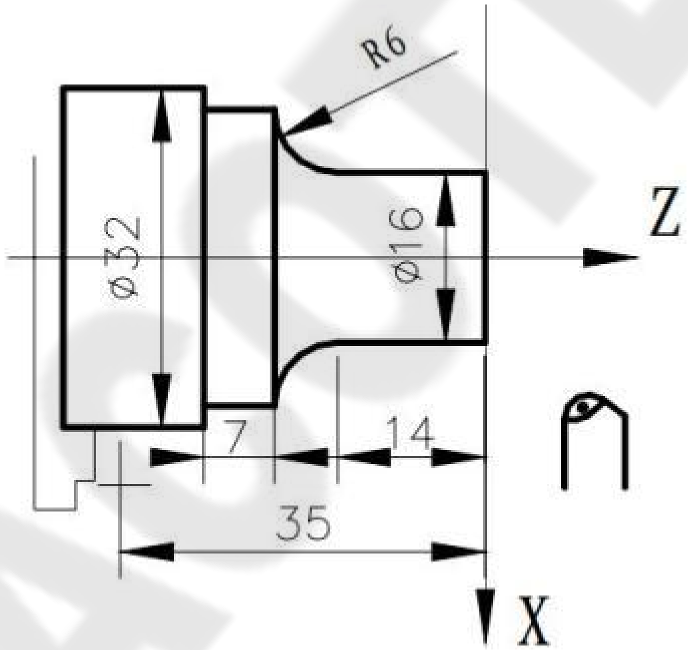


Figure 4-13

When setting the tool in the offset cancel mode, the Z axis is usually offset by a tool nose radius value after setting. The direction of the offset is related to the imaginary tool nose direction and the setting point, otherwise, the tool nose radius value will be overcut when starting the tool.

In the tool offset setting page, the tool nose radius R and the imaginary tool nose direction are set as follows:

Table 4-2

No.	X	Z	R	T
001			2.000	3
002	…	…	…	…
…	…	…	…	…
007	…	…	…	…
008	…	…	…	…

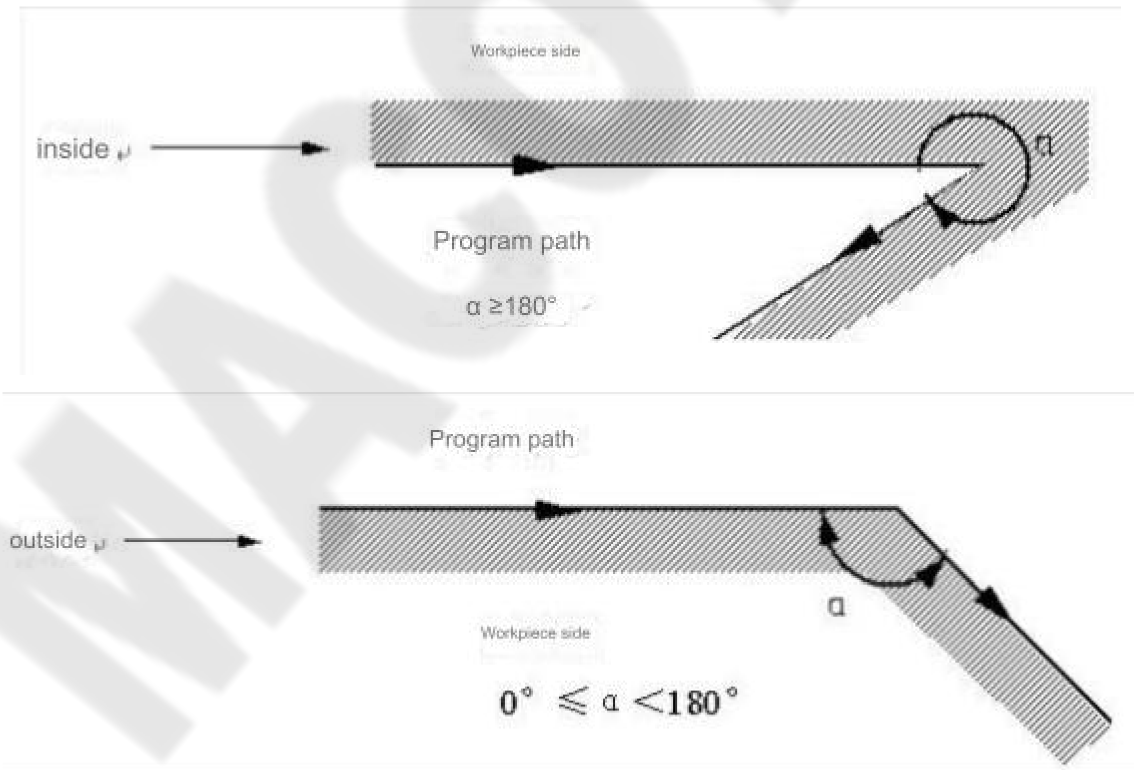
Example:

```
G00 X100 Z50 M3 T0101 S600;   (Positioning, spindle start, tool change and tool compensation)
G42 G00 X0 Z3;                (Establish tool nose radius compensation)
G01 Z0 F300;                  (Start cutting)
X16;
Z-14 F200;
G02 X28 W-6 R6;
G01 W-7;
X32;
Z-35;
G40 G00 X90 Z40;              (Cancel tool nose radius compensation)
G00 X100 Z50 T0100;
M30;
```

4.2 Description of Tool Nose Radius Compensation Offset Trajectory

4.2.1 Concepts of Inside and Outside

The two terms ‘inside’ and ‘outside’ will be used in the following description. When the angle between the intersection of two moving program segments is greater than or equal to 180°, it is called ‘inside’; when the angle between the intersection of two moving program segments is between 0 and 180°, it is called ‘outside’.



4.2.2 Tool Movement at Tool Start

To achieve tool nose radius compensation, there are three steps: establish tool compensation, execute tool compensation, and cancel tool compensation.

The tool movement from the offset cancellation mode to the start of the execution process of establishing the G41 or G42 code is called tool compensation establishment (or tool start). Note: In the following figure, S, L, and C, unless otherwise specified, have the following meanings:

S - single-segment stop point; L - straight line; C - circular arc.

There are two types of actions for tool path when compensation starts or cancels: Type A and Type B, which are selected by parameter N0165.

(a) Move along the inside of the corner (α≥ 180°)

1) Straight line-straight line

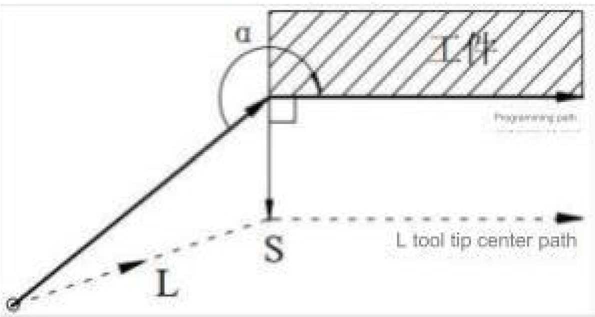


Figure 4-14a Straight line - straight line (starting from the inside)

2) Straight line – circle arc

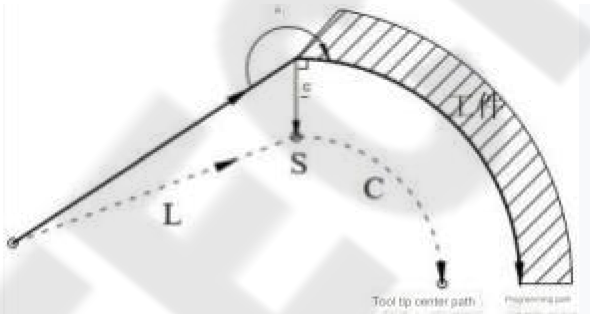


Figure 4-14b Straight line-circular arc (starting from the inside)

(b) Move along the outside of the obtuse corner (180°>α≥90°)

1) Straight line-straight line

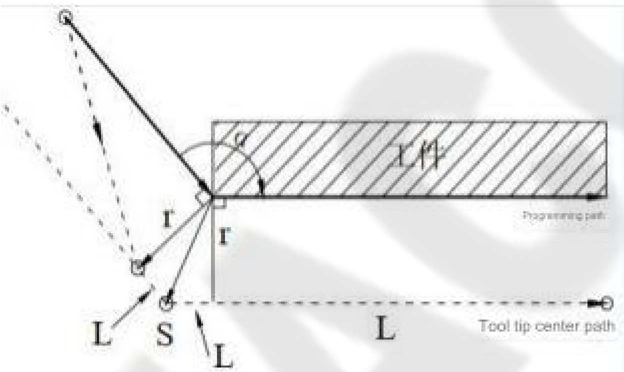


Figure 4-15a Straight line - straight line (starting from the outside)

2) Straight line-circular arc

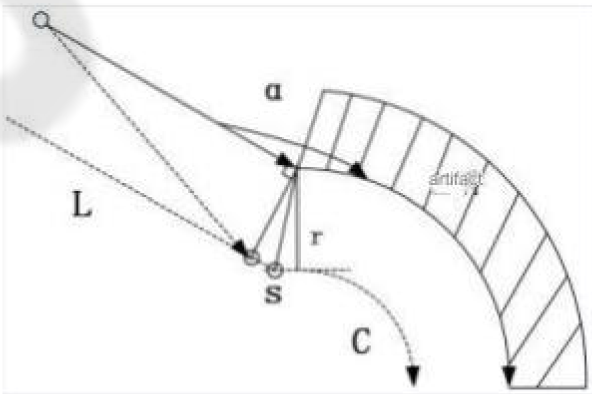


Figure 4-15b Straight line - circular arc (starting from the outside)

(c) Move along the outside of the acute corner (α ＜ 90°)

1) Straight line - straight line

2) Straight line - circular arc

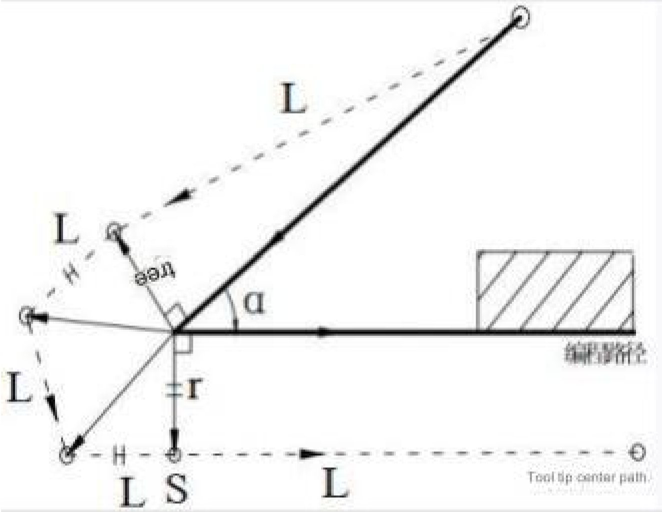


Figure 4-16a Straight line - straight line (starting from the outside)

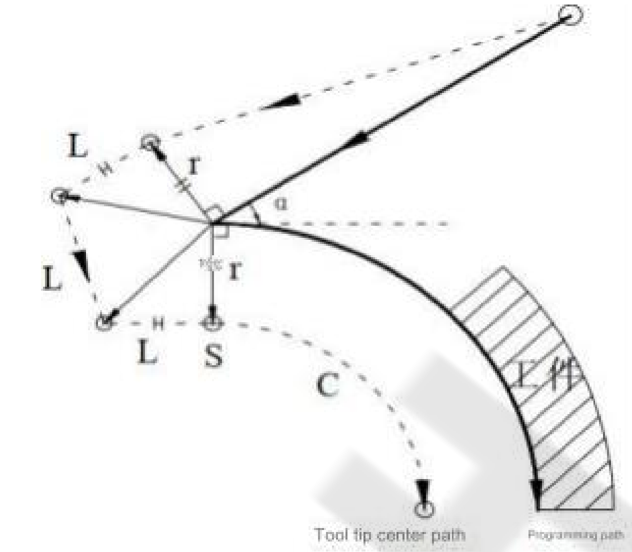


Figure 4-16b Straight line - circular arc (starting from the outside)

(d) Move along the outside of the acute corner less than 1 degree, straight line→straight line. (α ≦ 1 °)

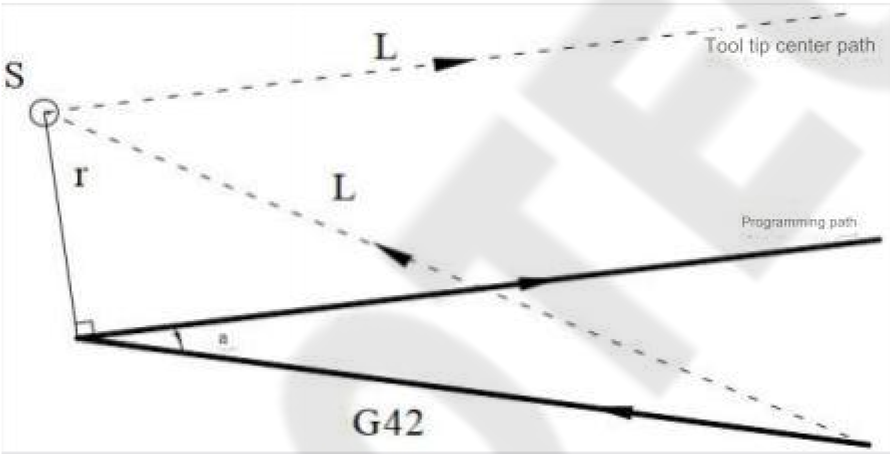


Figure 4-17 Straight line-straight line (angle less than 1 degree, starting from the outside)

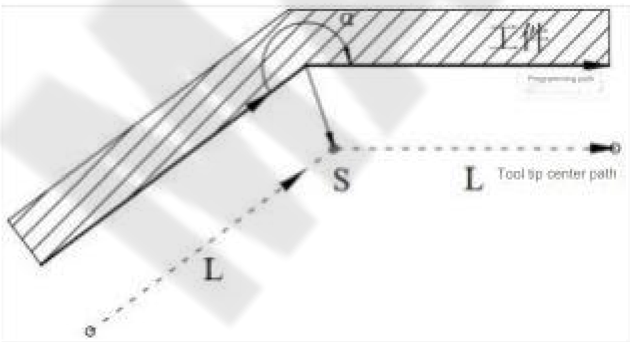
4.2.3 Tool Movement in Offset Mode

After establishing tool nose radius compensation and before canceling tool nose radius compensation, it is called offset mode.

* Offset trajectory of compensation direction isn't changed in compensation mode

(a) Moving along the inside of a corner (α ≥ 180°)

1) Straight line-straight line



2) Straight line-circular arc

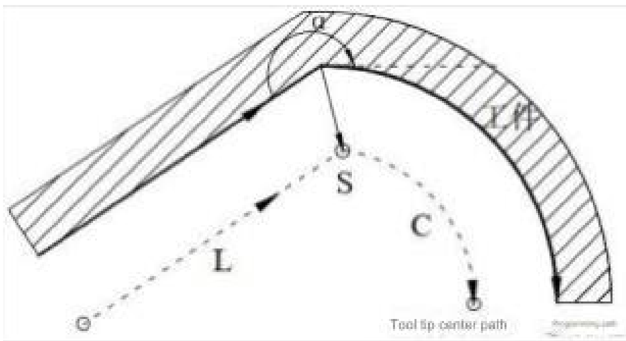


Figure 4-18a Straight line - straight line (moving inside)

Figure 4-18b Straight line - circular arc (moving inside)

3) Circular arc - straight line

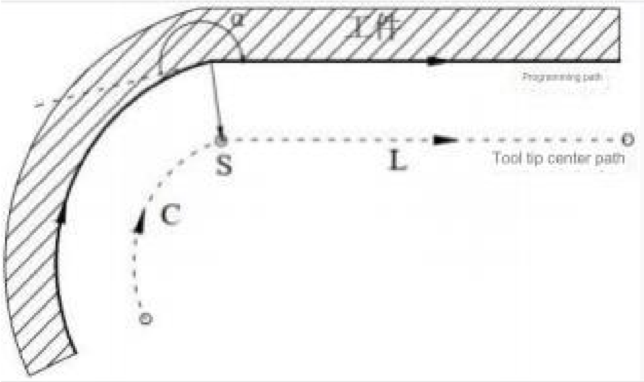


Figure 4-18c Circular arc - straight line (moving inside)

4) Circular arc - circular arc

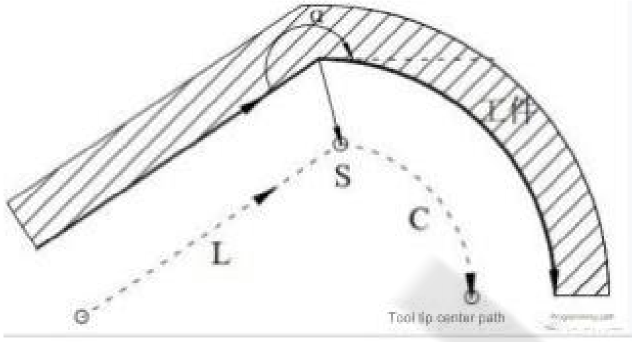


Figure 4-18d Circular arc - circular arc (moving inside)

(b) Moving along the outside of an obtuse corner (180° > α ≥ 90°)

1) Straight line-straight line

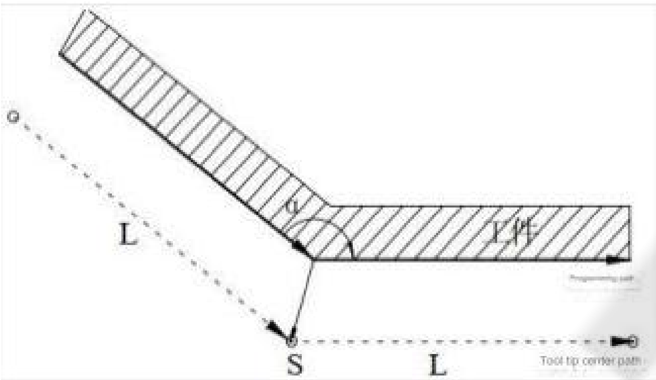


Figure 4-19a Straight line - straight line (obtuse angle, moving outside)

2) Straight line-circular arc

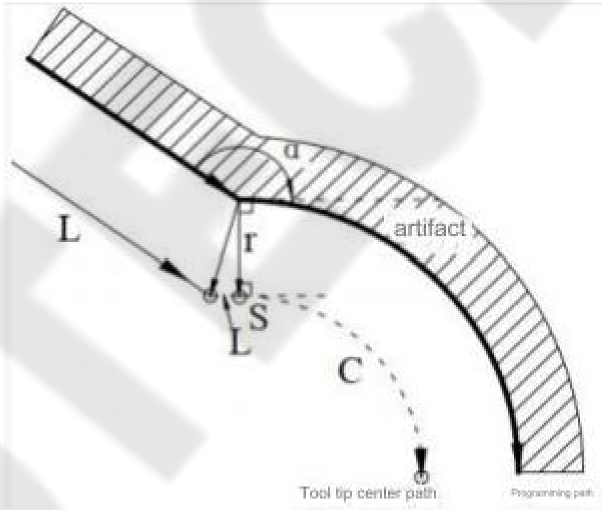


Figure 4-19b Straight line - circular arc (obtuse angle, moving outside)

3) Circular arc - straight line

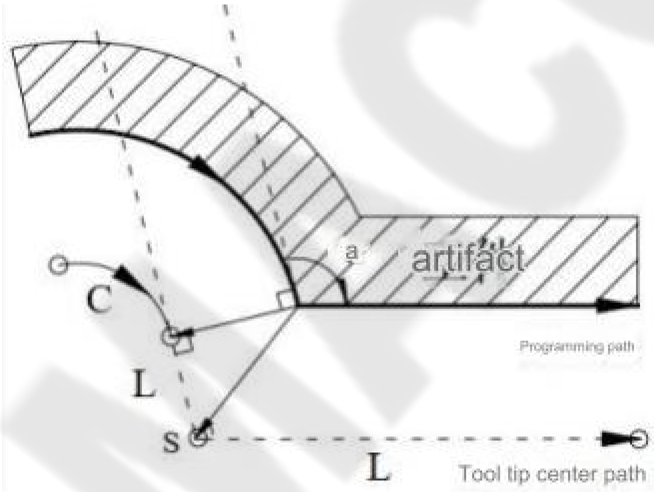


Figure 4-19c Circular arc - straight line (obtuse angle, moving outside)

4) Circular arc - circular arc

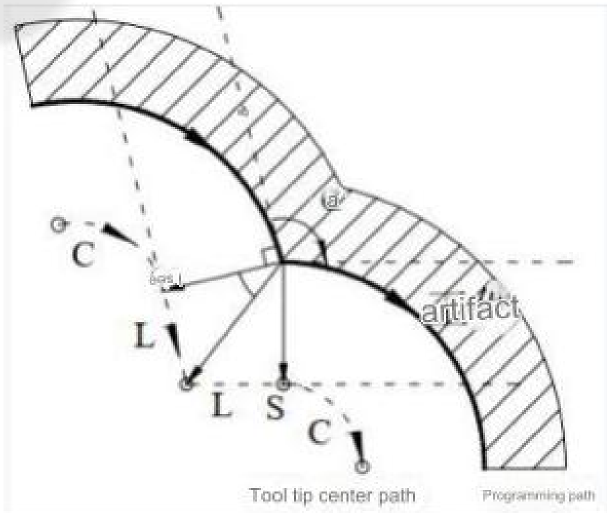


Figure 4-19d Circular arc - circular arc (obtuse angle, moving outside)

(c) Move along the outside of the acute corner (α ＜ 90°)

1) Straight line-straight line

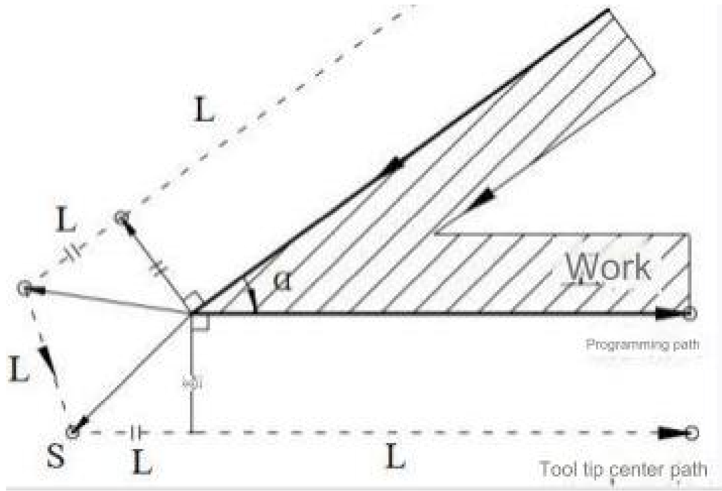


Figure 4-20a Straight line - straight line (acute angle, moving outside)

2) Straight line-circular arc

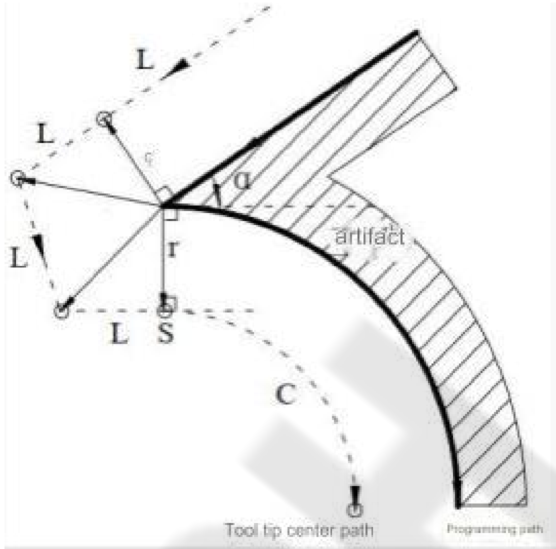


Figure 4-20b Straight line - circular arc (acute angle, moving outside)

3) Circular arc - straight line

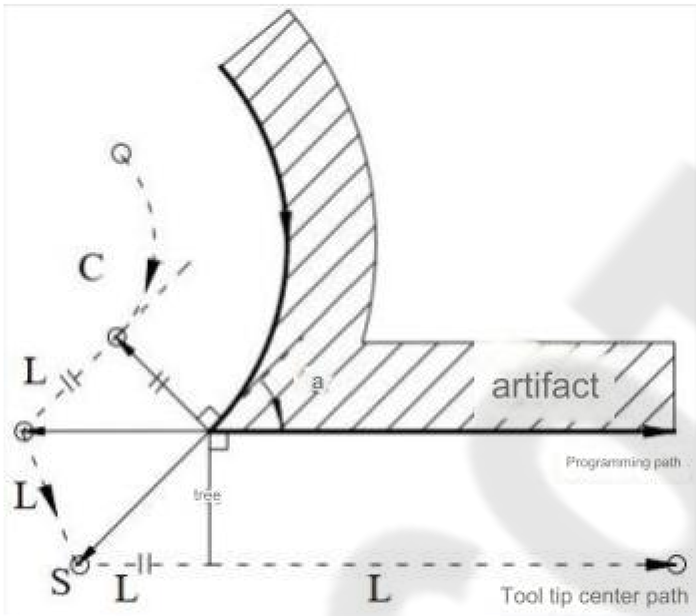


Figure 4-20c Circular arc - straight line (acute angle, moving outside)

4) Circular arc - circular arc

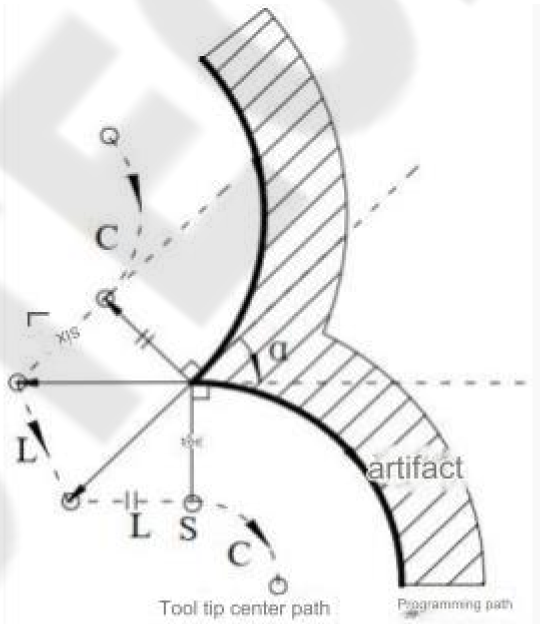


Figure 4-20d Circular arc - circular arc (acute angle, moving outside)

5) Inner processing less than 1 degree and compensation vector enlargement

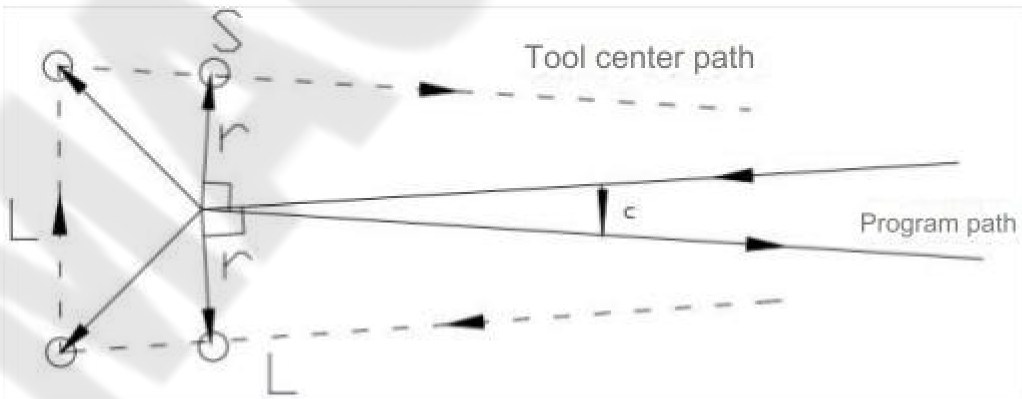
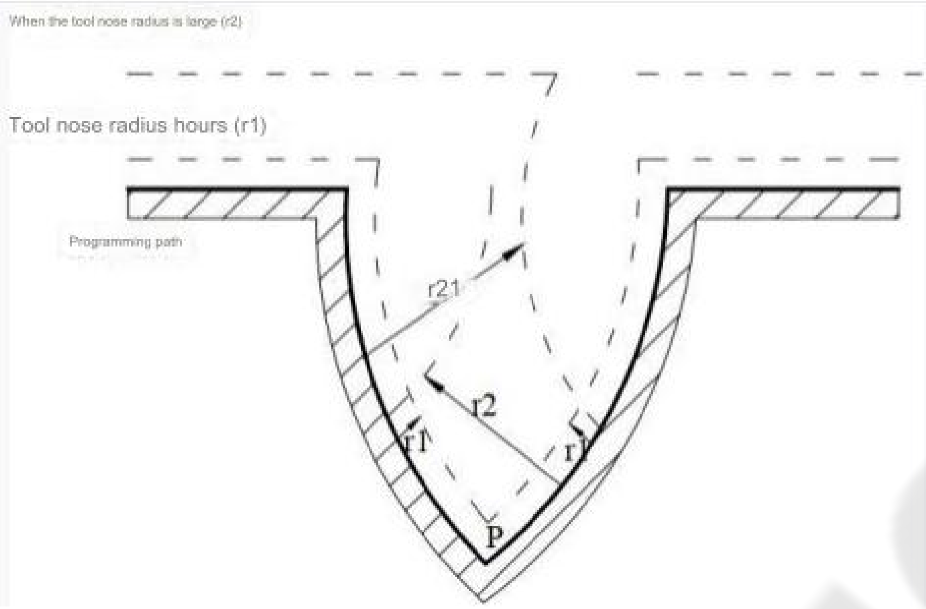


Figure 4-20e Straight line - straight line (angle less than 1 degree, moving inside)

(d) Special cases

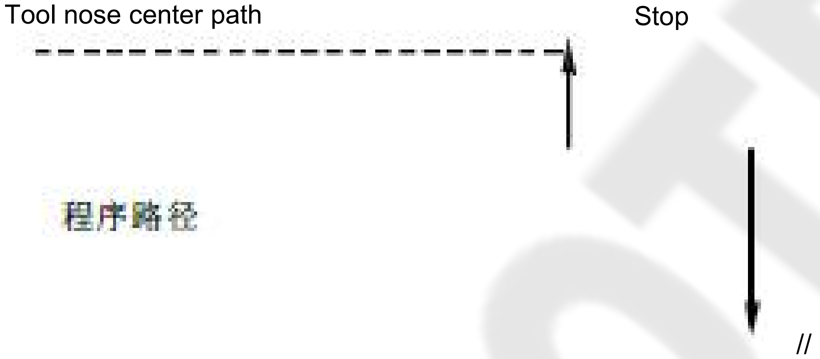
1) When there is no intersection



When the tool nose radius is small, the circular arc and the circular arc compensation path have an intersection P. When the tool nose radius is large, the circular arc and the circular arc compensation path may have no intersection, and the system will prompt an alarm

Figure 4-21 Special case - no intersection after offset

2) Circular arc center is consistent with the start point or end point



Alarm generated: the circular arc start point or end point is the same as the circle center; and stop at the end point of the previous program segment (G41)

N5 G01 W20;

N6 GO2 W10 I K0;
N7 G03 U-10 I-10;

Figure 4-22 Circular arc center is consistent with the start point or end point

* Offset trajectory of compensation direction is changed in compensation mode

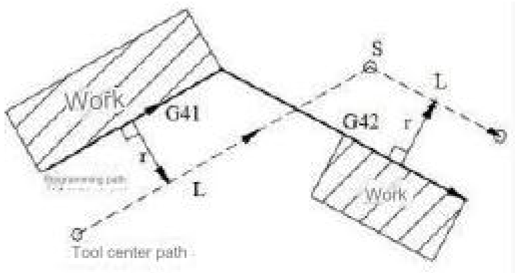
Tool nose radius compensation G41 and G42 codes determine the compensation direction, and the sign of the compensation amount is as follows

Table 4-3

G code Compensation amount sign	+	-
G41	Left compensation	Right compensation
G42	Right compensation	Left compensation

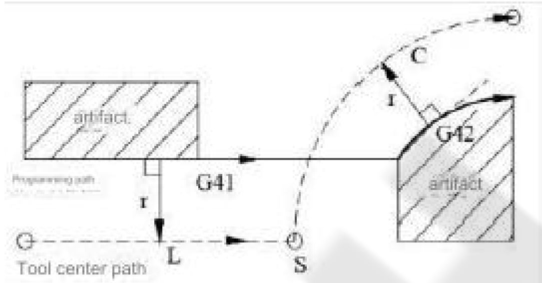
In special cases, the compensation direction can be changed in compensation mode. However, it cannot be changed in the starting segment and the following segments. When the compensation direction is changed, there is no concept of inside and outside for all conditions. The following compensation amounts are assumed to be positive.

