doMotion Control System User Manual

VERSION: V3.0

Contents

Chapter 1 Control System Introduction

1.1 System Constitution

doMotion series robot control system consists of TPU (teach pendant unit), IPC (industrial computer), driver and IO module. The basic-version doMotion control system includes TPU and IPC; meanwhile, the IO module is optional to adapt to the driver based on motor models.

1.1.1 Full Touch Screen Handheld TPU

Fig. 1 Full Touch Screen Handheld TPU

As shown in Fig. 1, the TPU adopts the handheld design with full touch screen to help users with less operation difficulty and develop towards robotic collaboration and intelligence. Please refer to Table 1 for detailed parameters.

Table 1 TPU Parameters

LCD	10 inch TFT 1024 x 768
Touch	Reinforced 4-wire resistive
OS	Linux (support 3D)
virtual keyboard	Support

External USB	2.0 x 1
Power module	12V，30W
Protection level	IP65

1.1.2 IPC

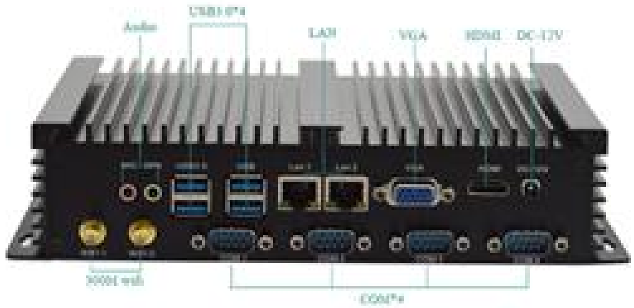

Fig. 2 IPC

As shown in Fig. 2, IPC adopts the OS architecture combined with Linux kernel and industrial real-time kernel, which supports all drive systems with standard EtherCAT protocol and digital IO module expansion. The detailed parameters are shown as follows in Table 2.

Table 2 Parameters of IPC

Hardware Parameters	Dual-core 1.8GHz CPU, integrated GPU, RAM 2GB, User Storage 4GB
Control Servo	BUS type (EtherCAT)
Control Axis	6-axis
Extension Port	16 digital inputs + 16 relay outputs
Operation	demonstration, reproduction, remote, support secondary

Mode	development
Motion Function	joints, straight lines, computation, arcs, NURBS splines, complex curve continuous motion
Command System	motion, logic, I/O, flow, process
Coordinate System	joint coordinates, cartesian coordinates, user coordinates, tool coordinates, base coordinates
Adapted Model	Standard 6-axis, UR type
Power Module	DC 12V， 30W
Working Environment	Temperature 0°C~50°C (no freezing), humidity: 10%-90% (no condensation)

1.1.3 Digital IO Module

Fig. 3 1.1.3 Digital IO Module

IO module is used for communication between the robot system and the tool equipment on site; doMotion control system adopts IO module supporting EtherCAT as the extension port as shown in Fig. 3; please refer to Table 3 for detailed parameters.

Table 3 Specifications of Digital IO Module

Digital Input	
Channel Number	16
Input Type	PNP
Filtering Time	0.5ms
Rated Level	24VDC (-15%/+20%)
"1" Signal Level	15~30VDC (7mA)
"0" Signal Level	0~5VDC
Port Protection	Overvoltage Protection
Digital Output	
Channel Number	16
Output Type	Relay (passive dry contact)
Rated Level	30VDC/250VAC
"1" Signal Level	Relay dry contact closed
"0" Signal Level	Relay dry contact open
Maximum Output Current (single channel)	1A
Maximum Stroke Current (single channel)	2A
Maximum Total Output Current	16A
Contact Resistance	＜ 0.05 Ω

1.1.4 Drive

doMotion control system supports motor drives with various EtherCAT bus structures.

Tsino DYNATRON CoolDrive R/RC

DELTA ASDA A2-E

INOVANCE IS620N

1.2 System Connection

Fig. 4 shows the connection of doMotion control system with EtherCAT bus; the whole system consists of TPU, IPC, drive, digital IO module and robot body.

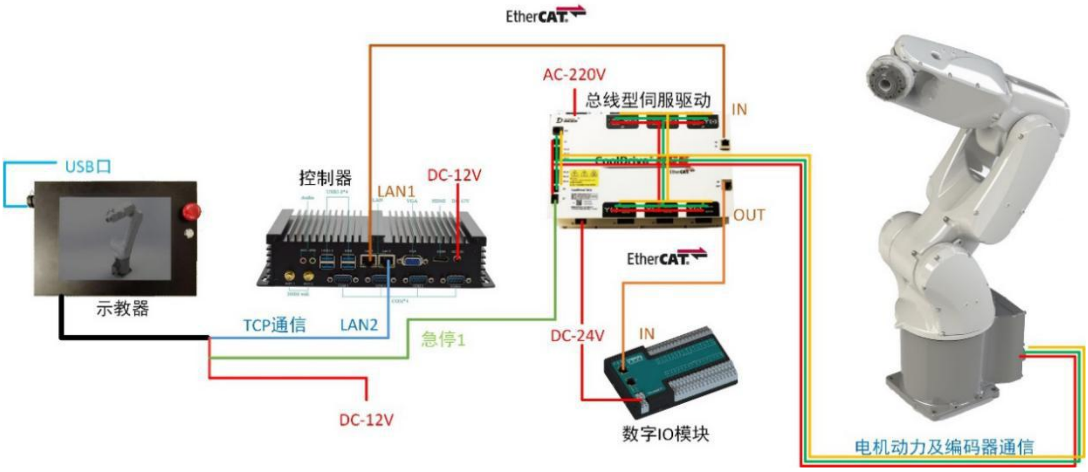

Fig. 4 EtherCAT BUS Robot Arm Control System

USB port is located at the upper left corner of the TPU and the outgoing cable at the bottom left corner includes three wires, the twisted pair cable (TCP communication) is connected to LAN2 of the controller, Emergency Stop 1 is connected to corresponding port of the drive, power cable is connected to DC-12V; the LAN1 port of IPC is connected to IN port of the drive (EtherCAT protocol); the OUT port of the driver is connected to IN port of the digital IO module by a network cable; please refer to Diver Manual for the power cable and encoder cable connection between the drive and the robot body. Cables and wires must be correctly connected.

Power supply to TPU and controller is DC-12V (not less than 5A), power supply to digital IO module and control voltage of driver are both DC-24V (not less than 5A), power supply to drive is AC- 220V (not less than 10A).

1.3 Pulse-version Scheme

The pulse-version control system is specially designed for various stepping and servo systems. Being equipped with industrial-grade software and DeX-Platform development platform, the pulse-version control system owns powerful functions and expandability.

Fig. 5 shows the design of pulse-version control system, which consists of Mocon motion control card, touch screen TPU and education-level main control board. Touch screen TPU and main control board are connected by Ethernet network cable supporting TCP protocol communication; the main control board and Mocon motion control card is connected by a USB-to-485-port cable supporting serial port communication.

Fig. 5 Education-version Control System

The pulse-version system is equipped with high performance FPGA motion control board Mocon, which support various types of stepping and servo motor drives and widely adapt to the robot bodies at different levels such as education level, desktop level and 3D printing level.

Table 4 Parameters of Mocon Motion Control Board

Overall size 220 × 170 × 30mm	9-axis stepping or digital servo control
Default communication cycle 15ms	16-way universal input switch signals

Output pulse frequency 500KHz	16-way universal output switch signals
Motor rotation direction optional	1-way brake control signal
Compatible with single-ended and differential signal drives	485 serial port communication
High performance FPGA chip	24V power input
Overcurrent protection	Output pulse 5V\24V optional
Single-axis two-way limit switch	Single axis enabling and error interruption

Table 5 Parameters of Touch Screen TPU

LCD	10.1" TFT 1280 x 800
Touch Screen	Capacitive Screen
Operating System	Linux (Support 3D simulation)
Virtual Keyboard	Supported
External USB	2.0 x 1
Power Module	5V， 10W

Table 6 Parameters of Main Control Board

Hardware Parameters	Dual-core 1.8GHz CPU, integrated GPU, 2GB RAM, 4GB user storage
Control Servo	Universal pulse type (stepper motor)
Control axis	6 axis
Operation Mode	Demonstration, reproduction, remote, support secondary development
Motion Function	joints, straight lines, computation, arcs, NURBS splines, complex curve continuous motion
Command	motion, logic, I/O, flow, process

System	
Coordinate System	joint coordinates, cartesian coordinates, user coordinates, tool coordinates, base coordinates
Adapted Model	Standard 6-axis, UR type
Power Module	DC 12V， 30W
Working Environment	Temperature 0°C~50°C (no freezing), humidity: 10%-90% (no condensation)

Chapter 2 Basic Settings and Operations

2.1 Start TPU

Press the power button on the TPU, the system will boot and automatically enter the interface shown in Fig. 6 below. Now, the system has not been fully operating and most buttons are in a gray and non-clickable state.

Click on the button at the upper left corner [Connect System]

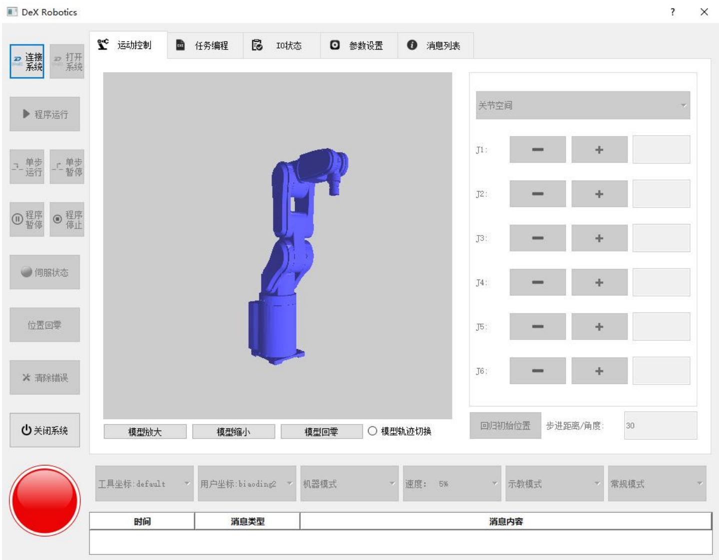

Fig. 6 Boot Interface

2.2 Robot Parameter Setting

Switch to the tab [Parameter Setting], shown in Fig. 7 as follows:

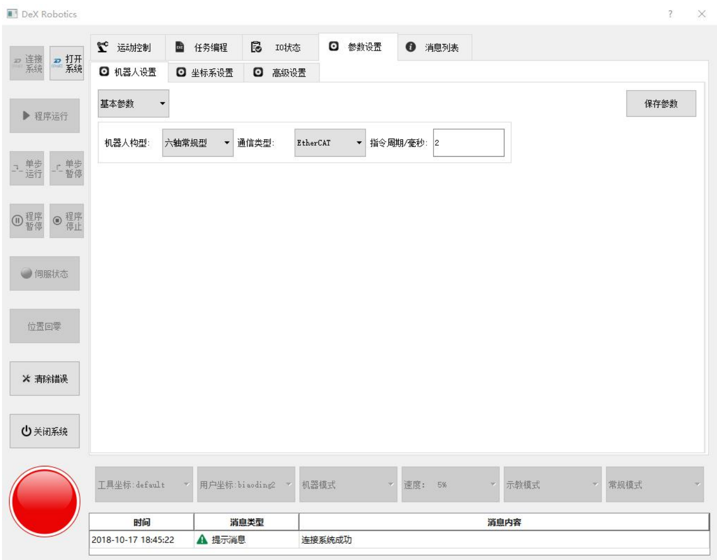

Fig. 7 Parameter Setting Interface

[Basic Parameter] includes setting options of [Robot Structure], [Communication Type] and [Command Period]. Notes: the settings aforesaid are default, please contact us for modification.

Select [Robot Setting] to find [Drive Parameter], [Model Parameter], [Motion Parameter] and [Ratio Parameter] in the drop-down option at the top left corner. Notes: when changing settings, click [Save Parameter] at the top right to save after input has been completed.

2.2.1 Drive Parameter

The interface of drive parameter setting is shown in Fig. 8.

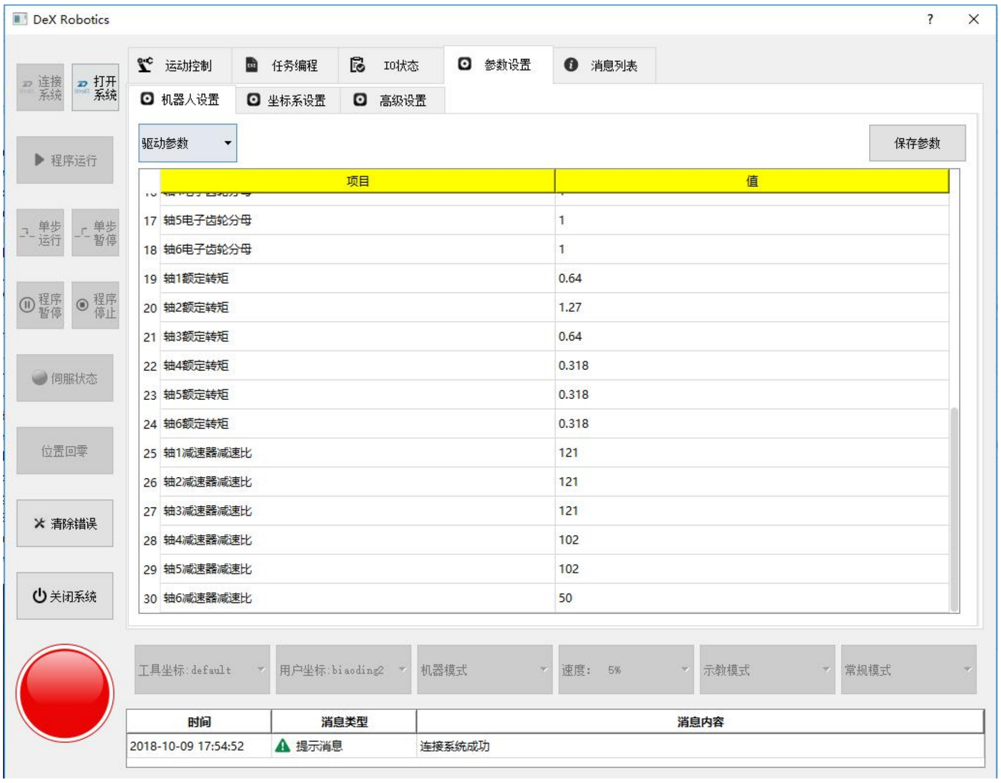

Fig. 8 Drive Parameter Setting

Drive parameters consist of the following parts:

1) **Axis encoder resolution:** Fill in accordance with parameter settings of axis motor manual.

2) **Axis electronic gear numerator/denominator:** Fill in accordance

with parameter settings of motor and drive manual.

3) **Rated axis torque：** Fill in accordance with parameter settings of axis motor manual.

4) **Axis reduction ratio：** Fill in accordance with actual settings of robot.

2.2.2 Model Parameter

Model parameter is referred to as DH parameter of industrial robot. The setting interface is shown in Fig. 9. “alpha”, “a”, “d” and “theta” are four kinematic parameters to be set for each connecting rod.

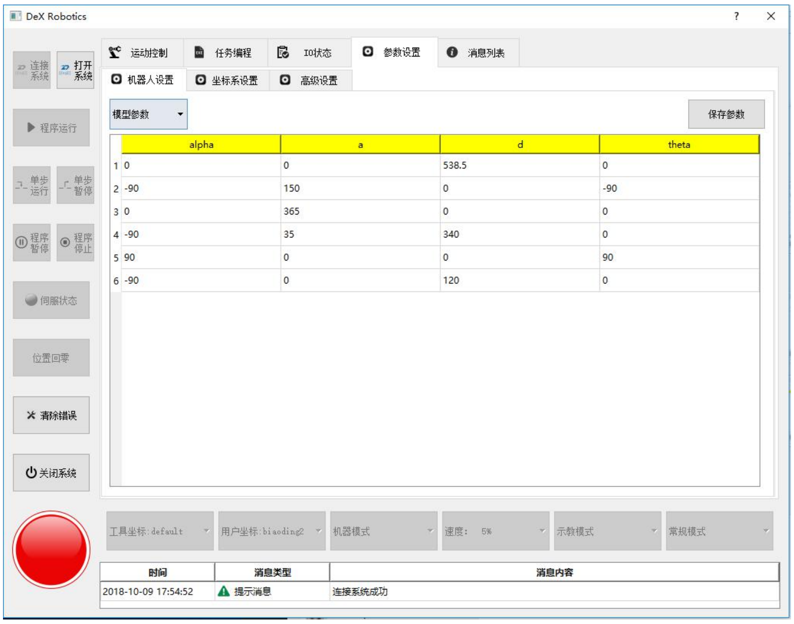

Fig. 9　Model Parameter Setting

doMotion series control system is integrate with two types of 6-axis industrial robot structures: conventional type and collaborative type. The conventional type robot structure is shown in Fig. 10, which refers to the zero position of conventional robot.

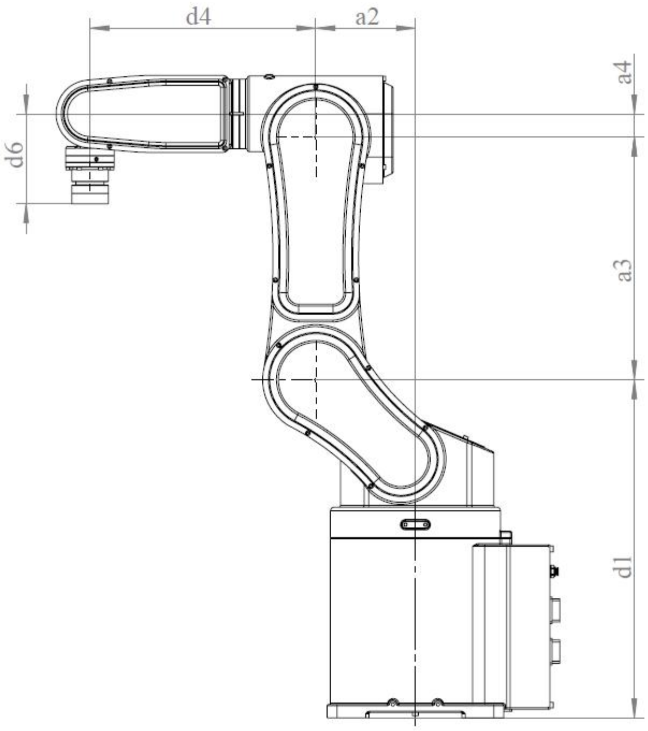

Fig. 10 Structure of Conventional Robot

The DH parameters of conventional robot are shown in Fig. 7 as follows. Normally, the values of alpha and theta are non-changeable; all the zero values in a and d are also non-changeable. The non-zero values should be filled according to the actual robot sizes shown in Fig. 10.

