# RC6F series robot controller

## user's manual

Wuxi Xinje Electric Co., Ltd.

Data ID: RC01 20240329 1.3

# Basic Description

- Thank you for purchasing the Xinje RC6F series robot controller.
- This manual provides a comprehensive guide to operating the RC6F series of robot controllers.
- Before using the product, read this manual carefully and program only after fully understanding its contents.
- Please deliver this manual to the end user.

# User Notice

- Only operators with basic electrical knowledge are authorized to perform wiring and related operations on the product. If any unclear usage occurs, please
  Consult our technical staff.
- The examples in manuals and other technical materials are provided for users' reference only and do not guarantee specific actions.
- When using this product with other products, ensure it meets the relevant specifications and principles.
- When using this product, verify that it meets the requirements and is safe.
- Please configure backup and safety features to prevent machine malfunctions or losses caused by this product.

# Disclaimer

- The contents of the manual have been carefully checked, but errors are inevitable, and we cannot guarantee complete consistency.
- We will review the manual regularly and make corrections in future versions.
- If the contents in the manual are changed, please understand that no separate notice will be given.

# Contact way

If you have any questions about using this product, please contact the purchasing agent or office, or directly reach out to Xinje Company.


- General: 0510-85134136
- Hotline: 400-885-0136
- Fax: 0510-85111290
- Website: https://www.xinje.com/
- Mailbox: xinje@xinje.com
- Address: No.816, West Jianzhu Road, Binhu District, Wuxi City, Jiangsu Province

# Catalog

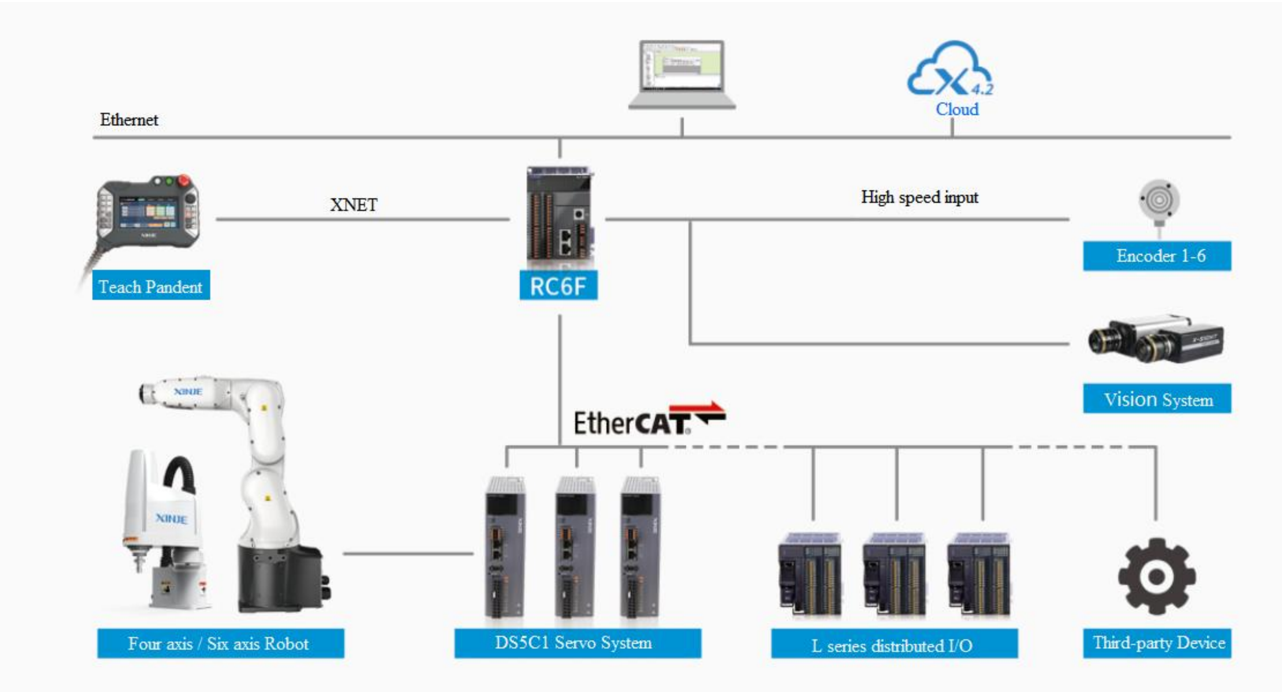
# Preface

Thank you for purchasing the RC6 series industrial robot controller developed by Xinje Electric.

The RC6 series industrial robot controller features a modular design with integrated motion control and PLC-based logic control capabilities. It supports complex motion control through industrial fieldbus protocols including EtherCAT, CAN, and MODBUS. This controller enables simultaneous control of six external axes, making it widely applicable to serial six-axis robots, SCARA systems, and vertical four-axis robots.

The RC6 series industrial robot controller utilizes EtherCAT bus for high-speed communication with XJ Electric's servo system and expansion modules. It connects industrial vision terminals, remote monitoring terminals, and teaching devices through TCP/IP communication, ensuring seamless interoperability. This results in a simple, efficient, and reliable system, making it a cost-effective industrial robot controller.

**The system configuration is as follows:**




**Common functions include:**

This manual covers essential robot functions, including basic motion control commands, JUMP function, arc swinging, external axis operation, external signal interaction, belt tracking, TCP calibration, simple self-calibration, safety zone, non-motion commands, and collision detection.

# Safety Notes

## Security Statement

1. Read and follow these safety precautions before installing, operating, or maintaining the product.
2. To ensure personal and equipment safety, follow all safety precautions indicated on the product and in the manual during installation, operation, and maintenance.
3. The "caution", "warning" and "danger" matters in the manual do not represent all the safety matters that should be observed, but only as a supplement to all safety matters.
4. This product must be used in environments that meet the specified design requirements. Failure to comply may cause malfunctions, and any functional failures or component damage resulting from non-compliance will not be covered by the product's warranty.
5. Our company shall not be held liable for any personal injury or property damage caused by improper operation of our products.

## Security Level Definition

| | |
|---|---|
| ⚠ **Note !** | "Danger" means that failure to follow the instructions may result in death or serious physical harm. |
| ⚠ **Note !** | "Warning" means that failure to follow the instructions may result in death or serious physical harm. |
| ⚠ **Note !** | "Caution" means that failure to follow the instructions may result in minor physical injury or equipment damage. |

## Safety Notes

The specific procedures cover the entire lifecycle: from unpacking and acceptance, through storage and transportation, installation, wiring, power-up, operation, maintenance, repair, to final disposal.

Before using this product, please read this section carefully and understand the product's usage, safety, and precautions before operating. Ensure proper wiring of the product with strict safety precautions.

The safety precautions for potential issues during product use are outlined, categorized as 'Caution' or 'Danger'. For any unresolved matters, please follow the basic electrical operating procedures.

- Ensemble

| ⚠ Warning |
|---|
| 1. Please have the correct installation done by a skilled technician, otherwise there is a risk of electric shock or injury.<br>2. When starting the machine after connecting the mechanism, set it to an emergency stop position at all times to avoid injury.<br>3. If the power is restored after a temporary outage during operation, the machine may restart suddenly. |

Do not approach the machine. Take precautions to ensure personal safety during the restart, as it may cause injury.

4. Never touch the interior of the product, as it may cause electric shock.
5. Therefore, never remove the outer cover, cables, connectors, or accessories while powered on, as this may cause electric shock, malfunction, or damage.
6. Do not damage or forcefully pull the cable, nor subject it to excessive force, place it under heavy objects, or clamp it. These actions may cause electric shock, product malfunction, or damage.
7. Do not modify this product, as it may cause injury or damage to the machine.

- Storage and transportation

Warning

1. **Store in the following environment. Otherwise, it may cause fire, electric shock, or machine damage.**
    - places where there is no direct sunlight.
    - places where the ambient temperature does not exceed the storage temperature conditions.
    - places where the relative humidity does not exceed the storage humidity conditions.
    - places with small temperature difference and no condensation.
    - places where non-corrosive gases and combustible gases are present.
    - places with less dust, dust, salt and metal powder.
    - places where water, oil, and chemicals are not easily splashed.
    - Vibration or shock will not affect the product.
2. When handling the product, always hold the main body. Only holding the cable or connector may cause connector damage, cable breakage, or personal injury.
3. Do not pile this product too high (follow instructions) to avoid injury or machine malfunction.
4. In all transportation scenarios, products must not be exposed to environments containing halogens (fluorine, chlorine, bromine, iodine, etc.), as this may cause malfunctions or damage.
5. When disinfecting and pest control for wooden packaging materials (including wooden frames, plywood, and shelves), always use methods other than fumigation.
    - Example: Heat treatment (core temperature above 56°C for over 30 minutes)
    - In addition, it must be addressed at the material stage before packaging, rather than as a whole after packaging.
    - **When using fumigated wood to package electrical products (either standalone units or those mounted on machinery), the gases and vapors released from the wood can cause irreversible damage to electronic components. Halogen-based disinfectants (fluorine, chlorine, bromine, iodine, etc.) may particularly corrode the internal structures of capacitors.**

- Install

Warning

1. **Install in the following environment. Otherwise, it may cause fire, electric shock, or machine damage.**
    - Sunshine exposure is not recommended.
    - places where the ambient temperature does not exceed the installation temperature conditions.
    - places where the relative humidity does not exceed the installation humidity conditions.
    - places with small temperature difference and no condensation.
    - places where non-corrosive gases and combustible gases are present.
    - places with less dust, dust, salt and metal powder.
    - places where it is not easy to spill water, oil, and medicines.
    - Vibration or shock will not affect the product.
2. Under no circumstances shall the product be exposed to environments containing halogens (fluorine, chlorine, bromine, iodine, etc.), as this may cause malfunction or damage.
3. Do not sit on or place heavy objects on this product, as it may cause injury or machine malfunction.
4. Do not block the air intake and exhaust port. Do not let foreign objects enter the product, otherwise it may cause failure or fire due to aging internal components.

5. Please follow the installation instructions to avoid potential malfunctions.
6. During installation, ensure the product maintains the required clearance from both the control cabinet's interior surface and adjacent equipment, as failure to do so may cause fire hazards or equipment malfunctions.
7. Do not apply strong impact, as it may cause malfunction.
8. Please have the battery installed by a qualified technician to avoid electric shock, injury, or equipment damage.
9. Do not touch the battery electrodes, as static electricity may cause damage.

- Wiring

**Warning**

1. Ensure proper and reliable wiring, as improper connections may cause motor loss of control, personal injury, or equipment failure.
2. Use the specified power voltage to avoid fire or malfunction.
3. When using the device under poor power supply conditions, ensure the input power is supplied within the specified voltage range. Otherwise, there is a risk of damaging the machine.
4. Install circuit breakers and other safety devices to prevent external wiring short circuits, as this could cause fire hazards.
5. Take appropriate shielding measures when using in the following places.
   - when interference is caused by static electricity.
   - A place where a strong electric field or magnetic field is generated.
   - places where there may be radiation.
   - Keep the machine away from electrical wires.
6. When designing the circuit, ensure the CPU unit or module is powered by the 24V input/output power supply before any external power. If the CPU unit or module is powered after the 24V input/output power is connected, its output may turn on momentarily, potentially causing unintended actions that could result in injuries or equipment damage.
7. Ensure the installation of safety protection-related emergency stop circuits, interlock circuits, and limit circuits in the external control circuit of the product. Failure to do so may cause injuries or machine damage.
8. When using the MECHATROLINK input/output module, ensure the established MECHATROLINK communication serves as the interlock output condition. Failure to do so may cause machine damage.
9. Connect the battery with the correct polarity to avoid damage or explosion.
10. Please consider the following factors when selecting input/output signal lines (external wiring) to connect products and peripherals.
    - mechanical strength；
    - noise disturbance；
    - Wiring distance.
    - signal voltage。
11. In order to suppress the noise from the power cable, the control circuit input and output signal cable wiring and routing should be separated from the power cable, whether inside or outside the control cabinet. If the separation is not thorough, it may lead to false action.

- Move

**Warning**

1. Please follow the steps and instructions in the user manual for the product to operate and test-run. If an operational error occurs while the servo motor and mechanical connections are engaged, it may not only damage the machine but also cause injuries or fatalities.
2. Set up interlock signals and other safety circuits on the product's exterior to ensure overall system safety even in the following situations.
   - An abnormal state caused by product failure or external factors.
   - The product detects abnormalities through its self-diagnosis function, halts operation, and either shuts off or maintains the output signal status.

- The output remains either ON or OFF due to the output relay's melting/fusing or the output transistor's damage.
- The product's DC24V output becomes unresponsive due to low voltage caused by overload or short circuit.
- If the product fails to detect abnormal conditions in the power supply, input/output, or memory through its self-diagnostic function, it may cause unintended outputs that could result in injuries, equipment damage, or burn injuries.

- Maintenance and Inspection

⚠ Warning

1. Do not disassemble or repair the product, as this may cause electric shock, injury, or mechanical damage.
2. Do not change the wiring during power supply, otherwise it may cause electric shock, injury, or mechanical damage.
3. Please have a skilled technician replace the battery correctly. Otherwise, it may cause electric shock, injury, or machine damage.
4. Always replace the battery with the controller powered on. If the battery is replaced while the controller is powered off, the data stored in the controller's memory may be lost.
5. When replacing the battery, avoid touching the electrode area to prevent electrostatic damage.
6. When replacing the CPU unit or module, do not skip the following steps.
   - Back up the program and parameters of the CPU unit or module to be replaced.
   - Transfer the saved program and parameters to the new CPU unit or module.
   - When a new CPU unit or module is activated without prior data transfer, it may cause mechanical damage or system failure due to unintended operations.
7. Do not touch the CPU unit's heat sink when powered on or immediately after power-off, as it may become hot enough to cause burns.

- Discard

⚠ Warning

1. Please dispose of it as general industrial waste.
2. Used batteries should be disposed of in accordance with local regulations.

# 1. Product Overview

## 1.1 Product Features

### 1.1.1 Main Features of RC6F Controller

- Higher instruction processing speed RC6F, with a faster instruction processing speed of 0.01~0.03us.
- Supports up to 16 right expansion modules and 1 left expansion module, with control capability for up to 6 external axes.
- Compatible with most standard features of the XC series.
- Compatible with original XC series programs: The Xinje PLC programming tool can directly open the original XC\XL\XD series PLC programs. However, some instructions in the program may differ between RC6F and XC\XL\XD series, which will be highlighted in red error messages in the software. You only need to manually modify these parts.
- The RC6F features a compact design that maximizes space efficiency. With its sleek, slim profile, it significantly reduces installation space requirements.
- The X-NET bus, with RC6F support, enables X-NET fieldbus communication.
- The RC6F Ethernet communication module features an RJ45 port and supports TCP/IP protocol, enabling Ethernet-based MODBUS-TCP and free-form communication. It supports program download, online monitoring, remote monitoring, and communication with other TCP/IP devices.

### 1.1.2 Hardware Specifications

| Project | | Specifications |
|---|---|---|
| Product model | | RC6F |
| Programming Method | | Command, Ladder Diagram and C Language |
| Processing speed | | 0.01~0.03us |
| Power outage | | Use FlashROM |
| User program capacity | | 4MB |
| Source | Rated voltage | DC24V |
| IO check the number | Total points | 30 |
| | Input points | 14 |
| | Output points | 16 |
| High speed counting input | OC import | Channel 2 (single-phase up to 80kHz, AB phase up to 50kHz) |
| | differential input | The maximum frequency for channel 2 is 1 MHz. |
| High speed positioning | | 4-axis 100kHz |
| Expansion capability | | Right expansion module *16 + left expansion module *1. supports up to 6 external axes |
| Suspend | | External, Timed, High-Speed Count, Pulse |
| Communication | communication port | 1 RS232 port, 1 RS485 port, 2 RJ45 ports |
| | protocol | Standard MODBUS ASCII/RTU, free-form communication, Ethernet, TCP/IP, UDP |
| Nus function | | Point-to-point, linear interpolation, circular interpolation, JUMP, transition, forward, swing arc, and collaboration |
| Robot control function | | Joint coordinates, Cartesian coordinates, user coordinates, tool coordinates, workpiece coordinates |

 User Manual of RC6F Series Robot Controller 2 Body Specifications

# 1.2 Model Composition and Model List

## 1.2.1 Model Composition and Model Table of RC6F Controller

1) Naming rules

RC 6 F
① ② ③

| | |
|---|---|
| ①: product line | RC: industrial robot |
| ②: Version code | 6: The sixth generation |
| ③: Supported device models | F: 4x6 universal |

2) Basic unit of RC6 series robot controller

| Series name | product model |
|---|---|
| RC6 series | RC6F |

## 1.2.2 Composition of Extended Unit Models and Model List

1) Model composition of I/O expansion modules

XL－E ○ □ ○ □－□
① ② ③ ④ ⑤ ⑥ ⑦

| | | |
|---|---|---|
| ①: | Series name | XL: XL Series Expansion Module |
| ②: | Referential extension module | E: indicates the expansion module |
| ③: | Input points | 8 or 16 or 32 |
| ④: | Input point type | X: indicates the input type as NPN<br>PX: indicates the input type is PNP |
| ⑤: | Output points | 8 or 16 or 32 |
| ⑥: | Output form | YT: Transistor Output<br>YR: relay output |
| ⑦: | Interface type | No: European terminal interface<br>A: A bull-jaw connector requiring an external terminal block |

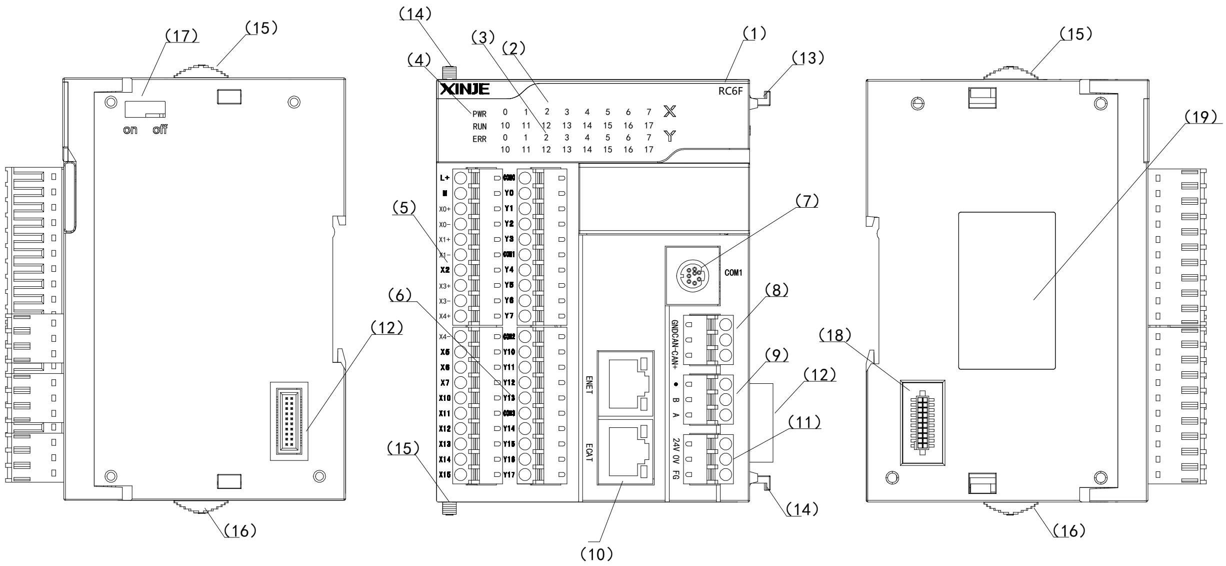
2) List of I/O expansion module models

| Model | | | | Total number of input/output points | Input points (DC24V) | Output points (R,T) |
|---|---|---|---|---|---|---|
| Type | Import | Output | | | | |
| | | Relay output | Transistor output | | | |
| NPN mould | - | XL-E8X8YR | XL-E8X8YT | 16 | 8 | 8 |
| | XL-E16X | - | - | 16 | 16 | - |
| | - | XL-E16YR | XL-E16YT | 16 | - | 16 |
| | - | - | XL-E16YT-A | 16 | - | 16 |
| | - | - | XL- | 32 | 16 | 16 |

| Model | | | | Total number of input/output points | Input points (DC24V) | Output points (R,T) |
|---|---|---|---|---|---|---|
| Type | Import | Output | | | | |
| | | Relay output | Transistor output | | | |
| | | | E16X16YT | | | |
| | - | - | XL-E16X16YT-A | 32 | 16 | 16 |
| | XL-E32X | - | - | 32 | 32 | - |
| | XL-E32X-A | - | - | 32 | 32 | - |
| | - | - | XL-E32YT | 32 | - | 32 |
| | - | - | XL-E32YT-A | 32 | - | 32 |
| PNP mould | - | XL-E8PX8YR | XL-E8PX8YT | 16 | 8 | 8 |
| | XL-E16PX | - | - | 16 | 16 | - |
| | - | - | XL-E16PX16YT | 32 | 16 | 16 |
| | XL-E32PX | - | - | 32 | 32 | - |

# 1.3 Explanation of each section

Product Structure




**The names of the sections are as follows:**

(1): Robot controller model
(2): Enter labels and indicator lights
(3): Output labels and indicator lights
(4): System indicator light
PWR: Power Indicator
RUN: Run indicator light
ERR: Error indicator light
(5): Input terminal
(6): Output terminal
(7): RS232 communication port (COM1)
(8): CANopen communication port (COM6)
(9): RS485 communication port (COM2)
(10): Communication ports 4 and 5 (RJ45)
(11): Power input terminal
(12): Right expansion module interface
(13): Fixed module hook (upper)
(14): Fixed module hook (lower)
(15): Slide Lock (Top)
(16): Slide Lock (Lower)
(17): DIP switch
(18): Left expansion module interface
(19): Product Label

The side switch on the controller's body determines its role as a terminal during RS485 communication. When the controller is at the bus's start or end, set the switch to ON.

# 2. Specifications and parameters of the entity

## 2.1 Specifications and Parameters

### 2.1.1 General Specifications

| Project | Specifications |
|---|---|
| Insulation voltage | DC 500V,2MΩ or higher |
| Anti-noise | Noise voltage 1000Vp-p 1us pulse 1 minute |
| Air | Non-corrosive, flammable gas |
| Ambient temperature | 0℃~55℃ |
| Ambient humidity | 5% RH to 95% RH (no condensation) |
| Communication port 1 | RS232, MODBUS, connect to the host computer, human-machine interface programming or debugging |
| Communication port 2 | RS485 and MODBUS/X-NET fieldbus connect smart meters, frequency converters, and other devices. |
| Communication port 3 | Used to connect the left extended ED module |
| Communication port 4 | RJ45, TCP/IP protocol, supports Ethernet-based X-NET communication, MODBUS-TCP communication, and free-form communication. |
| Communication port 5 | RJ45, supports EtherCAT bus control |
| Communication port 6 | CANopen, supports CAN bus control (not yet supported) |
| Install | Install directly on the rail |
| Landing | Third type of grounding (not shared with high-voltage systems) |

### 2.1.2 Performance Specifications

| Project | | Specifications | |
|---|---|---|---|
| Program execution mode | | Circular scanning | |
| Programming method | | Combine instructions and ladder diagrams | |
| Processing speed | | 0.01~0.03us | |
| Power outage | | Use flashrom | |
| User program capacity ① | | 4mb | |
| I/O check the number ② | Total points | 30 points | |
| | Input points | 14:00 x0~x15 | |
| | Output points | 16:00 y0~y17 | |
| Inner coil (x) ③ | | 1280 points | X0~x77、x10000~x11777、x20000~x20177、x30000~x30077 |
| Inner coil (y) ④ | | 1280 points | Y0~y77、y10000~y11777、y20000~y20177、y30000~y30077 |
| Internal coil (m, hm) | | 220,000 points | M0~m199999【hm0~hm19999】⑤ |
| Flow path (s) | | 22000 points | S0~s19999【hs0~hs1999】⑤ |
| Timer | Check the number | 22000 points | T0~t19999【ht0~ht1999】⑤ |

| Project | | Specifications | |
|---|---|---|---|
| (t) | Specifications | 100ms timer: Set time from 0.1 to 3276.7 seconds<br>10ms timer: Set time from 0.01 to 327.67 seconds<br>1ms timer: Set time from 0.001 to 32.767 seconds | |
| Counter (c) | Check the number | 22000 points | C0~c19999【hc0~hc1999】⑤ |
| | Specifications | 16-bit counter: Set value k0~32,767<br>32-bit counter: Set value from-2147483648 to +2147483647 | |
| Data register (d) | | 1,000,000 words | D0~d2000000【hd0~hd4394304】⑤ |
| Flashrom register (fd) | | 65,584 words | Fd0~fd65535 |
| | | | Special secret storage register fs0~fs47 |
| High speed processing function | | High speed counting, pulse output, external interrupt | |
| Password protection | | 6-bit ascii | |
| Self-diagnostic function | | Power-on self-check, monitoring timer, and grammar check | |

①: User program capacity refers to the maximum program size for confidential downloads.

②: I/O points refer to the number of terminals through which users can input or output signals externally.

③: X denotes an internal input relay. Any X exceeding the number of I points can serve as an intermediate relay.

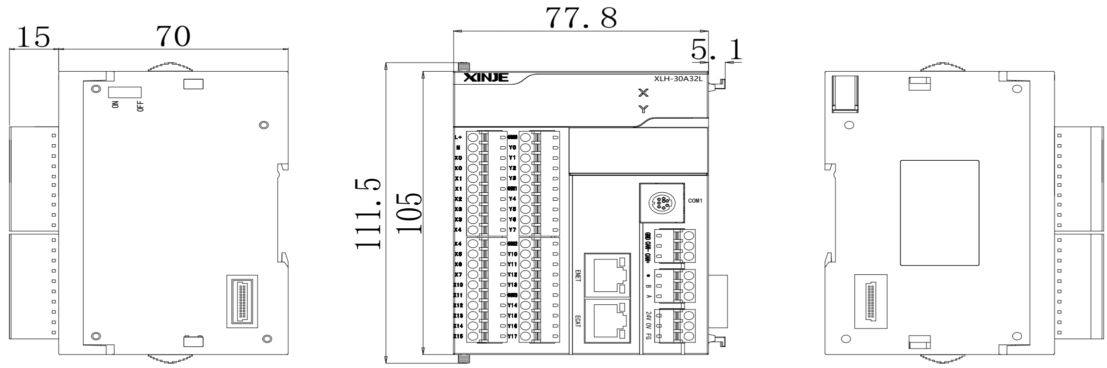
④: Y, referring to the internal output relay, can serve as an intermediate relay when the Y value exceeds the O-point threshold.

⑤: [] is the default power-off retention area and cannot be modified.

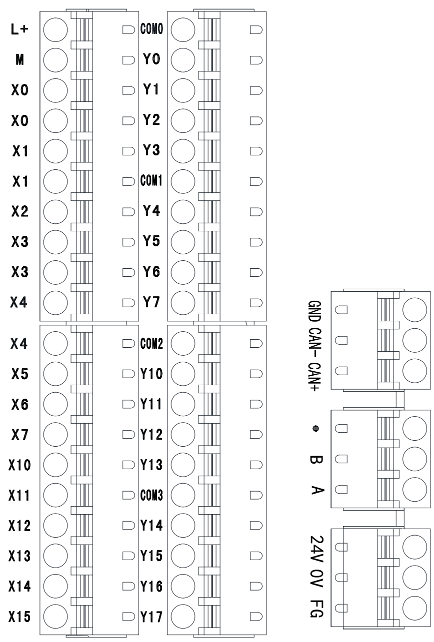
| Special input and output terminals | |
|---|---|
| High speed pulse output bit | Y0-y3 (all remaining y ports are standard transistor output points) |
| Maximum current | 50ma |
| Maximum output frequency of pulse | 100khz |
| Differential input point | Ports x0, x1, x3, and x4 (all other x ports are npn-type collector high-speed input points) |
| Incoming signal | 5v differential signal |
| Enter the highest frequency | 1mhz |

## 2.2 External dimensions

unit： mm




## 2.3 Terminal Arrangement

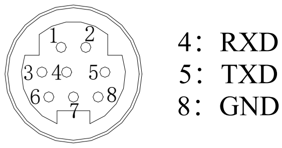



## 2.4 Communication Interface

The RC6F features five communication ports: one RS232 serial port (COM1), one RS485 port (COM2), two RJ45 ports (ENT and ECT), and one CANopen port. An additional RS232 or RS485 port (COM3) can be added via the ED module (XL-NES-ED).

The RS232 serial port (COM1) supports both MDOBUS and X-NET communication modes. It can be used to connect computer-based programming software with PLCs, as well as to interface with touchscreens and various instruments.

1) The pin layout of communication port COM1 is shown below:

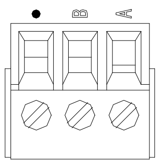



2) The cable wiring for programming is as follows:


Mini Din 8-pin plug (pin) / DB9 plug (socket) Mini Din 8-pin plug (pin) / DB9 plug (socket)

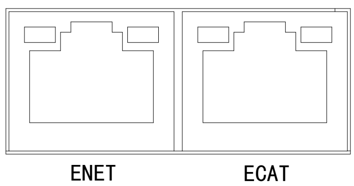
The diagram above shows the DVP line wiring. For the XVP line, connect the front terminal (Mini DIN8) 1 to the rear terminal (DB9) 7.

3) RS485 communication terminal block



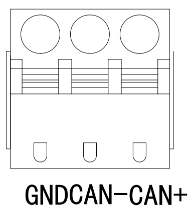

A is RS485+ and B is RS485-. During communication, A connects to A and B to B.

4) RJ45 port (ENET, ECAT)




The ENT port supports TCP/IP-based MODBUS TCP communication, free-form communication, and X-NET fieldbus communication. The ECT port supports EtherCAT bus and can control up to 6 external axes. The CAN port supports CANopen bus, with CAN communication protocol as the default, which can be modified to CAN free-form communication.

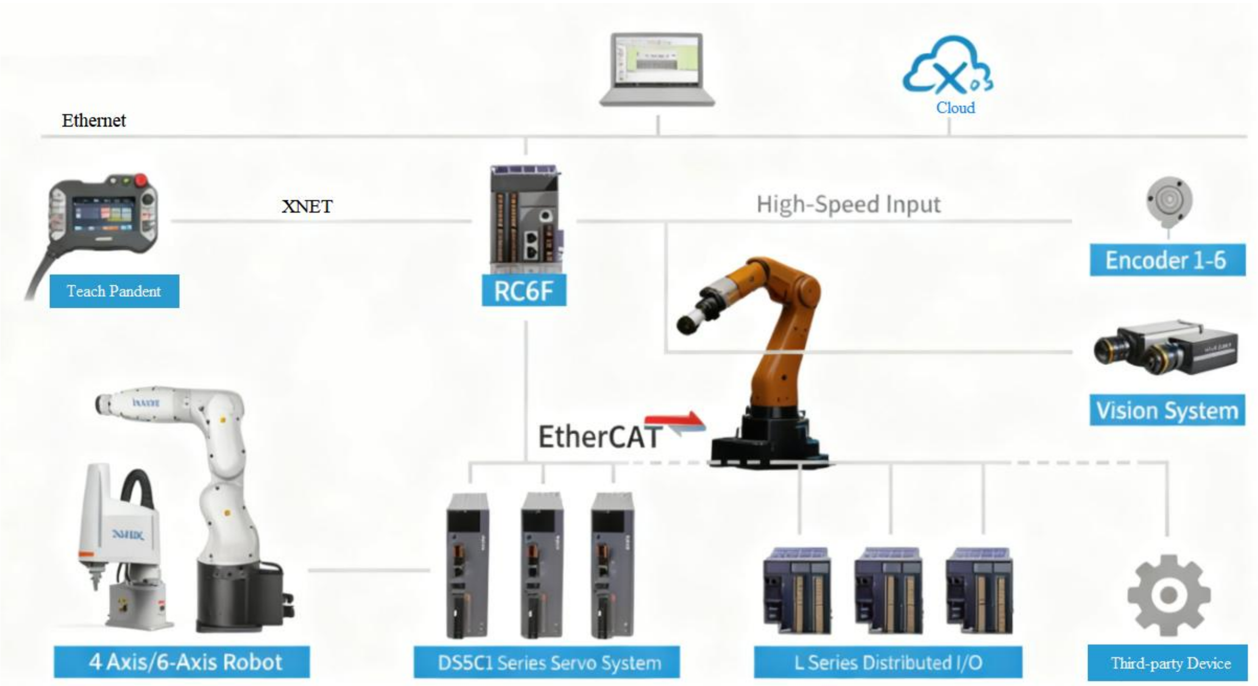
5) CANopen communication terminal block




The CAN port supports CANopen bus (currently unsupported). The default protocol is CAN communication, but it can be modified to CAN free format communication.

# 3. System Composition

## 3.1 System Composition

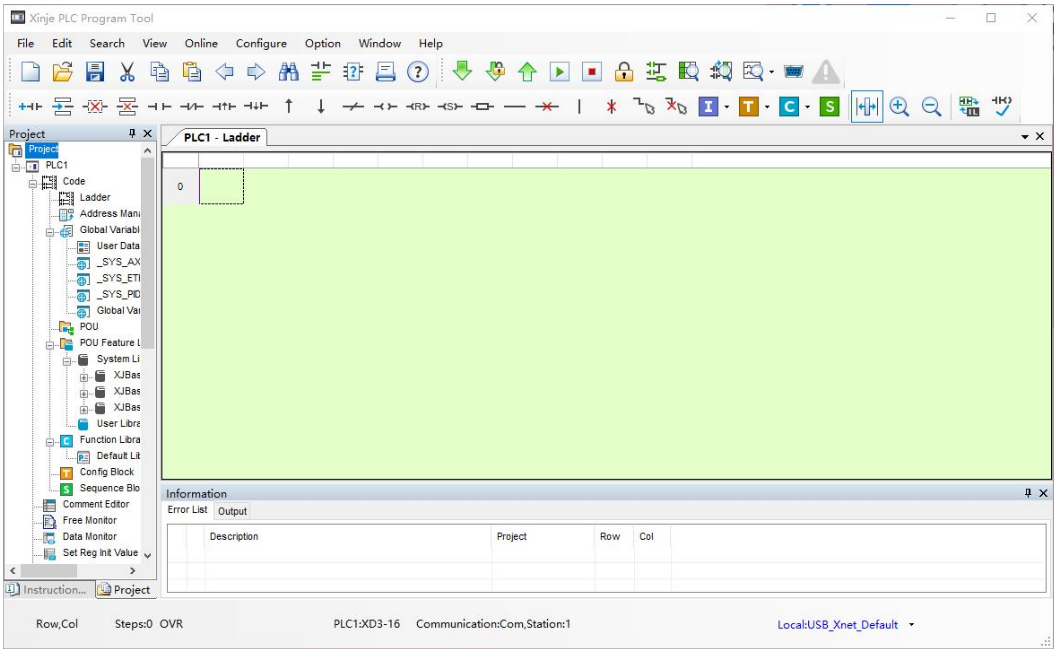



## 3.2 Peripheral Devices

### 3.2.1 Programming software

The software enables programming functions including writing or uploading programs to controllers, real-time monitoring of controller operations, and configuration. After installing the Xinje PLC Programming Tool Software on a personal computer, the controller can be connected to the software via USB download or programming cable through the basic unit's RJ45 port.

Software interface:




## 3.2.2 Human-Machine Interface

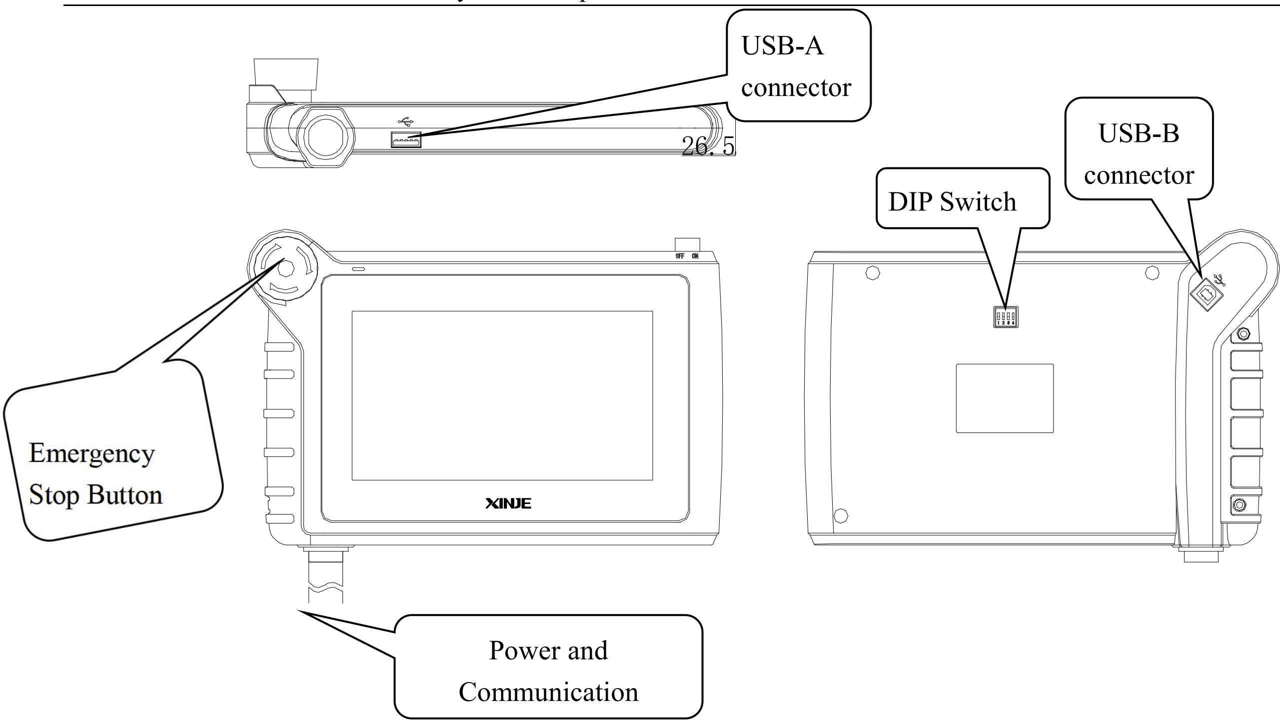
The human-machine interface is the interface to realize the interaction between the controller and the operator. The human-machine interface can easily and quickly deliver the operator's action to the controller, and the controller performs the action.

The basic unit of RC6 series controller supports various human-machine interface (HMI) connections, which are established on the basis of consistent communication protocols. Typically, the MODBUS protocol is used, with specific parameters determined by the connected HMI.

Our company's human-machine interface (HMI) can directly communicate with basic units (with consistent communication parameters). Currently, the robot HMI product is the RT series.

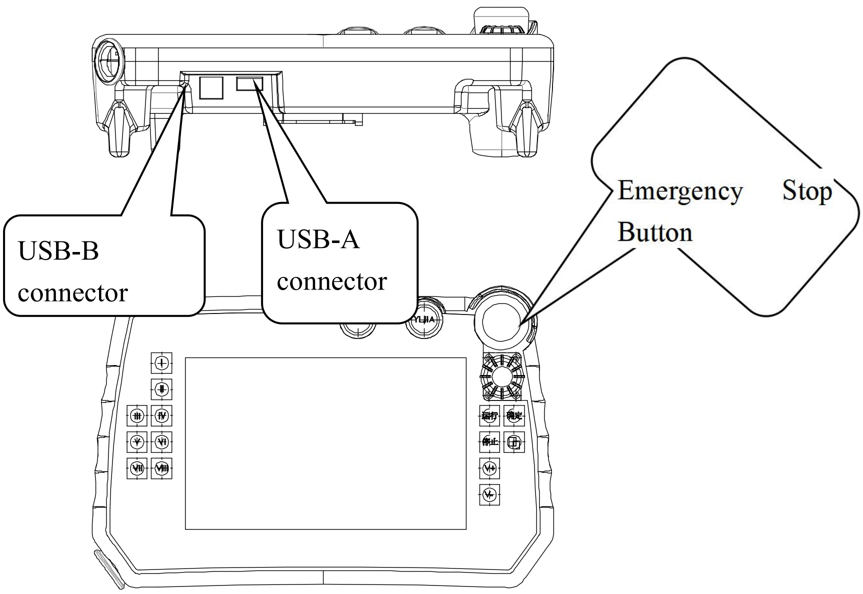
1） RT711

- Size: 7.0″
- Display: 16.77 million colors
- Brightness: adjustable (system register PFW100)
- Touch panel: four-wire resistive touch panel
- Ports: USB-A, USB-B
- Button: Stop button



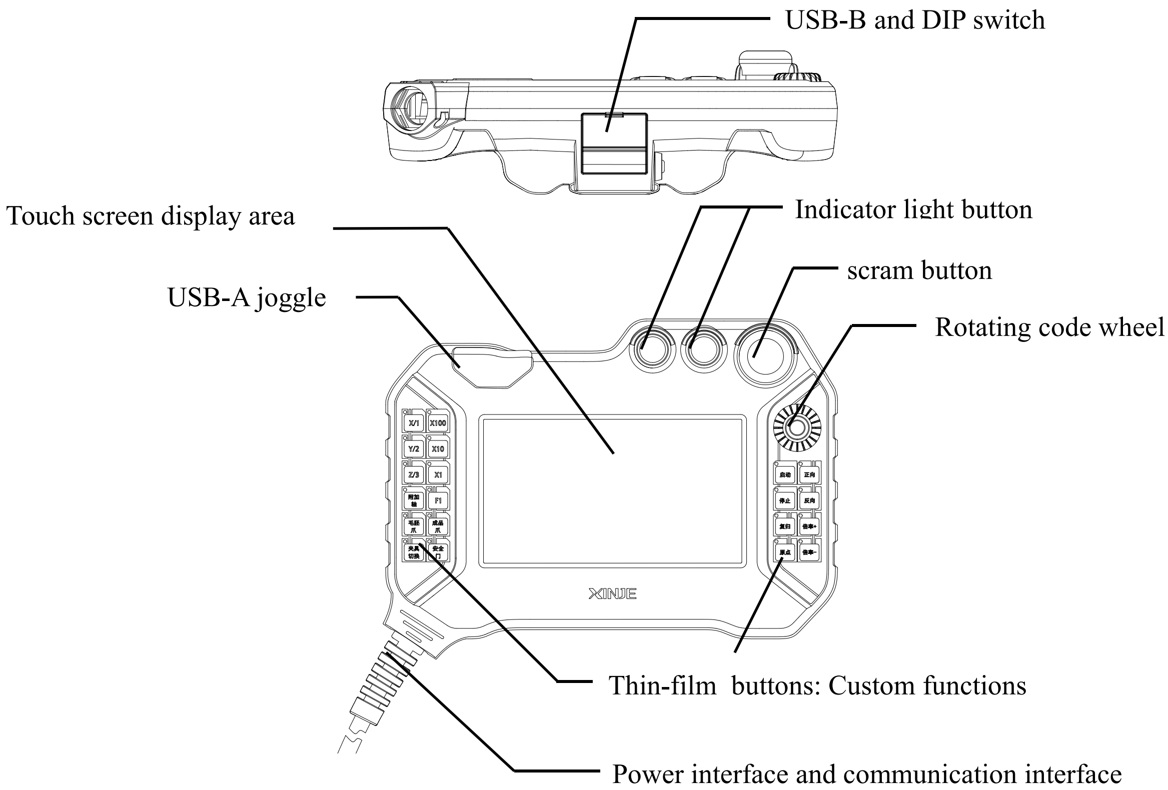

2) RT750

- Size: 7.0″
- Display: 16.77 million colors
- Brightness: adjustable (system register PFW100)
- Touch panel: four-wire resistive touch panel
- Ports: USB-A, USB-B
- Key: 12 function keys, enable button, emergency stop button, code wheel, mode selection, 2 safety switches



3）RT760

- Size: 7.0″
- Display: 16.77 million colors
- Brightness: adjustable (system register PFW100)
- Touch panel: four-wire resistive touch panel
- Ports: USB-A, USB-B
- Key: 20 function keys, 2 indicator light keys, 1 emergency stop button, code wheel, 1 safety switch



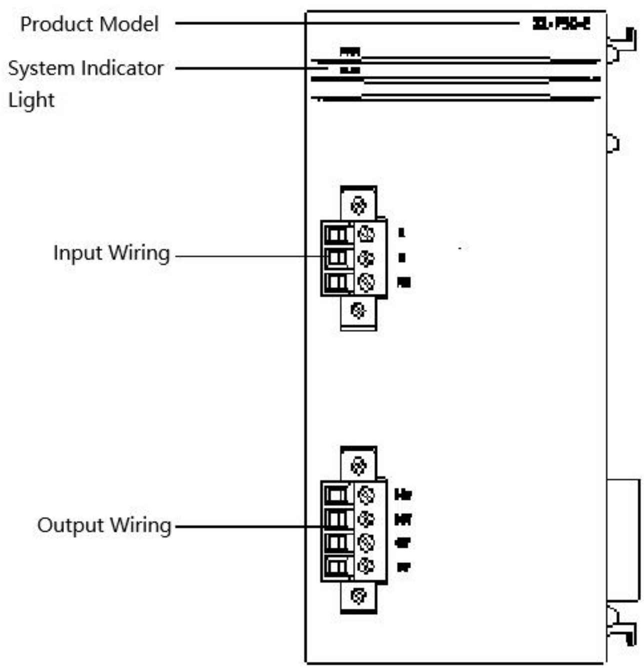
### 3.2.3 Power Supply Adaptation

The RC6 series controller can be powered by either an external switch-mode power supply or the dedicated XL-P50-E power module.

1) Basic Specifications

| Project | Specifications |
|---|---|
| Power supply | AC85-265v |
| Output voltage | DC24v |
| Output | 2A |
| Air | Non-corrosive, flammable gas |
| Ambient | 0℃~60℃ |
| Ambient | 5% rh to 95% rh (no condensation) |
| Install | Install directly on the rail |
| Landing | Third type of grounding (not shared with high-voltage |

2) Structure Description



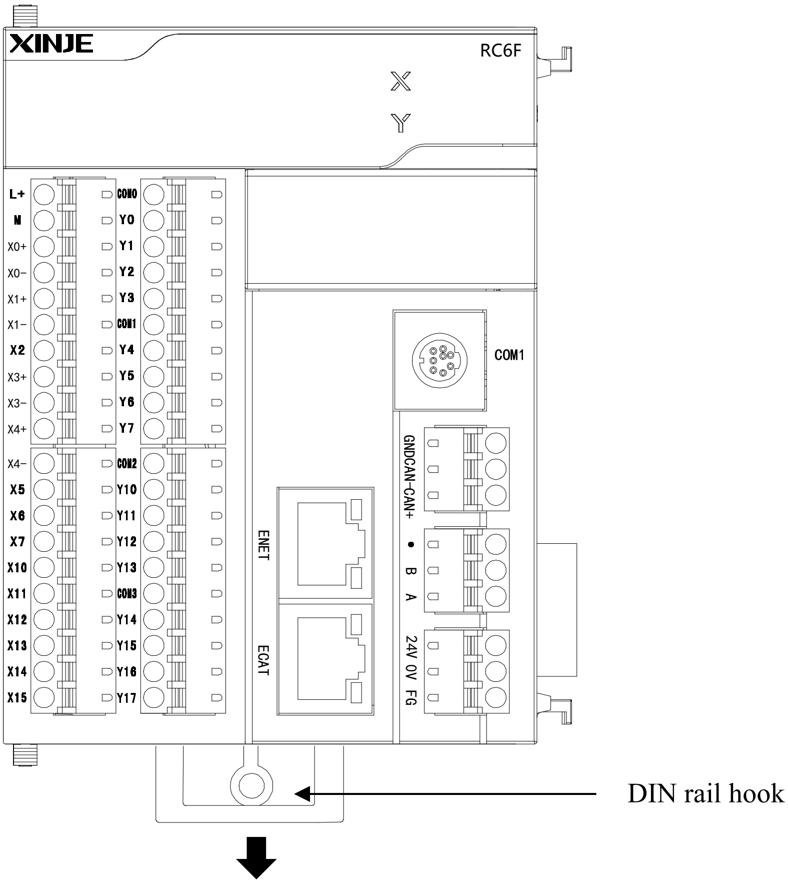

| Structure name | Function declaration |
|---|---|
| Product model | The specific model of the product |
| System indicator | Pwr: Power indicator light. The green light stays on when the power module is powered.<br>Run: The indicator light stays green when the power module is operating normally. |
| Input wiring | L, N: External power input terminals of the power module<br>FG： Earth terminal |
| Output connection | It can output a 24v/0v power supply, primarily powering the XL main unit. |

# 3.3 Assignment of the Expansion Module Definition Number

| Identification sign | Name | Scope | | Check the number |
|---|---|---|---|---|
| X | Input points | XD<br>XL | X10000~x10077 (#1 extension module)<br>......<br>X11000~x11177 (extension module #10)<br>......<br>X11700~x11777 (#16 expansion module) | 1024 |
| | | XD | X20000~x20077 (bd extended #1)<br>X20100~x20177 (#2 extended bd) | 128 |
| | | XD<br>XL | X30000~x30077 (extended ed #1) | 64 |
| Y | Output points | XD<br>XL | Y10000~y10077 (#1 extension module)<br>......<br>Y11000~y11177 (#10 extension module)<br>......<br>Y11700~y11777 (#16 extension module) | 1024 |
| | | XD | Y20000~y20077 (#1 extended bd)<br>Y20100~y20177 (#2 extended bd) | 128 |
| | | XD<br>XL | Y30000~y30077 (#1 extended ed) | 64 |
| ID | Extension module | XD<br>XL | Id10000~id10099#1 expansion module)<br>......<br>Id10900~id10999 (extension module #10)<br>......<br>Ids 11500 to 11599 (extension module #16) | 1600 |
| | Expand bd | XD | Id20000 to id20099 (bd extended #1)<br>Id20100 to id20199 (bd extended version #2) | 200 |
| | Expand ed | XD<br>XL | Id30000~id30099 (extended ed #1) | 100 |
| QD | Extension module | XD<br>XL | Qd10000~qd10099#1 expansion modules<br>......<br>Qd10900~qd10999 (#10 extension module)<br>......<br>Qd11500~qd11599 (#16 expansion module) | 1600 |
| | Expand bd | XD | Qd20000~qd20099 (#1 extended bd)<br>Qd20100~qd20199 (#2 extended bd) | 200 |
| | Expand ed | XD<br>XL | Qd30000~qd30099 (#1 extended ed) | 100 |

# 3.4 Product Installation

Use DIN46277 guide rails for installation.




This unit and its extension module are mounted on a DIN46277 rail (35mm wide). To remove it, simply pull down the assembly hook on the DIN rail and slide the product to the right.

# 4. ECAT parameter configuration

## 4.1 Overview of Functions

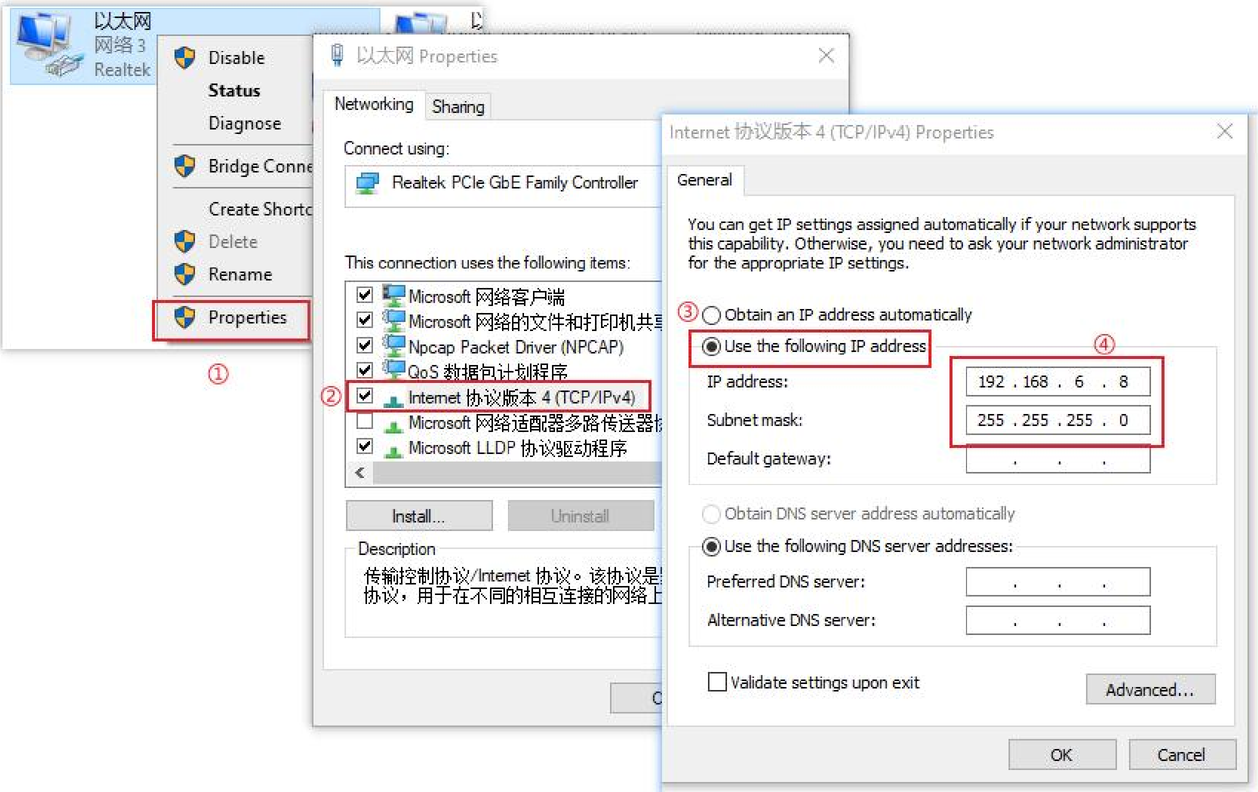
It is used to establish the communication between PLC and servo.

## 4.2 Instructions for Use

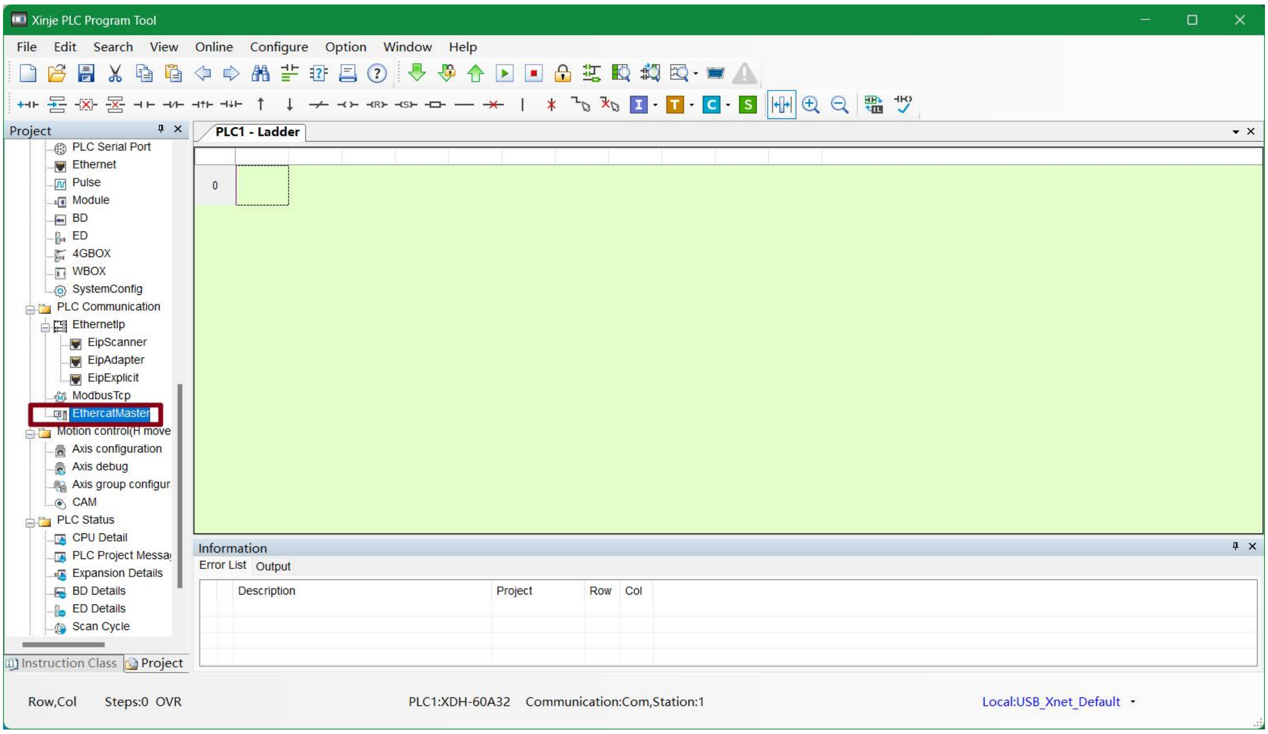
XDPPro selects version 3.7.14 2021-09-10 and above. The XDPPro host computer software that does not support 3.8.0a is used. The PLC Ethernet port is connected to the computer.

1. Go to the 'Network and Internet Settings' in your computer and locate 'Change Adapter Options'.

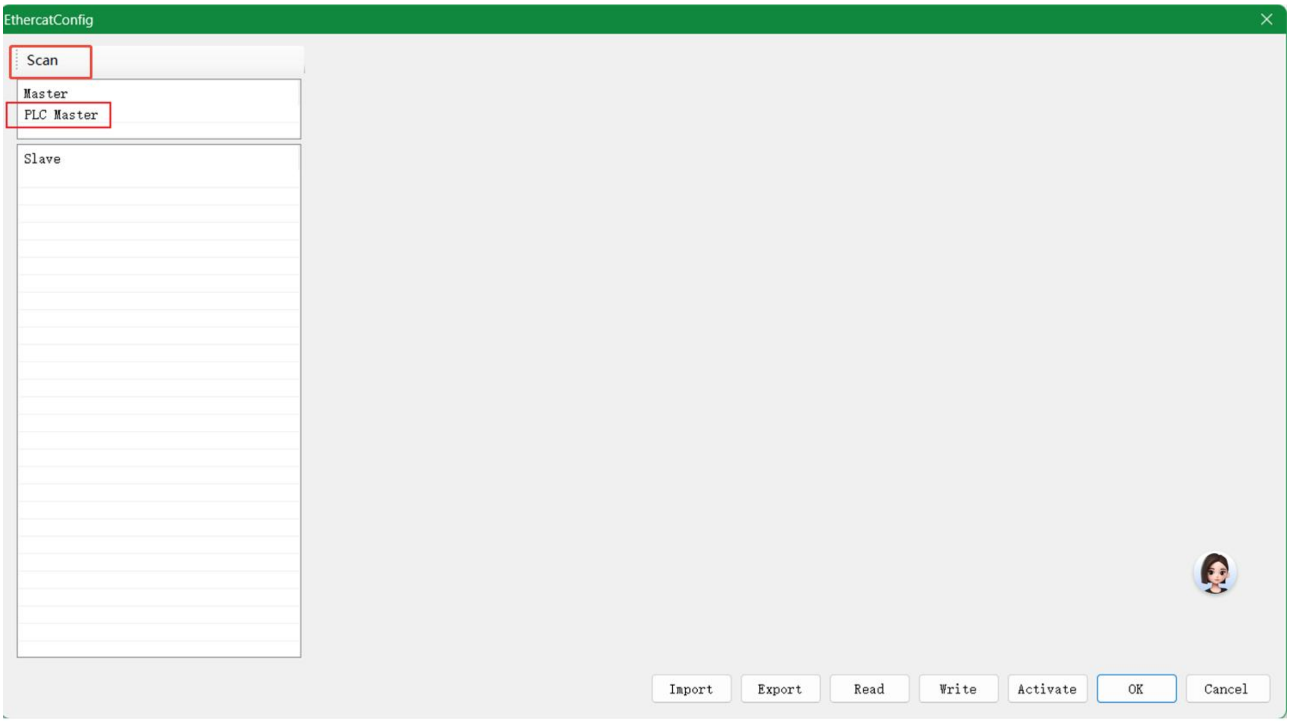
2. As shown in the diagram, click in sequence. Note that the fourth digit of the IP address in option 5 cannot be 6, meaning 192.168.6.6 is an invalid address.



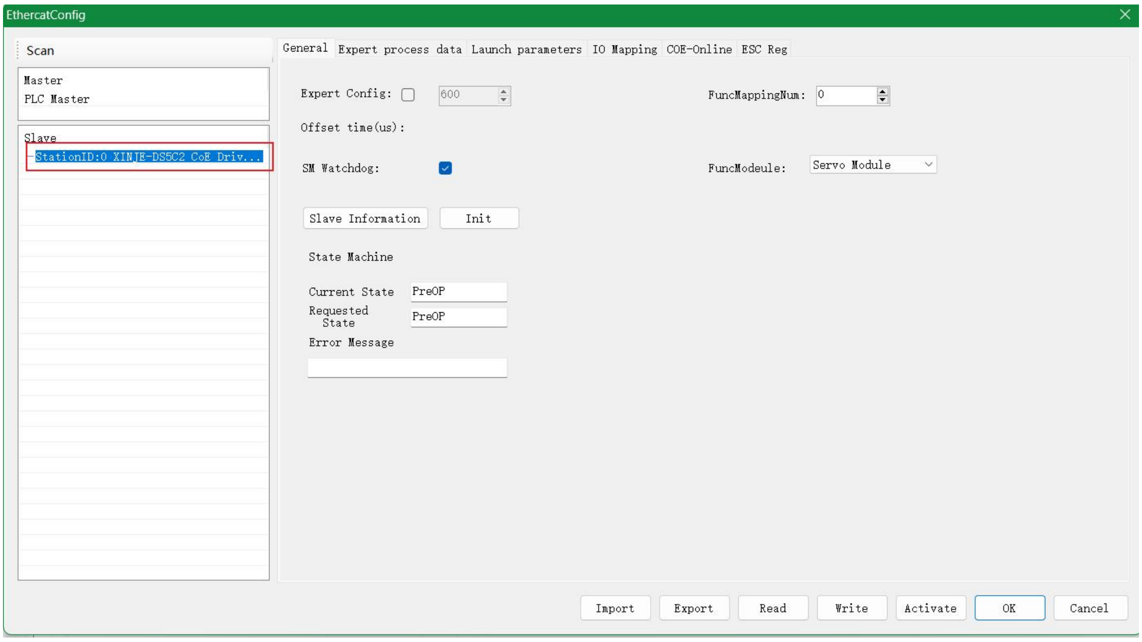
3. Launch the XDPPro software and select [EtherCAT] from [PLC Configuration] in the lower-right corner.



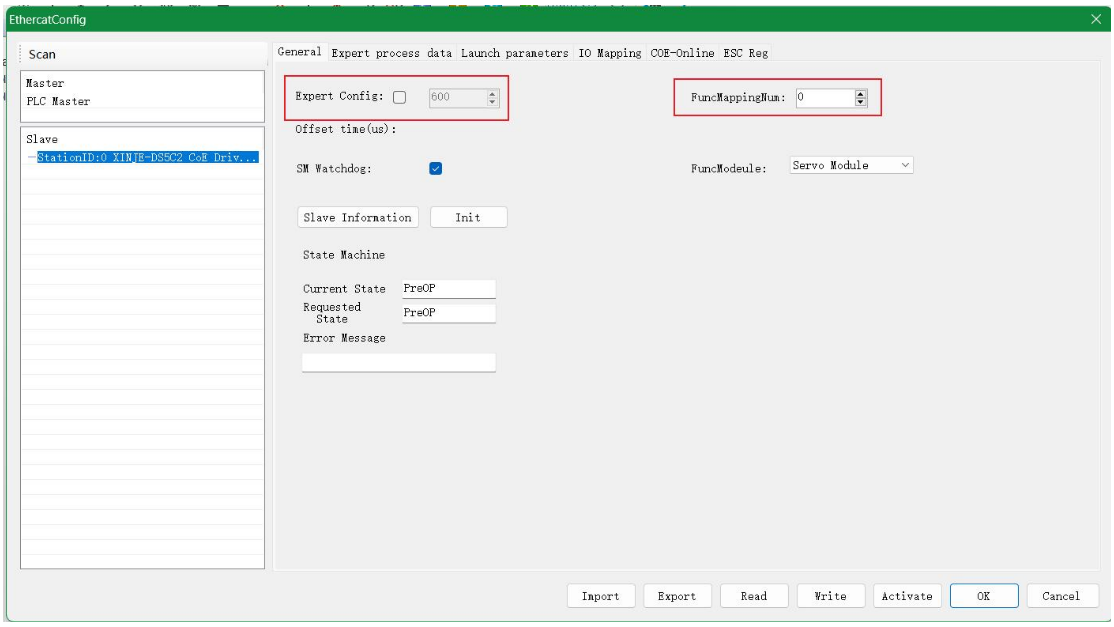
4. In the EtherCAT parameter configuration interface, click 'Scan' to display a list of slaves, with the number of slaves displayed.



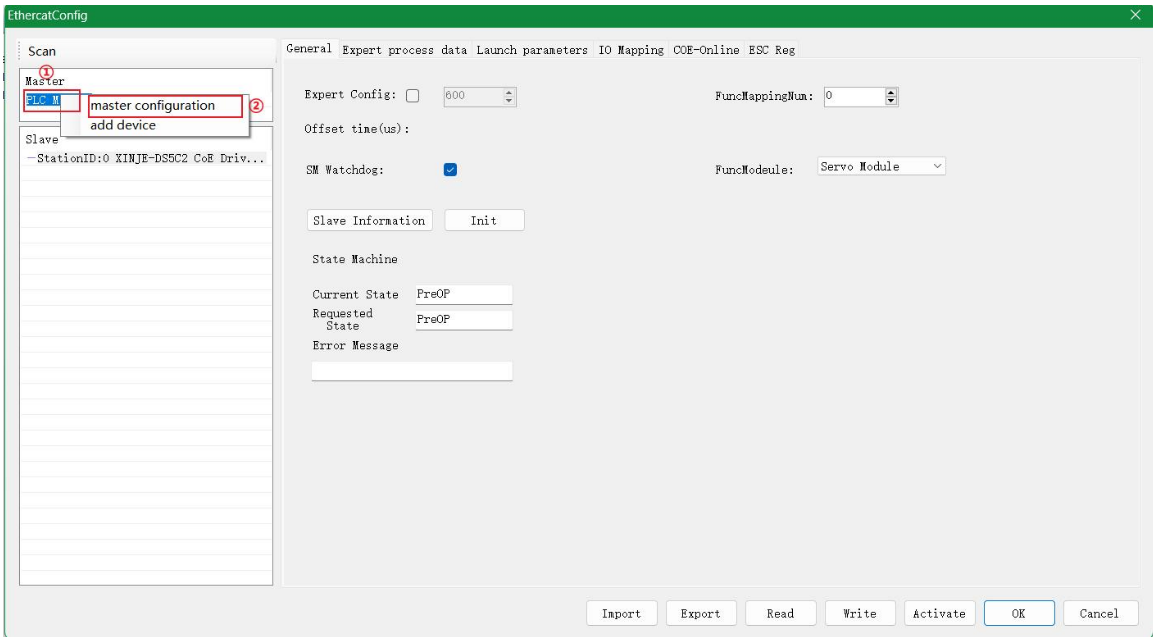
5. Select the slave station from the [From Station] list.



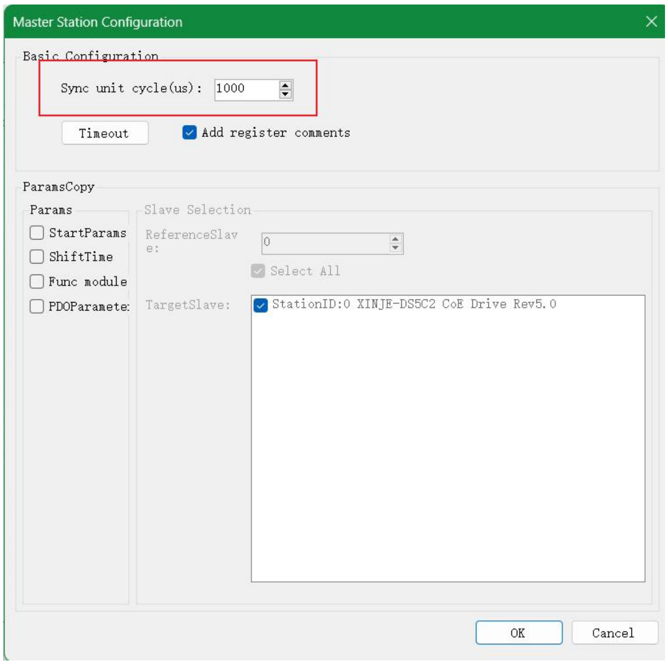

6. Modify the slave's [Offset Time] to 1000 and update the mapping number. The mapping numbers are sequentially assigned from 0 to 10 (0-5 for robot axes, 6-7 for External Axis System 1, 8 for ground rail axis, 9-10 for External Axis System 2). Slave "One-to-One" configuration: Each slave connects to one servo driver. Slave "One-to-Many" configuration: Supports one slave controlling two or three servo drivers. Specifically, 4-axis models use "One-to-Two" (two slaves controlling four axes), while 6-axis models use "One-to-Three" (two slaves controlling six axes). Note: Mapping numbers are fixed in "One-to-Many" configurations, whereas "One-to-One" slaves allow modification of function mapping numbers for axis switching.



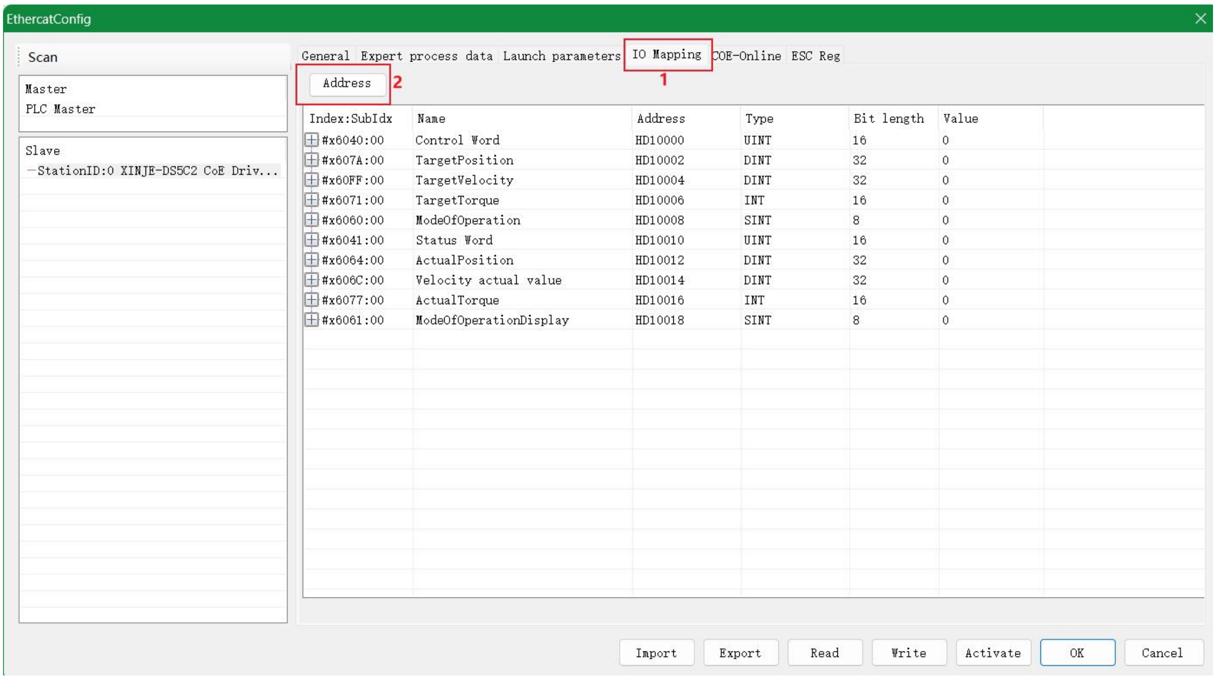

7. Right-click [PLC Master] in [Main Station], then select [Main Station Configuration].



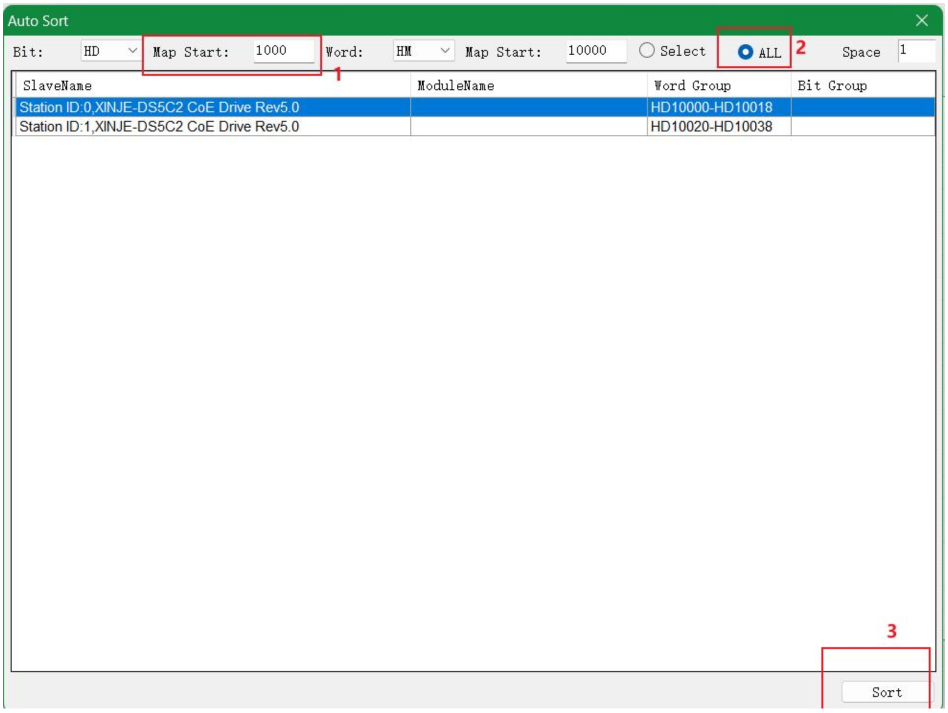

8. In the interface that appears, set the [Sync Unit Cycle] to 2000 and click [Confirm].



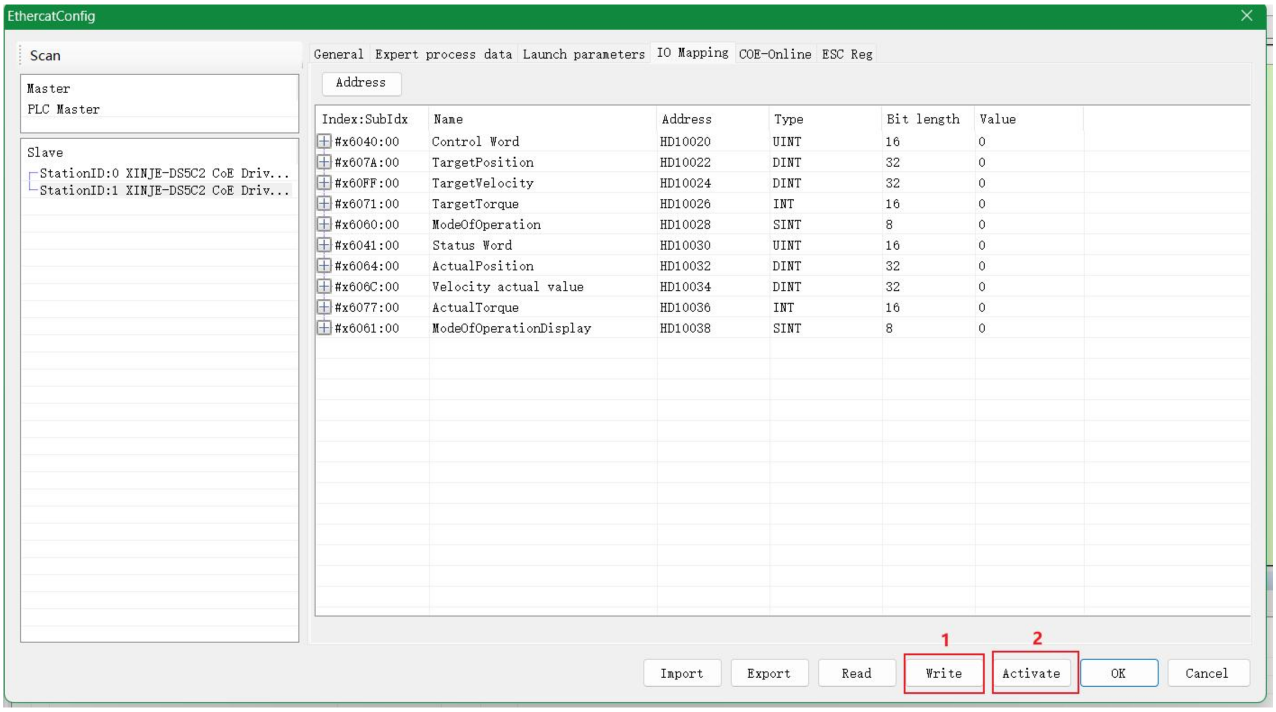

9. Select [Address Configuration] from [IO Mapping].



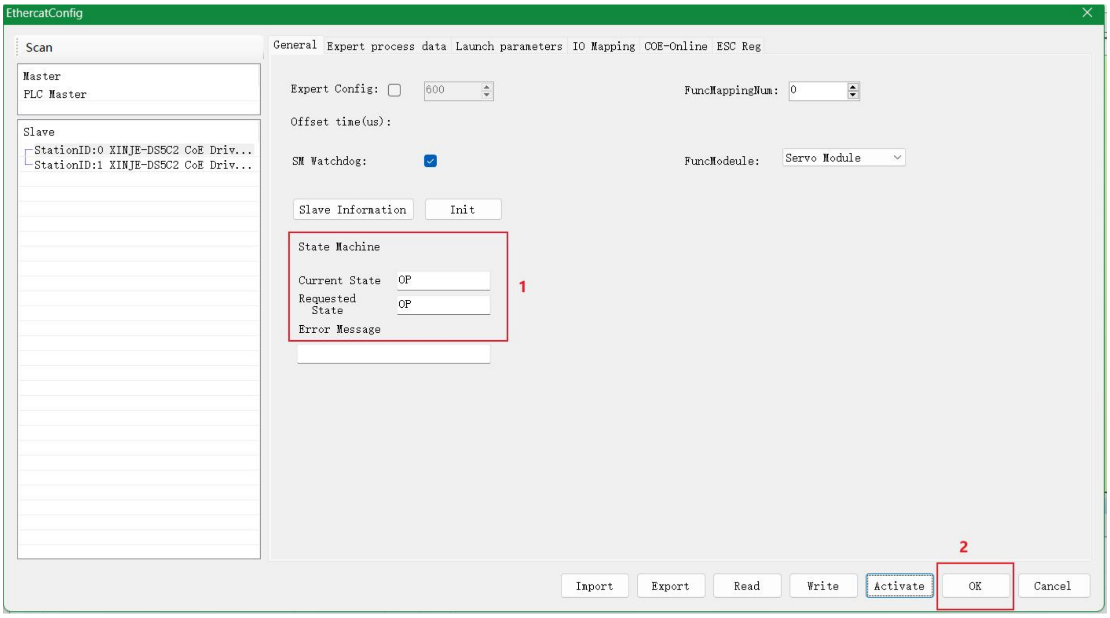
10. In the interface that appears, change [Offset Start] to 99500, select [All Slaves] with an interval of 1, click [Auto Sort], and then close the interface. (Note: The first address can be modified but cannot be reused.)



11. Click [Configure Write]. After writing, click [Activate] and wait for activation. Subsequent configuration changes will also activate the configuration when triggered by SM3212.



12. Click [Common] to view the status machines of each slave station. When both [Current Status] and [Request Status] are [OP], click [Confirm].



# 4.3 Guide to Configuring Robot Virtual Joints

## 4.3.1 Applicable Scenarios

Regarding the virtualization of body axes in common robot models, this configuration is only applicable under specific conditions and generally not recommended. The critical requirement is that the virtualized joint axis must maintain a constant joint angle during motion. Failure to do so would cause discrepancies between the actual joint angle and the inverse-engineered angle, resulting in mismatches between the actual and displayed pose. This requirement necessitates theoretical validation. Current simulation results confirm two models can utilize virtualized axes: a serial six-axis robot can virtualize Joint 4, while a collaborative six-axis robot can virtualize

Joint 5. To ensure the virtualized axis maintains a consistent angle, the robot must adopt a specific posture during operation (with the end effector flange perpendicular to the ground), as detailed in the following section.

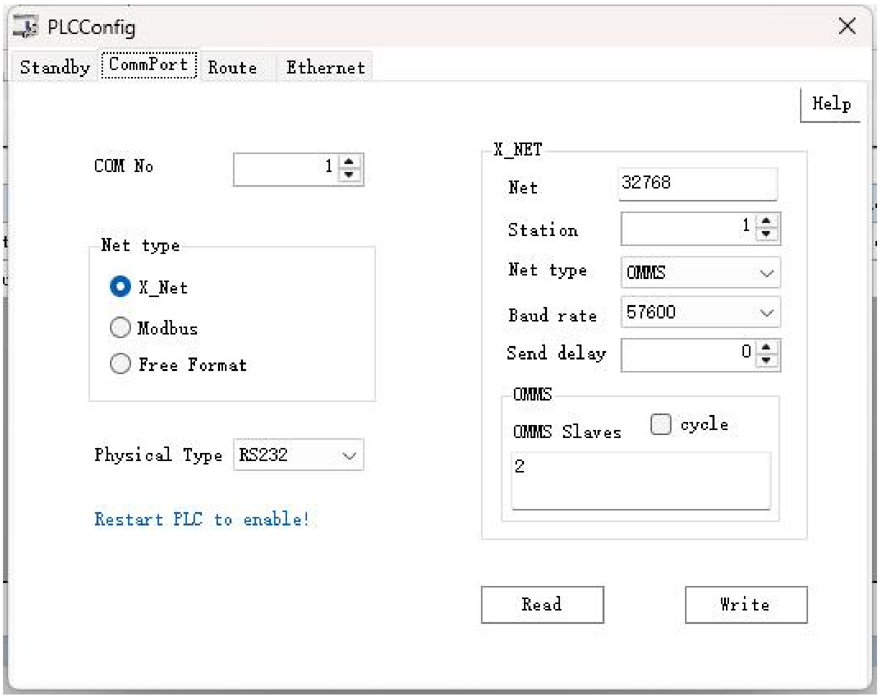
You can use this application if you meet the above conditions. See the following for the specific configuration methods and steps of the robot system.

### 4.3.2 Information Confirmation

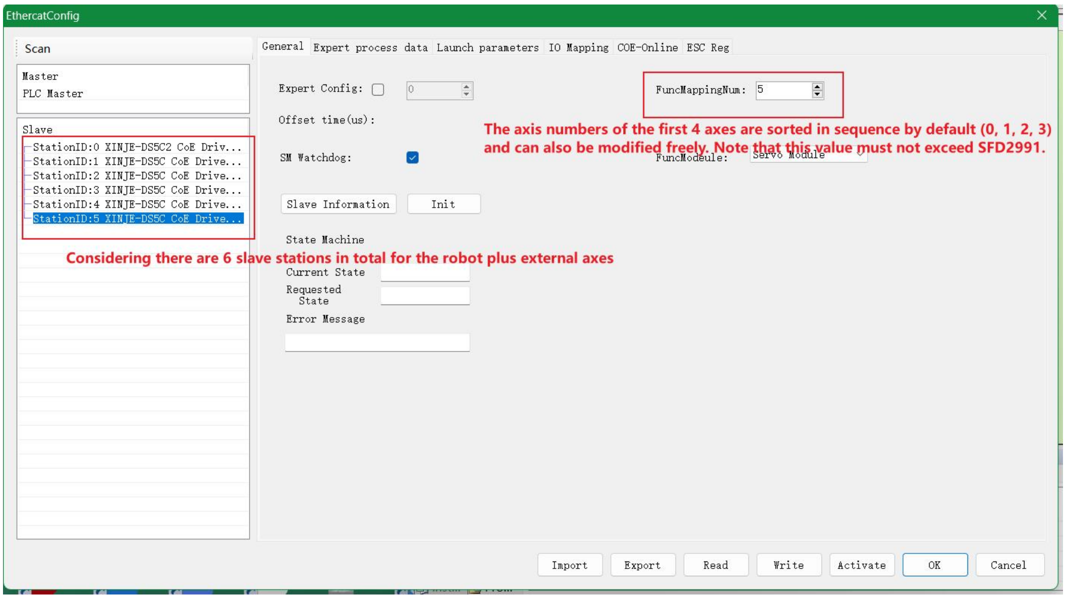
1. Verify the DH parameters, gear ratio, and motor encoder lead length of the robot (configuration errors may cause discrepancies between actual rotation angles and preset values). Refer to the model diagram to confirm the forward rotation direction of each axis (incorrect direction may result in non-linear Cartesian motion). Also check parameters such as joint limit positions.
2. Verify software versions, including firmware, platform system, XDP host computer, and TouchWin version. Report any issues promptly to enhance efficiency.
3. Verify that the motor model matches the driver model and that the XML file is added correctly. (Adding an incorrect model will cause activation failure)

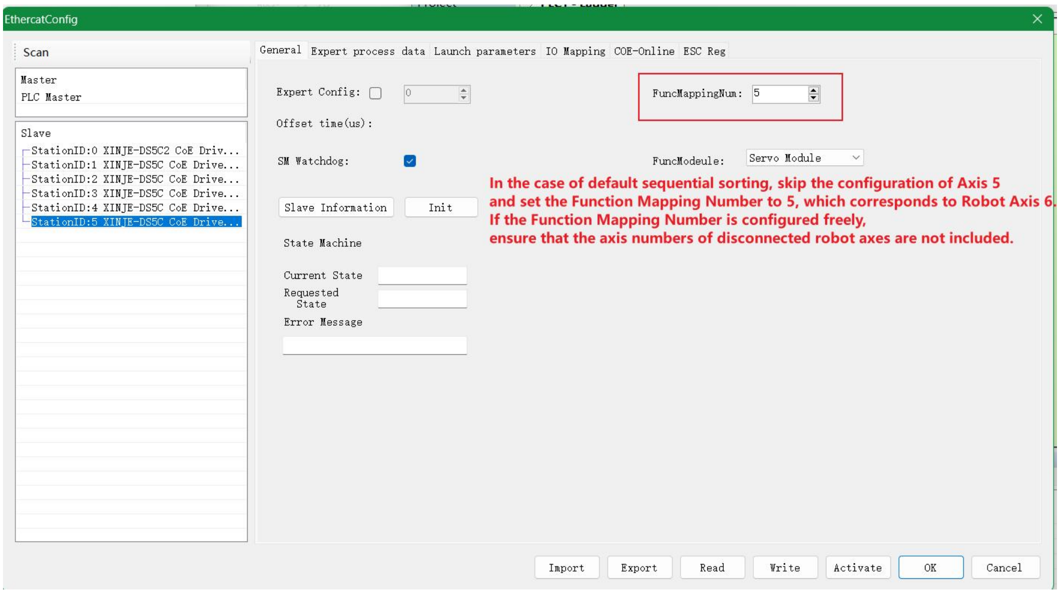
### 4.3.3 Operating Procedures

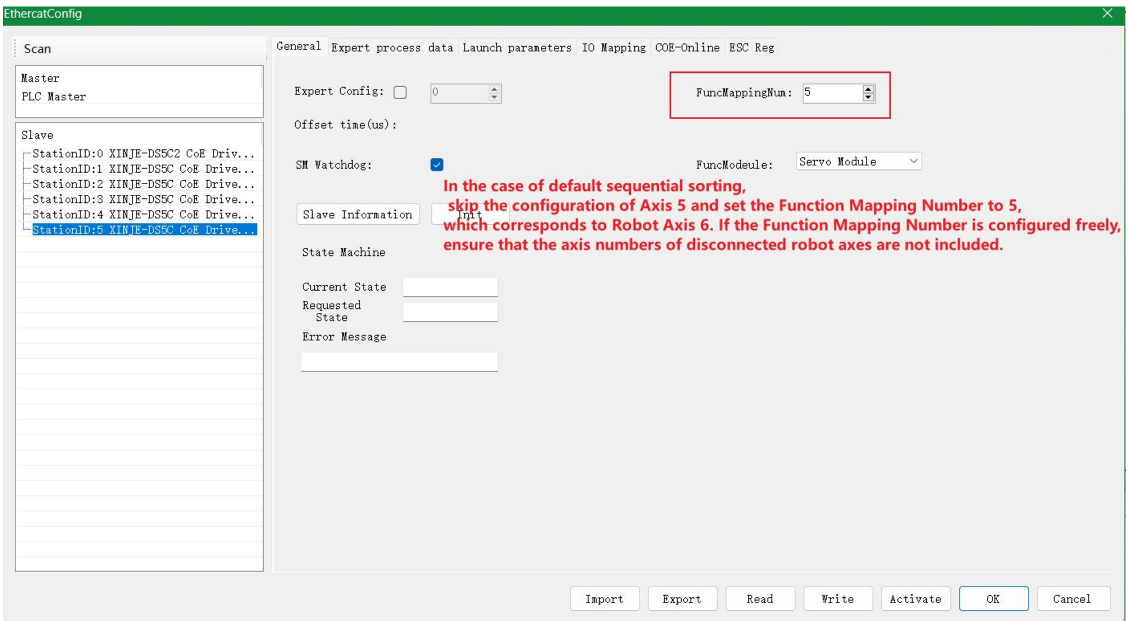
1. Inspect hardware wiring to ensure proper connections without any loose connections. Test the brake function to confirm it operates normally without dropping after power-on. After verification, power on the system and check if the PLC and servo are in normal standby mode. Download the corresponding firmware and platform program, then verify that the PLC panel's ERR light remains off and the servo panel stays in bb state.
2. Configure the teaching device's communication interface as shown in the figure below:




Ensure proper EtherCAT configuration. After scanning all slaves, verify each slave's function mapping number (default starts at 0, corresponding to axis 1, with subsequent axes numbered sequentially). Remove any disconnected axes, skip the current mapping number, and enter the next axis number. The diagram below illustrates a one-to-one slave configuration with axis 5 disconnected.

