Nanjing Yueqing Information Technology Co., Ltd

# iRobotCAM

# User Manual 1.2

Intelligent Computer Aided Manufacturing for Robotics

# Table of Contents

# Chapter 1: iRobotCAM Installation Guide

This guide will walk you through the process of installing iRobotCAM on your Windows operating system.

## 1.1 Prerequisites

- Computer Windows specifications: Supports Win10 64bit or higher versions.

- ZW3D version requirements: The minimum support is ZW3D 2023X educational version, and ZW3D 2024 version has no requirements.

- This product is developed at the expert level of ZW3D software.

## 1.2 Installation Trial+Activation Version

**Precautions before installation:**

Here are the precautions you need to take before installing the software package:

**1. Trial period:**

- The software package has a 30-day trial period.
- After 30 days, you will need activation authorization to continue using it.

**2. ZW3D installation:**

- Make sure you have ZW3D software installed on your system (download from the official website).

**3. Before installing the iROBOTCAM software package (English version), complete the following step:**

- Close the ZW3D software.

**4. Installing the software package:**

- Double-click the software package to begin the installation process.
- Follow the on-screen instructions to complete the installation.

**Additional notes:**

- Ensure a stable Internet connection when activating authorization

- If you encounter any problems during installation, please refer to the software documentation or contact the software vendor.

## Step-by-step instructions on IROBOTCAM plugin for ZW3D 2024 installation.

### Step 1:

- Double-click the installation package file "IROBOTCAM-V1.2-plugins_en-for-ZW3D2024.exe".
- Click "Next" to proceed. (Figure 1-1)

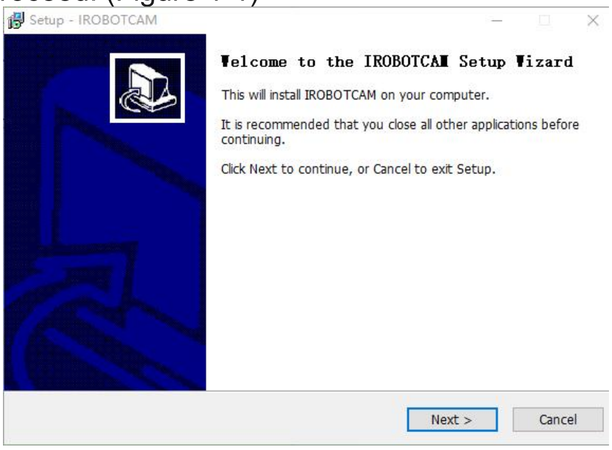



**Figure 1–1: Setup**

### Step 2:

- Read the license information carefully.
- If you agree with the terms, click "I accept".
- Fill in your personal information and click "Next". (Figure 1-2)

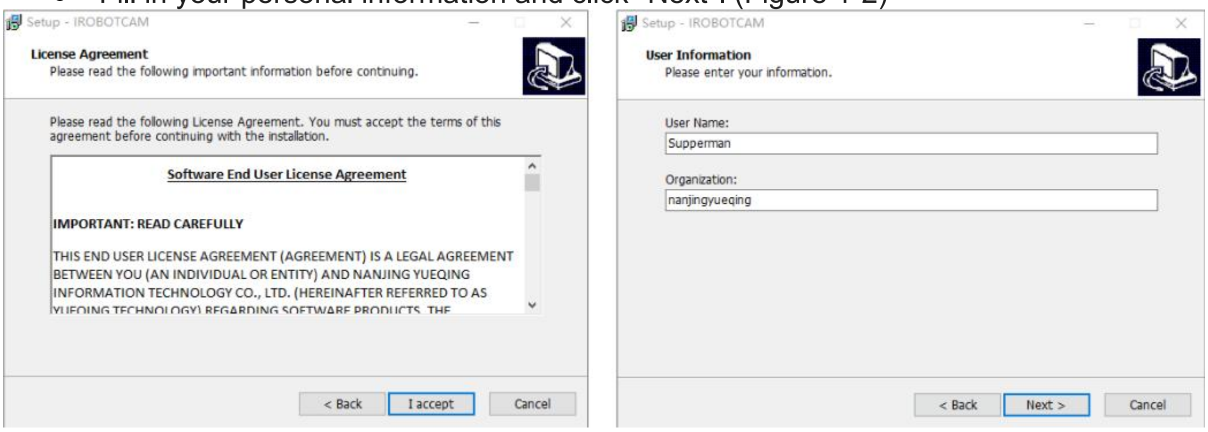



**Figure 1–2： License and User Information**

### Step 3 (Figure 1-3)：

- Select "installation path".

- Click the "Browse" button to change the installation directory if needed.
- Click "Next".
- Select start menu folder.
- Click "Next".

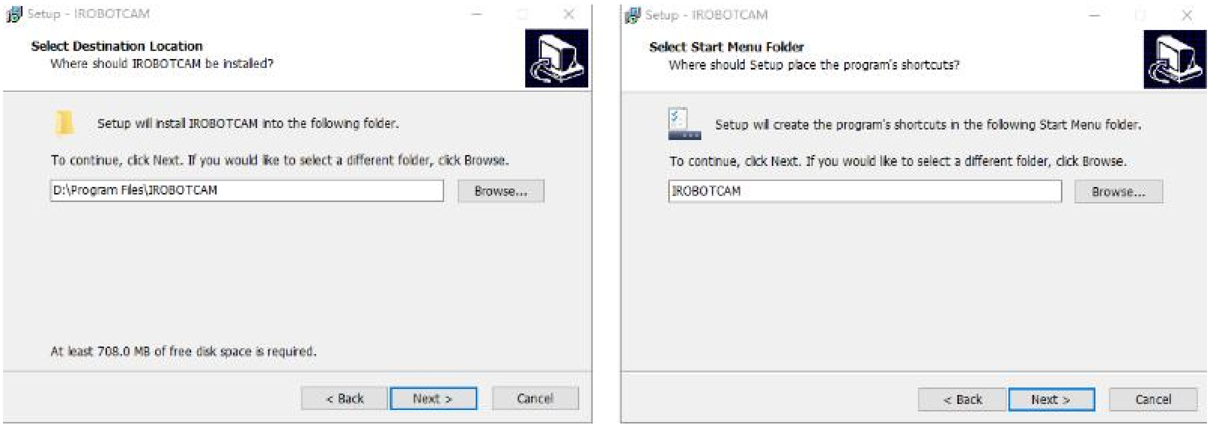



**Figure 1–3：Installation and Start menu folder**

**Step 4**(Figure 1-4)**:**

- A summary of the installation information will be displayed.
- You can click "Back" to make changes or "Install" to proceed.

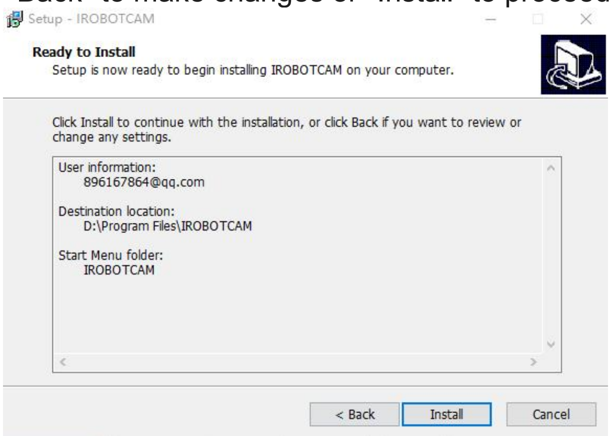



**Figure 1–4: Installation information**

**Step 5:**

- Click "Finish" to complete the installation process. (Figure 1-5)

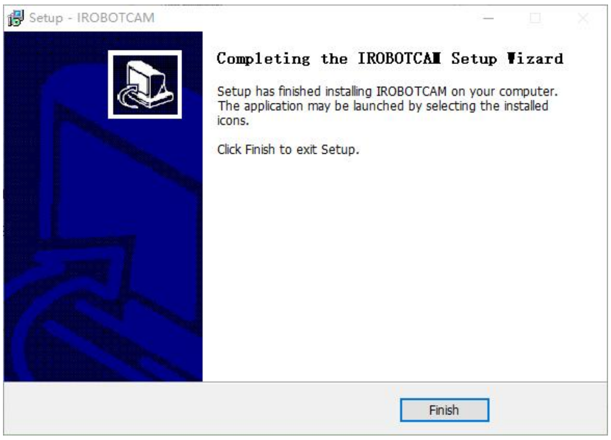



**Figure 1–5: Complete installation**

### Step 6:

- When you first run ZW3D after installation, some default settings will be displayed.
- Click "OK" to acknowledge them. (Figure 1-6)

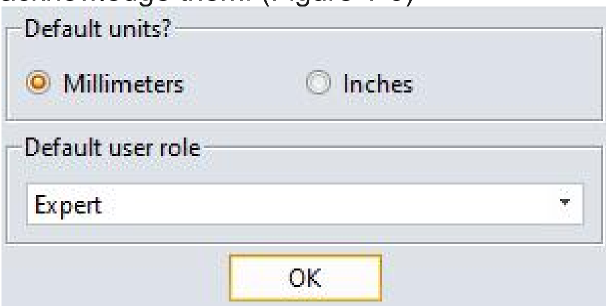



**Figure 1–6：Default setting**

### Step 7:

- Open the ZW3D interface and click on the "IROBOTCAM" module.
- Click "Activation" to activate the software package.(Figure 1-7)

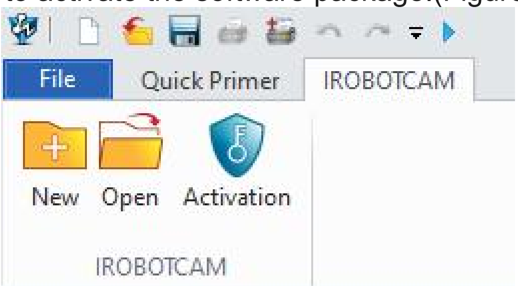



**Figure 1–7: Activeation**

### Step 8:

- The first installation of the plugin provides a 30-day trial period.
- After the trial expires, you will need to activate the plugin using a valid activation code.
- In the "License" window, under the "IROBOTCAM" column, you will see your

current status (trial or inactive).

- To activate the plugin, copy or enter the activation code you received from Yue Qing robot in the "Key ID" field.
- Click "Apply".
- The "IROBOTCAM" status will update to "activated" if the code is valid. (Figure 1-8)

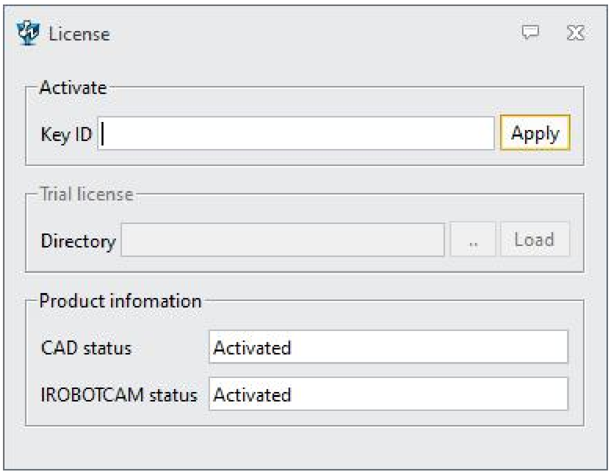



**Figure 1–8：Activate Authorization**

### Step 9:

- You have successfully installed and activated the IROBOTCAM plugin for ZW3D 2024.

### Note:

- The "-" symbol in the activation code is important and should not be omitted.
- If you encounter any problems during installation or activation, please refer to the software documentation or contact the software vendor.

# Chapter 2: Product Key Features

iRobotCAM is dedicated to creating an open and comprehensive platform for digital solutions related to robotics, focusing on various aspects of production line mechatronic. This includes:

**Design concept :**

- Assists in the design of robot-based production lines.
- Provides tools for mechatronic concept design.

**Robot off-line Programming & Simulation:**

- Enables users to program and simulate robot machining processes.
- Offers advanced features for virtual commissioning , ensuring accurate and efficient program development.

**virtual commissioning & Real-world Synchronization:**

- Utilizes a reliable physical simulation engine for realistic virtual testing.
- Achieves seamless debugging by synchronizing virtual and actual robot movement.

**Process Integration & Customization:**

- Features convenient modules for integrating various industrial processes.
- Allows for the creation of customized process modules to suit specific needs.

**Industrial Software Platform:**

- Leverages the powerful ZW3D platform for robust and reliable operation.
- Offers dedicated solutions for robot welding, robot spraying, and so on.

**Continuous Improvement:**

- iRobotCAM is constantly evolving to enhance the capabilities and application value of industrial robots.

**Benefits:**

- Streamlined production line design and development.
- Improved accuracy and efficiency of robot programming.
- Reduced risk of errors and rework.
- Enhanced flexibility and customization options.

**Target User:**

- Manufacturers and engineers involved in robot-based production.
- Companies seeking to optimize their robotic processes.
- Developers of industrial automation solutions.

## 2.1 Robot production line design

1. Robot workstation unit equipment library, importing robots, workpieces, tools, additional axes (guide rails, displacement machines, gantry frames, etc.) from different manufacturers, as well as other related work units (floors, safety barriers, control cabinets, etc.).

2. Virtual sensor modeling, cylinder component modeling, claw component modeling, conveyor belt component modeling.

3. Work unit layout, precise control of the position layout and pose relationship of the equipment imported into the work unit.

**Robot Workstation Unit Equipment Library:**

- **Comprehensive Scope:** The library should include a wide range of equipment from various manufacturers, including robots, workpieces, tool hands, additional axes (guide rails, displacement machines, gantry frames), floors, safety barriers, control cabinets, and other relevant components.
- **Standardized Representation:** Equipment should be represented in a consistent and standardized format, including 3D models, kinematic parameters, dynamic properties, and simulation sequence.
- **Interoperability:** Equipment models could compatible with different simulation platforms and software tools to facilitate seamless integration into existing workflows.
- **Parametrization:** Equipment models could parametrically defined, allowing for customization and adaptation to specific application requirements.
- **Search and Filtering:** Efficient search and filtering functionalities could implemented to allow users to quickly find and access the equipment they need.

**Virtual Sensor Modeling:**

- **Sensor Types:** The library including various sensor types commonly used in robot workstations, such as force sensors, and proximity sensors.
- **Customization:** Sensor models could customize the parameters to match the specific properties
- **Integration with Other Models:** Sensor models could seamlessly integrated with other equipment models within the robot workstation, allowing for accurate simulation of sensor-based interactions.

**Benefits:**

- **Reduced Design Time:** Access to a comprehensive library of pre-built equipment models can significantly reduce the time and effort required for robot workstation design.
- **Improved Design Quality:** Virtual sensor modeling allows for early identification and mitigation of potential issues, leading to more robust and reliable robot systems.
- **Enhanced Collaboration:** The library can facilitate collaboration between engineers, designers, and other stakeholders involved in robot workstation development.
- **Cost Optimization:** By reducing design time and improving design quality, the library can significantly contribute to cost optimization in robot workstation projects.

## 2.2 Mechatronics Integration Design with Physical Simulation

Mechatronics integration design, by defining geometric bodies as rigid bodies and collision bodies, realistically restoring the physical properties of geometric bodies such as mass, inertia, friction, material, and collision, creating motion pairs and constraints, sensors and actuators, simulation sequences, and signal adapters, thereby achieving physical simulation of the model.

**Key Aspects of Physical Simulation for Mechatronics Integration Design:**

- **Geometric Body Representation:**
  - ✧ **Rigid Bodies:** Each physical component is modeled as rigid body and collision body, accurately capturing its mass, inertia, and collision properties.
  - ✧ **Material Properties:** Realistic material properties, including friction coefficients, are assigned to each body to simulate their interactions and behaviors.
- **Motion Pairs and Constraints:**
  - ✧ **Motion Pairs:** Define the permitted relative motion between two rigid bodies, encompassing joints, gears, and other kinematic connections.
  - ✧ **Constraints:** Restrict the motion of bodies in specific ways, limiting their movement to prescribed paths or preventing unwanted rotations.
- **Sensors and Actuators:**
  - ✧ **Sensors:** Simulated devices that gather information about the system's state, including position, velocity, and force, providing critical feedback for control algorithms.
  - ✧ **Actuators:** Virtual representation of robot, tool, positioner, cylinder, or other mechanisms that apply forces or torques to manipulate the motion of the system's components.
- **Simulation Sequences:**
  - ✧ Defining the sequence of events that occur during the simulation. This includes specifying initial conditions, control inputs, and IO interactions.
- **Signal Adapters:**

✧ Facilitate data exchange between different components by converting signals between compatible formats, such as translating analog sensor data to digital values for software processing.

**Benefits of Physical Simulation:**

- **Realistic Motion Prediction:** Accurately reflects the physical properties and interactions of the components, providing valuable insights into the system's behavior under various operating conditions.
- **Early Design Verification:** Potential design flaws and unforeseen interactions are identified early in the development process, saving time and resources by avoiding costly physical prototypes.
- **Performance Optimization:** Allows for iterative design refinement and parameter adjustments to improve system performance, efficiency, and robustness.
- **Reduced Development Costs:** Physical simulation can significantly reduce the need for expensive and time-consuming physical prototypes, streamlining the development process.

## 2.3 Robot off-line programming

Robot teaching, performing robot teaching operations in joint space.Cartesian space in base coordinate system, tool coordinate system, etc.Robot interpolation algorithms, including several basic interpolation algorithms such as lines, arcs, and joints.Robot job simulation run, actual animation demonstration process.

**1. Robot Teaching:**

- **Joint space teaching:** Define robot movements by specifying the desired joint angles at each point in the program.
- **Cartesian space teaching:** Teach robot movements directly by guiding the end-effector through the desired path using a virtual teach pendant or other input device.
- **Coordinate system selection:** Specify the coordinate system for teaching, such as base coordinate system, tool coordinate system, or user-defined coordinate systems.

**2. Robot Interpolation Algorithms:**

- **Linear interpolation:** Generate straight-line motions between programmed points.
- **Circular interpolation:** Generate smooth, circular motions between programmed points.
- **Joint interpolation:** Generate coordinated movements of all robot joints to reach the desired end-effector positions.

**3. Robot Operation Simulation and Animation:**

- **Simulate the robot's movement:** Utilize the programmed path and interpolation algorithms to visually simulate the robot's motion and interaction with its environment.
- **Animation demonstrations:** Create realistic animations of the programmed movements to demonstrate the robot's performance and identify potential collisions or issues.

**Additional Features:**

- **Collision detection:** Detect potential collisions between the robot and its environment during simulation, allowing for early identification and correction of potential problems.
- **Program editing and optimization:** Edit and refine the programmed path to improve efficiency, accuracy, and smoothness of robot movements.
- **Program generation:** Generate robot program code based on the simulated program, supporting various robot controller languages.
- **Integration with CAD/CAM software:** Import and export 3D models of robots and workcells from CAD/CAM software for enhanced simulation realism.

**Benefits of Robot Off-line Programming:**

- **Reduced Programming Time:** Develop and test robot programs without needing a physical robot, significantly reducing programming time and effort.
- **Improved Program Accuracy:** Simulate the program beforehand to identify and correct errors before deployment, leading to more accurate and reliable programs.
- **Enhanced Efficiency:** Streamline the robot programming process and optimize robot movements for better performance and efficiency.
- **Reduced Downtime:** Avoid production disruptions by testing and debugging programs offline without impacting real-world operations.
- **Improved Safety:** Identify potential safety hazards through virtual simulation before the robot operates in the real world.

## 2.4 Robot virtual commissioning

Equipped with virtual commissioning and monitoring, capable of synchronizing the actions of virtual robot system workstations with actual robot system workstations; Supports multi machine IO communication simulation, multi robot synchronization, and multi axis robot linkage planning.

**1. Synchronization with Real Robot Systems:**

- Connect the virtual robot system and the actual robot system, enabling synchronized execution of the program in both environments.

**2. Multi-Machine I/O Communication Simulation:**

- Simulate the communication between multiple machines involved in the robot system, including controllers, sensors, and actuators.
- Test and debug the program's interaction with various I/O devices and ensure proper data exchange and communication protocols.

**3. Multi-Robot Synchronization:**

- Simulate the coordinated movement of multiple robots operating in the same environment.
- Analyze potential collisions, coordination issues, and optimize robot paths for smooth and efficient collaboration.

**4. Multi-Axis Robot Linkage Planning:**

- Simulate the synchronized motion of robots with multiple axes, such as articulated arms or gantry robots.
- Verify the program's ability to control complex robot kinematics and ensure accurate and coordinated movements.

**Benefits of Robot Virtual Debugging:**

- **Early Identification of Errors:** Identify and correct program errors in real-time during simulation, significantly reducing debugging time and effort.
- **Enhanced Program Robustness:** Test the program under various simulated conditions and scenarios to ensure its robustness and adaptability in real-world situations.
- **Improved System Integration:** Verify the interaction and communication between robots and other machines within the system, facilitating smoother integration and deployment.
- **Reduced Downtime:** Eliminate the need for repeated testing and debugging on the physical robot, minimizing downtime and production disruptions.
- **Enhanced Safety:** Identify and mitigate potential safety hazards in the virtual environment before the robot operates in the real world, ensuring a safe working environment.

# Chapter 3: iRobotCAM User Interface

The start interface allows users to create new projects or open existing ones. Additionally, it provides options to activate iRobotCAM. (Figure 3-1)

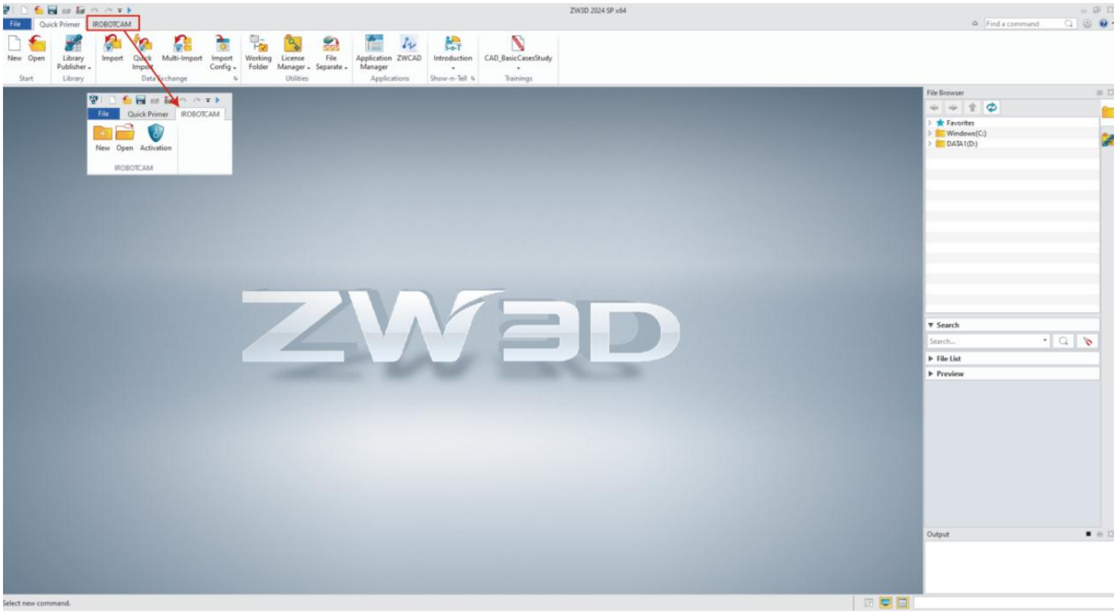


**Figure 3–1: iRobotCAM User Interface**

## 3.0 New project file

Under the "IROBOTCAM" module, click "New" to "Create a new project file". (Figure 3-2)

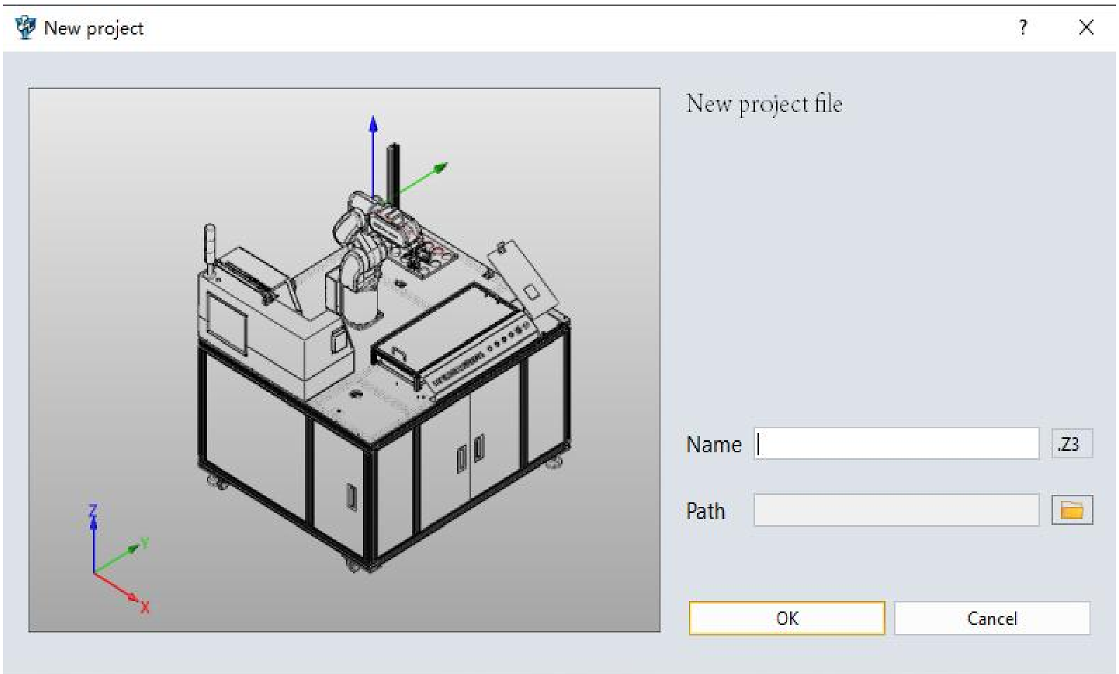


**Figure 3–2: New Project**

## 3.1 Component Library

On the "IROBOTCAM" toolbar, in the leftmost first position. (Figure 3-3)

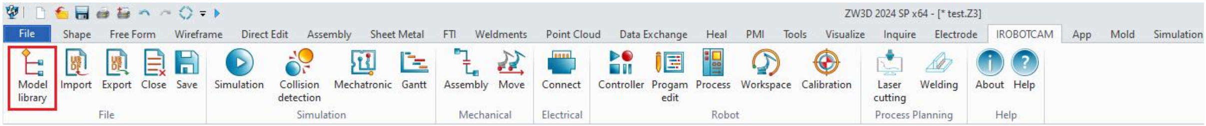



**Figure 3–3: Component library location**

The Model library offers a vast selection of pre-built robot models, categorized for ease of access. Users can browse through the library, and readily incorporate desired components into their projects. (Figure 3-4)


**Figure 3–4: Model library**

## 3.2 Mechatronic

This section constitutes the heart of iRobotCAM, encompassing a diverse set of tools for building realistic and functional robot models. (Figure 3-5)

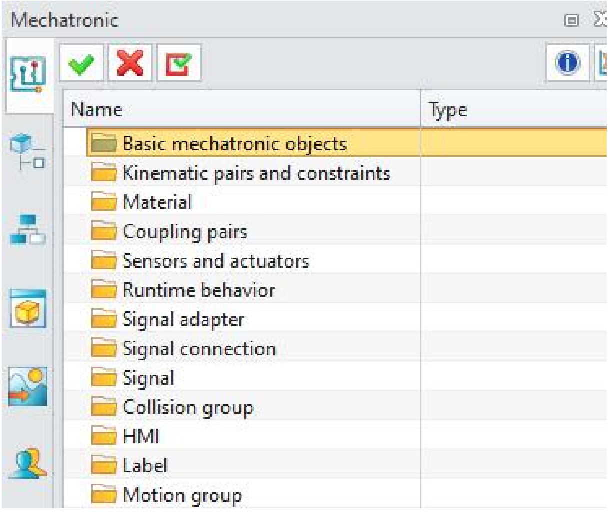



**Figure 3–5: Basic Mechatronic Objects**

- Rigid body: It can make geometric objects move under the control of physical systems, and any geometric object can only be affected by gravity or other forces if a rigid body component is added.
- Physical property: make the rigid body have a mass effect and respond to forces. Collision response: Collision with the environment or other objects during task processing. Support perception: Objects support being perceived. (Figure 3-6)

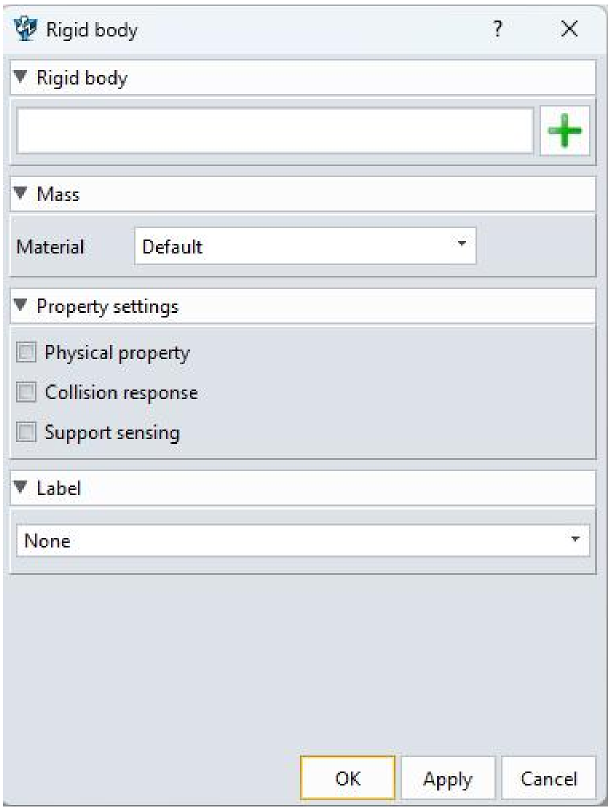



**Figure 3-6：Rigid Body**

- Collision: Collision is a type of physical component that can only trigger collisions when added to geometric objects along with rigid bodies. It defines the collision mode between components and other colliding components. In physical simulation, rigid bodies without colliding bodies pass through each other.
- Collision shape: Convex hull calculation speed is fast, can envelop, and accuracy may be slightly poor. Convex decomposition is decomposed into many small envelopes to fit the model, and the time for convex decomposition is slow but very accurate.
- Collision material: defines the basic characteristics of the material in collision.(Figure 3-7)

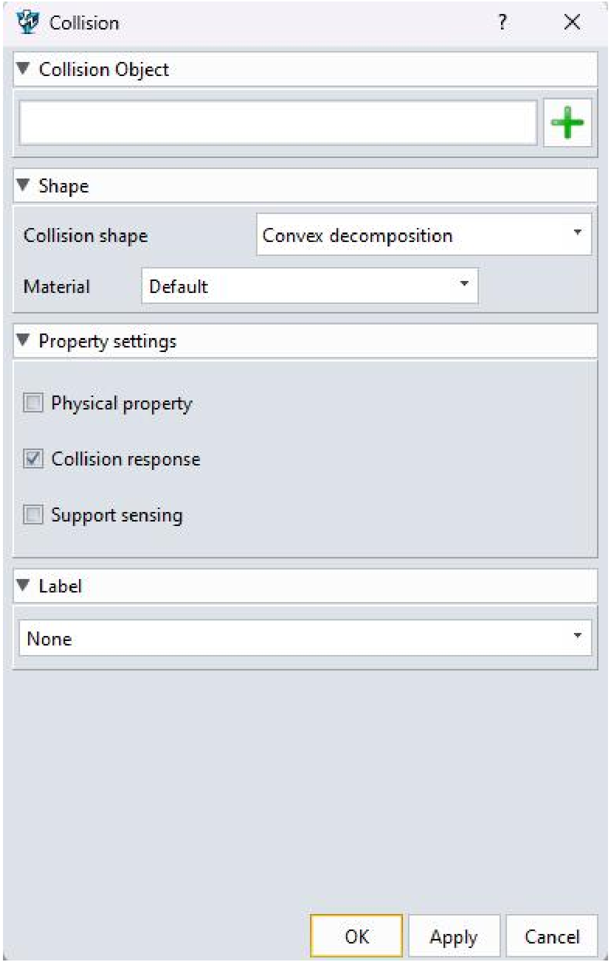



**Figure 3-7: Collision**

- Revolute joint: Establish a motion pair between two objects, allowing a degree of freedom to rotate around a certain axis.
- Parent node: Select another rigid body connected to the connector.
- Child node: Select the rigid body that needs to add hinge constraints.
- Specify axis vector: Specify rotation axis, starting angle: The angle of the connector relative to the basic component before the simulation starts.(Figure 3-8)

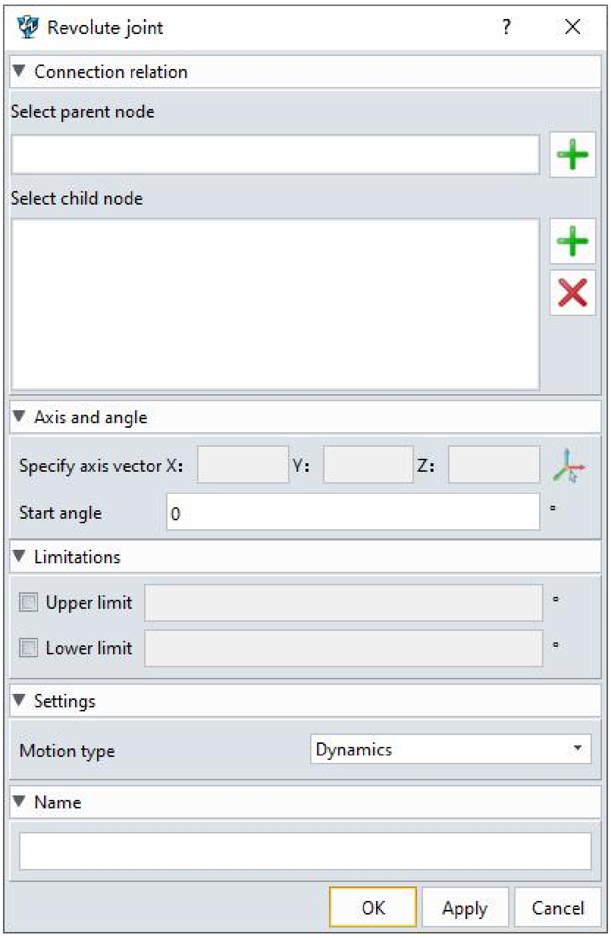



**Figure 3-8: Revolute joint**

- Prismatic joint: A joint consisting of two components that can only move relative to each other in a certain direction, and has one degree of freedom for translation.
- Offset: The distance between the connector and the basic component before the simulation starts.(Figure 3-9)

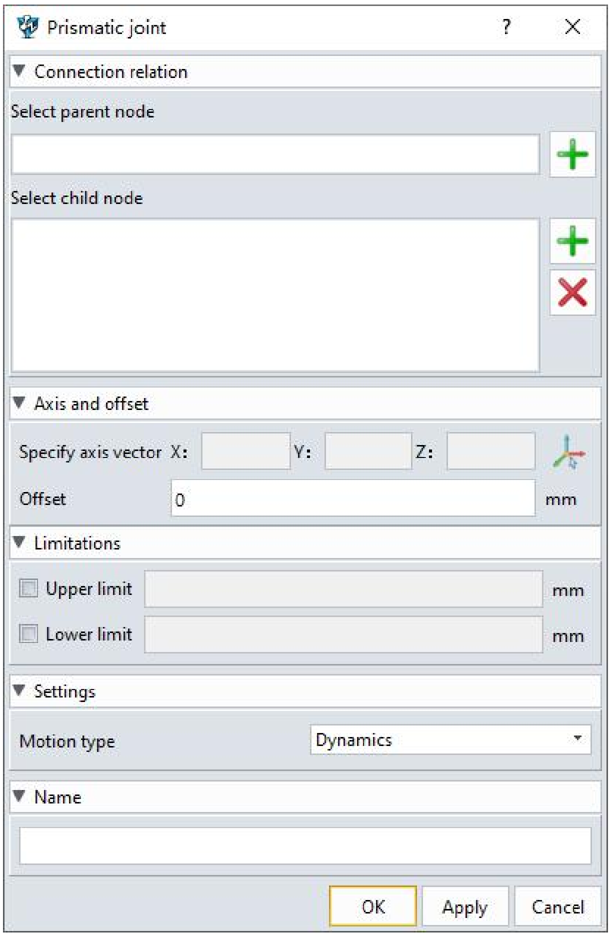



**Figure 3-9: Prismatic joint**

- Materials:The input friction coefficient and recovery coefficient of collision materials enable the object to have frictional force, which can more vividly simulate the simulation form of the object. (Figure 3-10)

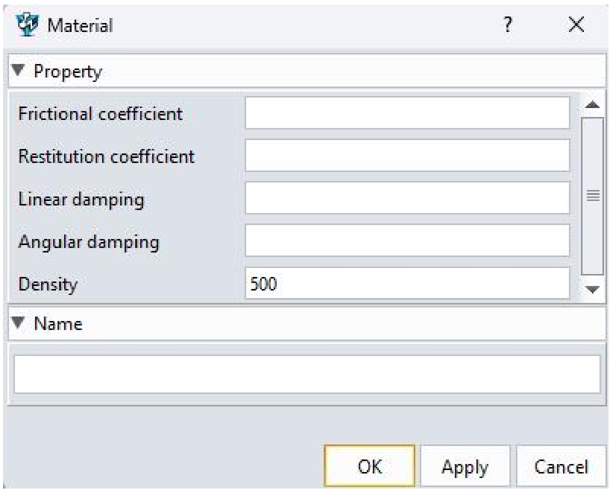



**Figure 3-10: Materia**

- Sensor: Use collision sensors to collect collision events. The size of the collision shape,length, height, and width depend on the type of collision shape.(Figure 3-11)

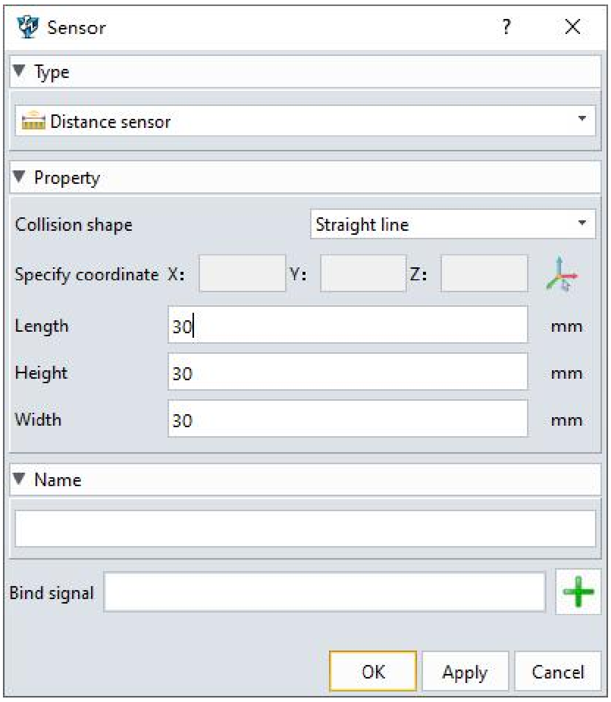



**Figure 3-11: Sennor**

- Transfer plane: It is a physical property that converts the selected plane into a conveyor belt.
- Specify vector: Specify the transmission direction of the transfer plane.
- Parallel: specifies the speed magnitude in the transmission direction.
- Vertical: Specify the speed magnitude perpendicular to the transmission direction.(Figure 3-12)

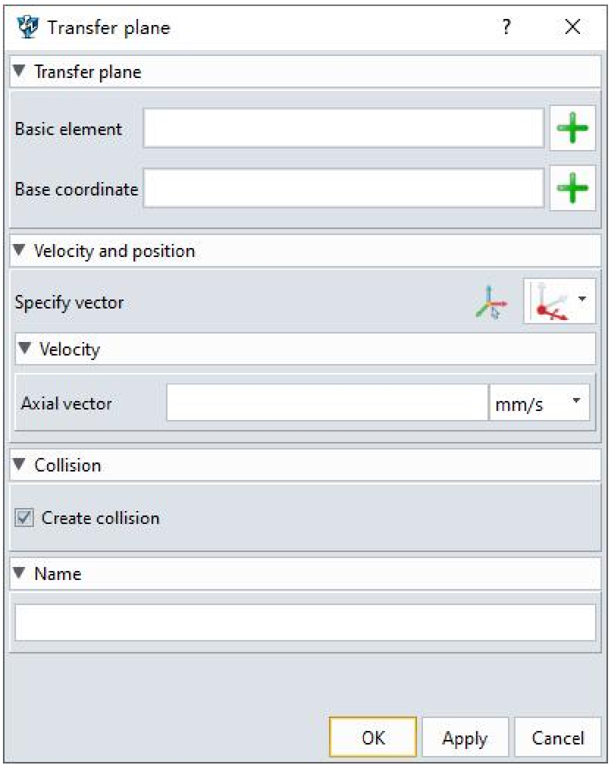



**Figure 3-12: Trasfer plance**

- Position control: Command to create an actuator that drives an axis defined by a kinematic pair to move to a predetermined position at a predetermined constant speed.
- Axis type-cylinder-angle path option: used to define the rotation scheme of the axis kinematic pair.
- Target: Specify a target location.
- Velocity: Specify a constant speed.
- Base marker: composed of the robot base point and coordinate position, it is the foundation of other coordinate systems of the robot. (Figure 3-13)

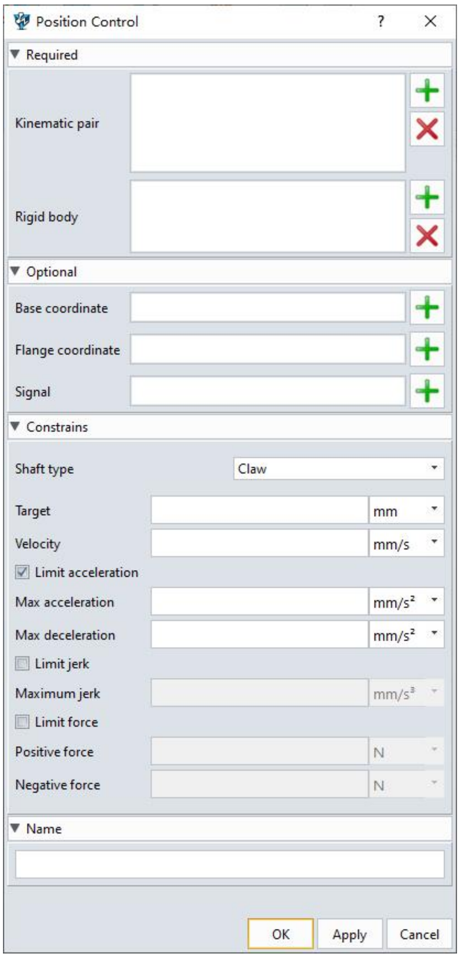



**Figure 3-13: Position control**

Add joint kinematic pairs, base coordinates, flange coordinates, robot parameters, etc. to create a robot. (Figure 3-14)

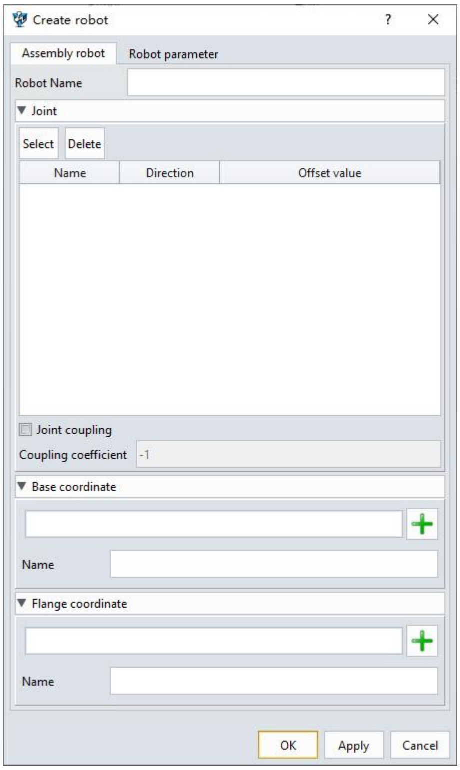



**Figure 3-14: Create robot**

Select joint positions, add external data, maximum acceleration, maximum acceleration, base coordinates, and tool center coordinates to create robot tools. (Figure 3-15)

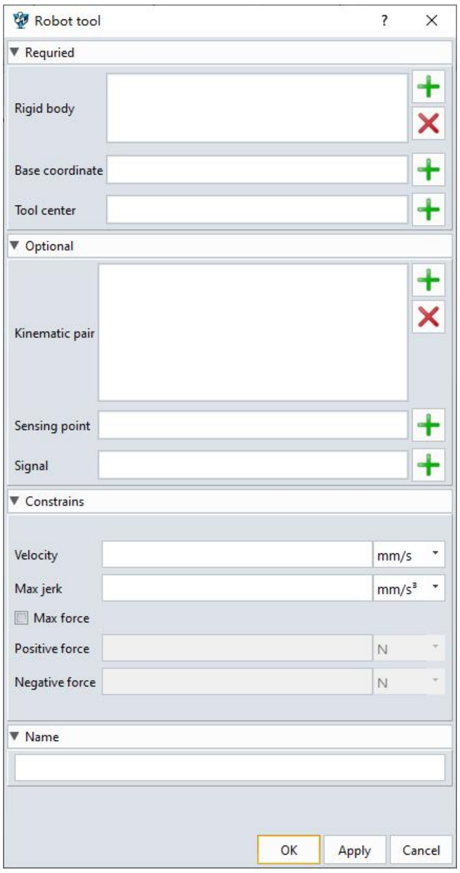


**Figure 3-15: Robot tool parameter interface**

Add motion array units, select basic objects, array elements, add speed, and simulate conveyor belt operation. (Figure 3-16)

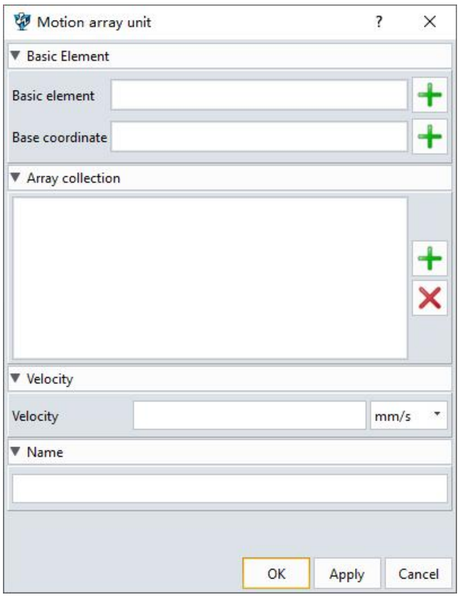



**Figure 3-16: Motion array unit parameter interface**

Add an external axis to allow the robot to move on it by selecting the joints, base, base coordinates, and flange coordinates. (Figure 3-17)

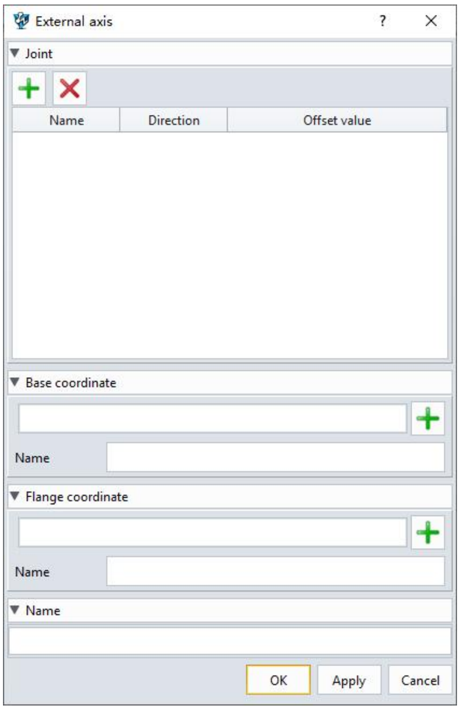



**Figure 3-17: External axis**

Add joint positions, select kinematic pairs, input type and speed, and establish internal signals. (Figure 3-18)

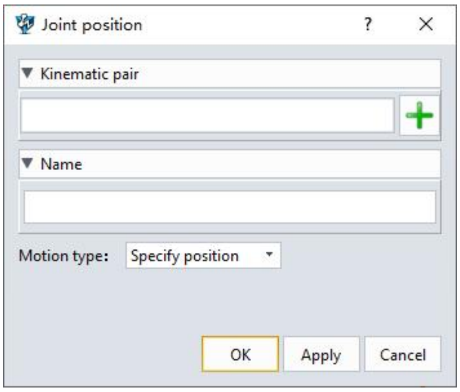



**Figure 3-18: Joint position**

Material production unit: Use materials to create multiple objects with the same appearance and attributes at specific time intervals. (Figure 3-19)

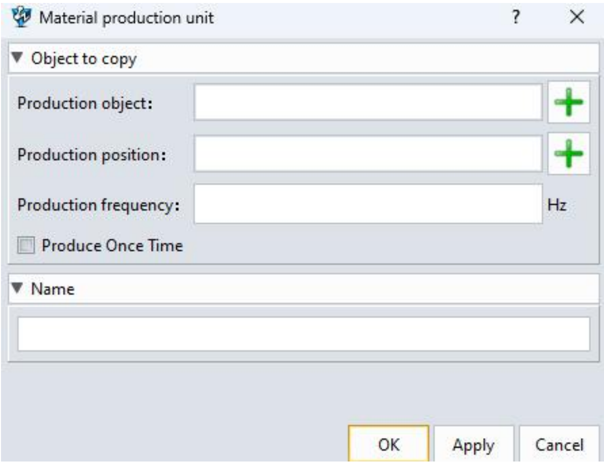



**Figure 3-19: Material production unit**

Simulation sequence: It is a shortcut for logic control and simulation verification, which can associate various actuators with sensors to achieve logical control. (Figure 3-20)

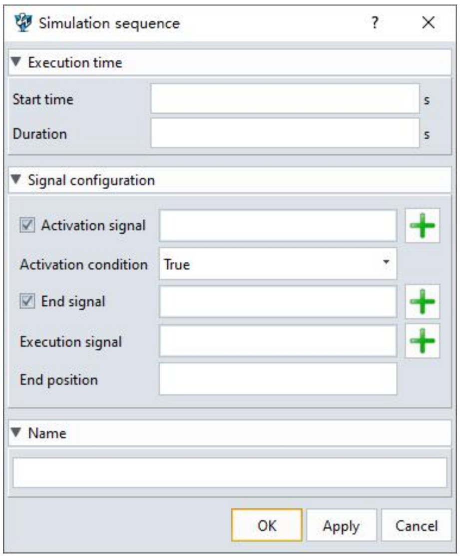


**Figure 3-20: Simulation sequence**

- Signal adapter: Encapsulates runtime formulas and signals using its commands, which are included under the "Mechatronic Modeling" window signals, creating signal objects that can be used to connect external signals.
- Formula: Assign the formula displayed in the formula box to the selected signal. (Figure 3-21)

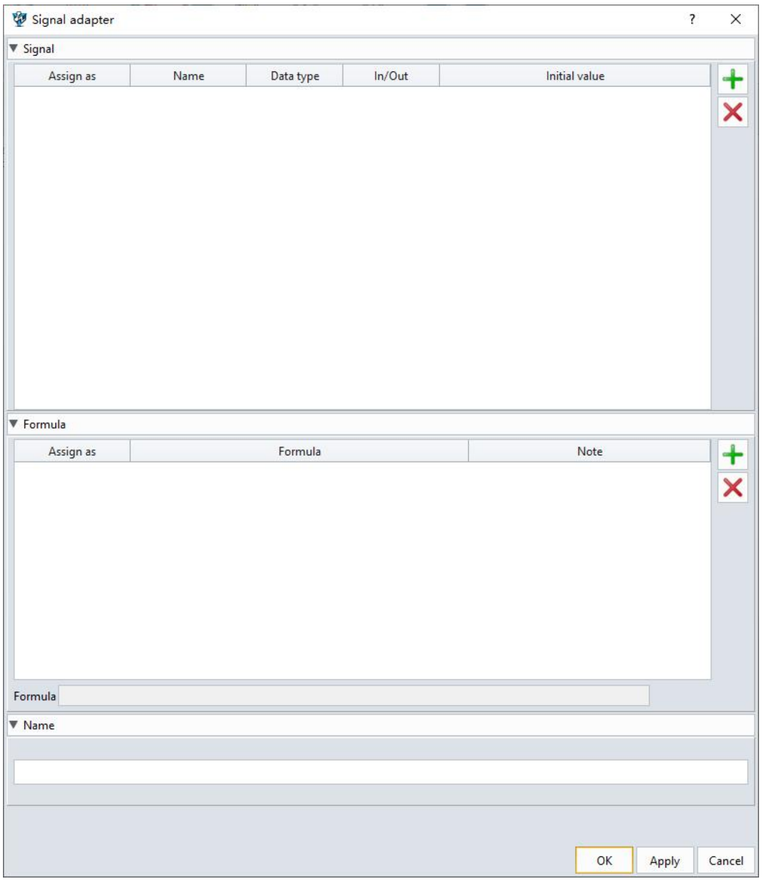



**Figure 3-21: Signal adapter**

Signal connection: Connect internal signals with external signals, and control the internal signals through external signals on external devices to view the simulation effect of the robot. (Figure 3-22)

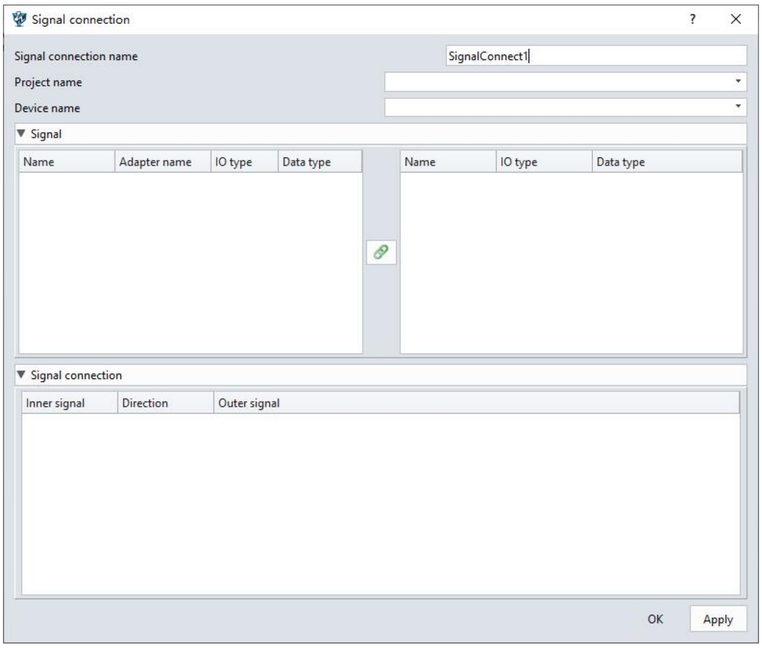



**Figure 3-22: Signal connection**

Signal: Divided into output signal and input signal, creating different signals to generate different simulation motion forms for the robot. (Figure 3-23)

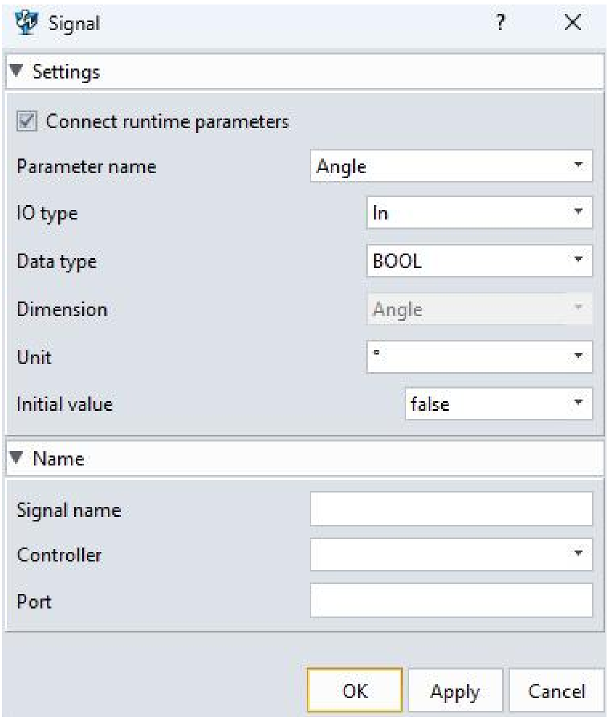



**Figure 3-23: Signal**

- Label: By adding labels, parent and child assemblies can be assembled.
- Parent Label: Select the parent label from the label type drop-down list, and select the corresponding rigid body for the parent assembly and child assembly respectively.(Figure 3-24)

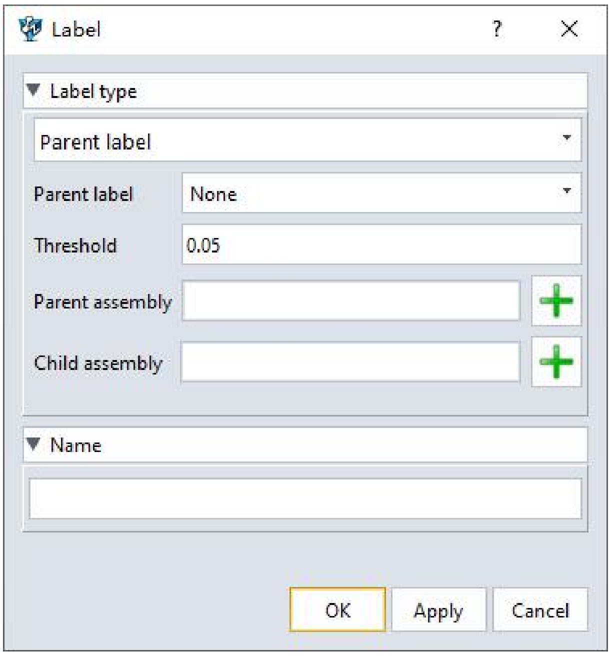



**Figure 3-24: Parent label**

Child Label: Select a child label under the label type, and select the parent label that has already been created from the drop-down list in the parent label box.(Figure 3-25)

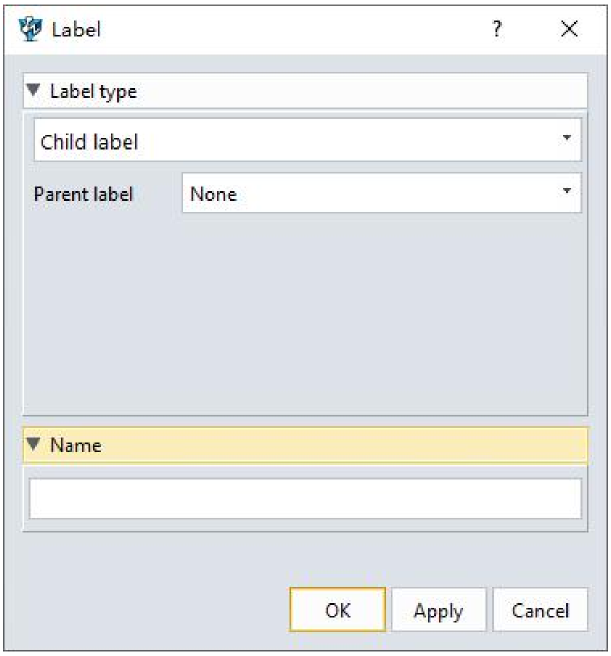



**Figure 3-25: Child Label**

## 3.3 Gantt chart

After adding the simulation sequence, perform simulation verification, and the real-time sequence time will be displayed in the Gantt chart. (Figure 3-26)

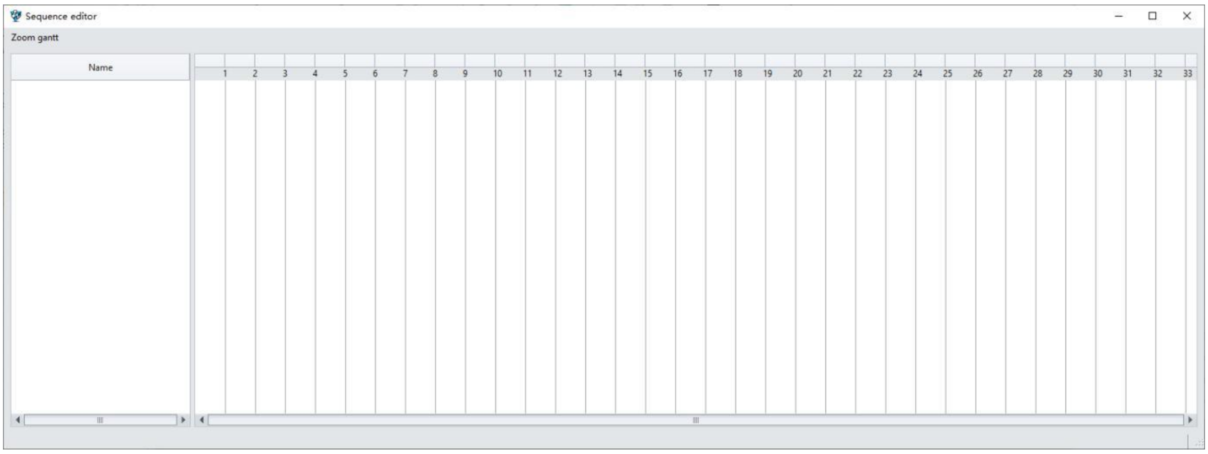



**Figure 3-26: Gant chart**

## 3.4 Assembly relationship

Add assembly relationships, select child devices to install on a certain device, and adjust the posture to ensure a tight fit between the sub devices and the device. (Figure 3-27)

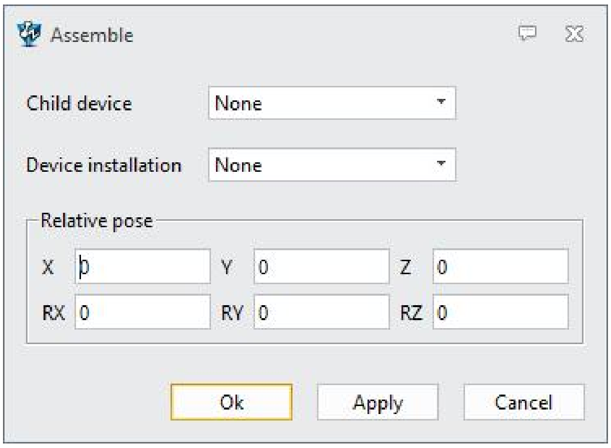



**Figure 3-27: Assembly relationship**

## 3.5 Teach programming

Manipulate the right angle space and joint space of the robot based on a 3D model, and add marker points. (Figure 3-28)

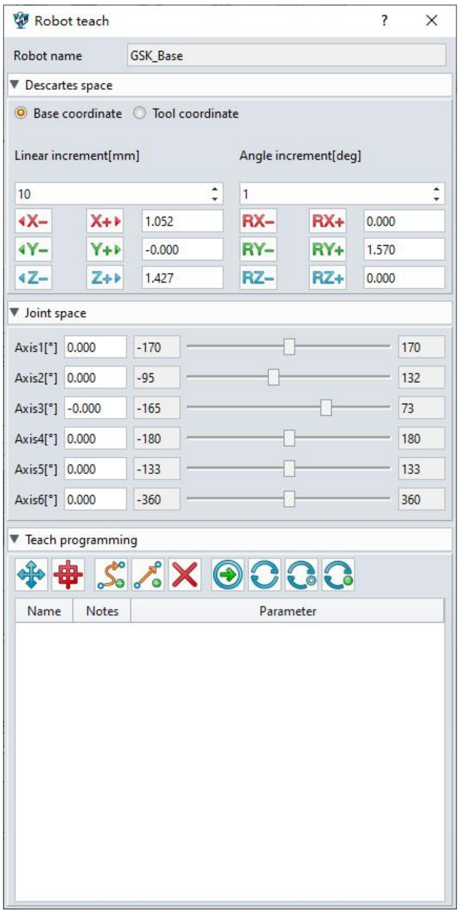



**Figure 3-28: Teach**

## 3.6 Collision detection

By adding collision groups and then collision pairs, you can see collision detection between robots and objects, which is highlighted. (Figure 3-29)

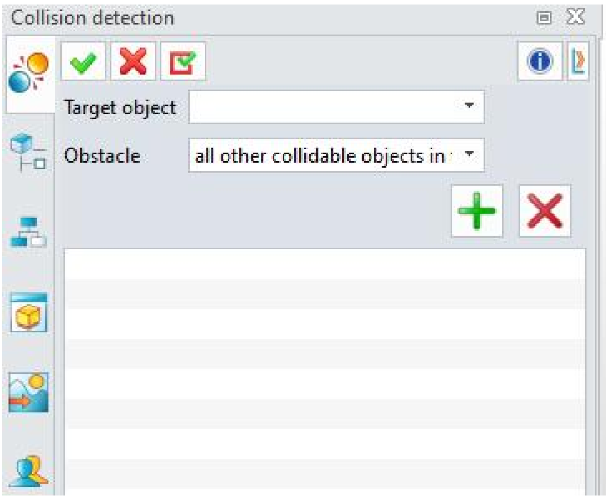


**Figure 3-29: Collision Detection**

## 3.7 Programming

In the "Programming" window, add a JOB program, add program program1, and then right-click to edit the program. (Figure 3-30)

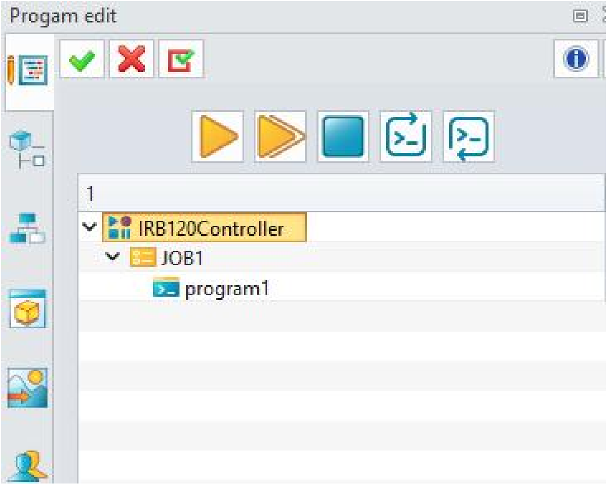



**Figure 3-30: Programming**

In the "Program Parameters" window, perform the add command. (Figure 3-31)

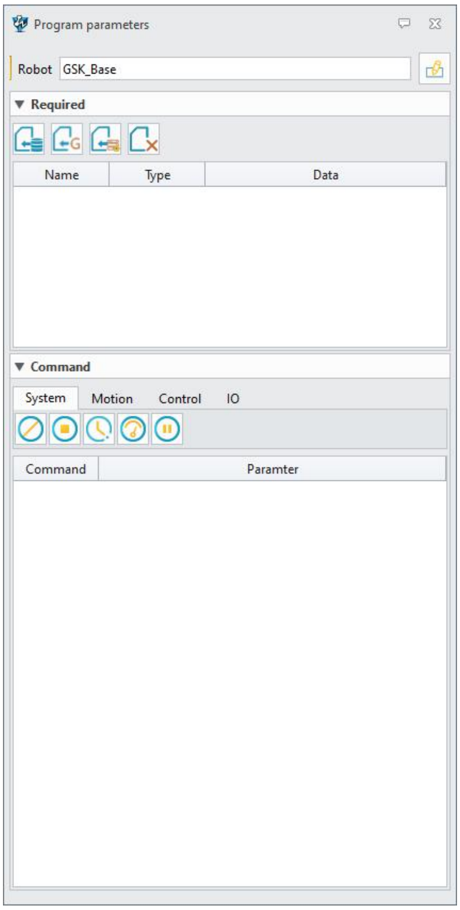


**Figure 3-31: Program Parameters**

Control robot movement by adding instructions. Program instructions are classified as follows:

**System instructions :**

1. NOP (empty instruction, no action taken)
2. END (program end instruction, program stops when it appears, and even if there are instructions remaining, it will no longer run)
3. WAIT (system wait instruction, unit: ms)
4. SPEED
5. PAUSE (system stop instruction, when the program runs to this instruction, the system stops and no longer runs).

(Figure 3-32)

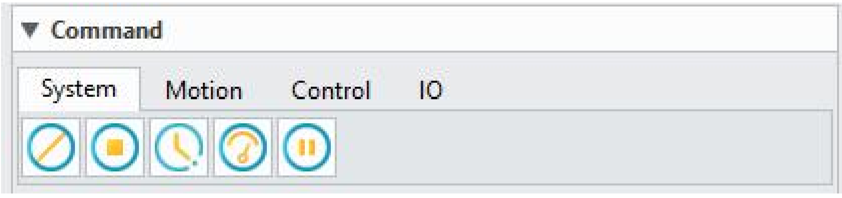



**Figure 3-32: System instructions**

**Motion commands :**MOVJ, MOVL, MOVA, MOVC, etc. (Figure 3-33)

Attention: Only one motion command can be used for each loc point, otherwise it may cause robot motion trajectory errors; If the same point needs to be used multiple times, please mark it multiple times with different names.