1) Straight line - straight line



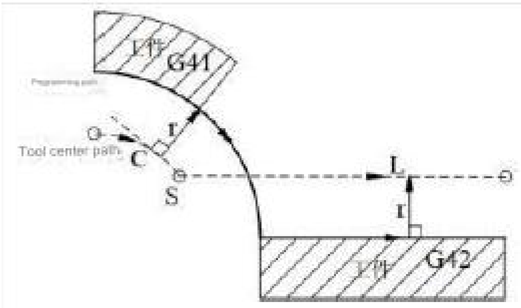
/

2) Straight line - circular arc



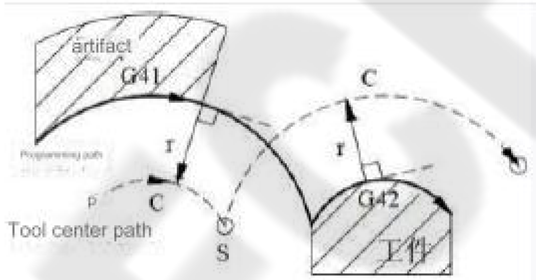
/

3) Circular arc - straight line



/

4) Circular arc - circular arc

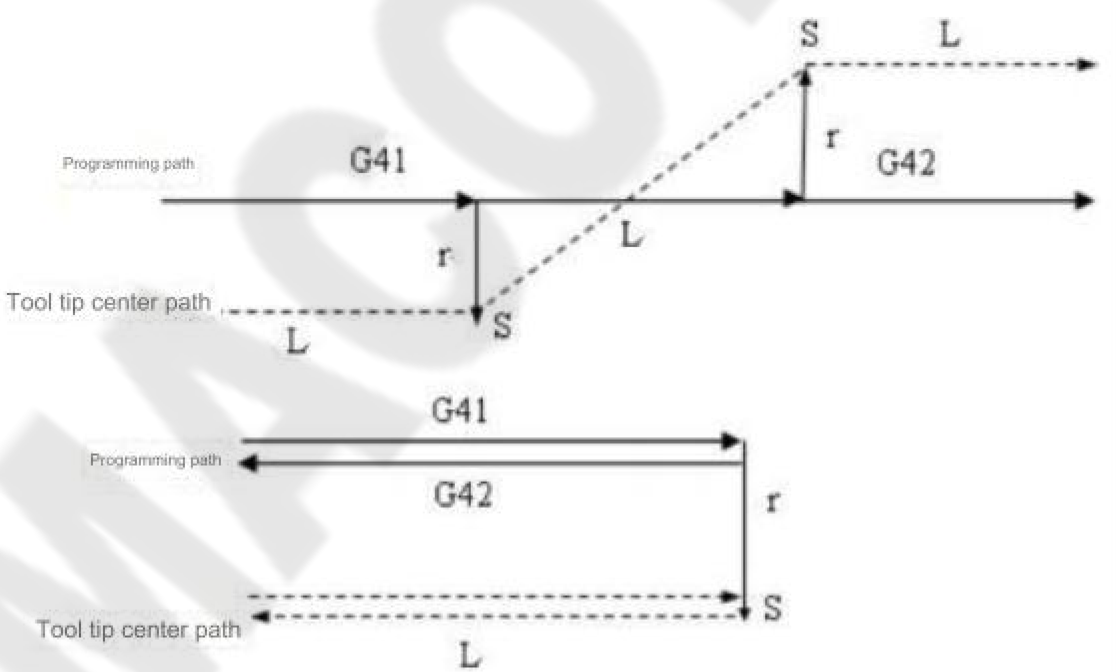


/

5) If compensation is executed normally, but there is no intersection

When G41 and G42 are used to change the offset direction from block A to block B, if the intersection of the offset path is not required, a vector perpendicular to block B is made at the starting point of block B.

i) Straight line - straight line



/

Figure 4-27a Straight line - straight line, no intersection (change compensation direction)

ii) Straight line - circular arc

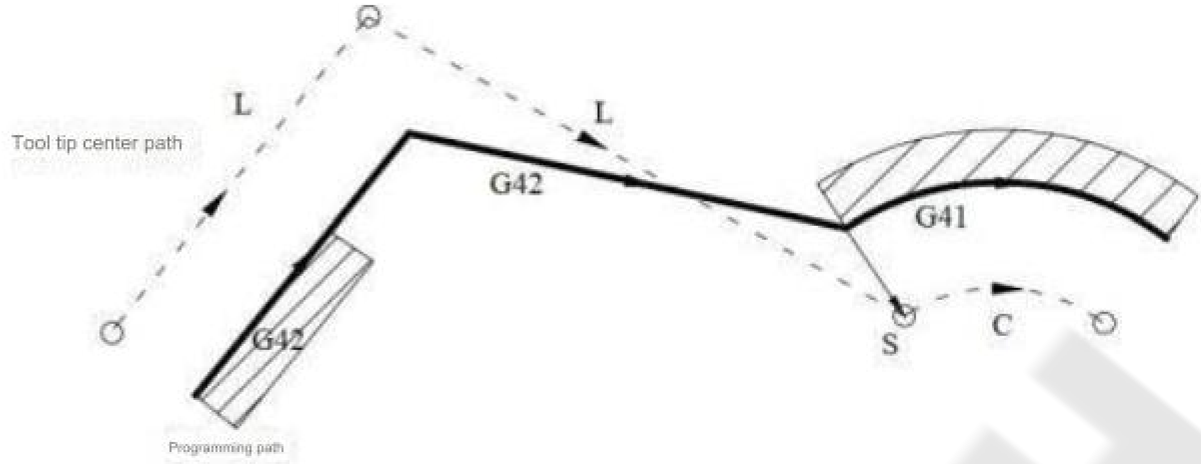


Figure 4-27b Straight line - circular arc, no intersection (change compensation direction)

iii) Circular arc - circular arc

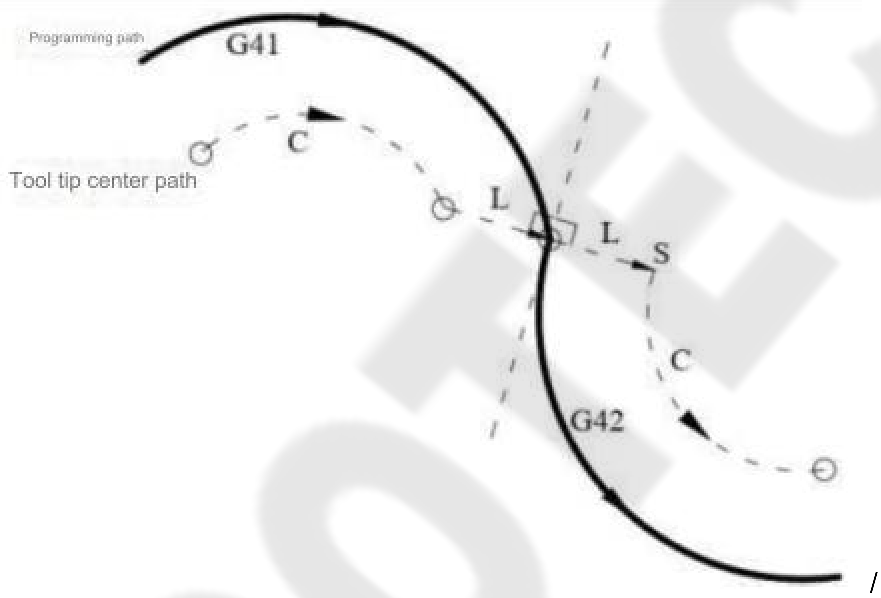


Figure 4-27c Circular arc - circular arc, no intersection (change compensation direction)

4.2.4 Tool Movement in Offset Cancellation Mode

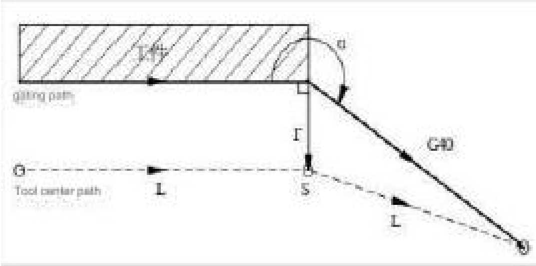
In compensation mode, when a block meets any of the following conditions, the CNC enters compensation cancellation mode, and the action of this block is called compensation cancellation.

1. G40 code is used in the program;
2. M30 code is executed.

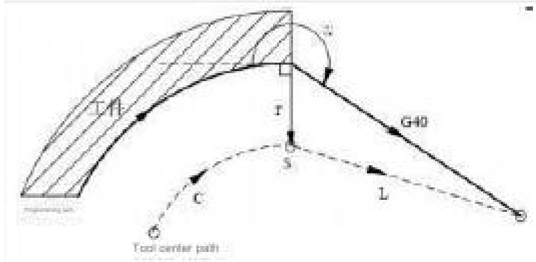
When C tool compensation is canceled, circular arc codes (G02 and G03) cannot be used. If a circular arc is instructed, an alarm will be generated and the operation will stop. In compensation cancellation mode, it controls execution of the current segment and the segment in the tool nose radius compensation buffer register. At this time, if the single segment switch is on, it stops after executing one segment. The next segment is executed without reading the next segment when the start button is pressed again.

(a) Move along the inside of the corner (α ≥ 180°)

1) Straight line-straight line

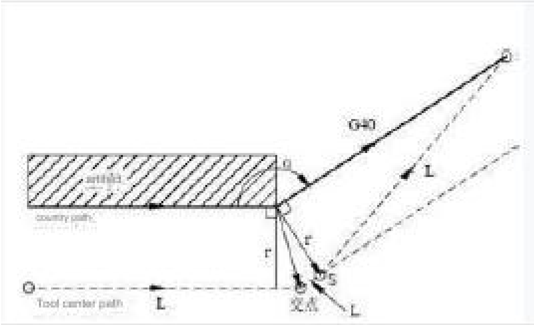


2) Circular arc→ straight line

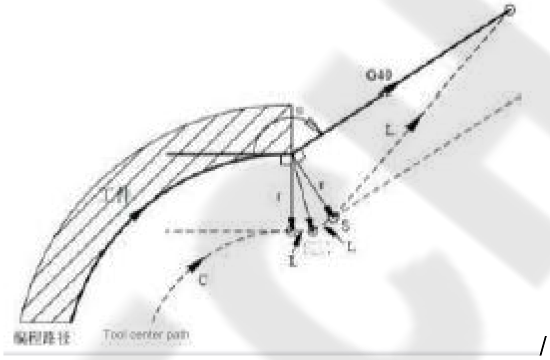


(b) Moving along the outside of an obtuse corner (180° > α ≥ 90°)

1) Straight line → straight line

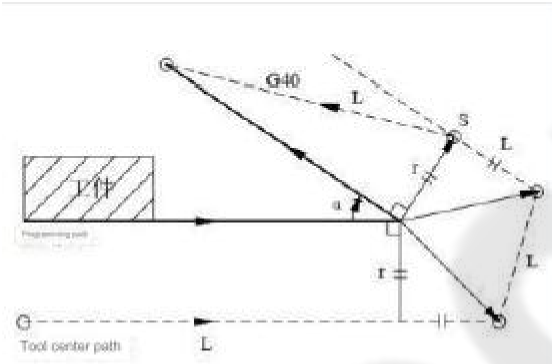


2) Circular arc→ straight line

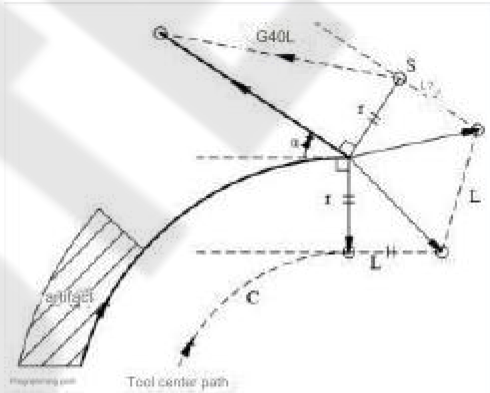


(c) Move along the outside of the acute corner (α ＜ 90°)

1) Straight line → straight line



2) Circular arc→ straight line



(d) Move along the outside of the acute corner less than 1 degree; Straight line → straight line. (α ＜ 1 °)

Figure 4-31 Straight line--straight line (angle less than 1 degree, outside, offset canceled)

4.2.5 Tool Interference Check

Tool transition cutting is called "interference". Interference can pre-check tool transition cutting. Interference check will be performed even if transition cutting does not occur. However, not all tool interference can be checked.

(1) Basic conditions for interference

1) The direction of the tool path is different from the direction of the program path. (The angle between the paths is 90~270 degrees).

2) When processing an arc, in addition to the above conditions, the angle between the start and end of the tool center path is very different from the angle between the start and end of the program path (more than 180 degrees).

Example: Linear processing

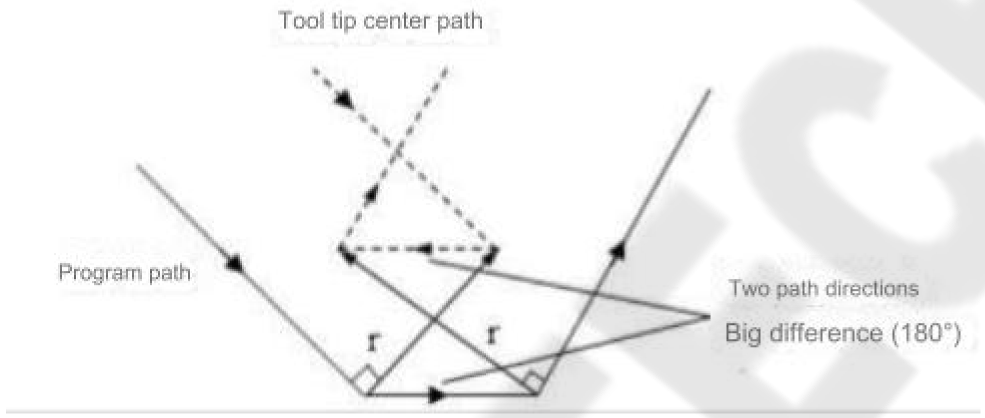


Figure 4-32a Processing interference (1)

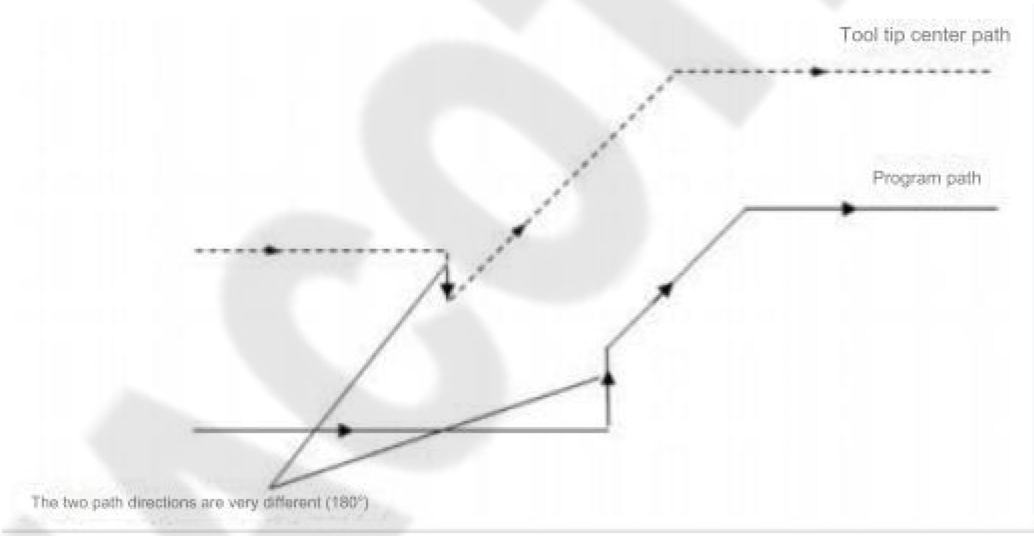


Figure 4-32b Processing interference (2)

(2) There is actually no interference, but it is also treated as interference.

1) The groove depth is less than the compensation amount

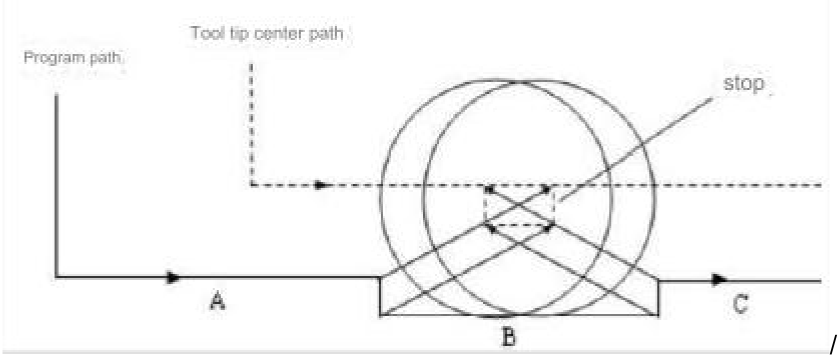


Figure 4-33 Special cases for interference processing (1)

There is actually no interference, but the direction of the program in segment B is opposite to the path of tool nose radius compensation, the tool stops and an alarm is displayed.

2) The groove depth is less than the compensation amount

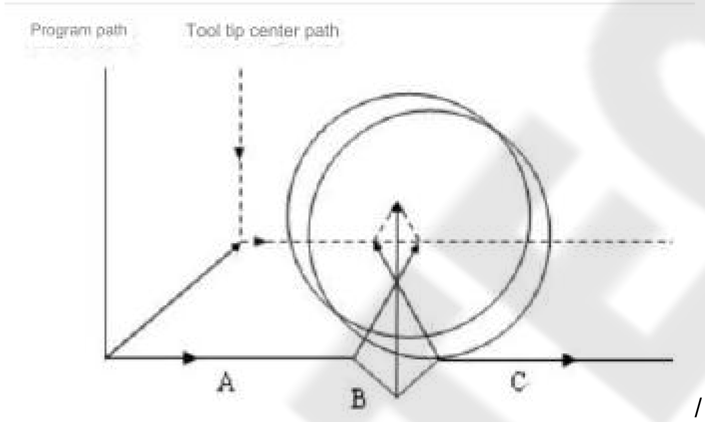


Figure 4-34 Several special cases for interference processing (2)

There is actually no interference, but the direction of the program in segment B is opposite to the path of tool nose radius compensation, the tool stops and an alarm is displayed.

4.2.6 Code for Temporarily Cancelling Compensation Vector

In compensation mode, if G50 and G71 ~ G76 codes are specified, the compensation vector will be temporarily canceled. After executing the code, the compensation vector will be automatically restored. The temporary cancellation of compensation at this time is different from the compensation cancellation mode. The tool moves directly from the intersection to the instruction point for canceling the compensation vector. When the compensation mode is restored, the tool moves directly to the intersection again.

* Coordinate system setting G50 code

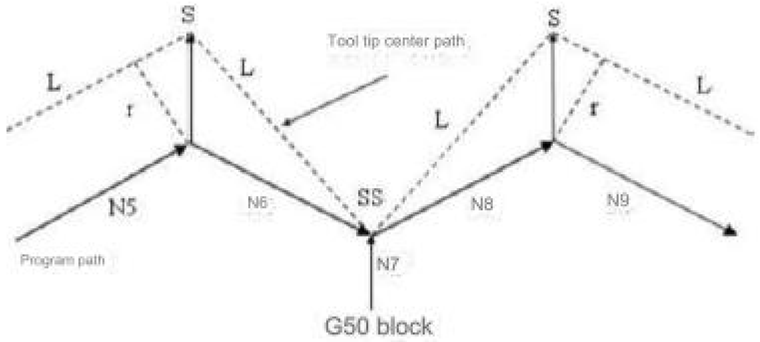


Figure 4-35 G50 temporarily cancels compensation vector

Note: SS indicates the point where the tool stops twice in single program segment mode.

In compensation mode, if G28 is specified, compensation will be canceled at the middle point, and the compensation mode will be automatically restored after returning to the reference point.

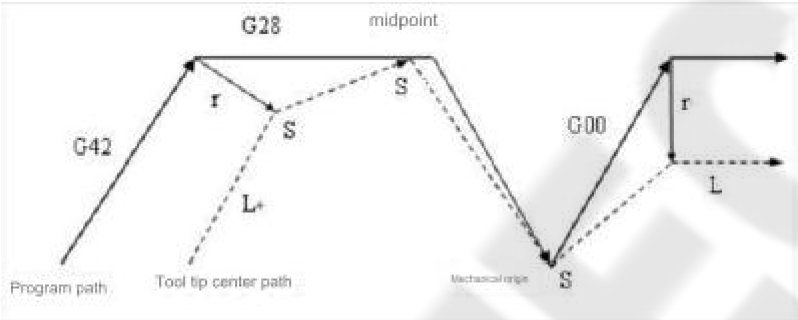


Figure 4-36 G28 temporarily cancels compensation vector

*G71 ~ G75 compound cycle; G76, G92 thread cutting

When executing G71 ~ G76 fixed cycle code; G92 thread cutting code, tool nose radius compensation is not executed during the cycle, tool nose radius compensation is temporarily canceled, and there are G00, and G01 codes in the following program segment, and CNC will automatically restore the compensation mode.

*G32, G33, G34 and other thread cutting

Can’t be run in tool nose radius compensation mode. If it is run, alarm 131# "... Instruction cannot be used in C tool compensation".

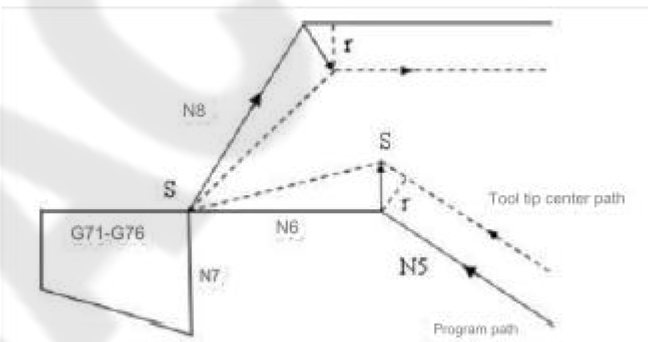


Figure 4-37 G71 ~ G76 temporarily cancel compensation vector

*G90, G94 code

G90 or G94 code performs tool nose radius compensation mode:

A. The original tool nose radius compensation will be canceled when positioning to the cycle start point;

B. The previous C compensation is established before cutting starts, and the trajectory ① in the figure below will establish the original radius compensation mode;

C. The trajectories ② and ③ in the figure below are cutting with radius compensation;

D. The trajectory ④ in the figure below will cancel the radius compensation and return to the cycle start point; there are G00 and G01 codes in the following program segment, and the CNC will automatically restore the compensation mode;

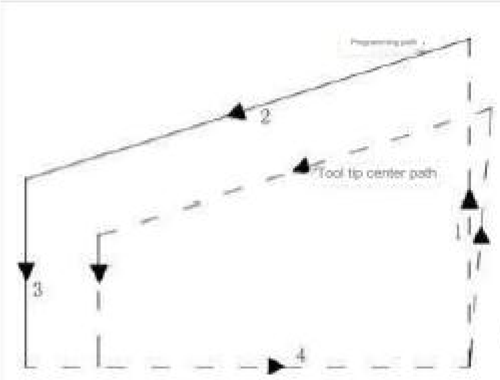


Figure 4-38 G90 tool nose radius compensation offset direction

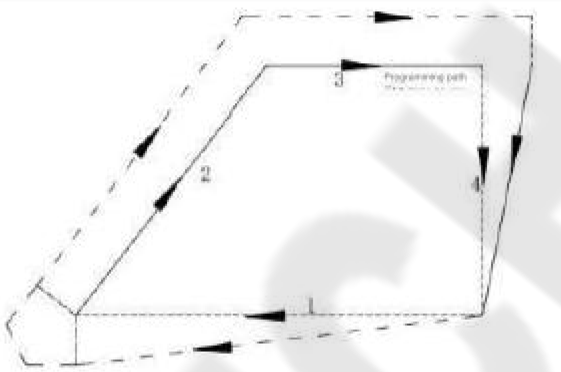


Figure 4-39 G94 tool nose radius compensation offset direction

4.2.7 Special cases

* When the inner corner processing is smaller than the tool nose radius, the inner offset of the tool will cause excessive cutting. After the start or corner move of the previous segment, the tool movement stops and an alarm is displayed (P/S41). However, if the 'single segment' switch is ON, the tool will stop at the end point of the previous segment.

* When machining a concave shape smaller than the tool tip diameter, when the tool tip radius compensation causes the tool tip center to move in the opposite direction to the program path, overcutting will occur. In this case, after the start or corner move of the previous segment, the tool movement stops and an alarm is displayed.

* When machining a step smaller than the tool nose radius, the program contains a step smaller than the tool tip radius and the step is an arc, the tool center path may form a movement direction opposite to the program path. In this case, the first vector will be automatically ignored and the tool will move directly to the end point of the second vector in a straight line. In single segment, the program will stop at this point. If it is not in single segment mode, the cycle operation will continue. If the step is a straight line, compensation will be performed correctly without generating an alarm. (However, the uncut portion will still remain)

* When the G code contains a subroutine

Before calling the subroutine (i.e. before executing M98), the CNC must be in compensation cancel mode. After entering the subroutine, the offset can be started, but before returning to the main program (i.e. before executing M99), it must be in compensation cancel mode. Otherwise, an alarm will be issued.

* When changing the compensation amount

(a) Usually, the compensation amount value is changed when changing the tool in cancel mode. If the compensation amount is changed in compensation mode, the new compensation amount will only be effective after the tool change.

(b) Positive and negative compensation amount and tool tip center path

If the compensation amount is negative (-), G41 and G42 are interchanged in the program. If the tool center moves along the outside of the workpiece, it will move along the inside, and vice versa.

As shown in the following example. Generally, the compensation amount is (+) when making a program. When the tool path is made as shown in (a), if the compensation amount is negative (-), the tool center moves as shown in (b), and vice versa.

* The end point of the programmed circular arc is not on the circular arc

When the end point of the circular arc in the program is not on the arc, the tool movement stops and displays the alarm message "The end point of the circular arc is not on the arc".

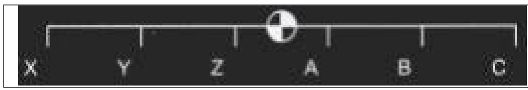
Part 2

Operating Instructions

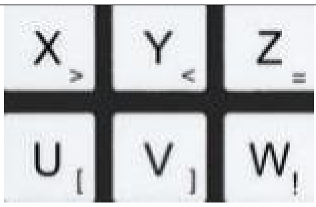
Chapter 1 Operation Mode and Display Interface

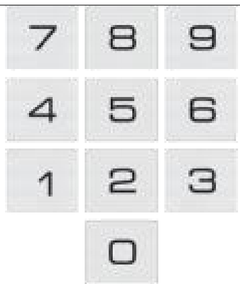
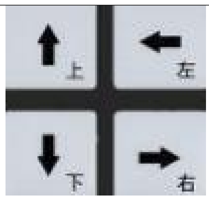
1.1 Panel Division

1.1.1 Status Indication

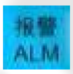
	Axis homing end indicator

1.1.2 Editing Keypad

Key	Name	Function description
	Reset key	CNC reset, feed, output stop, etc.
	Address key	Address input
		With dual address keys, press repeatedly to switch between the two

	Number key	Number input
	Input key	Confirmation of parameter, compensation amount and other data input
	Conversion key	Information, display switching
	Edit key	Delete previous and next when editing the program
	EOB key	Input of program segment end character
	Cursor moving key	Control cursor movement
	Page turn key	Switch pages in the same display interface

1.1.3 Display Menu

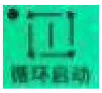
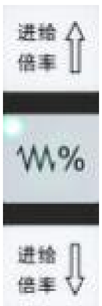
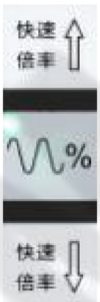
Menu key	Remarks
	Enter the position interface. The position interface has four pages: relative coordinates, absolute coordinates, comprehensive coordinates, and program monitoring
	Enter the program interface. The program interface has four pages: program content, program status, program directory, and file directory
	Enter the compensation interface, macro variable interface, and tool life management (parameter setting function), and press repeatedly to switch among the three interfaces. The compensation interface displays tool offset wear; the macro variable interface displays CNC macro variables; tool life management displays the current tool life usage and sets the tool group number
	Enter the alarm interface and alarm log, and press the key repeatedly to switch between the two interfaces. The alarm interface has two pages: CNC alarm and PLC alarm; the alarm log displays the history of alarm generation and alarm elimination
	Enter the setting interface and graphic interface, and press the key repeatedly to switch between the two interfaces. The setting interface has switch settings, G54-G59, data operations, authority settings, and time settings; the graphic interface displays the movement trajectory of the feed axis

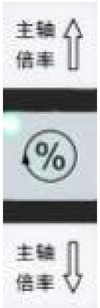
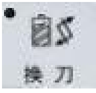
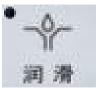
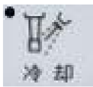
	Enter the status parameter, data parameter, and screw compensation parameter interfaces. Press the key repeatedly to switch among the interfaces.
	Enter the CNC diagnosis interface, PLC status, PLC data, machine tool soft panel, and version information interface. Press the key repeatedly to switch among the interfaces. CNC diagnostic interface, PLC status, PLC data display, CNC internal signal status, PLC address, and data status information; machine tool soft panel can be used for machine tool soft keypad operation; version information interface displays CNC software, hardware and PLC version number.
	Enter the ladder diagram interface, PLC version overview, PLC status, PLC data, ladder diagram interface, and press the key repeatedly to switch among the interfaces.

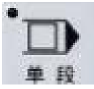
1.1.4 Machine Tool Panel

The function of the buttons in the machine tool panel is defined by the PLC program (ladder diagram). For the specific function meaning of each key, please refer to the manual of the machine tool.

The functions of the keys on the machine tool panel defined by the series standard PLC program are shown in the table below:

Key	Name	Function description	Operation mode when the function is valid
	Feed hold key	Program, MDI code running pause	Automatic mode, input mode
	Cycle start key	Program, MDI code running start	Automatic mode, input mode
	Feed rate override key	Adjustment of feed speed	Automatic mode, input mode, edit mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing
	Rapid override key	Fast moving speed adjustment	Automatic mode, input mode, machine tool homing, manual mode, program homing

	Spindle override key	Spindle speed adjustment (spindle speed analog control mode is effective)	Automatic mode, input mode, edit mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing
	Manual tool change key	Manual tool change	Machine tool homing, handwheel mode, single step mode, manual mode, program homing
	Jog switch key	Spindle jog status on/off	Machine tool homing, handwheel mode, single step mode, manual mode, program homing
	Lubrication switch key	Machine tool lubrication on/off	Automatic mode, input mode, edit mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing
	Coolant switch key	Coolant on/off	Automatic mode, input mode, edit mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing
	Spindle control key	Spindle forward, spindle stop, spindle reverse	Machine tool homing, handwheel mode, single step mode, manual mode, program homing
	Rapid switch	Rapid speed/feed speed switching	Automatic mode, input mode, manual mode
	X-axis feed key	In manual and single-step operation modes, each axis moves in a positive/negative direction	Machine tool homing, single-step mode, manual mode, program homing
	Z-axis feed key		
	Y-axis feed key		
	4th axis feed key		

	Handwheel/single-step increment selection and rapid override selection key	Handwheel movement per grid 1/10/100/1000 * least equivalent single-step movement per step 1/10/100/1000 * least equivalent rapid override Fo, 25%, F50%, F100%	Automatic mode, input mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing
	Selection stop	When the selection stop is valid, execute M01 Pause	Automatic mode, input mode
	Single-segment switch	Switch between single-segment operation and continuous operation of the program. When single-segment operation is valid, the single-segment operation indicator is on	Automatic mode, input mode
	Program segment skip switch	Whether the program segment with a "/" at the beginning of the segment is skipped or not. When the segment skip switch is turned on, the segment skip indicator is on.	Automatic mode, input mode
	Machine tool lock switch	When the machine tool is locked, the machine tool lock indicator is on, and the feed axis output is invalid	Automatic mode, input mode, edit mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing
	Auxiliary function lock switch	When the auxiliary function is locked, the auxiliary function lock indicator is on, and the M, S, T function output is invalid	Automatic mode, input mode
	Dry run switch	When dry run is valid, the dry run indicator is on, and the processing program/MDI code segment is dry run	Automatic mode, input mode
	Edit mode selection key	Enter the editing operation mode	Automatic mode, input mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing
	Automatic mode selection key	Enter automatic operation mode	Input mode, edit mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing

	Input mode selection key	ut (MDI) operation mode	Automatic mode, edit mode, machine tool homing, handwheel mode, single-step mode, manual mode, program homing
	Machine tool homing mode selection key	Enter machine tool homing operation mode	Automatic mode, input mode, edit mode, handwheel mode, single-step mode, manual mode, program homing
	Single-step/handwheel mode selection key	Enter single-step or handwheel operation mode (one of the two operation modes is selected by parameters)	Automatic mode, input mode, edit mode, machine tool homing, manual mode, program homing
	Manual mode selection key	Enter manual operation mode	Automatic mode, input mode, edit mode, machine tool homing, handwheel mode, single-step mode, program homing
	Program homing mode selection key	Enter program homing operation mode	Automatic mode, input mode, edit mode, machine tool homing, handwheel mode, single-step mode, manual mode

1.2 Operation Mode Overview

There are seven operation modes: edit, automatic, input, machine tool homing, single-step/handwheel, manual, and program homing.

● Edit mode

In edit mode, you can create, delete, and modify the machining program.

● Automatic mode

In automatic mode, the program runs automatically.

● Input mode

In input mode, you can input parameters, and input and execute code segments.

● Machine tool homing mode

In machine tool homing mode, you can perform machine tool homing operations for the feed axes respectively.

● Handwheel/single-step mode

In single-step/handwheel feed mode, the CNC moves according to the selected increment.

● Manual mode

In manual mode, you can perform manual feed, manual fast, feed rate override adjustment, rapid override adjustment, spindle start and stop, coolant switch, lubricant switch, spindle jog, manual tool change and other operations.

- Program homing mode

In program homing mode, you can perform program homing operations for the feed axes respectively.

1.3 Display Interface

There are 8 interfaces such as position interface and program interface, and each interface has multiple display pages. Each interface (page) is independent of the operation mode.

Chapter 2 Power-on, Power-off and Safety Protection

2.1 Power on

Before the system is powered on, it should be confirmed that:

1. The machine tool is in a normal state.
2. The power supply voltage meets the requirements.
3. The wiring is correct and firm.

2.2 Power off

Before powering off, it should be confirmed that:

1. The feed axis of CNC is in a stopped state;
2. Auxiliary functions (such as spindle, water pump, etc.) are turned off;
3. Cut off the CNC power supply first, and then cut off the machine power supply.

Note: For the operation of cutting off the machine power supply, please refer to the manual of the machine tool.

2.3 Emergency Operation

During the processing, some unexpected results may occur due to user programming, operation and product failure. At this time, the SZGH880T/SZGH1080T/S280ti series must be stopped immediately. This section describes the processing that can be performed in an emergency. For the processing of CNC machine tools in emergency situations, please refer to the relevant instructions of the machine tool.

2.3.1 Reset

When the system output is abnormal or the coordinate axis moves abnormally, press the 

key to reset the system:

1. All axis movements stop;
2. M and S function outputs are invalid (whether the spindle counterclockwise/clockwise rotation, lubrication, cooling and other signals are automatically turned off by pressing the 

key is set by the parameter, defined by the PLC ladder diagram);
3. Automatic operation ends, and the modal function and state are maintained.

2.3.2 Emergency Stop

When the emergency stop button is pressed in a dangerous or emergency situation during the operation of the machine tool (when the external emergency stop signal is valid), the CNC enters the emergency stop state. At this time, the machine tool movement stops immediately, and the spindle rotation, coolant and other outputs are all turned off. Release the emergency stop button to cancel the emergency stop alarm, and the CNC enters the reset state.

Note 1: Before canceling the emergency stop alarm, confirm that the fault has been eliminated;

Note 2: Pressing the emergency stop button before powering on and off can reduce the electrical shock of the equipment;

Note 3: After the emergency stop alarm is canceled, the machine homing operation should be performed again to ensure the correctness of the coordinate position (if the machine tool origin is not installed, the machine tool homing operation shall not be performed);

Note 4: Only when the system parameter 130 is set to 0, the external emergency stop is effective.

2.3.3 Feed Hold

During the operation of the machine tool, the operation can be paused by pressing the 

key. It should be noted that during thread cutting and cycle code operation, this function cannot stop the operation immediately.

