Design, optimization, debugging, and simulation of an agricultural biomimetic hexapod robot

Wentao Zhang

School of Engineering, University of Aberdeen, King Street, Aberdeen, Scotland, AB24 3FX, United Kingdom.
waynezhang1999@outlook.com

Abstract: This paper comprehensively studies the model establishment, kinematic analysis, hardware design, software development, and agricultural applications of hexapod robot, laying a solid foundation for their practical application in the agricultural field and proposing prospects for future research and development. This article mainly conducts in-depth research on the mechanical design, kinematic analysis, hardware implementation, software development, and application in the agricultural field of hexapod robots. Firstly, a brief introduction to biomimetic robots will be provided, and a comparative analysis will be conducted on existing common legged robots. Afterwards, use solidworks-2021 for 3D modeling to draw the legs and main structure of the robot. In terms of kinematic analysis, by studying the mechanism structure and motion characteristics of hexapod robot, the forward and inverse kinematic equations are derived, and the gait planning and motion control algorithms of the robot are analyzed in detail. These analyses provide a theoretical basis for the robot's program and actual motion. In terms of hardware, a lightweight and high-performance hexapod robot hardware platform has been designed using mainstream controllers such as Borboardino and ultrasonic sensors. This hardware platform has excellent responsiveness and adaptability, and can efficiently move and operate in complex agricultural environments. In terms of software, a comprehensive control system has been developed, including motion planning, action execution, and obstacle avoidance modules. By integrating triangular gait technology and related algorithms, robot can achieve navigation and obstacle avoidance, and perform tasks such as crop planting and spraying in the agricultural field. The simulation file of the hexapod robot was established using CoppeliaSim software, achieving motion based on the robot body, six legs, and end effectors. Finally, a 3D-printer is used to print and assemble all components, and complete the design through debugging and verification. The hexapod robot is capable of autonomous cruising in farmland, completing tasks such as planting and watering crops, providing efficient and intelligent agricultural production solutions for farmers.

Keywords: biomimetic, hexapod robot, triangular gait, kinematics, control, 3D printing, Borboardino, CoppeliaSim.

1. Introduction

1.1. Overview of Biomimetic Robots

In the past few decades, the inspiration or design ideas for robots have mainly come from mathematics, mechanics, electronics, and computer science. On the one hand, this approach undoubtedly solidified the technical foundations of the discipline and led to the production of highly successful products, especially in the field of industrial robotics. [1] But with the changing times, humans have gained a deeper understanding and higher pursuit of robots. For one thing, nature provides scientists with rich sources of inspiration; On the other hand, with the rise of disciplines such as materials science, computer science, and artificial intelligence, the intersection of bionics and robotics has become possible. For the moment, we have mainly discussed about principles and techniques to design efficient artificial systems, but it is also more and more acknowledged that bio-inspired robotics is also a very good methodology to better understand cognition and its biological foundations. [2] Generally speaking, the principles of bionics are abstracted from the observation of a certain organism (such as behavior, action), and bionic robots can also replace classical engineering solution by imitating their mechanisms or process, achieving good results. Therefore, there is a mutually inspiring relationship between the biological world and engineering. The biological community has provided many complex and efficient solutions, inspiring engineers to develop smarter and more flexible robots. At the same time, technological advancements in engineering have also provided new tools and

methods for biological research, promoting a deeper understanding of the structure and behavior of organisms.