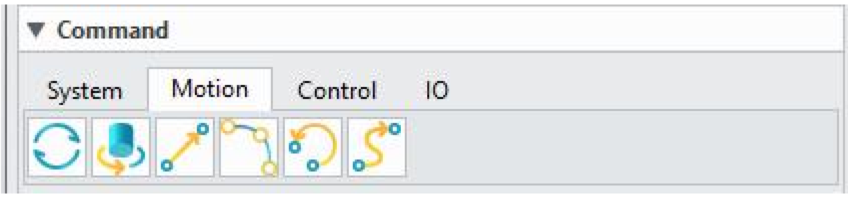



**Figure 3-33: Motion commands**

(1) HOME:

- **Function:** Use joint interpolation motion to return the robot to zero point;
- **Format:** Simply click the HOME button;
- **Explanation:** When using HOME, it can only be added directly, and motion instructions cannot be added by inserting instructions;

(2) MOVJ:

- **Function:** Move to the target point through joint interpolation;
- **Format:** MOVJ {LOC (target point), VEL [] (velocity), ACC [] (acceleration), JERK [] (acceleration)};

- **Explanation:** The unit of VEL is °/s; ACC and JERK both use percentage form, and the input range is a positive number. If it is less than or equal to 0, an error will be reported; (Figure 3-34)

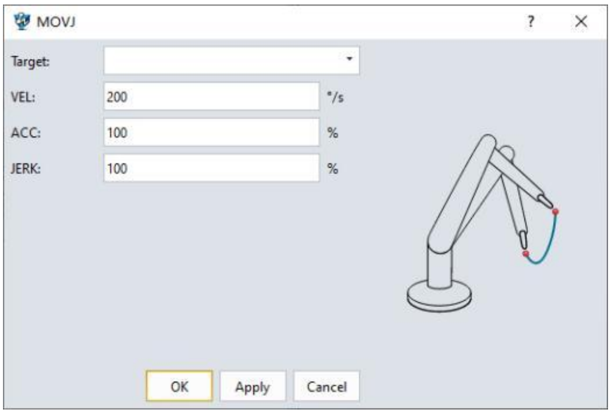



**Figure 3-34: MOVJ**

（3）MOVL：

- **Function:** Move to the target point through linear interpolation;
- **Format:** MOVL {LOC (target point), VEL [] (velocity), ACC [] (acceleration), JERK [] (acceleration)};
- **Explanation:** The unit of VEL is m/s; ACC and JERK both use percentage form, and the input range is a positive number. If it is less than or equal to 0, an error will be reported; (Figure 3-35)

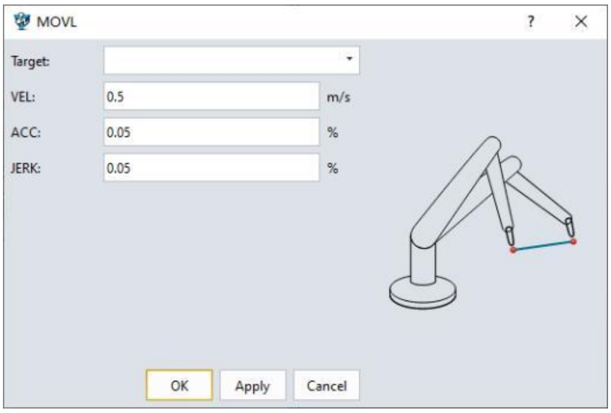



**Figure 3-35: MOVL**

（4）MOVA：

- **Function:** Circular interpolation motion mode passes through the middle point to the target point;
- **Format:** MOVA {LOC1 (target point), LOC2 (intermediate point), VEL [] (velocity), ACC [] (acceleration), JERK [] (acceleration)};
- **Explanation:** The unit of VEL is m/s; ACC and JERK both use percentage form, and the input range is a positive number. If it is less than or equal to 0, an error will be reported; (Figure 3-36)

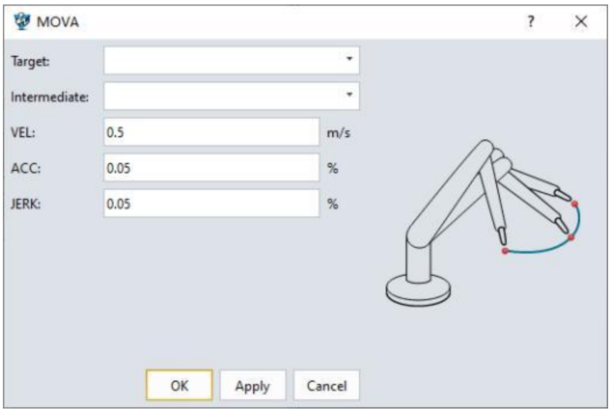



**Figure 3-36: MOVA**

（5）MOVC：

- **Function:** Move in a circular interpolation motion along the current point, target point, and intermediate point, forming a circle, and then return to the current position. When moving, it first passes through the target point and then passes through the intermediate point;
- **Format:** MOVC {LOC1 (target point), LOC2 (intermediate point), VEL [] (velocity), ACC [] (acceleration), JERK [] (acceleration)};
- **Explanation:** The unit of VEL is m/s; ACC and JERK both use percentage form, and the input range is a positive number. If it is less than or equal to 0, an error will be reported; (Figure 3-37)

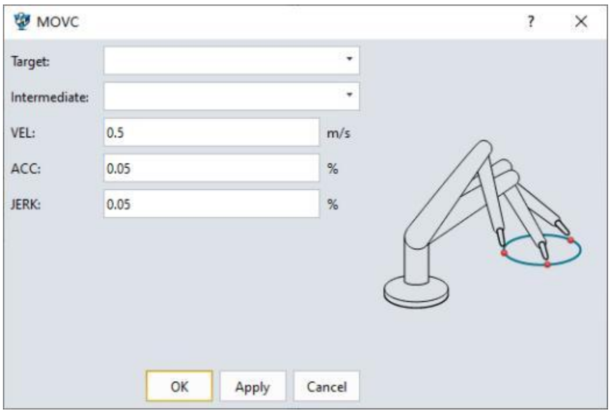



**Figure 3-37: MOVC**

（6）MOVS：

- **Function:** Move to the target point through spline interpolation motion;
- **Format:** MOVS {LOC1 (target point), LOC2 (passing point), VEL [] (velocity), ACC [] (acceleration), JERK [] (acceleration)}; (Figure 3-38)

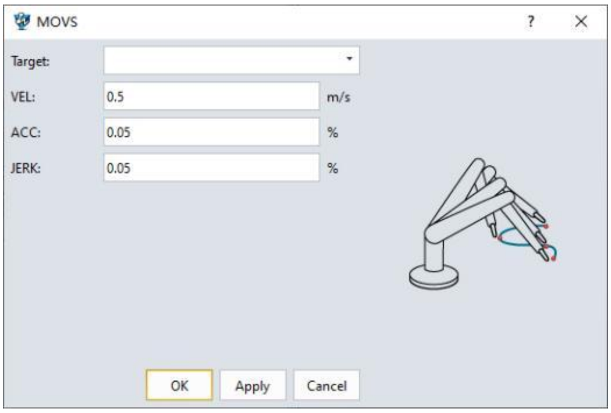



**Figure 3-38: MOVS**

**Control instructions** (Figure 3-39)**:**

- FLYBY # ON :smooth transition on instruction;
- FLYBY # OFF :smooth transition off instruction;
- LOOP :loop start instruction, the number of times the partial loop of the END_LOOP instruction is set after this instruction, which is used in combination with END_LOOP;
- END_LOOP :loop bundle instruction, paired with LOOP;
- WHIL : While loop start instruction, parameter is loop start condition, is a conditional judgment statement, used in conjunction with END_WHILE;
- END_ WHILE :WHILE loop content end instruction, used in conjunction with WHILE;
- IF : select structure instruction, parameter is a conditional statement, instruction after IF is an instruction that satisfies the condition and can be executed until ELSE,ELSE_IF, or END_IF is encountered to end the branch content;
- ELSE : another selection branch of IF or ELSE_IF statement, with END_IF as the end of the branch statement;
- ELSE_ IF : used for branch start instructions in multi branch structures, with ELSE as the end of the branch statement;
- END_ IF: used for the end of all conditional statements, paired with IF;
- LABEL : constructs a label at the specified position, which can be used together with the GOTO instruction to adjust the instruction process;
- GOTO : jumps to the instruction position marked by LABEL to start execution, generally does not jump to the previous label, which can easily form a dead loop;

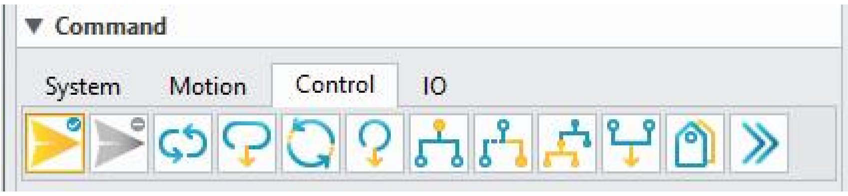


**Figure 3-39: Control instructions**

**The IO instructions** (Figure 3-40):

- DIN :digital input, reading the signal of the specified port number into the BOOL type variable;
- DOUT : digital output, making different responses to the controller based on the read signal;

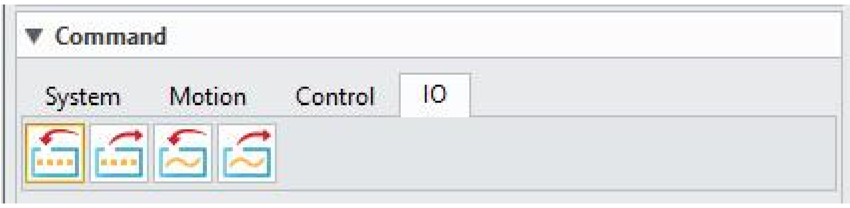



**Figure 3-40: IO instructions**

## 3.8 URDF Import and Export

URDF is a format based on the XML specification used to describe the structure of robots, mainly used to define the structure of robots, including the components of robots and the relationships between them. URDF files can be used for robot simulation, visualization, and analysis in ROS (Robot Operating System) to describe the relationships between robot components and their relationships.(Figure 3-41)

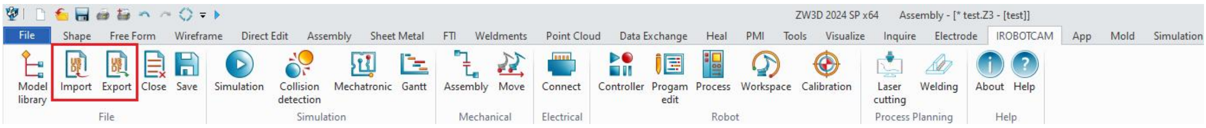



Figure 3-41: Import and Export

Import URDF: Select the path where the file is stored and click OK.(Figure 3-42)

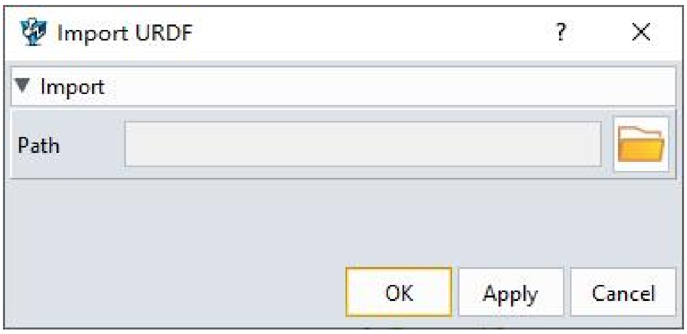



Figure 3-42: Import URDF

Export URDF: In the device bar, click the "+" button, select the robot to export in the assembly relationship bar, select the path to store in the path bar, and click OK.(Figure 3-43)


Figure 3-43: Export URDF

## 3.9 Workspace

In the IROBOTCAM status bar, click the workspace button.(see Figure 3-44)

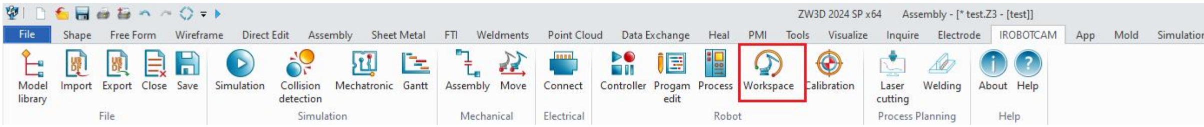



Figure 3-44: Workspace

In the mechatronic bar, click on the collision group, and in the collision group window, create collision groups for the robot's joints. The collision body of each joint of the robot needs to be checked to support sensing. Click on the workspace, select the controller in the workspace parameters, and adjust the sensing range that the detection robot can collide with by modifying the spatial point density.(Figure 3-45)

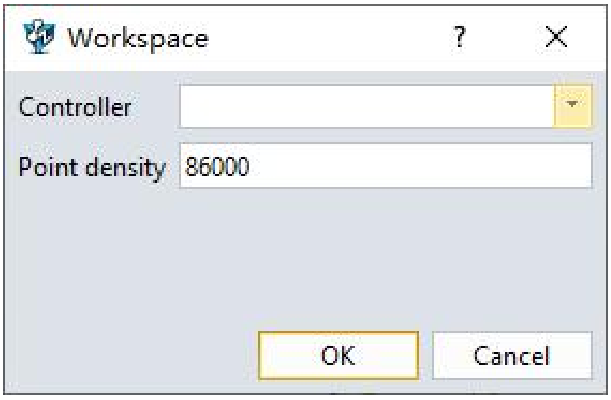



Figure 3-45: Workspace window

## 3.10 Euler angle

- Euler angle is a set of three independent angular parameters used to uniquely determine the position of a fixed point rotating rigid body.(Figure 3-46)

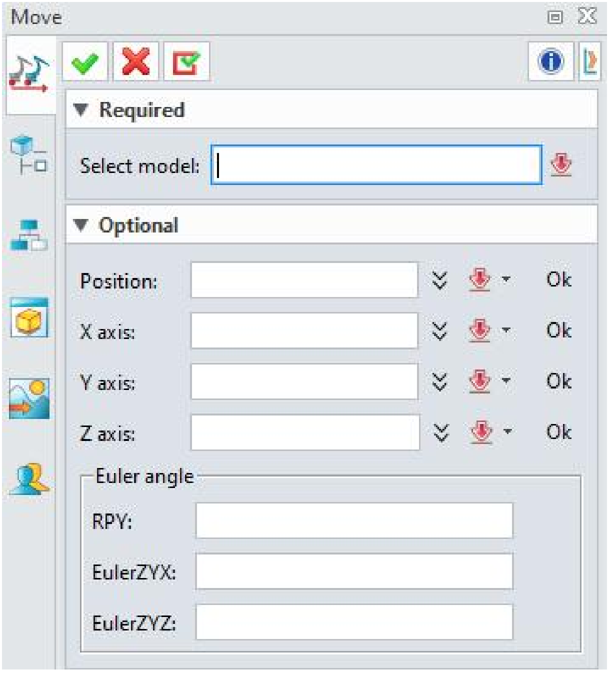



Figure 3-46: Euler angle

## 3.11 Connect module

- By adding devices, accessing the device server, and controlling internal signals through external signals from the server, communication functions are sequentially achieved. (Figure 3-47)

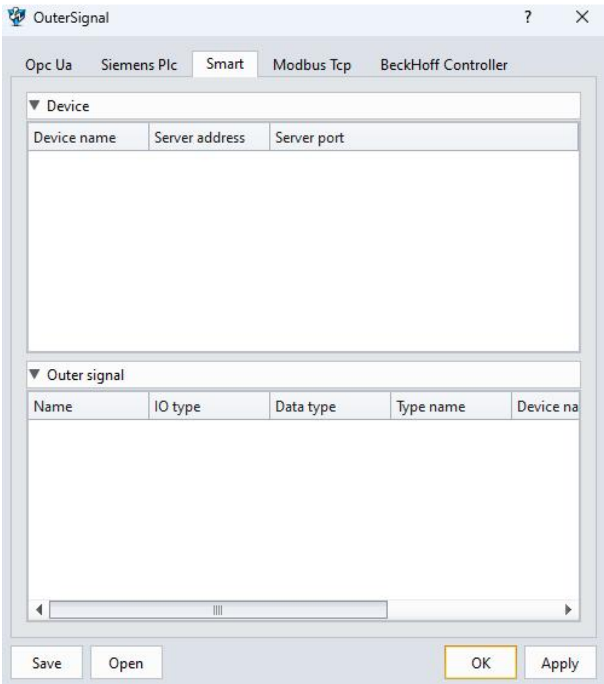


Figure 3-47: Communication Module

## 3.12 Simulation verification

- After programming, simulation verification can be conducted to demonstrate the robot's motion trajectory.
- Switch to real-time mode for mechatronic motion simulation. (Figure 3-48)

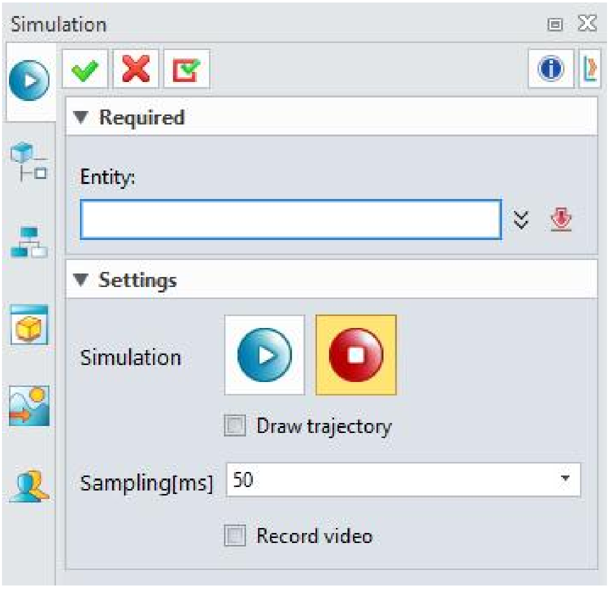
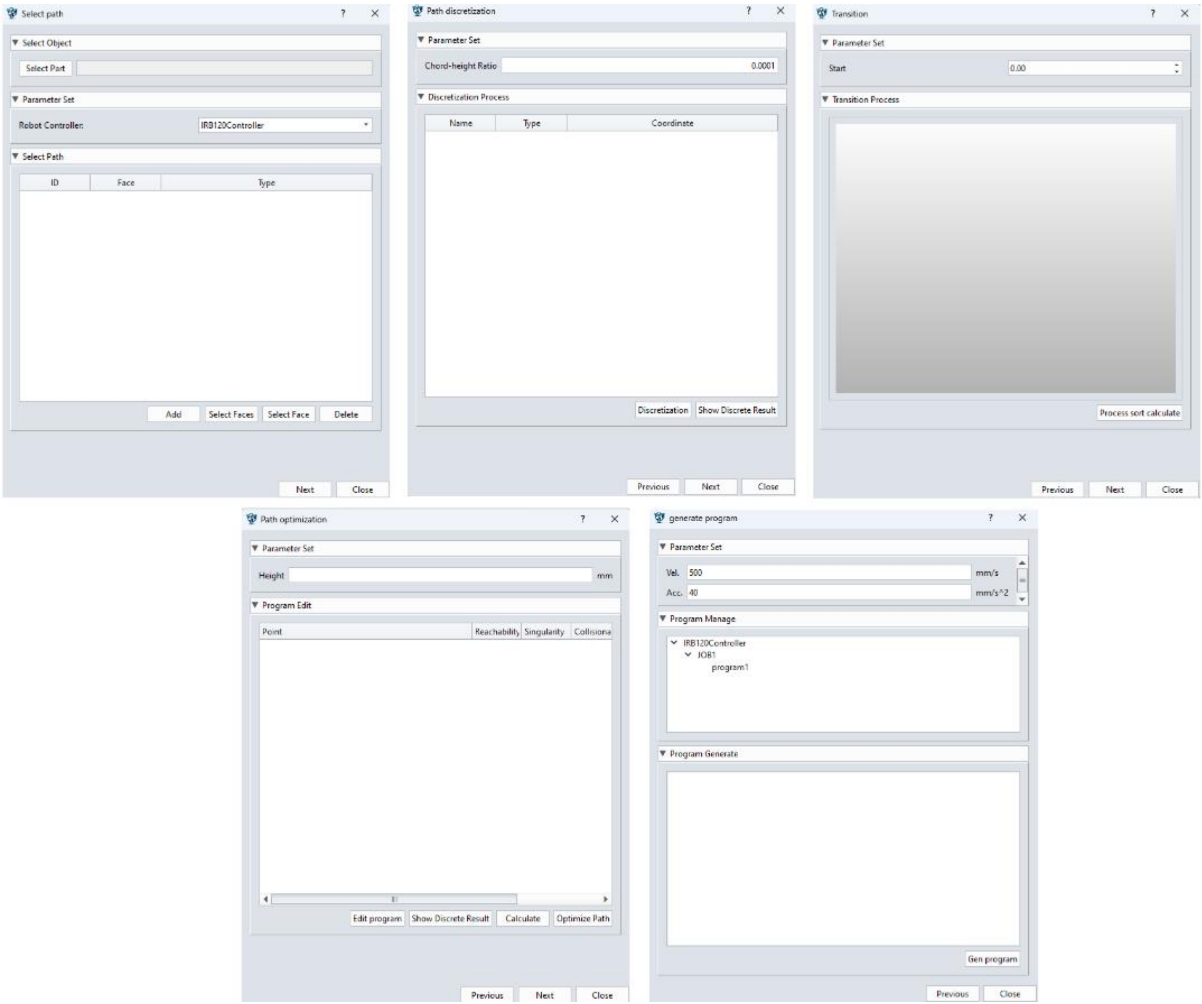



Figure 3-48: Simulation verification

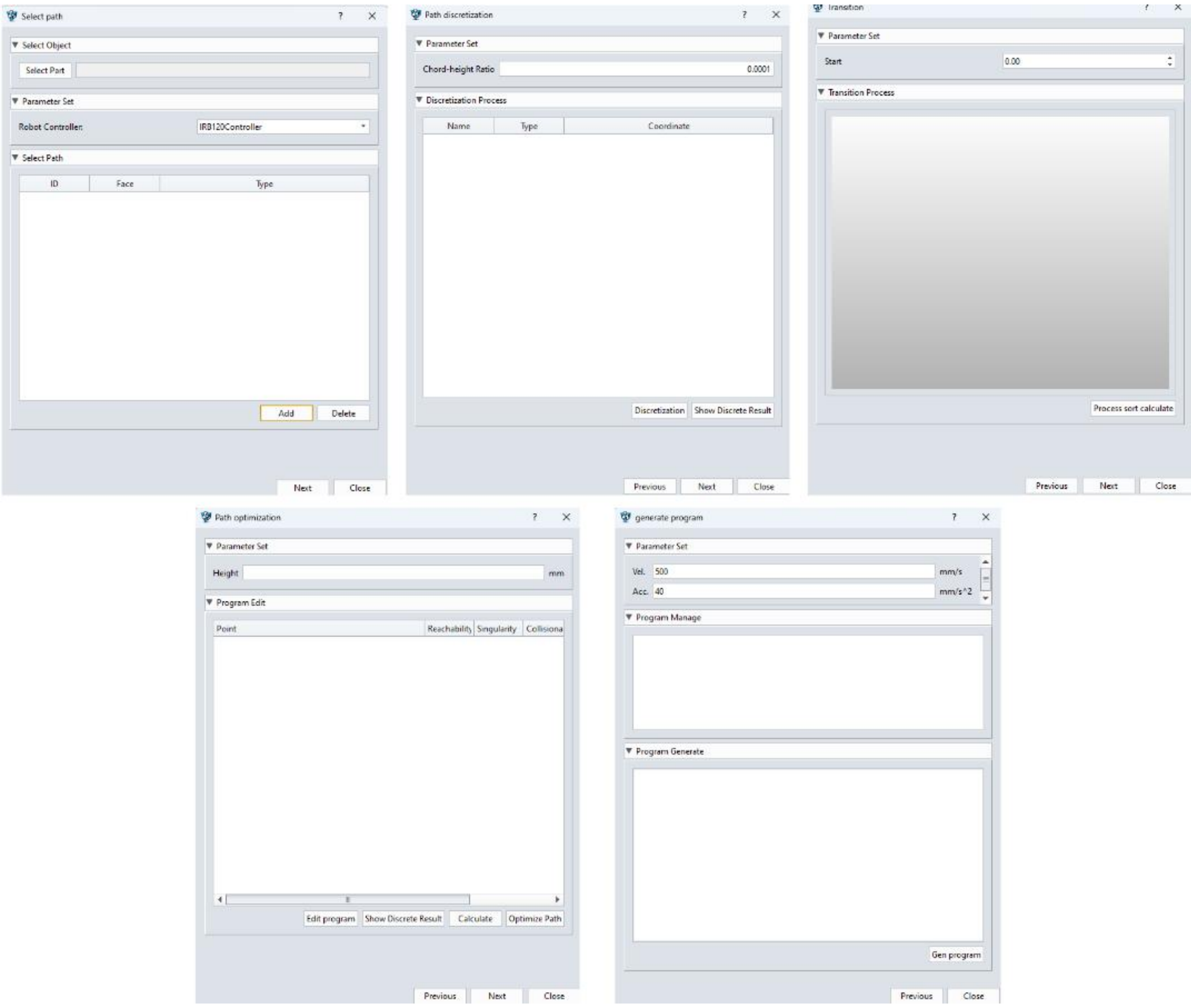
## 3.13 Laser cutting

- The laser cutting function is a functional module composed of multiple windows by clicking "Next".
- Step sequence: Select path - Discretize path - Sort transition points - Optimize path - Generate program.
- Specific operation steps are introduced with reference to application cases.




## 3.14 Welding

- The welding function is a functional module composed of multiple windows by clicking "Next".
- Step sequence: Select path - Discretize path - Sort transition points - Optimize path - Generate program.
- Specific operation steps are introduced with reference to application cases.




## 3.15 Screen recording function

- Click on the simulation button in the "IROBOTCAM" tab to open the "Simulation Control" interface.
- Check to enable video recording, and then click on the start simulation button on the left side of the "Simulation Control" window to start recording.
- Wait for the machine to run the entire process, and then finally click on the end simulation button on the right side of the "Simulation Control" window to end recording.
- The video recording is complete. (Figure 3-49)

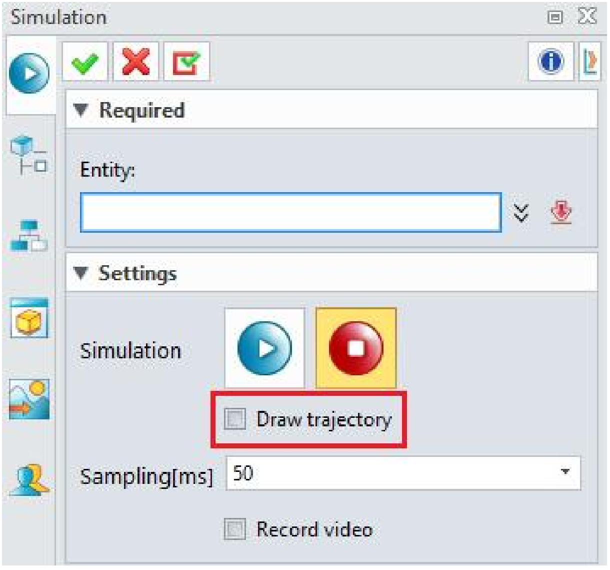



**Figure 3-49: Screen recording function**

Additional Tips:

- Video recording storage location: Open the installation location of ZW3D 2024 software, pull up to see the video recording file, click on the video MP4 file, and you will see the entire process of machine operation(see Figure 3-50).
- If the file for the screen recording function does not exist, you need to run the ZW3D with administrator rights.

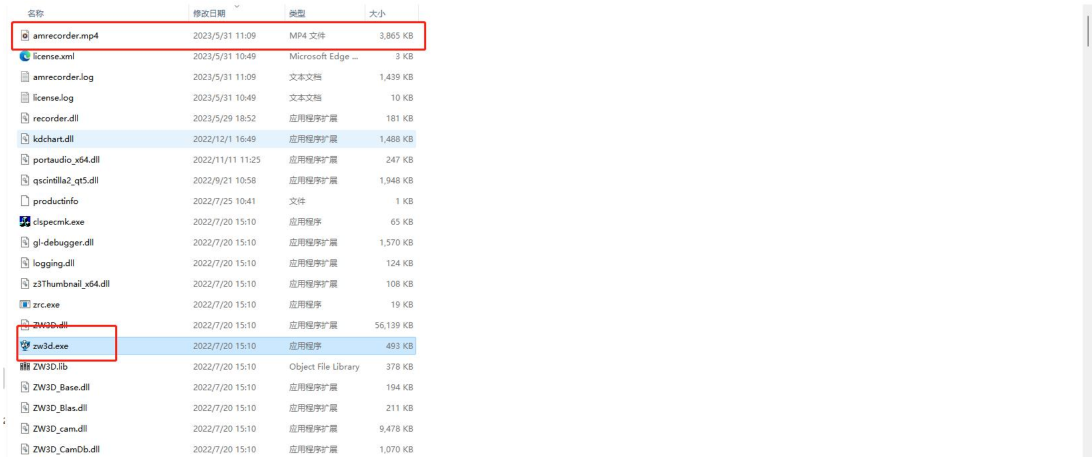



**Figure 3-50: Video recording storage location**

# Chapter 4: Notes and Save

IROBOTCAM User Interface: Notes and Storage

## 1. Shortcut Keys:

- Some shortcut keys are not available within the IROBOTCAM tab. This likely includes common shortcuts like Ctrl+S (save), Ctrl+Z (undo), Ctrl+X (cut), Ctrl+C (copy), and Ctrl+V (paste).

## 2. Storage:

- Encourage users to explore other methods of saving file data in order to gain a more comprehensive understanding of IROBOTCAM's storage capabilities.

Notes:

- In the IROBOTCAM module, this save button saves the content of  the mechatronic.
- This save button does not save the file. (Figure 4-1)

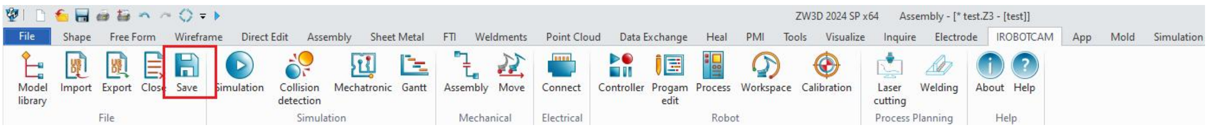



**Figure 4-1: Save**

Click Save and a save progress bar will pop up. After saving, the output box on the right will display the words "Successfully saved mechatronic data". (Figure 4-2)

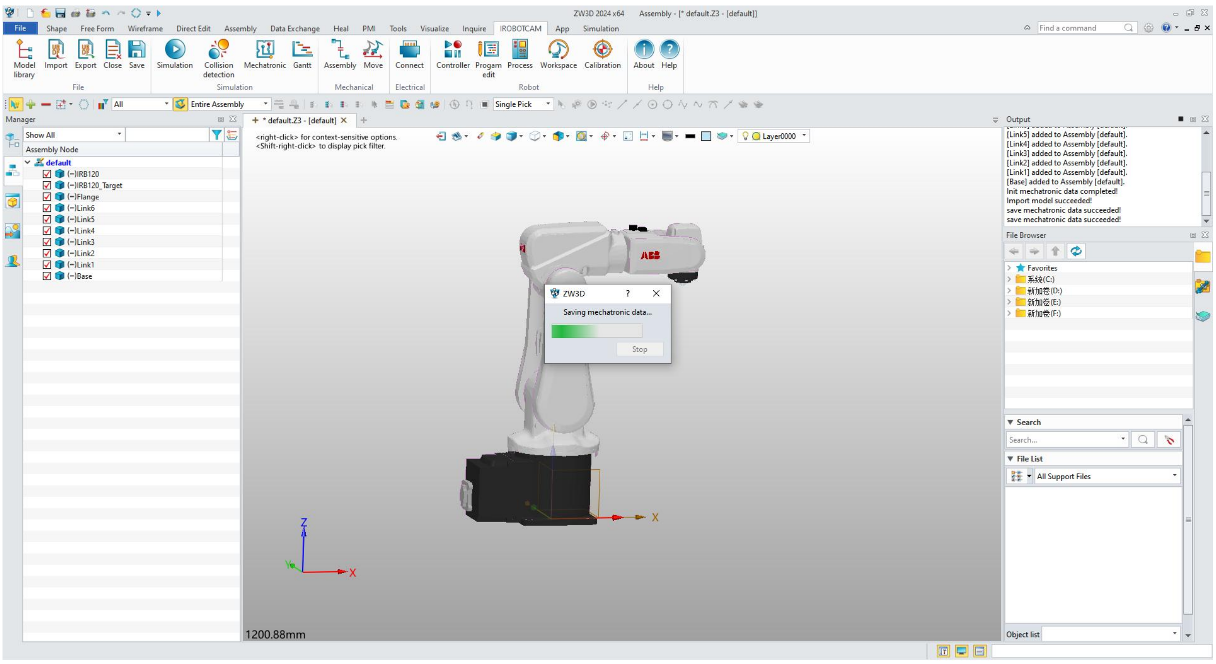


Figure 4-2: Output box

**Save File:** There are two ways to save: one is to overwrite the original imported file and save it, and the other is to save it as a new file.

- **Overwrite the original file:** In the "IROBOTCAM", click the save button in the upper left corner of the interface to overwrite the original file and save it.(Figure 4-3)

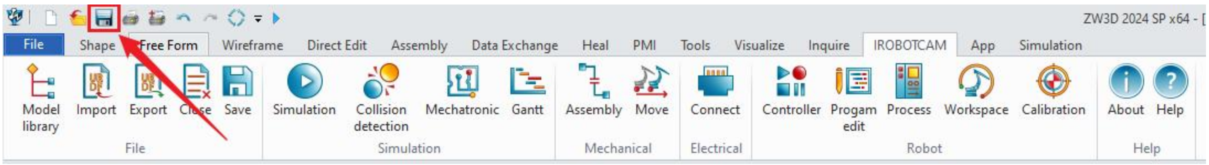



Figure 4-3: Close window

- **Save as a new file:** After successfully clicking the 'Save' button, click the 'File' button on the toolbar.(Figure 4-4)

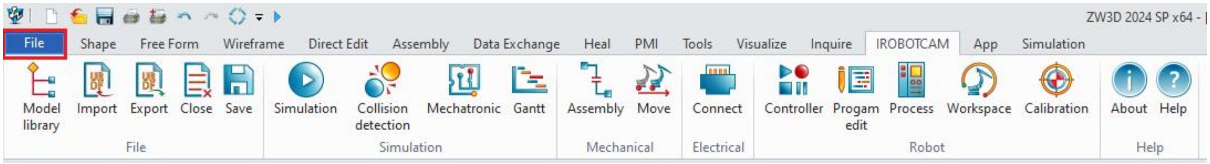



Figure 4-4: File button

Click Save under the "File Box" and select the "Save As" option in the box on the right. (Figure 4-5) (Whether the save was successful will be displayed as text in the output box on the right.)

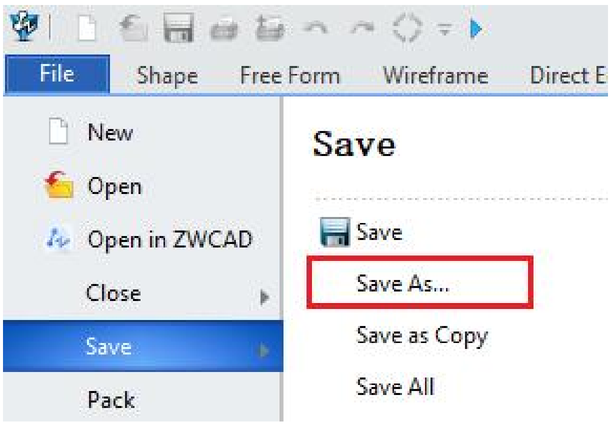



Figure 4-5: Save As

1. **Case file storage location:**

In the directory where ZW3D 2024 software is installed, click on the following path,and in the example folder and help folder of the plugins\iRobotCAM\example folder, it contains the project case and PDF version of the user manual. (Figure 4-6)

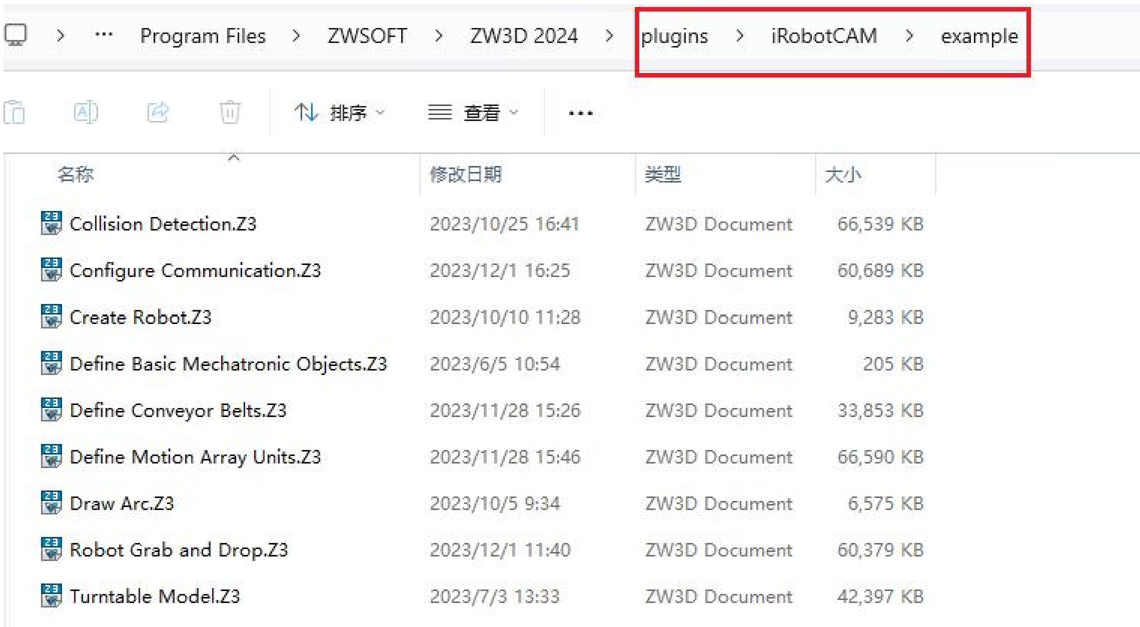



**Figure 4-6: example**

Location of the user manual on the software interface: Enter the "IROBOTCAM" tab and click the help button on the far right to pop up the PDF version of the user manual. (Figure 4-7)

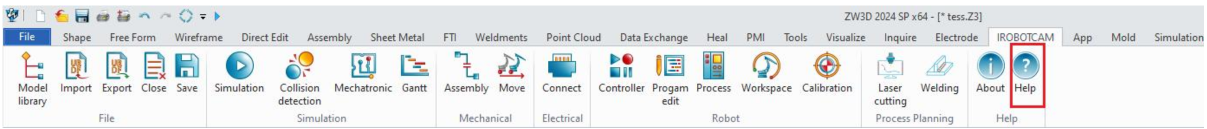



**Figure 4-7: Help button**

2. **Entrance to IROBOTCAM module**

In the start interface, next to Quick Start, click on the IROBOTCAM module, which includes the New, Open, and Activate buttons.

**New:** You can create a new project file.(Figure 4-8)

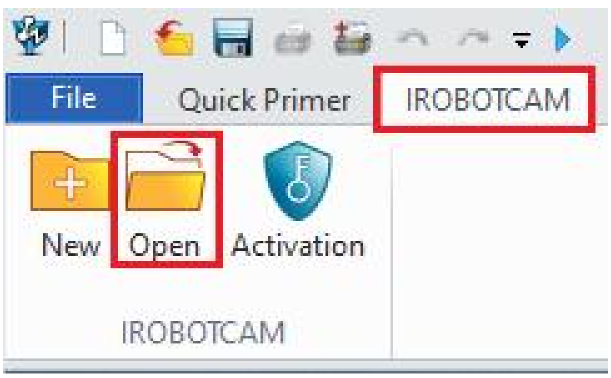



**Figure 4-8: Open button**

**Open:** Click the open button to open the corresponding Z3 file. Attention: When creating in the IROBOTCAM module Z3 files must be opened from this window in order to have relevant IROBOTCAM module content.(Figure 4-9)

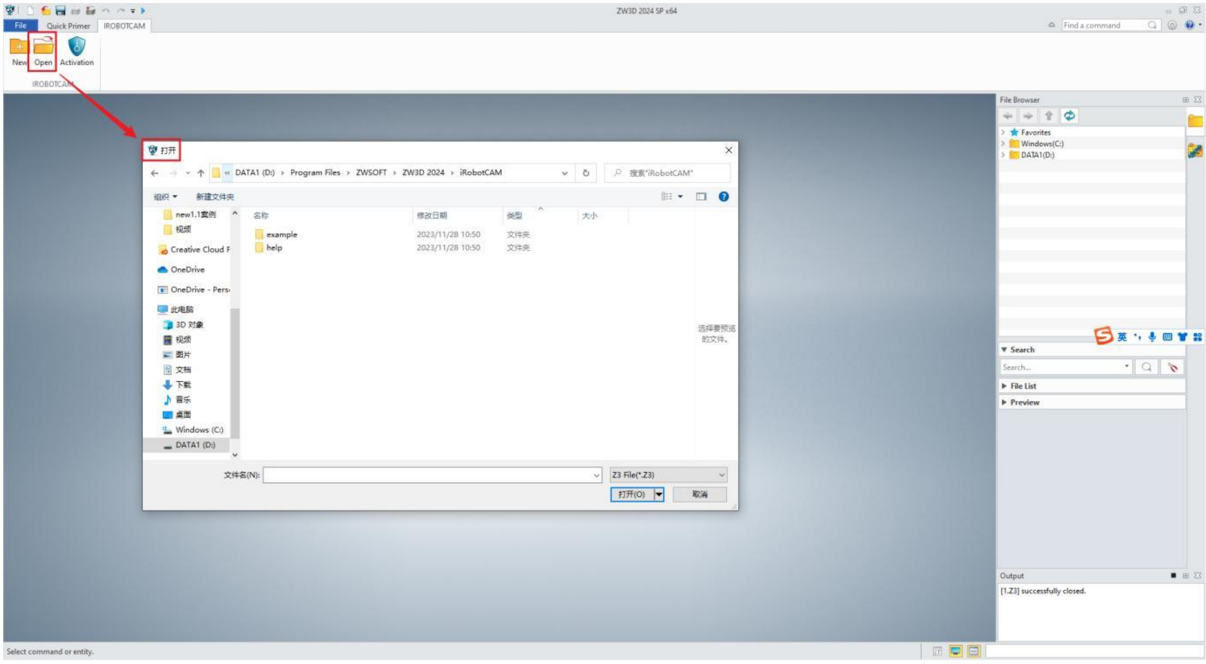



**Figure 4-9: Open**

**Activation:** You can see the product user manual of the software package

- **Activation authorization**: Fill in the activation code entry, click the application button, and the output box will prompt whether activation was successful.
- **Product information:** The CAD kernel status represents the status of ZW3D, which can be divided into three states: inactive, trial, and activated.
- **iRobotCAM status:** It also includes three states: inactive, trial, and activated. When inactive is showing, it indicates that the software trial period of 30 days has ended. The output box has relevant prompts.(Figure 4-10)

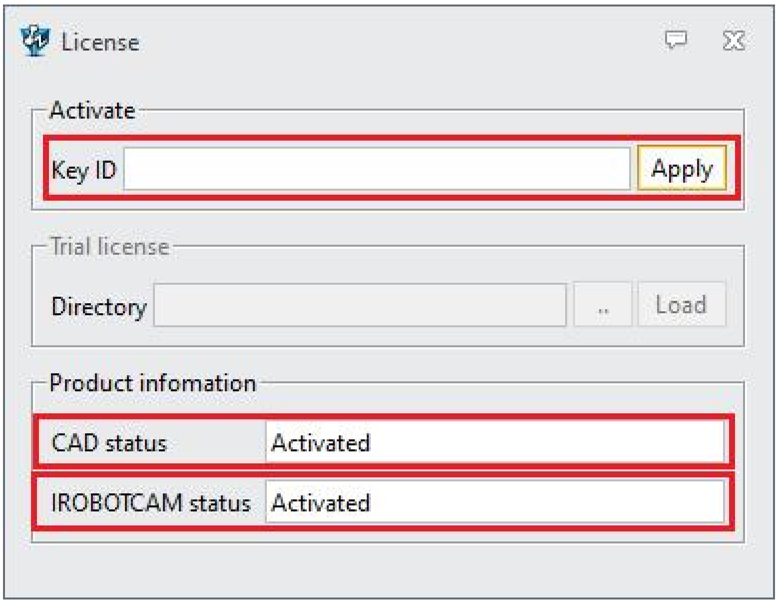



**Figure 4-10: License**

# Chapter 5: Define Basic Mechatronic Objects

This guide provides step-by-step instructions on defining basic mechatronic objects in IROBOTCAM software.(Figure 5-1)

**Note:**Compared to before, the software interface has been improved by adding density parameter settings in the material window, and removing mass settings in the rigid body and collision body windows. Users can pay attention to the changes in materials, rigid bodies, and collision body windows during the creative process.

When users create other cases, they may not be updated in a timely manner due to changes in other case images or videos. Users should pay attention to this issue when using the software. thanks

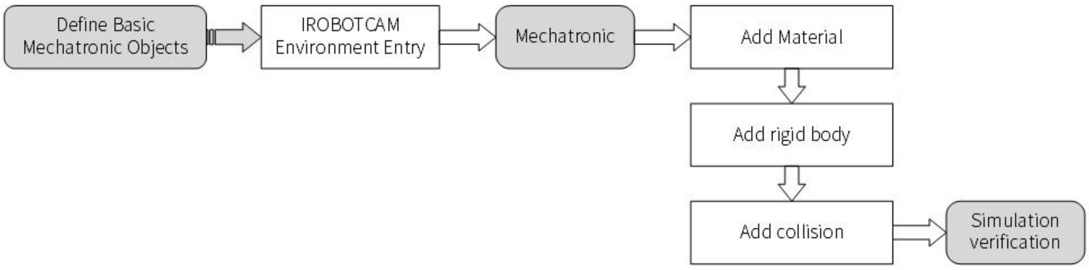



**Figure 5-1：flow chart of define basic mechatronic objects**

## Step 1: Open iRobotCAM project file

- Open iRobotCAM project file named "Define Basic Mechatronic Objects. Z3" to access the IROBOTCAM environment. (Figure 5-2)

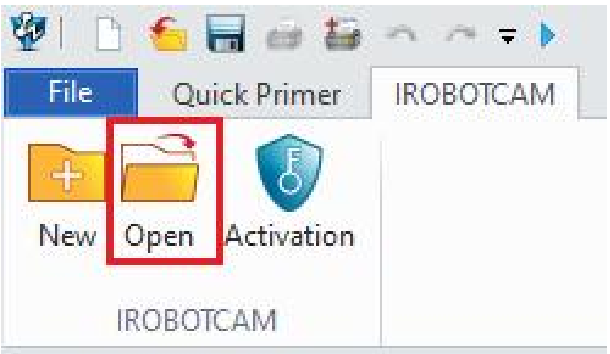



**Figure 5-2： IROBOTCAM Environment Entry**

- The interface will display three components. (Figure 5-4)

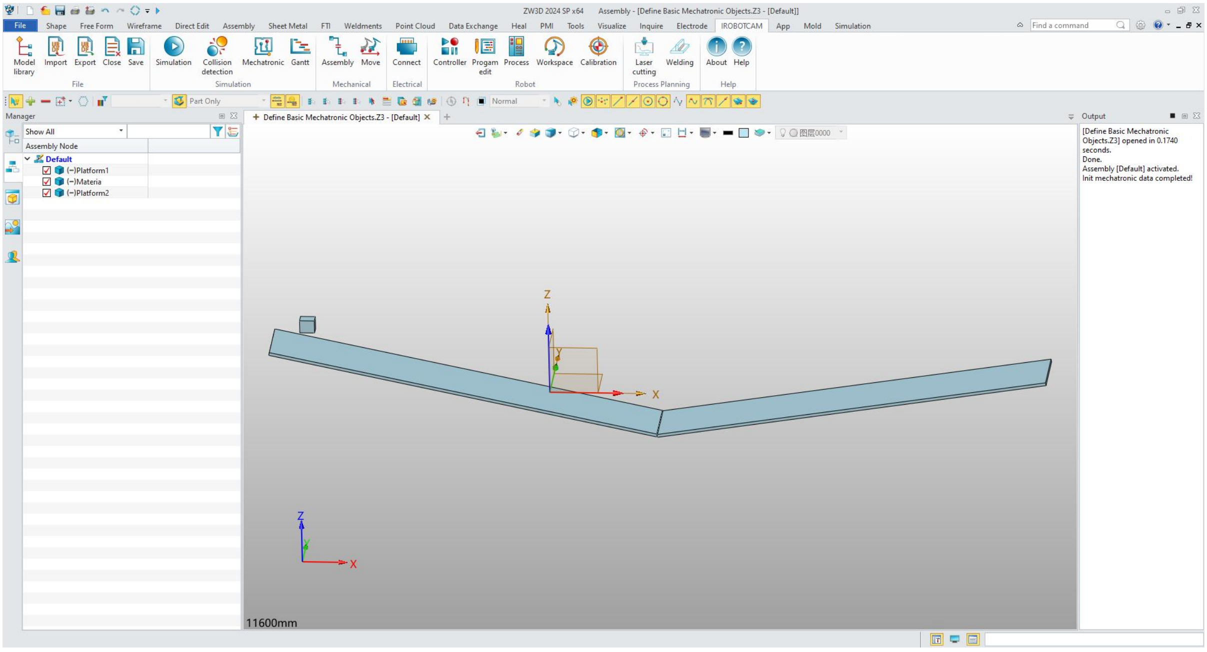


Figure 5-4：Component Details

## Step 2: Create materials

- Click on the "IROBOTCAM" to enter the mechatronic interface. (Figure 5-5)

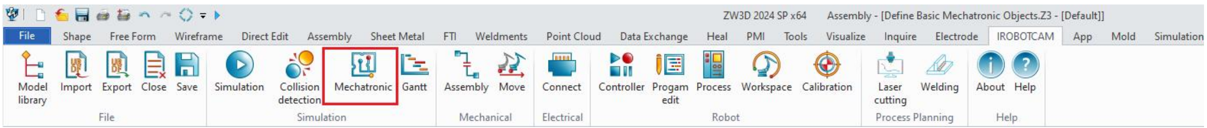



Figure 5-5：Mechatronic

- Right-click on the material and select “Add Material”. (Figure 5-6)

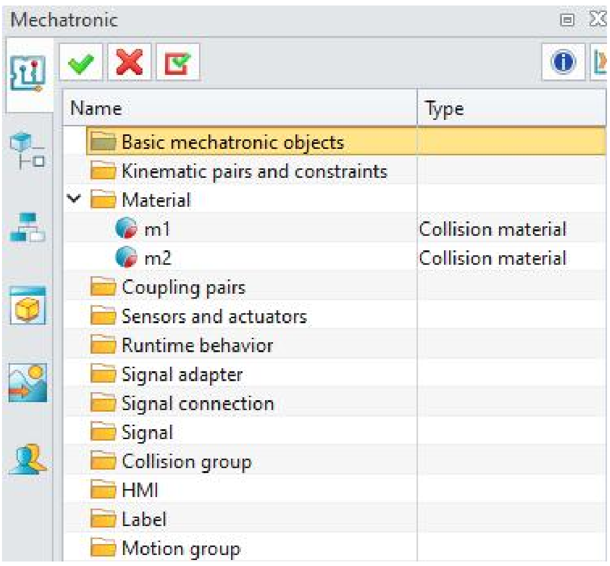


Figure 5-6：Add Material

- Create a new material named “m1” for the two slopes. (Figure 5-7)

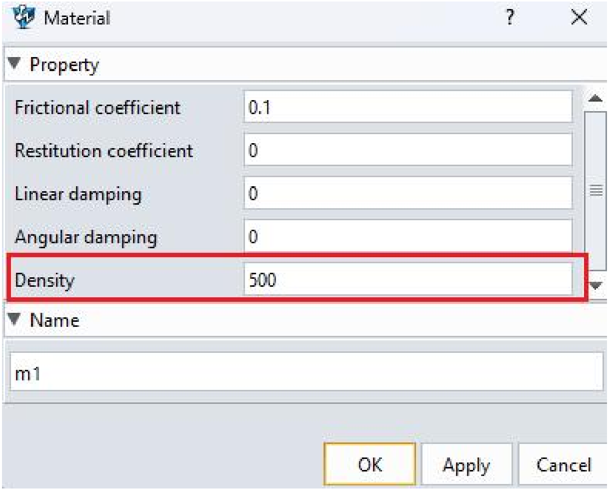



Figure 5-7：Material m1

- Create another material named “m2” for the remaining components. (Figure 5-8)

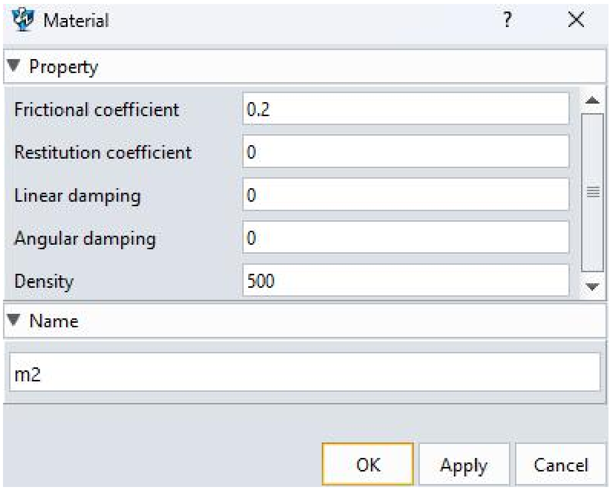



Figure 5-8：Material m2

## Step3：Create Rigid Body

- Right-click on "Basic mechatronic Object", and select "Add Rigid Body". (Figure 5-9)

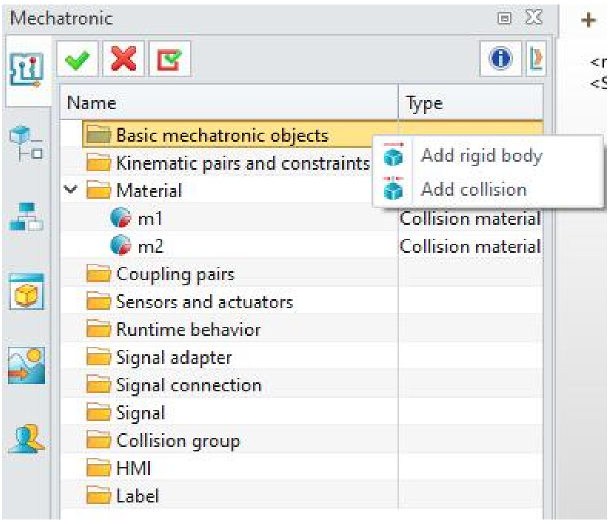



**Figure 5-9：Add Rigid Body**

- In the "Rigid Body" window, click the "+" button to select one of the slope components.
- Uncheck the property settings and click “OK”.
- Repeat above steps to add the other slope as a rigid body. (Figure 5-10)

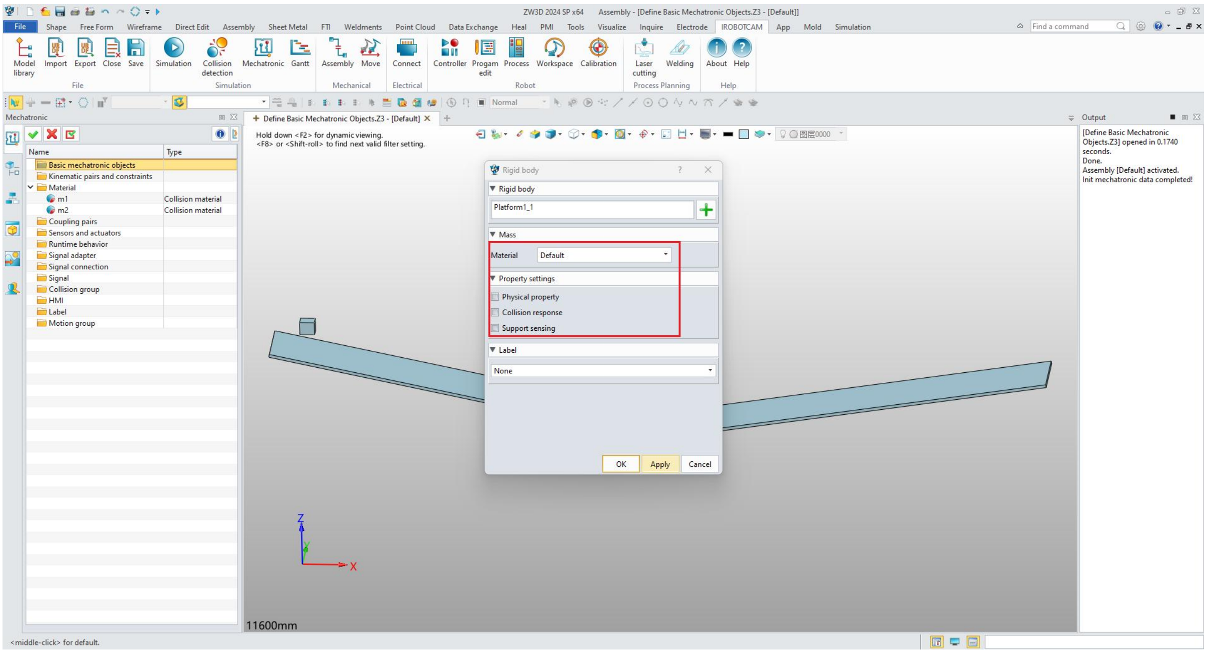



**Figure 5-10：Adding a obliquity Rigid Body**

- Select the "material" indicated by the arrow to create a rigid body.
- Uncheck the property settings and click “OK”.

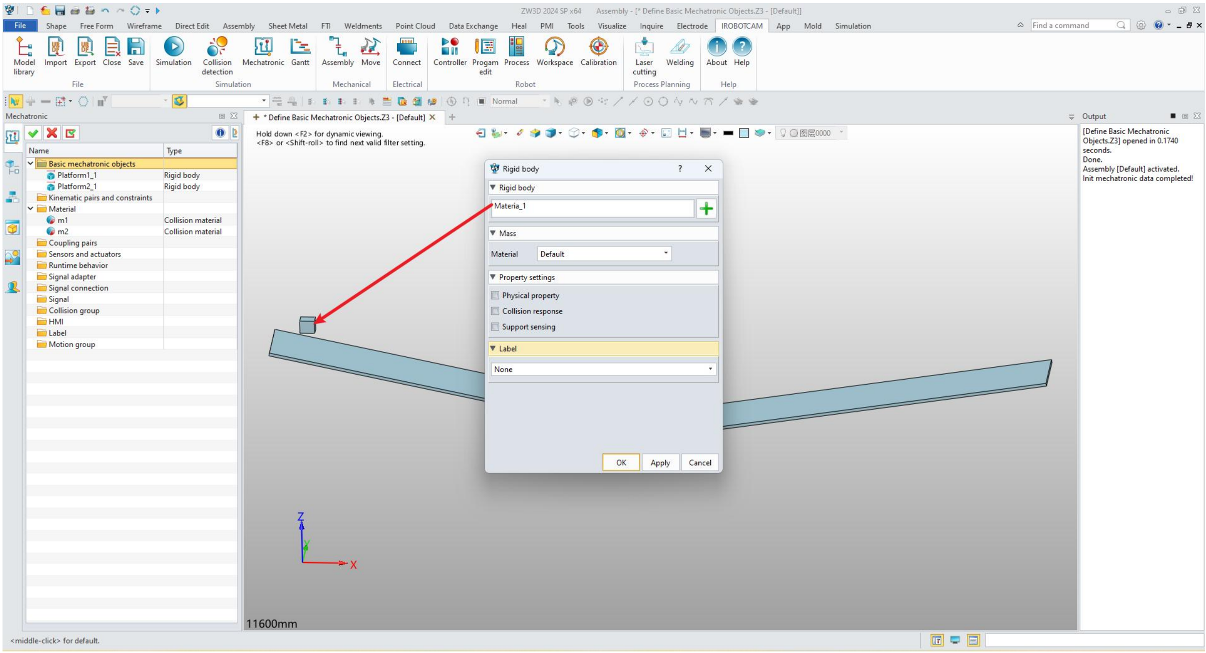


Figure 5-11：Add Materia rigid body

## Step4：Creat Collision

- Right-click on "Basic mechatronic Object" and select "Add Collision".
- In the "Collision" window, click the "+" button and select one of the slope components.
- Select "Convex Decomposition" for the collision shape.
- Check "Collision Response" for the property setting.
- Select "m1" for the material.
- Repeat above steps to create a collision for the other slope. (Figure 5-12)

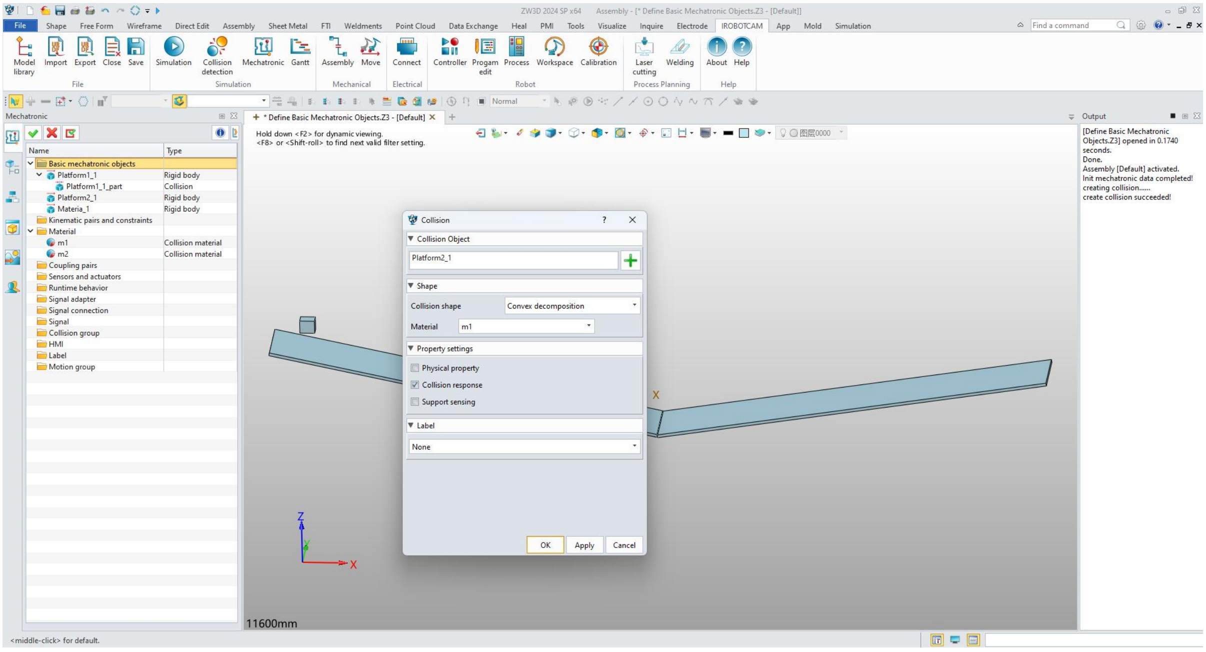



**Figure 5-12：Add obliquity collision body**

- Click the "+" button again and select the box to create a collision.
- Select "convex decomposition" for the collision shape.
- Check "physical property" and "collision response" for the collision setting.
- Select "m2" for the collision material. (Figure 5-13)

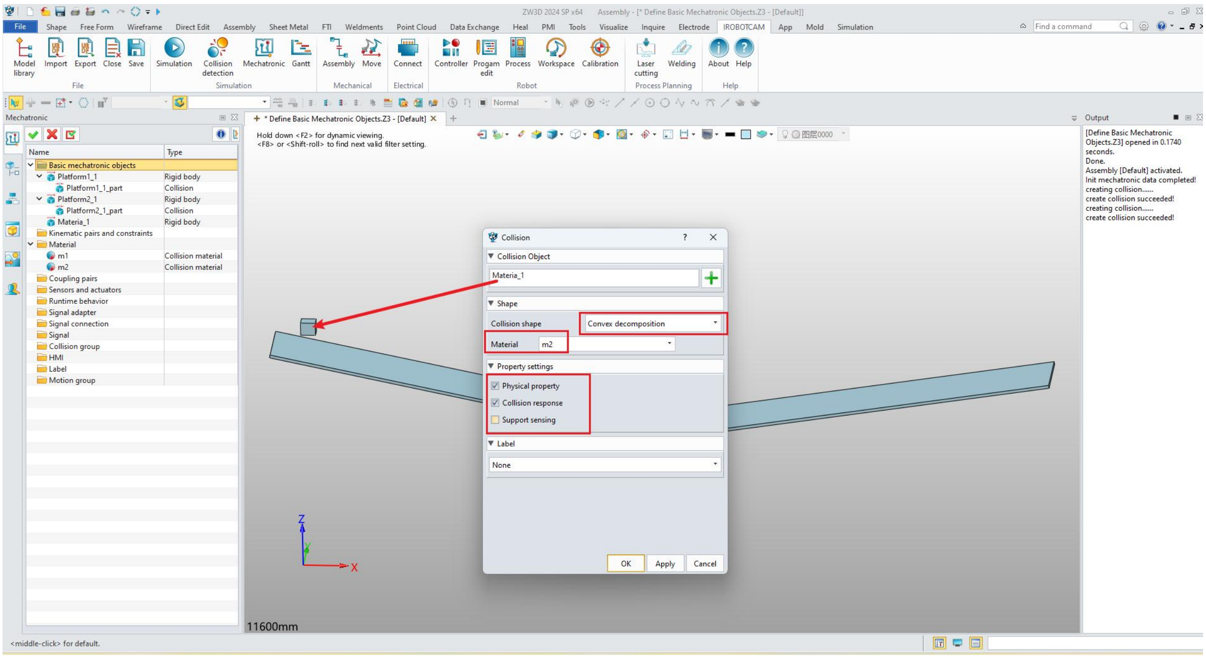



**Figure 5-13：Add Materia collision body**

## Step 5：View the framework for mechatronic

- At this point, you should have successfully created materials, rigid bodies, and collisions for all components. (Figure 5-14)

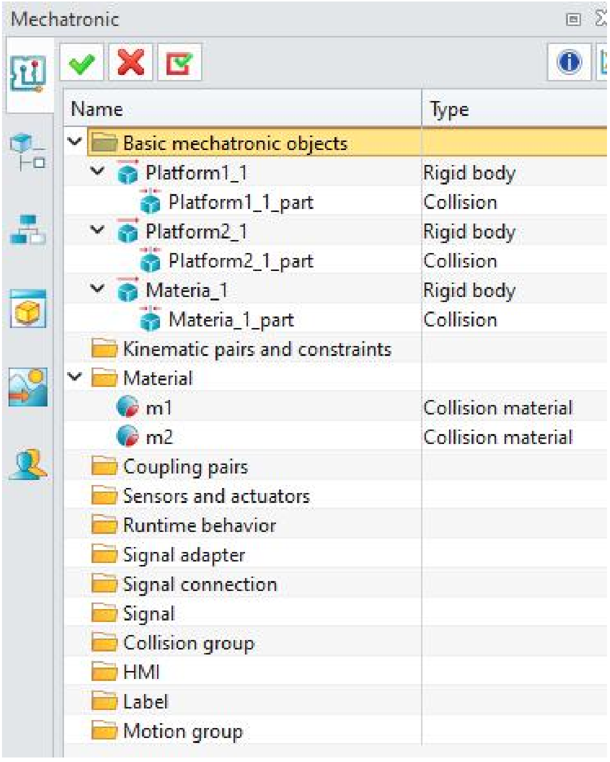



**Figure 5-14：The entire mechatronic framework**

## Step 6：Simulation verification

- Click on the “IROBOTCAM” tab.(Figure 5-15)


**Figure 5-15：Simulation button**

- Click the simulation button to verify the defined mechatronic objects. (Figure 5-16)

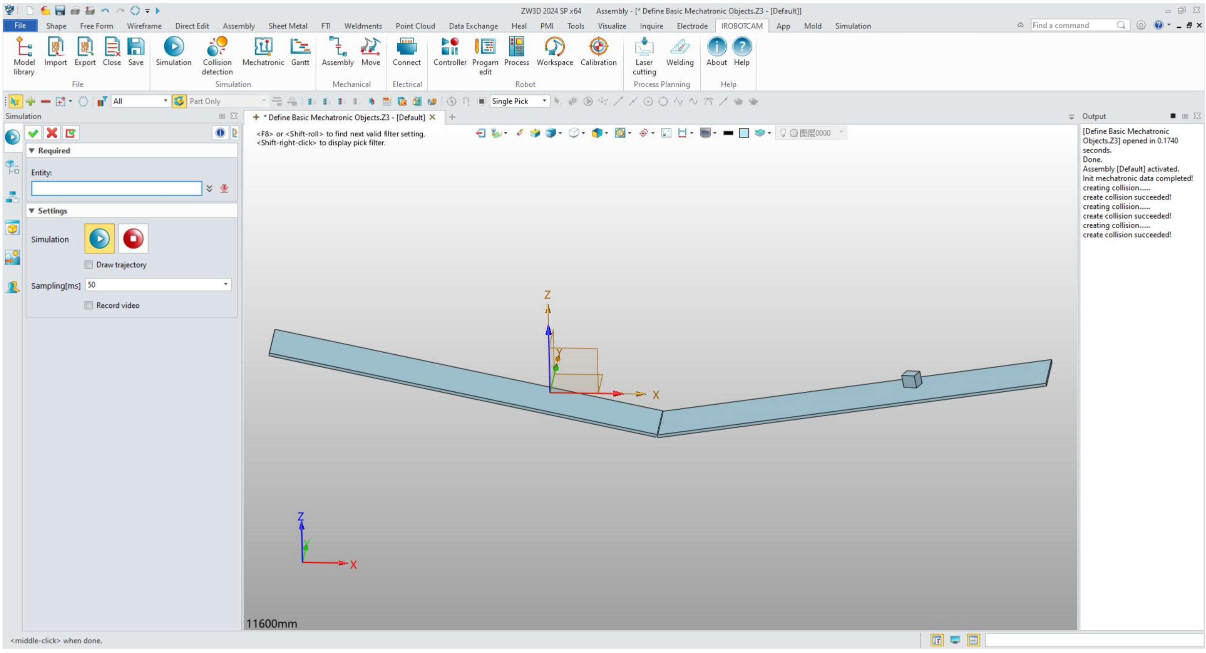


Figure 5-16：Simulation verification

# Chapter 6: Define Conveyor Belts

This document provides a step-by-step guide on how to define a conveyor belt in the IROBOTCAM software.(Figure 6-1)

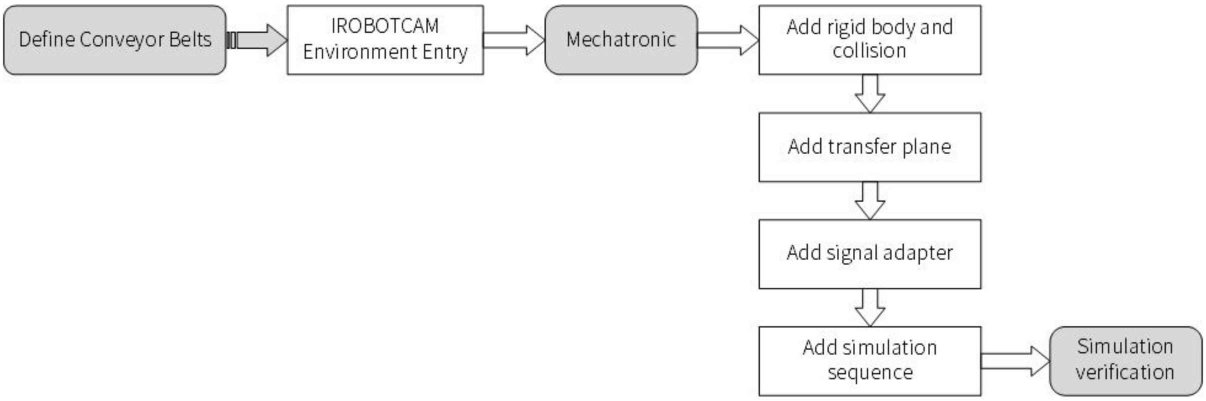



**Figure 6-1： Flow chart of define conveyor belts**

## Step 1：Open iRobotCAM project file

- Open iRobotCAM project file "Define Conveyor Belts.Z3" to access the IROBOTCAM environment. (Figure 6-2)

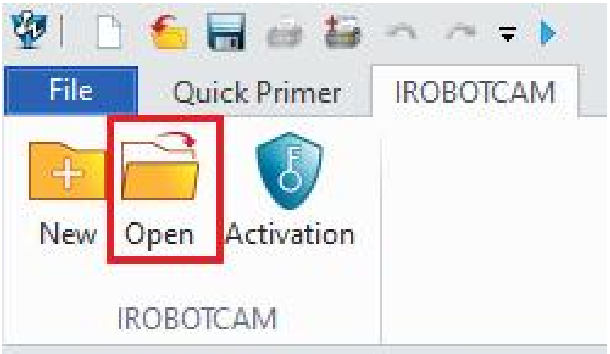



**Figure 6-2：Open iRobotCAM project file**

- Wait for the import to complete. A message will be displayed in the output box confirming completion. (Figure 6-3)

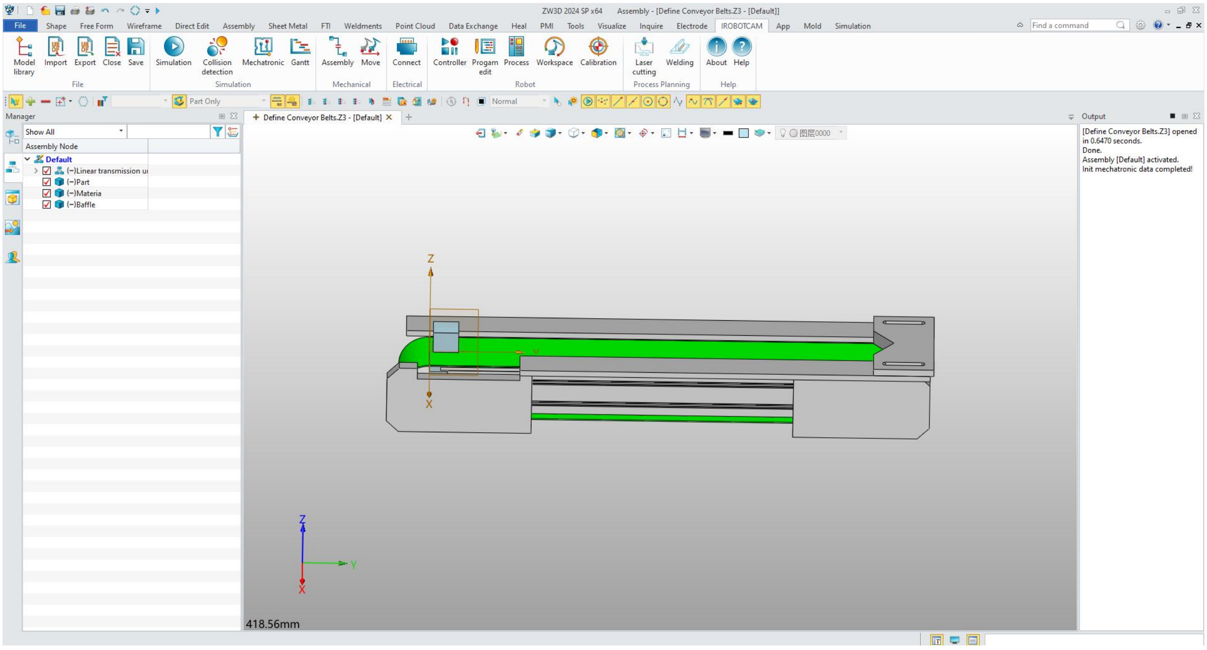


**Figure 6-3：Component Details**

## Step 2：Create rigid body and collision.

- Go to the "IROBOTCAM" tab and click the "Mechatronic" button. (Figure 6-4)

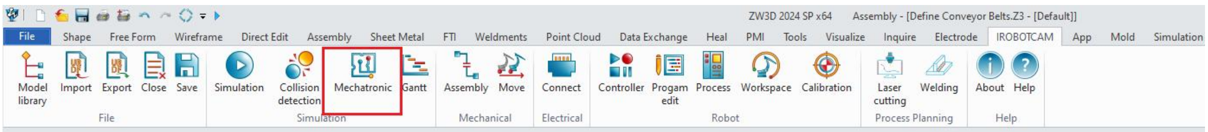



**Figure 6-4：Mechatronic**

- Under "Basic Mechatronic Objects", right-click and choose “Add Rigid Body”. (Figure 6-5)

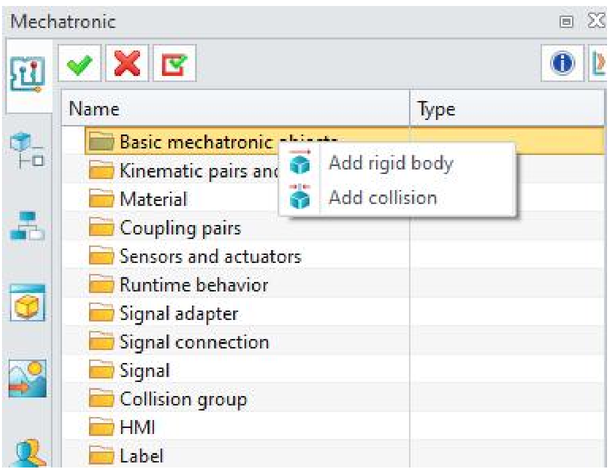


**Figure 6-5：Add Rigid Body**

- **Part:** Click the "+" button to enter the object selection status.
- Click the left mouse button to select the component , which is the long board, and create a rigid body.
- The material is "default", and the property setting is "unchecked".
- Click the OK button to create the first component as a rigid body. (Figure 6-6)