2.3.4 Cut off Power

During the operation of the machine tool, the power supply of the machine tool can be immediately cut off in dangerous or emergency situations to prevent accidents. However, it must be noted that after the power is cut off, the CNC displayed coordinates may deviate greatly from the actual position, and re-calibration and other operations must be performed.

Chapter 3 Manual Operation

The functions of the keys on the machine tool panel are defined by the PLC program (ladder diagram). Please refer to the manual of the machine tool for the functions and meanings of each key.

Please note that the following functions related to the keys on the operation panel in this chapter are described for standard PLC programs!

Press the  key to enter the manual mode, in which you can perform manual feeding, spindle control, override adjustment, tool change and other operations.

3.1 Coordinate Axis Movement

In the manual mode, you can make two axes manually feed and manually move quickly.

3.1.1 Manual Feed

Press and hold  or  X-axis direction key in the feed axis direction selection key  to feed the X-axis in the negative or positive direction, and release the key to stop the axis movement; press and hold  or  Z-axis direction key to feed the Z-axis in the negative or positive direction, and release the key to stop the axis movement stops; press and hold  or  Y-axis direction key to feed the Y -axis in the negative or positive direction, and release the key to stop the axis movement; press and hold  or  4th axis direction key to feed the 4th axis in the negative or positive direction, and release the key to stop the axis movement.

When manual feed is performed, press the  key to make the indicator light up to enter the manual rapid movement state.

3.1.2 Manual Rapid Movement

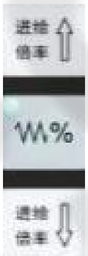
Press the  key until the rapid movement indicator light is on. Press the  or  key to make the X-axis move rapidly in the negative or positive direction, and release the key to stop the axis movement; press the  or  key to make the Z-axis move rapidly in the negative or positive direction, and release the key to stop the axis movement; press the  or  Y-axis direction key to make the Y-axis feed in the negative or positive direction, and release the key to stop the axis movement; press the  or  4th axis direction key to make the 4th axis feed in the negative or positive direction, and release the key to stop the axis movement. Rapid override adjustment is effective in real-time.

When performing manual rapid movement, press the  key to turn off the indicator light, the rapid movement is invalid, and the feed is at manual speed.

Note 1: If there is no return to the reference point after the power is turned on, the rapid movement speed is the manual feed speed or the rapid movement speed is selected by parameter N0048 when the rapid movement switch is turned on (the rapid movement indicator light is on);

Note 2: In edit/handwheel mode, the key is invalid.

3.1.3 Speed Adjustment

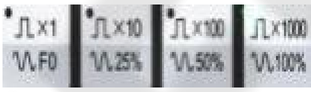
During manual feeding, you can press  to modify the manual feed rate override, with a total of 16 levels. When parameter N0051 is set to 1260, the relationship between the feed rate override and the feed speed is shown in the following table

Feed rate override (%)	Feed speed (mm/min)
0	0
10	126
20	252
30	378
40	504
50	630
60	756

70	882
80	1008
90	1134
100	1260
110	1386
120	1512
130	1638
140	1764
150	1890

Note: This table has an error of about 2%.

During manual rapid movement, you can press 

or 

to modify the manual rapid movement override, which has four levels: Fo, 25%, 50%, and 100%. (Fo speed is set by data parameter N0022)

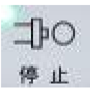
The rapid override selection is effective in the following situations:

(1) G00 rapid movement; (2) Rapid movement in fixed cycle; (3) Rapid movement in G28; and (4) Manual rapid movement

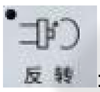
3.2 Other Manual Operations

3.2.1 Counterclockwise/Clockwise Rotation, Stop Control

: In manual mode, press this key to rotate the spindle counterclockwise;



: In manual mode, press this key to stop the spindle;



: In manual mode, press this key to rotate the spindle clockwise.

3.2.2 Spindle Jog

: At this time, the spindle is in a jog state.

Function description: Press the 

key on the panel to enter the jog state. The spindle jog function can be turned on/off only when the spindle is stopped. In the spindle jog state, press the 

key to jog in the counterclockwise direction; press the 

key to jog in the clockwise direction. The jog speed is set by parameter N0118.

When the spindle is jogging, press the 

key to stop the spindle jogging. When the jogging stops, the spindle braking signal will not be output.

When K10.4 is set to 1, the spindle jogging is valid in any mode. When the spindle is in a jogging state in automatic or input mode, running the program will turn off the spindle jogging and the jogging function.

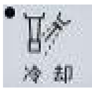
Parameter setting:

PLC parameter K10.4 1/0: Spindle jogging is valid in any mode/manual, handwheel, and homing mode.

System parameter N0118: Spindle rotation speed during jogging

Timer T12: Spindle jogging time

3.2.3 Coolant Control



: In any operation mode, press this key to switch the coolant between switches.

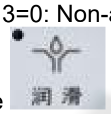
Parameter setting: PLC parameter K10.1 1/0: Spindle lubrication and cooling output are kept/turned off during reset

3.2.4 Lubrication Control

Function description:

1. Non-automatic lubrication:

DT13=0: Non-automatic lubrication.

Press the 

key on the machine tool panel to start the lubrication output. Press again to cancel the lubrication output. When M32 is executed, the lubrication output is activated, and then M33 is executed to cancel the lubrication output.

2. Automatic lubrication:

DT13>0: Automatic lubrication, you can set lubrication time DT13 and lubrication interval time DT53

After power-on, lubrication starts for the time set by DT13, and then stops output. After the time set by DT53, the lubrication output is repeated, and the cycle continues. During automatic lubrication, M32 and M33 codes, and machine tool panel 

key are also valid.

Parameter setting:

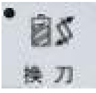
PLC parameter: K10.1 1/0: Spindle lubrication and cooling output are held/turned off during reset

PLC parameter: K16.2 1/0: Whether to lubrication output at power on when automatic lubrication is effective

PLC data: DT53: Automatic lubrication interval time (ms)

PLC data: DT13: Automatic lubrication output time (ms)

3.2.5 Manual Tool Change



: In manual mode, press this key to manually change tools in sequence (if the current tool is the first, press this key to change the tool to the second; if the current tool is the last, press this key to change the tool to the first).

3.2.6 Adjustment of Spindle Override

In manual mode, the spindle speed can be adjusted when analog voltage output is selected to control the spindle speed.

Press the


key to adjust the spindle override to change the spindle speed. The spindle override can be adjusted in 8 levels in real time from 50% to 120%.

Chapter 4 Handwheel/Single-step Operation

In a handwheel/single-step mode, the machine tool moves according to the selected increment value.

The functions of the keys on the machine tool panel are defined by the PLC program (ladder diagram). Please refer to the manual of the machine tool for the functions and meanings of each key.

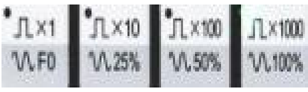
Please note that the following functions related to the keys on the operation panel in this chapter are described for standard PLC programs!

4.1 Single-step Feed

Set system parameter N0052 to 0 and press the 

key to enter the single-step mode.

4.1.1 Increment Selection

Press the 

key to select the movement increment, and the movement increment will be displayed on the page. When BIT7 (SINC) of PLC status parameter K016 is 1, the 

step value is invalid; when BIT7 is 0, it is valid.

4.1.2 Movement Direction Selection

Press the 

or 

key once to feed the X axis in the negative or positive direction in a single-step increment; press the 

or 

key once to feed the Z-axis in the negative or positive direction in a single-step increment. Press the 

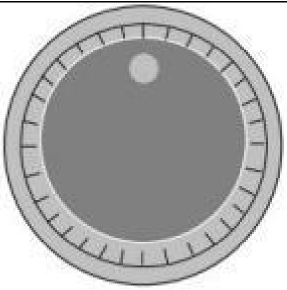
or 
key once to feed the Y-axis in a single-step increment in the negative or positive direction.

4.2 Handwheel (Hand-Crank Pulse Generator) Feed

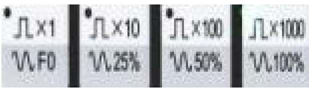
Set system parameter N0052 to 1, and press the 

key to enter the handwheel operation mode.

The appearance of the handwheel is shown in the figure below:



4.2.1 Increment Selection

Press the 

key to select the movement increment, which will be displayed on the page.

When BIT7 (SINC) of PLC parameter K016 is 1, the 

step value is invalid; when BIT7 is 0, it is valid.

4.2.2 Selection of Moving Axis and Direction

In the handwheel mode, press the keys of 

and 

to select the corresponding axis. The handwheel feed direction is determined by the handwheel rotation direction. Generally, the handwheel feeds clockwise for positive feeding and counterclockwise for negative feeding. If the handwheel feeds clockwise for negative feeding and counterclockwise for positive feeding, the A and B signals of the handwheel end can be exchanged. The feed direction when the handwheel rotates can also be selected by parameter N0061.

4.2.3 Notes

1. The relationship between the handwheel scale and the machine tool movement is shown in the following table:

	The movement of each scale on the handwheel			
Handwheel increment	0.001	0.01	0.1	1
Coordinate specified value	0.001mm	0.01mm	0.1mm	1mm

(Take the least input increment of 0.001mm as an example)

Note 1: The handwheel increment is related to the current metric input status of the system and the least input increment of the system;

Note 2: The speed of handwheel rotation shall not exceed 5r/s. If it exceeds 5r/s, the scale value and the movement amount may not match.

Chapter 5 Input Operation

In the input mode, parameters can be set, code words can be entered, and code words can be executed.

The functions of the keys on the machine tool panel are defined by the PLC program (ladder diagram). Please refer to the manual of the machine tool for the functions and meanings of each key.

Please note that the following functions related to the keys on the operation panel in this chapter are described for standard PLC programs!

5.1 Code Word Input

Select the input mode, enter the program status page, and enter a program segment G50 X50 Z100. The operation steps are as follows:

1. Press the 

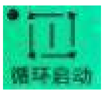
key to enter the input mode;

2. Press the 

key to enter the MDI page;

3. Enter the instruction.

5.2 Code Word Execution

Press the 

key to execute the input program segment. During the operation, you can press the 

key, 

key and emergency stop key to stop the program segment.

Note: Subroutine call codes (M98 P; etc.), compound cutting cycle codes (G70, G71, G72, G73, G74, G75, G76, etc.) are invalid under MDI.

5.3 Parameter Setting

In the input mode, enter the parameter interface to modify the parameter value.

5.4 Other Operations

1. The spindle override can be adjusted

In the handwheel/single-step mode, the spindle speed can be adjusted when analog voltage output is selected to control the spindle speed.

Press the 

key to adjust the spindle override to change the spindle speed. The spindle override can be adjusted in 8 levels in real time from 50% to 120%.

2. Rapid override can be adjusted

Press the 

key to adjust the rapid moving feed speed, which can be adjusted in 4 levels in real time.

3. Feed rate override can be adjusted

In the input mode, press the 

key to adjust the feed override to change the feed speed, and the actual speed can be adjusted in 16 levels from 0 to 150% of the feed speed specified by the F code.

Chapter 6 Tool Offset and Tool Setting

In order to simplify programming, it is allowed not to consider the actual position of the tool during programming. Three tool setting methods are provided: fixed-point, trial cutting and machine tool homing. Tool offset data is obtained through tool setting operations.

6.1 Fixed-point Tool Setting

The operation steps are as follows:

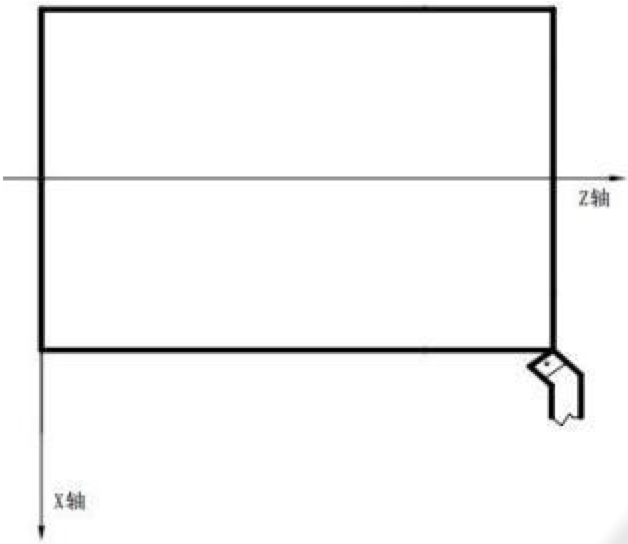


Figure A

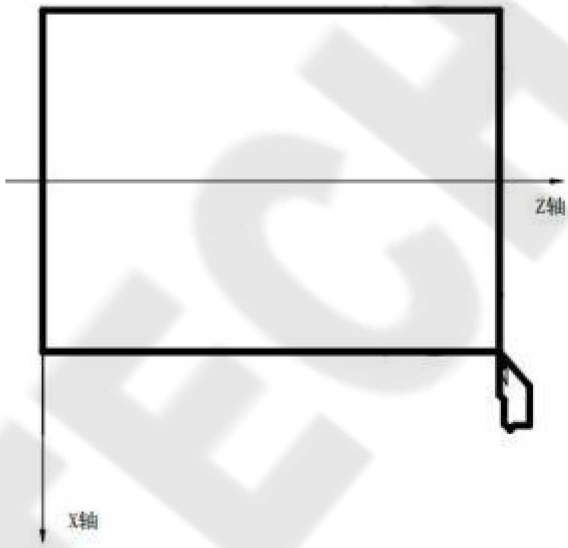
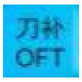
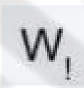


Figure B

1. First, determine whether the tool compensation values in the X and Z directions are zero. If not, the tool compensation values of all tool numbers must be cleared;
2. Set the offset number in the tool to 00 (such as T0100, T0300);
3. Select any tool (usually the first tool in the processing, which is used as the reference tool);
4. Position the tip of the reference tool to a certain point (tool setting point), as shown in Figure A;
5. Use G50 X_Z_ code to set the workpiece coordinate system in the input mode and program status page;
6. Clear the coordinate value of the relative coordinate (U, W);
7. After moving the tool to a safe position, select another tool and move it to the tool setting point, as shown in Figure B;

8. Press the  key,  key,  key to move the cursor to select the tool offset number corresponding to the tool;

9. Press the address key  , and then press the  key to set the X-axis tool offset value to the corresponding offset number;

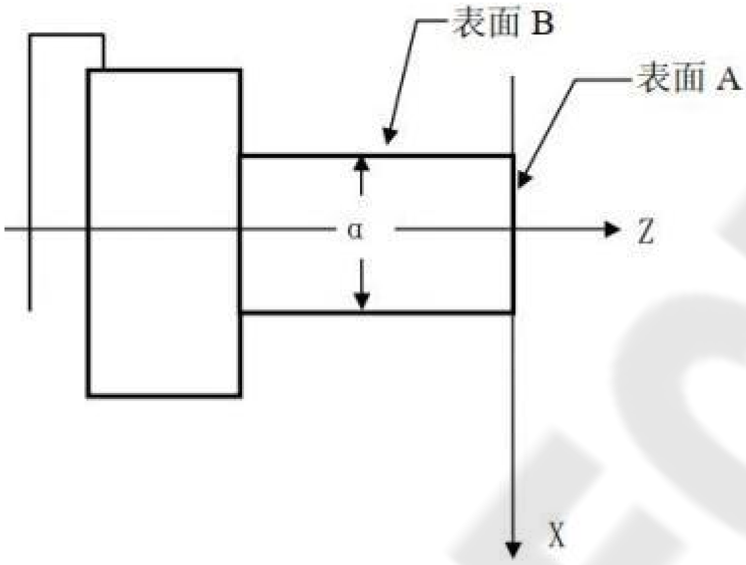
10. Press the address key  , and then press the  key to set the Z-axis tool offset value to the corresponding offset number;

11. Repeat steps 7~10 to set other tools.

Note: During fixed point tool setting, the original tool offset in the system must be cleared first. When pressing U and W to enter a new tool offset value, it cannot be repeated multiple times, but can only be entered once. For details on how to clear the tool offset value, see Section 7.4.4 of this part.

6.2 Trial Cutting Tool Setting

The operation steps are as follows (the workpiece coordinate system is established with the workpiece end face):



1. Select any tool and make the tool cut along the A surface;

2. With the Z-axis not moving, withdraw the tool along the X-axis and stop the spindle rotation;

3. Press the 刀补 OFT key to enter the offset interface, select the tool offset page, and press the ↑ 上 key and ↓ 下 key to move the cursor to select the offset number corresponding to the tool;

4. Enter the address key Z =, number key 0 and 输入 IN key in sequence;

5. Make the tool cut along the B surface;

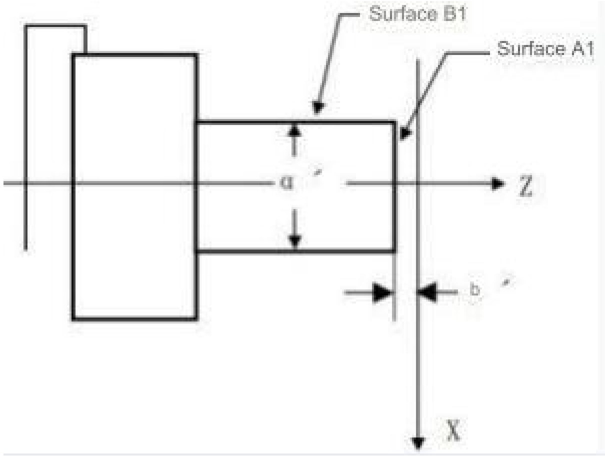
6. With the X-axis not moving, withdraw the tool along the Z-axis and stop the spindle rotation;

7. Measure the diameter "α" (assuming α=15);

8. Press the 刀补 OFT key to enter the offset interface, select the tool offset page, and press the ↑ 上 key and ↓ 下 key to move the cursor to select the offset number corresponding to the tool;

9. Enter the address key X >, number key 1, 5 and 输入 IN key in sequence;

10. Move the tool to the safe tool change position and change to another tool;



11. Make the tool cut along the A1 surface;

12. With the Z-axis not moving, withdraw the tool along the X-axis and stop the spindle rotation;

13. Measure the distance "β´" between the A1 surface and the origin of the workpiece coordinate system (assuming β´=1);

14. Press the 刀补 OFT key to enter the offset interface, select the tool offset page, and press the ↑ 上 key and ↓ 下 key to move the cursor to select the offset number corresponding to the tool;

15. Press the address key Z =, symbol key –, number key 1 and 输入 IN key in sequence;

16. Make the tool cut along the B1 surface;

17. With the X-axis not moving, withdraw the tool along the Z-axis and stop the spindle rotation;

18. Measure the distance "α´" (assuming α´=10);

19. Press the 刀补 OFT key to enter the offset interface, select the tool offset page, and press the ↑ 上 key and ↓ 下 key to move the cursor to select the offset number corresponding to the tool;

20. Enter the address key X, number key 1, 0 and 输入 IN key in sequence;

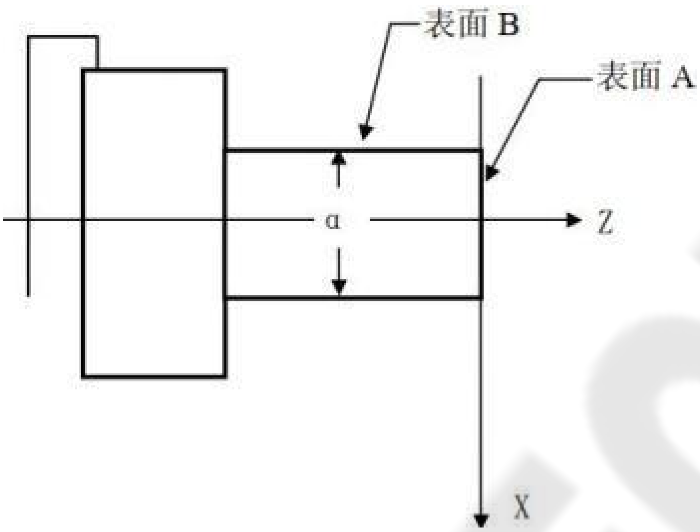
21. Repeat steps 10~20 for other tool setting methods.

Note: The tool compensation value of this tool setting method may be very large, so the CNC must be set to perform tool compensation in coordinate offset mode (CNC parameter N0163 is set to 1), and the first program segment uses T code to perform tool length compensation or the first movement code program segment of the program contains T code for performing tool length compensation.

6.3 Machine Tool Homing Tool Setting

This tool setting method does not have the problem of reference tool or non-reference tool. When the tool is worn or any tool is adjusted, just re-set the tool. Perform machine tool homing once before tool setting. After power off, just perform machine tool homing once to continue processing. The operation is simple and convenient.

The operation steps are as follows (the workpiece coordinate system is established with the workpiece end face):



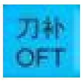
1. Press the 

key to enter the machine tool homing mode and perform machine tool homing for two axes;

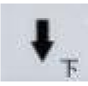
2. Select any tool and set the offset number in the tool to 00 (such as T0100, T0300);

3. Make the tool cut along the A surface;

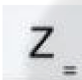
4. With the Z-axis not moving, withdraw the tool along the X-axis and stop the spindle rotation;

5. Press the 

key to enter the offset interface, select the tool offset page, and press the 

key and 

key to move the cursor to select a certain offset number;

6. Press the address key 

, number key 

and 

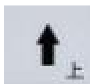
key in sequence, and the Z-axis offset value is set;

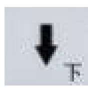
7. Make the tool cut along the B surface;

8. With the X-axis not moving, withdraw the tool along the Z-axis and stop the spindle rotation;

9. Measure the distance "α" (assuming α=15);

10. Press 

to enter the offset interface, select the tool offset page, and press the 

key and 

key to move the cursor to select the offset number;

11. Type the address key 

, number key 

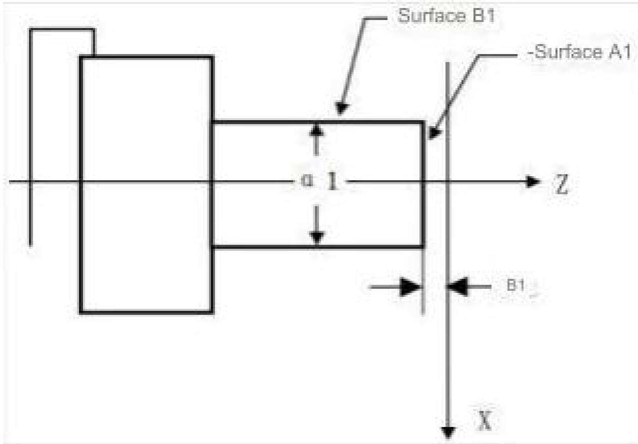
, 

and 

key in sequence to set the X-axis tool offset value;

12. Move the tool to the safe tool change position;

13. Change another tool and set the offset number in the tool to 00 (such as T0100, T0300);



14. Make the tool cut along the A1 surface;

15. With the Z-axis not moving, withdraw the tool along the X-axis and stop the spindle rotation; measure the distance "β´" between the A1 surface and the origin of the workpiece coordinate system (assuming β´=1);

16. Press the 刀补 OFT key to enter the offset interface, select the tool offset page, and press the ⬆ 上 key and ⬇ 下 key to move the cursor to select the offset number corresponding to the tool;

17. Press the address key Z =, symbol key –, number key 1 and 输入 IN key in sequence to set the Z-axis tool offset value;

18. Make the tool cut along the B1 surface;

19. With the X-axis not moving, withdraw the tool along the Z-axis and stop the spindle rotation;

20. Measure the distance "α1" (assuming α1=10);

21. Press the 刀补 OFT key to enter the offset interface, select the tool offset page, and press the ⬆ 上 key and ⬇ 下 key to move the cursor to select the offset number corresponding to the tool;

22. Press the address key X, the numeric key 1, 0 and 输入 IN key in sequence to set the X-axis tool offset value;

23. Move the tool to the safe tool change position;

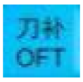
24. Repeat steps 12~23 to complete the tool setting of all tools.

Note 1: The machine tool must be equipped with a machine tool origin to perform the machine tool homing tool setting operation.

Note 2: After the machine tool homing tool setting, the G50 code cannot be executed to set the workpiece coordinate system.

Note 3: CNC must be set to perform tool compensation in coordinate offset mode (CNC parameter N0163 is set to 1), and the first program segment uses T code to perform tool length compensation or the first movement code program segment of the program contains T code for performing tool length compensation.

6.4 Setting and Modifying Tool Offset Values

Press the 

key to enter the offset interface, and use the 

key and 

key to display the No.000~No.032 offset numbers respectively.

6.4.1 Setting Tool Offset Value

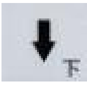
1. Press the 

key to enter the tool offset page, and press the 

key and 

key to select the required page;

2. Move the cursor to the position of the tool offset and wear the number to be entered. Scanning method: Press the 

key and 

key to move the cursor in sequence.

3. Press the address key 

or

and enter a number (decimal point is allowed);

4. After pressing the 

key, the CNC automatically calculates the tool offset and displays it on the page.

6.4.2 Modifying Tool Offset Value

1. Move the cursor to the position of the tool offset number to be changed according to the method described in the previous section;

2. If you want to change the tool offset value of the X-axis, type U; for the Z-axis, type W;

3. Type the incremental value;

4. Press 

to add the current tool offset value to the incremental value entered, and the result is displayed as the new tool offset value.

Example: The tool offset value of the X-axis has been set to 5.678. If you input the increment U1.5 on the keypad, the new tool offset value of the X-axis will be 7.178 (=5.678+1.5).

6.4.3 Clearing Tool Offset Value

1. Move the cursor to the position of the compensation number to be cleared.
2.

Method 1:

To clear the tool offset value of the X-axis, press the 

key, and then press the 

key;

To clear the tool offset value of the Z-axis, press the

key, and then press the 

key;

Method 2:

If the current tool offset value of the X-axis is α, enter U-α, then press the 

key, and the tool offset value of the X-axis will be zero;

If the current tool offset value of the Z-axis is β, enter W-β, then press the 

key, and the tool offset value of the Z-axis will be zero;

6.4.4 Setting and Modifying Tool Wear Values

In order to prevent misoperation when setting and modifying tool offset values (failure to enter a decimal point, incorrect decimal point position, etc.), resulting in excessive tool offset value modification, causing tool collision and other phenomena, and to help operators intuitively judge the degree of wear of each tool, the SZGH880T/SZGH1080T series has set up a tool wear page. When the tool compensation value needs to be modified due to tool wear and other reasons, it can be set or modified in the tool wear value. The input range of the tool wear value is set by the data parameter N0170. Tool wear data is saved after power failure.

The setting and modification methods of tool wear value are basically the same as those of tool offset value. Use U (X-axis), W (Z-axis), and V (Y-axis) to input the wear amount.

7.4.5 0# Tool Offset Translation Workpiece Coordinate System

When CNC parameter N0164 is set to 1, 0# tool offset translation workpiece coordinate system is valid. After entering a value in 0# tool offset, the workpiece coordinate system will shift according to the input value.

Chapter 7 Automatic Operation

The functions of the keys on the machine tool panel are defined by the PLC program (ladder diagram). Please refer to the manual of the machine tool for the functions and meanings of each key.

Please note that the following functions related to the keys on the operation panel in this chapter are described for standard PLC programs!

7.1 Automatic Operation

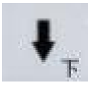
7.1.1 Selection of Running Program

a) Select automatic mode (must be in non-running state)

b) Press the 

key to enter the program directory display page;

c) Press the 

and 

key to move the cursor to the program name to be selected.

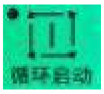
d) Press the 

key.

7.1.2 Starting Automatic Operation

1. Press the 

key to select the automatic mode;

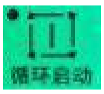
2. Press the  key to start the program, and the program will run automatically.

Note: The program starts from the line where the cursor is located, so before pressing the  key to run, you should first check whether the cursor is on the program segment to be run.

7.1.3 Stopping Automatic Operation

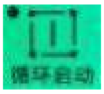
* Code stop (M00)

1. M00

After the program segment containing M00 is executed, the automatic operation is stopped, and all modal functions and states are saved. After pressing the panel key  or the external run key, the program continues to execute.

2. M01

Press the 

key, the stop indicator light is on, and the selection stop function is effective. After the program segment containing M01 is executed, the automatic operation is stopped, and all modal functions and states are saved. After pressing the panel key 

or the external run key, the program continues to execute.

* Press the relevant key to stop

1. After pressing the 

key or the external pause key during automatic operation, the machine tool will be in the following states:

(1) The machine tool feed decelerates and stops;

(2) The modal functions and states are saved;

(3) After pressing the 

key, the program continues to execute.

2. Press the reset key 

(1) All axis movements stop;

(2) M and S function outputs are invalid (whether the spindle counterclockwise/clockwise rotation, lubrication, cooling and other signals are automatically turned off by pressing the key is set by the parameter);

(3) Automatic operation ends, and the modal function and state are maintained.

3. Press the emergency stop key

When the emergency stop button is pressed in a dangerous or emergency situation during the operation of the machine tool (when the external emergency stop signal is valid), the CNC enters the emergency stop state. At this time, the machine tool movement stops immediately, and all outputs (such as spindle rotation, coolant, etc.) are turned off. Release the emergency stop button to cancel the emergency stop alarm, and the CNC enters the reset state.

4. Switching operation mode

When switching to machine tool homing, handwheel/single-step, manual, or program homing mode during automatic operation, the current program segment will be "paused" immediately; when switching to edit or input mode during automatic operation, "pause" will be displayed after the current program segment is run.

Note 1: Before canceling the emergency stop alarm, confirm that the fault has been eliminated;

Note 2: Pressing the emergency stop button before powering on and off can reduce the electrical shock of the equipment;

Note 3: After the emergency stop alarm is canceled, the machine homing operation should be performed again to ensure the correctness of the coordinate position (if the machine tool origin is not installed, the machine tool homing operation shall not be performed);

Note 4: External emergency stop is only effective when parameter N0173 is set to 0.

7.1.4 Automatic Operation from Any Segment

Press the edit key to enter the edit operation mode, press the program key to enter the program interface, and press the 

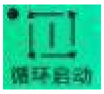
or 

key to select the program content page;

1. Move the cursor to the program segment to be run (such as starting from the second line, move the cursor to the beginning of the second line);

2. If the mode (G, M, T, F code) of the program segment where the current cursor is located is default and inconsistent with the mode of running the program segment, the corresponding modal function must be executed before continuing to the next step;

3. Press the 

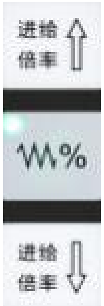
key to enter the automatic mode, and press the 

key to start the program running.

7.1.5 Adjustment of Feed and Rapid Speed

During automatic operation, the operation speed can be changed by adjusting the feed and rapid movement ratio without changing the speed value set in the program and parameters.

* Adjustment of feed override

Press the 

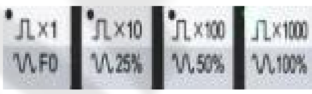
key to achieve real-time adjustment of 16 levels of feed override.

Note 1: The value specified by F in the feed override adjustment program;

Note 2: Actual feed speed = value specified by F × feed override.

* Adjustment of rapid override

Press 

or 

key to achieve real-time adjustment of four levels of rapid override: FO, 25%, 50%, 100%.

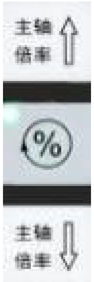
Note 1: CNC parameter N0021 sets the rapid movement rate of the X-axis and Z-axis respectively;

The actual rapid movement rate of X-axis and Z-axis = the value set by N0021 × rapid override

Note 2: When the rapid override is F0, the minimum rate of rapid movement is set by CNC parameter N0022.

7.1.6 Spindle Speed Adjustment

During automatic operation, when analog voltage output is selected to control the spindle speed, the spindle speed can be adjusted.

Press the 

key to adjust the spindle override to change the spindle speed, which can achieve a spindle override of 50%~120%, with a total of 8 levels of real-time adjustment.

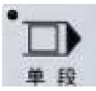
Note: The actual output analog voltage value = analog voltage value calculated according to the parameters × spindle override

7.2 Status During Operation

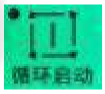
7.2.1 Single-Segment Operation

When executing the program for the first time, single-segment operation can be selected in order to prevent programming errors from happening unexpectedly.

In automatic mode, the method of turning on the single-segment program switch is as follows:

Press the 

key to make the single-segment indicator light up, indicating that the single-segment function is selected;

During single-segment operation, the CNC stops running after executing the current program segment; when continuing to execute the next program segment, you need to press the 

key again, and repeat until the program is completed.

Note 1: In G28 code, the single segment stops at the middle point;

Note 2: When executing fixed cycle G90, G92, G94, G70 ~ G76 codes, the single segment status can be found in Part 1 of *Programming* Instructions;

Note 3: The single segment is invalid when executing subroutine call (M98) and subroutine call return code (M99). However, in M98 and M99 program segments, except for addresses other than N, O, and P, the single segment stops are valid.

7.2.2 Dry Run

Before automatically running the program, you can select the dry run state to verify the program in order to prevent programming errors from happening unexpectedly. In automatic operation mode, the method to turn on the dry run switch is as follows:

Press the 

key to turn on the dry run indicator light in the status indication area, indicating that the dry run state has been entered; in the dry run state, the machine tool feed and auxiliary functions are valid (if the machine tool is locked and the auxiliary lock switch is in the off state), that is, the state of the dry run switch has no effect on the execution of the machine tool feed and auxiliary functions, the speed specified in the program is invalid, and the CNC moves at the speed in the table below.

	Program instruction	
	Quick movement	Cutting feed
Rapid movement button on	Quick movement	Manual feed maximum speed
Rapid movement button off	Manual feed speed or rapid movement (see note)	Manual feed speed

Note 1: CNC parameter N0015 can be used to set whether it is manual feed speed or rapid movement.

Note 2: In the dry run state, the rapid switch has no effect on the running speed of the current program segment, and all take effect on the next program segment.

Note 3: The standard ladder diagram defines that in the automatic state (automatic mode, input mode running), the dry run switch operation is invalid.

7.2.3 Machine Tool Lock Operation

In the automatic mode, the method of turning on the machine tool lock switch is as follows: Press the 

key to make the machine tool lock operation indicator light up, indicating that the machine tool has entered the locked operation state;

Machine tool lock operation is often used together with the auxiliary function lock function for program verification. When the machine tool is running in a locked state:

1. The machine tool pallet does not move, the "machine tool coordinates" in the comprehensive coordinate page under the position interface do not change, and the relative coordinates, absolute coordinates and residual movement amount are constantly refreshed, the same as when the machine tool lock switch is in the off state;

2. M, S, T codes can be executed normally.

7.2.4 Auxiliary Function Lock Operation

In automatic mode, the method to open the auxiliary lock switch is as follows: press the 

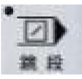
key to make the auxiliary function lock operation indicator light up, indicating that the auxiliary function lock operation state has entered;

At this time, M, S, T codes are not executed, and the machine tool pallet moves. It is usually used together with the machine tool lock function for program verification.