Table 7 DH Parameters of Conventional Robot

SN	α_i (°)	a_i (mm)	d_i (mm)	θ_i (°)
1	0	0	d_1	0
2	-90	a_2	0	-90
3	0	a_3	d_3	0
4	-90	a_4	d_4	0
5	90	0	0	90
6	-90	0	d_6	0

Notes: d3 of most conventional industrial robots is set as 0. If offset occurs, fill in according to actual structural parameters.

The structure of collaborative robot is shown in Fig. 11 as follows, which refers to the zero position of collaborative robot.

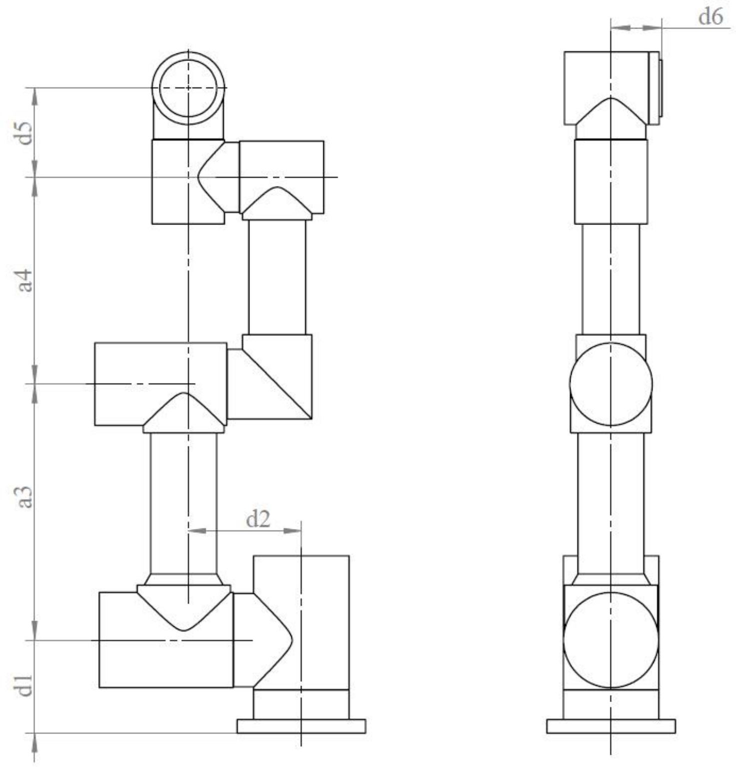

Fig. 11 Structure of Collaborative Robot

The DH parameters of collaborative robot are shown in Fig. 8 as follows. Similarly, the non-zero values in a and d should be filled according to the actual robot sizes shown in Fig. 11 and the other values should not be changed.

Table 8 DH Parameters of Collaborative Robot

SN	α_i (°)	a_i (mm)	d_i (mm)	θ_i (°)
1	0	0	d_1	0
2	-90	0	$-d_2$	-90
3	0	a_3	0	0
4	0	a_4	0	90
5	90	0	d_5	90
6	90	0	d_6	0

While entering model parameters, positive and negative rotations of each axis should be correct. **Rotation direction setting usually needs to be done on motor drive.** The positive rotation directions of conventional robot are shown in Fig. 12.

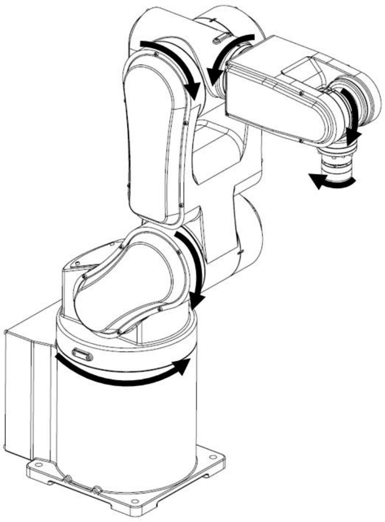

Fig. 12 Positive Joint Rotation of Conventional Robot

The positive rotation directions of collaborative robot are shown in Fig. 13.

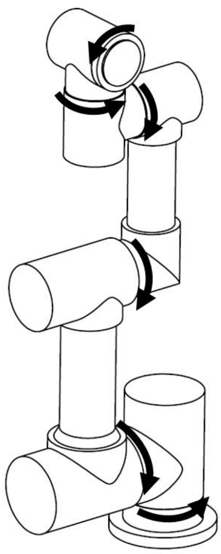

Fig. 13 Positive Joint Rotation of Collaborative Robot

2.2.3 Motion Parameter

The interface of motion parameter is shown in Fig. 14. These settings are related to the robot's motion efficiency, performance and service life. Excessive values may result in short service life and poor motion performance. The values should be carefully set according to specific workload, rigidity of robot body and the performance of reducer and servo motor.

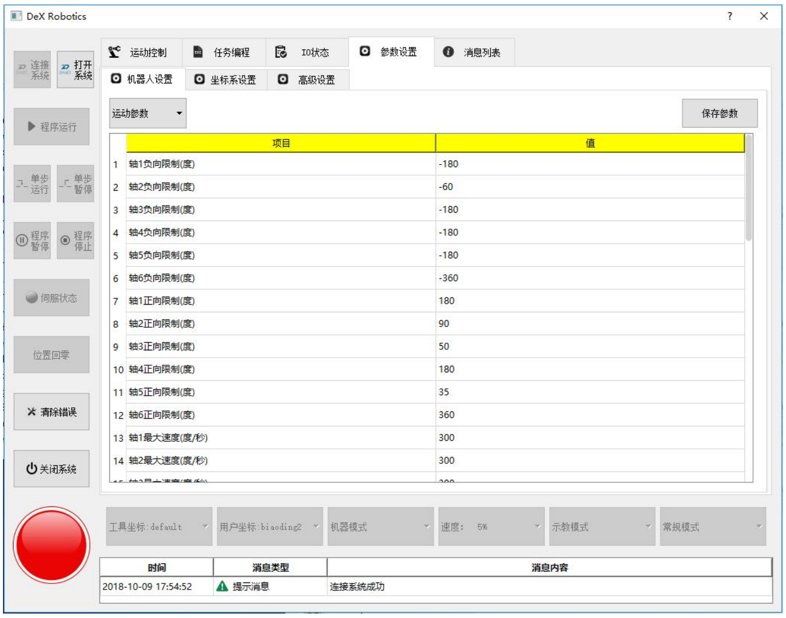

Fig. 14 Motion Parameter Setting

Motion parameters consist of the following parts:

1) **Axis positive/negative angle limit:** Set positive and negative soft limits of each joint according to actual robot structures and sizes.

2) **Axis maximum speed / acceleration / impact limit:** Carefully set the maximum speed, acceleration and impact limit of every joint based on robot performance an specific workload. Notes: setting too low values will make robot incapable of high speed motion; while setting too high values will reduce robot's service life and even cause damages on robot body.

3) **Cartesian space positive/negative limit:** Set Cartesian space X\Y\Z range limit according to robot arm length and space limitation.

4) **Cartesian space velocity / acceleration / impact limit:** Cartesian space motion consists of three parts: "End Displacement", "Select Along Z Axis" and "Rotate Along X Axis" (see Chapter 4 for the geometric meaning). Similar to joint space setting, the performance limit value of Cartesian space motion needs to be carefully set according to specific robot conditions.

2.2.4 Ratio Parameter

The interface of ratio parameter is shown in Fig. 15 as follows. The ratio parameter can further adjust the speed, acceleration and impact of robot joint space and end Cartesian space motion; the input range is 0-1, which means that based on the values of [Motion Parameter] "axis maximum speed / acceleration / impact limit" and "Cartesian space speed / acceleration / impact limit", it's multiplied by the scale factor as the current maximum performance limit; for trajectory planning, the values multiplied with the scale factor will be used as performance limit for calculation.

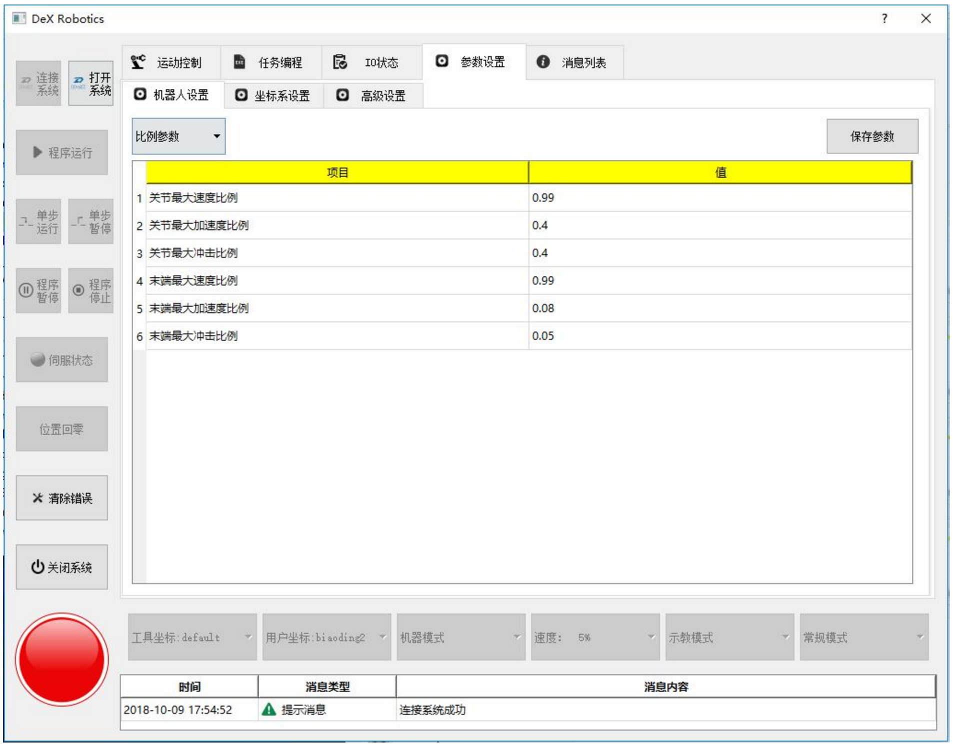

Fig. 15 Performance Parameter Setting

2.3 Start System

After setting is completed, click on the button [Start System] on the top left corner as shown in Fig. 6 to wait for system initialization; after the initialization is completed, the information prompted at the bottom shows “Successfully Start System” and the whole interface is illuminated as shown in

Fig. 16.

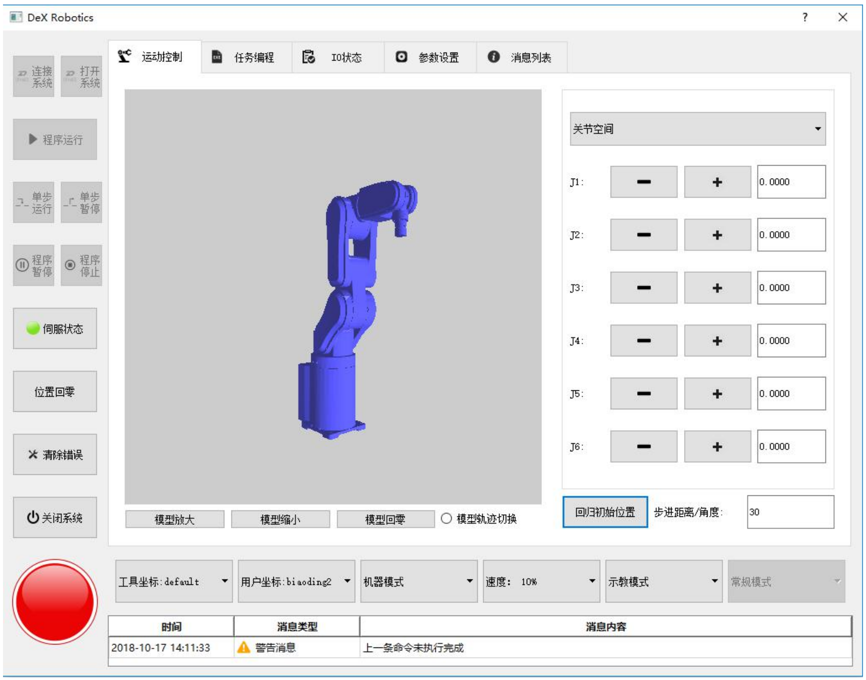

Fig. 16 Interface of Starting System Successfully

After system initialization, robot motion cannot be directly controlled yet. Click on the grey button [Servo Status] on the left as shown in Fig. 16, the indicator icon turns green, which means the servo is successfully activated. Then start normal operation.

2.4 Robot Zero Position Setting

Robot zero position needs to be set before officially starting system start in the tab [Advanced Setting] in [Parameter Setting] as shown in Fig. 17, which includes two types: manual zero position and automatic zero position.

2.4.1 Manual Zero Position

Set zero position as per following steps:

1) Switch to [Motion Control] for demonstration (see Chapter 3 for details) and adjust the robot to the zero-position shown in Fig. 10 or Fig. 11;
2) Inactivate [Servo Status];

3) Switch back to [Advanced Info Setting] under [Parameter Setting], click on the button [Zero Position] for partial or all axis;

4) Prompt shows successful zero position.

Fig. 17 Interface of Zero Position

After successful zero position, switch back to [Motion Control] to observe whether 3D robot model has returned to zero position.

2.4.2 Automatic Zero Position

Proximity switch needs to be installed on axis of robot arm and connected to system IO to realize the function of automatic zero position. The axis stops when rotating to trigger the proximity switch and automatically returns to zero position. In [Advanced Setting], fill in the axis sequence of zero position in "Zero Position Setting" as well as automatic rotation speed; fill in the angle values of proximity switches on axis to the zero position in "Reset Current Position".

After setting is completed, click on "Zero Position" button on the right column, then the robot arm will start to return to zero position in sequence and stop when the proximity switch is triggered. When all the 6 axes have stopped, the current joint angle is the value to be filled in "Reset Current Position". If the value is accurate, they will return to their respective zero positions on the axes.

2.5 Shut Down System

Before shutting down system, please click on [Servo Enable] button to inactive the function and confirm. Then, click on "Shut Down System" and confirm whether it's power off.

Chapter 3 Basic Demonstration Method

3.1 Inching

After starting system and activating Servo Enable, demon-programming operation can be conducted. Demonstration is conducted on the tab “Motion Control”.

Inching is the most commonly used demon method, which includes joint inching and end inching.

3.1.1 Joint Space Inching

On the tab [Motion Control], the left side shows robot 3D model and the right side shows demo buttons (- means negative, + means positive) shown as in Fig. 18. Select “Joint Space” in the drop-down box above the demon button, which means joint space inching function has been selected. Then, click the “-“ or “+” buttons corresponding to different axes to conduct inching operation on all the axes of the robot: press button to move while release to stop.

Select “Speed” percentage on the bottom to adjust the inching speed.

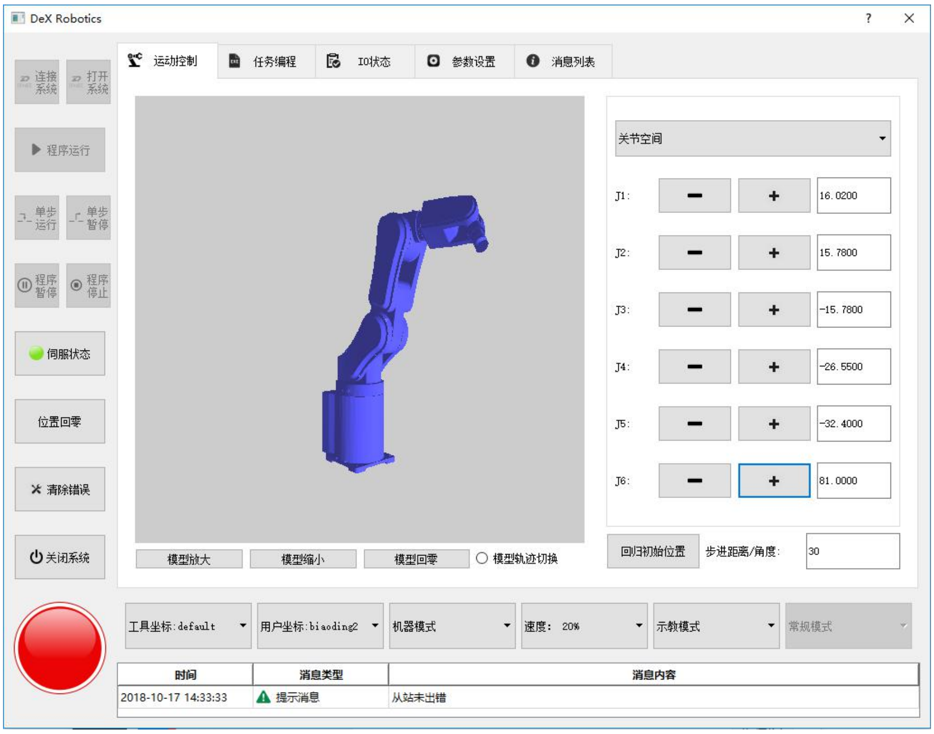

Fig. 18 Joint Inching

3.1.2 End Cartesian Right-angle Space Inching

Select “Right Angle” in the drop-down box above the demo button, which means the inching function of the end Cartesian coordinate system has been selected shown as in Fig. 19. Then, click the “-“ or “+” buttons corresponding to different axes to conduct inching operation on 6 component values of Cartesian coordinate system: X direction, Y direction, Z direction, A angle, B angle and C angle; similarly, the “Speed” percentage on the bottom can be selected to adjust the speed.