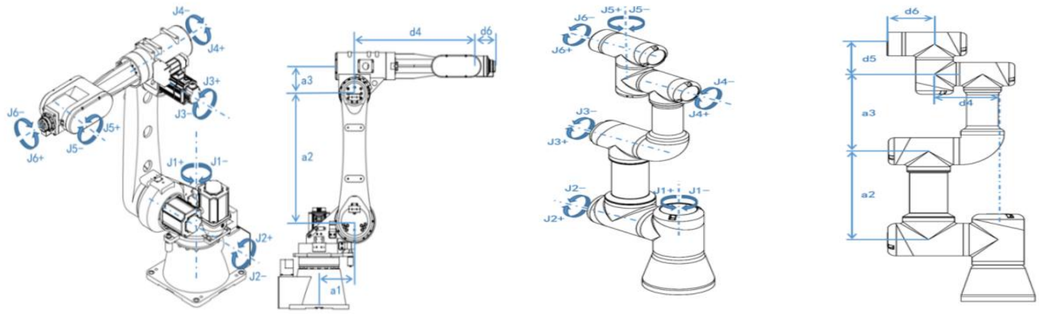



When a shaft connection is disconnected, the shaft simulation mode must be enabled for the SM2018+20i. In the scenario shown above, the SM2098 must be set to ON. Additionally, verify the maximum number of slave devices supported by the SFD2991 register, which should be configured to at least the maximum function mapping number plus 1. Check the write status of other SFD parameters to ensure the robot operates normally.

3. Set the zero position: The standard zero joint angles for a six-axis system are (0, -90,0, 0, 0, 0), while those for a collaborative robot are (0, 0, 0, 0, 0, 0). When using the simulated axis function, however, a specific axis must be disconnected, requiring adjustment of the zero angles.



Ensure the flange end is vertically aligned downward from the ground. The six-axis robot's zero joint angle is (0, -90,0, 0, 90,0), while the collaborative robot's zero joint angle is (0, 0, 90,0, -90,0). If the robot loses its zero position, reset it to the specified joint angles. Note: As the 4th axis of the six-axis robot and the 5th axis of the collaborative robot are disconnected, the robot body cannot move. To simulate movement, simply position the joint to the corresponding zero position in the simulated axis mode.

Detailed steps: (using a 5-axis collaborative robot as an example)

- The J5 of the actual object is kept vertically downward.
- The actual body's J3 movement is perpendicular to the upper arm.
- The other axis moves to the mechanical zero position.
- At this stage, mark all axes on the actual object (excluding those with pre-set mechanical zero positions).
- Shut down the PLC and reset the HSD104 (four-digit display) for axes 1, 2, 4, and 6 to zero.
- Shut down the PLC and set the HSD104 (four-digit) on shaft three to the pulse count corresponding to a 90° joint position.
- Shut down the PLC and set the HSD104 (four-digit) on shaft five to the pulse count corresponding to the 90° joint position.
- Record the single-turn value of SD6400's four-character pulse count for each axis at this time (using the remainder of the pulse count per turn).

Note: As Axis 5 is a virtual axis, its SD6416 value should remain zero.

Calibrate the zero position by using the platform system's "Zero Position Calibration" function.

- Maintain the actual physical position unchanged (update the code to ensure the platform system resets to zero).
- The 'Zero Position Calibration' interface displays 'Single Cycle Position', where the value is the remainder of the SD6400's measurement for each axis.
- After calibration, move each axis to any position and return to the (0,0,90,0, -90,0) point to verify if the mechanical zero mark on the body is correct. (Axis 5 is a virtual axis, so keep it stationary).

Pulse count calculation formula: (SFD3002 + 2i × SFD5020 + 2i × reduction ratio) × zero angle / 360°

4. During the trial run, observe that the flange remains vertically downward with both RX and RY values consistently at 0.

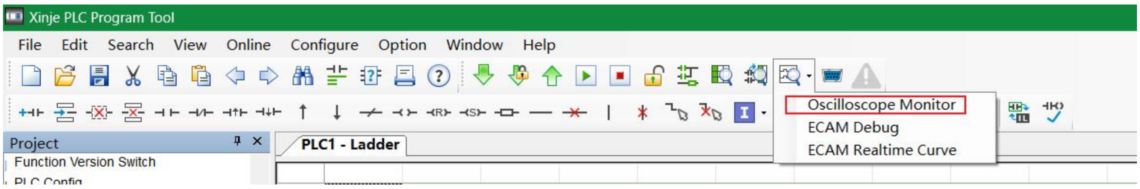
# 5. Oscilloscope functions

## 5.1 Overview of Functions

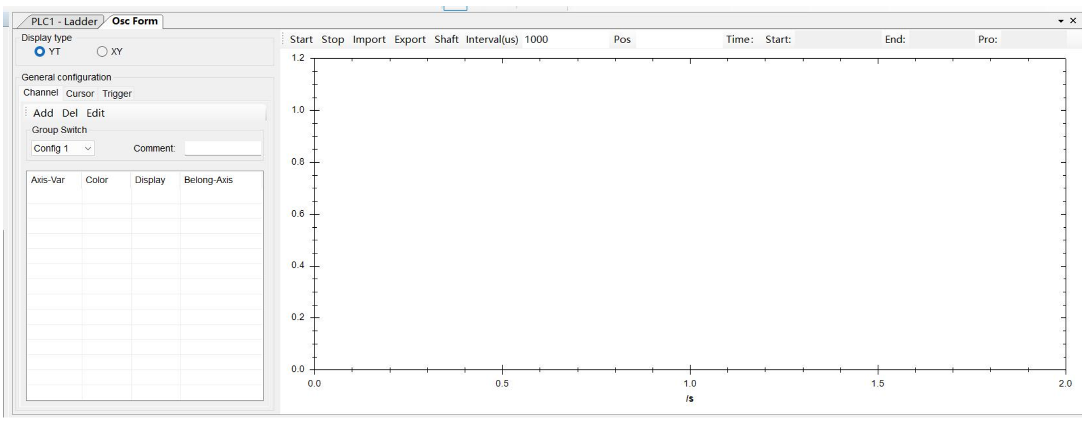
It is used to monitor the continuous change of the register value in a period of time.

## 5.2 Instructions for Use

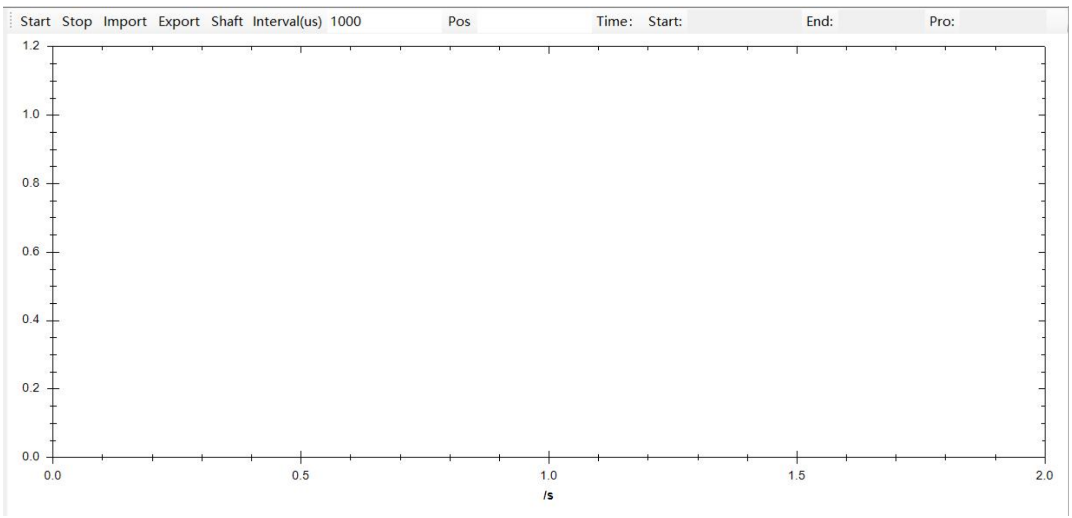
1) Open the oscilloscope function interface




Open the following feature interface:




2) Oscilloscope display unit



| Name | Explain |
|---|---|
| Begin | The oscilloscope is now operational. |
| Cease | The oscilloscope stopped working. |
| Leading-in | Open saved oscilloscope data. |
| Leading-out | Save all oscilloscope data in the current scenario, including curve configuration, cursor, trigger, image data, and operating time. |
| Split axle | Display different y-axis subregions in the same display area.<br>Note: This feature is available only when the curve is configured with multiple axes. If only one axis is assigned, axis splitting cannot be achieved. When users configure different axes, multiple y axes will be displayed. Axis splitting is only available when there are multiple y axes. |
| Time interval (us) | The time interval between two sampling points, measured in microseconds (default value is the synchronization unit cycle in ethercat). |
| Fixed position | Locate a curve at a specific time point (in seconds) or starting from a value. |
| Time | Display start, end, and oscilloscope time. |

Direction for use

<table>
<tr><th>Name</th><th colspan="2">Explain</th></tr>
<tr><td>Amplify</td><td colspan="2">Hold the left mouse button and drag to select the area you want to zoom in. The default zoom mode is both horizontal and vertical (area zoom). Right-click to display the menu and change the zoom mode (horizontal zoom, vertical zoom).</td></tr>
<tr><td>Reduce</td><td colspan="2">Right-click the area, then select restore to original zoom or restore to previous zoom in the display menu to zoom out.</td></tr>
<tr><td>Drag</td><td colspan="2">There are three drag methods:<br>① hold ctrl+left mouse button, the cursor turns into a hand shape, then drag the image.<br>② hold the middle mouse button (wheel) and drag the image.<br>③ when both the horizontal and vertical zoom options in the right-click menu are unselected (no zoom function available), hold down the left mouse button and drag the image.</td></tr>
<tr><td rowspan="7">Right-click menu</td><td>Save chart</td><td>Save the current interface image as a picture.</td></tr>
<tr><td>Derive data</td><td>Save image data in excel format.</td></tr>
<tr><td>Restore to original zoom</td><td>Display the entire curve.</td></tr>
<tr><td>Display node values</td><td>When the mouse hovers over a node on the curve, the axis values of that node are displayed.</td></tr>
<tr><td>Restore to last zoom level</td><td>The image is reduced to the last display ratio and display area.</td></tr>
<tr><td>Horizontal zoom</td><td>Zoom only on the x-axis.</td></tr>
<tr><td>Zoom vertically</td><td>Zoom only the y-axis (zooms a specific area only when both horizontal and vertical zoom are selected).</td></tr>
</table>

If the interface displays data for more than one minute, the data curve from the previous minute will be cleared. To view all data, click the export option in the right-click menu.

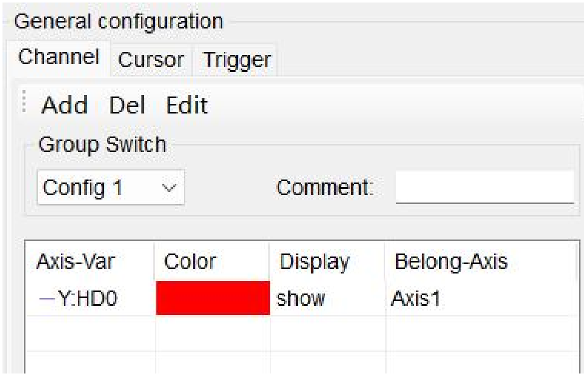
3) Oscilloscope Type Configuration Function

| Name | Explain |
|---|---|
| YT | The horizontal axis represents time, and the vertical axis represents individual register variables. When configuring the curve, only individual register variables need to be configured. |
| XY | Both the horizontal and vertical coordinates are register variables. When configuring the curve, you need to set up two register variables. |

When switching oscilloscope types, previous configuration information will not be cleared.

4) Register configuration function

Each row in the table displays an oscilloscope curve.

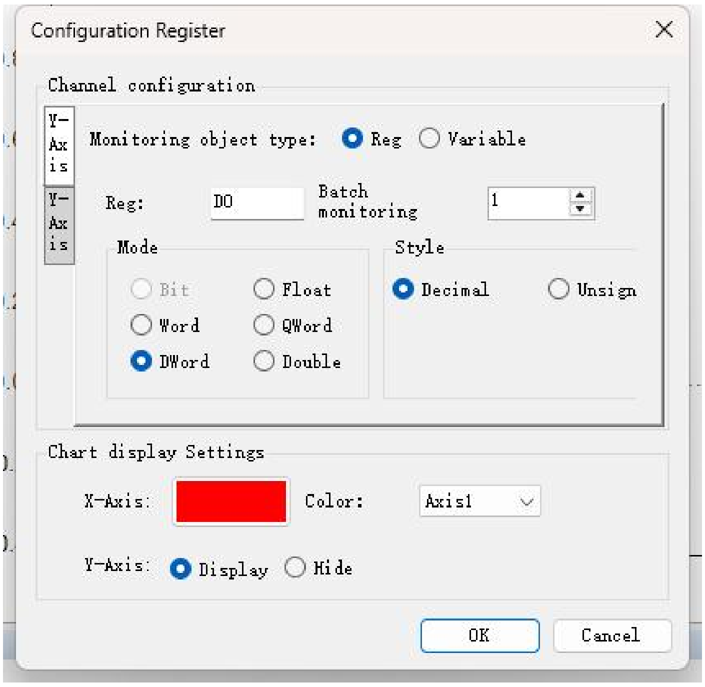



| Name | Explain |
|---|---|
| Add | Add curves (up to 15). |
| Delete | Delete the curve. |
| Edit | Edit curve properties. |

When the oscilloscope is running, you cannot add or delete curves. You can only edit curve properties.

5) Curve configuration

Add a curve to the oscilloscope display.

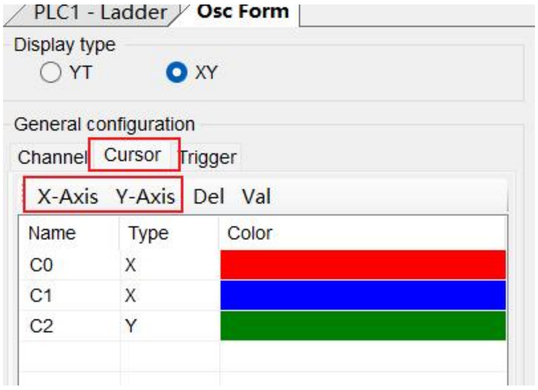



| Name | Explain |
|---|---|
| X axle | Register type (hd, d, sd, m, sm) + register offset (number) + register data type. |
| Y axle | Register type (hd, d, sd, m, sm) + register offset (number) + register data type. |
| Pigment | The color of the curve (click the color block to change the curve color). |
| Show | Whether the curve is displayed on the oscilloscope interface. |

| Name | Explain |
|---|---|
| Axle | The axis on which the curve appears on the oscilloscope display interface (for implementing the sub-axis function). |

When the oscilloscope type is YT, the [X-axis] cannot be configured, and the horizontal axis displays time. Once the oscilloscope starts working, you can only adjust the curve color, display, and axis properties, but the XY axis registers cannot be modified.

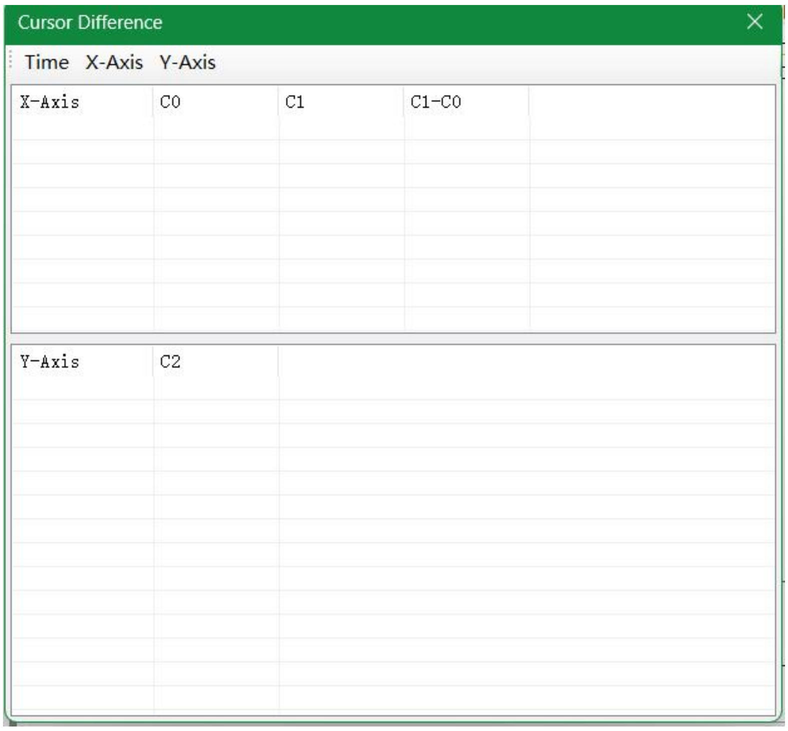
6) Cursor function




| Name | Explain |
|---|---|
| X axle | Add an x-axis cursor (vertical cursor, perpendicular to the x-axis). |
| Y axle | Add an y-axis cursor (horizontal cursor, perpendicular to the y-axis). |
| Delete | Delete the cursor. |
| D-value | Display cursor data. |

Click the table color block to change its color and cursor.

7) Difference interface



| Name | Explain |
|---|---|
| Hex | Dec to hex conversion. |
| Times | Show/hide the statustime area (available only for yt oscilloscope models). |
| X-axes | Show/hide the channel/x-axes area. |
| Y-axes | Show/hide y-axis range. |

StatusTime area display rules:

- Displays two times: computer time (PC time) and the oscilloscope's working display time.
- Time data source: The value of the X-axis cursor on the X-axis (time axis).

Channel display rules:

- Data source: Y-axis register data corresponding to the X-axis cursor (data on the Y-axis corresponding to the X-axis in the coordinate system). For example, the data from the Y-axis register variable at 1s when the X-axis cursor is at 1s on the X-axis serves as the display data source.
- Channel column: Displays all register variables monitored on the oscilloscope.

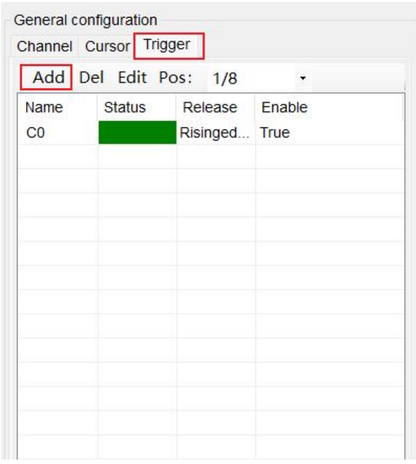
X-Axes area display rules:

- Data source: X-axis cursor data on the horizontal axis.
- Each additional X-axis adds a data point to the table.

Y-Axis range display rules:

- Data source: Data from the Y-axis cursor on the vertical axis.
- Each additional Y-axis adds a data point to the table.

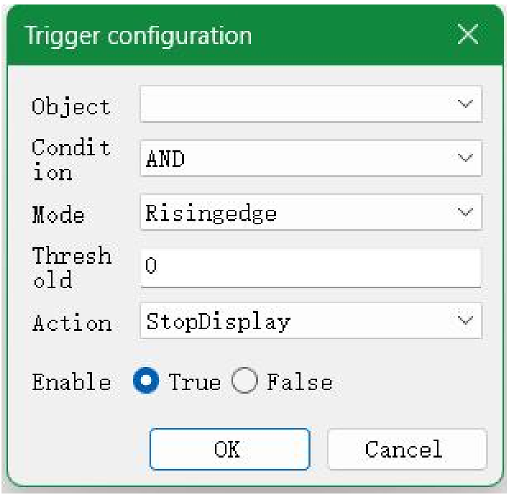
8) Trigger function




| Name | Explain |
|---|---|
| Add | Add a trigger. |
| Delete | Delete the selected trigger. |
| Edit | Edit the selected trigger. |
| Position | The position displayed on the screen when the trigger is activated. |

Trigger location description: For example, when the trigger location is 1/8, the trigger stops but not immediately. The display halts when the triggered data occupies 7/8 of the current interface. The trigger turns red and displays a dashed line at the trigger location. If the trigger version is XY, it stops immediately upon activation.

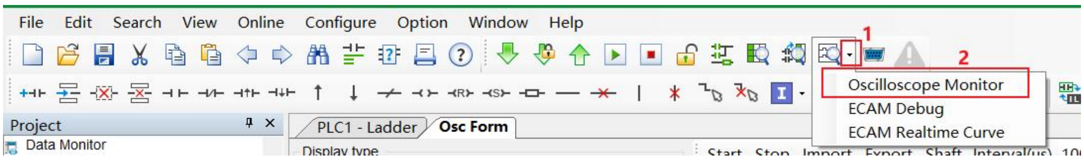
9) Trigger configuration



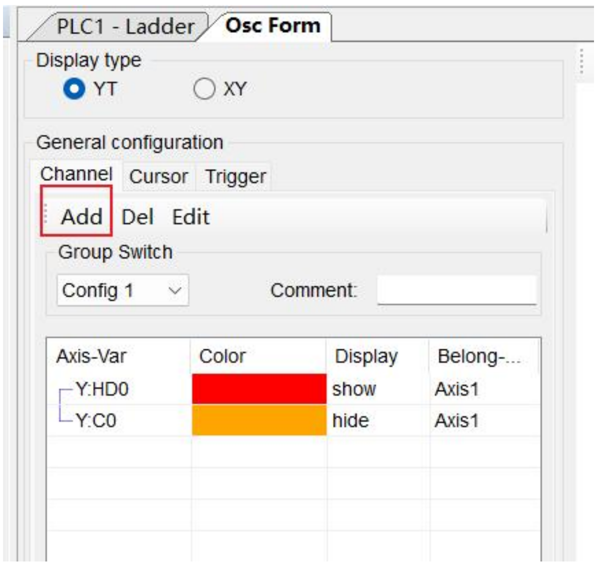

| Name | Explain |
|---|---|
| Target | The configured register variable. |
| Condition | The logical relationship between the flip-flop of the same register object. |
| Way | Trigger edge (rising edge: Rising edge. falling edge: Falling edge). |
| Threshold | Trigger threshold. |
| Action | Actions triggered by the trigger (stopdisplay: Stops display. restartdisplay: Restarts display). |
| Enable | Check if the trigger is working. |

10) Case: Register configuration for the XYZ and UVW axes of the surveillance robot's end effector

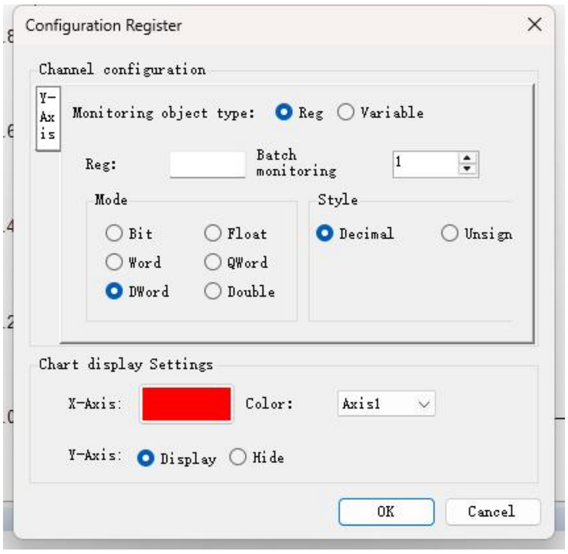
1. Open [Oscilloscope Monitor].



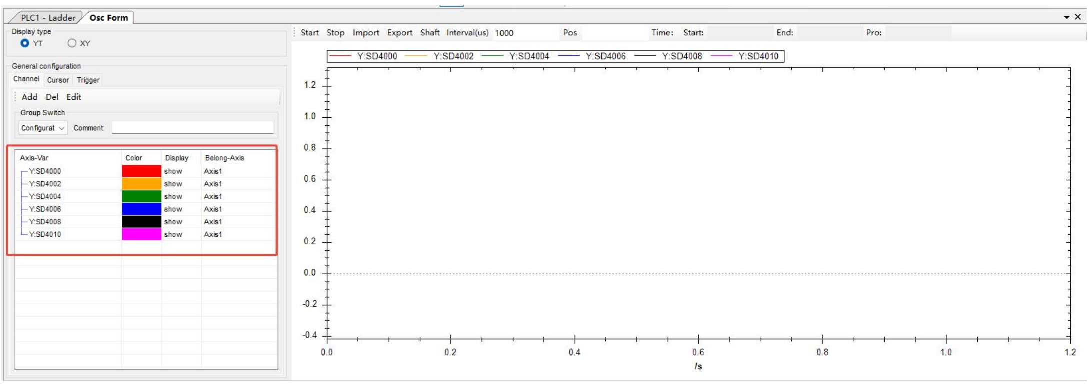

2. Select [Display Type] [YT], then click [Add].



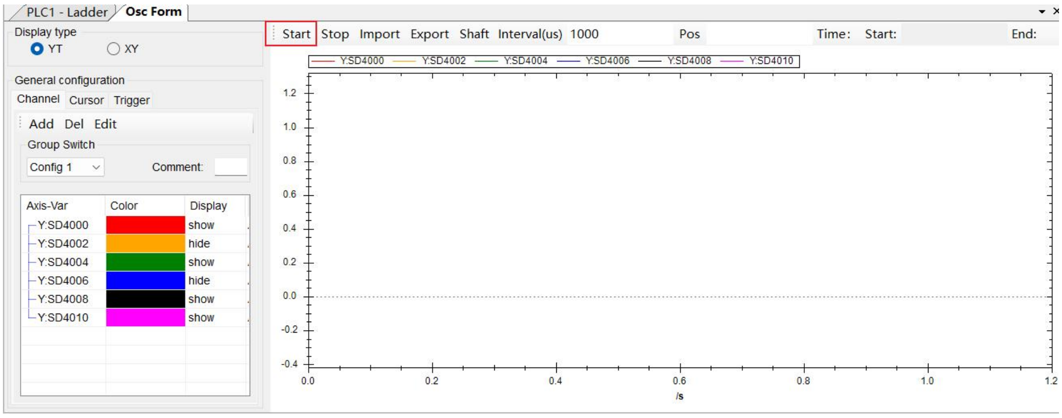

3. In the interface that appears, add the required monitoring parameters: [Register], [Batch Monitoring Count], [Monitoring Mode], and [Display Mode], then click [OK]. To monitor the robot's [X, Y, Z, U, V, W] coordinates, select the corresponding registers [SD4000, SD4002, SD4004, SD4006, SD4008, SD4010]. Enter [SD4000] as [Register], set [Batch Monitoring Count] to 6, and choose [Floating Point] as [Monitoring Mode].



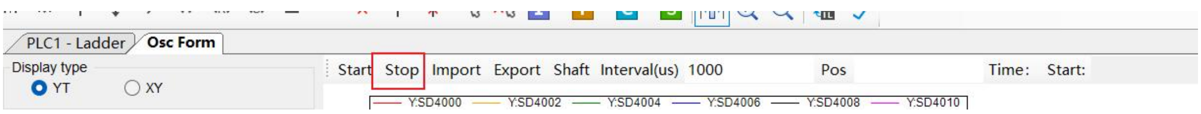

4. When the interface shown in the figure appears, the register to monitor is located in the lower left corner.



5. Click [Start], then activate the robot to monitor the register curve.




6. Click [Stop] to analyze the register curve.



# 6. Log feature

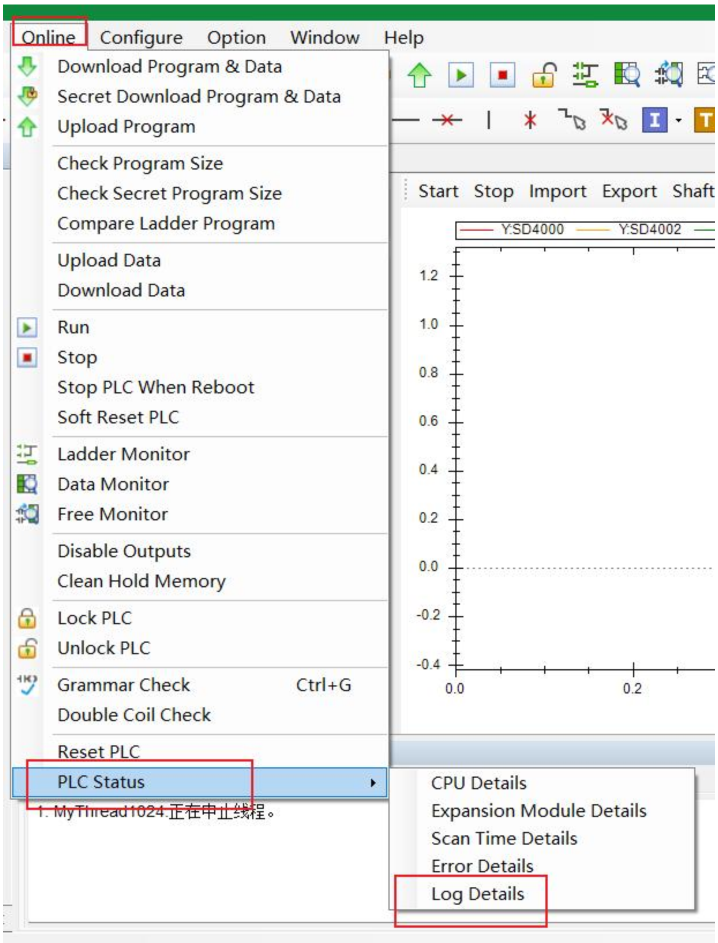
## 6.1 Overview of Functions

The log function of the robot control system is used to monitor the output of debug and error logs during the operation of the system, which is convenient for the use of the control system and troubleshooting.

## 6.2 Instructions for Use

### 6.2.1 Log Collection Interface

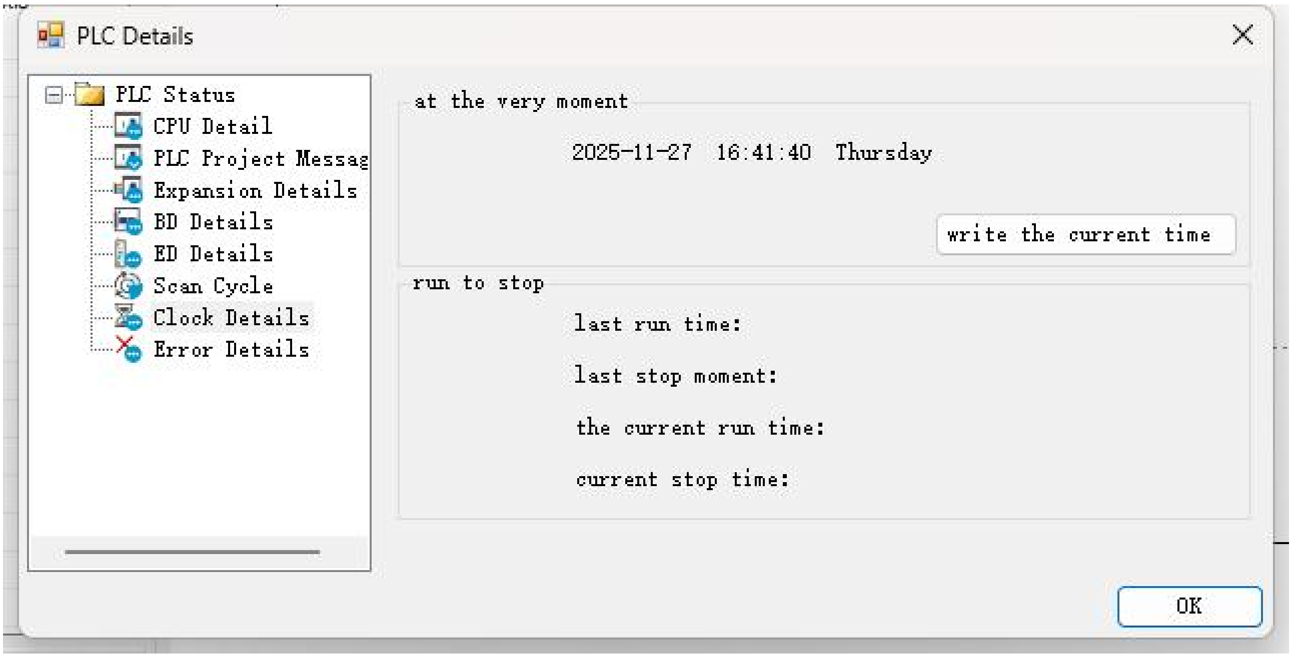
To use the logging feature, the host computer must support version 3.7.14 or later. The log interface opens as shown in the figure below. Set the log level to DEBUG and assign the value SFD803=2, which takes effect after the PLC restarts. By default, SFD803=0, and only ERROR-level logs are displayed.




| Register | Monitored value | Type | Base |
|---|---|---|---|
| SFD803 | 2 | Word | Decimal |

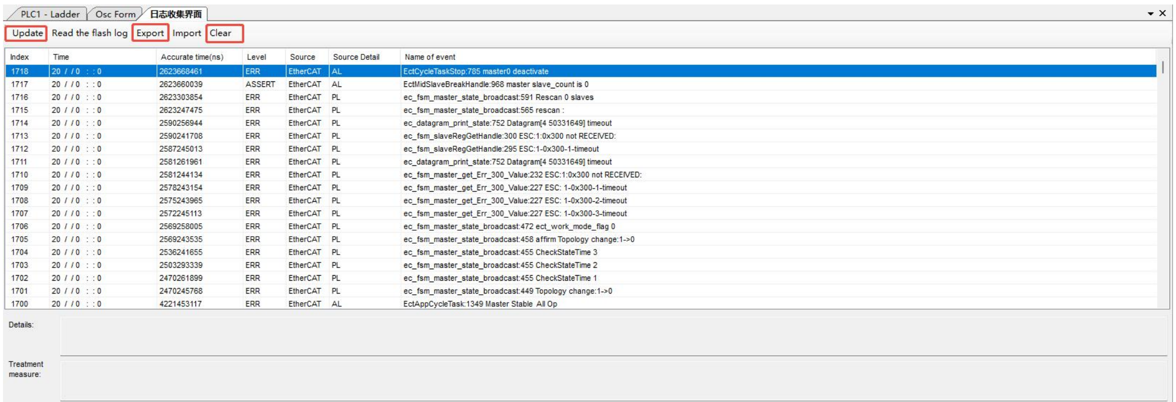
### 6.2.2 Time Configuration

To ensure accurate log timestamps, select the clock information module and click the current time field to write the current time.




### 6.2.3 Log Collection

The host machine's log interface automatically prints logs when the' Refresh 'button is clicked or the robot is successfully enabled. By default, it outputs 2,500 log entries per session, capturing approximately 2 seconds of data. Logs are sorted in descending chronological order. Each operation updates the log without overwriting existing entries in the display column. The' Clear' button removes all previously updated logs from the lower log bar.



### 6.2.4 Log closure

When the robot enters an error or warning state (SM3004 or SM3005) with ON status, logging will be suspended. Clicking 'Refresh' at this point will upload the final log record before the error occurred (to help analyze the cause after reproducing the anomaly). To resume logging, manually reset SM3004 and SM3005, then either re-enable the robot or restart the PLC. Alternatively, log operations can be performed via the SD1800-1804 register.

| Register address | Type | Definition | Explain |
|---|---|---|---|
| Sd1800 | Int16s | Log status | 1: Log off, 2: Log on, other: Exception |
| Sd1801 | Int16s | Log enabled | Set to 1 to enable. The sd1800 will automatically clear 0 to indicate success. |
| Sd1802 | Int16s | Log off | Set to 1 to enable. The sd1800 will automatically clear 0 to indicate success. |
| Sd1804 | Int16s | Clear flash log | Clear all log information from the current flash |

### 6.2.5 Log Enable

When the log function is disabled due to operational errors or warnings from the robot, manual reset of SM3004 and SM3005 is required before restarting the PLC.

### 6.2.6 Log Saving

Click the Export button to export the log information as a *.csv file.

# 7. Introduction to the Robot Model

This table contains general parameters for all robots. Adjust them based on the number of axes. Configure structural and motion parameters according to different models.

The following registers are the necessary configuration of the robot to move normally.

| Essential parameter | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| Sfd690 | INT32U | Firmware generation date | Firmware related | R | |
| Sfd803 | INT16U | Enable logging | Default: 0 | R/W | Re-enable the robot or power on the plc |
| Sfd2990 | INT16U | Control cycle | The default is 2000μs, matching the synchronization unit cycle in ethercat master station configuration. | R/W | Stop plc re-run |
| Sfd2991 | INT16U | Number of stops | Default: 20 | R/W | Stop plc re-run |
| Sfd2992 | INT16U | Retry count for communication | Default: 3 times | R/W | Stop plc re-run |
| Sfd3000+60*i (where i ranges from 0 to 12, with the maximum number of axes being 12 minus 1) | INT16U | Axis control mode | Default 0, location control mode | R/W | Stop plc re-run |
| Sfd3001+60*i (where i ranges from 0 to 12, with the maximum number of axes being 12) | INT16U | Motor type | Default 3, multi-turn absolute encoder | R/W | Stop plc re-run |
| Sfd3002+60*i (where i ranges from 0 to 12, with the maximum number of axes being 12 minus 1) | INT32U | The number of encoder lines per revolution, which is the count value fed back per full rotation of the encoder. Set this register according to the actual number of encoder lines on the motor. (for example, if the motor encoder uses a 17-bit code, set this register to 2^17, i.e., 131072) | The encoder's feedback count after one rotation defaults to 17-bit encoder pulse 131072 | R/W | Stop plc re-run |

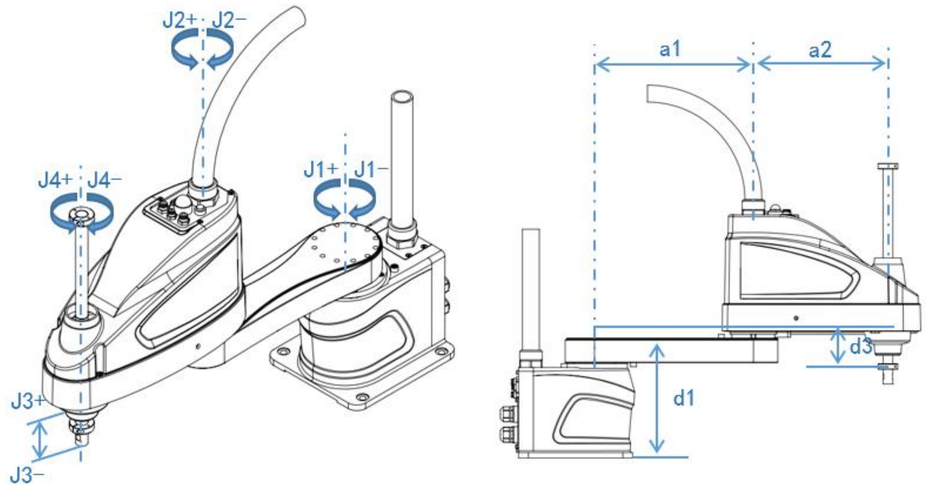
| Essential parameter | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| Sfd3004+60*i (where i ranges from 0 to 1, with the maximum number of axes being 1 less than the total number) | INT32U | The movement quantity/turn, the reference unit of motion (screw pitch), is measured in pulses. The number of pulses required for the motor to complete one full rotation is specified. The plc sends exactly the number of pulses equal to the value set in this register to rotate the motor. | The value matches sfd3002+60*i, with the default 17-bit encoder 131072 pulse | R/W | Stop plc re-run |
| Sfd3014+60*i (where i ranges from 0 to 12, with the maximum number of axes being 12) | Int32s | Minimum position of motor | Give tacit consent to -1000000000 pulses | R/W | Stop plc re-run |
| Sfd3016+60*i (where i ranges from 0 to 12, with the maximum number of axes being 12) | Int32s | Maximum position of motor | Give tacit consent to 10000000000 pulses | R/W | Stop plc re-run |
| Sfd3018+60*i (where i ranges from 0 to 1, with the maximum number of axes being 1 less than the total number) | INT32U | The motor's maximum speed is set based on its highest or rated speed. Exceeding this limit will cause the motor to operate at the maximum speed. | The system typically operates at either the maximum or rated speed, with a default 17-bit encoder and motor speed of 3000 r/min. 6553600 pulses/s | R/W | Stop plc re-run |
| Sfd5112 | Float | Position x completes the signal detection width | Mm | R/W | Stop plc re-run |
| Sfd5114 | Float | Position y completes signal detection width | Mm | R/W | Stop plc re-run |
| Sfd5116 | Float | Position z completes the signal detection width. | Mm | R/W | Stop plc re-run |
| Sfd5118 | Float | Width of attitude completion signal detection | Linear measure | R/W | Stop plc re-run |
| Sfd6500 | INT16U | Robot main version number | Firmware related | R | |
| Sfd6501 | INT16U | Robot subversion number | Firmware related | R | |
| Sfd6502 | INT16U | Robot function version number | Firmware related | R | |
| Sfd6503 | INT16U | Temporary robot version number | Firmware related | R | |
| Hsd104+20*i (i=0,1,2... Maximum number of axes minus 1) | Int64s | Set the zero position for the current encoder feedback value | Relative absolute zero encoder count | R/W | Stop plc re-run |

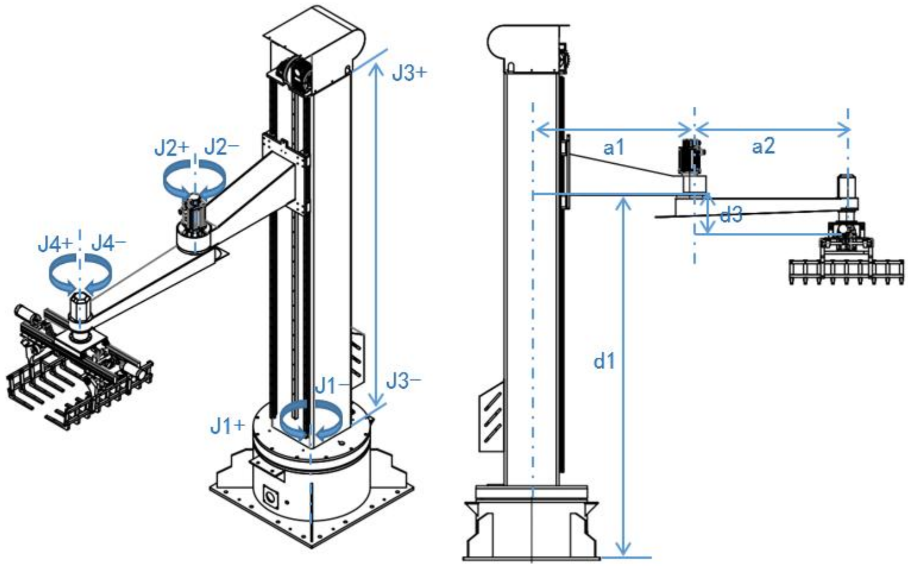
Zero Point Calibration Method: When the robot loses its zero position, calibration can be performed by writing values to the HSD104+20i register. First, position the robot to the zero position shown in the figure (where the robot is at the zero position). If the joint angle corresponding to the zero position is 0, write 0 to the corresponding HSD register. If there is an initial angle value, calculate the offset as: SFD3002+2i (based on the number of pulses per full revolution) * SFD5020+2i (reduction ratio) * zero position angle / 360°.

The calculation formula for the moving joint is: SFD3002+2i (based on the number of single-cycle pulses) × zero-distance / SFD5020+2i (reduction ratio).

For example, consider a six-axis robot with a 121:1 reduction ratio on axis 2, 131,072 pulses per revolution, and a zero position angle of-90. The zero position pulse count is calculated as 131,072 × 121-90 / 360= -39,649,28. Since all other axes have zero position angles of 0, the configuration data (0, -39,649,28,0, 0, 0, 0) should be written to the HSD104+20i.

## 7.1 SCARA robot




| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| Sfd3047+60*i (where i ranges from 0 to 1, with the maximum number of axes being 1 less than the total number) | INT16U | Motor direction logic<br>The configuration must align with the joint's forward and reverse rotation to ensure its movement matches the predefined model. The motor's final rotation direction is determined by the joint's rotation and the axis's forward rotation direction in the servo parameters. ⑤ | 0: Positive logic<br>1: Anti-logic | R/W | Stop plc re-run |
| **Sfd5000** | **INT16U** | **Robot type** | **2** | **R/W** | **Stop plc re-run** |
| Sfd5002 | Float | Connecting rod parameter a1① | mm | R/W | Stop plc re-run |
| Sfd5004 | Float | Connecting rod parameter a2② | mm | R/W | Stop plc re-run |
| Sfd5006 | Float | Connecting rod parameter d3③ | mm, default to 0 | R/W | Stop plc re-run |

| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| Sfd5008 | Float | Connecting rod parameter d1④ | mm, default to 0 | R/W | Stop plc re-run |
| Sfd5012 | Float | Scara – screw coupling ratio<br>How many millimeters (mm) does joint 3 move when joint 4 rotates one full turn? ⑧ | mm/ shut in a pen | R/W | Stop plc re-run |
| Sfd5020+2*i (where i=0,1,2... To the maximum number of axes minus 1) | Float | Gear ratio of each joint<br>The actual deceleration ratio of the joint is consistent with ⑦ | Rotate joint: Circle/circle<br>Moving joint: Mm/turn | R/W | Stop plc re-run |

①: First arm length: the vertical distance a1 from the center of the first joint to the center of the second joint.

②: Second arm length: the vertical distance a2 from the center of the second joint to the center of the third joint.

③: Screw height: The vertical distance d3 from the flange end to the origin of the second joint coordinate system, which may be positive or negative.

④: Base height: The vertical distance d1 from the origin of the first joint coordinate system to the origin of the base coordinate system, which may be positive or negative.

⑤: Motor motion direction logic: Based on the joint settings in the diagram, configure the rotation axis direction and define the positive/negative motion direction for each axis. Only two settings (0 or 1) are allowed to ensure that the joints' motion directions comply with the right-hand rule. Incorrect settings will result in erroneous Cartesian motion trajectories.

⑥: Robot Zero Position: The zero position configuration is shown in the figure above, with the input joint angles set to (0,0,0,0).

⑦: Motor speed calculation formula for joint angular velocity: For rotating joints, (pulse/s) / (pulse/revolution) / (reduction ratio) * (degree/revolution) = (degree/s). For moving joints: (pulse/s) / (pulse/revolution) * (reduction ratio) = (mm/s). where (pulse/s) = (rev/min) * (pulse/revolution) / 60.

⑧: In the third axis of SCARA robots, various transmission methods exist. The reduction ratio ultimately depends on the motor rotation of the three axes driving the movement of the three-axis lead screw. For SCARA robots with ball screw mechanisms, the actual reduction ratio is determined by the motor-to-nut transmission ratio and the lead screw pitch. In robots with non-ball screw structures, such as those using gear-rack transmission, the actual reduction ratio is determined by the gear-to-rack ratio.

⑨: Screw coupling ratio example: When the coupling ratio is set to 0, a full rotation of joint 4 causes joint 3 to move 20mm in the positive direction, resulting in a coupling ratio of 20/1. Conversely, a 20mm negative movement of joint 3 corresponds to a coupling ratio of-20/1. Similarly, a 0.5mm positive movement of joint 3 sets the coupling ratio to 0.5/1.

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
| Sfd5040+2*i (where i=0,1,2... To the maximum number of axes minus 1) | Float | The maximum speed of each joint (joint movement) is calculated by subtracting the motor-side sfd3018+60*i from the preset maximum motor speed to obtain the joint output-side speed. ① | M/s or mm/s | R/W | Stop plc re-run |
| Sfd5060+2*i (where i=0,1,2... To the maximum number of axes minus 1) | Float | Maximum acceleration of each joint<br>The default setting is 4-10 times joint speed | $M/s^2$ or $mm/s^2$ | R/W | Stop plc re-run |
| Sfd5250+2*i (where i=0,1,2... To the maximum number of axes minus 1) | Float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $M/s^3$ or $mm/s^3$ | R/W | Stop plc re-run |
| Sfd5080 | Float | Maximum speed at the end, [linear velocity] ② | Mm/s or degrees/s | R/W | Stop plc re-run |
| Sfd5090 | Float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear speed | $Mm/s^2$ or $degrees/s^2$ | R/W | Stop plc re-run |
| Sfd5270 | Float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear acceleration, with specific values depending on the actual operating conditions of the device. | $M/s^3$ or $mm/s^3$ | R/W | Stop plc re-run |
| Sfd5140+2*i (where i=0,1,2... To the maximum number of axes minus 1) | Float | Maximum angular limits for each joint<br>The set value should be smaller than the actual maximum mechanical limit angle of each joint. | Degree, axis 3 in mm | R/W | Stop plc re-run |
| Sfd5160+2*i (where i=0,1,2... To the maximum number of axes minus 1) | Float | Minimum angular limit of each joint<br>Set the value to be greater than the actual mechanical minimum limit angle of each joint. | Degree, axis 3 in mm | R/W | Stop plc re-run |
| Sfd5230+2*i (where i=0,1,2... To the maximum number of axes minus 1) | Float | The maximum speed of each joint (cartesian motion) is typically set to match the sfd5040+2i configuration. | Linear measure /s | R/W | Stop plc re-run |

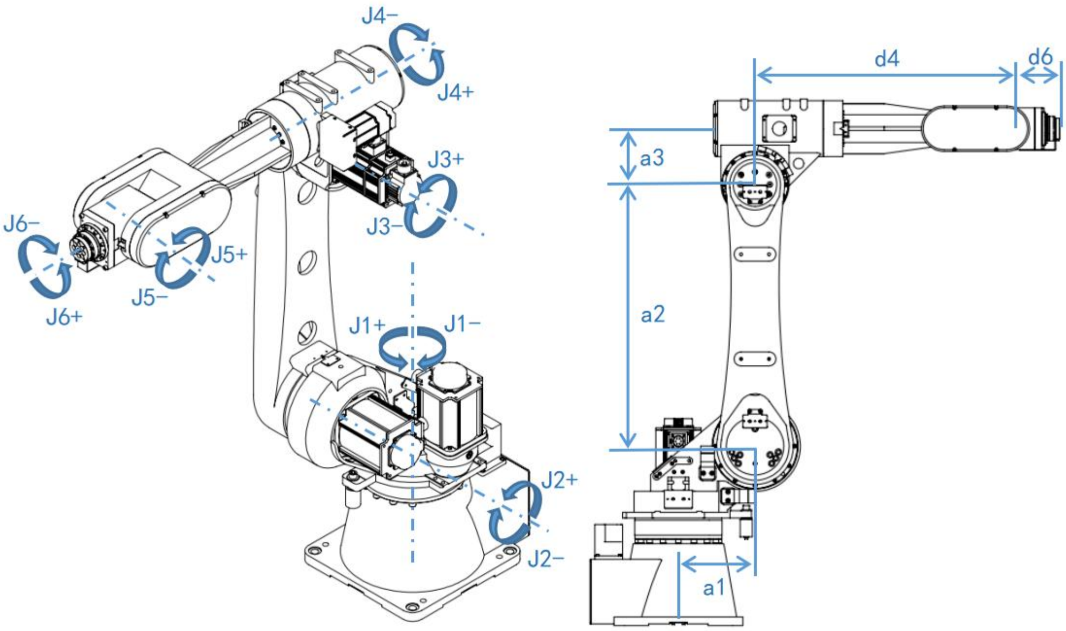
| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
| Sfd5370+i (where i ranges from 0 to 1,2,..., the maximum number of axes minus 1) | INT16U | Whether the axis is controlled. All robot axes must be set to 1. | 0: Uncontrolled Controlled | R/W | Stop plc re-run |
| Sfd6030 | Float | Maximum speed of quaternion terminal attitude | Default: 100 degrees/s | R/W | Stop plc re-run |
| Sfd6032 | Float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 degrees/s² | R/W | Stop plc re-run |
| Sfd6034 | Float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s³ | R/W | Stop plc re-run |

①: When configuring SFD5040, it must be compared with SFD3018. Since their units differ, the values must first be converted to the same unit. The corresponding pulse count is calculated as [SFD5040+2*i]*[SFD5020+2*i]*[SFD3002+60*i]/360. This value must be less than the corresponding axis's SFD3018+60i. Otherwise, the converted value of SFD3018+60i will be used as the maximum operating speed per axis of the robot.

②: The maximum terminal velocity can be derived from the joint's maximum velocity. For this model, the peak terminal velocity occurs when rod 1 and rod 2 are fully extended. The horizontal linear velocity is determined by the first-axis angular velocity multiplied by the rod length, while the vertical linear velocity is directly determined by the third-axis linear velocity. The resultant velocity represents the final maximum terminal linear velocity, with the calculation formula as follows:

$$V_{线\max} = \sqrt{[V_{J1\max}(L_1 + L_2) + V_{J2\max}L_2]^2 + V_{J3\max}{}^2}$$

# 7.2 Series six-axis robot




| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD3047+60*i (where i ranges from 0 to 1, with the maximum number of axes being 1 less than the total number) | INT16U | Motor direction logic The configuration must align with the joint's forward and reverse rotation to ensure its movement matches the predefined model. The motor's final rotation direction is determined by the joint's rotation and the axis's forward rotation direction in the servo parameters⑥ | 0: Positive logic 1: Anti-Logic | R/W | Stop PLC re-run |
| **SFD5000** | **INT16U** | **Robot type** | **6** | **R/W** | **Stop PLC re-run** |
| SFD5002 | float | Six-axis-d1 | mm, default is 0 | R/W | Stop PLC re-run |
| SFD5004 | float | Six-axis-a1① | mm | R/W | Stop PLC re-run |
| SFD5006 | float | Six-axis-a2② | mm | R/W | Stop PLC re-run |
| SFD5008 | float | Six-axis-A3③ | mm | R/W | Stop PLC re-run |
| SFD5010 | float | Six-axis-d4④ | mm | R/W | Stop PLC re-run |
| SFD5012 | float | Six-axis-d6⑤ | mm/ shut in a pen | R/W | Stop PLC re-run |

| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5014 | float | five axis six axis coupling ratio<br>The positive coupling ratio is 1 when the joint 5 is rotated forward by one degree, and the negative coupling ratio is 1 when the joint 6 is rotated forward by one degree. | | R/W | Stop PLC re-run |
| SFD5016 | float | six six axis coupling ratio<br>The coupling ratio is positive when the joint 4 is rotated one degree forward, and negative when the joint 6 is rotated one degree backward. | | R/W | Stop PLC re-run |
| SFD5018 | float | four five axis coupling ratio<br>The coupling ratio is positive when the joint 4 is rotated forward by one degree, and negative when the joint 5 is rotated forward by one degree. | | R/W | Stop PLC re-run |
| SFD5020+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | gear ratio of each joint<br>Consistent with the actual deceleration ratio of the joint | Rotate joint: Circle/Circle<br>Moving joint: mm/turn | R/W | Stop PLC re-run |

①: Connecting rod parameter 1: The vertical distance a1 between the axis centers of the first and second joints, as shown in the figure above.

②: Connecting rod parameter 2: The vertical distance a2 between the axis center of the second joint and the axis center of the third joint, which is parameter 2 in the diagram above.

③: Connecting rod parameter 3: The vertical distance a3 between the axis center of the third joint and the axis center of the fourth joint, which corresponds to parameter 3 in the diagram above.

④: Connecting rod parameter 4: The vertical distance d4 between the axis center of the 4th joint and that of the 5th joint, as shown in the figure above.

⑤: Connecting rod parameter 5: The vertical distance d6 between the axis center of the 5th joint and that of the 6th joint, which corresponds to parameter 5 in the diagram above.

⑥: Motor motion direction logic: Based on the joint settings in the diagram, configure the rotation axis direction and define the positive/negative motion direction for each axis. Only two settings (0 or 1) are allowed to ensure that the joints' motion directions comply with the right-hand rule. Incorrect settings will result in erroneous Cartesian motion trajectories.

⑦: Robot zero position: The zero position pose is shown in the figure above, with the input joint angles set as (0, -90,0, 0, 0, 0).

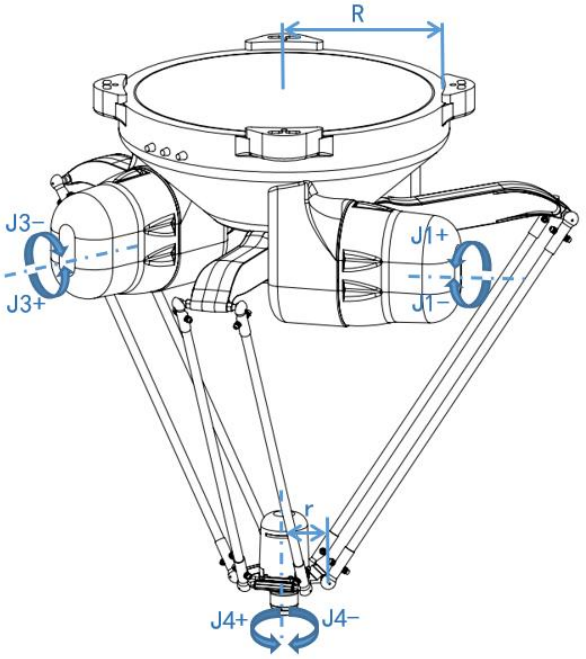
⑧: Example of joint 56 coupling ratio settings: When the coupling ratio is set to 0, joint 5 rotates forward by 1 degree. If joint 6 rotates forward by 2 degrees, the coupling ratio is set to 2/1. if joint 6 rotates backward by 2 degrees, the coupling ratio is set to-2/1.

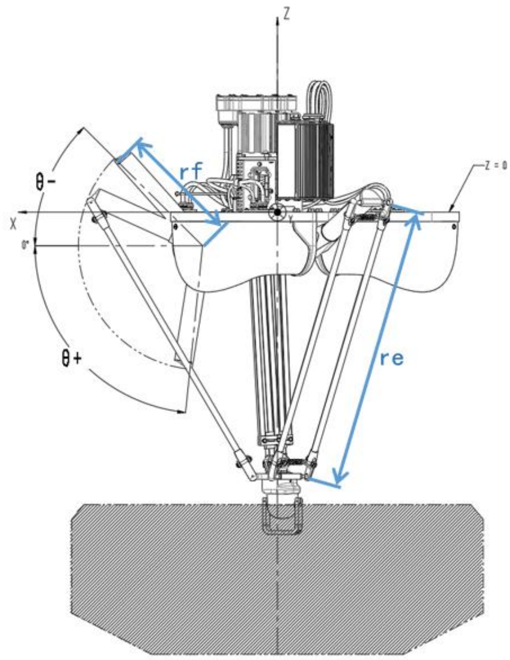
| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5040+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (joint motion) is calculated by setting the motor-side SFD3018+60*i to the maximum motor speed, which is then applied to the joint output side. | m/s or mm/s | R/W | Stop PLC re-run |
| SFD5060+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint The default setting is 5-10 times the joint speed | $m/s^2$ or $mm/s^2$ | R/W | Stop PLC re-run |
| SFD5250+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5080 | float | Maximum speed at the end, [linear velocity] | mm/s or degrees/s | R/W | Stop PLC re-run |
| SFD5090 | float | Maximum acceleration at the end, [linear acceleration] The default setting is 5-10 times the linear speed | $mm/s^2$ or $degrees/s^2$ | R/W | Stop PLC re-run |
| SFD5270 | float | Maximum acceleration at the end, [linear acceleration] The default setting is 5-10 times the linear acceleration, and the specific parameter values depend on the actual operating conditions of the device. | $mm/s^3$ or $degrees/s^3$ | R/W | Stop PLC re-run |

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5140+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum angular limits for each joint<br>The set value should be smaller than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5160+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | minimum angular limit of each joint<br>Set the value to be greater than the actual mechanical minimum limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5230+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (Cartesian motion) is typically set to match the SFD5040+2i configuration. | linear measure /s | R/W | Stop PLC re-run |
| SFD5370+i (where i ranges from 0 to 1,2,..., the maximum number of axes minus 1) | INT16U | Whether the axis is controlled. All robot axes must be set to 1. | 0: Uncontrolled<br>Controlled | R/W | Stop PLC re-run |
| SFD6000+2*i (i=0,1,2...10) | float | Kinematic model calibration compensation values | A total of 11 floats are compensated, with specific parameters as specified in the calibration manual. | R/W | Stop PLC re-run |
| SFD6030 | float | Maximum speed of quaternion terminal attitude | Default: 100 degrees per second | R/W | Stop PLC re-run |
| SFD6032 | float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 degrees/s² | R/W | Stop PLC re-run |
| SFD6034 | float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop PLC re-run |

①: When configuring SFD5040, it must be compared with SFD3018. Since their units differ, the values must first be converted to the same unit. The corresponding pulse count is calculated as [SFD5040+2*i]*[SFD5020+2*i]*[SFD3002+60*i]/360. This value must be less than the corresponding axis's SFD3018+60i. Otherwise, the converted value of SFD3018+60i will be used as the maximum operating speed per axis of the robot.

# 7.3 Parallel DELTA Robot




| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD3047+60*i (where i ranges from 0 to 1,2,..., down to the maximum number of axes minus 1) | INT16U | motor direction logic Configure the joint rotation in both forward and reverse directions to ensure it matches the specified model. | 0: Positive logic 1: Anti-Logic | R/W | Stop PLC re-run |
| **SFD5000** | **INT16U** | **Robot type** | **1** | **R/W** | **Stop PLC re-run** |
| SFD5002 | float | active rod length rf | mm | R/W | Stop PLC re-run |
| SFD5004 | float | The length of the follower rod is re. | mm | R/W | Stop PLC re-run |
| SFD5006 | float | The radius of the stationary platform is R. | mm | R/W | Stop PLC re-run |
| SFD5008 | float | The radius of the moving platform is r. | mm | R/W | Stop PLC re-run |
| SFD5020+2*i (where i ranges from 0 to 1, with the maximum number of axes being 1 less than the total number) | float | gear ratio of each joint Consistent with the actual deceleration ratio of the joint | Rotate joint: Circle/Circle Moving joint: mm/turn | R/W | Stop PLC re-run |

①: Active rod: The upper arm section directly connected to the motor shaft, rotating with the motor as shown in the figure above.

②: The follower rod is a closed-loop parallelogram formed by four ball joints and a rod, as shown in the figure above.

③: Static Platform: A virtual platform formed by the axes of three motors on the same plane (where the active rod connects to the motors), as shown in the simplified structure. No actual mechanical components exist.

④: Mobile platform: A small platform at the bottom that moves with the robot's linkage.

⑤: Motor motion direction logic: Based on the joint settings in the diagram, configure the rotation axis direction and define the positive/negative motion direction for each axis. Only two settings (0 or 1) are allowed to ensure that the joints' motion directions comply with the right-hand rule. Incorrect settings will result in erroneous Cartesian motion trajectories.

⑥: Robot Zero Position: The zero position configuration is shown in the figure above, with the input joint angles set to (0,0,0,0).

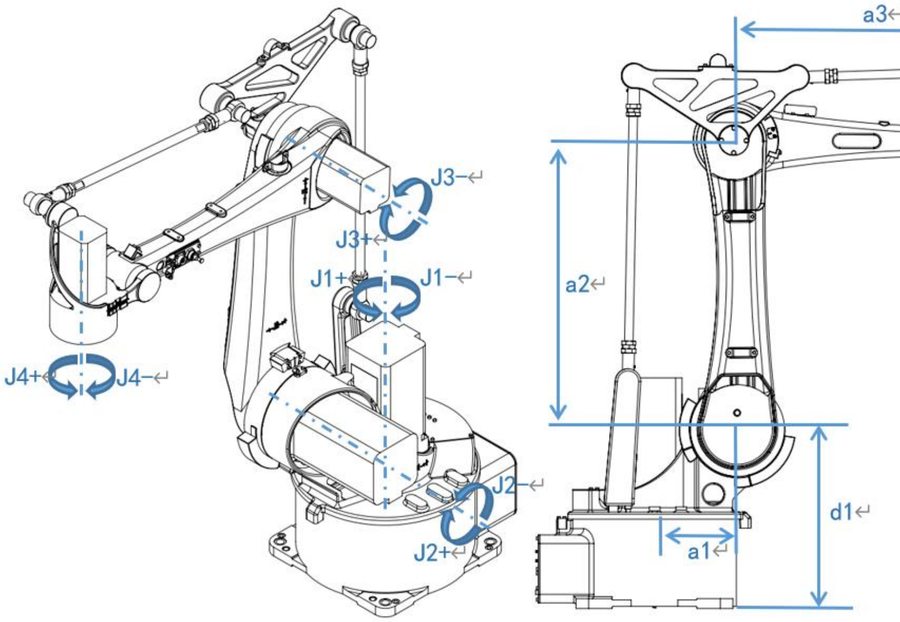
| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
| SFD5040+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (joint motion) is calculated by setting the motor-side SFD3018+60*i to the maximum motor speed, which is then applied to the joint output side ①. | m/s or mm/s | R/W | Stop PLC re-run |
| SFD5060+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint speed | $m/s^2$ or $mm/s^2$ | R/W | Stop PLC re-run |
| SFD5250+2*i (where i ranges from 0 to 1, with the maximum number of axes being 1 less than the total number) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5080 | float | Maximum speed at the end, [linear velocity] | mm/s or degrees/s | R/W | Stop PLC re-run |
| SFD5090 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear speed | $mm/s^2$ or $degrees/s^2$ | R/W | Stop PLC re-run |
| SFD5270 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear acceleration, and the specific parameter values depend on the actual operating conditions of the device. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
| SFD5140 | float | maximal angle restriction of joint 1<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5142 | float | maximal angle limitation of joint 2<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5144 | float | maximal angle restriction of joint 3<br>The set value should be smaller than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5146 | float | Maximum rotation angle of Rz | linear measure | R/W | Stop PLC re-run |
| SFD5148 | float | Maximum X-axis movement | mm | R/W | Stop PLC re-run |
| SFD5150 | float | Maximum movement radius | mm | R/W | Stop PLC re-run |
| SFD5160 | float | minimum angular restriction of joint 1<br>The set value should exceed the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5162 | float | minimum angular restriction of joint 2<br>The set value should exceed the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5164 | float | minimum angular restriction of joint 3<br>The set value should exceed the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5166 | float | Rz minimum rotation angle | linear measure | R/W | Stop PLC re-run |
| SFD5168 | float | Minimum position of the Z-axis | mm | R/W | Stop PLC re-run |
| SFD5170 | float | Minimum movement radius | mm | R/W | Stop PLC re-run |
| SFD5230+2*i (where i ranges from 0 to 1, 2,..., the maximum number of axes minus 1) | float | The maximum speed of each joint (Cartesian motion) is typically set to match the SFD5040+2i configuration. | linear measure /s | R/W | Stop PLC re-run |
| SFD5370+i (where i ranges from 0 to 1,2,..., the maximum number of axes minus 1) | INT16U | Whether the axis is controlled. All robot axes must be set to 1. | 0: Uncontrolled Controlled | R/W | Stop PLC re-run |
| SFD6030 | float | Maximum speed of quaternion terminal attitude | Default: 100 degrees per second | R/W | Stop PLC re-run |
| SFD6032 | float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 degrees/$s^2$ | R/W | Stop PLC re-run |

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
| SFD6034 | float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop PLC re-run |

①: When configuring SFD5040, it must be compared with SFD3018. Since their units differ, the values must first be converted to the same unit. The corresponding pulse count is calculated as [SFD5040+2*i]*[SFD5020+2*i]*[SFD3002+60*i]/360. This value must be less than the corresponding axis's SFD3018+60i. Otherwise, the converted value of SFD3018+60i will be used as the maximum operating speed per axis of the robot.

## 7.4 Four-joint palletizing robot




| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD3047+60*i (where i ranges from 0 to 1,2,..., down to the maximum number of axes minus 1) | INT16U | motor direction logic<br>The configuration must be performed in both forward and reverse joint rotations to ensure the joint's bidirectional motion aligns with the predefined model ⑦ | 0: Positive logic<br>1: Anti-Logic | R/W | Stop PLC re-run |
| **SFD5000** | **INT16U** | **Robot type** | **4** | **R/W** | **Stop PLC re-run** |
| SFD5002 | float | d1① | mm | R/W | Stop PLC re-run |
| SFD5004 | float | a1② | mm | R/W | Stop PLC re-run |
| SFD5006 | float | a2③ | mm | R/W | Stop PLC re-run |
| SFD5008 | float | a3④ | mm | R/W | Stop PLC re-run |
| SFD5010 | float | a4⑤ | mm | R/W | Stop PLC re-run |
| SFD5012 | float | d4⑥ | mm | R/W | Stop PLC re-run |
| SFD5014 | float | 2. The coupling ratio of the 3rd joint is ⑨ | mm | R/W | Stop PLC re-run |
| SFD5020+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | gear ratio of each joint Consistent with the actual deceleration ratio of the joint | Rotate joint: Circle/Circle<br>Moving joint: mm/turn | R/W | Stop PLC re-run |

①: Connecting rod parameter 1: The distance from the axis center of the first joint to the ground, as shown in Figure d1 above.

②: Connecting rod parameter 2: The horizontal distance between the centers of the first and second joint axes, as shown in Figure a1 above.

③: Connecting rod parameter 3: The vertical distance between the centers of the second and third joint axes, as shown in Figure a2 above.

④: Connecting rod parameter 4: The horizontal distance from the axis of the third joint to the wrist, as shown in Figure a3 above.

⑤: Connecting rod parameter 5: The horizontal distance from the wrist to the axis center of the fourth joint, as shown in Figure a4 above.

⑥: Connecting rod parameter 6: The vertical distance between the centers of the third and fourth joint axes, as shown in Figure d4 above.

⑦: Motor motion direction logic: Based on the joint settings in the diagram above, configure the rotation axis direction and define the positive/negative motion direction for each axis. Only two settings (0 or 1) are allowed to ensure that the joints' motion directions comply with the right-hand rule. Incorrect settings will result in erroneous Cartesian motion trajectories.

⑧: Robot zero position: The zero position pose is shown in the figure above, with the input joint angles set to (0,0,0,0).

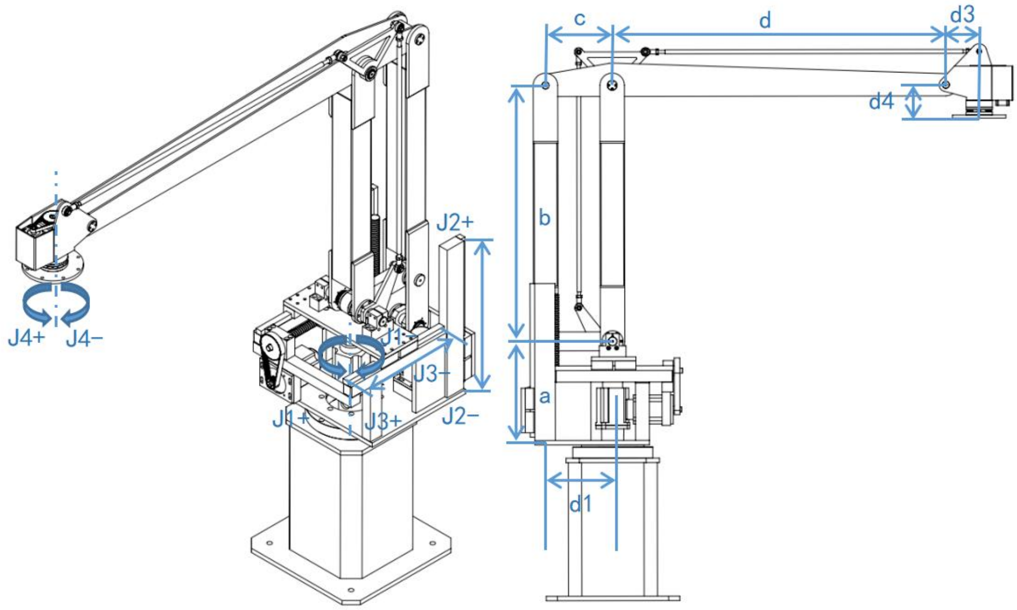
⑨: Example of joint 23 coupling ratio settings: When the coupling ratio is set to 0, joint 2 rotates forward by 1 degree. If joint 3 rotates forward by 2 degrees, the coupling ratio is set to 2/1. if joint 3 rotates backward by 2 degrees, the coupling ratio is set to-2/1.

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5040+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (joint motion) is calculated by setting the motor-side SFD3018+60*i to the maximum motor speed, which is then applied to the joint output side ①. | m/s or mm/s | R/W | Stop PLC re-run |
| SFD5060+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times joint speed | $m/s^2$ or $mm/s^2$ | R/W | Stop PLC re-run |
| SFD5250+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5080 | float | Maximum end speed, [linear velocity] Linear velocity determined by joints 1, 2, and 3 | mm/s or degrees/s | R/W | Stop PLC re-run |
| SFD5090 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear speed | $mm/s^2$ or degrees/$s^2$ | R/W | Stop PLC re-run |

| | | | | | |
|---|---|---|---|---|---|
| SFD5270 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear acceleration, and the specific parameter values depend on the actual operating conditions of the device. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5140+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum angular limits for each joint<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5160+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | minimum angular limit of each joint<br>Set the value to be greater than the actual mechanical minimum limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5230+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (Cartesian motion) is typically set to match the SFD5040+2i configuration. | linear measure /s | R/W | Stop PLC re-run |
| SFD5370+i (where i ranges from 0 to 1,2,..., the maximum number of axes minus 1) | INT16U | Whether the axis is controlled. All robot axes must be set to 1. | 0: Uncontrolled Controlled | R/W | Stop PLC re-run |
| SFD6030 | float | Maximum speed of quaternion terminal attitude | Default: 100 degrees/s | R/W | Stop PLC re-run |
| SFD6032 | float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 $degrees/s^2$ | R/W | Stop PLC re-run |
| SFD6034 | float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop PLC re-run |

①: When configuring SFD5040, it must be compared with SFD3018. Since their units differ, the values must first be converted to the same unit. The corresponding pulse count is calculated as [SFD5040+2*i]*[SFD5020+2*i]*[SFD3002+60*i]/360. This value must be less than the corresponding axis's SFD3018+60i. Otherwise, the converted value of SFD3018+60i will be used as the maximum operating speed per axis of the robot.

# 7.5 Hybrid Palletizing Robot




| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
| SFD3047+60*i (where i ranges from 0 to 1,2,..., down to the maximum number of axes minus 1) | INT16U | motor direction logic The configuration must be performed in both forward and reverse joint rotations to ensure the joint's bidirectional motion aligns with the predefined model ⑦ | 0: Positive logic 1: Anti-Logic | R/W | Stop PLC re-run |
| **SFD5000** | **INT16U** | **Robot type** | **10** | **R/W** | **Stop PLC re-run** |
| SFD5002 | float | a① | mm | R/W | Stop PLC re-run |
| SFD5004 | float | b② | mm | R/W | Stop PLC re-run |
| SFD5006 | float | c③ | mm | R/W | Stop PLC re-run |
| SFD5008 | float | d④ | mm | R/W | Stop PLC re-run |
| SFD5010 | float | d3⑤ | mm | R/W | Stop PLC re-run |
| SFD5012 | float | d4⑥ | mm | R/W | Stop PLC re-run |
| SFD5014 | float | d1⑦ | mm | R/W | Stop PLC re-run |
| SFD5020+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | gear ratio of each joint Consistent with the actual deceleration ratio of the joint | Rotate joint: Circle/Circle Moving joint: mm/turn | R/W | Stop PLC re-run |

①: Connecting rod parameter 1: The vertical distance a from the center of the sliding block in the zero position to the hinge.

②: Connecting rod parameter 2: Length of the front arm (b).

③: Connecting rod parameter 3: The perpendicular distance c between the front and rear arms.

④: Connecting rod parameter 4: forearm length d.

⑤: Connecting rod parameter 5: The vertical distance d3 from the forearm tip to the axis center of the wrist.

⑥: Connecting rod parameter 6: The vertical distance d4 from the motor end to the material bottom.

⑦: Connecting rod parameter 7: The distance between the perpendicular lines of shaft 1 and shaft 2, denoted as rod length d1.

⑧: The connecting rod parameters of this model satisfy a*d=b*c, with both b and d being non-zero, which verifies the correctness of the connecting rod parameters.

⑨: Motor motion direction logic: Based on the joint settings in the diagram above, configure the rotation axis direction and define the positive/negative motion direction for each axis. Only two settings (0 or 1) are allowed to ensure that the joints' motion directions comply with the right-hand rule. Incorrect settings will result in erroneous Cartesian motion trajectories.

⑩: Robot Zero Position: The zero position configuration is shown in the figure above, with the input joint angles set to (0,0,0,0).

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5040+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (joint motion) is calculated by setting the motor-side SFD3018+60*i to the maximum motor speed, which is then applied to the joint output side ①. | m/s or mm/s | R/W | Stop PLC re-run |
| SFD5060+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times joint speed | $m/s^2$ or $mm/s^2$ | R/W | Stop PLC re-run |
| SFD5250+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5080 | float | Maximum speed at the end, [linear velocity] ②<br>The linear velocity is determined by joint 1,2,3 | mm/s or degrees/s | R/W | Stop PLC re-run |
| SFD5090 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear speed | $mm/s^2$ or $degrees/s^2$ | R/W | Stop PLC re-run |

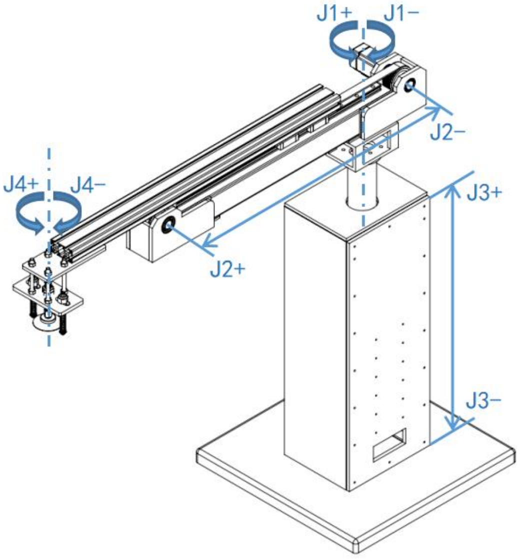
| | | | | | |
|---|---|---|---|---|---|
| SFD5270 | float | Maximum acceleration at the end, [linear acceleration] The default setting is 5-10 times the linear acceleration, and the specific parameter values depend on the actual operating conditions of the device. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5140+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum angular limits for each joint The set value should be less than the actual maximum mechanical limit angle of each joint. | Degree, Axis 2 and Axis 3 are in mm | R/W | Stop PLC re-run |
| SFD5160+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | minimum angular limit of each joint Set the value to be greater than the actual mechanical minimum limit angle of each joint. | Degree, Axis 2 and Axis 3 are in mm | R/W | Stop PLC re-run |
| SFD5230+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (Cartesian motion) is typically set to match the SFD5040+2i configuration. | linear measure /s | R/W | Stop PLC re-run |
| SFD5370+i (where i ranges from 0 to 1,2,..., the maximum number of axes minus 1) | INT16U | Whether the axis is controlled. All robot axes must be set to 1. | 0: Uncontrolled Controlled | R/W | Stop PLC re-run |
| SFD6030 | float | Maximum speed of quaternion terminal attitude | Default: 100 degrees/s | R/W | Stop PLC re-run |
| SFD6032 | float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 $degrees/s^2$ | R/W | Stop PLC re-run |
| SFD6034 | float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop PLC re-run |

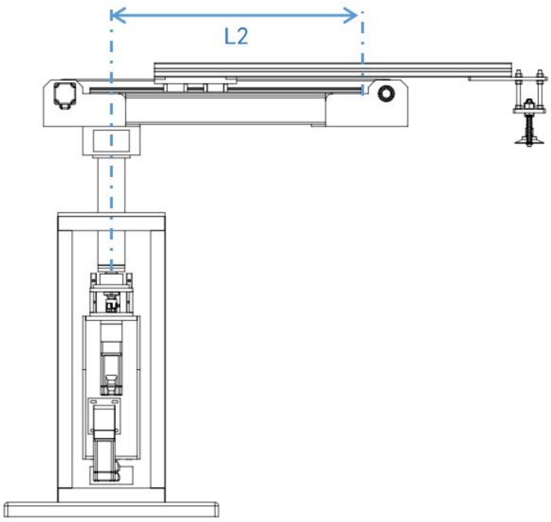
①: When configuring SFD5040, it must be compared with SFD3018. Since their units differ, the values must first be converted to the same unit. The corresponding pulse count is calculated as [SFD5040+2*i]*[SFD5020+2*i]*[SFD3002+60*i]/360. This value must be less than the corresponding axis's SFD3018+60i. Otherwise, the converted value of SFD3018+60i will be used as the maximum operating speed per axis of the robot.

②: The maximum terminal velocity can be deduced from the maximum joint velocity. Taking this model as an example, the maximum terminal velocity is calculated by combining the velocity of shaft one when rod 2 extends to its maximum length with the maximum velocities of shafts two and three. The formula for calculating the maximum terminal linear velocity is:

$$V_{\text{linear vebcity max}} = \sqrt{[V_{J1\max}(L_2 + V_{J2\max})]^2 + V_{J2\max}{}^2 + V_{J3\max}{}^2}$$

## 7.6 Stamping loading and unloading robot




| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD3047+60 *i (where i ranges from 0 to 1,2,..., down to the maximum number of axes minus 1) | INT16U | motor direction logic<br>Configure the joint rotation in both forward and reverse directions to ensure it matches the specified model. | 0: Positive logic<br>1: Anti-Logic | R/W | Stop PLC re-run |
| **SFD5000** | **INT16U** | **Robot type** | **5** | **R/W** | **Stop PLC re-run** |
| SFD5002 | float | L1: The offset between axis 1 and axis 2. | mm | R/W | Stop PLC re-run |
| SFD5004 | float | L2: Length of parameter 5 | mm | R/W | Stop PLC re-run |
| SFD5020+2* i (where i=0,1,2... to the maximum number of axes minus 1) | float | gear ratio of each joint<br>Consistent with the actual deceleration ratio of the joint | Rotate joint: Circle/Circle<br>Moving joint: mm/turn | R/W | Stop PLC re-run |

①: Connecting rod parameter 1: The horizontal distance between the centers of the first and second joint axes, measured along the Y-axis of the base coordinate system, as shown in Figure L1 above.

②: Connecting rod parameter 2: Crankshaft length, as shown in Figure L2 above.

③: Motor motion direction logic: Based on the joint settings in the diagram above, configure the rotation axis direction and define the positive/negative motion direction for each axis. Only two settings (0 or 1) are allowed to ensure that the joints' motion directions comply with the right-hand rule. Incorrect settings will result in erroneous Cartesian motion trajectories.

④: Robot Zero Position: The zero position configuration is shown in the figure above, with the input joint angles set to (0,0,0,0).

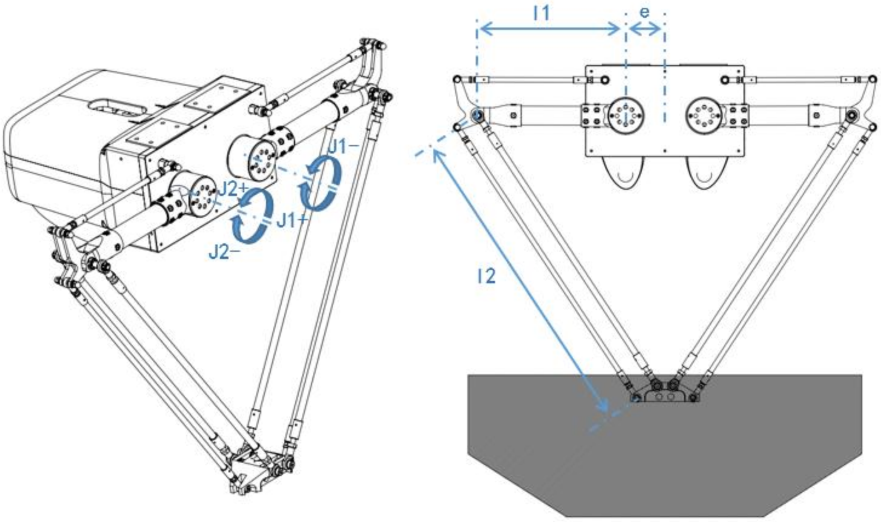
⑤: L1 offset: This parameter can be either positive or negative. The second axis line is offset along the positive Y-axis of the base coordinate system. A positive value indicates a forward offset, while a negative value indicates a backward offset.