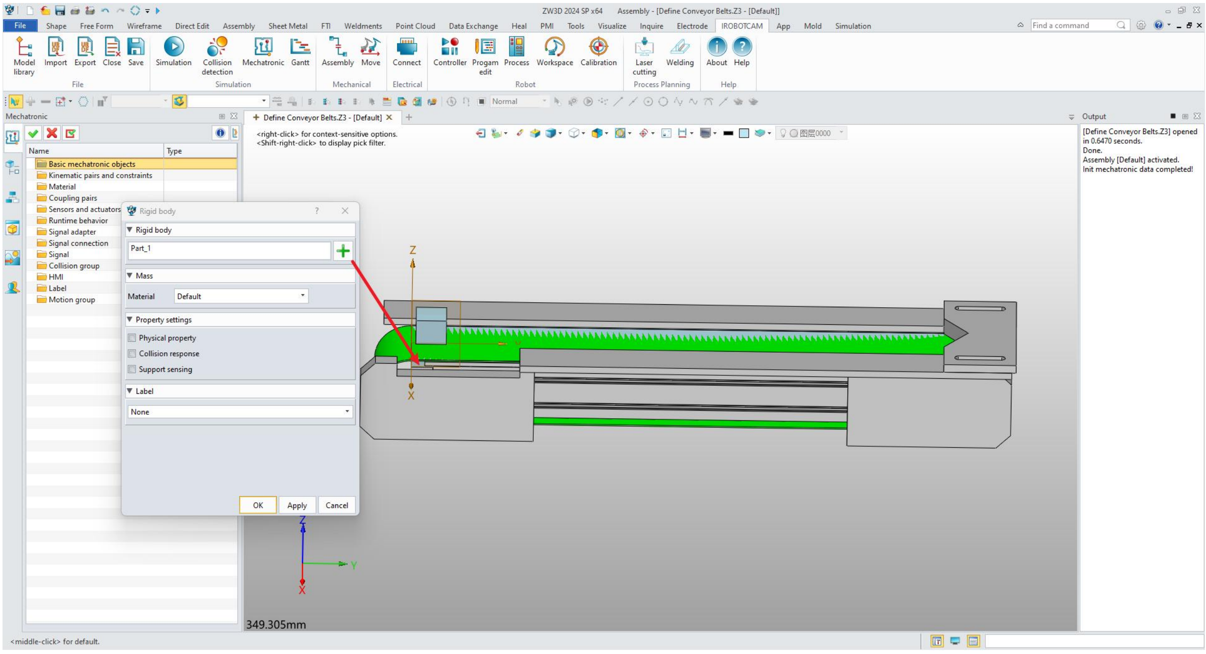


**Figure 6-6：Add Long Board Rigid Body**

- Click the "+" button to enter the object selection status, select the component , which is the long board, to create a collision.
- Select "convex decomposition" for the collision shape, "collision response" for the property settings, and "default" for the material. (Figure 6-7)

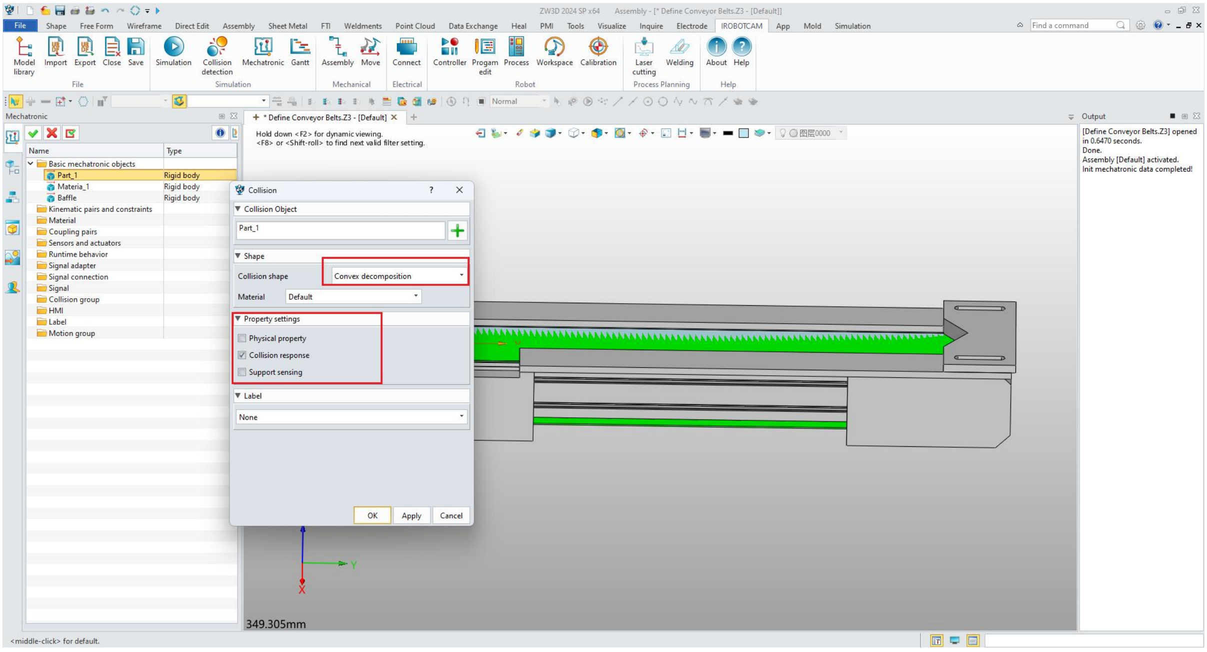


Figure 6-7：Add Long Board Collision

- **Material:** With the "+" button, select the material to create it as a rigid body, with a default material, and an property setting of "unchecked". (Figure 6-8)

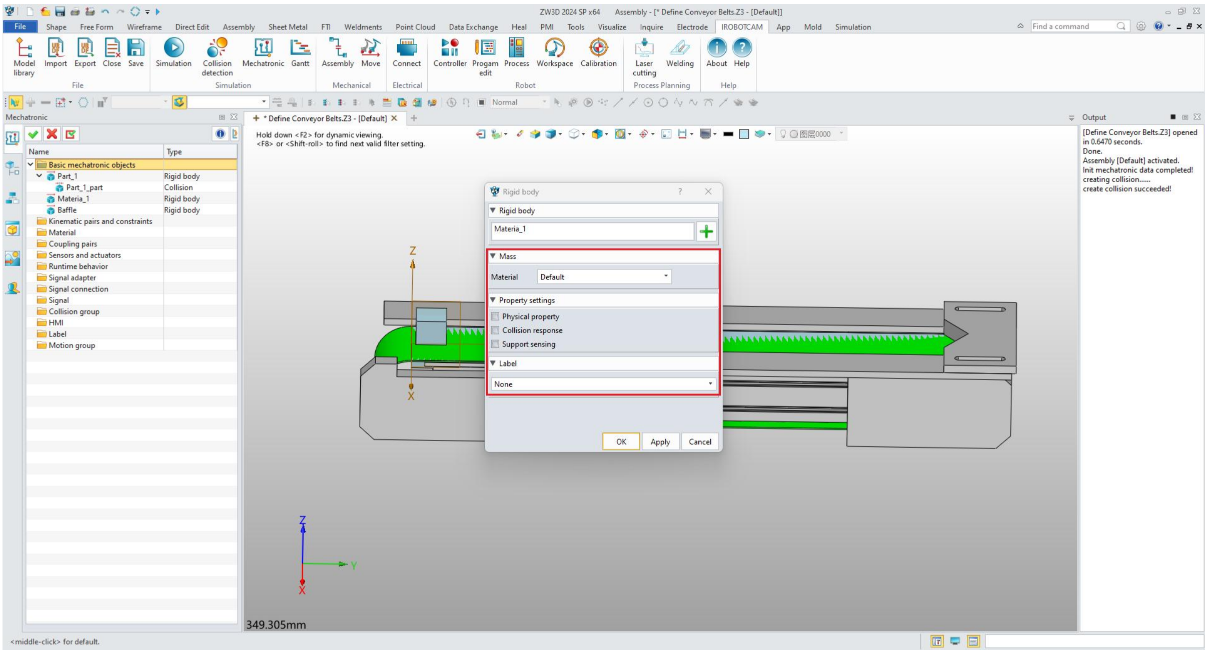


Figure 6-8：Add Rigid Body

- Click the "+" button to enter the object selection status, select the component on the right to create a collision for the material.
- Select "convex decomposition" for the collision shape.
- Select "physical properties", "collision response", "support sensing" for property settings, and "default recognition" for material. (Figure 6-9)

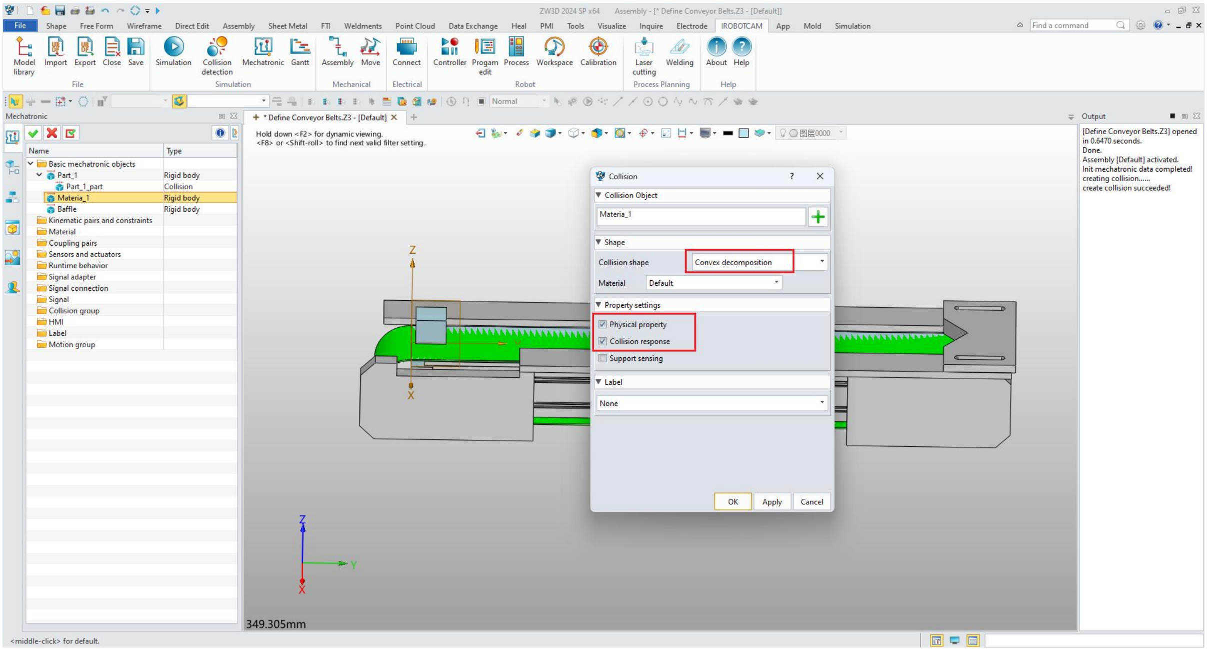



**Figure 6-9：Add Small Wooden Collision**

- **Barrier:** Use the "+" button to select the material barrier plate zdk and create it as a rigid body. The default mass is "1", the material is "default", and the property setting is "unchecked". (Figure 6-10)

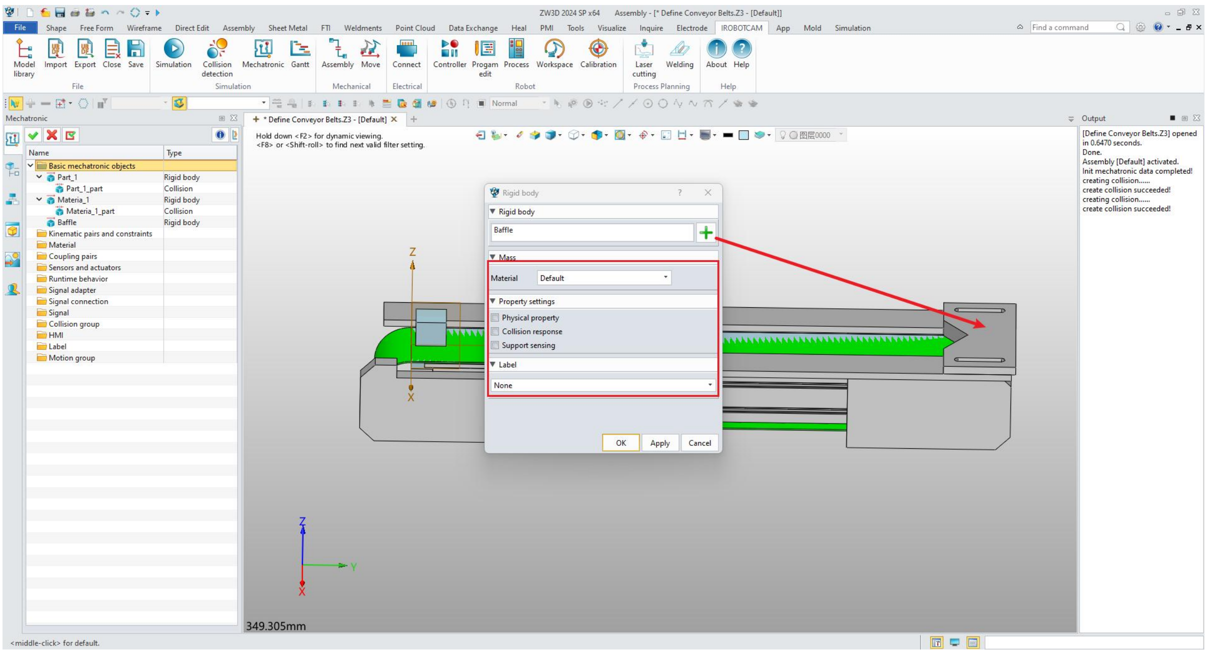



**Figure 6-10：Add Barrier Plate Rigid Body**

- Click the "+" button to enter the object selection status.
- Select the component zdk on the right, which is the barrier, to create a collision.

- Select "convex decomposition" for the collision shape,"collision response" for the property settings, and "default" for the material.
- Click OK. (This process needs to be patient for a while) (Figure 6-11)

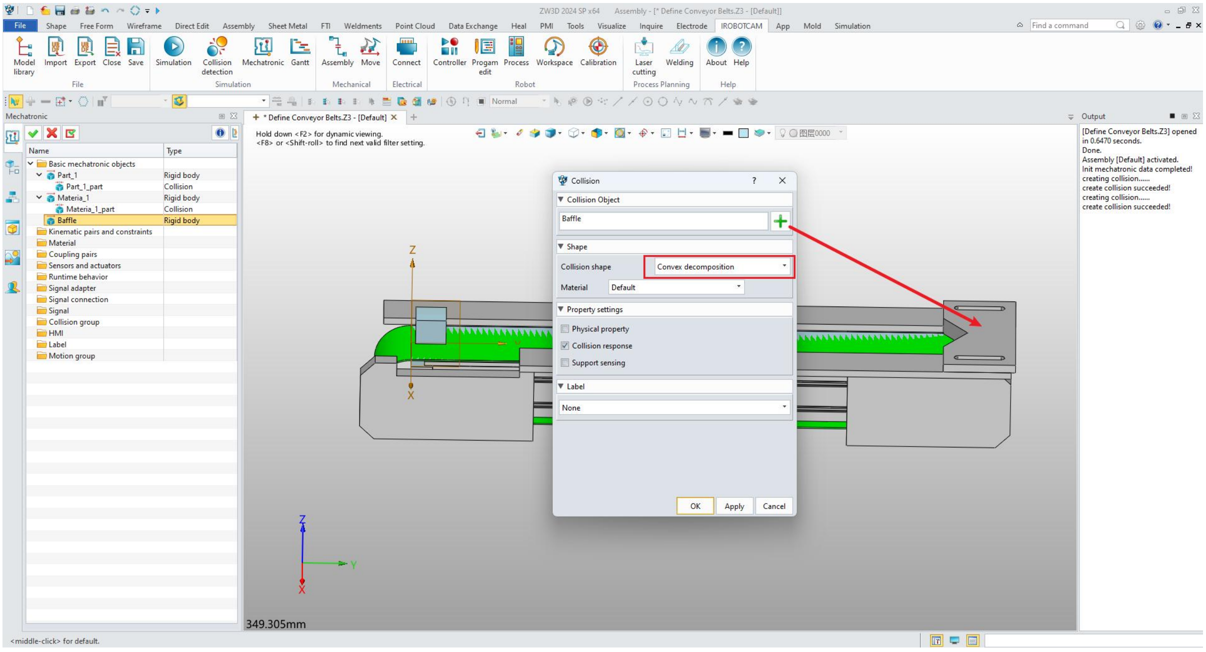



**Figure 6-11：Add Barrier Plate Collision Body**

## Step 3：Basic mechatronic object framework

- All rigid bodies and collisions should now be listed under "Basic Mechatronic Objects." (Figure 6-12)

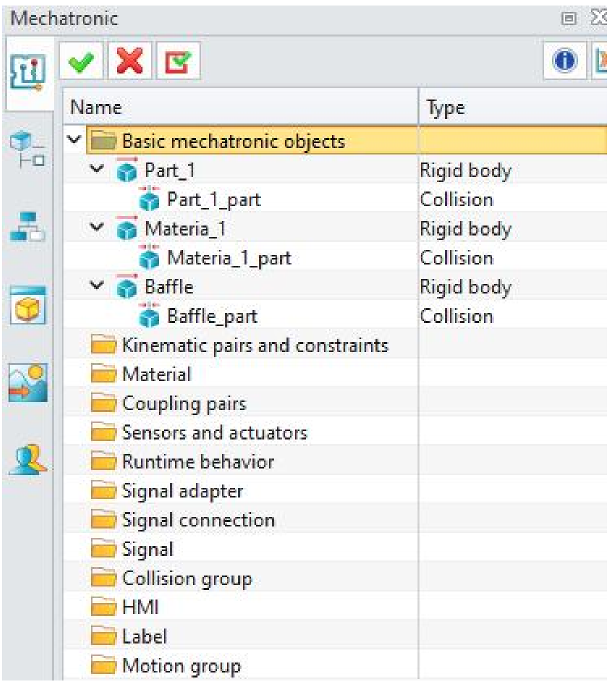


Figure 6-12：Basic mechatronic object framework

## Step 4：Add transfer plane

- Right-click "Sensors and actuators" and select "Add transfer plane". (Figure 6-13)

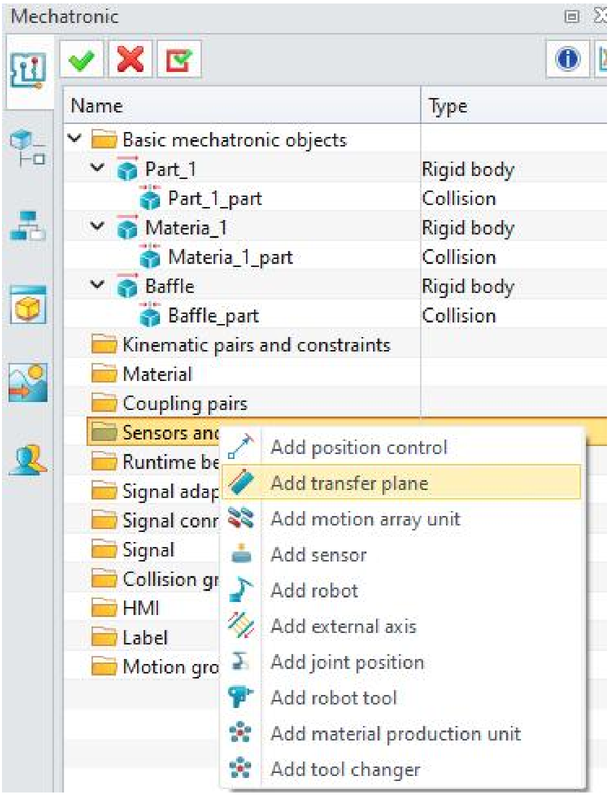



Figure 6-13：Add Transfer plane

- Select "Part" as the collision object.
- Click the "+" button and select CSYS1 as the base coordinate.
- In the "Transfer Surface" window, select "Second Entry" from the "Specify Vector" drop-down list. (conveyor movement along Y-axis).
- Enter the conveyor belt speed as "100".
- Named the transfer plane.
- Click “OK” to generate the conveyor belt object. (Figure 6-14)

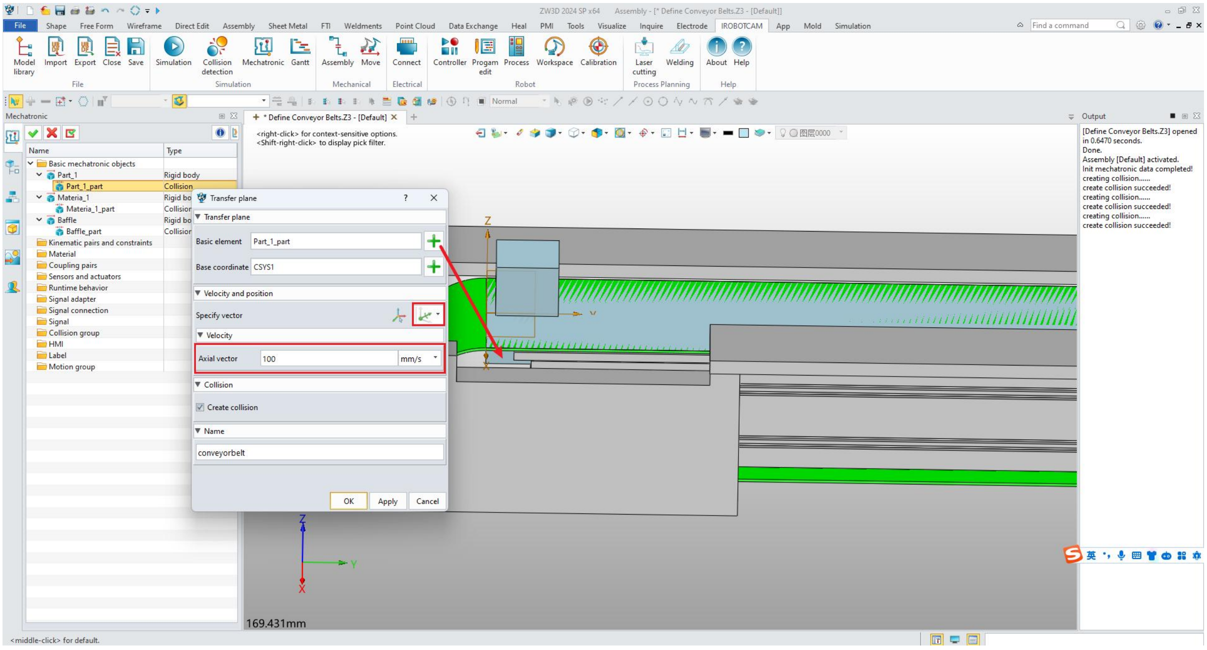


Figure 6-14: Transfer plane

## Step 5: Add a signal adapter

- Click "Signal Adapter" and right-click to select "Add Signal Adapter". (Figure 6-15)

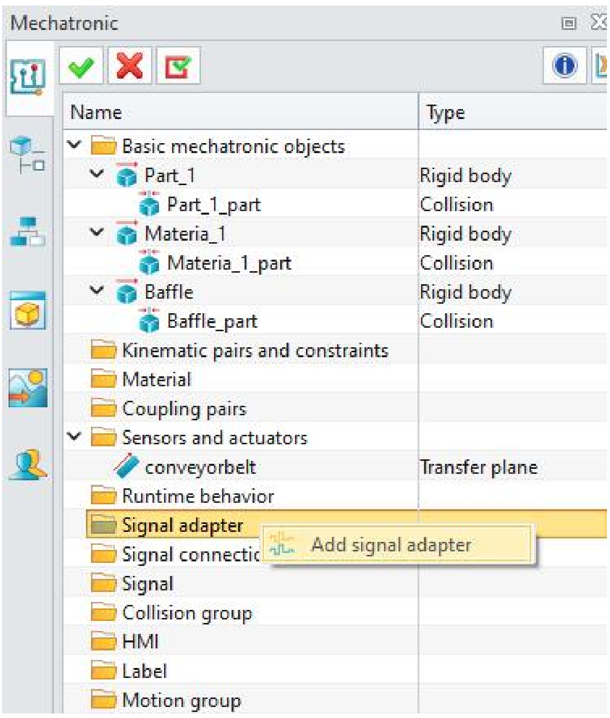



Figure 6-15: Signal Adapter

- Click "Sensors and actuators" then click the "+" button.

- Select "Transmission surface".
- Add parameters as shown in the provided figures, including signals, formulas, and names. Click OK to generate signal adapter objects and signal objects. (Figure 6-16)

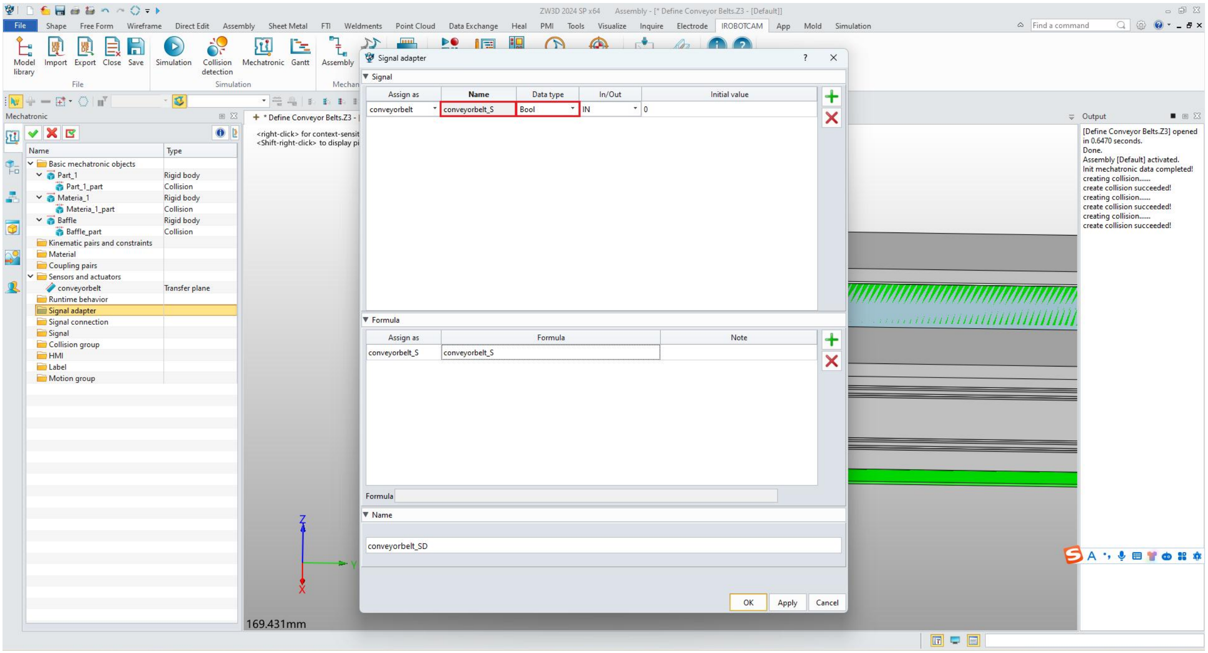



**Figure 6-16：Add Signal Adapter**

## Step 6：Add runtime behavior

- Click “runtime behavior”, right-click and select “Add Simulation Sequence”. (Figure 6-17)

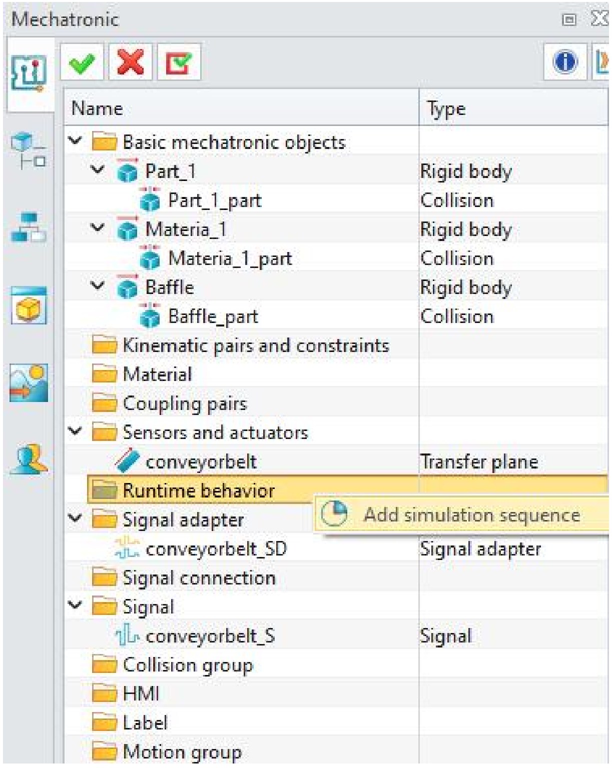



**Figure 6-17：Add simulation sequence**

- Set the Start time as "0" and the Duration as "999999".
- Uncheck the "Activation Signal" and "End Signal".
- Click the "+" button and select "Input signal" as the execution signal.
- Set the End Position to "1".
- Click “OK”. (Figure 6-18)

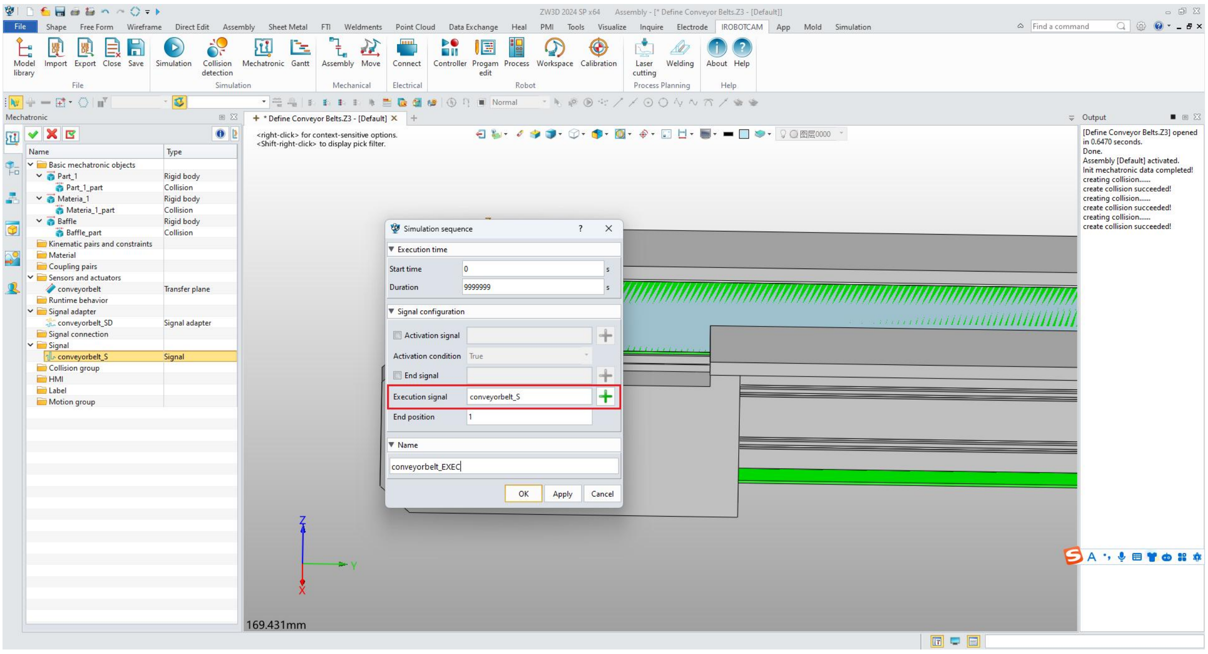


**Figure 6-18：Simulation Sequence**

## Step 7：Simulation Verification

- Click the simulation button Under the "IROBOTCAM". (Figure 6-19)

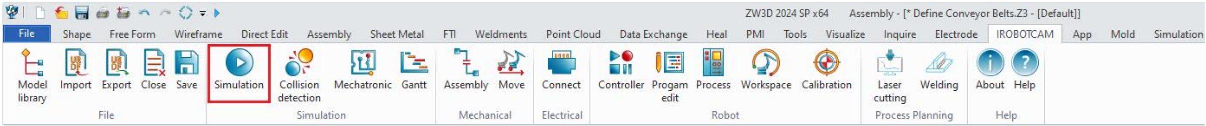



**Figure 6-19：Simulation Button**

- Click the “Start Simulation” button in the "Simulation Control" window.
- Observe the material being transported to the right side of the conveyor belt. (Figure 6-20)

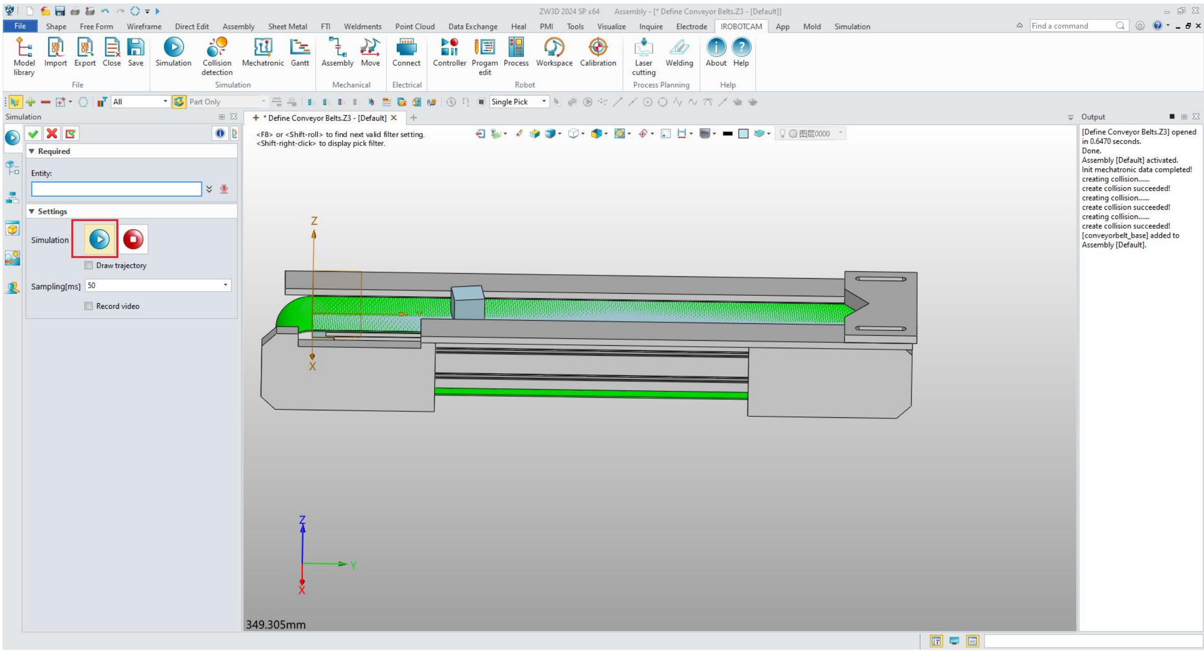



**Figure 6-20：Simulation Verification**

# Chapter 7: Define Motion Array Units

This document outlines the process of defining motion array units within the iRobotCAM environment. This guide is intended for users familiar with the basic functionalities of iRobotCAM and seeks to provide a detailed and professional approach to setting up and simulating conveyor belt systems.(Figure 7-1)

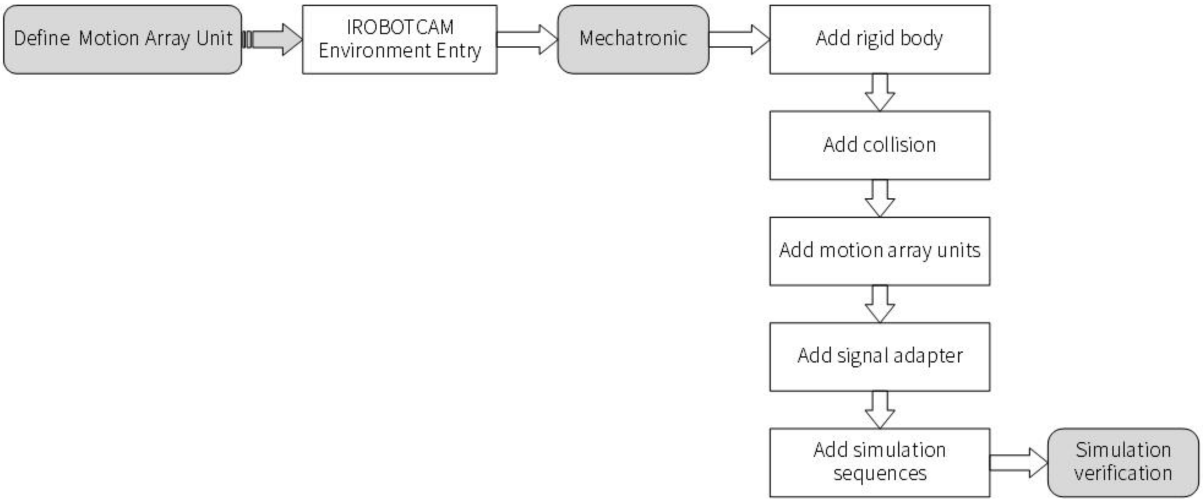



**Figure 7-1：Flow chart of define motion array unit**

## Step 1：Opening and Selecting the Assembly

- Open iRobotCAM project file "Define motion array Unit. Z3" to enter the IROBOTCAM environment. (Figure 7-2)

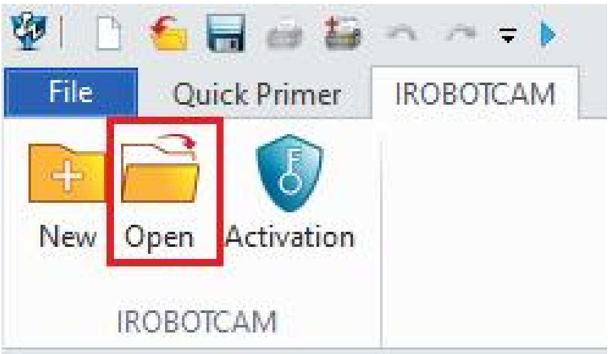



**Figure 7-2：Open iRobotCAM project file**

- Open the details of the interface components. (Figure7-4)

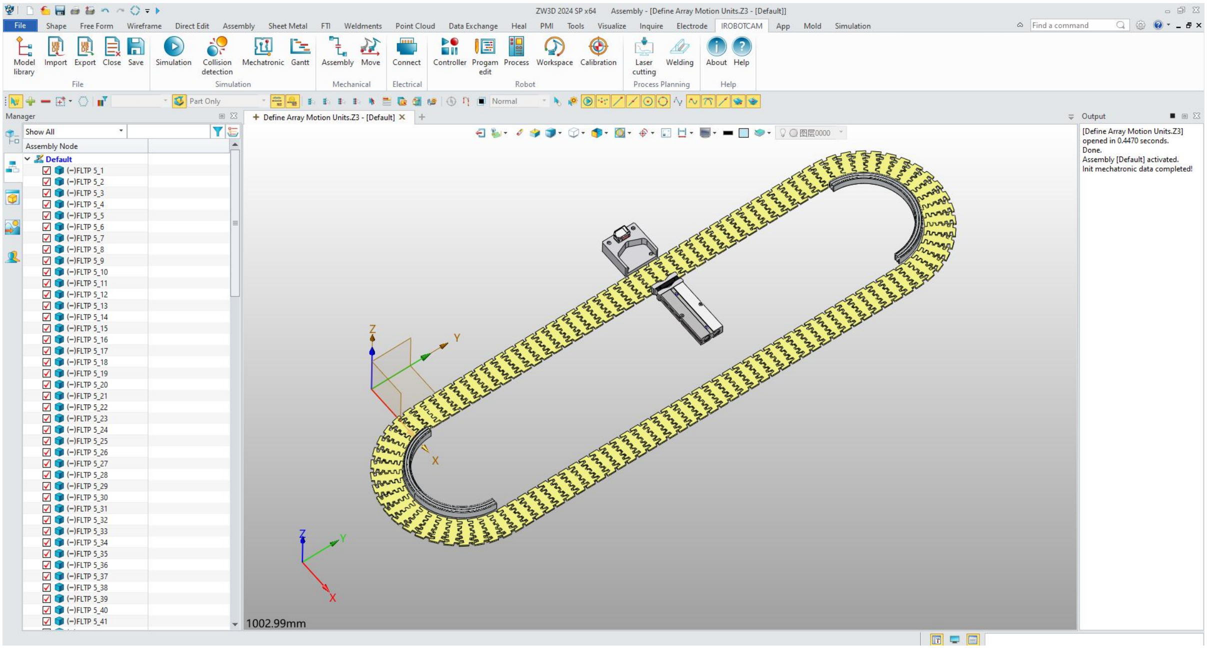


Figure 7-4：Component Details

## Step 2：Creating Rigid Body for FLTP5_7

- Navigate to the "Mechatronic" tab within the "IROBOTCAM" interface. (Figure 7-5)


Figure 7-5：Mechatronic

- Right-click within the "Basic Mechatronic Objects" panel and select "Add RigidBody" to open the "Add Rigid Body" window. (Figure7-6)

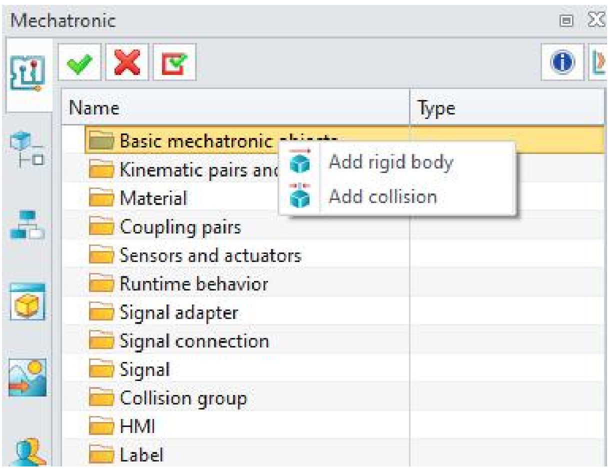


**Figure 7-6：Add rigid body**

- Click Click the "+" button and choose the component "FLTP5_7" from the available list.
- "Property setting" parameters to suit your specific requirements.
- **Notes:** Material use "default", property setting to "unchecked".
- Click "OK" to create the rigid body for "FLTP5_7". (Figure 7-7)

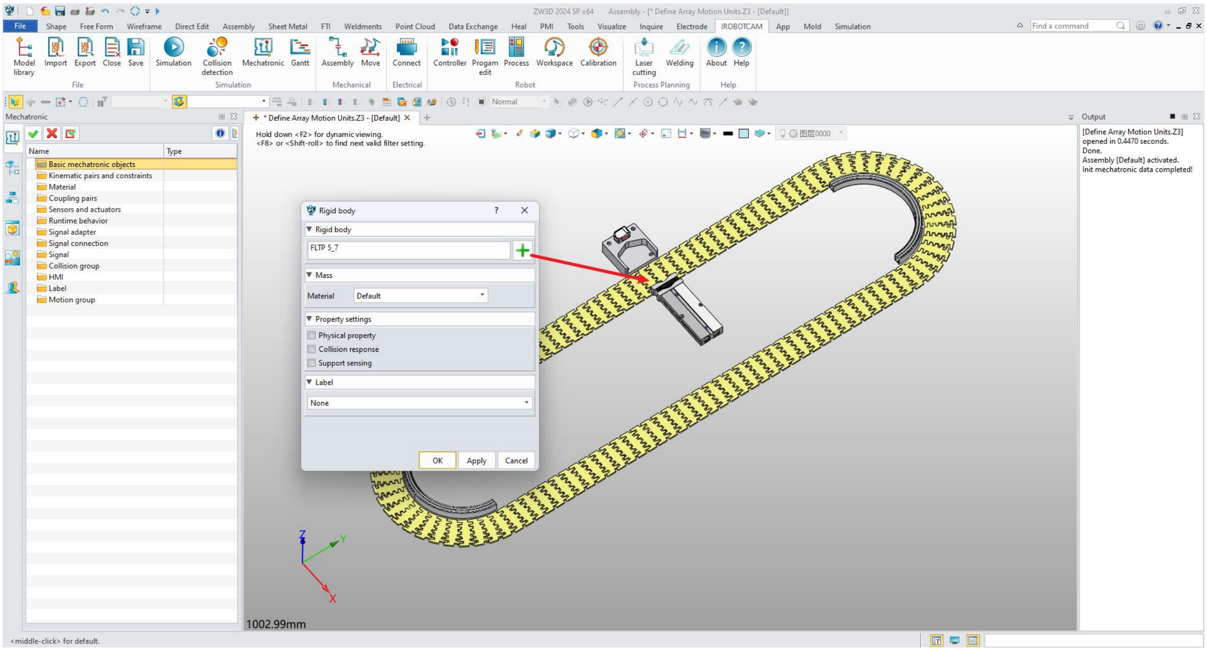



**Figure 7-7：Rigid Body Information**

## Step 3：Creating Collision Body for FLTP5_7

- The Right-click within the "Basic Mechatronic Objects" panel and select "Add Collision" to open the "Add Collision Body" window.
- Click the "+" button and choose the component "FLTP5_7" from the available list.
- **Notes:** The motion array unit needs to select a conveyor belt component (including rigid body and collision) as the basic model to create the entire conveyor belt. Therefore, the components selected for creating collision bodies here should be consistent with the components selected for creating rigid bodies in step 2.
- Modify the "Mass" "Material" and "Property setting" parameters to suit your specific requirements.
- **Notes:** Mass could be input 0.1kg. the material use "default", the collision shape needs to be selected as "convex decomposition", check the “Collision response” under the Property settings.
- Click "OK" to create the Collision body for "FLTP5_7". (Figure 7-8)

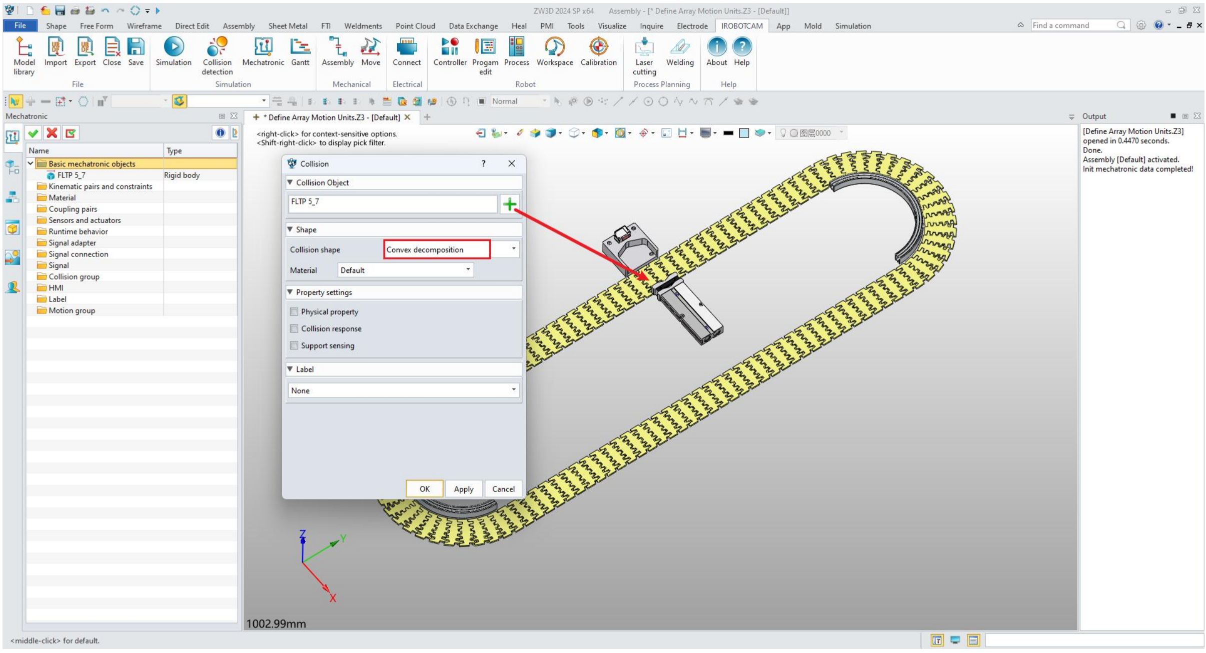



**Figure 7-8：Collision Information**

- **Notes:** It should be noted here that the waiting time of the convex decomposition algorithm varies depending on the complexity of the model and the computer performance of the user. The waiting time of this example model is approximately 10 seconds.

## Step 4：Adding motion array Units

- Click Right-click within the "Sensors and actuators" panel and choose "Add motion array unit". (Figure 7-9)

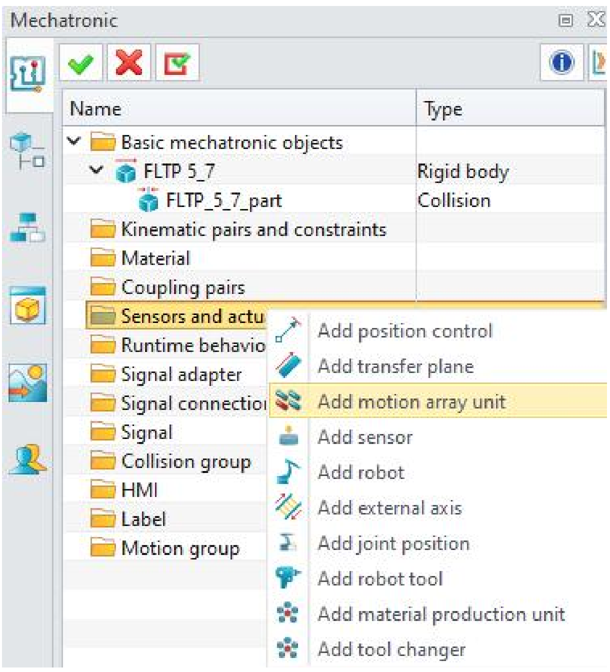


**Figure 7-9：Add motion array**

- Click the "+" button and select the previously created "FLTP5_7" rigid body as the base element for the motion array unit.
- Click the “+” button and select a CSYS as the base coordinate.
- Define the desired "Speed" (eg, 100) at which the conveyor belt will operate.
- Assign a descriptive "Name" for easy identification. (Figure 7-10)

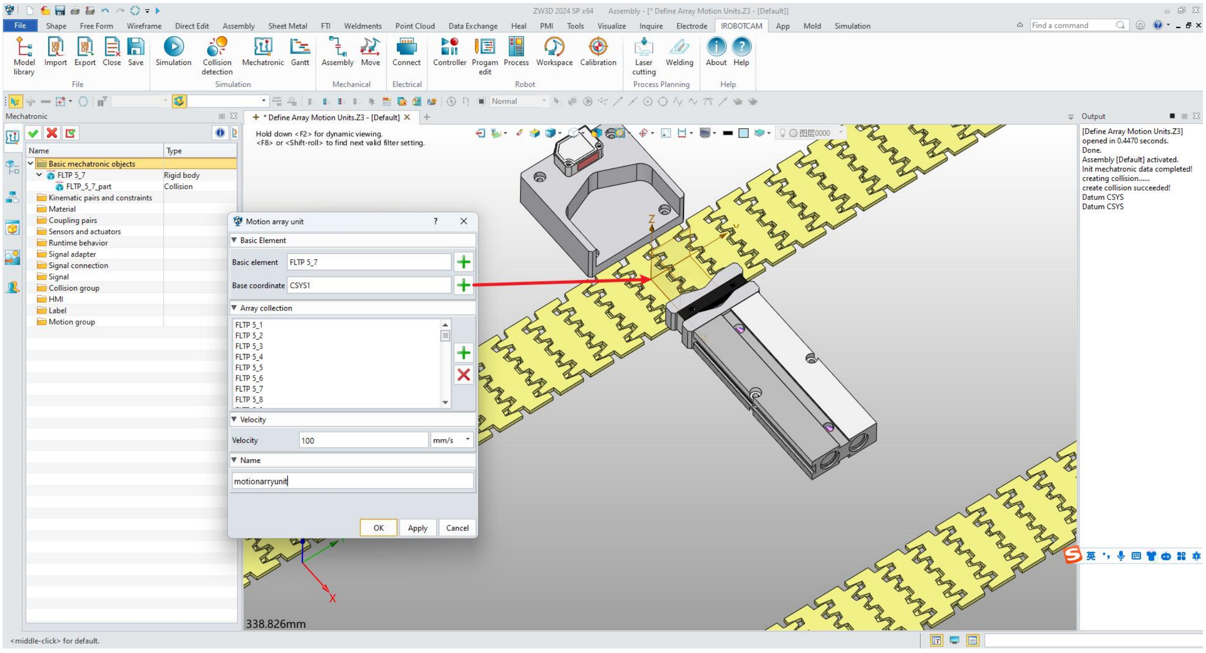



**Figure 7-10：Basic Object**

- Click Click the "+" button again to initiate the selection of all components comprising the conveyor belt as individual array elements.
- Carefully highlight and select each element within the 3D viewport by pressing the middle mouse button. (Figure 7-11)

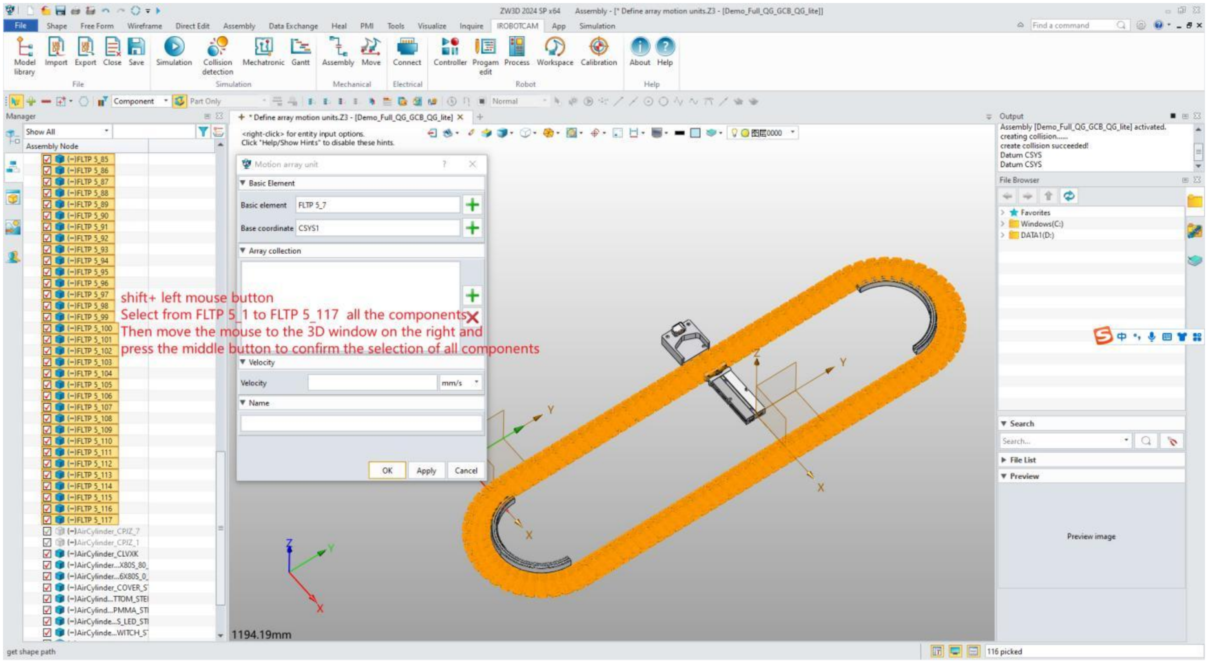



**Figure 7-11：Array Elements**

- Review and refine the details of each array element as necessary. (Figure 7-12)

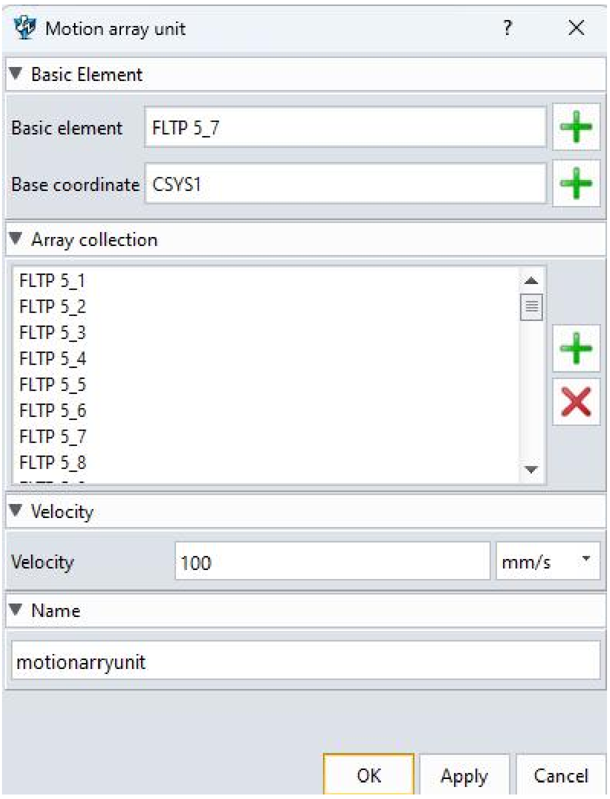



**Figure 7-12：motion array Unit**

- Click "OK" to finalize the creation of the motion array unit.

- Wait for the generation of the circular conveyor belt and all rigid and collision bodies on the conveyor belt under the basic mechatronic objects. (Figure 7-13)

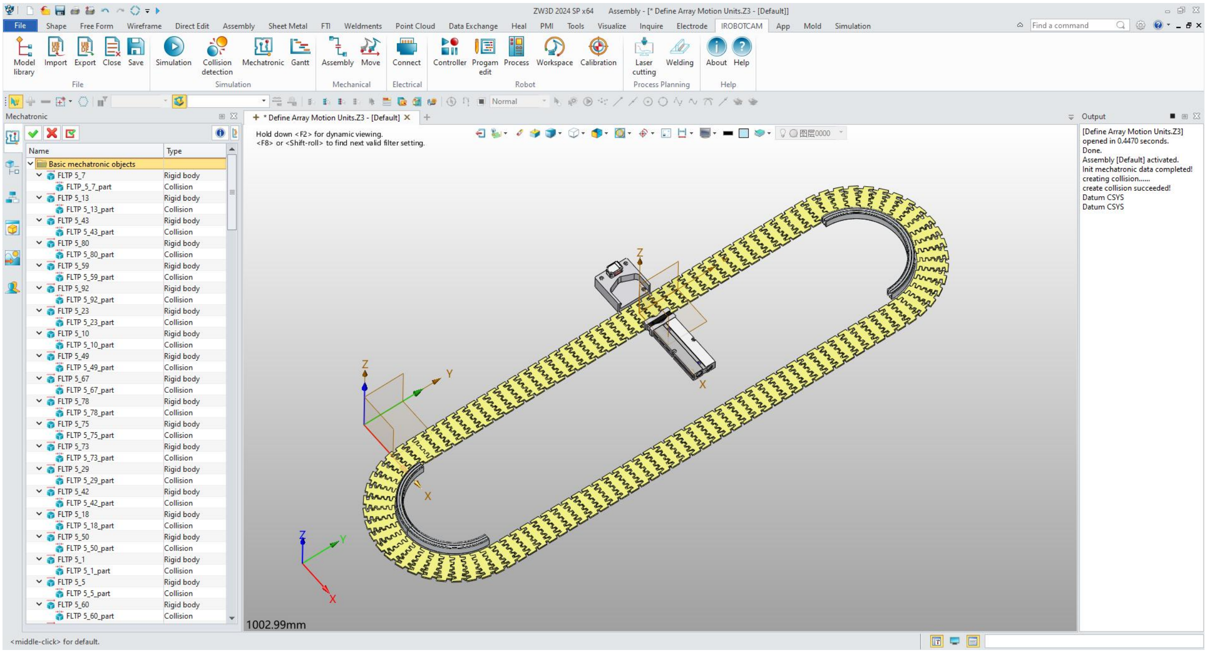


Figure 7-13：Array Elements

## Step 5：Add Signal Adapter

- Right-click within the "Signal Adapter" panel and choose "Add Signal Adapter". (Figure 7-14)

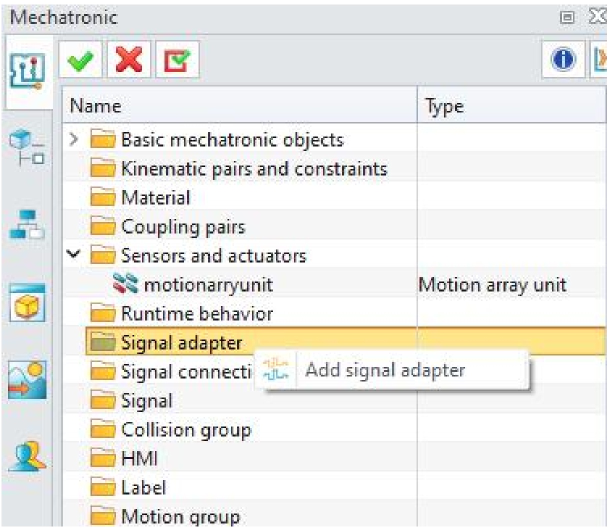



Figure 7-14：Add Signal Adapter

- Click the "+" button within the "Signal" section and select the newly created conveyor belt.

- Specify the desired signals, formulas, and names to define the operational behavior of the conveyor belt. (Figure 7-15)

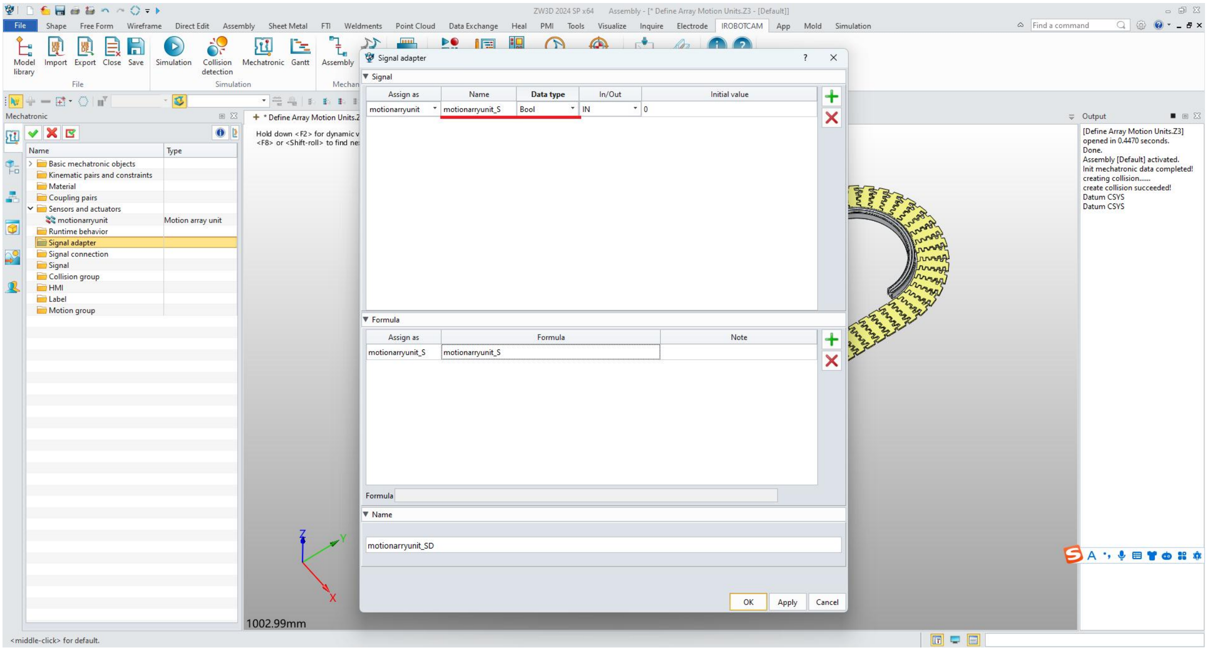


**Figure 7-15：Signal Adapter Information**

- Click "OK" to generate the corresponding signal adapter and signal objects.

## Step 6：Add simulation sequences

- Right-click within the "runtime behavior" panel and choose "Add Simulation Sequence" to open the "Simulation Sequence" window. (Figure 7-16)

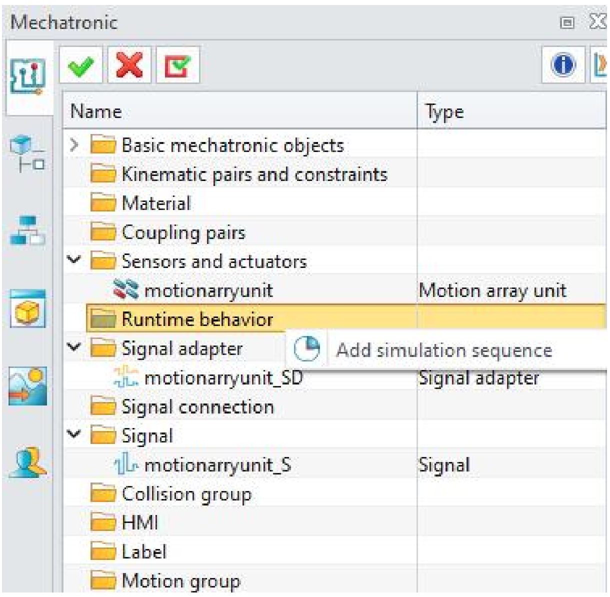


**Figure 7-16：Add simulation sequences**

- Click Define the "Start time" (eg, 0) at which the simulation should commence.
- Set the "Duration" (eg, 999999) to ensure the motion array unit operates for the desired duration.
- Uncheck the "Activation Signal" and "End Signal" options to initiate the simulation automatically.
- Specify the "Execution signal" to link the simulation sequence to the previously defined motion array unit.
- Set the "End Position" to "1" to ensure the motion array unit completes its full rotation.
- Assign a descriptive "Name"for the simulation sequence. (Figure 7-17)

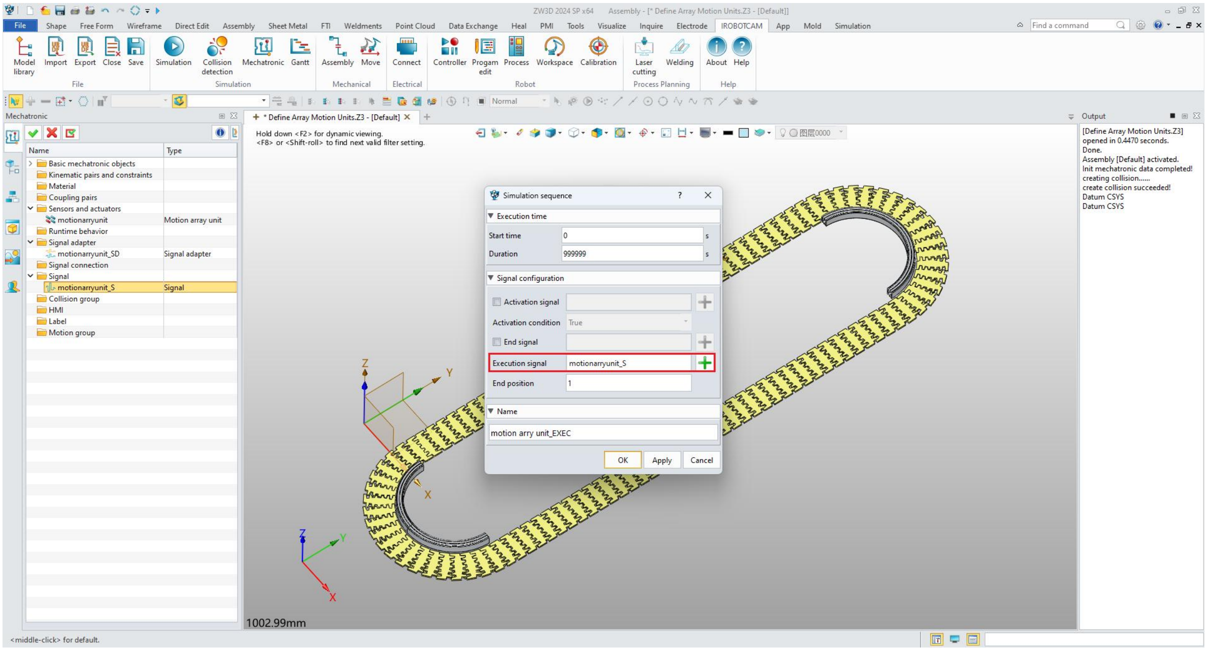


**Figure 7-17：T1_EXEC**

## Step 7：Starting the Simulation

- Click the "Simulation" button located under the "IROBOTCAM" tab. (Figure 7-18)

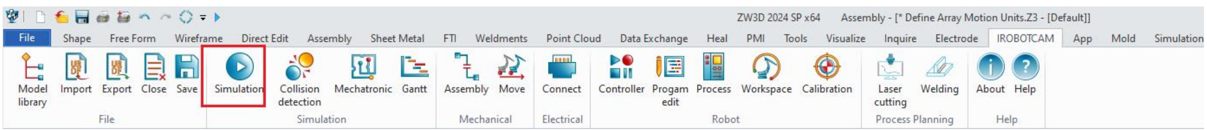


**Figure 7-18：Simulation Button**

- Click the "Start Simulation" button within the "Simulation Control" window to initiate the simulated motion of the conveyor belt. (Figure 7-19)

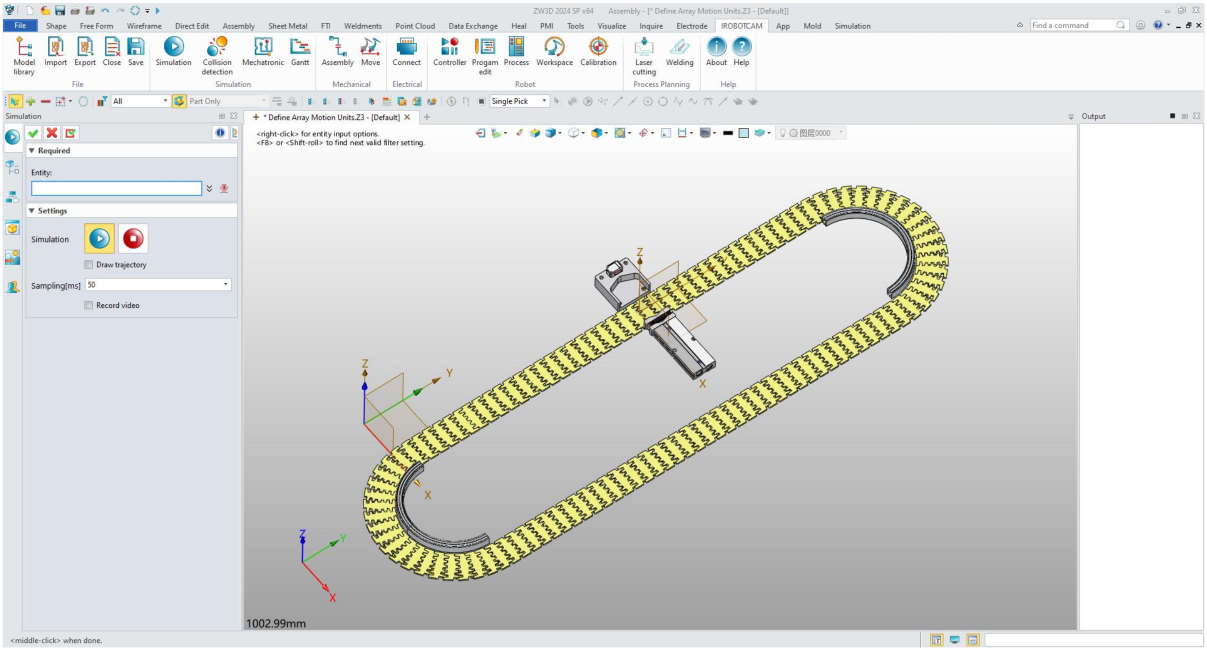


Figure 7-19：Simulation Verification

**Conclusion:**

By following this detailed and professional guide, you can effectively define and utilize motion array units within the iRobotCAM environment to simulate complex motion array unit systems with accuracy and precision. Adapt and modify the provided parameters and settings to achieve your specific desired behavior and explore the advanced functionalities offered by iRobotCAM for further customization and control.