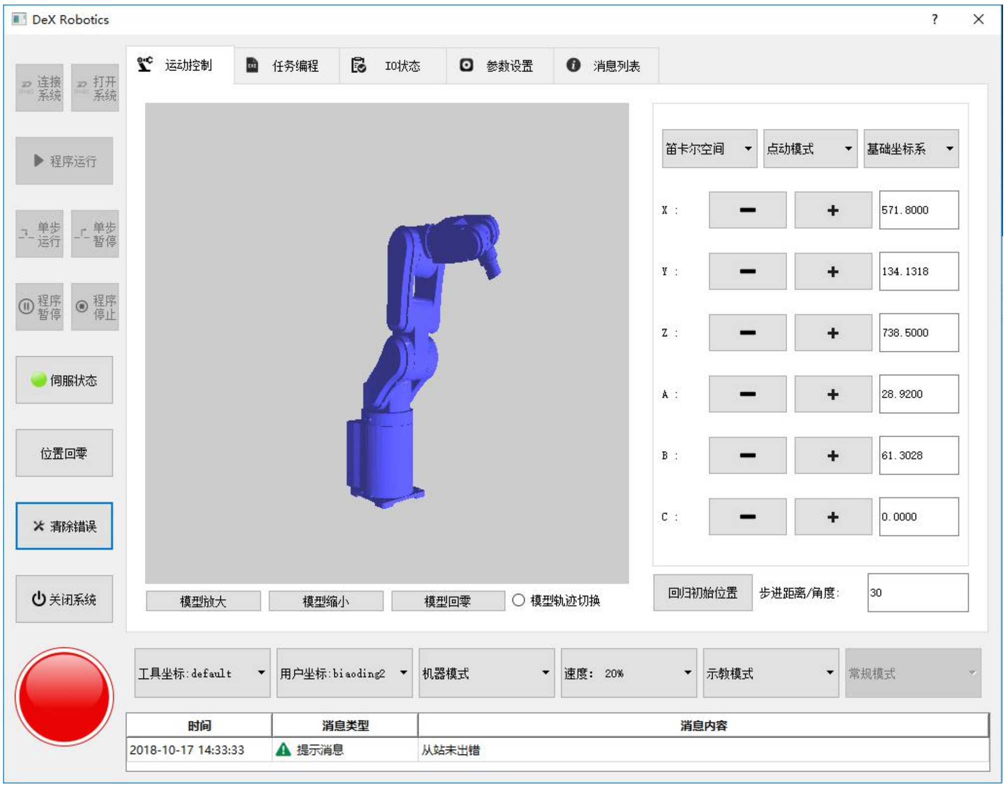

Fig. 19 Inching in Cartesian Right-Angle Space

Compared with joint space, Cartesian right-angle space has the option of selecting coordinate system, which means Cartesian space motion in different coordinate systems can be realized (see Chapter 4 for details related to coordinate system). In the drop-down box under “Inching” button, there are three reference coordinate systems, “Base Coordinate System", "User" and "End":

1) **Base Coordinate System** refers to the default coordinate system of robot, the position is based on robot’s DH parameter modeling;

2) **User Coordinate System** refers to the coordinate system which be calibrated and used by users. Users can add or calibrate their own coordination systems on the tab “Coordination System Setting” under “Parameter Setting”； and select the coordination system for current operation in the drop-down box on the tab “User Coordinates” at the bottom left of “Motion Control”;

3) **End Coordinate System** refers to the coordinate system of robot's current end tooling.

3.2 Stepping

"Stepping" demo can be selected in the mode of Cartesian coordinate system: on the tab "Stepping Distance/Angle" below the demon button, the stepping distance can be set (when stepping X, Y, Z, the unit is mm; when stepping A, B, C, the unit is degree). Then, click on demon "**-**" or "**+**" buttons to make robot end to step as per preset distance or angles in different direction. Similarly, the stepping speed can also be adjusted by selecting "speed" percentage.

3.3 Other Demon Operations

On the tab "Motion Control", press the button "Back to Initial Position" below the demon button to make robot return from current position to zero position and the speed can be adjusted by selecting "speed" percentage.

Notes:

This function demands extreme caution. Make sure robot will not be damaged due to any obstacles when moving from current position to zero position.

The 3D simulation model on the interface can display robot motion in real time. The options [Model Zoom In], [Model Zoom Out], [Model to Zero] and [Model / Trajectory Switch] are listed below. Meanwhile, the perspective can be changed by sliding the model interface and [Model to Zero] can make 3D model perspective return to the initial position. [Model / Trajectory Switch] can select robot modal or end motion trajectory to be displayed in 3D view.

Chapter 4 Coordinate System Introduction

4.1 Robot Coordinate System

The coordinate system is a very important concept in robot control system, because it affects most of the demon operations and motion trajectory. The doMotion series control system mainly includes three types of coordinate systems: base coordinate system, tool coordinate system and user coordinate system as shown in Fig. 20.

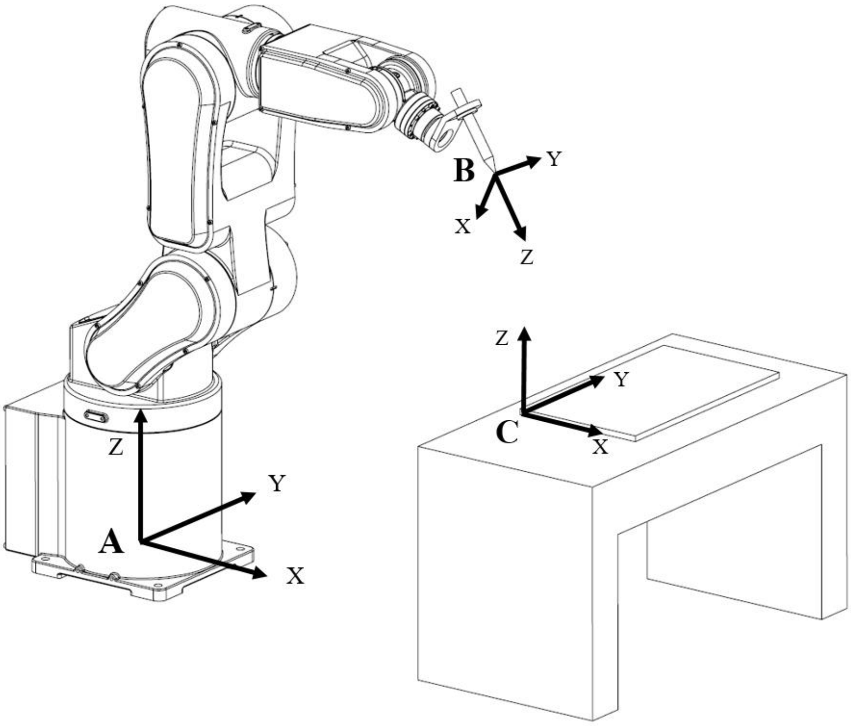

Fig. 20 Robot Coordinate System

A. **Base Coordinate System.** The robot base coordinate system is set when building robot DH model, usually at the base. For simple applications, programming can be conducted within robot base coordinate system.

B. **Tool Coordinate System.** The tools on end flange need tool coordinate system to define its tool center point (TCP). When changing tools, only redefine the tool coordinate system to complete original tasks with the new tools. The doMotion control system describes the tool coordinate system with relate to the flange coordinate system.

C. **User Coordinate System.** The robot may need to work at multiple positions on the workbench, the user coordinate system can be defined in this case. When changing workpiece positions, only need to change the corresponding user coordinate system. The doMotion control system describes the user coordinate system with relate to the base coordinate system.

4.2 Coordinate System Description Method

Object in 3D Cartesian space has six free dimensions, therefore at least six independent quantities are needed to describe a spatial coordinate system. Among them, X, Y, Z are position coordinates of coordinate system origin, which is very simple and intuitive; however, it’s a little difficult to describe the spatial positions (A, B, C angles).

As for the doMotion series controller, the description and setting of coordinate system posture are slightly different from that of the conventional controller. The conventional controllers normally use Euler angles or quaternions to describe the spatial posture, while this controller adopts the latitude and longitude angles. Compared with the Euler angle, the latitude and longitude angles can be used to describe the posture for better demon effect without such shortcomings as mutual coupling between Euler angles or locking of universal joint; compared with quaternions, the latitude and longitude angles are more intuitive and coordinate system values can be directed input.

There are two parts to describe the coordinate system posture with latitude and longitude angles: Z-axis direction and the rotation part (i.e. the X-axis direction) around Z-axis. Two parts are mutually decoupled. When describing the Z-axis direction, it can be considered that the vertex of Z-axis moves on the unit spherical surface centered on the origin of the coordinate system. Therefore, the direction of Z-axis can be uniquely determined by the latitude and longitude angles of the coordinate system. When the Z-axis direction is determined and it rotates around itself, the X-axis direction can be determined, so that arbitrary postures in the space can be described in the coordinate system.

Like the Euler angle, three angles are also needed to describe postures by the latitude and longitude angles, e.g. A, B, and C in the controller, where A refers to latitude angle, B refers to longitude angle and C refers to self-rotation angle. Once one point on the unit sphere is determined with the longitude and latitude angles, the vector from the spherical center determines the Z axis of the coordinate system; once the Z axis is determined, rotate the coordinate system around Z axis to determine the directions of the other two axes. The angle to be rotated is described by the C angle (self-rotation angle).

According to the unit homogeneous matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The base coordinate system can be determined. Take the origin of base coordinate system as the spherical center, the vertex of Z axis as the north pole, make the 0° longitude passes the vertex of X axis to establish a unit sphere, as shown in Fig. 21.

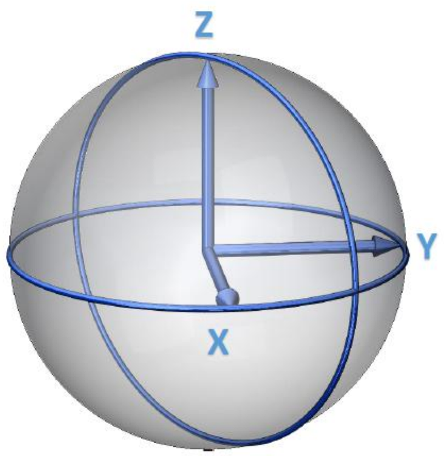

Fig. 21 Unit Sphere of Base Coordinate System

The definitions of A, B and C angles are shown as follows:

Latitude angle A: The latitude angle, by definition, refers to the angle at which the Z axis of the posture coordinate system moves along the same latitude, resulting in latitude changes. Different from the definition of Earth's latitude, the controller's 0° latitude is not from the equator but from the south pole of the unit sphere and it's 180° to the North Pole, so the range is 0° to 180°, as shown in Fig. 22.

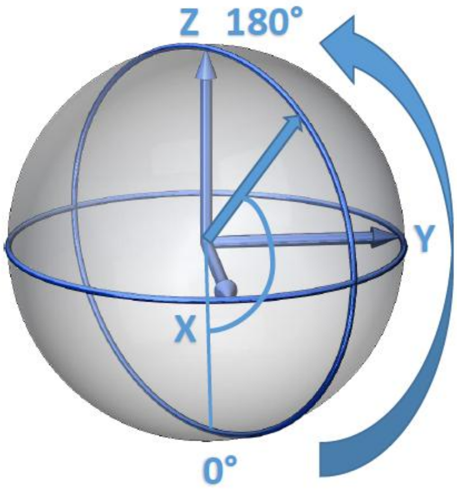

Fig. 22 Definition of Latitude Angle

Longitude Angle B: The longitude angle refers to the angle at which the Z axis of the posture coordinate system moves along the same latitude, resulting the longitude changes. By definition, the 0° longitude line is referred to as the line passing through the positive vertex of the X axis in base coordinate system. Like the globe, the east longitude 180° line coincides with the west longitude 180° line and the range is -180° (west longitude 180° line) to 180° (east longitude 180° line), as shown in Fig. 23.

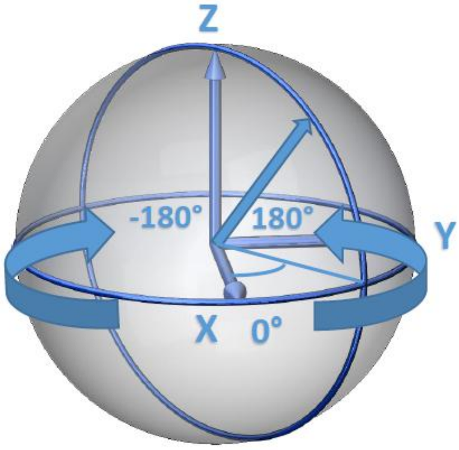

Fig. 23 Definition of Longitude Angle

Rotation Angle C： Once the latitude and longitude are determined, the direction of Z axis is also determined, then there's only rotation angle of Z axis itself. This angle can be defined by the angle with the X axis in the coordinate system; wherein, the 0° position of X axis is defined as the position of X-axis in base coordinate system when the Z axis of base coordinate system is rotated to coincide with the Z axis of current coordinate system. The angle of rotation angle is between -180° to 180°, as shown in Fig. 24.

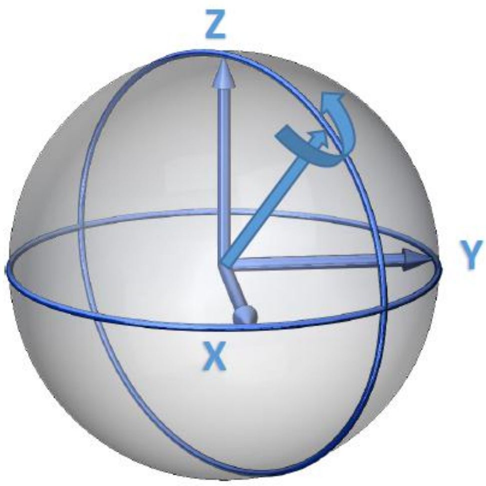

Fig. 24 Definition of Rotation Angle

According to definition of latitude and longitude angles, there are two singularities based on posture description by this method: South Pole and North Pole. At the two points, the latitudes are 0° and 180° respectively, but the longitude cannot be defined. The controller specifies that when the latitude is 0° and 180°, the longitude is 0°.

According to the definition, the posture of base coordinate system is described as **(180°, 0°, 180°)** with latitude and longitude angles.

4.3 Setting Coordinate System

Switch to the page [Parameter Setting] and select the tab [Coordinate System Setting], as shown in Fig. 25.

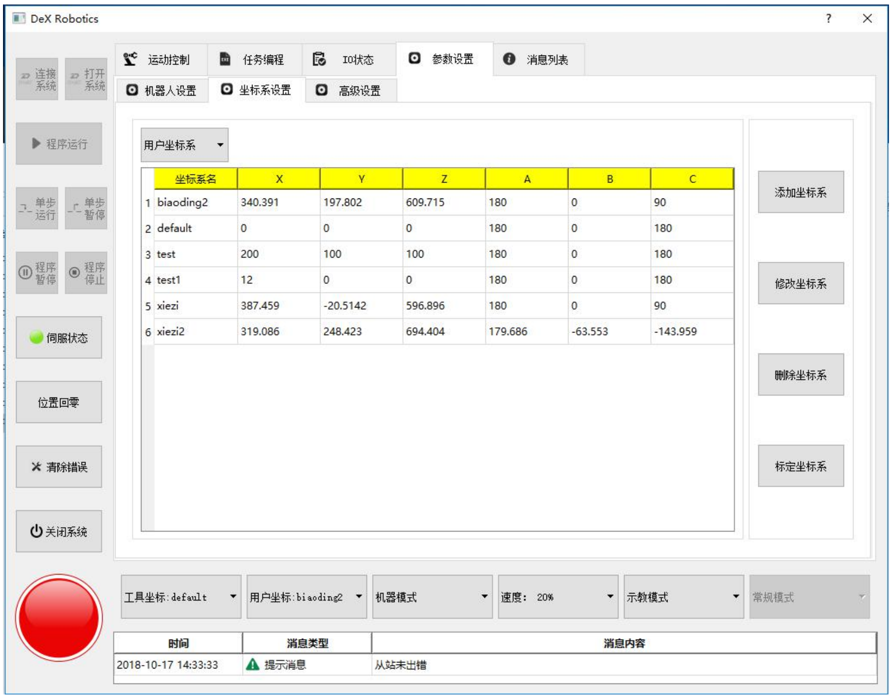

Fig. 25 Coordinate System Setting

Set and add coordinate systems according to the following steps:

1) Select [User Coordinate System] or [Tool Coordinate System] in the drop-down box at the top left corner;

2) Click on [Add Coordinate System] button on the right, input “coordinate system name” (to better distinguish categories, it’s recommended to add prefix “user” in front of user coordinate system, and prefix “tool” in front of tool coordinate system);

3) Enter values in the input boxes below in the order of X (mm), Y (mm), Z (mm), A (°), B (°), C (°); Note: to create a base coordinate system, just input (**0,0,0,180,0,180**).

4) Click on [OK] button to complete adding coordinate system.

Select a coordinate system, click on [Modify Coordinate System] button to modify the values; click on [Delete Coordinate System] button and confirm to delete the coordinate system.

In the drop-down boxes under [User Coordinate System] and [Tool Coordinate System] at the bottom left corn, the coordinate systems can be selected and added by names; the coordinate systems selected here will be reflected in the demo and point collecting process, as well as the setting of [Coordinate System Switching] command in programming process. Please refer to Chapter 6 for details.

4.4 Calibration of Coordinate System

4.4.1 Calibration of User Coordinate System

Select [User Coordinate System] on the tab [Coordinate System Setting], click on [Calibrate Coordinate System] button on the right, switch to the page shown in Fig. 26.

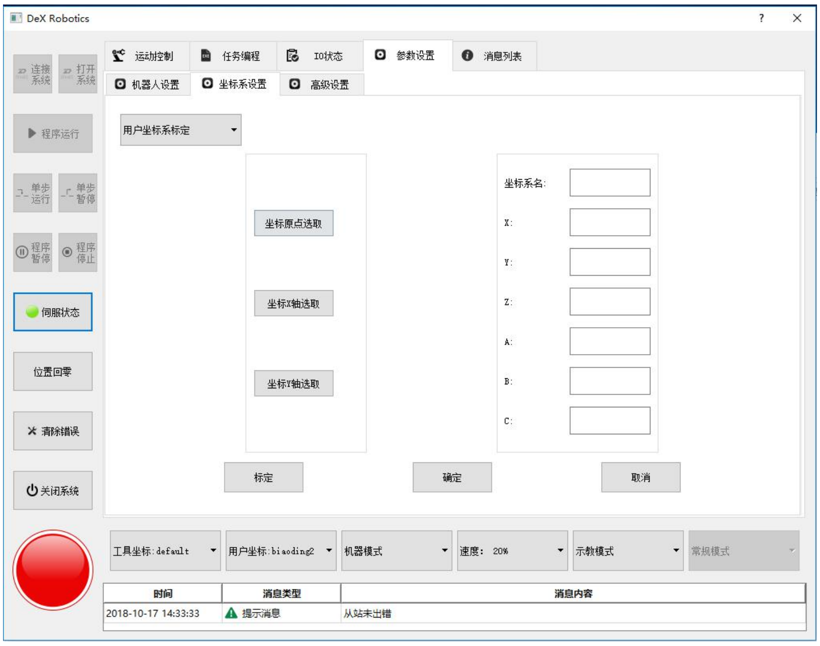

Fig. 26 Calibration of User Coordinate System

Conduct calibration according to following steps:

1) Fill in “name of coordinate system”;
2) Switch to the page [Motion Control], move robot end to the origin of the coordinate system to be calibrated, return to the page [Parameter Setting], click on [Select Coordinate Origin] button, the words on the button change into [√ Coordinate Origin Selected];
3) Move demon end to the point on X axis of the coordinate system to be calibrated, click on [Select X Axis];
4) Move demo end to the point on Y axis of the coordinate system to be calibrated, click on [Select Y Axis];
5) Click on the button [Calibration] below to show calibration result;
6) Check calibration result, click on [OK] button, return to the page [User Coordinate System].