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
| SFD5040+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (joint motion) is calculated by setting the motor-side SFD3018+60*i to the maximum motor speed, which is then applied to the joint output side ①. | m/s or mm/s | R/W | Stop PLC re-run |
| SFD5060+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint speed | $m/s^2$ or $mm/s^2$ | R/W | Stop PLC re-run |
| SFD5250+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5080 | float | Maximum end speed, [linear velocity] Linear velocity determined by joints 1, 2, and 3 | mm/s or degrees/s | R/W | Stop PLC re-run |
| SFD5090 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear speed | $mm/s^2$ or $degrees/s^2$ | R/W | Stop PLC re-run |
| SFD5270 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear acceleration, with specific values depending on the actual operating conditions of the device. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
| SFD5140+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum angular limits for each joint<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | Degree, Axis 2 and Axis 3 are in mm | R/W | Stop PLC re-run |
| SFD5160+2*i (where i ranges from 0 to 1, 2,..., down to the maximum number of axes minus 1) | float | minimum angular limit of each joint<br>Set the value to be greater than the actual mechanical minimum limit angle of each joint. | Degree, Axis 2 and Axis 3 are in mm | R/W | Stop PLC re-run |
| SFD5230+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (Cartesian motion) is typically set to match the SFD5040+2i configuration. | linear measure /s | R/W | Stop PLC re-run |
| SFD5370+i (where i ranges from 0 to 1,2,..., the maximum number of axes minus 1) | INT16U | Whether the axis is controlled. All robot axes must be set to 1. | 0: Uncontrolled Controlled | R/W | Stop PLC re-run |
| SFD6030 | float | Maximum speed of quaternion terminal attitude | Default: 100 degrees per second | R/W | Stop PLC re-run |
| SFD6032 | float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 degrees/$s^2$ | R/W | Stop PLC re-run |
| SFD6034 | float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop PLC re-run |

①: When configuring SFD5040, it must be compared with SFD3018. Since their units differ, the values must first be converted to the same unit. The corresponding pulse count is calculated as [SFD5040+2*i]*[SFD5020+2*i]*[SFD3002+60*i]/360. This value must be less than the corresponding axis's SFD3018+60i. Otherwise, the converted value of SFD3018+60i will be used as the maximum operating speed per axis of the robot.

## 7.7 Parallel SCARA Robot




| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD3047+60*i (where i ranges from 0 to 1,2,..., down to the maximum number of axes minus 1) | INT16U | motor direction logic Configure the joint rotation in both forward and reverse directions to ensure it matches the specified model. | 0: Positive logic<br>1: Anti-Logic | R/W | Stop PLC re-run |
| **SFD5000** | **INT16U** | **Robot type** | **13** | **R/W** | **Stop PLC re-run** |
| SFD5002 | float | Half of the distance between the rotation axes of the two active arms: e | mm | R/W | Stop PLC re-run |
| SFD5004 | float | Active arm length: l1 | mm | R/W | Stop PLC re-run |
| SFD5006 | float | Luffing arm length: l2 | mm | R/W | Stop PLC re-run |
| SFD5020+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | gear ratio of each joint Consistent with the actual deceleration ratio of the joint | Rotate joint: Circle/Circle<br>Moving joint: mm/turn | R/W | Stop PLC re-run |

①: Connecting rod parameter 1: Half of the distance between the rotation axes of the two active arms, i.e., parameter e in the diagram above.

②: Connecting rod parameter 2: the length of the active arm, which corresponds to parameter l1 in the diagram above.

③: Connecting rod parameter 3: the length of the follower arm, which corresponds to parameter l2 in the diagram above.

④: Motor motion direction logic: Based on the joint settings in the diagram above, configure the rotation axis direction and define the positive/negative motion direction for each axis. Only two settings (0 or 1) are allowed to ensure that the joints' motion directions comply with the right-hand rule. Incorrect settings will result in erroneous Cartesian motion trajectories.

⑤: Robot Zero Position: The zero position configuration is shown in the figure above, with the input joint angle set to (0,0). For joints 1 and 2, when the connected active arm is horizontal (on the x-axis of the coordinate system), the joint angle is 0°. When the active arm moves downward, the joint angle becomes positive, and conversely, negative.

⑥: The parallel SCARA robot, also known as a Diamond robot, has its end effector driven by two motors on a stationary platform to move along the XOY plane, which serves as the working plane. An additional Z-axis perpendicular to the working plane and an end-effector rotation axis can be added for spatial expansion. These extra axes can be configured as either controlled or uncontrolled based on practical needs.

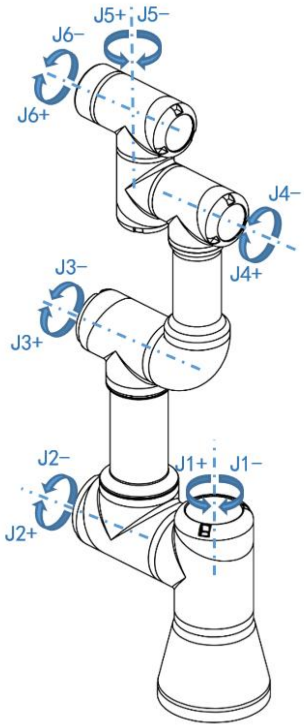
| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5040+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (joint motion) is calculated by setting the motor-side SFD3018+60*i to the maximum motor speed, which is then applied to the joint output side ①. | m/s or mm/s | R/W | Stop PLC re-run |
| SFD5060+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint The default setting is 5-10 times joint speed | $m/s^2$ or $mm/s^2$ | R/W | Stop PLC re-run |
| SFD5250+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5080 | float | Maximum speed at the end, [linear velocity] The linear velocity is determined by joint 1,2,3 | mm/s or degrees/s | R/W | Stop PLC re-run |
| SFD5090 | float | Maximum acceleration at the end, [linear acceleration] The default setting is 5-10 times the linear speed | $mm/s^2$ or $degrees/s^2$ | R/W | Stop PLC re-run |

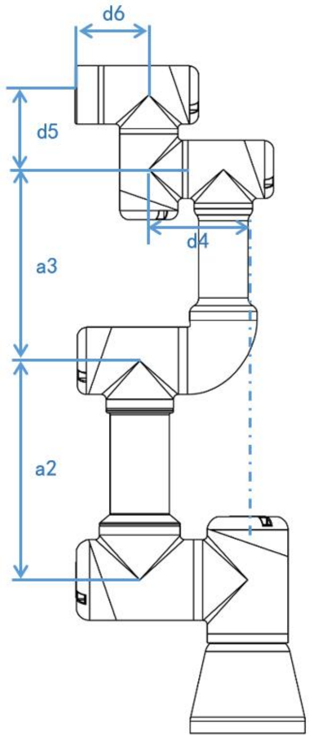
| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5270 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear acceleration, and the specific parameter values depend on the actual operating conditions of the device. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5140 | float | maximal angle restriction of joint 1<br>The set value should be smaller than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5142 | float | maximal angle limitation of joint 2<br>The set value should be smaller than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5144 | float | maximal angle restriction of joint 3<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | mm | R/W | Stop PLC re-run |
| SFD5146 | float | maximal angle restriction of joint 4<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5148 | float | Maximum X-axis position | mm | R/W | Stop PLC re-run |
| SFD5150 | float | Maximum Y position | mm | R/W | Stop PLC re-run |
| SFD5160 | float | minimum angular restriction of joint 1<br>The set value should be smaller than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5162 | float | minimum angular restriction of joint 2<br>The set value should be smaller than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5164 | float | minimum angular restriction of joint 3<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | mm | R/W | Stop PLC re-run |
| SFD5166 | float | minimum angular restriction of joint 4<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5168 | float | Minimum position in X direction | mm | R/W | Stop PLC re-run |
| SFD5170 | float | Minimum Y position | mm | R/W | Stop PLC re-run |

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5230+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (Cartesian motion) is typically set to match the SFD5040+2i configuration. | linear measure /s | R/W | Stop PLC re-run |
| SFD5370+i (where i ranges from 0 to 1,2,..., the maximum number of axes minus 1) | INT16U | Whether the axis is controlled. All robot axes must be set to 1. | 0: Uncontrolled Controlled | R/W | Stop PLC re-run |
| SFD6030 | float | Maximum speed of quaternion terminal attitude | Default: 100 degrees/s | R/W | Stop PLC re-run |
| SFD6032 | float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 degrees/s² | R/W | Stop PLC re-run |
| SFD6034 | float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop PLC re-run |

①: When configuring SFD5040, it must be compared with SFD3018. Since their units differ, the values must first be converted to the same unit. The corresponding pulse count is calculated as [SFD5040+2*i]*[SFD5020+2*i]*[SFD3002+60*i]/360. This value must be less than the corresponding axis's SFD3018+60i. Otherwise, the converted value of SFD3018+60i will be used as the maximum operating speed per axis of the robot.

## 7.8 Collaborative six-axis robot




| Robot model parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD3047+60*i (where i ranges from 0 to 1, with the maximum number of axes being 1 less than the total number) | INT16U | motor direction logic<br>The configuration must align with the joint's forward and reverse rotation to ensure its movement matches the predefined model. The motor's final rotation direction is determined by the joint's rotation and the axis's forward rotation direction in the servo parameters⑥ | 0: Positive logic<br>1: Anti-Logic | R/W | Stop PLC re-run |
| **SFD5000** | **INT16U** | **Robot type** | **14** | **R/W** | **Stop PLC re-run** |
| SFD5002 | float | Collaboration d1 | mm, default is 0 | R/W | Stop PLC re-run |
| SFD5004 | float | Collaboration-a2① | mm | R/W | Stop PLC re-run |
| SFD5006 | float | Collaboration-a3② | mm | R/W | Stop PLC re-run |
| SFD5008 | float | Collaboration d4③ | mm | R/W | Stop PLC re-run |
| SFD5010 | float | Collaboration d5④ | mm | R/W | Stop PLC re-run |
| SFD5012 | float | Collaboration d6⑤ | mm/ shut in a pen | R/W | Stop PLC re-run |

| | | | | | |
|---|---|---|---|---|---|
| SFD5020+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | gear ratio of each joint<br>Consistent with the actual deceleration ratio of the joint | Rotate joint: Circle/Circle<br>Moving joint: mm/turn | R/W | Stop PLC re-run |

①: Connecting rod parameter 1: The vertical distance a2 between the axis centers of the second and third joints, as shown in the figure above.

②: Connecting rod parameter 2: The vertical distance a3 between the axis center of the third joint and the axis center of the fourth joint, which corresponds to parameter 2 in the diagram above.

③: Connecting rod parameter 3: The vertical distance d4 from the axis center of the first joint to that of the fifth joint, as shown in the figure above.

④: Connecting rod parameter 4: The vertical distance d5 between the axis center of the 4th joint and the axis center of the 6th joint, which corresponds to parameter 4 in the diagram above.

⑤: Connecting rod parameter 5: The vertical distance d6 between the axis center of the 5th joint and the axis center of the 6th joint, which corresponds to parameter 5 in the diagram above.

⑥: Motor motion direction logic: Based on the joint settings in the diagram, configure the rotation axis direction and define the positive/negative motion direction for each axis. Only two settings (0 or 1) are allowed to ensure that the joints' motion directions comply with the right-hand rule. Incorrect settings will result in erroneous Cartesian motion trajectories.

⑦: Robot zero position: The zero position pose is shown in the figure above, with the input joint angles set as (0,0,0,0,0,0).

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5040+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (joint motion) is calculated by setting the motor-side SFD3018+60*i to the maximum motor speed, which is then applied to the joint output side. | m/s or mm/s | R/W | Stop PLC re-run |
| SFD5060+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint speed | $m/s^2$ or $mm/s^2$ | R/W | Stop PLC re-run |
| SFD5250+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC re-run |
| SFD5080 | float | Maximum speed at the end, [linear velocity] | mm/s or degrees/s | R/W | Stop PLC re-run |
| SFD5090 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear speed | $mm/s^2$ or degrees/$s^2$ | R/W | Stop PLC re-run |

| Robot operation parameters | | | | | |
|---|---|---|---|---|---|
| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| SFD5270 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear acceleration, with specific values depending on the actual operating conditions of the device. | $mm/s^3$ or $degrees/s^3$ | R/W | Stop PLC re-run |
| SFD5140+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Maximum angular limits for each joint<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5160+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | minimum angular limit of each joint Set the value to be greater than the actual mechanical minimum limit angle of each joint. | linear measure | R/W | Stop PLC re-run |
| SFD5230+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | The maximum speed of each joint (Cartesian motion) is typically set to match the SFD5040+2i configuration. | linear measure /s | R/W | Stop PLC re-run |
| SFD5370+i (where i ranges from 0 to 1,2,..., the maximum number of axes minus 1) | INT16U | Whether the axis is controlled. All robot axes must be set to 1. | 0: Uncontrolled, 1: Controlled | R/W | Stop PLC re-run |
| SFD6000+2*i (i=0,1,2...10) | float | Kinematic model calibration compensation values | A total of 11 floats are compensated, with specific parameters as specified in the calibration manual. | R/W | Stop PLC re-run |
| SFD6030 | float | Maximum speed of quaternion terminal attitude | Default: 100 degrees/s | R/W | Stop PLC re-run |
| SFD6032 | float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 $degrees/s^2$ | R/W | Stop PLC re-run |
| SFD6034 | float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop PLC re-run |

①: When configuring SFD5040, it must be compared with SFD3018. Since their units differ, the values must first be converted to the same unit. The corresponding pulse count is calculated as [SFD5040+2*i]*[SFD5020+2*i]*[SFD3002+60*i]/360. This value must be less than the corresponding axis's SFD3018+60i. Otherwise, the converted value of SFD3018+60i will be used as the maximum operating speed per axis of the robot.

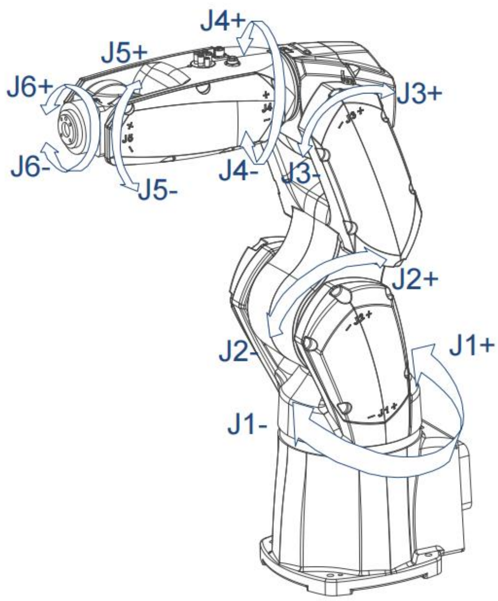
# 8. Robot Coordinate System

Common coordinate systems are shown in the table below.

| Coordinate system type | Description |
|---|---|
| Robot base coordinate system | The coordinate system is also called the base coordinate system, which is the coordinate system of the machine base mounting surface and coincides with the coordinate system of the robot joint 1, which is convenient for internal calculation and simplifying the body parameters. |
| World coordinate system | Known as the absolute coordinate system or geodetic coordinate system, this fixed reference frame is independent of robotic motion and uses earth as its reference point. Typically, the base coordinate system aligns with the world coordinate system. When the base coordinate system undergoes relative displacement from the geodetic coordinate system, an absolute coordinate system is established. For example, when a robot is mounted on a ground track, the world coordinate system coincides with the base coordinate system when the track is at the zero position. |
| Joint coordinate system | It is mainly used to determine the direction of the joint's rotation axis. |
| Tool coordinate system | The coordinate system used for tool tips, also known as the tcp coordinate system. |
| Workpiece coordinate system | A user-defined coordinate system, or a coordinate system defined on a part. |
| Other coordinate systems | It is generally used for specific processes, such as camera coordinate systems and belt coordinate systems. |

## 8.1 Joint Coordinate System

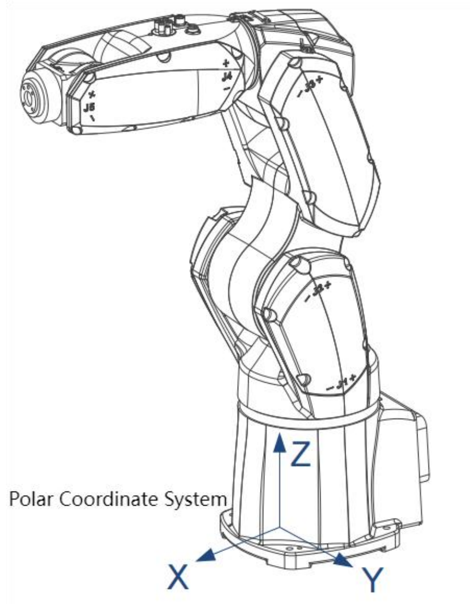
Take the six-axis robot as an example, each joint establishes a joint coordinate system, and each joint satisfies the right-hand rule. The establishment of each joint coordinate system constitutes the kinematic model of the robot body. Special attention should be paid to the logical direction of motor movement, which should be consistent with the positive and negative directions of the joint.



 User Manual of RC6F Series Robot Controller 8 Robot Coordinate System

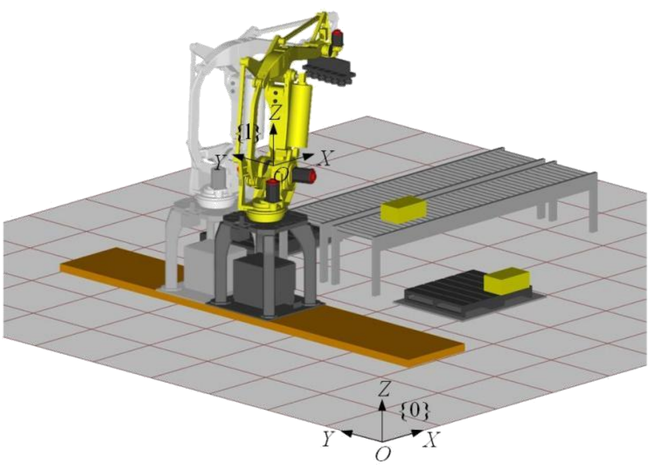
## 8.2 Base Coordinate System

Taking the six-axis robot as an example, in order to simplify the calculation, the base coordinate system of the robot is generally coincident with the joint coordinate system of the first axis.




## 8.3 World Coordinate System

The absolute coordinate system of a robotic system is defined as a reference frame that remains stationary relative to the geodetic coordinate system. Typically, the robot's base coordinate system serves as the absolute reference. However, when the robot's base coordinates shift relative to the geodetic system, this reference becomes invalid. For instance, if the robot is mounted on an orbital track, a stationary reference frame—such as the world coordinate system—must be established. This can be achieved by setting the origin on the orbital track or at another fixed stationary location.



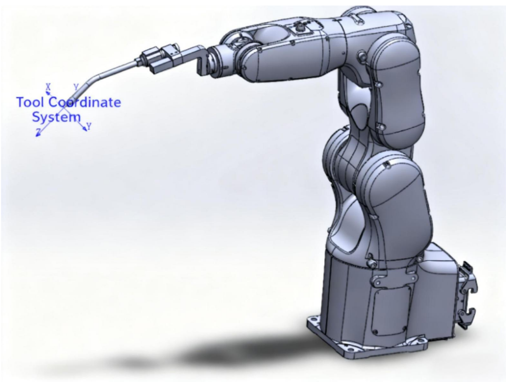
## 8.4 Tool Coordinate System

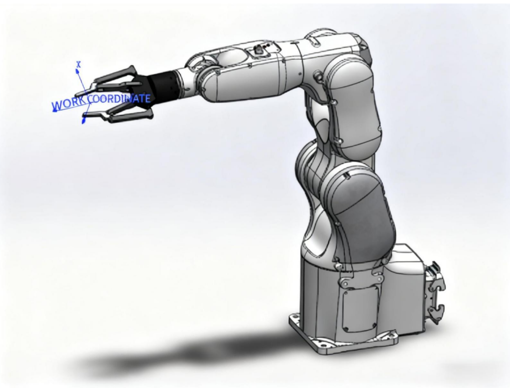
The tool coordinate system sets the tool center point as the zero position, which defines the tool's location and orientation. It is commonly abbreviated as TCPF (Tool Center Point Frame), with the center point abbreviated as TCP (Tool Center Point).

When executing the program, the robot moves the TCP to the programmed position. This means that if you change the tool (and its coordinate system), the robot will adjust its movement to ensure the new TCP reaches the target.

All robots have a predefined tool coordinate system called tool0 at the wrist. This allows one or more new tool coordinate systems to be defined as offsets from tool0.

The tool coordinate system is very useful when controlling a robot with point motion if you don't want to change the tool direction during movement.





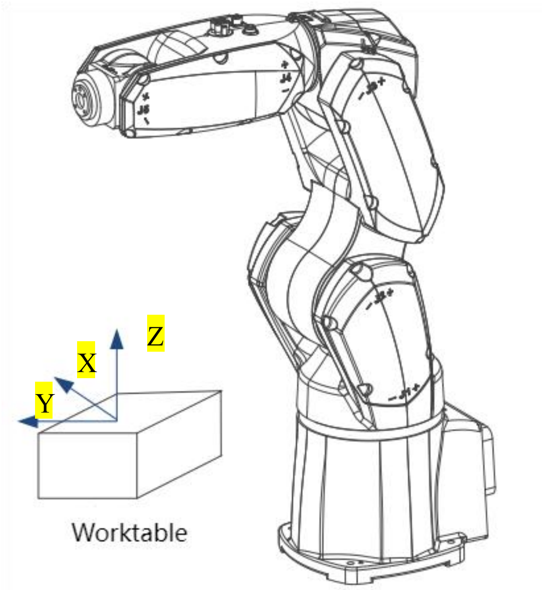

## 8.5 Workpiece Coordinate System

Workpiece coordinate system: It defines the position of the workpiece relative to the geodetic coordinate system (or other coordinate systems).

A robot can maintain multiple workpiece coordinate systems, either to represent different workpieces or multiple copies of the same workpiece at different positions. This offers significant advantages:

- To reposition a workpiece in the workstation, simply change the position of the workpiece coordinate system, and all paths will update immediately.
- Allow workpieces to move along other axes or transfer rails, as the entire workpiece can move with its path.

Manual of RC6F Series Robot Controller 8 Robot Coordinate System



User Manual of RC6F Series Robot Controller 9 Coordinate Value Representation

# 9. Coordinate values

In the reference coordinate system for position and orientation, position is represented by [x, y, z] while orientation is expressed through Euler angles [Rz, Ry, Rx]. For specific applications, orientation may alternatively be represented by unit quaternions [w, x, y, z]. Both tool parameters, workpiece parameters, and the base coordinate system of the positioner are represented by position and Euler angles in the world coordinate system. After writing the tool/workpiece parameters into the corresponding SFD registers (SFD5500, SFD5740), the parameters can be activated immediately by setting SM3018.

The manipulator can move to any position and display real-time information such as base coordinates and joint coordinates. The position display can be observed through corresponding parameters. The reference coordinate system for SD4000+2*i to display coordinate values can be adjusted by setting the SD4018 reference coordinate system type. For example:

- Joint position display in coordinate system, showing current joint angle values (unit: degrees or mm). Set SD4018=5 and read the value from SD4000.
- Display position in Cartesian coordinate system, showing the current end-effector pose coordinates (unit: mm or degrees). Set SD4018=0 and SD4015=0, then read the value from SD4000.
- The tool 1 coordinate system displays the current end effector pose coordinates (unit: mm or degrees). Set SD4018=0/4 and SD4015=1, then read the value from SD4000.
- The workpiece 1 coordinate system displays its position, showing the current end effector pose coordinates (unit: mm or degrees). Set SD4018=6 and SD4019=1, then read the value from SD4000.

| Register address | Assignment | Representative coordinate system |
|---|---|---|
| Sd4015 | 0 | Tool 0 |
| | 1 | Tool 1 |
| | ... | |
| | 19 | Tool 19 |
| Sd4019 | 0 | Workpiece 0① |
| | 1 | Workpiece 1 |
| | ... | |
| | 19 | Workpiece 19 |
| Sd4018 | 0 | Coordinate basis |
| | 1 | World coordinate system |
| | 4 | Tool coordinate system |
| | 5 | Joint coordinate system |
| | 6 | Workpiece coordinate system |

Using the 400mm-long Scara robot as an example, when the robot is at the zero position, SD4018=0 and SD4015=0, and the display shows the end position and orientation of the flange in the base coordinate system.

| SD4000(X) | SD4002(Y) | SD4004(Z) | SD4006(Rz) | SD4008(Ry) | SD4010(Rx) |
|---|---|---|---|---|---|
| 400 | 0 | 0 | 0 | 0 | 0 |

When SD4018 is set to 5 and SD4015 to 0, the system displays the angles of each axis in the joint coordinate system.

| SD4000 (Axis 1 Angle) | SD4002 (Axis 2 Angle) | SD4004 (Axis 3 Position) | SD4006 (Axis 4 Angle) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |

When tool 1 is set (X=-20, Y=0, Z=30) with no pose change, configure SD4018=0/4 and SD4015=1. The

User Manual of RC6F Series Robot Controller 9 Coordinate Value Representation

display will show the end-effector pose value in the tool coordinate system.

| SD4000(X) | SD4002(Y) | SD4004(Z) | SD4006(Rz) | SD4008(Ry) | SD4010(Rx) |
|---|---|---|---|---|---|
| 380 | 0 | 30 | 0 | 0 | 0 |

When setting workpiece 1 (X=200, Y=200, Z=100, Rz=45°, Ry=0, Rx=0), set SD4018=6 and SD4019=1. The system will then display the end-effector pose values in the workpiece coordinate system.

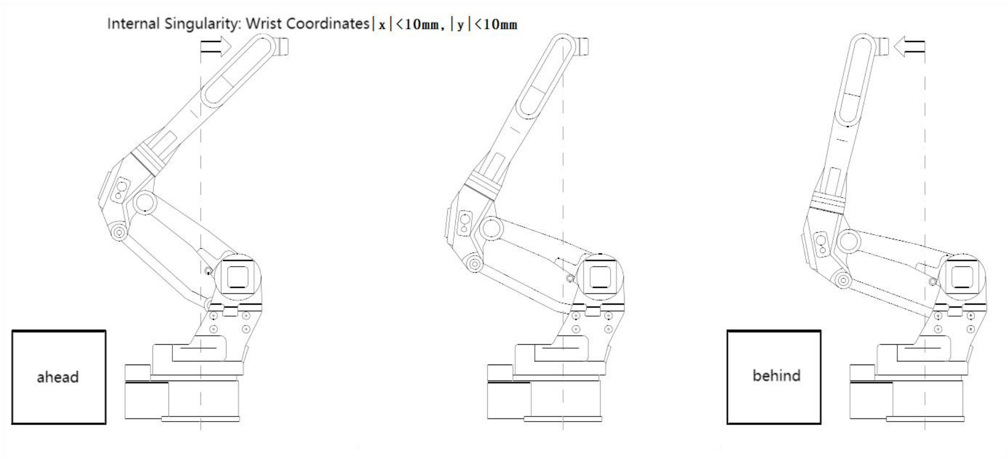
| SD4000(X) | SD4002(Y) | SD4004(Z) | SD4006(Rz) | SD4008(Ry) | SD4010(Rx) |
|---|---|---|---|---|---|
| 0 | -282.8422 | -100 | -45 | 0 | 0 |

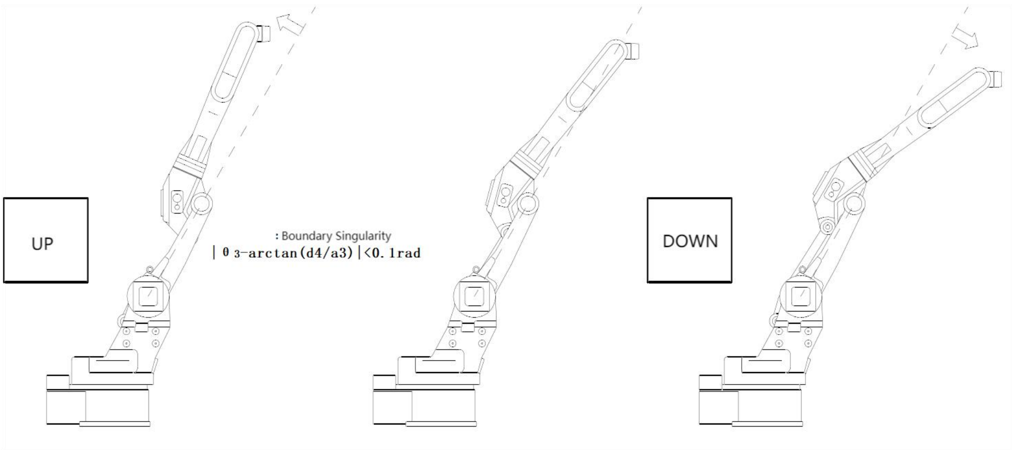
Before writing tool/workpiece parameters, all tool/workpiece parameters are set to 0, and writing values to tool 0 and workpiece 0 is invalid. The system always remains in a tool-free or workpiece-free state.

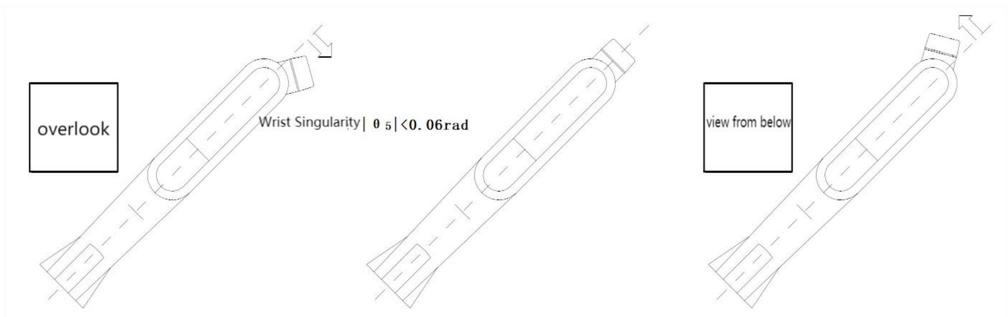
# 10. The Oddities and the Hand

Kinematic singularity is the inherent mechanical characteristic of industrial robot, which restricts the working space of robot on one hand, and on the other hand, when the manipulator moves to the singularity position, the inverse of Jacobian matrix is ill-conditioned, which requires infinite joint angular velocity and acceleration, and easily causes mechanical impact or vibration.

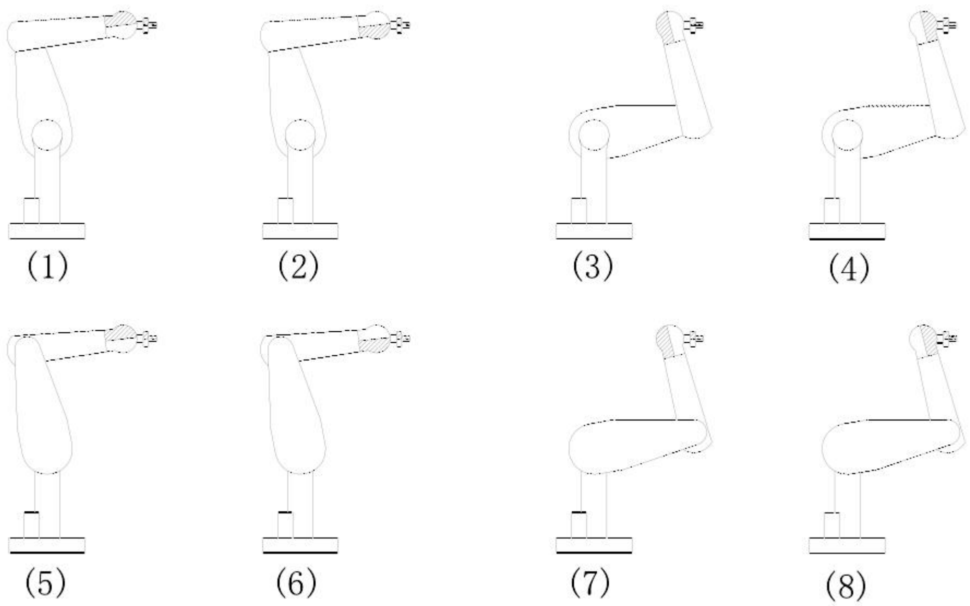
There are three kinds of singularities in the configuration of the six-axis robot: internal singularity, boundary singularity and wrist singularity.



The six-axis robot can reach the same spatial pose in eight forms, which represents the eight kinematic inverse solutions of the robot. The joint space is divided into eight regions by three singular regions, and the eight inverse solutions correspond to eight hand systems in the joint space.

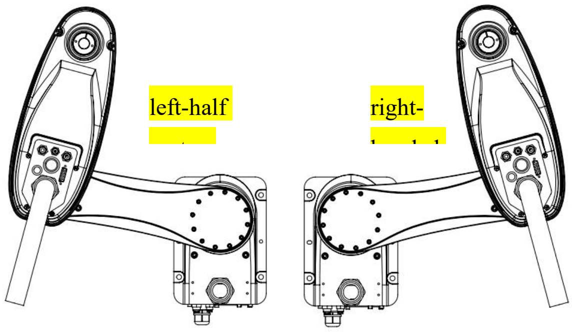



Define backForthType: 0 for forward, 1 for backward. handType: 0 for downward, 1 for upward. wristType: 0 for forward, 1 for backward.

Six-axis wrist joint parameter values: posType=wristType + handType*2 + backForthType*4

| Hand parameter value | The wrist center relative to axis 1 | Wrist center relative to axis 234 joint line | Axis 5 angle value |
|---|---|---|---|
| 0 | At the fore | Hereunder | The first month of the lunar year |
| 1 | At the fore | Hereunder | Burden |
| 2 | At the fore | On top | The first month of the lunar year |
| 3 | At the fore | On top | Burden |
| 4 | Rearwards | Hereunder | The first month of the lunar year |
| 5 | Rearwards | Hereunder | Burden |
| 6 | Rearwards | On top | The first month of the lunar year |
| 7 | Rearwards | On top | Burden |

The SCARA robot has only one singularity, known as the boundary singularity. The joint space is divided into two left-handed systems (joint angle <0°) and one right-handed system (joint angle> 0°) based on the angle of Joint 2. Note that if the joint angle exceeds ±180 degrees, two additional singularities will occur, specifically when the joint angle is 180 degrees. The figure below shows the left and right-handed systems of the SCARA robot in a top view.




| register address | type | definition | Value/Unit | read-write | Trigger time |
|---|---|---|---|---|---|
| SD4016 | INT16U | Current hand binding value of the robot | Model-dependent | R | |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

# 11. Robot stop

## 11.1 Stopping Mode

The point motion can be divided into joint point motion and Cartesian point motion according to the coordinate system, and the reference coordinate system of Cartesian point motion includes base coordinate system, tool coordinate system, world coordinate system and workpiece coordinate system.

Pressing the coordinate point movement SM3050+i or the joint point movement register SM3070+i initiates a single-step point movement, with the step length SD4100/SD4110 adjustable. Holding the coordinate point movement SM3050+i or the joint point movement register SM3070+i activates continuous point movement.


The robot can perform point movements according to different coordinate systems, and the selection of these coordinate systems follows the SD4018 and SD4015 standards.

| Parameter category | Address | Definition | Read-write | Trigger time |
|---|---|---|---|---|
| Condition monitoring | SM3001 | The robot is moving<br>0: End of exercise<br>1: Exercise in progress | R | |
| | SM3009 | Robot command given completed<br>0: Setting the instruction position<br>1: End of instruction position given | R | |
| Cartesian point control word | SM3050 | X forward direction | W/r | With immediate effect |
| | SM3051 | X negative direction | W/r | With immediate effect |
| | SM3052 | Y forward direction | W/r | With immediate effect |
| | SM3053 | Y negative direction | W/r | With immediate effect |
| | SM3054 | Z forward direction | W/r | With immediate effect |
| | SM3055 | Z negative direction | W/r | With immediate effect |
| | SM3056 | Rz forward direction | W/r | With immediate effect |
| | SM3057 | Rz negative direction | W/r | With immediate effect |
| | SM3058 | Ry forward direction | W/r | With immediate |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Parameter category | Address | Definition | Read-write | Trigger time |
|---|---|---|---|---|
| | | | | effect |
| | SM3059 | Ry negative direction | W/r | With immediate effect |
| | SM3060 | Rx forward direction | W/r | With immediate effect |
| | SM3061 | Rx negative direction | W/r | With immediate effect |
| Joint dynamic control word | SM3070 | J1 forward | W/r | With immediate effect |
| | SM3071 | J1 negative | W/r | With immediate effect |
| | SM3072 | J2 forward | W/r | With immediate effect |
| | SM3073 | J2 negative | W/r | With immediate effect |
| | SM3074 | J3 forward | W/r | With immediate effect |
| | SM3075 | J3 negative | W/r | With immediate effect |
| | SM3076 | J4 forward | W/r | With immediate effect |
| | SM3077 | J4 negative | W/r | With immediate effect |
| | SM3078 | J5 forward | W/r | With immediate effect |
| | SM3079 | J5 negative | W/r | With immediate effect |
| | SM3080 | J6 forward | W/r | With immediate effect |
| | SM3081 | J6 negative | W/r | With immediate effect |

## 11.2 Pointing step

The step distance indicates the movement distance for a single step, measured in millimeters or degrees. When a robot performs a single step, you can set the step distance register before executing the step.

| Basic point movement step value | | | | | | |
|---|---|---|---|---|---|---|
| Parameter category | Address | Register type | Definition | Unit | Read-write | Trigger time |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| | | | | | | |
|---|---|---|---|---|---|---|
| Point movement parameter | SFD5120 | Float | Cartesian point step size | Mm | R/W | Stop plc re-run |
| | SFD5130 | Float | Joint dynamic step size | Linear measure | R/W | Stop plc re-run |

| Effective point movement step | | | | | | |
|---|---|---|---|---|---|---|
| Parameter category | Address | Register type | Definition | Unit | Read-write | Trigger time |
| Point movement parameter | SD4100 | Float | Cartesian point step size | Mm | R/W | With immediate effect |
| | SD4110 | Float | Joint dynamic step length | Linear measure | R/W | With immediate effect |

During each power-on initialization, the PLC copies the value of the SFD register (Stop/Start Frequency) to the corresponding SD register. The SFD register retains its value after power-off, requiring a PLC power-off and reboot to apply the changes. Modifications to the SD register take effect immediately, with the data being reset to its original value upon the PLC's restart. The robot's stop/start parameters are determined by the values stored in the corresponding SD register.

## 11.3 Stop speed

The point speed indicates the maximum achievable speed during a point movement, measured in percentage. Before executing the point movement, you can set the point speed percentage register to adjust it. Due to the point movement distance limit, the actual running speed may not reach the set maximum value.

| Basic values of dot movement parameters | | | | | | |
|---|---|---|---|---|---|---|
| Parameter category | Address | Register type | Definition | Unit | Read-write | Trigger time |
| Point movement parameter | SFD5122 | Float | Percentage of coordinate point movement speed | % | R/W | Stop plc re-run |
| | SFD5124 | Float | Percentage of acceleration of the coordinate point | % | R/W | Stop plc re-run |
| | SFD5132 | Float | Percentage of the axis point's moving speed | % | R/W | Stop plc re-run |
| | SFD5134 | Float | Percentage of the axis point's dynamic acceleration | % | R/W | Stop plc re-run |
| Joint motion control parameters | SFD5040+2*I (WHERE I RANGES FROM 0 TO 1, 2,..., THE MAXIMUM NUMBER OF AXES MINUS 1) | Float | Maximum speed of each joint | Linear measure /s | R/W | Stop plc re-run |
| | SFD5060+2*I (WHERE I RANGES | Float | Maximum acceleration of each joint | Linear measure | R/W | Stop plc re-run |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Basic values of dot movement parameters | | | | | | |
|---|---|---|---|---|---|---|
| Parameter category | Address | Register type | Definition | Unit | Read - write | Trigger time |
| | FROM 0 TO 1, 2,..., THE MAXIMUM NUMBER OF AXES MINUS 1) | | | /s² | | |
| | SFD5250+2*I (WHERE I RANGES FROM 0 TO 1,2,..., THE MAXIMUM NUMBER OF AXES MINUS 1) | Float | Maximum acceleration of each joint | M/s³ | R/W | Stop plc re-run |
| Cartesian motion control parameters | SFD5080 | Float | X/Y/Z linear velocity | Mm/s | R/W | Stop plc re-run |
| | SFD5090 | Float | X/Y/Z line acceleration | Mm/s² | R/W | Stop plc re-run |
| | SFD5270 | Float | X/Y/Z line acceleration | Mm/s³ | R/W | Stop plc re-run |
| | SFD6030 | Float | Maximum speed of quaternion terminal attitude | Linear measure /s | R/W | Stop plc re-run |
| | SFD6032 | Float | Maximum acceleration of quaternion terminal attitude | Linear measure /s² | R/W | Stop plc re-run |
| | SFD6034 | Float | Maximum acceleration of quaternion terminal attitude | M/s³ | R/W | Stop plc re-run |

| Active value of dot movement parameter | | | | | | |
|---|---|---|---|---|---|---|
| Parameter category | Address | Register type | Definition | Unit | Read-write | Trigger time |
| Point movement parameter | SD4102 | Float | Percentage of coordinate point movement speed | % | R/W | With immediate effect |
| | SD4104 | Float | Percentage of acceleration of the coordinate point | % | R/W | With immediate effect |
| | SD4112 | Float | Percentage of the axis point's moving speed | % | R/W | With immediate effect |
| | SD4114 | Float | Percentage of the axis point's dynamic acceleration | % | R/W | With immediate effect |
| Joint motion parameters | SD4700+2*I (WHERE I=0,1,2... TO THE MAXIMUM NUMBER OF AXES MINUS 1) | Float | The maximum speed of each joint is transmitted by the sfd5040 and can be modified before the user executes the command. | Linear measure /s | R/W | With immediate effect |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Active value of dot movement parameter | | | | | | |
|---|---|---|---|---|---|---|
| Parameter category | Address | Register type | Definition | Unit | Read-write | Trigger time |
| | SD4720+2*I (WHERE I=0,1,2... TO THE MAXIMUM NUMBER OF AXES MINUS 1) | Float | The maximum acceleration of each joint is transmitted by the sfd5060, and users can modify it before running the command. | Linear measure /$s^2$ | R/W | With immediate effect |
| | SD5000+2*I (WHERE I=0,1,2... TO THE MAXIMUM NUMBER OF AXES MINUS 1) | Float | The maximum acceleration of each joint is transmitted by the sfd5250 and can be modified before the user executes the command. | M/$s^3$ | R/W | With immediate effect |
| Cartesian motion parameters | SD4750+2*I (I=0,1,2,3) | Float | The maximum speed at the end is transmitted by the sfd5080 and can be modified before the user executes the command. | Mm/s | R/W | With immediate effect |
| | SD4770+2*I (I=0,1,2,3) | Float | The maximum terminal acceleration is transmitted by the sfd5090 and can be modified before the user executes the command. | Mm/$s^2$ | R/W | With immediate effect |
| | SD5020+2*I (I=0,1,2,3) | Float | The maximum terminal acceleration is transmitted by the sfd5270 and modified before the user executes the command. | Mm/$s^3$ | R/W | With immediate effect |
| Continuous stop-and-go parameter | SD5595 | INT16U | Maximum stopping time for continuous point motion | Default: 500ms | R/W | With immediate effect |
| Continuous point filtering time | SFD5136 | INT16U | Set the continuous point filtering duration. The larger the value, the longer the continuous point trigger wait time. The smaller the value, the more responsive the response. Range: 3-50, unit: 10ms cycles. | Default 3 | R/W | Stop plc re-run |
| | SFD5137 | INT16U | Set the stop filter time for point stops. A larger value increases the stop duration, while a smaller value improves | Default 1 | R/W | Stop plc re-run |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Active value of dot movement parameter | | | | | | |
|---|---|---|---|---|---|---|
| Parameter category | Address | Register type | Definition | Unit | Read-write | Trigger time |
| | | | responsiveness. Range: 1-50, unit: 10ms cycles. | | | |
| | SD4116 | INT16U | After the plc starts, the sfd5136 transmits the data. Users can modify it before executing the command. | Default 3 | R/W | With immediate effect |
| | SD4118 | INT16U | After the plc starts, the sfd5137 transmits the data. Users can modify it before executing the command. | Default 1 | R/W | With immediate effect |

- During each power-on initialization, the PLC copies the stop speed parameter value from the SFD register to the corresponding SD register. The SFD register retains its value after power-off, requiring a PLC power-off and reboot for the change to take effect. SD register values are updated in real-time, with the data being reset to its original state after the PLC restarts. The robot's stop speed parameters are determined by the values stored in the corresponding SD register.
- During continuous point motion, resetting the point control word and executing the stop motion will cause the robot to remain stationary for a specified duration, with the calculation formula as follows:

$$Time = SD4102/SD4112 * SD5595$$

The pause duration is determined by the current stop speed as a percentage of the maximum pause time, with a minimum of 0.1 times SD5595.

- For robots with movable joints, the Cartesian point speed reference is determined by the joint's maximum speed. For example, in the SCARA model's axis 3, the reference speed is set to the axis 3's maximum speed (SFD5044), rather than the Cartesian maximum speed.
- The speed, acceleration and corresponding SD register value of the point motion of the coordinate point or the axis point will be automatically adjusted in the range of 0.01~100.

## 11.4 Pulsating dynamic speed control

The speed of the stop-motion is changed by modifying the stop-motion speed percentage register (SD4102, SD4112) in real time.

| Address | Register type | Definition | Unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5575 | INT16U | The shift to the gear takes the most time | Default: 50ms | R/W | With immediate effect |

When the robot changes its speed value SD4102 or SD4112 during continuous point movements, it will complete the speed switch within SD5575 time. The calculation formula is as follows:

$$Time = \frac{\text{Speed Ratio Difference}}{100} * SD5575$$

$$(\text{The speed ratio difference refers to the difference between the pre} - \text{adjustment and post} - \text{adjustment values of SD4102 or SD4112.})$$

The minimum time was 0.1 times of SD5575.

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

# 12. Basic movement commands for robots

## 12.1 Robot-enabled RBON instructions

### 12.1.1 Overview of the Instructions

Enable the robot's axes and system, only after enabling can each module operate normally.

### 12.1.2 Command Application


- When M0 is ON, all robot axes and the robot system are enabled.
- When M0 is set to OFF, disable all robot axes and the robot system.

## 12.2 Joint Movement RBPTP Directive

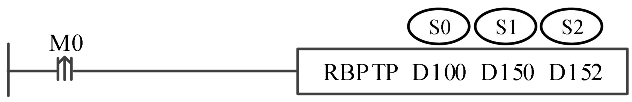
### 12.2.1 Overview of the Instructions

From the current point to the target point, the motion path is not fixed, and the interpolation motion of the joint space is performed.

| Two-point trajectory motion rbptp s0 s1 s2 | | |
|---|---|---|
| Operand | Function | Type |
| S0 | First address number of the target point (target point coordinates) | Float mould |
| S1 | Soft component address number specifying the maximum speed percentage, 0 to 100 (register) | Float mould |
| S2 | Address of the soft component specifying the maximum acceleration percentage, ranging from 0 to 100 (register). | Float mould |

### 12.2.2 Command Applications

The trajectory between the current point and the target point is not fixed.




- Move from the current point to the target point during the M0 rising edge.
- During the operation, the real-time coordinate change of SD4000+2*N can be monitored.
- S0-The first address D100 for trajectory point positions, with each point occupying 20 bytes. The fields are X, Y, Z, RZ, RY, RX, and attributes. X, Y, Z, RZ, RY, and RX are coordinate values, all of float type. The address table is as follows:

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Address | Content | Remarks |
|---|---|---|
| S0+0（float) | X-axis position of the target point | The four axes are x, y, z, and rz. If the joint coordinate system is selected, the corresponding axis angles are displayed. |
| S0+2（float) | Target point y-axis position | |
| S0+4（float) | Z-axis position of the target point | |
| S0+6（float) | Target point rz axis position | |
| S0+8（float) | Target point ry axis position | |
| S0+10（float) | Rx axis position of the target point | |
| S0+12-18 | Attribute | |

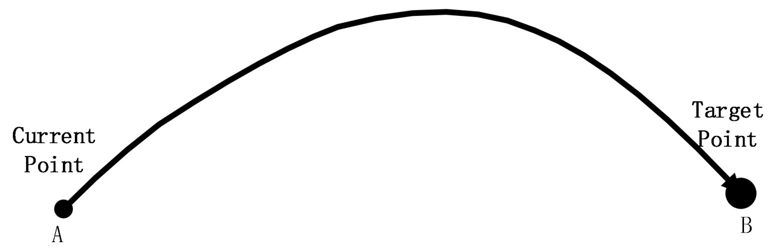
The properties include the smoothness of the point, hand binding, and reference tool coordinate system. The property table is as follows:

| Address | Meaning | Numerical meaning |
|---|---|---|
| S0+15(INT16U) | Reference id | 0: No tools<br>Tool 1<br>2: Tool 2<br>...<br>19: Tool 19 |
| S0+16(INT16U) | Attitude property | Hand assignment |
| S0+17(INT16U) | Smoothness property | Indicates the smoothness of the point. The lower the value, the closer it is to the target point (range: 0-200). |
| S0+18(INT16U) | Reference coordinate system | 0: Base coordinate system<br>1: World coordinate system<br>4: Tool coordinate system<br>5: Joint coordinate system<br>6: Workpiece coordinate system |
| S0+19(INT16U) | Workpiece coordinate system | 0: No workpiece<br>1: Workpiece 1<br>2: Workpiece 2<br>...<br>19: Workpiece 19 |

- S1 — Motion speed setting, a float data type indicating the maximum speed percentage. For example, 10.5 means 10.5% of the maximum speed. If the value is negative, the system will automatically operate at a positive speed. If the value exceeds 100, it will run at 100% of the maximum speed. The speed reference value is configured in SFD5040+2i.
- S2 — Motion acceleration setting, a float data type indicating the maximum acceleration percentage. For example, 10.5 means 10.5% of the maximum acceleration. If the value is negative, motion is automatically set to positive. If the value exceeds 100, motion is set to 100% of the maximum acceleration. The acceleration reference value is configured in SFD5060+2i.
- When the percentage settings for speed and acceleration in S1/S2 are ≤0.01, alarms 10120 (no speed set) and 10121 (no acceleration set) will be triggered. Priority 10120> 10121, and motion commands will not be executed. If the value exceeds 100, it will be adjusted to 100.

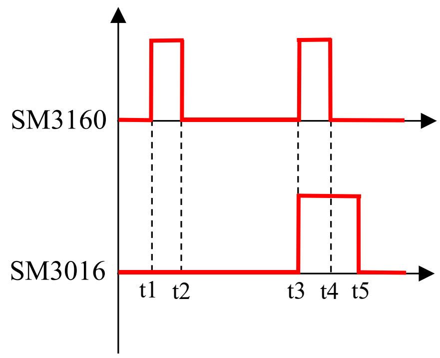
For example, programming a robotic arm to move from point A to point B (using SCARA as an example).

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot




Schematic diagram of the path planning for the robotic arm

（1）The position coordinates of target point B are stored in the register starting from D100.
（2）The percentage of the speed and acceleration of the manipulator is stored in register D150 and D152 respectively.
（3）The property defaults to no tool, default base coordinate system, and free hand binding. To use the tool coordinate system, set the SD5503 to the same tool number to ensure proper monitoring.
（4）Write the instruction RBPTP D100 D150 D152.
（5）When the robotic arm moves from point A to B, if multiple RBPTP motions are triggered, the smoothness can be set by the user.




The first RB instruction is triggered at t1. After instruction parsing, the robot initiates movement. When the first instruction reaches the constant-speed phase at t3, SM3016 is turned on, enabling pre-processing of the next trajectory's point data. The completion of the first trajectory depends on the smoothing factor: a value of 0 indicates full execution, while a non-zero value suggests incomplete execution. For trajectories with a smoothing factor, SM3016 is turned off at t5 when transition is permitted. For trajectories without a smoothing factor, SM3016 is turned off at t5 when the current trajectory's speed drops to zero.

## 12.3 Straight-line movement RBLINE instruction

### 12.3.1 Overview of the Instructions

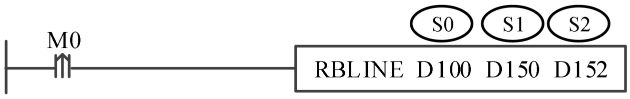
The path of different types of manipulators is the same from the current point to the target point.

| Two-point trajectory movement rbline s0 s1 s2 | | |
|---|---|---|
| Operand | Function | Type |
| S0 | First address number of the target point (target point coordinates) | Float mould |
| S1 | Soft component address number specifying the maximum speed percentage, 0 to 100 (register) | Float mould |
| S2 | Address of the soft component specifying the maximum acceleration percentage, ranging from 0 to 100 (register). | Float mould |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

### 12.3.2 Command Application

The path is planned by linear interpolation from the current point to the set point. The path is the same for different types of manipulators.




- When the M0 rising edge occurs, it moves in a straight line from the current edge to the target point.
- During the operation, the real-time coordinate change of SD4000+2*N can be monitored.
- S0-The first address D100 of the trajectory point positions, with each point occupying 20 bytes. The fields are arranged sequentially as X, Y, Z, RZ, RY, RX, and attributes. X, Y, Z, RZ, RY, and RX are coordinate values, all of which are float-type. The address table is as follows:

| address | content | remarks |
|---|---|---|
| S0+0（float) | X-axis position of the target point | The four axes are X, Y, Z, and RZ. If the joint coordinate system is selected, the corresponding axis angles are displayed. |
| S0+2（float) | Target point Y-axis position | |
| S0+4（float) | Z-axis position of the target point | |
| S0+6（float) | RZ axis position of target point | |
| S0+8（float) | Target point RY axis position | |
| S0+10（float) | RX axis position of the target point | |
| S0+12-18 | attribute | |

The properties include the smoothness of the point, hand binding, and reference tool coordinate system. The property table is as follows:

| Address | Meaning | Numerical meaning |
|---|---|---|
| S0+15(INT16U) | Reference id | 0: No tools<br>Tool 1<br>2: Tool 2<br>...<br>19: Tool 19 |
| S0+16(INT16U) | Attitude property | Hand assignment |
| S0+17(INT16U) | Smoothness property | Indicates the smoothness of the point. The lower the value, the closer it is to the target point (0-200). |
| S0+18(INT16U) | Reference coordinate system | 0: Base coordinate system<br>1: World coordinate system<br>4: Tool coordinate system<br>5: Joint coordinate system<br>6: Workpiece coordinate system |
| S0+19(INT16U) | Workpiece coordinate system | 0: No workpiece<br>1: Workpiece 1<br>2: Workpiece 2<br>...<br>19: Workpiece 19 |

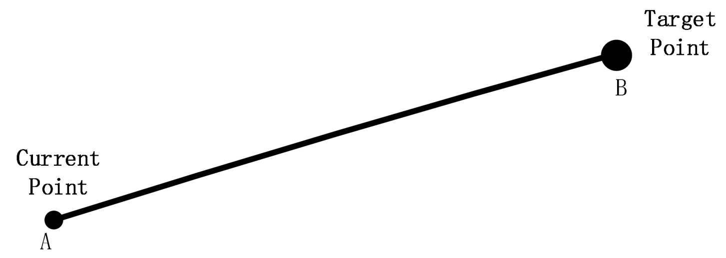
- S1 — Motion speed setting, a float data type indicating the maximum speed percentage. For example, 10.5 means 10.5% of the maximum speed. If the value is negative, the system will automatically operate at a positive speed. If the value exceeds 100, it will run at 100% of the maximum speed. The speed reference value is configured in SFD5080+2i.
- S2 — Motion acceleration setting, a float data type indicating the maximum acceleration percentage. For example, 10.5 means 10.5% of the maximum acceleration. If the value is negative, motion is

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

automatically set to positive. If the value exceeds 100, motion is set to 100% of the maximum acceleration. The acceleration reference value is configured in SFD5090+2i.

- When the percentage settings for speed and acceleration in S1/S2 are ≤0.01, alarms 10120 (no speed set) and 10121 (no acceleration set) will be triggered. Priority 10120> 10121, and the motion command will not be executed.

For example, programming a robotic arm to move from point A to point B (using SCARA as an example).




Schematic diagram of the path planning for the robotic arm

（1）The position coordinates of target point B are stored in the register starting with D100.

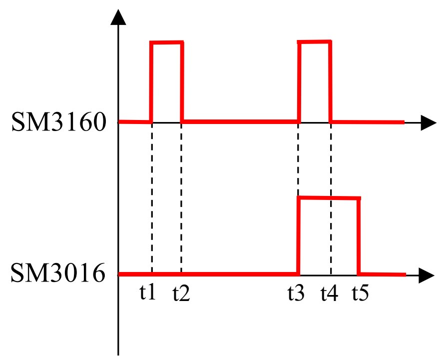
（2）The percentage of the speed and acceleration of the manipulator is stored in register D150 and D152 respectively.

（3）Read the SD4016 (INT16) value and write it into D116, ensuring that the low 8 bits of the current point and the target point at position S0+18 are identical. otherwise, the instruction will execute abnormally.

（4）To ensure proper monitoring, set the SD5503 to the same tool number when using the tool coordinate system.

（5）Write the instruction RBLINE D100 D150 D152.

（6）The robotic arm moves from point A to B along a preset trajectory. When multiple RBLINE motions are triggered, users can set the smoothness level.




The first RB instruction is triggered at t1. After instruction parsing, the robot initiates movement. When the first instruction reaches the constant-speed phase at t3, SM3016 is turned on, enabling pre-processing of the next trajectory's point data. The completion of the first trajectory depends on the smoothing factor: a value of 0 indicates full execution, while a non-zero value suggests incomplete execution. If a smoothing factor is applied between trajectories, SM3016 is turned off at t5 to allow transition. If no smoothing factor exists, SM3016 is turned off at t5 when the current trajectory's speed drops to zero.

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

# 12.4 Circular Motion RBCIRCLE Instruction

## 12.4.1 Overview of the Instructions

Move from the current point to the target point along an arc path with a fixed movement route.

| Two-point trajectory movement rbcircle s0 s1 s2 | | |
|---|---|---|
| Operand | Function | Type |
| S0 | Transition point first address number (transition point coordinate address) | Float mould |
| S1 | First address number of the target point (target point coordinates) | Float mould |
| S2 | Soft component address number specifying the maximum speed percentage, 0 to 100 (register) | Float mould |
| S3 | Address of the soft component specifying the maximum acceleration percentage, ranging from 0 to 100 (register). | Float mould |

## 12.4.2 Command Applications

Move from the current point to the target point along an arc path with a fixed trajectory between the two points.


- Move from the current point to the target point along an arc during the M0 rising edge. If the robot is in motion, this step is skipped.
- During the operation, the real-time coordinate change of SD4000+2*N can be monitored.
- For S0 and S1 (transition points), the first address D100 and target point D120 must have identical attributes. Each point occupies 20 bytes, with the following fields in sequence: X, Y, Z, RZ, RY, RX, and attributes. The X, Y, Z, RZ, RY, and RX fields are coordinate values, all of float type. The address table is as follows:

| Address | Content | Remarks |
|---|---|---|
| S0+0（float) | X-axis position of the target point | The four axes are x, y, z, and rz. If the joint coordinate system is selected, the corresponding axis angles are displayed. |
| S0+2（float) | Target point y-axis position | |
| S0+4（float) | Z-axis position of the target point | |
| S0+6（float) | Rz axis position of target point | |
| S0+8（float) | Target point ry axis position | |
| S0+10（float) | Rx axis position of the target point | |
| S0+12-18 | Attribute | |

The properties include the smoothness of the point, hand binding, and reference tool coordinate system. The property table is as follows:

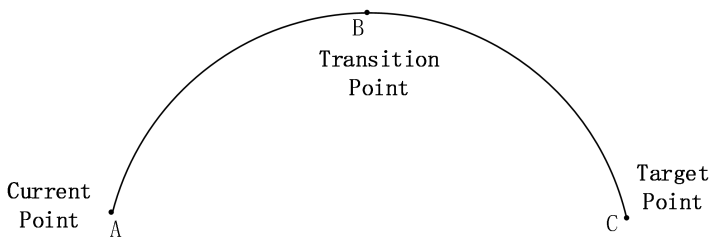
| Address | Meaning | Numerical meaning |
|---|---|---|
| S0+15(INT16U) | Reference id | 0: No tools<br>Tool 1<br>2: Tool 2<br>...<br>19: Tool 19 |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| | | |
|---|---|---|
| S0+16(INT16U) | Attitude property | Hand assignment |
| S0+17(INT16U) | Smoothness property | Indicates the smoothness of the point. The lower the value, the closer it is to the target point (range: 0-200). |
| S0+18(INT16U) | Reference coordinate system | 0: Base coordinate system<br>1: World coordinate system<br>4: Tool coordinate system<br>5: Joint coordinate system<br>6: Workpiece 0 coordinate system |
| S0+19(INT16U) | Workpiece coordinate system | 0: No workpiece<br>1: Workpiece 1<br>2: Workpiece 2<br>...<br>19: Workpiece 19 |

- S2 — Motion speed setting, a float data type indicating the maximum speed percentage. For example, 10.5 means 10.5% of the maximum speed. If the value is negative, the system will automatically operate at a positive speed. If the value exceeds 100, it will run at 100% of the maximum speed. The speed reference value is configured in SFD5080+2i.
- S3 — Motion acceleration setting, a float data type indicating the maximum acceleration percentage. For example, 10.5 means 10.5% of the maximum acceleration. If the value is negative, the system will automatically accelerate as if it were positive. If the value exceeds 100, the system will accelerate at 100% of the maximum. The acceleration reference value is set in SFD5090+2i.
- When the percentage settings for speed and acceleration in S2/S3 are ≤0.01, alarms 10120 (no speed set) and 10121 (no acceleration set) will be triggered. Priority 10120> 10121, and the motion command will not be executed.

For example, programming a robotic arm to move from point A through transit point B to point C (using SCARA as an example).




Schematic diagram of the path planning for the robotic arm

（1）The position coordinates of the transition points and target points B and C are stored in registers starting from D100 and D120. The given points should be reasonable to ensure that the arc trajectory can be completed within the working radius.

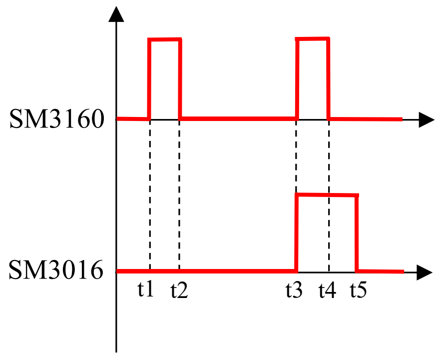
（2）The percentage of the speed and acceleration of the manipulator is stored in register D150 and D152 respectively.

（3）Read the SD4016 (INT16) value and write it into D116 and D136, ensuring that the low 8 bits of the current point, transition point, and target point at position S0+18 are identical. otherwise, the instruction will execute abnormally. Additionally, the tool numbers and workpiece numbers for the auxiliary point and target point must also be consistent.

（4）Write the instruction RBCIRCLE D100 D120 D150 D152.

（5）When the robotic arm moves from point A to C via transition point B, the system allows users to set the smoothness level for multiple RBCIRCLE motions.

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot




The first RB instruction is triggered at time t1. After instruction parsing, the robot initiates movement. When the first instruction reaches the constant-speed phase at t3, SM3016 is turned on, enabling pre-processing of the next trajectory's point data. The completion of the first trajectory depends on the smoothing factor: a value of 0 indicates full execution, while a non-zero value suggests incomplete execution. For trajectories with a smoothing factor, SM3016 is turned off at t5 when transition is permitted. For trajectories without a smoothing factor, SM3016 is turned off at t5 when the current trajectory's speed drops to zero.

## 12.5 Stop Movement RBSTOP Command

### 12.5.1 Overview of the Instructions

You can stop the robot.

| Stop movement rbstop s0 | | |
|---|---|---|
| Operand | Function | Type |
| S0 | Stop method:<br>0: Slow down and stop.<br>-1 indicates an emergency stop.<br>If greater than 0, this value is used as the given length to decelerate and stop along the tangent direction of the current trajectory. | Float |

### 12.5.2 Command Application


- When the M0 rising edge occurs, the robotic arm decelerates and stops or brakes according to the command. If the robot is stationary, this action is ignored. After decelerating and stopping with an input of 0, the system will resume the RBGOON command to continue the unfinished movement. The RBGOON command cannot be executed after a brake or a fixed-length stop.
- D0 is a floating-point register. A value of 0 triggers a deceleration stop, -1 a sudden stop, and any value greater than 0 sets the deceleration stop duration.
- This instruction is related to SD4171. When the slow stop is triggered, a variable-period slow stop will be executed. The slow stop duration is calculated as follows:

$$Time = \frac{SD4140}{SD4750} * SD4171$$

The minimum hold time is 0.1 times of SD4171.

For example, when the current linear speed is 1000 mm/s, the SD4750 supports a maximum linear speed

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

of 2000 mm/s, and the SD4171 sets the maximum pause time to 1000 ms. In this case, the pause will be triggered, and the robot will stop after 500 ms.

| register | type | explain | unit | read-write | Trigger time |
|---|---|---|---|---|---|
| SD4171 | INT16U | variable cycle deceleration time | Default: 1000ms | R/W | Modify before triggering RBSTOP instruction |

- S0 — Stop mode: 0 indicates deceleration stop, -1 indicates emergency stop, and values greater than 0 indicate stop for a specified duration.
- The RBSTOP instruction supports the RBPTP/RBLINE/RBCIRCLE/RCPATH motion commands.

## 12.6 Continue Movement RBGOON Command

### 12.6.1 Overview of the Instructions

After the robot executes the RBSTOP(0.0f) instruction to stop, it will continue to execute the unfinished motion from the stop.

| Restore Movement RBGOON |
|---|
| Operation: None |

### 12.6.2 Command Application


- When the M0 rising edge occurs, continue the previous motion from the stop. If the robot is in motion, it is ignored.
- The RBPTP/RBLINE/RBCIRCLE/RCPATH instructions support the RBGOON instruction.
- After executing the RBSTOP pause, the SM3166 register is set to ON, enabling RBGOON execution. Upon RBGOON execution, the modified register is reset.

## 12.7 RCPATH instruction

### 12.7.1 Overview of the Instructions

Forward-looking motion is primarily designed for continuous Cartesian line segment trajectories, employing spline transitions between Cartesian trajectories to eliminate the uncertainties of CP transitions. This ensures trajectory precision, maintains speed stability, and guarantees smoothness. Users must submit all path points in a single batch, after which the controller will group the points forward-looking to achieve the preset speed control.

The non-forward-looking motion command function is designed to differentiate from multi-segment trajectory continuous forward-looking motion. It is applicable in scenarios requiring no forward-looking processing or spline transitions, while maintaining compatibility with the RB motion command specification. This function enables single-segment motion command issuance and CP time transition processing between

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

adjacent commands. It achieves independent path and attitude planning for Cartesian trajectories, effectively preventing 10104 alarms.

Move to the target point by following the preset path from the current point, supporting multiple points. Before triggering the RCPATH command, configure the initial address of the RCPATH point.

| Address | Type | Explain | Unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD5500 | INT16U | Storage address type for processed data<br>0：D<br>1：Hd<br>If sd5500 equals 1 and sd5588 equals 1000, hd1000 is designated as the starting address for storing dynamic point data. | | R/W | Next time the location is triggered |
| SD5588 | INT32U | Processing station storage address | | R/W | Next time the location is triggered |
| SD5400 | INT16U | Number of points issued each time<br>Sd5400 <= sd5403. If sd5400> sd5403, the out-of-range data points cannot be read and will be automatically discarded. | | R/W | Next time the location is triggered |
| SD5401 | INT16U | The number of actual points issued each time<br>The actual number of distribution points after removing queue overflow points | | R | |
| SD5402 | INT16U | Maximum value of instruction point buffer space<br>Maximum number of points a user can send at once | Default 200 | R | |
| SD5403 | INT16U | Available space size, number of space pointers<br>The maximum number of points a user can send at a time, SD5403 <=SD5402 | | R | |

## 12.7.2 RCPATH instruction data point format

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-d12 | Target point coordinates [x, y, z, q0, q1, q2, q3]⑨ | Float |
| D14 | Arc angle, range [0-360]*100 | INT16U |
| D15 | Tool id, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-200 to indicate the percentage of cp time, and set 0 or-1 to indicate the zero-speed point | INT16S |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and posture ③ | FLOAT |
| D22 | Maximum acceleration percentage of running path and posture | FLOAT |
| D24 | Maximum deceleration percentage of running path and posture | FLOAT |
| D26 | Path transition error percentage | FLOAT |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: Ptp, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D29 | 0: Ptp with joint speed planning (unfixed trajectory). 1: Ptp with joint speed planning (fixed trajectory). 2: Ptp with cartesian speed planning (fixed trajectory). 3: Ptp with acceleration optimization planning (fixed trajectory) ⑥ | INT16U |
| D30 | Cf1, joint 1 attribute configuration. For specific definitions, refer to the cf quadrant division. | INT16S |
| D31 | Cf4, joint 4 attribute configuration. For specific definitions, refer to the cf quadrant division. | INT16S |
| D32 | Cf6, joint 6 attribute configuration. For specific definitions, refer to the cf quadrant division. | INT16S |
| D33 | Cfx, value 0-7 | INT16U |
| D34 | External axis motion type: 0: The external axis does not move. 1: The external axis moves in coordination with the robot. 2: The external axis moves independently while the robot remains stationary. 3: The external axis moves independently while the robot does not. 5: The external axis and the robot move simultaneously but asynchronously. 6: The external axis and the robot move simultaneously and synchronously but not in coordination. | INT16U |
| D35 | External axis transition parameter, value 0-100⑤ | INT16U |
| D36 | Cp transition based on displacement length | FLOAT |
| D38-39 | Obligate | |
| D40 | External shaft 1 angle, fixed as the positioner shaft 1 | FLOAT |
| D42 | External shaft 2 angle, fixed as the positioner shaft 2 | FLOAT |
| D44 | External shaft 3 displacement, fixed ground rail shaft | FLOAT |
| D46 | External shaft system speed percentage | FLOAT |
| D48 | External shaft system acceleration percentage | FLOAT |
| D50 | Whether the arc follows the pose: 0-no, 1-yes | INT16U |
| D51 | Swing mode: 0-no swing, 1-z-shaped swing, 2-v-shaped swing, 3-triangle swing, 4-circular swing, 5-octagonal swing | INT16U |
| D52 | Bend arc length, 1-10mm | FLOAT |
| D54 | Bend arc range: 1-10mm | FLOAT |
| D56 | Swing height, replacing the original v-type swing angle, 1-10mm | FLOAT |
| D58 | Left retention length: 1-10mm | FLOAT |
| D60 | Retention time, 1-10mm | FLOAT |
| D62 | Left retention length: 1-10mm | FLOAT |
| D64-d67 | Obligate | |
| D68 | The connecting length at the intersection line's end point is less than one-fourth of the arc length, measured in millimeters. | FLOAT |
| D70 | Sense function corresponds to m register address offset | INT32U |
| D72 | The till function corresponds to the m register address offset. | INT32U |
| D74 | Find function corresponds to m register address offset | INT32U |
| D76 | Jump instruction trajectory type: 0: No jump, 1: Three-stage ptp combination, 2: Three-stage line combination, 3: Three-stage line+ptp+line combination | INT16U |
| D77 | The jump instruction performs a carry-forward (cp) transition with the next instruction. 0: No transition, 1: Transition | INT16U |
| D78 | Archs: Jump instruction starts with a rising distance limit | FLOAT |
| D80 | Arche: Jump instruction ends the distance limit | FLOAT |
| D82 | Limz: Jump instruction height limit for vertical axis lifting | FLOAT |
| D84 | Point number, user-defined, monotonically increasing (for easy correlation with sd5404 or log viewing of anomalies) | INT32U |
| D86 | The number of transition points for attitude is up to 5⑦. | INT16U |
| D87 | Wait wait time, in milliseconds. Setting 0 disables the time setting and waits for the signal only. | INT16U |
| D88 | Wait for the m signal sequence number (e.g., 1000) to turn m[1000] on. A positive number indicates high-level waiting, a negative number indicates low-level waiting, and 0 means no signal effect—only time waiting (not supported in the | INT32S |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| | sd6288-compatible non-forwarding version). | |
| D90 | Load processout1 serial number, range 1 to 100 | INT16U |
| D91 | Load processout2 number, range 1 to 100 | INT16U |
| D92 | External shaft lifting height ⑩ for jump models with additional shafts | FLOAT |
| D94 | Wrist singular avoidance: 0: Singular avoidance is off. 1: Singular avoidance is on, and the singular avoidance is performed on the joint angle change range of 4/5/6 of the starting and ending points. 2: Singular avoidance is on, and the singular avoidance is performed with the wrist hand system unchanged. 3: Singular avoidance is on, and the singular avoidance is performed with the wrist hand system changed. | INT16U |
| D95 | Follow motion type: 0: No follow motion. 1: Chase motion. 2: Synchronized motion. 3: Deceleration motion | INT16U |
| D96 | The number of the material grabbed in a single chase action | INT16U |
| D97-D99 | Obligate | |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. When the external axis collaborative motion types are 1, 2, or 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to reference coordinate system D18=6, D19=1 (workpiece coordinate system). When setting D18=4 (tool coordinate system), the point position information is relative displacement. For instance: If the robot's tool-end position is (400,200,100) and the point position x, y, z information is set to (50,20,-20), the robot will reach (450,220,80) after movement.

③: When speed, acceleration, or both are not configured, the system automatically adjusts the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points: an arc auxiliary point and an arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint position is the starting point position rather than the endpoint information in the point input. Trajectory type 7 represents an incomplete intersection line, composed of the current point, auxiliary point, and endpoint. Trajectory type 8 represents a complete intersection line, consisting of the current point, auxiliary point 1, auxiliary point 2, auxiliary point 3, and the endpoint, with the requirement that the starting and endpoint poses must be identical. If the specified number of points is not met, the trajectory is forcibly converted to a straight line.

⑤: CP Time Transition Percentage parameter, indicating the percentage of remaining deceleration time for the current segment before initiating the next segment.

⑥: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: speed planning is performed in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: speed planning is performed in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values represent the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑦: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4,

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

and P5 will be proportionally linearly allocated according to the path length ratio, with the P3-P6 posture change amount being equally distributed.

⑧: The transition length of the intersecting line trajectory is determined by the arc length extending from the end point of the current trajectory, inheriting the end point's orientation. This length is defined by the transition length in the end point's position format, with auxiliary point transition length disabled. The unit is mm. A value ≤ 0 indicates no transition length, while a value exceeding 1/4 of the intersecting line's arc length triggers a range warning. This value can be approximated using the circumference calculation formula: 2πr/4.

⑨: When the intersection line has a high point or low point, the starting point, auxiliary point and ending point should be selected at the high point or low point as far as possible to obtain a better welding posture. When the intersection line is short and must pass through high and low points, or the intersection line is long and passes through many high and low points, the auxiliary point should be located at the middle position of the starting point and ending point as far as possible.

The point teaching sequence must be performed sequentially, either clockwise or counterclockwise.

⑩: When the robot is equipped with a height-adjustable external axis, this register allows configuration of the axis's elevation during JUMP trajectory transitions (similar to LimZ in JUMP trajectories). The value remains independent of JUMP group parameters, with the movement duration matching the robot's JUMP time. To enable this function, configure the external axis as a gantry type, designate it as the only controlled axis (moving axis), and set the axis mode to 6.

After completing the point and first address configuration, follow these steps to execute the RCPATH instruction:

- S1: Configure the SD5500 register type for processing point data storage.
- S2: Set the initial address SD5588 for the storage register containing machining point data.
- S3: Configure the number of SD5400 processing points for the initial distribution.
- S4: Set up the machining tool SD5503.
- S5: Forward-looking motion: All point data is sent in one batch, where SD5400 equals the number of points, and the SM3160 triggers the motion. Non-forward-looking motion: Only one point data is sent, where SD5400 equals 1 (arc = 2), and the SM3160 triggers the motion.
- S6: Set SM3016 to 1 when the trajectory reaches the end of the acceleration phase.
- S7: After detecting the SM3016 bit set during the scanning cycle, the user can utilize the cyclic point distribution function to continue writing points according to the configuration method in S6. Finally, the SM3160 bit is set to execute newly added points, with subsequent points following the S6-S8 steps in a cyclic manner.

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

## 12.7.3 Description of Registers

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD5500 | INT16U | Storage address type for processed data<br>0：D<br>1：Hd<br>If sd5500 equals 1 and sd5588 equals 1000, hd1000 is designated as the starting address for storing dynamic point data. | | R/W | The next delivery point will take effect |
| SD5588 | Int32s | Process the storage address. If a negative value is entered, it will be treated as an absolute value. | | R/W | The next delivery point will take effect |
| SD5400 | INT16U | Number of points issued each time Sd5400<=sd5403. If sd5400>sd5403, the out-of-range data points cannot be read and will be automatically discarded. | | R/W | The next delivery point will take effect |
| SD5401 | INT16U | The number of actual points issued each time<br>The actual number of distribution points after removing queue overflow points | | R | |
| SD5402 | INT16U | Maximum value of instruction point buffer space<br>Maximum number of points a user can send at once | Default 200 | R | |
| SD5403 | INT16U | Available space size, number of space pointers<br>The maximum number of points a user can send at a time, sd5403 <=sd5402 | | R | |
| SD5404 | INT16U | Current data reading processing point number<br>Monotonic increasing, user d84 settings | | R | |
| SD5450 | INT32U | The rc command specifies the total number of points, which is displayed only for reference and cannot be modified by the user. After the rc command is triggered, it sends the accumulated value to the sd5401 each time. | | R | |
| SD5452 | INT32U | Data read processing line number | | R | |
| SD5454 | INT32U | Filter data row numbers | | R | |
| SD5456 | INT32U | Forward processing line numbers | | R | |
| SD5458 | INT32U | Number of forward-looking groups | | R | |
| SD5460 | INT32U | Track processing line number | | R | |
| SD5462 | INT32U | Real-time interpolation of track line numbers | | R | |
| SD5464 | INT32U | Displays the actual interpolation line segment number (excluding filtered segments) and indicates the current movement to the nth machining point. When sd5573 arc instruction is not enabled, only the arc endpoint number is displayed. With sd5573 enabled, the auxiliary point number is shown. | | R | |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD5466 | INT16U | Number of machining points filtered out due to short paths and poses in data filtering | | R | |
| SD5468 | INT16U | Real-time interpolation skips filtered point counts, logging the execution status of filtered points during interpolation. Each filtered point increments the register by 1. The trajectory completes execution when the number of transition segments equals the count. | | R | |
| SD5503 | INT16U | Rc instruction processing point tool number<br>0~19: User-defined tool number during operation. Changing tools during motion is not supported. This register differs from the machining point coordinate system type. | | R/W | The next delivery point will take effect |
| SD5510 | Float | Path hard transition angle limit<br>Angle between line segments: The complement of the angle between two adjacent vectors. If the value exceeds this threshold, a direct transition is applied. | Default 175 degrees | R/W | The next delivery point will take effect |
| SD5512 | Float | Transition error<br>This value will somewhat limit the maximum allowable speed of the arc transition section and the previous section, to be used in conjunction with point d26. Setting it too large will cause the trajectory to deviate from the inflection point, increasing the distance and the trajectory error. Setting it too small will result in a smaller radius of the transition arc trajectory, leading to a correspondingly larger curvature, which in turn restricts the inflection point speed too much. When setting a larger transition error, due to the limitation of the trajectory length, the actual execution of the transition error may be less than the set value. Setting it to 0 will force the underlying layer to handle it as 100.0mm. | Default 100.0mm | R/W | The next delivery point will take effect |
| SD5514 | Float | Arc height error<br>A smaller setting imposes a stricter speed limit on the arc command, while a larger setting relaxes the speed limit but increases the fitting error of the arc trajectory by approximating curves with straight lines. | Default: 0.2mm | R/W | The next delivery point will take effect |
| SD5518 | Float | Direct transition error<br>For path hard transition speed limit calculation, use a straight transition. Setting it too large increases the distance from the inflection point, while setting it too small restricts the inflection point speed. | Default: 0.2mm | R/W | The next delivery point will take effect |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD5520 | Float | Minimum angle constraint for small line segments<br>Determines whether the point speed needs to be reduced to 0. If the angle between adjacent line segments is less than this value, no transition processing is performed, as an excessively small angle would result in a highly curved transition trajectory, severely limiting the speed. The system does not determine the minimum angle between a straight line and an arc. | Default: 5 degrees | R/W | The next delivery point will take effect |
| SD5522 | Float | Minimum path length limit<br>Filtering out machining points with overly short paths for robots or external axes, and working with sd5570 to exclude cases where two adjacent machining points are too close. | Default: 0.01mm or degree | R/W | The next delivery point will take effect |
| SD5524 | Float | Adjusting the approximation degree of circular arcs with three postures interpolation<br>Transition coefficient in attitude interpolation | Default 0.2 | R/W | The next delivery point will take effect |
| SD5526 | Float | Restrictions on the attitude changes of the starting and ending points of a line segment<br>Used to detect excessive changes in the relative path of attitude between adjacent segments, assessing the transformation of the relative path of attitude between adjacent trajectory segments. | Default: 100 degrees | R/W | The next delivery point will take effect |
| SD5528 | INT16U | Maximum search segment length for merged segments<br>Maximum number of segments allowed for multi-segment merging and maximum number of trajectory segments processed for merging | Default 10 | R/W | The next delivery point will take effect |
| SD5529 | INT16U | How many paragraphs before interpolation can be applied | Default 20 | R/W | The next delivery point will take effect |
| SD5532 | Float | Transition error limit of trajectory fusing path | Default: 0.1mm | R/W | The next delivery point will take effect |
| SD5534 | Float | Fusion trajectory transition angle limit | Default value: 179.908752 degrees | R/W | The next delivery point will take effect |
| SD5536 | INT16U | Forward window size<br>Number of trajectory segments processed in advance per plc scan cycle | Default 10 | R/W | The next delivery point will take effect |
| SD5537 | INT16U | Forward window step size<br>The difference of the forward processing track head pointer between two adjacent plc scan cycles | Default 5 | R/W | The next delivery point will take effect |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD5538 | INT16U | Enable initial merge<br>0: Close<br>1: Turn on | Default 0 | R/W | The next delivery point will take effect |
| SD5539 | INT16U | Enable termination merge<br>0: Close<br>1: Turn on | Default 0 | R/W | The next delivery point will take effect |
| SD5540 | Float | Dynamic adjustment, real-time speed ratio, minimum 0.01 | Default: 100% | R/W | The next delivery point will take effect |
| SD5542 | Float | Maximum radius of transition of trajectory attitude | Default: 100mm | R/W | The next delivery point will take effect |
| SD5544 | Float | Centripetal acceleration<br>The value is determined by the actual machine rigidity and servo performance. If there is an impact at the inflection point, this value can be reduced to 3000-500 until no abnormal noise is caused by excessive impact. | Default 4000 mm/s | R/W | The next delivery point will take effect |
| SD5546 | Float | Interpolation time | Default: 0.1s | R/W | The next delivery point will take effect |
| SD5548+ 2*I (WHERE I RANGES FROM 0 TO 1, 2,..., THE MAXIMUM NUMBER OF AXES MINUS 1) | Float | Given joint angle at the end of pause | Linear measure | R | |
| SD5567 | INT16U | Display the current interpolation trajectory type<br>1: Straight line<br>2: Circular arc<br>3： Ptp<br>4: Incomplete intersection line<br>5: Spiral<br>6: Complete intersection line<br>20: Single attitude trajectory<br>21: Robot follows external axis<br>22: B-spline trajectory | | R | |
| SD5568 | INT16U | Rc command execution hand system | | R | |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD5570 | Float | Non-forward-looking attitude trajectory length judgment limit | Default: 0.0001rad | R/W | The next delivery point will take effect |
| SD5572 | INT16U | Number of data points read each time | Default 25 | R/W | The next delivery point will take effect |
| SD5573 | INT16U | Consider arc auxiliary point attitude<br>0: Not considered<br>1: Consider | Default 0 | R/W | The next delivery point will take effect |
| SD5574 | INT16U | The welding torch posture is calculated according to the set welding torch angle | | R/W | The next delivery point will take effect |
| SD5575 | INT16U | Maximum time for dynamic speed control, in milliseconds<br>Change the running time of the shifting process after sd5540 | Default: 50ms | R/W | The next delivery point will take effect |
| SD5576 | Float | The transition radius of the trajectory fusion path is set to the percentage of the smaller value between the lengths of the adjacent two trajectory segments.<br>Use with sd5532 | Default: 20% | R/W | The next delivery point will take effect |
| SD5578 | INT16U | Time to complete dynamic speed adjustment at the rc starting point, in milliseconds | Default: 50ms | R/W | The next delivery point will take effect |
| SD5579 | INT16U | The tool attitude in the mode 1 of the outer shaft motion is kept unchanged in the world coordinate system<br>0: Do not enable<br>1: Turn on | | R/W | The next delivery point will take effect |
| SD5597 | INT16U | External axis forward processing control word<br>0: Conventional cp transition processing<br>1: Forward processing<br>When processing external axes, the cp parameter is invalid. Setting the external axis speed appropriately matches the robot's collaborative motion processing speed. Setting the external axis speed too low may cause overspeed or excessive acceleration after synchronization. Setting the external axis speed too high may result in prolonged zero-speed movement or extended deceleration time after synchronization. | | R/W | The next delivery point will take effect |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD6260 | Float | Minimum length limit for straight or circular arc trajectory without monotonic speed planning<br>When the straight-line length after transition exceeds this value, the segment can execute acceleration-deceleration planning. Regular trajectories can only accelerate or decelerate monotonically to avoid frequent acceleration-deceleration in short segments. | Default: 10mm | R/W | The next delivery point will take effect |
| SD6960+2I (I=0,1,2... MAXIMUM NUMBER OF AXES MINUS 1) | Float | Offset compensation for machining point coordinates, used to compensate for offset values of machining point coordinates or joint angles. The default attitude compensation is in the form of euler angles. | Degree or mm | R/W | The next delivery point will take effect |
| SD6972 | INT16U | Compensation type | 0: Pose compensation<br>5: Joint compensation | R/W | The next delivery point will take effect |
| SD6974 | INT16U | Coordinate offset compensation function switch | 0: Turn off compensation<br>Enable compensation | R/W | The next delivery point will take effect |
| SD4130 | INT16U | Wrong | | R/W | With immediate effect |
| SD4131 | INT16U | Warn | | R/W | With immediate effect |
| SD4171 | INT16U | Maximum deceleration time of variable cycle | Default: 1000ms | R/W | With immediate effect |
| SFD5040+2I (I=0,1,2... MAXIMUM NUMBER OF AXES MINUS 1) | Float | Maximum speed of each joint (joint movement)<br>The motor-side sfd3018+60*i determines the maximum motor speed, which is then calculated for the joint output side. | M/s or mm/s | R/W | Stop plc re-run |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SFD5060 +2I (I=0,1,2... MAXIMUM NUMBER OF AXES MINUS 1) | Float | Maximum acceleration of each joint<br>The default setting is 5-10 times joint speed | $M/s^2$ or $mm/s^2$ | R/W | Stop plc re-run |
| SFD5250 +2I (I=0,1,2... MAXIMUM NUMBER OF AXES MINUS 1) | Float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $M/s^3$ or $mm/s^3$ | R/W | Stop plc re-run |
| SFD5080 | Float | Maximum speed at the end, [linear velocity]<br>The linear velocity is determined by joint 1,2,3 | Mm/s perhaps Linear measure /s | R/W | Stop plc re-run |
| SFD5090 | Float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear speed | $Mm/s^2$ perhaps Linear measure $/s^2$ | R/W | Stop plc re-run |
| SFD5270 | Float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear acceleration, with specific values depending on the actual operating conditions of the device. | $M/s^3$ or $mm/s^3$ | R/W | Stop plc re-run |
| SFD6030 | Float | Maximum speed of quaternion terminal attitude | Default: 100 degrees/s | R/W | Stop plc re-run |
| SFD6032 | Float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 degrees/$s^2$ | R/W | Stop plc re-run |
| SFD6034 | Float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop plc re-run |
| SM3160 | Bit | Rc command trigger control bit | | R/W | |
| SM3161 | Bit | Rc instruction execution status bit | | R | |
| SM3162 | Bit | The data read module is in the execution state bit | | R | |
| SM3163 | Bit | The data filter module is in execution status | | R | |
| SM3164 | Bit | The foreground processing module is executing the status bit | | R | |
| SM3165 | Bit | Rc instruction interpolation is in execution status | | R | |
| SM3166 | Bit | Variable cycle slow stop state | | R | |
| SM3167 | Bit | Rc command error | | R | |

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SM3168 | Bit | Dynamic adjustment module is executing the status bit | | R | |
| SM3001 | Bit | The robot is moving. | | R | |
| SM3009 | Bit | Command given end flag | | R | |
| SM3012 | Bit | Position track transition sign<br>1: Enter transition<br>0: End transition | | R | |
| SM3014 | Bit | Enable cp transition states. Set on when the robot is stationary and off when rc commands begin interpolation. Set on when the current interpolation segment reaches the cp transition point or when the current segment is completed, and set off when interpolation begins for the next segment. | | R | |
| SM3016 | Bit | Determine the end flag of the acceleration phase | 1: Start the constant speed section.<br>0: Non-uniform section | R | |
| SM3017 | Bit | Check for filter flag | 1: Filter points exist.<br>0: Filter points do not exist. | R | |

### 12.7.4 External functional points and their performance

1) Movement commands

Overview: During RC instruction execution, RBSTOP can be executed, but point-stop and similar instructions cannot be executed in sequence.