1.2. Objectives

By imitating the gait and movement mechanisms of insects or animals, scientists have invented footed robots. Compared to traditional wheeled robots, legged robots can achieve efficient movement in complex terrain and narrow spaces, and have better stability and load capacity, which can adapt to the complex terrain and constantly changing environment in nature. Based on the straight knee walking gait of humans, Boston Dynamics has developed a more mature bipedal robot that has a larger workspace and is more easily accepted by humans. But balance poses high technical requirements for attitude control, making it difficult to control. Research has shown that bipedal robots have not fully achieved human locomotion capabilities because the leg configuration and driving form of the robot have not reached the level of human skeletal muscle locomotion ability [3]. In the field of continuous control, inspired by common vertebrates such as dogs and cats, there is currently a deep deterministic strategy gradient for reinforcement learning based on the arbitration mechanism of the lower prefrontal cortex layer. This algorithm has shown good results in gait control of quadruped robots [4]. Similarly, reference to arthropods and mollusks such as octopuses, eight legged robots, and multi legged robots have also been developed. Walkers with eight legs may maintain four legs on the ground at all times, and have four legs in the return stroke. The support polygon is a quadrilateral, with an area approximately twice that of the triangular support polygon of the hexapod with the same leg-base and track.[5] In the field of continuous control, inspired by common vertebrates such as dogs and cats, there is currently a deep deterministic strategy gradient for reinforcement learning based on the arbitration mechanism in the lower prefrontal cortex. This algorithm has achieved good results in gait control of quadruped robots [4]. Similarly, arthropods and mollusks such as octopuses, eight legged robots, and multi legged robots have also been developed. Compared to bipedal and quadruped robots, hexapod robots have stable support points, faster movement speed, and stronger adaptability to terrain; Compared to an eight-legged robot, it has fewer degrees of freedom, simpler and more stable control, and requires lower energy consumption for gait. Overall, hexapod robots not only outperform other legged robots in many aspects, but can also promote the development of robotics technology and lay the foundation for more complex and diverse tasks in the future.

1.3. Design and Application

The research content of hexapod robots mainly includes model building, motion control, path planning, and other aspects. In the establishment of robot models, it is usually necessary to model the geometry, kinematics, and dynamics of the hexapod robot for simulation and controller design; This article will focus on establishing coordinate systems and solving forward and inverse kinematics. In terms of motion control, it is necessary to design efficient gait and motion control algorithms to enable the robot to walk or even climb stably. This article will mainly focus on achieving common hexapod robot gait. Path planning is a key issue in selecting the optimal path and avoiding obstacles. In order to improve perception and decision-making ability of robot, multiple sensors (such as cameras, laser rangefinders, gyroscopes, etc.) are usually fused together to obtain more comprehensive environmental information; This article will consider a certain load capacity and use common sensors for simple path planning and obstacle avoidance.

Hexapod robots have broad application potential, mainly involving the following fields:

Exploration and survey: In unknown or dangerous environments, such as disaster sites, steep terrain, etc., it can be used for exploration and survey tasks.

Search and rescue: Hexapod robots can be used to search for trapped individuals, provide information, and more.

Agriculture: Hexapod robots can be used for agricultural inspections, plant monitoring and collection, as shown in Figure 1.

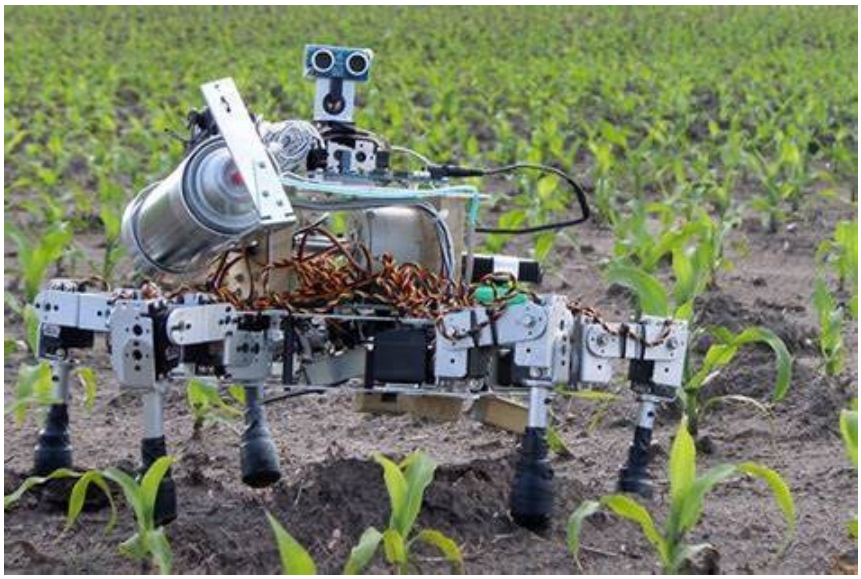

Figure 1

Indoor cleaning: In indoor environments, hexapod robots can be used for cleaning and inspection tasks, such as home cleaning and factory inspections.