4.4.2 Calibration of Tool Coordinate System

The method of "fixed point to change posture" is generally adopted for calibration of tool coordinate system. The center point of the tool coordinate system to be calibrated is pointed to a fixed point in various postures, as shown in Fig. 27.

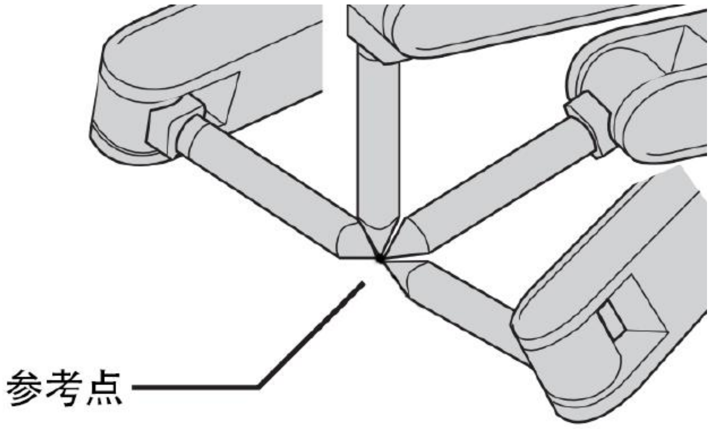

Fig. 27 Calibration Method of Tool Coordinate System

Select [Tool Coordinate System] in [Coordinate System Setting], click on the button [Calibrate Coordinate System] on the right side, switch to the page shown as in Fig. 28. In the columns on the left side, there are three selection: [TCP Calibration], [TCP + Z Calibration] and [TCP +Z/X Calibration].

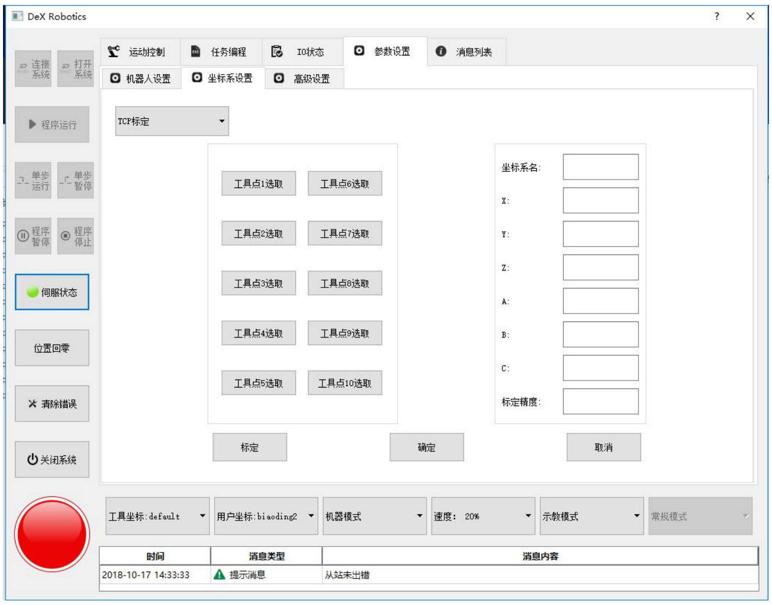

Fig. 28 Calibration of Tool Coordinate System

4.4.2.1 TCP Calibration

In the mode of [TCP Calibration], only the central point of the tool coordinate system will be calibrated, the axes of coordinate system are not included. Follow the following steps to conduct the calibration:

1) Fill in "Name of Coordinate System";
2) Switch to the page [Motion Control], move the central point of robot's tool coordinate system to point at a fixed reference point, return to [Parameter Setting] and click on [Tool Selection Site] button;
3) Repeat the aforesaid step 10 times to fill in all the 10 columns. Notes: The more difference for posture pointing, the more accurate for the result;
4) Click on [calibrate] button below to show calibration result;
5) Check the calibration result, click on [OK] button, return to [Tool Coordinate System].

4.4.2.2 TCP+Z Calibration

In [TCP+Z Calibration] mode, the central point and Z axis of the tool coordinate system can be calibrated. The basic calibration method and sequence are the same as that in [TCP Calibration]. The only difference is that after calibration of 10 base points, it is necessary to move to the Z-axis vector origin and a point on the positive Z-axis in the tool coordinate system to determine the Z-axis direction as shown in Fig. 29.

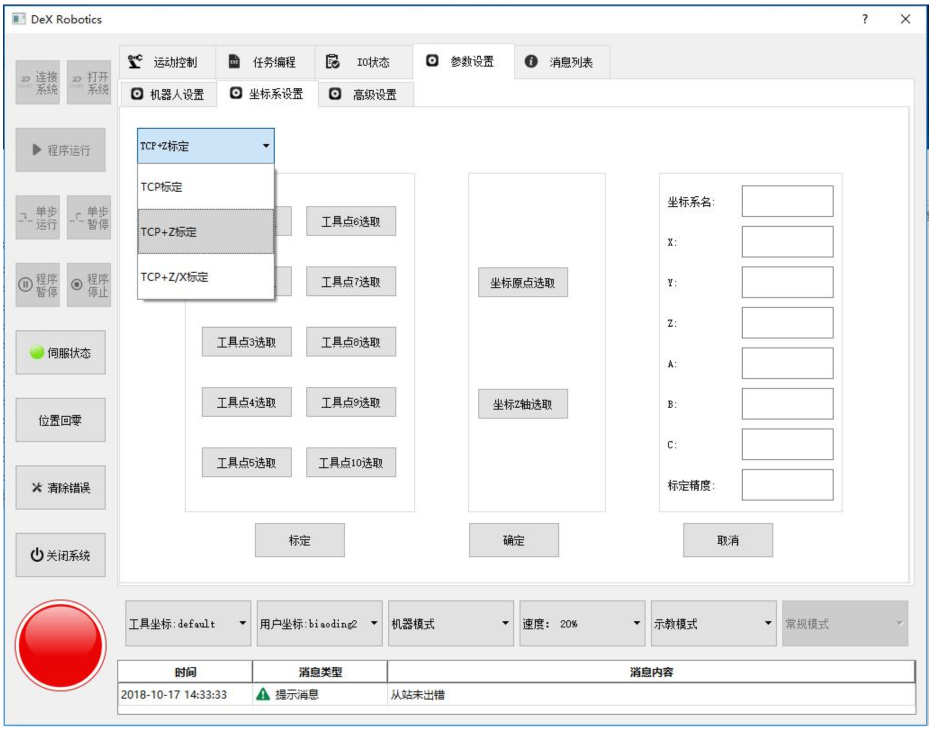

Fig. 29 TCP+Z Calibration Mode

4.4.2.3 TCP+Z/X Calibration

In [TCP+Z/X Calibration] mode, the central point, Z axis and X axis of the tool coordinate system can be calibrated. The basic calibration steps are the same with that of "TCP+Z Calibration" mode. The only difference is that after calibration of 10 base points, it is necessary to move to vector origin, a point on the positive Z-axis and a point on the positive X-axis in the tool coordinate system to determine the direction of Z-axis and X-axis as shown in Fig. 30.

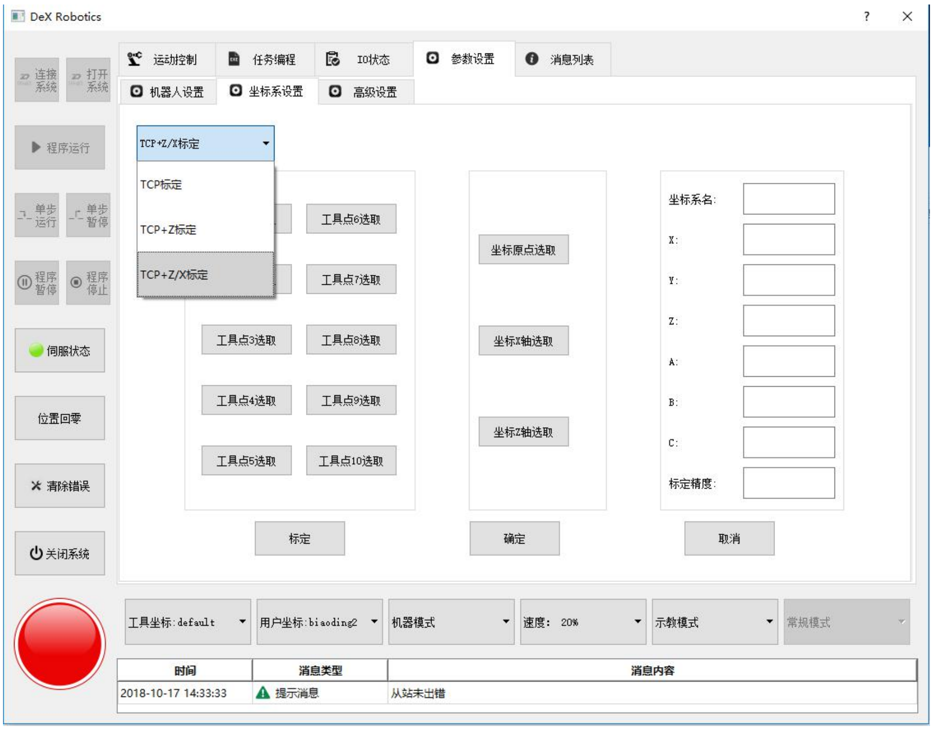

Fig. 30 TCP+Z/X Calibration Mode

Chapter 5 Program Debugging and Operation

5.1 Programming Overview

A complete programming for industrial robot usually includes the following parts:

1) **Trajectory setting**. When the robot moves from the previous position to the next position, many types of path trajectories are involved during the motion, the most common ones include straight lines, arcs, joint motion and splines. When programming, it is necessary to splice multiple basic trajectories as per the mission sequence to form a complete robot motion. Please refer to Chapter 6 for the method of trajectory edition.
2) **Logical instruction**. Logic instructions can realize complicated mission programs that robot can perform more complex tasks. The most common logic instructions include loops, conditional loops, decisions, etc. Please refer to Chapter 7 for application of logic instructions.
3) **IO Control.** Robot usually needs to cooperate with other equipment to complete mission. The IO port of the control system can be used to communicate with the device. It can send control signals to other devices while receive their feedback signals. The most common equipment includes clamps, suction cups, nozzles, welding torches, etc. Please refer to Chapter 8 for the method of IO control.

The program writing and debugging can be conducted in [Task Programming] as shown in Fig. 31.

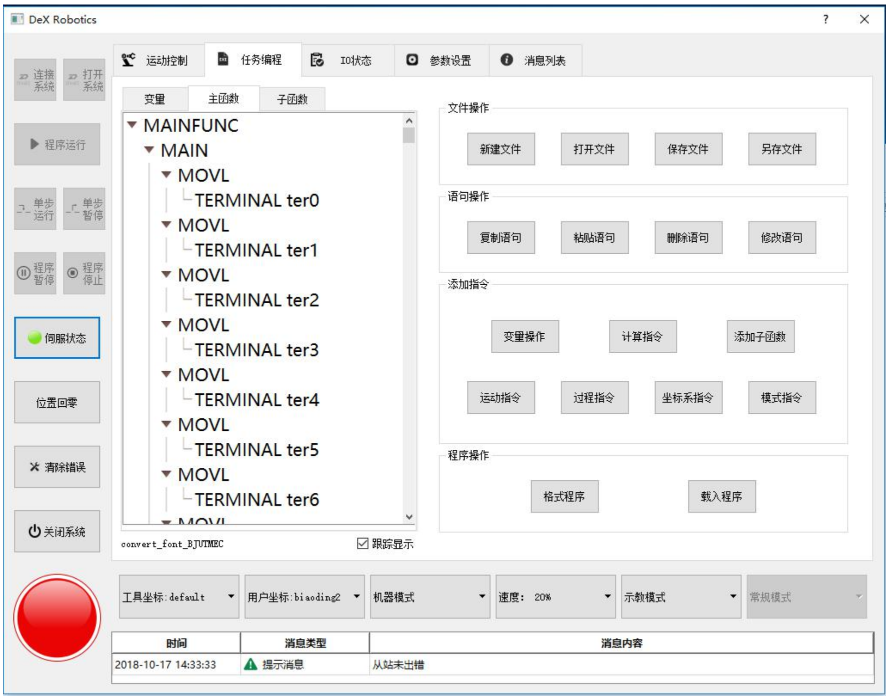

Fig. 31 Programming Interface

On the left side there are three tabs: "Variables", "Main Functions" and “Sub Functions". In the tab “Variables”, all current types of variables can be edited. In the tabs "Main Functions" and “Sub Functions", the programming statements can be edited.

5.2 File Operation

In [Mission Programming], there is a part named “File Operation”, which includes:

1) [Create New File]: create a new programming item;

2) [Open File]: open an existing programming item;

3) [Save File]: save the programming item being edited;

4) [Save As]: save the programming item being edited as a new file.

5.3 Statement Operation

The common operations on function statement code are as follows:

[Copy/Paste Statement]： To copy and paste a line of code, select this line and click the [**Copy Statement**] button, select a new line of code, click the [**Paste Statement**] button, then the content to be copied will be insert below the new line of code. Notes: the variables cannot be copied and pasted, the code on the [**Function**] tab cannot be copied and pasted to the [**Variable**] tab; the code can be copied and pasted between the [**Main Function]** and [**Sub Function**];

[Delete Statement]： Select a line of code or variables, click on [**Delete Statement**] and confirm for deletion. Notes: Cautious to delete "variable" type. If this variable is being used in [**Function**], syntax error will occur when loading.

[Modify Statement]: Select a line of statements, click on [**Modify Statement**] button to switch to corresponding modification interface for modification.

Application methods of motion trajectories, variables, logic instruction will be introduced in subsequent chapters.

5.4 Working Mode

doMotion series control system has two basic working modes: [Demo Mode] and [Representation Mode], which can be selected in the 5th drop-down box in the TPU interface.

5.4.1 Demon Mode

In demon mode, most of the demon functions such as robot inching and stepping, file operation and programming operation can be conducted. In this mode, user edits the program so as to enter [Representation Mode] for continuous operation.

5.4.2 Representation Mode

Representation mode enables robot and 3D model to execute programming code.

When user edits program in [**Demon Mode**], click [**Save File**] button in "File Operation"; then click on [**Load Program**] button in "Program Operation". If no syntax problem, "Load Successfully" will be prompted in the information bar below. Click on the drop-down box in "Demon Mode", select [**Representation Mode**], then most buttons in the interface will turn grey. [**Servo Enable**] button will be automatically activated. No demon edition operation is allowed at this moment.

In the 3rd drop-down box at the bottom, [**Machine Mode**] or [**Virtual Mode**] are optional. If [**Machine Mode**] is selected, all the motions in [**Representation Mode**] are the actual robot motions; if [**Virtual Mode**] is selected, it's the operation of 3D model. To ensure the security, **it is strongly recommended that before entering [Representation Mode] to run program, first select [Virtual Mode] to observe the motion of 3D model, confirm the programming command and the robot motions are consistent with expectations, then switch to [Machine Mode] for actual motions.**

In the last drop-down box, [**Debug Mode**] or [**Conventional Mode**] are optional.

In [**Debug Mode**], user can conduct debug the program by steps to check if it conforms to the expected motions; enable [Program Operation], [Step Operation] and [Step Pause] on the leftmost side, click on [Program Operation] button and the robot will execute the program according to the codes, but the motion segments are not continuous; [Step Operation] button enables the robot to execute motion trajectory by steps; [Step Pause] enables the robot to halt after executing current motion trajectory;

After motions confirmed, switch to [**Conventional Mode**], enable [Program Motion] button on the leftmost side, then the robot will continuously execute the program according to the codes. User can click on [Program Pause/Continue] and [Program Stop] below to conduct corresponding operations.

In [Representation Mode], the overall operation speed can be adjusted by selecting the “Speed” percentage at the bottom. Meanwhile, below the program box on the left, “Track Display” can be selected to make the cursor jump to the currently running code segment in real time.

Notes: when robot is moving in [Representation Mode], be sure to keep a safe distance!

When switching from [Representation Mode] to [Demon Mode], [Servo Enable] button will be automatically inactivated, then the demon programming can be conducted again.

5.5 Variable

Variable is an important concept of programming. In doMotion series control system there are several types of variables such as TERMINAL, JOINT, FRAME, INT, DOUBLE and BOOL which support variable calculation. The variable so TERMINAL and JOINT can be added into MOVE motion command, the variable of FRAME type can be used for “coordinate system switching”. The variables of INT, DOUBLE and BOOL can be used for logic command.

When adding variables, select [Variable Operation] button in “Add Command” under [Mission Programming] as shown in Fig. 32:

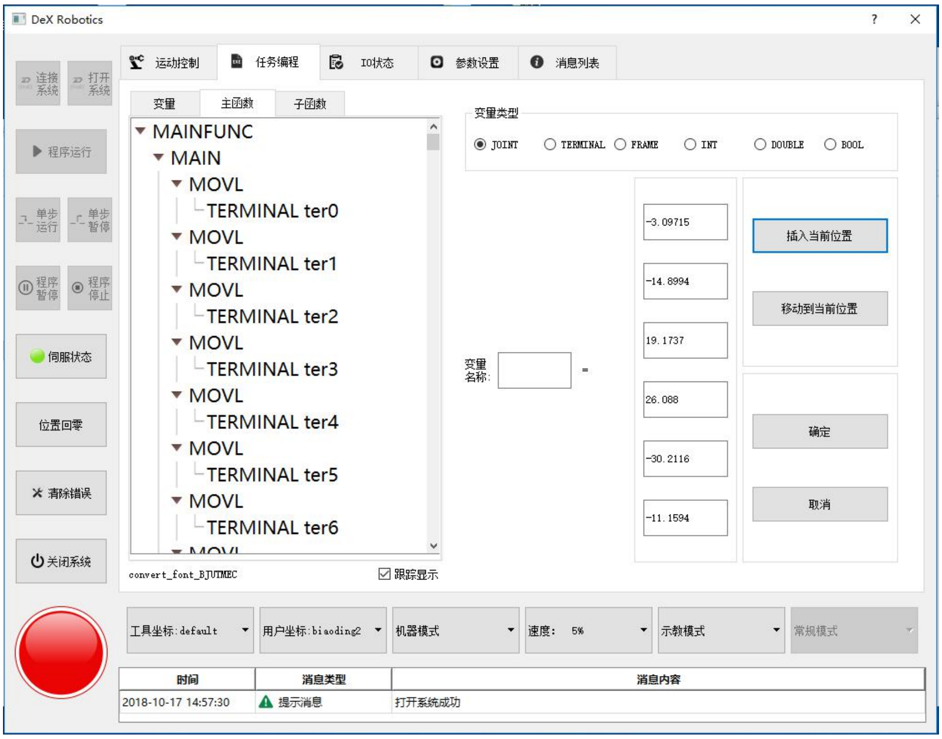

Fig. 32 Interface of Variable Generation

Select the variable type from the aforesaid interface, fill in “variable name” and values, click on [OK] to add variables in [Variable] list. Notes: variable type should be reflected when it is named for the convenience of future operation. Please refer to the following chapters for detailed setting methods.