Processing: During RC execution, a stoppage alarm is triggered without interrupting the RC command.

2) Small line segment filtering

Overview: When the path length and attitude trajectory length of a line segment between two adjacent processing points both fall below the preset threshold, the segment is deemed too short and discarded, with the next point read directly. If only the path length is insufficient, the segment remains valid as a single-attitude trajectory, with both start and end velocities set to 0.

3) Dynamic compensation

Overview: When executing RC commands, configuration parameters enable bit-pose or joint compensation for machining points. The compensation format is [x, y, z, Rz, Ry, Rx] or [J1, J2, J3, J4, J5, J6], with Euler angles as the default compensation method for attitude.

Procedure: Activate the Cartesian and Joint Offset function switches on SD6974, configure the SD6972 selection compensation type, and set the offset compensation value for SD6960+2(i).

4) Dynamically add points

Overview: Add distribution points in real time.

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

Operation: The SM3160 acts as a dynamic adjustment control word for point positions. After the bottom-level scan reaches the rising edge, it compares the number of points stored in the register start addresses of SD5500 and SD5588 (SD5400 and SD5403) to obtain the actual number of points that can be issued (SD5401). Based on the tail pointer of the buffer ring queue, it sequentially copies the point data into the buffer area. The data reading module is then activated, preparing to initiate a new round of point data reading operations from the start address set by the host and slave computers.

Three processing results for new points:

Add point operations are valid. The last segment can be modified, and the new first segment transitions with the original last segment. Add point operations are valid. The last segment cannot be modified, and the new first segment does not transition with the original last segment. Interpolation has ended. Add point operations are invalid.

### 5) Zero-speed machining point setting

Overview: Set the speed to 0 at a specific processing point, and set two consecutive processing points as zero-speed points. Both the start and end speeds of this segment are 0, and the system will execute an acceleration-uniform-deceleration planning.

### 6) Set different coordinate systems for points

Overview: You can set different tool coordinate systems and point data reference coordinate systems for machining points, including base coordinate system, joint coordinate system, and workpiece coordinate system. Note: The point information input in the tool coordinate system is a change quantity, not an absolute pose.

### 7) Trajectory type

Overview: The motion trajectory includes straight lines, arcs, and PTP (Point-to-Point) paths, considering both coplanar and non-coplanar scenarios. PTP trajectories can only execute zero-speed start/stop motions but allow multiple PTP transitions to achieve smooth transitions.

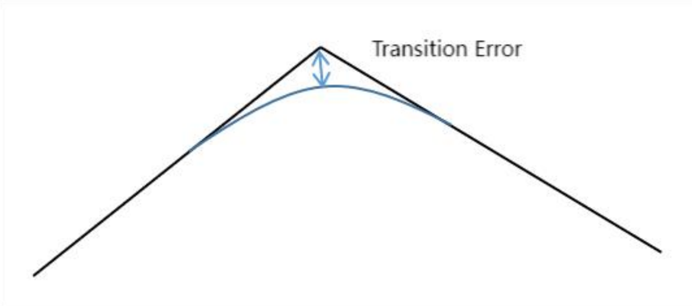
### 8) Spline transition processing

Path Transition: The forward processing method employs spline curves to handle transitions between consecutive Cartesian trajectory segments, with the curvature radius of the transition spline influenced by both transition error and segment length. This approach is only applicable when multiple points are issued in a single operation (forward mode). JUMP2, however, can utilize this method as it internally processes segments as three straight lines.

Attitude transition: spline curve transition.

When both adjacent trajectory types are Line or Circle, the transition radius (D17) in point data serves as a switch: values greater than 0 permit transitions. Key transition parameters include transition error (D26: path transition error. D86: number of attitude redistribution points). A higher D26 value enhances transition visibility and increases deviation from original points. D86 controls when attitude changes begin (advanced or delayed by specified points) and when changes conclude (delayed arrival). Attitude redistribution softens abrupt changes, reducing joint overspeeding caused by sudden shifts.

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot




9) Merge start and end segments

Overview: Multiple small segments with stepwise acceleration or deceleration. When this feature is enabled, multiple segments can be merged into one. The pose inherits the original data from multiple segments, and the speed can be accelerated, constant, or decelerated uniformly.

10) Dynamic speed control

Overview: During operation, the terminal speed is adjusted in real time by regulating the speed ratio SD5540float, with the required time for speed change as follows:

$$Time = \frac{\text{Speed Regulation Ratio Difference}}{100} * SD5575$$

$$（Pre - andPost - adjustmentSpeedRegulationRatioDifference: SD5540）$$

Speed adjustment is not supported during the pause, speed regulation, or trajectory recovery processes. During speed regulation, RBSTOP pause is supported, after which RBGOON interpolation restores the speed ratio to the last set value. Speed adjustment ranges from 0% to 100%, with a minimum duration of 0.1 SD5575.

11) Resume

Overview: The RBSTOP command can pause the system and resume the RC instruction from the last paused state. The recovery must return to the paused position, with the following time:

$$Time = \frac{\text{Speed Regulation Ratio Difference}}{100} * \text{Soft Stop Time+Soft Stop Time}$$

The minimum recovery time was 0.1 times SD4171.

For example, with a 1000ms dwell time, when the trajectory stops, the SD5540 is adjusted from 100 to 50.

The trajectory recovery time is calculated as: (50-100)/100 × 1000 + 1000 = 500ms.

Operation: RBSTOP initiates variable-period deceleration stop, followed by RBGOON command execution to resume motion. Both deceleration and recovery phases employ parabolic curves for interpolation time adjustment. The joint angle deviation between the end of the deceleration phase and the start of recovery must not exceed 0.001 degrees.

12) Arc auxiliary point attitude

Overview: Whether the preset attitude has reached the arc auxiliary point.

Operation: To set the attitude at the arc auxiliary point, activate the SD5573 function and split the arc command into two arcs for planning.

13) CP transition processing

Overview: CP transition is simply understood as the superposition of two trajectories with zero initial and final velocities and accelerations, ensuring continuous velocity and acceleration. When Trajectory 1 and

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

Trajectory 2 undergo CP transition, the transition parameters are determined by D17 or D35 set at the endpoint of Trajectory 1. When the CP time transition percentage is set as k (in%), where k<100 indicates the percentage of deceleration time, and k>100 indicates the percentage of the half-time uniform motion period for the excess duration. This means Trajectory 2 begins executing immediately after Trajectory 1 enters deceleration, forming a superimposed trajectory. Note that t1 equals Trajectory 1's half-uniform motion time multiplied by (k-100)/100 plus Trajectory 2's deceleration time multiplied by k (maximum 100)/100, while t2 equals Trajectory 2's acceleration time plus half-uniform motion time. If the calculated t1 exceeds t2 based on CP parameters, the transition time t1 must be adjusted to the smaller value t3 of the two. This may cause discrepancies between the CP settings for the next trajectory and actual conditions.

For the PTP trajectory: t1 is calculated by adding the acceleration phase (half of the total duration) and the deceleration phase to the deceleration phase of all joint motions, then taking the minimum value. t2 is determined by adding the acceleration phase to the deceleration phase of all joint motions, then taking the minimum value. For the LINE trajectory: t1 is calculated by adding the deceleration phase to the deceleration phase of all joint motions, then taking the minimum value. t2 is determined by adding the acceleration phase to the deceleration phase of all joint motions, then adding the deceleration phase to the acceleration phase, and taking the minimum value.

When t1 reaches its maximum value, some trajectories may begin to overlap with the next segment during the uniform speed phase. Similarly, in t2, certain trajectories might exhibit a phase where their velocity first increases and then decreases before the CP overlap concludes.

| D17(INT16S) | Definition |
| --- | --- |
| <=100 | Percentage of deceleration time |
| >100 | The percentage of time spent in the constant speed segment for the portion exceeding 100 |

Any Trajectory CP Time Transition Settings

The RC command parameters specify: D17 represents the CP transition time percentage, D26 indicates the spline path transition error percentage, and D36 controls distance-based CP transition. When both adjacent trajectories are Cartesian motions, set D26=0 and use D17 for CP transition. If both trajectories are Cartesian motions, set D26=0 with D36>0 to enable distance-based transition (D36 currently unsupported for JUMP and external axis transitions). The PTP (Point-to-Point) CP transition specification remains unchanged for any trajectory. Exceeding D17=100 or setting D36 too high may cause overspeeding.

| D17(INT16S) | D26(float) | D36(float) | Transitional form |
| --- | --- | --- | --- |
| -1 or 0 | arbitrary value | 0 | The target point deceleration is 0, indicating it is a passing point. |
| >0 | 0 | 0 | CP time transition serves as the transition point. CP transitions can be performed between any trajectories. |
| >0 | Not equal to 0 | 0 | The spline path transitions at transition points. if a point does not support spline transitions, no transition occurs. The PTP trajectory will clear the D26 value to 0 internally. |
| >0 | 0 | >0 | CP distance transition is the transition point. Cartesian trajectory is applicable to distance transition. if the trajectory is PTP, it degenerates to ordinary CP time transition. |
| >0 | Not equal to 0 | >0 | The spline path transitions at transition points. if a point does not support spline transitions, no transition occurs. The PTP trajectory will clear the D26 value to 0 internally. |

14) The posture is not forward-looking.

Overview: The non-forward-looking attitude function can effectively resolve the 10104 error code issue

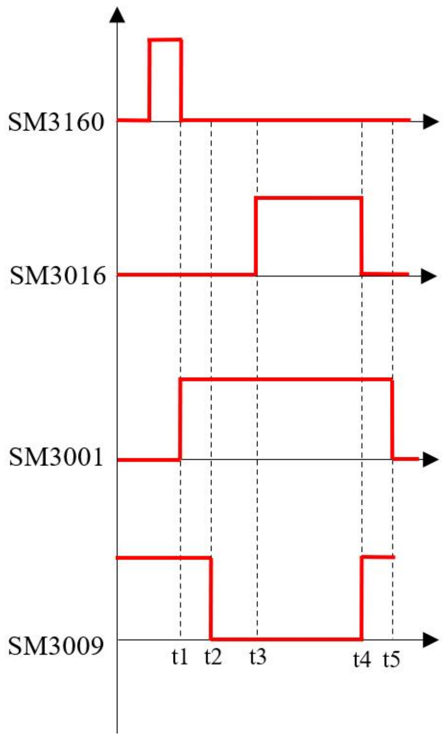
User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

when the attitude transitions from A to B and back to A. However, using a global non-forward-looking control word may cause oscillations during continuous short segments with attitude changes.

Operation: When a non-forward-looking segment occurs, synchronize the time of the preceding and following segments, and plan and interpolate the non-forward-looking segment separately.

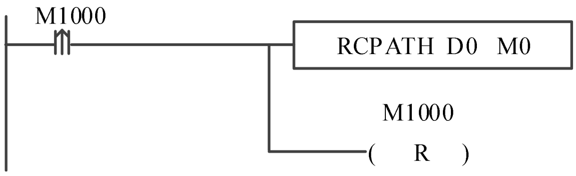
15) Instruction timing

Overview: RC commands typically utilize status bits to determine the current execution state, including SM3016 (for point judgment), SM3009 (for interpolation check), and SM3001 (for motion detection). The following shows the timing diagram of these registers. When the SM3160 signal is set to ON (t1), point analysis begins while SM3160 is turned off and SM3001 is activated. Upon completing analysis (t2), point interpolation starts with SM3009 deactivated. During the uniform speed phase (t3), SM3016 signals turn on to determine the next trajectory segment. When interpolation for the current segment ends (t4), SM3016 deactivates while SM3009 reactivates after completing remaining interpolation points in the queue, with the interval duration determined by the interpolation queue length. When the final interpolation point is delivered (t5), a time delay occurs between SM3001 and SM3009 due to servo mechanism transmission and positioning confirmation, influenced by positioning width and servo communication speed.




In the forward mode, each forward segment triggers state changes for SM3001 and SM3009. In the non-forward mode, these states only change at the final segment. For this mode, the increment operation must be triggered by SM3160 when SM3016 is at the rising edge (at t3).

## 12.7.5 Command Application



User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

The source file sample program is as follows:

```
#include"RCPATH.h"

#define pNum 4
RcPointInformation RcPoint[pNum] = {
   {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
   {{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0},
   {{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,,0,0,0,0,2,0,0,0,0,0},
   {{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0},
}.

void RCPATH(PINT16S W, BIT B)
{
   if (SD[5500] == 0)
   {
      for (int i = 0. i < SD[5401]. i++)
      {
// Transmit pose data
         *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
         *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
         *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
         *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
         *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
         *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q2.
         *(float*)&D[(SD[5588] + 100 * i + 14)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType.//hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
      }
   }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
```

The header file sample program is as follows:

```
#ifndef _RCPATH_H_
#define _RCPATH_H_

```

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

```
#define ExtAxisJointNum 3

typedef struct
{
    double x.
    double y.
    double z.
}rbStruct_pos.

typedef struct
{
    double q0.
    double q1.
    double q2.
    double q3.
}rbStruct_orient.

typedef struct
{
    rbStruct_pos pos.
    rbStruct_orient orient.
}CoordPos.

typedef struct
{
CoordinatesPos targetPos. //The target's pose (joint space coordinates)
float weaveAngleZ. // Angle of Z-shaped pendulum swing
INT16U toolIndex. // Tool number
INT16U posType. // Hand type
INT16S transR. // Attitude transition radius
INT16U coordType. //Reference coordinate system
INT16U weaveMode. //Arc swing mode
float targetSpeed. //Percentage of processing point speed
float targetAcc. // Processing point acceleration percentage
float targetDec. // Percentage of deceleration at machining point
float TransErr. // Path transition error
INT16U trajType. //Track type
INT16U ptpPlanType. //PTP speed planning mode
INT16S cf1. //Configuration parameters for Robot Joint 1
INT16S cf4. //Configuration parameters for Robot Joint 4
INT16S cf6. //Configuration parameters for Robot Joint 6
INT16U cfx. //Robot's singular correlation configuration parameters
INT16U ExtAxisMoveType. //External axis movement type
INT16U ExtAxisTranR. //External axis transition radius
float ExtAxisJointTheta[ExtAxisJointNum]. //External axis given angle
float ExtAxisSpeed. //External axis speed percentage
```

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

```
float ExtAxisAcc. //External axis acceleration percentage
INT16U weaveFollow. // Plane-following attitude
float weaveLength. // Waving arc length
float weaveWidth. // swing amplitude
float weaveHeight. // Swing arc height
float weaveLdWell. // Left dwell length
float weaveCdWell. // Length of stay in the middle
float weaveRdWell. // Right dwell length
INT32U SenseAddr. // Address of the M register for the SENSE function
INT32U TillAddr. //TILL function M register address
INT32U FindAddr. //FIND function M register address
INT16U JumpType. //Jump type
INT16U CPEnable. // Check if the current jump is the next CP
float ArchS. // Rise distance
float ArchE. // Distance to descend
float LimZ. // Height of lifting the upper and lower axes
INT32U pIndex. //Point index number
INT16U OrientInfluenceNum. // Number of transition influence points
INT16U waitTime. //Wait time
INT32S waitOrder. //Wait for the M signal sequence number
INT16U processOut1Num. // Load the processOut1 sequence number
INT16U processOut2Num. // Load the processOut2 sequence number
}RcPointInformation.

#endif
```

The code above specifies that point information can be written into registers designated by SD5500 and SD5588, starting from the fixed register D1000. Alternatively, values may be directly written into the designated registers, for example:

```
void RC(PINT16S W,BIT B)
{
   #define SysRegAddr_HD_D_HM_M_SM
   static int point_num = 0.
   unsigned char i .
   static unsigned char num = 0.
   SD[5500] = 0.
   SD[5588] = 1000.
   for(i=0. i<SD[5401].i++)
   {
* (float *) &D[1000 + 100 * i] = rand()% 20. //Randomly assign angle to joint 1
* (float *) &D[1002 + 100 * i] = rand()% 20. // Randomly assign angle to joint 2
* (float *) &D[1004 + 100 * i] = rand()% 20. // Randomly assign angle to joint 3
* (float *) &D[1006 + 100 * i] = rand()% 20. //Randomly assign angle to joint 4
* (float *) &D[1008 + 100 * i] = rand()% 10. // Randomly assign angle to joint 5
* (float *) &D[1010 + 100 * i] = rand()% 20. // Randomly assign angle to joint 6
D[1017 + 100 * i] = 10. //Transition radius
```

User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

```
D[1018 + 100 * i] = 5. //Reference coordinate system
* (float *) &D[1020] = 25. // Path and attitude speed percentages
* (float *) &D[1022] = 50. // Path and attitude acceleration percentage
* (float *) &D[1024] = 50. // Path and attitude deceleration percentage
* (float *) &D[1026 + 100 * i] = 10. // Path transition error
D[1028 + 100 * i] = 3. // Trajectory type
    }
}
```

After executing the function, the corresponding register is written with the bit information.

| Register | Type | Numeric value | Explain |
|---|---|---|---|
| Sd5500 | INT16U | 0 | Process the register type. Set to 0 to use the d register. |
| Sd5588 | INT32U | 1000 | Set the first address of the machining point register to 1000, indicating that d1000 is the machining point address. |
| D1000 | Float | 13.0 | Process the point pose parameters: Position and quaternion attitude in non-joint coordinate system, and joint angle values in joint coordinate system. |
| D1002 | Float | 3.0 | |
| D1004 | Float | 2.0 | |
| D1006 | Float | 9.0 | |
| D1008 | Float | 0.0 | |
| D1010 | Float | 10.0 | |
| D1012 | Float | 0.0 | |
| D1017 | INT16U | 10 | Transitional radius of attitude of machining point |
| D1018 | INT16U | 5 | Set the processing point coordinate system type to 5 for joint coordinates |
| D1020 | Float | 25.0 | Set the maximum operating speed for the processing point |
| D1022 | Float | 50.0 | Maximum operating acceleration for processing points |
| D1024 | Float | 50.0 | Maximum deceleration rate for processing points |
| D1026 | Float | 10.0 | Processing path transition error |
| D1028 | Float | 3.0 | Process point trajectory type. Set to 3 for ptp motion. |

This point indicates that the robot will perform a PTP motion (D28=3) in the joint coordinate system (D18=5), with a speed of 25 (D20=25), acceleration of 50 (D22=50), and transition radius of 10 (D17=10).

If you want to maintain the speed of the previous point, you do not need to set a speed value for subsequent points.

For example, when moving three points, the first point moves at 25% speed with 50% acceleration/deceleration, while the other two points use the speed data of the first point. The speed settings are as follows:

| Register | Type | Numeric value | Explain |
|---|---|---|---|
| D1020 | Float | 25.0 | Set the maximum operating speed for the processing point |
| D1022 | Float | 50.0 | Maximum operating acceleration for processing points |
| D1024 | Float | 50.0 | Maximum deceleration rate for processing points |
| D1120 | Float | 0.0 | Set the maximum operating speed for the processing point |
| D1122 | Float | 0.0 | Maximum operating acceleration for processing points |
| D1124 | Float | 0.0 | Maximum deceleration during processing |
| D1220 | Float | 0.0 | Set the maximum operating speed for the processing point |
| D1222 | Float | 0.0 | Maximum operating acceleration for processing points |
| D1224 | Float | 0.0 | Maximum deceleration rate for processing points |

Functionality for non-forward-point issuance: Upon completing each acceleration phase of a robot's trajectory segment, the system generates an SM3016 signal to initiate the next point's issuance, ensuring seamless transition to the current position. After finishing a point interpolation, the SM3016 resets. When the user detects the SM3016's rising edge, the system writes the new point through SM3160's set function for

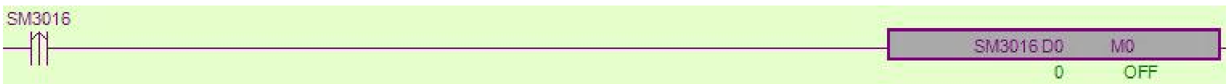
User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

point expansion. Each SM3160 trigger consistently sets the new point's initial address to SD5500/SD5588, maintaining this configuration regardless of trigger frequency.

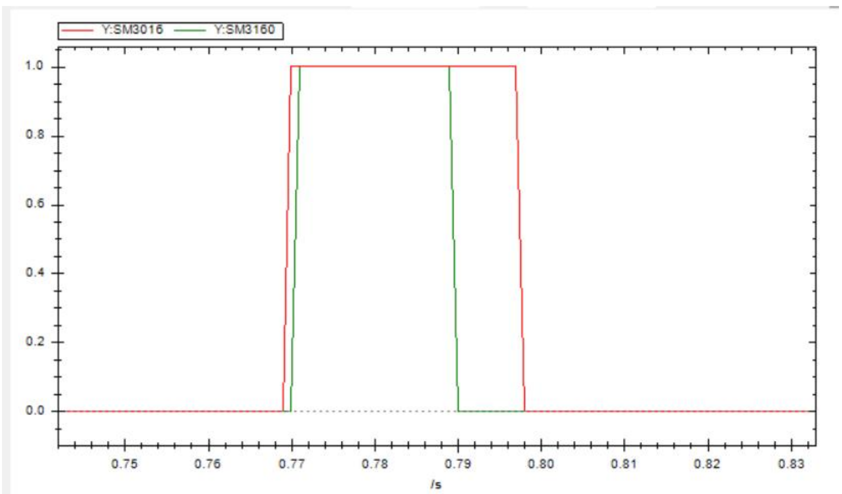
For example, the C function block function is as follows:

```
void SM3016(PINT16S W,BIT B)
{
  #define SysRegAddr_HD_D_HM_M_SD_SM
  SD[5500] = 0.
  SD[5588] = 1000.
  INT16S i .
  static int point_num = 0.

  if (HD[1028 + D[200] * 100] == 1 || HD[1028 + D[200] * 100] == 3 )
  {
    SD[5400] = 1.
    memcpy(D+1000,HD+(1000+100*D[200]),100*2).
    D[200] = D[200] + 1.
  }
  else if ( HD[1028 + D[200] * 100] == 2 )
  {
    SD[5400] = 2.
    memcpy(D+1000,HD+(1000+100*D[200]),100*4).
    D[200] = D[200] + 2.
  }
   SM[3160] = 1.
}
```

D[200] is a register that counts the number of recorded points, with the initial address set to D1000. Point data is stored in HD1000 and subsequent registers. When this function is triggered, the point data from HD1000 is copied to D1000, after which HD1000 advances by 100 words to prepare for copying new point data. The PLC invocation method is as follows:




The images of SM3016 and SM3160 are captured as follows:



User Manual of RC6F Series Robot Controller 12 Basic Movement Command of Robot

When the PLC detects the rising edge of SM3016, it executes the C function block to configure auxiliary point data and activates SM3160 for motion initiation. The SM3001 determines the 0.77-0.79 wait period. After SM3001 turns off, the system parses commands during the 0.79-0.8 duration. SM3016 remains deactivated until the motion begins.

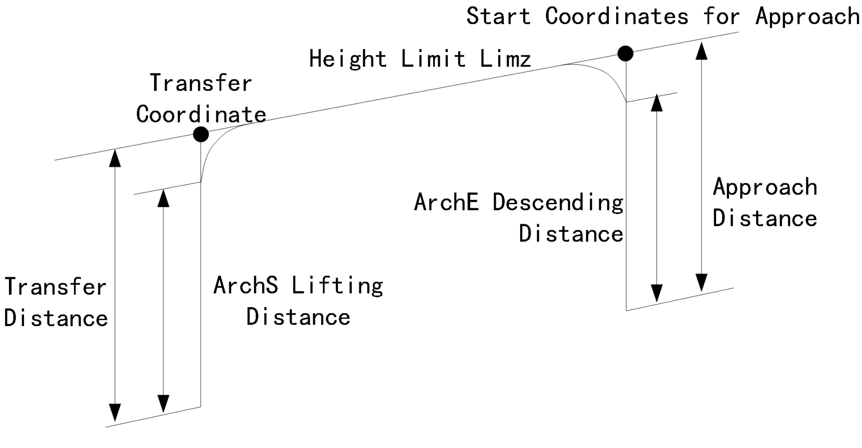
# 13. Jump function

## 13.1 Function Overview

The JUMP function is developed based on the RCPATH command, adding relevant point attributes. This command moves the robotic arm from its current position to the target coordinates via an "arch motion." The principle involves breaking a single Jump into three sequential actions: the transfer phase, the crossing phase, and the approach phase.

- Transfer process: The end of the robotic arm moves from the current point along the Z-axis to the transfer coordinate point.
- During the crossing process, the end of the robotic arm moves horizontally from the transfer coordinate point to the approach coordinate point.
- Approach process: The end of the robotic arm moves from the approach coordinate point to the target point along the Z-axis.
- Archs Rise Distance: The vertical movement height from the starting position, ensuring the minimum vertical movement distance from the starting point (unit: mm).
- ArchE descent distance: the minimum vertical movement height from the endpoint to the finish line (unit: mm).
- LimZ distance: The height limit that determines the Z-axis coordinate of the midpoint (the Z-coordinate of the robot end in the base coordinate system, in mm).

Note: When the D76-bit JUMP path is set, the D28 will no longer function. If the current JUMP path is parsed and identified as a regular path, it will execute according to the corresponding path type in the JUMP mode.




## 13.2 JUMP Classification

The jump instruction comprises three sequential movement phases, each executable through distinct trajectory combinations. Currently, the jump instruction features three distinct types:

### 13.2.1 Three-stage PTP combined motion

The three-stage process of gate motion is all of PTP trajectory type, the start and end velocity of single PTP trajectory is 0, the transition between three stages is handled by CP, the CP transition time is determined by the sum of the acceleration time of the middle stage and half of the uniform time, in order to ensure that the vertical distance of the start and end stage does not participate in the transition, the calculation of CP transition time needs to increase the constraint of the rising distance and the falling distance.

The transition between the initial and the middle segment of the JUMP is determined by the time and the distance of the middle segment, and the transition is only started when the time and the distance of the middle segment are satisfied.

### 13.2.2 Three-stage LINE Combination Movement

The three stages of gate motion are all LINE trajectory type, and there are two ways to transition between the trajectories.

Method 1: All LINE trajectory segments start and end at 0 speed, with CP mode for transitions as described above.

Method 2: Interpolating adjacent LINEs with spline transitions, where the spline transition error is adjusted according to the ascending or descending distance. This method ensures uniform transition of the linear speed across most trajectories.

### 13.2.3 Three-stage LINE+PTP+LINE combined motion

As with the three-stage PTP motion, each individual segment starts and ends at zero velocity. Refer to the chapter on three-stage PTP combined motion.

## 13.3 RCPATH instruction data point format

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-d12 | Target point coordinates [x, y, z, q0, q1, q2, q3] | Float |
| D15 | Tool id, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-200 to indicate the percentage of cp time, and set 0 or-1 to indicate the zero-speed point | Int16s |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and posture ③ | Float |
| D22 | Maximum acceleration percentage of running path and posture | Float |
| D24 | Maximum deceleration percentage of running path and posture | Float |
| D26 | Path transition error percentage | Float |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: Ptp, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: Ptp with joint speed planning (unfixed trajectory). 1: Ptp with joint speed planning (fixed trajectory). 2: Ptp with cartesian speed planning (fixed trajectory). 3: Ptp with acceleration optimization planning (fixed trajectory) ⑤ | INT16U |

| | | |
|---|---|---|
| D76 | Jump instruction trajectory type: 0: No jump, 1: Three-stage ptp combination, 2: Three-stage line combination, 3: Three-stage line+ptp+line combination | INT16U |
| D77 | The jump instruction performs a carry-forward (cp) transition with the next instruction. 0: No transition, 1: Transition | INT16U |
| D78 | Archs: Jump instruction starts with a rising distance limit | Float |
| D80 | Arche: Jump instruction ends the distance limit | Float |
| D82 | Limz: Jump instruction height limit for vertical axis lifting | Float |
| D84 | Point sequence number, user-defined, monotonically increasing (for easy correlation with sd5404 or log viewing of anomalies) | INT32U |
| D86 | The number of transition points for attitude changes is limited to 5⑥. | INT16U |
| D92 | External shaft lifting height ⑩ for jump models with additional shafts | Float |

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. For external axis cooperative motion types 1, 2, and 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to reference coordinate system D18=6, D19=1 (workpiece coordinate system). When setting D18=4 (tool coordinate system), the point position information is relative displacement. For instance: If the robot's tool-end position is (400,200,100) and the point position x, y, z information is set to (50,20,-20), the robot will reach (450,220,80) after movement.

③: When speed, acceleration, or both are not configured, the system automatically adjusts the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points, one being the arc auxiliary point and the other the arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint is the starting point, not the endpoint information in the point input.

⑤: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: performing speed planning in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: performing speed planning in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values correspond to the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time.

For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑥: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated according to the path length ratio, with the P3-P6 posture change being equally distributed.

⑦: When the robot is equipped with a height-adjustable external axis, this register allows configuration of the external axis's elevation during JUMP trajectory transitions (similar to LimZ in JUMP trajectories). The value remains independent of JUMP group parameters, with the movement duration matching the robot's JUMP time. To enable this function, configure the external axis as gantry type, select only controlled external axis 1 (moving axis), and set the external axis mode to 6.

## 13.4 Notes on parameter settings

- For jump instruction trajectory types, only D76 needs to be set, with no need to configure D28, and it is unaffected by D28 values, meaning D76 takes precedence over D28. If other trajectory types are used, D76 must be set to 0. If the D76 value exceeds 3, the underlying alarm trajectory type will be incorrectly set to 10103. If the jump instruction degrades into a segment trajectory, that segment will execute according to the trajectory type specified in the jump instruction.
- Under normal conditions, the height parameters must satisfy limz> initial point height + arches and limz> termination point height + arches. For abnormal settings that fail to meet these conditions, non-gate-shaped movements may occur, which can be used in special scenarios.
- In D82, set Limz as the z-axis coordinate value, which represents the robot end's z-axis position in the base coordinate system. For D18 reference coordinate system, note the conversion of this value when referencing other coordinate systems.
- D78 and D80 are distance limits. If the value is set to a negative number, the underlying system takes the absolute value. If the value exceeds the length from the starting point (end point) to the midpoint, the underlying system adjusts it to the maximum allowed length.
- D77 is the control word that determines whether a jump instruction and the next segment's trajectory form a continuous path (CP). If the subsequent instruction is a jump, this control word becomes invalid, meaning no transition is permitted between consecutive jumps. Since the jump instruction internally sets D17 to 200, even with D77 enabled, the transition degree of this CP process cannot be controlled via D17.
- D17 represents the percentage of CP transition time. For jump instructions, this value is forcibly set to 200 at the hardware level to maximize CP transition.
- D26 is the percentage of path transition error, which is used to control the spline transition radius.
- JUMP Degradation Phenomenon: When the target point height equals limz but differs from the starting point height, the JUMP trajectory degrades to only lifting and translation segments. If no horizontal displacement occurs, it reverts to lifting segments only. When the starting point height equals limz but differs from the target point height, the jump trajectory degrades to only translation and descent segments. If no horizontal displacement occurs, it reverts to descent segments only. When no horizontal displacement exists and the starting point height, target point height, and limz are pairwise unequal, the jump trajectory degrades to only lifting and descent segments. If the starting point height, target point height, and limz are equal, the point is completely filtered out.

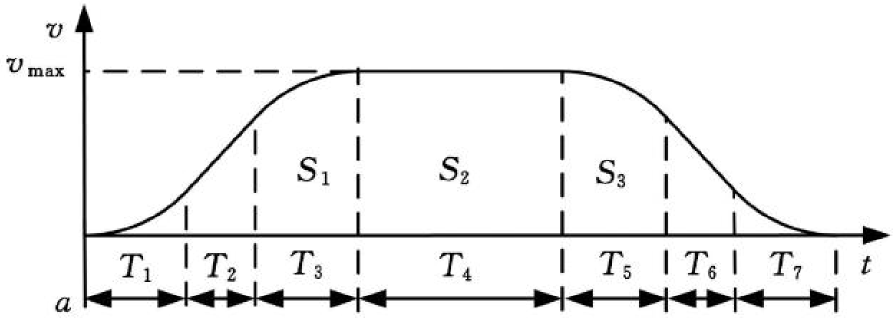
jump instruction transition specification

- Jump instructions do not support spline transitions with preceding or following motion instructions.
- A jump instruction may transition to a consecutive non-jump instruction via a control point (CP).
- The Jump instruction internally supports CP transitions, while the Jump type 2 can handle spline transitions.
- Jump instruction transition mode settings:

| D17（INT16S） | D26（float） | D76 | transitional form |
|---|---|---|---|
| 200 (Internal Mandatory) | 0 | 1，3 | CP transition |
| | | 2 | CP transition |
| 200 (Internal Mandatory) | Not zero | 1，3 | No transition |
| | | 2 | spline transition |

Jump instruction transition effect settings

- CP transition: The system employs trajectory overlay for transitions, enabling CP transitions between all trajectories (including PTP-PTP, PTP-Cartesian, and Cartesian-Cartesian).




CP processing at Jump points:

The current segment is the middle segment. The cp time is calculated as: min(T1 + T2 + T3 + (D17-100) * T4/2), ArchS

This segment is the end segment, with cp time calculated as min(T5 + T6 + T7 + (D17-100) * T4/2, ArchE)

The current segment is the starting segment. The cp time is calculated as: min(T5 + T6 + T7 + (D17-100) * T4/2, ArchS)

The previous segment is the end segment. If the cp flag is set to 1, the cp time is calculated as: min(T1 + T2 + T3 + (D17-100) * T4/2), ArchE.

This demonstrates that the cp transition within the jump instruction requires no D17 configuration, with its actual effect determined by the distance limit parameter and segment length.

- Spline Overlap: Uses B-splines to transition between Cartesian trajectories with angles, only between Cartesian trajectories (line to line, line to arc, arc to arc).

Handle the spline transition radius at the Jump point:

The current segment is a middle segment. Transition radius: min(Start segment length-arches, Middle segment length * 0.5)

The current segment is the end segment. Transition radius = min(remaining intermediate segment length, end segment length-archae)

This indicates that for the jump instruction type 2 with internal spline transition, D26 must be configured, though its actual performance is determined by the distance limit parameter and segment length.

## 13.5 Function Application and Examples

The source file sample program is as follows:

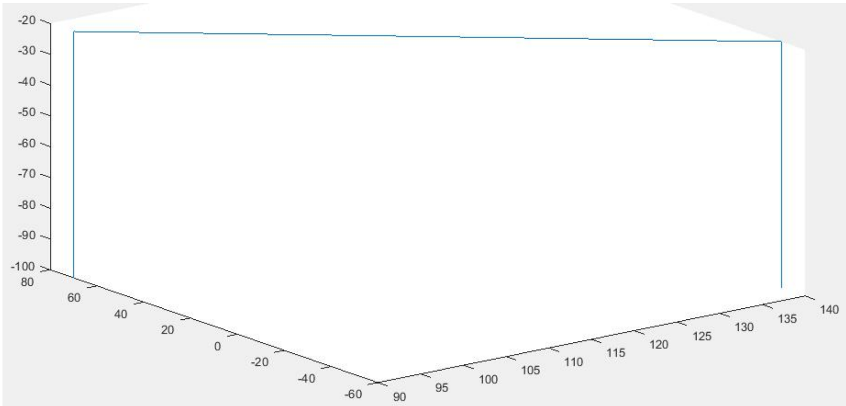
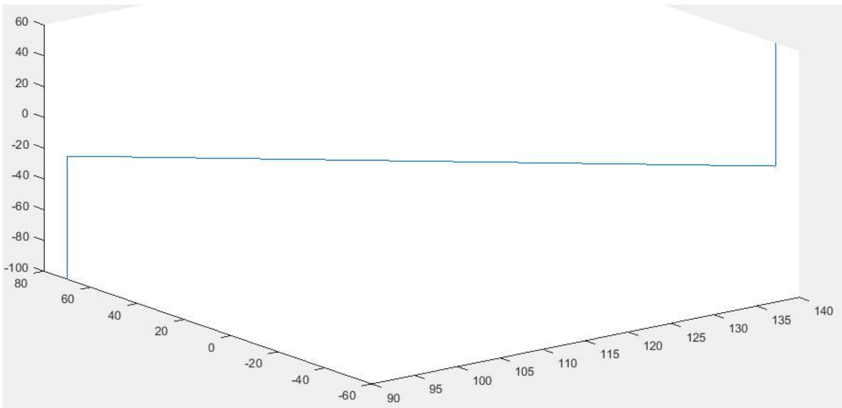
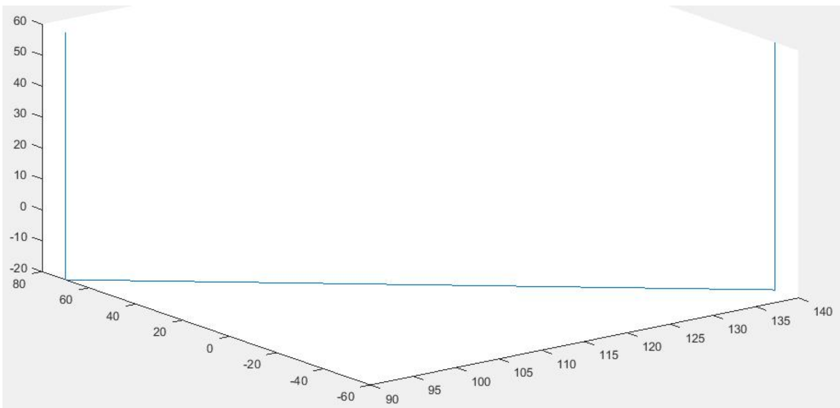
```
#include"RCPATH.h"

#define pNum 4
RcPointInformation RcPoint[pNum] = {
  {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,50,20,100,1,0,0,0,0,0},
{{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,60,30,100,2,0,0,0,0,0},
{{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,50,30,80,3,0,0,0,0,0},
}.

void RCPATH(PINT16S W, BIT B)
{
  if (SD[5500] == 0)
  {
    for (int i = 0. i < SD[5401]. i++)
    {
// Transmit pose data
      *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
      *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
      *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
      *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
      *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
      *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q2.
      *(float*)&D[(SD[5588] + 100 * i + 14)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType.//hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
D[(SD[5588] + 100 * i + 76)] = RcPoint->JumpType.//Jump type
D[(SD[5588] + 100 * i + 77)] = RcPoint->CPEnable. // Check if JUMP enables CP
*(float)* & D[(SD[5588] + 100 * i + 78)] = RcPoint->ArchS. // Ascending distance
*(float) & D[(SD[5588] + 100 * i + 80)] = RcPoint->ArchE. //Decrease distance
*(float)* & D[(SD[5588] + 100 * i + 82)] = RcPoint->LimZ. // Lift height limit
```

 User Manual of RC6F Series Robot Controller 13 JUMP Function

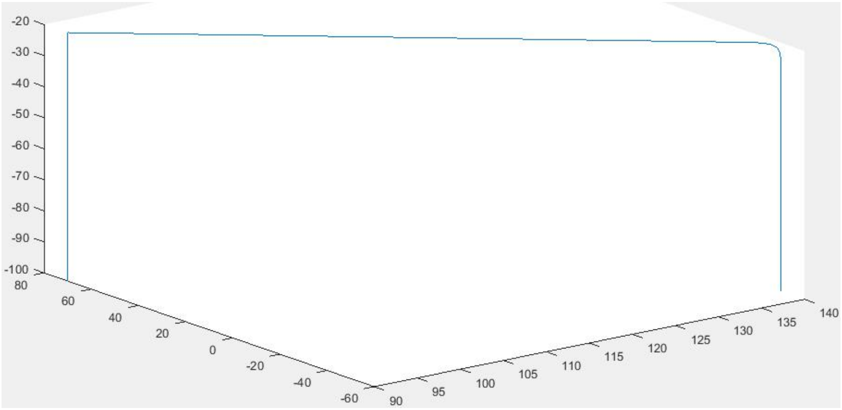
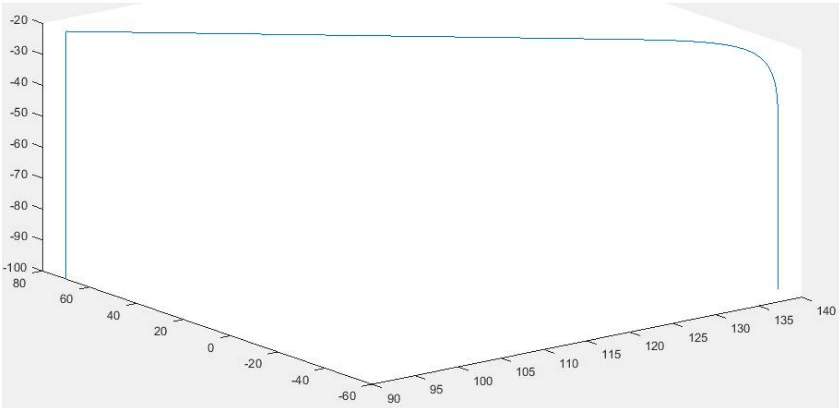
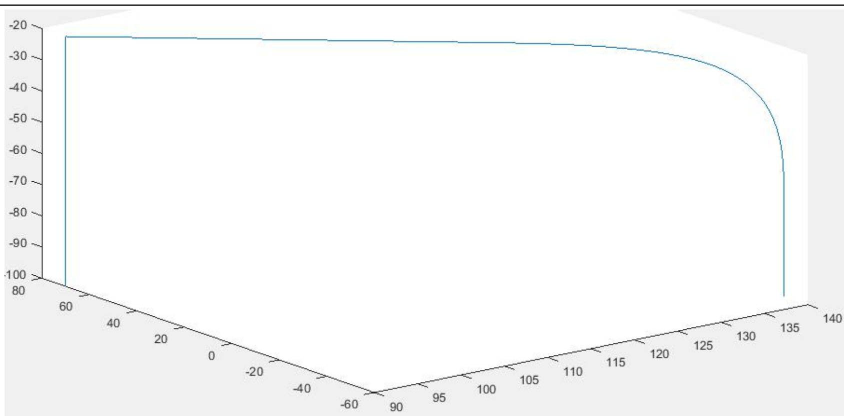
```
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
    }
  }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
}
```

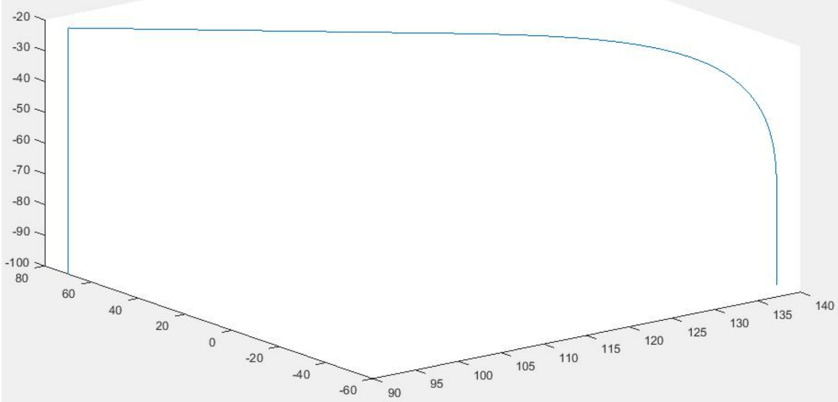
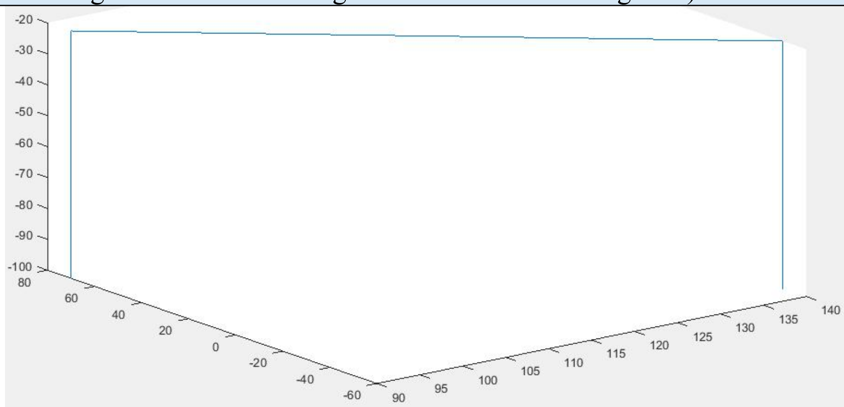
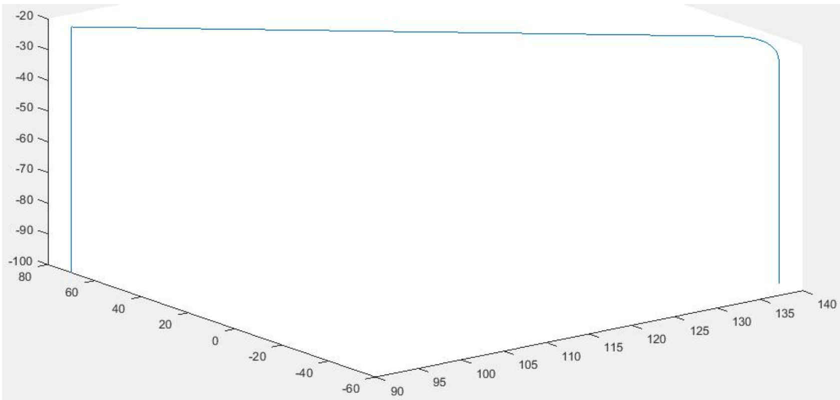
1) Trajectory parameter settings

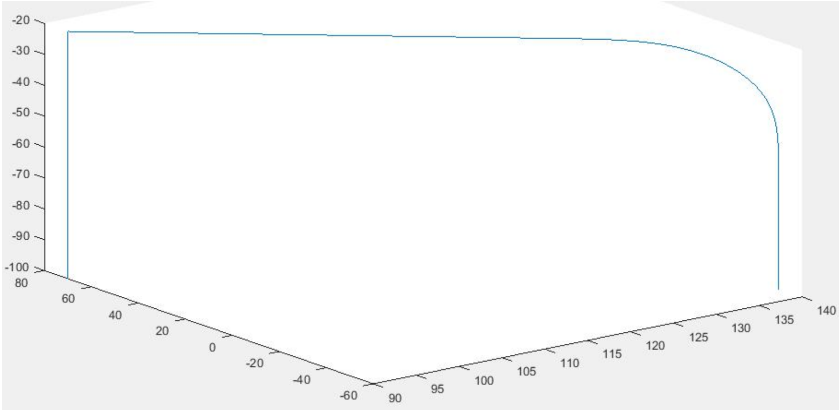
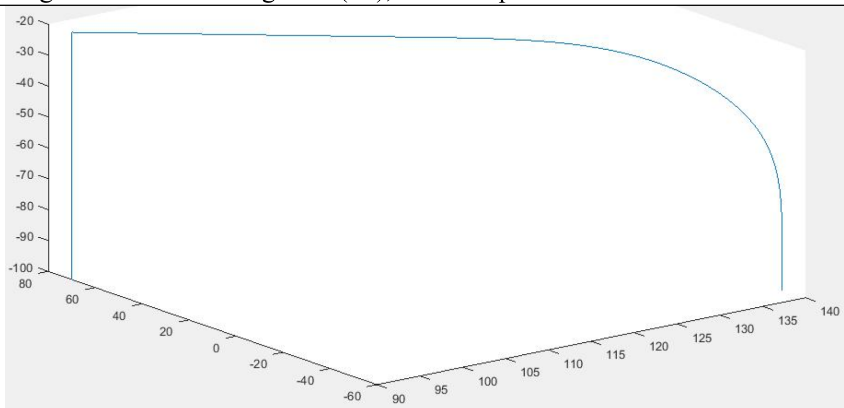
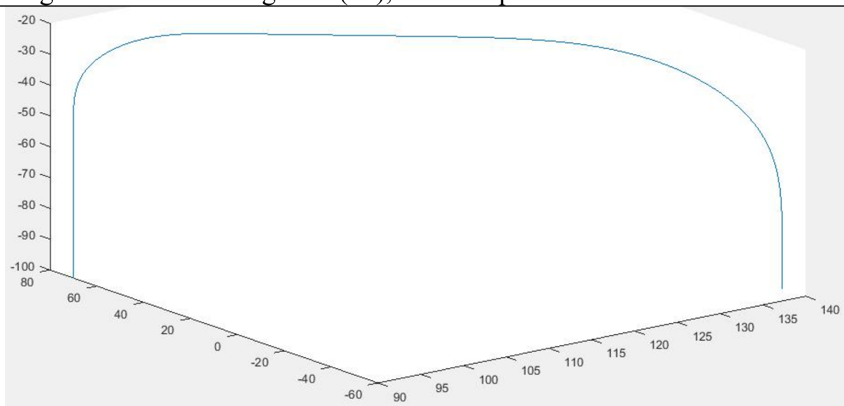
The parameters that determine the trajectory curve include: the coordinates of the starting and ending points, and the limz value.

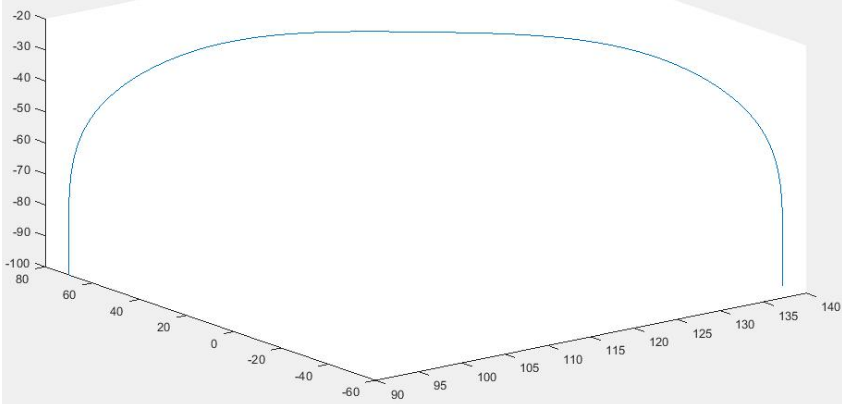
| parameter setting | Trajectory graph (determined by the z values of the start and end points and the limit z) |
|---|---|
| Start coordinates:<br>140,-50,-100,1,0,0,0<br>End coordinates:<br>90,70,-100,1,0,0,0<br>Jump type: 2<br>archs：90<br>arche：90<br>limz：-20 |  |
| Start coordinates:<br>140,-50,60,1,0,0,0<br>End coordinates:<br>90,70,-100,1,0,0,0<br>Jump type: 2<br>archs：90<br>arche：90<br>limz：-20 |  |
| Start coordinates:<br>140,-50,60,1,0,0,0<br>End coordinates:<br>90,70,60,1,0,0,0<br>Jump type: 2<br>archs：90<br>arche：90<br>limz：-20 |  |

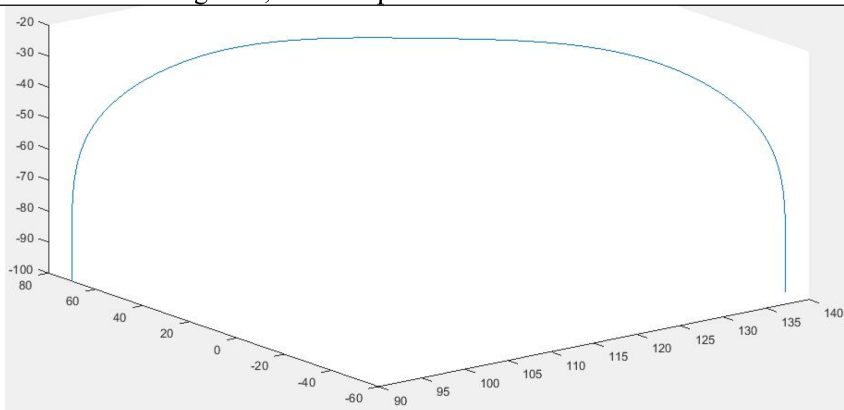
2) Transition parameter settings

| CP transition mode (determined by the intermediate segment's movement duration and the distance of ascent or descent) | |
|---|---|
| Start coordinates: 140,-50,-100,1,0,0,0<br>End coordinates: 90,70,-100,1,0,0,0<br>Jump type: 2<br>archs： 70<br>arche： 90<br>limz： -20 | <br>The rise distance is set to 70, with 10 remaining for the CP transition. Since the midpoint's half-time exceeds the initial segment's remaining distance time, the CP transition duration is determined by the rise distance. |
| Start coordinates: 140,-50,-100,1,0,0,0<br>End coordinates: 90,70,-100,1,0,0,0<br>Jump type: 2<br>archs： 50<br>arche： 90<br>limz： -20 | <br>The rise distance is set to 50, with 30 remaining for the CP transition. Since the midpoint's movement time exceeds half of the initial distance's duration, the CP transition time is determined by the rise distance. |
| Start coordinates: 140,-50,-100,1,0,0,0<br>End coordinates: 90,70,-100,1,0,0,0<br>Jump type: 2<br>archs： 30<br>arche： 90<br>limz： -20 | <br>The ascent distance is set to 30, with 50 remaining for the CP transition. Since the intermediate phase takes less than half the time required for the initial phase, the CP transition duration is determined by the intermediate phase duration. |

| | |
|---|---|
| Start coordinates: 140,-50,-100,1,0,0,0<br>End coordinates: 90,70,-100,1,0,0,0<br>Jump type: 2<br>archs：10<br>arche：90<br>limz：-20 | <br>The ascent distance is set to 10, with 70 remaining for the CP transition. Since the intermediate phase takes less than half the time required for the initial phase, the CP transition duration is determined by the intermediate phase duration. |
| B-spline transition method (determines the maximum path transition radius based on the ascending or descending distance and the length of the intermediate segment) | |
| Start coordinates: 140,-50,-100,1,0,0,0<br>End coordinates: 90,70,-100,1,0,0,0<br>D26 transition error: 100<br>Jump type: 2<br>archs：90<br>arche：90<br>limz：-20 | <br>The spline transition radius in the Jump instruction is determined by the minimum of the remaining distance after subtracting the ascending (or descending) distance and the intermediate segment length. With the ascending and descending distances set to 90, the internal variable takes the maximum distance of 80, and the remaining distances for both the start and end segments are 0, the spline transition radius becomes 0, and the trajectory shows no transition. |
| Start coordinates: 140,-50,-100,1,0,0,0<br>End coordinates: 90,70,-100,1,0,0,0<br>D26 transition error: 100<br>Jump type: 2<br>archs：70<br>arche：90<br>limz：-20 | <br>The remaining distance of the initial segment is 10, which is less than half of the length of the middle segment (65), and the spline transition radius is 10. |

| | |
|---|---|
| Start coordinates:<br>140,-50,-100,1,0,0,0<br>End coordinates:<br>90,70,-100,1,0,0,0<br>D26 transition error: 100<br>Jump type: 2<br>archs：40<br>arche：90<br>limz：-20 | <br>The remaining distance of the initial segment is 40, which is less than half the length of the middle segment (65), and the spline radius is 40. |
| Start coordinates:<br>140,-50,-100,1,0,0,0<br>End coordinates:<br>90,70,-100,1,0,0,0<br>D26 transition error: 100<br>Jump type: 2<br>archs：10<br>arche：90<br>limz：-20 | <br>The remaining distance of the initial segment is 70, which is larger than half the length of the middle segment (65), and the spline radius is 65. |
| Start coordinates:<br>140,-50,-100,1,0,0,0<br>End coordinates:<br>90,70,-100,1,0,0,0<br>D26 transition error: 100<br>Jump type: 2<br>archs：10<br>arche：50<br>limz：-20 | <br>The remaining distance of the initial segment is 70, the length of the middle segment is half of 65, and the spline radius is 65. the remaining distance of the final segment is 30, the remaining length of the middle segment is 65, and the spline radius is 30. |

<table>
<tr>
<td>Start coordinates: 140,-50,-100,1,0,0,0<br>End coordinates: 90,70,-100,1,0,0,0<br>D26 transition error: 100<br>Jump type: 2<br>archs：10<br>arche：10<br>limz：-20</td>
<td>


The remaining distance is 70 for the initial segment, 65 for the middle segment, and the spline radius is 65. The remaining distance is 70 for the final segment, 65 for the middle segment, and the spline radius is 65.</td>
</tr>
<tr>
<td>Start coordinates: 140,-50,-100,1,0,0,0<br>End coordinates: 90,70,-100,1,0,0,0<br>D26 transition error: 50<br>Jump type: 2<br>archs：10<br>arche：10<br>limz：-20</td>
<td>


The remaining distance is 70 for the initial segment, 65 for the middle segment, and the spline radius is 65. For the final segment, the remaining distance is 70, the middle segment is 65, and the spline radius is 65. These settings are independent of the D26 configuration.</td>
</tr>
</table>

User Manual for RC6F Series Robot Controller 14 Non-Motion Command Functions

# 14. Non-sporting instruction functions

## 14.1 Function Overview

The non-motion instruction function is designed to reduce the artificial segmentation of RCPATH instruction in application layer, including the wait instruction and the process output instruction. The automatic segmentation of the wait instruction is realized at the lower level, and the process instruction does not interrupt the motion output signal.

## 14.2 Function Specifications and User Instructions

### 14.2.1 Wait instruction wait

- The wait instruction (wait) operates as a non-motion segment. When the previous trajectory completes execution, if the current segment is a non-motion segment, the wait configuration parameter activates, continuously awaiting signal input and time condition fulfillment. If a time condition is specified, the next motion instruction executes upon either signal condition or time condition fulfillment. Without a time condition, the next motion instruction only executes after the signal condition is satisfied.
- Input parameter: signal sequence number index plus wait time Twait. The input signal is represented by M coils, where M[index]=ON indicates the signal condition is met. The wait time Twait (in milliseconds) is required when the time condition is satisfied.

Note: To enable this feature when only waiting for time and not for signal input, turn on the corresponding switch.

- Annotation feature: When the wait time and wait signal of the wait instruction are both set to 0, the annotation feature is enabled. This hides the current point without affecting the transition effect to the next point under this annotation.
- When the wait instruction is triggered by the waiting time t, the SM3016 signal begins output after t*SD6288[0~1). (Note: SD6288 value cannot be set to 1, and the SM3016 signal ceases output after trajectory interpolation is completed.)
- In the non-forward-point transmission mode, the non-moving segment waits for the signal but not for the time. Upon successful signal response, the SM3016 signal is not output.
- Non-motion commands do not support external signal interaction or processOut commands.
- The wait time is set to a value other than 0, with the minimum setting being 20ms.

### 14.2.2 Process Output Instruction processOut

The signal output function is opened during the movement, and the signal is output according to the degree of completion or the set time.

- Trigger upon distance from starting point: The command activates when the current segment's movement distance exceeds the preset threshold. If the threshold exceeds the total trajectory length, the signal will respond at the last interpolation point of this segment. This mode is not applicable to PTP trajectories or JUMP1/JUMP3 segments. Enabling this setting may cause abnormal response times or no response at all.

User Manual for RC6F Series Robot Controller 14 Non-Motion Command Functions

- Trigger when distance reaches a percentage of the total length: The command activates when the current segment's movement distance exceeds a specified percentage of the total length from the starting point. The percentage ranges from 0 to 100. This mode does not apply to JUMP1 and JUMP3. If enabled, it may cause abnormal or no signal response time.
- End-point distance trigger: The system activates when the remaining distance of the current trajectory segment is less than the preset threshold, calculated from the end point. If the preset threshold exceeds the total trajectory length, the system will respond to the current trajectory signal. This mode is not applicable to PTP trajectories or JUMP1/JUMP3 segments. Enabling this setting may cause abnormal response times or no response at all.
- Trigger when the remaining distance reaches a percentage of the total length: The instruction is triggered when the remaining distance of the current segment is less than a specified percentage of the total length, calculated from the end point. The percentage range is [0,100]. JUMP1 and JUMP3 do not support this mode. If enabled, it may cause abnormal signal response time or no response.
- Time-triggered: The instruction is triggered when the remaining time of the current segment is less than the set time, calculated from the end point. If the set time exceeds the total duration of the trajectory, the current trajectory signal will be triggered.
- Delayed trigger: Timing starts from the end point, counting backward from the completion of the current segment. The command activates when the duration exceeds the preset time. If the RC command has ended (SM3001 resets) and the delay period hasn't elapsed, the signal remains inactive.

Distance-length and distance-percent triggers can be configured with delay times for response signal output. When a transition segment exists, the path length or total runtime calculation for the current segment starts from the previous transition's end point and ends at the next transition's end point. A single motion command supports up to two processOut settings. The system provides 100 configurable processOut parameter blocks, each occupying 8 bytes. Dynamic modifications to processOut parameters are updated via SM registers. For arc and full-circle trajectories, when auxiliary points are considered, the trajectory is processed as two segments. When using inverted signals for output, two processOut signals are generated.

User Manual for RC6F Series Robot Controller 14 Non-Motion Command Functions

## 14.3 RCPATH instruction data point format

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-D12 | Target point coordinates [x, y, z, q0, q1, q2, q3] | Float |
| D15 | Tool id, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-200 to indicate the percentage of cp time, and set 0 or-1 to indicate the zero-speed point | Int16s |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and attitude ③ | Float |
| D22 | Maximum acceleration percentage of running path and posture | Float |
| D24 | Maximum deceleration percentage of running path and posture | Float |
| D26 | Path transition error percentage | Float |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: Ptp, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: Ptp with joint speed planning (unfixed trajectory). 1: Ptp with joint speed planning (fixed trajectory). 2: Ptp with cartesian speed planning (fixed trajectory). 3: Ptp with acceleration optimization planning (fixed trajectory) ⑤ | INT16U |
| D84 | Point number, user-defined, monotonically increasing (for easy correlation with sd5404 or log viewing of anomalies) | INT32U |
| D86 | The number of transition points for attitude changes is limited to 5⑥. | INT16U |
| D87 | Wait wait time, in milliseconds. Set to 0 to disable the wait time and wait for the signal only. | INT16U |
| D88 | Wait for the m signal sequence number (e.g., 1000) to turn m[1000] on. A positive value indicates high-level waiting, a negative value indicates low-level waiting, and 0 means no signal effect—only time waiting (not supported in the sd6288-compatible non-forwarding version). | Int32s |
| D90 | Load processout1 serial number, range 1 to 100 | INT16U |
| D91 | Load processout2 number, range 1 to 100 | INT16U |

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. When the external axis collaborative motion types are 1, 2, or 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to reference coordinate system D18=6, D19=1 (workpiece coordinate system). When setting D18=4 (tool coordinate system), the point position information is relative displacement. For instance: If the robot's tool-end position is (400,200,100) and the point position x, y, z information is set to (50,20,-20), the robot will reach (450,220,80) after movement.

User Manual for RC6F Series Robot Controller 14 Non-Motion Command Functions

③: When speed, acceleration, or both are not configured, the system will automatically adjust the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points: one as the arc assist point and the other as the arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the point's trajectory type is changed to straight line. For a full circle, the endpoint is set to the starting point rather than the endpoint information in the point input.

⑤: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: performing speed planning in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: performing speed planning in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values correspond to the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑥: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated based on the path length ratio, with the posture change amount from P1 to P6 being equally distributed.

## 14.4 ProcessOut parameter point format

| Register | Data type | Explain |
|---|---|---|
| D0 | INT16U | Processout determines the output condition type: 1: Triggered by starting distance length, 2: Triggered by starting distance percentage, 3: Triggered by ending distance length, 4: Triggered by ending distance percentage, 5: Triggered by time advance, 6: Triggered by time delay |
| D1 | INT16U | The processout delay time (in milliseconds) is active only when the output condition type of processout is 1 to 4, triggering the instruction delay signal. Note: If the delay time exceeds the complete rc period, no corresponding signal will be output. |
| D2 | Float | Process output trigger condition value, path length (mm)/path percentage (%)/time (ms) |
| D4 | INT32U | The processout function outputs the sequence number of the m or y signal. For example, assigning 1000 results in m[1000]. The y signal value is a decimal number, which corresponds to an octal number in the plc. For example, setting y[12] assigns the value 10. |
| D6 | INT16U | The output signal is valid for both high and low levels: 0: Processout triggers a high-level output, 1: Processout triggers a low-level output, 2: Processout outputs an inverted signal. |
| D7 | INT16U | Process output signal type: 0: M, 1: Y |

User Manual for RC6F Series Robot Controller 14 Non-Motion Command Functions

# 14.5 Description of Registers

Non-motion instructions are based on RCPATH instructions and therefore include all registers of RCPATH. The additional registers are listed below.

| Register | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| Sd6285 | INT16U | Processout parameter for setting the type of storage register<br>0: Hd register<br>1: Hsd register | Default value 0 | R/W | Active next time |
| Sd6286 | INT32S | Offset address of the storage register for the region setting parameter<br>For example, if sd6285 equals 0 and sd6286 equals 200, the hd200-hd999 module records 100 processout parameter block settings. Negative values are processed as absolute values. | | R/W | Active next time |
| Sd6288 | Float | Wait deceleration start percentage | [0,1) | R/W | Active next time |
| Sm3151 | Bit | Processout parameter refresh control word | | R/W | With immediate effect |

User Manual for RC6F Series Robot Controller 14 Non-Motion Command Functions

## 14.6 Function Applications

```
#include"RCPATH.h"
#include"processout.h"

#define pNum 4
#define ProcessOutNum 2
RcPointInformation RcPoint[pNum] = {
    {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,2},
    {{325.3,-125.12,289.17,0.217784,-0.882695,-
0.203444,0.363360},0,0,0,0,0,1,30,80,80,0,9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,50,200,1,0},
    {{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0},
}.

//Non-motion instruction configuration
ProcessOutPoint ProcessOut[ProcessOutNum] = {
{2,0,40,1000,0,0}, //Set signal M[1000] to ON at 40% of the starting point
{2,0,80,1000,1,0}, //Set signal M[1000] to OFF at 80% of the starting point
}.

void RCPATH(PINT16S W, BIT B)
{
//Configure non-motion commands
    ProcessOutCfg(0, 1000, ProcessOut, (int)(sizeof(ProcessOut) / sizeof(ProcessOut[0]))).

    if (SD[5500] == 0)
    {
        for (int i = 0. i < SD[5401]. i++)
        {
// Transmit pose data
            *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
            *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
            *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
            *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
            *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
            *(float*)&D[(SD[5588] + 100 * i + 10)] = RcPoint->targetPos.orient.q2.
            *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 14)] = RcPoint->weaveAngleZ. //Arc angle
D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType.//hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
```

User Manual for RC6F Series Robot Controller 14 Non-Motion Command Functions

```
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
D[(SD[5588] + 100 * i + 87)] = RcPoint->waitTime. //Set the wait time
*(INT32S*) & D[(SD[5588] + 100 * i + 88)] = RcPoint->waitOrder. //Wait for M signal sequence number
D[(SD[5588] + 100 * i + 90)] = RcPoint->processOut1Num. //ProcessOut1 sequence number
D[(SD[5588] + 100 * i + 91)] = RcPoint->processOut2Num. //ProcessOut2 sequence number
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
        }
    }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
}

void ProcessOutCfg(INT16U Type, INT16U Addr, ProcessOutPoint* Inf, int lenth)
{
SD[6285] = Type. //Configure the register type
*(INT32U)* & SD[6286] = Address. //Set the initial address
    for (int i = 0. i < lenth. i++)
    {
if (SD[6285] == 0) //Check if it's the HD register
        {
HD[(INT32U & SD[6286]) + 8 * i + 0] = Inf[i]. TriggerCondition. // Process output trigger condition
HD[(INT32U & SD[6286]) + 8 * i + 1] = Inf[i].DelayTime. //Process Output delay time
*(float)* & HD[(*(INT32U*) & SD[6286]) + 8 * i + 2] = Inf[i]. ConditionValue. // Process Out triggers the condition value
*(INT32U)* & HD[(INT32U)* & SD[6286] + 8 * i + 4] = Inf[i]. OutAddr. // Process Out M signal address
HD[(INT32U & SD[6286]) + 8 * i + 6] = Inf[i].level. //PrcessOut trigger level setting
        HD[(INT32U* & SD[6286]) + 8 * i + 7] = Inf[i]. OutType. // Process Out triggers the register type setting
        }
else if (SD[6285] == 1) //If the HSD register is detected, execute the same procedure as above
        {
            HSD[(*(INT32U*)&SD[6286]) + 8 * i + 0] = Inf[i].TriggerCondition.
            HSD[(*(INT32U*)&SD[6286]) + 8 * i + 1] = Inf[i].DelayTime.
            *(float*)&HSD[(*(INT32U*)&SD[6286]) + 8 * i + 2] = Inf[i].ConditionValue.
            *(INT32U*)&HSD[(*(INT32U*)&SD[6286]) + 8 * i + 4] = Inf[i].OutAddr.
            HSD[(*(INT32U*)&SD[6286]) + 8 * i + 6] = Inf[i].level.
            HSD[(INT32U* & SD[6286]) + 8 * i + 7] = Inf[i]. OutType. // Process Out triggers the register type setting
        }
    }
SM[3151] = 1. //Refresh the ProcessOut information
}
```

User Manual for RC6F Series Robot Controller 14 Non-Motion Command Functions

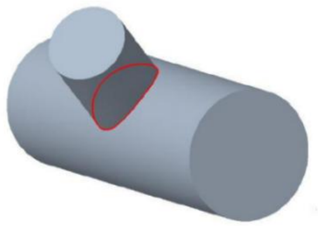
The non-motion command header file is as follows:

```
#ifndef _PROCESSOUT_H_
#define _PROCESSOUT_H_

typedef struct
{
INT16U TriggerCondition. //Trigger condition
   INT16U DelayTime.
   float ConditionValue.
   INT32U OutAddr.
   INT16U level.
 INT16U OutType.
}ProcessOutPoint.

void ProcessOutCfg(INT16U Type, INT16U Addr, ProcessOutPoint* Inf, int lenth).

#endif
```

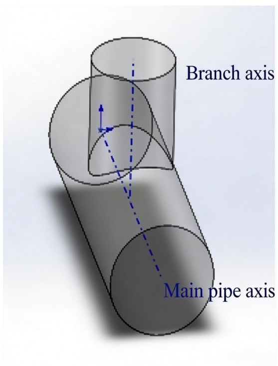
# 15. Intersection Line Function

## 15.1 Function Overview

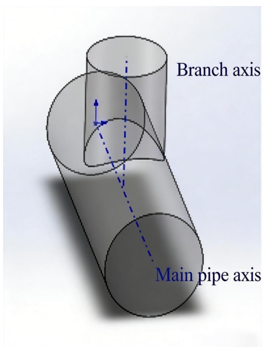
- The spatial curve formed by the intersection of two circular tubes is termed the intersection line, with four models: orthogonal, oblique, orthogonal offset, and oblique offset. To develop the robot's trajectory along the intersection line, the main coordinate system matrix is calibrated based on teaching points, and the intersection line trajectory equation is derived from these points. This ensures the robot follows the preset trajectory while maintaining smooth transitions in the welding torch's posture between teaching points.
- The intersection line function is implemented based on speed prediction, integrated into the RCPATH instruction, and supports external axis and swing arc functions.
- This function supports all kinds of intersection line models.



- The axis of the main pipe is perpendicular to and intersects with the axis of the branch pipe, as shown in the figure below.




- The orthogonal offset intersection line is where the axis of the main pipe is perpendicular to the axis of the branch pipe but does not intersect, as shown in the figure below:



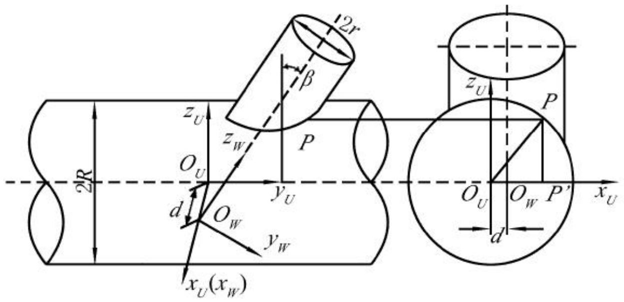
# 15.2 Function Specifications and User Instructions

## 15.2.1 Types of Intersection Lines

The intersection line supports all types of intersection line models.
The complete intersection line trajectory is calculated using five points: the starting point (the robot's current position), auxiliary point 1, auxiliary point 2, auxiliary point 3, and the endpoint. The starting point is either the endpoint of the previous segment or the robot's current position, requiring no manual teaching. Therefore, four points need to be manually taught, with three auxiliary points ideally selected as the highest and lowest points on the intersection line. The endpoint must coincide with the starting point.
The incomplete line trajectory is calculated by three points, namely the starting point, the auxiliary point and the end point. The auxiliary point and the end point are two teaching points, and the end point cannot be the same as the starting point. The error of whether the starting point and the end point are the same is determined by the register SD5522.
The type of intersection line is determined by the D28 code in the RC point format. A D28 value of 7 indicates an incomplete intersection line, while a value of 8 denotes a complete one (as detailed below).

## 15.2.2 Intersection Line Coordinate System

The mathematical model of the intersection line trajectory is established by using the form of fixed coordinate system. The intersection line trajectory is determined by the main pipe coordinate system and the branch pipe coordinate system, where the main pipe is the pipe that is crossed, and the branch pipe is the pipe that is crossed. Zu-Xu-Yu represents the main pipe coordinate system, and Zw-Xw-Yw represents the branch pipe coordinate system.




## 15.2.3 Intersection Line Parameters

1) Radius of Inclination (R)
When two pipes are connected, the pipe with the largest radius is considered the main pipe, and its radius is: main pipe radius. If the input parameter is negative, it will be automatically corrected to a positive number and executed.

2) Branch radius (r)
When two pipes are connected, the smaller radius pipe is considered a branch pipe. The radius of this pipe is: branch pipe radius. If the input parameter is negative, it will be automatically corrected to a positive number and executed.

3) The offset (d) of the branch pipe relative to the main pipe
The offset is the straight line distance between the axis of the main pipe and the axis of the branch pipe in

space. When the offset is 0, it means that the axis of the main pipe and the axis of the branch pipe intersect, which is the orthogonal intersection line. If the offset is not 0, it means that the axis of the main pipe and the axis of the branch pipe do not intersect, which is the orthogonal offset intersection line.
When the offset is positive, it is equivalent to the branch pipe is offset along the X-axis of the main pipe coordinate system, and when the offset is negative, it is equivalent to the branch pipe is offset along the negative X-axis of the main pipe coordinate system.

4) The deflection angle (β) of the branch pipe relative to the main pipe

This deflection angle represents the angle between the z-axis of the main pipe and the z-axis of the branch pipe. A clockwise rotation around the x-axis is considered positive, with values ranging from-π/2 to π/2 in degrees. The angle shown in the figure above is positive.

5) Teaching error
This indicates the deviation between the intersection line trajectory parameters (main pipe radius and branch pipe radius) calculated from user-specified points and the user-input parameters. The default error threshold is 0.5mm. The alarm code activates when the teaching error exceeds this threshold.

6) Transformation Matrix from Robot Base Coordinate System to Master Coordinate System
Expressed in Eulerian angle, with position in mm and attitude in degrees.

7) Transformation matrix from the end coordinate system of the positioner to the main coordinate system
Expressed in Eulerian angle, with position in mm and attitude in degrees.

### 15.2.4 Usage Specifications

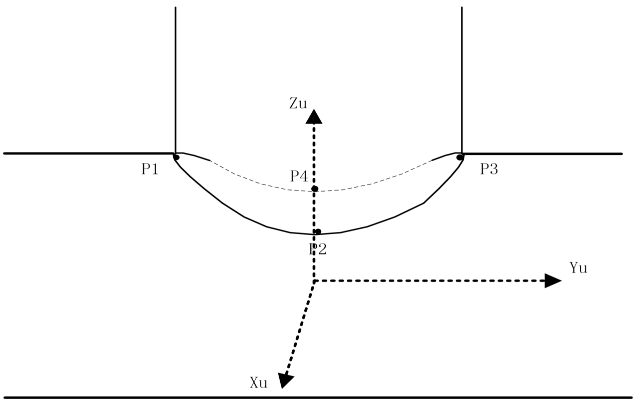
- Control word register must be set before instruction execution, otherwise it will affect instruction execution.
- The default values for the system power-on operation are 0 for the control radius, branch radius, offset, and deflection angle. Before executing RCPATH each time, the system reads data from the register into an internal variable, and the internal variable takes the absolute value (except for the deflection angle).
- Support the main pipe radius is greater than the branch pipe radius, the offset is less than the difference between the main pipe radius and the branch pipe radius.
- The default teaching error threshold is set to 0.5. Before executing RCPATH each time, data is read from the register and stored in the internal variable.
- When the external axis coordination function is disabled, enter the Euler angle of the rotational transformation between the intersecting line's main coordinate system and the base coordinate system in SFD6808+2i. When the function is enabled, input the Euler angle of the rotational transformation between the intersecting line's main coordinate system and the end coordinate system of the positioner's second axis in SFD6820+2i. Before executing RCPATH each time, the data is read from the register to the internal variable.
- Coordinate system calibration process:

1) To verify the external axis motion type: When no external axis is involved, simply input the Euler angle transformation from the robot base coordinate to the master coordinate in SFD6808+2i. For collaborative motion modes, the Euler angle transformation from the end-effector coordinate system of the positioner's second axis to the master coordinate system must be written into SFD6820+2i. If this Euler angle is missing, a calibration procedure is required.
2) When calibrating the Euler angle for the base coordinate system to the main coordinate system, ensure

the body accuracy meets the standard. The tool requires calibration, and the reference coordinate system should be selected as the base coordinate system. With the tool in place, select four teaching points corresponding to the initial values on the intersection line. Write the coordinate values into the first 12 floating-point registers starting from D29000, ensuring that the registers corresponding to the main pipe radius (SD7400), branch pipe radius (SD7402), offset (SD7404), and deflection angle (SD7406) have valid values. Trigger SM3208, and the calculation results will be displayed in the register SD7422+2i. The error from this calculation will be shown in SD7420. If the error falls within the allowable range, the result can be written into the corresponding SFD6808+2i. If the error is significant, recalibration is required.

3) When calibrating the Euler angle for transforming the end coordinate system of the 2-axis positioner to the master coordinate system, ensure the accuracy of the main body while first calibrating system components including the tool, ground rail, and positioner. Finally, calibrate the master coordinate system. Teach four points corresponding to initial values on the intersection line, which must be taught in the workpiece coordinate system (SD5593 matches the workpiece number, e.g., SD5593=1 corresponds to reference coordinate system settings SD4018=6 and SD4019=1). Write the coordinate values into the first 12 floating-point registers starting from D29000, ensuring values exist in registers corresponding to intersection line parameters: master radius (SD7400), branch radius (SD7402), offset (SD7404), and deflection angle (SD7406). Trigger SM3208 to calculate results displayed in register SD7422+2i, with calculation errors shown in SD7420. If errors fall within allowable range, write results to corresponding SFD6820+2i. If errors are significant, recalibration is required.

4) Initial value definition in calibration calculation: The initial value is the rotation angle of the intersection line (the angle between the line connecting a point on the intersection line and the center point and the x-axis, ranging from [-π, π], similar to the central angle of a circle). The unit is radians, with default values in the order of-1.57,0, 1.57,3.14. Teaching points must correspond one-to-one with these values, as shown in the figure.




The teaching points are arranged in the order p1--p2--p3--p4, with the rotation angles of p1, p2, p3 and p4 being-1.57 (approximate to-pi/2),0, 1.57 (pi/2) and 3.14 (pi) respectively.

These points should be positioned as close as possible to the initial value selection area during demonstration, though precise alignment with the target location is not required, each point must correspond one-to-one.

In some cases, the robot cannot reach certain points, so the initial values can be adjusted according to the actual situation, as long as a one-to-one correspondence is maintained.

The principle of selecting teaching points is to distribute them as evenly as possible along the entire line, preferably choosing initial points that are easily identifiable. For instance, the default four points are the two highest and two lowest points of the intersection line.

When the four selected teaching points are located on one side, the trajectory on that side will be more accurate, while the trajectory on the other side may be less precise, depending on the accuracy of the body, the tool, and the teaching points.

- A line segment must ensure that both auxiliary points and endpoints are of the same trajectory type. If the trajectory types are inconsistent, the original line segment will be forcibly modified to a straight line.
- The starting point, the auxiliary point and the end point must be on the intersection line, otherwise the intersection line cannot be formed.

## 15.3 RCPATH instruction data point format

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-D12 | Target point coordinates [x, y, z, q0, q1, q2, q3]⑧ | Float |
| D15 | Tool id, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-200 to indicate the cp time percentage, and set 0 or-1 to indicate the zero-speed point | INT16S |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and posture ③ | Float |
| D22 | Maximum acceleration percentage of running path and posture | Float |
| D24 | Maximum deceleration percentage of running path and posture | Float |
| D26 | Path transition error percentage | Float |
| D28 | Path type: 1: Straight line, 2: Circular arc, 3: Ptp, 7: Incomplete intersection line, 8: Complete intersection line ④, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: Ptp with joint speed planning (unfixed trajectory). 1: Ptp with joint speed planning (fixed trajectory). 2: Ptp with cartesian speed planning (fixed trajectory). 3: Ptp with acceleration optimization planning (fixed trajectory) ⑤ | INT16U |
| D68 | The connecting length at the intersection line's end point is less than one quarter of the arc length, measured in millimeters. | Float |
| D84 | Point sequence number, user-defined, monotonically increasing (for easy correlation with sd5404 or log viewing of anomalies) | INT32U |
| D86 | The number of transition points for attitude changes is limited to 5⑥. | INT16U |

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. When the external axis collaborative motion types are 1, 2, or 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to reference coordinate system D18=6, D19=1 (workpiece coordinate system). When setting D18=4 (tool coordinate system), the point position information is relative displacement. For instance: If the robot's tool-end position is (400,200,100) and the point position x, y, z information is set to (50,20,-20), the robot will reach (450,220,80) after movement.

③: When speed, acceleration, or both are not configured, the system automatically adjusts the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points: an arc auxiliary point and an arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint position is the starting point position rather than the endpoint information in the point input. Trajectory type 7 represents an incomplete intersection line, composed of the current point, auxiliary point, and endpoint. Trajectory type 8 represents a complete intersection line, consisting of the current point, auxiliary point 1, auxiliary point 2, auxiliary point 3, and the endpoint, with the requirement that the starting and endpoint poses must be identical. If the specified number of points is not met, the trajectory is forcibly converted to a straight line.

⑤: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: speed planning is performed in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: speed planning is performed in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values represent the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑥: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated based on the path length ratio, with the P3-P6 posture change amount being equally distributed.

⑦: The transition length of the intersecting line trajectory is determined by extending the current trajectory arc length from its endpoint, inheriting the endpoint's orientation. This value is defined by the transition length in the endpoint position format, with auxiliary point transition length disabled. The unit is mm. A value ≤ 0 indicates no transition length, while a value exceeding 1/4 of the intersecting line arc length triggers a range warning. This value can be approximated using the circumference calculation formula: 2πr/4.

⑧: When the intersection line has a high point or low point, the starting point, auxiliary point and ending point should be selected at the high point or low point as far as possible to obtain a better welding posture. When the intersection line is short and must pass through high and low points, or the intersection line is long and passes through many high and low points, the auxiliary point should be located at the middle position of the starting point and ending point as far as possible.

The point teaching sequence must be demonstrated sequentially, either clockwise or counterclockwise.

## 15.4 Description of Registers

| Address | Type | Explain | Unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SM3208 | Bit | Control point for coordinate system calibration calculation | | R/W | With immediate effect |
| SD5522 | Float | The threshold ① used in the intersection line trajectory to determine whether the given or taught point has the same point | Default: 0.01 | R/W | The next delivery point will take effect |
| SD7400 | Float | The radius of the intersecting line's main pipe is entered by the user | Mm | R/W | The next delivery point will take effect |
| SD7402 | Float | The radius of the intersecting line branch is entered by the user | Mm | R/W | The next delivery point will take effect |
| SD7404 | Float | The offset of the branch pipe relative to the main pipe is entered by the user (positive on the x-axis) | Mm | R/W | The next delivery point will take effect |
| SD7406 | Float | The deflection angle of the branch pipe relative to the main pipe is input by the user (positive for clockwise rotation around the x-axis, range: -π/2 to π/2). | Linear measure | R/W | The next delivery point will take effect |
| SD7408 | Float | Intersection line teaching error threshold, in mm (default: 0.5) | Default: 0.5 | R/W | The next delivery point will take effect |
| SFD6800+2*I (I=0,1,2,3) | Float | The initial value for the intersection line's main coordinate system calibration calculation requires that each teaching point position correspond exactly to its initial value. | The default is-1.57,0, 1.57,3.14 | R/W | The next delivery point will take effect |
| SFD6808+2*I (I=0,1...5) | Float | The euler angle for transforming the intersection line's main coordinate system relative to the robot's base coordinate system, with position in mm and attitude in | | R/W | The next delivery point will take effect |

 User Manual of RC6F Series Robot Controller 15 Intersection Line Function

| Address | Type | Explain | Unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| | | degrees. | | | |
| SFD6820+2*I (I=0,1...5) | Float | The euler angle for transforming the intersection line's main coordinate system to the 2-axis coordinate system of the positioner, with position units in mm and attitude units in degrees. | | R/W | The next delivery point will take effect |
| D29000+2*J+6*I (I=0,1,2,3, J=0,1,2) | Float | The coordinate system determines the required teaching point positions for calibration, with 4 points needed. Each point only requires xyz position information. | | R/W | |
| SD7410 | Float | Error of intersection line teaching | Mm | R | |
| SD7412 | Float | The calculated radius of the main pipe of the intersection line | Mm | R | |
| SD7414 | Float | The calculated radius of the intersecting line branch | Mm | R | |
| SD7416 | Float | The calculated offset of the intersecting line branch pipe relative to the main pipe | Mm | R | |
| SD7418 | Float | The calculated deflection angle of the intersecting line branch pipe relative to the main pipe | Linear measure | R | |
| SD7420 | Float | Calibration error of the main coordinate system for the intersection line | Mm | R | |
| SD7422+2*I (I=0,1...5) | Float | Calibration results of the intersection line's main coordinate system, euler angles (zyx), position units (mm), and attitude units (°) | | R | |

①: For all teaching points on the complete and incomplete intersection lines, form vectors pairwise. Check if the modulus is below the threshold set by SD5522. If so, the points are considered identical, preventing the intersection line from being executed and triggering alarm 10199.