Note: When the auxiliary function lock is valid, it does not affect the execution of M00, M30, M98, and M99.

7.2.5 Program Segment Jump

When you do not want to execute a certain program segment in the program and do not want to delete it, you can select the program segment skip function. When the program segment has a "/" sign at the beginning and the program segment skip switch is turned on (the machine tool panel key or program skip external input is valid), this program segment will be skipped and not run during automatic operation. In automatic mode, the method to turn on the program segment skip switch is as follows:

Press the 

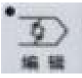
key to make the program segment skip indicator light up;

Note: When the program segment skip switch is not turned on, the program segment with a "/" sign at the beginning of the segment will not be skipped during automatic operation and will be executed as usual.

7.3 Other Operations

1. In automatic mode, press the 

key to switch the coolant on/off;

2. Press any key among 

, 

, 

, 

, 

or 

to achieve the conversion of operation mode;

3. Press the 

key to reset the CNC.

4. Automatic lubrication function (see Chapter 3 of this part for details).

Chapter 8 Homing Operation

The functions of the keys on the machine tool panel are defined by the PLC program (ladder diagram). Please refer to the manual of the machine tool for the functional meaning of each key.

Please note that the following functions related to the keys on the operation panel in this chapter are described for standard PLC programs!

8.1 Program Homing

8.1.1 Program Home

When the part is clamped on the machine tool, the absolute coordinates of the current position of the tool are set with the G50 code according to the relative position of the tool and the workpiece, and the workpiece coordinate system is established in the CNC. The current position of the tool is called the program home, and it returns to this position after executing the program homing operation.

8.1.2 Operation Steps of Program Homing

1. Press the 

key to enter the program homing operation mode, and the bottom line of the page displays "Program home";

2. Press any direction key of the X, Z, or Y axis to return to the X, Z, or Y axis program home;

3. The machine tool axis moves along the program home direction. After returning to the program home, the axis stops moving and the homing end indicator light is on.

Note 1: After program homing, the current tool offset state is not changed. If there is a tool offset, the position returned to is the position set by G50, which is the position containing the tool offset.

8.2 Machine Tool Homing

8.2.1 Machine Tool Home

The machine tool coordinate system is the reference coordinate system for CNC coordinate calculation and is the inherent coordinate system of the machine tool. The origin of the machine tool coordinate system is called the machine tool home (or machine tool reference point). The machine tool home is determined by the home switch or homing switch installed on the machine tool. Usually, the home switch or homing switch is installed at the maximum stroke in the positive direction of the X-axis and Z-axis.

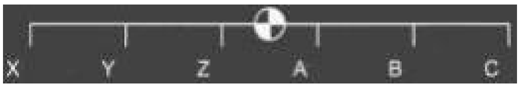
8.2.2 Operation Steps for Machine Tool Homing

1. Press the 

key to enter the machine tool homing mode. The bottom line of the page displays "machine home";

2. Press , ,  or  key to select the machine tool home of X, Z, Y or 4th axis;

3. The machine tool moves along the machine tool home direction and returns to the machine tool home after the deceleration signal and home signal detection. At this time, the axis stops moving and the homing end indicator light is on.



Note 1: If the CNC machine tool is not installed with the machine home, the machine tool homing operation shall not be used;

Note 2: The homing end indicator light goes out in the following cases:

1) Move out of the home;

2) CNC is powered off;

Note 3: After the machine tool homing operation, the CNC cancels the tool length compensation;

Note 4: For details on the parameters related to machine tool homing, see Chapter 4 *Installation and Connection*;

Note 5: After the machine tool homing operation is performed, the original workpiece coordinate system is reset and needs to be reset with G50.

8.3 Other Operations in Homing Mode

1. Press the 

key to rotate the spindle counterclockwise;

2. Press the 

key to stop the spindle;

3. Press the 

key to rotate the spindle clockwise;

4. Press the 

key to switch coolant on/off;

5. Lubrication control (see Chapter 3 of this part for details);

6. Press the 

key for manual relative tool change;

7. Adjustment of spindle override;

8. Adjustment of rapid override;

9. Adjustment of feed rate override.

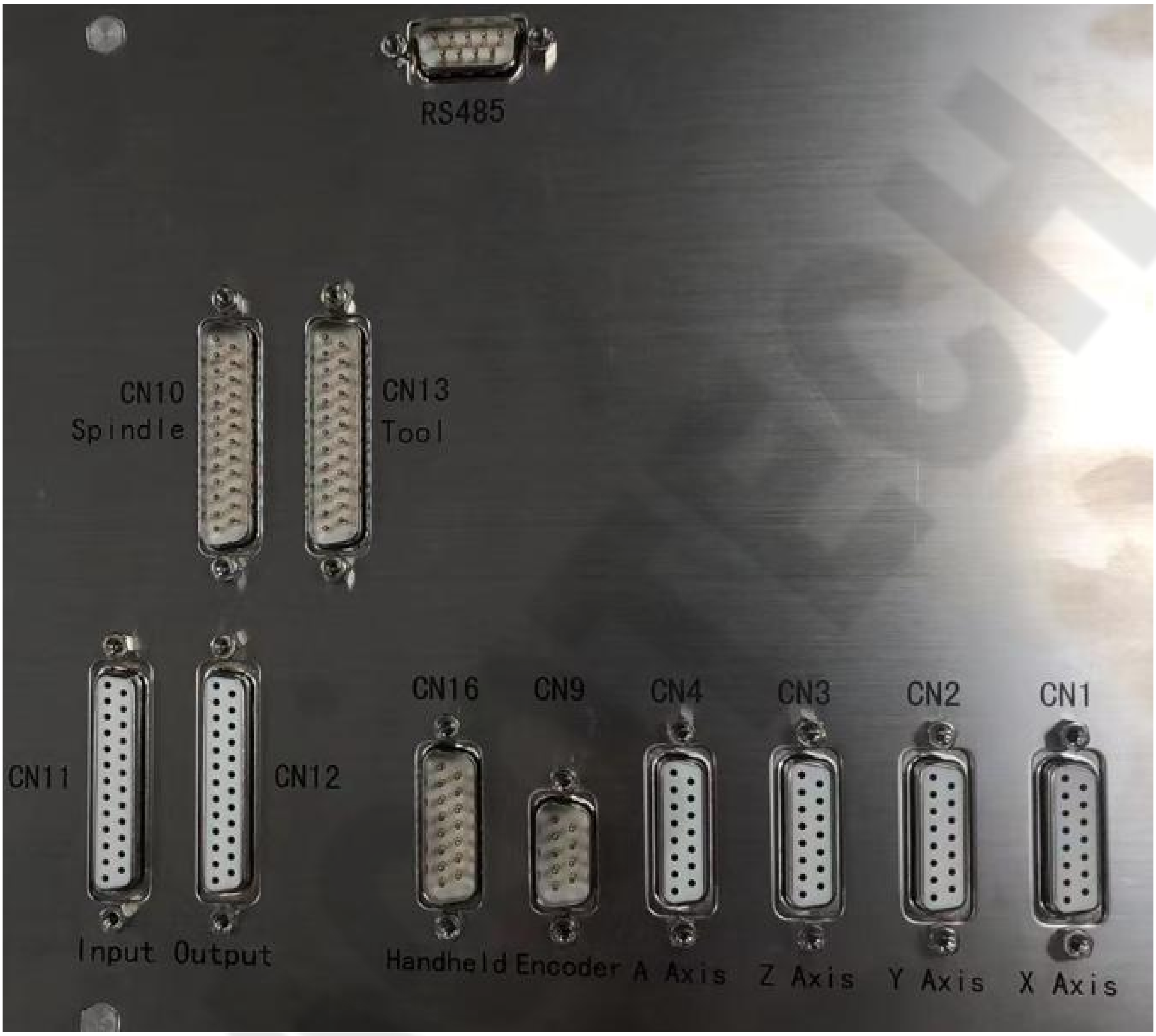
Part 3

Installation and Connection Instructions

Chapter 1 Installation Layout

1.1 System Connection

1.1.1 Rear Cover Interface Layout



1.1.2 Interface Description

◎ XYZA axis: 15-pin D-type hole socket, connected to the axis drive unit

◎ Servo spindle: 25-pin D-type hole socket, connected to the servo spindle drive unit

◎ Encoder: 15-pin D-type pin socket, connected to the spindle encoder

◎ Sub-panel: 15-pin D-type pin socket, connected to the handwheel

◎ Analog: 9-pin D-type pin socket, connected to the spindle drive unit

◎ Input 1: 25-pin D-type pin socket, connected to the machine tool input

◎ Input 2: 25-pin D-type pin socket, connected to the machine tool input

◎ Output 1: 25-pin D-type hole socket, connected to the machine tool output

◎ Output 2: 25-pin D-type hole socket, connected to the machine tool output

1.2 System Installation

1.2.1 Installation Conditions of Electric Cabinets

◎Electric cabinets must be able to effectively prevent the entry of dust, coolant and organic solution;

◎When designing electric cabinets, the distance between the CNC back cover and the chassis should not be less than 20cm. It is necessary to consider that when the temperature inside the electric cabinet rises, the temperature difference between inside and outside of the cabinet must not exceed 10°C;

◎To ensure effective heat dissipation, it is advisable to install a fan in the electric cabinet;

◎The display panel must be installed in a place where can't be sprayed by the coolant;

◎When designing electric cabinets, it is necessary to consider minimizing external electrical interference to prevent interference from being transmitted to the CNC.

1.2.2 Methods to Prevent Interference

CNC has taken anti-interference measures such as shielding space electromagnetic radiation, absorbing impact current, and filtering power supply noise during design, which can prevent the influence of external interference sources on the CNC itself to a certain extent. In order to ensure the stable operation of CNC, it is necessary to take the following measures when installing and connecting CNC:

1. CNC should be kept away from interference-generating equipment (such as inverters, AC contactors, electrostatic generators, high-voltage generators, and segmentation devices of power lines, etc.).

2. To power the CNC through an isolation transformer, the machine tool where the CNC is installed must be grounded, and the CNC and drive unit must be connected to independent grounding wires from the grounding point.

3. Suppress interference: connect an RC circuit in parallel at both ends of the AC coil (as shown in Figure 1-3). The RC circuit should be installed as close to the inductive load as possible; connect a freewheeling diode in reverse parallel at both ends of the DC coil (as shown in Figure 1-4); and connect a surge absorber in parallel at the winding end of the AC motor (as shown in Figure 1-5).

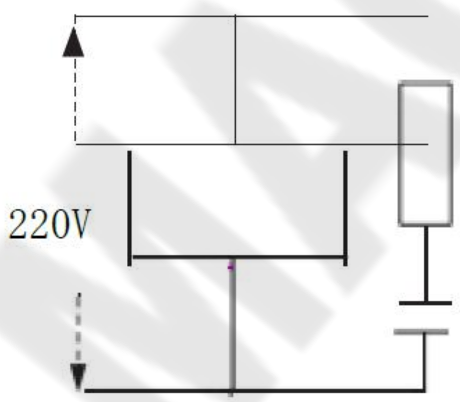


Figure 1-3

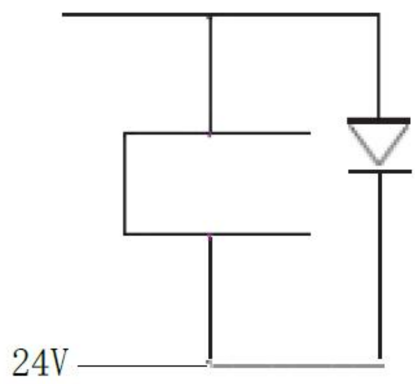


Figure 1-4

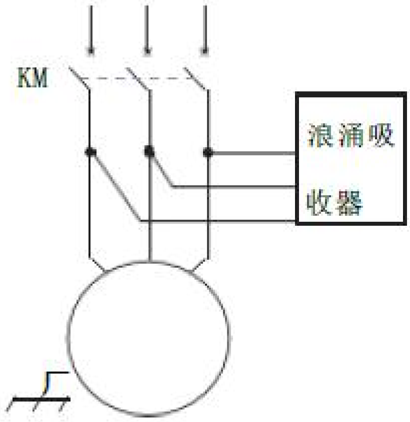


Figure 1-5

4. The lead-out cable of the CNC uses a twisted shielded cable or a shielded cable. The shielding layer of the cable is single-ended grounded on the CNC side, and the signal line should be as short as possible.

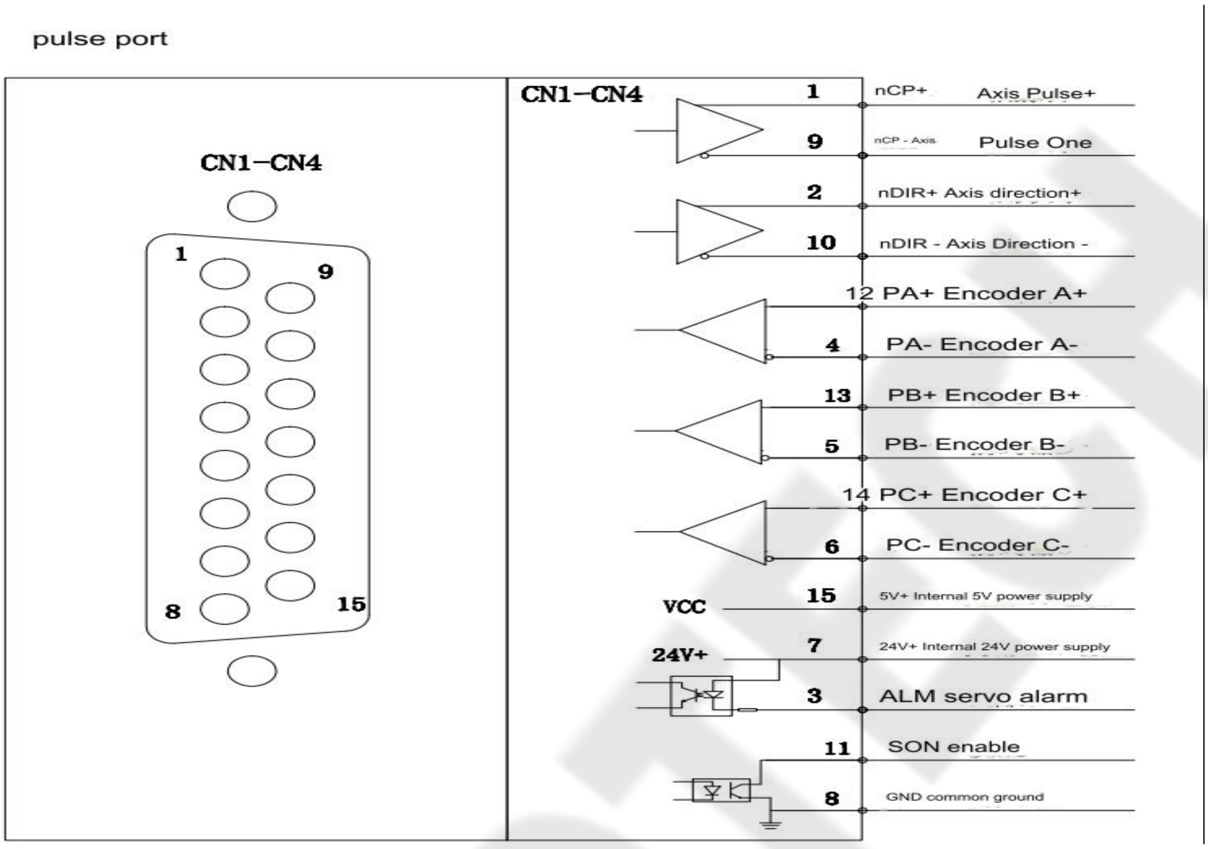
5. In order to reduce mutual interference between CNC signal cables and between CNC signal cables and high-voltage cables, the following principles should be followed when wiring:

Group	Cable type	Wiring requirements
A	AC power cable	Bundle the cables of group A separately from those of group B and group C, and keep the distance between them at least 10cm, or electromagnetically shield the cables of group A
	AC coil	
	AC contactor	
B	DC coil (DC24V)	Bundle the cables of group B separately from those of group A or shield the cables of group B; the farther the cables of group B and group C are, the better
	DC relay (DC24V)	
	Cables between CNC and power cabinet	
	Cables between CNC and machine tool	
C	Cables between CNC and servo drive unit	Bundle the cables of group C separately from those of group A, or shield the cables of group C. The distance between the cables of group C and group B is at least 10cm, and the cables are twisted pair
	Position feedback cable	
	Position encoder cable	
	Handwheel cable	
	Other shielded cables	

Chapter 2 Definition and Connection of Interface Signals

2.1 Connection with Drive Unit

2.1.1 Definition of Drive Interface



2.1.2 Instruction Pulse Signal and Instruction Direction Signal

CP+, CP- are instruction pulse signals, DIR+, DIR- are instruction direction signals, both sets of signals are differential (AM26LS31) output, external AM26LS32 is recommended for reception, the internal circuit is shown in Figure 2-2:

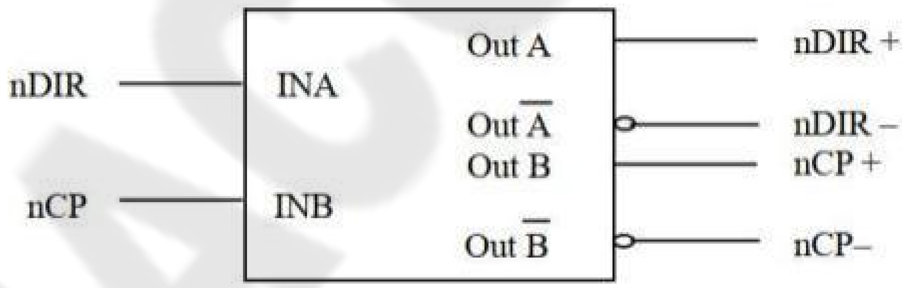


Figure 2-2 Internal circuit of instruction pulse signal and instruction direction signal

2.1.3 Drive Unit Alarm Signal nALM

The CNC parameter N0176 sets the drive unit alarm level to be low or high. The internal circuit is shown in Figure 2-3:

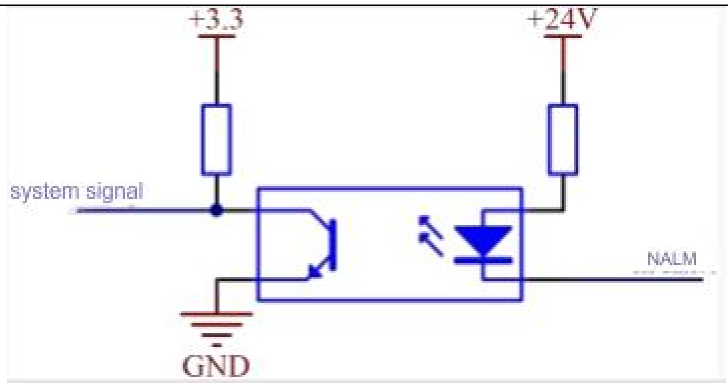


Figure 2-3 Internal circuit of drive unit alarm signal

This type of input circuit requires the drive unit to provide signals in the manner shown in Figure 2-4 below:

Method 1: Method 2:

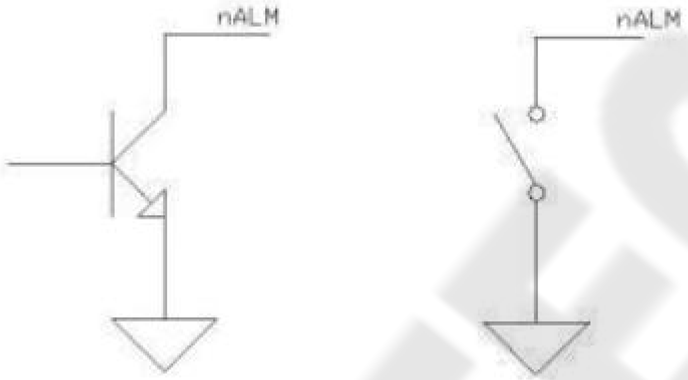


Figure 2-4 Ways of drive unit providing signals

2.1.4 Axis Enable Signal nEN

When CNC works normally, the nEN signal output is valid (nEN signal is connected to 0V). When the drive unit alarms, CNC turns off the nEN signal output (nEN signal is disconnected from 0V). The internal interface circuit is shown in Figure 2 -5 below:

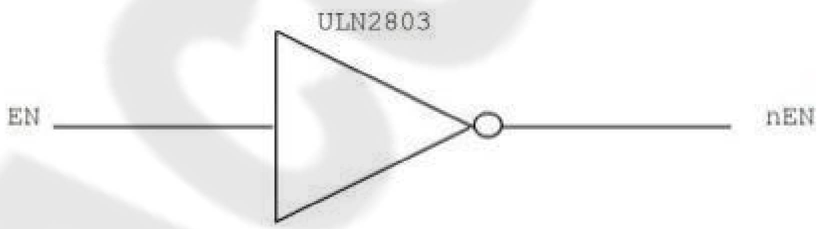


Figure 2-5 Internal interface circuit of axis enable signal

2.1.5 Home Signal nPC

For machine tool homing, the one-turn signal of the motor encoder or the proximity switch signal is used as the home signal. The internal connection circuit is shown in Figure 2-7 below:

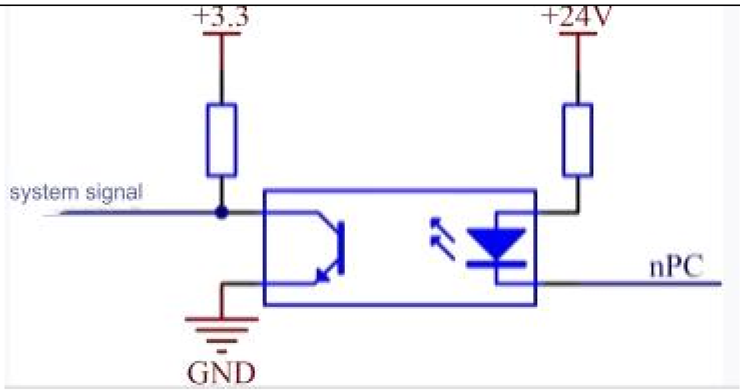
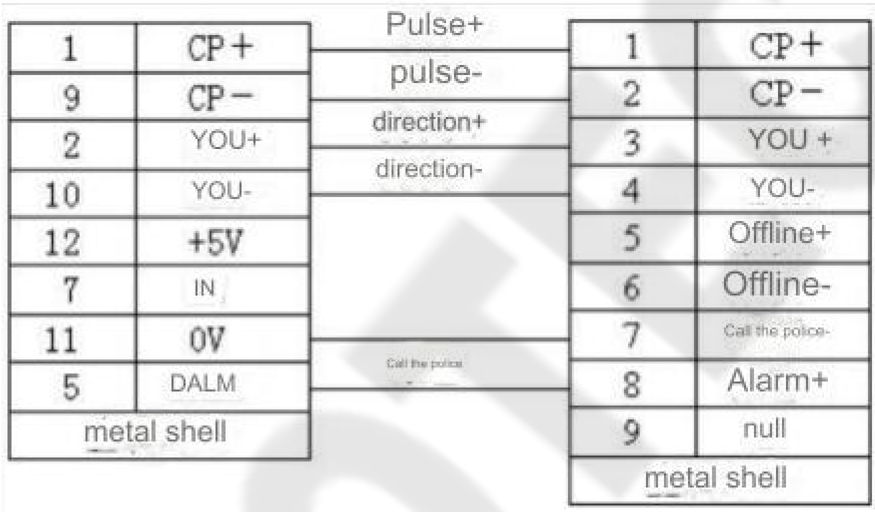


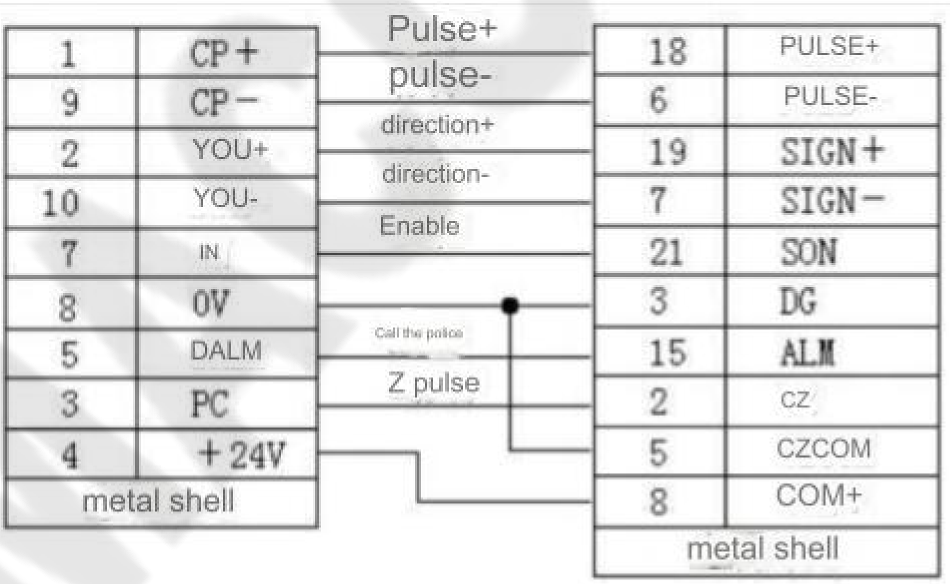
Figure 2-7 Home signal circuit

2.1.6 Connection with Drive Unit

The connection with stepper drive unit is shown in Figure 2-11 below:

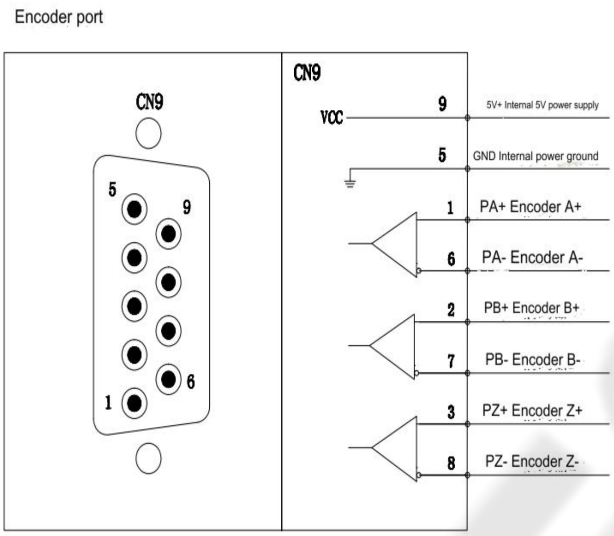


Connection with servo drive



2.2 Connection with Spindle Encoder

2.2.1 Definition of Spindle Encoder Interface



2.2.2 Signal Description

*PCS/PCS, *PBS/PBS, *PAS/PAS are the differential input signals of encoder phase C, phase B, and phase A, respectively, and are received by 26LS32; *PAS/PAS, *PBS/PBS are orthogonal square waves with a phase difference of 90, and the highest signal frequency is <1MHz; the number of lines of the encoder used is set by the parameter (range 100~5000).

The internal connection circuit is shown in Figure 2-13: (n=A, B, C in the figure)

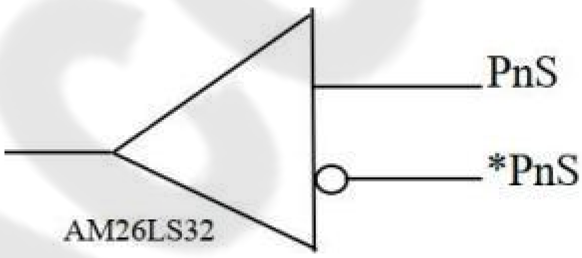
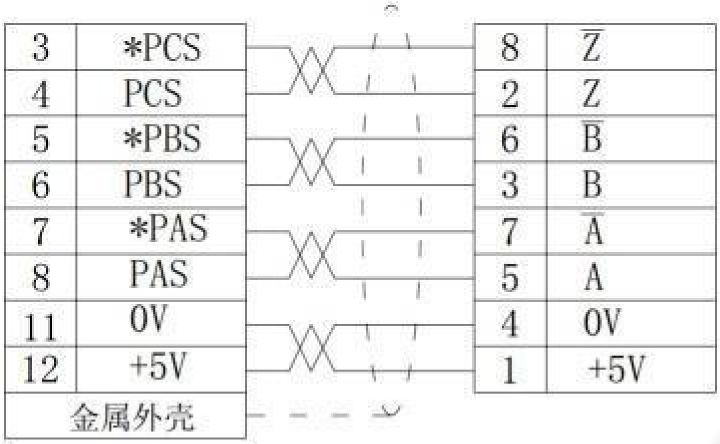


Figure 2-13 Encoder signal circuit

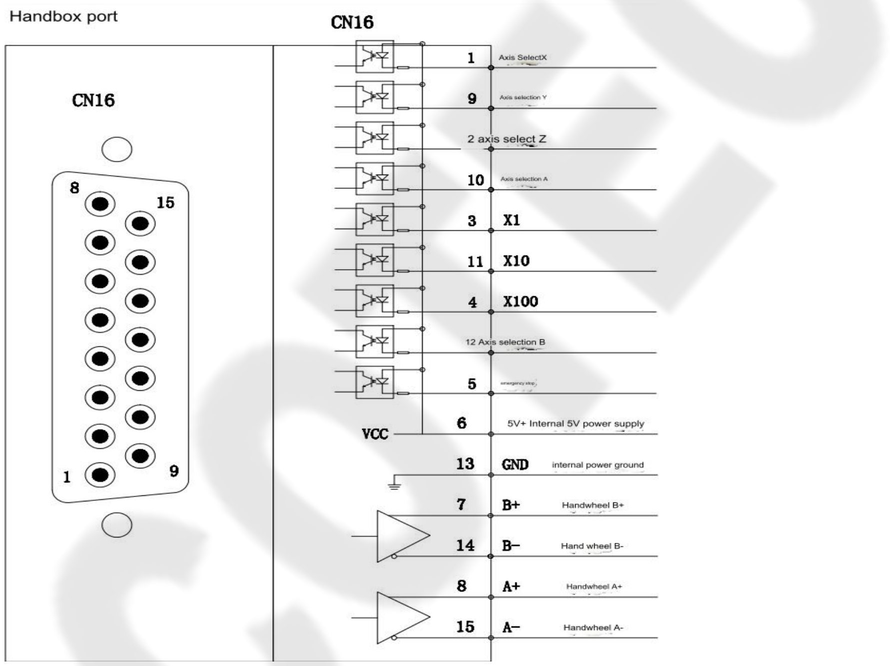
2.2.3 Spindle Encoder Interface Connection

The connection between the SZGH880T/SZGH1080T series and the spindle encoder is shown in Figure 2-14. Twisted pair cables are used for connection.

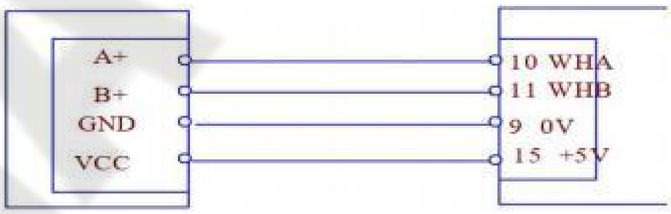


2.3 Connection with Handwheel

2.3.1 Definition of Sub-panel Interface

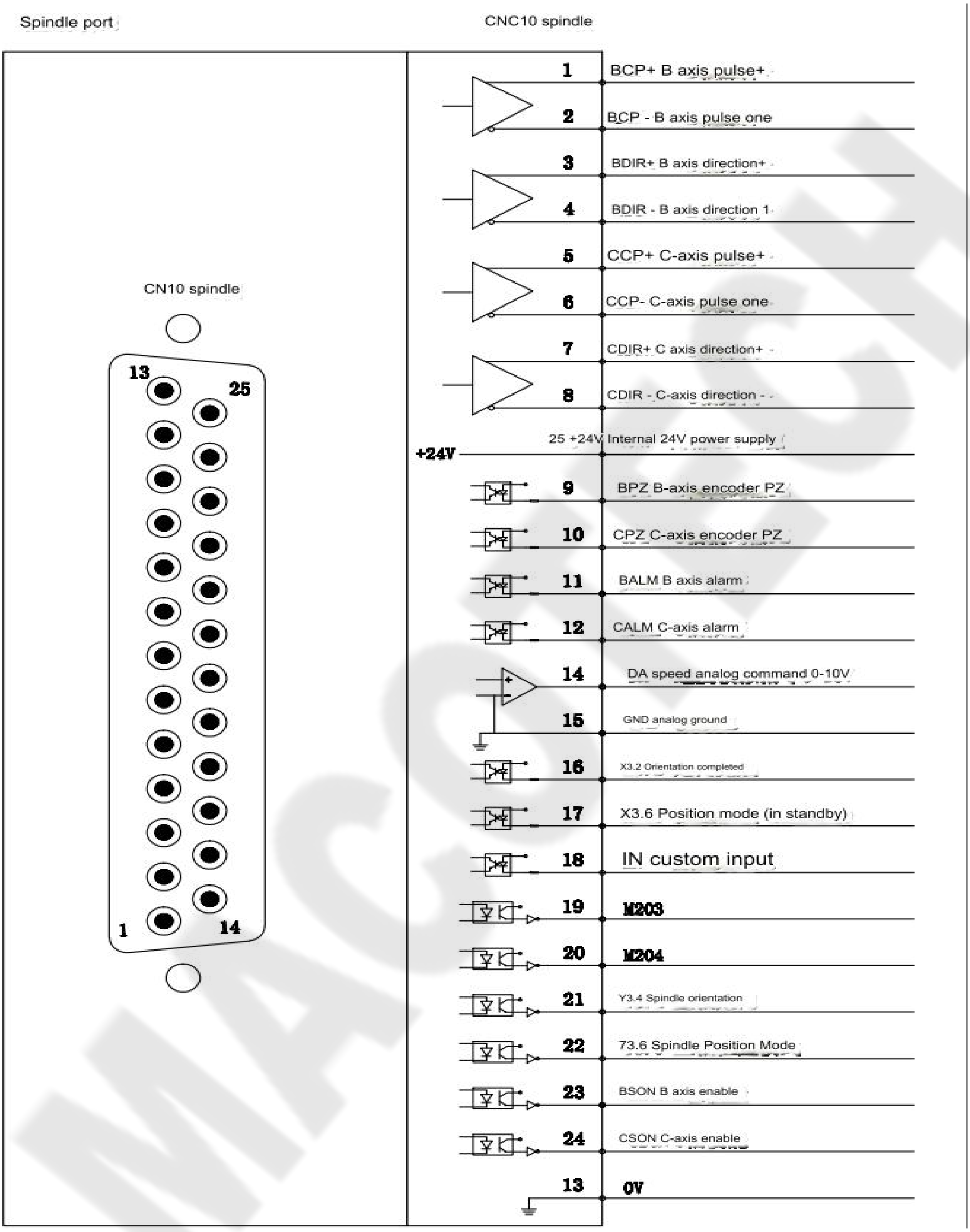


2.3.2 Wiring Diagram of Handwheel and CNC System

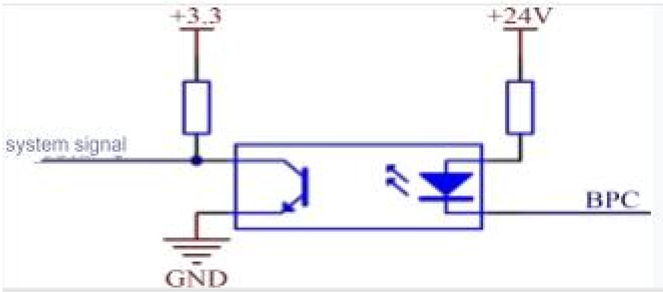


Depending on the output signal mode, there are generally two types of handwheels: two-signal line type (A+, B+ signals) and four-signal line type (A+, A-, B+, B-). For four-signal line handwheels, A- and B- signals are not connected.