# Chapter 8: Create Robot

This document provides a step-by-step guide on how to create a robot in the IROBOTCAM software.(Figure 8-1)

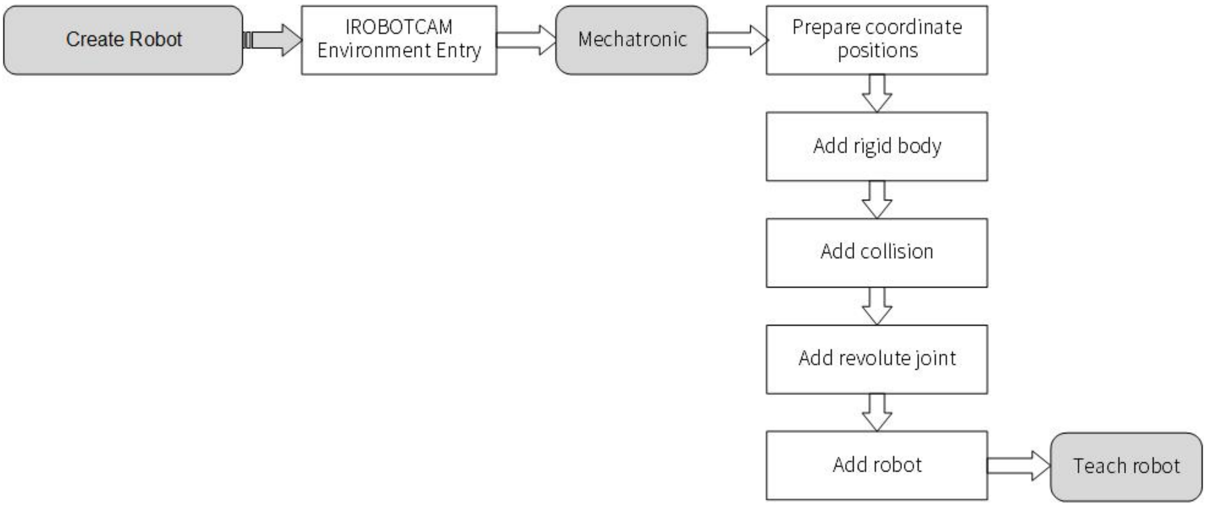



**Figure 8-1：Flow chart of Create Robot**

## Step 1：Open iRobotCAM project file

- Open iRobotCAM project file “Create Robot.Z3”. (Figure 8-2)

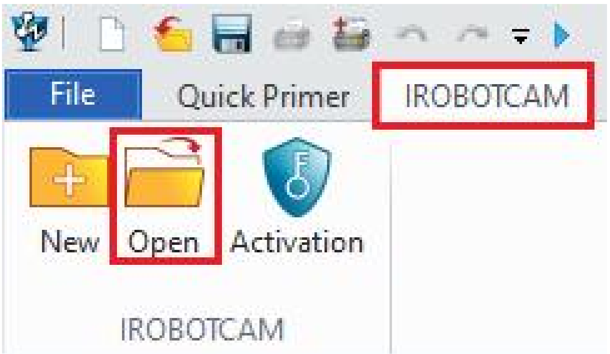



**Figure 8-2：Open iRobotCAM project file**

- View the details of interface components. (Figure 8-4)

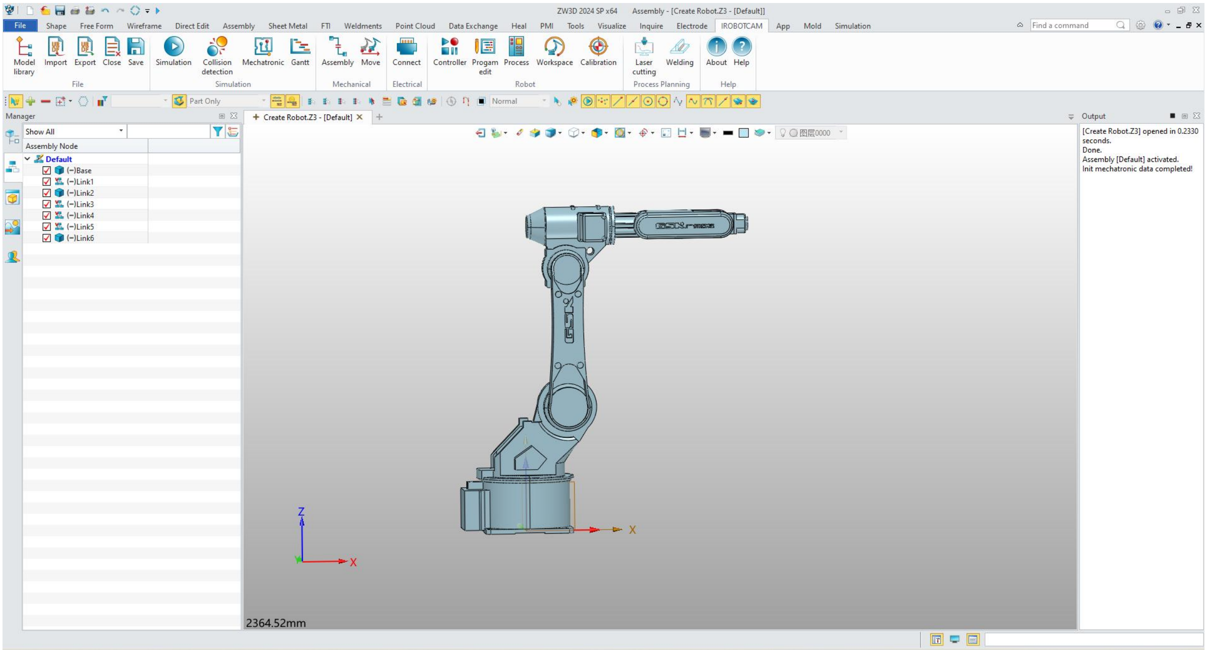


Figure 8-4：Component Details

## Step 2：Prepare coordinate positions

- Select the "Datum CSYS" option in the "Shape" tab. (Figure 8-5)

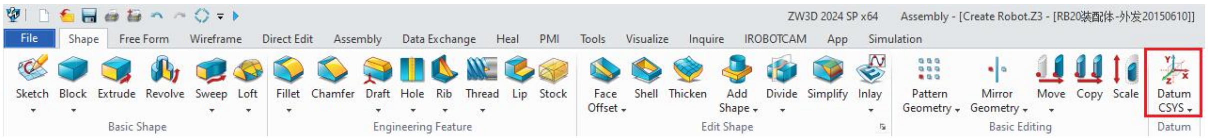


Figure 8-5：Datum CSYS

- Set "Datum CSYS" to "Dynamic". (Figure 8-6)

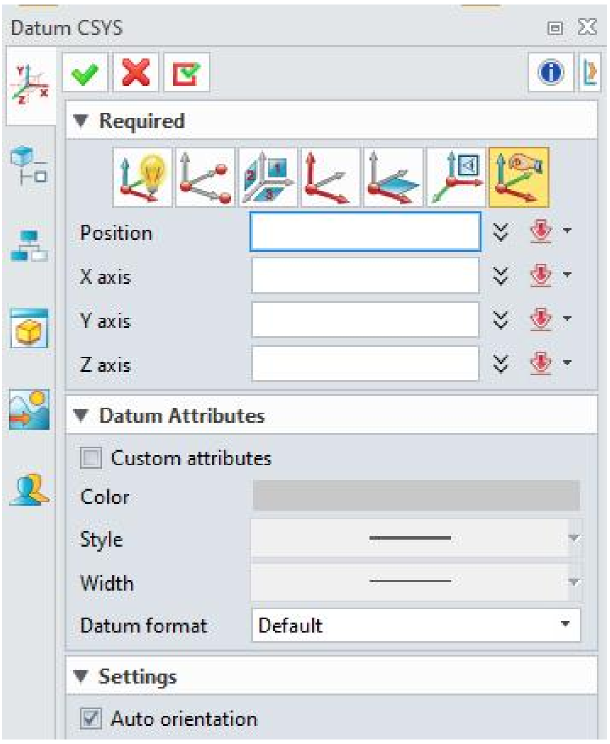



**Figure 8-6：Dynamic**

- Hide or uncheck "component Link1". (Figure 8-7)

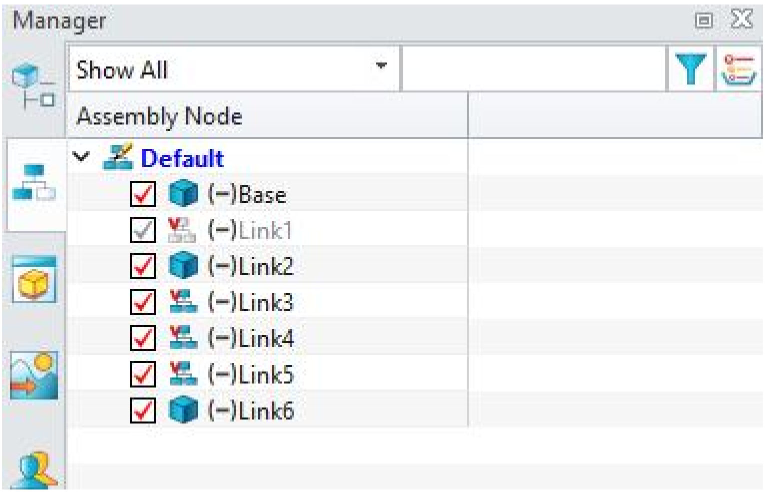



**Figure 8-7：Uncheck Link1**

- Create robot coordinate systems:
  - ✧ Link1 to Link6 joint rotation axis coordinate system.
  - ✧ Base coordinates.
  - ✧ Flange coordinates.
- Place and adjust the coordinate systems correctly, except for base and flange coordinates, which need only their Z-axis aligned. (Figure 8-8)

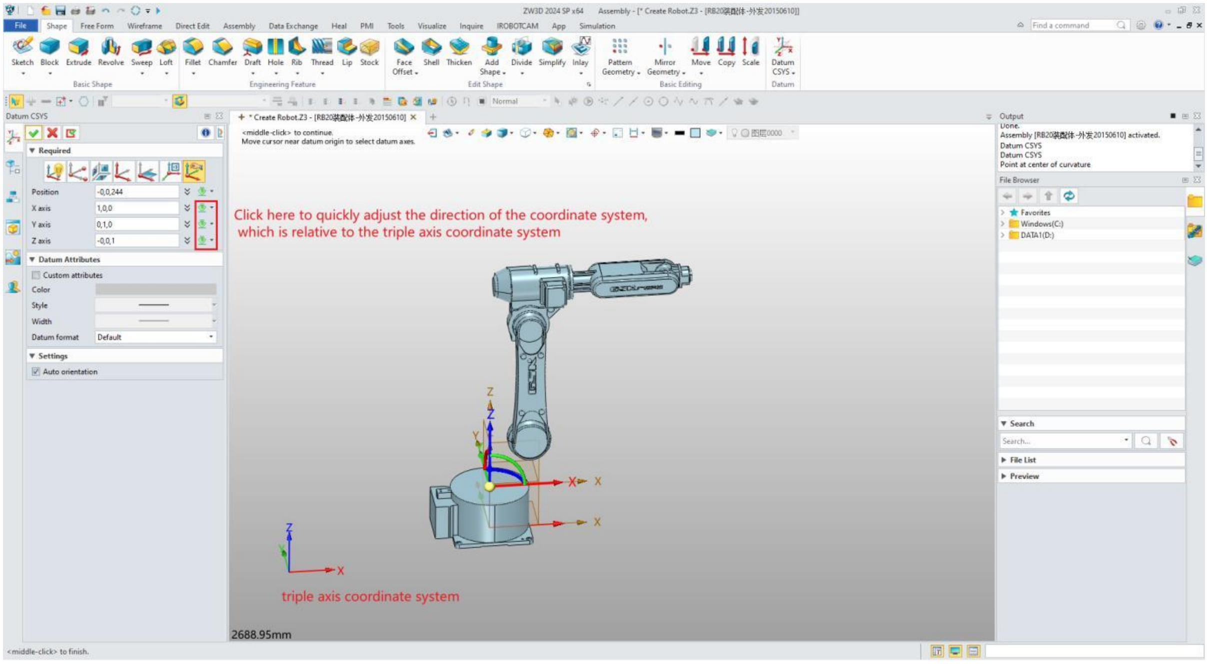



**Figure 8-8：Coordinate system construction**

- Create the base coordinate (CSYS8) using "Critical" and selecting the central triple axis coordinate system.
- Create the flange coordinate (CSYS7) at the desired position and direction on Link6.

(Figure 8-9)

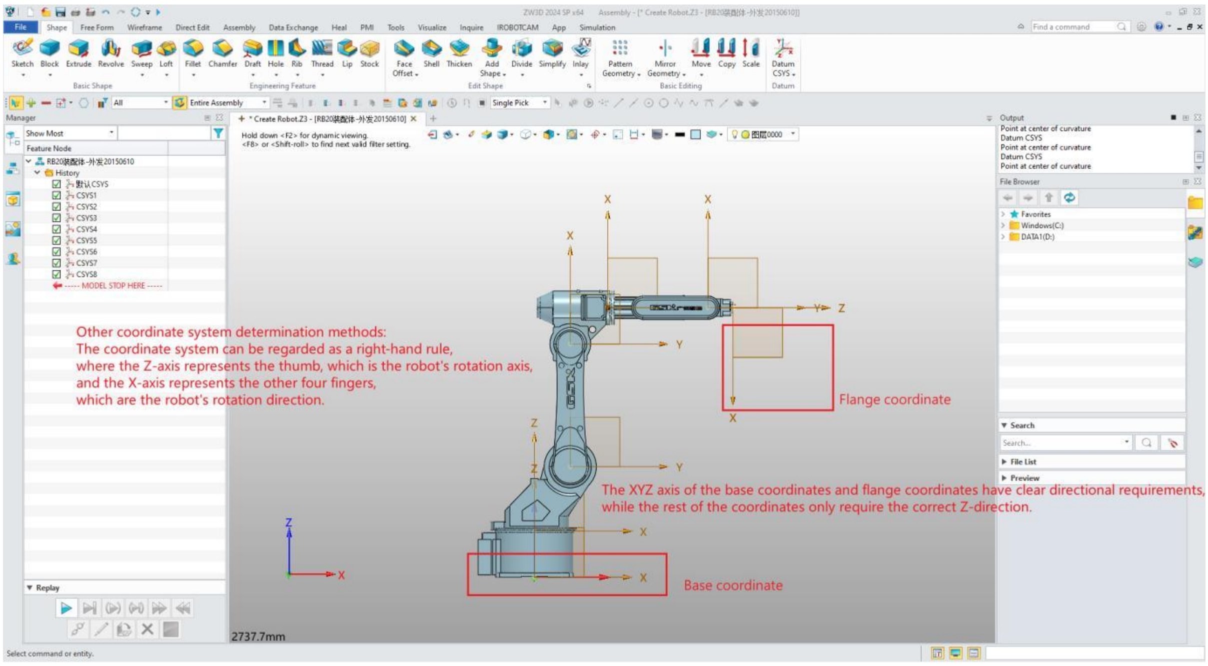



**Figure 8-9：Coordinate system position**

## Step 3：Create rigid bodies and collisions

- Go to the "IROBOTCAM" tab add click the "Mechatronic" button. (Figure 8-10)

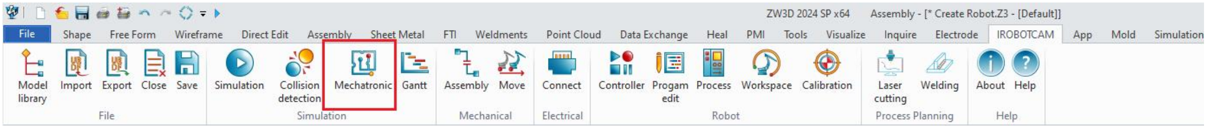



**Figure 8-10：Mechatronic**

- In the "Mechatronic" window, click on "Basic Mechatronic Objects".
- Right-click and select "Add Rigid Body" for each of the seven joint component objects. (Figure 8-11)

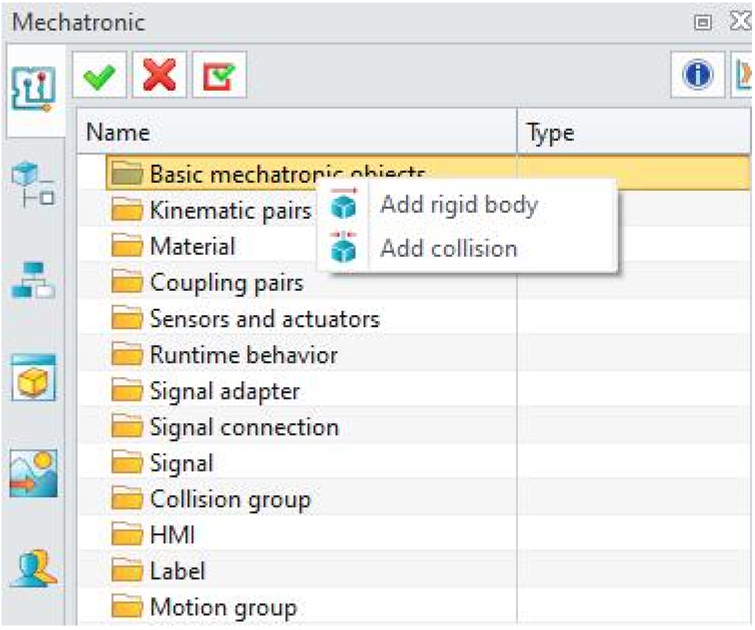



**Figure 8-11：Add Rigid Body**

- Property settings to 'Unchecked', and material to 'Default'. (Figure 8-12)

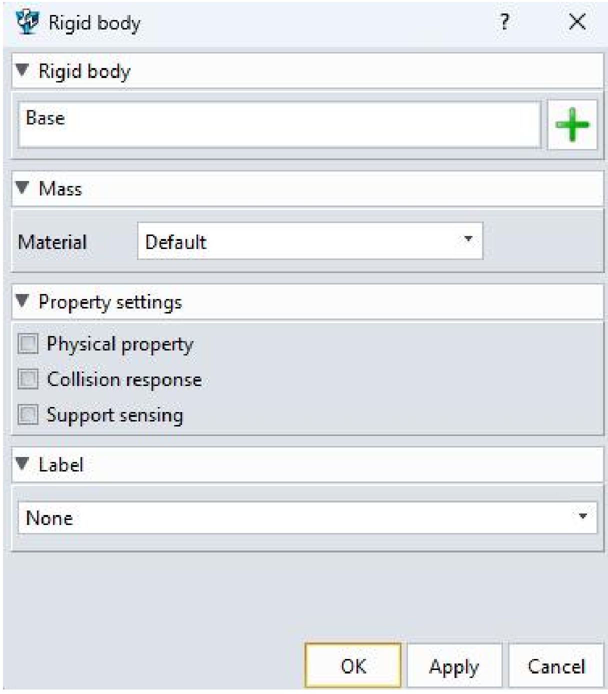



**Figure 8-12：Rigid body parameters**

- After adding rigid bodies to the seven joint objects, the component details will be displayed. (Figure 8-13)

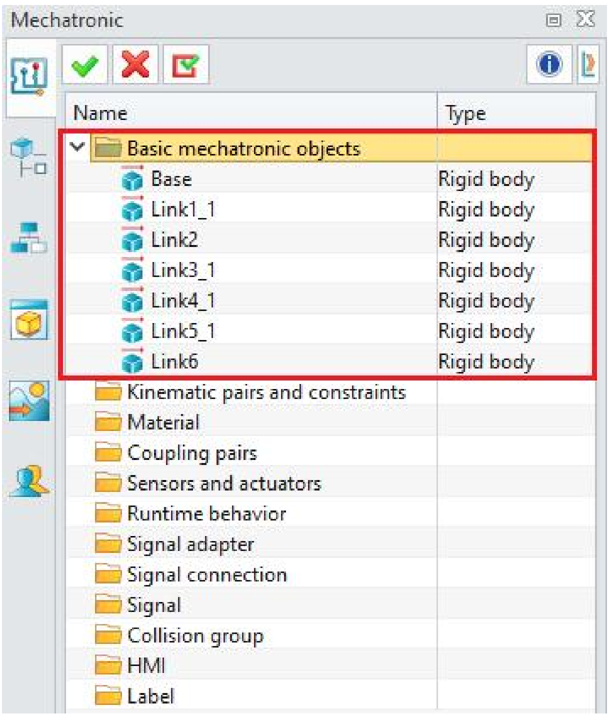



**Figure 8-13：seven rigid bodies**

- Right-click and select "Add collision" for each of the seven component objects.
- Select "convex hull" for the collision shape.

- Property settings to "unchecked", and material to "default". (Figure 8-14)

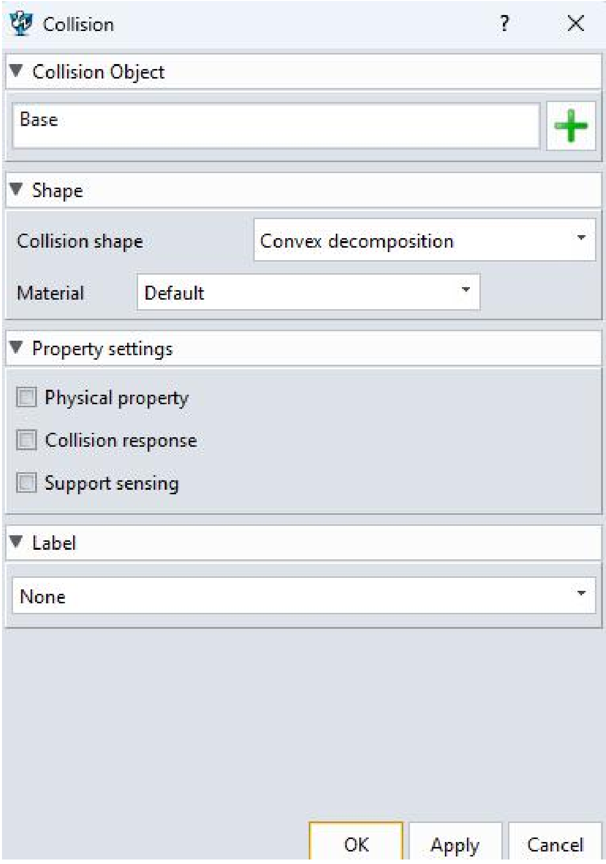



**Figure 8-14：Collision Parameters**

- After adding seven objects to the collisions, the component details will be displayed. (Figure 8-15)

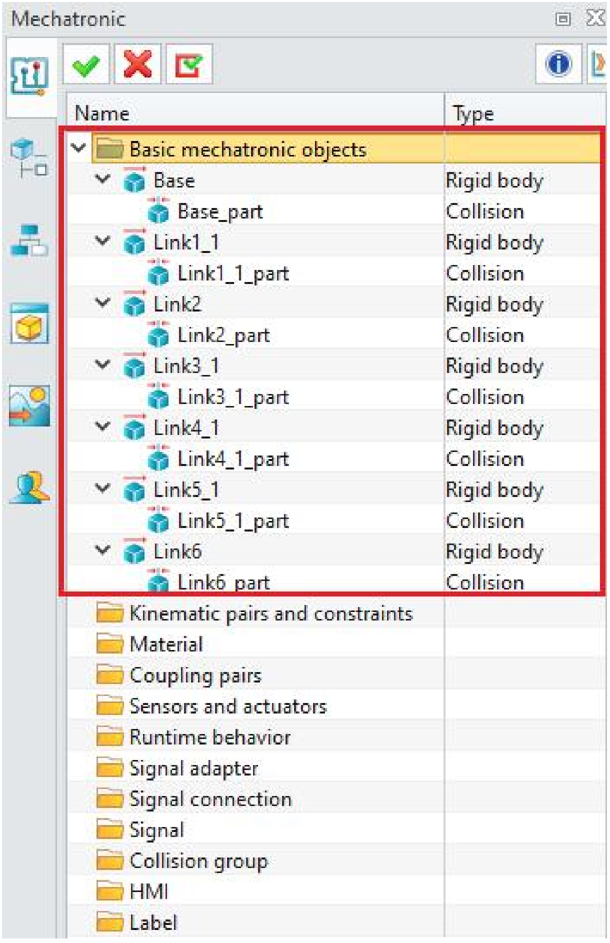



**Figure 8-15：Seven Collisions**

## Step 4：Create joints

- In the "Mechatronic" window, click "Kinematic Pairs and Constraints". Right-click and select "Add Revolute Joint". (Figure 8-16)

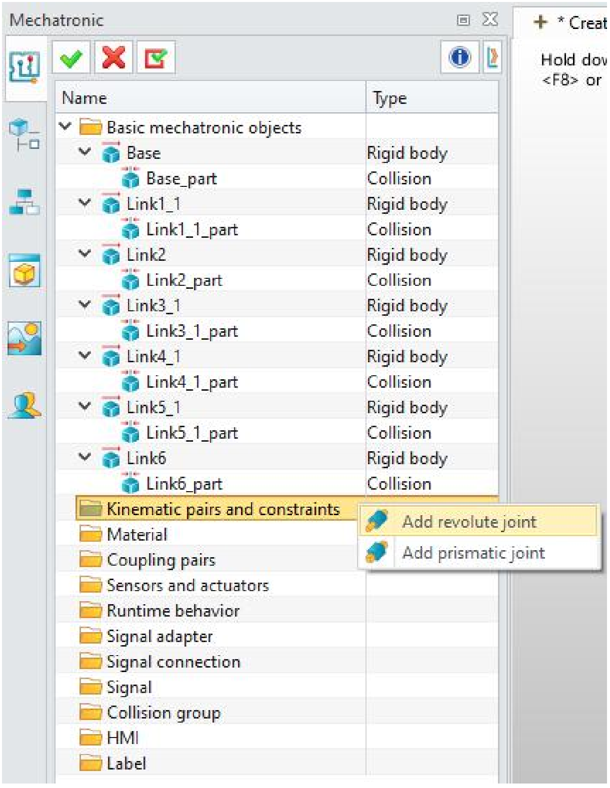



**Figure 8-16：Add Revolute Pairs**

- Create six revolute joints for each robot joint, specifying parameters for each. (Figure 8-17)

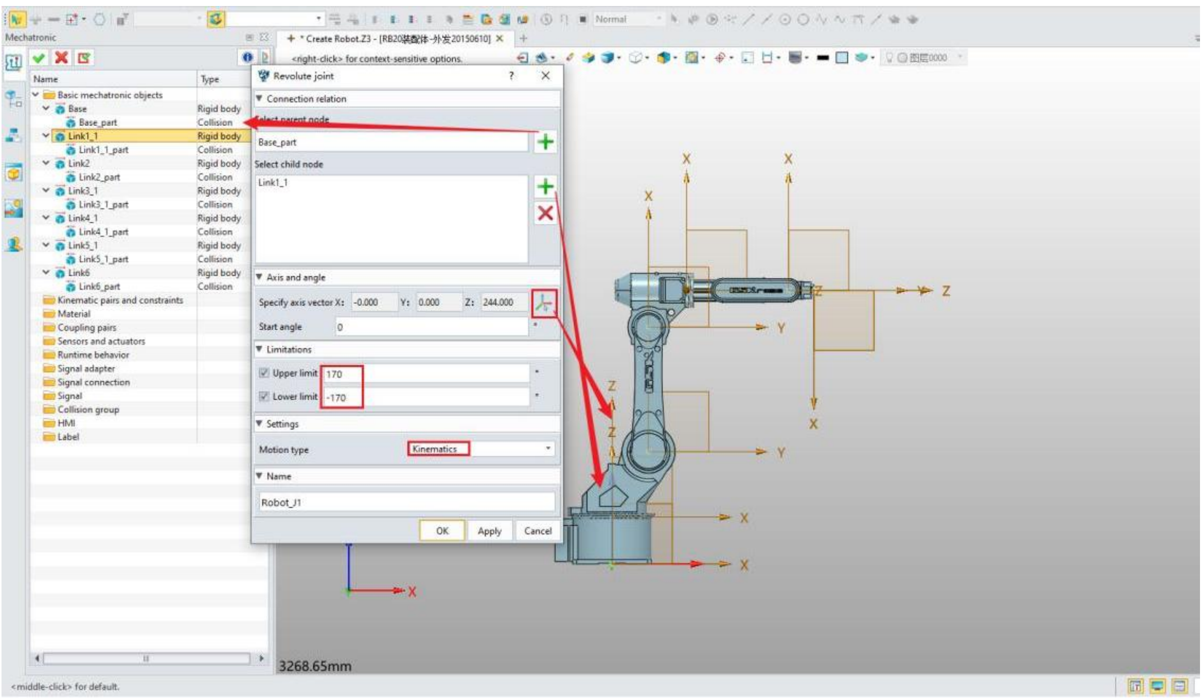



**Figure 8-17：Robot_J1**

- Create the six joints of the robot in sequence, with the following parameters:

- ✧ Child node rigid body and parent node collision.
- ✧ Specified axis vector (Datum CSYS1-6).
- ✧ Upper and lower limits of the joint.
- ✧ Start angle ("0").
- ✧ Joint type ("Kinematic").
- ✧ Name (Robot_J1-6).
- Six joint hinge sub interface details. (Figure 8-18)

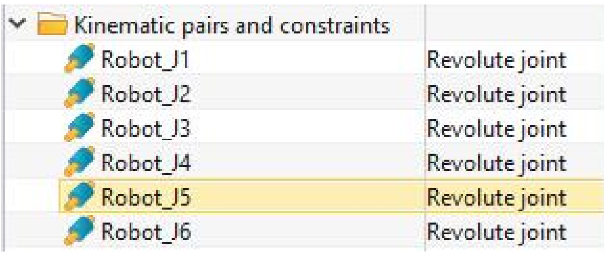



**Figure 8-18：Six Revolute Joints**

## Step 5：Create robot

- In the "Mechatronic" window, click "Sensors and actuators", right-click and select "Add Robot". (Figure 8-19)

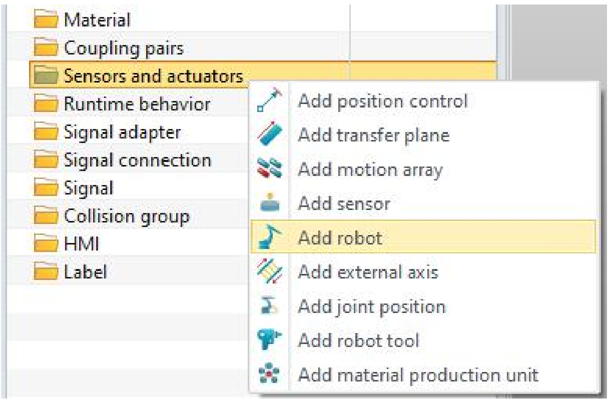



**Figure 8-19：Add Robot**

- Enter the robot name "GSK".
- Add joints in order with "positive" rotation direction and "0, 0, -90, 0, 0, 0" offset values.
- Choose "GSK_Base" for "Base Coordinate CSYS8" and "GSK_Flange" for "Flange Coordinate CSYS7". (Figure 8-20)

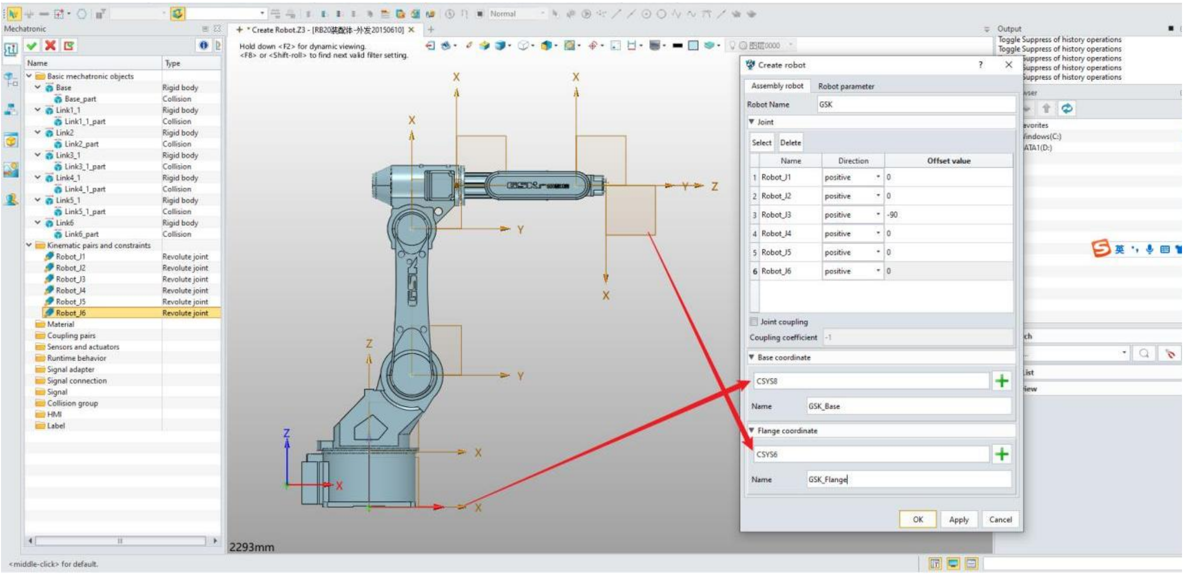


**Figure 8-20：Assembly Robot**

- Robot parameter settings, select “analytical solution” for the algorithm and confirm other default options. Click OK to create the robot. (Figure 8-21)

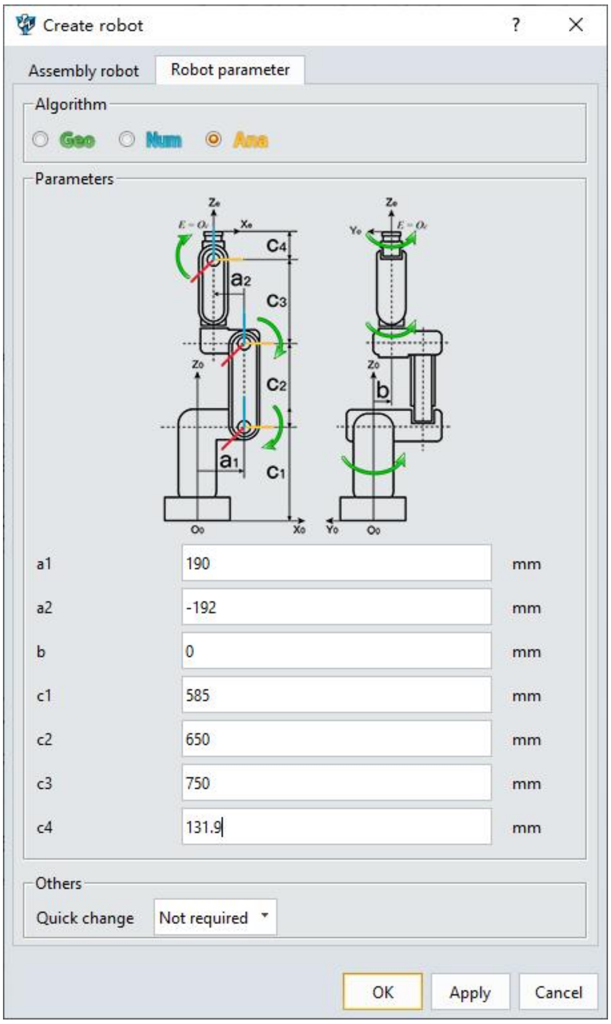


**Figure 8-21：Create Robot**

- Wait for the created robot to appear in the Mechatronic on the left, and the robot will be successfully created. (Figure 8-22)

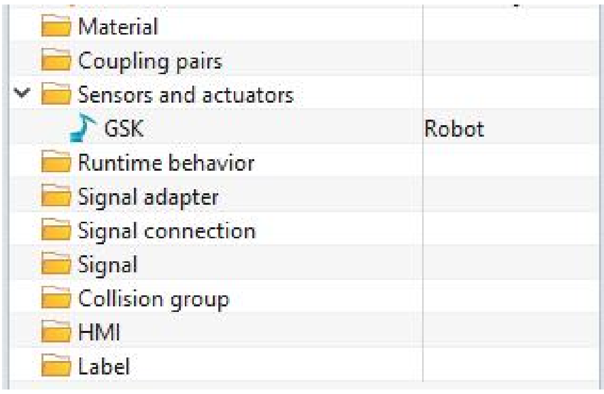



**Figure 8-22：Robot creation completed**

## Step 6：Measure Robot parameters

- Click "Inquire" and select “Distance”. (Figure 8-23)

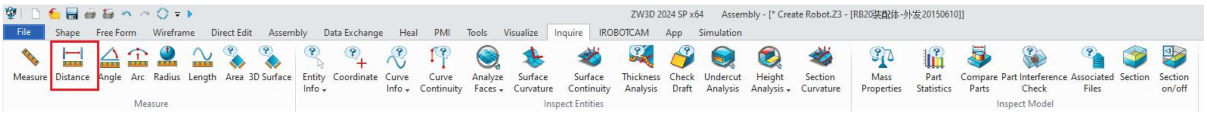



**Figure 8-23：Distance Option**

- Measure the lengths of a1, a2, b, c1, c2, c3, and c4 according to the robot parameter diagram.
- Enter the corresponding values in the "Add Robot" window.

  (Figure 8-24)

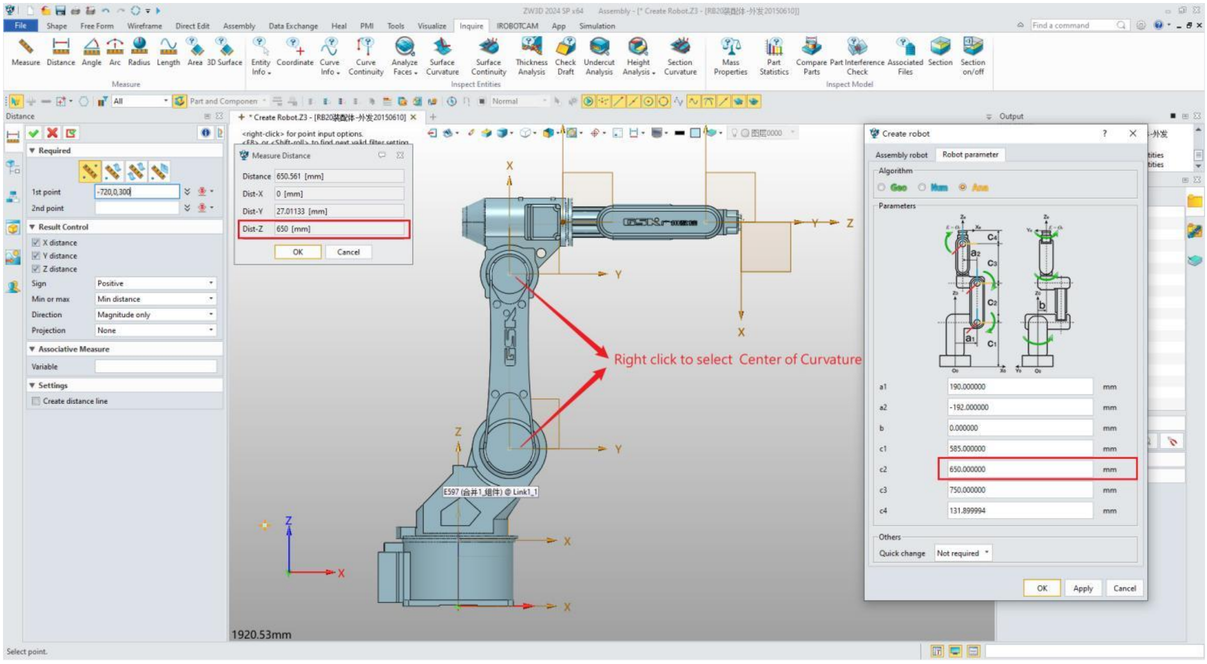



Figure 8-24：Robot parameter measurement

## Step 7：Teach robot

- Right-click on the created robot and select "Teach Robot". (Figure 8-25)

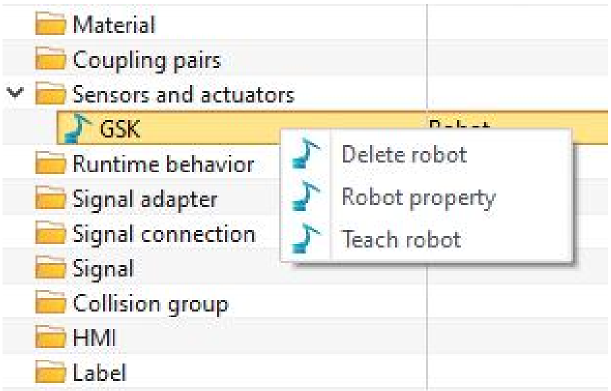



Figure 8-25：Teach Robot

- Click "Drag Teach" and Drag the XYZ axis to check if the robot is moving normally. (Figure 8-26)

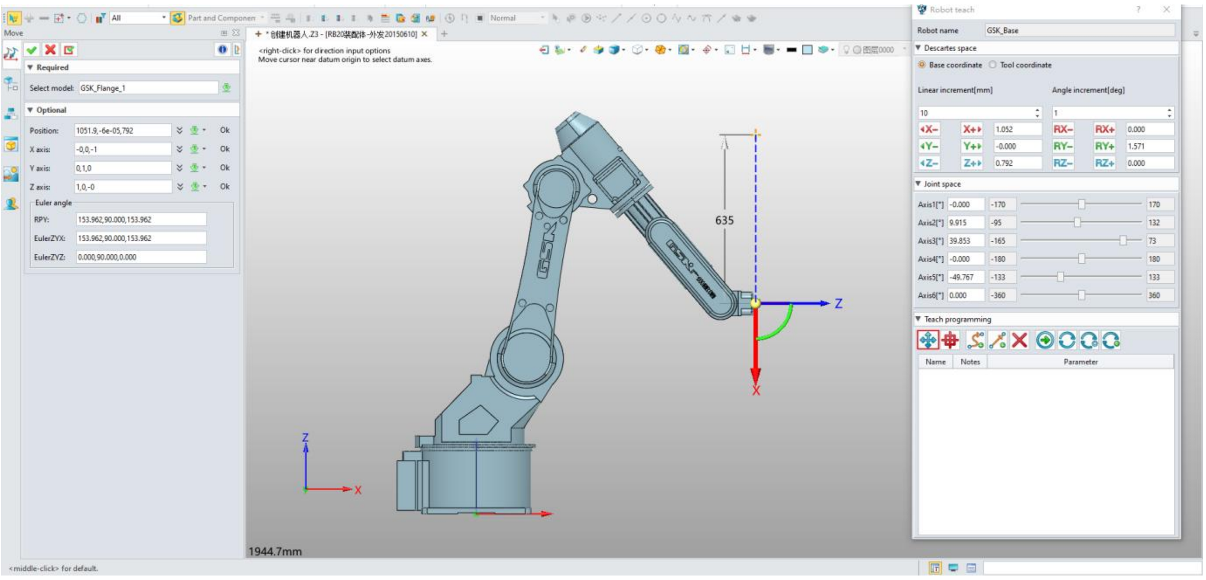


Figure 8-26：Drag Teach

# Chapter 9: Teach Program Draw Arc

Draw Arc: A Comprehensive and Informative Guide with Advanced Techniques

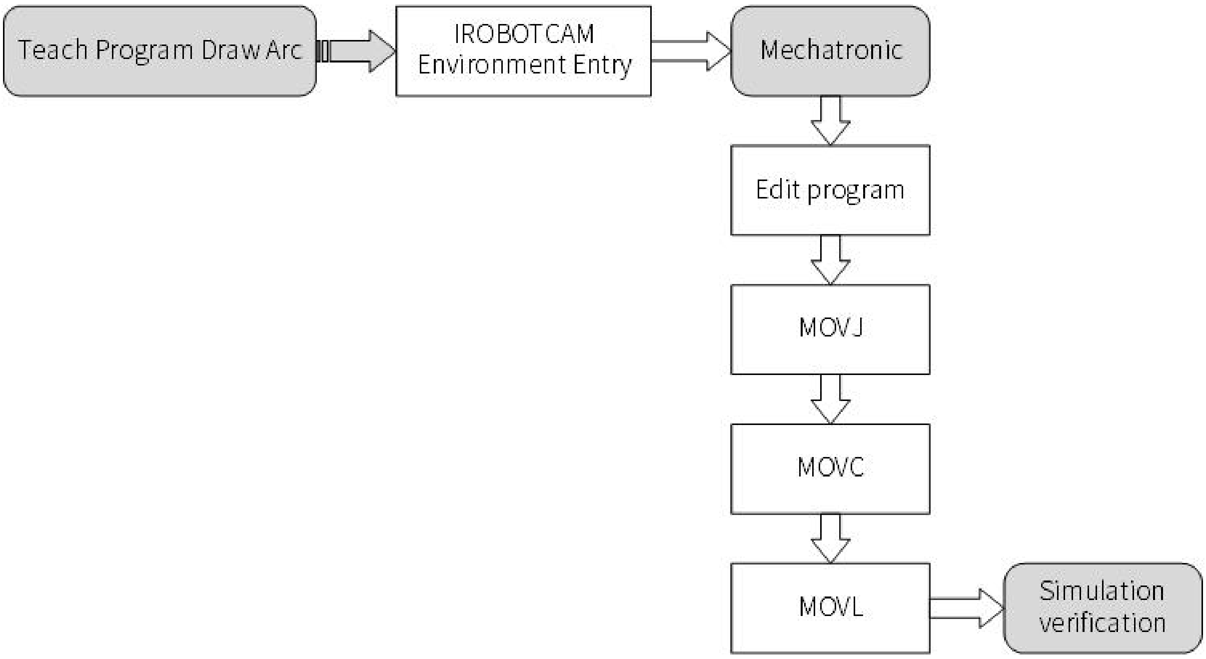



**Figure 9-1：Flow chart of teach program-draw arc**

## Step 1：Opening and Activating the Assembly

- **Entering iRobotCAM:** Launch the iRobotCAM software by opening the file "Draw Arc.Z3". This will initiate the iRobotCAM environment, providing you with the necessary tools for robot programming and simulation.

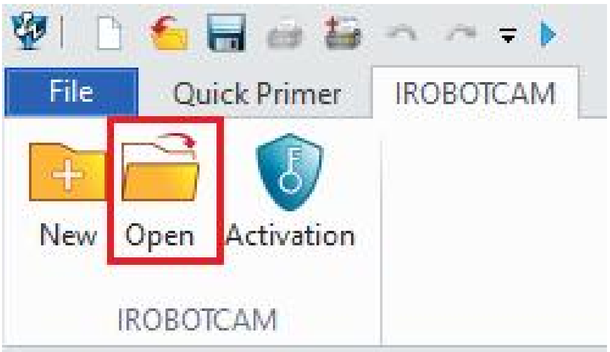



**Figure 9-2：Open iRobotCAM project file**

- Open interface details.

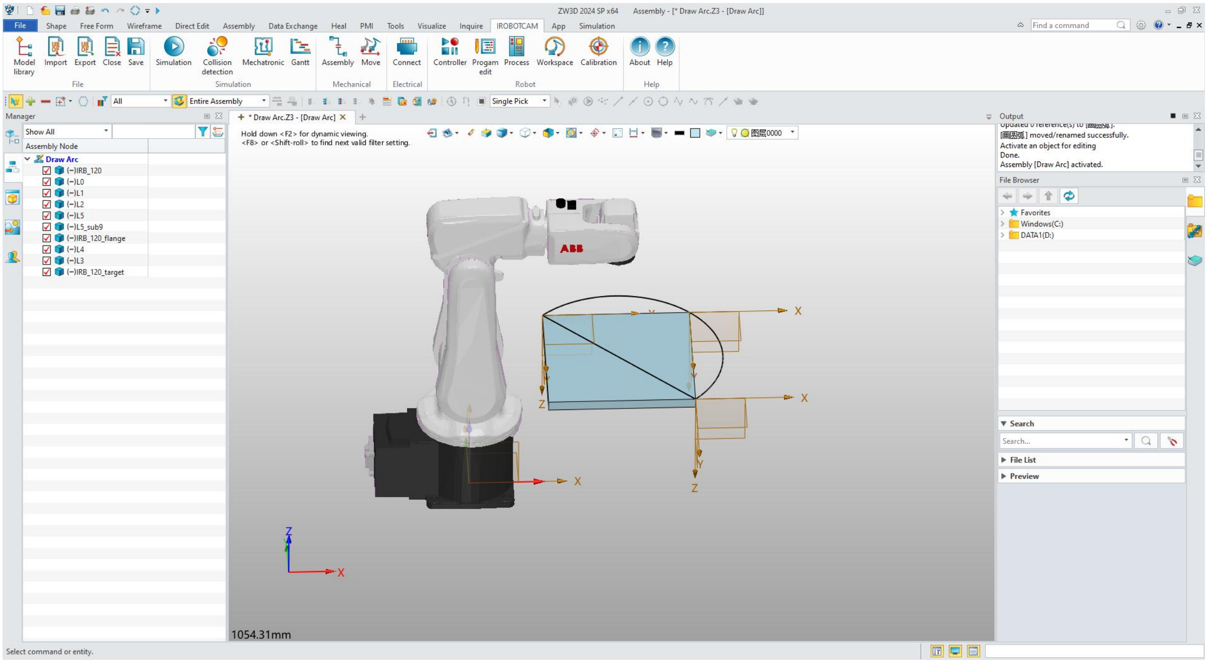


Figure 9-4：Component Details

## Step 2：Entering Teach Mode and Adding Marker Points

- Enter Mechatronic: Navigate to the "Mechatronic" option within the iRobotCAM interface.(Figure 9-5)


Figure 9-5：Mechatronic

- Right Initiating Teach Mode: Right-click on the robot and select "Teach Robot" from the context menu. This will activate the Teach Mode, allowing you to record robot movements and program its trajectory. (Figure 9-6)

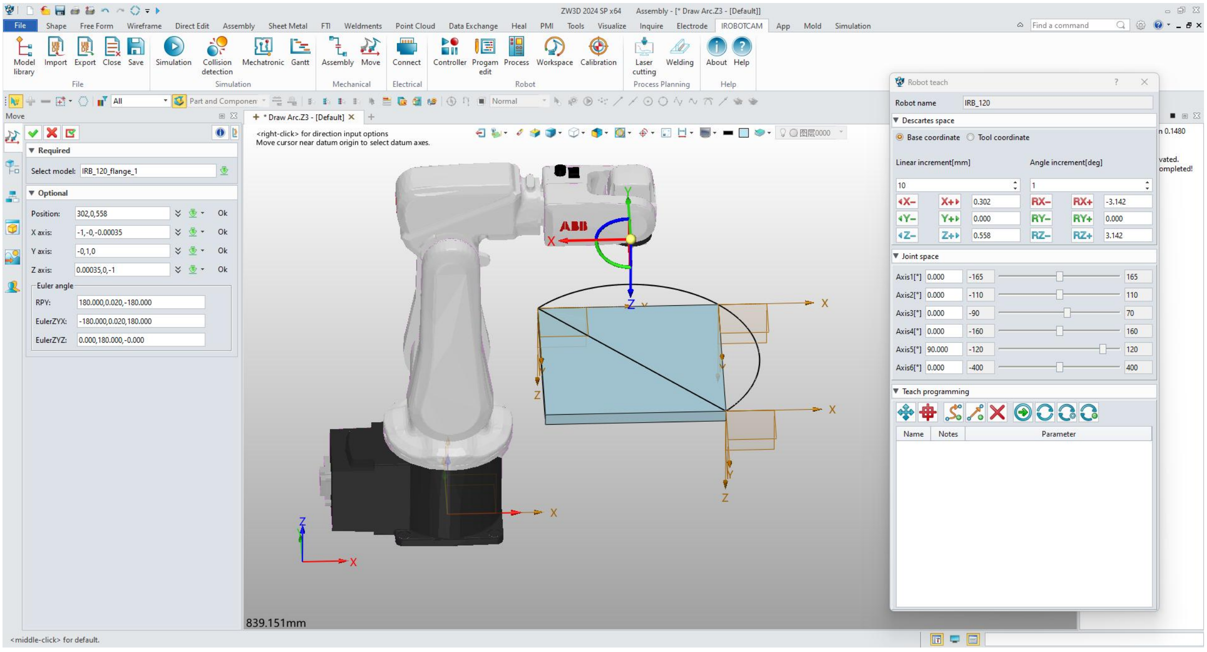



Figure 9-6：Teach Robot

## Step 3：Add marker points

- Record Joint Point: Click on "Mark point" in the "Robot Teach" window.
- Select the CSYS1 reference coordinate system.
- Select the suitable item in the pop-up box, the robot will move to that position, and then click the "Record Joint point" button. (Figure 9-7)

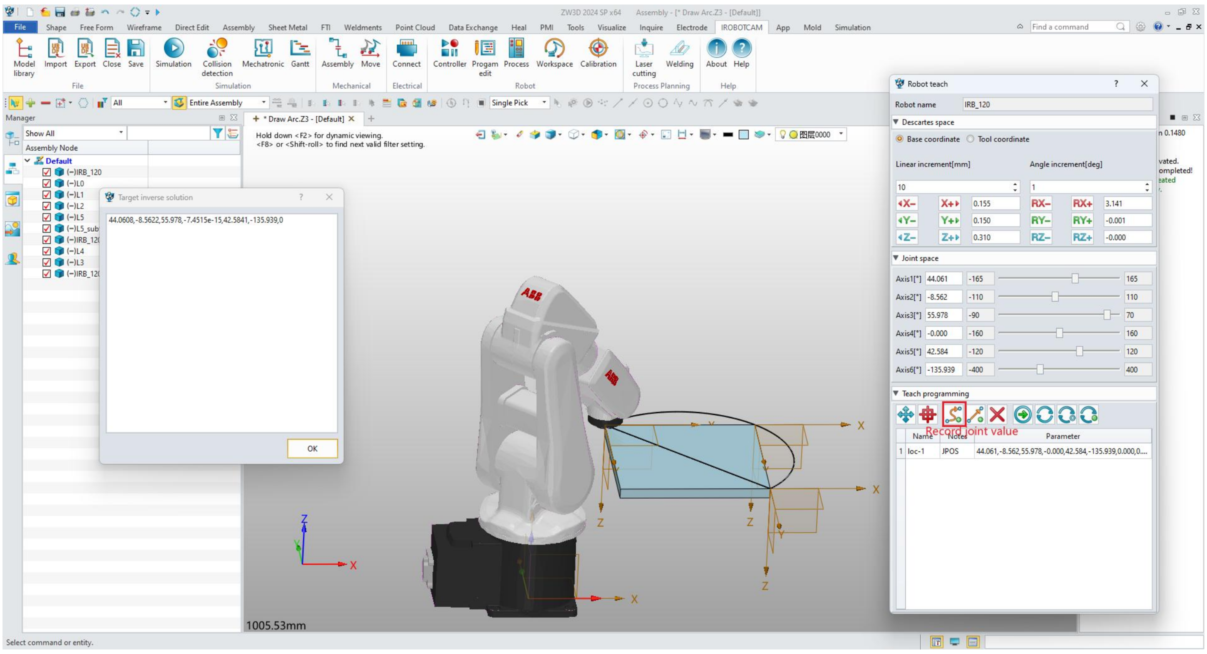



Figure 9-7：Add Marker Points

- Record End Point: Click on "Mark Point",Select "CSYS2" and "CSYS3", and click on the "Record End point" button to add a record. (Figure 9-8)

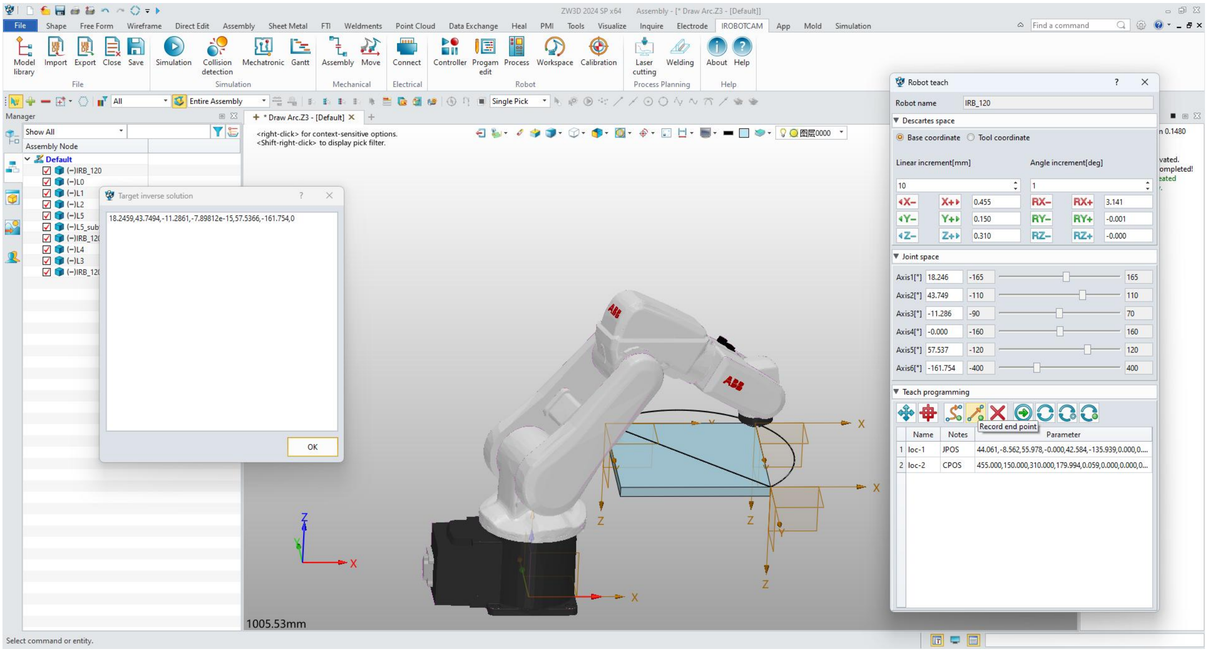


**Figure 9-8：Add markings for points 2 and 3**

- Arrow Record Point: Click loc-1 in the "Robot Teach" window.
- click the "arrow" to move to the marked point.
- move the robot to the position of point 1 again, click the "Record End point" button to mark, add Record point 4, and finally obtain 4 record point.
- In subsequent program editing, the same point cannot have both MOVJ and MOVL instructions, so add different Record point to the same position to avoid program instruction conflicts. (Figure 9-9)

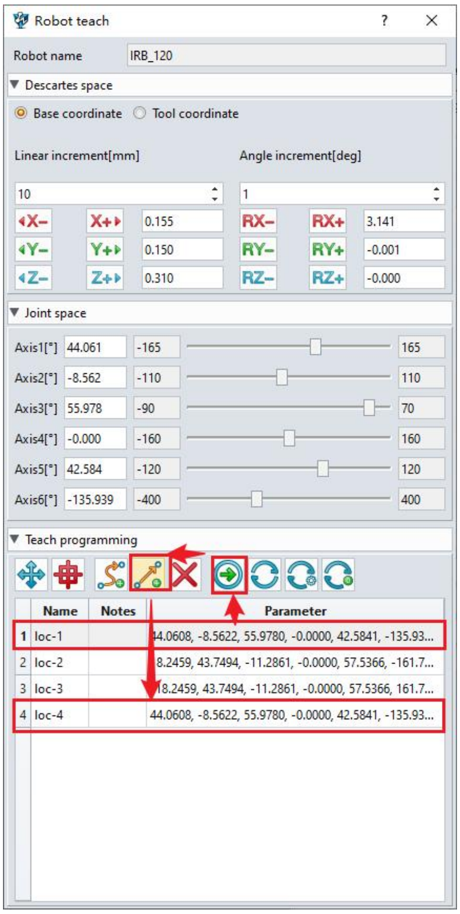


Figure 9-9：Marking of point 4

- Go Home Point: click Move to Home button to bring the robot to its original position. (Figure 9-10)

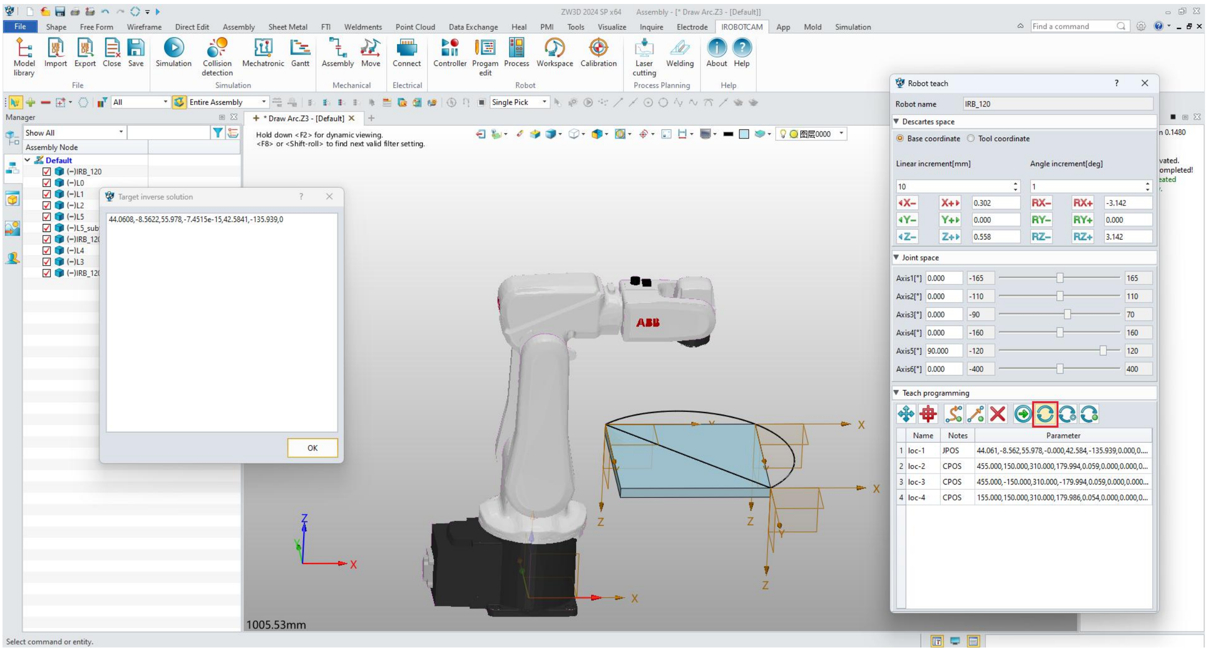


**Figure 9-10：Move to Home**

## Step 4：Program Editing

- On the "IROBOTCAM", click the "Program Edit" button. (Figure 9-11)

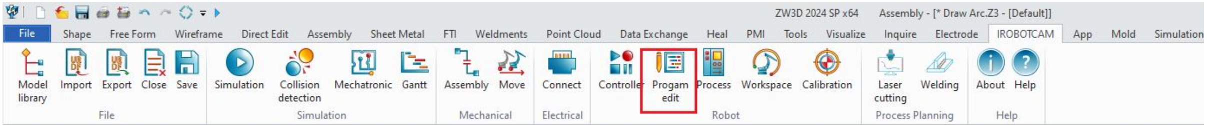



**Figure 9-11：Program Edit**

- In Edit Selected Program: In the "Program Edit" window, right-click on "program1" and select "Edit Program" to open the program parameter interface. (Figure 9-12)

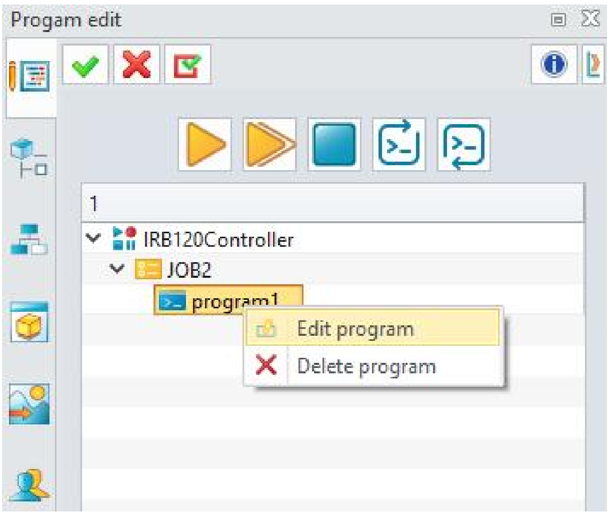



**Figure 9-12：Edit Program**

- Add MOVJ Command: In the "Program Parameters" window, select the "MOVJ" in the Motion command, add the MOVJ instruction, and set the parameters as shown in the figure. The target point is loc-1, and other parameters remain default. Click OK. (Figure 9-13)

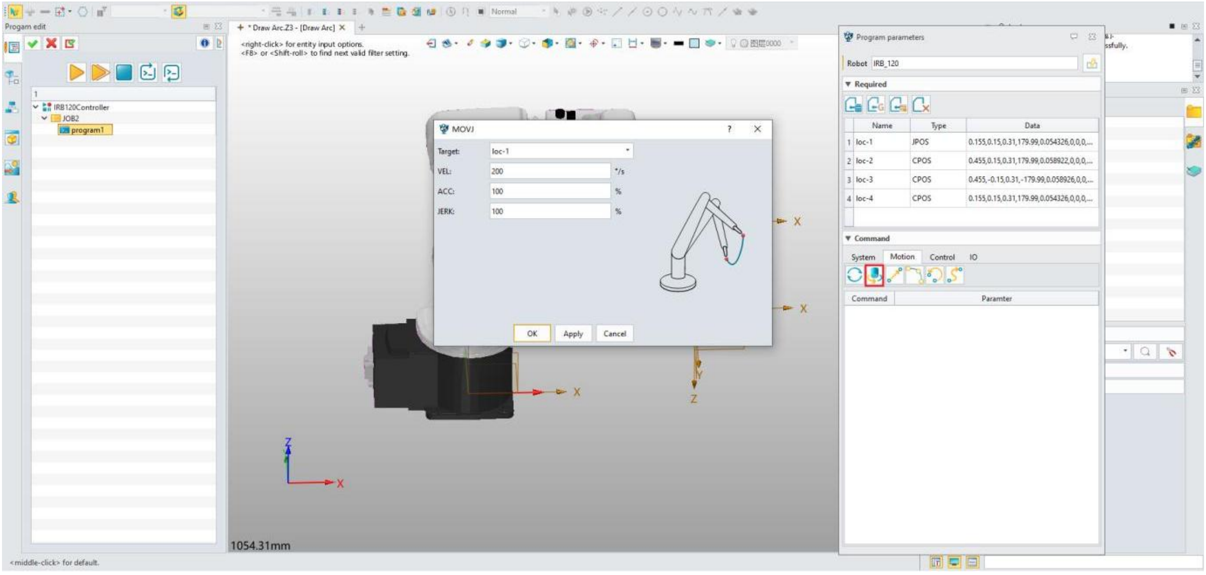


Figure 9-13：MOVJ

- Add MOVA Command: Then add the MOVA instruction, with parameter settings as shown in the figure. The target point is "loc-3", the middle point is "loc-2", and other parameters remain default. Click OK. (Figure 9-14)

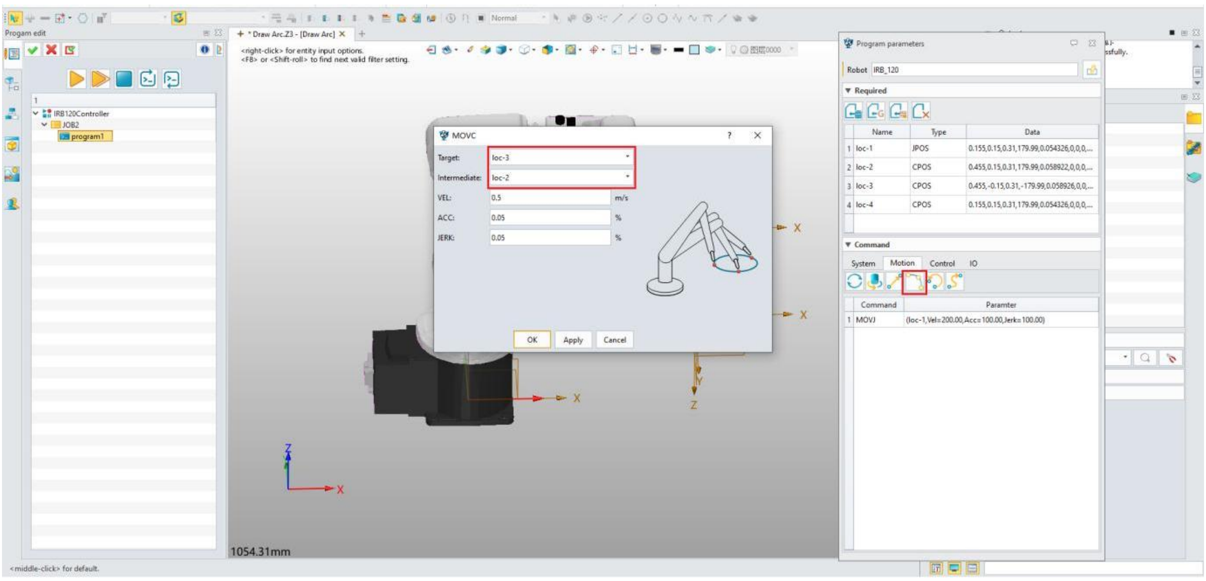


Figure 9-14：MOVA

- **Add MovL Command:** Then add the MOVL instruction, with the parameter settings shown in the figure, the target point being loc-4, and other parameters remaining at default. (Figure 9-15)

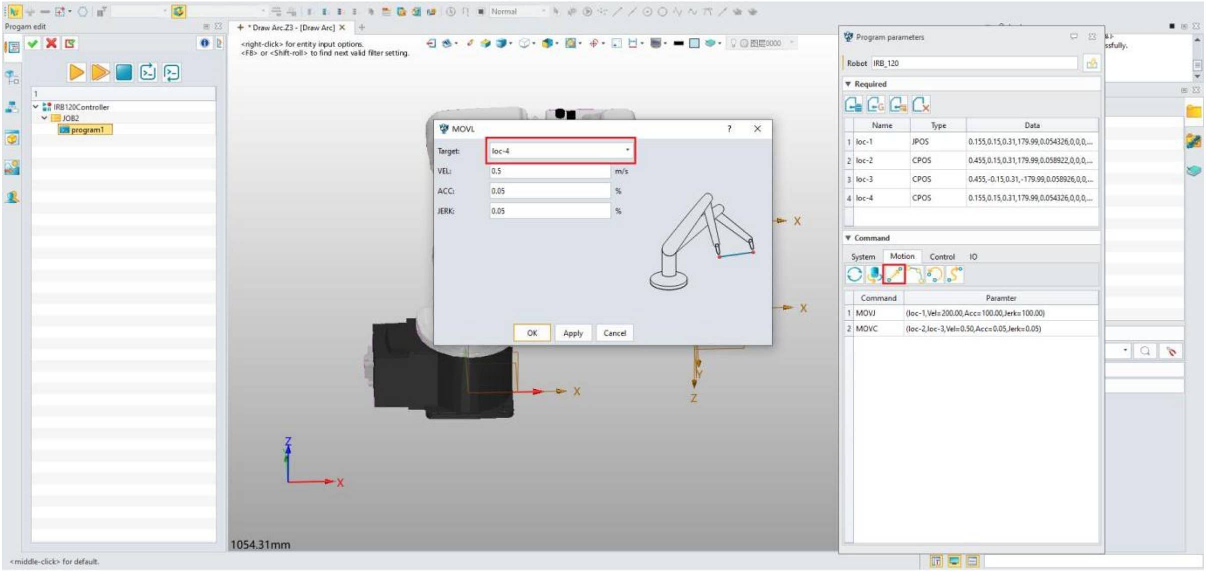


Figure 9-15：MOVL

## Step 5：Simulation Verification

- **Initiating Simulation:** Click on the "Run" button within the "Program Edit" window. This will initiate the simulation process and execute the programmed instructions for the robot. (Figure 9-16)

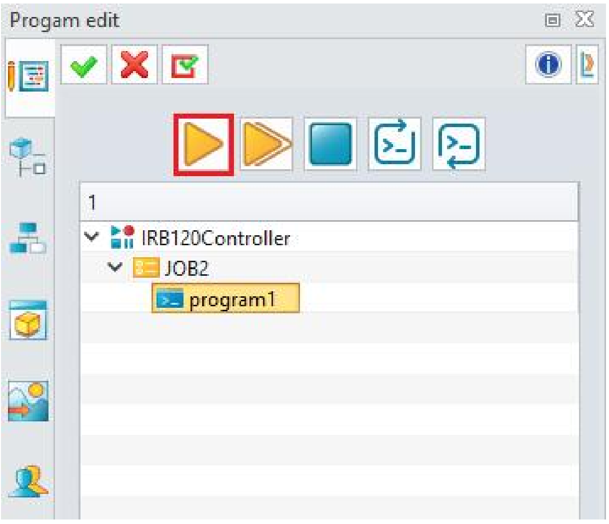



Figure 9-16：Simulation Button

- **Analyzing the Motion:** Observe the robot's movement during the simulation and compare it to the desired outcome. Pay close attention to the robot's path, speed, and overall behavior. (Figure 9-17)

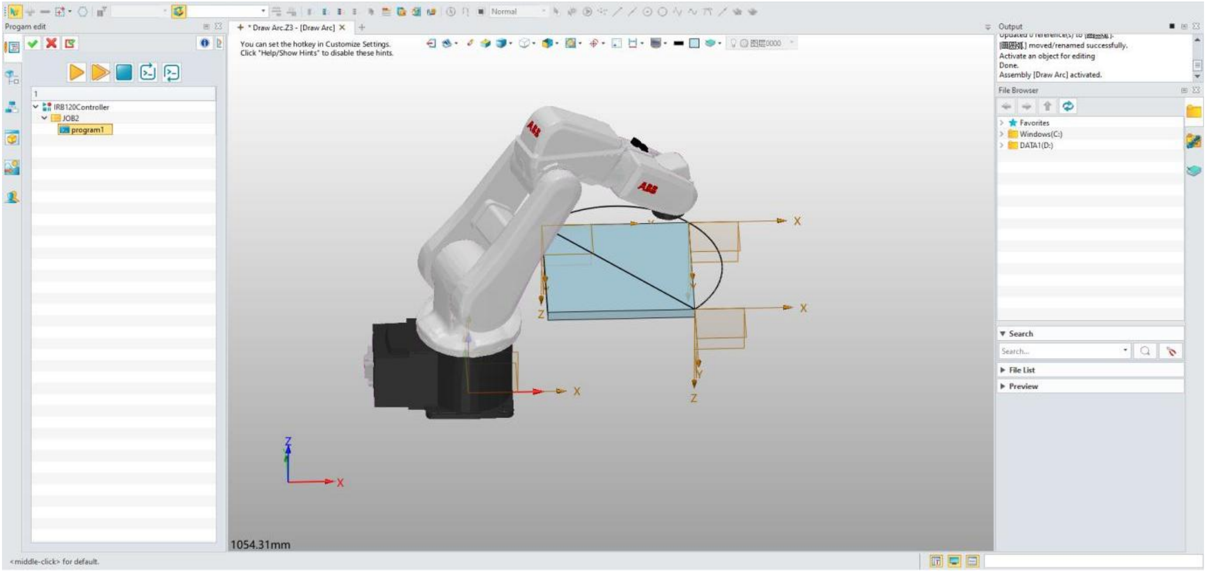



Figure 9-17：Simulation Verification

# Chapter 10:Robot Pick And Place

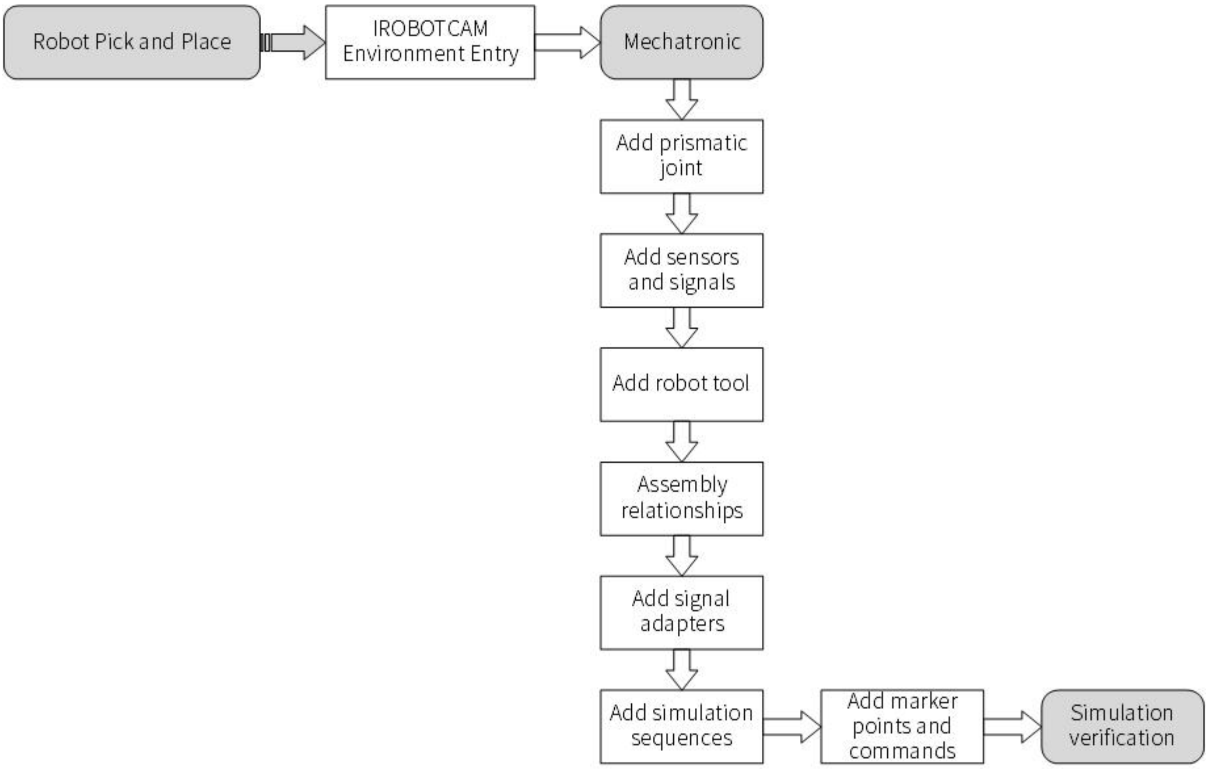



**Figure 10-1：Flow chart of robot pick and place**

## Step 1：Open iRobotCAM project file

- Open iRobotCAM project file "Robot Pick and Place.Z3" and enter the IROBOTCAM environment. (Figure 10-2)

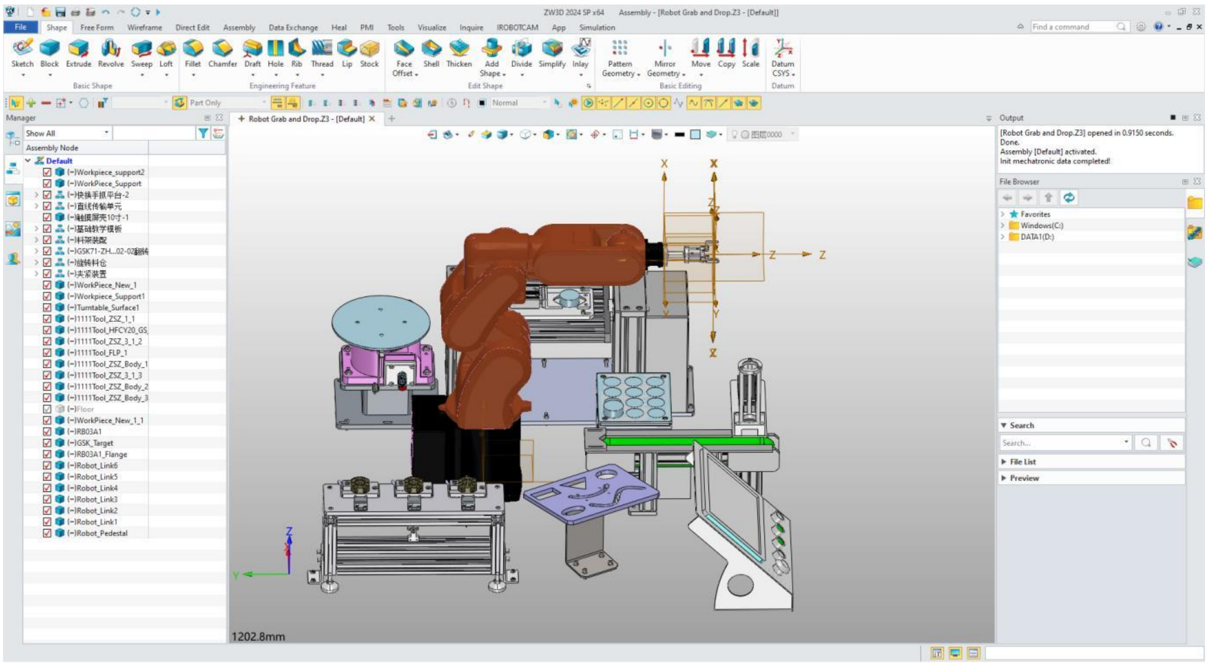


**Figure 10-2：Interface details**

## Step 2：Add Prismatic Joints

- With the robot GSK created and its rigid and collision bodies defined, the next step is to add prismatic joints for the robot tool. Here's how:

✧ Go to the "IROBOTCAM" tab and click the "Mechatronic" button.