# 16. Arc placement function

## 16.1 Function Overview

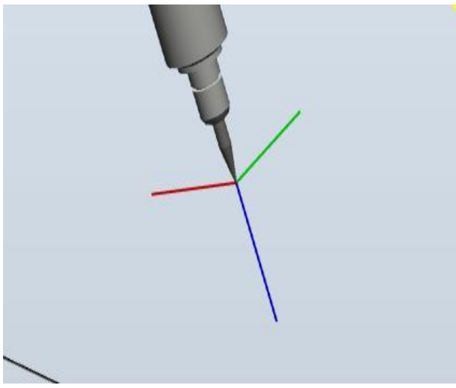
The system implements multi-segment arc motion through speed anticipation planning, capable of receiving discrete point data from the application layer. It performs speed anticipation, trajectory fusion, spline transitions, arc planning, and interpolation based on the received point data and parameters. This enables the manipulator to execute arc motions under any given mixed linear and circular segment commands, though PTP trajectory support is not available. The system supports dynamic speed adjustment, RBSTOP, and RBGOON functions, but does not currently support combined start-stop segment functionality.

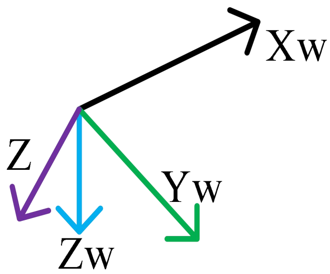
The arc swing function is implemented through the RCPATH instruction. After configuring the relevant arc swing parameters in the RCPATH instruction point attributes, the arc swing motion will be executed upon instruction execution.

## 16.2 Basic Definitions

### 16.2.1 Arc Reference Plane

To facilitate understanding of the swing arc plane, the reference plane is defined as follows: Xw: the direction of the weld seam and the tangent of the main trajectory. Yw: the normal vector of the plane formed by the Z-axis of the welding torch TCP and Xw. Zw: the normal vector of the plane formed by Xw and Yw. When Z is perpendicular to Xw, Z coincides with Zw.



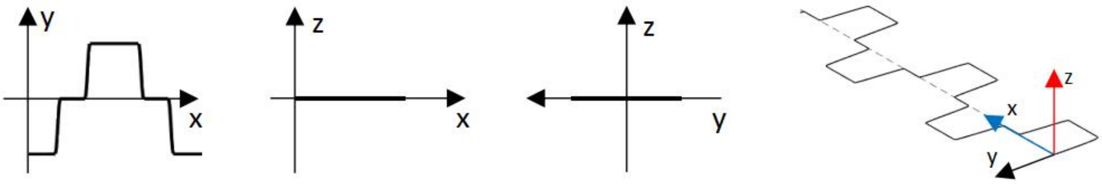


### 16.2.2 Types of Bending Arcs

The arc swing function supports five swing modes, which can be selected by setting the robot's RC command arc swing mode register D51. By default, this function is disabled.

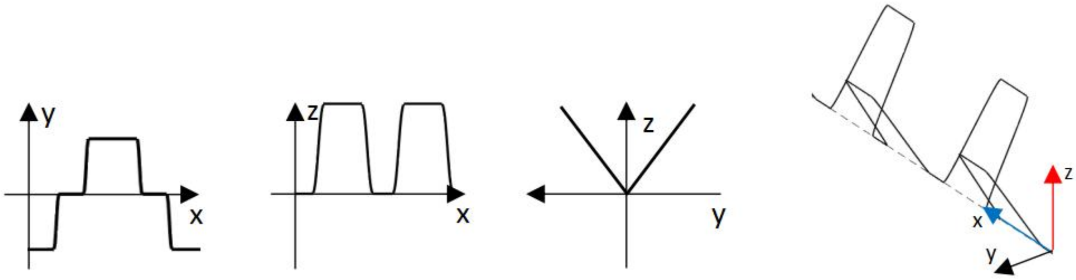
1) Z-shaped swing arc

- The welding seam is horizontally swayed.



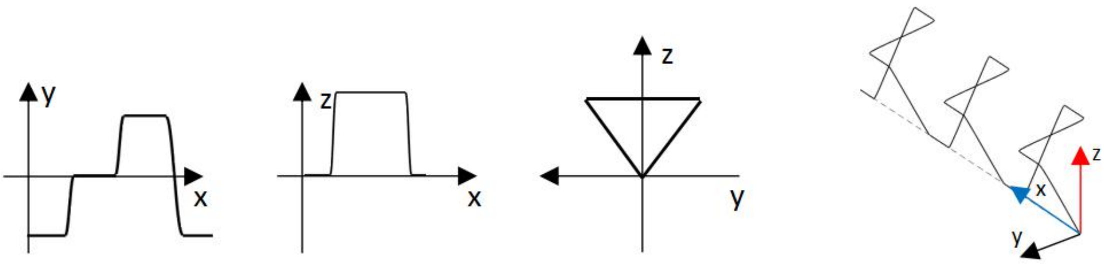

2) V-shaped swing arc

- The welding seam is swayed in a V-shape perpendicular to the weld.



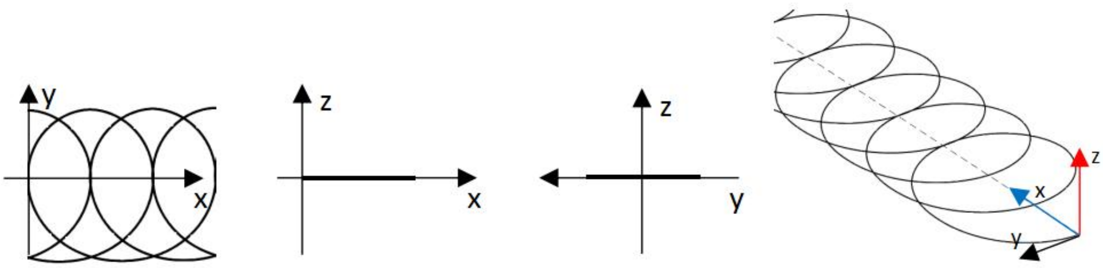

3) Triangular arc swing

- The triangle swing is perpendicular to the weld.



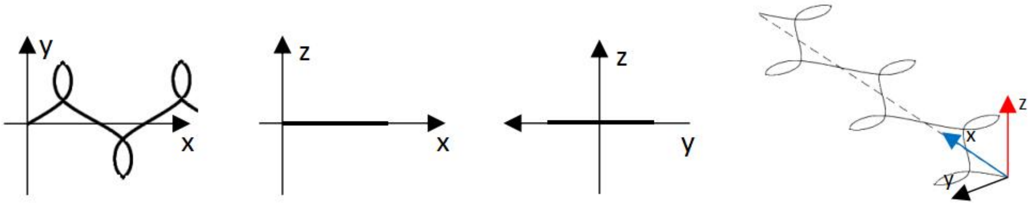

4) Circular arc swing

- The arc swing is perpendicular to the weld.



5) 8-shaped arc

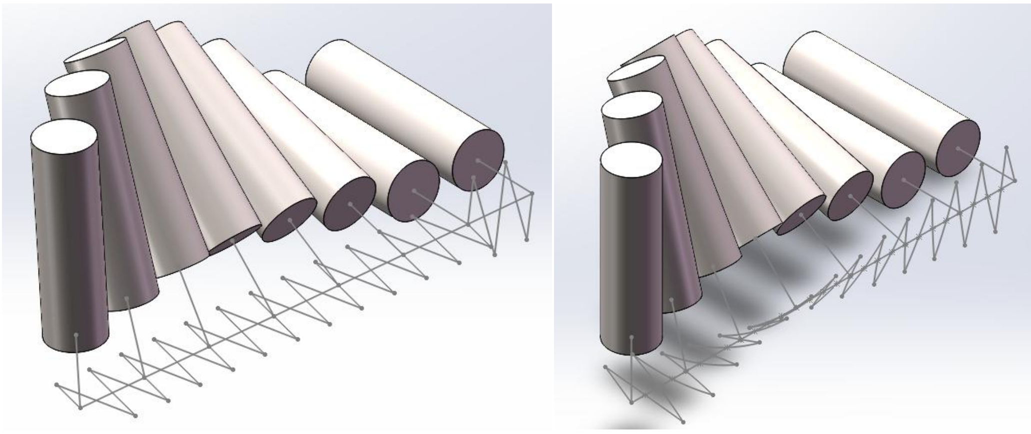
- The welding seam is swayed in an 8-shaped pattern perpendicular to the weld.




### 16.2.3 Arc Placement Parameters

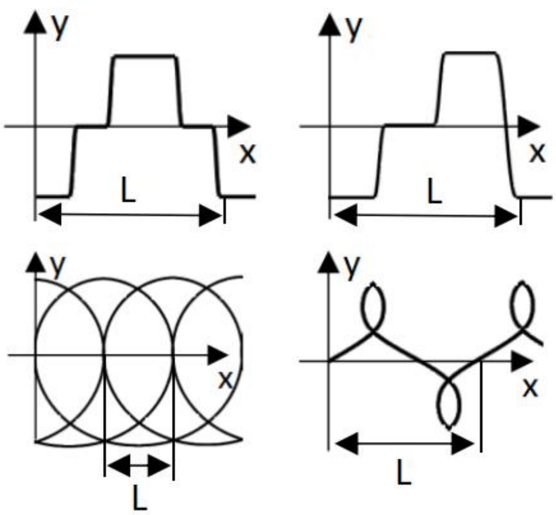
1) Follow orientation

- Whether the arc plane follows the TCP's attitude is determined by the robot's RC command and the attitude register D50.
- The arc plane is only related to the final attitude and does not follow the attitude change in the process.
- The arc plane always follows the attitude change.



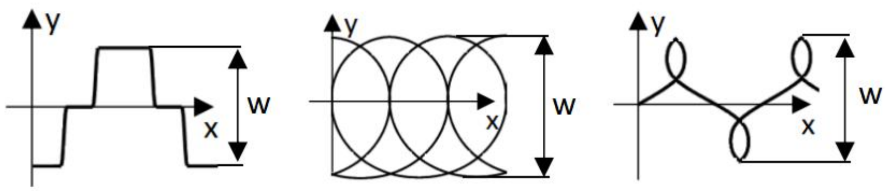
2) Swing length

- The length of a complete arc on the main trajectory X.



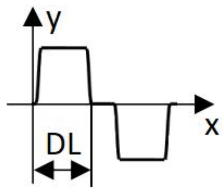
3) Swing amplitude width

- The length of the projection of a complete arc on Y.
- The amplitude of the arc pendulum is the length of the diameter.
- The arc of the 8-character arrangement is also the projection length on the Y-axis.



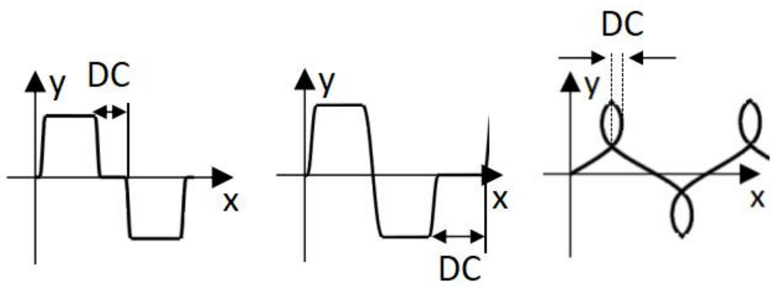

4) Left dwell length dwell_left

- The stop edge length of TCP is only along the direction of the weld at the most left point of the swing.
- Not for arc swing and eight character swing.



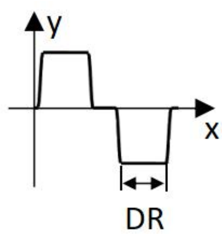

5) dwell_center

- The stop edge length of TCP is only along the direction of the weld at the swing centerline.
- The Z-shaped pendulum and V-shaped pendulum each have two stops at the centerline, while the triangular pendulum has only one stop.
- The shape of the Eight-Character Swing is determined by two parameters: the radius of the upper and lower semicircles, and the swing amplitude.
- Not provided for the pendulum.



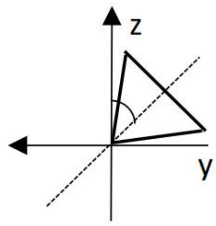

6) Right dwell length dwell_right

- The stop edge length of TCP is only along the direction of the weld at the most right point of the swing.
- Not for arc swing and eight character swing.



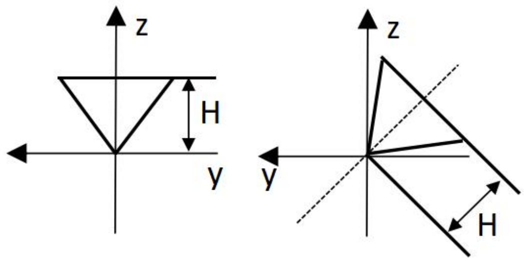
7) The orientation angle of the arc

- The angle between the normal vector of the oscillating plane and the Z-axis of the reference plane.



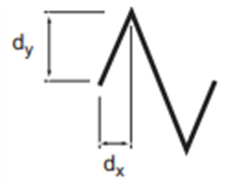

8) Swing height

- The height of a complete swing arc.




9) Restrictions

- Z-type pendulum, V-type pendulum, or triangular pendulum: The pendulum arc slope, defined as the ratio of the Y-axis projection length to the X-axis projection length in a complete pendulum arc, typically maintains dy/dx<10 to limit Y-direction velocity and prevent overspeed.




- Circumferential pendulum: the arc should be greater than the length.
- Bazi Swing: The swing amplitude exceeds 4 times the radius (dwell_center), and the swing length is greater than the swing amplitude.

Currently, in the arc swing parameters, except for the arc swing slope, which is set as a global parameter, all other parameters are located in the point information. Each point can be configured with different arc swing parameters to achieve different shapes for different line segments. The PTP trajectory arc swing mode remains constant at 0.

## 16.3 Functional Specifications and Usage Guidelines

### 16.3.1 Configuration initialization

- S1: Configure the SD5500 register type for processing point data storage.
- S2: Set the initial address SD5588 for the storage register of machining point data.
- S3: Configure the number of SD5400 processing points for the initial distribution.
- S4: Set up the machining tool SD5503.

### 16.3.2 RCPATH instruction data point format

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-d12 | Target point coordinates [x, y, z, q0, q1, q2, q3] | FLOAT |
| D14 | Arc angle, range [0-360]*100 | INT16U |
| D15 | Tool id, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-200 to indicate the percentage of cp time, and set 0 or-1 to indicate the zero-speed point | INT16S |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and attitude ③ | FLOAT |
| D22 | Maximum acceleration percentage of running path and posture | FLOAT |
| D24 | Maximum deceleration percentage of running path and posture | FLOAT |
| D26 | Path transition error percentage | FLOAT |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: Ptp, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: Ptp with joint speed planning (unfixed trajectory). 1: Ptp with joint speed planning (fixed trajectory). 2: Ptp with cartesian speed planning (fixed trajectory). 3: Ptp with acceleration optimization planning (fixed trajectory) ⑤ | INT16U |
| D50 | Whether the arc follows the pose: 0-no, 1-yes | INT16U |
| D51 | Swing mode: 0-no swing, 1-z-shaped swing, 2-v-shaped swing, 3-triangle swing, 4-circular swing, 5-octagonal swing | INT16U |
| D52 | Bend arc length, 1-10mm | FLOAT |
| D54 | Bend amplitude, 1-10mm | FLOAT |
| D56 | Swing height, replacing the original v-type swing angle, 1-10mm | FLOAT |
| D58 | Left retention length: 1-10mm | FLOAT |
| D60 | Retention time, 1-10mm | FLOAT |
| D62 | Left retention length: 1-10mm | FLOAT |
| D84 | Point number, user-defined, monotonically increasing (for easy correlation with sd5404 or log viewing of anomalies) | INT32U |
| D86 | The number of transition points for attitude is up to 5⑥. | INT16U |

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. When the external axis collaborative motion types are 1, 2, or 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to reference coordinate system D18=6, D19=1 (workpiece coordinate system). When setting D18=4 (tool coordinate system), the point position information is relative displacement. For instance: If the robot's tool-end position is (400,200,100) and the point position x, y, z information is set to (50,20,-20), the robot will reach (450,220,80) after movement.

③: When speed, acceleration, or both are not configured, the system automatically adjusts the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points, one being the arc auxiliary point and the other the arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint is the starting point, not the endpoint information in the point input.

⑤: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: speed planning is performed in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: speed planning is performed in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values represent the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑥: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated based on the path length ratio, with the posture change amount from P1 to P6 being equally distributed.

Actuating logic：

- S1: Enter all processing point data into the [Dynamic Point Data Storage Area] (referred to as the data area) according to the format.

- S2: The set position Arc instruction activates register SM3160, first performs instruction execution environment check, then automatically resets.
- S3: The instruction automatically executes in the module sequence, and the lower computer reads all the machining data from the upper computer at once.

Check the command execution environment:

- S1: Initialize the arc module.
- S2: The robot error alarm has been cleared (SM3004 and SM3005 turned off, SD4130 and SD4131 reset to zero).
- S3: The robot is in enable state (SM3040 ON).
- S4: The robot is not in motion (SM3001 is turned off).
- S5: No other motion commands are being executed.

## 16.4 Description of Registers

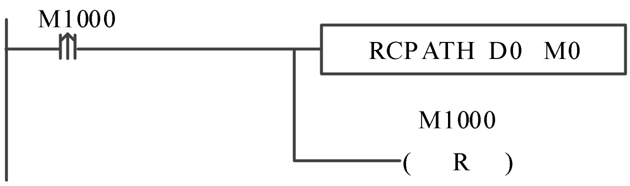
The Arc instruction is based on the RCPATH instruction and therefore includes all registers of the RCPATH instruction. The following registers are added.

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD4861 | INT16U | Maximum slope of the arc | Default: 10 degrees | R/W | Rc will take effect next time |
| SD4862 | Float | Minimum swing arc length<br>If the swing length at the point is less than this value, it will be adjusted to this value.<br>The last swing does not swing when the swing length is less than this value | Default 1mm | R/W | Rc will take effect next time |
| SD4864 | Float | Maximum swing arc length<br>If the swing length exceeds this value, it will be adjusted to this value. | Default: 10mm | R/W | Rc will take effect next time |
| SD4866 | Float | Minimum arc amplitude | Default 1mm | R/W | Rc will take effect next time |
| SD4868 | Float | Maximum arc amplitude | Default: 20mm | R/W | Rc will take effect next time |
| SD4870 | Float | Minimum arc height | Default 1mm | R/W | Rc will take effect next time |
| SD4872 | Float | Maximum arc height | Default: 10mm | R/W | Rc will take effect next time |
| SD4874 | Float | The minimum length of the last swing. If the remaining length is less than this value, no arc will be swung. | Default 1mm | R/W | Rc will take effect next time |
| SD5312 | Float | Swing x-axis displacement to monitor whether the swing direction interpolation is correct | Mm | R | |
| SD5314 | Float | Swing z-axis displacement to monitor whether the swing direction interpolation is normal | Mm | R | |
| SD5316 | Float | Z-axis oscillation velocity | Mm/s | R | |
| SD5318 | Float | Swing y-axis displacement to monitor whether the swing direction interpolation is correct | Mm | R | |
| SD5320 | Float | Y-axis oscillation velocity | Mm/s | R | |
| SD5322 | Float | The maximum angle between the two interpolation points is set. If the angle exceeds this value, an alarm (10095) will be triggered. | 0-10 degrees Default: 5 degrees | R/W | Rc will take effect next time |
| SD5324 | Float | Adjust the swing plane angle of the two interpolation points. If the angle exceeds this set value, the swing plane of the next interpolation point will rotate 180 degrees around the x-axis. | 170-180 degrees, default 175 degrees | R/W | Rc will take effect next time |
| SD5830+2*I (I=0,1,2,3) | Float | Primary trajectory speed, the real-time interpolation speed of the robot tool relative to the reference coordinate system, without superimposed swing arc | Mm/s | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6200+2*I (I=0,1,2...6) | Float | The robot tool's real-time interpolation pose and position display in the rc command mode, relative to the reference coordinate system, is shown with 7 floating-point values: [x, y, z, q0, q1, q2, q3]. The trajectory does not include superimposed swing arcs. In external axis operation mode 1, the relative position is displayed. | | R | |
| SD6220+2*I (I=0,1,2...6) | Float | When the rc command performs arc swinging, the robot tool displays its real-time position and orientation in the relative reference coordinate system [x, y, z, q0, q1, q2, q3], superimposes the arc trajectory, and operates in external axis mode 1 for relative position. | | R | |
| SD6240+2*I (I=0,1,2,3) | Float | Real-time interpolation speed of robot tool relative to reference coordinate system and superposition of swing arc | Mm/s | R | |
| SM3001 | Bit | Robot movement flag | | R | |
| SM3009 | Bit | Instruction completion flag | | R | |
| SM3171 | Bit | The robot is performing arc interpolation with the flag set to 1. After the interpolation calculation starts, the flag is set to 0. | | R | |
| SM3173 | Bit | Arc instruction error in multi-stage pendulum | | R | |

The SM3171 timing is identical to the RCPATH instruction, while the SM3165 timing follows the same pattern.

## 16.5 Function Applications




The source file sample program is as follows:

```
#include"RCPATH.h"

#define pNum 4
RcPointInformation RcPoint[pNum] = {
    {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,3,1,1,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0},
{{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,3,1,1,0,1,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0},
{{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,3,1,1,0,1,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0},
}.

```

 User Manual of RC6F Series Robot Controller 16 Arc Function

```
void RCPATH(PINT16S W, BIT B)
{
  if (SD[5500] == 0)
  {
    for (int i = 0. i < SD[5401]. i++)
    {
// Transmit pose data
      *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
      *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
      *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
      *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
      *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
      *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q2.
      *(float*)&D[(SD[5588] + 100 * i + 14)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 14)] = RcPoint->weaveAngleZ. // Arc angle
D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType.//hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
D[(SD[5588] + 100 * i + 50)] = RcPoint->weaveFollow. //Arc-following gesture
D[(SD[5588] + 100 * i + 51)] = RcPoint->weaveMode. //Swing arc mode
*(float) & D[(SD[5588] + 100 * i + 52)] = RcPoint->weaveLength. //Arc length
*(float)* &D[(SD[5588] + 100 * i + 54)] = RcPoint->weaveWidth. //Arc width
*(float) & D[(SD[5588] + 100 * i + 56)] = RcPoint->weaveHeight. //Arc height
*(float)* &D[(SD[5588] + 100 * i + 58)] = RcPoint->weaveLdWell. //Left dwell length of swing arc
*(float)* &D[(SD[5588] + 100 * i + 60)] = RcPoint->weaveCdWell. // Right arc dwell length
*(float)* &D[(SD[5588] + 100 * i + 62)] = RcPoint->weaveRdWell. // dwell length during arc swing
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
    }
  }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
}
```

# 17. External axis functions

## 17.1 Function Overview

The external axis is a device that creates an optimal workspace between robots and workpieces, primarily functioning for collaborative operations such as rotational and translational coordination. It can be modeled as a robot, and calibrating the external axis is equivalent to calibrating the robot's workpiece. The external axis's motion commands are executed through RCPATH instructions. After configuring the relevant axis parameters in the RCPATH instruction's point attributes, the system will perform the axis movement upon command execution.

## 17.2 External Axis System Configuration

| Type | Content |
|---|---|
| External shaft system serial number | 0-2: Default 0 indicates no external axes, supports up to 2 external axis systems |
| Number of external shafts | The number of single axes for an external axis system, up to 3 |
| Single axis kinematic pair type | Rotation/translation, supporting dual-axis positioner with ground track, and three-axis/two-axis gantry types |
| Axial controlled configuration | The actual composition of the external shaft system is determined by whether the control of the shaft is assigned. |

- External axes are typically assigned starting from axis 7 (function mapping number 6). They feature soft limit switches, with their default settings usually lower than the electrical limit of bus axes.
- The system supports two external axis systems: the first (7-8-9) and the second (10-11-12), with the robot's main axis numbered 1-6.
- An external axis system supports up to three fixed axes: Positioner 1, Positioner 2, and the ground rail axis. If no ground rail is present, configure the ground rail axis as an uncontrolled SFD5370+i. For single-axis positioners, positioner 2 should be set as uncontrolled. When only a ground rail is present, both positioners 1 and 2 must be configured as uncontrolled. The feedback joint position of the uncontrolled axis remains at 0, rendering it unresponsive to motion commands. For example, jogging the uncontrolled axis is invalid and will not trigger alarms (e.g., 10140).
- Common orbital configuration: the first external axis system is controlled, while the second is not. the maximum number of accessible axes is 5.
- The three-axis gantry type external axis enables coordinated motion between the gantry and the robot, while the two-axis gantry only supports uncontrolled setting of axis 3.

| Number of shafts in the external shaft system | Axis composition | Configuration |
|---|---|---|
| 1 | 1 geostationary orbit | The outer shaft system shaft 1 and shaft 2 are not controlled, but shaft 3 is controlled. |
| | 1 variable-speed machine shaft | The outer shaft system is controlled by shaft 1, while shafts 2 and 3 are not controlled.<br>Or the external shaft system shaft 2 is controlled, while shaft 1 and shaft 3 remain uncontrolled. |

| | | |
|---|---|---|
| 2 | 1 variable shaft, 1 ground rail | The outer shaft system is controlled by shaft 1 and shaft 3, but not by shaft 2.<br>Or the external shaft system (axles 2 and 3) is controlled while shaft 1 remains uncontrolled. |
| | 2 variable-speed machine shafts | The outer shaft system is controlled by shaft 1 and shaft 2, but not by shaft 3. |
| 3 | 2 variable-speed machine shafts, 1 ground rail | The outer shaft system (shaft 1, shaft 2, shaft 3) is controlled. |

The external axis is set as a simulation axis, which still responds to motion commands and participates in trajectory movement. The axis is set as an uncontrolled axis, which does not respond to motion commands and does not participate in trajectory movement.

## 17.3 External axis calibration

External axis calibration comprises two procedures: ground track calibration and positioner calibration. The current system only supports single-axis ground track calibration, with two external axis systems permitted to share a single ground track but not operate on separate ground tracks. Precise calibration of the positioner's relationship with the world coordinate system requires prior completion of ground track calibration and correction. The automatic reset function for calibration control bits indicates active calibration computation, while failure to reset after activation confirms absence of calibration process.

During calibration, each axis must move in the joint's forward direction. Prior to calibration, verify the rotation direction of all axes in the positioner and the logic of the ground rail's travel direction (using the right-hand spiral rule). Ensure the speed reduction ratios between the positioner and ground rail are set to theoretical values. The external axis system sequence is fixed as: positioner axis 1, positioner axis 2, ground rail. If the actual physical axis connection differs, first modify the EtherCAT function mapping numbers of the corresponding axes to match this sequence. During positioner calibration, the total rotation angle of the three points must not exceed 360 degrees. External axis calibration depends on accurate tool positioning, requiring sufficiently high tool calibration precision. Ensure the positioner carries correct tool parameters during calibration and uses the base coordinate with tools. When selecting calibration points for positioner, position them as far away from the axis rotation center as possible—closer to the center increases fitting axis errors caused by coordinate inaccuracies. During positioner and ground rail calibration, avoid changing tool positions to prevent tool errors introduced by posture changes. Do not move the ground rail during positioner calibration, nor move the positioner during ground rail calibration. Positioner axis 1 can start from negative angles, but axis 2 must be at zero position. Ensure the travel direction remains positive. After collecting positioner calibration points, the robot may not be at zero position. When axis 1 and 2 of the external axis system return to zero position (with ground rail present), SM3202 and SM3207 flags can be triggered. After writing parameters to the SFD, re-power the system to activate.

### 17.3.1 Ground Track Calibration

1) Configure the SD5593 (single-word) external axis system serial number. Each positioner and ground rail forms an external axis system. Under current specifications, two external axis systems (External Axis System 1 and External Axis System 2) are supported. During calibration, the positioner and ground rail in the actual external axis system must match the system serial number.
2) The ground track is calibrated using the two-point method, with a fixed point on the positioner designated as

the calibration point P. Verify that the robot tool ID in SD4015 matches the calibration tool, align the fixed point P at the robot tool's end, and record the robot base coordinate system position as P1. The X, Y, Z coordinates of the SD4000-SD4004 robot tool's end, along with the first three external axis angles (SD5620+2i+6* (SD5593-1)), are stored in the six float registers starting from SD5720.

3) The ground track is driven to move forward (actual movement exceeding 200mm), which determines its forward direction. The robot tool's fixed point P is realigned, and the robot base coordinate system's position is recorded as P2: the X, Y, and Z coordinates of the SD4000-SD4004 robot tool's end, along with the first three external axis angles (SD5620+2i+6* (SD5593-1)) stored in the six float registers starting from SD5732.

4) After completing the two calibration point recordings, the SM3201 must be activated when the robot is enabled to obtain the ground track calibration parameters and deceleration ratio correction coefficient [x, y, z, k3]. Here, [x, y, z] represents the ground track's movement unit vector in the world coordinate system, and k3 is the actual deceleration ratio. The calibration parameters are automatically stored in the first four float registers of the SD5870. During calibration, alarms may occur at 10141/10143/10148/10162. refer to the error code explanation for details.

5) Ground track specification: The dual external axis system supports only one shared ground track, requiring only one calibration. If no ground track is present, calibration is unnecessary. In such cases, verify that the first three floating-point bits of the SFD6040 in the ground track must all be 0, or configure the ground track as uncontrolled.

## 17.3.2 Transducer Calibration (Method 1)

### 17.3.2.1 Calibration of the dual-axis positioner

1) Configure the SD5593 (single-word) external axis system serial number. Each positioner and ground rail forms an external axis system. Under current specifications, two external axis systems (External Axis System 1 and External Axis System 2) are supported. During calibration, the positioner and ground rail must match the external axis system serial number. The ground rail must remain stationary during positioner calibration, and one axis must be at the joint zero position while the other axis is calibrated.
2) The positioner is calibrated using the six-point axis measurement method, with the non-calibrated axis at zero position and the calibrated axis at arbitrary position. A fixed point on the positioner is designated as the calibration point P. Check if the robot tool number in (INT16U)SD4015 matches the calibrated tool. If (INT16U)SD4018=0, set the reference coordinate system to the robot base coordinate system. Move the robot tool end to the calibration point P and record its position relative to the robot base coordinate system as P1. The X, Y, and Z coordinates of the robot tool end (SD4000-SD4004) and the three external axis angles starting from SD5620+2i+6* (SD5593-1) are stored in the six float registers starting from SD6040.
3) Rotate the first axis of the positioner forward by a specific angle (actual movement exceeding 15 degrees, with clockwise rotation as positive direction and the Z-axis perpendicular to the positioner plane). Position the robotic tool tip at the target point P, where its position relative to the robot base coordinate system is denoted as P2. The X, Y, and Z coordinates of the robotic tool tip (SD4000-SD4004) along with the three external axis angles starting from SD5620+2i+6* (SD5593-1) are stored in the six float registers beginning at SD6052.
4) Subsequently, rotate the first axis forward by a specific angle (actual movement exceeding 15 degrees, with rotation direction defined by the right-hand rule, where the Z-axis of the positioner is perpendicular to its plane and oriented upward) to position the robotic tool tip at the target point P. The position of this point relative to the robot base coordinate system is designated as P3. The X, Y, and Z coordinates of the robotic tool tip (SD4000-SD4004) along with the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6064.
5) Rotate the first axis of the manipulator forward by a specific angle (greater than 15 degrees, with the clockwise direction as positive, and the manipulator's Z-axis perpendicular to the manipulator plane and oriented upward) to position the robotic tool end at the target point P. Record the position of this point relative to the robot base coordinate system as P4: SD4000-SD4004 (X, Y, Z coordinates of the robotic tool end), along with the three external axis angles starting from SD5620+2i+6* (SD5593-1), and store them in the six float registers starting from SD6076.
6) Subsequently, rotate the first axis forward by a specific angle (greater than 15 degrees, with the clockwise direction as positive, and the Z-axis of the positioner oriented vertically upward relative to its plane) to position the robotic tool tip at the target point P. Record the position of this point relative to the robot base coordinate system as P5: the X, Y, and Z coordinates of the robotic tool tip (SD4000-SD4004), along with the three external axis angles starting from SD5620+2i+6* (SD5593-1), and store them in the six float registers starting from SD6088.
7) Rotate the first axis of the positioner forward by a specific angle (greater than 15 degrees, with the clockwise direction as positive, and the Z-axis perpendicular to the positioner plane). Position the robotic tool end at the target point P, denoted as P6. The X, Y, and Z coordinates of the tool end (SD4000-SD4004) and the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6100. The six calibration points required for single-axis calibration have been completed. During the entire calibration process of the first axis, the other external axes remain stationary.

8) Similarly, position all axes of the positioner to zero, designate a fixed point on the positioner as the calibration point P, and verify that the robot tool number in SD4015 matches the calibration tool. Move the end effector of the robot tool to the calibration point P, recording its position relative to the robot base coordinate system as P7: the X, Y, and Z coordinates of the SD4000-SD4004 robot tool end effector, along with the three external axis angles starting from SD5620+2i+6* (SD5593-1), which are stored in the six float registers starting from SD6112.
9) Rotate the second axis of the positioner forward by a specific angle (actual movement exceeding 15 degrees, with clockwise rotation as positive direction, and the Z-axis oriented vertically upward relative to the positioner plane) to position the robotic tool tip at the target point P. Record the position of this point relative to the robot base coordinate system as P8: The X, Y, and Z coordinates of the robotic tool tip (SD4000-SD4004) and the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6124.
10) Subsequently, rotate the second axis forward by a specific angle (actual movement exceeding 15 degrees, with rotation direction defined by the right-hand rule, where the Z-axis of the positioner is perpendicular to its plane and oriented upward) to position the robotic tool tip at the target point P. Record the position of this point relative to the robot base coordinate system as P9: SD4000-SD4004 (X, Y, Z coordinates of the robotic tool tip), along with the three external axis angles starting from SD5620+2i+6* (SD5593-1), and store them in the six float registers starting from SD6136.
11) Rotate the second axis of the positioner forward by a specific angle (greater than 15 degrees, with clockwise rotation as positive direction, and the Z-axis perpendicular to the positioner plane). Position the robotic tool end at the target point P, denoted as P4 in the robot base coordinate system. The X, Y, and Z coordinates of the robotic tool end (SD4000-SD4004) along with the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6148.
12) Next, rotate the second axis forward by a specific angle (greater than 15 degrees, with the clockwise direction as positive, and the Z-axis perpendicular to the manipulator's plane with the Z-axis pointing upward). Position the robotic tool end at the target point P, denoted as P5 in the robot base coordinate system. The X, Y, and Z coordinates of the tool end (SD4000-SD4004) and the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6160.
13) Rotate the second axis of the positioner forward by a specific angle (greater than 15 degrees, with the clockwise direction as positive, and the Z-axis perpendicular to the positioner plane). Position the robotic tool end at the target point P, denoted as P6 in the robot base coordinate system. The X, Y, and Z coordinates of the tool end (SD4000-SD4004) and the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6172. The six calibration points required for single-axis calibration have been completed. During the entire calibration process of the second axis, the other external axes remain stationary.
14) When the robot is enabled, set SM3202 to 1. Based on the calibration data, calculate the positioner's DH parameters and the reduction ratio correction coefficient SD5880. The six float registers [a, d, alpha, theta, k1, k2] starting from SD5880 and the six float registers TW_PB starting from SD5900 are then calculated. Here, [a, d, alpha, theta] represent the DH parameters of the positioner's second axis, while [k1, k2] indicate the actual reduction ratios of the two axes. During calibration, alarms 10142/10143/10149/10162/10163 may occur. For details, refer to the error code explanation.
15) The user will write the calibration results SD5870+2i, SD5880+2i, and SD5900+2i into the corresponding SFD6040+2i, SFD6050+12* (SD5593-1), and SFD5752+12* (SD5593-1) based on the external axis system.
16) Write the actual calibrated reduction ratio into SFD6080+6* (SD5593-1).

Explanation: When the joint is at zero position, the robot's base coordinate system aligns with the world coordinate system. The positioner's base coordinate system matches its axis 1 coordinate system, with the

DH parameter of axis 1 fixed at [0,0,0,0] requiring no calibration. After external axis calibration, the calibration accuracy and multi-axis synchronization performance can be verified through point-to-point collaborative motion.

### 17.3.2.2 Calibration of Single-axis Positioning Machine

1) Configure the SD5593 (single-word) external axis system serial number. Each positioner and ground rail forms an external axis system. Under current specifications, two external axis systems (External Axis System 1 and External Axis System 2) are supported. During calibration, the positioner and ground rail must match the external axis system serial number. The ground rail must remain stationary during calibration, and only the axis under calibration should be activated.
2) Calibrate the positioner using the three-point axis measurement method to align its single axis at zero position. Designate a fixed point on the positioner as the calibration point P. Verify that the robot tool ID in (INT16U)SD4015 matches the calibrated tool. If (INT16U)SD4018=0, set the reference coordinate system to the robot base coordinate system. Move the robot tool end to the calibration point P and record its position relative to the robot base coordinate system as P1. Save the X, Y, Z coordinates of the robot tool end (SD4000-SD4004) and the three external axis angles (SD5620+2i+6* (SD5593-1)) into the six float registers starting from SD6040.
3) Rotate the manipulator's single-axis forward by a specific angle (greater than 15 degrees, with clockwise rotation as positive direction and the Z-axis perpendicular to the manipulator's plane). Position the robotic tool end at the target point P, denoting its position relative to the robot base coordinate system as P2. The X, Y, and Z coordinates of the robotic tool end (SD4000-SD4004) along with the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6052.
4) Subsequently, rotate the single-axis in the forward direction by a specific angle (greater than 15 degrees, with the rotation direction defined by the right-hand rule, where the Z-axis of the positioner is perpendicular to its plane and oriented upward). Position the robotic tool end at the target point P, and record its position relative to the robot base coordinate system as P3. The X, Y, and Z coordinates of the robotic tool end (SD4000-SD4004) along with the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6064.
5) Rotate the manipulator's single-axis forward by a specific angle (greater than 15 degrees, with clockwise rotation as positive direction and the Z-axis perpendicular to the manipulator plane). Position the robotic tool tip at the target point P, where its coordinates relative to the robot base frame are denoted as P4: The X, Y, and Z positions of the tool tip (SD4000-SD4004) and the three external axis angles (SD5620+2i+6* (SD5593-1)) are stored in the six float registers starting from SD6076.
6) Subsequently, rotate the single-axis in the forward direction by a specific angle (greater than 15 degrees, with the rotation direction defined by the right-hand rule, where the Z-axis of the positioner is perpendicular to its plane and oriented upward). Position the robotic tool end at the target point P, and record its position relative to the robot base coordinate system as P5. The X, Y, and Z coordinates of the robotic tool end (SD4000-SD4004) along with the three external axis angles starting from SD5620+2i+6* (SD5593-1) are stored in the six float registers starting from SD6088.
7) Rotate the manipulator's single-axis forward by a specific angle (greater than 15 degrees, with clockwise rotation as positive direction and the Z-axis perpendicular to the manipulator plane). Position the robotic tool end at the target point P, denoted as P6. The X, Y, and Z coordinates of the tool end (SD4000-SD4004) along with the three external axis angles (SD5620+2i+6* [SD5593-1]) are stored in the six float registers starting from SD6100. The six calibration points for single-axis alignment are now complete. During the entire single-axis calibration process, no movement occurs in the other external axes.

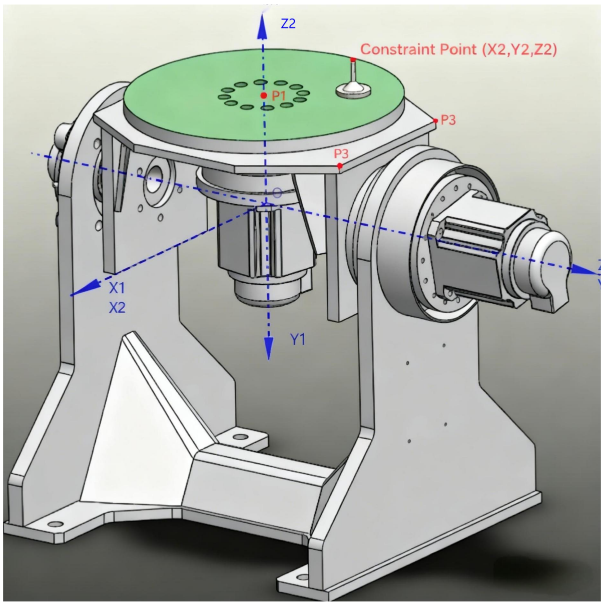
8) When the robot is enabled, set SM3202 to 1. Based on the calibration data, calculate the positioner's DH parameters and the reduction ratio correction coefficient SD5880. The six float registers [a, d, alpha, theta, k1, k2] starting from SD5880 and the six float registers TW_PB starting from SD5900 are then calculated. Here, [a, d, alpha, theta] represent the positioner's DH parameters (all set to 0), while [k1, k2] indicate the actual reduction ratios of the two axes (the uncontrolled axis's actual reduction ratio is 0). During calibration, alarms 10143/10149/10162/10163 will be triggered. For details, refer to the error code explanation.
9) The user will write the calibration results SD5870+2i, SD5880+2i, and SD5900+2i into the corresponding SFD6040+2i, SFD6050+12* (SD5593-1), and SFD5752+12* (SD5593-1) based on the external axis system.
10) Write the actual calibrated reduction ratio into SFD6080+12* (SD5593-1).

When recalibrating using the saved reference points, ensure that the tool's mechanical structure remains unchanged. Before recalibration, adjust the actual reduction ratio of the external shaft (SFD6080) to the value used during the reference point teaching.

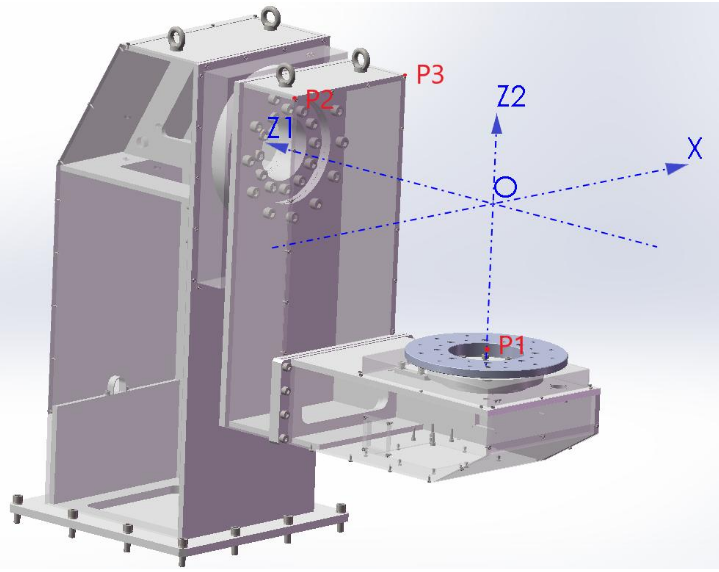
### 17.3.3 Transducer Calibration (Method 2)

The calibration of the positioner error model primarily involves using the robot as a measurement tool. By modeling the positioner error model, ten parameters of the positioner are identified to obtain an accurate kinematic description from the world coordinate system to the positioner's two-axis coordinate system. The calibration process collects eight points in total, including three initial points (when the positioner is at zero position) and five additional points collected during the calibration process. Note that the constraint points are measured in millimeters (mm), which should be carefully noted when writing the data.

P 掉中文，将图片中文翻译为英文并 P 到图片上，并生成一张图片，坐标不变，保留




instance 1




instance 2

### 17.3.3.1 Before the calibration begins

1) Set the SD5593 (single-word) external axis system serial number. Each positioner and ground rail forms an external axis system. Under current specifications, two external axis systems (External Axis System 1 and External Axis System 2) are supported. During calibration, the positioner and ground rail must match the external axis system serial number. The ground rail must remain stationary during the calibration process.
2) Confirm that the selected tool number in the teaching device matches the tool currently attached to the robot's end effector. All subsequent measurements will be performed with the tool in the base coordinate system.
3) The Z-axis orientation (initial axis movement direction) of the positioner coordinate system is defined as follows: For Positioner 1, the Z-axis extends from the non-motor side toward the motor side. For Positioner 2, the Z-axis points upward (as shown in Example 1 and Example 2 above), with counterclockwise rotation around the Z-axis defined as positive. Rotate both Positioner axes 1 and 2 separately to verify if the initial axis movement direction conforms to the specified orientation. If discrepancies are detected, the corresponding axis movement initial direction register must be modified first.
4) After verifying the initial axis orientation, reset the positioner to zero. For the 0/1/2 axis coordinate system of the positioner, define the X-axis orientation as follows: X0/X1/X2 equals Z1 cross multiplied by Z2, following the right-hand rule—where the right-hand index finger moves from Z1 to Z2, and the thumb points in the X0/X1/X2 direction (during positioner modeling, X0/X1/X2 coincide, hence collectively referred to as the positioner's X-axis, as shown in Example 1 and Example 2 above).
5) When the positioner is at zero position, it initiates the acquisition of the first calibration point P1. The function and measurement method of P1 are as follows: P1 represents the initial position value of the positioner's base coordinate system in the world coordinate system. During P1 measurement, the robot end effector should be positioned as close as possible to the intersection point of the positioner's 1/2 axis (due to mechanical constraints, approximate alignment suffices. see Example 1 and Example 2 in the figure above). The X, Y, and Z coordinates of the SD4000-SD4004 robot tool end effector are recorded at this point and stored in the SD6040+2*i register.
6) After completing the initial measurement, the positioner resets to zero and proceeds to measure points P2 and P3. The measurement procedure for P2/P3 is as follows: These points are primarily used to determine the initial angle between the positioner's X-axis and the global coordinate system's $\overrightarrow{P2P3}$ $\overrightarrow{P2P3}$ X-axis. The

measurement steps are: Select a side of the positioner's mechanical structure parallel to its X-axis, designating P2 as the starting point and P3 as the endpoint, ensuring the vector aligns with the positioner's X-axis direction. The robot sequentially measures points P2 and P3 (order must not be reversed, as shown in Example 1 and Example 2 above), recording the X, Y, and Z coordinates of the SD4000-SD4004 robot tool end at each point. These coordinates are then written into the SD6046+2*i and SD6052+2*i registers, respectively.

7) The user must calculate the position coordinates X2/Y2/Z2 of the fixed constraint point on the positioner in its 2-axis coordinate system, and store them in the corresponding SD6058+2*i register (as shown in Example 1 above).

The constraint points must be written before the five-point method calibration.

After the above calibration work is completed, the error model calibration of 5-point method displacement machine can be started.

### 17.3.3.2 Start the calibration process

The 5-point method error model calibration operation is summarized as follows: operate the 1/2 axis of the positioner to combine 5 sets of different joint angles, operate the robot to point, after the completion of the point, record the corresponding robot joint pulse value, positioner joint pulse value, ground track feedback angle for calculation.

The following principles apply to the operation process:

- When the manipulator manipulates the point, the end-effector posture should be kept unchanged as far as possible.
- The five points collected should be distributed in the larger range of the joint angles of the 1st and 2nd joints of the manipulator (180° for the 1st axis and 360° for the 2nd axis).
- The sampling point should not be at the zero position of the 1/2 axis.

The recommended angles for the mechanism joints of the five point groups are:

| | $\theta_1$ (°) | $\theta_2$ (°) |
|---|---|---|
| 0.1 | -90 | -90 |
| 0.2 | -50 | -20 |
| 0.3 | -10 | 50 |
| 0.4 | 40 | 120 |
| 0.5 | 90 | 200 |

The specific sampling procedure is as follows:

1) Rotate the 1/2 shaft of the transformer to-90° and-90° respectively (approximate, no precision required).

2) The manipulator moves along Cartesian coordinates X/Y/Z, progressively approaching the constraint point. When nearing the constraint point, the manipulator adjusts the manipulator's single-step distance to the minimum step size, ensuring the end-effector tool point aligns as closely as possible with the constraint point. The manipulator records joint pulses at the current position (HSD104/HSD124/HSD144/HSD164/HSD184/HSD204) and stores them in the SD6070+2*j+10*i register, where j=0 and i=0,1,2,3,4,5. The positioner's joint pulses at the current position are recorded and stored in the

SD6130+2*j+10*i register, where j=0 and i=0,1. The ground track feedback angle at the current position (SD5624) is recorded and stored in the SD6150+2*j register, where j=0.

3） Move the manipulator along the Cartesian axes X/Y/Z to a distance from the constraint point. Rotate the 1/2-axis of the positioner to-50° and-20° respectively (approximate values are acceptable).

4） The manipulator moves along Cartesian coordinates X/Y/Z, progressively approaching the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm to ensure the end-effector tool point aligns as closely as possible. Record the joint pulses of the manipulator at the current position (HSD104/HSD124/HSD144/HSD164/HSD184/HSD204) and store them in the SD6070+2*j+10*i register (where j=1, i=0,1,2,3,4,5). Record the positioner's joint pulses at the current position and store them in the SD6130+2*j+10*i register (where j=1, i=0,1). Record the ground track feedback angle at the current position (SD5624) and store it in the SD6150+2*j register (where j=1).

5） Move the manipulator along the Cartesian axes X/Y/Z to a distance from the constraint point. Rotate the 1/2-axis of the positioner to angles of-10° and 50° (approximate values are acceptable).

6） The manipulator moves along Cartesian coordinates X/Y/Z, progressively approaching the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm to ensure the end-effector tool point aligns as closely as possible. Record the joint pulses of the manipulator at the current position (HSD104/HSD124/HSD144/HSD164/HSD184/HSD204) and store them in the SD6070+2*j+10*i register (where j=2, i=0,1,2,3,4,5). Record the positioner's joint pulses at the current position and store them in the SD6130+2*j+10*i register (where j=2, i=0,1). Record the ground track feedback angle at the current position (SD5624) and store it in the SD6150+2*j register (where j=2).

7） Move the manipulator along the Cartesian axes X/Y/Z to a distance from the constraint point. Rotate the 1/2-axis of the positioner to angles of 40° and 120° (approximate values are acceptable).

8） The manipulator moves along Cartesian coordinates X/Y/Z, progressively approaching the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm to ensure the end-effector tool point aligns as closely as possible. Record the joint pulses of the manipulator at the current position (HSD104/HSD124/HSD144/HSD164/HSD184/HSD204) and store them in the SD6070+2*j+10*i register (where j=3, i=0,1,2,3,4,5). Record the positioner's joint pulses at the current position and store them in the SD6130+2*j+10*i register (where j=3, i=0,1). Record the ground track feedback angle at the current position (SD5624) and store it in the SD6150+2*j register (where j=3).

9） Move the manipulator along the Cartesian axes X/Y/Z to a distance from the constraint point. Rotate the 1/2-axis of the positioner to angles of 90° and 200° (approximate values are acceptable).

10） The manipulator moves along Cartesian coordinates X/Y/Z, progressively approaching the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm to ensure the end-effector tool point aligns as closely as possible. Record the joint pulses of the manipulator at the current position (HSD104/HSD124/HSD144/HSD164/HSD184/HSD204) and store them in the SD6070+2*j+10*i register (j=4, i=0,1,2,3,4,5). Record the positioner's joint pulses at the current position and store them in the SD6130+2*j+10*i register (j=4, i=0,1). Record the ground track feedback angle at the current position (SD5624) and store it in the SD6150+2*j register (j=4).

11） Point data collection is complete. Reset the positioner to zero.

### 17.3.3.3 Parameter Identification and Compensation

1） During PLC operation, set SM3207=1. The DH parameters and speed reduction correction values of the positioner are calculated from the calibrated data. The lower-level system writes these values into the first six float registers [a, d, alpha, theta, k1, k2] of SD5880. The position and orientation of the positioner's base coordinate system relative to the world coordinate system are recorded in register SD5900+2i, where [a, d, alpha, theta] represent the DH parameters of the positioner's second axis, and [k1, k2] indicate the actual speed reduction ratios of the two axes.

2) The positioner calibration errors (MaxError, MinError, and MeanError) are written into the SD6190+2*i register and output to the user terminal. The validity of the current positioner calibration is confirmed based on whether MaxError is less than 1.
3) If this calibration is valid, the user will write the calibration results SD5880+2i and SD5900+2i to the corresponding SFD6050+12* (SD5593-1) and SFD5752+12* (SD5593-1) according to the external axis system.
4) If this calibration is valid, the user will write the actual calibrated reduction ratio into SFD6080+6* (SD5593-1).
5) After the calibration of the positioner, the calibration accuracy and multi-axis synchronization performance can be checked by the external axis point motion.

### 17.3.4 Dragon Gate Calibration

1) Set the SD5593 (single word) external axis system serial number, and set SFD (6200+SD5593-1)=1 to calibrate the external axis system as gantry type. The current specifications support two external axis systems: External Axis System 1 and External Axis System 2.
2) The three axes of the gantry are calibrated using the two-point method, with a fixed point P established as the calibration reference relative to the world coordinate system. Verify that the robot tool ID in SD4015 matches the calibration tool, and ensure the tool's end fixed point P aligns with the reference. Record the robot base coordinate system position as P1: the X, Y, and Z coordinates of the SD4000-SD4004 robot tool end, along with the three external axis angles starting from SD5620+2i+6* (SD5593-1), and store them in the six float registers starting from SD5720.
3) Move the axis 1 of the gantry in the forward direction (actual movement exceeding 200mm, with the forward direction aligned with the X/Y axes of the robot base coordinate system). This movement will determine the forward direction of axis 1. Realign the tool end fixed point P, recording the robot base coordinate system position as P2: the X, Y, and Z coordinates of the SD4000-SD4004 robot tool end, along with the three external axis angles starting from SD5620+2i+6* (SD5593-1), which are stored in the six float registers starting from SD5732.
4) Move the axis 2 of the gantry in the forward direction (actual movement exceeding 200mm, with the forward direction aligned with the X/Y axes of the robot base coordinate system). This movement will determine the forward direction of axis 2. Realign the tool end fixed point P, recording the robot base coordinate system position as P3: the X, Y, and Z coordinates of the SD4000-SD4004 robot tool end, along with the three external axis angles starting from SD5620+2i+6* (SD5593-1), which are stored in the six float registers starting from SD5744.
5) Move the axis 3 of the gantry in the forward direction (actual movement exceeding 200mm, with the forward direction aligned with the X/Y axes of the robot base coordinate system). This movement will determine the forward direction of axis 3. Realign the tool end fixed point P, recording the robot base coordinate system position as P4: the X, Y, and Z coordinates of the tool end (SD4000-SD4004), along with the three external axis angles starting from SD5620+2i+6* (SD5593-1), which are stored in the six float registers starting from SD5756.
6) After completing four calibration point recordings, the SM3201 must be activated during robot enable to obtain gantry calibration parameters and speed reduction correction coefficients [x1, y1, z1, k1. x2, y2, z2, k2. x3, y3, z3, k3]. Here, [xi, yi, zi] (i=1, 2, 3) denotes the world coordinate system movement unit vectors for each gantry axis, while ki represents the actual speed reduction ratio of each axis. The calibration parameters are automatically stored in the first 12 float registers of the SD5800. During calibration, alarms may occur at 10141/10143/10167/10162. refer to the error code explanation for details.
7) After verifying the calibration results, the user writes the 12 floating-point parameters starting from SD5800 into SFD6210+24* (SD5593-1), and simultaneously writes the three gear ratios calibrated by SD5806, SD5814, and SD5822 into SFD6080+6* (SD5593-1).
8) Based on the SFD (6202+SD5593-1) configuration, assign values to SFD5752+12* (SD5593-1).

- When all axes of the gantry are at the zero position, the gantry base coordinate system coincides with the world coordinate system. The SFD6202 robot can be configured for installation on the gantry in two modes: upright or inverted. In upright mode, the gantry base coordinate system aligns with the robot base coordinate system when the gantry is at zero position. In inverted mode, the alignment is reversed. The gantry axes may not be mutually perpendicular, but they must maintain a certain degree of straightness.
- World Coordinate System Pose Monitoring: To monitor the SD4000's feedback position, set SD4018=1, where the coordinate value indicates the world coordinate system. Note: SD5593 must be preconfigured. If External Axis System 1 consists of a positioner and ground rail, and External Axis System 2 is a gantry, different SD5593 values will correspond to different world coordinate systems. This implies that when a gantry is present in both external axis systems, the world coordinate system is not unique.

### 17.3.5 Reset and Body Parameter Calibration

The zero position correction value of the positioner is calculated by the external axis calibration process. Like the robot zero position calibration, the external axis system is returned to the zero position first. The zero position of the positioner axis 1 and the ground rail axis is determined by the user, and the zero position of the positioner axis 2 is determined by the calibration result.

1) Correction of the deceleration ratio in the Earth orbit

The deceleration ratio is corrected at the zero position of the ground orbit using correction coefficient k3 (SD5876 floating point), yielding the corrected deceleration ratio parameter for the ground orbit.

2) Zero position correction of the ground orbit

The ground orbit zero position is user-defined and requires no calibration. Before calibration, the ground orbit must be moved to the correct position, and the corresponding axis HSD104+20*i (INT64S) should be set to 0.

3) Correction of the speed reducer ratio of the transformer

To calibrate the speed reduction ratio at the zero position of the positioner, use correction coefficients k1 and k2 (SD5888 and SD5890 floating-point). The corrected speed reduction ratio parameters are then written into the corresponding registers.

4) Zero position calibration of the positioner

After completing the positioner calibration, return to its zero position. No adjustment is required for the zero position of shaft 1. instead, set the zero position of shaft 2. The calculated value is obtained by: HSD[264+60* (SD5593-1)] + SFD[6056+12* (SD5593-1)] / 360* 131072* shaft 2's actual reduction ratio, and write this value into HSD[264+60* (SD5593-1)].

## 17.4 External axis point movement commands

The external axis supports only single-step and continuous-point motions of joints, not real-time speed changes. Trigger the corresponding SM register, which is the same as the robot joint motion method. The stop time of continuous-point motion is not determined by the acceleration of the point motion, but by the slow-stop deceleration time register SD4171.

When the external axis is enabled, the system does not check if the current joint angle exceeds the soft limit. The RCPATH command checks whether the current position exceeds the limit. If a joint angle exceeds the soft limit range, it cannot be moved further but can be moved to the zero position. Both single-step and continuous point motions are supported.

| Address | Definition | Read-write | Trigger time |
|---|---|---|---|
| SM3300 | External shaft system: Shaft 1, forward point motion | R/W | With immediate effect |
| SM3301 | External shaft system: Shaft 1, negative point movement | R/W | With immediate effect |
| SM3302 | External shaft system: Shaft 1, shaft 2, forward point motion | R/W | With immediate effect |
| SM3303 | External shaft system: Shaft 1, shaft 2, negative point movement | R/W | With immediate effect |
| SM3304 | External shaft system: Shaft 1, positive point motion | R/W | With immediate effect |
| SM3305 | External shaft system: Shaft 1, negative point movement at shaft 3 | R/W | With immediate effect |
| SM3306 | External shaft system: 2 shafts, 1 positive point movement | R/W | With immediate effect |
| SM3307 | External shaft system: 2 shafts, 1 negative point movement | R/W | With immediate effect |
| SM3308 | External shaft system: 2 axes, 2 forward-pointing spot movements | R/W | With immediate effect |
| SM3309 | External shaft system: 2 shafts, 2 negative point movements | R/W | With immediate effect |

# 17.5 RCPATH instruction data point format

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-D12 | Target point coordinates [x, y, z, q0, q1, q2, q3] | Float |
| D15 | Tool id, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-200 to indicate the cp time percentage, and set 0 or-1 to indicate the zero speed point | INT16S |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and attitude ③ | FLOAT |
| D22 | Maximum acceleration percentage of running path and posture | FLOAT |
| D24 | Maximum deceleration percentage of running path and posture | FLOAT |
| D26 | Path transition error percentage | FLOAT |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: Ptp, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: Ptp with joint speed control (unfixed trajectory). 1: Ptp with joint speed control (fixed trajectory). 2: Ptp with cartesian speed control (fixed trajectory) ⑥ | INT16U |
| D34 | External axis motion type: 0: The external axis does not move. 1: The external axis moves in coordination with the robot. 2: The external axis moves independently while the robot remains stationary. 3: The external axis moves independently while the robot does not. 5: The external axis and the robot move simultaneously but asynchronously. 6: The external axis and the robot move simultaneously and synchronously but not in coordination. | INT16U |
| D35 | External axis transition parameter, value 0-100⑤ | INT16U |
| D36 | Cp transition based on displacement length | FLOAT |
| D38-D39 | Obligate | |
| D40 | External shaft 1 angle, fixed to the positioner shaft 1 | FLOAT |

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D42 | External shaft 2 angle, fixed as the positioner shaft 2 | FLOAT |
| D44 | External shaft 3 displacement, fixed ground rail shaft | FLOAT |
| D46 | External shaft system speed percentage | FLOAT |
| D48 | External shaft system acceleration percentage | FLOAT |
| D84 | Point number, user-defined, monotonically increasing (for easy correlation with sd5404 or log viewing of anomalies) | INT32U |
| D86 | The number of transition points for attitude is up to 5⑦. | INT16U |
| D92 | External shaft lifting height ⑧ for jump models with additional shafts | FLOAT |

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. When the external axis collaborative motion types are 1, 2, or 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to setting D18=6 and D19=1 (workpiece coordinate system). When D18=4 (tool coordinate system) is set, the point position information is relative displacement. For instance: If the current tool-bearing position of the robot end-effector is (400,200,100), setting point x, y, z information to (50,20,-20) will move the robot to (450,220,80) after movement.

③: When speed, acceleration, or both are not configured, the system will automatically adjust the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points, one being the arc auxiliary point and the other the arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint is set to the starting point rather than the endpoint information in the point input.

⑤: CP Time Transition Percentage parameter, indicating the percentage of remaining deceleration time for the current segment before initiating the next segment.

⑥: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: speed planning is performed in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: speed planning is performed in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values represent the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease

 User Manual for RC6F Series Robot Controller 17 External Axis Function

angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑦: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated according to the path length ratio, with the P3-P6 posture change being equally distributed.

⑧: When the robot is equipped with a height-adjustable external axis, this register allows configuration of the external axis's lifting height during JUMP trajectory setup. The value remains independent of JUMP group parameters, with the movement duration matching the robot's JUMP time. To enable this function, configure the external axis as a gantry type, designate it as the only controlled axis (moving axis), and set the external axis mode to 6.

### 17.5.1 Applicable Scenarios and Trajectory Types for External Axis Modes

| External axis mode | Track type | Swinging arc | Applicable scenarios |
|---|---|---|---|
| 0 | All track types | All arc types | Robotic motion only |
| 1 | Straight/arc/intersecting line | All arc types | Commonly used for robot operations in processing where manual handling is inconvenient |
| 2 | Regardless of the robot's trajectory type | Not have | This method is commonly used for external axis calibration checks and for teaching point positions through coordinated operation of external axis and point movements. |
| 3 | Regardless of the robot's trajectory type | Not have | External axis motion only |
| 4 | Rc movement is not supported | Not have | The external axis rotates to verify the calibration accuracy of the external axis. |
| 5 | Supports all track types | Not have | Fly back |
| 6 | Supports all track types | Not have | Quick reset and simultaneous arrival |

- External axis mode 0 and mode 1 support arc swinging.
- Multiple external axis modes can be configured in a RC circuit, which can be activated by triggering the SM3160.
- External Axis Mode 6 only supports JUMP trajectory.
- When the external axis angle and robot movement distance vary minimally, this segment is filtered out. If the external axis angle changes slightly while the robot angle varies significantly, Mode 5 and Mode 6 are modified to Mode 0. When the external axis angle changes substantially and the robot is short, Mode 1 replaces Mode 2, and Mode 5 and Mode 6 are adjusted to Mode 3. If Mode 1 involves only single-attitude robot movement, Mode 1 is modified to Mode 6.

## 17.6 Description of Registers

The external axis function is based on the RCPATH instruction and therefore includes all registers associated with the RCPATH instruction. The additional registers are listed below.

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD4110 | Float | Joint dynamic step length | Linear measure | R/W | Trigger next time |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD4112 | Float | Percentage of the dynamic speed of the joint | % | R/W | Trigger next time |
| SD4114 | Float | Percentage of the dynamic acceleration of the joint | % | R/W | Trigger next time |
| SD4171 | INT16U | Maximum hold time | Default: 1000ms | R/W | Trigger next time |
| SD4792 | Float | Acceleration percentage | % | R/W | Next trigger |
| SD5470 | INT16U | Current external axis interpolation segment number | | R | |
| SD5540 | Float | Dynamic adjustment, real-time speed ratio | Default 100% | R/W | With immediate effect |
| SD5573 | INT16U | Consider arc auxiliary point attitude<br>0: Not considered<br>1: Consider | Default 0 | R/W | Enable external axis next time |
| SD5574 | INT16U | Calculate the welding torch posture based on the welding torch angle<br>0: Teach the welding torch position<br>1: Welding torch posture calculated based on angle | Default 0 | R/W | Enable external axis next time |
| SD5580+2*I (I=0,1,2,3) | Float | Real-time interpolation of differential end velocities (linear velocity, rz angular velocity, ry angular velocity, and rx angular velocity) follows the same velocity pattern as sd4140, but the curve is advanced by approximately 20 interpolation cycles. | Mm/s perhaps Linear measure /s | R | |
| SD5590 | Float | Maximum transition reference value for external shafts | 0-10 degrees or 0-10mm | R/W | Enable external axis next time |
| SD5593 | INT16U | The current external axis system number can be used to specify the workpiece number for external axis calibration. | 1-2 | R/W | Enable external axis next time |
| SD5594 | INT16U | Current external axis motion type ① | 0-5 | R/W | Enable external axis next time |
| SD5595 | INT16U | Maximum variable cycle stop time for continuous point movement | Default: 500ms | R/W | Trigger next time |
| SD5596 | INT16U | The percentage of acceleration and deceleration time for the external axis, used for smaller acceleration and deceleration values, in conjunction with the external axis cp parameter. | 0-100% | R/W | Enable external axis next time |
| SD5600+2*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | Float | External axis joint angle given value | Degree or mm | R | |
| SD5620+2*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | Float | External shaft joint angle feedback value | Degree or mm | R | |
| SD5640+2*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | Float | Real-time speed setting for the outer shaft motion | Degree or mm | R | |
| SD5660+2*I | Float | The maximum joint speed limit for the outer shaft | Degree or | R/W | Enable |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | | is derived from the sfd6100 mapping. | mm | | external axis next time |
| SD5680+2*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | Float | The maximum joint acceleration limit for the outer shaft is mapped from the sfd6120. | $M/s^2$ or Mm/s | R/W | Enable external axis next time |
| SD5700+2*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | Float | The maximum joint acceleration limit for the outer shaft is mapped from the sfd6140. | $M/s^3$ or $mm/s^3$ | R/W | Enable external axis next time |
| SD5720+ 12*I+2EJ (I=0,1,2,3 J=0,1...5) | Float | Ground track and gantry calibration data: The ground track has two reference points, and the gantry has four reference points. Each reference point records the robot's end position coordinates [x, y, z] and three external axis angles, where i is the reference point and j is the point data. | Degree or mm | R/W | |
| SD5800+2*I (I=0,1...11) | Float | Copy the current gantry calibration parameters from the external axis system serial number to sfd6210 for actual motion participation of gantry parameters [x1, y1, z1, k1. X2, y2, z2, k2. X3, y3, z3, k3] | | R/W | |
| SD5830+2*I (I=0,1,2,3) | Float | The tool end display performs real-time interpolation in the reference coordinate system without including the end speed of the swing arc trajectory, which includes linear speed, rz, ry, and rx angular speed. | Mm/s perhaps Linear measure /s | R | |
| SD5840+2*I (I=0,1...2*NUMBER OF EXTERNAL SHAFT JOINTS-1) | Float | Record the angle of the external shaft during the pause | Degree or mm | R | |
| SD5870+2*I (I=0,1,2,3) | Float | Record the current ground orbit calibration parameters [x, y, z, k3]. After successful calibration, save them to sfd6040. | | R | |
| SD5880+2*I (I=0,1...5) | Float | Record the six dh parameters [a, d, alpha, theta, k1, k2] for the current positioner calibration. Upon successful calibration, save the data to sfd6050+12* (sd5593-1). | Degree or mm | R | |
| SD5900+2*I (I=0,1...5) | Float | The pose of the base coordinate of the calibration transducer in the world coordinate system | | R | |
| SD5980+2*I (I=0,1...6) | Float | The end-effector coordinates [x, y, z, q0, q1, q2, q3] of the current positioner system in its base coordinate system | | R | |
| SD6040+ 12*I+2*J (I=0,1...11. J=0,1,2...5 METHOD 2: I=0, J=0,1,2) | Float | In the first method of positioner calibration, each calibration point records the end position coordinates [x, y, z] and the angles of three external axes, with 12 calibration points in total across two joints. | | R/W | |
| | | Positioner calibration (method 2): Before calibration begins, record the first point p1 in the robot tool coordinate system as the initial position reference for the positioner. | Mm | R/W | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6046+2*I (I=0,1,2) | Float | Calibration of the positioner (method 2): Before calibration begins, record the second point p2 in the robot tool coordinate system. | Mm | R/W | |
| SD6052+2*I (I=0,1,2) | Float | Positioner calibration (method 2): Before calibration begins, record the third point $\overrightarrow{P2P3}$ $\overrightarrow{P2P3}$ p3 in the robot tool coordinate system. The vector is the x-axis direction vector of the positioner. | Mm | R/W | |
| SD6058+2*I (I=0,1,2) | Float | Before starting the calibration of the positioner (method 2), the user must input the estimated position values of the constraint points on the positioner in the 2-axis coordinate system. | Mm | R/W | |
| SD6064 | INT16U | Display of iteration times for positioner calibration (method 2) | | R | |
| SD6066 | Float | Calculation of alpha_0 display in the second method of calibration of the variator | Linear measure | R | |
| SD6070+2*J+10*I (I=0,1...5. J=0,1...4) | Float | Calibration of positioning machine (method 2) error model calibration of positioning machine pulse recording of robot joint | | R/W | |
| SD6130+2*J+10*I (I=0,1. J=0,1...4) | Float | Calibration of positioning machine (method 2) error model calibration of positioning machine joint pulse recording | | R/W | |
| SD6150+2*I (I=0,1...4) | Float | Calibration of positioning machine (method 2) error model calibration of positioning machine, ground track feedback angle record | | R/W | |
| SD6190+2*I (I=0,1,2) | Float | Positioner calibration (method 2): The maximum, minimum, and average errors of the positioner are measured with three floating-point values. The error is defined as the difference between the distance from each point to the center of the circle and the fitted radius. | Mm | R | |
| | | Positioner calibration (method 2): The system displays three error metrics-maxerror, minerror, and meanerror-which represent simulation-derived positional deviations. | | | |
| SD6200+2*I (I=0,1...6) | Float | Display the tool end's real-time interpolation in reference coordinate system for the pose [x, y, z, q0, q1, q2, q3] without swing arc | | R | |
| SD6262 | INT16U | Enable non-forward-looking attitude function in external axis collaborative mode<br>0: Plan in the workpiece coordinate system<br>1: You can select the pose for planning in the base coordinate system | Default 0 | R/W | Enable external axis next time |
| SFD3360+60*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | INT16U | Operation mode of external shaft motor | Default 0 | R/W | Cease plc Run again |
| SFD3361+60*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | INT16U | Types of external shaft motors | Default 3 | R/W | Cease plc Run again |
| SFD3362+60*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS | INT32U | Number of wires in the motor encoder | Default 131072 | R/W | Cease plc Run again |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| MINUS 1) | | | | | |
| SFD3364+60*I (I=0,1...NUMBER OF EXTERNAL SHAFT JOINTS MINUS 1) | INT32U | Move per circle | Default 131072 | R/W | Cease plc Run again |
| SFD6040+2*I (I=0,1,2,3) | Float | Orbit calibration parameters, [x, y, z, k3] | | R/W | Cease plc Run again |
| SFD6050+2*I (I=0,1...11) | Float | Dh parameters of the positioner, sequentially storing two sets of dh parameters, each containing 6 values [a, d, alpha, theta, k1, k2] | | R/W | Cease plc Run again |
| SFD6080+2*I (I=0,1...2*NUMBER OF EXTERNAL SHAFT JOINTS-1) | Float | The external shaft speed reduction ratio is configured with two external shaft systems stored sequentially. The rotary joint determines the speed reduction ratio, while the moving joint controls the motor's stroke per revolution. The current specification requires a shared ground rail for both systems, rendering the sfd6090 invalid. In this configuration, the sfd6084 serves as the common ground rail speed reduction ratio for both systems. | | R/W | Cease plc Run again |
| SFD5752+2*I (I=0,1...5) | Float | The position and orientation (x, y, z, rz, ry, rx) of the base coordinate in the world coordinate system for the positioner system 1 | | R/W | Cease plc Run again |
| SFD5764+2*I (I=0,1...5) | Float | The position and orientation (x, y, z, rz, ry, rx) of the 2-axis coordinate system in the world coordinate system | | R/W | Cease plc Run again |
| SFD6100+2*I (I=0,1...5) | Float | Maximum speed limit for external axes. For 6-axis robots, the data is copied to sd5660 after power-on. | Mm/s perhaps Linear measure /s | R/W | Cease plc Run again |
| SFD6120+2*I (I=0,1...5) | Float | Maximum acceleration limit for external axes. For 6-axis robots, the data is copied to sd5680 after power-on. | Mm/s perhaps Linear measure /s | R/W | Cease plc Run again |
| SFD6140+2*I (I=0,1...5) | Float | Maximum acceleration limit for external axes (6 axes). The robot will copy data to sd5700 after power-on. | Mm/s perhaps Linear measure /s | R/W | Cease plc Run again |
| SFD5370+I (I=6,7...10) | INT16U | Is the external axis controlled?<br>0: Uncontrolled<br>Controlled | Default 1 | R/W | Cease plc Run again |
| SFD6160+2*I (I=0,1...5) | Float | Maximum limit of the external shaft joint angle | Degree or mm | R/W | Cease plc Run again |
| SFD6180+2*I (I=0,1...5) | Float | Minimum angular limit of external shaft joint | Degree or mm | R/W | Cease plc Run again |
| SFD6200+I (I=0,1) | INT16U | Two external shaft system types, corresponding to the sfd6200 and sfd6201 respectively.<br>0: The external axis configuration consists of a 2-axis positioner and a 1-axis ground rail.<br>1: The external shaft assembly is a three-axis gantry. | Default 0 | R/W | Cease plc Run again |
| SFD6202+I (I=0,1) | INT16U | The robot's mounting configurations on the gantry correspond to models sfd6202 and sfd6203 respectively.<br>0: Inversion<br>1: Formal wear | Default 0 | R/W | Cease plc Run again |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SFD6210+2*I (I=0,1...11) | Float | Longmen calibration parameters: [x1, y1, z1, k1. X2, y2, z2, k2. X3, y3, z3, k3] | | R/W | Cease plc Run again |
| SM3041 | Bit | External axis system enable control bit. The external axis system can only be enabled after the robot is successfully enabled. | | R/W | With immediate effect |
| SM3190 | Bit | External shaft system enable status bit | | R | |
| SM3191 | Bit | External axis motion flag | | R | |
| SM3192 | Bit | External axis instruction interpolation completion flag | | R | |
| SM3193 | Bit | The welding system is in motion. | | R | |
| SM3194 | Bit | Welding system instruction given completion flag | | R | |
| SM3199 | Bit | External axis instruction given completion flag | | R | |
| SM3201 | Bit | Earth orbit calibration control point | | R/W | With immediate effect |
| SM3202 | Bit | Variable speed machine calibration control position | | R/W | With immediate effect |
| SM3300+2*I (I=0,1...5) | Bit | External shaft joint forward point movement | | R/W | With immediate effect |
| SM3301+2*I (I=0,1...5) | Bit | Negative point movement of the external shaft joint | | R/W | With immediate effect |

①: The motion state and interpolation state are updated according to the assigned values under different movement modes of the external axis.

(1) SD5594=0: The external axis is not in motion.
Update: Robot, RC command, and welding robot system.
(2) SD5594=1: The external axis synchronizes with the robot, with two kinematic chains working in tandem under positional and orientation constraints.
Update values: robot, external axis, RC command, welding robot system.
(3) SD5594=2: The external axis moves independently while the robot remains stationary relative to it, with both kinematic chains operating in coordination under a pose constraint.
Update values: robot, external axis, RC command, welding robot system.
(4) SD5594=3: The external axis moves independently without robot movement, with a single kinematic chain operating without relative constraints.
Update values: external axis, RC command, and welding robot system.
(5) External axis point movement
Update values: External axis, welding robot system. Set SD5594=4, the robot follows the external axis in point motion while remaining stationary relative to the external axis end. Update values: Robot, external axis, welding robot system.
(6) SD5594=5: The external axis moves synchronously with the robot but asynchronously, with two independent kinematic chains that lack relative constraints.
(7) SD5594=6: External axis. The robot synchronizes its motion with time, and the two kinematic chains are not constrained.
Update values: robot, external axis, RC command, welding robot system.

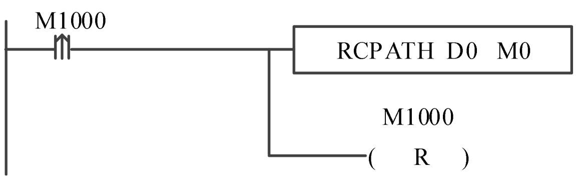
The rational application of external shaft technology can effectively solve the related application projects of external shaft systems. At present, there are mainly three processes:

- The attitude of the external axis is planned in the world coordinate system.
- Enable forward planning for external axes.
- The attitude of the outer axis is kept unchanged in the world coordinate system.

| Register type | Function | Remarks |
|---|---|---|
| SD5579 | The robot's posture remains unchanged | The default value is 0. This feature is available only in mode 1 within an rc. |
| SD6262 | Planning of the attitude of the outer shaft in the world coordinate system | The default value is 0. When the robot's attitude speed fluctuates during external axis coordination, sd6262=1 activates this function. |
| SD5597 | External axis forward processing | Refer to the two notes below |

- The SD5597 defaults to 0, disabling the external axis's forward-looking function. When using CP transition for external axes, their planning time exceeds the robot's, triggering error code "10145". During operation, the external axis speed can be calculated based on the robot's runtime, with the set speed potentially exceeding the calculated value. Applicable scenarios: Non-forward-looking point information distribution. The external axis system number cannot be modified during non-forward-looking point distribution. it can only be adjusted after all movements have stopped.
- When SD5597 is set to 1, the forward planning function for external axes is activated, enabling them to operate according to the predefined schedule. In this mode, the external axis control point (D35) becomes invalid. Key benefits include: 1) Using periodic synchronization to resolve time synchronization issues (e.g., 10145 alarm). 2) Ideal for scenarios requiring real-time distribution of point information.

## 17.7 Function Applications




The source file sample program is as follows:

```
#include"RCPATH.h"

#define pNum 4
RcPointInformation RcPoint[pNum] = {
    {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,1,0,0,0,0,0,1,50,0.2,0,1.1,5,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0},
    {{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,2,0,0,0,0,0,1,50,1.2,2.5,1.2,5,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0},
    {{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,2,0,0,0,0,0,1,50,1.5,3.1,-1.0,5,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0},
}.

void RCPATH(PINT16S W, BIT B)
{
  if (SD[5500] == 0)
  {
    for (int i = 0. i < SD[5401]. i++)
    {
```

 User Manual for RC6F Series Robot Controller 17 External Axis Function

```
// Transmit pose data
        *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
        *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
        *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
        *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
        *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
        *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q2.
        *(float*)&D[(SD[5588] + 100 * i + 14)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 14)] = RcPoint->weaveAngleZ. // Arc angle
D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType.//hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
D[(SD[5588] + 100 * i + 34)] = RcPoint->ExtAxisMoveType. //External axis mode
D[(SD[5588] + 100 * i + 35)] = RcPoint->ExtAxisTranR. //External axis transition error
*(float) & D[(SD[5588] + 100 * i + 40)] = RcPoint->ExtAxisJointTheta[0]. //Angle of external axis 1
*(float) & D[(SD[5588] + 100 * i + 42)] = RcPoint->ExtAxisJointTheta[1]. //Angle of the second external axis
*(float)* &D[(SD[5588] + 100 * i + 44)] = RcPoint->ExtAxisJointTheta[2]. //Angle of the third external axis
*(float) & D[(SD[5588] + 100 * i + 46)] = RcPoint->ExtAxisSpeed. //External axis speed
*(float) & D[(SD[5588] + 100 * i + 48)] = RcPoint->ExtAxisAcc. //External axis acceleration
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
      }
   }
SM[3160] = 1. //Trigger the RC command movement for the external axis
}
```

User Manual of RC6F Series Robot Controller 18 External Signal Interaction Function

# 18. External signal interaction function

## 18.1 Function Overview

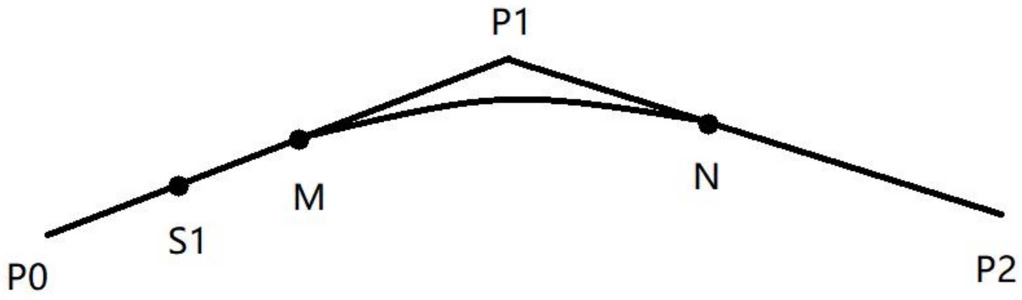
The interaction between RCPATH instructions and external signals is processed to achieve the ability to change or record the current motion state through external trigger signals, reducing the complex logic issues that previously arose at the application layer. This primarily includes SENSE/TILL/FIND functions. all corresponding external signals are uniformly set as M registers, with the platform program encapsulating which external signal corresponds to each M register. Each RCPATH can individually set trigger register addresses for three functions. SENSE and TILL cannot be triggered simultaneously, and SENSE has lower priority than TILL. The processing of all three functions occurs within the bus communication cycle. During the execution of RBSTOP(0.0f), SENSE and TILL cannot be triggered. A new control word for external signal interaction has been added to support the corresponding functions and prevent abnormal use of M registers. The three functions are implemented based on RCPATH instructions, and the corresponding parameter settings can be configured.

## 18.2 Function Specifications and User Instructions

### 18.2.1 SENSE Function

1) During continuous trajectory operation, the system determines whether to transition between trajectories and proceed to the next segment based on the external signal M[SENSE_ID] corresponding to the set function, where SENSE_ID is the offset address of the set register. When M[SENSE_ID]=ON and the SENSE trigger condition is met, the transition between current and next trajectories is disabled, and movement stops upon reaching the target point of the current trajectory. When M[SENSE_ID]=OFF, the trajectory executes normally without additional processing.
2) The SENSE function supports LINE, PTP, and JUMP trajectory types. Its stop point is user-defined, and the trajectory cannot be resumed after stopping. When triggered, the SENSE function executes RBSTOP(0.0) in motion, which is invalid, whereas RBSTOP(-1.0) takes effect, with emergency stop having the highest priority. For LINE trajectories, triggering SENSE essentially performs a fixed-length stop. Strictly requiring stopping at the target point may cause over-acceleration, and the default setting allows execution without alarm.
3) SENSE activates before trajectory transitions. When the current trajectory is in the forward-end or non-forward-end segment, only the interpolation of the current segment is required. If the current motion trajectory is the final segment of a continuous one, SENSE becomes invalid. If the remaining length of the current segment is insufficient for deceleration when SENSE is triggered, it will be processed as TILL, causing the trajectory to execute a slow stop. For transitions between trajectories, the judgment conditions are as follows:
   - For spline transitions, the effective point is set at the midpoint of the remaining trajectory after path transition, with the last triggered point serving as the starting point for sense signal analysis.
   - For CP transitions, the selection of the effective point follows the same principle as for spline transitions.

User Manual of RC6F Series Robot Controller 18 External Signal Interaction Function




For example, consider the straight lines P0P1 and P1P2. After transition processing, the trajectory is composed of P0M, MN, and NP2, with S1 as the active point. When the sense signal is triggered at any point between the S1M line segments during trajectory interpolation, the system initiates response. Upon stopping at P1, the subsequent trajectory is halted.

### 18.2.2 TILL Function

1) The trajectory determines whether to transition and proceed to the next segment based on the external signal M[TILL_ID] corresponding to the set function, where TILL_ID is the offset address of the set register. When M[TILL_ID]=ON, the current trajectory stops after the pause. When M[TILL_ID]=OFF, the trajectory proceeds normally without additional processing.
2) The TILL function supports full trajectory types. Its stop point is on the trajectory after transition processing, essentially performing a one-time pause. Once stopped, the trajectory cannot be restored. After TILL activation, executing RBSTOP(0.0) during stop motion is invalid. RBSTOP(-1.0) takes effect, with the emergency stop state having the highest priority.
3) The TILL function operates on a single segment, with the stop point at any location between the current segment and the end point, determined by the dwell time setting. For multi-segment operations, the stop point may extend to the next segment.

### 18.2.3 FIND Function

1) During trajectory execution, the system determines whether to latch the real-time feedback position based on the external signal M[FIND_ID] corresponding to the configured function, where FIND_ID is the register offset address. When the configuration signal is ON, M[FIND_ID]=ON, recording the real-time feedback position at the signal trigger moment, and M[FIND_ID]=OFF, with no further processing. Conversely, when the configuration signal is OFF, M[FIND_ID]=OFF, recording the real-time feedback position at the signal trigger moment, and M[FIND_ID]=ON, with no further processing.
2) The FIND function supports full trajectory tracking, with the bus cycle task recording all position data at the specified point. It also logs feedback pose or joint angles when trigger signals are detected (depending on the coordinate system type: joint coordinates record feedback joint angles, while other systems record the end flange's pose relative to the base coordinate). The function also records hand binding, CF configuration, and external axis feedback angles.

## 18.3 RCPATH instruction data point format

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-D12 | Target point coordinates [x, y, z, q0, q1, q2, q3] | Float |
| D15 | Tool id, 0-19 | INT16U |

User Manual of RC6F Series Robot Controller 18 External Signal Interaction Function

| | | |
|---|---|---|
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-200 to indicate the cp time percentage, and set 0 or-1 to indicate the zero speed point | Int16s |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and posture ③ | Float |
| D22 | Maximum acceleration percentage of running path and posture | Float |
| D24 | Maximum deceleration percentage of running path and posture | Float |
| D26 | Path transition error percentage | Float |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: Ptp, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: Ptp with joint speed planning (unfixed trajectory). 1: Ptp with joint speed planning (fixed trajectory). 2: Ptp with cartesian speed planning (fixed trajectory). 3: Ptp with acceleration optimization planning (fixed trajectory) ⑤ | INT16U |
| D70 | Sense function corresponds to m register address offset | INT32U |
| D72 | The till function corresponds to the m register address offset. | INT32U |
| D74 | Find function corresponds to m register address offset | INT32U |
| D84 | Point number, user-defined, monotonically increasing (for easy correlation with sd5404 or log viewing of anomalies) | INT32U |
| D86 | The number of transition points for attitude is up to 5⑥. | INT16U |

User Manual of RC6F Series Robot Controller 18 External Signal Interaction Function

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. When the external axis collaborative motion types are 1, 2, or 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to setting the reference coordinate system as D18=6, D19=1 (workpiece coordinate system). When D18=4 (tool coordinate system) is set, the point position information is relative displacement. For instance: If the current tool-bearing position of the robot end-effector is (400,200,100), setting the point position coordinates (x, y, z) as (50,20,-20) will move the robot to (450,220,80) after movement.