Chapter 6 Trajectory Edition

6.1 Overview of Trajectory Edition

The motion trajectory determines the way the robot moves from one posture to next posture. The motion trajectory in doMotion series control system includes multiple types in joint space and end space, which can enable the robot to complete complicated and diversified motions.

This chapter will introduce in turn the following content, including the variable types to be used in trajectory edition, how to switch between user coordinate system and tool coordinate system, add various motion trajectories and path points.

6.2 Related Variable

In this chapter, the variable types such as TERMINAL, JOINT and FRAME will be introduced. Select [Variable Operation] button in "Add Command" under [Mission Edition] and switch to [Variable] interface.

6.2.1 JOINT Type Variable

Select "JOINT" on the top of the interface as shown in Fig. 33.

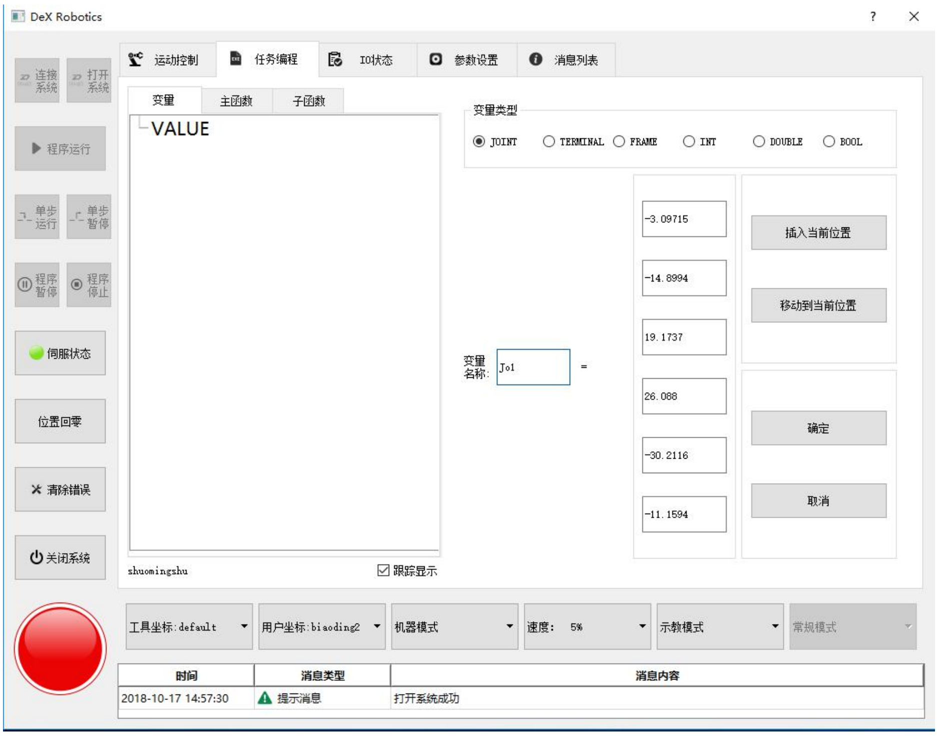

Fig. 33 JOINT Variable

Input name for the variable in "Variable Name". Notes: variable type should be reflected (add prefix "Jo", for instance).

Two ways to input values:

1) Manually input numbers in every box, then click on [Move to Input Position] button to make the robot move to the position. Notes: Ensure the input values are correct and the ambient security when the robot is moving, motion speed can be reduced by adjusting "Speed" percentage on the bottom; check the values and click on "Confirm" button, then JOINT-type variable will be added into the [Variable] list on the left side;
2) Click on [Insert Current Position], the JOINT values of robot's current position will be automatically input, click on [OK] button to successfully add the variable.

6.2.2 TERMINAL Variable

Select "TERMINAL" on the top of [Variable] interface, as shown in Fig. 34.

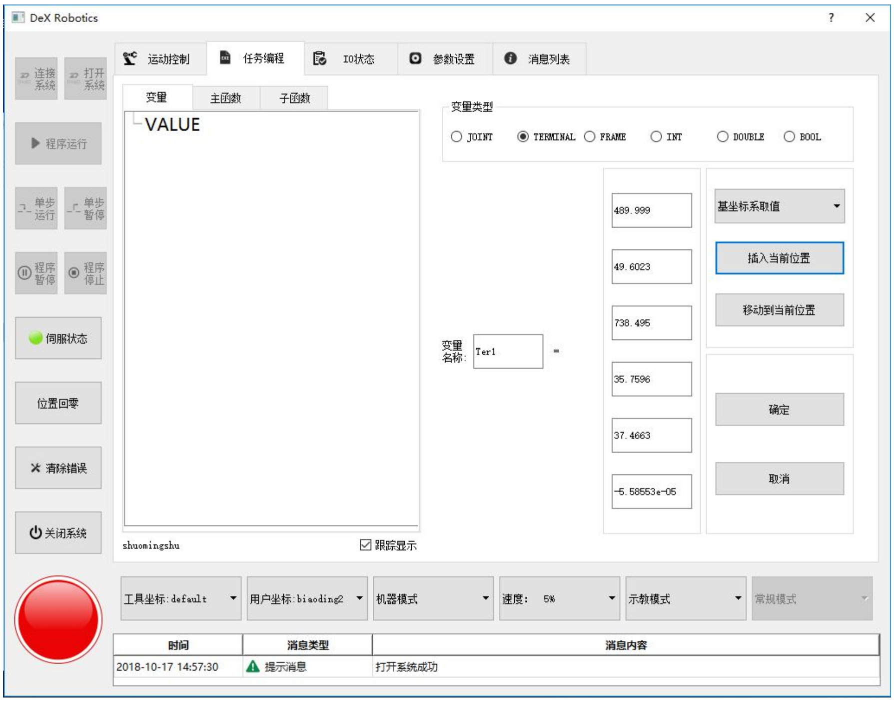

Fig. 34 TERMINAL Variable

Input name for the variable in “Variable Name”. Notes: variable type should be reflected (add prefix “Ter”, for instance). Compared to JOINT variable, TERMINAL needs selection of coordinate system to indicate which TERMINAL value is corresponding to which coordinate system. If “User Coordinate System Value” is selected in the drop-down box, it means the input values are corresponding to user coordinate system. The actual user coordinate system should is determined by the set values in [Coordinate System Switch] then (please refer to the next section).

Two ways to input TERMINAL values:

1) Manually input numbers in every box, then click on [Move to Input Position] button, check the values and click on “Confirm” button;
2) Click on [Current Position], the TERMINAL values of robot’s current position will be input automatically; if “User Coordinate System” is selected, the input values in current user coordinate system, e.g. the [User Coordinate System] on the bottom, are corresponding to this user coordinate system; on the other hand, the TERMINAL values will also

change with different selection of coordinate systems in [Tool]; therefore, it's necessary to confirm the selected [User] coordinate system and [Tool] coordinate system are correct; click on [OK] to successfully add the variable.

Notes:
If the input value of path point is relative variation, do not click on [Move To Input Position] button.

6.2.3 FRAME Variable

FRAME variable is used to add [Coordinate System Switch] command. Select "FRAME" on the top of [Variable] interface, as shown in Fig. 35.

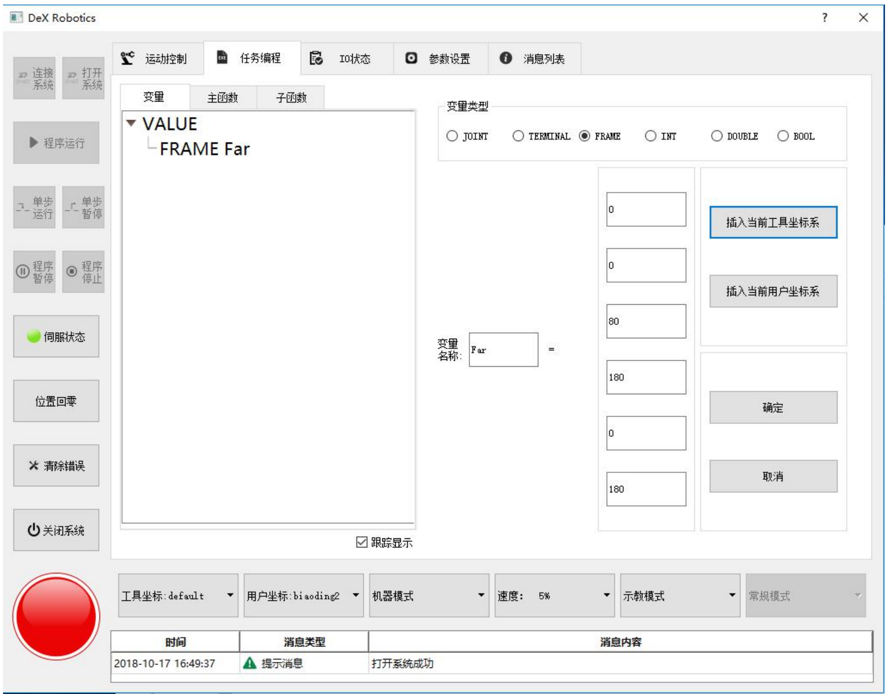

Fig. 35 FRAME Variable

Input the variable name in the input box below, notice that variable type should be reflected (add prefix "Fra", for instance).

Two ways to input FRAME values:

1) Manually input numbers in every box and click on [OK] button;

2) Click on [Insert Current User Coordinate System], input the selected values in [User] coordinate system at the bottom left of the interface; or click on [Insert Current Tool Coordinate System], input the selected values in [Tool] coordinate system at the bottom left of the interface, click on [OK] button;

6.3 Switching of Coordinate System

Insert [Coordinate System Command] in the edit list of [Function] to change user coordinate system or tool coordinate system, so that the motion trajectory segments and path points are within the changed coordinate system.

In “Add Command” on [Mission Programming] interface, click on [Coordinate System Command] and switch to corresponding interface, as shown in Fig. 36.

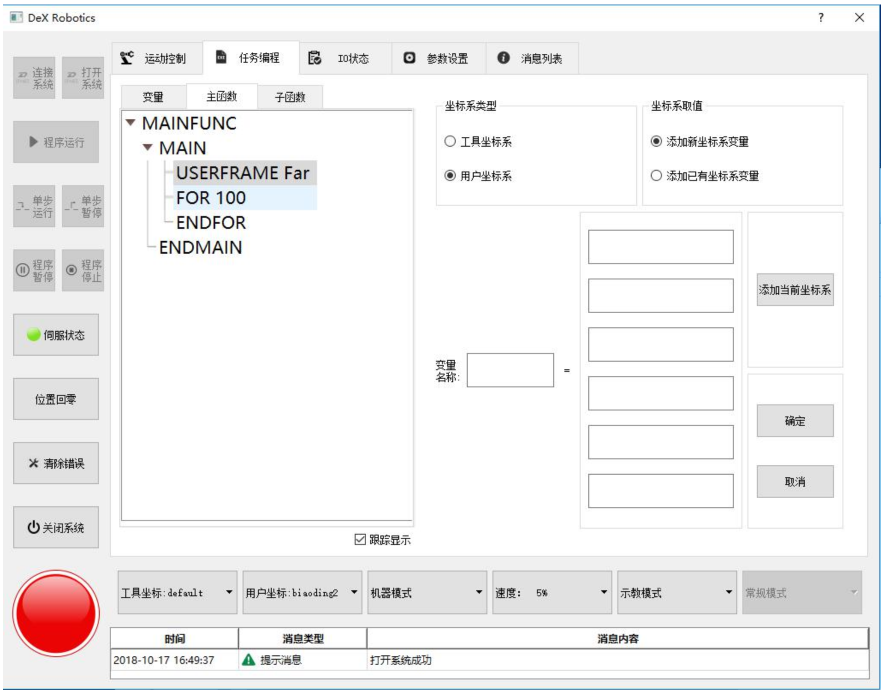

Fig. 36 Switching of Coordinate System

Then, “User Coordinate System” or “Tool Coordinate System” can be selected on the top.

Three ways to input coordinate system:

1) Add FRAME variable in the [Variable] list, click on [Add Variable of Existing Coordinate System] in [Coordinate System Values] on the top, input the name of the added variable in [Variable Name], click on “Confirm” button;
2) Click on [Add Variable of New Coordinate System] in in [Coordinate System Values] on the top right, input digit numbers in input box, click on [OK];
3) Click on [Add Variable of New Coordinate System] in [Coordinate System Values] on the top right, input “Variable Name”, click on [Add Current Coordinate System] button. If “User Coordinate System” is selected on the left, the current values of [User] coordinate system on bottom left will be automatically input; if “Tool Coordinate System” is selected, the current values of [Tool] coordinate system on the bottom left will be automatically input; click on [OK].

It’s a good programming habit to specify the user coordinate system and tool coordinate system at the very beginning, which can prevent the potential dangers caused by the uncertainty of coordinate system selection.

The user can insert the command [Coordinate System Transformation] at any position of program segment to enable the robot task to be converted between different coordinate systems at any time. However, it is necessary to ensure the correctness of coordinate systems and the security of the robot's motion path as well as the surrounding environment.

6.4 Edition of Motion Trajectory

Select a line of codes from [Function] list on the left side of [Programming] interface, the motion trajectory will be inserted into the line below the code; click [Motion Command] button in “Add Command” on the right to switch to the interface of motion edition.

6.4.1 Linear Motion

The linear motion is programmed in the Cartesian space at robot end to make robot end to move in a straight line. Linear motion includes absolute mode MOVL and relative motion MOVLR. The difference is that the path point values added in absolute mode is the absolute position values of the robot; while the path point values added in the relative mode is the relative change values compared to the previous path point.

Select MOVL or MOVLR on the top as shown in Fig. 37. Input speed, acceleration, impact and turning radius ratios (pay attention to the prompt information on input ranges); select the coordinate system corresponding to the motion segment; click on [Next] button.

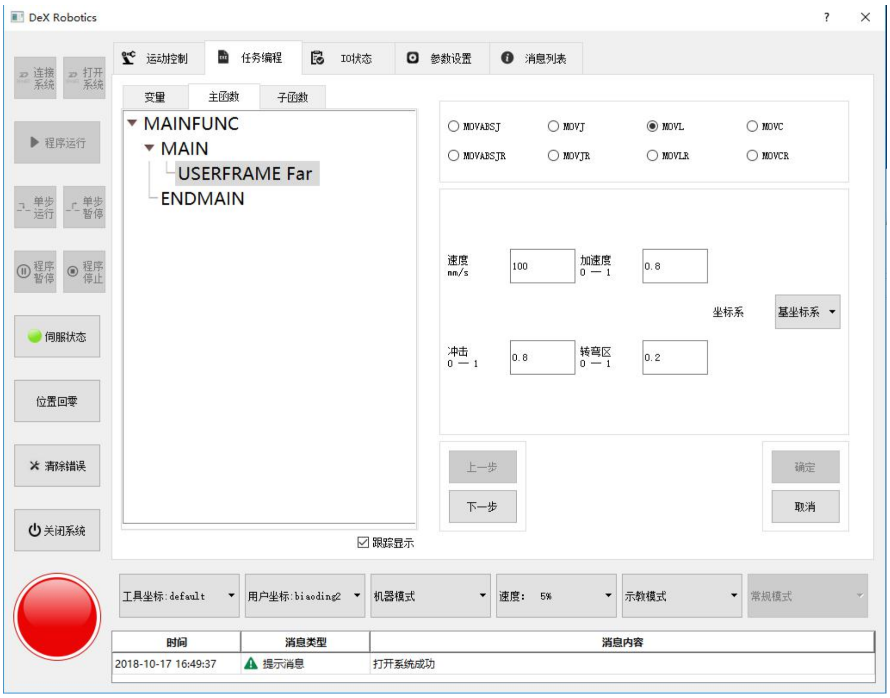

Fig. 37 Absolute Linear Motion

The interface of path point edition is shown in Fig. 38, which can add a TERMINAL path point for linear motion. Select [User Coordinate System] or [Base Coordinate System] to determine the coordinate system to which the added path point values refers (the user coordinate system is determined by the current selection of the [User] coordinate at the bottom left). Normally, the selection here is consistent with the coordinate system selection in previous interface.

Three ways to complete adding path points:

1) Select [New Path Point], input path point name in the box after "Path Point Name"; click on [Insert Current Point], the Cartesian coordinates of robot's current position will be automatically input, click on [Add Road Point] and [OK] to complete; then a New TERMINAL variable will be generated.
2) Select [New Path Point], input path point name in the box after "Path Point Name"; manually input values in the box, click on [Add Road Point] and [OK] to complete; then a New TERMINAL variable will be generated.
3) Select [Existing Path Point], input the name of TERMINAL variable which has been created in the box after "Path Point Name"; click on [Add Path Road] and [OK] to complete.

If "Road Point Name" is not input, the system will automatically add a TERMINAL variable named as "ter+ digital serial number" by the system to store this path point.

Select the path point, click on [Delete Path Point] and [OK] to delete the path point. Notes: deleting path point does not mean deleting corresponding variable.

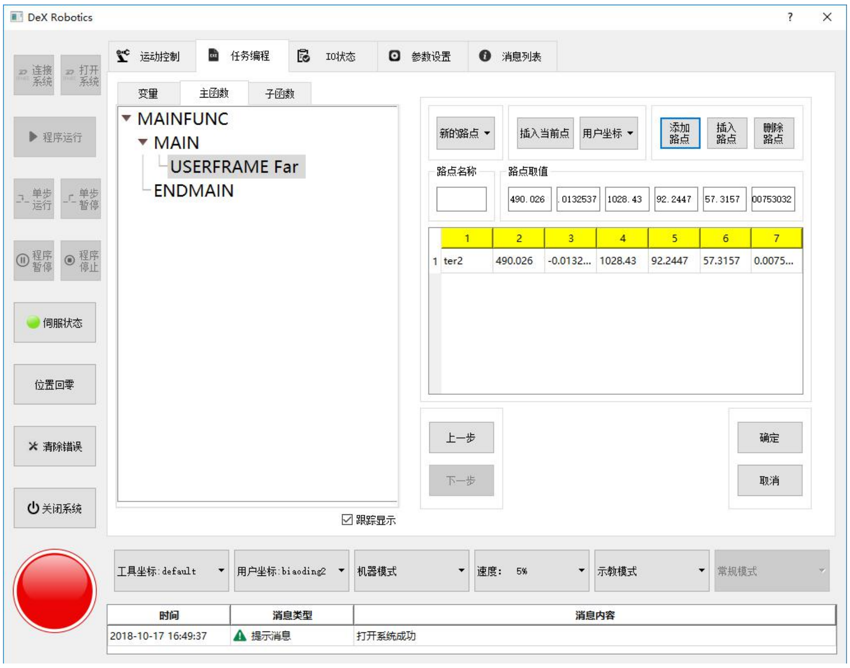

Fig. 38 Edit TERMINAL Path Point

After edition is completed, the motion trajectory of this segment will be added in [Function] list.