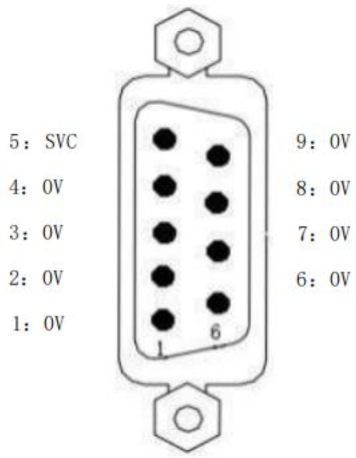
2.4 Spindle Interface Definition



Note: The internal circuit of BPC signal is shown in the figure below:



2.5 Analog Interface



Pin number	Signal name	Signal description
5	SVC	Analog 1
Other pins	0V	Ground

The analog spindle interface SVC terminal can output 0 ~ 10V voltage, and the internal circuit of the signal is shown in Figure 2-20 below:

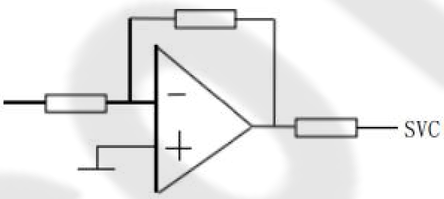


Figure 2-20 SVC signal circuit

The connection with the inverter is shown in Figure 2-21:

SZGH880T/SZGH1080T series Inverter

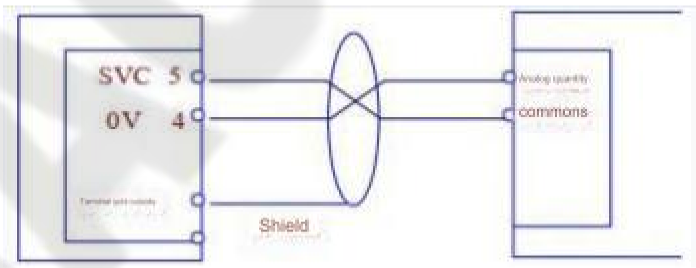


Figure 2-21 Connection with inverter

2.6 I/O Interface Definition

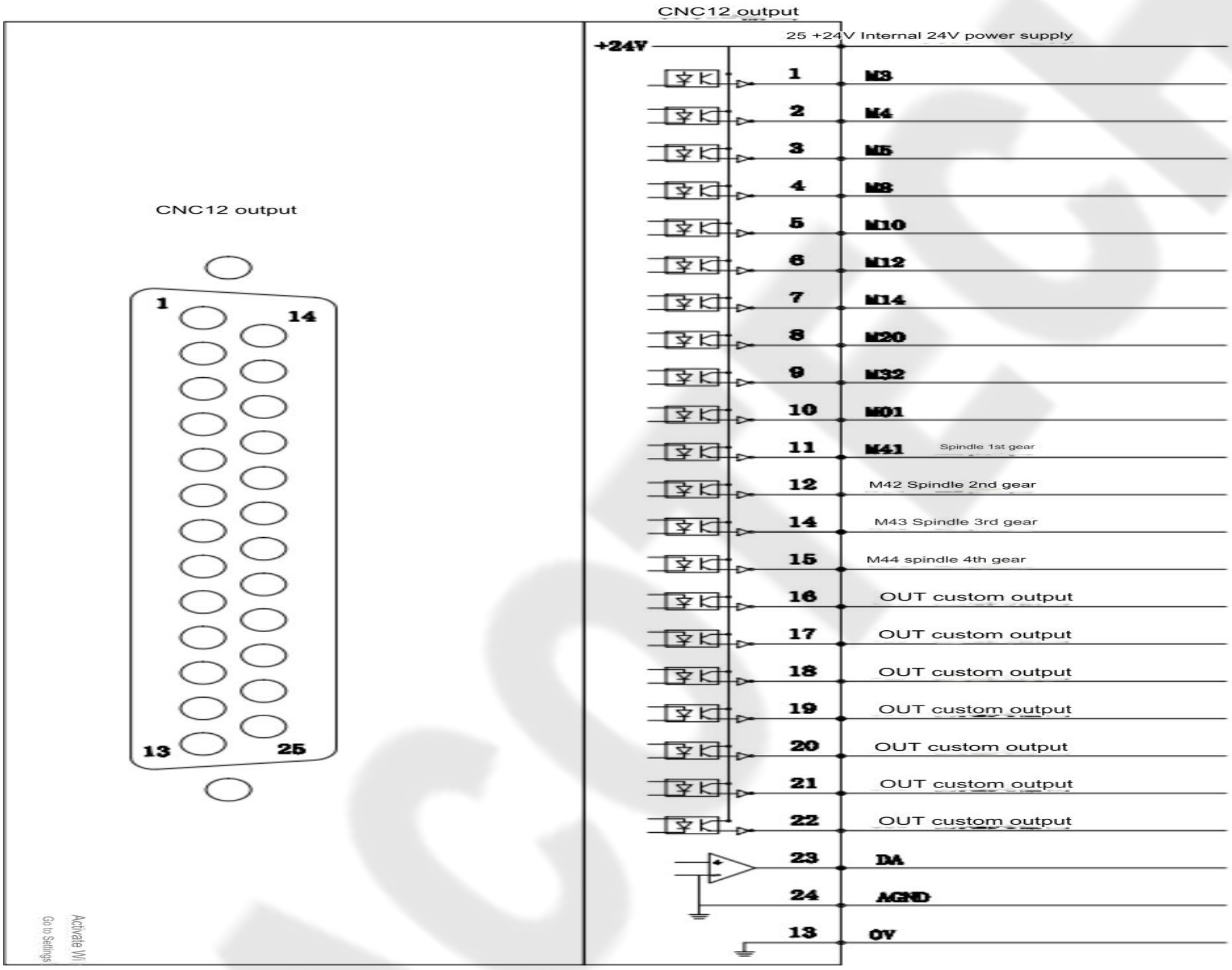
Note!

The meaning of the I/O functions without fixed addresses marked on the CNC of SZGH880T/SZGH1080T series lathes is defined by the PLC program (ladder diagram). When the CNC of SZGH880T/SZGH1080T series lathes is assembled on the machine tool, the I/O functions are designed and determined by the machine tool manufacturer. For details, please refer to the manual of the machine tool.

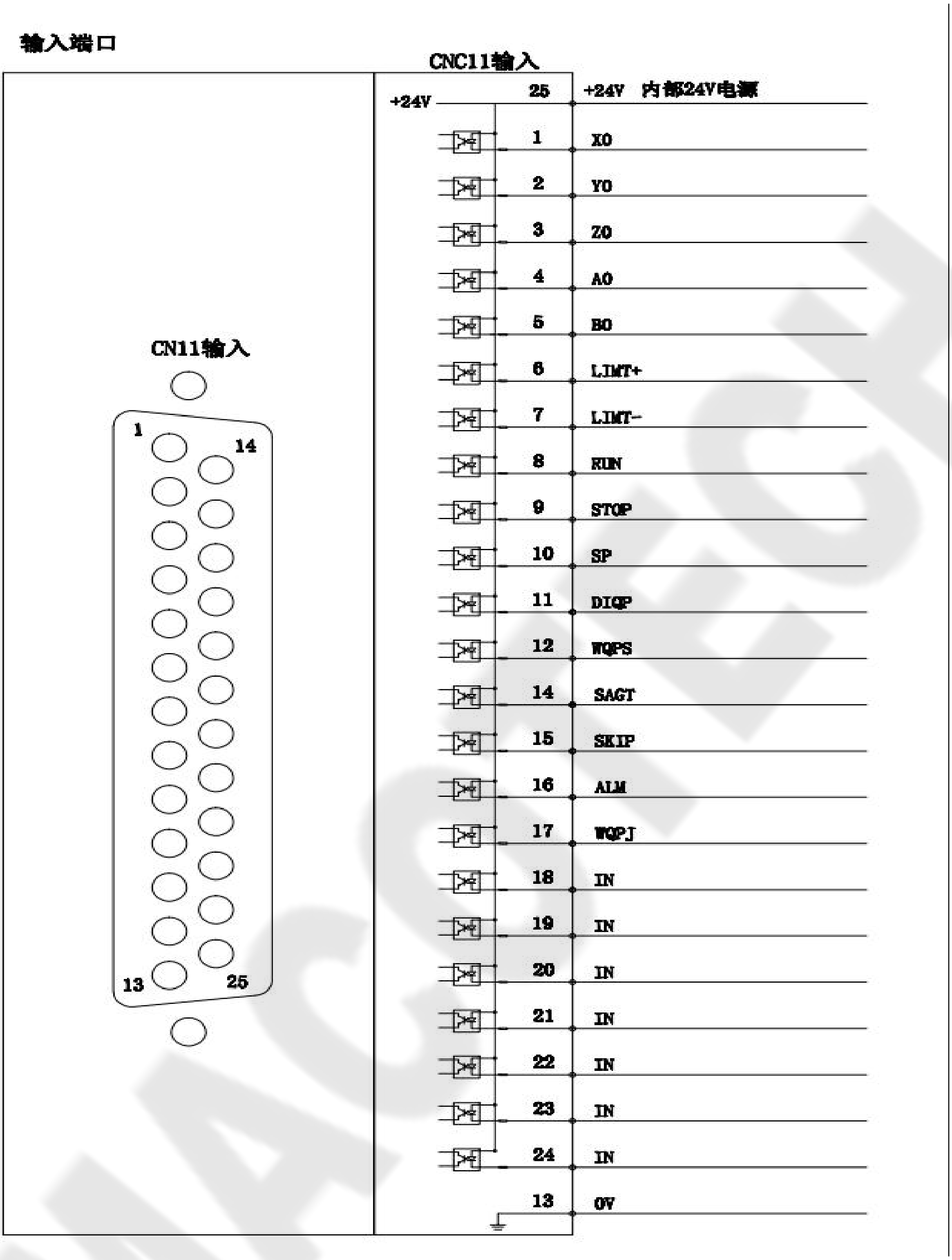
The I/O functions without fixed addresses marked in this section are described for standard PLC programs.

Input 1 pin definition

Output port



Input 2 pin definition



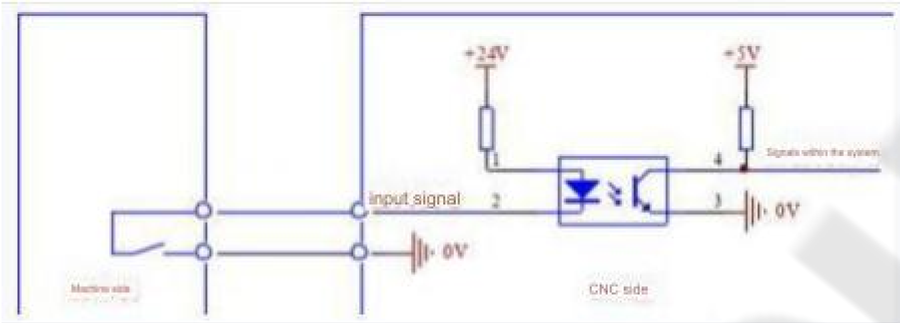
2.6.1 Input Signal

All input ports are photoelectrically isolated from the internal circuit of the system. The electrical specifications of each input port are as follows:

(1) Photoelectric isolation circuit, maximum isolation voltage 2500VRMS

(2) Input voltage range DC 0V~24V

The electrical schematic diagram of the input port is as follows:



2.6.2 Output Signal

The output signal is used to drive the relay and indicator light on the machine tool side. When the output signal is connected to 0V, the output function is valid; when it is disconnected from 0V, the output function is invalid. There are 32 digital outputs in the I/O interface, all with the same structure, as shown in Figure 2-29:

CNC side Machine tool side

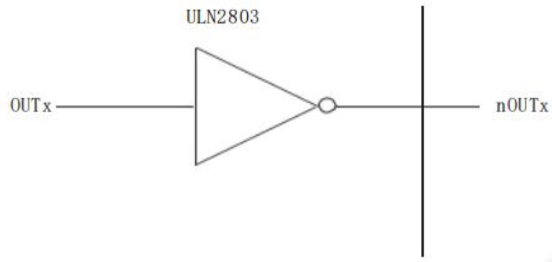


Figure 2-29 Digital output module circuit structure diagram

The logic signal OUTx output by the mainboard is sent to the input end of the inverter (ULN2803) through the connector. nOUTx has two output states: 0V output or high impedance. Typical applications are as follows:

Driving light-emitting diodes

Using ULN2803 output to drive light-emitting diodes, a resistor needs to be connected in series to limit the current flowing through the light-emitting diode (usually about 10mA).

As shown in Figure 2-30 below:

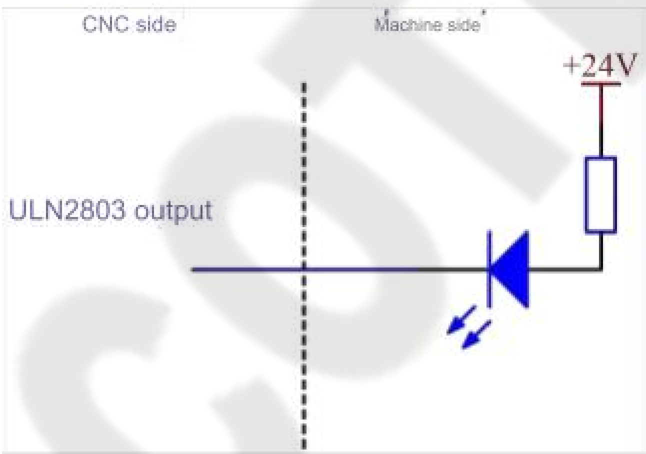


Figure 2-30

Driving filament indicator lights

Using ULN2803 output to drive filament indicator lights, an external preheating resistor is required to reduce the current impact when conducting. The resistance value of the preheating resistor is based on the principle that the indicator light does not light up, as shown in Figure 2-31 below.

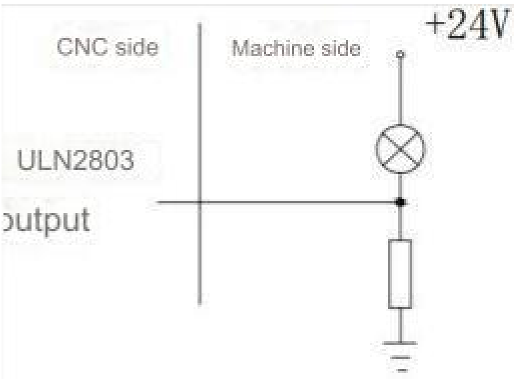


Figure 2-31

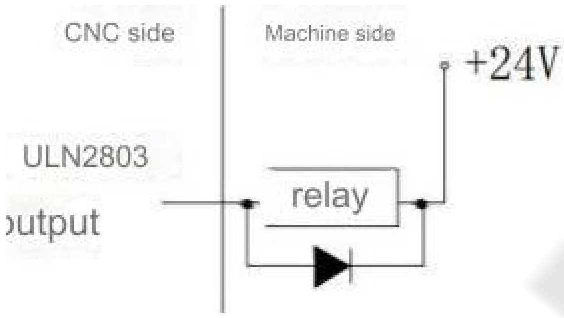


Figure 2-32

Driving inductive loads (such as relays)

Using ULN2803 output to drive inductive loads, it is necessary to connect a freewheeling diode near the coil to protect the output circuit and reduce interference.

As shown in Figure 2-32 above.

The meaning of the output signal in the I/O interface is defined by the PLC program. The output signals defined by the standard PLC program include S1~S4 (M41~M44), M3~M5, M8, M10, M11, M32, TL-, TL+, UO0~UO5, DOQPJ, DOQPS, SPZD, etc.

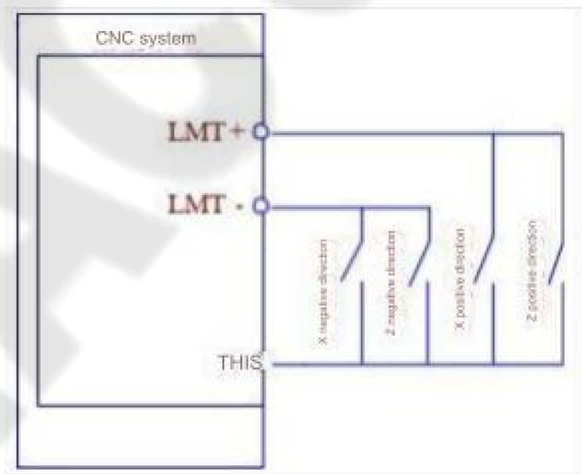
2.7 I/O Function and Connection

2.7.1 LMT+, LMT- Positive and Negative Hardware Limit Signals

LMT+, LMT- signals are low-level valid and are common signals for all axes. When wiring, connect the positive limit signal of each axis to LMT+, and the negative limit signal of each axis to LMT-. The limit switch should be in normally open mode.

When an overtravel alarm is generated, the system stops feeding in that direction and can be manually operated to feed in the reverse direction.

Wiring diagram:



2.7.2 Tool Change Control

Related signals

Signal type	Symbol	Signal function
Input signal	T01	Tool position signal 1
	T02	Tool position signal 2
	T03	Tool position signal 3
	T04	Tool position signal 4
	T05	Tool position signal 5
	T06	Tool position signal 6
	T07	Tool position signal 7
	T08	Tool position signal 8
	TCP	Tool holder locking signal
Output signal	TL+	Tool holder forward signal
	TL-	Tool holder reverse signal

Control parameters

K011			CHET	TCPS	CTCP	TSGN		CHT

CHT =0: Tool change mode selection mode B

=1: Tool change mode selection mode A

TSGN=0: Tool position signal high level is valid

=1: Tool position signal low level is valid

CTCP=0: Do not check tool holder locking signal

=1: Check tool holder locking signal

TCPS=0: Tool holder locking signal low level is valid

=1: Tool holder locking signal high level is valid

CHET=0: Do not check tool position signal at the end of tool change

=1: Check tool position signal at the end of tool change

DT004: Upper limit of the time to move the maximum tool position when changing tools

DT007: Delay time from the forward stop of the tool holder to the reverse output of the tool holder (ms)

DT008: Timeout time of the tool holder locking output

DT009: Reverse locking time of the tool holder

2.7.3 Machine Tool Homing

Related signals

DECX: X-axis deceleration signal;

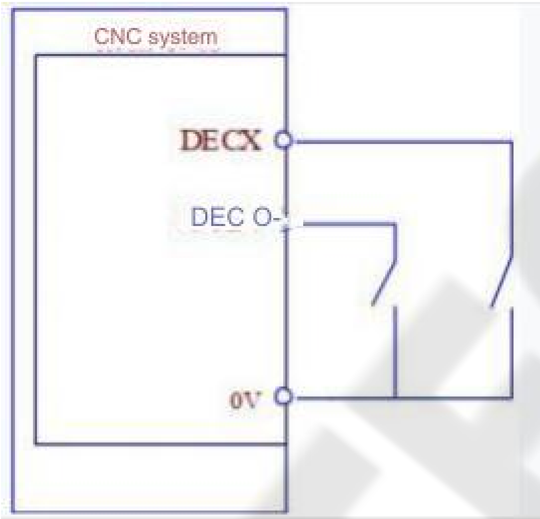
DECY: Y-axis deceleration signal;

DECZ: Z-axis deceleration signal;

DEC4: 4th axis deceleration signal;

DEC5: 5th axis deceleration signal;

Wiring diagram:



Control parameters

K022	DEC4T	DECY	DECZ	DECX				

DEC4T =0: 4th axis deceleration signal low level

=1: 4th axis deceleration signal high level

DECY =0: Y axis deceleration signal low level

=1: Y axis deceleration signal high level

DECZ =0: Z axis deceleration signal low level

=1: Z axis deceleration signal high level

DECX =0: X axis deceleration signal low level

=1: X axis deceleration signal high level

Under the system parameter homing category, you can set the type and speed of homing.

2.7.4 Spindle Control

Related signals (defined by standard PLC program)

Signal type	Symbol	Function description
Input signal	SAR	Spindle speed arrival signal
	SALM	Spindle abnormal alarm input
Output signal	M03	Spindle counterclockwise rotation (forward)
	M04	Spindle clockwise rotation (reverse)
	M05	Spindle stop
	SCLP	Spindle clamping
	SPZD	Spindle braking
	SVF	Spindle servo disconnected

2.7.5 Spindle Speed Switch Control

Related signals (defined by standard PLC program)

S01~S04: Spindle speed switch control signal, the S01~S04 signal interfaces defined by the standard PLC program are multiplexed interfaces, and S01~S04 and M41~M44 share the interface. Parameter N0105 sets the switch control.

Control logic (defined by standard PLC program)

When CNC is powered on, the output of S1~S4 is invalid. When any code among S01, S02, S03, and S04 is executed, the corresponding S signal output is valid and maintained, and the output of the other S signals is canceled. When the S00 code is executed, the output of S1~S4 is canceled, and only one output of S1~S4 is valid at the same time.

2.7.6 Spindle Automatic Shift Control

Related signals (defined by standard PLC program)

M41~M44: Spindle automatic shift output signal, when the spindle analog control (0~10V analog voltage output) is selected, it can support 4-gear spindle automatic shift control.

M41I, M42I: Spindle automatic 1st or 2nd gear shift in place signal, which supports 2 gear shift in place detection function.

Control parameters

K015					SHT	AGIM	AGIN	AGER

AGER=1: Spindle automatic shift function is valid;

=0: Spindle automatic shift function is invalid.

AGIN=1: When the spindle automatically shifts to 1st or 2nd gear, check the shift in place signals M41I and M42I;

=0: When the spindle automatically shifts to 1st or 2nd gear, do not check the shift in place signals M41I and M42I.

AGIM=1: Valid when the gear shift in place signal M41I, M42I is disconnected from +24V;

=0: Valid when the gear shift in place signal M41I, M42I is connected to +24V.

SHT =1: Spindle gear position is memorized after power off;

=0: Spindle gear position is not memorized after power off.

2.7.7 External Cycle Start and Feed Hold

Related signals (defined by standard PLC program)

ST: External automatic cycle start signal, which has the same function as the automatic cycle start key in the machine tool panel.

SP: External feed hold signal, which has the same function as the feed hold key in the machine tool panel.

2.7.8 Cooling Pump Control

Related instruction signals (defined by standard PLC program)

Signal type	Symbol	Function description
Output signal	M08	Cooling pump control output
Instruction format	M08	Coolant on
	M09	Coolant off

Signal connection

The internal circuit is shown in Figure 2-50 below:

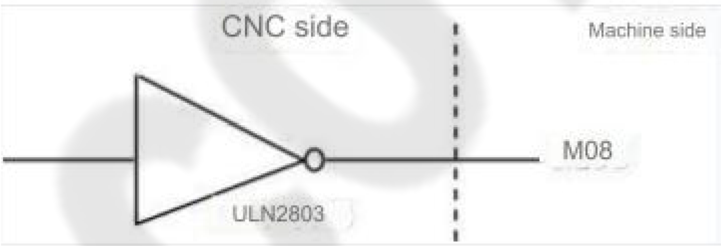


Figure 2-50

Function description (defined by standard PLC program)

After CNC is powered on, M09 is valid, that is, M08 output is invalid. Execute M08, M08 output is valid, and the cooling pump is on; execute M09, cancel M08 output, and the cooling pump is off.

2.7.9 Lubrication Control

Related instruction signals (defined by standard PLC program)

Signal type	Symbol	Function description
Output signal	M32	Lubrication control output
Instruction format	M32	Lubrication on
	M33	Lubrication off

Signal connection

The internal circuit is shown in Figure 2-51 below:

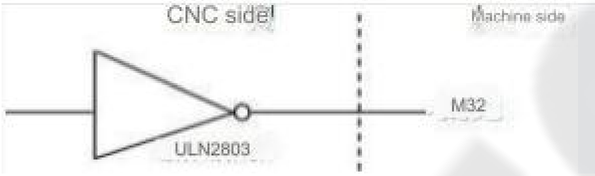


Figure 2-51

2.7.10 Chuck Control

Related signals (defined by standard PLC program)

DIQP: Chuck control input signal

DOQPJ: Internal chuck clamping output/external chuck release output signal

DOQPS: Internal chuck release output/external chuck clamping output signal

NQPJ: Internal chuck clamping in place/external chuck release in place signal

WQPJ: Internal chuck release in place/external chuck clamping in place signal

Control parameters

K012					CCHU	NYQP	SLSP	SLQP

SLQP=1: Chuck control function is valid;

=0: Chuck control function is invalid.

SLSP=1: When the chuck function is valid, do not check whether the chuck is clamped;

=0: When the chuck function is valid, check whether the chuck is clamped. If the chuck is not clamped, the spindle cannot be started and an alarm is generated.

NYQP=1: External chuck mode, NQPJ is the external chuck loose signal, WQPJ is the external chuck tight signal;

=0: Internal chuck mode, NQPJ is the internal chuck tight signal, WQPJ is the internal chuck loose signal.

CCHU=1: Check the chuck in place signal;

=0: Do not check the chuck in place signal.

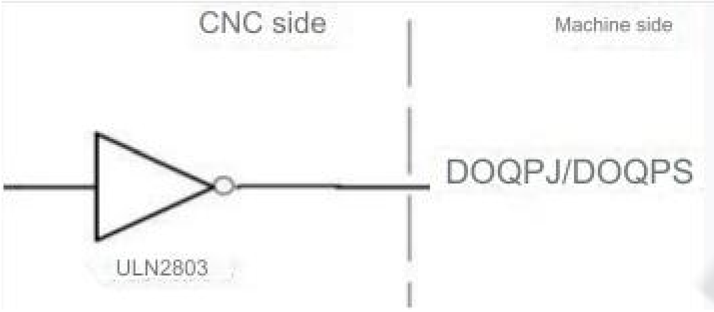
DT018	

DT18>0: Chuck clamping and releasing signals are pulse outputs, and the pulse width is set by DT18

=0: Chuck clamping and releasing signals are level outputs

Signal connection

The DOQPJ/DOQPS circuit is shown in Figure 2-52 below:



Action sequence

①When SLQP=1, SLSP=0, NYQP=0, CCHU=1, CNC selects the internal chuck mode, and the chuck in-place signal detection function is effective:

DOQPS: Chuck release output; WQPJ: Release in-place signal;

DOQPJ: Chuck clamping output; NQPJ: Clamp in-place signal.

When the power is turned on, DOQPJ and DOQPS both output high impedance. When CNC detects that the chuck control input signal DIQP is valid for the first time, DOQPJ is connected to 0V and the chuck is clamped.

After executing M12, DOQPS outputs high impedance, DOQPJ outputs 0V, the chuck is clamped, and the CNC waits for the NQPJ signal to be in place;

After executing M13, DOQPJ outputs high impedance, DOQPS outputs 0V, the chuck is released, and the CNC waits for the WQPJ signal to be in place.

②When SLQP=1, SLSP=0, NYQP=1, CCHU=1, CNC selects the external chuck mode, and the chuck in place signal detection function is effective:

DOQPS: Chuck clamping output. WQPJ: Clamping in place signal

DOQPJ: Chuck release output. NQPJ: Release in place signal.

When the power is turned on, DOQPJ and DOQPS both output high impedance. When CNC detects that the chuck control input signal DIQP is valid for the first time, DOQPS is connected to 0V and the chuck is clamped.

After executing M12, DOQPS outputs 0V, DOQPJ outputs high impedance, the chuck is clamped, and the CNC waits for the WQPJ signal to be in place;

After executing M13, DOQPJ outputs 0V, DOQPS outputs high impedance, the chuck is released, and the CNC waits for the NQPJ signal to be in place.

When the chuck control input is valid for the second time, DOQPS outputs 0V, the chuck is released, and the chuck clamping/releasing signals are interlocked and output alternately, that is, each time the chuck control input signal is valid, its output state changes once.

③ Interlocking relationship between chuck and spindle:

When SLQP=1, SLSP=0, M3 or M4 is valid, executing M13 generates an alarm and the output status remains unchanged;

When SLQP=1, SLSP=0, CCHU=1, executing M12 code in MDI or automatic mode, CNC will not execute the next code before CNC detects that the chuck is clamped in place and effective. When the chuck control input signal DIQP is valid in manual mode, the panel spindle forward and reverse keys are invalid before CNC detects that the chuck is clamped in place and effective. When the spindle is rotating or during automatic cycle processing, the DIQP signal input is invalid; the output status of DOQPS and DOQPJ remains unchanged when CNC resets or stops urgently.

2.7.11 Tailstock Control

Related signals (defined by standard PLC program)

DOTWJ: Tailstock forward output signal

DOTWS: Tailstock backward output signal

DITW: Tailstock control input signal

Control parameters

K013							SPTW	SLTW

SLTW=1: Tailstock control function is valid;

=0: Tailstock control function is invalid.

SPTW=1: Spindle rotation and tailstock forward/backward are not interlocked. No matter what state the spindle is in, the tailstock can forward/backward; no matter what state the tailstock is in, the spindle can rotate;

=0: Spindle rotation and tailstock forward/backward are interlocked. When the spindle rotates, the tailstock cannot exit; when the tailstock is not advancing, the spindle cannot be started.

Signal connection

The tailstock control signal circuit is shown in Figure 2-55 below:

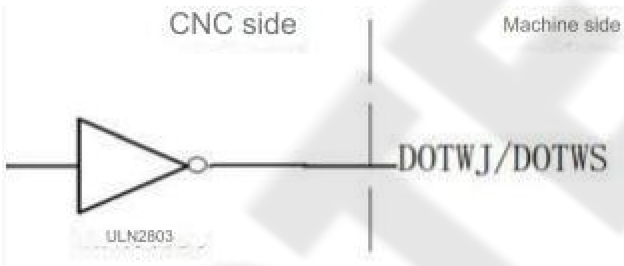


Figure 2-55

Action sequence (defined by standard PLC program)

When the power is turned on, the tailstock forward DOTWJ and the tailstock backward DOTWS are both invalid; when the first tailstock control input DITW is valid, the tailstock forward is valid; when the second tailstock control input is valid, the tailstock backward is valid, and the tailstock forward /tailstock backward signals are interlocked and output alternately, that is, each time the tailstock control input signal is valid, the output state changes once. After executing code M10, DOTWJ outputs 0V, and the tailstock advances; after executing code M11, DOTWS outputs 0V, and the tailstock retracts.

When the spindle rotates, the tailstock control input signal is invalid, and its output state remains unchanged; the output states of DOTWS and DOTWJ remain unchanged when the CNC is reset or emergency stopped.

2.7.12 Guard Door Detection

Related signals (defined by standard PLC program)

SAGT: Guard door detection input signal.

Control parameters

K014					SPB4	PB4		

PB4 =0: Guard door detection function is invalid;
=1: Guard door detection function is valid.
SPB4=0: When SAGT is low level, the guard door is closed;
=1: When SAGT is high level, the guard door is closed.

Function description (defined by standard PLC program)

① The guard door detection function is valid in automatic mode, but when the guard door is open, a warning prompt of "Guard door is open" will be given in all modes, which will not affect the execution of other functions;

② In automatic mode, if the CNC detects that the guard door is open when the automatic cycle is started, an alarm will be generated;

③ During automatic operation, if the CNC detects that the guard door is open, the axis feed will be paused, the spindle and cooling output will be turned off;

2.7.13 Segment Skip

If you do not want to execute a certain segment in the program and do not want to delete the segment, you can select the segment skip function. When the program segment has a "/" sign at the beginning and the program segment skip switch is turned on (the machine tool panel key or program skip external input is valid), this program segment will be skipped and not run during automatic operation.

Related signals (defined by standard PLC program)
AEY/BDT: program segment skip signal.

2.7.14 CNC Macro Variables

Related signals
Macro output signal: Standard PLC defines 5 #1100 ~ #1105 macro output ports;
Macro input signal: Standard PLC defines 16 #1000 ~ #1015 macro input ports.

Signal diagnosis

Macro variable number	#1105	#1104	#1103	#1102	#1101	#1100
Diagnostic address	Y3.7	Y3.6	Y3.5	Y3.4	Y3.3	Y3.2

Macro variable number	#1007	#1006	#1005	#1004	#1003	#1002	#1001	#1000
Diagnostic address	X0.7	X0.6	X0.5	X0.4	X0.3	X0.2	X0.1	X0.0

Macro variable number	#1015	#1014	#1013	#1012	#1011	#1010	#1009	#1008
Diagnostic address	X1.7	X1.6	X1.5	X1.4	X1.3	X1.2	X1.1	X1.0

Function description (defined by standard PLC program)

Assigning values to macro variables #1100 ~ #1105 can change the output signal status of UO0 ~ UO5; when assigned a value of "1", the output is 0V; when assigned a value of "0", the output signal is turned off.

The input status of input interfaces X0.0 ~ X0.7, X1.0 ~ X1.7 can be known by detecting the values of macro variables #1000 ~ #1015.

2.7.15 Tri-color Light

Related signals and function definitions (defined by standard PLC program)

Y3.1: Yellow light, indicating normal state (non-operating, non-alarm state)

Y2.7: Green light, indicating operating state

Y2.6: Red light, indicating alarm state

Chapter 3 Machine Tool Commissioning Methods and Steps

This chapter introduces the trial operation method and steps of the SZGH880T/SZGH1080T CNC system when it is powered on for the first time. After commissioning according to the following steps, the corresponding machine tool operation can be performed.