③: When speed, acceleration, or both are not configured, the system automatically adjusts the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points, one being the arc auxiliary point and the other the arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint is the starting point, not the endpoint information in the point input.

⑤: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: performing speed planning in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: performing speed planning in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values correspond to the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑥: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated based on the path length ratio, with the posture change amount from P1 to P6 being equally distributed.

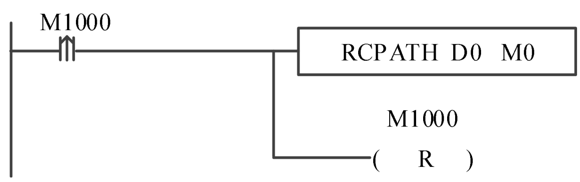
User Manual of RC6F Series Robot Controller 18 External Signal Interaction Function

## 18.4 Description of Registers

The external signal function is based on the RCPATH instruction and therefore includes all registers associated with the RCPATH instruction. The following registers are added.

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5432 | INT16U | Type of input signal for external signal function<br>0: M signal<br>1: X signal | | R/W | Next time rc movement is triggered |
| SD5433 | INT16U | Type of input signal for external signal functionality<br>0: On signal trigger<br>1: Off signal triggered | | R/W | Next time rc movement is triggered |
| SD5440 | INT16U | Does it support sense<br>0: Close<br>1: Turn on | Default 0 | R/W | Sense will take effect next time |
| SD5441 | INT16U | Does it support till<br>0: Close<br>1: Turn on | Default 0 | R/W | Next time till is triggered |
| SD5442 | INT16U | Supports find function<br>0: Close<br>1: Turn on | Default 0 | R/W | Next time find is triggered, it will take effect |
| SD6900 | INT16U | The find function records the offset address type of the data point: 0 for d register, 1 for hd register. | Default 0 | R/W | Rc will take effect next time |
| SD6902 | Int32s | The find function records the offset of point information addresses. For example, setting 0 for sd6900 and 1000 for sd6902 means the point information recorded by find is saved from d1000 to d1099. The point format is the same as rc. Negative values are treated as absolute values. | Default 0 | R/W | Rc will take effect next time |
| SM3153 | Bit | Till function trigger status bit | | R | |

## 18.5 Function Applications




The source file example program is as follows:

```
#include"RCPATH.h"

#define pNum 4
RcPointInformation RcPoint[pNum] = {
    {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
```

User Manual of RC6F Series Robot Controller 18 External Signal Interaction Function

```
{{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,100,101,102,0,0,0,0,0,1,0,0,0,0,0},
{{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,103,104,105,0,0,0,0,0,2,0,0,0,0,0},
{{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,106,107,108,0,0,0,0,0,3,0,0,0,0,0},
}.

void RCPATH(PINT16S W, BIT B)
{
SD[5440] = 1. //Enable the SENSE function
SD[5441] = 1. //Enable TILL function
SD[5442] = 1. //Enable the FIND function
  if (SD[5500] == 0)
  {
    for (int i = 0. i < SD[5401]. i++)
    {
// Transmit pose data
      *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
      *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
      *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
      *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
      *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
      *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q2.
      *(float*)&D[(SD[5588] + 100 * i + 14)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 14)] = RcPoint->weaveAngleZ. // Arc angle
D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType.//hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
*(INT32U)* & D[(SD[5588] + 100 * i + 70)] = RcPoint->SenseAddr. //Set the SENSE trigger address
*(INT32U)* & D[(SD[5588] + 100 * i + 72)] = RcPoint->TillAddr. //Set TILL trigger address
*(INT32U)* & D[(SD[5588] + 100 * i + 71)] = RcPoint->FindAddr. //Set FIND trigger address
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
    }
  }
SM[3160] = 1. //Trigger the RC command movement for the external axis
}
```

# 19. Definition of the four-axis U

## 19.1 Function Overview

The current RC6F controller lacks support for setting Rz beyond ±180 degrees in four-axis robot control, preventing the implementation of optimal arc motion (not inferior arc motion) for axis 4 during Cartesian trajectory movement. Additionally, joint 4 only supports ±360-degree soft limit settings, with no provision for arbitrarily large soft limits. The ±360-degree processing scheme also causes inverse solution values to be affected by the current joint angle setting. Any change in the given joint angle triggers corresponding adjustments to the inverse solution, resulting in operational inconvenience and poor user experience.

To address these issues, the quadrotor requires customized configurations: it supports arbitrary soft limit values for joint 4, Cartesian trajectory movement to reach specified joint angles, and Rz display exceeding ±180 degrees.

## 19.2 Functional Specifications

- The attitude of the tool and the workpiece is treated as consistent to solve the problem of losing Rz information in the process of attitude change. The attitude expression to the user is changed, but the attitude calibration data is not lost, which is reflected in the internal operation, such as the point movement along the Z-axis of the tool.
- The RCPATH command now supports Euler angle input for new attitude configurations, implemented via SFD5106=1 (default value 0). The original point coordinates [x, y, z, q0, q1, q2, q3] are updated to [x, y, z, Rz, Ry, Rx], with Euler angles expressed in degrees.
- For the quadrotor, when the SFD5106 is activated, all attitude trajectories are processed as non-forward-looking attributes, with interpolation calculations performed for the Euler angle Rz while preserving its variation trend.
- After retaining the Rz data, the inverse solution calculates Joint 4 using Rz, enabling support without additional processing (while Joint 6 requires special handling) to accommodate any soft limit settings.
- This feature cannot be used with the robot axis configuration feature. Six-axis models do not support this feature and will trigger alarm 10035.

In addition, the quaternion attitude parameter SFD6030 is removed, and the original parameters SFD5082+2*i, SFD5092+2*i, and SFD5272+2*i are directly adopted. When the robot performs a point movement, if the reference coordinate system is the tool coordinate system, the actual tool attitude must be calculated to determine the point movement vector.

## 19.3 Description of Registers

| Address | Type | Explain | Unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| Sfd5106 | INT16U | Rcpath point attitude representation type: 0: Unit quaternion, 1: Euler angle. The default is 0, valid only for quadrotor robots. | | R/W | Stop plc re-run |

## 19.4 Function Applications

After configuring the SFD5106, enter the Euler angle for attitude input in the RCPath command.

```
#include"RCPATH.h"
#define pNum 2
RcPointInformation RcPoint[pNum] = {
    {{400,-200,-22.5,243, 0,0,0},0,0,0,0,0,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {{-405.3,-142.12,0,-157,0,0,0},0,0,0,0,0,1,30,80,80,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,2},
}.

void RCPATH(PINT16S W, BIT B)
{
   if (SD[5500] == 0)
   {
      for (int i = 0. i < SD[5401]. i++)
      {
// Transmit pose data
         *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
         *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
         *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
         *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
         *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
         *(float*)&D[(SD[5588] + 100 * i + 10)] = RcPoint->targetPos.orient.q2.
         *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 14)] = RcPoint->weaveAngleZ. // Arc angle
D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType. // Hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
      }
   }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
}
```

User Manual of RC6F Series Robot Controller 20 Robot Axis Configuration Function

# 20. Robot axis configuration function

## 20.1 Function Overview

The default mode of inverse solution for robot is the shortest distance, which may result in the final joint angle being 360° different from the user's setting, or the wrong rotation direction causing alarm. Therefore, the function is optimized by configuring the robot's axes to expand the motion range of the 1/4/6 axes.

The axis configuration applies to all robot types but cannot be used with the four-axis U function. Additionally, when the point is configured in joint coordinate system, the CF configuration is not effective, and the target position will strictly follow the joint coordinate values.

## 20.2 Function Specifications and User Instructions

The configuration parameters of the robotic arm's axes include cf1, cf4, and cf6, defined by the current half-turn rotation of axes 1, 4, and 6. If an axis has a motion range of-900° to +900°, its quadrant division is as follows:

| Axis configuration parameter cf | Angle range | number of turns |
|---|---|---|
| 5 | (720°, 900°] | 2 |
| 4 | (540°, 720°] | |
| 3 | (360°, 540°] | 1 |
| 2 | (180°, 360°] | |
| 1 | [0,180°] | 0, no configuration required. The PTP final angle defaults to the shortest distance. |
| 0 | - | |
| -1 | [-180°,0) | |
| -2 | [-360°,-180°) | -1 |
| -3 | [ -540°,-360°) | |
| -4 | [-720°,-540°) | -2 |
| -5 | [-900°,-720°) | |

- For RC commands, simply configure the D30-D32 pins at the corresponding positions.
- For the RB instruction, 12-14 words (INT16S) from the 20-word sequence at each position correspond to cf1, cf4, and cf6. During position teaching, the system can directly copy the first 20 words starting from SD4000 as the current position data and execute the RB instruction. The robot's current axis configuration attributes can be retrieved from SD4012 to SD4014.

| register address | type | definition | Value/Unit | read-write | Trigger time |
|---|---|---|---|---|---|
| SD4012 | INT16S | Cf1 | | R | |
| SD4013 | INT16S | Cf4 | | R | |

User Manual of RC6F Series Robot Controller 20 Robot Axis Configuration Function

| | | | | | |
|---|---|---|---|---|---|
| SD4014 | INT16S | Cf6 | | R | |
| SFD5107 | INT16U | The posture can be optimized.<br>Switch: 0-off, 1-on | | R/W | Next trigger |

User Manual of RC6F Series Robot Controller 20 Robot Axis Configuration Function

## 20.3 RCPATH instruction data point format

| Rcpath instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-D12 | Target point coordinates [x, y, z, q0, q1, q2, q3] | Float |
| D14 | Arc angle, range [0-360]*100 | INT16U |
| D15 | Tool id, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-100 to indicate the percentage of cp time. Set 0 or-1 to indicate the zero-speed point. | Int16s |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and posture ③ | Float |
| D22 | Maximum acceleration percentage of running path and posture | Float |
| D24 | Maximum deceleration percentage of running path and posture | Float |
| D26 | Path transition error percentage | Float |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: Ptp, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: Ptp with joint speed planning (unfixed trajectory). 1: Ptp with joint speed planning (fixed trajectory). 2: Ptp with cartesian speed planning (fixed trajectory). 3: Ptp with acceleration optimization planning (fixed trajectory) ⑥ | INT16U |
| D30 | Cf1, joint 1 attribute configuration. For specific definitions, refer to the cf quadrant division. | Int16s |
| D31 | Cf4, joint 4 attribute configuration. For specific definitions, refer to the cf quadrant division. | Int16s |
| D32 | Cf6, joint 6 attribute configuration. For specific definitions, refer to the cf quadrant division. | Int16s |
| D33 | Cfx, value 0-7 | INT16U |
| D84 | Point number, user-defined, monotone increasing, non-continuous | INT32U |
| D86 | The number of transition points for attitude is up to 5⑦. | INT16U |
| D94-d99 | Obligate | |

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. When the external axis collaborative motion types are 1, 2, or 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to reference coordinate system D18=6, D19=1 (workpiece coordinate system). When setting D18=4 (tool coordinate system), the point position information is relative displacement. For instance: If the robot's tool-end position is (400,200,100) and the point position x, y, z information is set to (50,20,-20), the robot will reach (450,220,80) after movement.

User Manual of RC6F Series Robot Controller 20 Robot Axis Configuration Function

③: When speed, acceleration, or both are not configured, the system will automatically adjust the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points: an arc auxiliary point and an arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint position is the starting point position rather than the endpoint information in the point input. Trajectory type 7 represents an incomplete intersection line, composed of the current point, auxiliary point, and endpoint. Trajectory type 8 represents a complete intersection line, consisting of the current point, auxiliary point 1, auxiliary point 2, auxiliary point 3, and the endpoint, with the requirement that the starting and endpoint poses must be identical. If the specified number of points is not met, the trajectory is forcibly converted to a straight line.

⑤ : CP Time Transition Percentage parameter, indicating the percentage of remaining deceleration time for the current segment before initiating the next segment.

⑥: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: speed planning is performed in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: speed planning is performed in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values represent the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑦: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated according to the path length ratio, with the P3-P6 posture change amount being equally distributed.

## 20.4 Function Applications

```
#include"RCPATH.h"
#define pNum 4
RcPointInformation RcPoint[pNum] = {
    {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                                                                        {{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-
0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,50,20,100,1,0,0,0,0,0},
                                                                        {{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-
0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,60,30,100,2,0,0,0,0,0},
                                                                        {{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-
0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,50,30,80,3,0,0,0,0,0},
```

User Manual of RC6F Series Robot Controller 20 Robot Axis Configuration Function

```
}.

void RCPATH(PINT16S W, BIT B)
{
  if (SD[5500] == 0)
  {
    for (int i = 0. i < SD[5401]. i++)
    {
// Transmit pose data
      *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
      *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
      *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
      *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
      *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
      *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q2.
      *(float*)&D[(SD[5588] + 100 * i + 14)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType. // Hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
      D[(SD[5588] + 100 * i + 30)] = RcPoint->cf1. // Axis 1 configuration parameter
      D[(SD[5588] + 100 * i + 31)] = RcPoint->cf4. // Axis 4 configuration parameter
      D[(SD[5588] + 100 * i + 32)] = RcPoint->cf6. // Axis 6 configuration parameter
D[(SD[5588] + 100 * i + 76)] = RcPoint->JumpType.//Jump type
D[(SD[5588] + 100 * i + 77)] = RcPoint->CPEnable. // Check if JUMP enables CP
*(float)* & D[(SD[5588] + 100 * i + 78)] = RcPoint->ArchS. // Ascending distance
*(float) & D[(SD[5588] + 100 * i + 80)] = RcPoint->ArchE. //Decrease distance
*(float)* & D[(SD[5588] + 100 * i + 82)] = RcPoint->LimZ. // Lift height limit
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
    }
  }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
}
```

For any joint angle, the corresponding pose remains unchanged when its value is adjusted by k*360° (where k is an integer).

When the normal target position is based on the base coordinate system, the corresponding joint angle defaults to [-180,180], with the joint attribute set to-1 or 1.

**For PTP instructions:**

When CF is not configured by default, the end position angle is closest to the starting angle. For example, in a

User Manual of RC6F Series Robot Controller 20 Robot Axis Configuration Function

SCARA model's axis 4 joint, the starting angle is 170°, while the end position angle corresponds to-170° at joint 4. The final position will be 190°, as the-170° and 190° positions are equivalent. When configuring CF, the end position angle at joint 4 remains-170°. Setting CF4 to-1 allows reaching-170°, CF2 to 190°, and CF4 to 550°. If CF is set to 3, the system will reach 550°, as this angle is closest to the CF range (360°,540°).

**For Cartesian commands (Line, Circle):**

When CF is not configured by default, the joint motion direction is unknown, but the system predicts the motion direction and calculates the end joint angle based on the predicted direction.

When configuring the CF (Configuration Factor), if the CF value exceeds the starting CF, the joint will rotate forward. conversely, it will rotate backward. Note that the CF of the Cartesian command does not guarantee the final end angle will match the CF. This is because the Cartesian trajectory limits the joint's maximum rotation to 360°. For example, if the starting angle is 170° (CF=1) and the final end angle is-170°, a CF value of ≥2 will only stop at 190°, while a CF value of ≤-1 will only stop at-170°.

**Point CF configuration issue:**

For the teaching points, their CF values can be recorded during the teaching process and then input when executing the motion command.

Note: For PTP motion, the system can maintain the joint angle during teaching based on CF. However, for Line/CirCle commands, the joint angle difference between the trajectory's start and end points must be less than 180° to ensure the teaching joint angle.

For points generated through computational methods, it is generally recommended to set their CF values to match those of the next teaching point. For example, when programming a trajectory from point A to point B with precise CF parameters, the calculated points should adopt the CF values of point B. If no teaching points exist and all points are computed, it is advisable to set all CF values to 0.

User Manual of RC6F Series Robot Controller 21 Cartesian Trajectory Over-speed Protection Function

# 21. The Cartesian trajectory over-speed protection function

## 21.1 Function Overview

Current Cartesian trajectory planning methods calculate path lengths in Cartesian space based on constraints like maximum linear velocity and maximum angular velocity. During interpolation, the system computes the corresponding Cartesian space pose from time data and then inverses it to obtain joint angles. This approach neglects joint constraints, resulting in abrupt stoppages when joint overspeed is detected during operation, which compromises the integrity of the trajectory execution.

During transitions between two joint motion trajectories, the Cartesian trajectory and joint trajectory, as well as between two non-forward-looking Cartesian trajectories, utilize the CP method. This method directly superimposes the corresponding joint angles of the two trajectories based on the D17 transition time percentage set at the point. However, during interpolation, when the joint speed of the trajectory is high and the transition time percentage is large, it may cause joint overspeeding and result in motion cessation.

To solve the above problems, two improved speed planning algorithms are developed, namely the constrained Cartesian planning and the CP transition optimization.

Joint constraint Cartesian trajectory planning incorporates joint constraints (primarily maximum speed limits) during trajectory planning. The system predicts potential joint overspeeding and reduces the Cartesian maximum speed in overspeed zones to ensure full trajectory execution. This feature cannot resolve joint overspeeding caused by singularities.

CP transition optimization predicts potential joint overspeed during trajectory planning. If overspeed occurs, it automatically reduces the CP coefficient step by step until either the CP region no longer generates overspeed or the CP coefficient drops to zero. When combined with the Cartesian planning function for joint constraints, this mechanism effectively prevents most joint overspeed alarms, ensuring stable robot operation.

## 21.2 Functional Specifications

Building upon the RC command framework, this method optimizes the speed planning process for Cartesian trajectories. For trajectories with completed speed planning, the system first performs joint speed extremum search and over-speed zone calculation to determine potential joint overspeeding and its extent. During trajectory interpolation, dynamic speed adjustment is implemented before entering the over-speed zone to reduce joint speed and prevent overspeeding. After exiting the over-speed zone, the original speed is restored through subsequent dynamic speed adjustment.

Usage:

- S1: Activate the SD7441 over-speed detection optimization master switch.
- S2: Execute the RC instruction flow.
- CP transition optimization.

Similarly, based on the RC instruction, the CP region undergoes CP time optimization. For trajectories already planned with CP, an extremum search for joint speed is performed. When joint overspeed is predicted,

User Manual of RC6F Series Robot Controller 21 Cartesian Trajectory Over-speed Protection Function

the CP coefficient is progressively reduced by the preset adjustment step until the joint no longer overspeeds or the CP coefficient drops to zero.

Usage:

- S1: Configure the SD7442 CP to adjust the step size for overspeed adjustment.
- S2: Activate the SD7441 over-speed check optimization master switch.
- S3: Execute the RC instruction flow.

## 21.3 Register Description

| Address | Type | Explain | Unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5540 | Float | Rc command dynamic speed control ratio, in percentage<br>In the cartesian programming of joint restraint, the value is automatically adjusted during interpolation to prevent joint overspeed. | % | R/W | The next delivery point will take effect |
| SD6020 | Int16s | Current overspeed axis number | | R | |
| SD5575 | INT16U | Maximum time for forward instruction to reach the target position, in milliseconds<br>Reserve time before and after the speed limitation region of the cartesian planning | Ms | R/W | The next delivery point will take effect |
| SD7441 | INT16U | Speed check optimization switch: 0: Off, 1: On. Default is 0<br>The cp optimize switch and the save constraint cartesian switch are merged into one. When enabled, the internal forward window automatically adjusts to 5, and the forward step length automatically adjusts to 3. | | R/W | The next delivery point will take effect |
| SD7442 | INT16U | Cp adjusts step size percentage (unit:%). Default is 70, with a range of 1-90. | | R/W | The next delivery point will take effect |
| SD7443 | INT16U | The joint speed extremum search count is typically set to the default value (0). When the value exceeds 5, the algorithm increases search frequency for more accurate extremum prediction, though this may prolong computation time and potentially cause plc offline. Conversely, setting the value between 0 and 5 reduces search frequency to minimize computation time, but this compromises extremum prediction accuracy, potentially failing to predict overspeed joints. | | R/W | The next delivery point will take effect |
| SM3420 | Bit | During trajectory correction with dynamic speed regulation, the system automatically adjusts sd5540 and sm3420 to on when entering overspeed zones, then resets them to off upon exiting. If the system is interrupted or emergency stop occurs during adjustment, the original sd5540 values cannot be restored after stopping. | | R | |

User Manual of RC6F Series Robot Controller 21 Cartesian Trajectory Over-speed Protection Function

## 21.4 Function Applications

```
#include"RCPATH.h"
#define pNum 4
RcPointInformation RcPoint[pNum] = {
    {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,50,20,100,1,0,0,0,0,0},
    {{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,60,30,100,2,0,0,0,0,0},
    {{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,50,30,80,3,0,0,0,0,0},
}.

void RCPATH(PINT16S W, BIT B)
{
  if (SD[5500] == 0)
  {
    for (int i = 0. i < SD[5401]. i++)
    {
// Transmit pose data
        *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
        *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
        *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
        *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
        *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
        *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q2.
        *(float*)&D[(SD[5588] + 100 * i + 14)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType.//hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
        D[(SD[5588] + 100 * i + 30)] = RcPoint->cf1. // Axis 1 configuration parameter
        D[(SD[5588] + 100 * i + 31)] = RcPoint->cf4. // Axis 4 configuration parameter
        D[(SD[5588] + 100 * i + 32)] = RcPoint->cf6. // Axis 6 configuration parameter
D[(SD[5588] + 100 * i + 76)] = RcPoint->JumpType.//Jump type
D[(SD[5588] + 100 * i + 77)] = RcPoint->CPEnable. // Check if JUMP enables CP
*(float) & D[(SD[5588] + 100 * i + 78)] = RcPoint->ArchS. // Ascending distance
*(float) & D[(SD[5588] + 100 * i + 80)] = RcPoint->ArchE. //Decrease distance
*(float)* & D[(SD[5588] + 100 * i + 82)] = RcPoint->LimZ. // Lift height limit
```

User Manual of RC6F Series Robot Controller 21 Cartesian Trajectory Over-speed Protection Function

```
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
        }
    }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
}
```

Enabling this feature effectively prevents the 1008 overspeed alarm caused by CP overlap.

User Manual of RC6F Series Robot Controller 22 Speed Planning with Given Acceleration and Deceleration Time

# 22. Speed planning with given acceleration and deceleration time

## 22.1 Function Overview

The existing speed planning algorithm needs to set the acceleration and deceleration size and the acceleration size in each point information, which is not intuitive enough for the user. In the actual planning, due to the maximum speed and length limit, setting different acceleration and deceleration size may end up with the same effect, which is easy to cause ambiguity.

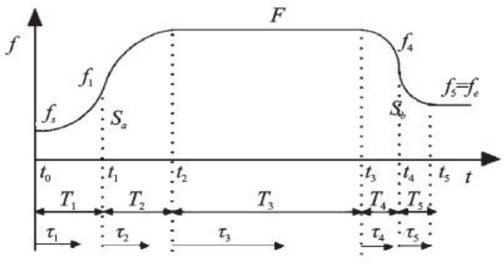
In non-forward mode, the speed change time of the moving segment is generally different because of the different parameter setting, so the transition effect cannot be realized from the uniform speed segment to the uniform speed segment.

To solve the above problems, the speed planning algorithm with given acceleration and deceleration time is developed.

- Users can set parameters more intuitively.
- All motion segments share identical acceleration/deceleration durations, enabling seamless CP transition effects that achieve smooth transitions between uniform speed phases.

## 22.2 Functional Specifications

The current seven-segment S-type speed planning scheme incorporates both constant acceleration and deceleration phases, resulting in non-linear relationships between acceleration, deceleration, and time during speed changes. To address this, the algorithm is modified by eliminating the constant acceleration and deceleration segments, retaining only the remaining five segments. This streamlined approach enables direct calculation of speed planning parameters based on acceleration and deceleration durations.




This solution offers another advantage. The seven-stage S-type speed planning algorithm employs methods like the bisection method to search for an optimal maximum speed when the preset limit cannot be reached. However, due to constraints such as iteration limits, the resulting solution is not perfect, ultimately leaving a small constant-speed segment. In contrast, the five-stage S-type speed planning algorithm can directly calculate the required maximum speed using a formula when the preset limit cannot be achieved, eliminating the need for iterative calculations and avoiding any constant-speed segments.

The given acceleration and deceleration time planning method is not suitable for the cases of B-spline transition or attitude change. It is only suitable for PTP trajectory and JUMP mode 1.

User Manual of RC6F Series Robot Controller 22 Speed Planning with Given Acceleration and Deceleration Time

## 22.3 Description of Registers

| Address | Type | Explain | Unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SFD7500 | INT16U | speed planning mode<br>0: Plan based on acceleration/deceleration values<br>1: Plan based on acceleration and deceleration time | | R/W | Stop PLC re-run |
| SFD7502 | float | acceleration time | ms | R/W | Stop PLC restart |
| SFD7504 | float | deceleration time | ms | R/W | Stop PLC re-run |

## 22.4 Function Applications

```
#include"RCPATH.h"
#define pNum 4
RcPointInformation RcPoint[pNum] = {
    {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,50,20,100,1,0,0,0,0,0},
    {{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,60,30,100,2,0,0,0,0,0},
    {{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,0,30,80,80,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,50,30,80,3,0,0,0,0,0},
}.

void RCPATH(PINT16S W, BIT B)
{
   if (SD[5500] == 0)
   {
      for (int i = 0. i < SD[5401]. i++)
      {
// Transmit pose data
         *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
         *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
         *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
         *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
         *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
         *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q2.
         *(float*)&D[(SD[5588] + 100 * i + 14)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType.//hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum operating speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
```

User Manual of RC6F Series Robot Controller 22 Speed Planning with Given Acceleration and Deceleration Time

```
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
        D[(SD[5588] + 100 * i + 30)] = RcPoint->cf1. // Axis 1 configuration parameter
        D[(SD[5588] + 100 * i + 31)] = RcPoint->cf4. // Axis 4 configuration parameter
        D[(SD[5588] + 100 * i + 32)] = RcPoint->cf6. // Axis 6 configuration parameter
D[(SD[5588] + 100 * i + 76)] = RcPoint->JumpType.//Jump type
D[(SD[5588] + 100 * i + 77)] = RcPoint->CPEnable. // Check if JUMP enables CP
*(float)* & D[(SD[5588] + 100 * i + 78)] = RcPoint->ArchS. // Ascending distance
*(float) & D[(SD[5588] + 100 * i + 80)] = RcPoint->ArchE. //Decrease distance
*(float)* & D[(SD[5588] + 100 * i + 82)] = RcPoint->LimZ. // Lift height limit
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
      }
   }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
}
```

Set the trajectory planning mode SFD7500 to 1 and configure the acceleration/deceleration time SFD7502 and SFD7504. This function resolves speed irregularities caused by overlapping trajectories during CP.

Note: The PTP mode 1 works best with this function.

# 23. TCP calibration function

The TCP calibration optimization algorithm provides two calibration methods: multi-point optimal TCP calibration and error model TCP calibration. The following explains the different calibration methods:

## 23.1 Multi-point TCP calibration

At present, the tool end is provided with 8 constraint points, and 6 or 7 points are selected for TCP solution.

### 23.1.1 Function Overview

Multi-point optimal TCP calibration is mainly through collecting several constraint points in the robot workspace, and then selecting the points with smaller error for TCP solution according to certain evaluation criteria, which improves the accuracy of TCP calibration.

### 23.1.2 Calibration Process

#### 23.1.2.1 Before calibration begins

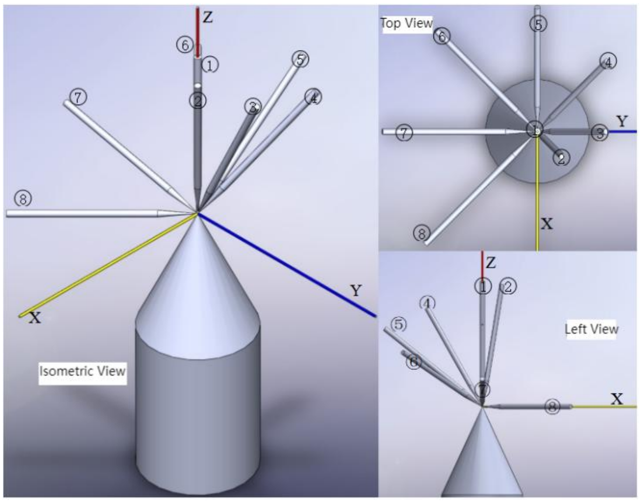
Before starting multi-point TCP optimization calibration, confirm the following:

- Whether the robot body calibration is completed. Multi-point optimal TCP calibration must be performed after the robot body calibration is completed. otherwise, the TCP calibration is invalid!
- Check if the tool parameter (SFD5500+2*i+12*j) corresponding to the tool number to be calibrated in the teaching device is 0 (as checked in the register). If not, set it to 0 first.
- Ensure the tool is securely and reliably installed. During the multi-point optimal TCP calibration process, the tool's end point must remain fixed relative to the six-axis flange. otherwise, the TCP calibration will be invalid.

#### 23.1.2.2 Start calibration

Point collection:

- The general principle of point acquisition is that the six-axis flange position and attitude of any two pairs of acquisition points are quite different.
- Recommended pose: The tool tip's eight distinct pose points are achieved by simultaneously adjusting two angles. Angle 1: The tool tip's angle with the Z-axis, ranging from 0° to 90°. Angle 2: The tool tip's projection on the XOY plane relative to the X-axis, ranging from 0° to 360°.
- The reference posture diagram for point collection (n=1,2,3,4,5,6,7,8) at the constraint point, based on the recommended pose, is shown below:




The robot workspace's orientation for the illustrated sampling points is defined as follows: the red point aligns with the Z-axis, the yellow with the X-axis, and the blue with the Y-axis.

The sampling process is as follows:

1） The calibration cone point is placed in the robot's dexterous workspace. The position of the cone point should be selected to ensure that the robot can reach the point in various postures, and the cone point should be fixed reliably and remain still during the point process.

2） The manipulator adjusts the tool tip's orientation to align it vertically with the ground (e.g., welding torch, as shown in Figure ①. approximation suffices, with Point 1 verifying the tool's orientation). After completing the adjustment, the manipulator maintains this orientation while moving along Cartesian coordinates X/Y/Z, progressively approaching the constraint cone point. When nearing the constraint point, the manipulator adjusts the manipulator's single-step distance to 0.1mm, ensuring the tool tip contacts the cone point as closely as possible. After determining the first sampling point position, the manipulator records the current joint pulse values of its position, corresponding to registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are stored in the register address +2*j+16*i INT32S, where j=0 and i=0,1,2,3,4,5.

3） Move the manipulator along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during tool pose adjustments. Adjust the tool's end-effector from its initial position (attitude ①) by first rotating counterclockwise along the Y-axis to approximately 13° (approximate precision is acceptable), then performing a 45° counterclockwise rotation around the Z-axis (approximate precision is acceptable), as shown in attitude ②. After completing the adjustment, maintain the current pose while moving the manipulator along the X/Y/Z axes to approach the constraint point. When nearing the target, set the manipulator's teaching tool step distance to 0.1mm and position the end-effector as close to the constraint point as possible. Once the second sampling point is secured, record the joint pulse values of the manipulator's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then stored in the register address +2*j+16*i INT32S, where j=1 and i=0,1,2,3,4,5.

4） Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during tool pose adjustments. Modify the tool's end-effector position as follows: First, rotate it counterclockwise 45° around the Z-axis of world coordinates (approximate precision is acceptable), then rotate clockwise along the X-axis to achieve an angle of approximately 26° relative to position ① (approximate precision is acceptable), as shown in position ③.

After completing the adjustment, maintain the position while moving the robot along the X/Y/Z axes of Cartesian coordinates to approach the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm and guide the robot to align the tool tip as closely as possible. Once the third sampling point is secured, record the joint pulse values of the robot's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. Store these values in the register address +2*j+16*i INT32S, where j=2 and i=0,1,2,3,4,5.

5) Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during tool pose adjustments. Adjust the tool's end-effector from its initial position (attitude ③) by first rotating clockwise along the X-axis to approximately 39° relative to attitude ① (approximate precision is acceptable), then performing a 45° counterclockwise rotation around the Z-axis (similarly approximate). After completing these adjustments (attitude ④), maintain the configuration while moving the robot along the X/Y/Z axes toward the constraint point. As the robot approaches, set the manipulator's single-step distance to 0.1mm to ensure optimal tool-point alignment. Once the fourth sampling point is secured, record the joint pulse values of the robot's current position in registers HSD104 /HSD124 /HSD144 /HSD164 /HSD184 /HSD204. These values are then stored in the register address +2*j+16*i INT32S, where j=3 and i=0,1,2,3,4,5.

6) Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during tool pose adjustments. Modify the tool's end-effector position as follows: First, rotate it counterclockwise 45° around the Z-axis of world coordinates (approximate precision is acceptable), then rotate clockwise along the Y-axis to achieve an angle of approximately 52° relative to position ① (approximate precision is acceptable), as shown in position ⑤. After completing the adjustment, maintain the position while moving the robot along the X/Y/Z axes of Cartesian coordinates to approach the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm and guide the robot to align the tool tip as closely as possible. Once the fifth sampling point is secured, record the joint pulse values of the robot's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. Store these values in the register address +2*j+16*i INT32S, where j=4 and i=0,1,2,3,4,5.

7) Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during tool pose adjustments. Begin with rotating the tool tip clockwise along the Y-axis from its initial position (attitude ⑤) to approximately 65° relative to attitude ① (approximate precision is acceptable). Then rotate counterclockwise 45° around the Z-axis (similarly approximate precision). After completing these adjustments (attitude ⑥), maintain the configuration while moving the robot along the X/Y/Z axes toward the constraint point. As the robot approaches, adjust the manipulator's single-step distance to 0.1mm to ensure optimal tool-tip contact. Once the sixth sampling point is secured, record the joint pulse values of the robot's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are stored in the register address +2*j+16*i INT32S, where j=5 and i=0,1,2,3,4,5.

8) Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during tool pose adjustments. Modify the tool's end-effector position as follows: First, rotate it counterclockwise 45° around the Z-axis of world coordinates (approximate precision is acceptable), then rotate counterclockwise along the X-axis to achieve an angle of approximately 78° relative to position ① (approximate precision is acceptable), as shown in position ⑦. After completing the adjustment, maintain the position while moving the robot along the X/Y/Z axes of Cartesian coordinates to approach the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm and guide the robot to align the tool tip as closely as

possible. Once the seventh sampling point is secured, record the joint pulse values of the robot's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. Store these values in the register address +2*j+16*i INT32S, where j=6 and i=0,1,2,3,4,5.

9) Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during tool pose adjustments. Begin with rotating the tool tip counterclockwise along the X-axis from its original position (attain an approximate 90° angle with position ①), then rotate it 45° counterclockwise around the Z-axis (similarly approximate). Finalize the adjustment with position ⑧. Maintain this configuration while moving the robot along the X/Y/Z axes toward the constraint point. As proximity increases, adjust the manipulator's single-step distance to 0.1mm to ensure optimal tool-point alignment. After establishing the eighth sampling point, record the joint pulse values of the robot's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are stored in the register address (register head address +2*j+16*i) of the INT32S register, where j=7 and i=0,1,2,3,4,5.

10) After the eight points are collected, the robot tool parameters are calculated.

If the cone point moves during the alignment process, restart the alignment from the first step.

23.1.2.3 Tool parameter calculation and writing

- Set SM3204=1, calculate the robot tool parameters [X, Y, Z, RZ, RY, RX] based on the collected data, write them into the designated register, and output the calibrated tool parameters to the user terminal for display.
- The tool parameter calibration errors (MaxError, MinError, MeanError) are written to the specified register and displayed on the client side as TCP calibration errors.

## 23.1.3 Description of Registers

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SFD6610 | float | Setting the 7-point to 6-point error limit | Default value: 0.2mm | R/W | Stop PLC re-run |
| SD6990 | INT16U | Configure joint pulse storage register type 0: D, 1: HD | | R/W | with immediate effect |
| SD6991 | INT16U | Configure the calibration result storage register type: 0: D, 1: HD | | R/W | with immediate effect |
| SD6992 | INT32S | Configure the first address of the joint pulse storage register, the pulse storage register type, INT32S, and handle negative values as absolute values. | | R/W | with immediate effect |
| SD6994 | INT32S | Configure the first address of the calibration result storage register and its type (float). When negative values are input, handle them as absolute values. | | R/W | with immediate effect |
| SM3204 | Bit | Multi-point Optimal TCP Calibration Trigger Control Bit | | R/W | with immediate effect |

| Order number | Corresponding parameter | Remarks | Register type |
|---|---|---|---|
| 0 | X | Tool calibration result X | float |
| 1 | Y | Tool calibration result Y | float |
| 2 | Z | Tool calibration result Z | float |
| 3 | Rz | Euler angle Rz, in degrees | float |
| 4 | Ry | Eulerian angle Ry, in degrees | float |
| 5 | Rx | Eulerian angle Rx, in degrees | float |
| 6 | MaxError | maximum change | float |
| 7 | MinError | least error | float |
| 8 | MeanError | average error | float |

## 23.2 Error Model TCP Calibration

### 23.2.1 Function Overview

The error model TCP optimization algorithm is mainly used to collect five points and use the error model to identify the tool parameters to achieve high-precision TCP calibration.

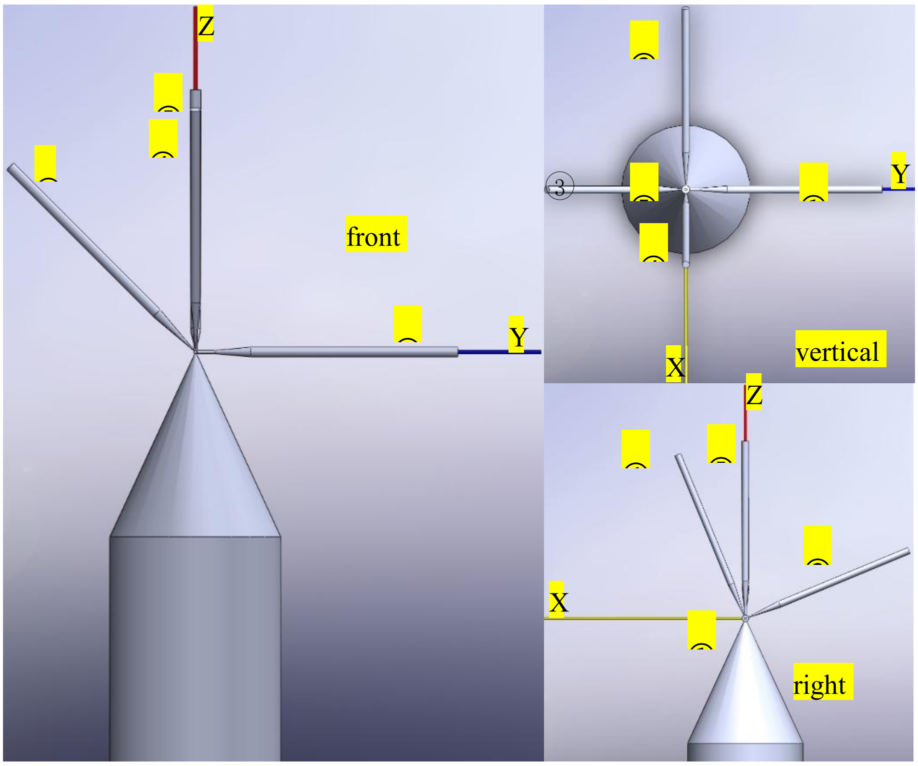
### 23.2.2 Calibration Process

#### 23.2.2.1 Before calibration begins

Before starting the first step of error model TCP calibration, confirm the following:

- Whether the robot body calibration is complete. The error model TCP calibration must be performed after the robot body calibration is completed. otherwise, the TCP calibration is invalid.
- Ensure the tool is securely and reliably installed. The error model TCP calibration process requires the tool end point to be fixed relative to the six-axis flange. otherwise, the TCP calibration will be invalid.

23.2.2.2 Start the calibration process




The sampling process is as follows:

1) The calibration cone point is placed in the robot's dexterous workspace. The position of the cone point should be selected to ensure that the robot can reach the point in various postures, and the cone point should be fixed reliably and remain still during the point process.
2) The manipulator adjusts the tool's end-effector orientation to align it parallel to the base coordinate Y (as shown in Figure ①, approximate alignment suffices without precision). After completing the orientation adjustment, the manipulator maintains this position while moving along the base coordinate axes X/Y/Z, progressively approaching the constraint point. When nearing the constraint point, the manipulator adjusts the manipulator's single-step distance to 0.1mm, ensuring the end-effector point closely contacts the constraint point. Once the first sampling point is secured, the manipulator's current joint pulse values are recorded and stored in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then written into the register address (first address + 2*j+14*i) of the INT32S register, where j=0 and i=0,1,2,3,4,5.
3) Move the manipulator along the positive Z-axis of the base coordinate system to maintain a safe distance from the constraint point, preventing potential collisions during subsequent tool pose adjustments. Rotate the tool tip counterclockwise along the X-axis from its initial position (attain an approximate 22.5° angle, as shown in Pose ②). After completing the adjustment, stabilize the tool's orientation while moving the manipulator along the X/Y/Z axes toward the constraint point. As the manipulator approaches, adjust the teaching manipulator's step size to 0.1mm to ensure optimal tool-point alignment. Once the second sampling point is secured, record the joint pulse values of the manipulator's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then stored in the register address (register head address +2*j+14*i) using the INT32S register, where j=1 and i=0,1,2,3,4,5.
4) Move the manipulator along the Z-axis in the forward direction to maintain a safe distance from the constraint point, preventing potential collisions during subsequent tool pose adjustments. Adjust the

tool's end-effector position as follows: First, rotate it counterclockwise around the X-axis from position ① to approximately 45° (approximate precision is acceptable), then rotate 90° counterclockwise around the Z-axis to achieve position ③ (as shown in the diagram). After completing the adjustment, maintain the position while moving the manipulator along the X/Y/Z axes toward the constraint point. As the manipulator approaches, set the teaching tool's step distance to 0.1mm and position the end-effector as close to the constraint point as possible. Once the third sampling point is secured, record the joint pulse values of the manipulator's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then stored in the register address +2*j+14*i INT32S, where j=2 and i=0,1,2,3,4,5.

5) Move the manipulator along the Z-axis in the forward direction to maintain a safe distance from the constraint point, preventing potential collisions during tool pose adjustments. Adjust the tool's end-effector to position it at approximately 22.5° counterclockwise rotation from Pose ③ (approximate precision is acceptable) before rotating 90° counterclockwise around the Z-axis to achieve Pose ④ (as shown in the diagram). After completing the adjustment, maintain the current pose while moving the manipulator along the X/Y/Z axes toward the constraint point. When approaching the constraint point, set the teaching tool's step distance to 0.1mm and position the end-effector to make it contact the constraint point. Once the fourth sampling point is secured, record the joint pulse values of the manipulator's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then stored in the register address +2*j+14*i INT32S, where j=3 and i=0,1,2,3,4,5.

6) Move the manipulator along the Z-axis of the base coordinate to a distance from the constraint point, preventing collision during subsequent tool attitude adjustments. Adjust the tool's end-effector posture as follows: First, rotate clockwise around the X-axis from posture ④ to approximately 22.5° (approximate precision is acceptable), then rotate counterclockwise 90° around the Z-axis to achieve posture ⑤ (denoted as P5). The teaching point ⑤ will assist in determining the Z-axis orientation of the tool coordinate system, which should be as perpendicular to the ground as possible. After completing the posture adjustment, maintain the current posture while moving the manipulator along the X/Y/Z axes of the base coordinate to approach the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm to ensure the tool tip contacts the constraint point. Once the fifth sampling point is secured, record the joint pulse values of the manipulator's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are stored in the register address +2*j+14*i INT32S, where j=4 and i=0,1,2,3,4,5.

If the constraint point shifts during the sampling process, restart the sampling from step (1).

When no special tool attitude requirements are specified, the tool calibration point process terminates, with the tool's Z-axis orientation automatically aligned with the welding wire direction.

When calibrating tool orientation, handle the following scenarios: (1) Calibrate only the X-axis direction. (2) Calibrate only the Z-axis direction. (3) Calibrate both X/Z axes. For X-axis calibration, the Z-axis defaults to the welding wire direction. In this case, teach the 6th point as P6, where vector P5P6 defines the tool's X-axis orientation.

When the tool is oriented along the Z-axis, its X-axis direction defaults to the Y-axis of the base coordinate system. In this case, if the sixth point is designated as P6, the vector P5P6 represents the tool's Z-axis orientation.

When both the X and Z axes of the tool are calibrated, to teach the 6th and 7th points, vector P5P6 represents the tool's X-axis orientation, while vector P7P6 represents its Z-axis orientation.

#### 23.2.2.3 Identification and Update of Tool Parameters

- Set SM3206=1, identify the robot tool parameters [X, Y, Z, RZ, RY, RX] based on the acquisition points, and write them into the corresponding registers. Then output these parameters to the user terminal to display the tool parameter calibration values.
- The tool parameter calibration errors (MaxError, MinError, MeanError) are written to the corresponding registers and displayed on the user side as TCP calibration errors.

### 23.2.3 Description of Registers

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6990 | INT16U | Configure joint pulse storage register type 0: D, 1: HD | | R/W | with immediate effect |
| SD6991 | INT16U | Configure the calibration result storage register type: 0: D, 1: HD | | R/W | with immediate effect |
| SD6992 | INT32S | Configure the first address of the joint pulse storage register, the pulse storage register type, INT32S, and handle negative values as absolute values. | | R/W | with immediate effect |
| SD6994 | INT32S | Configure the first address of the calibration result storage register and its type (float). When negative values are input, handle them as absolute values. | | R/W | with immediate effect |
| SD6996 | INT16U | TCP attitude calibration flag: 0: Write attitude to default value. 1: Calibrate only X-axis. 2: Calibrate only Z-axis. 3: Calibrate both Z/X axes. All other values are treated as 0. | | R/W | with immediate effect |
| SM3206 | Bit | Error Model TCP Calibration Trigger Control Bit | | R/W | with immediate effect |

| Order number | Corresponding parameter | Remarks | Register type |
|---|---|---|---|
| 0 | X | Tool calibration result X | float |
| 1 | Y | Tool calibration result Y | float |
| 2 | Z | Tool calibration result Z | float |
| 3 | Rz | Euler angle Rz, in degrees | float |
| 4 | Ry | Eulerian angle Ry, in degrees | float |
| 5 | Rx | Eulerian angle Rx, in degrees | float |
| 6 | MaxError | Maximum change | float |
| 7 | MinError | Least error | float |
| 8 | MeanError | Average error | float |

## 23.3 Least-squares TCP calibration

### 23.3.1 Function Overview

The tool1 is installed on the end of the robot flange, but the end coordinate system {T} is unknown relative to the end coordinate system {E} of the robot flange.

The primary objective is to rapidly obtain the pose information of an unknown tool. For scenarios where only the tool's position is required, the four-point method is employed when the tool's default orientation is used (with the z-axis aligned with the tool's tip and the x-axis corresponding to the y-axis in the base coordinate system). When the tool's position requires explicit definition of its x-axis or z-axis orientation, the five-point method is applied. For cases where both the x-axis and z-axis orientations must be specified, the six-point method is utilized.

The four-axis robot employs two-point and three-point methods to measure the tool's positional deviations along the x and y axes, as well as its rotational angle (Rz).

Two-point method is calculated directly by formula, and three-point method is calculated by least square method.

## 23.3.2 Calibration Process

### 23.3.2.1 Start the calibration process

The sampling process is as follows:

1) The calibration cone point is placed in the robot's dexterous workspace. When selecting the cone point position, it should be ensured that the robot can reach the point in various postures, and the cone point should be fixed reliably and remain stationary during the point process. Meanwhile, the tool end is determined as the reference point to ensure alignment with the cone point during calibration.
2) The manipulator adjusts the tool tip's orientation. After completing the adjustment, it maintains the position while moving along the base coordinate axes (X/Y/Z) to approach the cone point. When nearing the cone point, the manipulator adjusts the teaching manipulator's step distance to 0.1mm, ensuring the tool tip aligns as closely as possible with the cone point. Once the first sampling point is secured, the manipulator's current pose (X, Y, Z, Rz, Ry, Rx) is recorded and stored in the corresponding register +2*i+12*j float register, where j=0, i=0,1,2,3,4,5.
3) Move the manipulator along the Z-axis of the base coordinate to maintain a safe distance from the cone point, preventing potential collisions during tool pose adjustments. Rotate the tool tip to significantly alter its orientation relative to the initial reference point. Then, guide the manipulator along the X/Y/Z axes of the base coordinate to progressively approach the cone point. When nearing the target, set the teaching tool's step distance to 0.1mm and position the manipulator's tip as close to the cone point as possible. After determining the second sampling point, record the manipulator's current position and orientation (X, Y, Z, Rz, Ry, Rx) and store them in the corresponding register: +2*i+12*j float register, where j=1, i=0,1,2,3,4,5.
4) Move the manipulator along the Z-axis of the base coordinate to maintain a safe distance from the cone point, preventing potential collisions during tool pose adjustments. Rotate the tool tip to significantly alter its orientation relative to the first two reference points. Then, guide the manipulator along the X/Y/Z axes of the base coordinate while progressively approaching the cone point. When nearing the cone point, set the teaching tool's step distance to 0.1mm and position the manipulator's tip as close to the cone point as possible. After determining the third sampling point, record the manipulator's current position and orientation (X, Y, Z, Rz, Ry, Rx) and store them in the corresponding register +2*i+12*j float register, where j=2 and i=0,1,2,3,4,5.
5) For the first three points, the tool can be positioned in any orientation, but the fourth point requires the tool reference point to be perpendicular to the ground. The robot moves along the Z-axis of the base coordinate system, positioning the tool at a safe distance from the cone point to prevent collision during subsequent tool orientation adjustments. After aligning the tool's end with a vertical downward orientation, the robot advances along the X/Y/Z axes of the base coordinate system, gradually approaching the cone point. When nearing the cone point, the teaching tool's step distance is adjusted to 0.1mm, ensuring the tool's tip remains as close to the cone point as possible. Once the fourth measurement point is secured, the robot's current position and orientation (X, Y, Z, Rz, Ry, Rx) are recorded and stored in the corresponding register +2*i+12*j float register, where j=3 and i=0,1,2,3,4,5.
6) The fifth step involves moving the tool reference point from the fourth point along the x-axis direction of the target TCP (or z-axis direction if using the five-point method). For x-axis movement, the tool tip must remain stationary within the XOY plane, ensuring no z-axis displacement. Conversely, for z-axis

movement, the tool tip must stay fixed in the YOZ plane, preventing x-axis displacement. A translation distance of approximately 100mm is sufficient. Once the fifth sampling point is secured, the robot's current pose (X, Y, Z, Rz, Ry, Rx) is recorded and stored in the corresponding register, specifically the +2*i+12*j float register (where j=4 and i=0,1,2,3,4,5).

7) The sixth step involves moving the tool reference point from the fifth position along the z-axis of the target TCP. When setting the z-axis orientation, ensure the tool tip remains stationary in the YOZ plane (zero-yaw-z-axis plane), with zero x-axis displacement. A translation distance of approximately 100mm is sufficient. After determining the fourth sampling point's position, record the robot's current pose (X, Y, Z, Rz, Ry, Rx) and store it in the corresponding register +2*i+12*j float register, where j=5 and i=0,1,2,3,4,5.

When SD6996=0, the sampling process stops at step (5). When SD6996=1 or 2, it proceeds to step (6). When SD6996=3, it reaches step (7).

#### 23.3.2.2 Identification and Update of Tool Parameters

- Set SM3205=1, identify the robot tool parameters [X, Y, Z, RZ, RY, RX] based on the acquisition points, and write them into the corresponding registers. Then output these parameters to the user terminal to display the tool calibration values.
- The tool parameter calibration errors (MaxError, MinError, MeanError) are written to the corresponding registers and displayed on the user side as TCP calibration errors.

The calibration procedure for four-axis machine tools follows the same protocol as described above. Simply align the tool tip with the calibration cone point, then select two or three reference points as needed.

### 23.3.3 Description of Registers

| Register address | Type | Definition | Value/unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD6990 | INT16U | Configure the robot pose storage register type: 0: D, 1: HD | | R/W | with immediate effect |
| SD6991 | INT16U | Configure the calibration result storage register type: 0: D, 1: HD | | R/W | with immediate effect |
| SD6992 | INT32S | Configure the first address of the robot pose storage register, the type of pose storage register, float, and handle negative values as absolute values | | R/W | with immediate effect |
| SD6994 | INT32S | Configure the first address of the calibration result storage register and its type (float). When negative values are input, handle them as absolute values. | | R/W | with immediate effect |

| | | | | | |
|---|---|---|---|---|---|
| SD6996 | INT16U | TCP attitude calibration flag: 0: Write attitude to default value. 1: Calibrate only X-axis. 2: Calibrate only Z-axis. 3: Calibrate both Z/X axes. All other values are treated as 0.<br>Four-axis machine tool calibration: 0: two-point method. 1: three-point method. all values except 0 are processed using the three-point method | | R/W | with immediate effect |
| SM3205 | Bit | Least Squares TCP Calibration Trigger Control Bit | | R/W | with immediate effect |

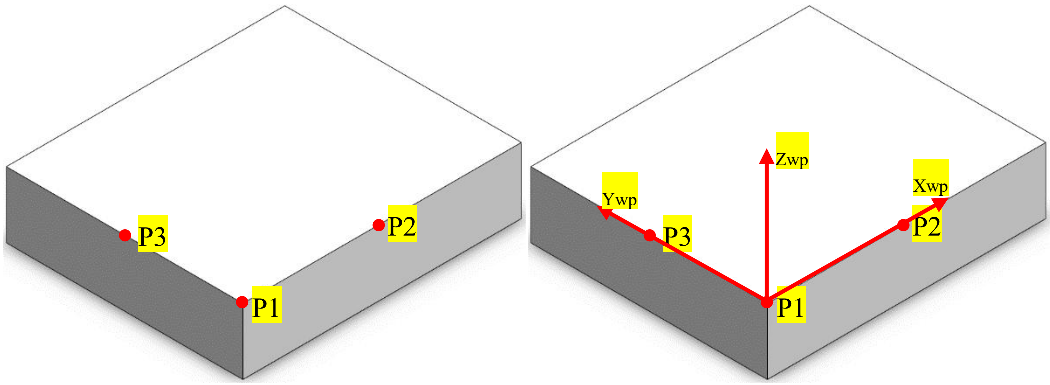
| Order number | Corresponding parameter | Remarks | Register type |
|---|---|---|---|
| 0 | X | Tool calibration result X | FLOAT |
| 1 | Y | Tool calibration result Y | FLOAT |
| 2 | Z | Tool calibration result Z | FLOAT |
| 3 | Rz | Euler angle Rz, in degrees | FLOAT |
| 4 | Ry | Eulerian angle Ry, in degrees | FLOAT |
| 5 | Rx | Eulerian angle Rx, in degrees | FLOAT |
| 6 | MaxError | Maximum change | FLOAT |
| 7 | MinError | Least error | FLOAT |
| 8 | MeanError | Average error | FLOAT |

## 23.4 Workpiece Calibration

### 23.4.1 Function Overview

The robot performs three-point teaching to calculate the relative coordinates of the workpiece between the robot's base coordinate system and either the world coordinate system or the ground orbit.

### 23.4.2 Calibration Process




1) Select the appropriate position to teach P1 point as the origin of the workpiece coordinate system, and store it in "register first address + 2*i + 6*j", where i∈[0,2] indicates the coordinate value of a single point, and j∈[0,2] indicates the j-th point.
2) Select the appropriate position to teach P2 point, as the point on the X-axis of the workpiece coordinate system, and store it in "register first address + 2*i + 6*j", where i∈[0,2] represents the coordinate value of a single point, and j∈[0,2] represents the j-th point.
3) Select an appropriate position to teach the P3 point as the point on the Y-axis of the workpiece coordinate system, and store it in the address register 'register first address + 2*i + 6*j', where i∈[0,2] represents the coordinate value of a single point, and j∈[0,2] indicates the j-th point.

Set SM3210=1, calculate the workpiece parameters [X, Y, Z, RZ, RY, RX] based on the teaching points, write them into the corresponding registers, and output the calibrated values to the user terminal for display.

### 23.4.3 Description of Registers

| Register address | Type | Definition | Value /unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6990 | INT16U | Configure the set-point register type: 0: D, 1: HD | | R/W | with immediate effect |
| SD6991 | INT16U | Configure the calibration result storage register type: 0: D, 1: HD | | R/W | with immediate effect |

| | | | | | |
|---|---|---|---|---|---|
| SD6992 | INT32S | Configure the starting address of the floating-point storage register and its type. When negative values are input, they are processed as absolute values. | | R/W | with immediate effect |
| SD6994 | INT32S | Configure the first address of the calibration result storage register and its type (float). When negative values are input, handle them as absolute values. | | R/W | with immediate effect |
| SM3210 | Bit | Workpiece calibration trigger control bit | | R/W | with immediate effect |

| Order number | Corresponding parameter | Remarks | Register type |
|---|---|---|---|
| 0 | X | Tool calibration result X | FLOAT |
| 1 | Y | Tool calibration result Y | FLOAT |
| 2 | Z | Tool calibration result Z | FLOAT |
| 3 | Rz | Euler angle Rz, in degrees | FLOAT |
| 4 | Ry | Eulerian angle Ry, in degrees | FLOAT |
| 5 | Rx | Eulerian angle Rx, in degrees | FLOAT |

# 24. Simple self-calibration function

## 24.1 Function Overview

Point constraint self-calibration primarily collects a series of constraint point groups in the robot's workspace to identify errors in 27 parameters of both the robot body and its end-effector. The identified parameters are then compensated into corresponding registers, thereby improving the absolute positioning accuracy of the robot body. Currently, it only supports six-axis systems.

> The current regulation requires collecting 3 constraint points at different locations in the workspace, with each point comprising 5 data points to form a set. This results in 3 sets totaling 15 data points.

## 24.2 Calibration Process

### 24.2.1 Before the calibration begins

Before starting point constraint calibration, confirm:

- Verify the correct rotation direction of all-axis motors in the robot (SFD3047+60* (n-1)).
- Check if the robot's axes are set to the zero position. If not, adjust them to the zero position first.
- Set the robot's DH parameters to their theoretical values: a1/a2/a3/d4/d6, corresponding to registers SFD5004-SFD5014 (e.g., a1=180, a2=600, a3=130, d4=622.5, d6=110.5). If a coupling ratio is required, set the theoretical coupling ratio parameter. Note that the theoretical DH parameters vary by robot type, so pay attention to this when configuring.
- The gear ratios must be known and set to their theoretical values: k1/k2/k3/k4/k5/k6, with corresponding registers SFD5020-SFD5030 (e.g., k1=121, k2=121, k3=121, k4=100, k5=150, k6=80). Note that theoretical gear ratios vary by device type, so careful configuration is required.
- After the above operations are completed, the end position of the robot tool can be determined through the positive kinematics of the robot, and the point constraint self-calibration can be started.

### 24.2.2 Start the calibration process

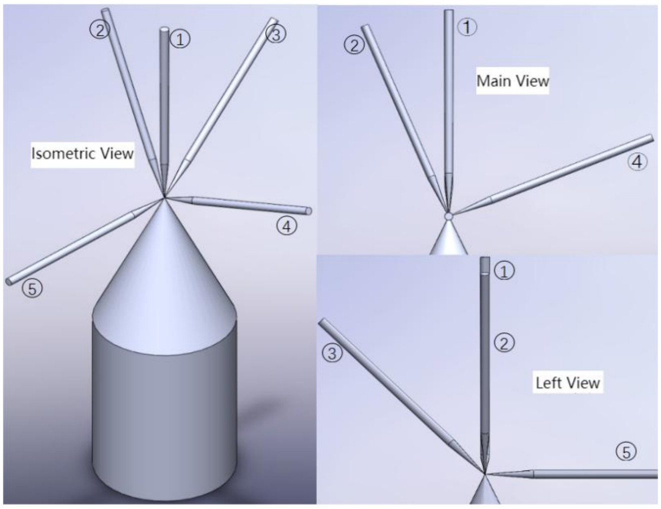
The calibration procedure is as follows:

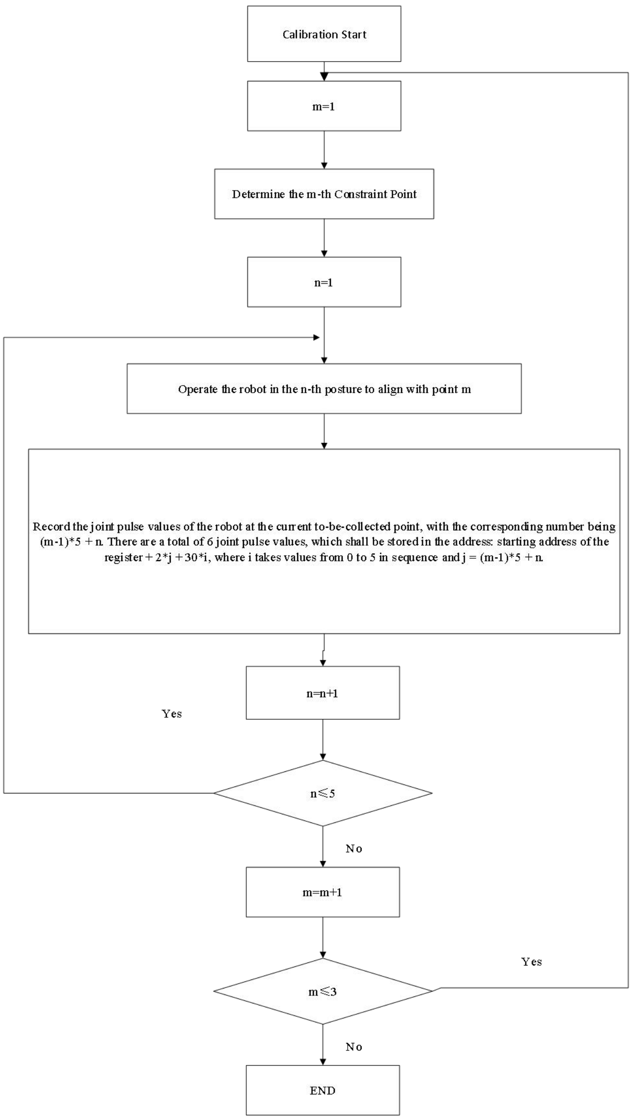
where m denotes the constraint points in the robot's workspace (m=1,2,3) and n represents the sampling points of each constraint point (n=1,2,3,4,5).

The schematic diagram of the acquisition points (n=1,2,3,4,5) for the constraint points is shown below:

The general orientation of the constraint point in the robot workspace is defined as follows: assuming the constraint point is the origin of the coordinate system, the direction of tool attitude ① from the constraint point is aligned with the positive direction of the Z-axis in the robot base coordinate system, and the direction of tool attitude ② is aligned with the positive direction of the X-axis in the robot base coordinate system.

User Manual of RC6F Series Robot Controller 24 Simple Self-Calibration Function



The sampling process is as follows:

1) Select the first constraint point in the robot's dexterous workspace (corresponding to m=1 in the flowchart). When selecting the point, ensure the robot's end effector can reach it in various postures, and the constraint point must be securely fixed without moving during the selection process.
2) The manipulator adjusts the tool's end-effector orientation to achieve vertical alignment with the ground (as shown in Figure ①, approximate precision is acceptable). After completing the orientation adjustment, the manipulator maintains this position while moving along Cartesian coordinates X/Y/Z toward the constraint point. As it approaches the constraint point, the manipulator adjusts the manipulator's single-step distance to 0.1mm, ensuring the tool's tip makes optimal contact with the constraint point. Once the first sampling point of the first constraint point is determined, the manipulator records the current joint pulse values of its position. These values are stored in registers HSD104, HSD124, HSD144, HSD164, HSD184, and HSD204, with the storage address calculated as the base address plus 2*j+30*i INT32S register, where j=0 and i=0,1,2,3,4,5.
3) Move the manipulator along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during subsequent tool pose adjustments. Rotate the tool tip counterclockwise along the X-axis from its initial position (attain an approximate 22.5° angle, as shown in Pose ②). After stabilizing the pose, continue moving the manipulator along the X/Y/Z axes while progressively approaching the constraint point. When nearing the target, set the manipulator's single-step distance to 0.1mm to achieve optimal tool-point alignment. Once the second sampling point of the first constraint point is secured, record the joint pulse values of the manipulator's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then stored in the register address (HSD104/HSD124/HSD144/HSD164/HSD184/HSD204) plus 2*j+30*i INT32S, where j=1 and i=0,1,2,3,4,5.
4) Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during subsequent tool pose adjustments. To modify the tool's end-effector position: First, rotate around the X-axis counterclockwise from position ① to approximately 45° (approximate precision is acceptable), then rotate clockwise 90° around the Z-axis to achieve position ③ (as shown in the diagram). After completing the adjustment, maintain the position while moving the robot along the X/Y/Z axes toward the constraint point. As the robot approaches, adjust the teaching tool's step distance to 0.1mm to ensure optimal contact between the tool tip and the constraint point. Once the third sampling point of the first constraint point is secured, record the joint pulse values of the robot's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then stored in the register address +2*j+30*i INT32S, where j=2 and i=0,1,2,3,4,5.
5) Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the constraint point, preventing potential collisions during subsequent tool pose adjustments. Modify the tool's end-effector posture as follows: First, rotate around the Y-axis clockwise from posture ③ to an angle of approximately 67.5° relative to posture ① (approximate precision is acceptable). Then, rotate 90° clockwise around the Z-axis to achieve posture ④ as shown in the diagram. After completing the posture adjustment, maintain the current configuration while moving the robot along the X/Y/Z axes to progressively approach the constraint point. When nearing the target position, adjust the manipulator's single-step distance to 0.1mm to ensure optimal contact between the tool tip and the constraint point. Once the fourth sampling point of the first constraint point is secured, record the joint pulse values of the robot's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then stored in the register address +2*j+30*i INT32S, where j=3 and i=0,1,2,3,4,5.
6) Guide the robot along the positive Z-axis of Cartesian coordinates to maintain a safe distance from the

constraint point, preventing potential collisions during subsequent tool pose adjustments. Modify the tool's end-effector posture as follows: First, rotate around the X-axis of the world coordinate system clockwise to approximately 90° relative to posture ① (approximate precision is acceptable), then rotate 90° clockwise around the Z-axis of the world coordinate system to achieve posture ⑤ (as shown in the figure). After completing the posture adjustment, maintain the current configuration while moving the robot along the X/Y/Z axes of Cartesian coordinates to progressively approach the constraint point. When nearing the constraint point, adjust the manipulator's single-step distance to 0.1mm to ensure optimal contact between the tool tip and the constraint point. Once the fifth sampling point of the first constraint point is secured, record the joint pulse values of the robot's current position in registers HSD104/HSD124/HSD144/HSD164/HSD184/HSD204. These values are then stored in the register address +2*j+30*i INT32S, where j=4 and i=0,1,2,3,4,5.

If the constraint point shifts during the sampling process, restart the sampling from step (1).
If the robot is on the ground track, the ground track must not move during the sampling process.

7) After completing the five posture point acquisitions, the collection of a single constraint point is finished. The position of the constraint point in the robot workspace is then altered (all X/Y/Z coordinates must be adjusted, with each change being no less than 500mm). Repeat steps (2)-(6) to acquire the second and third constraint points. The pulse values are stored in the register INT32S at the address (register address +2*j+30*i), where j=5,6,...,14 and i=0,1,2,3,4,5.
8) After completing data collection at 15 points, the robot resets to its starting position.

### 24.2.3 Parameter Identification and Selection

1) Set SM3203=1, calculate the robot compensation parameters based on the collected data, and write them into the designated register.
2) The parameter list for each row is as follows:

| Row i | Corresponding parameter | Corresponding register | Register type |
|---|---|---|---|
| 0 | a1 | SFD5004 | float |
| 1 | a2 | SFD5006 | float |
| 2 | a3 | SFD5008 | float |
| 3 | d4 | SFD5010 | float |
| 4 | d6 | SFD5012 | float |
| 5 | k1 | SFD5020 | float |
| 6 | k2 | SFD5022 | float |
| 7 | k3 | SFD5024 | float |
| 8 | k4 | SFD5026 | float |
| 9 | k5 | SFD5028 | float |
| 10 | k6 | SFD5030 | float |
| 11 | α1 | SFD6000 | float |
| 12 | α2 | SFD6002 | float |
| 13 | α3 | SFD6004 | float |
| 14 | α4 | SFD6006 | float |
| 15 | α5 | SFD6008 | float |
| 16 | β2 | SFD6010 | float |

| Row i | Corresponding parameter | Corresponding register | Register type |
|---|---|---|---|
| 17 | d2 | SFD6012 | float |
| 18 | d3 | SFD6014 | float |
| 19 | d5 | SFD6016 | float |
| 20 | a4 | SFD6018 | float |
| 21 | a5 | SFD6020 | float |
| 22 | θ2 | HSD124 | float |
| 23 | θ3 | HSD144 | float |
| 24 | θ4 | HSD164 | float |
| 25 | θ5 | HSD184 | float |
| 26 | TCPx | The TCP position information is updated, and the attitude remains the same as before calibration. The result has been written into the corresponding HD register. | float |
| 27 | TCPy | | float |
| 28 | TCPz | | float |
| 29 | 56 joint coupling ratio | SFD5014 | float |
| 30 | convergence error | HD3858 | float |
| 31 | iterations | HD3860 | float |
| 32 | X | Tool calibration result X | float |
| 33 | Y | Tool calibration result Y | float |
| 34 | Z | Tool calibration result Z | float |
| 35 | Rz | Euler angle Rz, in degrees | float |
| 36 | Ry | Eulerian angle Ry, in degrees | float |
| 37 | Rx | Eulerian angle Rx, in degrees | float |

### 24.2.4 Parameter Compensation

Parameter adjustment order:

1) First, stop the PLC and begin parameter compensation.
2) When the robot is at the zero position, the deceleration ratio and coupling ratio are compensated, and the identification values of these ratios are written into the corresponding SFD registers in the table.
3) Compensate the robot's zero position by taking the calibrated θ2/θ3/θ4/θ5 (pulse form), and write the corresponding zero position (pulse) value into the corresponding HSD register.
4) The remaining parameters of the robot are compensated, the calibrated a1/a2/a3/d4/d6 are written into the corresponding SFD register, the calibrated α1/α2/α3/α4/α5/β2/d2/d3/d5/a4/a5 are written into the corresponding SFD register.
5) The TCP calibration result position and Euler angle are written into the corresponding tool number register.
6) PLC backroll .
7) Reset the manipulator to zero.
8) Compensation complete!

### 24.2.5 Verification of Calibration Results

1) The robot randomly finds 3 constraint points in the flexible workspace.
2) Mobile robot and constraint point fitting.
3) Move the manipulator along the Z-axis by 2mm, which raises the tool tip by 2mm.
4) Move the manipulator along the Cartesian axes RZ, RY, and RX (order adjustable) to observe the constraint effect on the tool tip.

## 24.3 Description of Registers

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SFD6600 | INT | LM iteration count | Default 8 | R/W | Stop PLC restart |
| SFD6601 | INT | LM convergence error setting (sum of squared position errors) | Default 5 | R/W | Stop PLC restart |
| SFD6602 | INT | TCP error impact factor | Default: 10000 | R/W | Stop PLC restart |
| SFD6603 | INT | Zero error influence factor | Default 100 | R/W | Stop PLC re-run |
| SFD6604 | INT | rod length/bias error factor | Default 100 | R/W | Stop PLC restart |
| SFD6605 | INT | alpha/beta error factor | Default 1 | R/W | Stop PLC restart |
| SFD6606 | INT | Reduction ratio error influence factor | Default 1 | R/W | Stop PLC restart |
| SFD6607 | INT | upper limit of rod length/offset deviation | Default 2 | R/W | Stop PLC restart |
| SD6990 | INT16U | Configure joint pulse storage register type 0: D, 1: HD | | R/W | with immediate effect |
| SD6991 | INT16U | Configure the calibration result storage register type: 0: D, 1: HD | | R/W | with immediate effect |
| SD6992 | INT32S | Configure the first address of the robot pose storage register, the type of the pose storage register, INT32S, and handle negative values as absolute values. | | R/W | with immediate effect |
| SD6994 | INT32S | Configure the first address of the calibration result storage register and its type (float). When negative values are input, handle them as absolute values. | | R/W | with immediate effect |
| SM3203 | Bit | self-calibration trigger control bit | | R/W | with immediate effect |

# 25. Safety Zone Functions

## 25.1 Function Overview

The robot's end-effector is programmed with designated working zones, restricted zones, and signal-enabled zones as safety parameters. The working zone and restricted zone maintain inverse relationships: the working zone restricts the robot to movement within predefined spatial boundaries, with boundary detection triggering immediate stoppage. the restricted zone prohibits movement outside the defined spatial model, causing the robot to halt upon boundary entry. the signal-enabled zone imposes no stop restrictions, only generating response signals during entry or exit from specific spatial zones.

## 25.2 Function Specifications and User Instructions

- Set the region number: 0-15, with a total of 16 regions supported.
- Area enable setting: Enables or disables an area. The area's related judgment processing only takes effect after enabling.
- Area shape configuration: Supports three types of selection-working area, forbidden area, and signal working area. The selection process includes output M register address configuration.
- Work area: When the area is enabled, the robot is activated. During the execution of the command movement, if the movement is about to leave the work area, the system will internally execute a pause, stopping at the boundary of the area. Simultaneously, SM3005 is set to 1, and the corresponding warning alarm is output.
- Prohibited Zone: When the zone is enabled, the robot is activated. During the execution of command movements, if the robot is about to enter the prohibited zone, it will perform an internal pause and stop at the zone boundary. Simultaneously, SM3005 is set to 1, triggering the corresponding warning alarm.
- Signal working area: After the area is enabled, the robot detects the real-time relationship between its current position and the working area. When entering the area from outside, the M register is set to 1. when exiting the area from inside, the M register is set to 0.
- Shape settings: Supports two shapes: box and cylinder.
- Area setting: The application layer can obtain the point parameters of the calibration area through teaching or point offset. It is usually two diagonal points in teaching or one point in teaching plus an offset.
- The application layer and the base layer can agree the address of the register of the calibration parameter in each area, and the calibration parameter of each area can be allocated according to the 6 floating point registers.
- For a box-shaped region, the bottom layer only needs the [x, y, z] coordinates of two reference points. For a cylindrical region, the bottom layer only needs the [x, y, z] coordinates of the cylinder's base center, the radius R, and the height H.
- Reference coordinate system: This system specifies the coordinate system for regional calibration point locations. Only base coordinate system, workpiece coordinate system, and world coordinate system are supported.

## 25.3 Format for safety zone parameter points

| Register | Data type | Explain |
|---|---|---|
| D0 | INT16U | Not in use |
| D1 | INT16U | Region enable control word: 0: disable, 1: enable |
| D2 | INT16U | Area type: 1: Working area, 2: Restricted area, 3: Signal working area |
| D3 | INT16U | Shape: 1: Box, 2: Cylinder |
| D4 | INT32U | The output register M offset address is region-specific, and only the signal working area type is valid. |
| D6-D17 | FLOAT | Coordinate parameters for the region: box: [x1, y1, z1, x2, y2, z2], cylinder: [x, y, z, R, H], where H can be negative. |
| D18 | INT16U | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 6: Workpiece coordinate system |
| D19 | INT16U | Workpiece number, 0-19 |

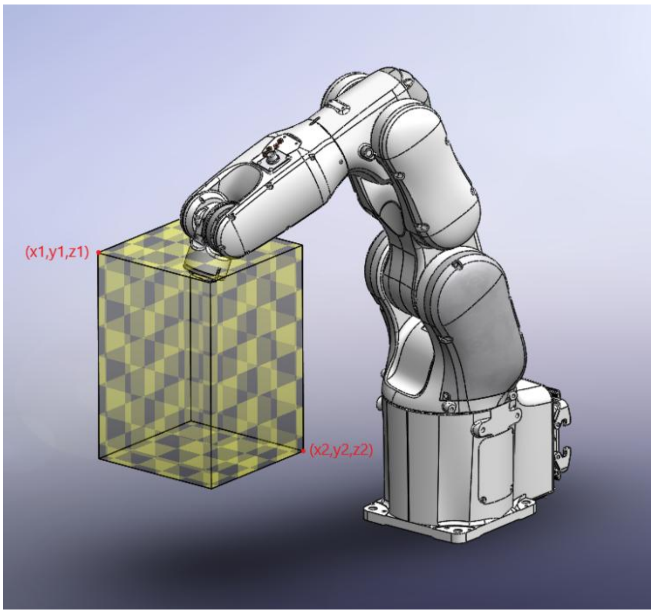
- Each region occupies 20 word registers, with power failure retention registers (e.g., HD or HSD) selectable, and the first address is user-defined. The 16 regions collectively occupy 320 word registers.
- The modified security zone parameters are updated via the SM3150 register.
- The spatial position of the safe area must be perpendicular to the XOY plane of the reference coordinate system.
- The safe zone is determined by prioritizing addresses in ascending order, with smaller addresses given higher priority. This means that only one working or prohibited zone can be detected at a time. Once one zone triggers an alarm, subsequent zones are no longer detected.

## 25.4 Description of Registers

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6280 | INT16U | Region setting parameter storage register type<br>0: HD register<br>1: HSD register | Default 0 | R/W | RC takes effect next time |
| SD6281 | INT16U | Display the current alarm active area or prohibited area number | 0-15 | R | |
| SD6282 | INT32S | The first address of the register storing the region setting parameters<br>For example, if SD6280=0 and SD6282=200, the HD200-HD519 module records parameter settings for 16 zones. Negative values are processed as absolute values. | | R/W | RC will take effect next time |
| SM3150 | Bit | secure area parameter refresh control word | | R/W | with immediate effect |

## 25.5 Function Application

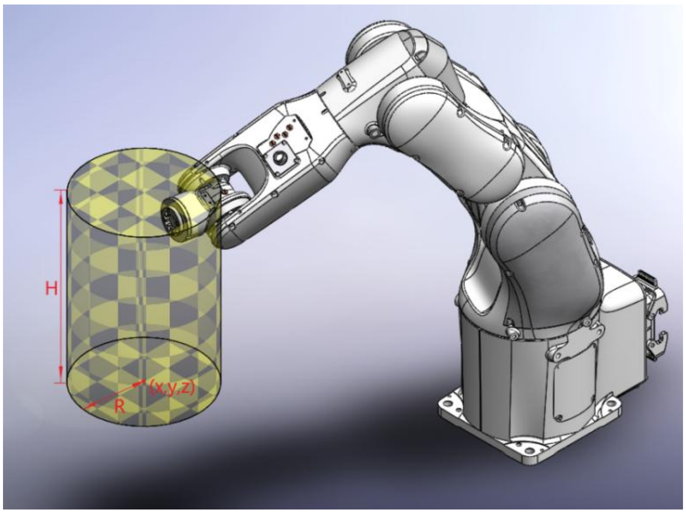
Square safety zone diagram:




Region configuration example

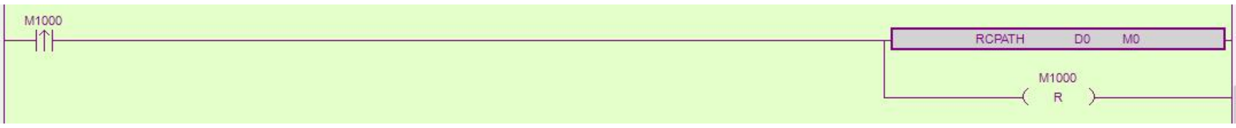
| Register configuration | Data type | Explain |
|---|---|---|
| SD6280=0 | INT16U | Set the first address of the secure zone to HD1000 |
| SD6282=1000 | INT32U | |
| HD1001=1 | INT16U | Enable the security zone |
| HD1002=1 | INT16U | Set this security zone type as a workspace |
| HD1003=1 | INT16U | Set this security zone as a box security zone |
| HD1006=200 | FLOAT | Set safety zone marker coordinates |
| HD1008=-100 | FLOAT | |
| HD1010=100 | FLOAT | |
| HD1012=100 | FLOAT | |
| HD1014=0 | FLOAT | |
| HD1016=-100 | FLOAT | |
| HD1018=0 | INT16U | Set the coordinate system to the base table system |

Schematic diagram of the cylindrical safety zone:



Region configuration example

| Register configuration | Data type | Explain |
|---|---|---|
| SD6280=0 | INT16U | Set the first address of the secure zone to HD1000 |
| SD6282=1000 | INT32U | |
| HD1001=1 | INT16U | Enable the security zone |
| HD1002=1 | INT16U | Set this security zone type as a workspace |
| HD1003=2 | INT16U | Set this security zone as a cylindrical security zone |
| HD1006=300 | float | Set safety zone marker coordinates |
| HD1008=-100 | float | |
| HD1010=0 | float | |
| HD1012=50 | float | |
| HD1014=300 | float | |
| HD1018=0 | INT16U | Set the coordinate system to the base table system |




The source file sample program is as follows:

```
#include"RCPATH.h"
#include"safezone.h"

#define pNum 4
#define SafeZoneNum 2
RcPointInformation RcPoint[pNum] = {
    {{-16.0,-34.0,22.5,0.5,12.0,-17.5},0,0,0,0,5,0,30,80,80,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
    {{405.3,-142.12,266.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0},
    {{325.3,-125.12,289.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0},
    {{402.3,-158.12,325.17,0.217784,-0.882695,-0.203444,-0.363360},0,0,0,0,0,1,30,80,80,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0},
}.
//Secure Zone Parameter Configuration
SafeZoneSet SafeZone[SafeZoneNum] = {
      {0,1,1,1,0,{50,-50,0,-50,50,100},{0,0,0,0,0},0 },
      {0,1,1,2,0,{0,0,0,0,0,0},{300,-100,0,50,300},0 },
}.

void RCPATH(PINT16S W, BIT B)
{
  SecurityZoneCfg(0,1000,SafeZone, (int)(sizeof(SafeZone) / sizeof(SafeZone[0]))).
  if (SD[5500] == 0)
  {
    for (int i = 0. i < SD[5401]. i++)
    {
// Transmit pose data
      *(float*)&D[(SD[5588] + 100 * i + 0)] = RcPoint->targetPos.pos.x.
```

```
        *(float*)&D[(SD[5588] + 100 * i + 2)] = RcPoint->targetPos.pos.y.
        *(float*)&D[(SD[5588] + 100 * i + 4)] = RcPoint->targetPos.pos.z.
        *(float*)&D[(SD[5588] + 100 * i + 6)] = RcPoint->targetPos.orient.q0.
        *(float*)&D[(SD[5588] + 100 * i + 8)] = RcPoint->targetPos.orient.q1.
        *(float*)&D[(SD[5588] + 100 * i + 10)] = RcPoint->targetPos.orient.q2.
        *(float*)&D[(SD[5588] + 100 * i + 12)] = RcPoint->targetPos.orient.q3.

D[(SD[5588] + 100 * i + 14)] = RcPoint->weaveAngleZ. //Arc angle
D[(SD[5588] + 100 * i + 15)] = RcPoint->toolIndex. // Tool ID
D[(SD[5588] + 100 * i + 16)] = RcPoint->posType. // Hand binding
D[(SD[5588] + 100 * i + 17)] = RcPoint->transR. // Attitude transition radius
D[(SD[5588] + 100 * i + 18)] = RcPoint->coordType.//Reference coordinate system
*(float) & D[(SD[5588] + 100 * i + 20)] = RcPoint->targetSpeed. //Maximum running speed
*(float) & D[(SD[5588] + 100 * i + 22)] = RcPoint->targetAcc. // Maximum operational acceleration
*(float) & D[(SD[5588] + 100 * i + 24)] = RcPoint->targetDec. // Maximum deceleration during operation
*(float) & D[(SD[5588] + 100 * i + 26)] = RcPoint->TransErr. // Path transition error
D[(SD[5588] + 100 * i + 28)] = RcPoint->trajType. // Trajectory type
*(INT32U)* & D[(SD[5588] + 100 * i + 84)] = RcPoint->pIndex. //Position index
    }
  }
SM[3160] = 1. //Trigger the RC instruction to initiate movement
}

void SecurityZoneCfg(INT16U Type,INT16U Addr,SafeZoneSet *Zone,int lenth)
{
SD[6280] = Type. //Configure the security zone register type
*(INT32U)* & SD[6282] = Address. //Configure the security area address
  for (int i = 0. i < lenth. i++)
  {
if (SD[6280] == 0) //Check if the register type is HD register
    {
HD[(SD[6282]) + 20 * i + 0] = Zone[i].ZoneNum. // Configuration of security zone sequence number
HD[(SD[6282]) + 20 * i + 1] = Zone[i].ZoneEnable. //Configure the security zone enable status
HD[(SD[6282]) + 20 * i + 2] = Zone[i].ZoneMod. // Configure the security zone type
HD[(SD[6282]) + 20 * i + 3] = Zone[i].ZoneType. // Configure the security zone shape
*(INT32U)* & HD[(SD[6282]) + 20 * i + 4] = Zone[i]. ZoneDeviationAddr. // Sets the register address for the working signal area M
if (Zone[i].ZoneType == 1) //Box-shaped security zone configuration
        {
            *(float*)&HD[(SD[6282]) + 20 * i + 6] = Zone[i].BoxZoneCalibration.SafePoint1.x.
            *(float*)&HD[(SD[6282]) + 20 * i + 8] = Zone[i].BoxZoneCalibration.SafePoint1.y.
            *(float*)&HD[(SD[6282]) + 20 * i + 10] = Zone[i].BoxZoneCalibration.SafePoint1.z.
            *(float*)&HD[(SD[6282]) + 20 * i + 12] = Zone[i].BoxZoneCalibration.SafePoint2.x.
            *(float*)&HD[(SD[6282]) + 20 * i + 14] = Zone[i].BoxZoneCalibration.SafePoint2.y.
            *(float*)&HD[(SD[6282]) + 20 * i + 16] = Zone[i].BoxZoneCalibration.SafePoint2.z.
        }
else if (Zone[i].ZoneType == 2) // Cylinder safety zone configuration
```

```
        {
            *(float*)&HD[(SD[6282]) + 20 * i + 6] = Zone[i].CylinderZoneCalibration.SafePoint1.x.
            *(float*)&HD[(SD[6282]) + 20 * i + 8] = Zone[i].CylinderZoneCalibration.SafePoint1.y.
            *(float*)&HD[(SD[6282]) + 20 * i + 10] = Zone[i].CylinderZoneCalibration.SafePoint1.z.
            *(float*)&HD[(SD[6282]) + 20 * i + 12] = Zone[i].CylinderZoneCalibration.R.
            *(float*)&HD[(SD[6282]) + 20 * i + 14] = Zone[i].CylinderZoneCalibration.H.
        }
        HD[(SD[6282]) + 20 * i + 18] = Zone[i].ZoneCoord.
    }
else//If it's the HSD register, configure as above
    {
        HSD[(SD[6282]) + 20 * i + 0] = Zone[i].ZoneNum.
        HSD[(SD[6282]) + 20 * i + 1] = Zone[i].ZoneEnable.
        HSD[(SD[6282]) + 20 * i + 2] = Zone[i].ZoneMod.
        HSD[(SD[6282]) + 20 * i + 3] = Zone[i].ZoneType.
        *(INT32U*)&HSD[(SD[6282]) + 20 * i + 4] = Zone[i].ZoneDeviationAddr.
        if (Zone[i].ZoneType == 1)
        {
            *(float*)&HSD[(SD[6282]) + 20 * i + 6] = Zone[i].BoxZoneCalibration.SafePoint1.x.
            *(float*)&HSD[(SD[6282]) + 20 * i + 8] = Zone[i].BoxZoneCalibration.SafePoint1.y.
            *(float*)&HSD[(SD[6282]) + 20 * i + 10] = Zone[i].BoxZoneCalibration.SafePoint1.z.
            *(float*)&HSD[(SD[6282]) + 20 * i + 12] = Zone[i].BoxZoneCalibration.SafePoint2.x.
            *(float*)&HSD[(SD[6282]) + 20 * i + 14] = Zone[i].BoxZoneCalibration.SafePoint2.y.
            *(float*)&HSD[(SD[6282]) + 20 * i + 16] = Zone[i].BoxZoneCalibration.SafePoint2.z.
        }
        else if (Zone[i].ZoneType == 2)
        {
                *(float*)&HSD[(SD[6282]) + 20 * i + 6] = Zone[i].CylinderZoneCalibration.SafePoint1.x.
                *(float*)&HSD[(SD[6282]) + 20 * i + 8] = Zone[i].CylinderZoneCalibration.SafePoint1.y.
                *(float*)&HSD[(SD[6282]) + 20 * i + 10] = Zone[i].CylinderZoneCalibration.SafePoint1.z.
                *(float*)&HSD[(SD[6282]) + 20 * i + 12] = Zone[i].CylinderZoneCalibration.R.
                *(float*)&HSD[(SD[6282]) + 20 * i + 14] = Zone[i].CylinderZoneCalibration.H.
            }
            HSD[(SD[6282]) + 20 * i + 18] = Zone[i].ZoneCoord.
        }
    }
    SM[3150] = 1. //Refresh the security zone configuration
}
```

The example of a secure area configuration header file is as follows:

```
#ifndef _SAFEZONE_H_
#define _SAFEZONE_H_
//Square area marker
typedef struct
{
    rbStruct_pos SafePoint1.
```

```
    rbStruct_pos SafePoint2.
}SafeZoneBox.
//Cylindrical area marker
typedef struct
{
    rbStruct_pos SafePoint1.
    float R.
    float H.
}SafeZoneCylinder.
//Safe Zone Status
typedef struct
{
INT16U ZoneNum. //Work area number
INT16U ZoneEnable. //Enable control for the working area
INT16U ZoneMod. //Work area type
INT16U ZoneType. //Work area shape
INT32U ZoneDeviationAddr. // Signal area offset address
SafeZoneBox BoxZoneCalibration. //Box-shaped workspace calibration points
SafeZoneCylinder CylinderZoneCalibration. //Cylinder workspace calibration points
    INT16U ZoneCoord.
}SafeZoneSet.

void SecurityZoneCfg(INT16U Type,INT16U Addr,SafeZoneSet* Zone, int lenth).

#endif
```

# 26. Collision detection function

## 26.1 Function Overview

Robot dynamics examines the relationship between a robot's operational state and required torque. Through dynamic models, collision detection and feedforward control functions can be implemented. The collision detection algorithm serves as a safety mechanism that identifies potential collisions during robotic movement. By monitoring the robot's operational state and calculating theoretical torque using dynamic models, the system compares the actual torque with theoretical values to determine whether a collision has occurred.