✧ Right-click the “Kinematic Pairs and Constraints” button and select “Add Prismatic Joint”.

✧ Define child nodes, select the following rigid bodies:(Figure 10-3)

Tool_2

Tool_21

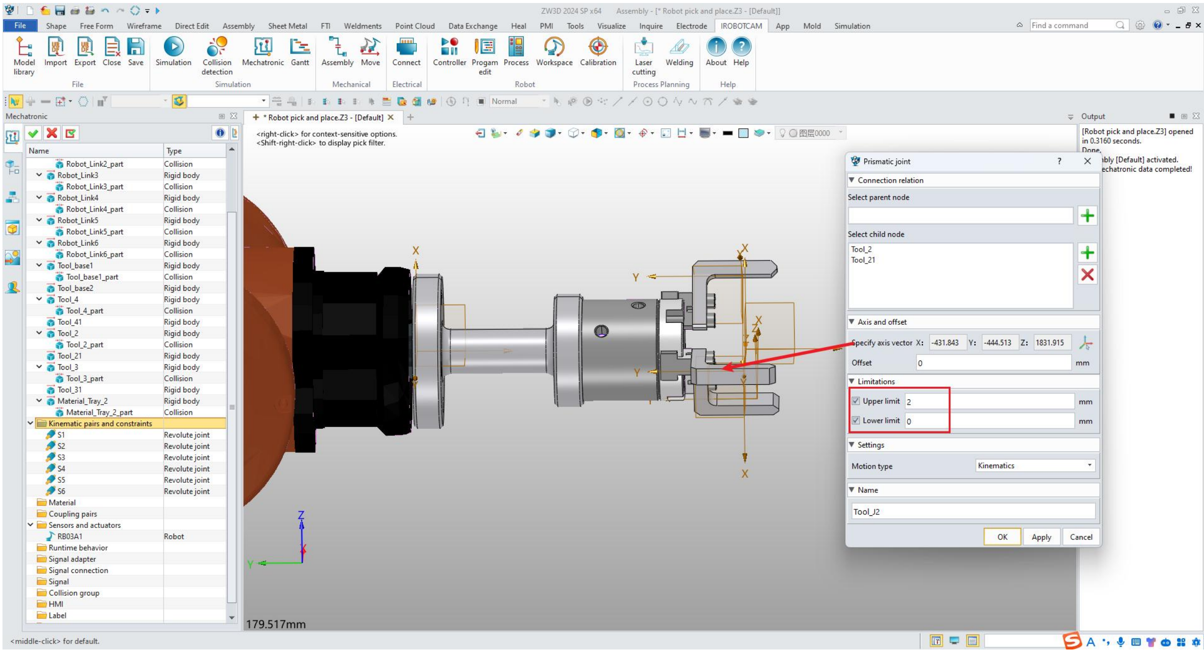



**Figure 10-3：W1 Prismatic Joint**

- Specify axis vector, click on the "Datum CSYS16" built on the selected rigid bodies to define the axis of movement for the joint.
- Set restrictions, set the "upper limit" to 2 and enter "0" in the "lower limit" field.These values represent the maximum distance the joint can move in the positive and negative directions.
- Choose motion type, select “Kinematics” for this joint.
- Name the joint: Enter a descriptive name for the joint, such as “W1”.
- Repeat the above operation steps for the remaining two pairs of claws.

✧ “Tool_J3”: Child nodes: “Too_3” and “Tool_31”, Axis vector: “Datum CSYS18”.

✧ “Tool_J4”: Child nodes: “Tool_4” and “Tool_41”, Axis vector: “Datum CSYS13”.

- Verify highlighting: After adding the three joints, click on each joint to verify that the corresponding part of the claw is highlighted correctly. (Figure 10-4)

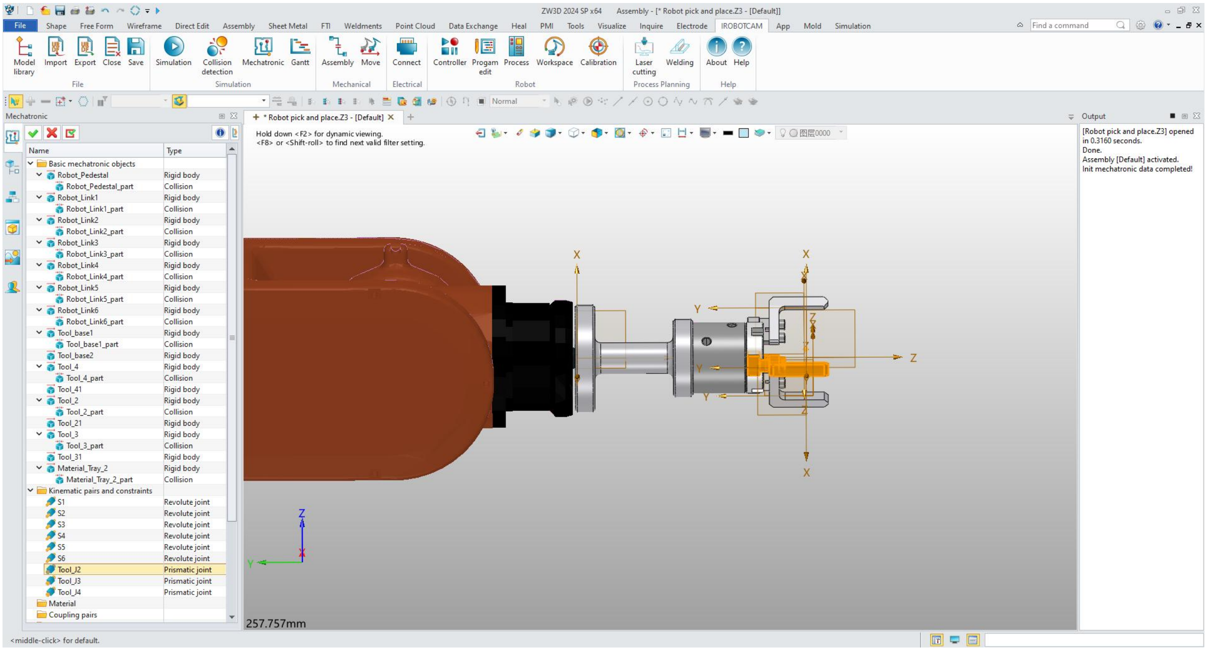


**Figure 10-4：Three Prismatic Joints**

## <u>Step 3：Add sensors</u>

- Right-click "Sensors and actuators" button.
- Select “Add Sensors”.
- Define sensor dimensions, set the “Length”, “Width”, and “Height” to 30 for each dimension.
- Specify coordinate system, Click "CSYS8" to refer to the coordinate system of the sensor.
- Name the sensor, enter a descriptive name for the sensor.
- Set the window to default values.
- Click “OK” to add the sensor to the scene. (Figure 10-5)

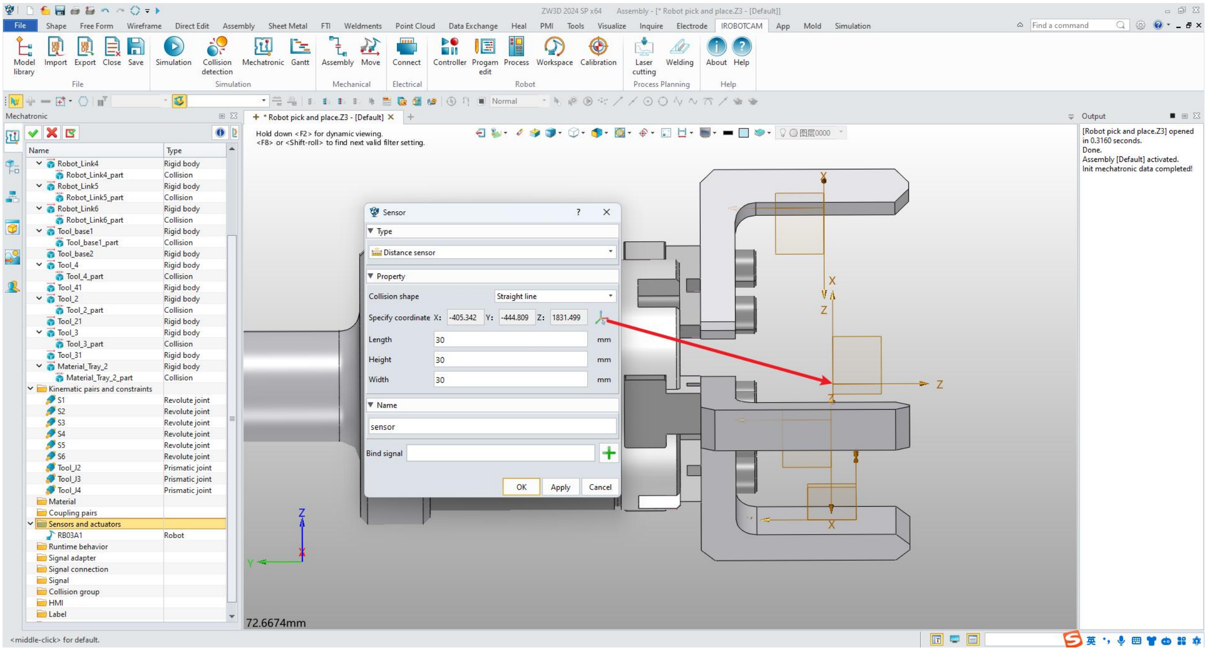


**Figure 10-5: Impact Sensor**

## Step 4: Add signals

- Right-click "Signal" button.
- select "Add Signal".
- Define signal type, in the “IO type” field, select “Out” to define the signal as an output signal.
- Name the signal, such as “Tool_out”.
- Set Port, enter -1 in the “port” field.
- Click “OK”. (Figure 10-6)

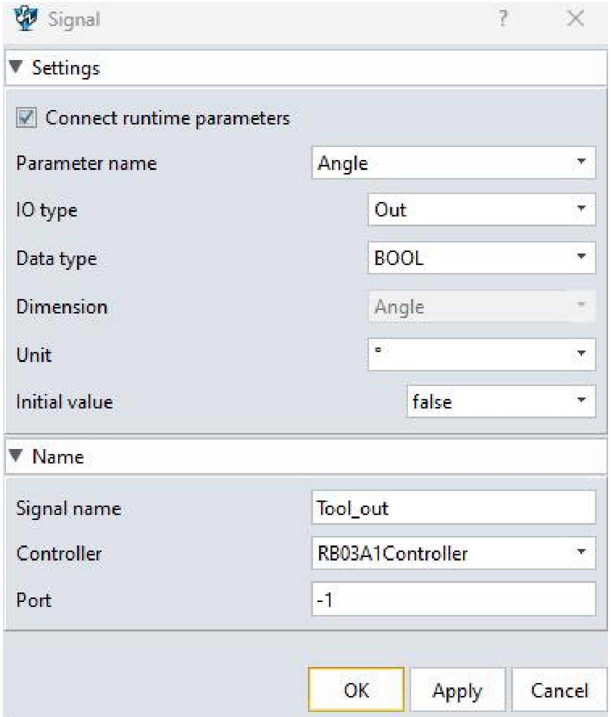



**Figure 10-6：Add out signal**

## Step 5：Add robot tool

- Right-click "Sensors and actuators" button.
- Select “Add Robot Tool”.
- Select motion pairs, Click on the + button and select the “three prismatic pairs” you created previously.
- Set speed, In the “Speed” field, enter “100” to define the speed of the tool movement.
- Select Rigid Bodies,select the rigid body representing the base part of the robot tool.
- Assign Sensor: In the “Sensing Point” field, select the sensor you created previously.
- Define Base, In the “Base coordinate” field, select the “CSYS9” coordinate system.
- Select the tool center and select the "CSYS8" coordinate system.
- Assign Signal, click the + button next to “Signal” and select the “Tool_out” signal you created previously.
- Name the tool: Enter a descriptive name for the robot tool, such as “Tool”.
- Click the “OK” button.(Figure 10-7)

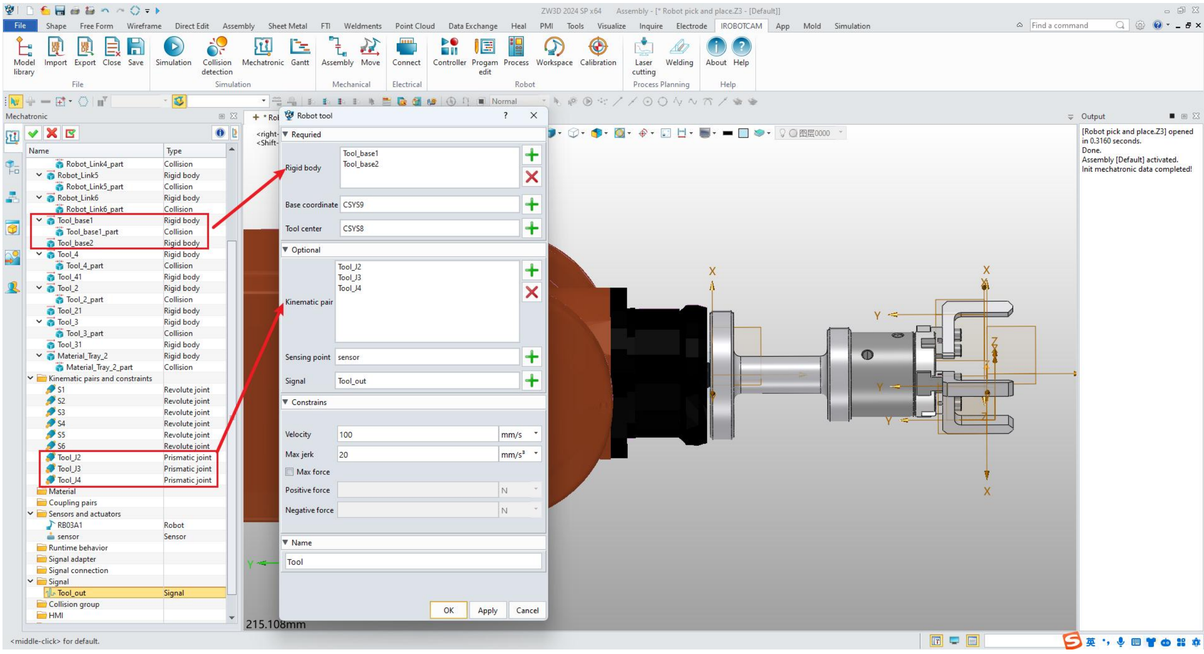



Figure 10-7：Robot Tool

## Step 6：Establish assembly relationships

- Click the “Assembly” button in the "IROBOTCAM" tab.
- Click the "Add" button.
- Select child device, select "Tool" robot tool you created in the previous step.
- Install on robot, in the “Parent Device” list, select the “RB03A1” robot.
- Click “OK” button the confirm. (Figure 10-8)

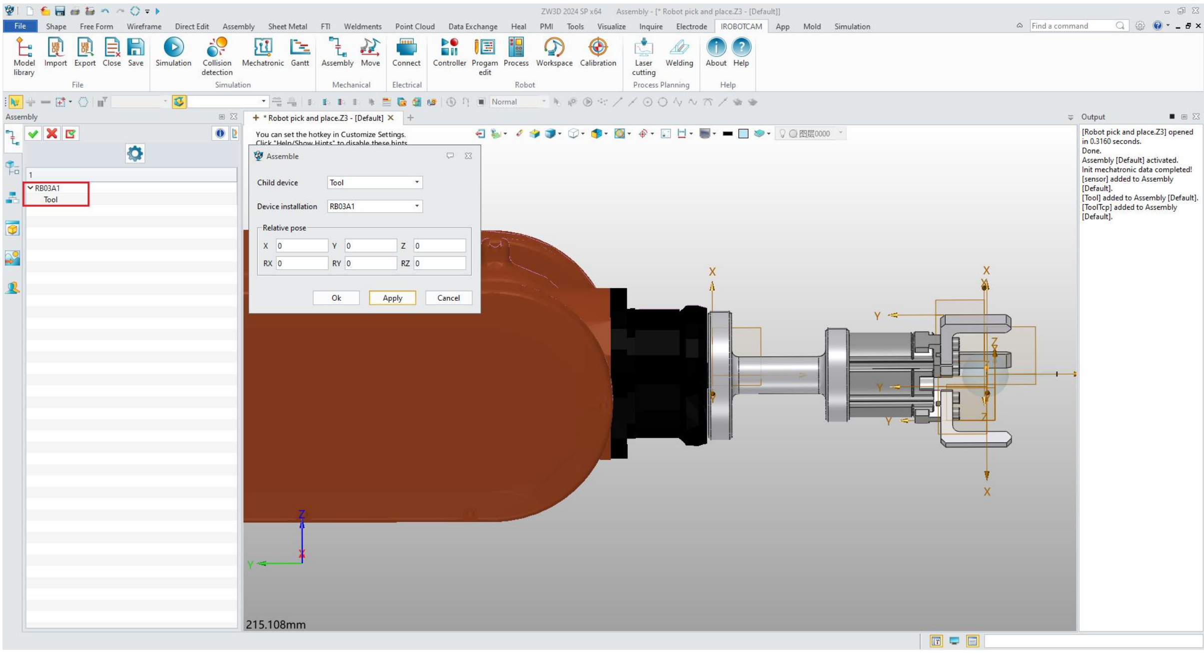


**Figure 10-8：Assembly relationships**

## Step 7：Add signal adapters (2)

- Right-click "Signal Adapter" button.
- Select "Add Signal Adapter". (Figure 10-9)

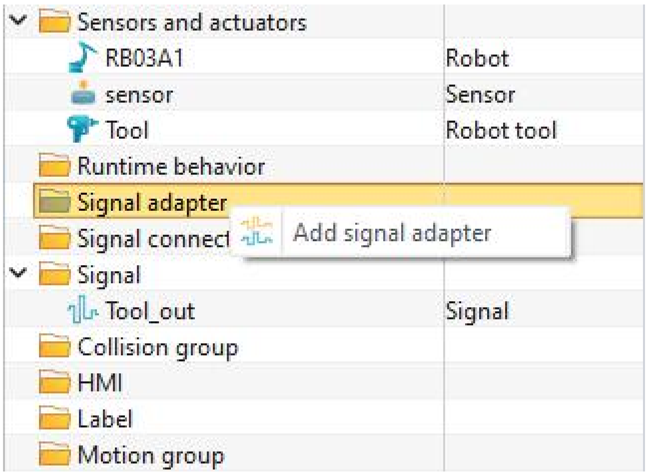



**Figure 10-9：Add signal adapters**

- Select Robot, click the + button and select the “RB03A1” robot.
- Name and Set Data Type, enter a descriptive name for the signal adapter, such as “RB03A1_S”. In the “Data Type” field, select “Bool” to indicate that the signal transmits Boolean values.
- In the formula bar, fill in “RB03A1_S” and name the signal “RB03A1_SD”.(Figure 10-10)

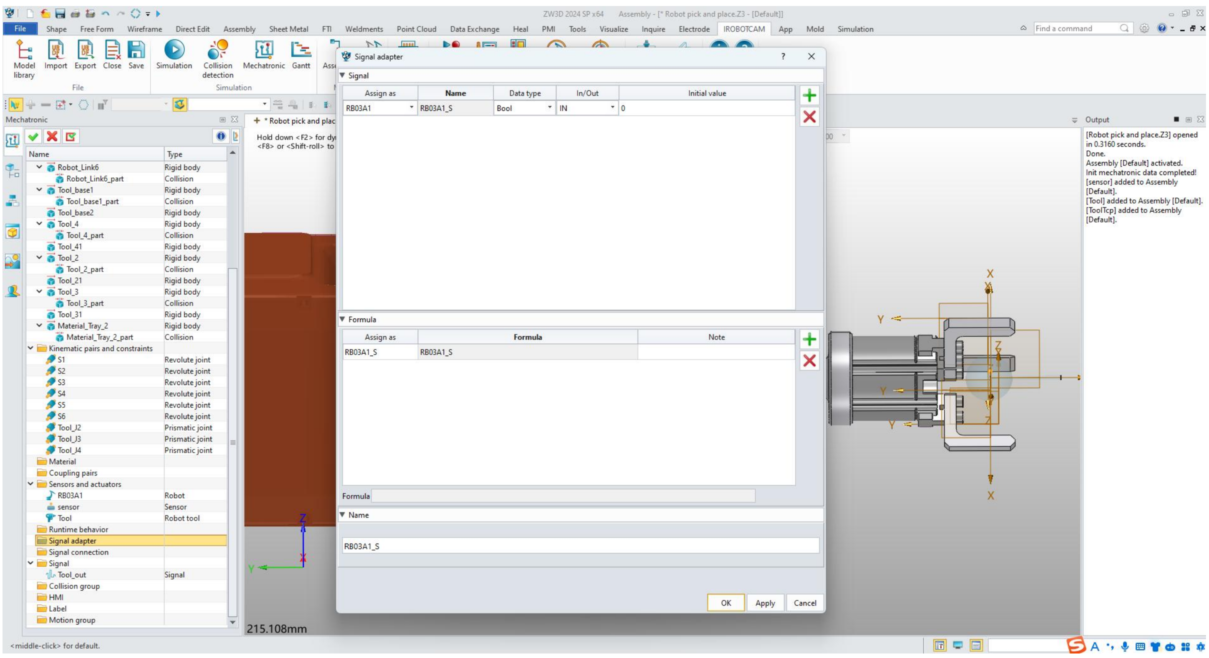


**Figure 10-10：Signal Adapter RB03A1**

- Repeat the above operations for the robot tools, assigning appropriate names and data types (such as "Tool_S" and "Float"). (Figure 10-11)

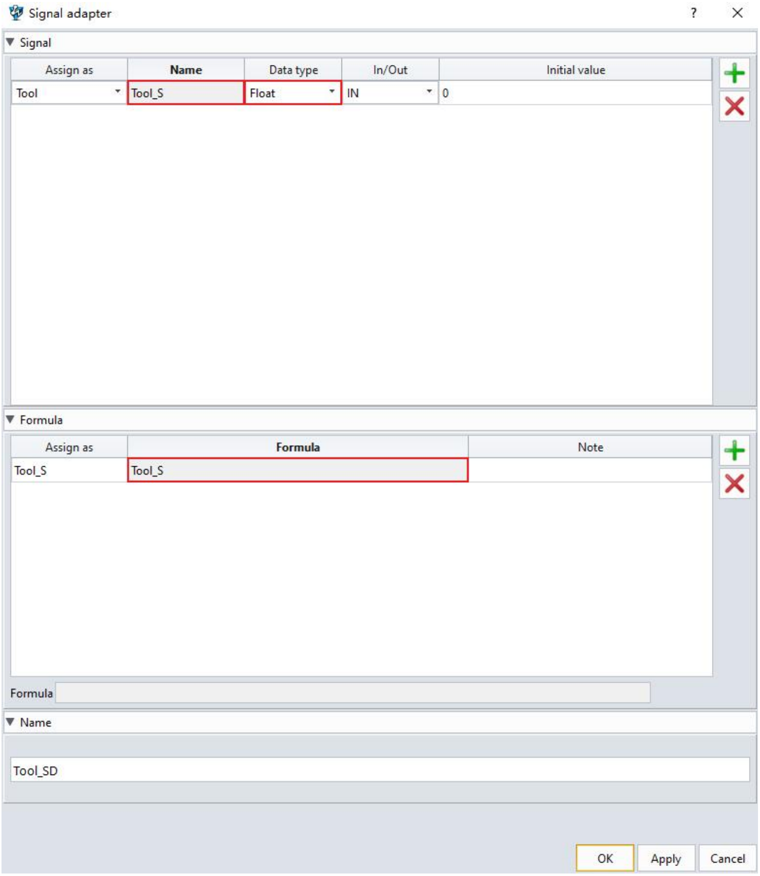



**Figure 10-11：Signal adapter tool**

## Step 8：Add signals

- Right-click on "Signal".
- Select “Add signal” button.(Figure 10-12)


**Figure 10-12：Add Signal**

- Set input type, in the “IO Type” field, select “In” to define the signal as an input signal.

- Name the Signal, such as “Tool_opt”.
- Set Port, In the “Port” field, enter 0 to specify the port on which the signal will be received.
- Click “OK” button to confirm. (Figure 10-13)

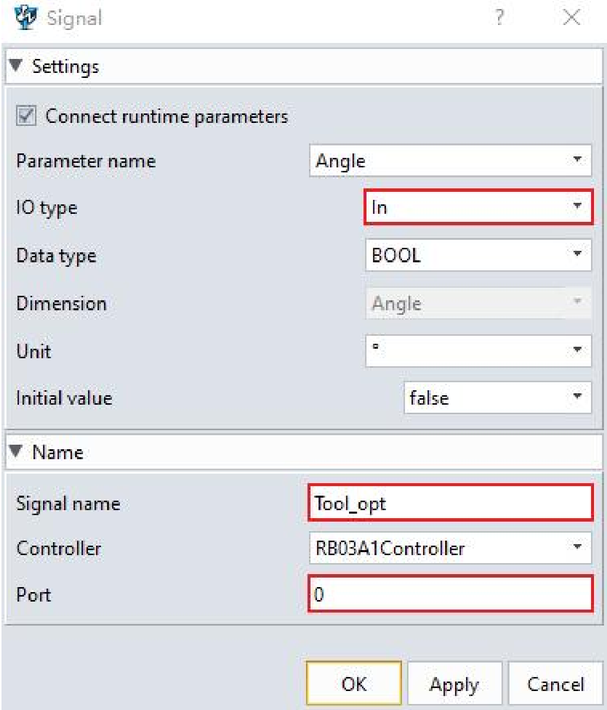



**Figure 10-13：Signal**

## Step 9：Add simulation sequences (3)

- Right-click "Runtime behavior" button.
- Select “Add simulation sequence”. (Figure 10-14)

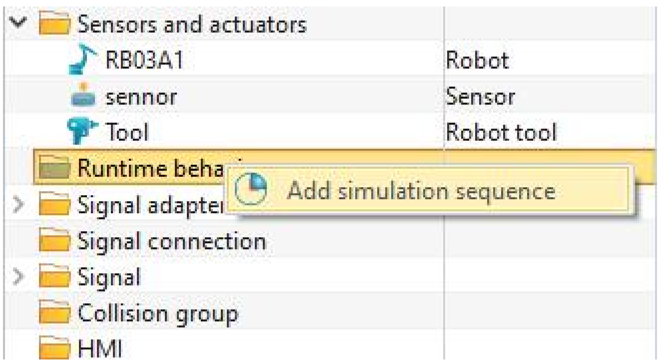



**Figure 10-14：Add simulation sequences**

- Define RB03A1_EXEC Sequence:(Figure 10-15)

✧ Start Time: 1

✧ Duration: 999999 (essentially constant execution)

✧ Activation/End Signals: Unchecked (no specific activation or end triggers)

✧ Execution Signal: RB03A1_S (controls the robot's overall movement)

✧ End Position: 1 (presumably defines the final position of the robot)

Name: RB03A1_EXEC.

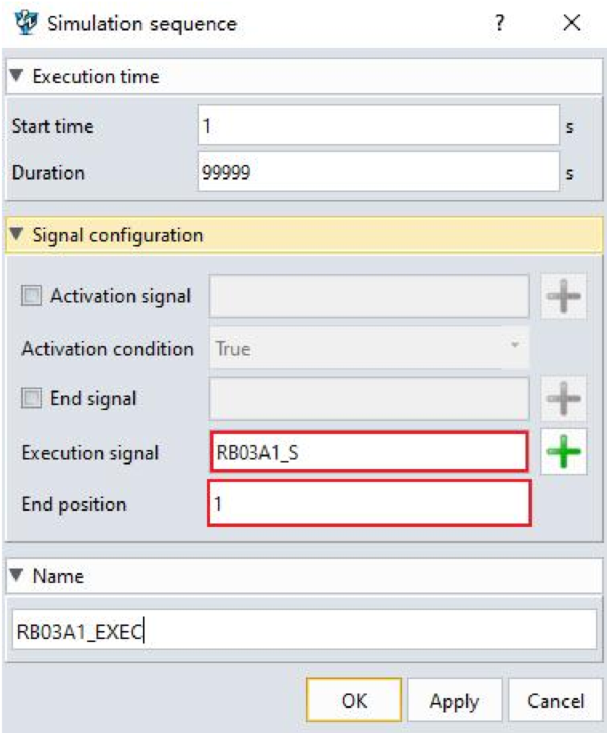



**Figure 10-15：GSKRobot_EXEC**

- Define Tool_catch_EXEC Sequence:(Figure 3-12)

✧ Start Time: 0

✧ Duration: 5

✧ Activation Signal: Checked (requires activation)

✧ Activation Condition: Tool_opt = True

✧ Execution Signal: Tool_S (controls the robot tool's movement)

✧ End Position: 1 (presumably defines the final position of the tool during grasping)

✧ Name: Tool_catch_EXEC

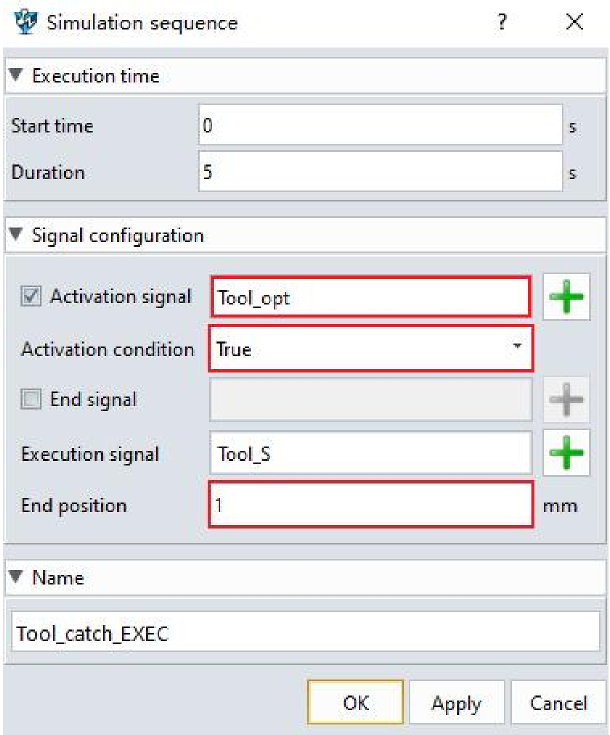



**Figure 10-1：gongju_catch_EXEC**

- Define Tool_pull_EXEC Sequence:(Figure 3-12)

✧ Start Time: 0

✧ Duration: 5

✧ Activation Signal: Checked (requires activation)

✧ Activation Condition: Tool_opt = False

✧ Execution Signal: Tool_S (controls the robot tool's movement)

✧ End Position: 0 (presumably defines the final position of the tool after releasing)

✧ Name: Tool_pull_EXEC

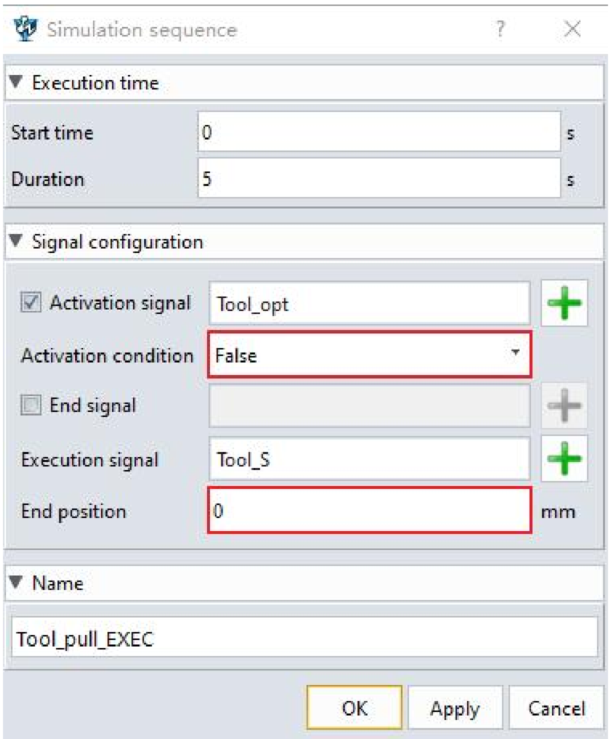



**Figure 10-1：gongju_pull_EXEC**

- Three simulation sequences define the overall behavior of the robot and robotic tool. They specify the timing, activation conditions and execution signals of different operations:

✧ RB03A1_EXEC: This sequence controls the main movement of the robot throughout the simulation.

✧ Tool_catch_EXEC: When signal "Tool_opt" becomes True, this sequence triggers the robot tool to move towards the object and catch it.

✧ Tool_pull_EXEC: This sequence triggers the robot tool to move away from the object and drop it when the signal "Tool_opt" becomes False

## Step 10：Add rigid body and collison

- Add a rigid body to the material.(Figure 3-12)

✧ Click on “Basic Mechatronic Objects” in the IROBOTCAM interface.

✧ Right-click and select “Add Rigid Body”.

✧ Select the desired rigid body (Figure 3-12).

✧ Choose the material as “default”.

✧ Leave the property settings unchecked.

✧ Click “OK” to confirm and add the rigid body.

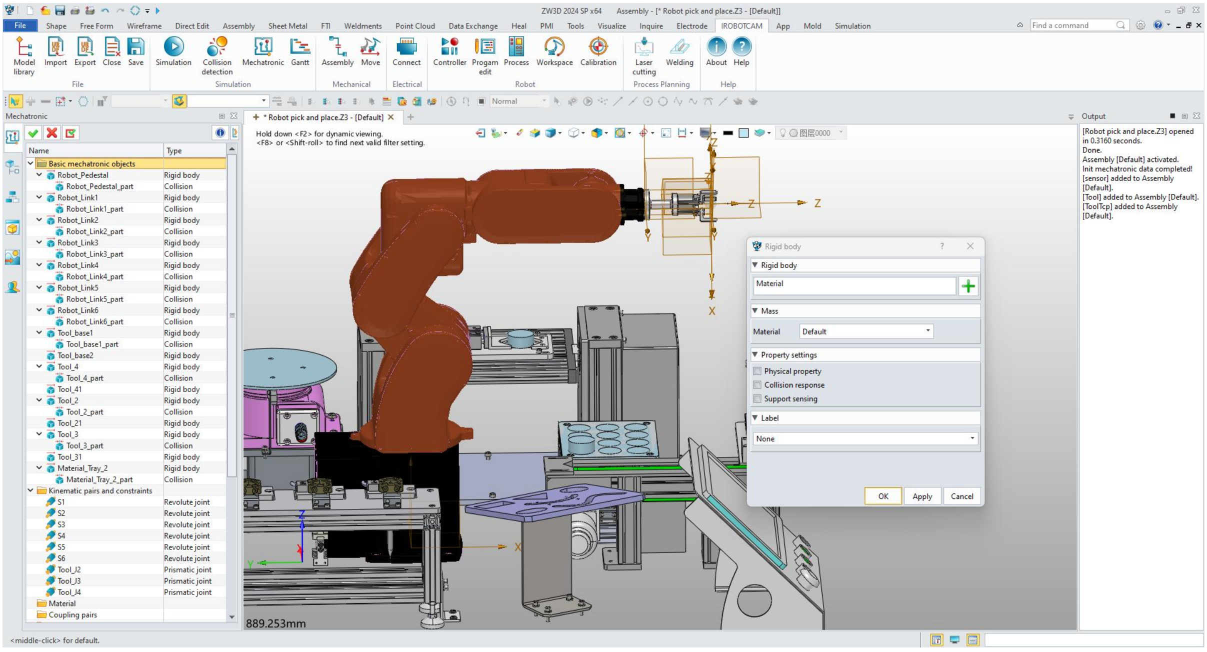



**Figure 10-1：Add rigid body**

- Add Collision to “Material”(Figure 3-12)

✧ Right-click on the “Basic Mechatronic Object” again.

✧ Select the rigid body you just added.

✧ Choose “Convex decomposition” as the collision shape.

✧ Check all three property settings:

  ➢ Physical properties

  ➢ Collision response

  ➢ Support induction

✧ Set the material to default.

✧ Click "OK" to confirm and add the collision object.

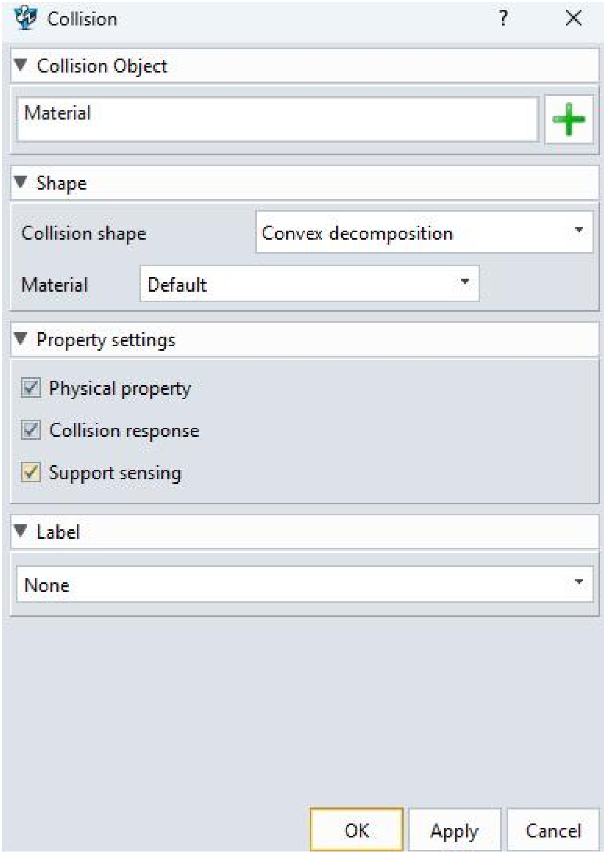



**Figure 10-1：Add Collison**

- Add Rigid Body to “Material_tray”:(Figure 3-12)

✧ Repeat steps similar to adding the rigid body to “Material_tray”.

✧ Select the rigid body representing the support table.

✧ Properties are set to default.

✧ Click “OK” to confirm and add the rigid body.

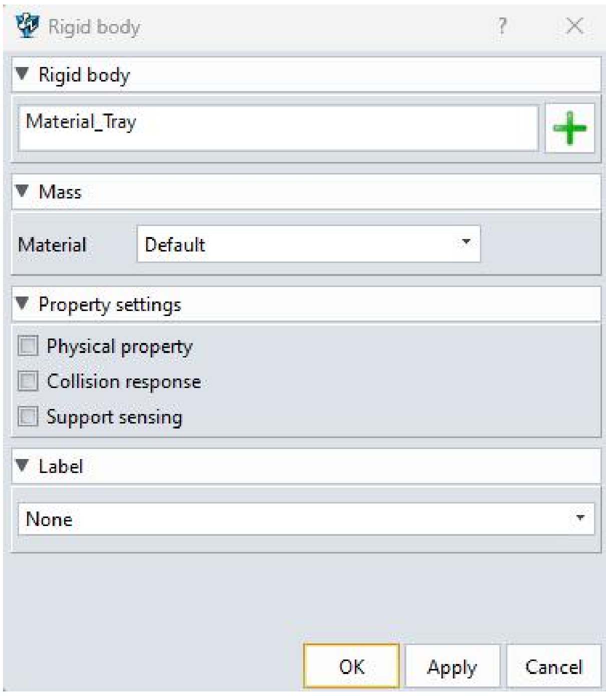



**Figure 10-1：Add Rigid Body**

- Add Collision to “Workpiece Support1”:(Figure 3-12)

✧ Repeat steps similar to adding the collision object to “WorkPieceNew_1”.

✧ Select the rigid body representing the support table.

✧ Choose “Convex decomposition” as the collision shape.

✧ Properties are set to default.

✧ Set the material to “default”.

✧ Click “OK” to confirm and add the collision object.

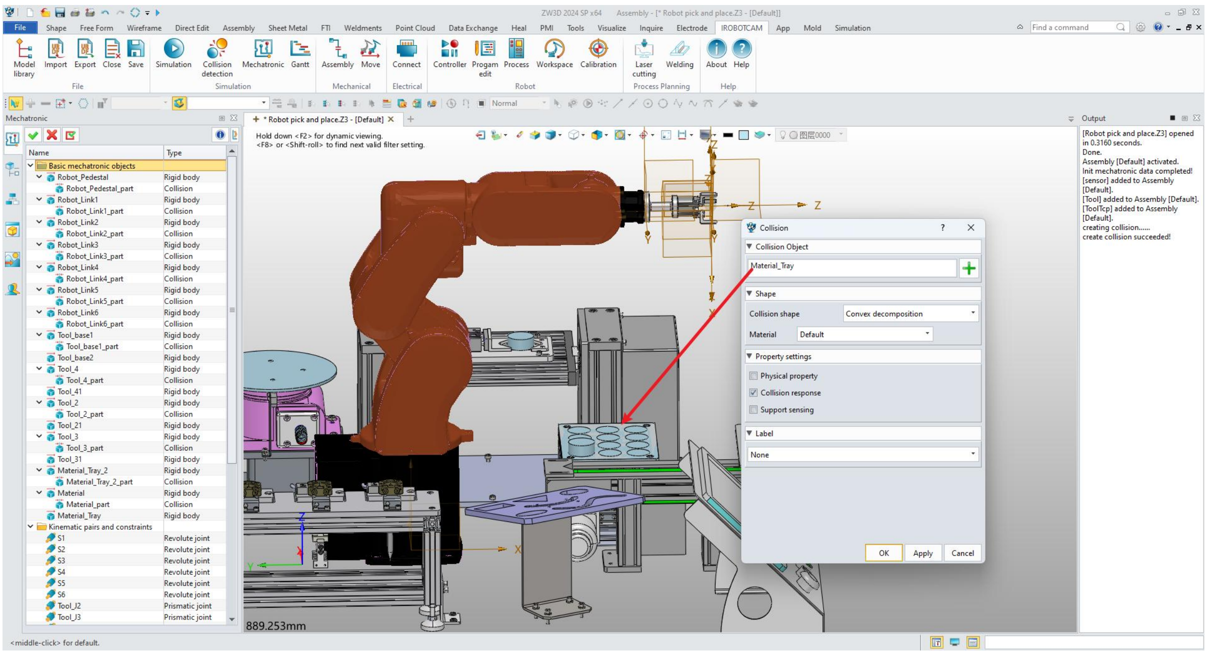


**Figure 10-1：Add Collision**

## Step 11：Prepare coordinate positions

- Click on the “DatumCSYS” option in the “Shape” column.
- This will allow you to create a coordinate system directly on the material itself.
- Ensure the “Z-axis” direction points downwards.
- Select the “Dynamic” option to enable real-time updates of the coordinate system during simulation.
- Right-click in an empty space and choose “Center of Curvature”.

(Figure 3-12)

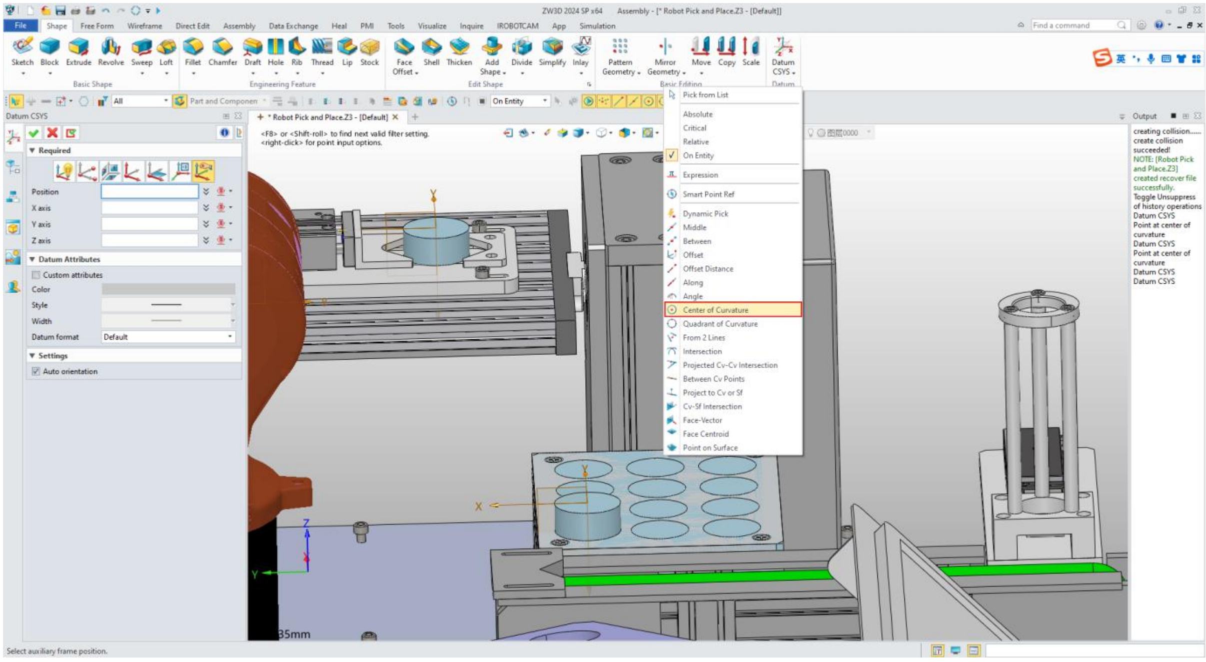


**Figure 10-1：Coordinate position**

## Step 12：Add marker points

- Hide Unnecessary Coordinate Systems:

✧ Select the CSYS8 and CSYS18 coordinate systems on the robot tool.

✧ Hold down the Shift key and selected all other displayed coordinate systems.

✧ Uncheck all selected coordinate systems, effectively hiding them from the scene.

(Figure 3-12)

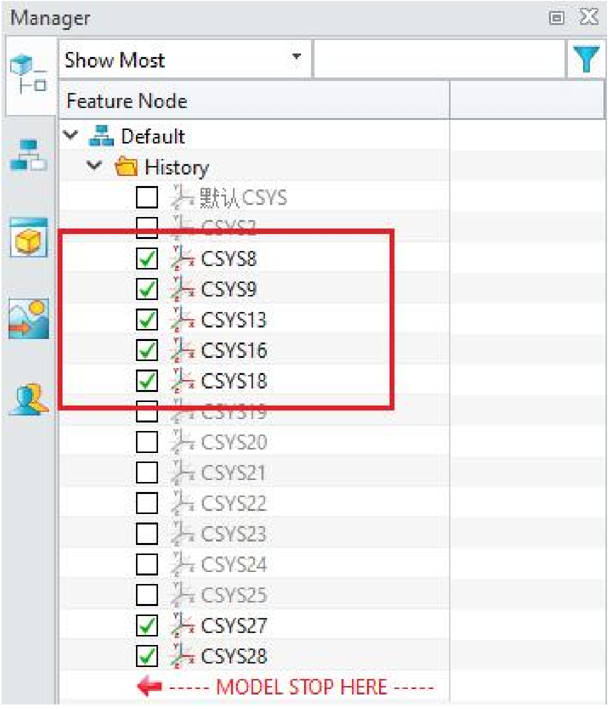



**Figure 10-1：Hide the Coordinate Systems**

- Teach Robot Movement:
- ✧ Navigate to the "Sensors and actuators" section under the "RB03A1" tab.
- ✧ Right-click and selected "Teach Robot" to begin adding marker points. (Figure 3-12)

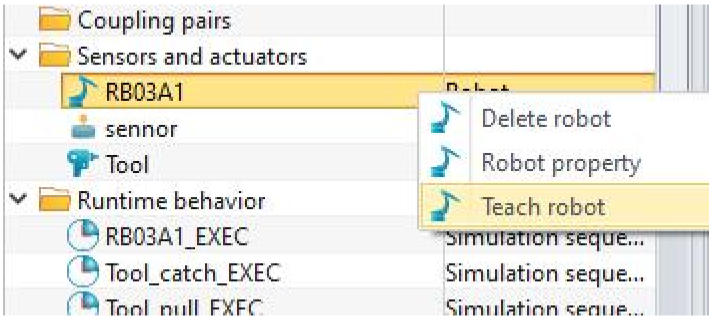



Figure 10-1：

- Adding Marker Points:
- ✧ Set the fifth joint of the robot arm to 90 degrees.
- ✧ Click "Record Joint Point" to mark this position as the first point. (Figure 3-12)

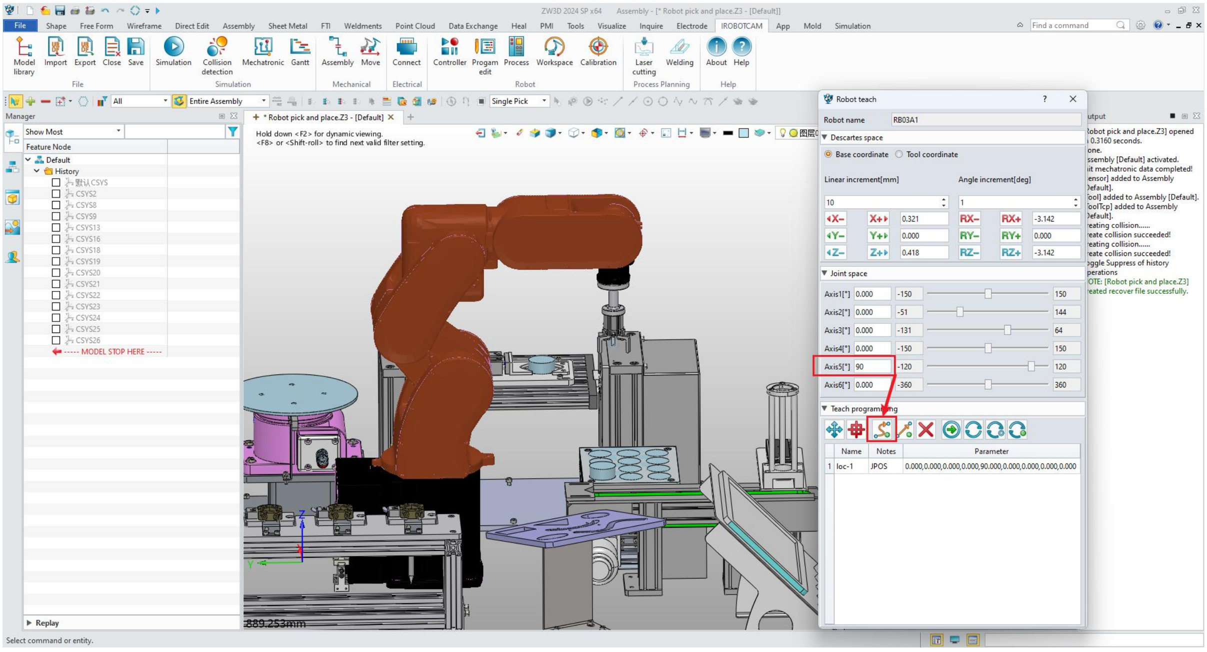



Figure 10-1：The First Point

- Click the "Mark Point" button and selected the coordinate system established on the rigid body.
- Utilize "Drag Teach" and adjusted the robot's gripper position using the XYZ axes.
- Click "Record End Point" to mark this as the second point.

  (Figure 3-12)

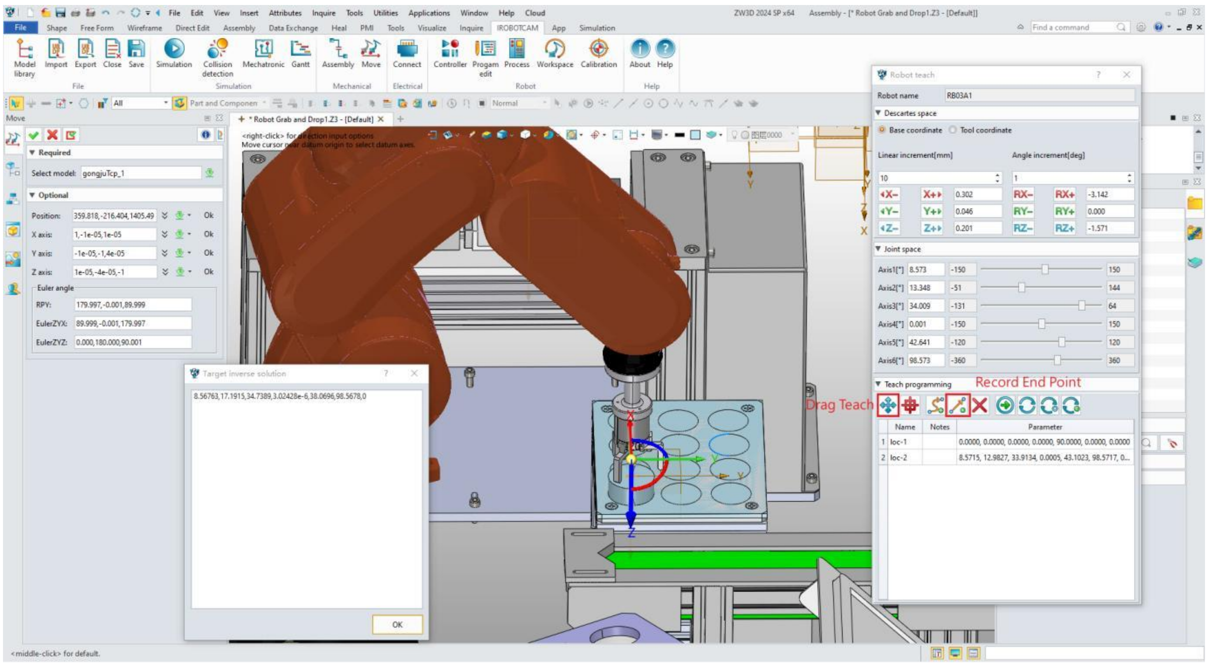



Figure 10-1：The Second Point

- Repeat the "Drag Teach" and "Record End Point" process to mark additional points for the robot's movements, including upwards positions and a point where the gripper separates from another rigid body.
- Finally, click "Move to Home" to return the robot to its initial position.

(Figure 3-12)

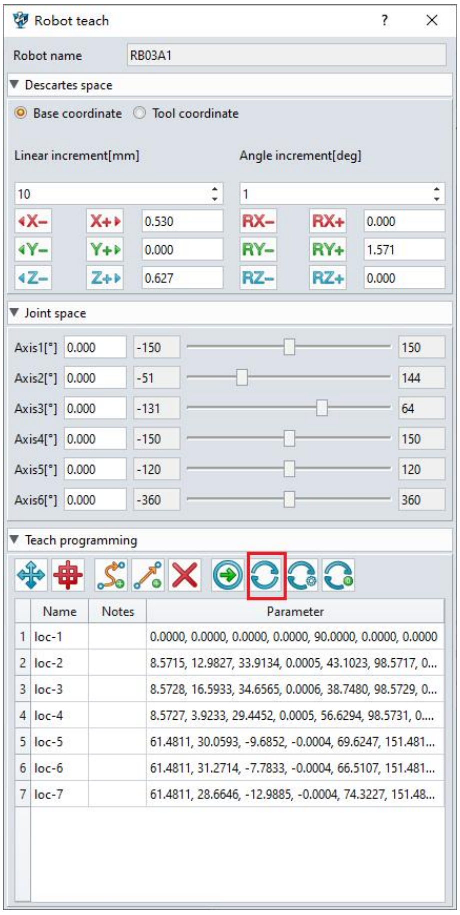



**Figure 10-1：Mark Completed**

## Step 13：Add commands

- Navigate to "IROBOTCAM" tab, click the "Program Edit" button. (Figure 3-12)

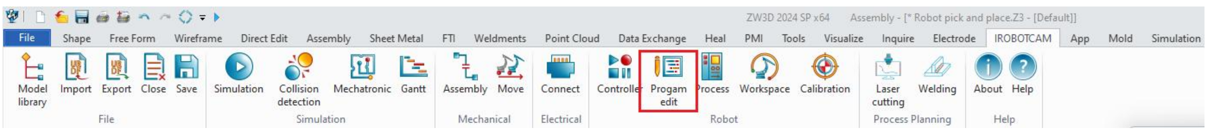



**Figure 10-1：Program Edit**

- Add JOB and Program:

✧ In the left window, right-click "RB03A1Controller" and select "Add JOB".

✧ Click on the newly added "JOB1" and right-click to choose "Add Program". (Figure 3-12)

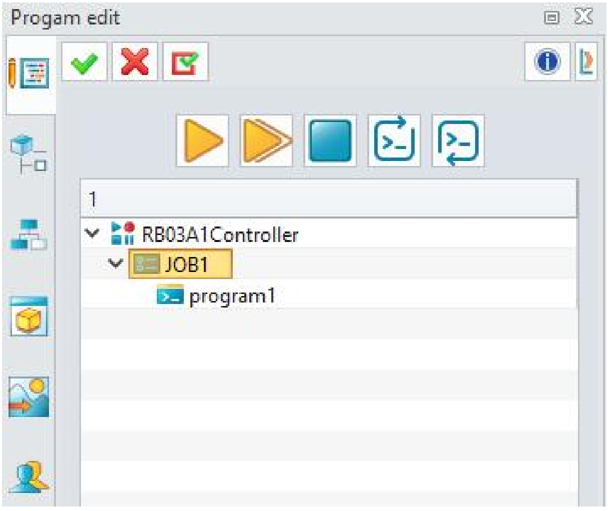



**Figure 10-1：Edit Program**

- Edit program parameters:

✧ Right-click on "program1" to edit it, and a window for program parameters appeared

✧ Add all command positions and names used in the program.
(Figure 3-12)

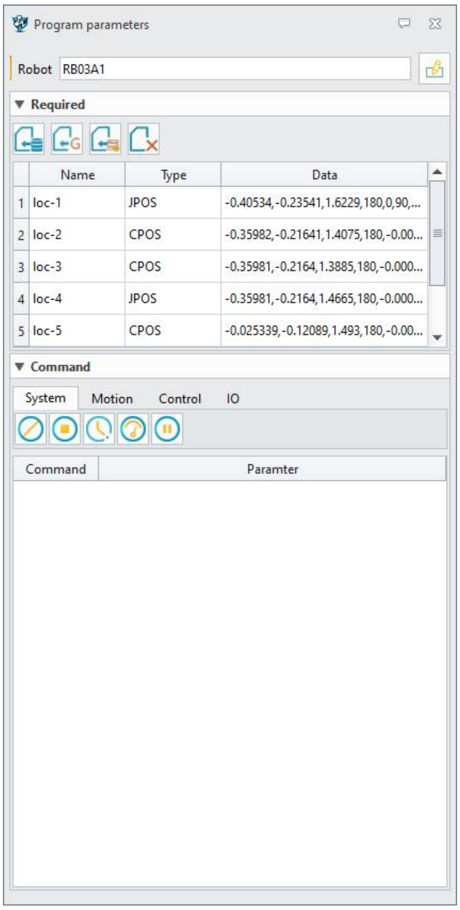



**Figure 10-1：Program Parameters**

- Define Wait time and DOUT values:
- ✧ Set the value of WAIT (ms) to "1000".
- ✧ Define the false value of DOUT as "0" and the true value as "1". (Figure 3-12)

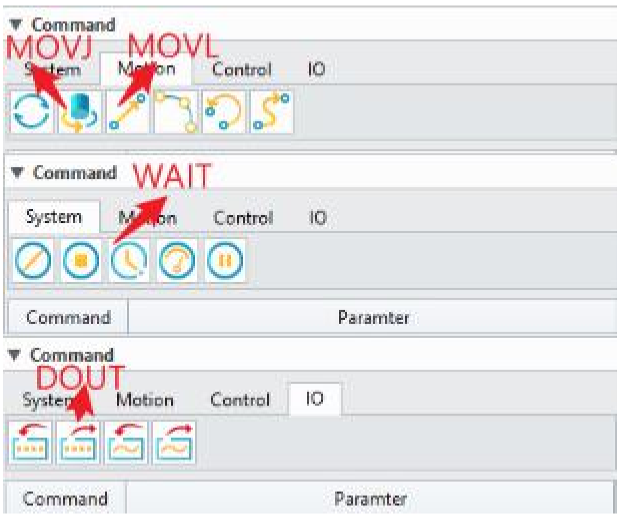



**Figure 10-1：Command instruction**

- Adjust MOVL parameters:

✧ Modify the parameter values under the "MOVL" command.

(Figure 3-12)

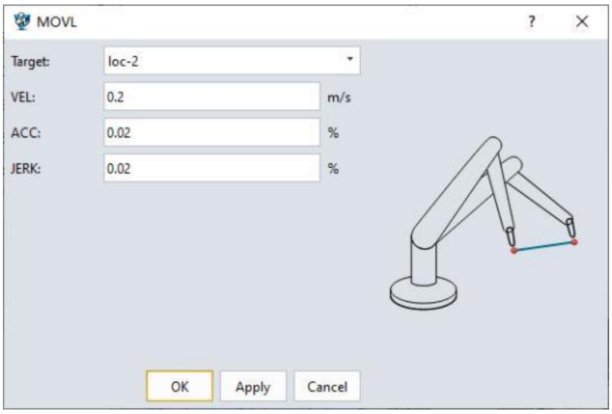



**Figure 10-1：MOVL**

- Add commands and modify MOVJ parameters:

✧ Added commands according to the provided style.

✧ Specifically, change the MOVJ command parameter in the seventh line to "100, 50, 50".

(Figure 3-12)

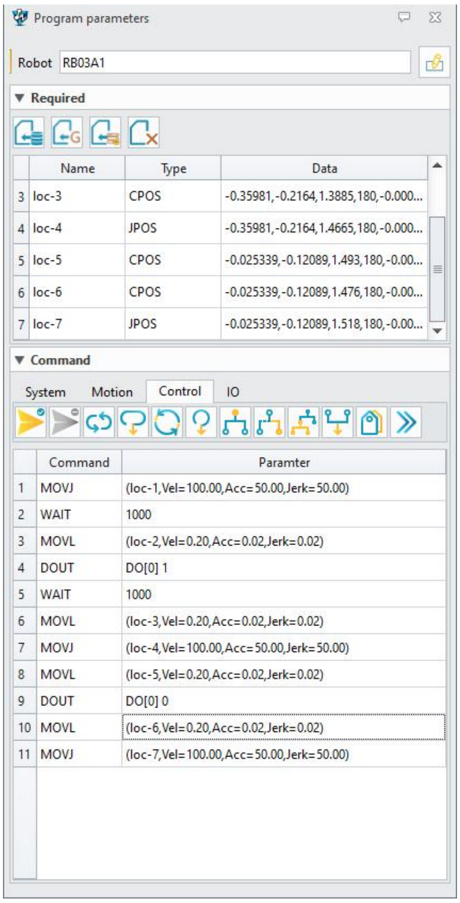


**Figure 10-1：Command**

- These steps ensure that the robot follows specific commands and performs the desired pick-and-place operation based on the marked points and defined parameters.

## Step 14：Simulation Verification

- Make the interface clean

✧ Hide all coordinate systems by unchecking them to improve the interface clarity.

✧ Additionally, you hid the component "ReferenceMateial" to better visualize the robot's gripper and its interaction with the object.

(Figure 3-12)

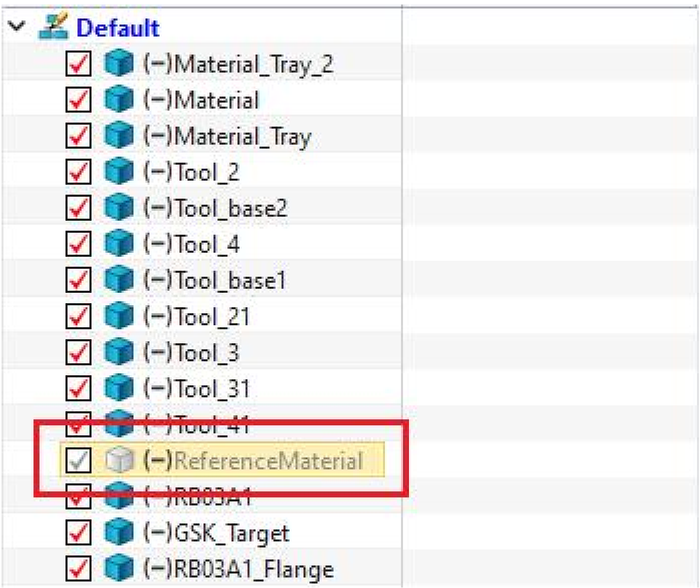



**Figure 10-1：Hide**

- Start simulation:

✧ Click the "Simulation" button in the "IROBOTCAM" tab.

(Figure 3-12)

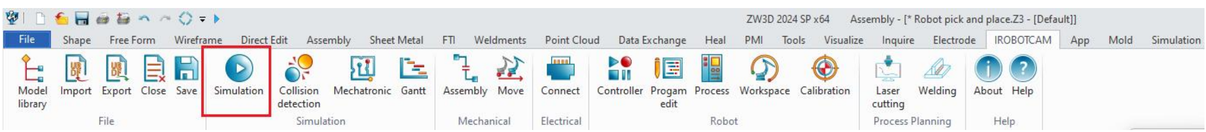



**Figure 10-1：Simulation Button**

- Record robot motion

✧ Optionally checked the "Record video" checkbox to capture the robot's movement during simulation.

● Run simulation and stop recording

✧ Click the "Start" button in the "Simulation Control" to run the programmed robot actions.

● After observing the entire process and confirming successful capture (if video recording was enabled), you clicked the "Stop" button.

(Figure 3-12)

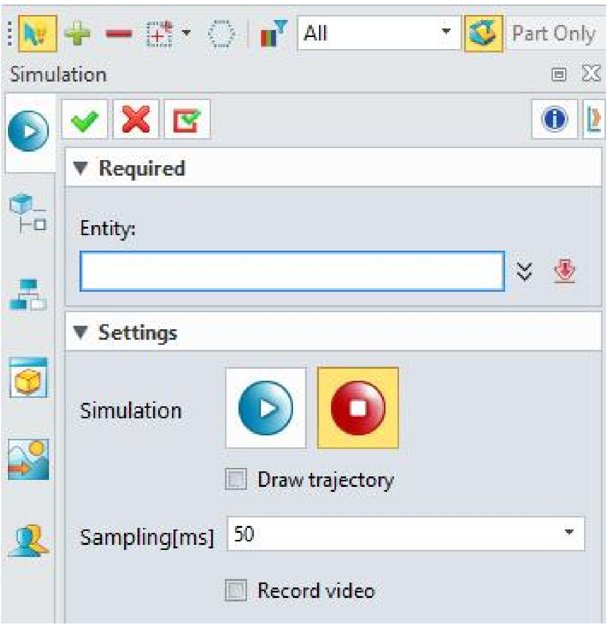



**Figure 10-1： Start**

● Verify successful pick and place:

✧ Confirmed that the robot successfully grasped and released the object based on the simulation results.

(Figure 3-12)

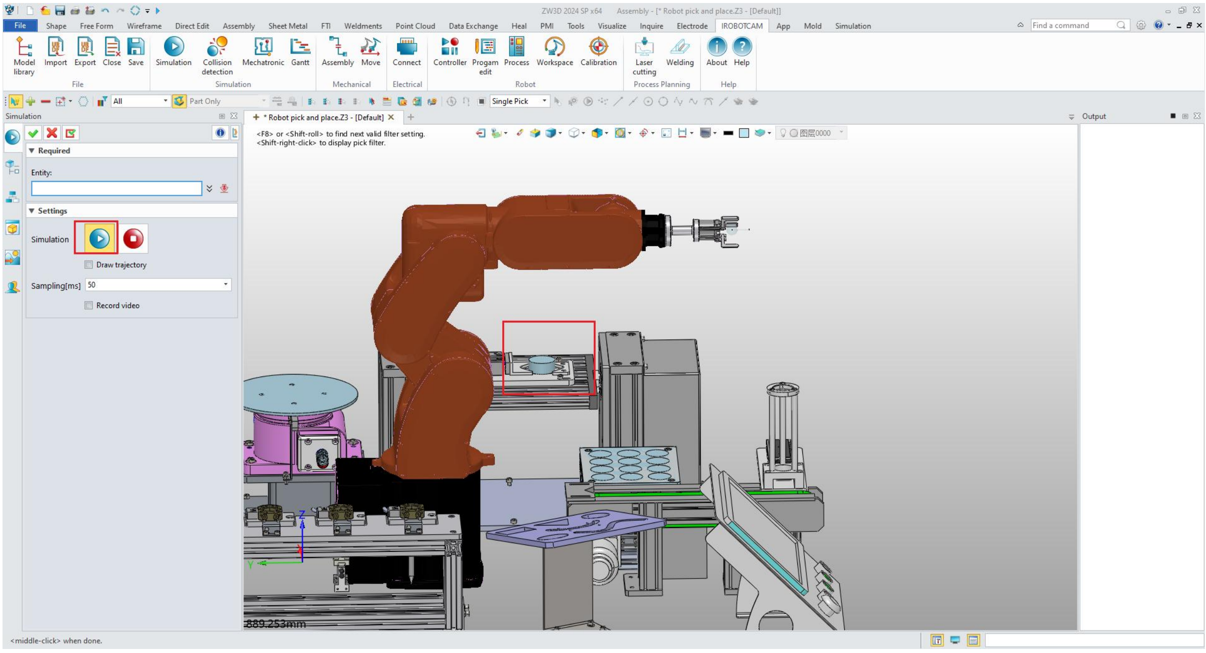


**Figure 10-1：Simulation verification**

This step ensures that the robot's movements and actions match the programmed commands and successfully execute the desired pick-and-place operation. Recording the simulation allows you to analyze the process and verify its accuracy and efficiency.