6.4.2 Circular Motion

Like linear motion, the circular motion also have two modes, absolute mode MOVC and relative motion MOVCR. Select MOVC in the drop-down box on the left, as shown in Fig. 39. Select [Full Circle] or [Partial Circle] motion; input speed, acceleration, impact and turning radius values (pay attention to the prompt information on input ranges); select coordinate system corresponding to this motion; click the [Next] button.

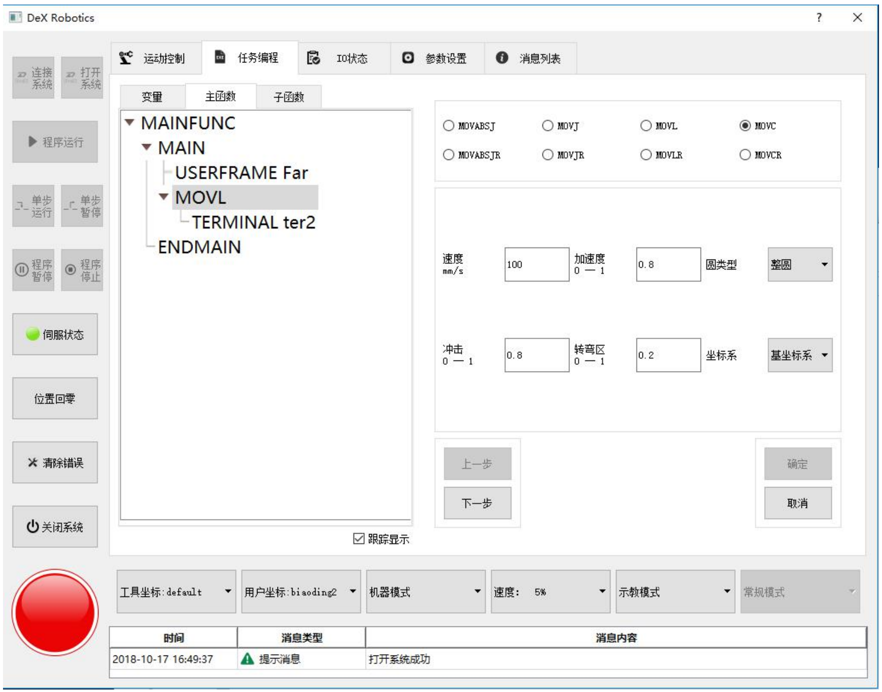

Fig. 39 Absolute Circular Motion

On the interface of path point edition, two TERMINAL path points can be added to circular motion, which successively correspond to the middle and end points on the circle. It can be added in the same way as linear motion.

6.4.3 Joint Motion

The joint motion is planned in robot's joint space with unknown end motion trajectory, therefore it's necessary to ensure there's no safety hazard; joint motion can be added to fit for multiple path points. It is recommended to add as many path points as possible to increase motion controllability and fitting smoothness.

Joint motion includes MOVJ and MOVABSJ, the difference is that the path points of the former belong to TERMINAL type while that of the latter belong to JOINT type; like the linear and circular motions of Cartesian rectangular space planning, the MOVJ and MOVABSJ motions also have the corresponding modes of relative motion, e.g. MOVJR and MOVABSJR.

Select MOVJ/MOVJR/MOVABSJ/MOVABSJR in the drop-down box on the left, as shown in Fig. 40. Input speed ratio value; select coordinate system corresponding to this motion (since the values of joint type are directly added in MOVABSJ and MOVABSJR, it's unnecessary to specify the coordinate system, as shown in Fig. 41); click the [Next] button.

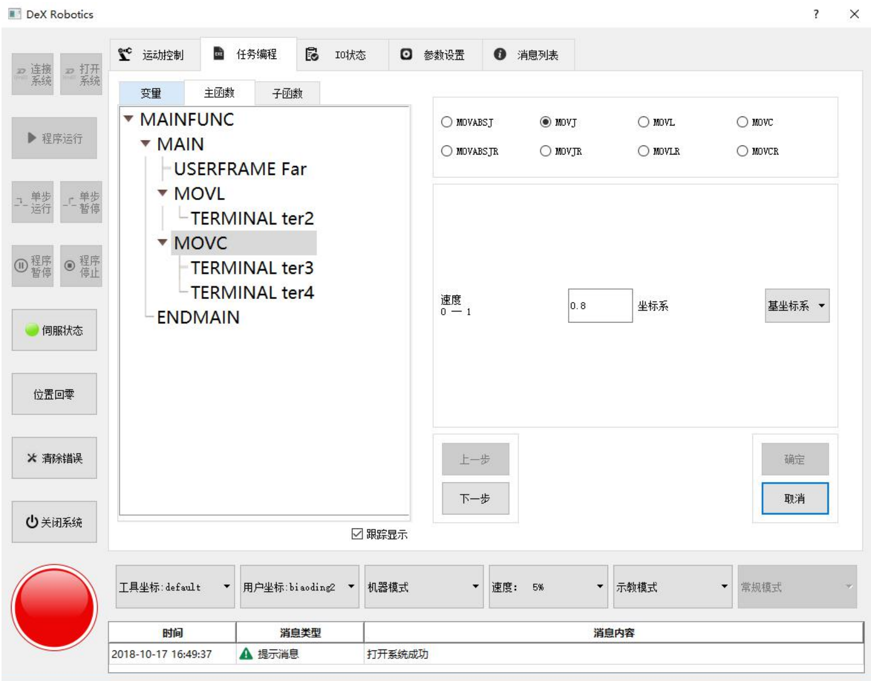

Fig. 40 Absolute Joint Motion

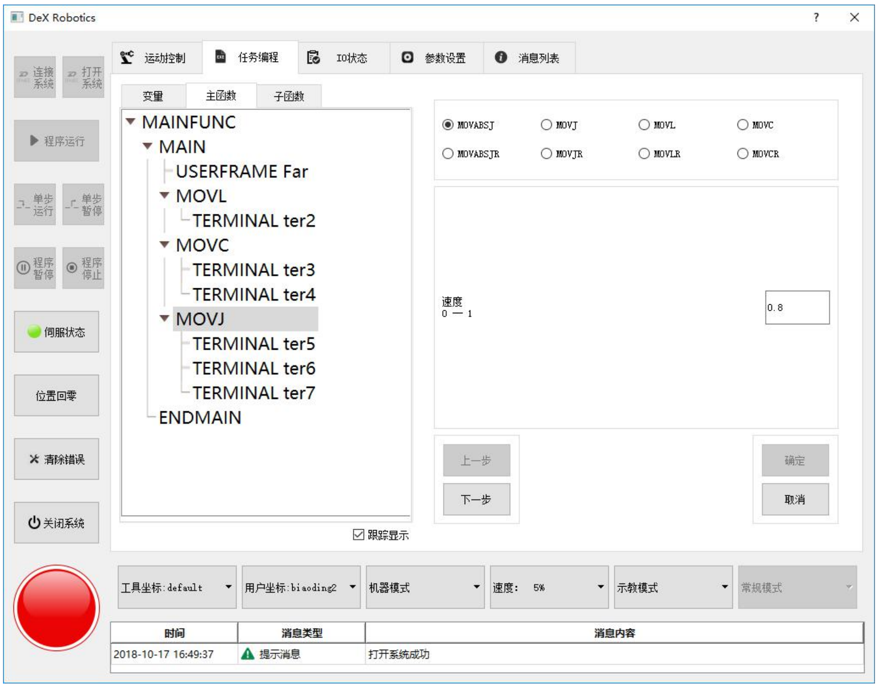

Fig. 41 MOVABSJ

The interface of path point edition is shown in Fig. 42, two arbitrary path points can be added to joint motion.

If MOVJ/MOVJR motions are selected, add TERMINAL path points in the same way as linear motion and circular motion. If MOVABSJ/MOVABSJR motions are selected, similarly there are three ways to add JOINT path points:

1) Select [New Path Point], input the name in the box after the "path point name"; click on [Insert Current Point], the robot's joint coordinates at current position will be automatically input, click on [Add Path Point];
2) Select [New Path Point], input the name in the box after the "path point name"; manually input values in the box, click on [Add Path Point];
3) Select [Existing Path Point], input the JOINT-type variables created in the box after the "path point name"; click on [Add Path Point];

If "path point name" is not input, the system will automatically add JOINT-type variables named as "joP + digital serial number" by the system to store this path point.

Select a path point first, click on [Insert Path Point] button, a point can be inserted above the path point selection.

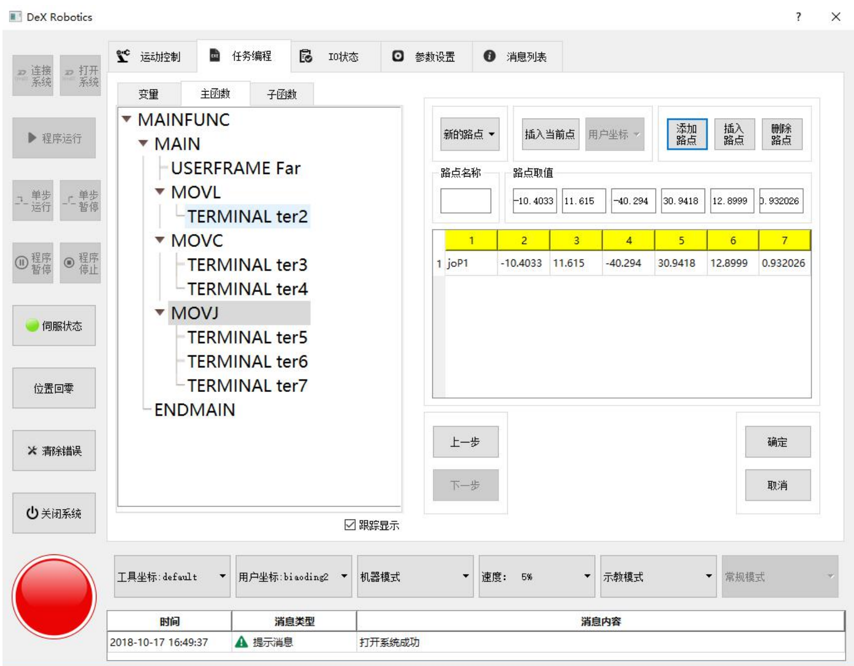

Fig. 42 Edit JOINT Path Point

Notes:

If relative motion types MOVJR/ MOVLR/ MOVCR/ MOVABSJR are the objects to be added with path points, the input values of path point should be relative change from the previous point, so DO NOT click on [Insert Current Point] button to avoid input absolute position values. It's necessary to manually input values or fill in variables (variables must be relative values.)

6.4.4 MOVB Spline Motion MOVB

The NURBS curve is a kind of advanced spline curve fitting algorithm which can realize high-order smoothing of curves; doMotion series control system has built-in B-spline curve trajectory which can realize high-quality fitting of multiple positions and postures.

Click [Mode Command] in "add command" on the [Mission Programming] interface, select "Spline Curve" in "Mode", as shown in Fig. 43.

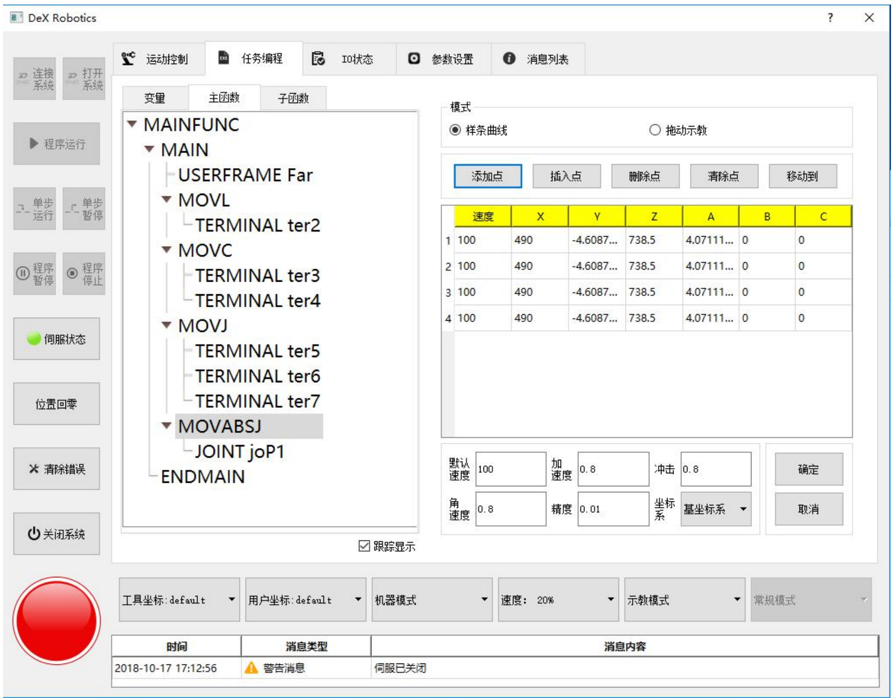

Fig. 43 B-spline Curve Motion

The speed, acceleration, angular velocity (adjust the angular rotation speed when the end rotates in pure posture) and impact equivalent; wherein, the unit of speed is mm/s, input proportional value between 0 and 1 for acceleration, angular velocity and impact.

“Accuracy” refers to the accuracy of B-spline fitting with mm as the unit. The smaller input values, the higher fitting accuracy, but the longer fitting calculation time; too high input values will reduce the fitting accuracy, even result in fitting failure, therefore the value should not exceed 0.5.

Finally, select the coordinate system corresponding to this segment of motion.

User can switch to [Motion Control] interface to conduct demon operation to move robot to the target point and return to the interface of [Spline Curve] edition, click on [Add Point] button to add current demon point into the end of the sequence; or click on [Insert Point] to inset before the point selected by current cursor; the [speed] of the first column can be modified to change the displacement velocity value between the two points, the unit is mm/s; the user can add any number of points;

Click on [Delete Point] button to delete the point selected by the cursor; click on [Empty Point] button to clear the list.

Select a point in the list, click [Move To] to move the robot to the selected point.

Click on [OK] to complete edition of B-spline motion trajectory and return to the main page. Then, the MOVB trajectory is added to the [Function] list.

6.4.5 Parameter Interpretation

6.4.5.1 Speed / Acceleration / Impact

The linear line, arc and B-spline curve in the doMotion series control system adopt S-type velocity curve planning (curves are shown in Fig. 44) to ensure high-order smoothing of motion. The **velocity, acceleration** and **impact** are corresponded to Vm, Am and Jm in the Fig.; the acceleration and shock ratio parameters are based on the end values set in the **Proportional Parameter**.

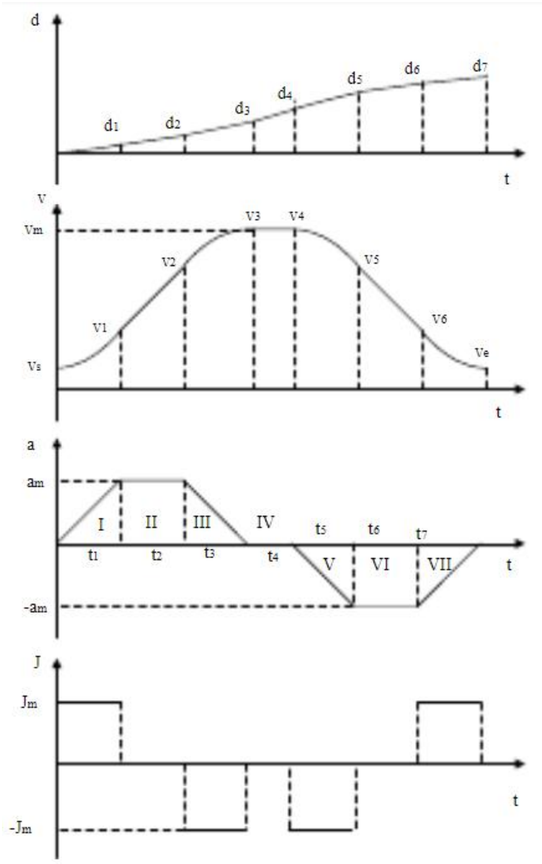

Fig. 44 S-type Velocity Curves

6.4.5.2 Joint Motion Velocity Ratio

MOVJ, MOVJR and MOVABSJ adopts fifth-power polynomial curve fitting to ensure high-order smoothing of the fitting, but the speed, acceleration and impact of the kinematic curves cannot be accurately set; when setting, compared to the joint **Proportional Parameter**, the input speed proportional value is roughly constrained by the moving speed of joint trajectory.

6.4.5.3 Turning Zone

The turning zone realizes a smooth transition of trajectory segments and reduces the tact time of the robot; the meaning of turning radius is shown in Fig. 45 in which the robot starts the transition to the next trajectory when it moves to the turning radius of actual target point. The input proportional value of turning zone radius is based on the smaller length of the front and rear trajectory segments. **Only linear motion and circular motion trajectory can set turning zone transition**; If the proportional value of turning zone is set to 0, it means there's no turning zone, and the end will move to the target point and stop, then continue the next motion.

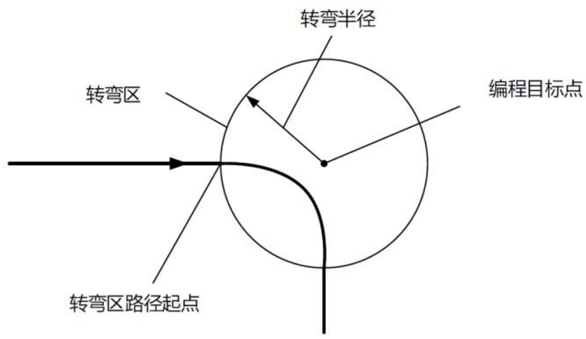

Fig. 45 Turning Zone

Chapter 7 Logic Command

7.1 Overview of Logic Command

The logic command is the necessary component to edit a complete mission program for robot. The doMotion series control system has multiple logic commands for complex missions.

This chapter will introduce the variable types of needed to edit logic command, ways of adding various types of logic commands, variable calculations and the application of sub-functions.

7.2 Related Variable

In this chapter, the INT, DOUBLE and BOOL variables will be mainly involved. Click [Variable Operation] in "Add Coomand" on the interface of [Mission Programming].

7.2.1 INT Variable

Select “INT” at the top of [Variable], as shown in Fig. 46.