The robot collision detection system comprises five core modules: dynamic parameter identification, status monitoring, filtering, torque calculation, collision detection, and collision handling. The dynamic parameter identification module integrates three key components: excitation trajectory design, robot joint Fourier motion drive, and data processing. This module determines the robot's dynamic parameters to minimize model errors, forming the foundation for effective collision detection. The system requires only initial parameter calibration during factory testing, with no need for repeated identification during subsequent operations.

## 26.2 Calibration of the dynamic model

- Collision detection and feedforward functions require the robot's dynamic parameters, including body and tool parameters, to be stored in registers after the SFD7200. For newly manufactured robots or those lacking dynamic parameters for other reasons, the robot's body dynamic parameters can be identified. Tool dynamic parameter identification is not supported.
- The offline identification of the robot's dynamic parameters is typically performed after the factory fatigue test and DH parameter calibration. The identification process must be conducted without any load, ensuring the base height exceeds the normal operational height. During operation, RBSTOP is supported but RBGOON is not. The specific procedure is as follows:
  - S1： Configure the monitoring data register type SD7002 and its starting address SD7004.
  - S2： Set the SD7000 operation count to 1.
  - S3： Set the SD7001 speed ratio to 10.
  - S4： Activate the SM3400 operation flag, which runs for approximately 100 seconds. Monitor the robot's movement to detect any ground contact, collisions, or speed violations. If a dangerous condition triggers RBSTOP, terminate the process and alert the developers. Proceed to the next step if normal operation is confirmed.
  - S5： Set the SD7001's operational speed ratio to 50.
  - S6： Activate the SM3400 operation flag, which takes approximately 20 seconds to execute. Monitor the robot's movement to detect any ground contact or collisions. If a hazard triggers RBSTOP, terminate the process and alert the developers. Proceed to the next step if normal operation is confirmed.
  - S7： Set the SD7001's operational speed ratio to 100.
  - S8： Activate the SM3400 operation flag, which takes approximately 10 seconds to execute. Monitor the robot's movement to detect any ground contact or collisions. If a hazard

triggers RBSTOP, terminate the process and alert the developers. Proceed to the next step if normal operation is confirmed.

- S9： Set the SD7000 operation count to 10.
- S10： The oscilloscope monitors the SD7012 and the feedback position and torque of each axis.
- S11： Activate the SM3400 operation flag, with an estimated runtime of 10×10 seconds.
- S12： Save the oscilloscope monitoring data as a CSV file and provide it to the developers.
- S13： The dynamic parameters identified by the monitoring data are written into the SFD7200+20*a+2*b register, and the threshold parameters are written into the SFD7140+2*a register, where a represents the axis number and b represents the dynamic parameter number.

## 26.3 Collision Detection Function

- The collision detection function supports all motion modes and verifies that the dynamic parameters in the SFD have been written. The process is as follows:
  - S1： Configure the parameters of the bus PDO register (SD7013-SD7020).
  - S2： Configure the collision detection enable HM register and the starting address SD7008.
  - S3： Set the starting address of the M flag collision axis to SD7010.
  - S4： configure the collision detection sensitivity HD register.
  - S5： Set the starting address of the sensitivity register SD7006.
- During robot operation, parameters in the bus PDO register and the start address SD register cannot be modified and must be updated before the SM3001's rising edge. In contrast, axis enable HM register and axis sensitivity HD register modifications take effect immediately. When a collision is detected, the corresponding M register is set to ON, allowing the robot's movement to be halted via this flag.
- When collision detection is enabled and the robot is moving, the torque deviation value is displayed in real time on the SD7060. If this value exceeds the preset collision detection threshold, the system will detect a collision. For repetitive motions, the value from the initial few cycles can be set as the collision detection threshold.

## 26.4 Collision Handling

When the bus communication task (interpolation task) is running and the corresponding collision signal M register is detected as ON, it will stop according to the set mode. Currently, two modes are supported. To enable collision handling, configure SD7074 and SD7075.

Mode 1: Slow Stop

The internal execution is suspended, functioning similarly to the external rbstop command. The suspension duration is set by SD4171, and the system will continue running along the original trajectory for a certain distance after the longer stop. However, it can resume movement by using the rbgoon command after stopping.

Mode 2: Normal stop

For forward collision (collision force and motion direction are the same), the internal execution queue is paused, and the total pause duration is 40 bus cycles. The stop time is short, but it will continue to run along the original trajectory for a certain distance, and it cannot continue to move after stopping.

For reverse collision (the collision force is opposite to the direction of motion), the queue first slows down, then retraces along the original trajectory, and finally stops at a position before the collision occurs. After stopping, the queue cannot continue moving.

In any stop mode, collision detection during movement is not supported, so mode 2 cannot prevent collisions with the rear during retraction.

## 26.5 Data Monitoring

- The dynamic function needs to be debugged to get a better effect, and the dynamic parameters need to be monitored during debugging.
- According to SD7002 and SD7004, the starting register of the storage of the dynamic monitoring parameters, the relevant offset point, a set of 20 words, and a set of points corresponding to one axis.

| Register | Data type | Explain |
|---|---|---|
| D0-D4 | float | Joint given position, given speed, given acceleration, angle mode |
| D6 | float | The joint torque corresponding to the current motion state, in Nm |
| D8 | float | The Deviation of the Joint's Current Feedback Torque and the Predicted Torque |
| D10-D14 | float | Joint feedback position, feedback speed, feedback acceleration, and angle mode |
| D16 | float | The torque currently fed back by the joint, in Nm |
| D18 | float | Predictive Torque of Joint Current Feedback State |

For example, when SD7002 equals 0 and SD7004 equals 1000, the floating-point value D[1000+6] represents the specified torque for joint 1, while D[1000+20+6] represents the specified torque for joint 2.

## 26.6 Description of Registers

| Register | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD7000 | INT16U | The number of Fourier trajectory runs can be modified during execution. If it is less than the current count, the segment will stop after completion. | Default 1 | R/W | with immediate effect |
| SD7001 | INT16U | The Fourier trajectory speed ratio. A value of 100 takes 10 seconds to run one Fourier trajectory. Changes made during the run are invalid. | 1-100 Default 50 | R/W | Active next time |
| SD7002 | INT16U | monitoring data storage register type<br>0: D register<br>1: HD register | Default 0 | R/W | Active next time |
| SD7004 | INT32S | The monitoring data storage register starts at the specified address and tracks dynamic operational parameters, requiring a continuous sequence of 20×axis words.<br>SD7004=0 indicates no monitoring. When a negative value is entered, it is treated as an absolute value. | | R/W | Active next time |

| Register | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD7006 | INT32S | The collision detection sensitivity HD register's starting address is processed as an absolute value when negative input is provided.<br>The sensitivity address of each joint is HD[SD7006+n]<br>The collision sensitivity of each joint is stored in a continuous HD register, type INT16U, with a range of 1-400.<br>for instance：<br>SD7006=1000, HD[1000]=100, HD[1001]=200, indicating joint 1's collision sensitivity is 100 and joint 2's sensitivity is 200. Higher sensitivity values indicate lower sensitivity.<br>SD7006=0 indicates no detection | | R/W | Active next time |
| SD7008 | INT32S | The collision detection enables the starting address of the HM register. When a negative value is input, it is processed as an absolute value.<br>HM[SD7008] General Collision Detection Switch<br>n Collision Detection Switch of HM[SD7008 +1 +n] Joint | | R/W | Active next time |
| SD7010 | INT32S | The starting address of the M flag for axis collision. A negative value is treated as an absolute value.<br>M[SD7010] Total collision flag<br>The [SD7010 +1 +n] joint n has collided | | R/W | Active next time |
| SD7012 | INT16U | Number of Fourier trajectories currently running | | R | |
| SD7003 | INT16U | The time constant of the excitation trajectory and the cycle of the excitation trajectory running at full speed | Default 10 | R/W | Active next time |
| SD7013 | INT16U | bus configuration register type<br>0: D register<br>1: HD register | Default 0 Match the bus mapping address | R/W | Active next time |
| SD7014 | INT32S | Set the bus configuration register starting address. If a negative value is entered, it will be treated as an absolute value. | | | |
| SD7016 | INT16U | bus configuration register axis offset | | | |
| SD7017 | INT16U | bus configuration register feedback position offset | | | |
| SD7018 | INT16U | bus configuration register feedback speed offset | | | |
| SD7019 | INT16U | bus configuration register feedback acceleration offset | | | |
| SD7020 | INT16U | bus configuration register feedback torque offset | | | |
| SD7030+2*i (i=0,1...maxAxisNum-1) | float | The dynamic threshold for the i-th axis is read from the SFD7140 during PLC operation, with changes taking effect in real time. | | R/W | Active next time |
| SD7050+i (i=0,1...maxAxisNum-1) | INT16U | Maximum consecutive collisions on the i-th axis | Default 15 | R/W | Active next time |

| Register | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD7060+i (i=0,1...maxAxisNum-1) | INT16U | The percentage deviation of the i-th axis's torque from the rated torque during motion. If this value exceeds the user-defined collision detection sensitivity for each axis, a collision signal will be generated.<br>When performing repetitive motions, you can set the sensitivity of axis collision detection for the first few rounds of motion. | | R/W | Active next time |
| SD7074 | INT16U | Collision Handling Mode:<br>0-Do not process internally, set M register only<br>1-Set M register, pause<br>2-Set M register, normal stop retraction | | R/W | Active next time |
| SD7075 | INT16U | Number of stop pullback cycles: The interval between the position where the reverse collision pullback stops and the current position in the queue. | Default 0 Maximum 99 | R/W | Active next time |
| SD7077 | INT16U | Collision type detected<br>0-No collision<br>1-forward collision<br>2-Reverse collision | | R | |
| SD7078 | INT16U | Current motion status with collision detection enabled<br>0-Still<br>1-normal interpolation<br>2-Slow down<br>3-cue hold<br>4. Retraction pause | | R | |
| SFD7000 +20*i+2*j (i=0,1...maxAxisNum-1. j=0,1...9) | float | The j-th Fourier trajectory parameter on the i-th axis, with 10 parameters per axis | | R/W | Active next time |
| SFD7120+2*i (i=0,1...maxAxisNum-1) | float | rated torque of the i-th shaft | MN·m | R/W | Active next time |
| SFD7140+2*i (i=0,1...maxAxisNum-1) | float | Dynamic threshold for axis i | MN·m | R/W | Active next time |
| SFD7160 | INT16U | Feedback data filter coefficient. The higher the value, the better the filtering effect. | Default: 50ms | R/W | Active next time |
| SFD7200 +20*i+2*j (i=0,1...number of maximum axes minus 1. j=0,1...9) | float | The j-th dynamic parameter of the i-th axis, with 10 parameters per axis<br>MDH modeling method, the representation of the connecting rod in the corresponding joint coordinate system, and the sequence of corresponding dynamic parameters<br>Lxx Lxy Lxz Lyy Lyz Lzz mx my mz m | | R/W | Active next time |

| Register | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SFD7320 +20*i+2*j (i=0,1... maximum load count-1. j=0,1...9) | float | The j-th dynamic parameter of the i-th load<br>i: 0-3 0-1 indicates tool load<br>j=5 is the moment of inertia under the center of mass (mg·m²)<br>j=6,7,8: The load center of gravity is represented in the robot's end coordinate system (m)<br>j=9 Load mass (mg)<br>j=0:4 Not in use<br>i=2 is the load on the forearm at joint 3 | | R/W | Active next time |
| SFD7400 +10*a+2*b (i=0,1...maxaxes-1. j=0,1...4) | float | The j-th friction parameter on the i-th axis<br>Each axis can have up to 5 friction parameters. The current friction model has only 2 parameters.<br>F=Fc*sign(dq)+Fv*dq | | R/W | Active next time |
| SM3400 | Bit | Fourier instruction start control bit | | R/W | with immediate effect |
| SM3401 | Bit | Fourier instruction run status bit | | R | |
| SM3402 | Bit | Fourier instruction exception status bit | | R | |
| SM3410 | Bit | load identification instruction start control bit | | R/W | with immediate effect |
| SM3411 | Bit | Load identification operation status bit | | R | |
| SM3412 | Bit | Load identification abnormal status bit | | R | |
| SM3600+i (i=0,1... maximum number of axes) | Bit | Total collision signal and collision signal of each axis<br>The collision detection function operates identically to the M register.<br>The internal setting is applied after the collision stops, and no external setting is required. | | R | |

# 27. Accessibility Check Module

## 27.1 Function Overview

Provide a point accessibility check module to verify whether the point exceeds the reach space of the robotic arm and the joint soft limit. The input point supports all coordinate systems: 0: Base coordinate system, 1: World coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system.

## 27.2 Point Format

| Register | Data type | Explain |
|---|---|---|
| D0-D11 | FLOAT | Pose coordinates [x, y, z, rz, ry, rx], in mm or degrees. Or joint angles, in degrees |
| D12 | INT16U | Whether the point is reachable: 0: Not reachable, 1: Reachable |
| D13 | INT16U | Unreachable point type: 1: Cartesian coordinate out of range, not within the robotic arm's reach. 2: Joint angle exceeds the soft limit |
| D14 | | Obligate |
| D15 | INT16U | Tool id, 0-19 |
| D16 | INT16U | Hand ligament |
| D17 | | Obligate |
| D18 | INT16U | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system |
| D19 | INT16U | Workpiece number, 0-19 |

## 27.3 Description of Registers

| Address | Type | Explain | Unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SM3015 | BOOL | accessibility check word | | R/W | with immediate effect |
| SD5447 | INT16U | The register type for storing the coordinates of the accessibility check input points | 0：D<br>1：HD | R/W | SM3015 will take effect next time |
| SD5448 | INT32S | The first address of the storage register for the input point coordinates is checked for accessibility. If a negative value is entered, it is treated as an absolute value. | | R/W | Next time SM3015 is triggered, it will take effect |

## 27.4 Function Applications

This feature checks whether a manually entered point falls within the robot's operational range, using the Scara model with a 600mm arm span as an example.

| Register configuration | Data type | Explain |
|---|---|---|
| SD5447=1 | INT16U | Set the first address of the configuration accessibility checkpoint to HD1000 |
| SD5448=1000 | INT32U | |

| Register configuration | Data type | Explain |
|---|---|---|
| HD1000=400 | float | Set accessibility checkpoint coordinates |
| HD1002=200 | float | |
| HD1004=-60 | float | |
| HD1006=80 | float | |
| HD1012=1 | float | Returns the value after triggering SM3015.1 indicates reachable. |
| HD1013=0 | float | Returns the value after SM3015 is triggered. 0 indicates normal operation. |
| HD1015=0 | float | The tool ID is set to 0 |
| HD1016=0 | float | The left hand is the left hand. |
| HD1018=0 | float | Set the coordinate system as the base coordinate system |
| HD1019=0 | float | The workpiece number is set to 0 |

| Register configuration | Data type | Explain |
|---|---|---|
| SD5447=1 | INT16U | Set the first address of the configuration accessibility checkpoint to HD1000 |
| SD5448=1000 | INT32U | |
| HD1000=400 | float | Set accessibility checkpoint coordinates |
| HD1002=600 | float | |
| HD1004=-60 | float | |
| HD1006=80 | float | |
| HD1012=0 | float | Returns the value after triggering SM3015.0 indicates unreachable. |
| HD1013=1 | float | After triggering SM3015, the return value is 1: Cartesian coordinates exceed the range and are beyond the reach of the robotic arm. |
| HD1015=0 | float | The tool ID is set to 0 |
| HD1016=0 | float | The left hand is the left hand. |
| HD1018=0 | float | Set the coordinate system as the base coordinate system |
| HD1019=0 | float | The workpiece number is set to 0 |

| Register configuration | Data type | Explain |
|---|---|---|
| SD5447=1 | INT16U | Set the first address of the configuration accessibility checkpoint to HD1000 |
| SD5448=1000 | INT32U | |
| HD1000=100 | float | Set accessibility checkpoint coordinates |
| HD1002=200 | float | |
| HD1004=-60 | float | |
| HD1006=80 | float | |
| HD1012=0 | float | Returns the value after triggering SM3015.0 indicates unreachable. |
| HD1013=1 | float | Return value after triggering SM3015:2: Joint angle exceeds the soft limit |
| HD1015=0 | float | The tool ID is set to 0 |
| HD1016=0 | float | The left hand is the left hand. |
| HD1018=5 | float | Set the coordinate system to joint coordinate system |
| HD1019=0 | float | The workpiece number is set to 0 |

User Manual of RC6F Series Robot Controller 28 Positive Solution Calculation Function

# 28. Correct calculation function

## 28.1 Function Overview

Provide the forward kinematics calculation for the robot. Enter the joint angle of the robot to get the end effector pose, and check whether the end effector pose corresponding to the joint angle meets the expected requirements.

## 28.2 Point Format

| Register | Data type | Explain |
|---|---|---|
| D0-D19 | float | User input joint angle |
| D20-D39 | float | The corresponding end pose generated after triggering SM3013, in the format [x, y, z, Rz, Ry, Rx] |

## 28.3 Description of Registers

| Address | Type | Explain | Unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SM3013 | BOOL | positive solution calculation control word | | R/W | with immediate effect |
| SD5443 | INT16U | The type of storage register for input point coordinates in forward solution calculation | 0：D<br>1：HD | R/W | SM3013 will take effect next time |
| SD5444 | INT32S | Enter the first address of the storage register for the input point coordinates. When a negative value is entered, it is treated as an absolute value. | | R/W | SM3013 will take effect next time |

## 28.4 Function Applications

After writing the joint coordinates, trigger SM3013 to obtain the end-effector coordinates at this joint angle.

| Register configuration | Data type | Explain |
|---|---|---|
| SD5443=1 | INT16U | Set the first address of the correct solution function point to HD1000 |
| SD5444=1000 | INT32U | |
| HD1000=40 | float | Set the joint angle for the forward function |
| HD1002=20 | float | |
| HD1004=-50 | float | |
| HD1006=60 | float | |
| HD1020=386.5251 | float | After triggering SM3013, return the spatial coordinate value |
| HD1022=447.0741 | float | |
| HD1024=-50 | float | |
| HD1026=120 | float | |

User Manual of RC6F Series Robot Controller 29 Coordinate System Conversion Function

# 29. Coordinate system conversion function

## 29.1 Function Overview

Provide coordinate system conversion to transform the current position across different coordinate systems and tools. Configure CF parameters to check the arrival status of the target position.

## 29.2 Point Format

D0-D19 is the original coordinate information. D20-D39 is the converted target coordinate information. D40-D59 is the joint angle of the previous point.

| Register | Data type | Explain |
|---|---|---|
| D0-D11 | float | The original pose coordinates [x, y, z, Rz, Ry, Rx], in mm and degrees. or joint angles, in degrees |
| D12-D14 | INT16S | The original joint attributes CF1, CF4, CF6② |
| D15 | INT16U | Original tool number, 0-19 |
| D16 | INT16U | primitive hand |
| D17 | INT16U | When the target coordinate reference system (D38) is the joint coordinate system, it can be represented as whether the point is reachable: 0: not reachable, 1: reachable. |
| D18 | INT16U | Original coordinate reference system, 0: Base coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system |
| D19 | INT16U | Original workpiece number, 0-19 |
| D20-D31 | float | Target pose coordinates [x, y, z, Rz, Ry, Rx] in mm and degrees. or joint angles in degrees |
| D32-D34 | INT16U | The original joint attributes CF1, CF4, CF6② |
| D35 | INT16U | Target tool number, 0-19 |
| D36 | INT16U | target hand |
| D37 | INT16U | When the target coordinate reference system (D38) is the joint coordinate system, it can be represented as: 1: Cartesian coordinate overrun, 2: Joint angle soft limit overrun. |
| D38 | INT16U | Target coordinate reference system, 0: Base coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system |
| D39 | INT16U | Target workpiece number, 0-19 |
| D40-D51 | float | The joint angle of the previous point, in degrees ③ |

①: When the target coordinate reference system (D38) is the joint coordinate system, the inverse solution function is activated, with the inverse solution angle dependent on the hand system. In this scenario, if the original coordinate reference system (D18) is the joint coordinate system, the hand system is determined by the target hand system (D36). Conversely, if D18 is not the joint coordinate system, the hand system is determined by the original hand system (D16).

②③: Typically, no configuration is required. When the target reference frame D38 is a joint coordinate system and the original coordinate system D18 is not a joint coordinate system, the coordinate transformation function functions as the robot's inverse kinematics to determine the corresponding joint angles for the pose. The inverse kinematics angles are related to the target hand frame D36. If the soft limit of the joint on axis 1, axis 4, or axis 6 exceeds ±180°, the corresponding joint attributes D12-D14

User Manual of RC6F Series Robot Controller 29 Coordinate System Conversion Function

must be configured. If the soft limit of the joint on axis 1, axis 4, or axis 6 exceeds ±180° but the corresponding joint attributes are not configured, the system will default to selecting the joint angle with the closest starting point within the joint limit based on the shortest distance principle. Therefore, the starting joint angle D40-D51 must be configured.

More hand binding and joint attribute settings:

| Original coordinate system<br>Target coordinate system | Non-articular coordinate system | Joint coordinate system |
|---|---|---|
| Non-articular coordinate system | You may omit D12-14, D16, and D40-51.<br>D32-34 equals D12-14, and D36 equals D16. | D16 needs to be configured.<br>The configurations of D12-14, D40-51, and D36=D16 can be customized as needed.<br>D32-34 is calculated according to the target joint angle. |
| Joint coordinate system | You may omit D12-14, D16, and D40-51.<br>D36 and D32-34 are calculated according to the original joint angle. | You can skip configuring D12-14 and D16.<br>D36 needs to be configured.<br>You can configure D32-34 and D40-51 as needed. |

## 29.3 Description of Registers

| Address | Type | Explain | Unit | Read-write | Effective date |
|---|---|---|---|---|---|
| SM3019 | Bit | Coordinate conversion control word | | R/W | With immediate effect |
| SD5482 | INT16U | Coordinate conversion input point storage register type | 0：D<br>1：HD | R/W | SM3019 will take effect next time |
| SD5480 | INT32S | Convert coordinates to the first address of the input point's storage register. When a negative value is entered, it is treated as an absolute value. | | R/W | SM3019 will take effect next time |

# 30. Timer function

## 30.1 Function Overview

For some beat tests or processes, you need to obtain the movement time of one cycle and provide a timing function. Up to ten timers can be used simultaneously.

## 30.2 Description of Registers

| Address | Type | Explain | Unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SD6980 | INT16U | Timer on/off storage address type | 0：M | R/W | With immediate effect |
| SD6982 | INT32S | The timer turns on and off the storage address. When a negative value is entered, it is treated as an absolute value. | | R/W | With immediate effect |
| SD6984 | INT16U | Timer function switch: 0: off, 1: on | Default 0 | R/W | With immediate effect |
| SD6986 | INT16U | Timer time storage address type | 0：D<br>1：HD | R/W | With immediate effect |
| SD6988 | INT32S | Timer time storage address. If a negative value is entered, it is treated as an absolute value. | | R/W | With immediate effect |

## 30.3 Function Applications

| Register configuration | Data type | Explain |
|---|---|---|
| SD6980=0 | INT16U | The address for configuring timer on/off functions is M1000. The subsequent addresses from M1000 to M1010 contain the switch control words for the 10 timers respectively. |
| SD6982=1000 | INT32S | |
| SD6984=1 | INT16U | Timer function master switch |
| SD6986=0 | INT16U | Set the timer time storage address to D1000-D1019 |
| SD6988=1000 | INT32S | |
| D1000+2*i(i<10) | float | Timer stores time values |

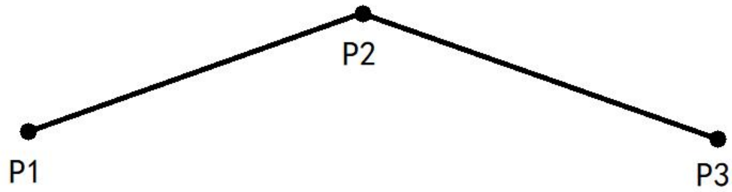
When SD6984 is set to 1, the timer function switch is activated. The remaining timers are controlled by their respective M registers. After setting the corresponding M register, the timer will start counting under two scenarios:

- When the robot has not started movement (SM3001 is OFF), the system records the current time and waits for the timer to expire.
- When the robot is performing RC motion and SM3001 is ON, the timer starts recording time during real segment number switching.

After resetting their respective M registers, the timer terminates under two conditions:

- When the robot has not started movement or the SM3001 is OFF after movement ends, the current time is immediately recorded and the total timing duration is calculated.
- When the robot is performing RC motion and SM3001 is ON, it records time during real segment number switching and calculates the total timing duration.

After the timer ends, the time is saved in the corresponding register.




For example: When the robot is at point P1, reset M1000. The robot starts moving and the timer begins. If M1000 is reset during the movement from P1 to P2, the timer will end when the robot reaches P2, and the time from P1 to P2 is output. If M1000 is reset during the movement from P2 to P3, the timer will end when the robot reaches P3, marking the end of the entire movement, and the total time from P1 to P3 is output.

# 31. Belt-following function

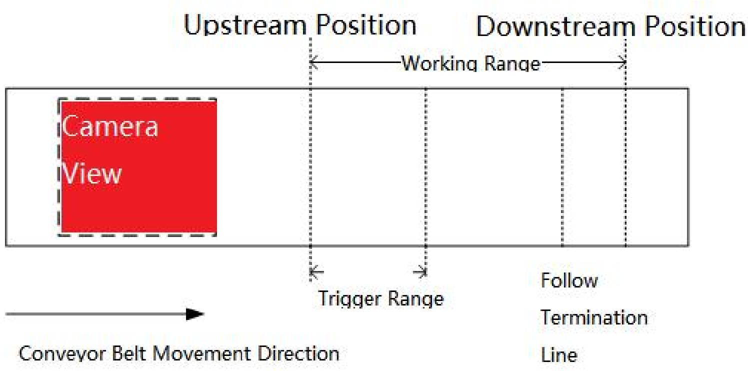
## 31.1 Function Overview

When the workpiece enters the trigger zone, the robot departs from its stationary position to perform a tracking operation. It initiates a chase grab along the belt's movement direction. During the chase phase, the robot accelerates from its starting position until achieving relative stationary position with the belt, awaiting synchronization commands. Upon reaching synchronization, it moves from the initial synchronization position to the target grab position, completing the grab action before decelerating. Once fully synchronized, the robot immediately decelerates from the grab completion position to a complete stop, preparing for subsequent operational procedures.

The standard workflow for belt tracking system operation is as follows:

- S1: The camera takes the image of the workpiece.
- S2: The visual supercomputer processes the photo information to get the position and posture of the workpiece.
- S3: The controller calculates the real-time position of the workpiece. When the robot is in a state that allows query and the workpiece enters the trigger range to meet the following condition, the robot will start following the action.
- S4: The robot finally returns to the "waiting position" after four processes of catching up, synchronization, deceleration and returning.

The schematic diagram of the main following parameter and robot following scene is as follows:




Mainly follow the parameter diagram

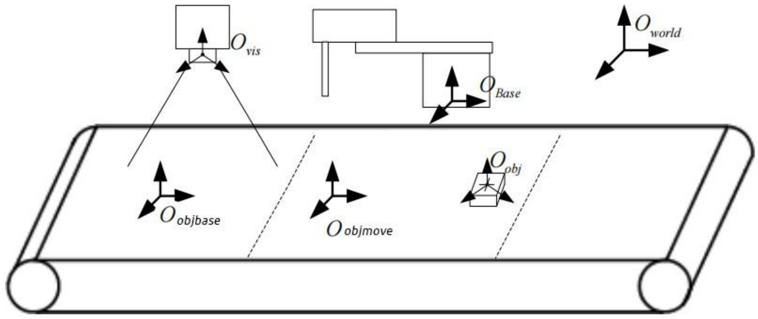
The motion of the belt workpiece is actually the coordinate obtained by the transformation between the coordinate systems, and its schematic diagram and meaning are as follows:

{0base}, base coordinate system

{0objbase}, a static belt coordinate system, is bound to a workpiece coordinate system and linked to the belt

{0objmove}, the moving belt coordinate system, bind the belt position

{0obj}, the moving workpiece coordinate system, and the belt position




Relationships between coordinate systems in the manufacturing process

## 31.2 Configuration initialization

The following external configuration tasks must be completed before using the belt follow function:

① Install the belt and its drive mechanism, ensuring the belt is parallel to the x-axis or y-axis of the robot's base coordinate system.

② The belt motor is debugged to ensure stable speed during normal operation, maintaining it within the range of [10 mm/s, 500 mm/s].

③ The belt calibration includes the belt direction vector calibration and the belt encoder pulse equivalent calibration.

④ The hand-eye relationship between the robot and the camera is obtained by camera calibration.

⑤ Work boundary and workpiece information entry address configuration.

After configuration, the robot's movement flow is as follows:

Step 1: Move to the waiting point.

Step 2: Check the workpiece. If successful, proceed to Step 3. otherwise, return to Step 2.

Step 3: Switch to the dynamic coordinate system, and the robot moves in sync with the belt's speed.

Step 4: Relative motion to the workpoint, typically the center of the workpiece surface.

Step 5: Move the relative motion to the safe point, typically positioned above the workpiece surface.

Step 6: If the required number of captures is not met, proceed to Step 7. otherwise, proceed to Step 8.

Step 7: Check the workpiece. If successful, proceed to Step 4. otherwise, proceed to Step 8.

Step 8: Switch to the static coordinate system (release the synchronization release signal), the robot decelerates and stops.

Step 9: If the required number of captures is not met, proceed to Step 2 (or Step 11). otherwise, proceed to Step 10.

Step 10: Proceed to the process critical point, typically the material release point.

Step 11: Return to the waiting point.

The above process is supplemented as follows:

(1) Additional movements (Cartesian movements) and external IO operations may be inserted between Step 4 and Step 5.

(2) The query operation in Step 7 can be performed once or multiple times.

(3) In Step 8, the synchronization release signal must be promptly released after the required movement is completed.

(4) The return-to-point motion in Step 11 may be composed of multiple segments.

(5) The query operation of workpiece mainly includes: ① when the robot is at the waiting point, ② when the last grab is not in synchronization, ③ when the last grab is not in the deceleration stop process of switching to the static coordinate system, ④ when the robot is moving back to the waiting point.

(6) When the system is in synchronization, the underlying system performs overload detection. If overload is detected, the following process will be forcibly stopped and the vehicle will decelerate and stop immediately.

## 31.3 RCPATH instruction data point format

When the wrist anomaly error occurs during RcPath operation, you can enable the wrist anomaly avoidance function.

| RCPATH instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-D12 | Target point coordinates [x, y, z, q0, q1, q2, q3] | FLOAT |
| D14 | Arc angle, range [0-360]*100 | INT16U |
| D15 | Tool ID, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-100 to indicate the percentage of CP time. Set 0 or-1 to indicate the zero-speed point. | INT16S |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and posture ③ | FLOAT |
| D22 | Maximum acceleration percentage of running path and posture | FLOAT |
| D24 | Maximum deceleration percentage of running path and posture | FLOAT |
| D26 | Path transition error percentage | FLOAT |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: PTP, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: PTP with joint speed planning (unfixed trajectory). 1: PTP with joint speed planning (fixed trajectory). 2: PTP with Cartesian speed planning (fixed trajectory). 3: PTP with acceleration optimization planning (fixed trajectory) ⑥ | INT16U |
| D84 | Point number, user-defined, monotonically increasing, may be discontinuous | INT32U |
| D86 | The number of transition points for attitude is up to 5⑦. | INT16U |
| D95 | Follow motion type: 0: No follow motion. 1: Chase motion. 2: Synchronized motion. 3: Deceleration motion | INT16U |
| D96 | The number of the material grabbed in a single chase action | INT16U |
| D97-D99 | Obligate | |

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. When the external axis collaborative motion types are 1, 2, or 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to reference coordinate system D18=6, D19=1 (workpiece coordinate system). When setting D18=4 (tool coordinate system), the point position information is relative displacement. For instance: If the robot's tool-end position is (400,200,100) and the point position x, y, z information is set to (50,20,-20), the robot will reach (450,220,80) after movement.

③: When speed, acceleration, or both are not configured, the system automatically adjusts the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points: an arc auxiliary point and an arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint position is the starting point position rather than the endpoint information in the point input. Trajectory type 7 represents an incomplete intersection line, composed of the current point, auxiliary point, and endpoint. Trajectory type 8 represents a complete intersection line, consisting of the current point, auxiliary point 1, auxiliary point 2, auxiliary point 3, and the endpoint, with the requirement that the starting and endpoint poses must be identical. If the specified number of points is not met, the trajectory is forcibly converted to a straight line.

⑤: CP Time Transition Percentage parameter, indicating the percentage of remaining deceleration time for the current segment before initiating the next segment.

⑥: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: speed planning is performed in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: speed planning is performed in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values represent the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid this, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑦: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated according to the path length ratio, with the P3-P6 posture change being equally distributed.

To designate the current trajectory as a follow motion, both the belt function enable bit and the belt motor connection enable bit must be activated. Set point D18 to 2, and configure D95 to 1, 2, or 3 based on current requirements. If only one register is set while the other remains at 0, the system will fail to recognize this as a follow motion and trigger alarm 10227. To execute multiple gripping actions within a single follow period (between a follow motion and a deceleration phase), different D96 values must be configured during synchronized motion.

When D18=2 and D95=1, this segment is a catch-up movement. The robot will move in the direction of the belt until its speed matches the belt's speed. If no additional movement is added, the robot will continue moving at the current speed until the lower limit is triggered. The catch-up segment defaults to a straight-line trajectory, and the value of D28 does not need to be set.

When D18=2 and D95=2 are set, the motion segment operates in synchronous mode. In this configuration, the position data from D0 to D12 in the point information represents the relative displacement of the workpiece during the current grab. If the position is set to (0,0,0), the motion endpoint will align with the workpiece's center. Synchronous motion segments support only three trajectory types: straight lines, arcs, and non-movement segments (D28=1, 2, 9). The JUMP type is limited to Mode 2 (three straight-line segments). External signal interactions and processout functions can be integrated with synchronous motion segments. If no subsequent movements are initiated after the synchronization phase, the robot will continue moving at its current speed until the lower limit is triggered.

When D18=2 and D95=3 are set, this phase is a deceleration phase. The robot will decelerate from its current speed to a complete stop. The catch-up phase defaults to a straight-line trajectory, and the value of D28 is optional. The deceleration phase marks the end of a single catch-up cycle. For multi-belt gripping, an interlaced deceleration phase must be used to complete a single belt's gripping action.

## 31.4 Description of Registers

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SFD9100+40i (i=0,1,2,3) | INT16U | Belt type: 0: straight, 1: disc | | R/W | Reactivation process switch |
| SFD9101+40i (i=0,1,2,3) | INT16U | Coordinate system serial number of the workpiece bound by the belt | | R/W | Reactivation process switch |
| SFD9102+40i (i=0,1,2,3) | INT16U | Types of belt encoders<br>0: Servo motor encoder<br>1: External encoder (high-speed counter) | | R/W | Reactivation process switch |
| SFD9103+40i (i=0,1,2,3) | INT16U | address of value storage register of belt encoder | | R/W | Reactivation process switch |
| SFD9104+40i (i=0,1,2,3) | float | Pulse equivalent of the belt encoder | | R/W | Reactivation process switch |
| SFD9106+40i (i=0,1,2,3) （obligate） | INT16U | The sensitivity coefficient of the encoder pulse fluctuation protection mechanism ranges from 0 to 9. A higher value indicates greater sensitivity. The default value is 9. | | R/W | Reactivation process switch |
| SFD9107+40i (i=0,1,2,3) | INT16U | Coordinate system type for visual points<br>0: Base coordinate system<br>1: Belt-bound workpiece coordinate system<br>2: Visual coordinate system | | R/W | Reactivation process switch |
| SFD9108+40i (i=0,1,2,3) | INT16U | Visual coordinate system index, values range from 0 to 4 | | R/W | Reactivation process switch |
| SFD9109+40i (i=0,1,2,3) | INT16U | Queue capacity, maximum 200 | | R/W | Reactivation process switch |
| SFD9110+40i (i=0,1,2,3) | float | Distance for repeated removal, default value 10 | mm | R/W | Reactivation process switch |
| SFD9112+40i (i=0,1,2,3) | float | upper bound of work | | R/W | Reactivation process switch |
| SFD9114+40i (i=0,1,2,3) | float | lower bound of work | | R/W | Reactivation process switch |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SFD9116+40i (i=0,1,2,3) ( obligate ) | float | Trigger upper bound | | R/W | Reactivation process switch |
| SFD9118+40i (i=0,1,2,3) | float | trigger lower bound | | R/W | Reactivation process switch |
| SFD9120+40i (i=0,1,2,3) | float | stop distance | mm | R/W | Reactivation process switch |
| SFD9122+40i+2j (i=0,1,2,3. j=0,1,2,...,5) | float | Pose compensation, with 6 floats in total. The Euler angle for pose compensation is measured in degrees, and the default value is 0. | mm or degree | R/W | Reactivation process switch |
| SFD9134+40i (i=0,1,2,3) | float | time bias | s | R/W | Reactivation process switch |
| SFD9136+40i (i=0,1,2,3) | INT16U | The filter window size for the belt position filter, with a maximum of 500 and a minimum of 5 | | R/W | Reactivation process switch |
| SFD9260+2i (i=0,1,2,...,9) | float | Workpiece height (relative to the belt surface, where the belt surface height is 0) corresponds to 10 workpiece types, totaling 10 floats | mm | R/W | Reactivation process switch |
| SD7100+10i (i=0,1,2,3) | INT16U | Number of workpieces to photograph, minimum 1, maximum 10 | | R/W | With immediate effect |
| SD7101+10i (i=0,1,2,3) | INT16U | The D register's first address (occupying 20 words with unused reserved space) stores workpiece information, including belt number (INT16U, D0), workpiece type (INT16U, D1), and Eulerian angle coordinates (6 floats, D2~D12). | | R/W | With immediate effect |
| SD7150 | INT16U | Query the belt number for the workpiece. The maximum value is 3, and the minimum value is 0. | | R/W | With immediate effect |
| SD7151 | INT16U | The type of workpiece you are querying, with a maximum of 9 and a minimum of 0 | | R/W | With immediate effect |
| SD7152 ( obligate ) | INT16U | Query the starting address of the D register storing workpiece information (occupying 20 words, with unused reserved space). The workpiece information includes belt number (INT16U, D0), workpiece type (INT16U, D1), and Euler angle coordinates (6 floats, D2~D12). | | R/W | With immediate effect |
| SD7153 ( obligate ) | INT16U | Track following processing method<br>0: Smoothing (sacrifices tracking accuracy)<br>1: Do not smooth | | R/W | With immediate effect |
| SD7160+4i (i=0,1,2,3) | double | belt feedback position | mm | R | |
| SD7180+4i (i=0,1,2,3) | double | belt filter location | mm | R | |
| SD7200+40i (i=0,1,2,3) | float | belt feedback position | mm | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD7202+40i (i=0,1,2,3) | float | belt filter location | mm | R | |
| SD7204+40i (i=0,1,2,3) | float | belt feedback speed | mm/s | R | |
| SD7206+40i (i=0,1,2,3) | float | belt filter speed | mm/s | R | |
| SD7208+40i (i=0,1,2,3) | INT16U | Current work queue length | | R | |
| SD7209+40i (i=0,1,2,3) | INT16U | Available capacity in the workpiece queue | | R | |
| SD7210+40i+2j (i=0,1,2,3. j=0,1,2,...,5) | float | The robot tool's end position is given in the workpiece coordinate system, with 6 floats. | | R | |
| SD7222+40i+2j (i=0,1,2,3. j=0,1,2,...,5) | float | Feedback of the robot tool end position to the workpiece coordinate system, 6 floats in total | | R | |
| SD7234+40i (i=0,1,2,3) | float | The belt's average feedback speed refreshes every 200ms | mm/s | R | |
| SD7236+40i (i=0,1,2,3) | float | The belt filter averages the speed and refreshes every 200ms | mm/s | R | |
| SD7360 | INT32U | Number of successful follow-up items | | R | |
| SD7362 | INT32U | Number of failed workpieces | | R | |
| SD7364 | INT16U | Follow result of the previous workpiece | | R | |
| SD7365 | INT16U | robotry | | R | |
| SD7370+2i (i=0,1,2,...,5) | float | The robot tool's end position in the workpiece coordinate system, 6 floats in total | | R | |
| SD7382+2i (i=0,1,2,3) | float | Time required for processing the photo information of the workpiece | ms | R | |
| SD6930+2i (i=0,1,2,...,5) | float | Eulerian Angle Coordinates of Workpiece When It Leaves the Queue | mm or degree | R | |
| SM3440 | bit | Belt follow function control bit, active on high level, off by default | | R/W | With immediate effect |
| SM3441+i (i=0,1,2,3) | bit | Photo signal, high level active | | R/W | With immediate effect |
| SM3445+i (i=0,1,2,3) | bit | Photo processing complete signal, high level valid | | R/W | With immediate effect |
| SM3449 | bit | Workpiece sort control bit, active on high level | | R/W | With immediate effect |
| SM3450 | bit | Workpiece query signal, high level valid | | R/W | With immediate effect |
| SM3451 | bit | Workpiece query success signal, high level valid | | R/W | With immediate effect |
| SM3452 | bit | Failed workpiece query signal, high level valid | | R/W | With immediate effect |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SM3453+i (i=0,1,2,3) | bit | Enable control bit for the belt, high level active | | R/W | With immediate effect |
| SM3458 | bit | Compensate queue length control bit, active on high level | | R/W | With immediate effect |
| SM3490 （Obligate） | bit | The belt stop function control bit is active at high level and is disabled by default. | | R/W | With immediate effect |
| SM3495 （Obligate） | bit | The robot signals the workpiece grasping action, with high level being valid. | | R/W | With immediate effect |
| SM3460 | bit | belt following motion state position | | R | |
| SM3461 | bit | belt following interpolation status bit | | R | |
| SM3462 | bit | belt following catch-up state | | R | |
| SM3463 | bit | belt following synchronization status bit | | R | |
| SM3464 | bit | belt following deceleration state position | | R | |
| SM3465 | bit | state of completion of following motion | | R | |

# 32. The function of avoiding strangeness

## 32.1 Function Overview

When a robotic arm encounters a singularity, its trajectory becomes erratic, with joint velocities exceeding control limits and potentially damaging hardware. This necessitated the development of a singularity avoidance mechanism. The function primarily addresses point-motion and RC command operations: SCARA models and serial six-joint models can utilize point-motion singularity avoidance, while RC commands are exclusively applicable to serial six-joint models.

## 32.2 Description of Registers

When the single-step and continuous point-motion singular avoidance function is enabled, the robot can avoid joint overspeeding caused by singularities when approaching the singular region, but this comes at the cost of reduced pose accuracy. The relevant registers are as follows:

| Address | Type | Explain | Unit | Read - write | Trigger time |
|---|---|---|---|---|---|
| SM3220 | Bit | Singular avoidance function switch: 0: off, 1: on | | R/W | with immediate effect |
| SFD8280 | float | Setting of the threshold for shoulder abnormalities | Default value: 0.2 | R/W | with immediate effect |
| SFD8282 | float | Setting the threshold for elbow anomaly detection | Default value: 0.2 | R/W | with immediate effect |
| SFD8284 | float | Setting of the threshold for wrist abnormalities | Default value: 0.2 | R/W | with immediate effect |
| SFD8286 | float | Nominal damping coefficient setting | Default value: 0.2 | R/W | with immediate effect |
| SFD8288 | float | Translation singular error compensation coefficient | Default value: 300 | R/W | with immediate effect |
| SFD8290 | float | Rotation of the singular error compensation coefficient | Default value 100 | R/W | with immediate effect |
| SFD8292 | float | Setting the Strange Threshold of Scara Robot | Default value 1 | R/W | with immediate effect |
| SFD8294 | float | Nominal damping coefficient setting for Scara robot | Default value: 0.4 | R/W | with immediate effect |

## 32.3 RCPATH instruction data point format

When the wrist anomaly error occurs during RcPath operation, you can enable the wrist anomaly avoidance function.

| RCPATH instruction point information | | |
|---|---|---|
| Operand | Function | Type |
| D0-D12 | Target point coordinates [x, y, z, q0, q1, q2, q3] | FLOAT |
| D14 | Arc angle, range [0-360]*100 | INT16U |
| D15 | Tool ID, 0-19 | INT16U |
| D16 | Hand strap, 0-7 | INT16U |
| D17 | Transition parameter ①: Set 0-100 to indicate the percentage of CP time. Set 0 or-1 to indicate the zero-speed point. | INT16S |
| D18 | Reference coordinate system: 0: Base coordinate system, 1: World coordinate system, 2: Belt coordinate system, 4: Tool coordinate system, 5: Joint coordinate system, 6: Workpiece coordinate system ② | INT16U |
| D19 | Workpiece number, 0-19 | INT16U |
| D20 | Maximum speed percentage of running path and posture ③ | FLOAT |
| D22 | Maximum acceleration percentage of running path and posture | FLOAT |
| D24 | Maximum deceleration percentage of running path and posture | FLOAT |
| D26 | Path transition error percentage | FLOAT |
| D28 | Path type: 1: Straight line, 2: Circular arc ④, 3: PTP, 7: Incomplete intersection line, 8: Complete intersection line, 9: Non-moving segment, 10: Complete circle | INT16U |
| D29 | 0: PTP with joint speed planning (unfixed trajectory). 1: PTP with joint speed planning (fixed trajectory). 2: PTP with Cartesian speed planning (fixed trajectory). 3: PTP with acceleration optimization planning (fixed trajectory) ⑥ | INT16U |
| D84 | Point number, user-defined, monotonic increasing, may be discontinuous | INT32U |
| D86 | The number of transition points for attitude is up to 5⑦. | INT16U |
| D94 | Wrist singular avoidance: 0: singular avoidance is off. 1: singular avoidance is on, and the singular avoidance is performed on the joint angle change range of 4/5/6 of the starting and ending points. 2: singular avoidance is on, and the singular avoidance is performed with the wrist hand system unchanged. 3: singular avoidance is on, and the singular avoidance is performed with the wrist hand system changed. | INT16U |
| D97-D99 | Obligate | |

①: When the current segment is a PTP trajectory, or both the current and next segments are single-attitude trajectories, the transition parameter for the current segment is the time transition percentage. For external axis cooperative motion types 1, 2, and 4, the CP transition between PTP and non-PTP trajectories is disabled.

②: The external axis coordinate system is a special workpiece coordinate system. When SD5593 matches the workpiece number, the workpiece coordinate system becomes the end-effector coordinate system of the manipulator. When SD5593 differs from the workpiece number, the workpiece coordinate system becomes the manipulator base coordinate system (including ground track). For motion types 2 and 3, trajectory pose assignment is independent of D0-D12 coordinate values and depends solely on the starting pose. For motion types 1 and 2, the point reference coordinate system can only be set to the workpiece coordinate system corresponding to its external axis system. The external axis coordinate system must match the reference coordinate system when motion types 1 and 2 are configured. For example, SD5593=1 corresponds to setting the reference coordinate system as D18=6, D19=1 (workpiece coordinate system). When D18=4 (tool coordinate system) is set, the point position information is relative displacement. For instance: If the current tool-bearing position of the robot end-effector is

(400,200,100), setting the point position coordinates (x, y, z) as (50,20,-20) will move the robot to (450,220,80) after movement.

③: When speed, acceleration, or both are not configured, the system automatically adjusts the preset values of speed and acceleration to a range of 0.01 to 100 by referencing the previous data point.

④: The arc and full circle commands consist of two machining points: an arc auxiliary point and an arc endpoint. When only one arc trajectory type point exists between adjacent machining points, the trajectory type is modified to a straight line. For a full circle, the endpoint position is the starting point position rather than the endpoint information in the point input. Trajectory type 7 represents an incomplete intersection line, composed of the current point, auxiliary point, and endpoint. Trajectory type 8 represents a complete intersection line, consisting of the current point, auxiliary point 1, auxiliary point 2, auxiliary point 3, and the endpoint, with the requirement that the starting and endpoint poses must be identical. If the specified number of points is not met, the trajectory is forcibly converted to a straight line.

⑤: CP Time Transition Percentage parameter, indicating the percentage of remaining deceleration time for the current segment before initiating the next segment.

⑥: When mode 0 is selected, the speed planning mode remains the implementation scheme of the old PTP function: speed planning is performed in joint space with non-fixed trajectories. When mode 1 is selected, the speed planning mode becomes the implementation scheme of the new PTP function 1: speed planning is performed in joint space with fixed trajectories. When mode 2 is selected, speed planning is performed in Cartesian space with fixed trajectories. When mode 0 or 1 is selected, the percentage values of speed, acceleration, and double acceleration represent the upper limits of joint speed, joint acceleration, and joint double acceleration. When mode 2 is selected, these percentage values represent the upper limits of linear speed, linear acceleration, and linear double acceleration in Cartesian space. During mode 2 operation, actual linear speed, linear acceleration, and linear double acceleration may fall short of the specified values due to constraints from position curve length, angular velocity, angular acceleration, and angular double acceleration. This may also lead to speed planning based on angular displacement. To avoid such issues, one can reduce the specified linear speed, increase angular increments of the first, second, and third axes, or decrease angular increments of the first, second, and third axes. However, this may increase motion time. For applications requiring high time efficiency, modes 0 or 1 can be directly adopted. When mode 3 is selected, speed planning is performed in joint space with fixed trajectories and optimal acceleration, requiring configuration of the maximum torque register SFD7170.

⑦: The number of points affected by forward or backward transitions. For example, with 10 target points in total, if the bottom-level judgment of the 6th target point's posture change can be allocated forward, and if the D86 setting for P6 is 3, then the postures of P3, P4, and P5 will be proportionally linearly allocated according to the path length ratio, with the P3-P6 posture change being equally distributed.

# 33. Point offset function

## 33.1 Function Overview

The offset value of machining point is dynamically compensated under Cartesian coordinate system and joint coordinate system.

The current Rcpath function only supports compensating offset values for the x, y, and z axes of machining points in Cartesian coordinates. To address this limitation, the system should allow adding offsets for either the base coordinate system (Cartesian) or joint coordinate system. This feature enables users to construct offset points from known positions, eliminating the need for creating new points or performing manual teaching.

## 33.2 Instructions for Use

When executing RC commands, the configuration parameters enable offset compensation for machining points. This allows both Cartesian coordinate system (base coordinate system) pose compensation and joint coordinate system angle compensation. The compensation format is [x, y, z, Rx, Ry, Rz], with Euler angles being the default for attitude compensation.

Usage:

S1: Configure the corresponding RC point parameters.

S2: Activate the SD6974 Cartesian and Joint Offset function switches.

S3: Configure SD6972 to select coordinate system type and SD6960+2(i) offset compensation value

S4: Trigger the RC instruction to issue the SM3160 bit

## 33.3 Description of Registers

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6960+2(i) (i=0,1,2...5) | float | Offset compensation for machining point coordinates<br>Compensates for offset values of the [x, y, z, Rx, Ry, Rz] coordinates of the machining point. The default attitude compensation is in the form of Euler angles. | Degree or mm | R/W | The RC instruction takes effect after execution |
| SD6972 | INT16U | Offset coordinate system type | 5: Joint Coordinate System<br>Other values: Base coordinate system | R/W | The RC instruction takes effect after execution |
| SD6974 | INT16U | dynamic offset compensation function switch | Only 1: Effective<br>None of the other values are valid | R/W | The RC instruction takes Effect after execution |

1. When the offset coordinate system is the base coordinate system, if the original RC point is not in the base coordinate system, the original point information is processed into the base coordinate value without tools for Cartesian offset.

When the offset coordinate system is joint coordinate system, if the original RC point is not in joint coordinate system, the original point information will be processed into joint value in joint coordinate system for joint offset.

Currently, the offset function only supports offsets in the base coordinate system and joint coordinate system, and the type of offset coordinate system is selected based on user settings.

The offset function is also supported when using tool numbers. It converts the pose originally associated with a tool number to the base coordinate system without the tool, rather than offsetting based on the tool's coordinate system. Therefore, to view the offsetted pose, you need to switch the tool number to the corresponding one, rather than checking the coordinate value changes under the default tool 0.

2. At the origin, applying joint offsets to points in the base coordinate system may result in infinite solutions during inverse kinematics calculations due to the robot's singular position. Under the same hand coordinate system, the system selects the solution with the shortest distance, causing corresponding variations in the calculated poses Rz, Ry, and Rx. This is a normal phenomenon.

3. For the CF (Center of Force) scenario, joint offsets are incorporated during point position addition in the base coordinate system. The joint angles obtained from the inverse kinematics solution in the base coordinate system are then corrected based on the CF values in the point position data.

4. When the Euler angle offset exceeds a certain threshold, the Cartesian transformation may yield discrepancies between the post-offset Euler angle and the sum of the original Euler angle and the offset value. This occurs because quaternion-to-Euler angle conversion can produce two possible solutions, though both ultimately converge to the same angle.

## 33.4 Function Applications

1. Cartesian (Base Coordinate) Offset Cartesian/Joint

(1) Cartesian offset Cartesian

For instance, the Cartesian coordinate P0 (1092.53,350.73,556.07,36.58, -53.85,166.58),

Set SD6974 to 1, SD6972 to 0, and configure the offset values SD6960 to SD6970 as (10,10,10,10,10,10).

The Cartesian coordinates P0 after offsetting are (1102.53,360.73,566.07,46.58, -43.85,176.58).

(2) Cartesian offset joint

For example, the Cartesian coordinate P1 (1103.55,177.83,721.59,75.91, -50.93,119.16)

corresponding to joint coordinates (9.13, -73.23, -14.66,0.71,15.77, -36.08)

Set SD6974 to 1, SD6972 to 5, and configure the offset values SD6960 to SD6970 as (10,10,10,10,10,10).

The offset joint coordinates P1 are (19.13, -63.23, -4.66,10.71,25.77, -26.08).

Special case: When manually configuring CF,

P1 (-211.31, 551.99, -66.92, 7.24)

(With joint coordinates (120.01, -19.78, -66.66, -93), cf4=-1), perform offset adjustment on joint 4 at P1. Set SD6974=1, SD6972=5, and SD6966=10, resulting in the adjusted joint coordinates as

P1 coordinates (120.01, -19.78, -66.66, -83). When cf4=0 is set, inverse calculation may yield J4=-93 or 267 (indicating two solutions differing by 360 degrees, both representing the same J4 position within the ±360 limit).

To resolve this, the robot's current position must be analyzed. For example, if J4=149.08, moving the joint from 149.08 to 267 requires 117.92 units. Under cf4=0 (shortest path), the inverse solution selects the shortest path (J4=267), then offsets by 10 to obtain 277. The adjusted P1 coordinates become (120.01, -19.78, -66.66,277). cf1 and cf6 follow similar logic.

2. Joint Offset Cartesian/Joint
(1) Cartesian joint offset
For example, the joint coordinates P2 (20.05, -65.59, -5.35,6.89,10.28, -31.52)
corresponding to the Cartesian coordinates (1132.39,416.52,470.9, 63.85, -52.81,143.97)
Set SD6974 to 1, SD6972 to 0, and configure the offset values SD6960 to SD6970 as (10,10,10,10,10,10).
The Cartesian coordinates P2 after offsetting are (1142.39,426.52,480.9, 73.85, -42.81,153.97).

(2) Joint displacement
For example, the joint coordinates P3 (24.36, -60.01, -1.08,18.29,30.36, -0.43),
Set SD6974 to 1, SD6972 to 5, and SD6960 to 10.
The offset joint coordinates P3 are (34.36, -60.01, -1.08,18.29,30.36, -0.43).

3. Tool/workpiece coordinate offset Cartesian/Joint
(1) Cartesian tool/workpiece coordinate offset
For example, Cartesian coordinates with tools or Cartesian coordinates in the workpiece coordinate system.

P4（1122.84，509.09，528.95，105.76，-32.12，132.92），

Set SD6974 to 1, SD6972 to 0, and the offset values SD6960 to SD6970 to (10,10,10,10,10,10).
The Cartesian coordinates P4 after offset are (1132.84,519.09,538.95,115.76, -22.12,142.92).

(2) Tool/workpiece coordinate joint
For example, Cartesian coordinates with tooling or Cartesian coordinates in the workpiece coordinate system.

P5（1023.72，609.56，584.55，138.63，11.38，122.47），

corresponding to joint coordinates (29.58, -65.88, -15.14,24.21,25.31, -131.86)
Set SD6974 to 1, SD6972 to 5, and the offset values SD6960 to SD6970 to (10,10,10,10,10,10).
The offset joint coordinates P5 are (39.58, -55.88, -5.14,34.21,35.31, -121.86)

# 34. Handwheel retraction function

## 34.1 Function Overview

The manual return function operates on RC commands, supporting both forward and backward trajectory interpolation through direction and speed registers. It enables continuous motion based on input speed and direction. While CP transitions are supported during forward/reverse movement, external signal interaction commands, external axes, and swing arc functions are not available in either direction. When enabled, the function supports only 200-point loadout with identical configuration and usage to RC commands. During activation, RC commands function as continuous trajectory instructions.

## 34.2 Description of Registers

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SM3390 | Bit | Handwheel rollback function enable control word | 0 | R/W | With immediate effect |
| SM3391 | Bit | Hand Return Function Status Word | 0 | R | |
| SM3392 | Bit | The first trigger flag for the handwheel rollback function. When the handwheel function is enabled, the variable is initialized to 1. It resets when movement begins. | 1 | R | |
| SM3393 | Bit | Set the hand's return function trajectory end flag. This value is set when the point trajectory of the current movement direction has completed. It resets after the directional change movement response. | 0 | R | |
| SD7940 | Float | Hand cycle retreat speed, the percentage of the current instruction speed, similar to SD5540 | 0 | R/W | With immediate effect |
| SD7942 | INT16U | Hand rotation direction: 0-no direction, 1-forward, 2-reverse | 0 | R/W | With immediate effect |

## 34.2 Function Applications

This feature requires RC commands. After configuring multiple RC points, connect the handwheel to activate the motion function.

S1: Send the handwheel rotation speed as a percentage variable to SD7940, and input the specified rotation direction value to SD7942.

S2: Enable the SM3390 control word.

S3: Rotate the handwheel forward to check if the robot moves to the preset position.

S4: Rotate the handwheel in reverse direction and verify if the robot can return to its original position along the forward trajectory.

For handwheel debugging, when the register value of handwheel speed or direction changes, the system performs corresponding speed adjustment or interpolation direction adjustment. This mainly includes handwheel start, handwheel stop, and speed change. All three processes use variable cycle speed control and support setting the speed to 0, i.e., stopping motion.

The specific speed control strategy is:

$$usetime = \left| \frac{|SD[4171]| * (|SD[7940]| - speed)}{100} \right|$$

Number of variable cycles = time / 2

The speed must not exceed 100% and the usage time must be at least 10 milliseconds.

Usage Notes

1、 The sense and till functions are not supported

2、 Rbstop and rbgoon are not supported. The handwheel can control start/stop.

3、 External shafts and swing arcs are not supported

4、 Due to internal errors causing the SM3004 and SM3005 bits to be set, the system cannot proceed with movement even after clearing the error alarm, and the handwheel function is reset.

5、 RC instruction original register performance

| Address | Wheel function enabled |
|---|---|
| SM3016 | Reset only when the first forward interpolation occurs in this segment. The signal remains unchanged during reverse and re-interpolation. |
| SM3014 | Reset only when the first forward interpolation occurs in this segment. The signal remains unchanged during reverse and re-interpolation. |
| SM3161 | After the handwheel is opened, 3161 resets. When the RC command is triggered, it remains set even after movement stops. |
| SD5462 | The real-time interpolated segment number changes with the motion path and displays both forward and reverse directions. |
| SD5567 | The real-time interpolation track type adapts to the motion path and displays both forward and reverse directions. |
| The segment numbers in the remaining RC will only be updated during the initial forward motion. | |

6、 When the handwheel function is activated, the SM3016 and SM3042 signals in the non-motion command 'wait' are adjusted to set after interpolating the first point, while SD6288 is disabled. The handwheel controls the motion to the 'wait' trajectory segment, with the waiting time calculated from the first stop. When the handwheel is not activated, normal timing occurs and signals are transmitted as usual.

# Appendix

## Appendix 1 Summary of Registers

### SFD register

| Register address | Type | Definition | Value/unit | Read-write | Come into force Opportunity |
|---|---|---|---|---|---|
| SFD803 | INT16U | Enable logging after power-on with value 2<br>When sm3004 or sm3005 is set to 1, the log function will be automatically disabled. | | R/W | Re-enable the robot or power on the PLC |
| SFD690 | INT32U | Firmware generation date (hexadecimal) | YTD | R | Stop PLC restart |
| SFD2990 | INT16U | Control cycle | Default: 2000ms | R/W | Stop PLC restart |
| SFD2991 | INT16U | Number of stops | give tacit consent to 20 | R/W | Stop PLC restart |
| SFD2992 | INT16U | Retry count for communication | give tacit consent to 3 times | R/W | Stop PLC re-run |
| SFD3000+60*i (i=0,1,2... maximum number of axes minus 1) | INT16U | Axis control mode | Default 0, location control mode | R/W | Stop PLC restart |
| SFD3001+60*i (i=0,1,2... maximum number of axes minus 1) | INT16U | Motor type | Default 3, multi-turn absolute encoder | R/W | Stop PLC restart |
| SFD3002+60*i (i=0,1,2... maximum number of axes minus 1) | INT32U | The number of encoder lines per revolution, which is the count value fed back per full rotation of the encoder. Set this register according to the actual number of encoder lines on the motor. (for example, if the motor encoder is a 17-bit encoder, set this register to 2^17, i.e., 131072) | The encoder's feedback count after one rotation defaults to 17-bit encoder pulse 131072 | R/W | Stop PLC restart |
| SFD3004+60*i (i=0,1,2... maximum number of axes minus 1) | INT32U | The movement quantity/turn, the reference unit of motion (screw pitch), is measured in pulses. The number of pulses required for the motor to complete one full rotation is specified. The plc sends exactly the number of pulses equal to the value set in this register to rotate the motor. | The value matches SFD3002+60*i, with the default 17-bit encoder 131072 pulse | R/W | Stop PLC re-run |

| Register address | Type | Definition | Value/unit | Read-write | Come into force Opportunity |
|---|---|---|---|---|---|
| SFD3014+60*i (i=0,1,2... maximum number of axes minus 1) | INT32S | Minimum position of motor | give tacit consent to -1000000000 pulses | R/W | Stop PLC restart |
| SFD3016+60*i (i=0,1,2... maximum number of axes minus 1) | INT32S | Maximum position of motor | give tacit consent to 10000000000 pulses | R/W | Stop PLC re-run |
| SFD3018+60*i (i=0,1,2... maximum number of axes minus 1) | INT32U | The motor's maximum speed is set based on its highest or rated speed. Exceeding this limit will cause the motor to operate at the maximum speed. | The system typically operates at either the maximum or rated speed, with a default 17-bit encoder and motor speed of 3000 r/min. 6553600 pulses/s | R/W | Stop PLC re-run |
| SFD3047+60*i (where i=0,1,2,..., with the maximum number of axes being-1) | INT16U | Motor direction logic<br>Configure the joint rotation in both forward and reverse directions to ensure it matches the specified model. | 0: Positive logic<br>1: Anti-Logic | R/W | Stop PLC restart |
| SFD3360+60*i (i=0,1...number of external shaft joints minus 1) | INT16U | Operation mode of external shaft motor | Default 0 | R/W | Stop PLC restart |
| SFD3361+60*i (i=0,1...number of external shaft joints minus 1) | INT16U | Types of external shaft motors | Default 3 | R/W | Stop PLC restart |
| SFD3362+60*i (i=0,1...number of external shaft joints minus 1) | INT32U | Number of wires in the motor encoder | Default 131072 | R/W | Stop PLC re-run |
| SFD3364+60*i (i=0,1...number of external shaft joints minus 1) | INT32U | Mobile volume/1 circle | Default 131072 | R/W | Stop PLC restart |
| SFD5000 | INT16U | Robot type | | R/W | Stop PLC restart |
| SFD5002+2i (i=0,1...7) | float | Dh parameters for robots | | R/W | Stop PLC restart |
| SFD5020+2*i (where i=0,1,2... to the maximum number of axes minus 1) | float | Gear ratio of each joint<br>Consistent with the actual deceleration ratio of the joint | Rotate joint: Circle/Circle<br>Moving joint: mm/turn | R/W | Stop PLC re-run |

| Register address | Type | Definition | Value/unit | Read-write | Come into force Opportunity |
|---|---|---|---|---|---|
| SFD5040+2i (i=0,1,2... maximum number of axes minus 1) | float | Maximum speed of each joint (joint movement)<br>The motor-side sfd3018+60*i determines the maximum motor speed, which is then calculated for the joint output side. | m/s or mm/s | R/W | Stop PLC re-run |
| SFD5060+2i (i=0,1,2... maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint speed | $m/s^2$ or $mm/s^2$ | R/W | Stop PLC restart |
| SFD5080 | float | Maximum speed at the end, [linear velocity]<br>The linear velocity is determined by joint 1,2,3 | mm/s perhaps linear measure /s | R/W | Stop PLC restart |
| SFD5090 | float | Maximum acceleration at the end, [linear acceleration]<br>The default setting is 5-10 times the linear speed | $mm/s^2$ perhaps linear measure $/s^2$ | R/W | Stop PLC restart |
| SFD5105 | INT16U | Selection of robot inverse solving algorithm | Default 0 | R/W | Stop PLC re-run |
| SFD5106 | INT16U | Rcpath point attitude representation type: 0: Unit quaternion, 1: Euler angle. The default is 0, valid only for quadrotor robots. | Default 0 | R/W | Stop PLC restart |
| SFD5107 | INT16U | Attitude can be optimized. Switch 0: Off 1: On | | R/W | Next trigger |
| SFD5120 | float | Cartesian point step size | mm | R/W | Stop PLC restart |
| SFD5122 | float | Percentage of coordinate point movement speed | % | R/W | Stop PLC restart |
| SFD5124 | float | Percentage of acceleration of the coordinate point | % | R/W | Stop PLC re-run |
| SFD5130 | float | Joint dynamic step length | linear measure | R/W | Stop PLC restart |
| SFD5132 | float | Percentage of the axis point's moving speed | % | R/W | Stop PLC restart |
| SFD5134 | float | Percentage of the axis point's dynamic acceleration | % | R/W | Stop PLC restart |
| SFD5138 | INT16U | Extend the queue length, ranging from 5 to 50 | Default 20 | R/W | Stop PLC re-run |
| SFD5140+2*i (i=0,1,2... maximum number of axes minus 1) | float | Maximum angular limits for each joint<br>The set value should be less than the actual maximum mechanical limit angle of each joint. | linear measure | R/W | Stop PLC restart |
| SFD5160+2*i (i=0,1,2... maximum number of axes minus 1) | float | Minimum angular limit of each joint<br>Set the value to be greater than the actual mechanical minimum limit angle of each joint. | linear measure | R/W | Stop PLC re-run |

| Register address | Type | Definition | Value/unit | Read-write | Come into force Opportunity |
|---|---|---|---|---|---|
| SFD5230+2*i (i=0,1,2... maximum number of axes minus 1) | float | The maximum speed of each joint (cartesian motion) is typically set to match the sfd5040+2i configuration. | linear measure /s | R/W | Stop PLC re-run |
| SFD5250+2i (i=0,1,2... maximum number of axes minus 1) | float | Maximum acceleration of each joint<br>The default setting is 5-10 times the joint acceleration, with specific values depending on the actual operating conditions of the body. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC restart |
| SFD5270 | float | The maximum terminal acceleration, [linear acceleration, angular acceleration], is typically set to 5-10 times the linear acceleration. The specific parameter values depend on the actual operating conditions of the device. | $m/s^3$ or $mm/s^3$ | R/W | Stop PLC restart |
| SFD5370+i (i=6,7...10) | INT16U | Is the external axis controlled?<br>0: Uncontrolled<br>Controlled | Default 1 | R/W | Stop PLC restart |
| SFD5500+2i+12j (i=0,1...5) (j=0,1...19) | float | Tool parameters: The relative pose of the tool tip at the end of the robot flange, represented by [x, y, z, rz, ry, rx] in mm or degrees, where j is the tool number. | Default 0 | R/W | SM3018 will take effect next time |
| SFD5740+2i+12j (i=0,1...5) (j=0,1...19) | float | Workpiece parameters: The relative pose of the workpiece coordinate system in the robot base coordinate system, represented as [x, y, z, rz, ry, rx] in mm or degrees, where j is the workpiece number. | Default 0 | R/W | SM3018 will take effect next time |
| SFD5752+2*i (i=0,1...5) | float | The position and orientation (x, y, z, rz, ry, rx) of the base coordinate in the world coordinate system for the positioner system 1 | | R/W | Stop PLC restart |
| SFD5764+2*i (i=0,1...5) | float | The position and orientation (x, y, z, rz, ry, rx) of the 2-axis coordinate system in the world coordinate system | | R/W | Stop PLC restart |
| SFD6000+2*i (i=0,1,2...10) | float | Kinematic model calibration compensation values | A total of 11 floats are compensated, with specific parameters as specified in the calibration manual. | R/W | Stop PLC re-run |
| SFD6030 | float | Maximum speed of quaternion terminal attitude | Default: 100 degrees/s | R/W | Stop PLC restart |
| SFD6032 | float | Maximum acceleration/deceleration of quaternion terminal attitude | Default: 1000 $degrees/s^2$ | R/W | Stop PLC restart |
| SFD6034 | float | Maximum acceleration of quaternion terminal attitude | Default 10000 degrees/s3 | R/W | Stop PLC re-run |
| SFD6040+2*i (i=0,1,2,3) | float | Orbit calibration parameters, [x, y, z, k3] | | R/W | Stop PLC restart |

| Register address | Type | Definition | Value/unit | Read-write | Come into force Opportunity |
|---|---|---|---|---|---|
| SFD6050+2*i (i=0,1...5) | float | Dh parameters of the positioner, sequentially storing two sets of dh parameters, each containing 6 values [a, d, alpha, theta, k1, k2] | | R/W | Stop PLC restart |
| SFD6080+2*i (i=0,1...2*number of external shaft joints-1) | float | External shaft reduction ratio: Two external shaft systems are stored sequentially. The rotating joint is the reduction ratio, and the moving joint is the motor's movement distance per revolution. | | R/W | Stop PLC re-run |
| SFD6100+2*i (i=0,1...5) | float | Maximum speed limit for external axes. For 6-axis robots, the data is copied to sd5660 after power-on. | mm/s perhaps linear measure /s | R/W | Stop PLC re-run |
| SFD6120+2*i (i=0,1...5) | float | Maximum acceleration limit for external axes. For 6-axis robots, the data is copied to sd5680 after power-on. | mm/s perhaps linear measure /s | R/W | Stop PLC re-run |
| SFD6140+2*i (i=0,1...5) | float | Maximum acceleration limit for external axes (6 axes). The robot will copy data to sd5700 after power-on. | mm/s perhaps linear measure /s | R/W | Stop PLC restart |
| SFD6160+2*i (i=0,1...5) | float | Maximum limit of the external shaft joint angle | Degree or mm | R/W | Stop PLC re-run |
| SFD6180+2*i (i=0,1...5) | float | Minimum angular limit of external shaft joint | Degree or mm | R/W | Stop PLC restart |
| SFD6200+i (i=0,1) | INT16U | Two external shaft system types, corresponding to the sfd6200 and sfd6201 respectively.<br>0: The external axis configuration consists of a 2-axis positioner and a 1-axis ground rail.<br>1: The external shaft assembly is a three-axis gantry. | Default 0 | R/W | Stop PLC re-run |
| SFD6202+i (i=0,1) | INT16U | The robot's mounting configurations on the gantry correspond to models sfd6202 and sfd6203 respectively.<br>0: Inversion<br>1: Formal wear | Default 0 | R/W | Stop PLC re-run |
| SFD6210+2*i (i=0,1...11) | float | Longmen calibration parameters: [x1, y1, z1, k1. X2, y2, z2, k2. X3, y3, z3, k3] | | R/W | Stop PLC restart |
| SFD6400 | float | Inverse solution of six-axis robot with iterative position accuracy | Default: 0.005 | R/W | Stop PLC re-run |
| SFD6402 | float | Inverse iterative attitude accuracy of six-axis robot | Default: 0.01 | R/W | Stop PLC re-run |
| SFD6500 | INT16U | Robot main version number | firmware related | R | |
| SFD6501 | INT16U | Robot subversion number | firmware related | R | |
| SFD6502 | INT16U | Robot function version number | firmware related | R | |
| SFD6503 | INT16U | Temporary robot version number | firmware related | R | |
| SFD6600 | INT | Lm iteration count | Default 8 | R/W | Stop PLC re-run |

| Register address | Type | Definition | Value/unit | Read-write | Come into force Opportunity |
|---|---|---|---|---|---|
| SFD6601 | INT | Lm convergence error setting (sum of squared position errors) | Default 5 | R/W | Stop PLC re-run |
| SFD6602 | INT | Tcp error impact factor | Default: 10000 | R/W | Stop PLC restart |
| SFD6603 | INT | Zero error influence factor | Default 100 | R/W | Stop PLC restart |
| SFD6604 | INT | Rod length/bias error factor | Default 100 | R/W | Stop PLC restart |
| SFD6605 | INT | Alpha/beta error factor | Default 1 | R/W | Stop PLC restart |
| SFD6606 | INT | Reduction ratio error influence factor | Default 1 | R/W | Stop PLC re-run |
| SFD6607 | INT | Upper limit of rod length/offset deviation | Default 2 | R/W | Stop PLC re-run |
| SFD6610 | float | Set the 6-point error upper limit at 7 o'clock | Default value: 0.2mm | R/W | Stop PLC restart |
| SFD6800+2*i (i=0,1,2,3) | float | The initial value for the intersection line's main coordinate system calibration calculation. The teaching point must correspond one-to-one with the initial value. | The default is-1.57,0, 1.57,3.14 | R/W | The next delivery point will take effect |
| SFD6808+2*i (i=0,1...5) | float | The euler angle for transforming the intersection line's main coordinate system relative to the robot's base coordinate system, with position in mm and attitude in degrees | | R/W | The next delivery point will take effect |
| SFD6820+2*i (i=0,1...5) | float | The euler angle for transforming the main coordinate system of the intersection line relative to the 2-axis coordinate system of the positioner, with position units in mm and attitude units in degrees. | | R/W | The next delivery point will take effect |
| SFD7000 +20*i+2*j (i=0,1...maxaxes -1. j=0,1...9) | float | The j-th fourier trajectory parameter on the i-th axis, with 10 parameters per axis | | R/W | Stop PLC restart |
| SFD7120 +2*i | float | Rated torque of the i-th shaft | MN·m | R/W | Stop PLC re-run |
| SFD7140 +2*i | float | Dynamic threshold for axis i | MN·m | R/W | Stop PLC re-run |
| SFD7160 | INT16U | Feedback data filter coefficient. The higher the value, the better the filtering effect. | Default: 50ms | R/W | Stop PLC restart |
| SFD7170+2*i (i=0,1...maxAxis Num-1) | float | Maximum torque of the i-th axis | Default 0 | R/W | Stop PLC restart |
| SFD7200 +20*i+2*j (i=0,1...maxaxes -1. j=0,1...9) | float | The j-th dynamic parameter of the i-th axis, with 10 parameters per axis<br>Mdh modeling method, the representation of the connecting rod in the corresponding joint coordinate system, and the corresponding sequence of dynamic parameters<br>Lxx lxy lxz lyy lyz lzz mx my mz m | | R/W | Stop PLC restart |

| Register address | Type | Definition | Value/unit | Read-write | Come into force Opportunity |
|---|---|---|---|---|---|
| SFD7320 +20*i+2*j (i=0,1... maximum load count-1. j=0,1...9) | float | The j-th dynamic parameter of the i-th load<br>A: 0-3 0-1 indicates tool load<br>B=5 the moment of inertia under the load at the center of mass (mg·m²)<br>B=6,7,8: The load center of gravity is represented in the robot's end coordinate system (m)<br>B=9 load mass (mg)<br>B=0:4 not in use<br>A=2 is the load on the forearm at joint 3 | | R/W | Stop PLC restart |
| SFD7400 +10*a+2*b (i=0,1...maxaxes -1. j=0,1...4) | float | The j-th friction parameter of the i-th axis<br>Each axis can have up to 5 friction parameters. The current friction model has only 2 parameters.<br>F=fc*sign(dq)+fv*dq | | R/W | Stop PLC restart |
| SFD7500 | INT16U | Speed planning mode<br>0: Plan based on acceleration/deceleration values<br>1: Plan based on acceleration and deceleration time | | R/W | Stop PLC re-run |
| SFD7502 | float | Acceleration time | ms | R/W | Stop PLC re-run |
| SFD7504 | float | Deceleration time | ms | R/W | Stop PLC re-run |
| SFD8280 | float | Setting of the threshold for shoulder abnormalities | Default value: 0.2 | R/W | With immediate effect |
| SFD8282 | float | Setting the threshold for elbow anomaly detection | Default value: 0.2 | R/W | With immediate effect |
| SFD8284 | float | Setting of the threshold for wrist abnormalities | Default value: 0.2 | R/W | With immediate effect |
| SFD8286 | float | Nominal damping coefficient setting | Default value: 0.2 | R/W | With immediate effect |
| SFD8288 | float | Translation singular error compensation coefficient | Default value 300 | R/W | With immediate effect |
| SFD8290 | float | Rotation of the singular error compensation coefficient | Default value 100 | R/W | With immediate effect |
| SFD9100+40i (i=0,1,2,3) | INT16U | Belt type: 0: Straight, 1: Disc | | R/W | Reactivation process switch |
| SFD9101+40i (i=0,1,2,3) | INT16U | Coordinate system serial number of the workpiece bound by the belt | | R/W | Reactivation process switch |
| SFD9102+40i (i=0,1,2,3) | INT16U | Types of belt encoders<br>0: Servo motor encoder<br>1: External encoder (high-speed counter) | | R/W | Reactivation process switch |
| SFD9103+40i (i=0,1,2,3) | INT16U | Address of value storage register of belt encoder | | R/W | Reactivation process switch |

| Register address | Type | Definition | Value/unit | Read-write | Come into force Opportunity |
|---|---|---|---|---|---|
| SFD9104+40i (i=0,1,2,3) | float | Pulse equivalent of the belt encoder | | R/W | Reactivation process switch |
| SFD9106+40i (i=0,1,2,3) （obligate） | INT16U | The sensitivity coefficient of the encoder pulse fluctuation protection mechanism ranges from 0 to 9. A higher value indicates greater sensitivity. The default value is 9. | | R/W | Reactivation process switch |
| SFD9107+40i (i=0,1,2,3) | INT16U | Coordinate system type for visual points<br>0: Base coordinate system<br>1: Belt-bound workpiece coordinate system<br>2: Visual coordinate system | | R/W | Reactivation process switch |
| SFD9108+40i (i=0,1,2,3) | INT16U | Visual coordinate system index, values range from 0 to 4 | | R/W | Reactivation process switch |
| SFD9109+40i (i=0,1,2,3) （obligate） | INT16U | Queue capacity, maximum 200 | | R/W | Reactivation process switch |
| SFD9110+40i (i=0,1,2,3) | float | Distance for repeated removal, default value 10 | mm | R/W | Reactivation process switch |
| SFD9112+40i (i=0,1,2,3) | float | Upper bound of work | | R/W | Reactivation process switch |
| SFD9114+40i (i=0,1,2,3) | float | Lower bound of work | | R/W | Reactivation process switch |
| SFD9116+40i (i=0,1,2,3) （obligate） | float | Trigger upper bound | | R/W | Reactivation process switch |
| SFD9118+40i (i=0,1,2,3) | float | Trigger lower bound | | R/W | Reactivation process switch |
| SFD9120+40i (i=0,1,2,3) | float | Stop distance | mm | R/W | Reactivation process switch |
| SFD9122+40i+2j (i=0,1,2,3. j=0,1,2,...,5) | float | Pose compensation, with 6 floats in total. The euler angle for pose compensation is measured in degrees, and the default value is 0. | mm or degree | R/W | Reactivation process switch |
| SFD9134+40i (i=0,1,2,3) | float | Time bias | s | R/W | Reactivation process switch |
| SFD9136+40i (i=0,1,2,3) | INT16U | The filter window size for the belt position filter, with a maximum of 500 and a minimum of 5 | | R/W | Reactivation process switch |
| SFD9260+2i (i=0,1,2,...,9) | float | Workpiece height (relative to the belt surface, where the belt surface height is 0) corresponds to 10 workpiece types, totaling 10 floats | mm | R/W | Reactivation process switch |