3.1 Drive Unit Setting

Set parameter N0176 according to the alarm logic level of the drive unit.

If the machine tool movement direction is inconsistent with the instruction required direction, modify parameter N0177. The manual movement direction can be changed by parameter N0178.

3.2 Gear Ratio Adjustment

The system provides two setting methods. The first method uses the numerator and denominator method (parameters N0007 and N0008); the second method only uses the parameter setting without calculation (parameters N0009~N0012), but N0007 and N0008 must be set to 0.

Numerator and denominator calculation method:

When the machine tool moving distance is inconsistent with the displacement distance displayed by the CNC coordinates, the parameters N0007 and N0008 can be modified to adjust the electronic gear ratio to adapt to different mechanical transmission ratios. Calculation formula:

$$\frac{CMR}{CMD} = \frac{\delta \times 360}{\alpha \times L} \times \frac{Z_M}{Z_D}$$

CMR: numerator of electronic gear ratio

CMD: denominator of electronic gear ratio

α: pulse equivalent, the angle that the motor rotates when receiving a pulse L: lead of the screw

δ: the current minimum input unit of CNC ZM: the number of teeth of the gear at the screw end ZD: the number of teeth of the gear at the motor end

Example: the number of teeth of the gear at the screw end is 50, the number of teeth of the gear at the motor end is 30, the pulse equivalent α =0.075 degrees, and the lead of the screw is 4 mm;

X, Z axis electronic gear ratio:

$$\frac{CMR}{CMD}=\frac{\delta\times360}{\alpha\times L}\times\frac{Z_M}{Z_D}=\frac{0.001\times360}{0.075\times4}\times\frac{50}{30}=\frac{2}{1}$$

Then parameter N0007=2, N0008=1.

When the numerator of the electronic gear ratio is greater than the denominator, the maximum speed allowed by the CNC will decrease.

When the numerator and denominator of the electronic gear ratio are not equal, the positioning accuracy of the CNC may decrease. Example: when parameters N0007=1, N0008=5, no pulse is output when the input increment is 0.004, and one pulse is output when the input increment reaches 0.005. When matching stepper drive, try to use a drive unit with step subdivision function, choose the mechanical transmission ratio reasonably, and set the CNC electronic gear ratio to 1:1 to avoid the numerator and denominator of the CNC electronic gear ratio being too different.

3.3 Backlash Compensation

The backlash compensation value is the actual measured clearance value as the input value. The unit is mm (metric machine tool) or inch (inch machine tool). You can use a dial indicator, micrometer or laser detector to measure. The backlash compensation must be accurately compensated to improve the processing accuracy. Therefore, it is not recommended to use a handwheel or single-step method to measure the screw backlash. It is recommended to measure the backlash as follows:

Edit the program (take the Z axis as an example):

```
O0001;
N10  G01  W10 F800;
N20 W15;
N30  W1;
N40 W-1;
N50 M30.
```

The backlash error compensation value should be set to zero before measurement;

Single-stage run program, locate twice and find the measurement reference A, record the current data, then run 1mm in the same direction, and then run 1mm in the opposite direction to point B, and read the current data.

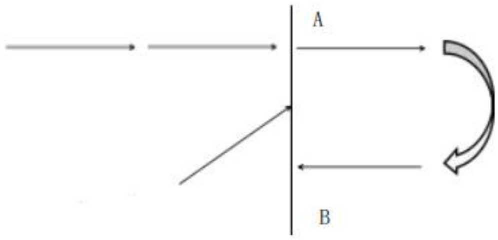


Figure 4-4 Schematic diagram of backlash measurement method

Backlash error compensation value = |Data recorded at point A – Data recorded at point B|; Input the calculated data into CNC data parameter N0063.

Data A: Data read from the dial indicator at point A;

Data B: Data read from the dial indicator at point B;

Note 1: CNC parameter N0062 can set the backlash compensation method, and parameter N0064 can set the compensation step length of the backlash in a fixed frequency mode;

Note 2: The machine tool needs to re-detect the backlash after every 3 months of use.

3.4 Tool Holder Commissioning

It can support multiple tool holders, and the specific parameter settings are subject to the machine tool manual. Related parameter settings for normal operation of the tool holder:

K parameter №011 Bit2 (TSGN): tool holder in place signal high/low level selection, if the tool in place signal is low level valid, a pull-up resistor must be connected in parallel;

K parameter №011 Bit3 (CTCP): check/not check tool holder locking signal when changing tools:

K parameter №011 Bit4 (TCPS): tool holder locking signal high/low level selection;

K parameter №011 Bit5 (CHET): check/not check tool position signal at the end of tool change;

The function of K parameter №011 tool change mode selection bit Bit0 (CHT) is detailed in the tool change control section;

T parameter №004: upper limit of time required for tool change;

T parameter №007: delay time from tool holder forward stop to reverse locking start;

System parameter N0126: total tool position selection;

T parameter №008: tool holder reverse locking time. When the tool holder does not rotate during the first power-on tool change, it may be due to incorrect phase sequence connection of the three-phase power supply of the tool holder motor. At this time, press the reset button immediately, cut off the power supply and check the wiring. If the three-phase power supply is incorrectly connected, any two phases of the three-phase power supply can be swapped.

The reverse locking time should be set appropriately. The setting time should not be too long or too short. If the reverse locking time is too long, the motor will be damaged; if the reverse locking time is too short, the tool holder may not be locked tightly. The method to check whether the tool holder is locked: use a dial indicator to close the tool holder, manually pull the tool holder, and the floating of the dial indicator pointer should not exceed 0.01mm.

During commissioning, each tool position and the tool position with the maximum conversion must be changed once to observe the correctness of the tool change and whether the time parameter setting is appropriate.

Chapter 4 Storage-type Pitch Error Compensation Function

4.1 Function Description

The pitch of the screws of each axis of the machine tool has more or less precision errors, which will inevitably affect the processing accuracy of the parts. The SZGH880T/SZGH1080T series has a storage-type pitch error compensation function that can accurately compensate for the pitch error of the screw.

4.2 Specifications

1. The set compensation amount is related to factors such as the compensation origin and compensation interval;
2. The pitch error compensation value is obtained by looking up the table based on the machine tool coordinate (mechanical coordinate) value and the pitch error compensation origin;
3. Number of compensation points: up to 256 for each axis;
4. Axes that can be compensated: X, Z, Y, 4th, 5th, 6th;
5. Compensation range: 0 ~ ±99 × least command increment;
6. Compensation interval: 1 ~ 9999.9999;
7. The compensation amount of compensation point N (N=0, 1, 2, 3, …255) is determined by the mechanical error of interval N and N-1;
8. The setting method is the same as the input method of CNC parameters. See *Operation Instructions* for details.

4.3 Parameter Setting

4.3.1 Pitch Compensation Function

System parameter N0067 sets whether to start the pitch error compensation function.

4.3.2 Pitch Error Compensation Origin

The compensation position number in the pitch error compensation table corresponding to the machine tool home is called the pitch error compensation origin (reference point); the pitch error compensation origin is set by system parameter N0068. According to actual needs, each axis can be set at any position between 0 and 255.

4.3.3 Compensation Interval

Pitch error compensation interval: N0069;

Input unit: metric machine tool: mm, imperial machine tool: inch;

Setting range: 1 ~ 9999.9999.

Note: The X-axis pitch error compensation interval is input as a radius value.

4.3.4 Compensation Amount

The pitch error compensation amount of each axis is set according to the parameter number in the table below. The compensation amount is fixedly input as a radius value, regardless of diameter programming or radius programming. The input value unit is mm (metric machine tool) or inch (imperial machine tool).

Compensation number	X	Z	Y
000	…	…	…
001	5	-2	3
002	-3	4	-1
…	…	…	…
255	…	…	…

4.4 Notes on Compensation Setting

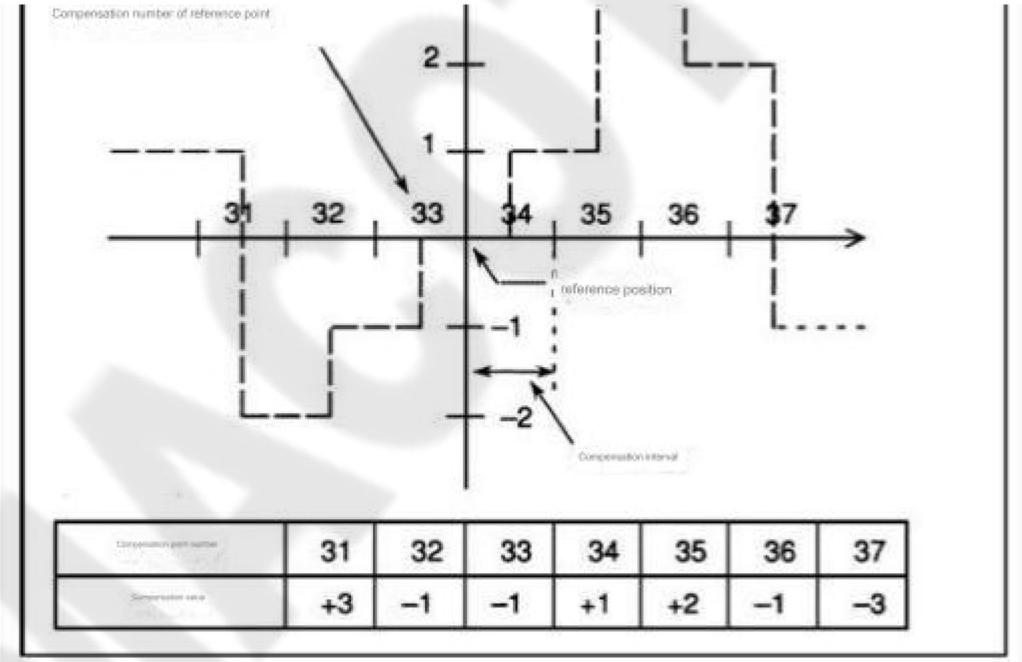
① The operation authority must be a secondary password to set and modify the pitch error compensation parameters.

② After setting the pitch error compensation parameters, the correct compensation can be performed only after machine tool homing.

4.5 Example of Compensation Parameter Setting

System parameter N0068 (pitch error compensation origin) = 33, system parameter N0069 (compensation interval) = 10.000mm

In the following example, the pitch error compensation point number of the reference point is 33.



4.6 Special Functions

4.6.1 Rotation and Stop of Power Head

M26 P speed Q axis number (0:X 1:Z 2:Y 3:A 4:B 5:C) //rotation axis forward; if Q is omitted, parameter 227 is used

M27 P speed Q axis number (0:X 1:Z 2:Y 3:A 4:B 5:C) //rotation axis reverse

M28 Q axis number (0:X 1:Z 2:Y 3:A 4:B 5:C) // rotation axis stop; if Q is omitted, all rotation axes are stopped

4.6.2 Workpiece Counting

M31

4.6.3 Automatic Loading and Unloading

M35 K_feeding output port (signed number, greater than 0, keep output when feeding in place; less than 0, the output is turned off when the material is in place)

I_in place signal input port (signed number, the sign is used to indicate the effective level)

Q_in place signal holding time, second

J_maximum waiting time for in place signal, second

P_delay between two feedings, second

L_number of repeated feedings

Note: (1) The output port refers to the port number of the output port custom page under diagnosis, not the pin number.

(2) Input port refers to the port number of the input port custom page under diagnosis, not the pin number.

4.6.4 Set Relative Coordinates

M46 X_Y_Z_

4.6.5 Force Port Switch

Port output: M80 K_port number (always output)

M80 K_port number J_delay time, unit: millisecond (output for a period of time and then close. The program waits until it is closed before continuing to execute)

M80 K_port number Q_delay time, unit: millisecond (output for a period of time and then close. The program does not wait, and closes in the backend)

Port closing: M81 K_port number (always closed)

M81 K_port number J_delay time, unit: millisecond (output after closing a period of time. The program waits until it is output, and then continues to execute)

M81 K_port number Q_delay time, unit: millisecond (output after closing a period of time. The program does not wait, and outputs in the backend)

Note: The port number refers to the port number of the output port custom page under diagnosis, not the pin number.

4.6.6 Waiting for Input Port Signal

Waiting is valid: M82 L_port number (always waiting)

M82 L_port number J_waiting time (unit: millisecond) (waiting timeout, alarm)

M82 L_port number J_waiting time (unit: millisecond) P_alarm number (waiting timeout, alarm, alarm content can be specified)

Waiting is invalid: M83 L_port number (always waiting)

M83 L_port number J_waiting time (unit: milliseconds) (waiting timeout, alarm)

M83 L_port number J_waiting time (unit: milliseconds) P_alarm number (waiting timeout, alarm, alarm content can be specified)

Note: (1) The port number refers to the port number of the input port custom page under diagnosis, not the pin number.

(2) In the input port custom page, if the port signal is valid, the green light is displayed; if the port signal is invalid, the white light is displayed.

4.6.7 Switching Instructions for 5 K Keys

M84 K_ (1/2/3/4/5 keys) I_ (0: off 1: on)

Description: (1) To use the K key for output, define the Y signal in the output port custom page

(2) K1 to K5 keys correspond to Y18.0 to Y18.4 respectively

4.6.8 M92 skips a specific line by an external signal

The first usage: M92 P_ (N number) //wait for external X3.1 to have a signal, and skip

The second usage: M92 P_ (N number) K_ (input port is high level) L_ (input port is low level) I_ (input port, signed number, >0 indicates high level, <0 indicates low level)

//K, L and I exist at the same time, I takes priority

4.6.9 M97 P_ (start N) Q_ (end N) L_ (number of calls)

After the M97 cycle is completed, it will start executing from the next line of M97

4.6.10 Enable axis and read current drive position at the same time, used for bus version

M111 Q axis number (0:X 1:Z 2:Y 3:A 4:B 5:C); if there is no Q, read all axes

4.6.11 Disable axis, used for bus version

M110 Q axis number (0:X 1:Z 2:Y 3:A 4:B 5:C); to specify Q

Appendix I System Parameters

Axis servo

No.	Note	Default Value	Range
0001	Programming method (0: diameter 1: radius)	0	0 or 1
0002	Axis type (0: linear axis 1: rotary axis)	0	0 or 1
0003	When it is a rotary axis, absolute coordinate cycle function (0: invalid 1: valid)	1	0 or 1
0004	When it is a rotary axis, (0: nearby rotation 1: rotation in the direction of the sign)	0	0 or 1
0005	When it is a rotary axis, relative coordinate cycle function (0: invalid 1: valid)	0	0 or 1
0006	C axis control type (0: pulse and direction 1: MII)	0	0 or 1

Gear ratio 1

No.	Note	Default Value	Range
0007	Numerator of electronic gear ratio	1	0~99999999
0008	Denominator of electronic gear ratio	1	0~99999999

Gear ratio 2

No.	Note	Default Value	Range
0009	Number of pulses per revolution	1	1~99999999
0010	Lead of screw	1	0~99999999
0011	Number of teeth of screw end gear	1	1~99999999
0012	Number of teeth of motor end gear	1	1~99999999

Speed

No.	Note	Default Value	Range
0013	Maximum control speed (universal for all axes mm/min)	8000	0~90000
0014	Minimum control speed (universal for all axes mm/min)	30	0~8000
0015	During dry run, the speed of G00 is (0: manual feed 1: rapid speed)	1	0 or 1
0016	G00 interpolation trajectory is (0: non-linear type 1: linear type)	0	0 or 1
0017	Rapid running mode (0: front acceleration and deceleration 1: rear acceleration and deceleration)	0	0 or 1
0018	Rapid running is (0: linear type 1: front acceleration and deceleration S type/rear acceleration and deceleration exponential type)	0	0 or 1
0019	Time constant for each axis before rapid acceleration and deceleration (ms)	100	1~4000
0020	Time constant for each axis after rapid acceleration and deceleration (ms)	80	1~4000

0021	Maximum speed of each axis rapid movement	5000	0~90000
0022	When the rapid feed override is Fo (0: no stop 1: stop)	0	0 or 1
0023	Rapid movement speed when the rapid movement override is F0 (mm/min)	400	6~4000
0024	Cutting feed mode (0: front acceleration and deceleration, 1: back acceleration and deceleration)	0	0 or 1
0025	Cutting feed (0: linear type 1: front acceleration and deceleration S type/back acceleration and deceleration exponential type)	0	0 or 1
0026	Pre-read acceleration and deceleration type (0: new 1: old)	0	0 or 1
0027	Cutting feed speed when power is turned on (mm/min)	300	0~9999
0028	Time constant for acceleration and deceleration before cutting feed (ms)	530	1~4000
0029	Time constant for acceleration and deceleration after cutting feed (ms)	100	1~4000
0030	Minimum speed of exponential acceleration and deceleration	0	0~9999
0031	Whether the acceleration of exponential acceleration and deceleration cutting feed is clamped (0: no 1: yes)	0	0 or 1
0032	Exponential acceleration and deceleration clamp constant	50	0~ 1000
0033	Automatic corner deceleration function (0: angle control, 1: speed difference control)	1	0 or 1
0034	Minimum feed speed of automatic corner deceleration (mm/min)	60	0~4000
0035	Critical angle of two program segments of automatic corner deceleration (degrees)	0	0~ 10
0036	Allowable deviation of each axis of speed difference mode deceleration function	10	0~4000
0037	Normal acceleration limit of circular interpolation	1000	0~9999
0038	Low speed limit of normal acceleration clamp of circular interpolation	200	0~9999
0039	Maximum value of circular radius error (mm)	0.01	0~ 1
0040	Circular interpolation control accuracy (mm)	0.03	0~ 1
0041	Maximum number of merged program segments	0	0~25
0042	Control accuracy of merged program segments	0	0~ 1
0043	Whether cutting feed is controlled to the in-place accuracy (0: no 1: yes) (only valid after acceleration and deceleration)	1	0 or 1
0044	Whether overlapping interpolation of the previous acceleration and deceleration program segment is valid (0: no 1: yes)	0	0 or 1
0045	Cutting feed in-place accuracy	0	0~ 1
0046	Pre-reading mode, cutting processing accuracy level	2	1~3
0047	Graphic preview speed (mm/min)	1000	0~9999

Manual

No.	Note	Default Value	Range
0048	Manual rapid movement before mechanical homing after power-on (0: Invalid 1: Valid)	1	0 or 1
0049	Manual operation selection (0: linear 1: exponential type acceleration and deceleration)	0	0 or 1
0050	Acceleration and deceleration time constant of each axis manual feed (ms)	100	1~4000
0051	Set speed when manual feed ratio is 100% (mm/min)	1260	0~9999

Handwheel

No.	Note	Default Value	Range
0052	MPG key action (0: single step 1: handwheel) mode	1	0 or 1
0053	X-axis handwheel or single step (0: coordinate 1: machine tool) movement amount	0	0 or 1
0054	Whether handwheel rotation displacement is fully operated (0: no 1: yes)	1	0 or 1
0055	Handwheel operation selection (0: linear 1: exponential type acceleration and deceleration)	1	0 or 1
0056	Handwheel acceleration and deceleration time constant (ms)	120	1~4000
0057	Maximum clamping speed when handwheel in incomplete operation mode	2000	0~3000
0058	Acceleration clamping constant when handwheel in incomplete operation mode	50	0~ 1000
0059	Handwheel/single step feed maximum clamping speed	1000	0~3000
0060	Handwheel trial cutting override	0.2	0~ 100
0061	Handwheel rotation direction of each axis	0	0 or 1

Compensation

No.	Note	Default Value	Range
0062	Backlash compensation method (0: fixed frequency 1: acceleration/deceleration)	0	0 or 1
0063	Backlash compensation amount for each axis (mm)	0	0~0.5
0064	Compensation step length for each axis backlash in fixed frequency mode	0.003	0~0.5
0065	Precision of reverse direction determined by backlash compensation (mm)	0.01	0.0001~ 1
0066	Time constant for backlash compensation in acceleration/deceleration mode	20	0~400
0067	Pitch error compensation function (0: invalid 1: valid)	0	0 or 1
0068	Pitch error compensation position number corresponding to the machine home position of each axis	0	0~225
0069	Pitch error compensation interval distance (mm)	10	1~ 10000
0070	Pitch error compensation override	0.001	0~ 10000

Homing

No.	Note	Default Value	Range
0071	Homing mode selection (0: after the block 1: before the block)	1	0 or 1
0072	Homing mode selection: (0: no 1: yes) one-turn signal	0	0 or 1
0073	Homing mode selection when there is no one-turn signal (0: U type 1: T type)	1	0 or 1
0074	Manual homing point selects multiple axes at the same time (0: yes 1: no)	0	0 or 1
0075	Relative coordinates for reference point return (0: not canceled 1: canceled)	0	0 or 1
0076	Speed for manual reference point return after reference point is established and memorized (0: fast 1: manual)	0	0 or 1

0077	When reference point is not established, execute instructions other than G28 (0: no alarm 1: alarm)	0	0 or 1
0078	G28 instruction when reference point is not established (0: use block 1: alarm)	0	0 or 1
0079	Machine home (0: not memorized 1: memorized)	0	0 or 1
0080	Deceleration signal (0: low, 1: high) level is valid for each axis	0	0 or 1
0081	Each axis returns to reference point in (0: positive 1: negative) direction	0	0 or 1
0082	Low speed for each axis to return to machine home (mm/min)	40	1~400
0083	High speed for each axis to return to machine home (mm/min)	4000	10~9999
0084	Machine tool home offset for each axis (mm)	0	-10000~10000
0085	Machine tool coordinates of the first reference point (mm)	0	-10000~10000
0086	Machine tool coordinates of the second reference point (mm)	0	-10000~10000
0087	Machine tool coordinates of the third reference point (mm)	0	-10000~10000
0088	Machine tool coordinates of the 4th reference point (mm)	0	-10000~10000

Coordinate system

No.	Note	Default Value	Range
0089	Automatic conversion of workpiece coordinate system values when converting between metric and imperial systems (0: no 1: yes)	1	0 or 1
0090	G54 workpiece origin offset (mm)	0	-99999~99999
0091	G55 workpiece origin offset (mm)	0	-99999~99999
0092	G56 workpiece origin offset (mm)	0	-99999~99999
0093	G57 workpiece origin offset (mm)	0	-99999~99999
0094	G58 workpiece origin offset (mm)	0	-99999~99999
0095	G59 workpiece origin offset (mm)	0	-99999~99999
0096	External workpiece origin offset (mm)	0	-99999~99999

Limit

No.	Note	Default Value	Range
0097	Whether to perform stroke detection before moving (0: no 1: yes)	1	0 or 1
0098	When issuing an overtravel command, alarm (0: before 1: after) overtravel	1	0 or 1
0099	Whether the soft limit is valid before mechanical homing (0: invalid 1: valid)	0	0 or 1
0100	Forbidden area of the second stroke limit (0: inside 1: outside)	1	0 or 1
0101	Maximum positive stroke (first stroke limit mm)	9999.999	-99999~99999
0102	Maximum negative stroke (first stroke limit mm)	-9999.999	-99999~99999
0103	Maximum positive stroke (second stroke limit mm)	9999	-99999~99999

0104	Maximum negative stroke (second stroke limit mm)	-9999	-99999~99999

Spindle

No.	Note	Default Value	Range
0105	Spindle speed (0: switch control 1: analog voltage control)	1	0 or 1
0106	Spindle speed arrival signal before cutting (0: no check 1: check)	0	0 or 1
0107	Whether G04 is paused per revolution in per-revolution feed mode (0: no 1: yes)	0	0 or 1
0108	Voltage offset compensation value when the spindle analog voltage output is 10V	0	-0.2~0.2
0109	Gain adjustment data of spindle speed analog output	1	0.98~ 1.02
0110	Number of spindle encoder lines	1024	100~5000
0111	Maximum spindle speed to be input (r/min)	6000	10~99999
0112	Maximum speed of corresponding gear of the spindle (r/min)	6000	10~99999
0113	Encoder and spindle gear ratio parameter: Number of spindle gears	1	1~255
0114	Encoder and spindle gear ratio parameter: Number of encoder gears	1	1~255
0115	Spindle instruction multiplication factor	512	0~9999
0116	Spindle instruction frequency division factor	125	0~9999
0117	Voltage output when the spindle shifts gears (mV)	100	0~ 10000
0118	Spindle rotation speed during jogging (r/min)	40	1~8000
0119	Backlash compensation amount for spindle reversal	0	0~ 100
0120	Spindle speed sampling period	1	0~32
0121	Maximum setting value corresponding to the inverter	65535	10~99999
0122	G0 positioning calculates G96 spindle speed according to (0: end coordinate 1: current coordinate)	0	0 or 1
0123	G96 spindle speed clamp (0: before spindle override 1: after spindle override)	0	0 or 1
0124	Minimum spindle speed (r/min) under constant line speed (G96) control	100	0~9999
0125	Axis used as counting reference during surface speed control	0	0~4

Tool holder

No.	Note	Default Value	Range
0126	Total tool position selection (tool post when less than 2)	4	1~32

System configuration

No.	Note	Default Value	Range
0127	Whether to ignore external user alarm (0: no 1: yes)	0	0 or 1

0128	Whether to ignore feed axis driver alarm (0: no 1: yes)	0	0 or 1
0129	Whether to ignore spindle driver alarm (0: no 1: yes)	0	0 or 1
0130	Whether to ignore emergency stop alarm (0: no 1: yes)	0	0 or 1
0131	Whether to ignore hard limit alarm (0: no 1: yes)	0	0 or 1
0132	Whether to ignore MII alarm (0: no 1: yes)	0	0 or 1
0133	Minimum moving unit: (0: 0.001 1: 0.0001)	0	0 or 1
0134	(0: metric 1: imperial) input	0	0 or 1
0135	(0: metric 1: imperial) output	0	0 or 1
0136	When the power is turned on or the status is cleared (0: G00 mode, 1: G01 mode)	0	0 or 1
0137	Setting when the power is turned on or the status is cleared (0: G12 1: G13)	1	0 or 1
0138	Whether to switch to the program interface by pressing the [Edit] key (0: no 1: yes)	1	0 or 1
0139	Tool life management function (0: invalid 1: valid)	0	0 or 1
0140	Number of controlled axes	2	1~6
0141	Axis name (0: X 1: Z 2: Y 3: A 4: B 5: C)		0~5
0142	Current ladder diagram number	66	1-99
0143	FPGA interpolation cycle	1	0.5 or 1
0144	Reset signal output time (ms)	200	50~400
0145	Total number of parts to be processed	0	0~99999
0146	Total number of parts processed	0	0~99999
0147	Accumulated value of power-on time (hours)	0	0~99999
0148	Accumulated value of cutting time (hours)	0	0~99999

Program

No.	Note	Default Value	Range
0149	Cursor (0: not return 1: return) to start after M02 execution	1	0 or 1
0150	Cursor (0: not return 1: return) to start after M30 execution	1	0 or 1
0151	When resetting, the cursor returning to the beginning of the program is valid in (0: edit 1: any) mode	1	0 or 1
0152	(0: not prohibit 1: prohibit) program editing of numbers 8000~8999	1	0 or 1
0153	(0: not prohibit 1: prohibit) program editing of numbers 9000~9999	1	0 or 1
0154	Whether to automatically insert the sequence number (0: no, 1: yes)	0	0 or 1
0155	Segment number increment value when automatically inserting the program segment number	10	
0156	Whether the single-piece processing time is automatically cleared (0: no 1: yes)	0	0 or 1
0157	When more than two identical addresses are specified in the same section (0: no alarm 1: alarm)	1	0 or 1
0158	Whether the reset key in the MDI interface deletes the compiled program (0: no 1: yes)	0	0 or 1

0159	Whether the compiled program is deleted after the MDI interface executes the program (0: no 1: yes)	0	0 or 1
0160	Whether the single-direction positioning G code is set as a modal code (0: no 1: yes)	0	0 or 1
0161	Movement sequence of each axis after power failure and restart (1: X 2: Z 3: Y 4: A 5: B 6: C)	123456	1~6

Tool compensation

No.	Note	Default Value	Range
0162	Tool compensation value is automatically converted when converting between metric and imperial units (0: no 1: yes)	1	0 or 1
0163	Execute tool offset in the form of (0: move 1: coordinate offset)	1	0 or 1
0164	No.0 tool compensation translation workpiece coordinate system (0: invalid 1: valid)	0	0 or 1
0165	Tool start form in tool radius compensation is (0: A type 1: B type)	0	0 or 1
0166	G28 and G30 instruction moves to the middle point, (0: not cancel 1: cancel) radius compensation	1	0 or 1
0167	Whether to perform radius compensation interference check (0: no, 1: yes)	1	0 or 1
0168	Limit value of ignoring vector when moving along the outside of the corner in tool radius compensation C	0	0~ 10000
0169	Maximum error value of tool radius compensation C	0.001	0.0001~0.01
0170	Positive/negative limit value of tool wear input each time (mm)	10	0.01~100

PLC

No.	Note	Default Value	Range
0171	External cycle start signal (0: valid 1: invalid)	0	0 or 1
0172	External pause signal (0: valid 1: invalid)	0	0 or 1
0173	External emergency stop signal (0: valid 1: invalid)	0	0 or 1
0174	Emergency stop effective level (0: high 1: low)	0	0 or 1
0175	Whether to use external editing lock (0: no 1: yes)	0	0 or 1
0176	Drive alarm level	0	0 or 1
0177	Feed direction level	0	0 or 1
0178	Is the axis movement key reversed (0: yes 1: no)	1	0 or 1
0179	Are all axis interlock signals valid (0: no 1: yes)	0	0 or 1

Thread

No.	Note	Default Value	Range
0180	Thread acceleration/deceleration mode (0: linear type 1: S type)	0	0 or 1
0181	Starting speed of X/Z axis for thread cutting (mm/min)	50	6~8000
0182	Linear acceleration/deceleration time constant in thread cutting (spindle gear position ms)	100	1~4000
0183	Thread back-off length TCH (*0.1*thread lead)	5	0~225

0184	Minor axis acceleration/deceleration time constant for thread back-off (ms)	100	1~4000
0185	Minor axis speed for thread back-off (set to 0 to back-off at the thread cutting feed rate)	0	0~8000
0186	Spindle speed fluctuation alarm limit value during thread machining (set to 0 to indicate no detection)	0	0~ 100

Fixed cycle

No.	Note	Default Value	Range
0187	Single feed amount during G71/G72 cycle turning (mm)	0.001	0.001~100
0188	Single retraction amount during G71/G72 cycle turning (mm)	0	0~ 100
0189	X-axis tool retraction amount during G73 cycle turning (mm)	0	-10000~10000
0190	Z-axis tool retraction amount during G73 cycle turning (mm)	0	-10000~10000
0191	Number of cuts during G73 cycle turning	1	1~9999
0192	Z/X-axis tool retraction amount during G74/G75 cycle turning (mm)	0	0~ 100
0193	Number of repetitions of G76 cycle finishing	1	1~99
0194	Tool nose angle during G76 cycle	0	0~99
0195	Minimum cutting depth during G76 cycle (mm)	0	0~ 100
0196	Finishing allowance during G76 cycle (mm)	0	0~ 100

Tapping

No.	Note	Default Value	Range
0197	Dry run during tapping (0: invalid 1: valid)	0	0 or 1
0198	Spindle control mode during tapping (0: follow 1: servo)	0	0 or 1
0199	Whether tapping becomes a high-speed deep hole tapping cycle (0: no 1: yes)	0	0 or 1
0200	Whether the spindle is ready to stop when flexible tapping starts (0: no 1: yes)	0	0 or 1
0201	Whether the same time constant is used for rigid tapping feeding and retraction (0: no 1: yes)	0	0 or 1
0202	Whether the override is valid during rigid tapping retraction (0: no 1: yes)	0	0 or 1
0203	Rigid tapping retraction override (0: 1% 1: 10%)	0	0 or 1
0204	Rigid tapping retraction override value	1	0~ 1
0205	Maximum spindle speed allowed for rigid tapping (r/min)	800	0~6000
0206	Retraction amount or space during deep hole tapping cycle	0	0~9999
0207	Linear acceleration and deceleration time constant of spindle and tapping axis	200	0~9999
0208	Time constant of spindle and tapping axis during retraction	200	0~9999

Macro program

No.	Note	Default Value	Range
0209	Macro program public variables #100~#199, (0: not clear 1: clear) after reset	0	0 or 1
0210	Macro program local variables #1~#50, (0: not clear 1: clear) after reset	0	0 or 1
0211	Whether single segment can be used in macro program instruction statement (0: no 1: yes)	0	0 or 1
0212	Whether to delay in macro program instruction statement (0: yes 1: no)	0	0 or 1

No.	Note	Default Value	Range
0213	Tool life management technology in times mode (0: mode 1: mode 2)	0	0 or 1
0214	X-axis handwheel/single-step acceleration/deceleration time (ms)	100	
	Y-axis handwheel/single-step acceleration/deceleration time (ms)	100	
	Z-axis handwheel/single-step acceleration/deceleration time (ms)	100	
	A-axis handwheel/single-step acceleration/deceleration time (ms)	100	
	B-axis handwheel/single-step acceleration/deceleration time (ms)	100	
	C-axis handwheel/single-step acceleration/deceleration time (ms)	100	
0215	Maximum speed of X-axis handwheel incomplete operation	2000	
	Maximum speed of Y-axis handwheel incomplete operation	2000	
	Maximum speed of Z-axis handwheel incomplete operation	2000	
	Maximum speed of A-axis handwheel incomplete operation	2000	
	Maximum speed of B-axis handwheel incomplete operation	2000	
	Maximum speed of C-axis handwheel incomplete operation	2000	
0216	Whether to modify the driver slave number (0: no 1: yes)	0	0 or 1
0217	1:X 2:Z 3:Y 4:A 5:B 6:C, press the reset button to take effect	1-6	1-6
0218	Lathe type (0: horizontal 1: vertical)	0	0 or 1
0219	Interval between consecutive key presses, unit: 15 milliseconds	14	
	Custom port page interface (0: prohibit 1: allow) modifying function column	0	0 or 1
	Parameter switch (0: on 1: off)	0	0 or 1
	Program switch (0: on 1: off)	0	0 or 1
	Reserved	0	
	Reserved*	0	0.500
0220	Tool setting mode (0: single trial cutting 1: memory)	0	0 or 1
0221	M function reset related settings		
	At M30, (0: output 1: no output) M05	0	0 or 1
	At M30, (0: output 1: no output) M09	0	0 or 1