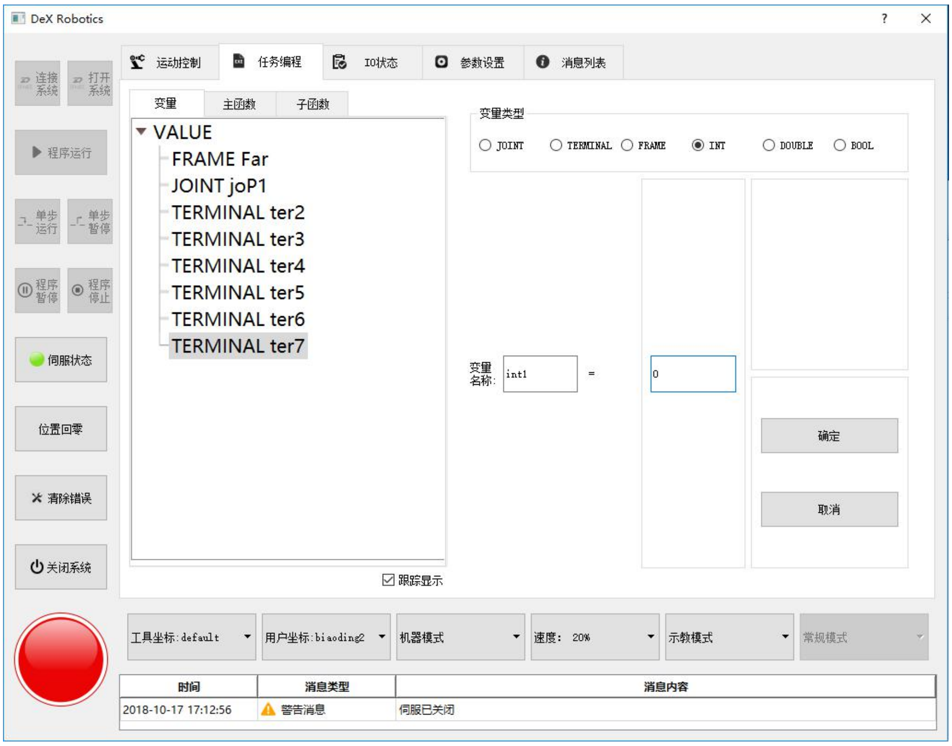

Fig. 46 INT Variable

Input name for the variable in “Variable Name”. Notes: variable type should be reflected (add prefix “int”, for instance). Then enter an integer number in the rear box; click [OK] to add the variable.

7.2.2 DOUBLE Variable

Select “DOUBLE” at the top of [Variable], as shown in Fig. 47.

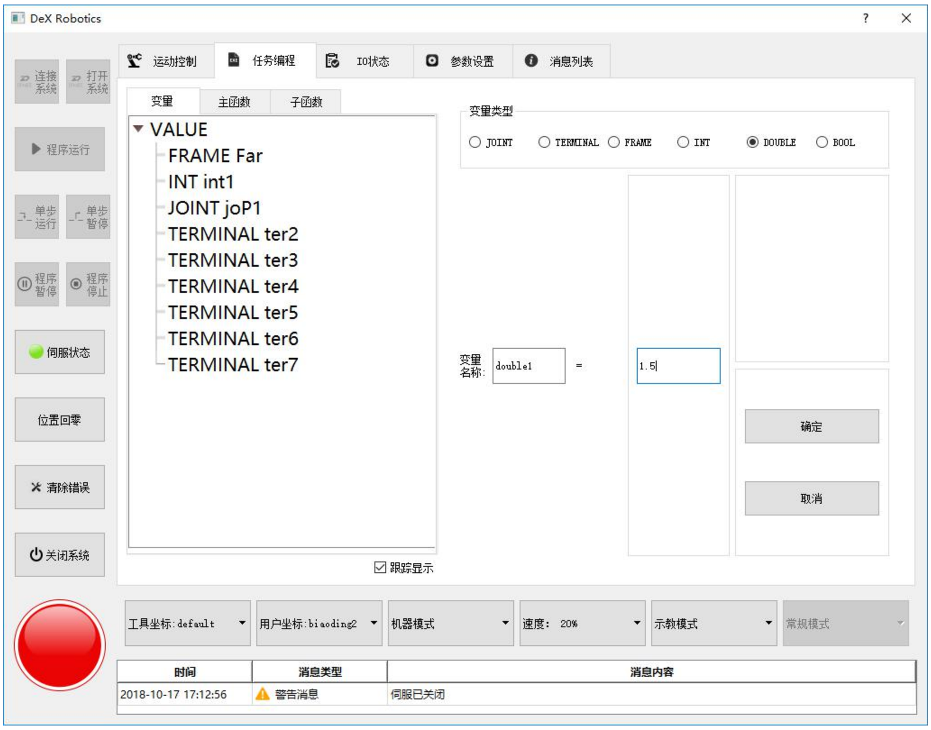

Fig. 47 DOUBLE Variable

Input name for the variable in “Variable Name”. Notes: variable type should be reflected (add prefix “do”, for instance). Then enter a double precision floating point number in the rear box; click [OK] to add the variable.

7.2.3 BOOL Variable

Select “BOOL” at the top of [Variable], as shown in Fig. 48.

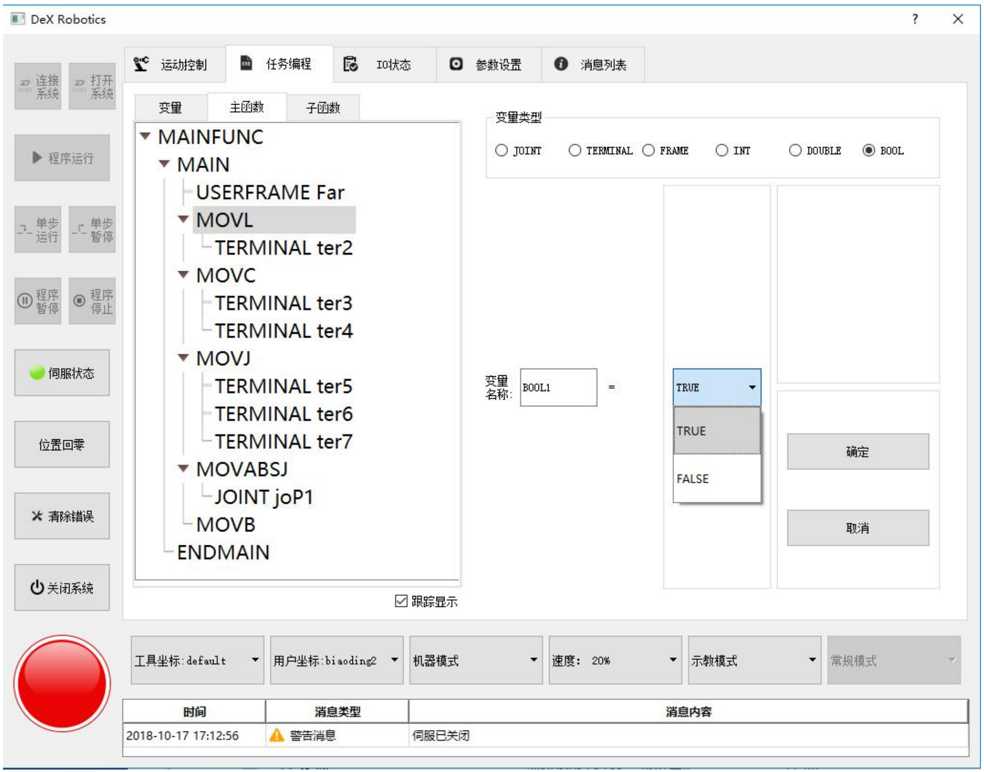

Fig. 48　BOOL Variable

Input name for the variable in "Variable Name". Notes: variable type should be reflected (add prefix "bool", for instance). Then select TRUE or FASLE in the rear drop-down box; click [OK] to add the variable.

7.3　Add Logic Command

Select a line of code in the [Function] list box on the left side of [Mission Programming], and the added logic command will be inserted after this line of code; click on [Process Command] button on the right side.

7.3.1 Loop Command FOR

Select [Loop] at the top, as shown in Fig. 49; select [FOR] in the second drop-down box, then enters the times to be looped in the box [Loop Times] below; click on [OK] to add;

Then select one line of code in the [Function] list, select [Process Command] - [Loop], select [ENDFOR] in the second drop-down box, and click the [OK] button to add.

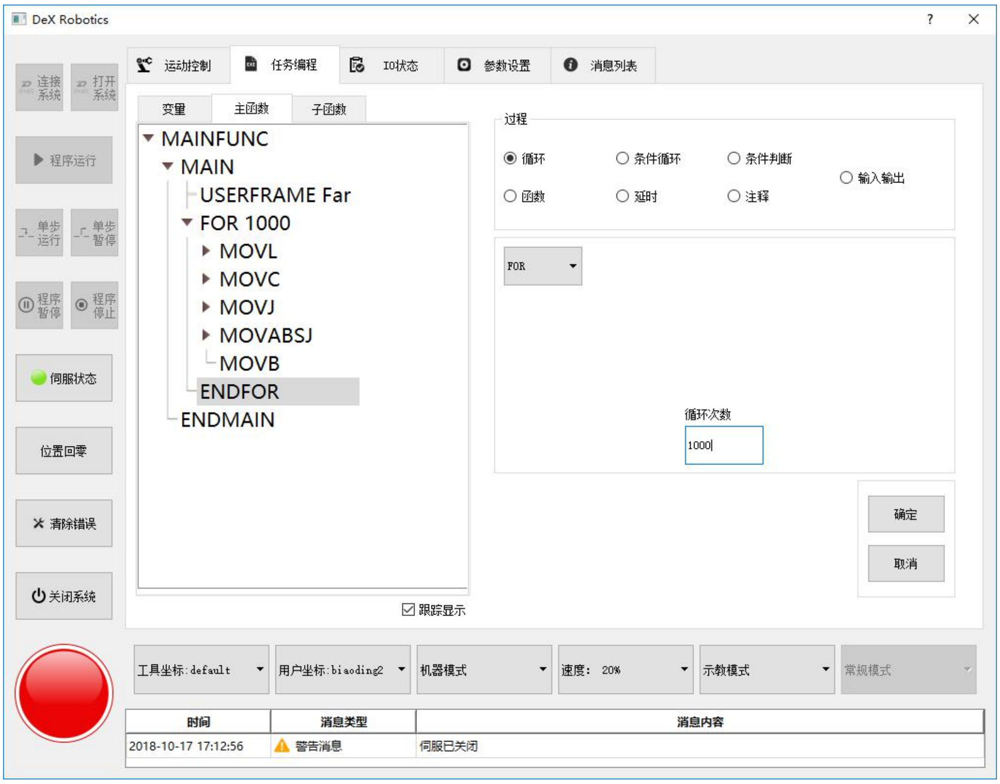

Fig. 49 Loop Command

The code loop between FOR and ENDFOR will be realized after the adding process is completed.

7.3.2 Conditional Loop Command

Select [Conditional Loop] at the top, as shown in Fig. 50; select [WHILE] in the second drop-down box; in the input box below, only variable name can be input before judgment symbol, therefore the variable should be created in advance; the boxes after judgment symbol should be input based on the rules shown in Table 9:

Table 9 Rules to Input Conditional Loop

Variable Type	Corresponding Judgment Input
INT	INT/ input number
DOUBLE	DOUBLE/ input number

BOOL	BOOL/ input TRUE/FALSE

After the input is completed, click the [OK] button to add.

Then select one line of code in the [Function] list, select [Process Command] - [Conditional Loop], select [ENDWHILE] in the second drop-down box, and click the [OK] button to add.

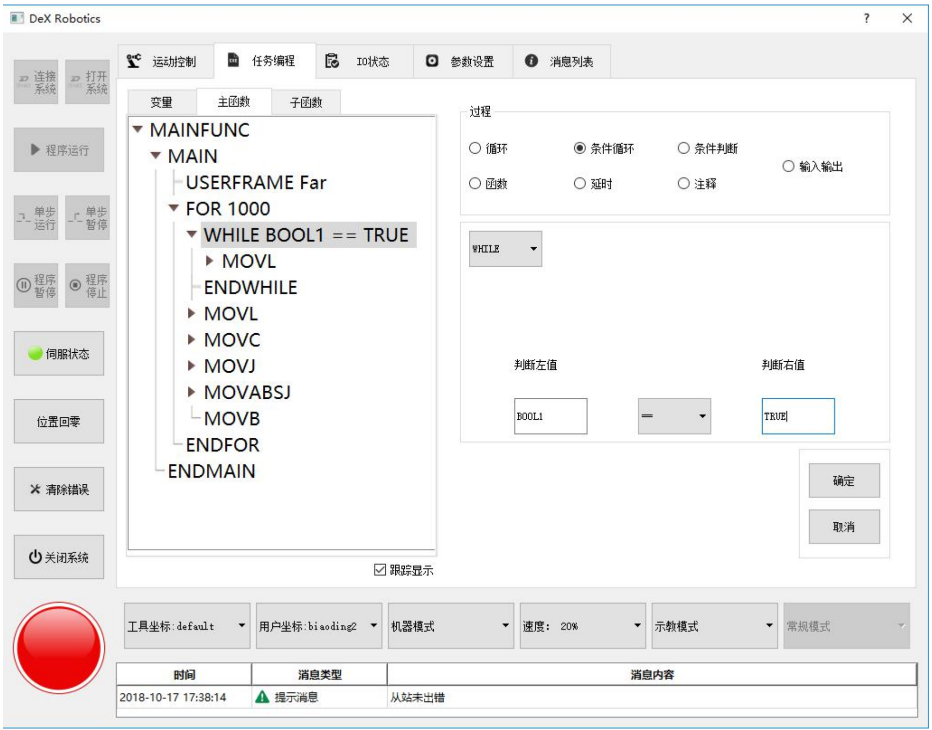

Fig. 50 Conditional Loop Command

The conditional loop between WHILE and ENDWHILE will be realized after the adding process is completed.

7.3.3 Judgment Command

Select [Conditional Judgment] at the top, as shown in Fig. 51; select [IF] in the drop-down box; the rules for input box below are the same as in Table 9. After input is completed, click the [OK] button to add.

Then select one line of code in the [Function] list, select [Process Command] - [Conditional Judgement], select [ENDIF] in the drop-down box, and click the [OK] button to add.

The code judgement execution between IF and ENDIF will be realized after the adding process is completed.

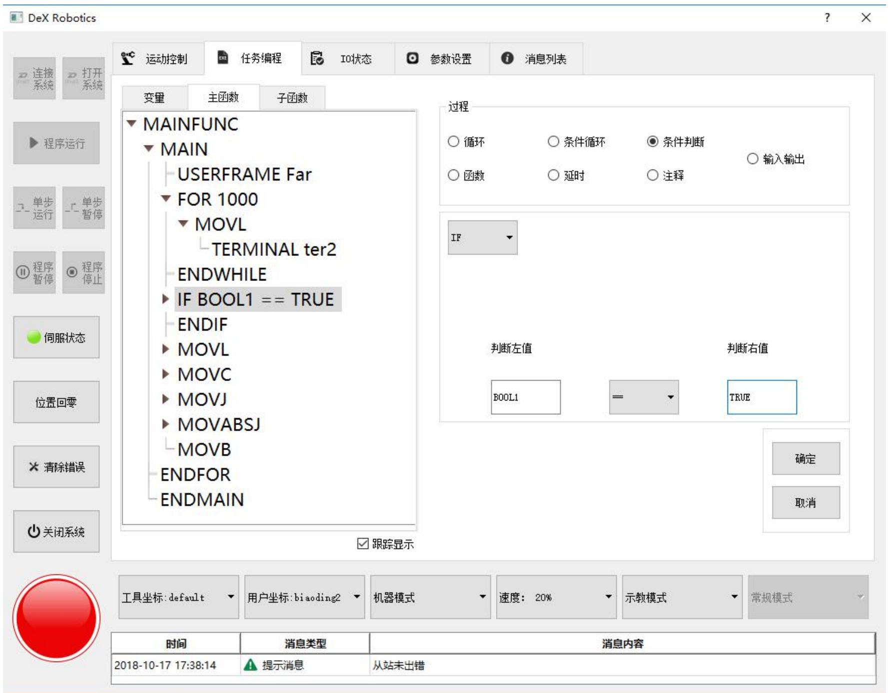

Fig. 51 Judgment Command

Meanwhile, ELSE can also be inserted after IF and ENDIF can be inserted at the end, as shown in Fig. 52. If the IF-ELSE command is in a loop, CONTINUE or BREAK command can be inserted to continue or jump out of the loop.

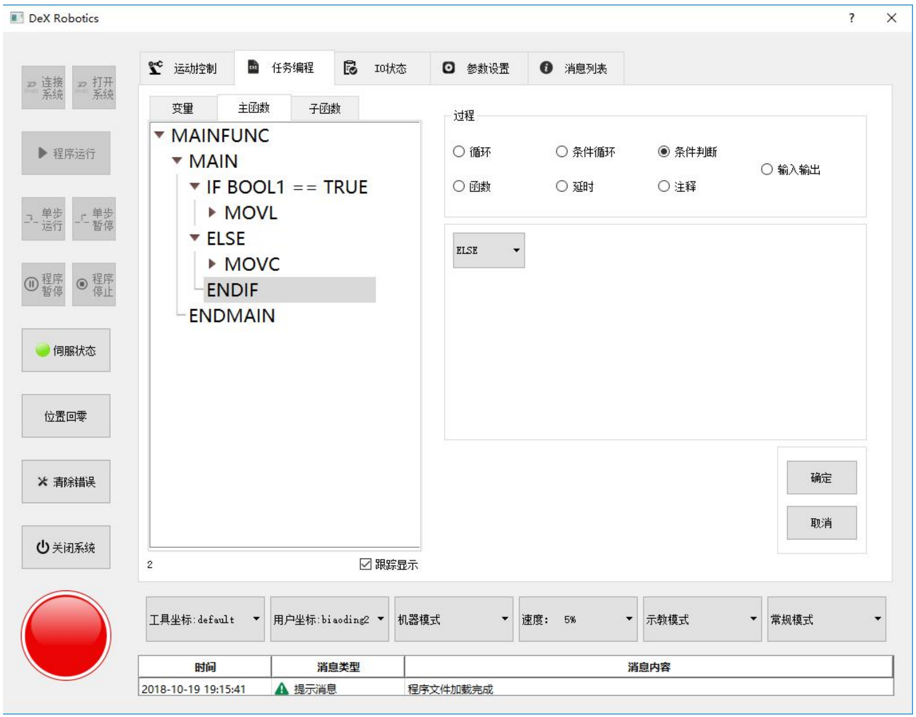

Fig. 52 IF-ELSE Command

Notes: All logic commands in this section must be added with "END" command, otherwise a syntax error will occur.