**Here are some additional thoughts on each error type:**

- **Speed, acceleration, and jerk parameters:**

1.Symptoms: The robot may move too quickly or slowly, or it may experience jerky movements.

2.Causes: The values for speed, acceleration, and jerk may be outside the acceptable range for the robot’s capabilities.

- **Solutions:**

1.Check the robot's specifications for the recommended values for these parameters.

2.Adjust the values in the program instructions until the robot moves smoothly and accurately.

3.Use the "Record joint point" and "Record end point" functions during teaching to ensure realistic speeds and accelerations.

- **Same point with two different commands:**

1.Symptoms: The robot may not move to the desired location, or it may move in an unexpected way.

2.Causes: The program may include two different instructions for the same point in the robot's path.

- **Solutions:**

1.Carefully review the program instructions to identify any duplicate commands.

2.Remove the duplicate command or choose the desired command for that point.

3.Use the "History management" bar to view and edit the teaching points and instructions.

4.Here are some additional tips for avoiding errors:

5.Test the program in stages: Start by testing the program with a small number of instructions and then gradually add more complexity. This will help you identify any errors more quickly.

6.Use the simulation visualization: Watch the robot's movements in the simulation window to identify any unexpected behavior.

7.Consult the IROBOTCAM documentation: The documentation provides detailed information on the software's features and functions, as well as troubleshooting tips.

By following these tips and carefully reviewing your program, you can avoid these common errors and ensure that your robot pick-and-place simulation runs smoothly and accurately.

# Chapter 11: Collision Detection

Step-by-Step Guide on Collision Detection Simulation in iRobotCAM.(Figure 11-1)

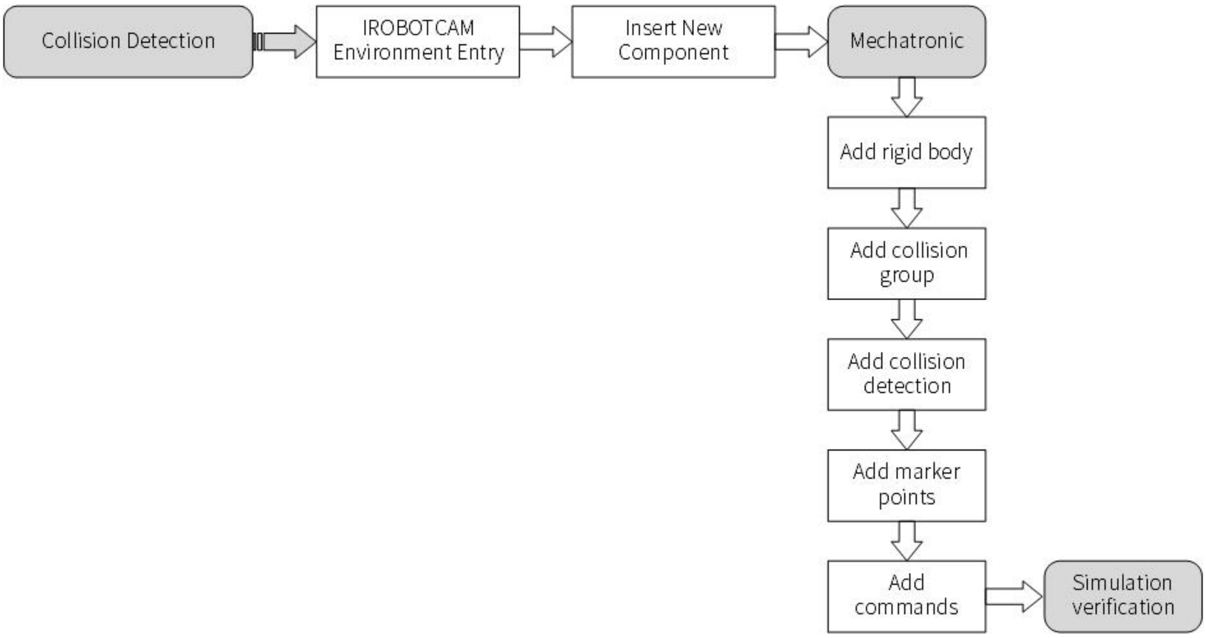



**Figure 11-1：Flow chart of collision detection**

## Step 1：Open iRobotCAM project file

- Open iRobotCAM project file “Collision Detection. Z3” to enter the IROBOTCAM environment. (Figure 11-2)

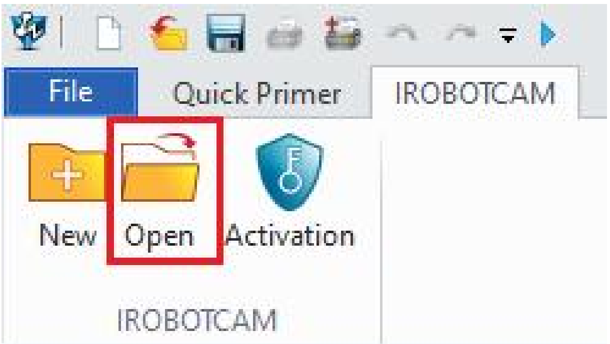



**Figure 11-2：Open iRobotCAM project file**

- The component details will show after enter it.(Figure 11-4)

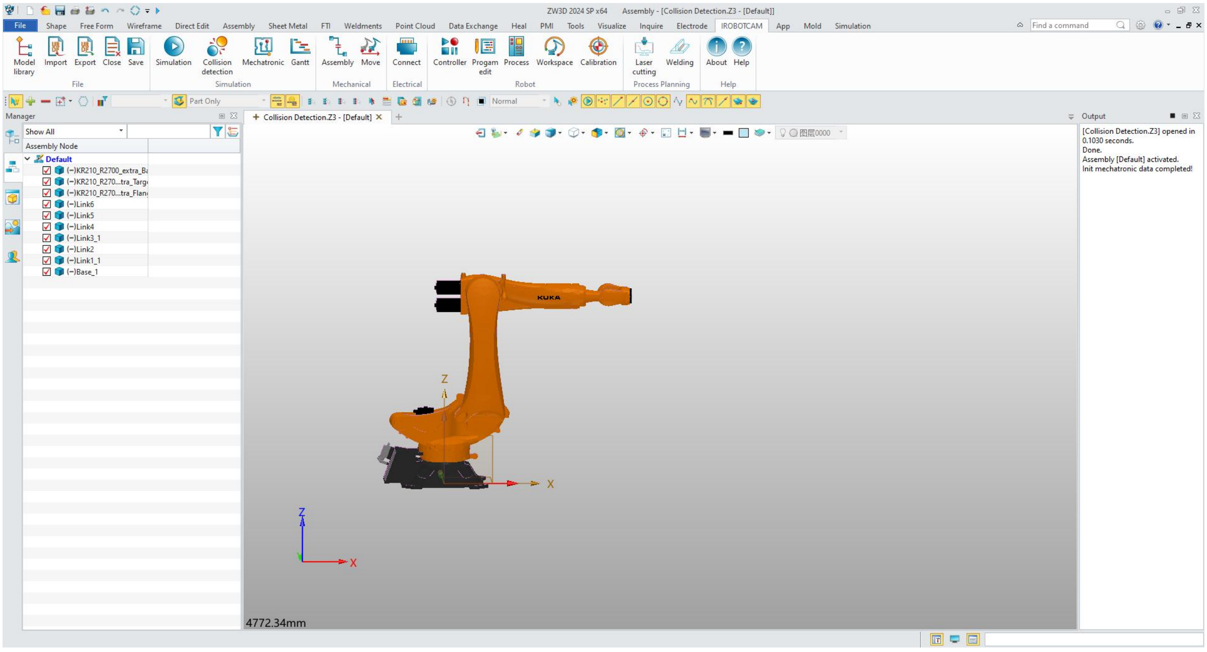


Figure 11-4：Component Details

## Step 2：Insert a new component

- Right click on "KR210_R2700_extra" to insert a new component.
- Name the part "COLLISION".
- Uncheck the “Automatic Activation” checkbox.
- Click the “Create” button to complete the creation of the new component. (Figures11-6)

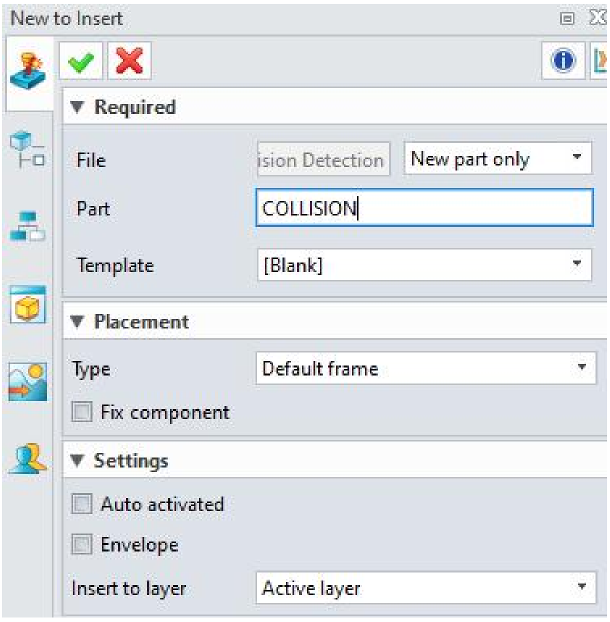



Figure 11-6：Insert New Component

- Right click to delete the coordinates under the constraint.Deleting the coordinates will also delete the constraints.(Figure 11-7)
- This creates the component named “COLLISION” that will be used for collision detection.

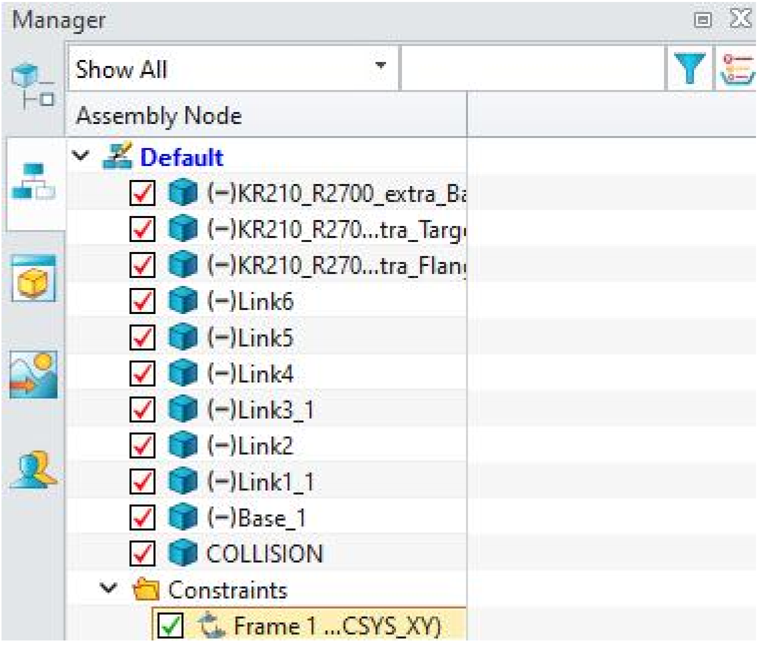



**Figure 11-7：Delete Constraint**

**Note:**

- The provided figures (11-2 to 11-7) are helpful for visualizing the described steps.
- Please refer to the software's documentation for confirmation.

**<u>Step 3：Insert and Adjust block</u>**

- Insert block.

✧ Find the “block” option in the “Shape” column and select it.

✧ Build the desired hexahedron shape within the interface, as shown in the provided figure.

✧ Click the “Checkmark” button to finish creating the hexahedron.(Figure 11-8)

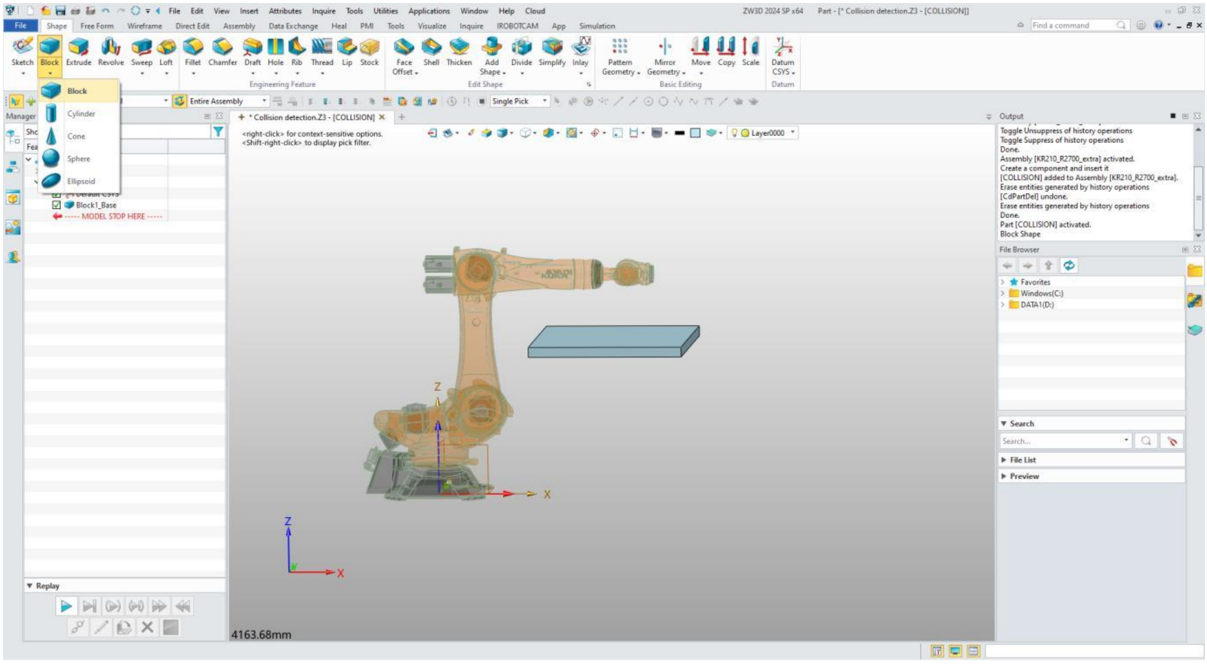


Figure 11-8：Insert Hexahedron

- Adjust Hexahedron Position:

✧ Double-click on the “KR210_R2700_extra” assembly to activate it as a whole. (Figure 11-9)

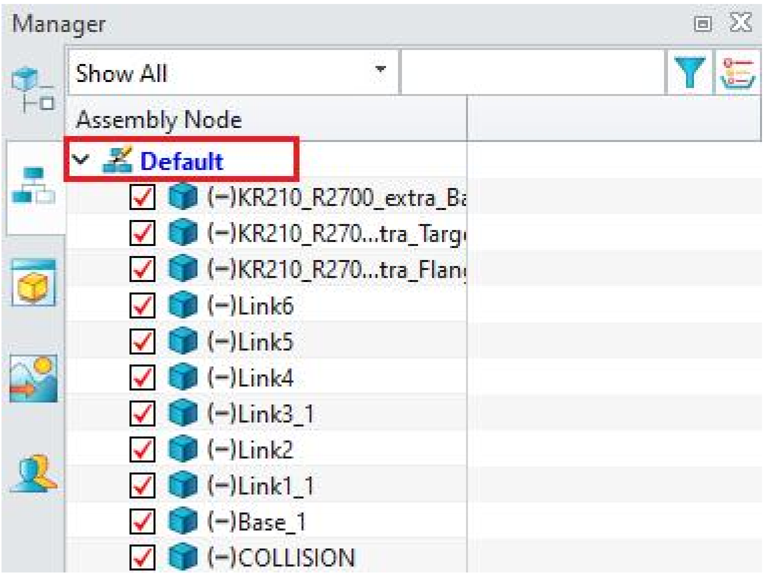



Figure 11-9：Activate Assembly

✧ Click on “Move” in the “Assembly” column.

✧ Select the hexahedron you created.

✧ Adjust the XYZ axes to move the hexahedron to the desired position within the assembly. (Figure 11-10)

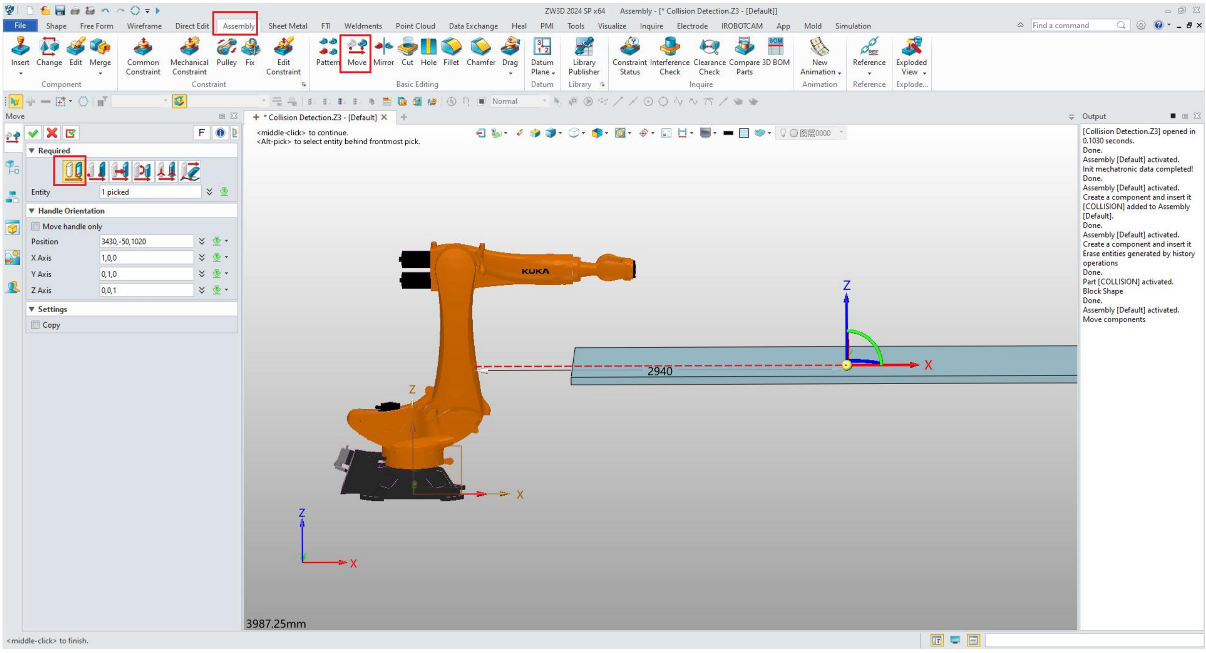


Figure 11-10：Adjust the hexahedron

## Step 4：Add Rigid Body and collision

- In the “IROBOTCAM” column, click on “Mechatronic”.
- Right-click on “Basic Mechatronic Objects” and select “Add Rigid Body and Collision”.
- Click the + button and choose the hexahedron you created from the list.
- Choose the “”Default” material.
- In the collision window, Under “Property Settings”, select “”Support Sensing”. (Figure 11-11)
- Click “OK” to finalize the creation of the rigid body and collision.

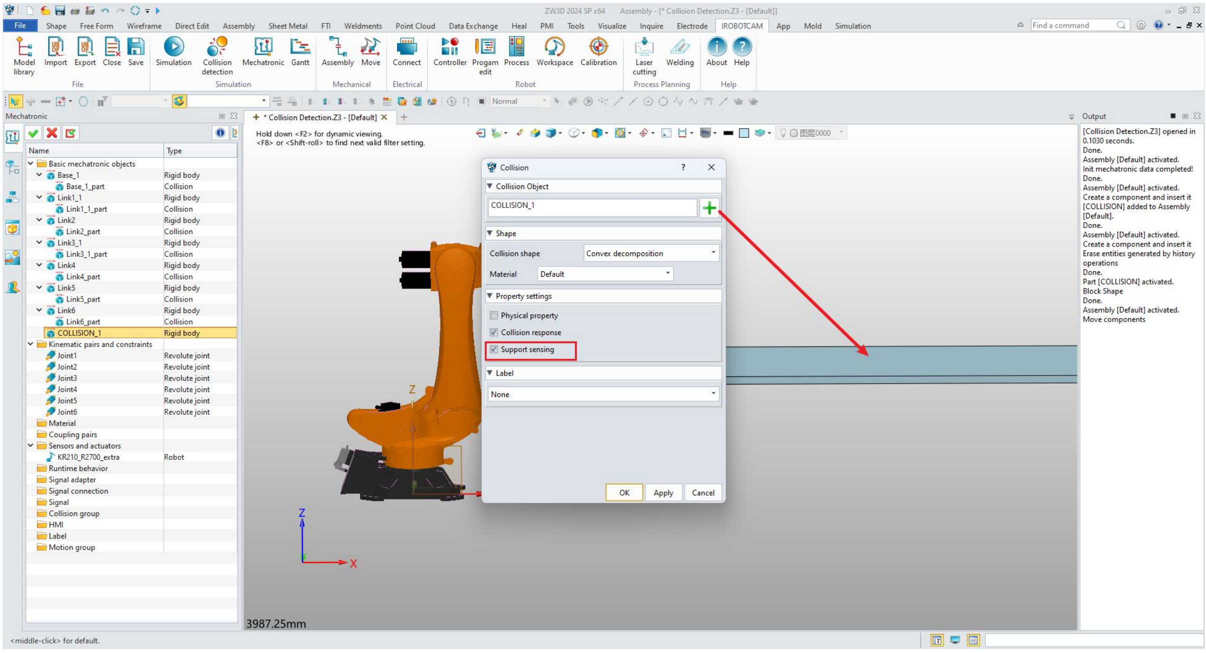



**Figure 11-11：Add rigid body and collision**

## Step 5：Add collision group

- Right-click on “Collision Group” to add a new group.
- Select multiple components representing the robot's rigid bodies.
- Click the “Add” button to add them to the box on the right.
- Alternatively, hold the “Shift” key and click to quickly add all robot rigid body components.
- Name the newly created collision group “Robot”. (Figure 11-12)

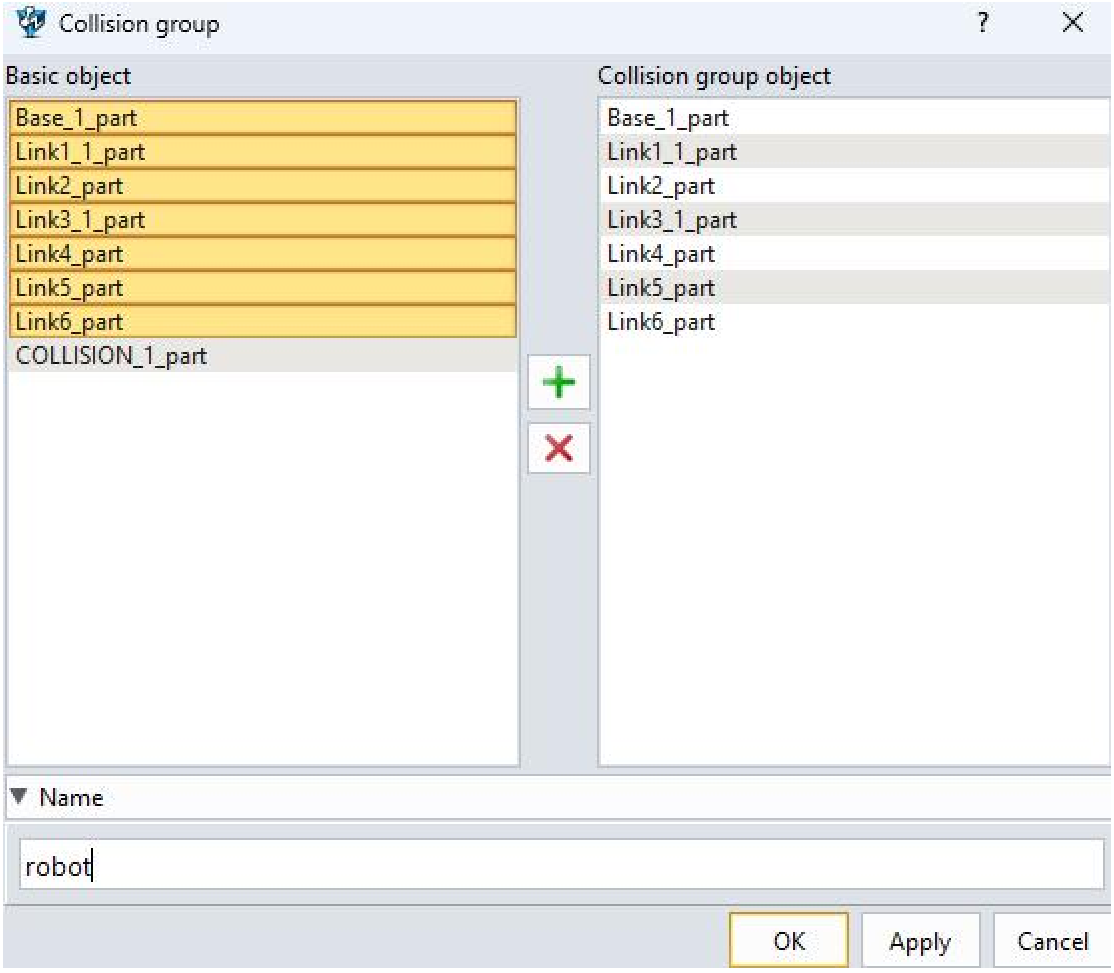



**Figure 11-12：Robot collision group**

- Repeat the above steps to create a new collision group for the hexahedral rigid body.
- Name this group “object”. (Figure 11-13)

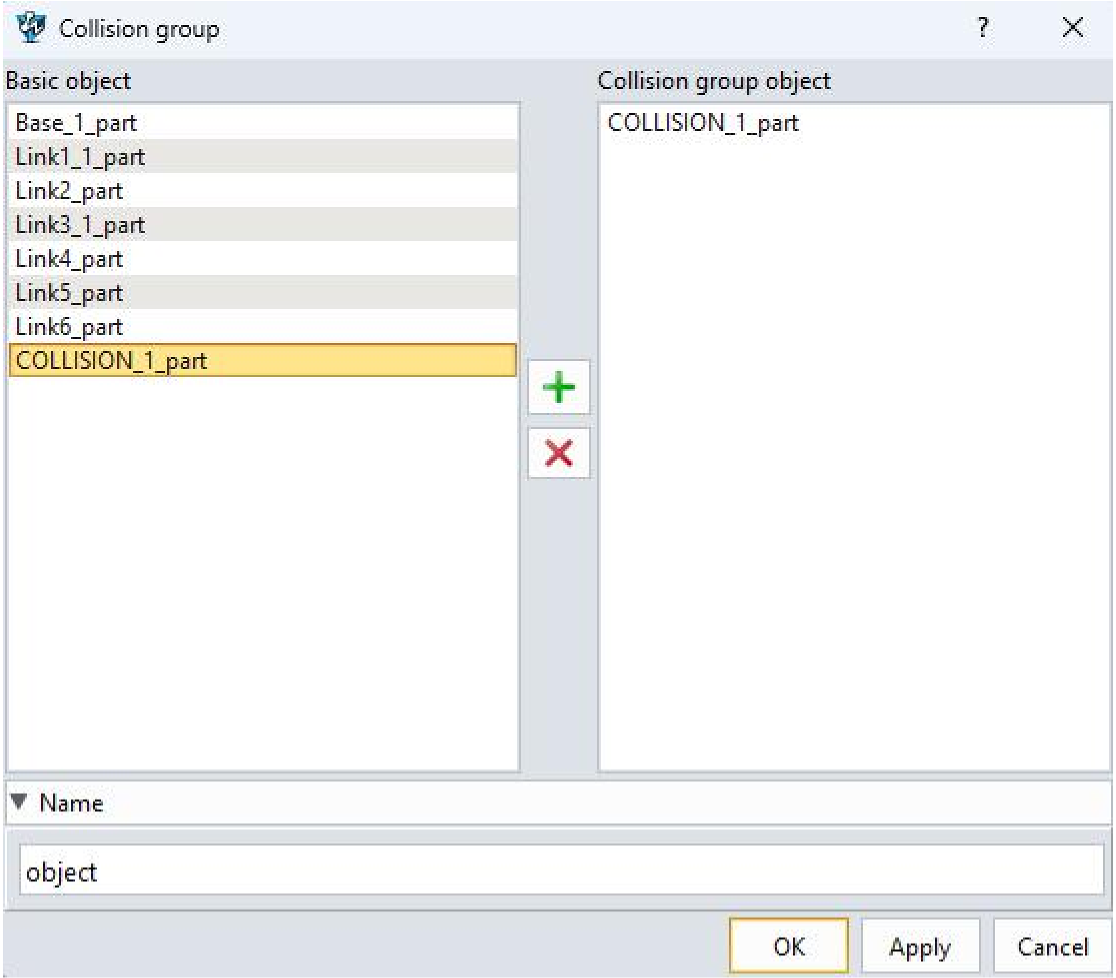



**Figure 11-13：Object collision group**

## Step 6：Add collision detection

- In the “”IROBOTCAM” column, click on the “Collision Detection” button. (Figure 11-14)

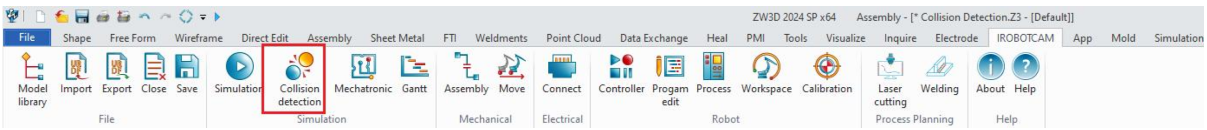



**Figure 11-14：Collision Detection**

- Select the content shown in the figure under the output boxes of “Target object” and “Obstacle” respectively.
- These should correspond to the previously created “object” and “robot” groups.
- Click “Add Collision” to create two collision pairs.
- Click the “checkmark” to complete the creation. (Figure 11-15)

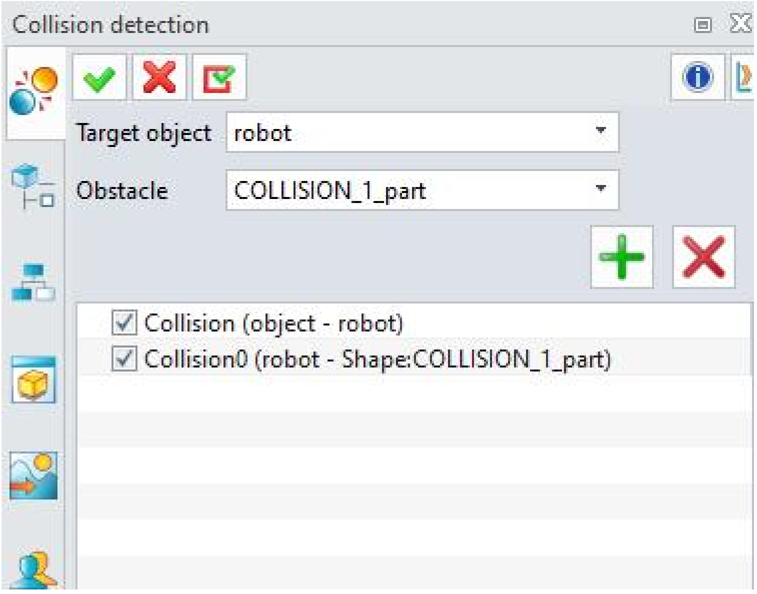



**Figure 11-15：Collision detection**

## Step 7：Add marker points and commands

- Under “Sensors and actuators”, right-click on “Robot” and select “Teach Robot”. (Figure 11-16)

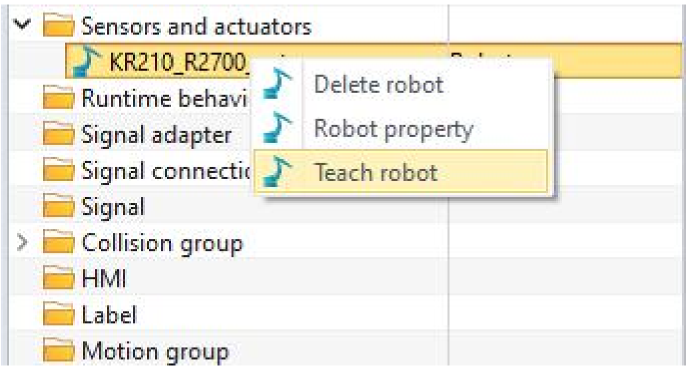



**Figure 11-16：Teach Robot**

- Click on “Drag Teach”, then click anywhere on the robot.
- Drag the X-axis downwards until you reach the bottom of the hexahedron.
- Click to “Record End point” at this location. (Figure 11-17)

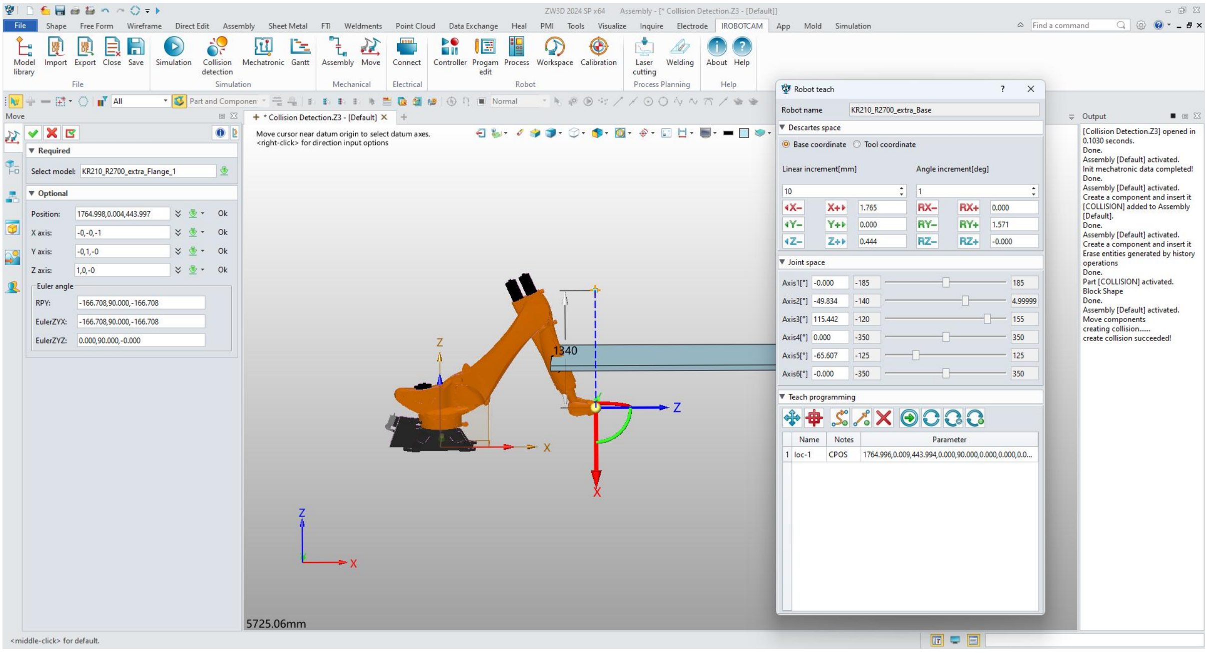


**Figure 11-17：First marking point**

- Close drag teach window and add a record point above the hexahedron. (Figure 11-18)
- This ensures the robot collides with the hexahedron when the command is executed.

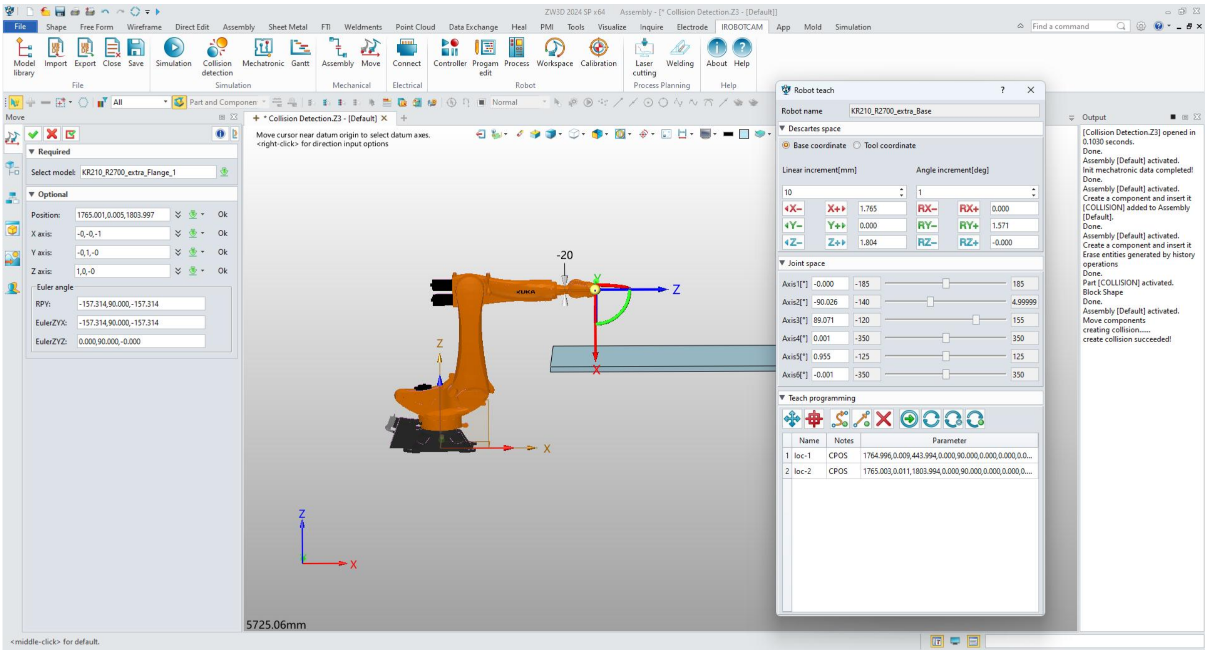


**Figure 11-18：Second marker**

- In the "IROBOTCAM" column, click on "Program Edit". (Figure 11-19)

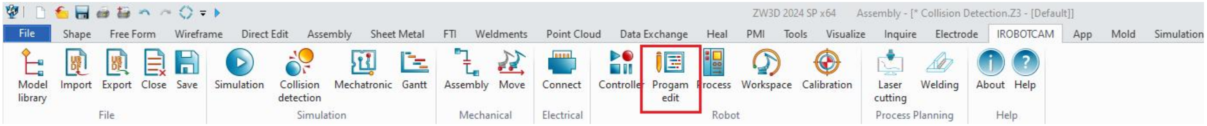



**Figure 11-19：Program edit button**

- Right-click on “Controller” and select “”Add JOB”.
- Right-click on “JOB1” and select “Add Program”.
- Right-click on “program1” and select “Edit Program”.
- Use the “MOVL” command and specify “loc-1” and “loc-2” as the motion targets. (Figure 11-20)

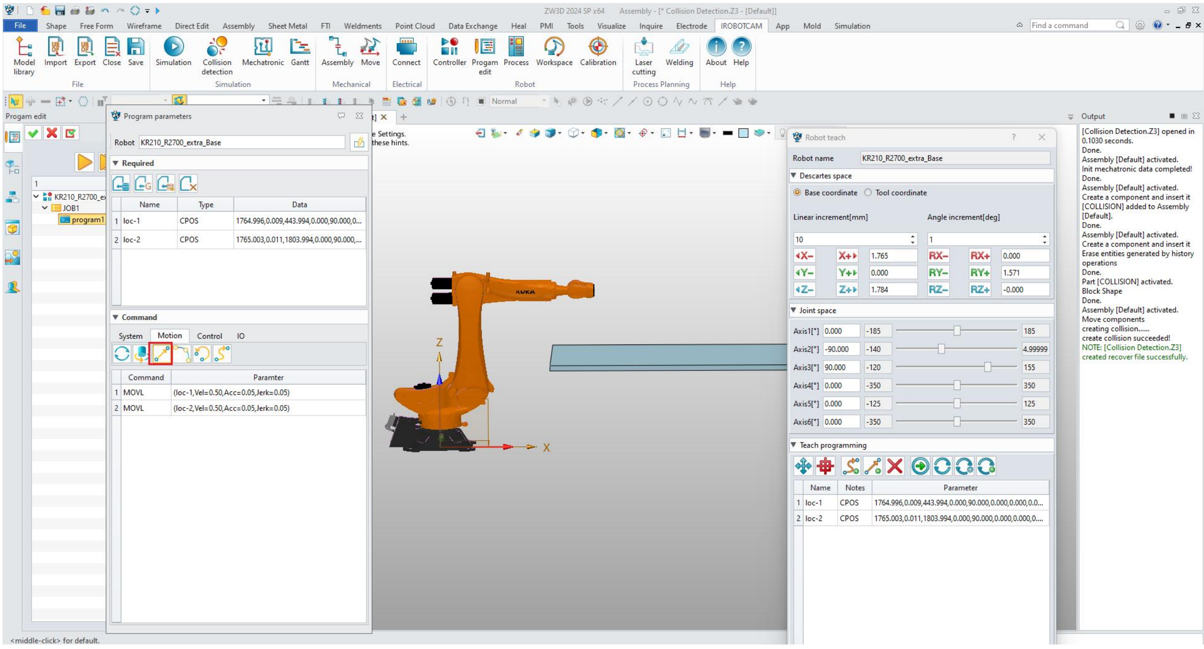



**Figure 11-20：Add command**

## Step 8：Simulation Verification

- In the “IROBOTCAM” column, click on “Program Edit”. (Figure 11-21)

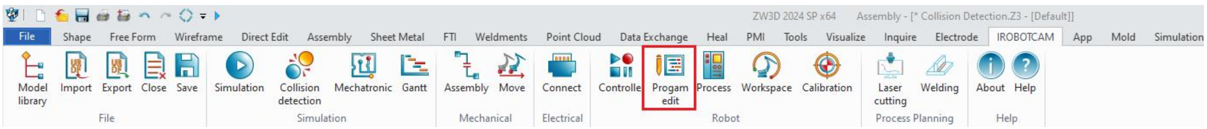



**Figure 11-21：Program edit**

- In the “Program Edit” window, click the “Run” button. (Figure 11-22)

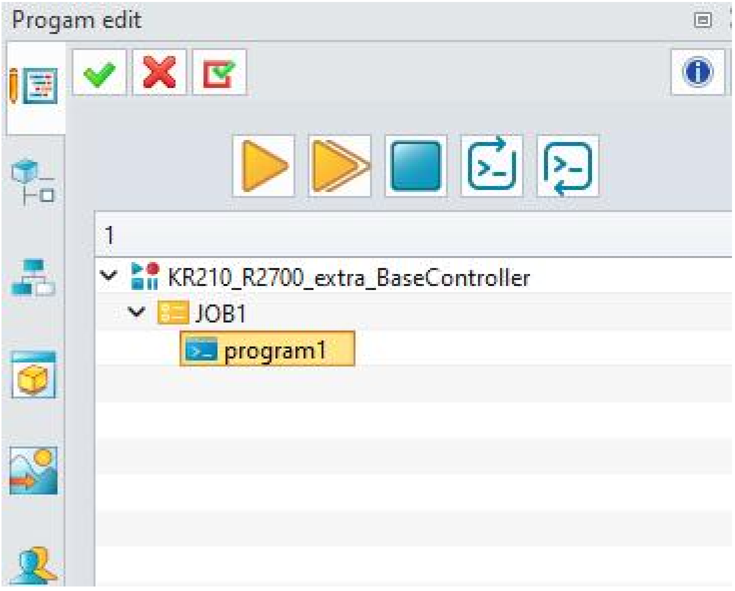



**Figure 11-22：Run button**

- If the program doesn't start running immediately, try clicking “Run” once, waiting a few seconds, and then clicking “Stop”.
- Refresh the interface and then click “Run” again. This sometimes triggers the program to start running.
- Once the program is running, you should see a highlighted status indicator, confirming successful collision detection. (Figure 11-23)

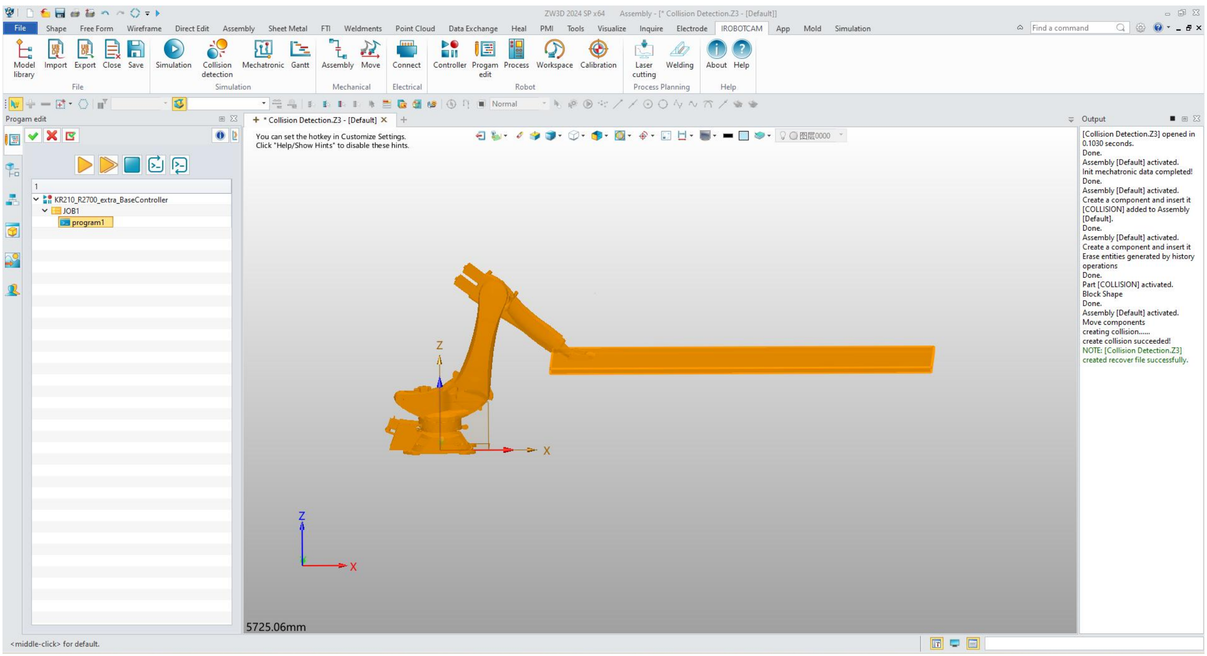


**Figure 11-23：Simulation verification**

# Chapter 12: Configure Communication

Connecting iRobotCAM Model to External Controller.(Figure 12-1)

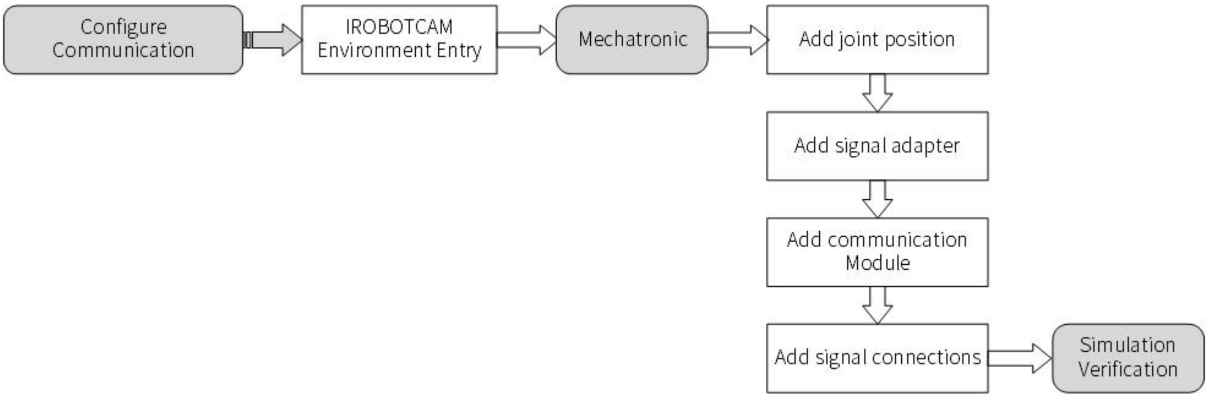



**Figure 12-1：Flow chart of configure communication**

## Step 1: Open iRobotCAM project file

- Open iRobotCAM project file "Configure Communication.Z3" to enter the iRobotCAM environment. (Figure 12-1)

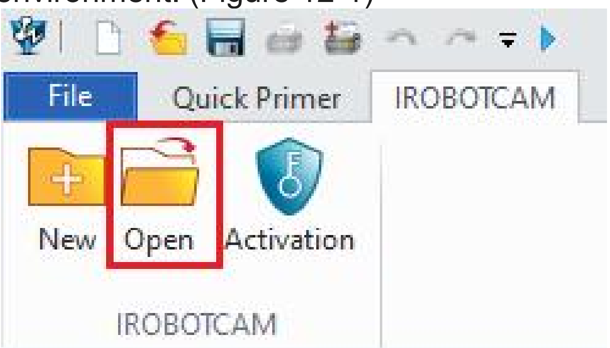



Figure 12-2：**Open iRobotCAM project file**

- Wait for the model to import. Once complete, there is a message will appear in the red box below the output box on the right. (Figure 12-3)

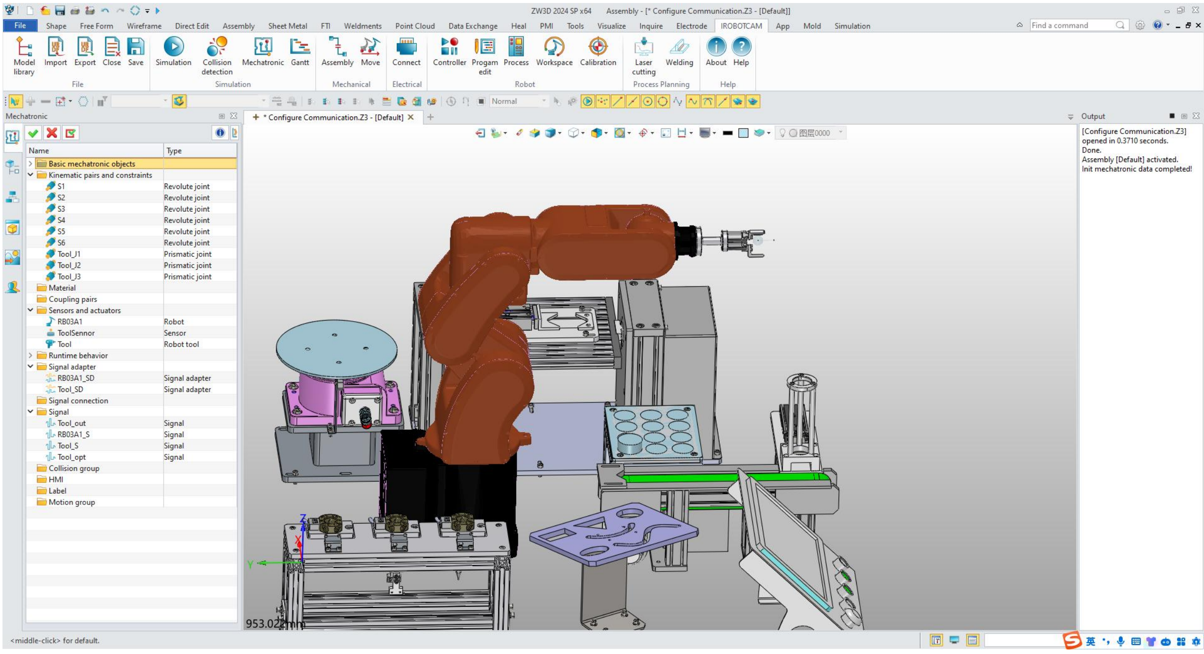


Figure 12-3：**Component details**

## Step 2: Add Joint Positions

- In the "IROBOTCAM" column, click the "Mechatronic" button, then "Runtime behavior".
- Right-click on "Robot_EXEC" and delete the simulation sequence. (Figure 12-4)

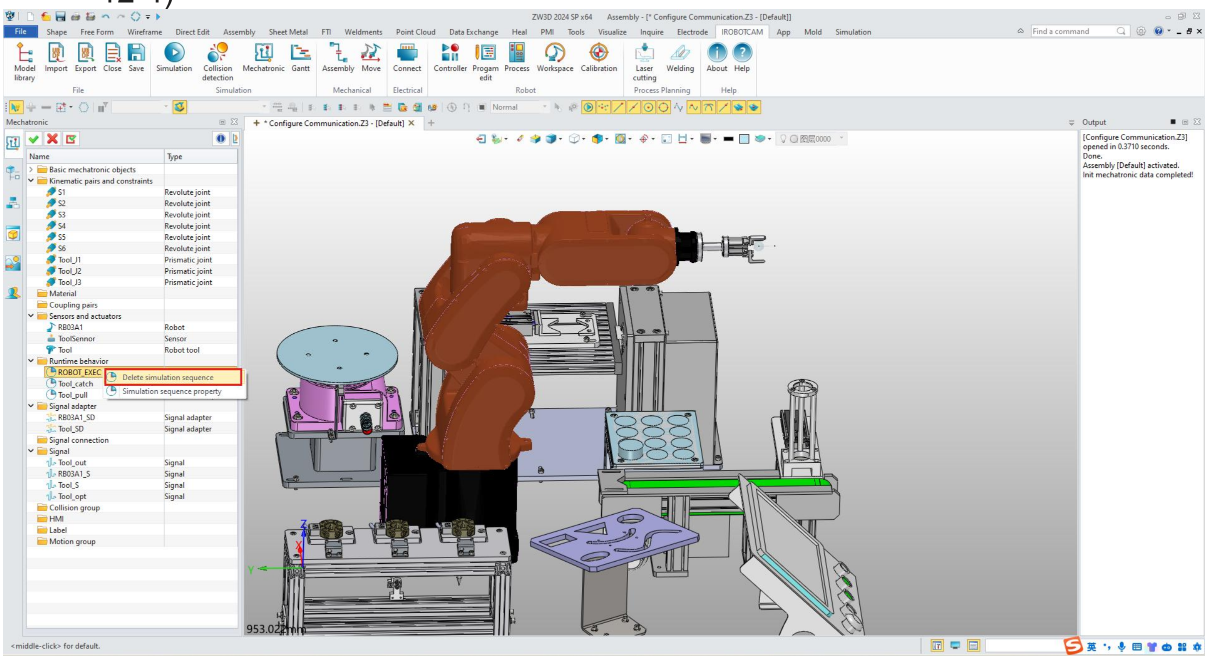


Figure 12-4：**Delete the simulation sequence**

- Click on "Sensors and actuators" and right-click to add joint positions. (Figure 12-5)

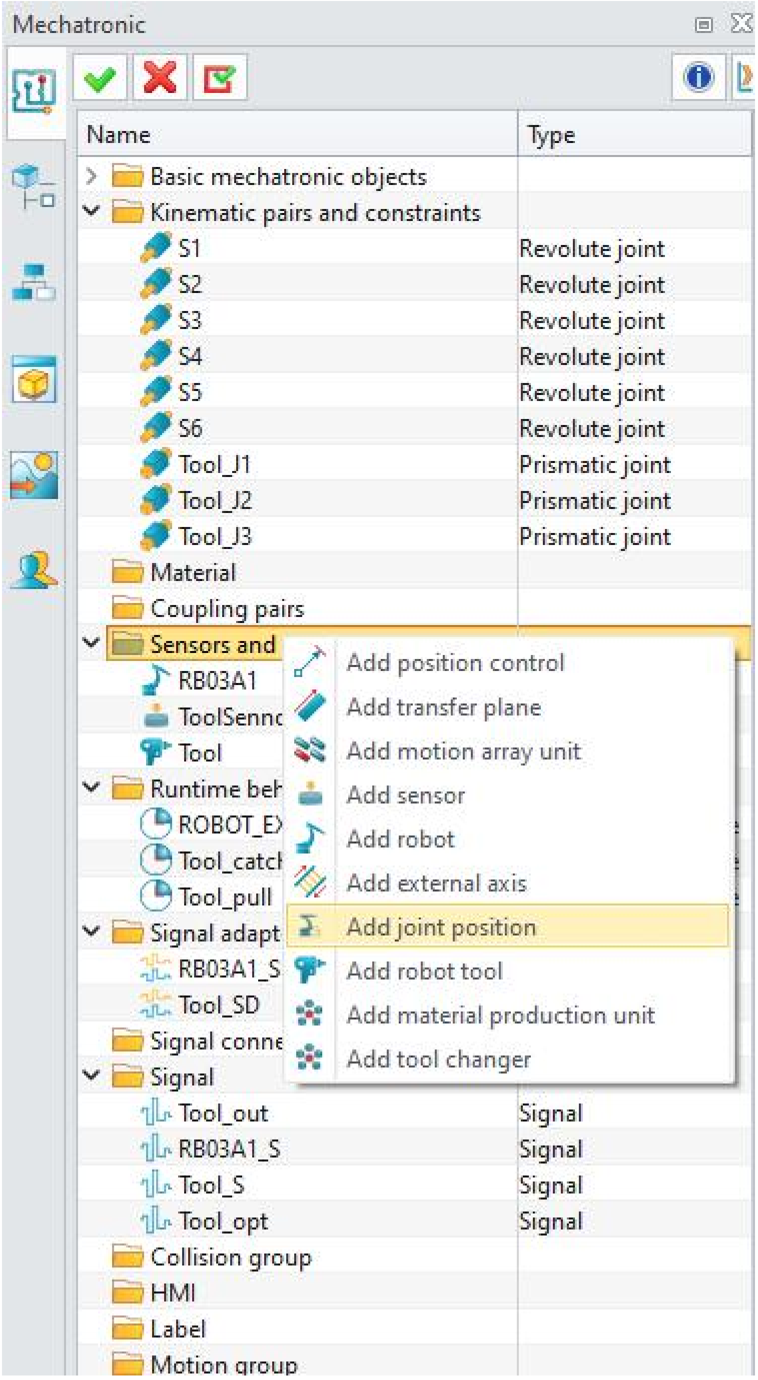


Figure 11-23：**Add joint position**

- For each joint "S1" to "S6", follow these steps:

✧ Click the "+" button.
✧ Select the joint (eg, "S1").
✧ Name it "S1Pos" (replace "1" with the appropriate joint number).
✧ Set the motion type to "Specify position".
✧ Click OK. (Figure 12-6)

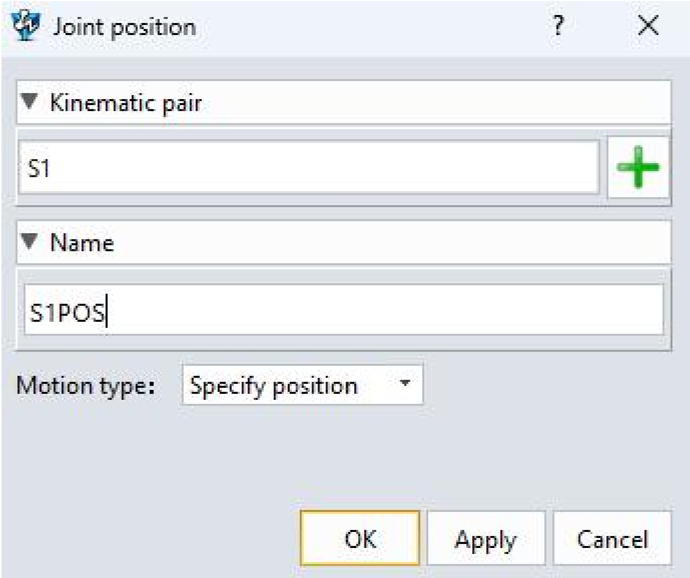



Figure 12-6：**Joint position parameters**

- After completing steps for all joints, the interface should resemble.

(Figure 12-7)

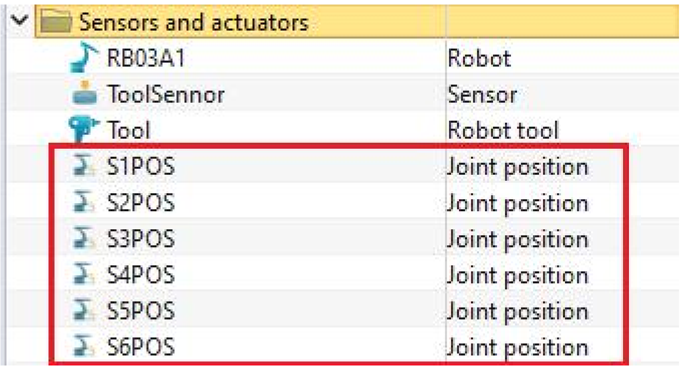



Figure 12-7：**Joint position completed**

## Step 3: Add Signal Adapter

- Click on "Signal Adapter".
- Right-click and select "Add Signal Adapter".
- Follow these steps for each joint: (Figure 12-8)

✧ Click the "+" button.
✧ Select the joint position (eg: "S1Pos").
✧ Double-click the second box and enter the name "S1Pos_S" (replace "1" with the appropriate joint number).
✧ Select "Float" for the data type in the third box.
✧ Click the "+" button on the right in the second column.
✧ Fill in "S1Pos_S, S1Pos_S * 0.017444" (the value source 3.14/180).
✧ Name the adapter "S1Pos_SD" (replace "1" with the appropriate joint number).

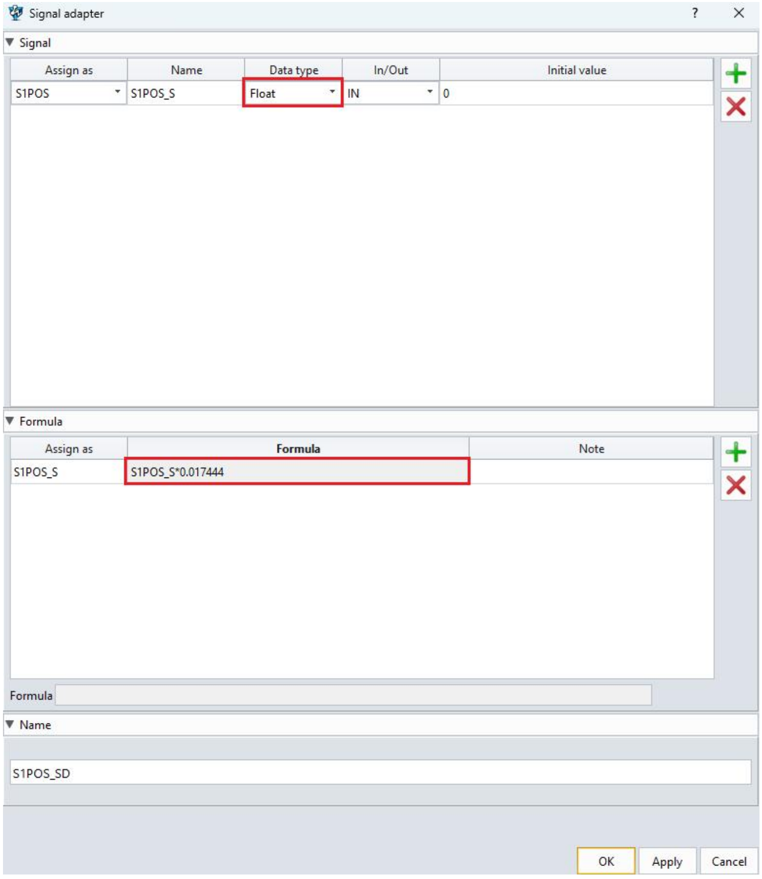


Figure 12-8：**Signal adapter S1Pos_SD**

- The final result should look similar to .(Figure 12-9)

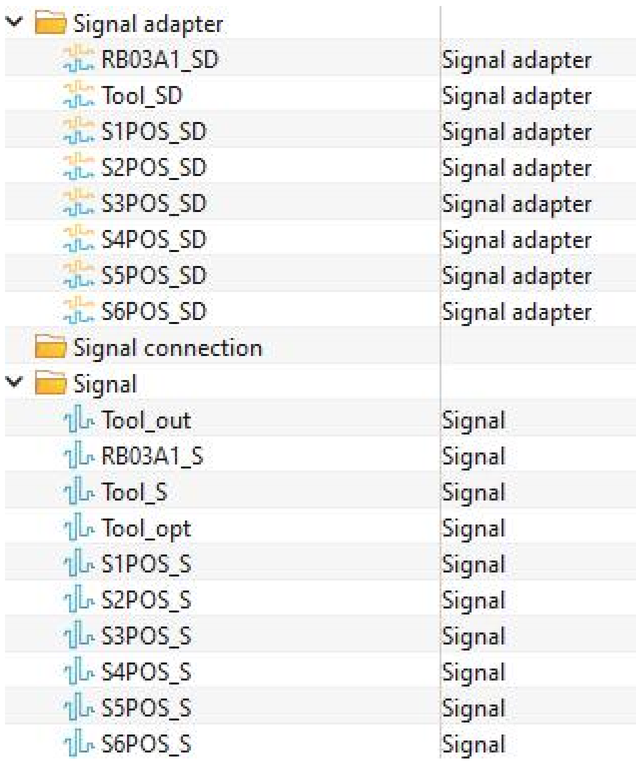


Figure 12-9：**Signal adapter completed**

## Step 4: Add Communication Module

- Click on the "Connect" button in the iRobotCAM column. (Figure 12-10)

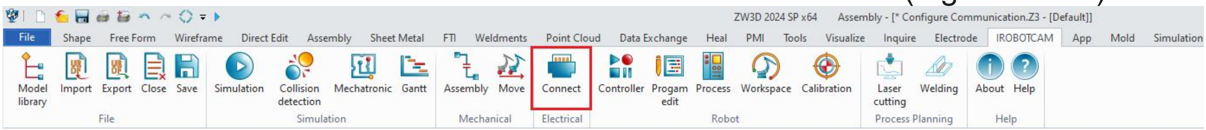


Figure 12-10：**Connect button**

- Under "Modbus Tcp", right-click the blank box under "Devices" and select "Add Device". (Figure 12-11)

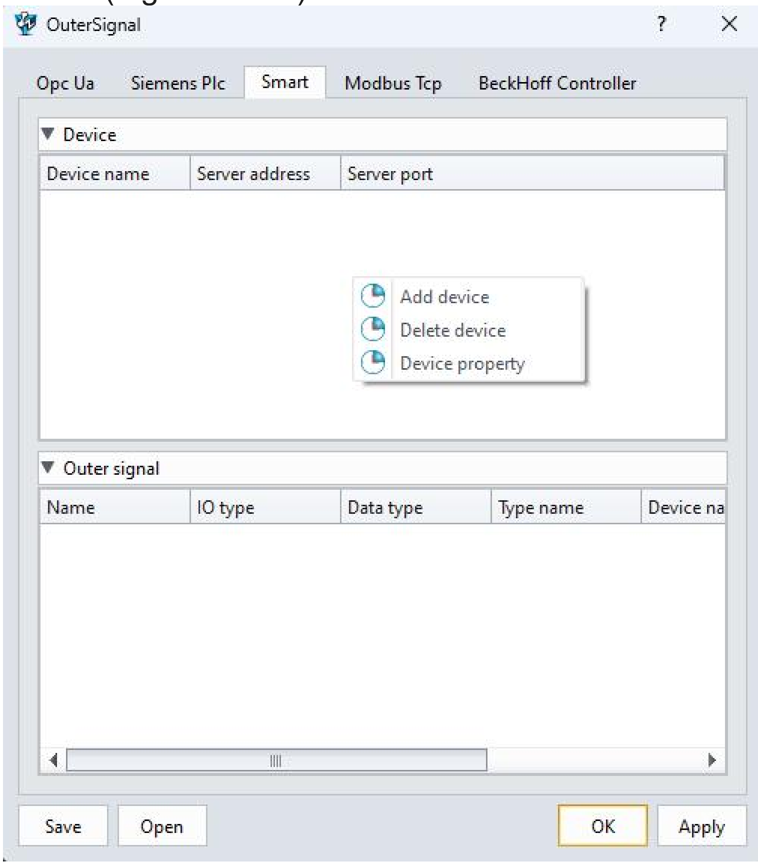



Figure 12-11：**Outer signal**

- In the pop-up window: (Figure 12-12)Here is an example setting.
- ✧ Name the device "ModbusTcp1".
- ✧ Set the server address to "192.168.1.10".
- ✧ Set the server port to "502".
- ✧ Set the mode to "1".
- ✧ Click OK.

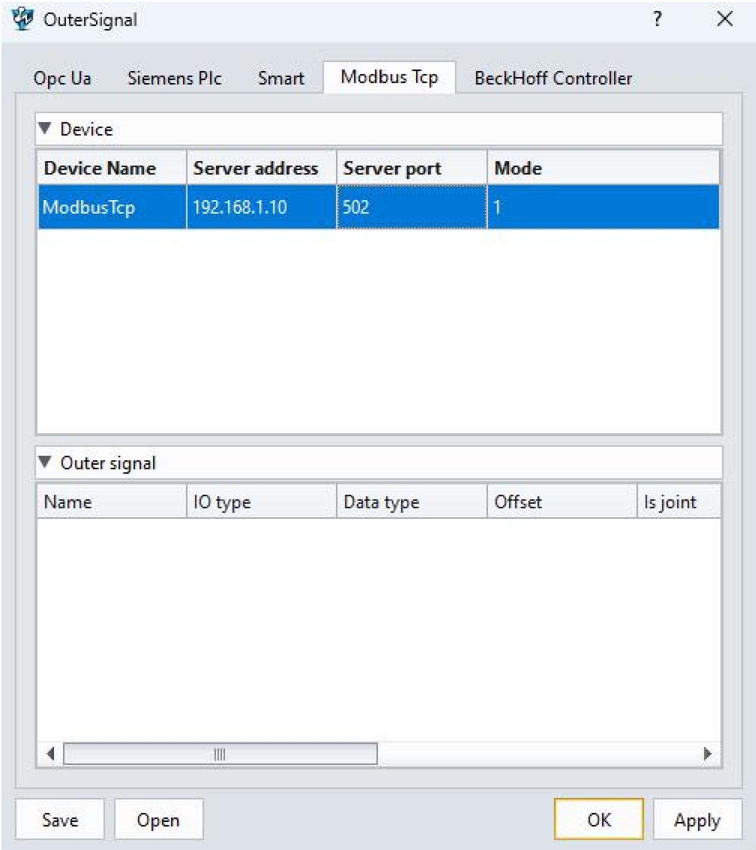



Figure 12-12：**Outer signal device parameters**

- Under "Modbus Tcp", right-click the blank box under "Outer Signal" and select "Add Signal". (Figure 12-13)

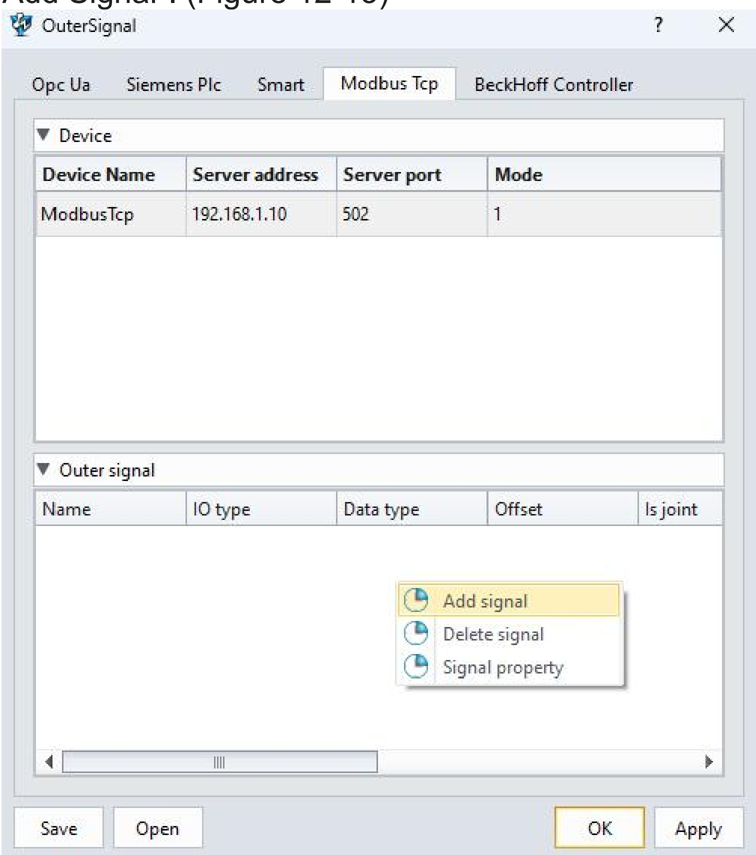


Figure 12-13：**Add signal**

- In the pop-up window: (Figure 12-14)
  - ✧ Name the signal "J1".
  - ✧ Select "Double" for the signal type.
  - ✧ Select "OUT" for the IO type.
  - ✧ Select "Yes" for the joint.
  - ✧ Set the offset address to "65536".
  - ✧ The remaining window settings are filled in according to the actual device.
  - ✧ Click Add.