		0	
		0	
	At resetting, (0: output 1: no output) M05	0	0 or 1
	At resetting, (0: output 1: no output) M09	0	0 or 1
	At resetting, (0: output 1: no output) M33	0	0 or 1
	Parameter 222 (0: reset or emergency stop 1: emergency stop only) turns off the output port	0	0 or 1
0222	Turn off the output port when resetting/emergency stop		
0001	(0: not turn off 1: turn off) output port 8-1	0	0 or 1
0002	(0: not turn off 1: turn off) output port 16-9	0	0 or 1
0003	(0: not turn off 1: turn off) output port 24-17	0	0 or 1
0004	(0: not turn off 1: turn off) output port 32-25	0	0 or 1
0223	When the spindle stops, whether to output analog quantity (0: no 1: yes)	0	0 or 1
0224	Axis protection		
	X-axis (0: none 1: negative 2: positive 3: positive and negative) protection	0	0-3
	Y-axis (0: none 1: negative 2: positive 3: positive and negative) protection	0	0-3
	Z-axis (0: none 1: negative 2: positive 3: positive and negative) protection	0	0-3
	A-axis (0: none 1: negative 2: positive 3: positive and negative) protection	0	0-3
	B-axis (0: none 1: negative 2: positive 3: positive and negative) protection	0	0-3
	C-axis (0: none 1: negative 2: positive 3: positive and negative) protection	0	0-3
0225	Axis protection brake opening time (ms)	100	
0226	Axis protection speed condition: during axis protection, open brake at (0: high speed and low speed 1: only high speed)	0	0 or 1
0227	M26/M27 rotation axis setting (012345 XYZABC)	2	0-5
0228	Maximum speed of rotation axis (rpm)	2000	
0229	Acceleration and deceleration time of rotation axis (ms)	100	
0230	Default speed of rotation axis (rpm)	500	
0231	Set the follower axis XYZABC of the active axis and the follower axis 123456 XYZABC of the axis respectively	0	0-6
0232	Whether to turn on the servo spindle function (0: off 1: on)	0	0 or 1
0233	Whether Y-axis homes after square cutting (0: no 1: yes)	0	0 or 1
0234	MII communication digital format (0: 17 1:32)	0	0 or 1
0235	Spindle speed stabilization time for thread detection (ms)	60	
0236	Spindle speed error range before thread	60	
0237	NC axis or PMC axis		

	(0: NC axis 1: PMC axis) XYZABC	0	0 or 1
0238	Channel number 1-4 of PMC axis corresponding channel XYZABC	1	1~4
0239	PMC axis speed selection XYZABC (0: system parameter 1: AXCTL parameter)	0	0 or 1
0240	Tool wear value compensation type is represented by (0: follow system programming 1: diameter value 2: radius value)	0	0-2
0241	Only use home signal for homing XYZABC (0: no 1: yes)	0	0 or 1
0242	Set the ratio of the follower axis to the active axis The ratio of each of XYZABC axis divided by the active axis (positive number in the same direction, negative number in the opposite direction)	1	
0243	Default spindle speed after power-on	100	
0244	Time waiting for spindle speed stabilization before cutting, unit: milliseconds	0	
0245	K key calls program number, greater than 0 is effective K1 K2 K3 K4 K5	0	
0246	Set K key to control quick retraction function 0-5 None, K1-K5	0	0-6
0247	Quick retraction function control button 0-6 None, XYZABC	0	0-6
0248	Quick retraction position, machine tool coordinates	0	
0249	Whether to turn off the spindle after quick push (0: no 1: yes)	0	0 or 1
0250	Clear each group of G codes when resetting or emergency stop		
	0	0	
	Clear group 07 G codes (G40/G41/G42)	1	
	0	0	
	0	0	
	0	0	
	Clear group 03 G codes (G98/G99)	1	
	Clear group 02 G codes (G96/G97)	1	
	Clear group 01 G codes (G00/G01/G02/G03)	0	

0251	Switch tool offset and wear interface position (0: no 1: yes)	0	0 or 1
0252	Shield correction number in position interface (0: no 1: yes)	0	0 or 1
0253	Number of tool posts	4	
0254	Switch program to clear processing number (0: no 1: yes)	1	0 or 1
0255	G130 call program number, valid when greater than 0 (0xxxx)	0	
0256	G131 call program number, valid when greater than 0 (0xxxx)	0	
0257	G132 call program number, valid when greater than 0 (0xxxx)	0	
0258	G133 call program number, valid when greater than 0 (0xxxx)	0	
0259	G134 call program number, valid when greater than 0 (0xxxx)	0	
0260	G135 call program number, valid when greater than 0 (0xxxx)	0	
0261	G136 call program number, valid when greater than 0 (0xxxx)	0	
0262	G137 call program number, valid when greater than 0 (0xxxx)	0	
0263	G138 call program number, valid when greater than 0 (0xxxx)	0	
0264	G139 call program number, valid when greater than 0 (0xxxx)	0	
0265	After power-on, each axis homes		
	After power-on, X-axis/YZABC00 (0: need 1: not need) homing (for pulse version)	0	0 or 1
0266	Rotation axis angle range (0: 0-360 1: determined by parameters 101 and 102)	0	0-360
0267	Starting program number for external signal calls The external signal starts from X50.0; the program number is valid when it is greater than 0 (0xxxx)	0	
0268	Number of external signals The number of external signals in the calling program (up to 32)	0	0-32
0269	Turn off optimized trajectory speed (0: no 1: yes)	0	0 or 1
0270	(0: cancel 1: reserve) tool offset when homing completed	0	0 or 1
0271	Set the maximum load of the driver (the lower three bits are the load, and the upper bits are the overload time)	0	
0272	Reset and clear the handwheel interruption amount (0: no 1: yes)	0	0 or 1
0273	Input port jitter time, unit: milliseconds	24	
0274	Delay detection of spindle speed fluctuation, unit: seconds	10	
0275	Spindle speed fluctuation alarm limit value (set to 0 for no detection)	0	
0276	Driver load detection conditions (0: real-time detection 1: cutting detection)	0	0 or 1
0277	Delay spindle shutdown when load alarms, unit: milliseconds	0	
0278	Interval time of enabling motor sequence, unit: milliseconds	1	
0279	Parallel axis corresponding to G7.1 rotation axis (0: X 1: Y 2: Z)	5	0-2
0280	G12.1 linear axis (0-5, XYZABC)	0	0-5
0281	G12.1 rotary axis (0-5, XYZABC)	0	0-5

0282	Tool compensation adjustment direction is (0: opposite to 1: same as) actual dimension direction	0	0 or 1
0283	(Bus version) motor enable, delay encoder, unit: milliseconds	0	
0284	Source of first round fourth axis signal (0: external input 1: software judgment)	0	0 or 1
0285	Press the reset key to execute the O9949 program (0: no 1: manual 2: automatic)	0	0, 1, 2
0286	Before M70 is executed, call M70 again (0: wait 1: alarm 2: none)	0	0, 1, 2
0287	Oblique axis setting (0: none 1: X axis 2: Y axis)	0	0, 1, 2
0288	Oblique axis angle	0	
0289	Tool change axis clamping and releasing function (0: off 1: on)	0	0, 1
0290	Tool change axis number (0-5, XYZABC)	0	0-5
0291	In the case of the M16, Y-axis movement alarm (0: no 1: yes)	0	0, 1
0292	In case of the M16, A-axis movement alarm (0: no 1: yes)	0	0, 1
0293	Keys K1 and K2 control the 5th axis (0: none 1: control the 5th axis)	0	0, 1
0294	Keys K3 and K4 control the 5th axis (0: none 1: control the 5th axis)	0	0, 1

Appendix II PLC Signals

X signal

No.	Bit	Note
X000	.0	X-axis deceleration signal
	.1	Tailstock control input
	.2	Tool position signal 04
	.3	Tool position signal 03
	.4	Tool position signal 02
	.5	Tool position signal 01
	.6	Tool position T08/tool holder overheating
	.7	Positive limit (all axes)
X001	.0	T07/clamp in place
	.1	Negative limit (all axes)
	.2	T06/release in place
	.3	Z-axis deceleration signal
	.4	Tool position signal 05
	.5	Y-axis deceleration signal
	.6	Chuck input
	.7	Tool holder locking signal
X002	.0	External cycle start
	.1	External feed pause
	.2	
	.3	4TH axis deceleration signal
	.4	5TH axis deceleration signal
	.5	6TH axis deceleration signal
	.6	Spindle gear 3 in place level Guard door detection signal
	.7	Guard door detection signal
X003	.0	Spindle gear 4 in place level
	.1	G31/M92 skip signal
	.2	M51-58 orientation completion signal
	.3	Chuck release in place
	.4	Spindle gear 1 in place

	.5	Chuck clamp in place
	.6	Spindle 1 position/speed signal
	.7	Spindle gear 2 in place
X004	.2	M51-58 orientation completion signal
	.6	Spindle 1 position/speed signal
X005	.0	Spindle 1 speed reached
	.1	Teaching point
	.2	Detect M08 output
	.3	
	.4	
	.5	M16 release in place
	.6	M17 clamping in place
	.7	External reset
X006	.0	Auxiliary panel start signal
	.1	Auxiliary panel pause signal
	.2	Auxiliary panel emergency stop signal
	.3	External handwheel magnification X100
	.4	External handwheel axis selection X
	.5	External handwheel magnification X10
	.6	External handwheel axis selection Y
	.7	External handwheel magnification X1
X007	.0	External handwheel axis selection Z
	.1	Tailstock forward in place
	.2	Tailstock backward in place
	.3	Cylinder alarm input
	.4	3-position switch left interface signal
	.5	3-position switch right interface signal
	.6	Inverter alarm input
	.7	Lubrication alarm input
X008	.0	Cut back in place signal
	.1	Cut forward in place signal
	.2	External K1
	.3	External K2

	.4	External K3
	.5	External K4
	.6	External K5
	.7	Quick retraction
X009	.0	Corresponding to Y9.0
	.1	Corresponding to Y9.1
	.2	Corresponding to Y9.2
	.3	Corresponding to Y9.3
	.4	Corresponding to Y9.4
	.5	Corresponding to Y9.5
X010	.0	External handwheel axis selection A
	.1	2nd spindle drive in position mode signal
	.2	2nd spindle alarm
	.3	External handwheel axis selection B
	.4	External handwheel axis selection C
X012	.5	Turn off cooling (input port)
	.6	Turn off lubrication (input port)
X013	.0	PMC channel 1 start
	.1	PMC channel 2 start
X014	.0	PMC channel 1 pause
	.1	PMC channel 2 pause
	.2	PMC channel 3 pause
	.3	PMC channel 4 pause
	.4	PMC axis ignore alarm
X015	.0	External feed rate override knob A
	.1	External feed rate override knob F
	.2	External feed rate override knob B
	.3	External feed rate override knob E
X016	.0	External rapid override knob A
	.1	External rapid override knob F
	.2	External rapid override knob B
	.3	External rapid override knob E
X017	.0	External spindle override knob A
	.1	External spindle override knob F

	.2	External spindle override knob B
	.3	External spindle override knob A
X018	.0	Close Y18.0
	.1	Close Y18.1
	.2	Close Y18.2
	.3	Close Y18.3
	.4	Close Y18.4
X020	.0	Edit mode key
	.1	Automatic mode key
	.2	Input mode key
	.3	Machine tool home mode key
	.4	Handwheel mode key
	.5	Manual mode key
	.6	Program home mode key
	.7	MES mode key
X021	.0	Single-segment key
	.1	Skip-segment key
	.2	Machine tool lock key
	.3	Auxiliary lock key
	.4	Dry run key
	.5	Selection stop key
	.6	Coolant key
	.7	Lubricant key
X022	.0	Increase rapid override
	.1	Cancel rapid override (100%)
	.2	Decrease rapid override
	.3	Increase feed rate override
	.4	Cancel feed rate override (100%)
	.5	Decrease feed rate override
	.6	Increase spindle override
	.7	Cancel spindle override (100%)
	.0	Decrease spindle override

X023	.1	Rapid override Fo
	.2	Rapid override 25%
	.3	Rapid override 50%
	.4	Rapid override 100%
	.5	Single step length X1
	.6	Single step length X10
	.7	Single step length X100
X024	.0	Single step length X1000
	.1	Manual rapid switch key
	.2	X-axis positive direction/handwheel selection key
	.3	Press Y+ key/handwheel selection key
	.4	Z-axis positive direction/handwheel selection key
	.5	4TH axis positive direction/handwheel selection key
	.6	5TH axis positive direction/handwheel selection key
	.7	Press X-key
X025	.0	Press Y-key
	.1	Press Z-key
	.2	4TH axis negative direction button
	.3	5TH axis negative direction button
	.4	Spindle forward key
	.5	Spindle stop key
	.6	Spindle reverse key
	.7	Spindle jog key
X026	.0	Spindle stop/spindle orientation
	.1	Program restart
	.2	Lighting lamp
	.3	Guard door
	.4	Chip conveyor
	.5	Tailstock key
	.6	Chuck key
	.7	Tool change key
	.0	FUN1
	.1	FUN2

X027	.2	FUN3
	.3	Tool magazine forward
	.4	Tool magazine backward
	.5	Clamp and release tool
	.6	Cycle start key
	.7	Pause key
X028	.0	Lock 1 (operation)
	.1	Lock 2 (edit)
	.2	Emergency stop
	.3	Reset (panel key)
	.4	External cycle start
	.5	External pause
	.6	Rapid override 1/three-stage switch first gear
	.7	Rapid override 2/three-stage switch second gear
X029	.0	Spindle override 1
	.1	Spindle override 2
	.2	Spindle override 3
	.3	Feed rate override 1
	.4	Feed rate override 2
	.5	Feed rate override 3
	.6	Feed rate override 4
X030	.0	TH6 axis positive direction
	.1	TH6 axis negative direction key
	.2	MPG trial cut key
X031	.0	K1 key
	.1	K2 key
	.2	K3 key
	.3	K4 key
	.4	K5 key
X032	.0	Single step key

Y signal

No.	Bit	Note
Y000	.0	Tool holder reverse output
	.1	Tool holder forward output
	.2	Chuck release output
	.3	Tailstock reverse output
	.4	Mechanical gear 4 output
	.5	Spindle forward output
	.6	Lubrication output
	.7	Mechanical gear 1 output
Y001	.0	Spindle braking
	.1	Chuck clamping output
	.2	Spindle stop output
	.3	Spindle reverse output
	.4	Cooling output
	.5	Tailstock forward output
	.6	Mechanical gear 3 output
	.7	Mechanical gear 2 output
Y002	.0	Brake
	.1	
	.2	
	.3	
	.4	
	.5	Lighting
	.6	Tri-color light - red
	.7	Tri-color light - green (running)
Y003	.0	
	.1	Tri-color light - yellow (RDY)
	.2	Release
	.3	Tool holder locking output
	.4	M51-58 directional output signal
	.5	Tool holder brake/release
	.6	Spindle 1 speed/position switching output

	.7	Spindle clamping
Y004	.7	Spindle 2 speed/position switching output
Y005	.0	Wenchang tool number code 1
	.1	Wenchang tool number code 2
	.2	Wenchang tool number code 3
	.3	Wenchang tool number code 4
	.4	Tool holder enable
	.5	MDO
	.6	MD1
	.7	Wenchang tool number code 5
Y006	.0	Spindle 2 reverse
	.1	Spindle 2 forward
	.2	Spindle 2 stop
	.3	
	.4	
	.5	M16 release
	.6	M17 clamp
	.7	Spindle servo disconnect (reduce spindle servo excitation)
Y007	.0	X-axis enable
	.1	Y axis enabled
	.2	Z-axis enable
	.3	4TH axis enable
	.4	Spindle 1 enable
	.5	Spindle 2 enable
	.6	Spindle orientation
	.7	Speed/position
Y009	.0	Corresponding to X9.0
	.1	Corresponding to X9.1
	.2	Corresponding to X9.2
	.3	Corresponding to X9.3
	.4	Corresponding to X9.4
	.5	Corresponding to X9.5
	.0	X-axis brake open output

Y017	.1	Y-axis brake open output
	.2	Z-axis brake open output
	.3	A-axis brake open output
	.4	B-axis brake open output
	.5	C-axis brake open output
Y018	.0	K1 key output
	.1	K2 key output
	.2	K3 key output
	.3	K4 key output
	.4	K5 key output
Y019	.0	4th axis homing end light
	.1	Z-axis homing end light
	.2	Y-axis homing end light
	.3	X-axis homing end light
	.4	5th axis homing end light
	.5	6th axis homing end light
Y020	.0	Edit mode light
	.1	Automatic mode light
	.2	Input mode light
	.3	Machine tool home mode light
	.4	Handwheel mode light
	.5	Manual mode light
	.6	Program home mode light
	.7	MES mode indicator light
Y021	.0	Single-segment indicator light
	.1	Skip-segment indicator light
	.2	Machine tool lock indicator light
	.3	Auxiliary lock indicator light
	.4	Dry run indicator light
	.5	Selection stop indicator light
	.6	Coolant indicator light
	.7	Lubricant indicator light
Y022	.1	Rapid override (100%) indicator light
	.4	Feed rate override (100%) indicator light

	.7	Spindle override (100%) indicator light
Y023	.1	Rapid override Fo indicator light
	.2	Rapid override 25%
	.3	Indicator light rapid override 50%
	.4	Indicator light rapid override 100% indicator light
	.5	Single step length x1 indicator light
	.6	Single step length x10 indicator light
	.7	Single step length x100 indicator light
Y024	.0	Single step length x1000 indicator light
	.1	Manual rapid switch indicator light
	.2	X-axis handwheel selection indicator light
	.3	Y-axis handwheel selection indicator light
	.4	Z-axis handwheel selection indicator light
	.5	4TH-axis handwheel selection indicator light
	.6	5TH-axis handwheel selection indicator light
Y025	.4	Spindle forward indicator light
	.5	Spindle stop indicator light
	.6	Spindle reverse indicator light
	.7	Spindle jog indicator light
Y026	.0	Spindle stop/spindle orientation indicator light
	.1	Program restart indicator light
	.2	Lighting indicator light
	.3	Guard door indicator light
	.4	Chip conveyor indicator light
	.5	Tailstock indicator light
	.6	Chuck indicator light
	.7	Tool change indicator light
Y027	.0	Tool magazine homing indicator light
	.1	Tool magazine forward rotation indicator light
	.2	Tool magazine reverse rotation indicator light
	.3	Tool magazine forward indicator light
	.4	Tool magazine reverse indicator light
	.5	Tool clamping and unclamping indicator light

	.6	Cycle start key indicator light
	.7	Pause key indicator light
Y028	.0	Current gear position nixie tube indication 1
	.1	Current gear position nixie tube indication 2
	.2	Current tool number nixie tube indication 1
	.3	Current tool number nixie tube indication 2
	.4	Current tool number nixie tube indication 3
	.5	Current tool number nixie tube indication 4
	.6	Current tool number nixie tube indication 5
	.7	Machine tool panel ALM indicator light
Y029	.0	Machine tool panel RDY indicator light
	.1	X-axis homing indicator light
	.2	Y-axis homing indicator light
	.3	Z-axis homing indicator light
	.4	4TH axis homing indicator light
	.5	5TH axis homing indicator light
	.6	External cycle start indicator light
	.7	External pause indicator light

F signal

No.	Bit	Note
F000	.4	Feed pause light signal
	.5	Cycle start light signal
	.6	Servo ready signal
	.7	Automatic operation signal
F001	.0	Alarm signal
	.1	Reset signal
	.2	Spindle speed arrival signal
	.3	Distribution end signal
	.4	Spindle enable signal
	.5	Tapping signal
	.7	CNC ready signal
F002	.1	Rapid feed signal
	.3	Thread cutting signal
	.4	Program start signal
	.5	Selection stop detection signal
	.6	Cutting feed signal
	.7	System sends dry run signal
F003	.0	System in single step mode
	.1	System in handwheel mode
	.2	System in manual mode
	.3	System in input mode
	.4	DNC operation selection detection signal
	.5	System in automatic mode
	.6	System in edit mode
	.7	System sends MPG trial cutting signal
F004	.0	System sends skip segment
	.1	System sends machine tool lock signal
	.3	System sends single segment signal
	.4	System sends auxiliary lock signal
	.5	System in mechanical home mode
	.6	System in program home mode

F007	.0	Auxiliary function selection signal
	.2	Execute S when spindle gear control
	.3	Tool function selection signal
F009	.4	M30 decoding signal
	.5	M02 decoding signal
	.6	M01 decoding signal
	.7	M00 decoding signal
F010	.0	Auxiliary function code MB00
	.1	Auxiliary function code MB01
	.2	Auxiliary function code MB02
	.3	Auxiliary function code MB03
	.4	Auxiliary function code MB04
	.5	Auxiliary function code MB05
	.6	Auxiliary function code MB06
	.7	Auxiliary function code MB07
F014	.0	PLC enters debugging mode
	.1	Switch mode prohibition signal
F015	.1	Z-axis selection (whether enabled)
	.2	Y-axis selection (whether enabled)
	.3	4TH axis selection (whether enabled)
	.4	5TH axis selection (whether enabled)
	.5	6TH axis selection (whether enabled)
F018	.0	Actual spindle speed AR00
	.1	Actual spindle speed AR01
	.2	Actual spindle speed AR02
	.3	Actual spindle speed AR03
	.4	Actual spindle speed AR04
	.5	Actual spindle speed AR05
	.6	Actual spindle speed AR06
	.7	Actual spindle speed AR07
F019	.0	Actual spindle speed AR08
	.1	Actual spindle speed AR09
	.2	Actual spindle speed AR10

	.3	Actual spindle speed AR11
	.4	Actual spindle speed AR12
	.5	Actual spindle speed AR13
	.6	Actual spindle speed AR14
	.7	Actual spindle speed AR15
F021	.3	Shield external emergency stop signal
	.4	External emergency stop level
	.5	Shield external pause signal
	.6	Shield external cycle start signal
F022	.0	Spindle speed code signal SB00
	.1	Spindle speed code signal SB01
	.2	Spindle speed code signal SB02
	.3	Spindle speed code signal SB03
	.4	Spindle speed code signal SB04
	.5	Spindle speed code signal SB05
	.6	Spindle speed code signal SB06
	.7	Spindle speed code signal SB07
F026	.0	Tool function code signal TB00
	.1	Tool function code signal TB01
	.2	Tool function code signal TB02
	.3	Tool function code signal TB03
	.4	Tool function code signal TB04
	.5	Tool function code signal TB05
	.6	Tool function code signal TB06
	.7	Tool function code signal TB07
F030	.0	S12 bit code signal R01O
	.1	S12 bit code signal R02O
	.2	S12 bit code signal R03O
	.3	S12 bit code signal R04O
	.4	S12 bit code signal R05O
	.5	S12 bit code signal R06O
	.6	S12 bit code signal R07O
	.7	S12 bit code signal R08O

F031	.0	S12 bit code signal R09O
	.1	S12 bit code signal R10O
	.2	S12 bit code signal R11O
	.3	S12 bit code signal R12O
F032	.4	Step length X1 soft key
	.5	Step length X10 soft key
	.6	Step length X100 soft key
	.7	Step length X1000 soft key
F033	.0	Tapping mode, issued simultaneously when executing M29
	.1	G84 rigid tapping mode signal
	.2	G88 rigid tapping mode signal
F034	.2	X+ soft key
	.3	X- soft key
	.4	Z+ soft key
	.5	Z- soft key
	.6	Spindle forward soft key
	.7	Spindle stop soft key
F035	.0	Rapid feed soft key
	.1	Soft panel dry run
	.2	Soft panel machine tool lock
	.3	Soft keypad single segment
	.4	Soft keypad skip segment key
	.5	Soft panel auxiliary lock
	.6	Soft panel selection stop
	.7	Spindle reverse soft key
F036	.0	Feed rate override increase soft key
	.1	Feed rate override decrease soft key
	.4	Rapid override increase soft key
	.5	Rapid override decrease soft key
	.6	Spindle override increase soft key
	.7	Spindle override decrease soft key
F037	.0	X homing end signal ZP1
	.1	Y homing end signal ZP2

	.2	Z homing end signal ZP3
	.3	4TH homing end signal ZP4
	.4	5TH homing end signal ZP5
	.5	6TH homing end signal ZP6
F038	.0	X-axis movement signal MV1
	.1	Z-axis movement signal MV2
	.2	Y-axis movement signal MV3
	.3	4TH axis movement signal MV4
	.4	5TH axis movement signal MV5
	.5	6TH axis movement signal MV6
F039	.0	Axis movement direction signal MVD1
	.1	Axis movement direction signal MVD2
	.2	Axis movement direction signal MVD3
	.3	Axis movement direction signal MVD4
	.4	Axis movement direction signal MVD5
F040	.0	Reference point establishment signal ZRF1
	.1	Reference point establishment signal ZRF2
	.2	Reference point establishment signal ZRF3
	.3	Reference point establishment signal ZRF4
	.4	Reference point establishment signal ZRF5
F041	.0	End signal of X-axis returning to the first reference point
	.1	End signal of Z-axis returning to the first reference point
	.2	End signal of Y-axis returning to the first reference point
	.3	End signal of 4TH axis returning to the first reference point
	.4	End signal of 5TH axis returning to the first reference point
	.5	End signal of 6TH axis returning to the first reference point
F042	.0	Program homing end signal PRO1
	.1	Program homing end signal PRO2
	.2	Program homing end signal PRO3
	.3	Program homing end signal PRO4
	.4	Program homing end signal PRO5
F043	.0	System in jog state
F044	.1	Spindle is in (0: speed 1: position) mode

	.2	2nd spindle Cs contour control switching end signal
	.4	(0: switch control 1: analog voltage control)
F045	.1	2nd spindle is in (0: speed 1: position) mode
F047	.0	Tool location number 0
	.1	Tool location number 1
	.2	Tool location number 2
	.3	Tool location number 3
	.4	Tool location number 4
	.5	Tool location number 5
	.6	Tool location number 6
	.7	Tool location number 7
F051	.0	Is X-axis movement key inverted (0: no 1: yes)
	.1	Is Z-axis movement key inverted (0: no 1: yes)
	.2	Is Y-axis movement key inverted (0: no 1: yes)
	.3	Is the 4TH axis movement key inverted (0: no 1: yes)
	.4	Is the 5TH axis movement key inverted (0: no 1: yes)
	.5	Is the 6TH axis movement key inverted (0: no 1: yes)
F054	.0	Macro output signal UO00
	.1	Macro output signal UO01
	.2	Macro output signal UO02
	.3	Macro output signal UO03
	.4	Macro output signal UO04
	.5	Macro output signal UO05
	.6	Macro output signal UO06
	.7	Macro output signal UO07
F055	.0	Macro output signal UO08
	.1	Macro output signal UO09
	.2	Macro output signal UO10
	.3	Macro output signal UO11
	.4	Macro output signal UO12
	.5	Macro output signal UO13
	.6	Macro output signal UO14
	.7	Macro output signal UO15

F057	.0	End signal of X axis returning to the second reference point
	.1	End signal of Z-axis returning to the second reference point
	.2	End signal of Y-axis returning to the second reference point
	.3	End signal of 4TH axis returning to the second reference point
	.4	End signal of 5TH axis returning to the second reference point
	.5	End signal of 6TH axis returning to the second reference point
F058	.0	End signal of X axis returning to the third reference point
	.1	End signal of Z axis returning to the third reference point
	.2	End signal of Y axis returning to the third reference point
	.3	End signal of 4TH axis returning to the third reference point
	.4	End signal of 5TH axis returning to the third reference point
	.5	End signal of 6TH axis returning to the third reference point
F059	.0	End signal of X axis returning to the fourth reference point
	.1	End signal of Z axis returning to the fourth reference point
	.2	End signal of Y axis returning to the fourth reference point
	.3	End signal of 4TH axis returning to the fourth reference point
	.4	End signal of 5TH axis returning to the fourth reference point
	.5	End signal of 6TH axis returning to the fourth reference point
F060	.0	All tools in the same group has reached end of life
F061	.1	Signal of reaching required number of parts

G signal

No.	Bit	Note
G004	.3	Auxiliary function end signal
G005	.4	All axis interlock signal
	.6	Auxiliary lock signal
	.7	Edit lock signal
G006	.0	Program restart signal
	.1	Selection stop signal
	.2	Manual absolute value signal
	.4	Feed rate override cancel signal
	.6	G31 skip signal
	.7	M92 skip signal
G007	.2	Cycle start signal
G008	.4	Emergency stop signal
	.5	Feed hold signal (0 valid)
	.6	Reset signal
G009	.0	Cooling signal
	.1	Lubrication signal
	.2	Chuck clamping signal
	.3	Tailstock forward signal
G010	.0	Manual movement override signal JV00
	.1	Manual movement override signal JV01
	.2	Manual movement override signal JV02
	.3	Manual movement override signal JV03
	.4	Manual movement override signal JV04
	.5	Manual movement override signal JV05
	.6	Manual movement override signal JV06
	.7	Manual movement override signal JV07
G012	.0	Feed rate override signal FV00
	.1	Feed rate override signal FV01
	.2	Feed rate override signal FV02
	.3	Feed rate override signal FV03
	.4	Feed rate override signal FV04

	.5	Feed rate override signal FV05
	.6	Feed rate override signal FV06
	.7	Feed rate override signal FV07
G014	.0	Rapid feed override signal RV01
	.1	Rapid feed override signal RV02
	.2	Rapid feed override signal RV03
	.3	Rapid feed override signal RV04
	.4	Rapid feed override signal RV05
	.5	Rapid feed override signal RV06
	.6	Rapid feed override signal RV07
	.7	Rapid feed override signal RV08
G016	.3	Spindle speed arrival signal
	.4	Spindle speed arrival
G017	.0	X axis homing deceleration signal detection
	.1	Z axis homing deceleration signal detection
	.2	Y axis homing deceleration signal detection
	.3	4TH axis homing deceleration signal detection
	.4	5TH axis homing deceleration signal detection
	.5	6TH axis homing deceleration signal detection
G018	.0	X axis handwheel feed selection signal
	.1	Z axis handwheel feed selection signal
	.2	Y axis handwheel feed selection signal
	.3	4TH axis handwheel feed selection signal
	.4	5TH axis handwheel feed selection signal
	.5	6TH axis handwheel feed selection signal
G019	.4	Handwheel override signal MP1
	.5	Handwheel override signal MP2
	.7	Manual rapid feed selection signal
G021	.0	Spindle speed override signal SOV0
	.1	Spindle speed override signal SOV1
	.2	Spindle speed override signal SOV2
	.3	Spindle speed override signal SOV3
	.4	Spindle speed override signal SOV4

	.5	Spindle speed override signal SOV5
	.6	Spindle speed override signal SOV6
	.7	Spindle speed override signal SOV7
G025	.0	Multi-spindle 1st spindle selection signal
	.1	Multi-spindle 2nd spindle selection signal
	.4	Spindle forward signal
	.5	Spindle reverse signal
G026	.0	1st spindle analog voltage off signal
	.1	2nd spindle analog voltage off signal
	.6	2nd spindle CS contour control switching signal
	.7	CS contour control switching signal
G027	.0	X+ feed axis and direction selection signal
	.1	Z+ feed axis and direction selection signal
	.2	Y+ feed axis and direction selection signal
	.3	4TH+ feed axis and direction selection signal
	.4	5TH+ feed axis and direction selection signal
	.5	6TH+ feed axis and direction selection signal
G028	.0	X- feed axis and direction selection signal
	.1	Z- feed axis and direction selection signal
	.2	Y- feed axis and direction selection signal
	.3	4TH- feed axis and direction selection signal
	.4	5TH- feed axis and direction selection signal
	.5	6TH- feed axis and direction selection signal
G030	.0	X axis positive overtravel signal +L1 (low effective)
	.1	Z axis overtravel signal +L2 (low effective)
	.2	Y axis overtravel signal +L3 (low effective)
	.3	4TH axis overtravel signal +L4 (low effective)
	.4	5TH axis overtravel signal +L5 (low effective)
	.5	6TH axis overtravel signal +L6 (low effective)
G031	.0	X axis negative overtravel signal -L1 (low effective)
	.1	Z axis overtravel signal -L2 (low effective)
	.2	Y axis overtravel signal -L3 (low effective)
	.3	4TH axis overtravel signal -L4 (low effective)

	.4	5TH axis overtravel signal -L5 (low effective)
	.5	6TH axis overtravel signal -L6 (low effective)
G032	.0	Axis overtravel release signal
G036	.0	In spindle jog state
	.6	Index table clamping completion signal
	.7	Index table release completion signal
G037	.0	Current tool number NT00
	.1	Current tool number NT01
	.2	Current tool number NT02
	.3	Current tool number NT03
	.4	Current tool number NT04
	.5	Current tool number NT05
	.6	Current tool number NT06
	.7	Current tool number NT07
G043	.0	Current working mode selection 1
	.1	Current working mode selection 2
	.2	Current working mode selection 3
	.5	DNC operation selection signal
	.7	Current working mode selection 4
G044	.0	Segment skip signal
	.1	Machine tool lock signal
	.7	Manual sequence tool change signal
G046	.1	Single segment signal
	.3	Memory protection signal
	.6	MPG trial cutting signal
	.7	Dry run signal
G048	.0	Gear selection signal, i.e. K3.1
	.1	Gear selection signal, i.e. K3.2
G054	.0	Macro input signal UI00
	.1	Macro input signal UI01
	.2	Macro input signal UI02
	.3	Macro input signal UI03
	.4	Macro input signal UI04

	.5	Macro input signal UI05
	.6	Macro input signal UI06
	.7	Macro input signal UI07
G055	.0	Macro input signal UI08
	.1	Macro input signal UI09
	.2	Macro input signal UI10
	.3	Macro input signal UI11
	.4	Macro input signal UI12
	.5	Macro input signal UI13
	.6	Macro input signal UI14
	.7	Macro input signal UI15
G061	.0	In rigid tapping mode

A signal

Address	Alarm number	Display content
A0000.0	1200	Tool change time is too long
A0000.1	1201	At the end of the tool change, alarm if the tool holder is not in place
A0000.2	1202	Alarm if tool change is not completed
A0000.3	1203	Alarm if lock signal is not received
A0000.4	1204	When tool change is completed, the lock signal is repeatedly detected, and the lock signal is invalid
A0000.5	1205	Tool change error before the system powers off
A0001.0	1208	The tailstock function is invalid, and M10 and M11 instructions cannot be executed
A0001.1	1209	The spindle is rotating, tailstock cannot be retracted
A0001.3	1211	The tailstock advance is not detected, and the spindle cannot be rotated
A0001.4	1212	Tool change mode A or B can have up to 8 tools
A0001.5	1213	Tool life is over
A0002.0	1216	Guard door is not closed/automatic operation is not allowed
A0002.1	1217	Low pressure alarm
A0002.3	1219	When the spindle is rotating, the chuck must not be released
A0002.4	1220	When the spindle is rotating, alarm if the clamping signal is invalid
A0002.5	1221	When the chuck clamping signal is invalid, the spindle must not be started
A0002.6	1222	When the chuck is released, the spindle must not be started
A0003.0	1224	Chuck function is invalid, and M12/M13 instructions cannot be executed
A0003.1	1225	Chuck clamping/releasing signal is not detected
A0003.7	1231	The total number of tool positions in the tool holder is greater than 4, and the external override cannot be connected (address reuse)
A0004.0	1232	Illegal M code
A0004.1	1233	Not an analog spindle currently, the jog function cannot be executed
A0004.2	1234	M03, M04 code specification error
A0004.4	1236	Spindle shift time is too long
A0004.5	1237	Spindle speed/position control switching time is too long
A0005.1	1241	Spindle servo or inverter abnormal alarm
A0007.1	1257	Guard door is open
A0007.3	1259	Cutter not locked alarm