7.3.4 Delay Command

Select [Delay] at the top, as shown in Fig. 53; enter the delay time (unit: second) in the [Delay Time] box below; click the [OK] button to add;

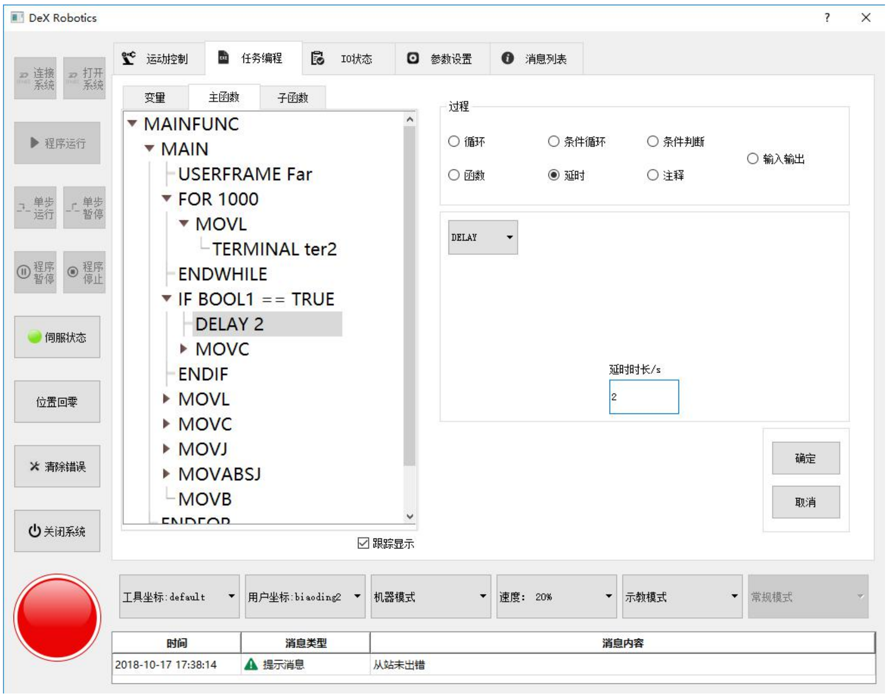

Fig. 53 Delay Command

7.4 Variable Calculation

Select a line of code in the [Function] list on the left side of [Task Programming], the added variable calculation command will be inserted after the line of code; click on [Calculate Command] in the "Add Command" on the right side, as shown in Fig. 54.

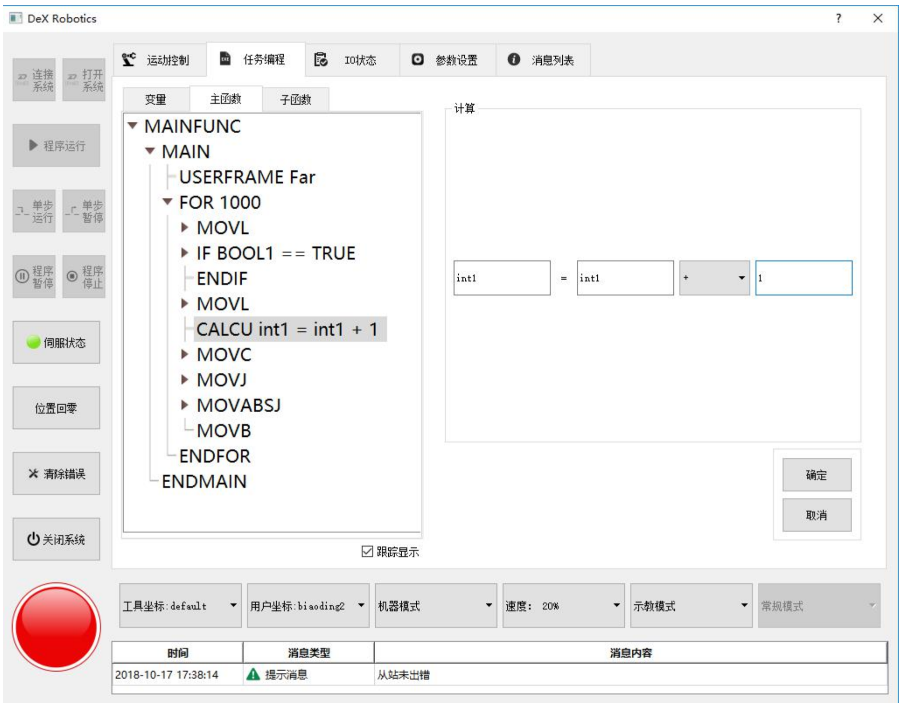

Fig. 54 Variable Calculation

In the input box below, only variable name can be input before the judgment symbol, so the variable should be created in advance; the box in the input box below, only variable name can be input before judgment symbol, therefore the variable should be created in advance; the boxes after judgment symbol should be input based on the rules shown in Table 10:

Table 10 Rules to Input Calculation Command

Variable	Box 1 Input	Supported Symbol	Box 2 Input
JOINT	JOINT	+,-	JOINT
TERMINAL	TERMINAL	+,-	TERMINAL
FRAME	FRAME	+,-	FRAME
INT	INT/ input number	+,-,*,/,%	INT/ input number
DOUBLE	DOUBLE/ input number	+,-,*,/	DOUBLE/ input number
BOOL	BOOL/ input TRUE/FALSE	None	None

The addition of TERMINAL variable is shown in Fig. 55. After input is completed, click on [OK] to add.

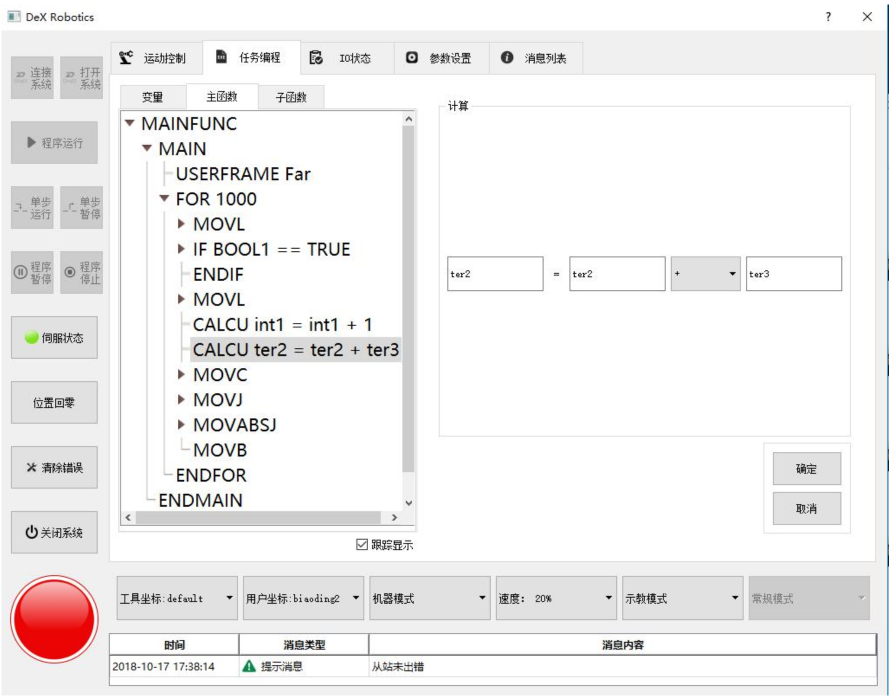

Fig. 55 Addition of TERMINAL Variable

7.5 Sub Function

7.5.1 Create Sub Function

Switch to [Sub Function] list on the left side of [Mission Programming], select SUBFUNC or ENDFUNC statement of previous sub-function, and click on [Add Sub Function] in "Add Command" on the right side, as shown in Fig. 56.

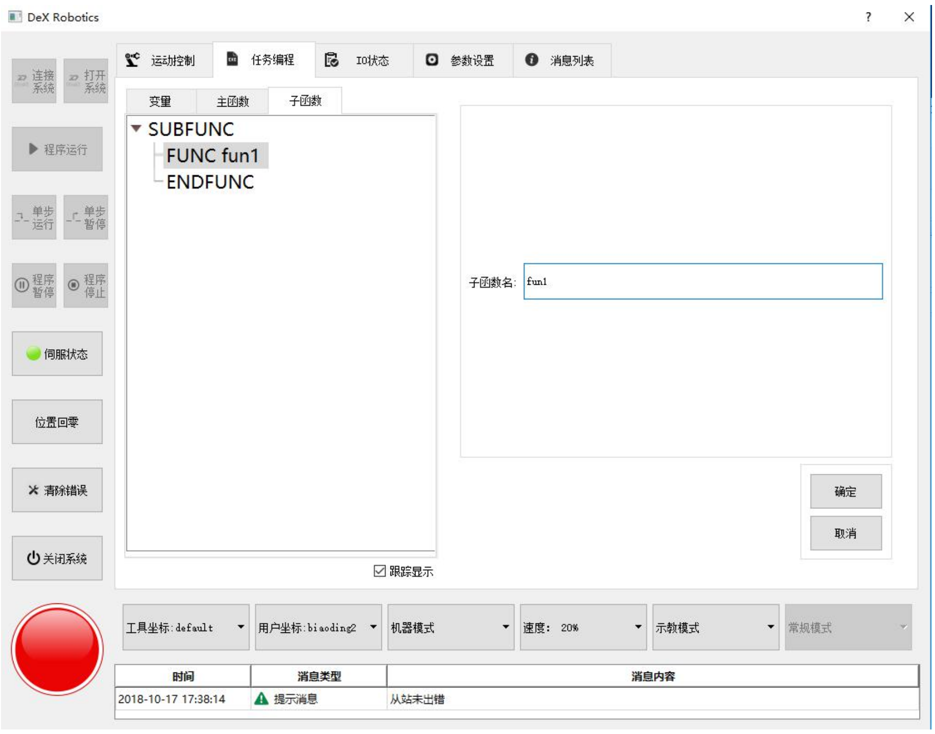

Fig. 56 Create Sub Function

Enter the name of sub-function in the "Function Name" below, click the [OK] button to create the sub-function.

Under the subfunction, various code statements can be added in the same way as the main function; the statements can be directly copied and pasted into the [**Subfunction**] from the [**Main Function**] list.

Users can create any number of subfunctions. Make sure the subfunction names are different.

7.5.2 Calling Subfunction

To call subfunction, switch to the [Main Function] list, select a line of code, click [Process Command] on the right, select [Function] on the top, as shown in Fig. 57.

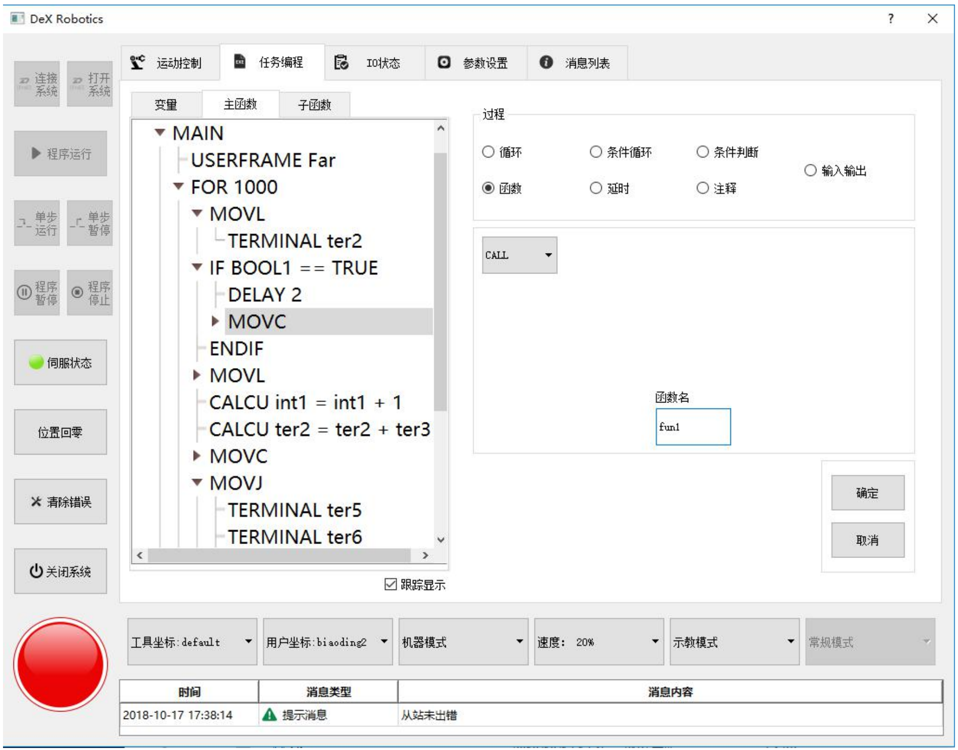

Fig. 57　Calling Subfunction

Enter the subfunction name in the "Calling Function Name" box below, and click the [OK] button to call the subfunction.

Meanwhile, RETURN can also be selected in the drop-down box to exit the subfunction or terminate the main function operation.

Chapter 8 IO Operation

8.1 Overview of IO Function

IO is the main way for the doMotion series control system to conduct interaction with external devices. The control system has 16 IO ports.

Switch to the [IO Status] interface to display the status of every IO port, as shown in Fig. 58. The left side displays the state of current input ports; the right side displays the status of output ports, OFF (grey button) means off, and ON (green button) means on; click on the round buttons to switch the corresponding input and output status.

Fig. 58 IO Status

8.2 Add IO Command

Select a line of code in the [Function] box on the left side of the [Mission Programming], then the added IO command will be inserted after the line of code; click on [Process Command] on the right and select [IO] at the top.

8.2.1 IO Input

Select [WEITDIN] in the drop-down box to display waiting for port input. In the "Waiting for Input" box below, enter the port number waiting to be input in the first box, the input format is "DI + number (1-16)", such as "DI1"; the rear box displays the state of waiting for input, select “ON” or “OFF”, as shown in Fig. 59.

BOOL variables can also be input as the state, TRUE means ON, FALSE means OFF.

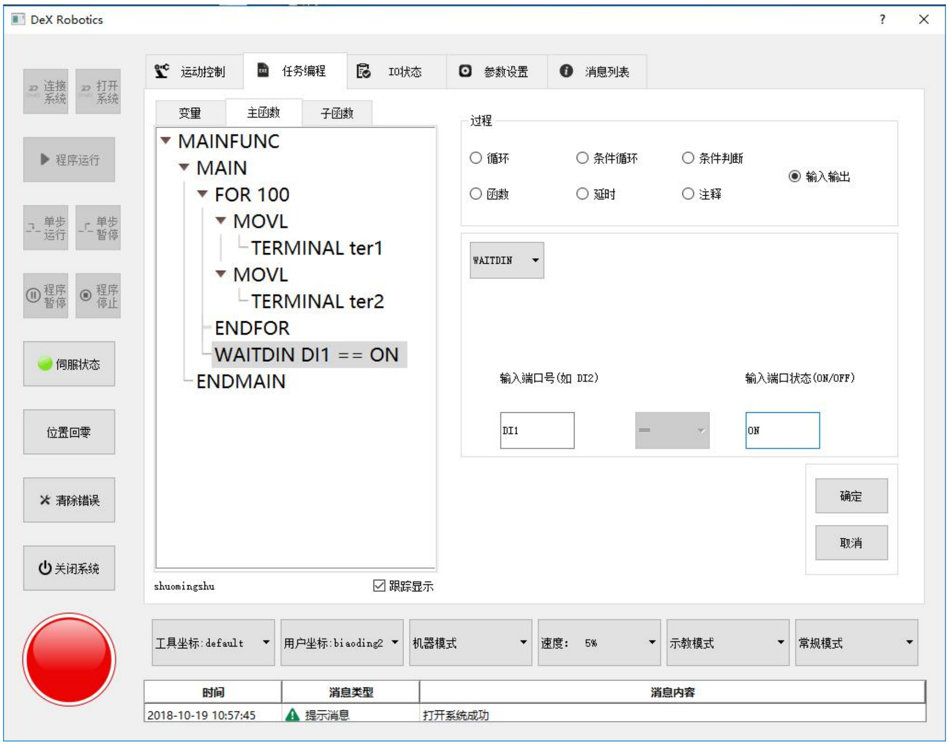

Fig. 59 Waiting for IO Command Input

Click the [OK] button to insert the IO command waiting to be input.

8.2.2 IO Output

Select [DOUT] in the second drop-down box to display port output. In the "Output" box below, enter the port number in the first box, the format is "DI + number (1-16)", such as "DO1"; the rear box displays the state to output, select “ON” or “OFF”, as shown in Fig. 60.

BOOL variables can also be input as the state, TRUE means ON, FALSE means

OFF.

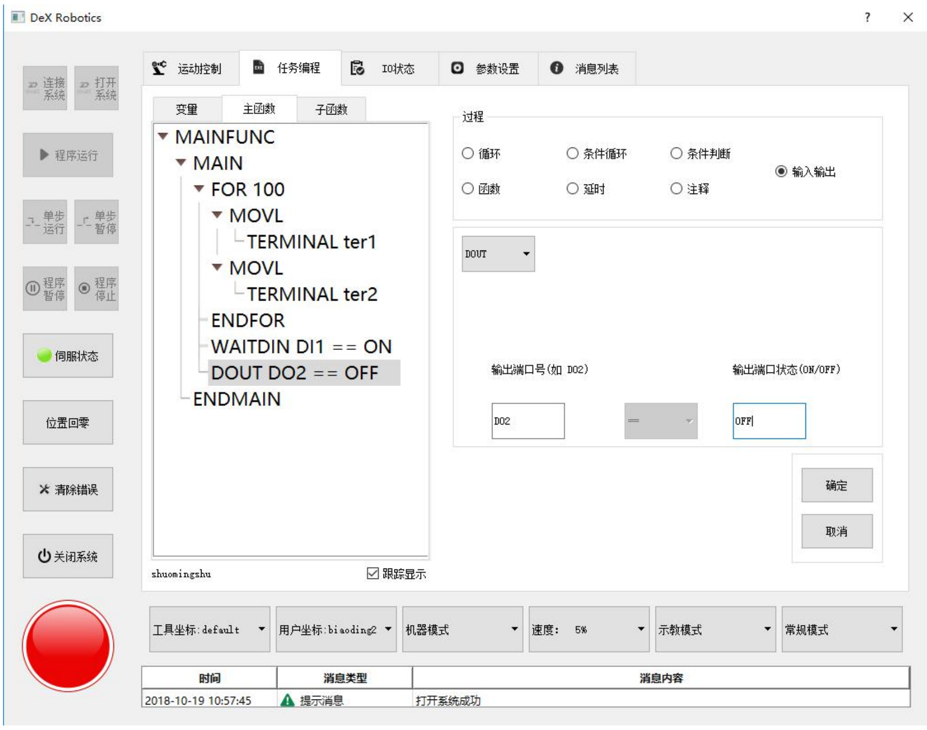

Fig. 60 IO Output Command

Click the [OK] button to insert the IO output command.