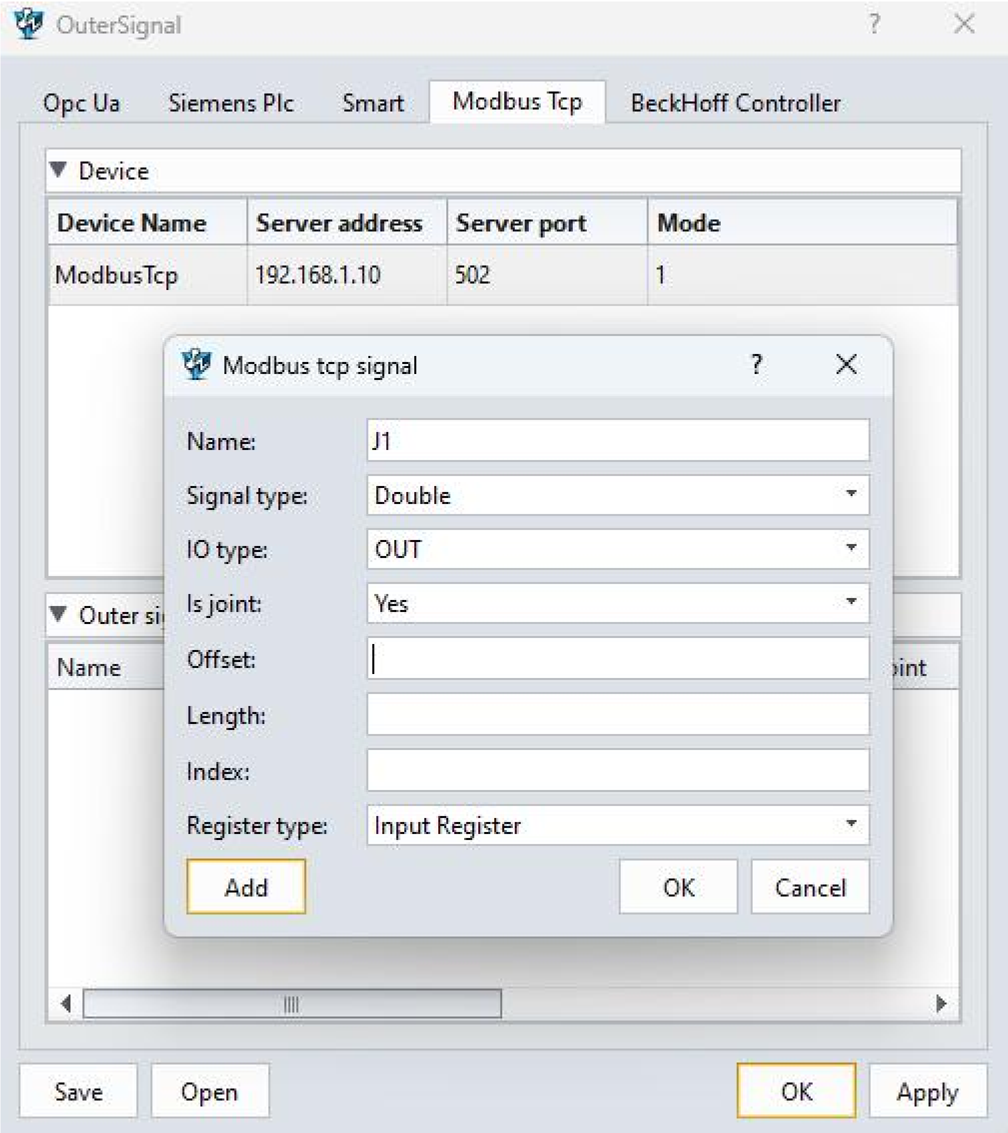



Figure 12-14：**Add joint signal**

- Repeat step 5 for the remaining joints, naming them "J2" to "J6", selecting "Double" for the signal type, "OUT" for the IO type, "Yes" for the joint, and offset addresses incrementally increasing by 4 (eg: 65540, 65544). (Figure 12-15)

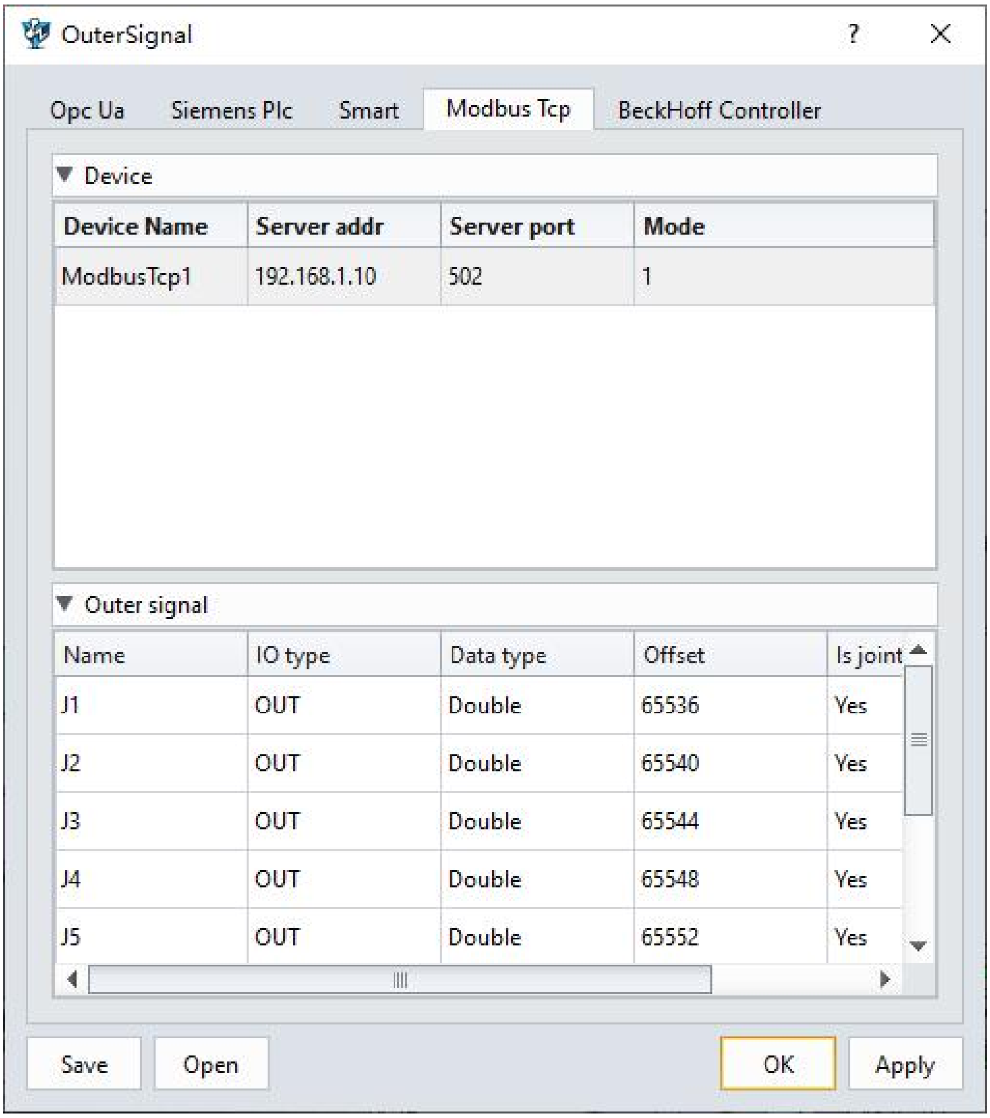



Figure 12-15：**Joint signal completed**

- Add communication for grabbing and releasing: (Figure 12-16)
- ✧ Name the signal "Grab".
- ✧ Select "Bool" for the signal type.
- ✧ Select "OUT" for the IO type.
- ✧ Select "No" for the joint.
- ✧ Set the offset address bit to "65560".
- ✧ The remaining window settings are filled in according to the actual device.

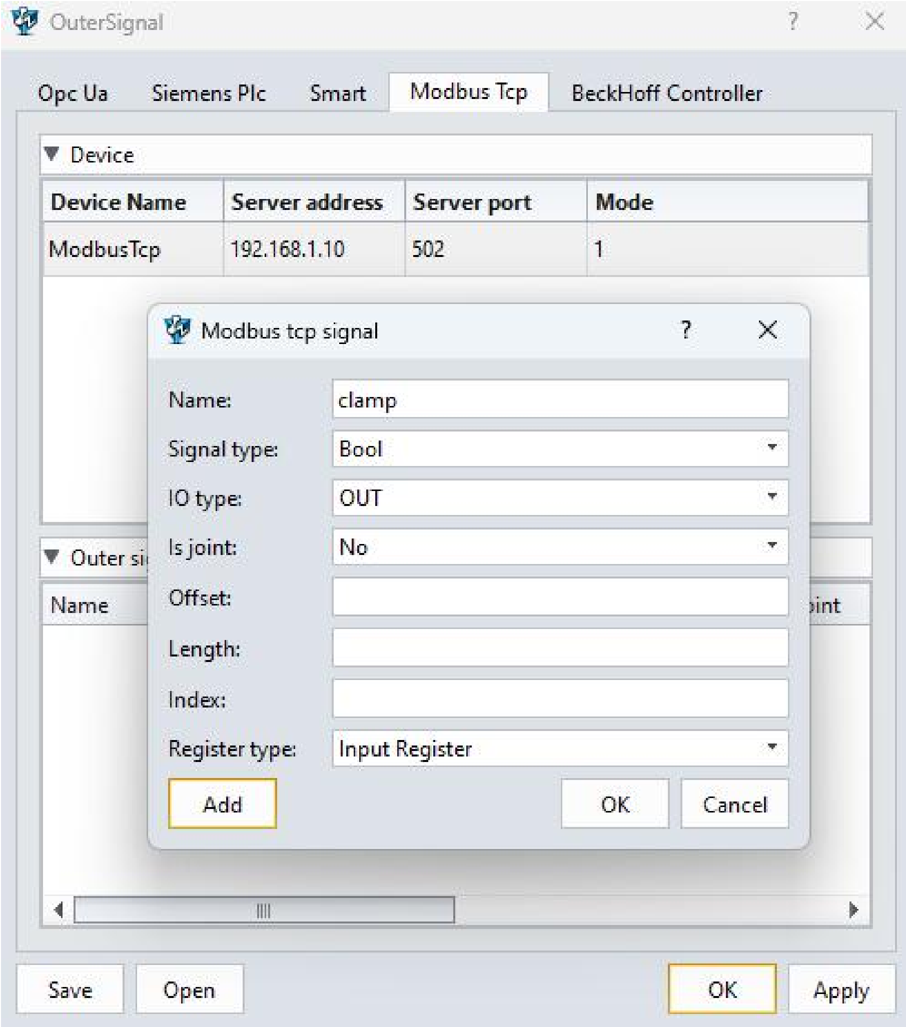



Figure 12-16：**Pick and place outer signals**

- Click the "Save" button to save the external signal connection information to the computer for future use. (Figure 12-17)

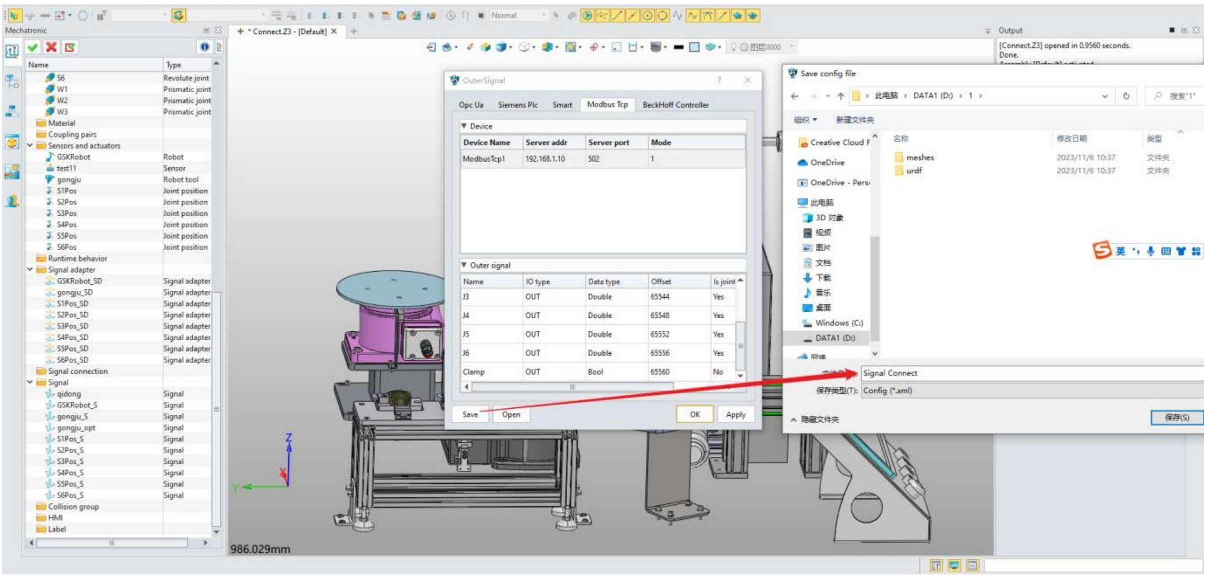


Figure 12-17：**Save signal**

Connecting iRobotCAM Model to External Controller (Continued)

## Step 5: Add Signal Connections

- Under "Mechatronic", locate "Signal Connection". (Figure 12-18)

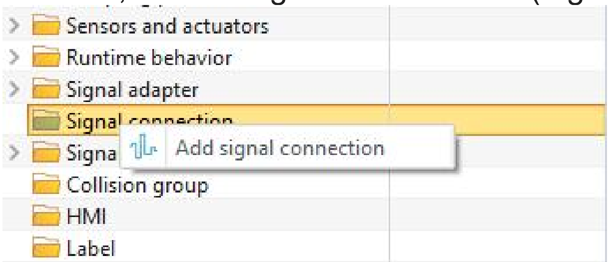



Figure 12-18: **Add signal connection**

- Right-click and select "Add Signal Connection".
- In the pop-up box, name it "SignalConnect1" and click "OK". (Figure 12-19)

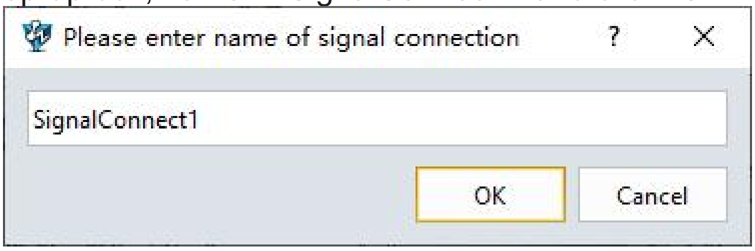



Figure 12-19: **Signal connect name**

- Within the "Signal Connection" box:
- ✧ Click the internal signal "S1Pos_S" in the left column. (Figure 12-20, point ①)
- ✧ Click the external signal "J1" in the right column. (Figure 12-20, point ②)
- ✧ Click the "Connect" button.

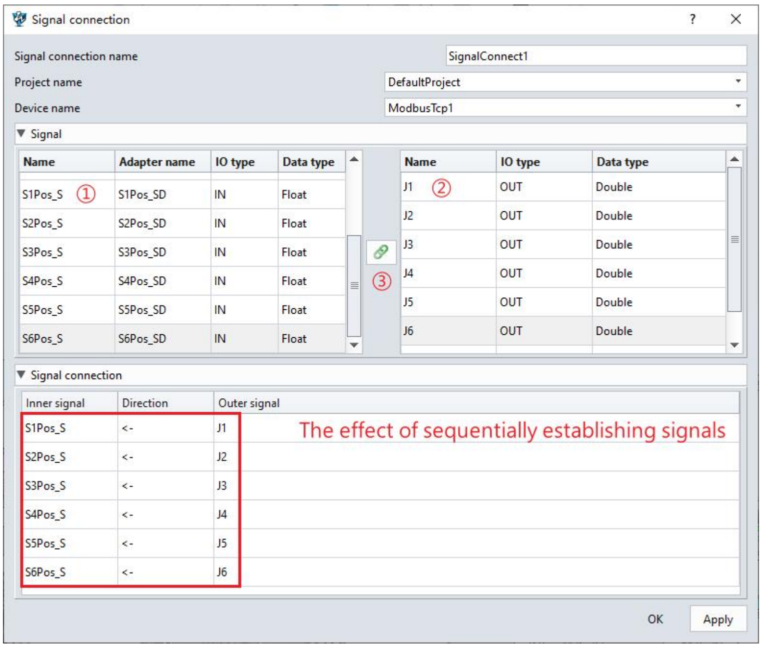



Figure 12-20: **Connection method**

- Repeat steps for all remaining pairs of internal and external signals, connecting them according to the order shown in Figure 12-20 (① -> ② -> ③).
- This step allows the robot to simulate grabbing and dropping by controlling its internal signals through external signals.

## Step 6: Simulation Verification

- In the "IROBOTCAM" column, click the "Simulation" button. (Figure 12-21)


Figure 12-21：**Simulation verification**

- In the "Simulation Control" window, click the "Start Simulation" button on the far left side. (Figure 12-22, point ①)
- You will see the device server you added, identified by ②, along with its ③ external signals. (Figure 12-22)

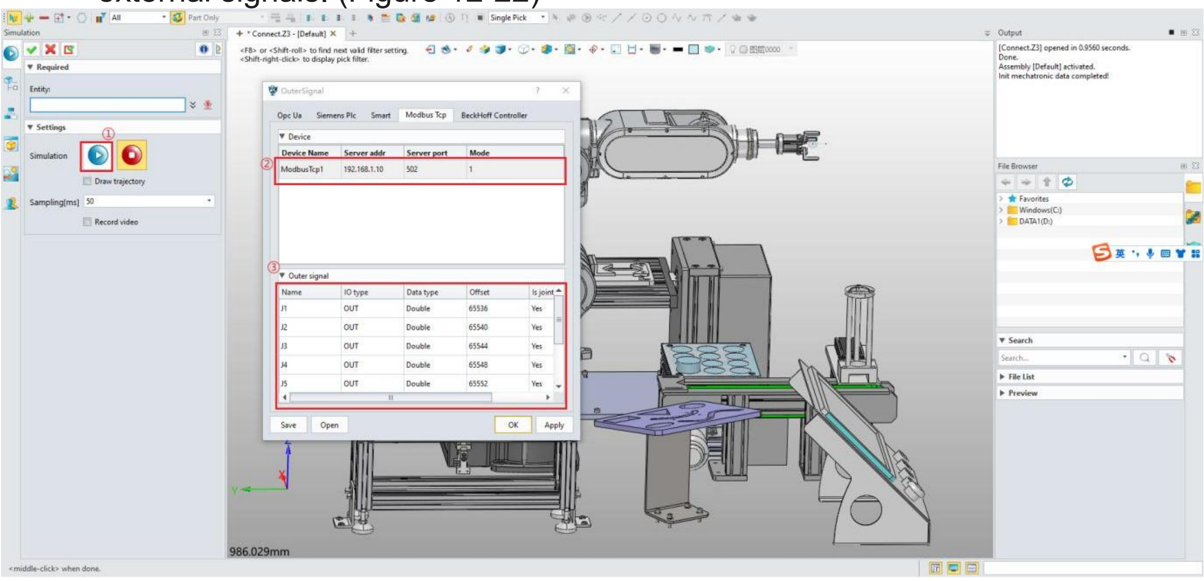



Figure 12-22：**Simulation verification**

- The external signals added through ③ control the internal signals of ④ sequentially.
- Finally, the internal signals control the ⑤ joint positions and the programmed simulation sequence, successfully establishing an external communication function. (Figure 12-23)

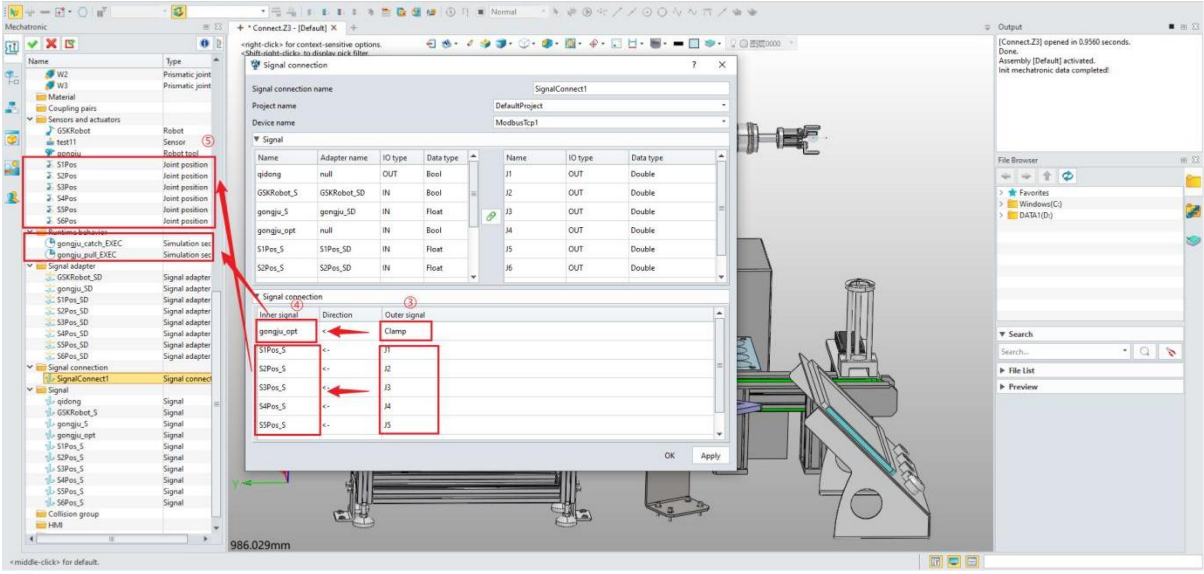


Figure 12-23：**Simulation verification**

- These steps guide you through connecting your iRobotCAM model to an external controller.

- By following these instructions, you can enable external communication and control your robot's movements through external signals.

# Chapter 13: Turntable Model

- This document presents a comprehensive guide for constructing a functional turntable model in IROBOTCAM, encompassing various elements and functionalities.(Figure 13-1)

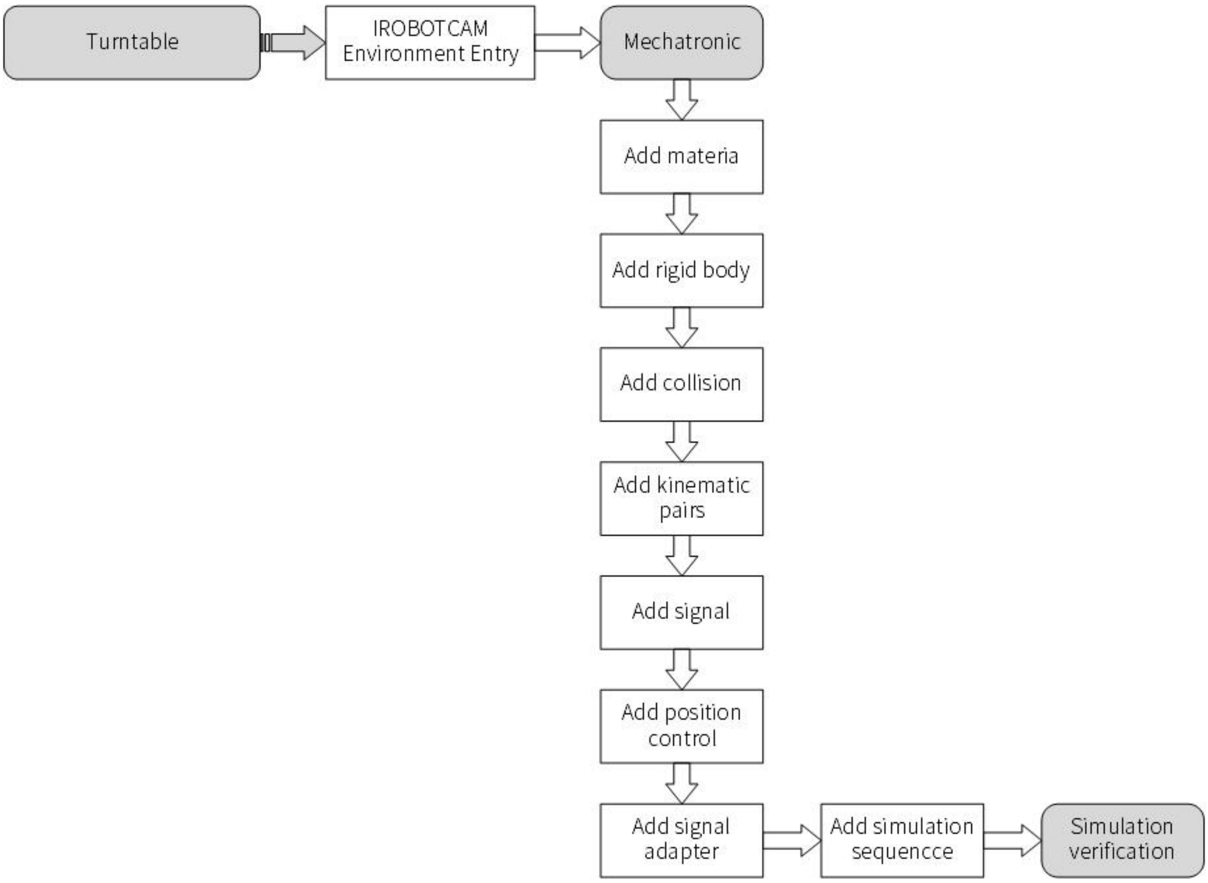



Figure 13-1：Flow chart of collision detection

## Prerequisites

- Familiarity with IROBOTCAM interface and basic functionalities.
- Understanding of mechatronic concepts like rigid bodies, collisions, joints, signals, and simulation sequences.

## Procedure:

### Step1: Model Import and Initialization

- Open iRobotCAM project file "Turntable.Z3" in IROBOTCAM.(Figure 13-2)
- Wait for the model import to complete.(Figure 13-3)
- Access the component details interface.(Figure 13-4)

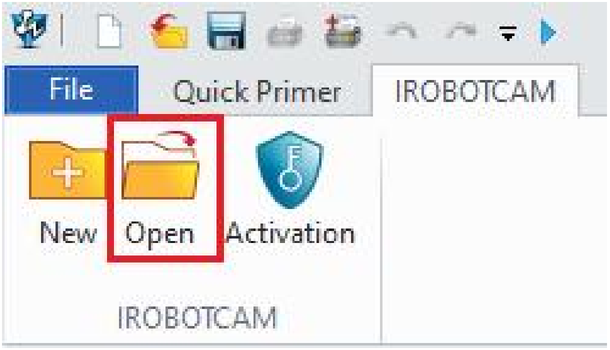



Figure 13-2：Open iRobotCAM project file

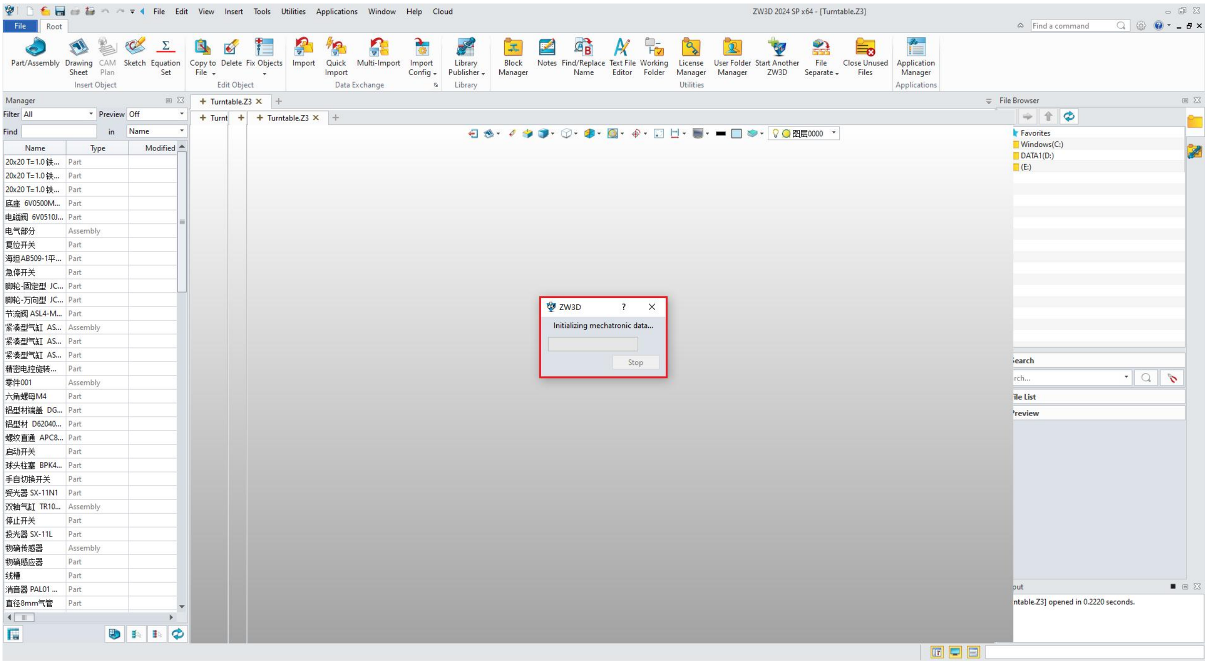



Figure 13-3：Waiting

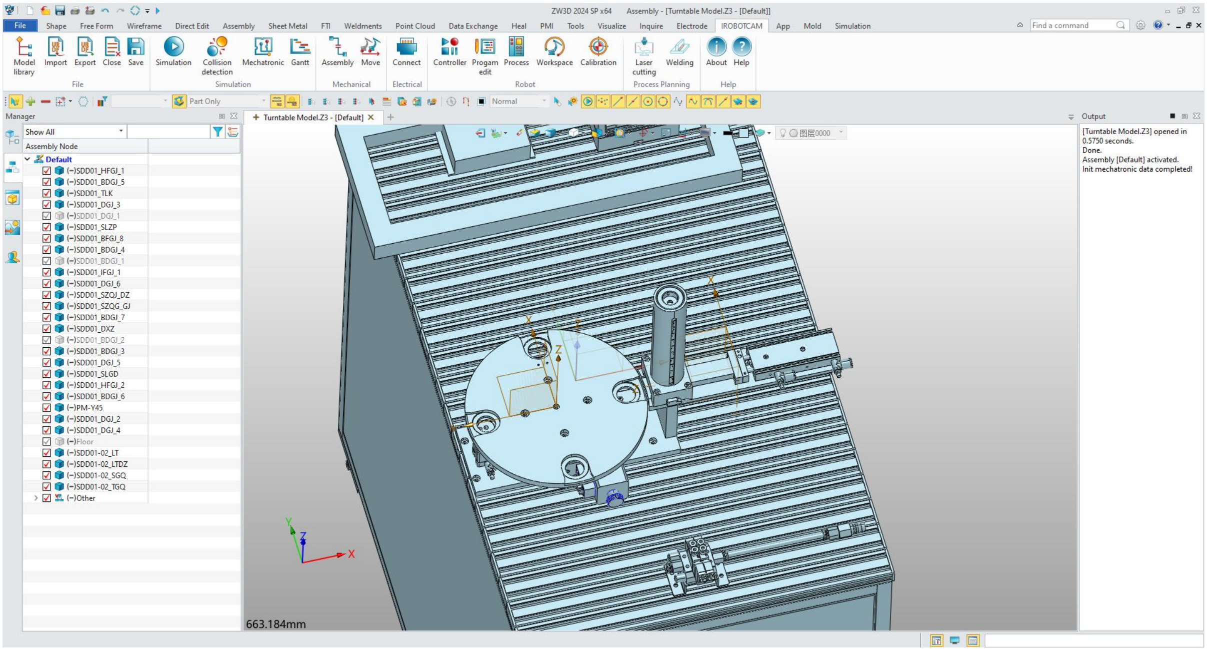


Figure 13-4：Component details

## Step2: Material Definition

- In the "IROBOTCAM" tab, click the "Mechatronic" button.(Figure 13-5)
- Open the "Materials" window.(Figure 13-6)
- Define two materials: "Material1" and "Material2".(Figure 13-7)
- Assign specific values for friction coefficient, restitution coefficient, linear and angular damping for each material.(Figure 13-7)

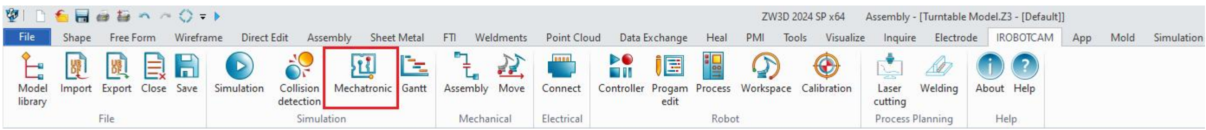



Figure 13-5：Mechatronic button

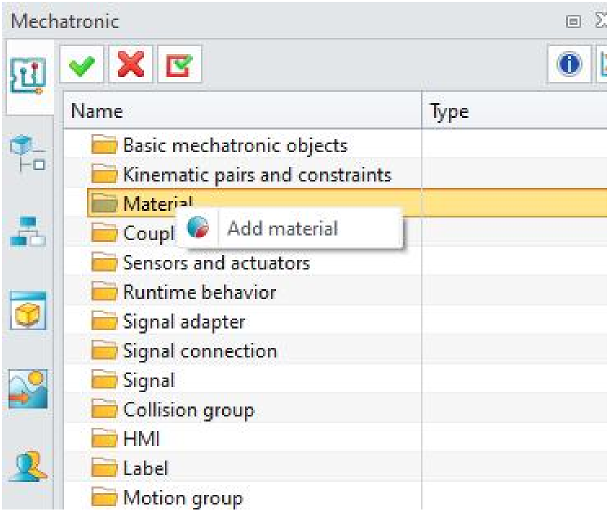



Figure 13-6：Add materia

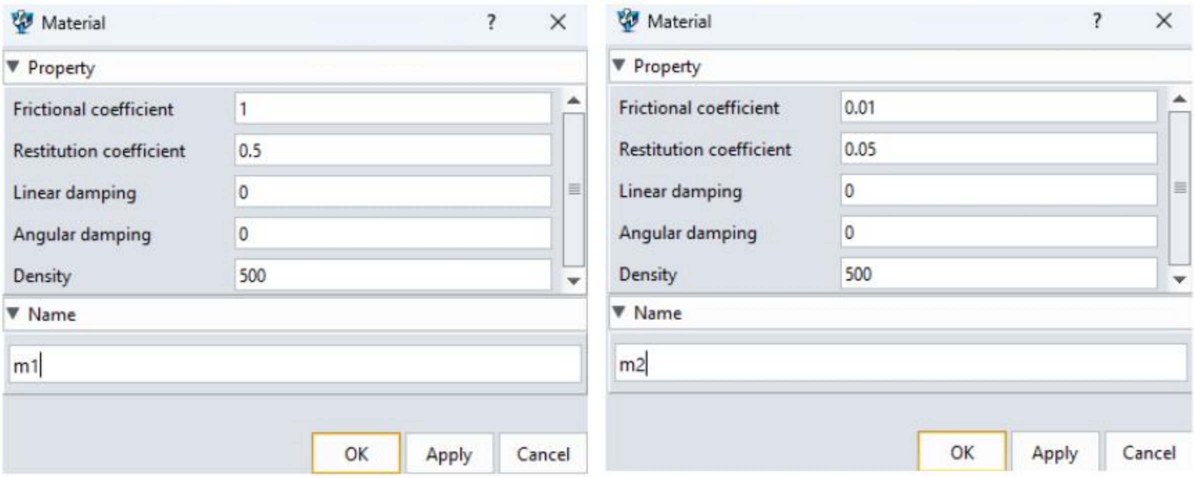



Figure 13-7：Materia1 and Materia2

## Step3: Rigid Body Creation

- Click the "Mechatronic" button.(Figure 13-8)
- Use the "+" button to create eight rigid bodies corresponding to the main turntable components.(Figure 13-9)
- Specify their masses, material assignments, and property settings.(Figure 13-10)
- Create two additional rigid bodies ("SDD01DGJ_2_1" and "SDD01DGJ3_1").(Figure 13-11)
- Assign "Material2", enable "Physical Property" and "Collision Response" for these rigid bodies.(Figure 13-12)

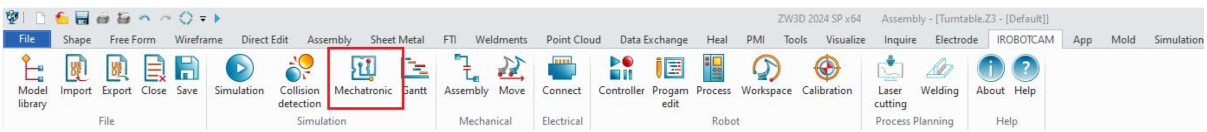


Figure 13-8：Mechatronic button

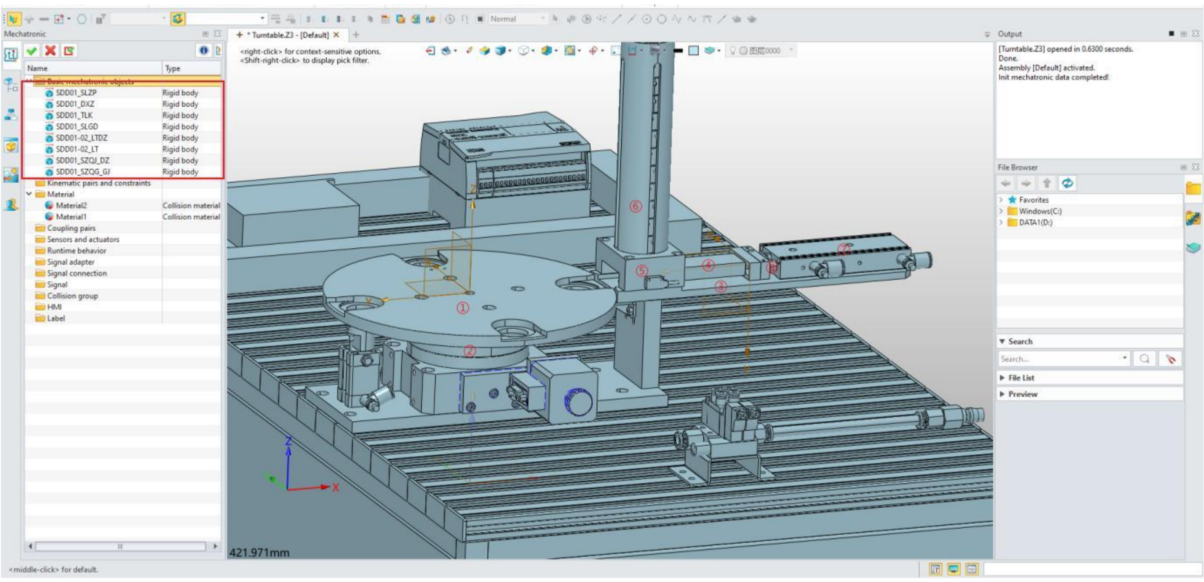


Figure 13-9：Add rigid body

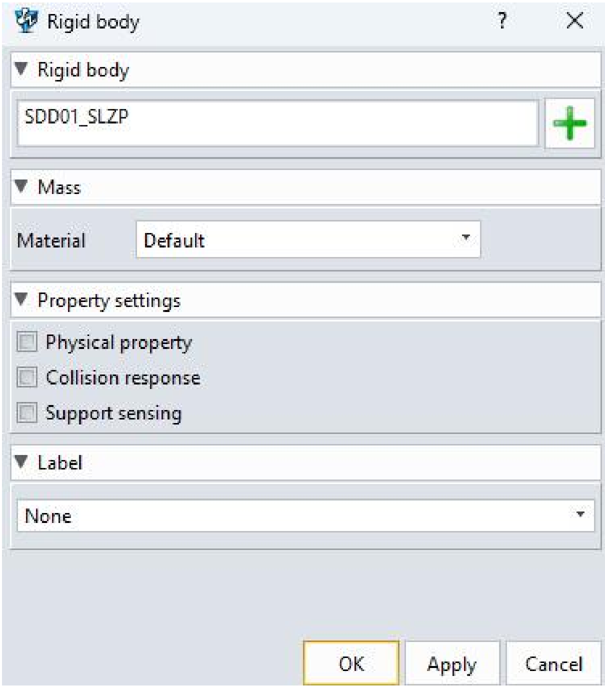



Figure 13-10：Rigid body message

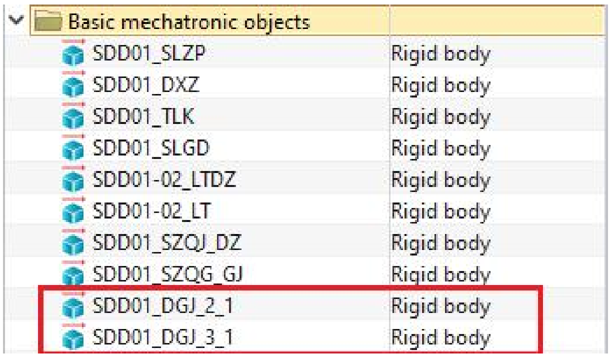



Figure 13-11：Two additional rigid body

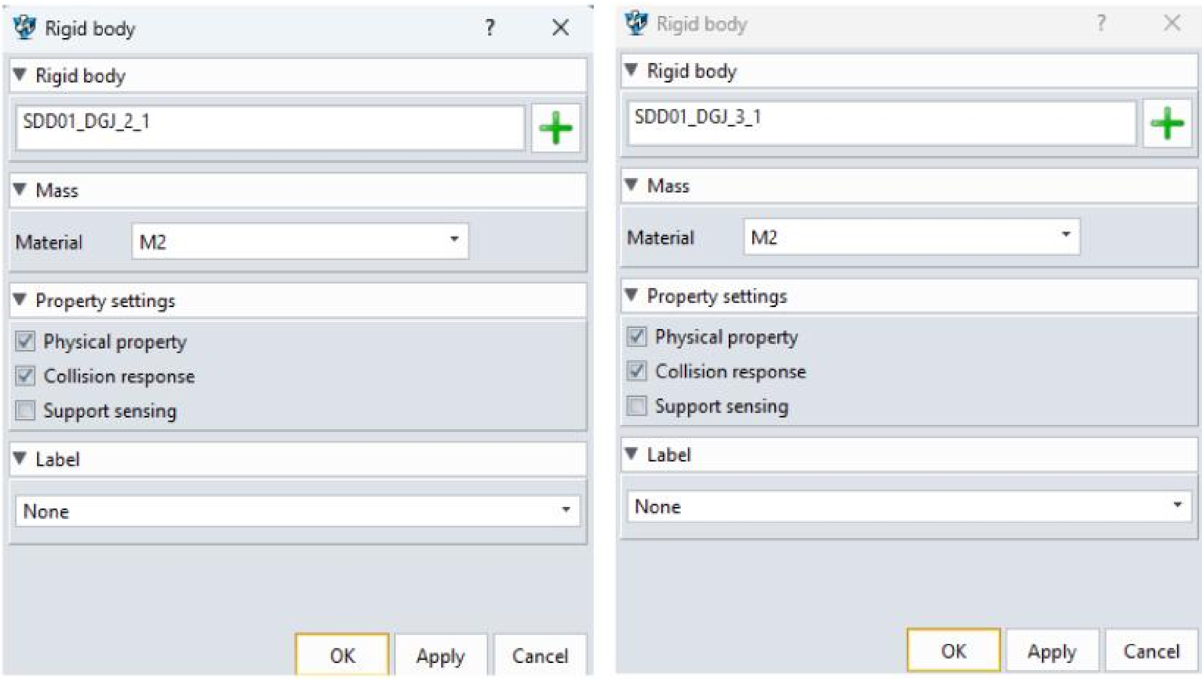



Figure 13-12：Rigid body message

## Step4: Collision Definition

- Click the "Mechatronic" button.(Figure 13-13)
- Open the "Basic Mechatronic Objects" window and select "Add Collision".(Figure 13-14)
- Create six collisions for the turntable components.(Figure 13-15)
- Choose appropriate collision shapes like "convex decomposition" or "convex hull".(Figure 13-16)
- Assign mass, material, and property settings for each collision.(Figure 13-16)

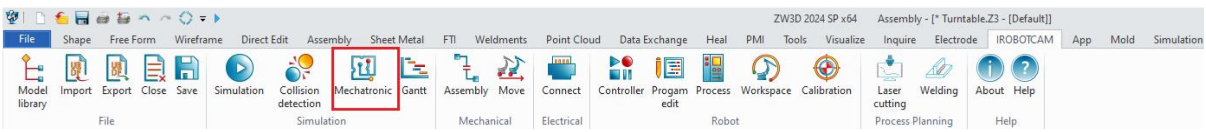


Figure 13-13：Mechatronic

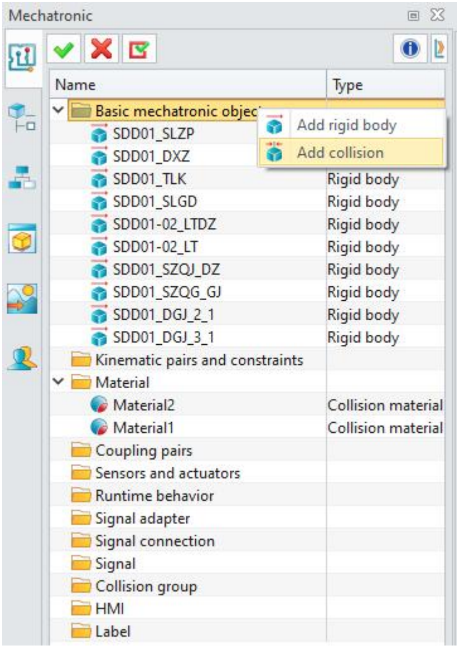



Figure 13-14：Add collision

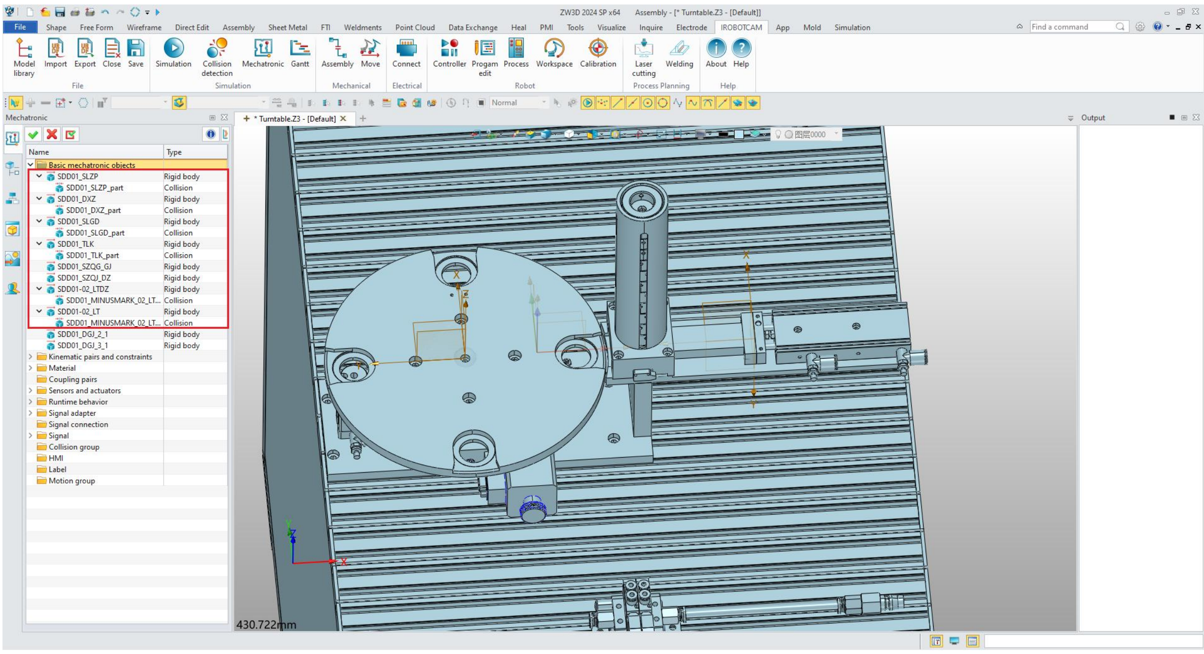



Figure 13-15：Six collision

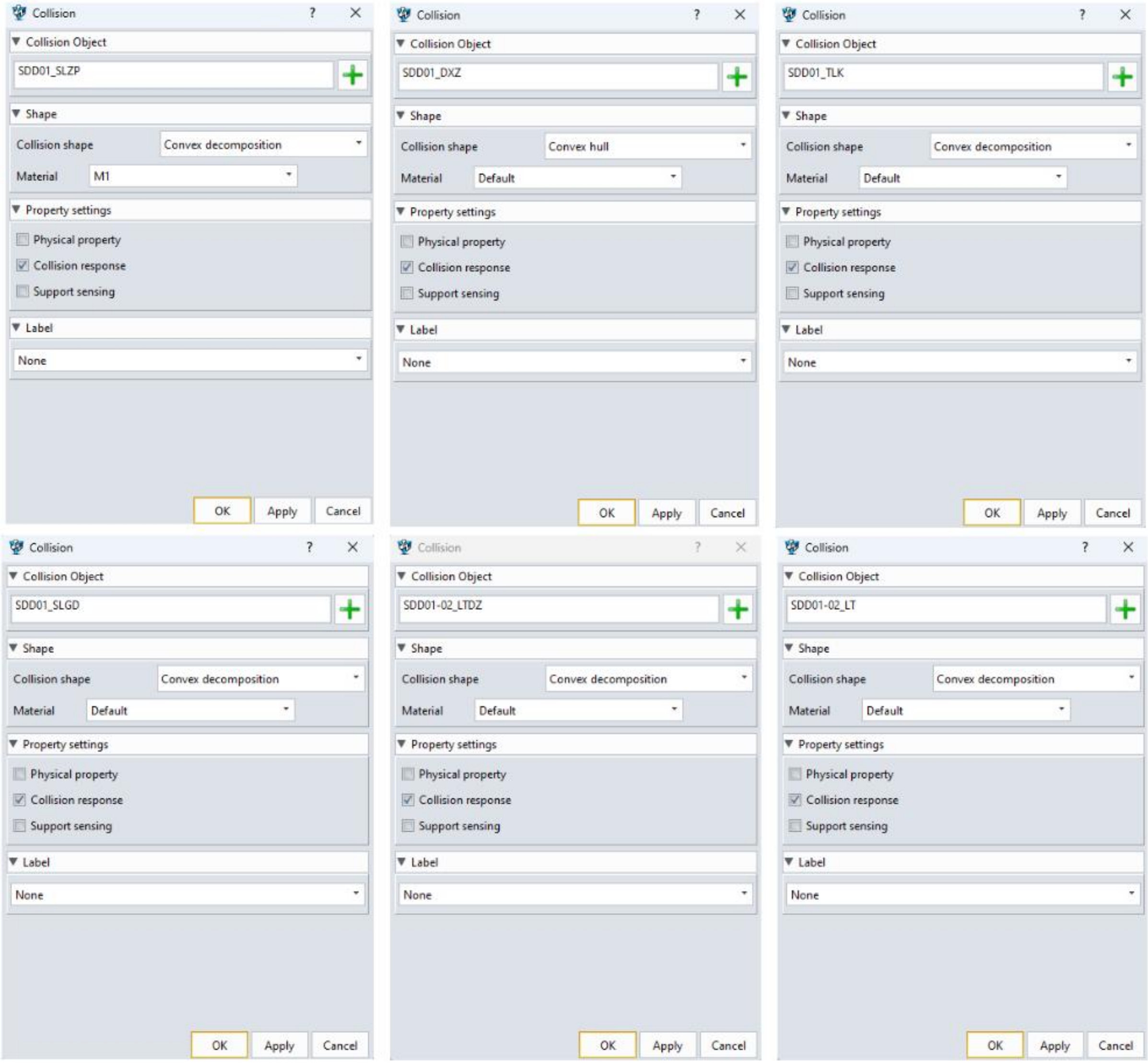



Figure 13-16：Collison message

## Step5: Kinematic Pair Definition

- Click the "Mechatronic" button.(Figure 13-17)
- Open the "Kinematic pairs and constraints" window and select "Add Revolute and Prismatic joints".(Figure 13-18)
- Create a revolute joint for the turntable base and specify the axis, start angle, limits, and motion type.(Figure 13-19)
- Create a prismatic joint (Cylinder_J1) for the cylinder and define the axis, offset, limits, and motion type.(Figure 13-20)

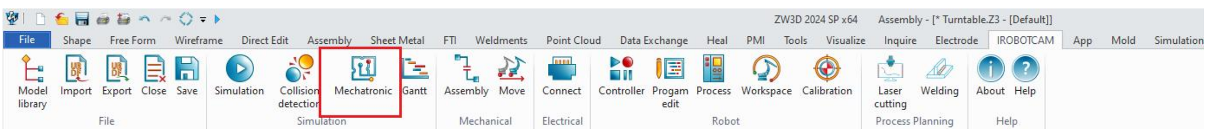



Figure 13-17：Mechatronic button

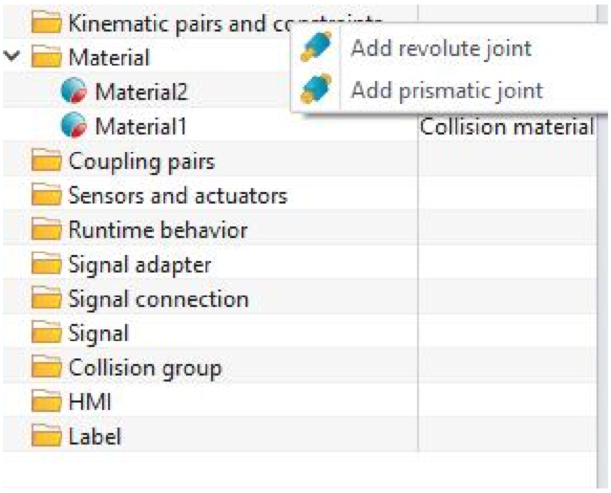



Figure 13-18：Add kinematic pairs

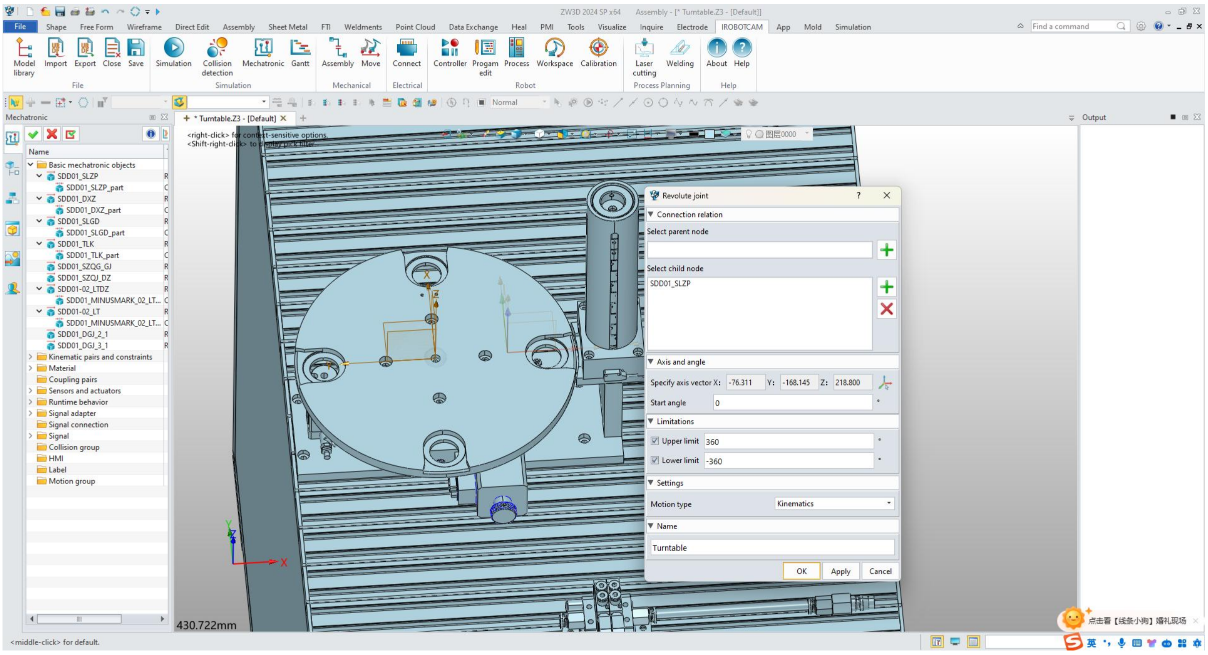


Figure 13-19：Revolute joint

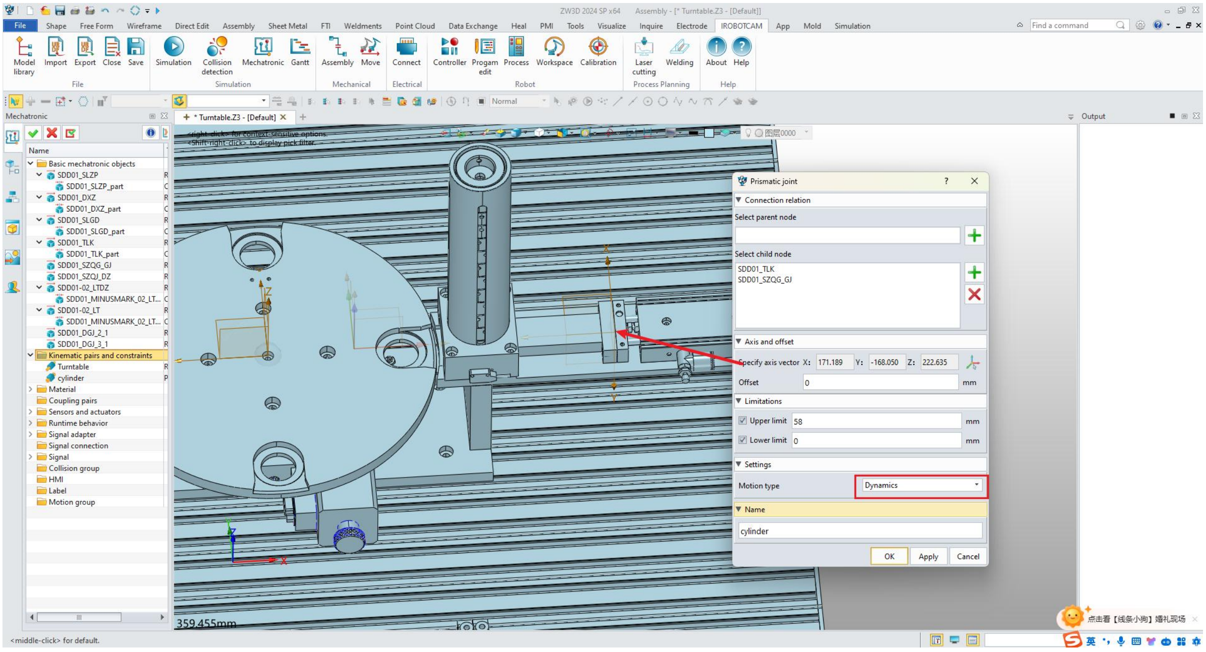



Figure 13-20：Prismatic joint

## Step6: Output Signal Definition

- Click the "Mechatronic" button.(Figure 13-21)
- Open the "Signal" window and select "Add Signal".(Figure 13-22)
- Create two output signals: "Turntable_Out" and "Cylinder_Out".(Figure 13-23)
- Specify "Out" as the IO type, assign names, and set port values.(Figure 13-23)


Figure 13-21：Mechatronic

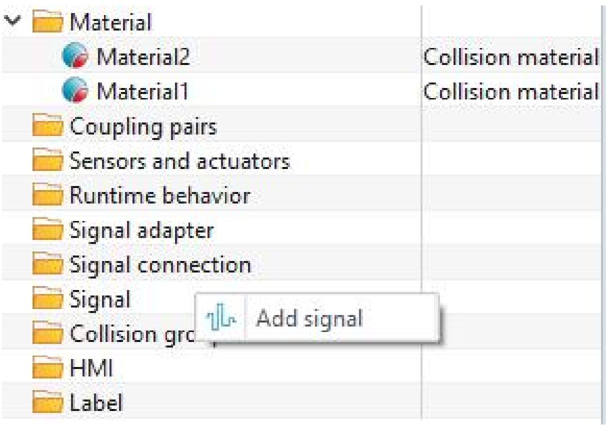



Figure 13-22：Add signal

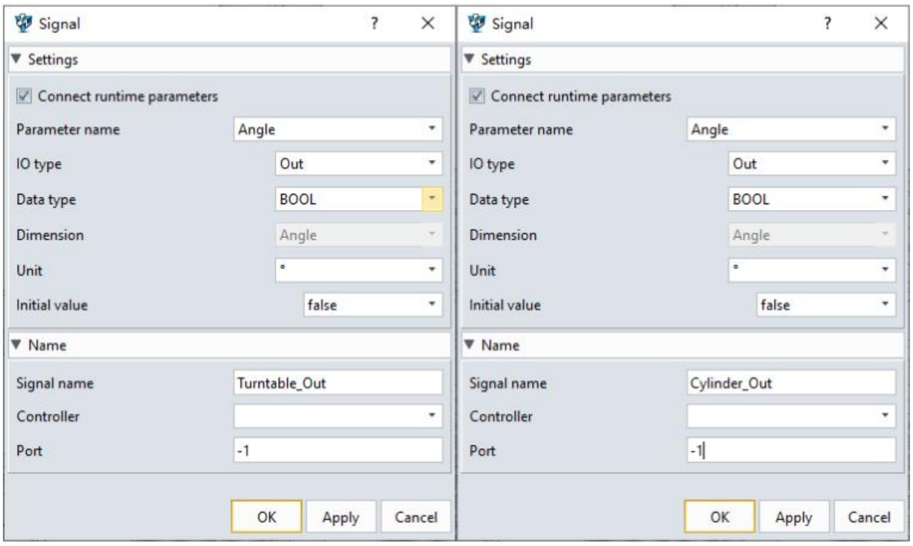



Figure 13-23：Output signal

## Step7: Position Control Implementation

- Click the "Mechatronic" button.(Figure 13-24)
- Open the "Sensors and actuators" window and select "Add Position Control".(Figure 13-25)
- Define position control for both "Turntable" and "Cylinder".(Figure 13-26/13-27)
- Associate the control with the corresponding kinematic pairs and rigid bodies.(Figure 13-26/13-27)
- Specify coordinate systems, signals, axis types, target positions, speeds, and acceleration limits for each control.(Figure 13-26/13-27)

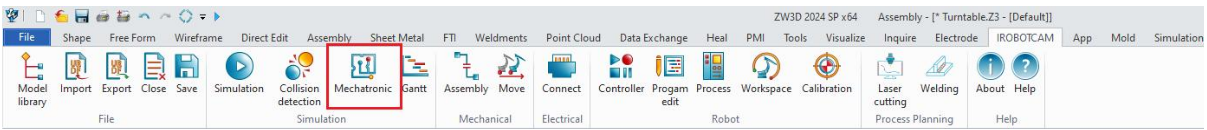



Figure 13-24：Mechatronic button

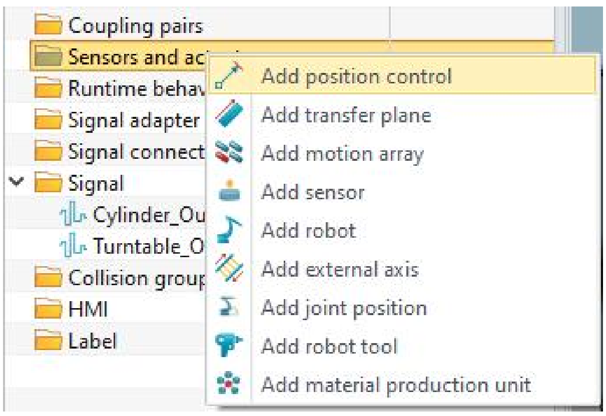



Figure 13-25：Add position control

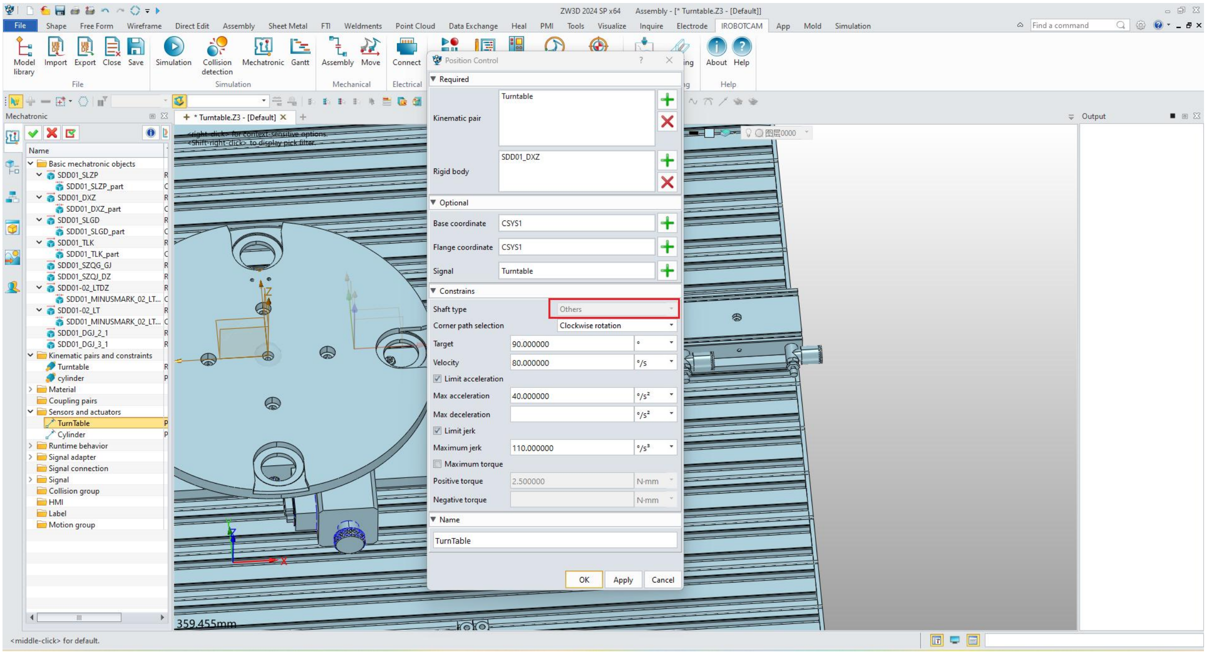


Figure 13-26：Turntable message

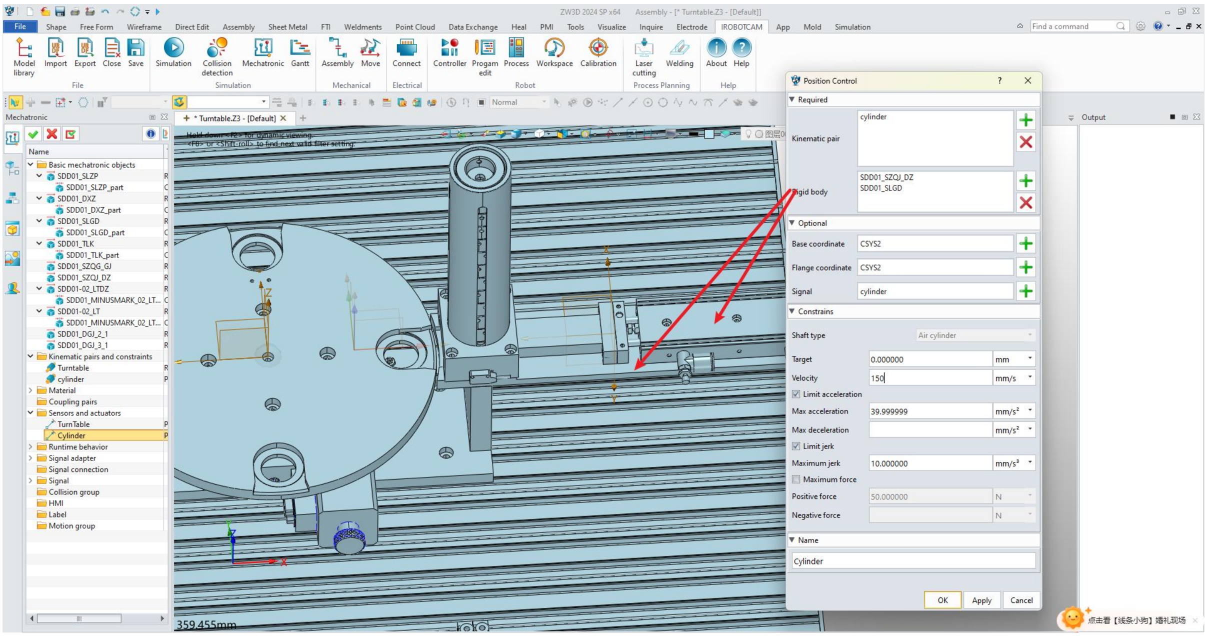


Figure 13-27：Cylinder

## Step 8: Signal Adapter Creation

- Click the "Mechatronic" button.(Figure 13-28)
- Open the "Signal Adapter" window and select "Add Signal Adapter".(Figure 13-29)
- Create two signal adapters: "Turntable_SD" and "cylinder_SD".(Figure 13-30)
- Assign the corresponding "Turntable" and "Cylinder" control signals.(Figure 13-30)

- Define data types, input/output direction, initial values, and formulas for each adapter.(Figure 13-30)

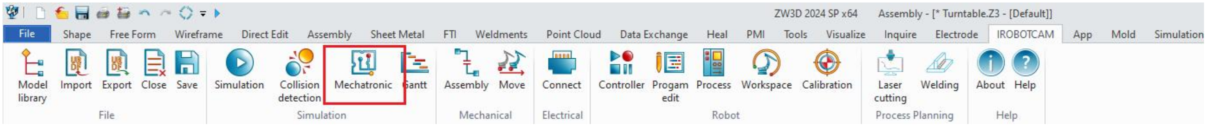



Figure 13-28：Mechatronic button

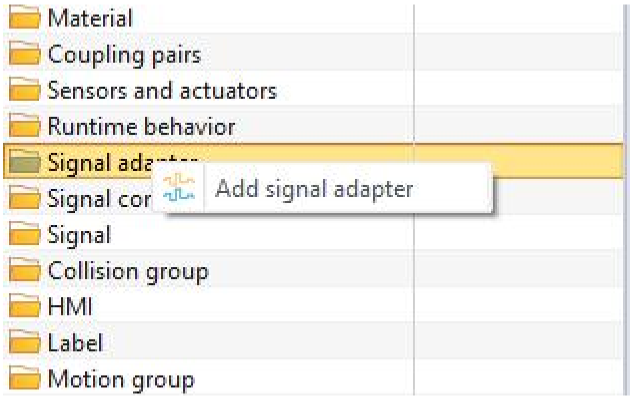



Figure 13-29：Add signal adapter

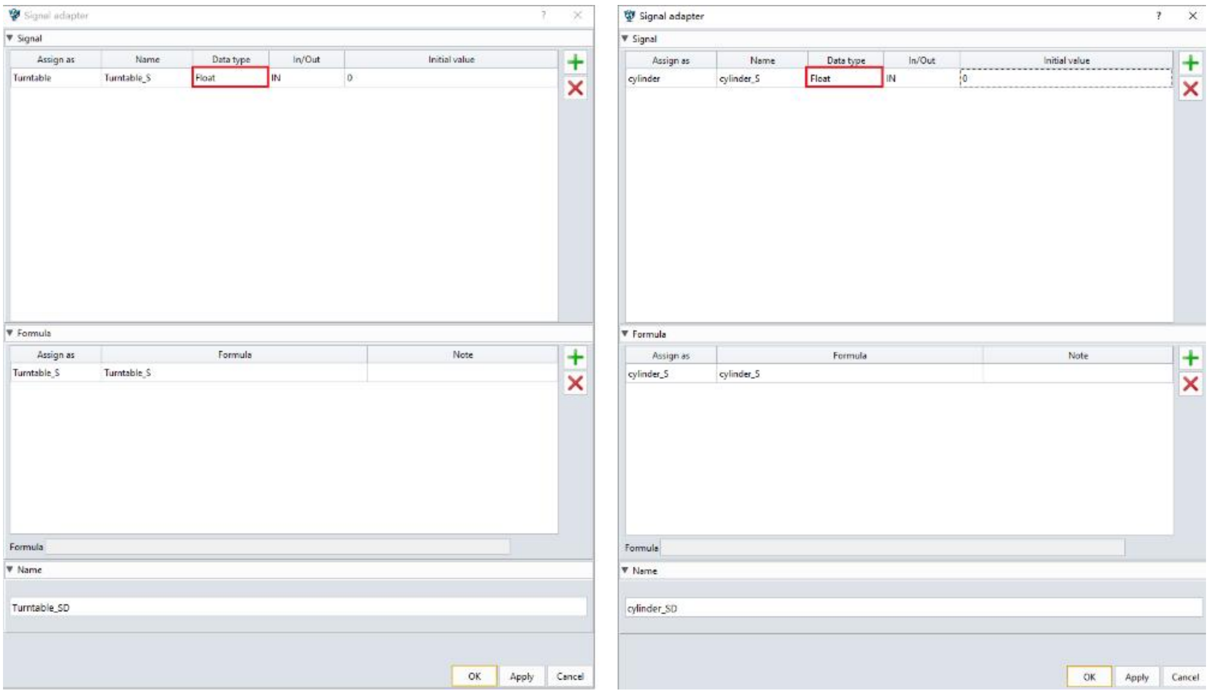



Figure 13-30：Two signal adapter

## Step 9: Input Signal Definition

- Click the "Mechatronic" button.(Figure 13-31)
- Open the "Signal" window and select "Add Signal".(Figure 13-32)

- Create input signals for both "Turntable" and "Cylinder".(Figure 13-33)
- Specify "In" as the IO type, assign names, and set port values.(Figure 13-34)

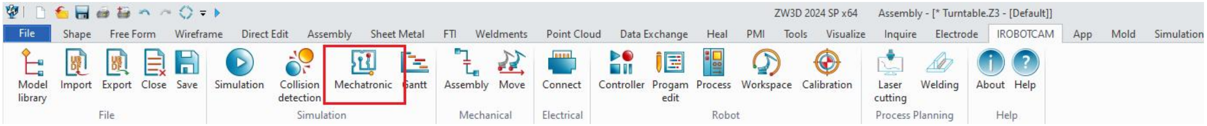



Figure 13-31：Mechatronic button

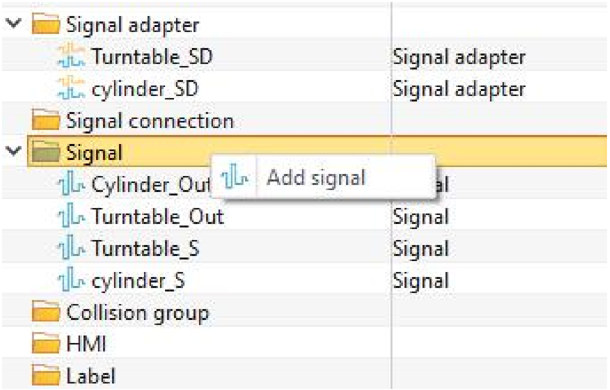



Figure 13-32：Add signal

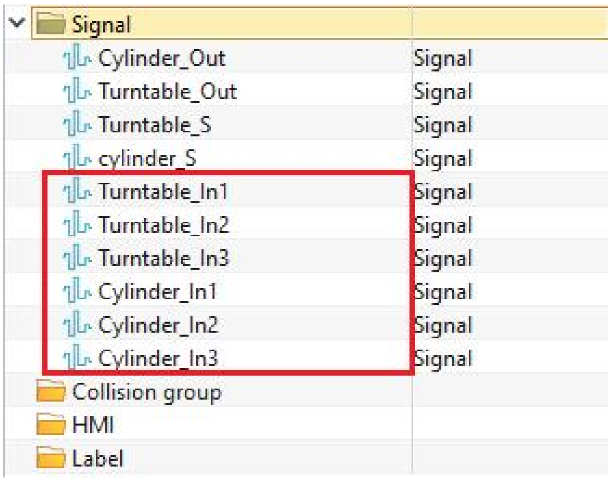



Figure 13-33：Six signal

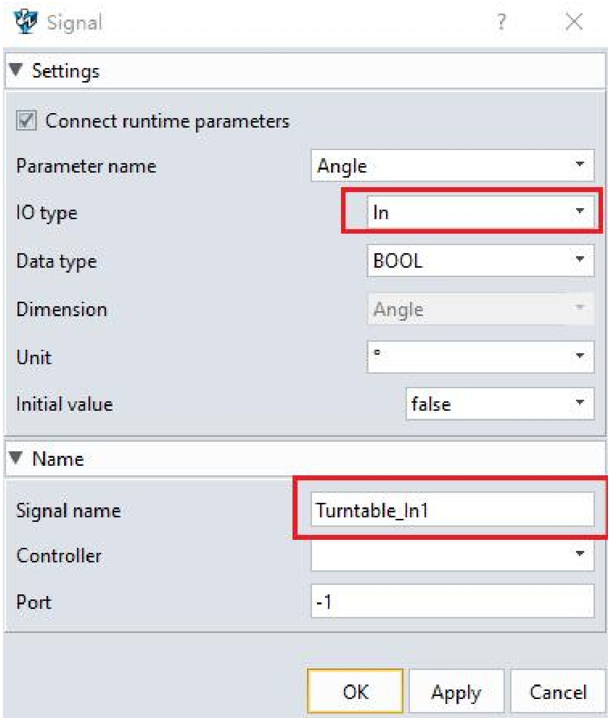



Figure 13-34：Signal message

## Step10: Simulation Sequence Design

- Click the "Mechatronic" button.(Figure 13-35)
- Open the "Runtime behavior" window and select "Add Simulation Sequence".
- Create three simulation sequences:(Figure 13-36)(Figure 13-37)

✧ "Turntable_EXEC1": activates the turntable to reach 90 degrees within 5 seconds based on specific input and output signals.