SD register

<table>
<tr><th>Register address</th><th>Type</th><th colspan="2">Definition</th><th>Value/unit</th><th>Read-write</th><th>Trigger time</th></tr>
<tr><td>SD10</td><td>INT16U</td><td colspan="2">Current PLC scan cycle size</td><td>0.1ms</td><td>R</td><td></td></tr>
<tr><td>SD11</td><td>INT16U</td><td colspan="2">Minimum PLC scan cycle</td><td>0.1ms</td><td>R</td><td></td></tr>
<tr><td>SD12</td><td>INT16U</td><td colspan="2">Maximum PLC scan cycle</td><td>0.1ms</td><td>R</td><td></td></tr>
<tr><td>SD1800</td><td>INT16S</td><td colspan="2">Log status<br>1: Log off<br>2: Log enabled<br>Other: Abnormal</td><td></td><td>R</td><td></td></tr>
<tr><td>SD1801</td><td>INT16S</td><td colspan="2">Log enabled<br>Set to 1 to enable. The SD1800 will automatically clear 0 to indicate success.</td><td></td><td>R/W</td><td>With immediate effect</td></tr>
<tr><td>SD1802</td><td>INT16S</td><td colspan="2">Log off<br>Set to 1 to enable. The SD1800 will automatically clear 0 to indicate success.</td><td></td><td>R/W</td><td>With immediate effect</td></tr>
<tr><td>SD1804</td><td>INT16S</td><td colspan="2">Clear Flash log<br>Clear all log information from the current flash</td><td></td><td>R/W</td><td>With immediate effect</td></tr>
<tr><td>SD2000+60i</td><td>INT16U</td><td colspan="2">Servo status, where i is the axis number<br>0: Connection lost<br>1： Ready<br>2： ON</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2002+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT16U</td><td colspan="2">Axis error code, where i is the axis number. For error code interpretation, see Appendix "Motion Bus Error Codes".</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2004+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT16U</td><td colspan="2">Number of bus communication errors, where i is the axis number<br>Check the communication parameters of the servo driver P7 group.<br>Check if the servo and controller communication wiring is working properly</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2006+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT32S</td><td rowspan="8">State parameters of each axis, where i is the axis number</td><td>Current mobile data usage</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2008+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT32S</td><td>Current location</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2010+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT32S</td><td>Current speed</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2012+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT32S</td><td>Instantaneous given velocity</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2014+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT32S</td><td>Current full closed-loop position</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2016+60i<br>i (i=0,1,2... max axes-1)</td><td>INT16U</td><td>Present segment</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2030+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT32S</td><td>Given location</td><td></td><td>R</td><td></td></tr>
<tr><td>SD2032+60i<br>i (i=0,1,2... max axial number-1)</td><td>INT32S</td><td>Given velocity</td><td></td><td>R</td><td></td></tr>
</table>

| Register address | Type | Definition | | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|---|
| SD2034+60i i (i=0,1,2... max axial number-1) | INT32S | | Acceleration time | | R | |
| SD2036+60i i (i=0,1,2... max axial number-1) | INT32S | | Deceleration time | | R | |
| SD2038+60i i (i=0,1,2... max axial number-1) | float | | Binding coefficient | | R | |
| SD2040+60i i (i=0,1,2... max Axis number-1) | INT32S | | Point step size | | R | |
| SD2042+60i i (i=0,1,2... max axial number-1) | INT32S | | Point velocity | | R | |
| SD2044+60i i (i=0,1,2... max axial number-1) | INT32S | | Positioning completion width | | R | |
| SD2046+60i i (i=0,1,2... max axial number-1) | INT32S | | Sync width | | R | |
| SD4000+2i i(i=0,1...5) | float | Real-time pose feedback from the end-effector | | Degree or mm | R | |
| SD4012 | INT16S | Cf1 | | | R | |
| SD4013 | INT16S | Cf4 | | | R | |
| SD4014 | INT16S | Cf6 | | | R | |
| SD4015 | INT16U | Current tool ID | | | R/W | With immediate effect |
| SD4016 | INT16U | Current hand binding value | | | R | |
| SD4018 | INT16U | Current coordinate system | | | R/W | With immediate effect |
| SD4019 | INT16U | Current work piece number | | | R/W | With immediate effect |
| SD4020+2i i (i=0,1,2... max_cuts-1) | float | The current joint angle, where i is the axis number | | Linear measure | R | |
| SD4040+2i i(i=0,1,2..6) | float | The trajectory's end-point position and orientation, along with the tool's Cartesian coordinates [x, y, z, Rz, Ry, Rx], are specified. When SM3152 is triggered, the coordinates are formatted as [x, y, z, q0, q1, q2, q3]. | | | R | |
| SD4060+2i i(i=0,1,2..5) | float | The current end provides real-time position and orientation, and real-time interpolation of Cartesian coordinate values for the tool, [x, y, z, Rz, Ry, Rx] | | | R | |
| SD4080+2i i (i=0,1,2... max_cuts-1) | float | Interpolates joint values in real time given a joint angle | | Linear measure | R | |
| SD4100 | float | Cartesian point step size | | mm | R/W | Trigger next time |
| SD4102 | float | Percentage of Descartes' point velocity | | % | R/W | Trigger next time |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD4104 | float | Percentage of acceleration from the Cartesian point movement | % | R/W | Trigger next time |
| SD4110 | float | Joint dynamic step size | Linear measure | R/W | Trigger next time |
| SD4112 | float | Percentage of the dynamic speed of the joint | % | R/W | Trigger next time |
| SD4114 | float | Percentage of the dynamic acceleration of the joint | % | R/W | Trigger next time |
| SD4130 | INT16U | Error. For error code analysis, see the appendix "Robot Error Codes". | | R/W | With immediate effect |
| SD4131 | INT16U | Warning. For warning code analysis, see the appendix "Robot Warning Exception Codes". | | R/W | With immediate effect |
| SD4140+2i i(i=0,1,2,3,) | float | The terminal provides real-time differential speeds, including linear velocity, Rz angular velocity, Ry angular velocity, and Rx angular velocity. | mm/s perhaps linear measure /s | R | |
| SD4240+2i i(i=0,1,2,3,) | float | The terminal provides real-time differential acceleration data, including linear acceleration, Rz angular acceleration, Ry angular acceleration, and Rx angular velocity. | mm/s perhaps linear measure /s | R | |
| SD4380+2i i (i=0,1,2... max_cuts-1) | float | real time joint given differential velocity | linear measure /s | R | |
| SD4480+2i i (i=0,1,2... max_cuts-1) | float | real time joint given differential acceleration | linear measure /s | R | |
| SD4171 | INT16U | maximum deceleration time of maximum period | Default: 1000ms | R/W | Next trigger |
| SD4700+2i (i=0,1,2... maximum number of axes minus 1) | float | The maximum speed of each joint is transmitted by the SFD5040 and can be modified before the user executes the command. | linear measure /s | R/W | Next trigger |
| SD4720+2i (i=0,1,2... maximum number of axes minus 1) | float | The maximum acceleration of each joint is transmitted by the SFD5060 and can be modified before the user executes the command. | linear measure /$s^2$ | R/W | Next trigger |
| SD4750 | float | The maximum speed at the end is transmitted by the SFD5080 and can be modified before the user executes the command. | mm/s | R/W | Next trigger |
| SD4770 | float | The maximum terminal acceleration is transmitted by the SFD5090 and can be modified before the user executes the command. | mm/$s^2$ | R/W | Next trigger |
| SD4792 | float | Acceleration percentage | % | R/W | Next trigger |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD4794 | INT16U | Current interpolation real-time trajectory type<br>0: Cartesian single-step motion<br>1: Single step joint<br>2: Cartesian continuous point motion<br>3: Joint continuous point movement<br>4: Slow down and stop moving<br>5: Stop state<br>6: Cartesian point motion state<br>7: Dynamic state of the joint<br>12: RBPTP/RBLINE/RBCIRCLE instruction<br>13: RCPATH instruction<br>14: External shaft single-step motion<br>15: External shaft continuous stop-and-go | | R | |
| SD4795 | INT16U | Selection of six-axis robot model<br>0: Xinje ontology model<br>1: For Panasonic's main model<br>Selects the iteration count | Default 0 | R/W | Restart the PLC to take effect |
| SD4800+2i (i=0,1,2...5) | float | Robot base coordinate system, real-time feedback pose of the current tool | | R | |
| SD4820+2i (i=0,1,2...5) | float | Robot base coordinate system, real-time tool position and orientation | | R | |
| SD4861 | INT16U | Maximum slope of the arc | Default: 10 degrees | R/W | RC will take effect next time |
| SD4862 | float | Minimum swing arc length<br>If the swing length at the point is less than this value, it will be adjusted to this value. The last swing does not swing when the swing length is less than this value | Default 1mm | R/W | RC will take effect next time |
| SD4864 | float | Maximum swing arc length<br>If the swing length exceeds this value, it will be adjusted to this value. | Default: 10mm | R/W | RC will take effect next time |
| SD4866 | float | Minimum arc amplitude | Default 1mm | R/W | RC will take effect next time |
| SD4868 | float | Maximum arc amplitude | Default: 20mm | R/W | RC will take effect next time |
| SD4870 | float | Minimum arc height | Default 1mm | R/W | RC will take effect next time |
| SD4872 | float | Maximum arc height | Default: 10mm | R/W | RC will take effect next time |
| SD4874 | float | The minimum length of the last swing. If the remaining length is less than this value, no arc will be swung. | Default 1mm | R/W | RC will take effect next time |
| SD5000+2i (i=0,1,2... maximum number of axes minus 1) | float | The maximum acceleration of each joint is transmitted by the SFD5250, and users can modify it before running the command. | $m/s^3$ | R/W | RC will take effect next time |
| SD5020+2i (i=0,1,2,3) | float | The maximum acceleration at the end is transmitted by the SFD5270 and modified before the user executes the command. | $mm/s^3$ | R/W | Next trigger |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5080+2i (i=0,1,2... maximum number of axes minus 1) | float | Maximum speed of each joint in Cartesian motion<br>The SFD5230 passes this data during PLC initialization, which users can modify before executing commands. | linear measure /s | R/W | Next trigger |
| SD5140+2i (i=0,1,2...6) | float | The robot tool coordinate system displays the current real-time feedback coordinates [x, y, z, Rz, Ry, Rx]. When SM3152 is triggered, the format changes to [x, y, z, q0, q1, q2, q3]. | mm | R | |
| SD5200+2i (i=0,1,2...6) | float | The robot's base coordinate system displays its real-time position coordinates [x, y, z, Rz, Ry, Rx]. When SM3152 is triggered, the coordinates change to [x, y, z, q0, q1, q2, q3]. | mm | R | |
| SD5220+2i (i=0,1,2...6) | float | The robot's base coordinate system displays the current real-time feedback tool coordinates [x, y, z, Rz, Ry, Rx]. When SM3152 is triggered, the coordinates change to [x, y, z, q0, q1, q2, q3]. | mm | R | |
| SD5240+2i i (i=0,1,2... max_cuts-1) | float | Robot joint coordinate system, real-time joint speed feedback | linear measure /s | R | |
| SD5260+2i (i=0,1,2,3) | float | The robot's base coordinate system provides real-time feedback on the end-effector speed, including linear velocity, Rz angular velocity, Ry angular velocity, and Rx angular velocity. | mm/s perhaps linear measure /s | R | |
| SD5280+2i (i=0,1,2) | float | The terminal pose display at the application layer, similar to the SD4000 at the system level, is controlled by the SD5298 coordinate system and is used solely for monitoring purposes. | | R | |
| SD5312 | float | Swing X-axis displacement to monitor whether the swing direction interpolation is correct | mm | R | |
| SD5314 | float | Swing Z-axis displacement to monitor whether the swing direction interpolation is normal | mm | R | |
| SD5316 | float | Z-axis oscillation velocity | mm/s | R | |
| SD5318 | float | Swing Y-axis displacement to monitor whether the swing direction interpolation is correct | mm | R | |
| SD5320 | float | Y-axis oscillation velocity | mm/s | R | |
| SD5322 | float | The maximum angle between the two interpolation points is set. If the angle exceeds this value, an alarm (10095) will be triggered. | 0-10 degrees Default: 5 degrees | R/W | RC will take effect next time |
| SD5324 | float | Adjust the swing plane angle of the two interpolation points. If the angle exceeds this set value, the swing plane of the next interpolation point will rotate 180 degrees around the X-axis. | 170-180 degrees, default 175 degrees | R/W | RC will take effect next time |
| SD5340+2i i (i=0,1,2... max_cuts-1) | float | Joint coordinate system of the robot, real-time feedback of joint acceleration | linear measure /s | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
| --- | --- | --- | --- | --- | --- |
| SD5360+2i (i=0,1,2,3) | float | The robot's base coordinate system provides real-time feedback on the end-effector acceleration, including linear acceleration, Rz angular acceleration, Ry angular acceleration, and Rx angular acceleration. | mm/s perhaps linear measure /s | R | |
| SD5400 | INT16U | Number of points issued each time SD5400 <= SD5403. If SD5400> SD5403, the out-of-range data points cannot be read and will be automatically discarded. | | R/W | The next delivery point will take effect |
| SD5401 | INT16U | The number of actual points issued each time<br>The actual number of delivery points after removing queue overflow points | | R | |
| SD5402 | INT16U | maximum value of instruction point buffer space<br>Maximum number of points a user can send at once | Default 200 | R | |
| SD5403 | INT16U | Available space size, number of space pointers<br>The maximum number of points a user can send at a time, SD5403 <=SD5402 | | R | |
| SD5404 | INT16U | Current data reading processing point number<br>Monotonic increasing, user D84 settings | | R | |
| SD5432 | INT16U | Type of input signal for external signal function<br>0: M signal<br>1: X signal | | R/W | Enable RC motion next time |
| SD5433 | INT16U | Type of input signal for external signal function<br>0: ON signal trigger<br>1: OFF signal trigger | | R/W | Next time RC movement is triggered |
| SD5440 | INT16U | Does it support SENSE?<br>0: Close<br>1: Turn on | Default 0 | R/W | SENSE will take effect next time |
| SD5441 | INT16U | Does it support TILL?<br>0: Close<br>1: Turn on | Default 0 | R/W | Next time TILL is triggered |
| SD5442 | INT16U | Supports FIND function<br>0: Close<br>1: Turn on | Default 0 | R/W | Next time FIND is triggered, it will take effect |
| SD5443 | INT16U | The type of register used to store input coordinates for solving equations | 0： D 1： HD | R/W | SM3013 will take effect next time |
| SD5444 | INT32S | Enter the first address of the storage register for the input point coordinates. When a negative value is entered, it is treated as an absolute value. | | R/W | SM3013 will take effect next time |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5447 | INT16U | The register type for storing the coordinates of the accessibility check input points | 0：D<br>1：HD | R/W | Next time SM3015 is triggered, it will take effect |
| SD5448 | INT32S | The first address of the storage register for the input point coordinates is checked for accessibility. If a negative value is entered, it is treated as an absolute value. | | R/W | Next time SM3015 is triggered, it will take effect |
| SD5450 | INT32U | The RC command specifies the total number of points, which is displayed only for reference and cannot be modified by the user. After the RC command is triggered, it sends the accumulated value to the SD5401 each time.<br>Change from the old SD5502 specification to dual characters | | R | |
| SD5452 | INT32U | Data read processing line number<br>Change from the old SD5560 specification to dual characters | | R | |
| SD5454 | INT32U | Filter data row numbers<br>Switch from the old SD5561 specification to dual-channel | | R | |
| SD5456 | INT32U | Forward processing line numbers<br>Switch from the old SD5562 specification to dual-channel | | R | |
| SD5458 | INT32U | Number of forward-looking groups<br>Change from the old SD5565 specification to dual characters | | R | |
| SD5460 | INT32U | Track processing line number<br>Change from old SD5563 to dual characters | | R | |
| SD5462 | INT32U | Real-time interpolation of track line numbers<br>Switch from the old SD5564 specification to dual-channel | | R | |
| SD5464 | INT32U | Displays the current interpolation real small segment number (considering filtered small segments) and indicates the movement to the nth machining point. When the SD5573 arc instruction is not enabled, only the arc endpoint number is displayed. When SD5573 is enabled, the auxiliary point number is displayed.<br>Change from old SD5566 to dual character | | R | |
| SD5466 | INT16U | Number of machining points filtered out due to short paths and poses in data filtering<br>Change from the old SD5569 specification to dual characters | | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5468 | INT16U | Real-time interpolation skips filtered point counts, logging the execution status of filtered points during interpolation. Each filtered point increments the register by 1. The trajectory completes execution when the number of transition segments equals the count.<br>Switch from the old SD5578 specification to dual-character format | | R | |
| SD5470 | INT16U | Current external axis interpolation segment number | | R | |
| SD5480 | INT32S | Convert coordinates to the first address of the input point storage register. If the input value is negative, it is treated as an absolute value. | | R/W | SM3019 will take effect next time |
| SD5482 | INT16U | Coordinate conversion input point storage register type | 0：D<br>1：HD | R/W | SM3019 will take effect next time |
| SD5500 | INT16U | Storage address type for processed data<br>0:D<br>1:HD<br>If SD5500 equals 1 and SD5588 equals 1000, HD1000 is designated as the starting address for storing dynamic point data. | | R/W | The next delivery point will take effect |
| SD5503 | INT16U | RC instruction processing point tool number<br>0~19: User-defined tool number during operation. Changing tools during motion is not supported. This register differs from the machining point coordinate system type. | | R/W | RC will take effect next time |
| SD5510 | float | Path arc transition angle limit<br>Angle between line segments: The complement of the angle between two adjacent vectors. If the angle is greater than this value, use a straight transition. if less, use a rounded transition. | Default: 175 degrees | R/W | RC will take effect next time |
| SD5512 | float | transition error<br>This value will somewhat limit the maximum allowable speed of the arc transition section and the previous section, to be used in conjunction with point D26. setting it too large will cause the trajectory to deviate from the inflection point, increasing the distance and the trajectory error. setting it too small will result in a smaller radius of the transition arc trajectory, leading to a correspondingly larger curvature, which in turn restricts the inflection point speed too much. when setting a larger transition error, due to the limitation of the trajectory length, the actual execution of the transition error may be less than the set value. setting it to 0 will force the underlying layer to handle it as 100.0mm. | Default 100.0mm | R/W | RC will take effect next time |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5514 | float | arc height error<br>A smaller setting imposes a stricter speed limit on the arc command, while a larger setting relaxes the speed limit but increases the fitting error of the arc trajectory by approximating curves with straight lines. | Default: 0.2mm | | RC will take effect next time |
| SD5518 | float | direct transition error<br>For path hard transition speed limit calculation, a straight transition is used. Setting it too large increases the distance from the inflection point, while setting it too small restricts the inflection point speed. | Default: 0.2mm | R/W | RC will take effect next time |
| SD5520 | float | Minimum angle constraint for small line segments<br>Determines whether the point speed needs to be reduced to 0. If the angle between adjacent line segments is less than this value, no transition processing is performed, as an excessively small angle would result in a highly curved transition trajectory, severely limiting the speed. The system does not determine the minimum angle between a straight line and an arc. | Default: 5 degrees | R/W | RC will take effect next time |
| SD5522 | float | minimum path length limit<br>Filtering out machining points with robot or external axis paths that are too short, and working with SD5570 to exclude cases where two adjacent machining points are too close. | Default: 0.01mm or degree | R/W | Next trigger |
| | | The threshold used to determine if the given or taught point coincides with the intersection line trajectory. | Default: 0.01 | R/W | The next delivery point will take effect |
| SD5524 | float | Adjusting the Approximation Degree of Circular Arcs with Three Postures Interpolation<br>Transition Coefficient in Attitude Interpolation | Default 0.2 | R/W | RC will take effect next time |
| SD5526 | float | Restrictions on the attitude changes of the starting and ending points of a line segment<br>Used to detect excessive changes in the relative path of attitude between adjacent segments, assessing the transformation of the relative path of attitude between adjacent trajectory segments. | Default: 100 degrees | R/W | RC will take effect next time |
| SD5528 | INT16U | Maximum search segment length for merged segments<br>Maximum number of segments allowed for multi-segment merging and maximum number of trajectory segments processed for merging | Default 10 | R/W | RC will take effect next time |
| SD5529 | INT16U | How many paragraphs before interpolation can be applied | Default 20 | R/W | RC will take effect next time |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5532 | float | Transition Error Limit of Trajectory Fusing Path | Default: 0.1mm | R/W | RC will take effect next time |
| SD5534 | float | Fusion trajectory transition angle limit | Default value: 179.908752 degrees | R/W | RC will take effect next time |
| SD5536 | INT16U | Forward window size<br>Number of trajectory segments processed in advance per PLC scan cycle | Default 10 | R/W | RC will take effect next time |
| SD5537 | INT16U | Forward window step size<br>The Difference of the Forward Processing Track Head Pointer Between Two Adjacent PLC Scan Cycles | Default 5 | R/W | RC will take effect next time |
| SD5538 | INT16U | Enable initial merge<br>0: Close<br>1: Turn on | Default 0 | R/W | RC will take effect next time |
| SD5539 | INT16U | Enable termination merge<br>0: Close<br>1: Turn on | Default 0 | R/W | RC will take effect next time |
| SD5540 | float | Dynamic adjustment, real-time speed ratio, minimum 0.01<br>In the joint constrained Cartesian programming, the system automatically adjusts the value during interpolation to prevent joint overspeed. | Default 100% | R/W | with immediate effect |
| SD5542 | float | maximum radius of transition of trajectory attitude | Default: 100mm | R/W | The next delivery point will take effect |
| SD5544 | float | centripetal acceleration<br>This value is determined by the actual machine rigidity and servo performance. You can reduce this value if there is an impact at the inflection point. | Default 4000 mm/s | R/W | RC will take effect next time |
| SD5546 | float | Interpolation time | Default: 0.1s | R/W | RC will take effect next time |
| SD5548+2i<br>i (i=0,1,2...<br>max_cuts-1) | float | The given joint angle at the end of the pause | linear measure | R | |
| SD5567 | INT16U | Display the current interpolation trajectory type<br>1: Straight line<br>2: Circular Arc<br>3：PTP<br>4: Incomplete intersection line<br>5: Spiral<br>6: Complete intersection line<br>20: Single attitude trajectory<br>21: Robot follows external axis<br>22: B-spline trajectory | | R | |
| SD5568 | INT16U | RC command execution hand system | | R | |
| SD5570 | float | Non-Forward-looking Attitude Trajectory Length Judgment Limit | Default: 0.0001rad | R/W | RC will take effect next time |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5572 | INT16U | Number of data points read each time | Default 25 | R/W | The next delivery point will take effect |
| SD5573 | INT16U | Consider arc auxiliary point attitude<br>0: Not considered<br>1: Consider | Default 0 | R/W | The next delivery point will take effect |
| SD5574 | INT16U | Calculate the welding torch posture based on the welding torch angle<br>0: Teach the welding torch position<br>1: Welding torch position calculated based on angle | Default 0 | R/W | The next delivery point will take effect |
| SD5575 | INT16U | Dynamic speed control completion time, in milliseconds<br>Change the running time of the shifting process after SD5540<br>Reserve Time Before and After the Speed Limitation Region of the Cartesian Planning | Default: 500ms | R/W | The next delivery point will take effect |
| SD5576 | float | The transition radius of the trajectory fusion path is set to the percentage of the smaller value between the lengths of the adjacent two trajectory segments.<br>Use with SD5532 | Default: 20% | R/W | The next delivery point will take effect |
| SD5578 | INT16U | Starting point speed variation time | Default 50 | R/W | The next delivery point will take effect |
| SD5579 | INT16U | The tool attitude in the mode 1 of the outer shaft motion remains unchanged in the world coordinate system<br>0: Do not enable<br>1: Turn on | | R/W | The next delivery point will take effect |
| SD5580+2i (i=0,1,2,3) | float | Real-time interpolation of differential end velocities (linear velocity, Rz angular velocity, Ry angular velocity, and Rx angular velocity) follows the same velocity pattern as SD4140, with the curve leading SD4140 by approximately 20 interpolation cycles. | mm/s perhaps linear measure /s | R | |
| SD5588 | INT32S | Process the storage address. If a negative value is entered, it will be treated as an absolute value. | | R/W | Next trigger |
| SD5590 | float | Maximum transition reference value for external shafts | 0-10 degrees or 0-10mm | R/W | Next trigger |
| SD5593 | INT16U | The current external axis system number can be used to specify the workpiece number for external axis calibration. | 1-2 | R/W | Enable external axis next time |
| SD5594 | INT16U | Current external axis motion type ① | 0-5 | R/W | Enable external axis next time |
| SD5595 | INT16U | Maximum variable cycle stop time for continuous point movement | Default: 500ms | R/W | Trigger next time |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5596 | INT16U | The percentage of acceleration and deceleration time for the external axis, used for smaller acceleration and deceleration values, in conjunction with the external axis CP parameter. | 0-100% | R/W | Enable external axis next time |
| SD5597 | INT16U | external axis forward processing control word<br>0: Conventional CP transition processing<br>1: Forward processing<br>When processing external axes, the CP parameter is invalid. Setting the external axis speed appropriately matches the robot's collaborative motion processing speed. Setting the external axis speed too low may cause overspeed or excessive acceleration after synchronization. Setting the external axis speed too high may result in prolonged zero-speed movement or extended deceleration time after synchronization. | | R/W | Enable external axis next time |
| SD5599 | INT16U | Attitude velocity percentage. D20 only indicates path velocity percentage or single attitude velocity percentage. | Default: 100% | R/W | The next delivery point will take effect |
| SD5600+2*i (i=0,1...number of external shaft joints minus 1) | float | External axis joint angle given value | Degree or mm | R | |
| SD5620+2*i (i=0,1...number of external shaft joints minus 1) | float | External shaft joint angle feedback value | Degree or mm | R | |
| SD5640+2*i (i=0,1...number of external shaft joints minus 1) | float | Real-time speed setting for the outer shaft motion | Degree or mm | R | |
| SD5660+2*i (i=0,1...number of external shaft joints minus 1) | float | The maximum joint speed limit for the outer shaft is derived from the SFD6100 mapping. | Degree or mm | R/W | Next trigger |
| SD5680+2*i (i=0,1...number of external shaft | float | The maximum joint acceleration limit for the outer shaft is mapped from the SFD6120 | $m/s^2$ or $mm/s^2$ | R/W | Next trigger |
| SD5700+2*i (i=0,1...number of external shaft joints minus 1) | float | The maximum joint acceleration limit for the outer shaft is mapped from the SFD6140. | $m/s^3$ or $mm/s^3$ | R/W | Next trigger |
| SD5720+ 12*i+2*j (i=0,1,2,3 j=0,1...5) | float | Ground track and gantry calibration data: The ground track has two reference points, and the gantry has four reference points. Each reference point records the robot's end position coordinates [x, y, z] and three external axis angles, where i is the reference point and j is the point data. | Degree or mm | R/W | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD5800+2*i (i=0,1...11) | float | Copy the current gantry calibration parameters from the external axis system serial number to SFD6210 for actual motion participation of gantry parameters [x1, y1, z1, k1. x2, y2, z2, k2. x3, y3, z3, k3] | | R/W | |
| SD5830+2*i (i=0,1,2,3) | float | The tool end display performs real-time interpolation in the reference coordinate system without including the end speed of the swing arc trajectory, which includes linear speed, Rz, Ry, and Rx angular speed. | mm/s perhaps linear measure /s | R | |
| SD5840+2*i (i=0,1...2*number of external shaft joints-1) | float | Record the angle of the external shaft during the pause | Degree or mm | R | |
| SD5870+2*i (i=0,1,2,3) | float | Record the current ground orbit calibration parameters [x, y, z, k3]. After successful calibration, save them to SFD6040. | | R | |
| SD5880+2*i (i=0,1...5) | float | Record the six DH parameters [a, d, alpha, theta, k1, k2] for the current positioner calibration. Upon successful calibration, save the data to SFD6050+12* (SD5593-1). | Degree or mm | R | |
| SD5900+2*i (i=0,1...5) | float | The Pose of the Base Coordinate of the Calibration Transducer in the World Coordinate System | | R | |
| SD5980+2*i (i=0,1...6) | float | The end-effector coordinates [x, y, z, q0, q1, q2, q3] of the current positioner system in its base coordinate system | | R | |
| SD6000+2i (i=0,1,2,3) | float | The unit quaternion attitude value [q0, q1, q2, q3] corresponding to the Euler angles of the SD4000's feedback pose coordinates | | R | |
| SD6010+2i (i=0,1,2,3) | float | The unit quaternion attitude value [q0, q1, q2, q3] corresponding to the Euler angles of the SD4060's position and orientation coordinates | | R | |
| SD6020 | INT16S | axis overlimit display register<br>For example, +3 indicates the maximum soft limit of axis 3 (robot joint 3), while-9 indicates the minimum soft limit of axis 9 (axis 2 of the external axis system 1).<br>axis superspeed display register<br>Before monitoring this register, check if the alarm code indicates over-limit or overspeed. Both conditions share this register. | | R | |
| SD6040+ 12*i+2*j (i=0,1...11. j=0,1,2...5 Method 2: i=0, j=0,1,2) | float | In the first method of positioner calibration, each calibration point records the end position coordinates [x, y, z] and the angles of three external axes, with 12 calibration points in total across two joints. | | R/W | |
| | | Positioner Calibration (Method 2): Before calibration begins, record the first point P1 in the robot tool coordinate system as the initial position reference for the positioner. | mm | R/W | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6046+2*i (i=0,1,2) | float | Calibration of the positioner (Method 2): Before calibration begins, record the second point P2 in the robot tool coordinate system. | mm | R/W | |
| SD6052+2*i (i=0,1,2) | float | Positioner calibration (Method 2): Before calibration begins, record the third point P3 in $\overrightarrow{P2P3}$ $\overrightarrow{P2P3}$ the robot tool coordinate system. The vector is the X-axis direction vector of the positioner. | mm | R/W | |
| SD6064 | INT16U | Positioning machine calibration (Method 2): Before starting the calibration, the user must input the estimated position values of the constraint points on the positioning machine in the 2-axis coordinate system. | mm | R/W | |
| SD6058+2*i (i=0,1,2) | float | Display of iteration times for positioner calibration (Method 2) | | R | |
| SD6066 | float | CALCULATION OF alpha_0 DISPLAY IN THE SECOND METHOD OF CALIBRATION OF THE VARIATOR | linear measure | R | |
| SD6070+2*j+10*i (i=0,1...5. j=0,1...4) | float | Calibration of Positioning Machine (Method 2) Error Model Calibration of Positioning Machine Pulse Recording of Robot Joint | | R/W | |
| SD6130+2*j+10*i (i=0,1. j=0,1...4) | float | Calibration of Positioning Machine (Method 2) Error Model Calibration of Positioning Machine Joint Pulse Recording | | R/W | |
| SD6150+2*i (i=0,1...4) | float | Calibration of Positioning Machine (Method 2) Error Model Calibration of Positioning Machine, Ground Track Feedback Angle Record | | R/W | |
| SD6190+2*i (i=0,1,2) | float | Positioner calibration (Method 1): The maximum, minimum, and average errors of the positioner are measured with three floating-point values. The error is defined as the difference between the distance from each point to the center of the circle and the fitted radius. | mm | R | |
| | | Positioner Calibration (Method 2): The system displays three error metrics-MaxError, MinError, and MeanError-which represent simulation-derived positional deviations. | mm | R | |
| SD6200+2*i (i=0,1...6) | float | The robot tool's real-time interpolation pose and position display in the RC command mode, relative to the reference coordinate system, is shown with 7 floating-point values: [x, y, z, q0, q1, q2, q3]. The trajectory does not include superimposed swing arcs. In external axis operation mode 1, the relative position is displayed. | | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6220+2*i (i=0,1,2...6) | float | When the RC command performs arc swinging, the robot tool displays its real-time interpolation pose relative to the reference coordinate system [x, y, z, q0, q1, q2, q3], superimposes the arc trajectory, and operates in external axis mode 1 for relative position. | | R | |
| SD6260 | float | Minimum length limit for straight or circular arc trajectory without monotonic speed planning<br>When the straight-line length after transition exceeds this value, the segment can execute acceleration-deceleration planning. Regular trajectories can only accelerate or decelerate monotonically to avoid frequent acceleration-deceleration in short segments. | Default: 10mm | R/W | Next trigger |
| SD6240+2*i (i=0,1,2,3) | float | Real-time interpolation speed of robot tool relative to reference coordinate system and superposition of swing arc | mm/s | R | |
| SD6262 | INT16U | Enable non-forward-looking attitude function in external axis coordination mode<br>0: Plan in the workpiece coordinate system<br>1: You can select the pose for planning in the base coordinate system | Default 0 | R/W | Enable external axis next time |
| SD6280 | INT16U | Region setting parameter storage register type<br>0: HD register<br>1: HSD register | Default 0 | R/W | RC will take effect next time |
| SD6281 | INT16U | Display the current alarm active area or prohibited area number | 0-15 | R | |
| SD6282 | INT32S | Offset address of the storage register for the region setting parameter<br>For example, if SD6280=0 and SD6282=200, the HD200-HD519 module records parameter settings for 16 zones. Negative values are processed as absolute values. | | R/W | RC will take effect next time |
| SD6285 | INT16U | ProcessOut parameter for setting the type of storage register<br>0: HD register<br>1: HSD register | Default value 0 | R/W | RC will take effect next time |
| SD6286 | INT32S | Offset address of the storage register for the region setting parameter<br>For example, if SD6285 equals 0 and SD6286 equals 200, the HD200-HD999 module records 800 processOut parameter block settings. Negative values are processed as absolute values. | | R/W | Active next time |
| SD6288 | float | Wait deceleration start percentage | | R/W | Active next time |
| SD6900 | INT16U | The FIND function records the offset address type of the data point: 0 for D register, 1 for HD register. | | R/W | RC will take effect next time |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6902 | INT32S | The FIND function records the offset of the location information address. When a negative value is entered, it is treated as an absolute value. | | R/W | RC will take effect next time |
| SD6930+2i (i=0,1,2,...,5) | float | Eulerian Angle Coordinates of Workpiece When It Leaves the Queue | mm or degree | R | |
| SD6960+2(i) (i=0,1,2...5) | float | Offset compensation for machining point coordinates<br>Compensates for offset values of the [x, y, z, Rx, Ry, Rz] coordinates of the machining point. The default attitude compensation is in the form of Euler angles. | Degree or mm | R/W | The RC instruction takes effect after execution |
| SD6972 | INT16U | Offset coordinate system type | 5: Joint Coordinate System<br>Other values: Base coordinate system | R/W | The RC instruction takes effect after execution |
| SD6974 | INT16U | Dynamic offset compensation function switch | Only 1: Effective<br>None of the other values are valid | R/W | The RC instruction takes effect after execution |
| SD6980 | INT16U | Timer on/off storage address type | 0：M | R/W | With immediate effect |
| SD6982 | INT32S | The timer turns on and off the storage address. When a negative value is entered, it is treated as an absolute value. | | R/W | With immediate effect |
| SD6984 | INT16U | Timer function switch: 0: off, 1: on | Default 0 | R/W | With immediate effect |
| SD6986 | INT16U | Timer time storage address type | 0：D<br>1：HD | R/W | With immediate effect |
| SD6988 | INT32S | Timer time storage address. If a negative value is entered, it is treated as an absolute value. | | R/W | With immediate effect |
| SD6990 | INT16U | Configure the robot pose storage register type: 0: D, 1: HD | | R/W | With immediate effect |
| SD6991 | INT16U | Configure the calibration result storage register type: 0: D, 1: HD | | R/W | With immediate effect |
| SD6992 | INT32S | Configure the first address of the robot pose storage register and the type of the pose storage register. When a negative value is input, it is processed as an absolute value. | | R/W | With immediate effect |
| SD6994 | INT32S | Configure the first address of the calibration result storage register and the type of the calibration result storage register. When a negative value is entered, it is treated as an absolute value. | | R/W | With immediate effect |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD6996 | INT16U | TCP attitude calibration flag: 0: Write attitude to default value. 1: Calibrate only X-axis. 2: Calibrate only Z-axis. 3: Calibrate both Z/X axes |  | R/W | With immediate effect |
| SD7000 | INT16U | The number of Fourier trajectory runs can be modified during execution. If it is less than the current count, the segment will stop after completion. | Default 1 | R/W | With immediate effect |
| SD7001 | INT16U | The Fourier trajectory speed ratio. A value of 100 takes 10 seconds to run one Fourier trajectory. Changes made during the run are invalid. | 1-100<br>Default 50 | R/W | Active next time |
| SD7002 | INT16U | monitoring data storage register type<br>0: D register<br>1: HD register | Default 0 | R/W | Active next time |
| SD7003 | INT16U | The time constant of the excitation trajectory and the period of the full-speed operation of the excitation trajectory | Default 10 | R/W | Active next time |
| SD7004 | INT32S | Monitor the starting address of the data storage register and the dynamic-related operational parameters. A continuous sequence of 20* axis values is required. When negative values are input, they are processed as absolute values.<br>SD7004=0 indicates no monitoring |  | R/W | Active next time |
| SD7006 | INT32S | The collision detection sensitivity HD register's starting address is processed as an absolute value when negative input is provided.<br>The sensitivity address of each joint is HD[SD7006+n]<br>The collision sensitivity of each joint is stored in a continuous HD register, of type INT16U, with a range of 1-400.<br>for instance：<br>SD7006=1000, HD[1000]=100, HD[1001]=200, indicating joint 1's collision sensitivity is 100 and joint 2's sensitivity is 200. Higher sensitivity values indicate lower sensitivity.<br>SD7006=0 indicates no detection |  | R/W | Active next time |
| SD7008 | INT32S | The collision detection enables the starting address of the HM register. When a negative value is input, it is processed as an absolute value.<br>HM[SD7008] General Collision Detection Switch<br>n Collision Detection Switch of HM[SD7008 +1 +n] Joint |  | R/W | Active next time |
| SD7010 | INT32S | The starting address of the M flag for axis collision. A negative value is treated as an absolute value.<br>M[SD7010] Total collision flag bit<br>The [SD7010 +1 +n] joint n has collided |  | R/W | Active next time |
| SD7012 | INT16U | Number of Fourier trajectories currently running |  | R |  |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD7013 | INT16U | bus configuration register type<br>0: D register<br>1: HD register | Default 0 Match the bus mapping address | R/W | Active next time |
| SD7014 | INT32S | Set the bus configuration register starting address. If a negative value is entered, it will be treated as an absolute value. | | | |
| SD7016 | INT16U | bus configuration register axis offset | | | |
| SD7017 | INT16U | bus configuration register feedback position offset | | | |
| SD7018 | INT16U | bus configuration register feedback speed offset | | | |
| SD7019 | INT16U | bus configuration register feedback acceleration offset | | | |
| SD7020 | INT16U | bus configuration register feedback torque offset | | | |
| SD7030+2*i (i=0,1...maxAxisnum-1) | float | The dynamic threshold for the i-th axis is read from the SFD7140 during PLC operation and updated in real time. | | R/W | With immediate effect |
| SD7050+i (i=0,1...maxAxisnum-1) | INT16U | Maximum consecutive collisions on the i-th axis | Default 15 | R/W | Active next time |
| SD7060+i (i=0,1...maxAxisnum-1) | INT16U | The percentage deviation of the i-th axis's torque from the rated torque during motion. If this value exceeds the user-defined collision detection sensitivity for each axis, a collision signal will be generated.<br>When performing repetitive motions, you can set the sensitivity of axis collision detection for the first few rounds of motion. | | R/W | Active next time |
| SD7074 | INT16U | Collision Handling Mode:<br>0: No internal processing, only set the M register<br>1: Set M register, pause<br>2: Set M register, normal stop retraction | | R/W | Active next time |
| SD7075 | INT16U | Number of stop retracement cycles: The interval between the position where the reverse collision stops retracement and the current position in the queue. | Default 0 Maximum 99 | R/W | Active next time |
| SD7077 | INT16U | Collision type detected<br>0: No collision<br>1: Forward collision<br>2: Reverse collision | | R | |
| SD7078 | INT16U | Current motion status with collision detection enabled<br>0: Still<br>Normal interpolation<br>2: Slow Stop<br>3: Queue stop<br>4: Pullback pause | | R | |
| SD7100+10i (i=0,1,2,3) | INT16U | Number of workpieces to photograph, minimum 1, maximum 10 | | R/W | With immediate effect |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD7101+10i (i=0,1,2,3) | INT16U | The D register's first address (occupying 20 words with unused reserved space) stores workpiece information, including belt number (INT16U, D0), workpiece type (INT16U, D1), and Eulerian angle coordinates (6 floats, D2~D12). | | R/W | With immediate effect |
| SD7150 | INT16U | Query the belt number for the workpiece. The maximum value is 3, and the minimum value is 0. | | R/W | With immediate effect |
| SD7151 | INT16U | The type of workpiece you are querying, with a maximum of 9 and a minimum of 0 | | R/W | With immediate effect |
| SD7152 （obligate） | INT16U | Query the starting address of the D register storing workpiece information (occupying 20 words, with unused reserved space). The workpiece information includes belt number (INT16U, D0), workpiece type (INT16U, D1), and Euler angle coordinates (6 floats, D2~D12). | | R/W | With immediate effect |
| SD7153 （obligate） | INT16U | Track following processing method<br>0: Smoothing (sacrifices tracking accuracy)<br>1: Do not smooth | | R/W | With immediate effect |
| SD7160+4i (i=0,1,2,3) | double | belt feedback position | mm | R | |
| SD7180+4i (i=0,1,2,3) | double | belt filter location | mm | R | |
| SD7200+40i (i=0,1,2,3) | float | belt feedback position | mm | R | |
| SD7202+40i (i=0,1,2,3) | float | belt filter location | mm | R | |
| SD7204+40i (i=0,1,2,3) | float | belt feedback speed | mm/s | R | |
| SD7206+40i (i=0,1,2,3) | float | belt filter speed | mm/s | R | |
| SD7208+40i (i=0,1,2,3) | INT16U | Current work queue length | | R | |
| SD7209+40i (i=0,1,2,3) | INT16U | Available capacity in the workpiece queue | | R | |
| SD7210+40i+2j (i=0,1,2,3. j=0,1,2,...,5) | float | The robot tool's end position is given in the workpiece coordinate system, with 6 floats. | | R | |
| SD7222+40i+2j (i=0,1,2,3. j=0,1,2,...,5) | float | Feedback from the robot tool end position to the workpiece coordinate system, with 6 floats | | R | |
| SD7234+40i (i=0,1,2,3) | float | The belt's average feedback speed refreshes every 200ms | mm/s | R | |
| SD7236+40i (i=0,1,2,3) | float | The belt filter averages the speed and refreshes every 200ms | mm/s | R | |
| SD7360 | INT32U | Number of successful follow-up items | | R | |
| SD7362 | INT32U | Number of failed workpieces | | R | |
| SD7364 | INT16U | Follow result of the previous workpiece | | R | |
| SD7365 | INT16U | robotry | | R | |
| SD7370+2i (i=0,1,2,...,5) | float | The robot tool's end position in the workpiece coordinate system, 6 floats in total | | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD7382+2i (i=0,1,2,3) | float | Time required for processing the photo information of the workpiece | ms | R | |
| SD7400 | float | The radius of the intersecting line's main pipe is entered by the user. | mm | R/W | The next delivery point will take effect |
| SD7402 | float | The radius of the intersecting line branch is entered by the user | mm | R/W | The next delivery point will take effect |
| SD7404 | float | The offset of the branch pipe relative to the main pipe is entered by the user (positive on the X-axis) | mm | R/W | The next delivery point will take effect |
| SD7406 | float | The deflection angle of the branch pipe relative to the main pipe is input by the user (positive for clockwise rotation around the x-axis, range: -π/2 to π/2). | linear measure | R/W | The next delivery point will take effect |
| SD7408 | float | Intersection line teaching error threshold, in mm (default: 0.5) | Default: 0.5 | R/W | The next delivery point will take effect |
| SD7410 | float | error of intersection line teaching | mm | R | |
| SD7412 | float | The calculated radius of the main pipe of the intersection line | mm | R | |
| SD7414 | float | The calculated radius of the intersecting line branch | mm | R | |
| SD7416 | float | The calculated offset of the intersecting line branch pipe relative to the main pipe | mm | R | |
| SD7418 | float | The calculated deflection angle of the intersecting line branch pipe relative to the main pipe | linear measure | R | |
| SD7420 | float | Calibration error of the main coordinate system for the intersection line | mm | R | |
| SD7422+2*i (i=0,1...5) | float | Calibration results of the intersection line's main coordinate system, Euler angles (zyx), position units (mm), and attitude units (°) | | R | |
| SD7441 | INT16U | Optimize speed check switch: 0: Off, 1: On. Default is 0.<br>The CP optimize switch and the save constraint Cartesian switch are merged into one. When enabled, the internal forward window automatically adjusts to 5, and the forward step length automatically adjusts to 3. | | R/W | The next delivery point will take effect |
| SD7442 | INT16U | CP adjusts the step size percentage (unit:%), default value 70, adjustment range 1-90 | | R/W | The next delivery point will take effect |

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| SD7443 | INT16U | The joint speed extremum search count is typically set to the default value (0). When the value exceeds 5, the algorithm increases search frequency for more accurate extremum prediction, though this may prolong computation time and potentially cause PLC offline. Conversely, setting the value between 0 and 5 reduces search frequency to minimize computation time, but this compromises extremum prediction accuracy, potentially failing to predict overspeed joints. | | R/W | The next delivery point will take effect |
| SD7940 | Float | Hand cycle retreat speed, the percentage of the current instruction speed, similar to SD5540 | 0 | R/W | With immediate effect |
| SD7942 | INT16U | Hand rotation direction: 0-no direction, 1-forward, 2-reverse | 0 | R/W | With immediate effect |
| SD8010 | INT32U | Current communication interval | ns | R | |
| SD8012 | INT32U | EtherCAT operation time | ns | R | |
| SD8014 | INT32U | Maximum communication interval | ns | R | |
| SD8016 | INT32U | Maximum time for EtherCAT operation | ns | R | |

D register

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| D29000+2*j+6*i (i=0,1,2,3， j=0,1,2) | float | Coordinate the teaching point positions required for coordinate system calibration, 4 points, each point only needs xyz position information | | R/W | |

HSD register

| Register address | Type | Definition | Value/unit | Read-write | Trigger time |
|---|---|---|---|---|---|
| HSD100+20i (i=0,1...maxAxisnum-1) | INT64S | The current pulse set value, the encoder count relative to the absolute zero position | | R/W | |
| HSD104+20i (i=0,1...maxAxisnum-1) | INT64S | The current encoder feedback value, relative to the encoder count value of the absolute zero position | | R/W | |
| HSD108+20i (i=0,1...maxAxisnum-1) | INT64S | The current displacement pulse count, the encoder count value relative to the absolute zero position | | R/W | |

SM register

| Register address | Type | Definition | Value/unit | Read-write | Trigger |
|---|---|---|---|---|---|
| SM1991 | Bit | Communication status flag, ON for errors | | R | |
| SM2000+20i (i=0,1...maxAxisnum-1) | Bit | Enable the axis. When enabled, it is ON. | | R | |
| SM2001+20i (i=0,1...maxAxisnum-1) | Bit | The command is running | | R | |
| SM2002+20i (i=0,1...maxAxisnum-1) | Bit | Command direction: ON for positive direction | | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger |
|---|---|---|---|---|---|
| SM2003+20i (i=0,1...maxAxisnum-1) | Bit | Command completed | | R | |
| SM2004+20i (i=0,1...maxAxisnum-1) | Bit | Axis error | | R | |
| SM2005+20i (i=0,1...maxAxisnum-1) | Bit | Minimum limit position of shaft | | R | |
| SM2006+20i (i=0,1...maxAxisnum-1) | Bit | Maximum limit position of shaft | | R | |
| SM2007 | Bit | Full closed loop state | | R | |
| SM2010+20i (i=0,1...maxAxisnum-1) | Bit | Axis enable | | R/W | With immediate effect |
| SM2011+20i (i=0,1...maxAxisnum-1) | Bit | Positive point motion of axis | | R/W | With immediate effect |
| SM2012+20i (i=0,1...maxAxisnum-1) | Bit | Axis reverse point motion | | R/W | With immediate effect |
| SM2013+20i (i=0,1...maxAxisnum-1) | Bit | Clear the servo alarm | | R/W | With immediate effect |
| SM2014+20i (i=0,1...maxAxisnum-1) | Bit | Positive idiosyncraticity | | R/W | With immediate effect |
| SM2015+20i (i=0,1...maxAxisnum-1) | Bit | Back to Origin | | R/W | With immediate effect |
| SM2016+20i (i=0,1...maxAxisnum-1) | Bit | Full closed-loop enablement | | R/W | With immediate effect |
| SM2017+20i (i=0,1...maxAxisnum-1) | Bit | Stop motion | | R/W | With immediate effect |
| SM2018+20i (i=0,1...maxAxisnum-1) | Bit | Axis Simulation. After enabling axis simulation, you can connect to the actual physical axis. | | R/W | With immediate effect |
| SM3000 | Bit | Robot enabled state | | R | |
| SM3001 | Bit | Robot movement flag | | R | |
| SM3004 | Bit | Report to the police | | R/W | With immediate effect |
| SM3005 | Bit | Warn | | R/W | With immediate effect |
| SM3009 | Bit | Command given end flag | | R | |
| SM3012 | Bit | Position track transition sign<br>1: Enter transition<br>0: End transition | | R | |
| SM3013 | BOOL | Positive solution calculation control word | | R/W | With immediate effect |

| Register address | Type | Definition | Value/unit | Read-write | Trigger |
|---|---|---|---|---|---|
| SM3014 | Bit | Enable CP transition state word. Set ON when the robot is stationary and OFF when RC commands start interpolation. Set ON when the current interpolation segment reaches the CP transition point or the current segment is completed, and set OFF when the next segment begins interpolation. | | R | |
| SM3015 | Bit | Accessibility check word | | R/W | With immediate effect |
| SM3016 | Bit | Determine the end flag of the acceleration phase | 1: Start the constant speed phase, 0: Non-uniform phase | R | |
| SM3017 | Bit | Check for filter flag | 1: Filter point exists. 0: Filter point does not exist. | R | |
| SM3018 | Bit | Tooling parameter refresh control bit | | R/W | With immediate effect |
| SM3019 | Bit | Coordinate conversion control word | | R/W | With immediate effect |
| SM3040 | Bit | Robot enabled control bit | | R/W | With immediate effect |
| SM3041 | Bit | External axis system enable control bit. The external axis system can only be enabled after the robot is successfully enabled. | | R/W | With immediate effect |
| SM3042 | Bit | Add a marker. The first interpolation point of each segment is discarded and must be reset manually. | | R/W | With immediate effect |
| SM3050 | Bit | X forward point movement | | R/W | With immediate effect |
| SM3051 | Bit | X negative point movement | | R/W | With immediate effect |
| SM3052 | Bit | Y positive point movement | | R/W | With immediate effect |
| SM3053 | Bit | Y negative point movement | | R/W | With immediate effect |
| SM3054 | Bit | Z positive point movement | | R/W | With immediate effect |
| SM3055 | Bit | Z negative point movement | | R/W | With immediate effect |
| SM3056 | Bit | RZ forward point movement | | R/W | With immediate effect |

| Register address | Type | Definition | Value/unit | Read-write | Trigger |
|---|---|---|---|---|---|
| SM3057 | Bit | RZ negative point movement | | R/W | With immediate effect |
| SM3058 | Bit | RY forward point movement | | R/W | With immediate effect |
| SM3059 | Bit | RY negative point movement | | R/W | With immediate effect |
| SM3060 | Bit | RX forward point movement | | R/W | With immediate effect |
| SM3061 | Bit | RX negative point movement | | R/W | With immediate effect |
| SM3070 | Bit | J1 forward point movement | | R/W | With immediate effect |
| SM3071 | Bit | J1 negative point movement | | R/W | With immediate effect |
| SM3072 | Bit | J2 forward point movement | | R/W | With immediate effect |
| SM3073 | Bit | J2 negative point movement | | R/W | With immediate effect |
| SM3074 | Bit | J3 forward point movement | | R/W | With immediate effect |
| SM3075 | Bit | J3 negative point movement | | R/W | With immediate effect |
| SM3076 | Bit | J4 forward point movement | | R/W | With immediate effect |
| SM3077 | Bit | J4 negative point movement | | R/W | With immediate effect |
| SM3078 | Bit | J5 forward point movement | | R/W | With immediate effect |
| SM3079 | Bit | J5 negative point movement | | R/W | With immediate effect |
| SM3080 | Bit | J6 forward point movement | | R/W | With immediate effect |
| SM3081 | Bit | J6 negative point movement | | R/W | With immediate effect |
| SM3150 | Bit | Secure area parameter refresh control word | | R/W | With immediate effect |
| SM3151 | Bit | Processout parameter refresh control word | | R/W | With immediate effect |

| Register address | Type | Definition | Value/unit | Read-write | Trigger |
|---|---|---|---|---|---|
| SM3152 | Bit | Control word for attitude form conversion of monitoring register | | R/W | With immediate effect |
| SM3153 | Bit | Till function trigger status bit | | R | |
| SM3154 | Bit | Variable cycle stop state bit | | R | |
| SM3155 | Bit | Variable cycle speed state bit | | R | |
| SM3160 | Bit | RC command trigger control bit | | R/W | With immediate effect |
| SM3161 | Bit | RC instruction execution status bit | | R | |
| SM3162 | Bit | The data read module is in the execution state bit | | R | |
| SM3163 | Bit | The data filter module is in execution status | | R | |
| SM3164 | Bit | The foreground processing module is executing the status bit | | R | |
| SM3165 | Bit | RC instruction interpolation is in execution status | | R | |
| SM3166 | Bit | Variable cycle slow stop state | | R | |
| SM3167 | Bit | RC command error | | R | |
| SM3168 | Bit | Dynamic adjustment module is executing the status bit | | R | |
| SM3171 | Bit | The robot is performing arc interpolation with the flag set to 1. After the interpolation calculation starts, the flag is set to 0. | | R | |
| SM3173 | Bit | Arc instruction error in multi-stage pendulum | | R | |
| SM3182 | Bit | Dynamic adjustment: Dynamic offset real-time modification control word | | R/W | With immediate effect |
| SM3190 | Bit | External shaft system enable status bit | | R | |
| SM3191 | Bit | External axis motion flag | | R | |
| SM3192 | Bit | External axis instruction interpolation completion flag | | R | |
| SM3193 | Bit | The welding system is in motion. | | R | |
| SM3194 | Bit | Welding system instruction given completion flag | | R | |
| SM3199 | Bit | External axis instruction given completion flag | | R | |
| SM3201 | Bit | Earth orbit calibration control point | | R/W | With immediate effect |
| SM3202 | Bit | Variable speed machine calibration control position | | R/W | With immediate effect |
| SM3203 | Bit | Self-calibration trigger control bit | | R/W | With immediate effect |
| SM3204 | Bit | Multi-point Optimal TCP Calibration Trigger Control Bit | | R/W | With immediate effect |
| SM3205 | Bit | Least Squares TCP Calibration Trigger Control Bit | | R/W | With immediate effect |

| Register address | Type | Definition | Value/unit | Read-write | Trigger |
|---|---|---|---|---|---|
| SM3206 | Bit | Error Model TCP Calibration Trigger Control Bit | | R/W | With immediate effect |
| SM3208 | Bit | Control Point for Coordinate System Calibration Calculation | | R/W | With immediate effect |
| SM3210 | Bit | Workpiece calibration trigger control bit | | R/W | With immediate effect |
| SM3212 | Bit | EtherCAT reactivation control bit | | R/W | With immediate effect |
| SM3214 | Bit | Some SFD parameters are being refreshed, including body parameters, axis control settings, tool and workpiece information, and speed configuration. | | R/W | Trigger when the robot is not enabled |
| SM3220 | Bit | Singular avoidance function switch: 0: off, 1: on | | R/W | With immediate effect |
| SM3300+2i (i=0,1...number of external shaft joints minus 1) | Bit | External shaft joint forward point movement | | R/W | With immediate effect |
| SM3301+2i (i=0,1...number of external shaft joints minus 1) | Bit | Negative point movement of the external shaft joint | | R/W | With immediate effect |
| SM3390 | Bit | Handwheel rollback function control word | 0 | R/W | With immediate effect |
| SM3391 | Bit | Hand Return Function Status Word | 0 | R | |
| SM3392 | Bit | The first trigger flag for the handwheel rollback function. When the handwheel function is enabled, the variable is initialized to 1 and will not be assigned a value thereafter. The platform can reset it. | 1 | R | |
| SM3393 | Bit | Set the hand's return function trajectory end flag. This value is set when the point trajectory in the current movement direction has completed. It resets after the directional change movement response. | 0 | R | |
| SM3400 | Bit | Fourier instruction start control bit | | R/W | With immediate effect |
| SM3401 | Bit | Fourier instruction run status bit | | R | |
| SM3402 | Bit | Fourier instruction exception status bit | | R | |
| SM3410 | Bit | Load identification instruction start control bit | | R/W | With immediate effect |
| SM3411 | Bit | Load identification operation status bit | | R | |
| SM3412 | Bit | Load identification of abnormal status bits | | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger |
|---|---|---|---|---|---|
| SM3420 | Bit | Improved trajectory planning execution dynamic speed regulation flag<br>During interpolation, the system automatically adjusts SD5540 when entering the overspeed zone, setting SM3420 to ON. After exiting the overspeed zone, SD5540 and SM3420 are reset to OFF. The automatic adjustment process may be interrupted or forced to stop, and the original SD5540 values cannot be restored after stopping. | | R | |
| SM3430 | Bit | The soft reset function of the robot system, when triggered, restarts the PLC program and updates SFD parameters via this register (Note: The PLC will briefly go offline after triggering). | | R/W | Trigger when the robot is not enabled |
| SM3440 | bit | Belt follow function control bit, active on high level, off by default | | R/W | With immediate effect |
| SM3441+i (i=0,1,2,3) | bit | Photo signal, high level active | | R/W | With immediate effect |
| SM3445+i (i=0,1,2,3) | bit | Photo processing complete signal, high level valid | | R/W | With immediate effect |
| SM3449 | bit | Workpiece sort control bit, active on high level | | R/W | With immediate effect |
| SM3450 | bit | Workpiece query signal, high level valid | | R/W | With immediate effect |
| SM3451 | bit | Workpiece query success signal, high level valid | | R/W | With immediate effect |
| SM3452 | bit | Failed workpiece query signal, high level valid | | R/W | With immediate effect |
| SM3453+i (i=0,1,2,3) | bit | The belt enable control bit is active at high level | | R/W | With immediate effect |
| SM3458 | bit | Compensate queue length control bit, active on high level | | R/W | With immediate effect |
| SM3490 ( obligate ) | bit | The belt stop function control bit is active at high level and is disabled by default. | | R/W | With immediate effect |
| SM3495 ( obligate ) | bit | The robot signals the workpiece grasping action, with high level being valid. | | R/W | With immediate effect |
| SM3460 | bit | Belt following motion state position | | R | |
| SM3461 | bit | Belt following interpolation status bit | | R | |
| SM3462 | bit | Belt following catch-up state | | R | |
| SM3463 | bit | Belt following synchronization status bit | | R | |

| Register address | Type | Definition | Value/unit | Read-write | Trigger |
|---|---|---|---|---|---|
| SM3464 | bit | Belt following deceleration state position | | R | |
| SM3465 | bit | State of completion of following motion | | R | |

# Appendix 2 Summary of Abnormalities and Treatment Plans

The robot control system outputs three types of abnormal codes, with monitoring addresses SD4130 (single word), SD4131 (single word), and SD2002+60i (single word, where i is the axis number, with 0 indicating axis 1). SD4130 and SD4131 display direct alarm codes. When SD4130=1009, it indicates that a specific axis has triggered motion bus exception handling. To identify the exact cause, check the SD2002+60i codes and their explanations for all robot axes.

Robot error class exception code

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 1 | Robot body parameter error | Incorrect robot arm length or gear ratio setting | Check the register for the robot's arm length or gear ratio settings |
| 2 | Server error | The robot's axis servo failed or the driver lost connection | Check the single-axis alarm cause or servo status |
| 5 | Axis control configuration error | The robot axis configuration is incorrect. | Check the controlled configuration register of the robot axis |
| 6 | The servo is not enabled | Not all single axes are enabled when enabling robots | Check if the robot axis is enabled |
| 7 | Device error | The selected device model is not in the device library. | Check if the selected SFD5000 model is correct |
| 1001 | Alarm for excessive movement of the soft limit of the current point joint | Move beyond the joint soft limit. The soft limit is determined and set based on a threshold. | The joint is dynamically returned to its normal soft limit range. |
| 1006 | The current limit setting is invalid | The current maximum limit is lower than the minimum limit | Check if the robot limit is reasonable |
| 1007 | The positioning width setting is incorrect | The positioning width settings for SFD5102, SFD5112, SFD5114, SFD5116, and SFD5118 exceed the range (0,10). | Check the SFD5102, SFD5112, SFD5114, and SFD5116. SFD5118 register value |
| 1008 | Articular tachycardia | The joint speed exceeds the maximum allowable value | The log analysis indicates that small displacement with large posture changes may cause joint overspeed (1008). To resolve this, reduce the robot's speed (D20) or minimize the magnitude of posture variations. |
| 1009 | Single bus layer alarm | The bus shaft has an alarm. | Check the SD2002+60*i bus axis alarm code |
| 1010 | External shaft joint overtravel alarm | The external shaft feedback angle exceeds the maximum and minimum limits | Check the relationship between the current external shaft feedback angle and the maximum and minimum limit positions |
| 1011 | External axis system enable failed | When enabling the external shaft system, the shaft is not fully enabled, resulting in no external shaft system servo being normally enabled. | Check if all external shafts are enabled successfully |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 1012 | The motor parameter is set to 0 | The robot or external axis has controlled-axis encoder pulses, motor displacement, or reduction ratio set to 0 | Check the settings for SFD3002+60 (i-1), SFD3004+60 (i-1), and SFD5020+2i. |
| 1013 | The robot sends out pulse signals that are not numerical in nature. | Generally not present | Contact developers |
| 1014 | External axis pulse non-numeric | Generally not present | Contact developers |
| 1015 | The torque value is too high. | The torque value is too large in the torque mode | Reduced torque |
| 1020 | Invalid belt type | Incorrect belt type setting | Set the correct belt type |
| 1021 | The coordinate system number of the belt-bonded workpiece is incorrect | The coordinate system number for the belt-bound workpiece is incorrect | Incorrect workpiece coordinate system sequence number |
| 1022 | The belt encoder has an error in the pulse equivalent value. | The pulse equivalent of the belt encoder is not positive. | Ensure the calibration result of the belt encoder pulse equivalent is positive |
| 1023 | The stop distance or boundary is not set properly | The relative position of the trailing line to the trigger lower boundary upstream or between boundaries is unreasonable | The follow stop line is positioned upstream of the work lower bound, with a stop distance from the work lower bound, ensuring the follow stop line is downstream of the trigger lower bound. The order of the boundary lines from upstream to downstream is: work upper bound, trigger lower bound, follow stop line, work lower bound. |
| 1025 | The belt number for the workpiece is incorrect | The belt for the workpiece is not in use | When multiple belts operate in rotation, avoid checking the workpiece in synchronized mode. |
| 1036 | Bus time setting error | The synchronization time of the master station differs from the SFD2990 value. | Check the configuration time of the Ether CAT master station and the SFD2990 values |
| 1040 | The initial values of the singular threshold, nominal damping coefficient, and error compensation coefficient are not set. | Initial value not set | Set initial values based on recommendations |
| 1041 | The robot is approaching the limit of its workspace. | The robot's workspace has exceeded its limits. | Reverse point moving robot |
| 1042 | The robot enters two or more singular regions simultaneously | The robot enters multiple singular regions simultaneously | Dynamic robot with joint avoidance in multi-singular region |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 1043 | robotic shoulder singularity | Robot enters the shoulder strange zone | For Cartesian continuous point motion, it is recommended to enable the continuous point motion singularity avoidance function.<br>For RcPath, it is recommended to reduce the Cartesian speed. |
| 1044 | Robot elbow singularity | Robot enters elbow singular region | For Cartesian continuous point motion, it is recommended to enable the continuous point motion singularity avoidance function.<br>For RcPath, it is recommended to reduce the Cartesian speed. |
| 1045 | Robot wrist singularity | Robot enters the strange zone of wrist | For Cartesian continuous point motion, it is recommended to enable the continuous point motion singularity avoidance function.<br>For RcPath, we recommend enabling the hybrid space singular avoidance feature. |
| 1046 | robotic joint overspeed | Joint Overspeed of Robot After Processing by the Strange Avoidance Algorithm | Reduce the Cartesian point speed |
| 1047 | RcPath wrist singular avoidance type setting error | The singular type is not in the range 0-3. | Set the correct singular avoidance type parameter |
| 1048 | Scara robot does not support Cartesian point motion at singular points | Scara robot's Cartesian point movement at the singularity | The Scara point motion is performed to avoid singularity by moving the joint point in a small angle along the 2-axis or in the opposite direction to the limit position. |

Robot warning type of abnormal code

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10001 | Alarm for excessive movement of the joint soft limit in the trajectory planning point | The target point exceeds the soft limit of the joint. | Check the collected log to see if the inverse-engineered joint values exceed the soft limit. |
| 10002 | No speed is set | The command speed percentage is less than or greater than a given minimum speed threshold. | Check the instruction speed setting |
| 10007 | Acceleration is not set | The command acceleration percentage is less than or greater than a given minimum speed threshold | Check the acceleration setting for the command. |
| 10012 | Tool ID settings exceed the limit | The tool ID must be 0, 1,..., 19 | Check tool number value |
| 10013 | The workpiece coordinate system exceeds the limit | The workpiece number must be 0, 1,..., 19 | Check the workpiece number |
| 10020 | The error message SM3004 is not cleared when executing the motion command | Cannot execute motion commands when SM3004=1 is invalid | Check error, clear alarm SM3004=0 |
| 10021 | When executing the motion command, the robot remains in motion, i.e., SM3001 is ON | Time sequence protection | The robot cannot trigger the motion command during movement. |
| 10022 | The robot did not enable the SM3000 when executing the motion command. | The robot cannot execute the motion command. | Check if the robot is enabled or the reason for enabling failure |
| 10031 | Kinematic inverse error | The point robot is unreachable | Check based on log information |
| 10032 | Incorrect motion solution | Some models have a linkage parameter check for the positive solution. An incorrect setting may prevent the positive solution from being calculated. | Refer to the manual for the connecting rod section based on the aircraft model |
| 10035 | Six-axis does not support Euler angle mode | The current model is a six-axis model with SFD5106=1 | Switch to quaternion mode |
| 10037 | Arc point error | The collinear points cannot form an arc trajectory | Check the arc or full circle instruction point information |
| 10063 | Invalid swing parameter value | The input swing parameter contains a negative number, the unit swing distance is less than 0, or the dwell time at the upper, middle, and lower swing points is too large. | Reset swing parameters |
| 10064 | Parameter calculation error after length adjustment | After applying the integer pendulum correction, the calculated distance between the upper, middle, and lower pendulum points exceeds the limit. | Check based on log information |
| 10082 | Swing time calculation error | Base-level errors rarely occur | Contact developers |
| 10083 | Sway direction displacement interpolation error | Base-level errors rarely occur | Contact developers |
| 10084 | Non-transition adjacent trajectory angle transition | The point angle setting is incorrect | Reset swing angle |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10086 | The number of single pendulums exceeds the limit. | The total displacement of the multi-tappend pendulum is too large, and the number of single pendulums exceeds the limit. | Reduce the number of oscillation points, execute multiple instructions, or increase the oscillation length (SD4840) |
| 10088 | Arc plane calculation error | The forward segment trajectory coincides with the z-axis of the TCP coordinate system and is related to the welding posture. | Identify issues at the underlying level based on log information |
| 10095 | The selection of the plane for the forward and backward interpolation points is too fast | The plane interpolation calculation protection is disabled because the main trajectory transition is too short or the main trajectory is moving too fast. | If the main trajectory accuracy permits, the transition error value may be increased or the SD5322 adjusted (which may cause overspeed risks). |
| 10096 | The arc mode supports only external axis motion modes 0 and 1 | The external axis motion mode is not set correctly | Checkpoint external axis motion mode settings |
| 10097 | Collaborative arc property error | When SD5579 is ON, the swing plane cannot follow the attitude change. | Turn off SD5579 or set the swing plane to not change with attitude |
| 10100 | RCPATH command failed to execute | The RCPATH instruction module is not initialized or has an error in state machine switching. | Restart the PLC. Contact the developer if needed. |
| 10101 | The RCPATH instruction sets an unreasonable number of machining point data | The number of processing points is set to a value less than 1 | The instruction is not executed. Check the corresponding register assignment. |
| 10103 | The user set the track type incorrectly | The path type is incorrect | Check if the D28 settings for the distribution point are correct |
| 10104 | The attitude setting is incorrect. The attitude change between adjacent processing points is too large. | The attitude changes too much relative to the position | Attitude changes cannot transition from A to B and back to A unless point B is set as the zero-speed point. |
| 10105 | Forward Planning Error | The input trajectory parameters are generally incorrect. | The collected log view typically shows that the acceleration or double-acceleration reference value is not set. |
| 10106 | Forward Backtracking Error | The input trajectory parameters are generally incorrect. | View collection logs |
| 10107 | Forward-backward backtracking failed. The backtracking to the maximum allowable small segment still does not meet the requirements. | The deceleration zone is usually too short to slow down sufficiently, causing the speed to drop too much. | Reduce the speed or increase the speed preview window size |
| 10108 | The RC instruction is executing when the motion command is being executed. | Instruction protection prevents RC from being interfered with by other instructions during execution. | Do not interrupt the RC instruction, only trigger an alarm |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10109 | trajectory planning error | The input trajectory parameters are incorrect. The speed, acceleration, and displacement parameters contain 0 or negative values.<br>and the maximum speed at the end SFD5080,<br>The maximum acceleration at the end of the SFD5090 model is...<br>The maximum acceleration at the end of the SFD5270 model<br>The maximum joint speed of the SFD5040 model is...<br>The maximum joint acceleration of the SFD5060 model<br>The maximum joint acceleration of the SFD5250 model<br>The maximum speed of the end position posture is SFD6030.<br>The maximum acceleration of the end position posture is SFD6032.<br>The maximum acceleration of the end position posture is SFD6034.<br>The maximum speed of the external shaft is SFD6100.<br>The maximum acceleration of the external shaft is SFD6120.<br>The maximum acceleration of the external shaft is related to the SFD6140 model. | Check the PLC power-up to verify the register settings. |
| 10110 | Trajectory planning error | The input parameters are invalid. The displacement length does not meet the speed change requirements. | The input data is invalid due to a trajectory calculation error. |
| 10111 | Error in solving a cubic equation | Speed planning calculation error | Contact developers |
| 10112 | Trajectory planning error | Velocity model error | This usually doesn't occur. If it does, it indicates data corruption. |
| 10113 | Trajectory planning error | Most of them are caused by the calculation accuracy | This usually doesn't happen due to calculation precision. |
| 10114 | Trajectory interpolation error. The trajectory planning buffer is empty. | Forward processing is too slow | You can increase the reserved time SD5546, or it might be caused by an unknown bug. |
| 10115 | Trajectory interpolation error | It doesn't usually happen | This usually doesn't occur. If it does, it indicates data corruption. |
| 10116 | Dynamic adjustment speed ratio is too small | The speed percentage is less than 0.01 (%). | Only report to police, without affecting current operations |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10117 | Rbgoon instruction execution status error | The RCPATH command's recovery robot joint angle or external axis joint angle is not at the paused position, and the joint deviation must be within 0.001 degrees to be valid. | Check the suspension stop for joint angle SD5548 backup of the robot and joint angle SD5840 backup of the external axis. |
| 10119 | The maximum operating speed of the first processing point is incorrect | No speed or percentage set beyond 100 | Check the first point speed percentage |
| 10120 | The maximum operating acceleration of the first processing point is incorrectly set | Acceleration or percentage is not set or exceeds 100 | Check the percentage of acceleration or deceleration at the first point |
| 10121 | The maximum operating acceleration setting for the first processing point is incorrect | No acceleration or percentage above 100 is set | Check the first point's acceleration percentage |
| 10122 | The pose of the input processing data is not a unit quaternion | Input quaternion is not a unit quaternion | Identify and correct erroneous processing points using SD5452 |
| 10124 | The angle limit set during the transition of the fusion trajectory is unreasonable | The SD5534 typically has an improper configuration. | Set SD5534 as valid data |
| 10125 | The input data hand binding does not match the current point hand binding | The inconsistency of the starting and ending points of the Descartes trajectory | Cartesian trajectory does not support variable hand system |
| 10126 | Data transition module calculation error: non-numeric or infinite value | This is not usually the case. Check the checkpoint information. | Contact developers |
| 10127 | The foreground processing window size is not properly set | It is generally greater than the forward translation size per step | Check the size relationship between SD5536 and SD5537 |
| 10128 | Incorrect judgment of the forward-looking attitude attribute | This usually doesn't happen. If the posture information is abnormal, collect the log and contact the developer. | Contact developers |
| 10129 | Incorrect number of data reads set | It is generally greater than the number of projections per session | Check the size relationship between SD5536 and SD5572 |
| 10133 | Error in interpolation time calculation | Time sequence errors rarely occur. Contact the developer if logging occurs. | Contact developers |
| 10134 | Error in interpolation time calculation | Time sequence errors rarely occur. Contact the developer if logging occurs. | Contact developers |
| 10135 | Error in assigning the fusion trajectory attitude forward-looking type | This usually doesn't happen. Contact the developer if logging occurs. | The motion command is not executed. |
| 10136 | The Jump instruction activates the external axis function | Jump instruction does not support external axis movement | Check for external shaft motion superimposed on the Jump instruction |
| 10137 | The Jump instruction activates the arc function | Jump instruction does not support arc motion | Check for arc motion superimposed on the Jump instruction |
| 10138 | Jump movement LimZ settings are incorrect | Setting the LimZ value causes the Jump midpoint to exceed the limit | reset limZ |
| 10140 | The external axis system is not enabled | The external axis system is not enabled for the current operation. | Check if SM3041 and the single axis are enabled |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10141 | External axis calibration error | The movement distance of each axis in the external axis calibration is too short, less than 10mm or degrees. | Verify that the actual movement of each external shaft during calibration does not exceed the recommended values of 200mm or 20 degrees. |
| 10142 | Calibration error of the positioner | The alignment of the positioner is either collinear or the axes of adjacent joints are parallel. | The calibration algorithm of the integrated position machine at the bottom does not support parallel axis calibration |
| 10143 | The external shaft system number is incorrect | The external axis system number is incorrect or does not match the part number set at the point. | Set the external axis SD5593 to 1 or 2 |
| 10144 | Incorrect setting for the external shaft motion type | All external axis point motion types are inconsistent. A rc instruction supports only one motion type. | The movement type D34 at all points must be consistent when the external axis executes the RCPATH instruction. |
| 10145 | Excessive transformation of the external axis collaborative motion trajectory | The external axis's movement duration is either too long or the robot's movement duration is too short, resulting in synchronization failure between the external axis and the robot. | You can increase the speed of the external axis, raise the transition percentage of the external axis, or reprogram to reduce the displacement of the external axis. |
| 10146 | The external axis is moving | The external axis is in motion and cannot execute other motion commands. | The external axis is in motion when the RCPATH instruction is executed. |
| 10147 | external shaft overspeed | This usually doesn't happen but may occur during external axis CP transitions when the overlay speed exceeds the limit. | For cooperative movement, try reducing the robot speed. For non-cooperative movement, try reducing the external axis speed. |
| 10148 | orbit calibration error | An external axis is moving when a certain axis is timed, with a threshold of 0.1 degree. | recalibrate the positioner |
| 10149 | Calibration error of the positioner | When calibrating one axis, if another external axis moves or one axis is not at zero position, the threshold is 0.1 degree or mm. | recalibrate the positioner |
| 10150 | External axis data read error | External axis mode is incorrect | Collect alarm logs and contact developers |
| 10151 | Spline parameter calculation error | Limit the iteration count of the radius to the maximum | Collect alarm logs and contact developers |
| 10154 | Spline parameter calculation error | The interpolation time t exceeds the maximum number of iterations. | Collect alarm logs and contact developers |
| 10155 | Spline parameter calculation error | The calculation of control points overlaps, and the iteration error causes | Collect alarm logs and contact developers |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10156 | PTP mode 2 interpolation error | The number of iterations in calculating the interpolation time t of PTP exceeds the limit | Collect alarm logs and contact developers |
| 10157 | The post-up/down model point is unreachable | The trajectory points of the rear elevation model after obstacle avoidance exceed the limits. | Check the information of the parameter inspection points for the connecting rod. |
| 10158 | Error in calculating spiral line data for the post-ascension and descent model | The input point information cannot form a spiral line | Collect alarm logs and contact developers |
| 10160 | The external axis coordinate system is set incorrectly | The reference coordinate system for the robot point is incorrect when the external axis moves in coordination. Set it to the workpiece coordinate system. | Under the RCPATH instruction, the point reference coordinate system for external axis collaborative motion can only be set to the workpiece coordinate system. |
| 10161 | External axis motion type setting exceeds limits | The external axis motion type number exceeds the limit. | Check the motion type setting for the external axis at the error point. RC commands only support motion types 0, 1, 2, 3, or 5. The external axis cannot exceed 5. |
| 10162 | The external axis calibration axis has no forward movement | The external calibration axis can only move in the forward direction and checks only the current calibration axis. | Check if the external shaft rotates in the opposite direction |
| 10163 | The axis movement of the positioner exceeds the limit during calibration. | During calibration, each axis of the positioner must not rotate forward beyond 360 degrees. | Check if the angle change during external axis calibration exceeds 360 degrees |
| 10164 | External shaft system configuration error | The common ground orbit configuration is incorrect. Currently, only System 1 ground orbit axis is controlled and System 2 ground orbit axis is uncontrolled. | Check the external axis controlled configuration parameters |
| 10165 | External axis target point position with hyperarticulation soft limit | The joint angle of the target point exceeds the maximum and minimum limit settings. | Reset target point |
| 10167 | The non-standard axis moves during the Longmen calibration. | When calibrating a specific axis, other axes cannot move. | Re-calibrate the external axis |
| 10171 | The number of reading points is too large | The SD5572 setting exceeds the maximum allowable value of 200 | Check if the SD5572 exceeds the maximum allowable value |
| 10172 | Cache read pointer movement error | Base errors are not typically encountered | Collect alarm logs and contact developers |
| 10181 | The given time is too short to complete time synchronization | Base errors are not typically encountered | Collect alarm logs and contact developers |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10182 | The Iteration Number Exceeding the Limit of S-type Acceleration and Deceleration Control Based on Time Synchronization | Base errors are not typically encountered | Collect alarm logs and contact developers |
| 10190 | The error in the teaching of intersection line points is large. | The demonstration point is incorrect or the user entered incorrect intersection line parameters | When the SD7410 value is high, check the user input parameter register or point input register. If the SD7410 value is within the acceptable range, modify the error threshold register SD7408. |
| 10191 | The Newton iteration of the intersection line exceeds the limit. | The iteration calculation has not converged after the set number of iterations | Contact developers |
| 10192 | The angle parameter for the intersection line arc length calculation is incorrect. | A negative number appears under the square root when calculating rotation angle parameters based on arc length. | Check if the intersection line model meets the requirements and if the two pipe parameters are within the range. |
| 10193 | Invalid rotation type assignment for the intersection line | The rotation type of the intersection line is not clockwise and counterclockwise. | Contact developers |
| 10194 | The curvature of the intersection line is calculated incorrectly. | The first derivative of position is too small, and the denominator for curvature calculation approaches zero. This rarely occurs. | Contact developers |
| 10195 | The intersection line exceeds the allowed length | The overlength exceeds one quarter of the arc length | The overconnection length is set to decrease |
| 10196 | The intersection line offset exceeds the limit. | The absolute value of the offset exceeds the absolute difference between the main pipe radius and the branch pipe radius. | Check the offset setting register. If correct, this type of intersection line is not supported. |
| 10197 | The coordinate transformation matrix of the intersection line is not orthogonal or non-numeric. | The transformation matrix between the intersecting line's principal coordinate system and the robot's base coordinate system is a non-orthogonal unit matrix, which rarely occurs. | Contact developers |
| 10198 | The radius of the main branch line is less than or equal to that of the branch line. | The supervisor radius is no larger than the branch radius | Check the radius value setting register. If correct, this type of intersection line is not supported. |
| 10199 | The intersection line points are in incorrect order or have duplicate points | The teaching sequence does not follow a sequential pattern or include identical teaching points. | Check the order and size of the input points |
| 10200 | The robot is not at the zero position when triggering the Fourier trajectory. | The robot must be at the zero position when running the Fourier trajectory. | returning a robot to zero point |
| 10201 | No joint maximum torque is set | The point attribute is PTP mode 3, but the joint's maximum torque is not set. | Set whether to configure the SFD7170 value |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10209 | The first overspeed point cannot be avoided effectively | The first overspeed point cannot be avoided with the current configuration parameters | Reduce the SD5575's gear shift arrival time or lower the point speed |
| 10210 | The initial value of the calibration parameter is not set | The algorithm initial value is not set when SM3203 is triggered | Check the SFD6600-6606 and set the correct initial values according to the register usage instructions. |
| 10211 | Error in the initial calibration point of the positioner | P2 and P3 are the same point | Measure P2 and P3 again |
| 10212 | Incorrect selection of the calibration tool number for the positioner | No tool parameters for the selected tool were selected during the positioner calibration | Check if the tool selection is correct before calibrating the positioner |
| 10213 | The first parameter of the error model calibration is not obtained | The initial value for the first TCP calibration step was not obtained during the second step of error model calibration. | Check if the corresponding tool number register has a parameter |
| 10214 | The iteration result deviates from the true value. | The iteration result deviates too much. Modify the parameters and recalculate. | Modify parameters and recalculate |
| 10215 | Four-axis TCP repeatability | Four-axis TCP repeatability | Check input |
| 10216 | The teaching points for workpiece calibration may overlap. | The teaching points for workpiece calibration may overlap. | Check the teaching point and re-teach it. |
| 10225 | Tracking is not enabled | The follow function is not enabled when executing related motion commands | Enable belt tracking |
| 10227 | Trigger error during motion tracking | The wrong follow motion section was triggered | The first segment after an ordinary RC point must be a pursuit movement. after a pursuit movement, it must be a synchronized movement or a deceleration movement. after a synchronized movement, it must be a synchronized movement or a deceleration movement. after a deceleration movement, it must be a pursuit movement or an ordinary RC command. |
| 10228 | The track type in the synchronization process is incorrect | RC Trajectory Type Error in the Synchronous Process of Following the Movement | The corresponding trajectory type must be one of the following: straight line, arc, three-line jump, or non-motion segment. |

| Alarm code | Definition | Reason for reporting | Processing and solutions |
|---|---|---|---|
| 10229 | Exceeding the Limit in the Process of Robot Following and Synchronization | The robot goes beyond the upper limit of work or follows the stop line during the synchronous movement | Set the start position of the chase downstream of the work upper boundary and advance it as much as possible. advance the trigger lower boundary appropriately |
| 10235 | The kinematics parameters of the return phase PTP trajectory are invalid | The kinematic parameters of PTP trajectory are non-numeric or infinite values. | Kinematic Parameters of PTP Trajectory |
| 10239 | The belt follows the trajectory of the motion, but the interpolation time calculation is incorrect. | Unknown error. This is highly unlikely to occur. | Contact developers |
| 10250 | The robot's end is about to leave or has already left the work area | The robot is about to move beyond the workspace | Replan the trajectory |
| 10251 | The robot's end is about to enter or has already entered the restricted area | The robot is about to enter the restricted area | Replan the trajectory |
| 10252 | The point loading process has exceeded the limit for the sequence number setting. | The processOut number exceeds the maximum limit | Check if the check point attributes D90 and D91 exceed 100 |
| 10253 | Changing the processOut parameter during the process is invalid | You cannot modify the processOut parameter during the operation | SM3151 modification can be triggered after exercise |

motion bus error code

| Alarm code | Explain | Internal processing method | Processing and solutions |
|---|---|---|---|
| 20001 | Maximum limit overtravel | Jerk | Move the reverse point out of the overtravel area to manually clear the alarm flag and alarm code. |
| 20002 | Minimum soft limit overtravel | Jerk | Move the reverse point out of the overtravel area to manually clear the alarm flag and alarm code. |
| 20003 | Maximum electrical limit overtravel | Jerk | Move the reverse point out of the overtravel area to manually clear the alarm flag and alarm code. |
| 20004 | Minimum electrical limit overtravel | Jerk | Move the reverse point out of the overtravel area to manually clear the alarm flag and code. |
| 20005 | Speed alarm | Stop the deceleration | Decrease target speed |
| 20006 | Position deviation alarm | Emergency stop, enable the shutdown function | Check if servo P0-05 is set correctly (should be 0). Inspect the mechanism for causes of excessive deviation in position commands and feedback, such as locked rotor. After troubleshooting, re-enable. |

| Alarm code | Explain | Internal processing method | Processing and solutions |
|---|---|---|---|
| 20010 | Servo alarm | Emergency stop, enable the shutdown function | Manual clearance of servo alarm messages is available via SM2013+20*N or the F0-00 function on the servo panel. For non-removable alarms, refer to the servo manual for troubleshooting. After resolving the alarm, the alarm flag and code can be manually cleared. |
| 20011 | Servo communication error | Stop, enable the function, and switch the controller's status to online download mode | Check communication parameter settings and connection of communication lines |
| 20020 | Movement command target point out-of-range alarm | Invalid motion command | Change to a reasonable target position |
| 20021 | The target speed for the movement command has exceeded the limit. | Invalid command execution | Change to a reasonable target speed for the instruction |
| 20022 | Too many multi-stage motion commands | Invalid motion command | Change to a reasonable number of multi-stage instructions |
| 20023 | The movement command has exceeded the acceleration and deceleration time limits. | Invalid motion command | Change to reasonable instruction movement acceleration and deceleration time |
| 20025 | The axis number is out of limit | Invalid motion command | Check the axis number of the bound command |
| 20026 | The input point of the return-to-zero terminal is set beyond the limit. | Cannot reset to zero | Check the settings for the return-to-zero input points, including the near-point and origin-point terminals. |
| 20028 | The closed-loop feedback counter port is misconfigured | | The total number of SFD3028+60*i units exceeds the high-speed computing capacity. |
| 20030 | The current motion state does not meet the conditions for instruction execution | Invalid motion command | Wait until the SM2001+20*N flag (indicating active motion) is turned off and the SM2000+20*N flag (indicating servo enable) is turned on before executing the command. |
| 20031 | The motion state of the bound axis does not meet the conditions for executing the binding command. | Invalid binding of the command | Wait until the flag SM2001+20*N (indicating the bound axis is moving) is OFF and the servo enable flag SM2000+20*N is ON before executing the MOSYN instruction. |
| 20032 | The current axis motion mode (SFD3000+60*N) is incorrect. | SM2010+60*N is invalid and cannot enable the servo | Check if the SFD3000+60*N single word is 0, 1, or 2. After correction, re-enable the servo. |
| 20033 | Continue to have | | |
| 20034 | The motor is already enabled when the user performs the enable operation. | This enable operation is invalid | Check the reasons for servo enable, such as enable mode, and re-run the enable operation after correction. |

| Alarm code | Explain | Internal processing method | Processing and solutions |
| --- | --- | --- | --- |
| 20035 | Motor type is not set | The motor cannot be enabled | Check the value of register SFD3001+60*i. After correction, the controller will restart.<br>SFD3001+60*i motor type code:<br>1: Incremental encoder motor<br>2: Single-turn absolute encoder motor<br>3: Absolute encoder motor<br>4: Stepper motor<br>5: XJ Encoder |
| 20036 | Failed to return to the origin. Try again. The movement stopped when searching for the perihelion. | | |

# Manual update log

The data number of this manual is recorded in the lower right corner of the manual cover.

| Order number | Document ID | Update content |
| --- | --- | --- |
| 1 | RC01 20230314 1.0 | The first edition of the manual has been released. |
| 2 | RC01 20230720 1.1 | 1. Modification of JUMP instruction specifications, tool calibration, self-calibration, and non-motion instruction specifications.<br>2. The log analysis section has been removed.<br>3. Known errors in the manual have been corrected. |
| 3 | RC01 20231121 1.2 | 1. Remove the belt tracking section and revise the feedback comments.<br>2. New axis configuration feature description.<br>3. The directory structure has been changed.<br>4. Some functional specifications have been revised. |
| 4 | RC01 20240710 1.3 | 1. A new 7.8-axis collaborative robot has been added.<br>2. Added belt tracking, coordinate system conversion, and timer functions with corresponding descriptions.<br>3. Adjust the specifications for the buffer stop time, CP parameter D17, FIND function, and external axis mode switching.<br>4. After completing the positioning, the system will add new registers and expand HD registers to HD4394304. However, once the value exceeds HD200000, it will no longer be monitorable via XDP.<br>5. Upgrade the main project to 3.7.4b, and reserve HSD10000-HSD12000 to find hardfault problems.<br>6. This manual is compatible with V1.2.0.0 and later versions. |
| 5 | RC01 20250314 1.4 | 1、 New specification for the belt tracking feature.<br>2、 Newly added risk aversion feature description.<br>3、 Add RC full circle trajectory description.<br>4、 Add explanations for some alarm codes.<br>5、 This manual is for System Screen version 3 or later. |

| | | |
|---|---|---|
| 6 | RC01 20250610 1.5 | 1、 Add specifications for the new point offset function.<br>2、 Add specifications for the hand cycle return function.<br>3、 Added D36 point attribute description.<br>4、 This manual is for version 1.2.3.0 and later. |
| 7 | RC01 20250614 1.6 | 1. New content regarding the avoidance of peculiarities<br>2. This manual is applicable to versions V1.2.3.0 and later. |
| 8 | RC01 20250818 1.7 | 1、 Specifications of the SM3392 rotary switch for manual operation<br>2、 Add the relevant register content for the monitoring acceleration curve<br>3、 Additional specifications for arc auxiliary points to execute processOut<br>4、 Add a description for the upper computer version<br>5、 Add CP transition speed limit explanation<br>6、 Update robot-related error warnings<br>7. This manual is applicable to version V1.2.3.0 and later. |

WeChat

WUXI XINJE ELECTRIC CO., LTD.

Address: No.816, West Jianzhu Road, Binhu District, Wuxi City, Jiangsu Province
Tel: 0510-85134136 Fax: 0510-85111290
Website: www.xinje.com Email: xinje@xinje.com