K parameter

No.	Bit	Note
K000	.1	Ladder diagram interface data is displayed as (0: decimal; 1: hexadecimal)
	.2	When the Y-axis is a rotating axis, press the emergency stop to (0: not cancel; 1: cancel) enable
	.3	Inverter alarm switch type (0: normally open; 1: normally closed)
	.4	Lubrication alarm switch type (0: normally open; 1: normally closed)
	.5	Turn on lighting upon power on (0: no; 1: yes)
	.6	Use system internal analog spindle speed
	.7	PLC enters (0: operation mode; 1: debugging mode)
K001	.0	Debug PLC
	.1	Panel start (0: press; 1: release) valid
	.2	Shield panel start key (0: no; 1: yes)
	.3	Shield machine tool home key (0: no; 1: yes)
	.4	Lower left corner key (0: cannot; 1: can) modify rapid override
	.5	External start is valid when (0: release; 1: press delay)
	.6	Reset when changing tools (0: allow; 1: not allow)
	.7	Rapid key (0: state hold; 1: press to take effect)
K002	.0	Use 3-position switch (0: no; 1: yes)
	.1	3-position switch signal (0: normally closed; 1: normally open)
	.2	3-position switch turns to start gear and continues to run (: yes; 1: no)
	.3	Teaching switch type (0: normally open; 1: normally closed)
	.4	When the spindle is rotating, reverse the spindle (: stop first and then reverse; 1: alarm)
	.5	Cut back in place switch type (0: normally open; 1: normally closed)
	.6	Cut forward in place switch type (0: normally open; 1: normally closed)
	.7	Output spindle direction signal type (0: level; 1: pulse)
K003	.1	Spindle gear holding register
	.2	Spindle gear holding register
	.4	External handwheel axis selection (0: valid; 1: invalid)
	.6	Handwheel override signal MP1
	.7	Handwheel override signal MP2
	.0	X positive overtravel signal
	.1	X negative overtravel signal
	.2	Z positive overtravel signal

K004	.3	Z negative overtravel signal
	.4	Y positive overtravel signal
	.5	Y negative overtravel signal
	.6	A positive overtravel signal
	.7	A negative overtravel signal
K005	.4	B positive overtravel signal
	.5	B negative overtravel signal
	.6	C positive overtravel signal
	.7	C negative overtravel signal
K006		Feed rate override holding register
K007		Rapid override holding register
K008		Spindle override holding register
K009		Working mode holding register
K010	.0	Cooling output at reset (0: Off; 1: Hold)
	.1	Spindle output at reset (0: off; 1: hold)
	.2	1: Start the program after the Z-axis returns to the first reference point
	.3	Feed rate override (0: adjustable; 1: fixed to 100%)
	.4	Spindle jog is valid in (0: manual, handwheel, homing mode; 1: any mode)
	.5	Lubrication output at reset (0: off; 1: hold)
	.6	Hard limit overtravel alarm switch type (0: normally open; 1: normally closed)
	.7	Hard limit overtravel (0: invalid; 1: valid)
K011	.0	Tool holder release signal (0: not detect; 1: detect) (hydraulic)
	.1	Switch tool holder forward and reverse output (0: no; 1: yes)
	.2	Tool holder position signal switch type (0: normally open; 1: normally closed)
	.3	Tool holder locking signal (0: no detect; 1: detect)
	.4	Tool holder locking signal switch type (0: normally open; 1: normally closed)
	.5	Tool position signal at the end of tool change (0: not check; 1: check)
	.6	Wenchang tool holder is the absolute value type (0: no; 1: yes)
	.7	Real-time detection of tool position signal (0: no; 1: yes)
K012	.0	Chuck control (0: invalid; 1: valid)
	.1	Chuck clamping before spindle start (0: check; 1: no check)
	.2	Chuck control mode (0: internal; 1: external)
	.3	Chuck in place signal (0: not check; 1: check)

	.4	Disconnect output when chuck clamping completed (0: no; 1: yes)
	.5	Disconnect output when chuck release completed (0: no; 1: yes)
	.6	Foot switch type (0: normally open; 1: normally closed)
	.7	Manually control chuck during operation (0: not allow; 1: allow)
K013	.0	Tailstock control (0: invalid; 1: valid)
	.1	Spindle rotation and tailstock forward and backward (0: interlock; 1: not interlocked)
	.2	Whether to detect tailstock in-place signal (0: no; 1: yes)
	.3	Record tailstock status when power is off (0: no; 1: yes)
	.4	Tailstock forward completion (0: no; 1: yes)
	.5	Disconnect output tailstock retraction completion (0: no; 1: yes)
	.6	Disconnect output tailstock forward in-place switch type (0: normally open; 1: normally closed)
	.7	Tailstock retraction in-place switch type (0: normally open; 1: normally closed)
K014	.0	Whether to detect cylinder alarm (0: no; 1: yes)
	.1	Cylinder alarm switch type (0: normally open; 1: normally closed)
	.2	Guard door function (0 invalid; 1: valid)
	.3	Guard door open signal (0: normally open; 1: normally closed)
	.4	When guard door is open (0: close spindle; 1: do not close spindle)
	.5	
	.6	
	.7	External feed rate override switch (0: invalid; 1: valid)
K015	.0	Spindle automatic gear shift function (0: invalid; 1: valid)
	.1	Gear shift in-place signal (0: do not check; 1: check)
	.2	M41-44 switch type (0: normally open; 1: normally closed)
	.3	Spindle gear power-off memory (0: no memory; 1: memory)
	.4	Key operation spindle during program running (0: valid; 1: invalid)
	.5	Spindle orientation function (0: invalid; 1: valid)
	.6	Spindle contour control during emergency stop/reset (0: not closed; 1: closed)
	.7	Cs axis function (0: invalid; 1: valid)
K016	.0	Whether to shield the driver feedback signal during spindle V/P switching (0: no; 1: yes)
	.1	Spindle position mode signal (0: normally closed; 1: normally open)
	.2	Automatic lubrication is effective, lubrication at startup (0: no output; 1: output)
	.3	M51-58 orientation completion signal (0: normally closed; 1: normally open)

	.4	After the spindle positioning is completed (0: do not disconnect; 1: disconnect) output
	.5	Whether to shield the orientation completion feedback signal (0: no; 1: yes)
	.6	Single-step mode X1000 increment (0: valid; 1: invalid))
	.7	Handwheel mode X1000 increment (0: valid; 1: invalid)
K017	.0	Spindle speed arrival signal before cutting (0: no check 1: check)
	.1	After power on, the spindle enters (0: speed; 1: position) mode
	.2	Spindle servo disconnected (0: spindle tightening; 1: spindle loosening)
	.3	
	.4	When the spindle rotates, (0: do not allow; 1: allow) clamping the spindle
	.5	Second spindle alarm switch type (0: normally open; 1 normally closed)
	.6	When multiple spindles are valid, the spindle is clamped (: 1st; 1: 2nd)
	.7	Multi-spindle function (0: invalid; 1: valid) requires restart
K018	.0	Operation mode setting MD1: 000: input 001: automatic 010: program homing
	.1	Operation mode setting MD2: 011: edit 100: handwheel 101: manual
	.2	Operation mode setting MD4: 110: mechanical homing
	.3	Operation mode (0: power-off memory; 1: specified mode)
K020	.0	External start (0: normally open; 1: normally closed)
	.1	External pause (0: normally open; 1: Normally closed)
	.2	External emergency stop (0: Normally open; 1: Normally closed)
	.3	External reset (0: Normally open; 1: Normally closed)
	.4	X3.1 switch type (0: normally open; 1: normally closed)
	.5	Detect M08 output switch type (0: normally open; 1: normally closed)
	.6	Detect M08 output (0: no; 1: yes)
	.7	Emergency retraction switch type (0: normally open; 1: normally closed)
K021	.0	Tool change successful, current tool == T instruction tool
K022	.2	5TH axis deceleration signal (0: normally closed; 1: normally open)
	.3	6TH axis deceleration signal (0: normally closed; 1: normally open)
	.4	X-axis deceleration signal (0: normally closed; 1: normally open)
	.5	Y-axis deceleration signal (0: normally closed; 1: normally open)
	.6	Z-axis deceleration signal (0: normally closed; 1: normally open)
	.7	4TH axis deceleration signal (0: normally closed; 1: normally open)

K023	.0	Rapid override (0: adjustable; 1: fixed at 100%)
	.1	Spindle override (0: adjustable; 1: fixed at 100%)
	.2	
	.3	
	.4	Power on (0: no output; 1: output M17)
	.5	M16 release in place (0: normally open; 1: normally closed)
	.6	M17 clamp in place (0: normally open; 1: normally closed)
	.7	M16/M17 (0: detect; 1: not detect) in place signal
K024	.0	X-axis handwheel feed selection hold signal
	.1	Z-axis handwheel feed selection hold signal
	.2	Y-axis handwheel feed selection hold signal
	.3	4TH axis handwheel feed selection hold signal
	.4	5TH axis handwheel feed selection hold signal
	.5	6TH axis handwheel feed selection hold signal
	.6	
	.7	
K025	.0	Feed rate override (0: panel; 1: external)
	.1	Rapid override (0: panel; 1: external)
	.2	Spindle override (0: panel; 1: external)
	.3	Knob binary (0: reverse code; 1: original code)
K030	.0	Tool change direction (0: forward; 1: nearby), hydraulic tool holder
	.1	After locking in place (0: hold; 1: disconnect) output
	.2	Tool holder overheating (0: not detect; 1: detect), Yantai AK31
	.3	Reverse rotation after reaching T tool number (0: off; 1: on)
K060	.0	Power-off tailstock status
K061	.0	C-axis function (0: servo spindle; 1: feed axis)
	.1	Tool life alarm (0: immediate alarm; 1: program end)
K062	.0	(Bus 197) uses (0: encoder at encoder port; 1: encoder at spindle port)
	.1	(Bus 107) uses (0: spindle port to send pulse; 1 encoder door to send pulse)
	.2	K1-K5 key light (0: no; 1: yes) (take effect after powering off and restarting)
	.3	
	.4	Keypad membrane B-axis and C-axis (0: no; 1: yes) (take effect after powering off and restarting)
	.5	Bus driver parameter default (0: Yaskawa; 1: other)
	.6	Master station (0: check; 1: not check) slave station watchdog counter
	.7	Forced port switch (0: port; 1: PLC signal)
K063	.0	Organize NC program upon power on (0: no; 1: yes)
	.1	
	.2	Use when the spindle speed is less than 1 revolution
	.3	Major and minor axis change and reverse thread (0: not detect; 1: detect) home signal
	.4	Whether to alarm when exceeding the maximum speed of the system during threading (0: no; 1: yes)
	.5	Spindle override during thread tapping (0: fixed 100%; 1: adjustable)
	.6	Port during reset (0: restore the corresponding Y signal state; 1: no change)
	.7	(0: Turn on; 1: Turn off) thread interruption alignment function

T parameter

No.	Note
T000	Time for the spindle to close original gear position when shifting gear, unit: millisecond
T001	Time from the output of new gear position of the spindle to the end, unit: millisecond
T002	Timeout for pressing the start key, unit: millisecond
T003	Time for the external start to be effective after pressing the start key, unit: millisecond
T004	Timeout for tool change, unit: millisecond
T005	M code execution duration, unit: millisecond
T006	Gear spindle S0~S4 execution time, unit: millisecond
T007	The time delay from the forward stop of the tool holder to the reverse locking (mode B), unit: millisecond
T008	Maximum time for receiving tool holder locking signal, unit: millisecond
T009	Tool holder reverse locking time, unit: millisecond
T010	Delay before braking when spindle stops, unit: millisecond
T011	Total time of spindle braking output, unit: millisecond
T012	Spindle jog time, unit: millisecond
T013	Lubrication start time, unit: second (lubrication is not limited by time when set to 0)
T014	Pulse time in output spindle direction, unit: millisecond
T015	Lubrication output pulse time, unit: milliseconds (set to 0 for level mode)
T016	Cooling output pulse time, unit: milliseconds (set to 0 for level mode)
T018	Chuck output maximum time, stop output when time is up. 0 Ignore, unit: milliseconds
T020	M08 output time before detecting M08, unit: milliseconds
T021	Delay before chuck control after the spindle stops, unit: milliseconds
T022	M20/M21 delay time, unit: milliseconds
T023	Spindle clamping output time, unit: milliseconds
T024	Chuck clamping output time, unit: milliseconds
T025	Chuck release output time, unit: milliseconds
T026	Tailstock forward output time, unit: milliseconds
T027	Tailstock backward output time, unit: milliseconds
T028	Release output delay (hydraulic), unit: milliseconds
T029	Lock output delay (hydraulic), unit: milliseconds
T030	Delay before opening brake if the driver does not alarm, unit: milliseconds
T031	Delay before braking if the driver alarms, unit: milliseconds

T032	Delay after reaching T tool number, (hydraulic), unit: milliseconds
T033	M16/M17 in place delay, unit: milliseconds
T034	M16/M17 timeout, unit: milliseconds
T036	Delay after rotation stops, unit: milliseconds
T037	Delay after locking (Xuyang), unit: milliseconds
T038	Release output delay (Xuyang), unit: milliseconds
T039	Delay after reaching T tool number (Xuyang), unit: milliseconds
T043	Chuck tightening/relaxing timeout (detecting in-position signal), unit: milliseconds
T045	Spindle automatic gear shift timeout, unit: milliseconds
T053	Auto-lubrication interval output shutdown time, unit: milliseconds
T055	Delay time before M51-M58 orientation, unit: milliseconds
T056	Delay time before spindle shutdown in M51-M58 orientation, unit: milliseconds
T057	M51-M58 orientation timeout, unit: milliseconds
T059	Tri-color light - red light alarm flashing time, unit: milliseconds

C parameter

No.	Note
C004	Panel feed rate override count
C005	Panel rapid override count
C006	Panel spindle override count
C007	Panel manual override count
C020	Press reset several times to cancel the tool change incomplete alarm

D parameter

No.	Note
D000	Total number of tools + 1
D001	Set to 0
D002-D009	Tool offset code No. 1 to No. 8
D021	Set to 0
D022-D029	Tool offset code No. 1 to No. 8
D080	Current T08-T01 total tool position signal
D081	DSCH S4, current tool number
D099	Spindle zero speed output range (r/min)
D100	Save current tool number

D110	PMC channel 1 control instruction
D111	PMC channel 1 data 1 low 4 bits
D112	PMC channel 1 data 1 high 4 bits
D113	PMC channel 1 data 2 low 4 bits
D114	PMC channel 1 data 2 high 4 bits
D115	PMC channel 2 control instruction
D116	PMC channel 2 data 1 low 4 bits
D117	PMC channel 2 data 1 high 4 bits
D118	PMC channel 2 data 2 low 4 bits
D119	PMC channel 2 data 2 high 4 bits
D1271	Tool holder selection (0: standard; 1: hydraulic; 2: Yantai; 3: Wen; 4: Liuxin; 5: Xuyang)

Appendix III System Alarms

No.	Content
0000	The parameter that must be cut off once has been modified.
0001	File opening failed
0002	Data input is out of range
0003	Copied or renamed program number exists.
0004	Address not found
0005	No data after address
0006	Illegal use of negative sign
0007	Illegal use of decimal point
0008	Program file is too large and not fully loaded.
0009	Illegal address input
0010	Incorrect G code
0011	No feed speed instruction
0012	Insufficient disk space
0013	The number of program files has reached the upper limit
0014	Cannot instruct G95, and spindle does not support
0015	Too many axes have been instructed
0016	Current pitch error compensation point is out of range
0017	No permission to modify
0018	Modification is not allowed
0019	Scaling function is not enabled
0020	Exceeding radius tolerance
0021	Instructed illegal plane axis
0022	R and IJK in the arc are all 0
0023	IJK and R are specified at the same time in circular interpolation
0024	Spiral interpolation rotation angle is 0
0025	G12 cannot be in the same segment as other G instructions
0026	File format that is not supported by the system.
0027	Length tool compensation instruction cannot be in the same segment as G92
0028	Illegal plane selection

0029	Illegal offset value
0030	Illegal compensation number
0031	Illegal P address is instructed in G10, or P address is not commanded.
0032	Illegal compensation value in G10
0033	No intersection in tool compensation C or chamfering
0034	Tool compensation cannot be established or canceled during circular arc instruction
0035	C tool compensation is not canceled before M99 instruction
0036	Cannot instruct G31
0037	Plane cannot be changed in tool compensation C
0038	Interference in circular arc program segment
0039	Tool nose positioning error in tool compensation C
0040	Workpiece coordinate system changed during tool compensation C
0041	Interference in tool compensation C
0042	More than ten non-movement instructions for compensation plane in tool compensation C
0043	Insufficient authority
0044	G27~G30 instructions are not allowed in fixed cycle
0045	Address Q is not found or Q value is 0 (G73/G83)
0046	Illegal reference point return instruction
0047	Mechanical homing must be executed before executing this instruction
0048	Z plane should be higher than R plane
0049	Z plane should be lower than R plane
0050	Position should be moved when changing fixed cycle mode
0051	Wrong movement after chamfering or chamfering value is too large
0052	Mirror function cannot be used in slot milling fixed cycle
0053	Incorrect chamfering or rounding instruction format
0054	DNC transmission error
0055	Wrong movement value in chamfer or reverse R
0056	M99 cannot be in the same section as macro program instruction
0057	File writing failed, power must be turned off and restarted.
0058	End point not found
0059	Program number not found
0060	Sequence number not found

0061	X-axis not at reference point
0062	Z-axis not at reference point
0063	Y-axis not at reference point
0064	A-axis not at reference point
0065	B-axis not at reference point
0066	C-axis not at reference point
0067	Setting format not supported by G10.
0068	Parameter switch not turned on
0070	Insufficient memory capacity, insufficient memory
0071	Data end not found
0072	Too many programs
0073	Program number already used
0074	Illegal program number
0075	Protection
0076	Address P not defined
0077	Subroutine nesting error
0078	Program number not found
0079	System usage time expired.
0080	Irrational data entry
0081	Macro program cannot call the subroutine.
0082	H code is instructed in G37
0083	Illegal axis instruction in G37
0084	Key timeout or short circuit occurs
0085	Communication error
0086	Plane cannot be switched in fixed cycle mode
0087	X-axis returning to reference point not completed
0088	Z-axis returning to reference point not completed
0089	Y-axis returning to reference point not completed
0090	A-axis returning to reference point not completed
0091	B-axis returning to reference point not completed
0092	C-axis returning to reference point not completed
0094	P type (coordinate) is not allowed

0095	P type is not allowed (EXT OFS CHG)
0096	P type is not allowed (WRK OFS CHG)
0097	P type is not allowed (auto execution)
0098	G28 is found in sequence return
0099	MDI is not allowed to execute after retrieval
0100	Parameter write is valid
0101	Power-off memory data is disordered, and please make sure the position is correct
0102	The motor model parameters of the system and the drive are inconsistent
0103	485 bus communication error
0104	Setting machine tool home timeout
0105	Getting drive data timeout
0106	Gear ratio of the drive and system servo parameters is inconsistent.
0107	Drive parameters are inconsistent with system servo parameters.
0109	Screw compensation value has changed, and please perform homing.
0110	The position data exceeds the allowable range. Please perform homing.
0111	Calculation data overflow
0112	Division by zero
0113	Incorrect instruction
0114	G39 format error
0115	Illegal variable
0116	Write protection variable
0117	This parameter does not support G10 online modification
0118	Brace nesting error
0119	M00~M02, M06, M98, M99, M30 cannot be in the same segment with other M instructions
0120	Some settings are restored
0121	Machine tool coordinates and encoder feedback value exceed the deviation setting value
0122	Quadruple macro mode-call
0123	Macro instructions cannot be used in DNC
0124	Illegal program end
0125	Macro program format error
0126	Illegal number of cycles
0127	NC and macro instructions in the same program segment

0128	Illegal sequence number of macro instructions
0129	Illegal independent variable address
0130	Illegal axis operation
0131	Too much external alarm information
0132	Alarm number not found
0133	Axis instructions not supported by the system
0134	Rigid tapping cannot be used when the number of system control axes is greater than 3
0135	Illegal angle instructions
0136	Illegal axis instructions
0139	Cannot change PLC control axis
0140	Macro instruction skip sequence number does not exist
0141	MDI current mode and DNC mode do not support macro instruction skip
0142	Illegal ratio
0143	Scaling motion data overflow
0144	Illegal plane selection
0148	Illegal data setting
0149	Format error in G10 L3
0150	Illegal tool group number
0151	Tool group number not found
0152	Tool data cannot be stored
0153	C tool compensation not canceled before tool change
0154	Tool in unused life group
0155	Illegal T code in M06
0156	P/L instructions not found
0157	Too many tool groups
0158	Illegal tool life data
0159	Tool data setting not completed
0160	Arcs can only use R programming in polar coordinate mode
0161	This instruction cannot be executed in polar coordinate mode
0162	G70~G76 instructions are used in input mode
0163	This instruction cannot be executed in rotation mode
0164	This instruction cannot be executed in scaling mode

0165	Please specify this instruction in a separate program segment
0166	No axis is specified when returning to the reference point
0167	Middle point coordinates are too large
0168	Minimum dwell time at the bottom of the hole should be less than the maximum dwell time at the bottom of the hole
0169	Tool radius compensation is not canceled when entering or exiting cylindrical or polar coordinate interpolation
0170	Tool radius compensation is not canceled when entering or exiting a subroutine
0172	In the program segment that calls a subroutine, P is not an integer or P is less than or equal to 0
0173	The number of subroutine calls should be less than 9999
0175	Fixed cycles can only be executed in the G17 plane
0176	Spindle speed is not specified before rigid tapping starts
0177	Spindle orientation function is not supported
0178	Spindle speed is not specified before fixed cycle starts
0181	Illegal M code
0182	Illegal S code
0183	Illegal T code
0184	Selected tool is out of range
0185	L is too small or L is not defined
0186	L is too large
0187	Tool radius is too large
0188	U is too large
0189	U value is less than tool radius
0190	V is too small or V is not defined
0191	W is too small or W is not defined
0192	Q is too small or Q is not defined
0193	I is not defined or I is 0
0194	J is not defined or J is 0
0195	D is not defined or D is 0
0198	P instruction of G96 exceeds the value range
0199	Macro instruction not defined
0200	Illegal S mode instruction
0201	Feed speed not found in rigid tapping
0202	Position LSI overflow

0203	Wrong program in rigid tapping
0204	M29 should be specified in G80 mode
0205	Rigid mode DI signal is off
0206	Cannot change plane (rigid tapping)
0207	Tapping data is wrong
0208	This instruction cannot be executed in G10 mode.
0209	Scaling and rotating polar coordinates mode does not support program restart.
0210	Program restart file name is inconsistent.
0212	Illegal plane selection
0213	Tool change macro program does not support G31 skip.
0214	Tool change macro program does not support segment skip operation.
0215	Tool change macro program does not support dynamic modification of the coordinate system and tool compensation.
0216	Scaling/rotation/polar coordinates do not support G31 skip.
0217	The skip state cannot be changed in the scaling/rotation/polar coordinate mode.
0218	Scaling/rotation/polar coordinates do not support the dynamic modification of the coordinate system and tool compensation.
0219	Tool magazine is not used, and tool change instruction M06 cannot be used.
0220	Scaling/rotation/polar coordinates do not support metric and inch input switching.
0221	Tool change macro program does not support metric and inch input switching.
0224	Return to reference point
0225	Waiting signal timeout
0226	Spindle is not in position mode
0231	Illegal format in G10 L50 or L51
0232	Too many helical interpolation axes are instructed
0233	Device busy
0235	Record end
0236	Program restart parameter error
0237	No decimal point
0238	Address duplication error
0239	Parameter 0
0240	G41/G42 is not allowed in MDI mode
0241	Abnormal handwheel pulse
0242	Bus connection error

0243	Spindle home deviation is too large
0244	Thread processing speed exceeds the upper limit value
0245	Spindle speed fluctuation exceeds the limit value during thread processing
0251	Emergency stop alarm
0255	Thread segment cannot specify the spindle speed
0256	Thread lead is out of range
0257	T instruction is used in the program segment of G71~G73 instructions
0258	The two program segments specified by address P or Q are instructed in M98/M99/M30/M02
0259	In G71/G72 instruction, the address Z(W)/X(U) is instructed in the P program segment
0260	Axis name is repeated, and please modify the parameters
0261	Tool offset number exceeds the valid range (0~32)
0262	Tool number is not within the parameter setting range
0263	In tool life management, the tool group number exceeds the range (1~32)
0264	The T instruction cannot be executed in C tool compensation, and please cancel C tool compensation
0265	G70~G76, G90, G92, G94 can only be used in the G18 plane
0266	The plane conversion instruction G17~G19 cannot be executed
0267	G11 or G13.1 is missing in the program
0268	In tool life management, there is no tool in the current tool group
0269	In tool life management, the current tool group is not defined
0270	All tools in the same group has reached end of life
0271	The tool life management function is invalid, and G10 L3 instruction cannot be used
0272	G11 cannot be programmed before G10
0273	The movement amount in the X direction during G33 tapping is not 0
0274	The number of thread indexing heads is greater than 65535
0275	The absolute value of R in the G90 and G92 instructions is greater than the absolute value of U/2
0276	The absolute value of R in the G94 instruction is greater than the absolute value of W
0277	The number of finishing program segments in the G70~G73 instructions exceeds 31
0278	The order of Ns and Nf in the finishing program segments in the G70~G73 instructions is wrong
0279	The cycle segment number Ns or Nf does not exist in the G70~G73 instructions
0280	The cycle start and end segment numbers are not entered in the G70~G73 instructions
0281	The subroutine is called in the G70~G73 cycle
0282	There is no instruction G00 or G01 in the G70~G73 cycle start segment

0283	The G instruction that is prohibited from use is used in the G70~G73 cycle start segment
0284	The G instruction that is prohibited from use is used in the G70~G73 cycle end segment
0285	The G70~G73 instructions are used in the input mode
0286	The coordinate change in G71~G72 cycle finishing program segment is non-monotonic
0287	The single feed amount in G71 or G72 exceeds the allowable range
0288	The single retract amount in G71 or G72 exceeds the allowable range
0289	The first segment of the G71 instruction specifies Z or W
0290	The first segment of the G72 instruction specifies Z or W
0291	The total cutting amount of G73 exceeds the allowable range
0292	The number of cycles of G73 is less than 1 or greater than 9999
0293	The single retract amount R(e) in G74 or G75 exceeds the allowable range
0294	The retract amount when cutting to the end point in G74 or G75 is a negative value
0295	The single cutting amount in the X or Z direction in G74 or G75 exceeds the allowable range
0296	Z value is not entered in the G74 instruction
0297	Q value in the G74 instruction is 0 or not entered
0298	X value is not entered in the G75 instruction
0299	P value in the G75 instruction is 0 or not entered
0300	The starting point is between the thread start point and the thread end point when G76 machining a tapered thread
0301	The minimum cutting amount in the G76 instruction exceeds the allowable range
0302	G76 finishing allowance exceeds the allowable range
0303	G76 thread height is less than the finishing allowance or less than 0
0304	The number of G76 cycles exceeds the allowable range
0305	G76 thread chamfer width exceeds the allowable range
0306	The tool nose angle in the G76 instruction exceeds the allowable range
0307	The X-axis or Z-axis movement in the G76 instruction is 0
0308	The thread height P value is not specified in the G76 instruction
0309	The first cutting depth Q value is not specified in the G76 instruction or the Q value is 0
0310	The cycle start point is in the closed area formed by the start point and end point of the finishing trajectory
0311	The pitch is less than 0 during variable pitch thread cutting
0312	The thread height in the G76 instruction is less than the X-axis movement
0313	More than 10 grooves
0320	The additional axis instruction has no chamfering function

0321	WHILE, END instruction is used in the input mode
0322	The macro program format is specified incorrectly
0323	DO, END labels in the macro program are not 1, 2, 3
0324	The DO, END format in macro program is specified incorrectly
0325	The brackets in the macro statement do not match or the format specification is incorrect
0326	The divisor in the macro statement cannot be 0
0327	The inverse tangent ATAN format specified in the macro statement is incorrect
0328	The antilog of LN in the macro statement is 0 or less than 0
0329	The square root in the macro statement cannot be negative
0330	The result of the tangent TAN in the macro statement is infinite
0331	The operand of ASIN or ACOS in the macro statement exceeds the range of -1 to 1
0332	Macro variable number or variable value in the macro statement is illegal (error)
0333	Too many WHILE nestings in the macro statement
0334	WHILE loop in the macro statement does not end with END
0340	The value of A or B cannot be 0
0341	A cannot be less than B
0342	The hyperbola A and B values cannot be 0
0343	The address value exceeds the value range
0344	M97 instruction error
0345	M97 has multiple start lines
0346	No start line or end line found
0347	No corresponding start line found
0350	SPC alarm n-axis pulse encoder
0351	SPC alarm n-axis communication
0360	Too many pulses in a single cycle
0401	Drive alarm 01: Overspeed
0402	Drive alarm 02: Main circuit overvoltage
0403	Driver alarm 03: Main circuit undervoltage
0404	Driver alarm 04: Position out of tolerance
0405	Driver alarm 05: Motor overheating
0406	Driver alarm 06: Speed amplifier saturation fault
0407	Driver alarm 07: Drive prohibition abnormal

0408	Driver alarm 08: Position deviation counter overflow
0409	Driver alarm 09: Encoder fault
0410	Driver alarm 10: Control power undervoltage
0411	Driver alarm 11: IPM module fault
0412	Driver alarm 12: Overcurrent
0413	Driver alarm 13: Overload
0414	Driver alarm 14: Braking fault
0415	Driver alarm 15: Encoder counting error
0420	Driver alarm 20: EEPROM error
0430	Driver alarm 30: Encoder Z pulse loss
0431	Driver alarm 31: Encoder UVW signal error
0432	Driver alarm 32: Encoder UVW signal illegal encoding
0433	Driver alarm 33: Communication interrupted
0434	Driver alarm 34: Encoder speed abnormal
0435	Driver alarm 35: Encoder status abnormal
0436	Driver alarm 36: Encoder counting abnormal
0437	Driver alarm 37: Encoder single-turn count overflow
0438	Driver alarm 38: Encoder multi-turn count overflow
0439	Driver alarm 39: Encoder battery alarm
0440	Driver alarm 40: Encoder battery power failure
0441	Driver alarm 41: Motor model does not match
0442	Driver alarm 42: Absolute position data abnormal alarm
0443	Driver alarm 43: Encoder EPPROM verification alarm
0449	Ethernet initialization failed
0450	Driver disconnected
0451	X-axis driver alarm
0452	Z-axis driver alarm
0453	Y-axis driver alarm
0454	A-axis driver alarm
0455	B-axis driver alarm
0456	Spindle driver alarm
0498	Soft limit overtravel: -C

0499	Soft limit overtravel: +C
0500	Soft limit overtravel: -X
0501	Soft limit overtravel: -+X
0502	Soft limit overtravel: -Z
0503	Soft limit overtravel: +Z
0504	Soft limit overtravel: -Y
0505	Soft limit overtravel: +Y
0506	Soft limit overtravel: -A
0507	Soft limit overtravel: +A
0508	Soft limit overtravel: -B
0509	Soft limit overtravel: +B
0510	Hard limit overtravel: -X
0511	Hard limit overtravel: +X
0512	Hard limit overtravel: -Z
0513	Hard limit overtravel: +Z
0514	Hard limit overtravel: -Y
0515	Hard limit overtravel: +Y
0516	Hard limit overtravel: -A
0517	Hard limit overtravel: +A
0518	Hard limit overtravel: -B
0519	Hard limit overtravel: +B
0520	Hard limit overtravel: -C
0521	Hard limit overtravel: +C
0523	The axis is not a PMC axis
0530	The program home is not defined
0531	The number of processed pieces has reached the set value
0600	The equipment trial has expired
0601	The machine code does not match
0602	Automatic loading error
0740	Rigid tapping alarm: out of tolerance
0741	Rigid tapping alarm: out of tolerance
0742	Rigid tapping alarm: LSI overflow

0751	First spindle alarm detected (AL-XX)
0754	Spindle abnormal torque alarm
1001	The address of the relay or coil is not set
1002	The function instruction of the input code does not exist
1003	The function instruction COM/COME is not used correctly
1004	The user ladder diagram exceeds the maximum number of allowed lines or steps
1005	Improper use of function instruction END1 or END2
1006	There are illegal outputs in the network
1007	Hardware failure or system interruption error causes PLC communication failure
1008	Function instruction is not connected correctly
1009	Network horizontal line is not connected or short-circuited
1010	Power is off during the ladder diagram editing, causing the network being edited to be lost
1011	Address data is not entered correctly
1012	Input symbol is not defined or input address is out of range
1013	Illegal characters are specified or the data is out of range
1014	CTR address is repeated
1015	Function instruction JMP/LBL is not processed correctly or exceeds the capacity
1016	Network structure is incomplete
1017	Currently unsupported network structure appears
1019	TMR address is repeated
1020	Parameter missing in function instruction
1021	When PLC execution timeout, the system automatically stops the PLC
1022	Function instruction name is missing
1023	Address or constant of the function instruction parameter is out of range
1024	There are unnecessary relays or coils
1025	Function instruction is not output correctly
1026	Number of network connection lines exceeds the supported range
1027	The same output address is used in another place
1028	Ladder diagram file format is wrong
1029	Ladder diagram file in use is missing
1030	There are incorrect vertical lines in the network
1031	The user data area is full, and please reduce the capacity of COD instruction data table

1032	The first level of the ladder diagram is too large and cannot be executed in time
1033	SFT instruction exceeds the maximum allowed number of uses
1034	Function instruction DIFU/DIFD is not used correctly
1035	The conversion of the currently opened ladder diagram file is not successful
1036	PLC abnormal stop alarm
1037	The opened ladder diagram is inconsistent with the data parameter setting ladder diagram
1039	Instruction or network is not within the executable range
1040	Function instruction CALL/SP/SPE is not used correctly
1041	Horizontal conduction line is connected in parallel with the node network
1042	PLC system parameter file has not been loaded Inverter