✧ "Cylinder_EXEC1": moves the cylinder to reach 58 units within 5 seconds based on the turntable's position.

✧ "Cylinder_EXEC2": activates the cylinder to reach 58 units within 5 seconds based on another input signal.

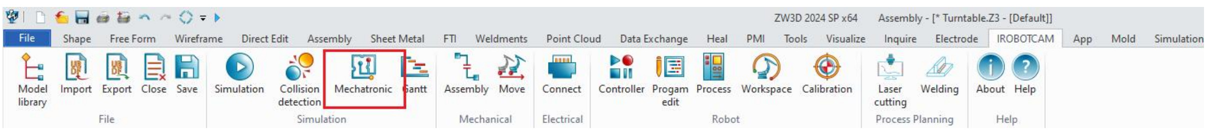



Figure 13-35：Mechatronic button

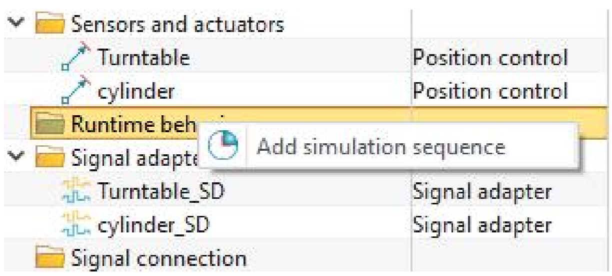


Figure 13-36：Add simulation sequencce

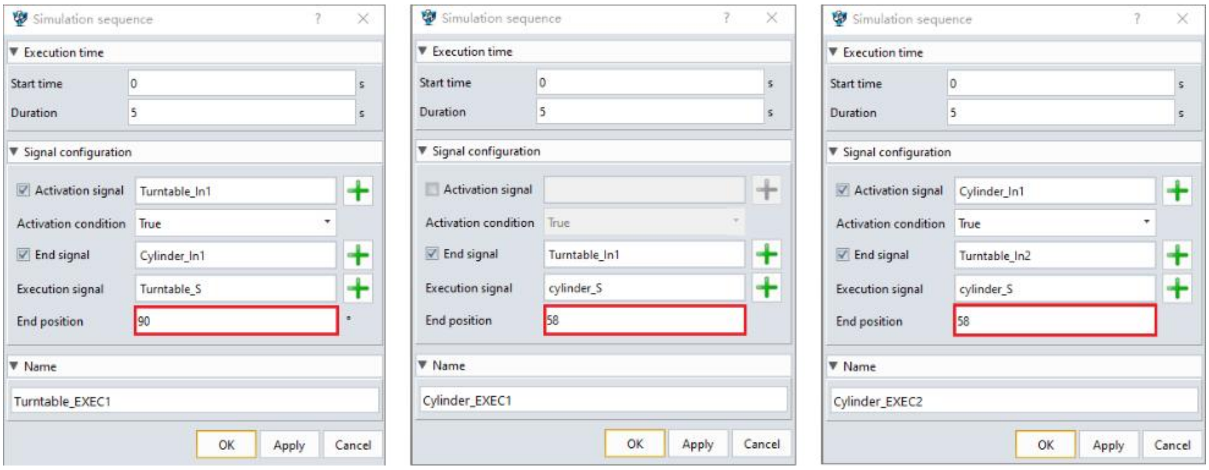



Figure 13-37：simulation sequencce message

## Step11: Simulation and Verification

- Click the "Simulation" button.(Figure 13-38)
- Click the "Start" button to initiate the turntable operation.(Figure 13-39)

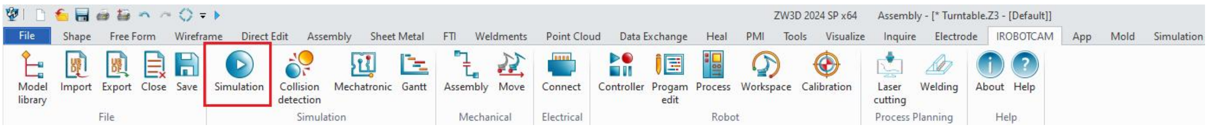


Figure 13-38：Simulation button

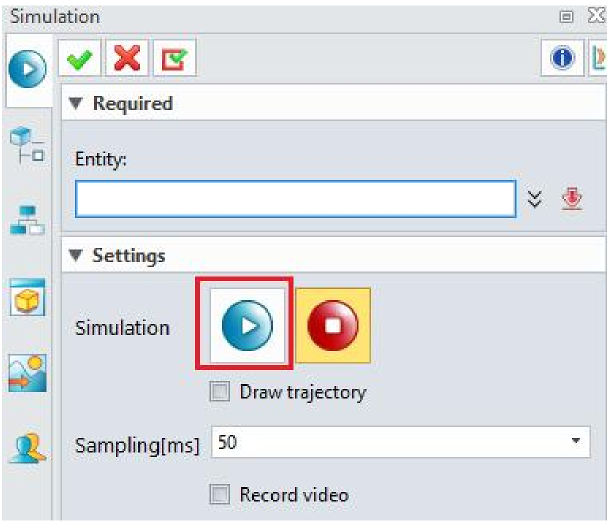



Figure 13-39：Start button

# Chapter 14: Laser Cutting

## Introduction

- Laser cutting is a technology that uses a laser beam to cut materials. It has the advantages of fast cutting speed, high precision, smooth cutting edge and small heat affected zone. It is widely used in sheet metal processing, electronic components, mechanical manufacturing, automobile manufacturing, aerospace and other fields.

- The software provides laser cutting simulation function, users can simulate the cutting process, view the cutting effect.

This guide provides step-by-step instructions for laser cutting in IROBOTCAM software.

## Step 1: Open the iRobotCAM project file

- Open the iRobotCAM project file named "Laser Cutting.Z3" to enter the IROBOTCAM environment.

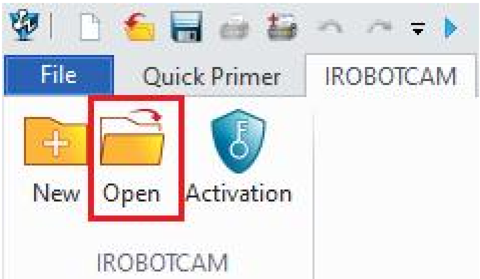



**Figure 14-1: Open button**

- The interface after opening is shown in the figure.

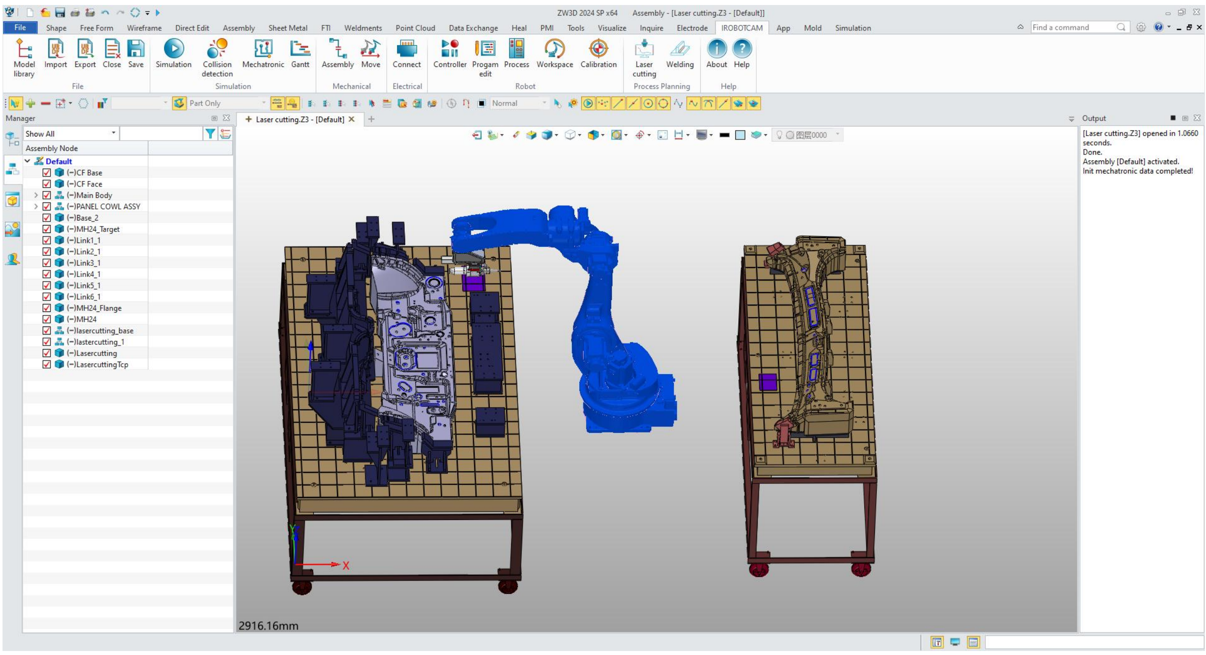


**Figure 14-2: Interface details**

## Step 2: Add collision detection

- Click "IROBOTCAM" to enter the mechatronic interface.

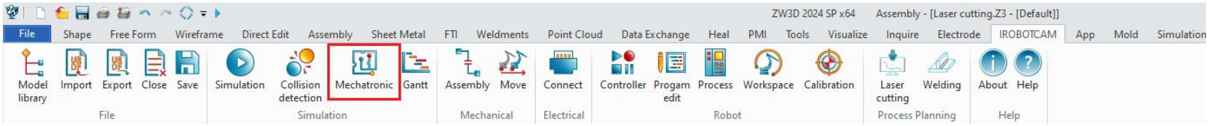



**Figure 14-3: mechatronic button**

- In the collision group window, select the collision body of the workpiece to create a collision group.
- The support induction checkbox needs to be selected in the collision body window of the workpiece.

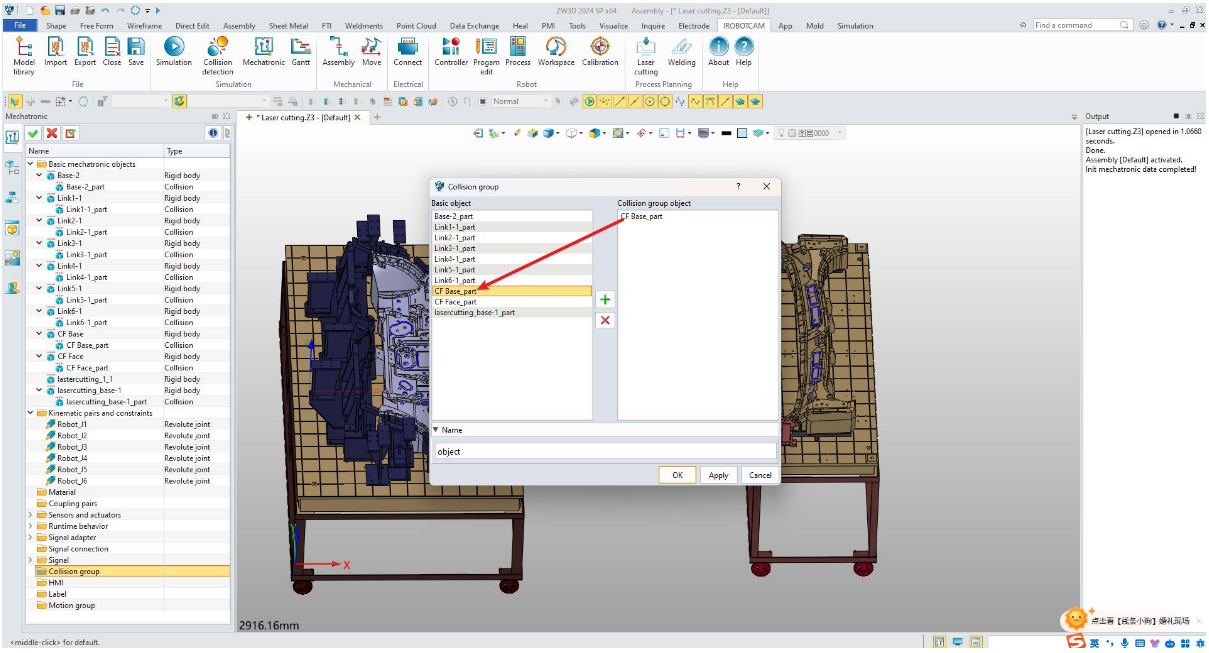



**Figure 14-4: Workpiece collision group**

- In the collision group window, select the collision body of the robot and the collision body of the robot tool to create a collision group.
- The support induction checkbox needs to be selected in the collision body window of the workpiece and the tool.

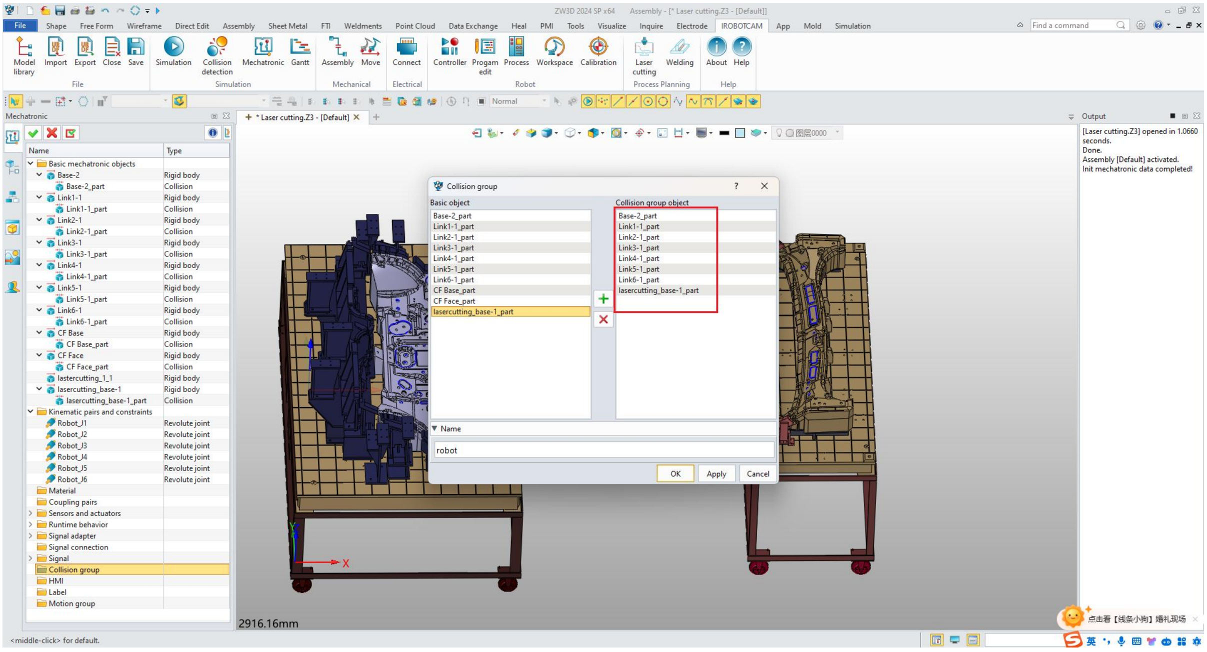


**Figure 14-5: Robot and robot tool collision group**

- Click the collision detection button.
- Select "robot" as the target object. Select "object" as the obstacle.
- Click the "+" button to add.

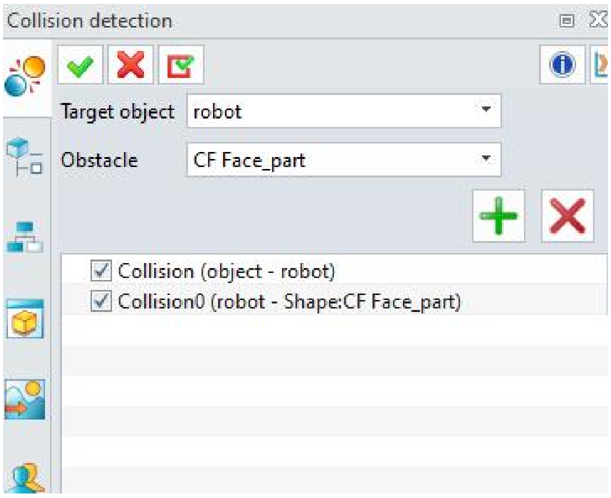



**Figure 14-6: Collision detection**

## Step 3: Add curve list

- Double-click to activate the component to be cut.
- Click the wireframe button to enter the curve list window.
- Select the edge to be cut to create a curve list.

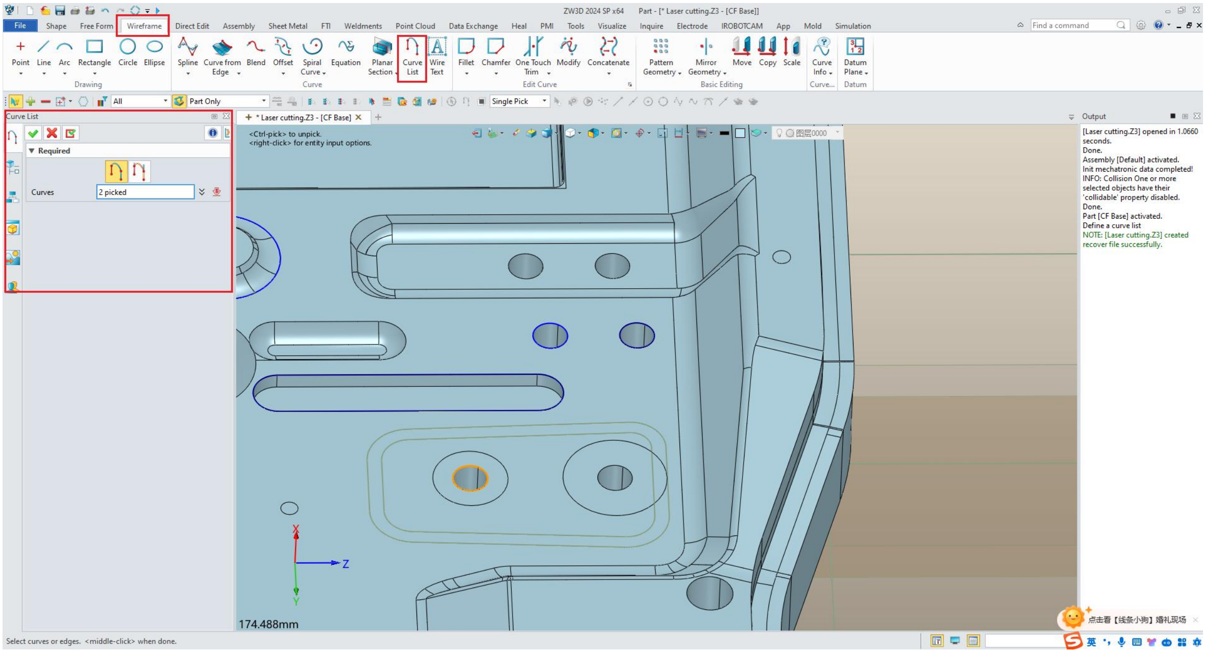


Figure 14-7: Curve list

## Step 4: Laser cutting - Select path

- Click the laser cutting button.

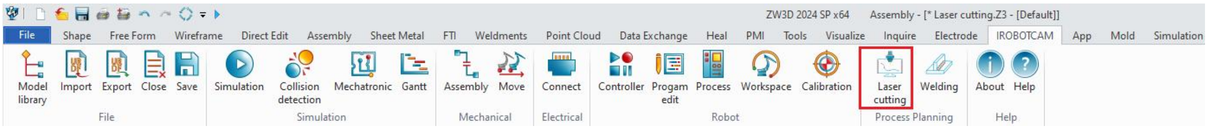



Figure 14-8: Laser cutting

- In the "Select Path" window, click the Select Part button to select the workpiece to be cut.
- Drop down to select the robot controller.
- In the Select Path bar, first add all the curve lists of the edges to be cut by clicking the Add button. (Note that clicking the Add button once can only add one curve list)

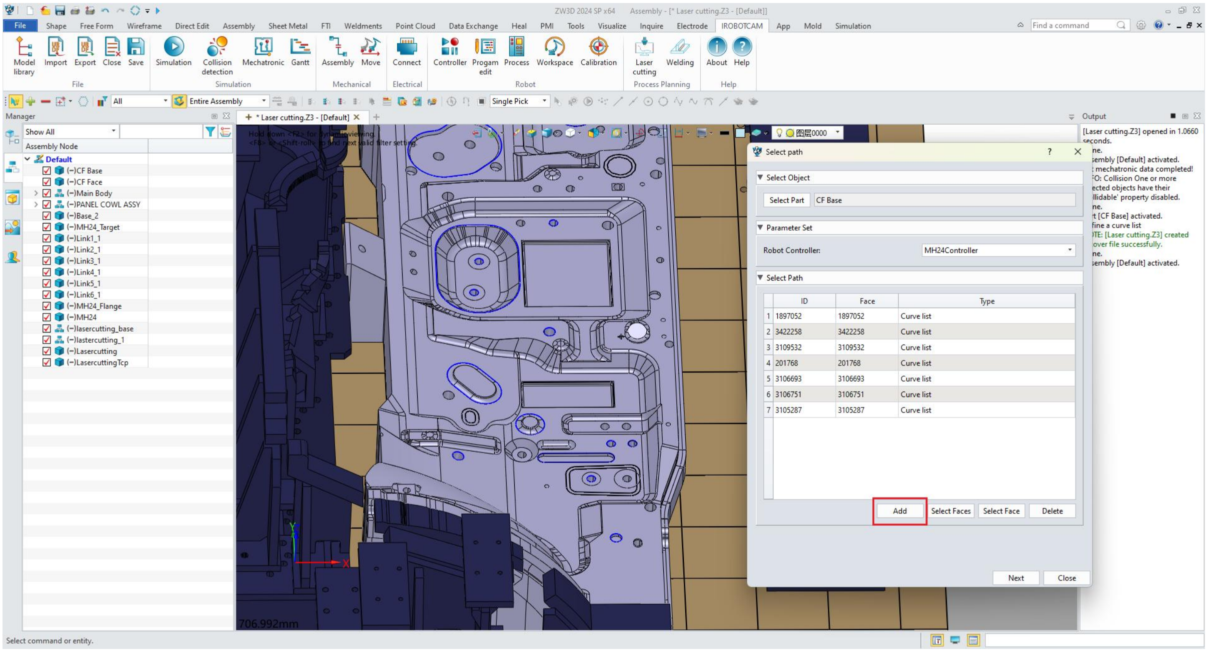



**Figure 14-9: Select path - Add curve list**

- Add all the faces where the cutting edge curve list is located.
- After adding all the curve lists, click all the faces where the curve lists are located, and select all of them by pressing the CTRL key.
- Then click the Select Multiple Faces button. After adding, if the output box does not prompt an error, the addition is successful.

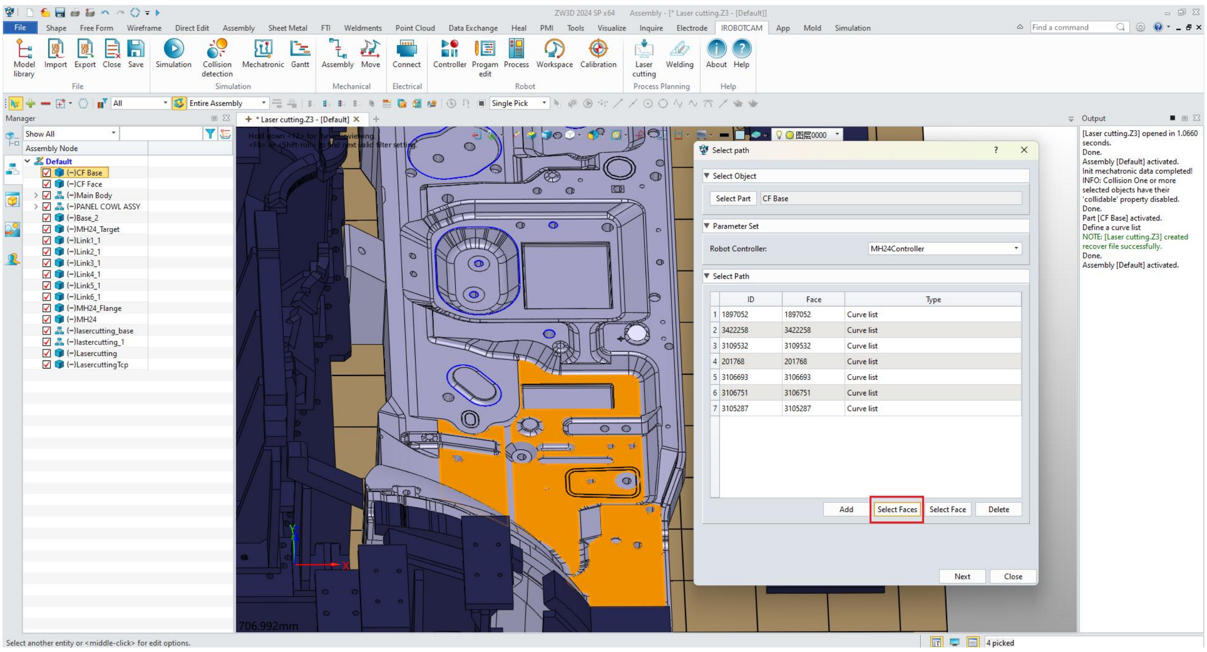



**Figure 14-10: Select multiple faces**

## Step 5: Path Discretization

- Click "Next".
- The default value for the chord height ratio.
- Click the Discretization button to generate discrete points in the Discretization window.
- Click Show Discretization Result to view the coordinate system direction of the discrete points.

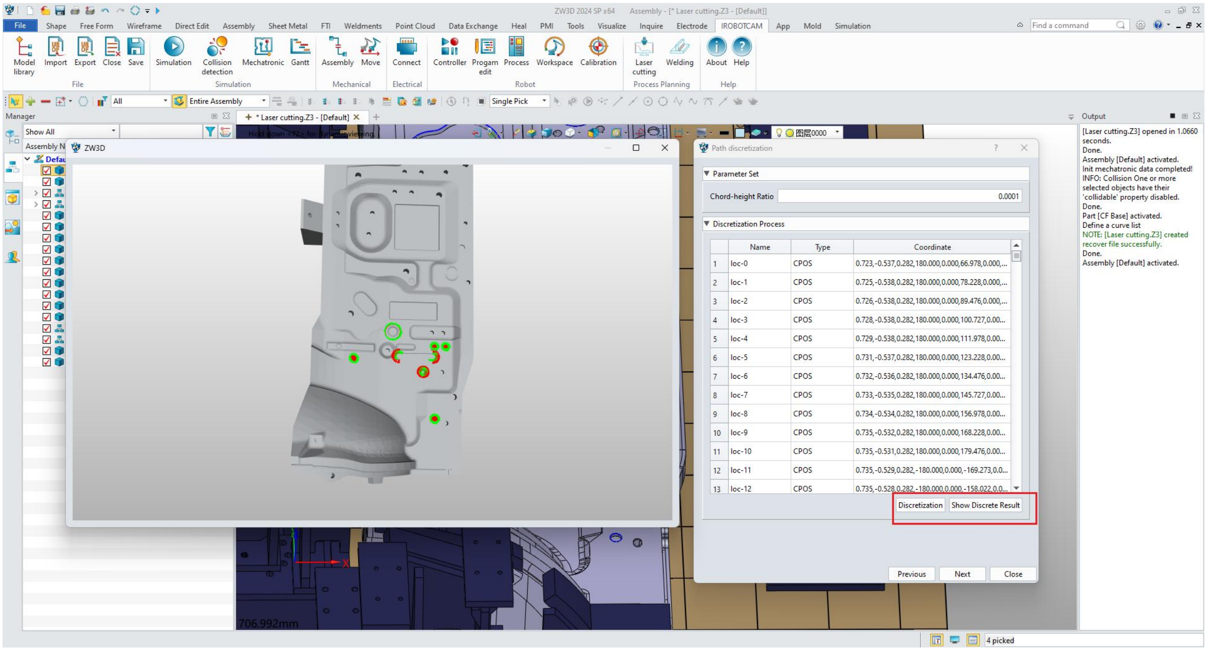



**Figure 14-11: Path discretization**

### Step 6: Transition Point Sorting

- Click "Next".
- Click the "process sort calculate" button.
- View the laser cutting order in Process sorting calculation chart.

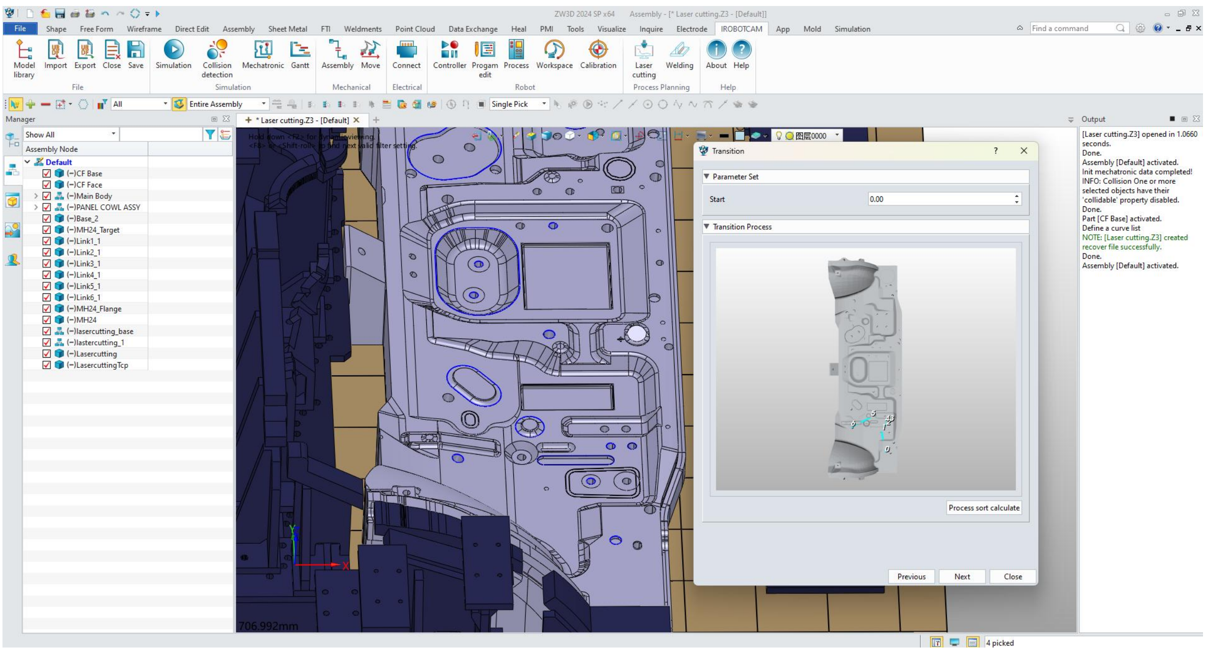


Figure 14-12: Transition point sorting

## Step 7: Path Optimization

- Click "Next".
- In the Raise Gun Height bar, enter the raise gun height value.
- Click the "edit program" button to view the coordinate position of the raise gun height after discretization.
- Click the Calculate button to automatically list the three detection tables of Reachable, Singular and Collision.
- Finally, click the Optimize Path button. In the Optimize Path window, click the "Change fill data" and "Path optimize" buttons in sequence.
- In the Optimize Path window, click Modify Fill Data. Red represents the collision detection.

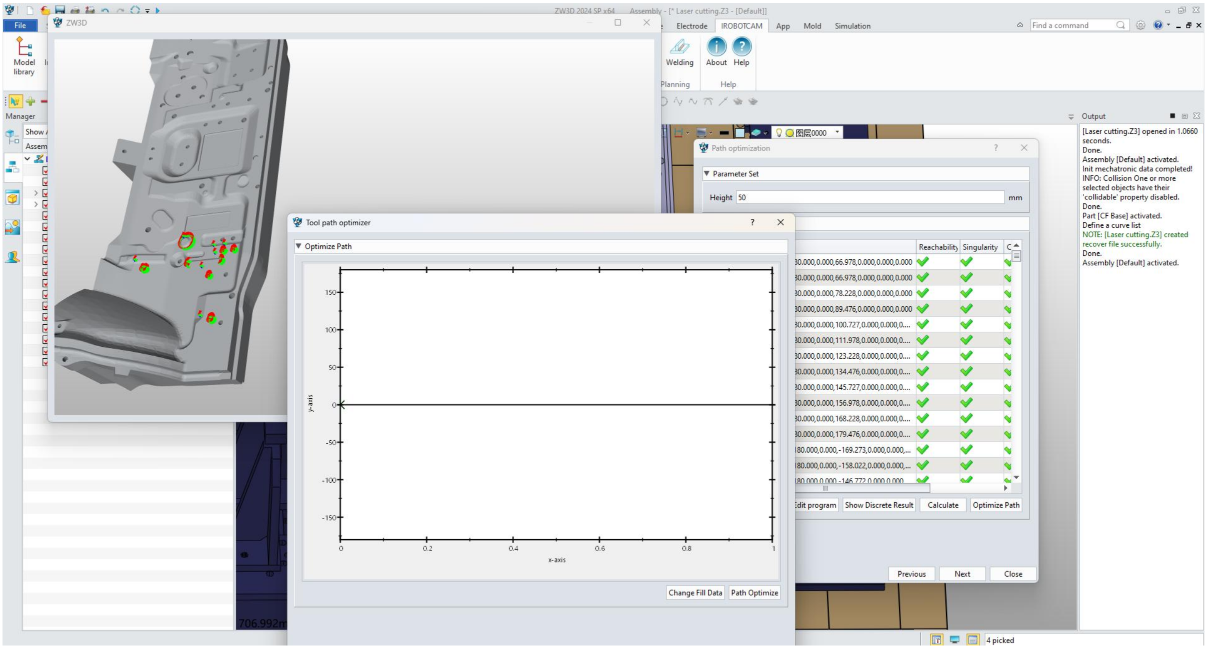
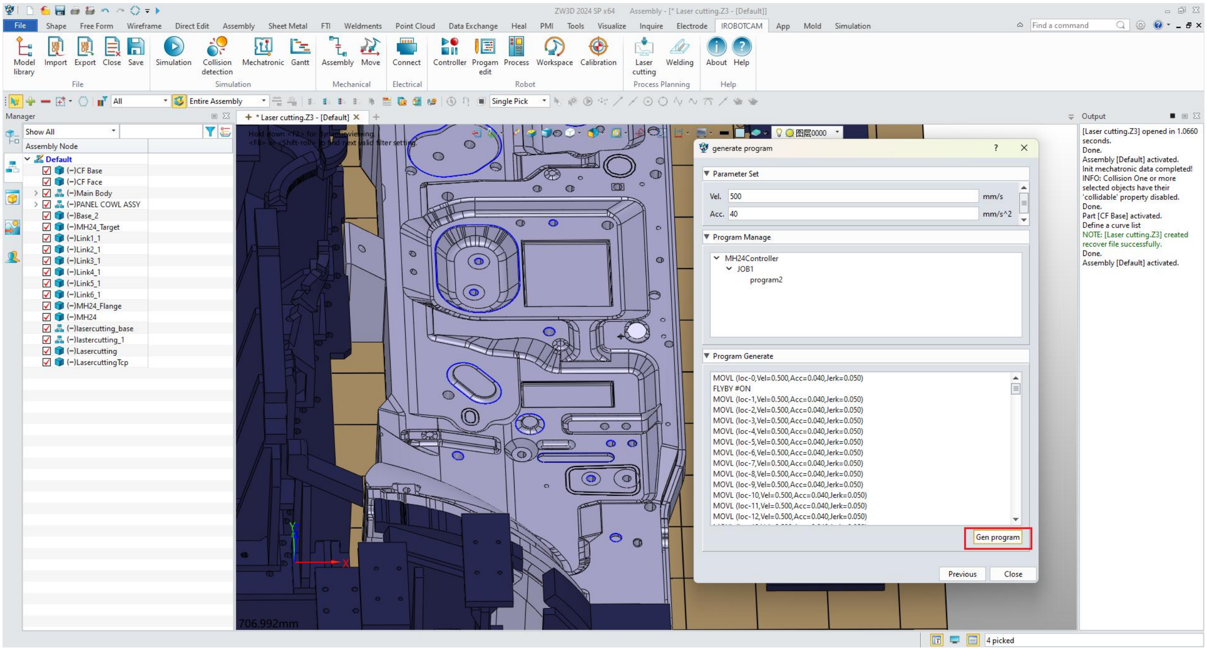



Figure 14-13: Path optimization

## Step 8: Generate program

- Click "Next".
- The default values for speed and acceleration are set in the Generate Program window.
- Under the Program Management bar, add JOB and Program programs under Program Editing.
- In the Program Generation bar, click the "Generate program" button.




## Step 9: Simulation verification

- If the program has added a simulation sequence, you can click the simulation button to verify it. If not, you can click Run in the program editing window to view the laser cutting effect.
- During laser cutting, if a collision is detected, the target object will be displayed in a highlighted manner.

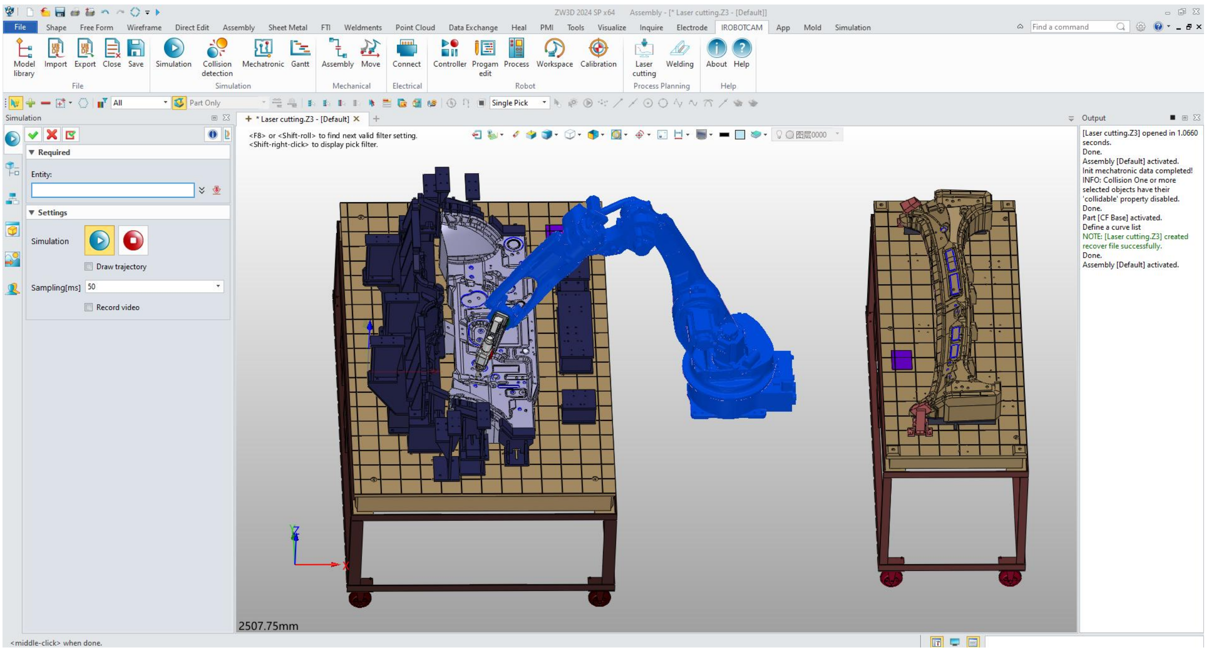


**Figure 14-14: Simulation verification**

The software provides two cutting workpieces in the scene, users can try the laser cutting function by themselves.

# Chapter 15: Welding

## Introduction

Welding is a forming method that uses heating or pressurizing, or both, to cause two separate objects to join together by interatomic bonding. Welding is an essential process in manufacturing and is widely used in mechanical manufacturing, construction, ships, aerospace, electronics and other fields.

The software provides welding simulation function, users can simulate the welding process, view the welding effect.

This guide provides step-by-step instructions for welding in IROBOTCAM software.

## Step 1: Open the iRobotCAM project file

- Open the iRobotCAM project file named "Welding.Z3" to enter the IROBOTCAM environment.

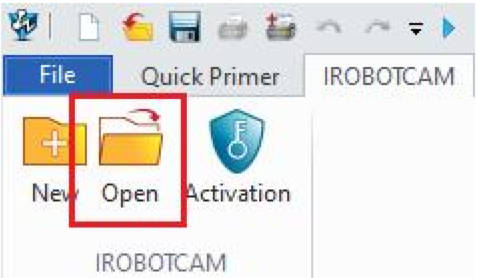



**Figure 15-1: Open button**

- The interface after opening is shown in the figure.

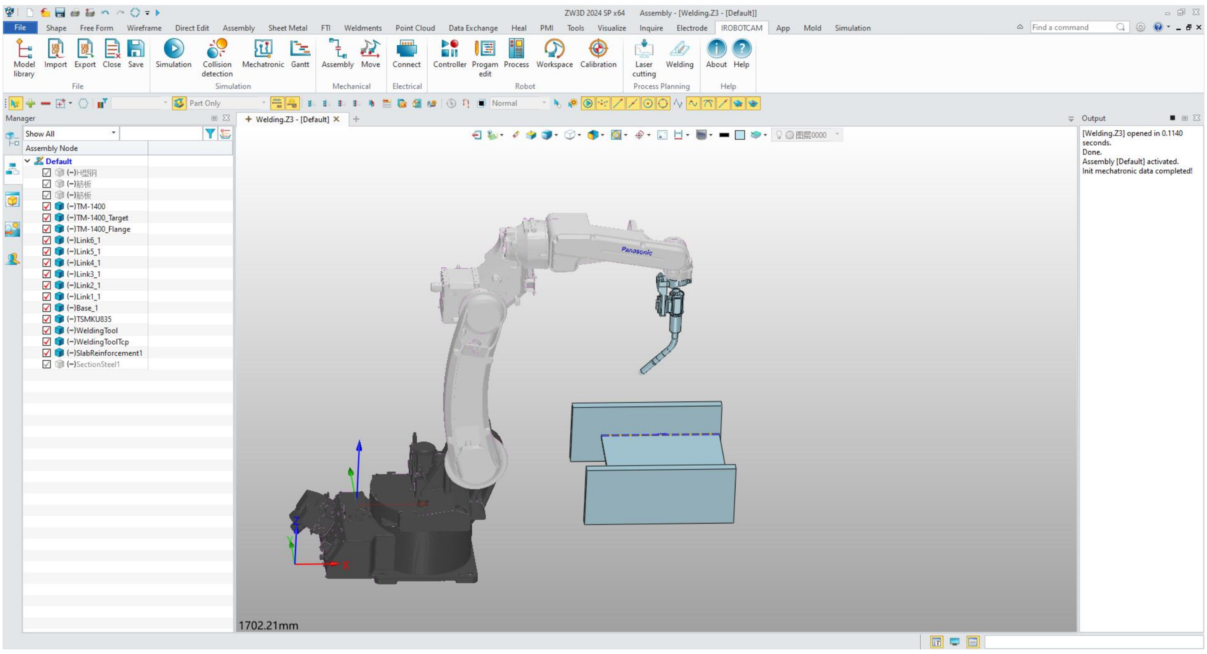


**Figure 15-2: Interface details**

## Step 2: Add collision detection

- Click "IROBOTCAM" to enter the mechatronic interface.

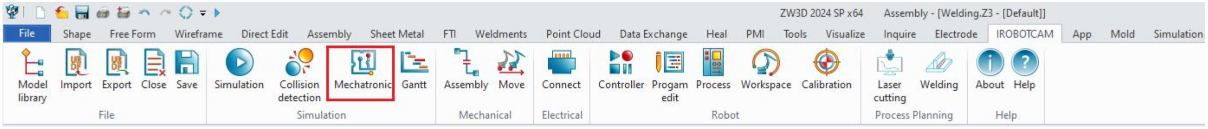



**Figure 15-3: mechatronic**

- In the collision group window, select the collision body of the workpiece to create a collision group. The support induction checkbox needs to be selected in the collision body window of the workpiece.

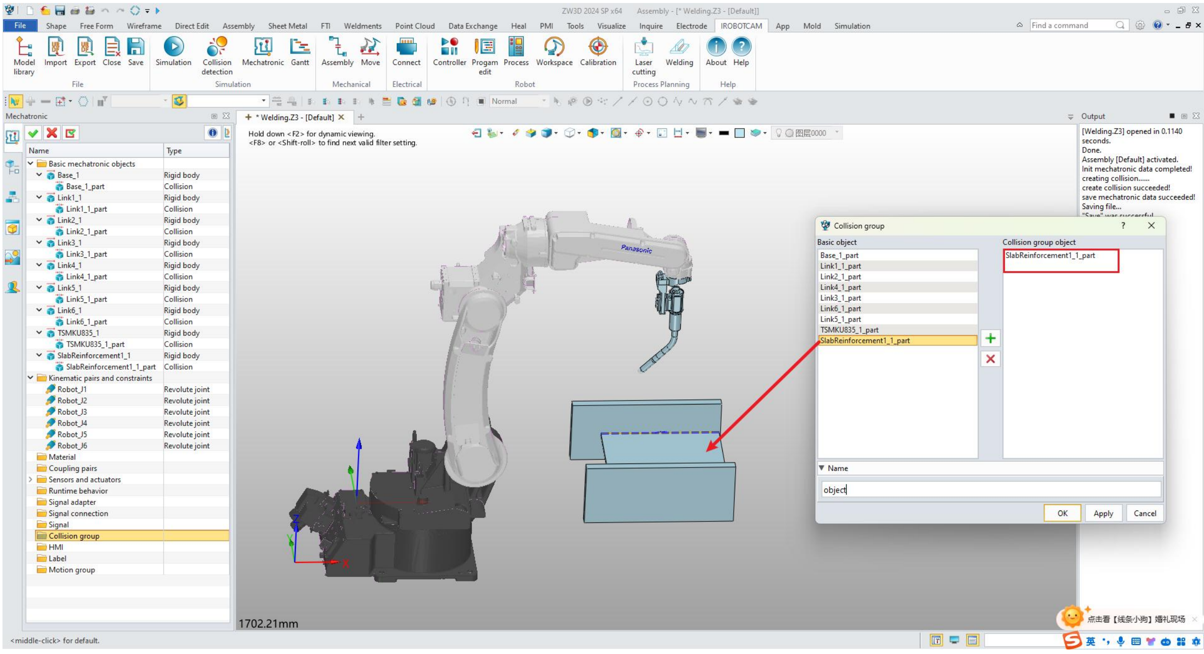



**Figure 15-4: Workpiece collision group**

- In the collision group window, select the collision body of the robot and the collision body of the robot tool to create a collision group. The support induction checkbox needs to be selected in the collision body window of the workpiece and the tool.

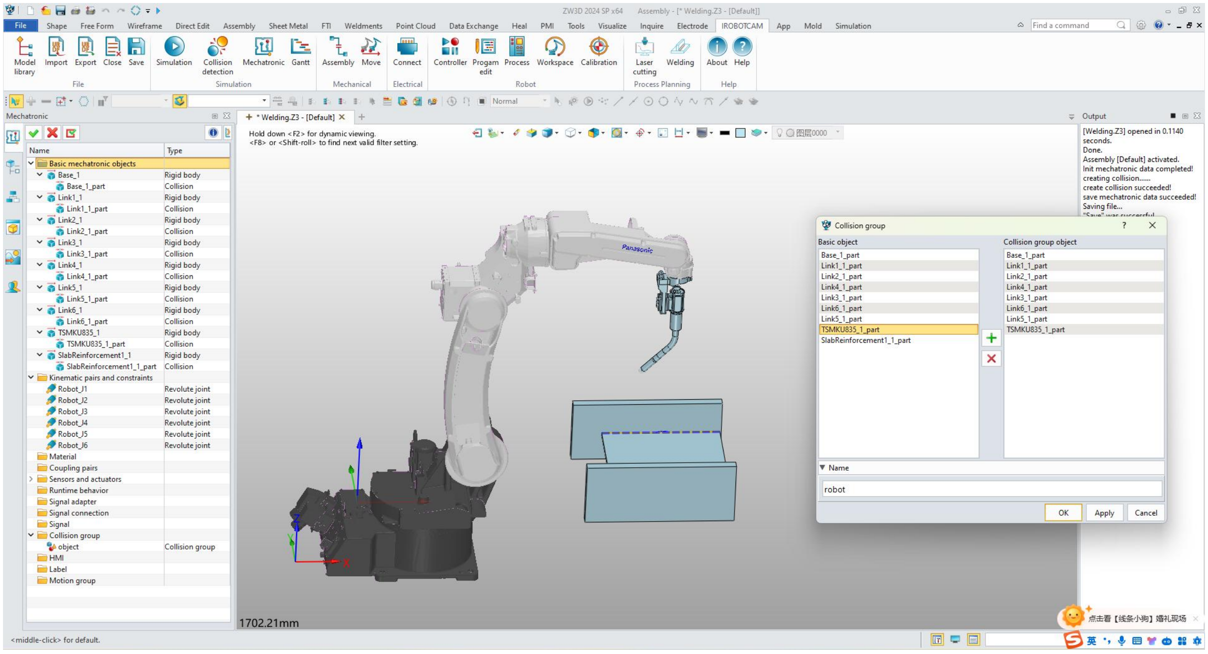


Figure 15-5: Robot and robot tool collision group

- Click the collision detection button.
- Select "robot" as the target object. Select "object" as the obstacle.
- Click the "+" button to add.

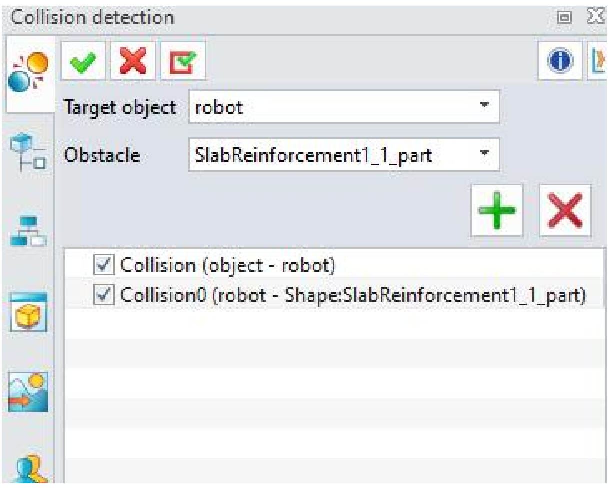



Figure 15-6: Collision detection

## Step 3: Add weldments

- Click the weld button.
- Click Fillet Welding. In the Fillet Weld bar, select two faces respectively and enter the weld leg width.
- The weld function can be used under a specific ZW3D authorization code.

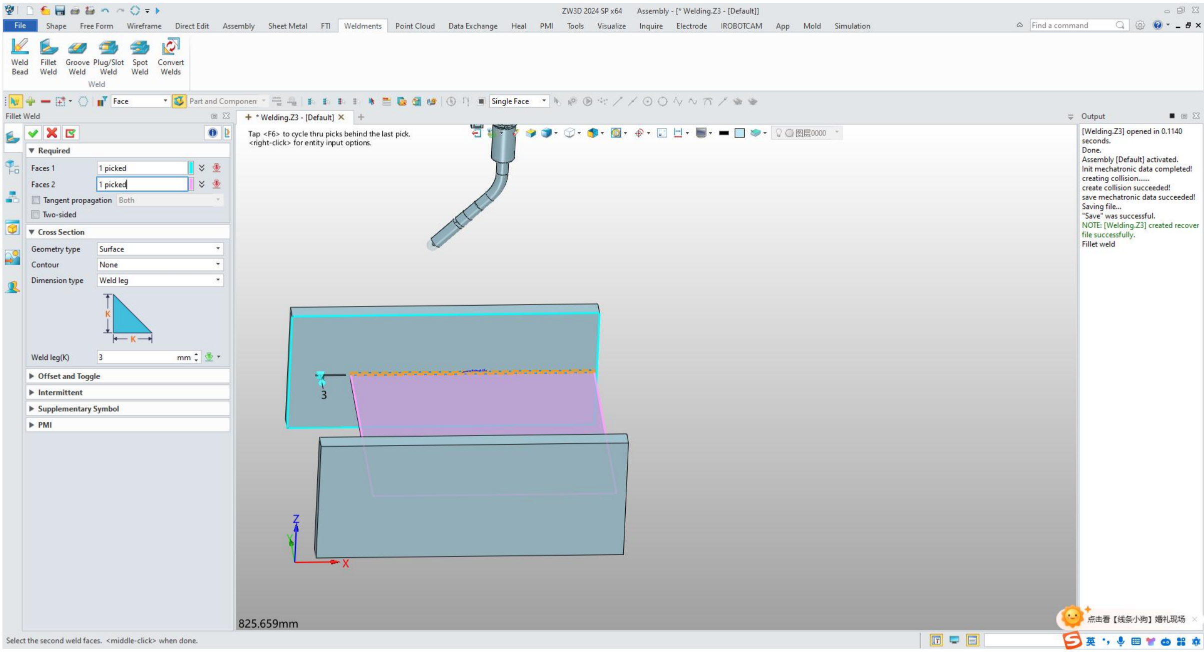


Figure 15-7: Add weld

## Step 4: Welding - Select path

- Click the welding button.

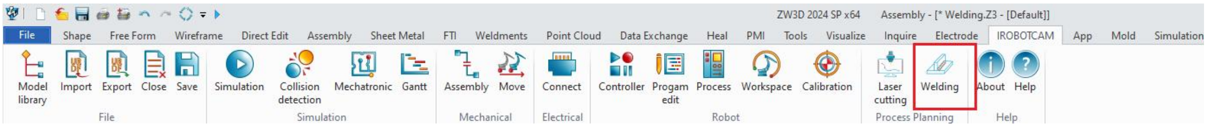



Figure 15-8: Welding

- In the "Select Path" window, click the Select Part button to select the workpiece to be welded.
- Drop down to select the robot controller.
- In the Select Path bar, first add the generated weld face by clicking the Add button.

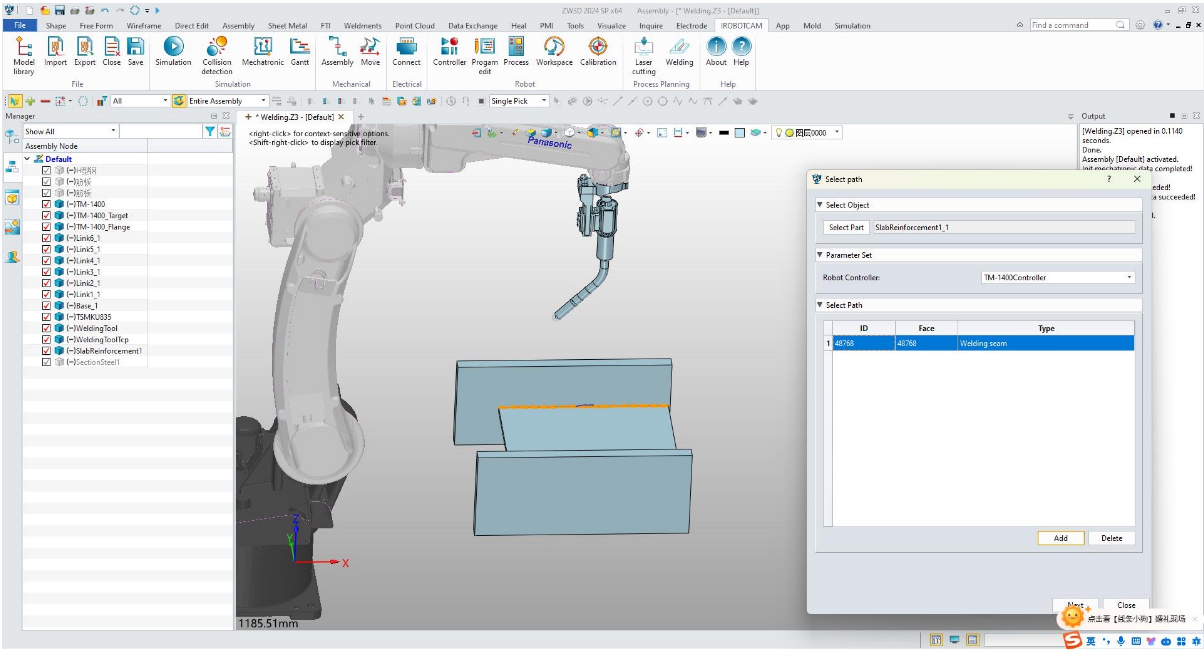


Figure 15-9: Select path

## Step 5: Path Discretization

- Click "Next".
- The default value for the chord height ratio.
- Click the Discretization button to generate discrete points in the Discretization window.
- Click Show Discretization Result to view the coordinate system direction of the discrete points.

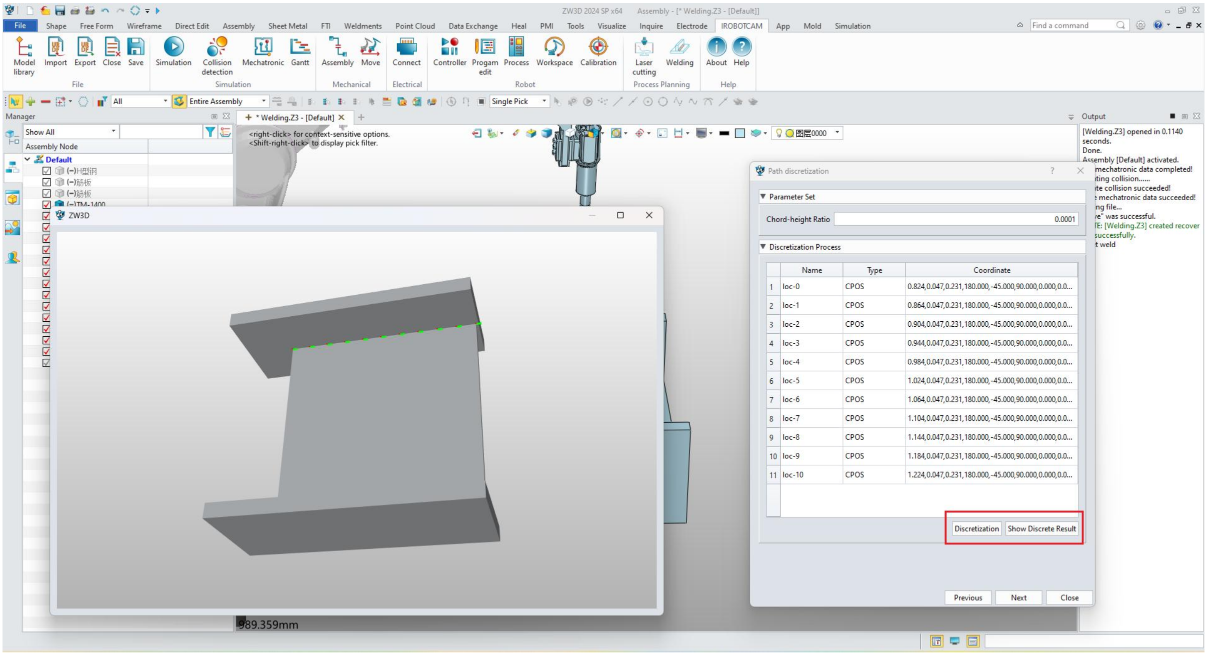


Figure 15-10: Path discretization

## Step 6: Transition Point Sorting

- Click "Next".
- Click the "process sort calculate" button.
- View the welding order in the process sort calculate chart.

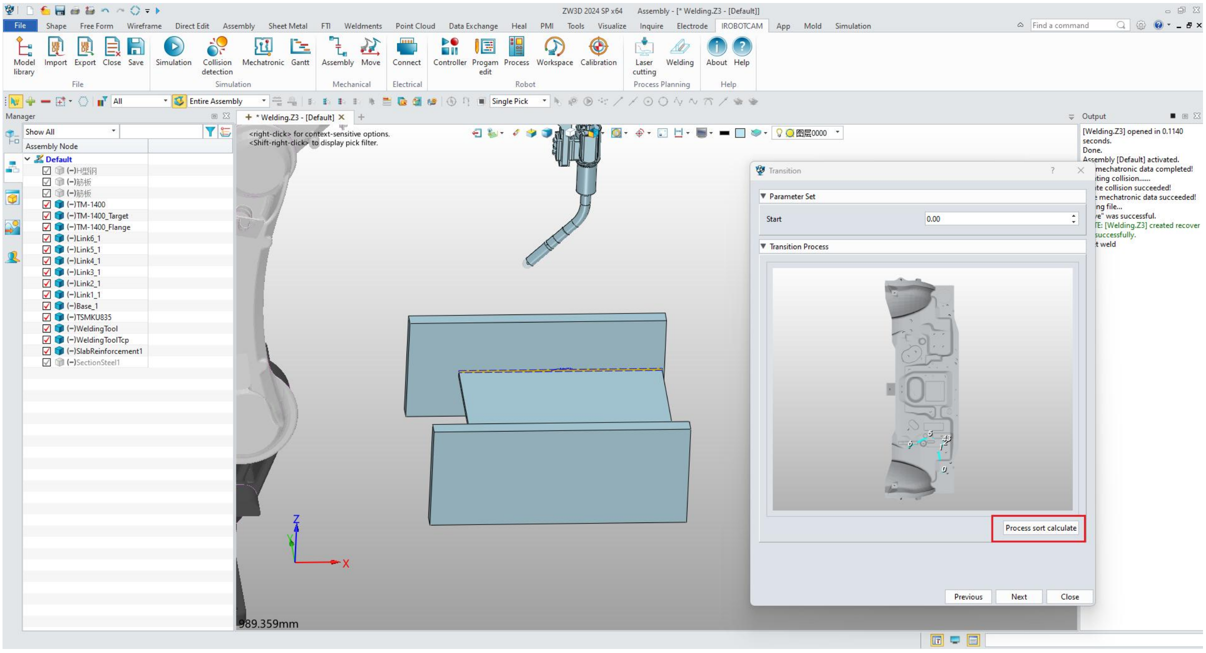


**Figure 15-11:process sort calculate**

### Step 7: Path Optimization

- Click "Next".
- In the Raise Gun Height bar, enter the raise gun height value.
- Click the "edit program" button to view the coordinate position of the raise gun height after discretization.
- Click the Calculate button to automatically list the three detection tables of Reachable, Singular and Collision.
- Finally, click the Optimize Path button. In the Optimize Path window, click the "Change fill data" and "Path optimize" buttons in sequence. (It takes about 1 minute to load the modified filling data)
- In the Optimize Path window, click Modify Fill Data. Red represents the collision detection.

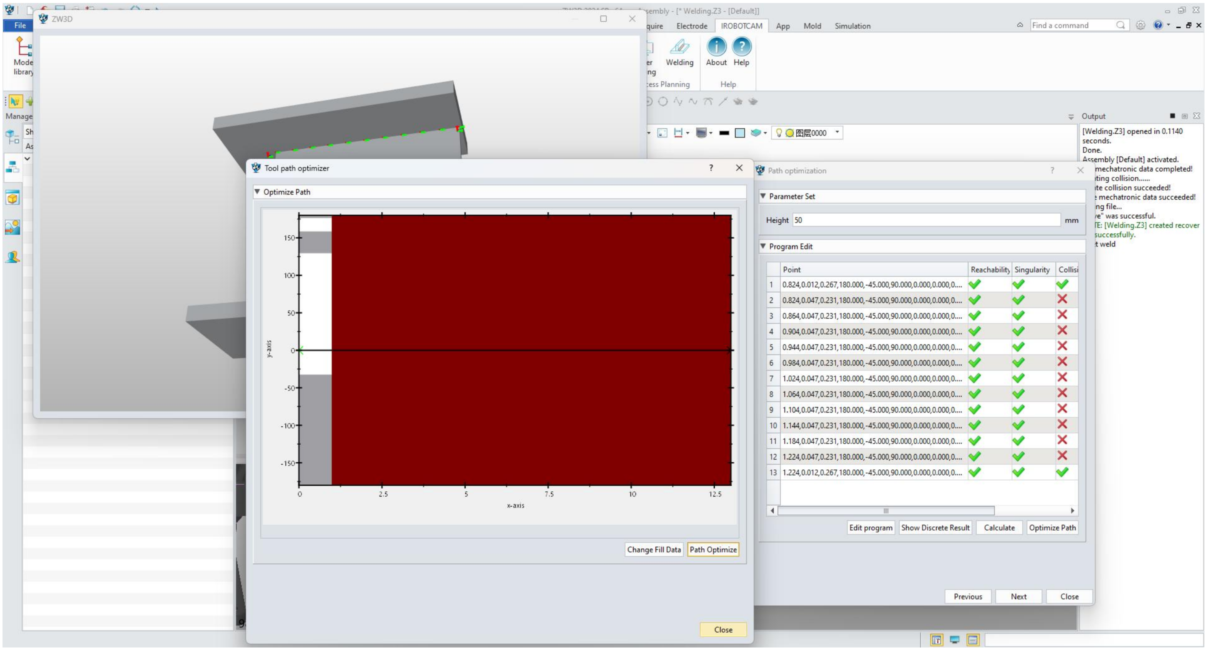


**Figure 15-12: Path optimization**

## Step 8: Generate program

- Click "Next".
- The default values for speed and acceleration are set in the Generate Program window.
- Under the Program Management bar, add JOB and Program programs under Program Editing.
- In the Program Generation bar, click the "generate program" button.
- Click the Close button.

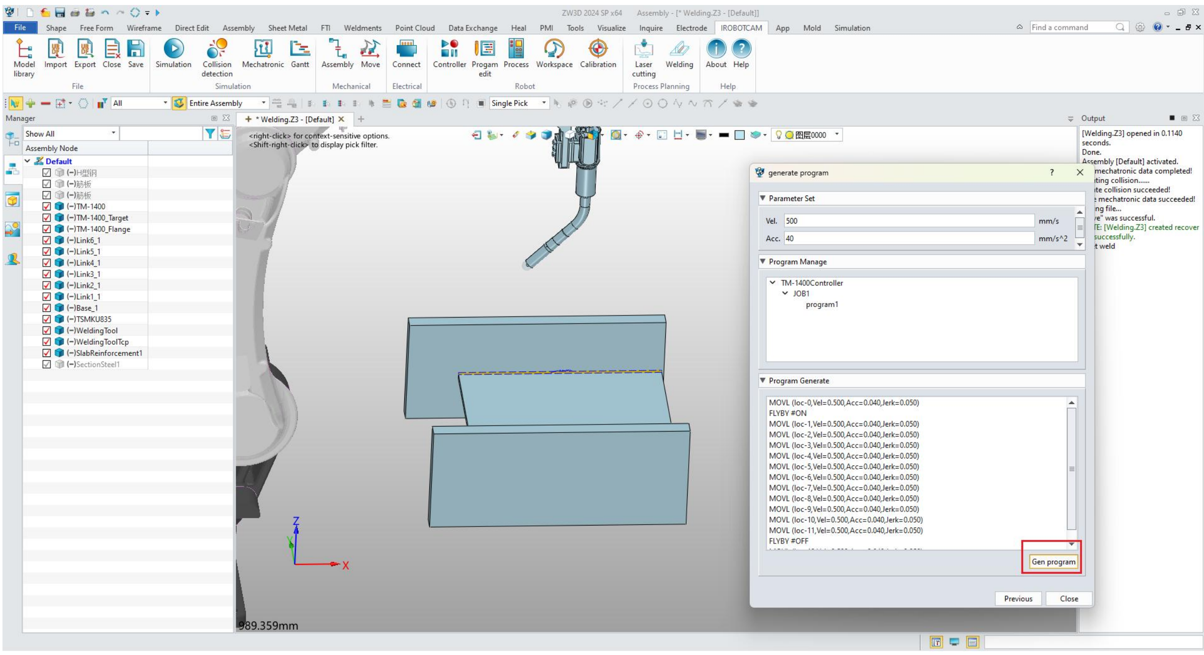


**Figure 15-13: Generate program**

## Step 9: Simulation

- If the program adds a simulation sequence, you can click the simulation button to verify. If not available, click on run in the program editing window to view the effect of laser cutting.
- During the welding process, if a collision is detected, the target object will be highlighted.

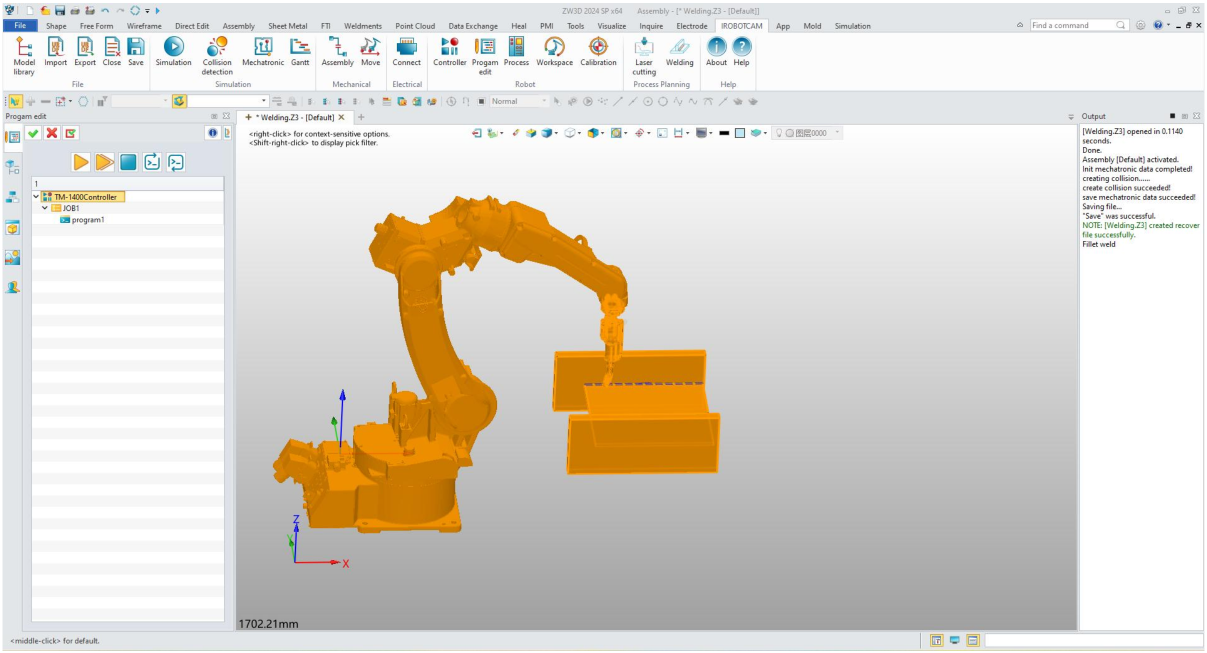


Figure 15-14: Simulation