This article mainly focuses on its application in agriculture.

2. Structural design

2.1. Mechanical Analysis

A hexapod robot consists of a main structure and six legs. Referring to the legs and feet of insects, the legs of the robot can be designed. Each leg of the biological prototype consists of a base joint, a femoral joint, and a tibial joint, with three joints rotating around the hip joint, root joint and knee joint in a single degree of freedom [7]. The legs of a robot typically consist of rigid segments connected at joints. Each joint has an actuator such as a servo motor for each degree of freedom (DOF)[8] The size and length of the legs, as well as the maximum rotation angle of the servo, will determine the range of motion of the robot. Based on the above characteristics, each leg can be designed as a series transmission mechanism with three degrees of freedom (one degree of freedom around the horizontal axis with two degrees of freedom around vertical axis), with three legs on each side of the torso and an overall axisymmetric distribution, totaling 18 degrees of freedom, as shown in Figure 2.

This design not only increases the stability of motion, but also increases the range of robot activity in a certain gait. At the same time, as a support structure and servo bearing structure, the legs need to reduce unnecessary weight as much as possible while ensuring a certain load capacity. An effective method is to hollow out some areas without changing mechanical stress. This not only reduces the material and time required for 3D-printing of the legs, but also reduces the energy consumption of movement. The sole of the foot needs to increase friction as much as possible, designed with a tip that not only enhances grip but also adapts to uneven terrain. Inspired by insect bodies, the main structure of hexapod robots usually comes in two forms: quadrilateral and hexagonal. As shown in Figure 3, in order to better accommodate the controller and battery, this design adopts a quadrilateral structure.

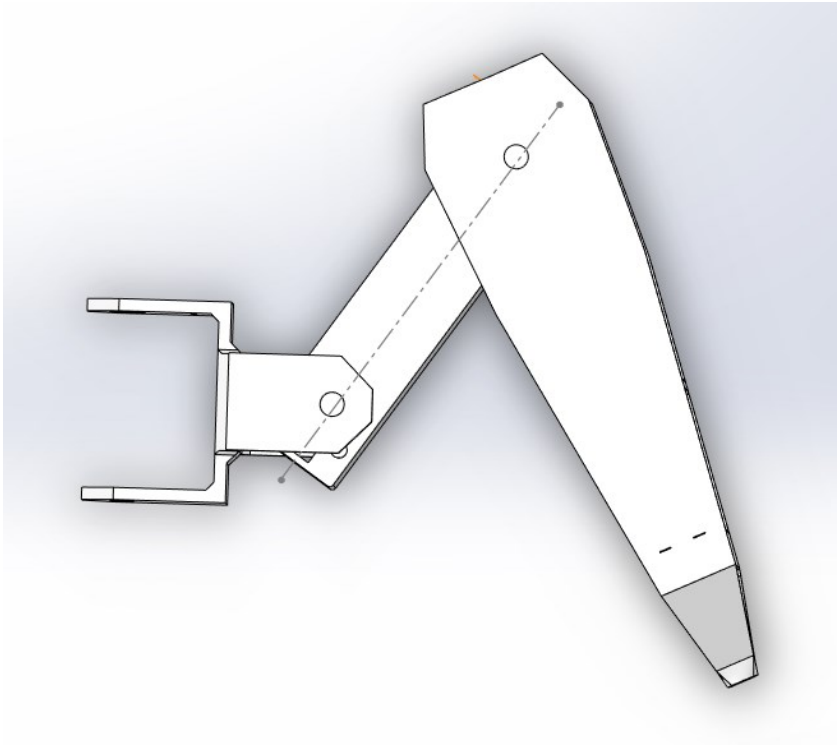

Figure 2

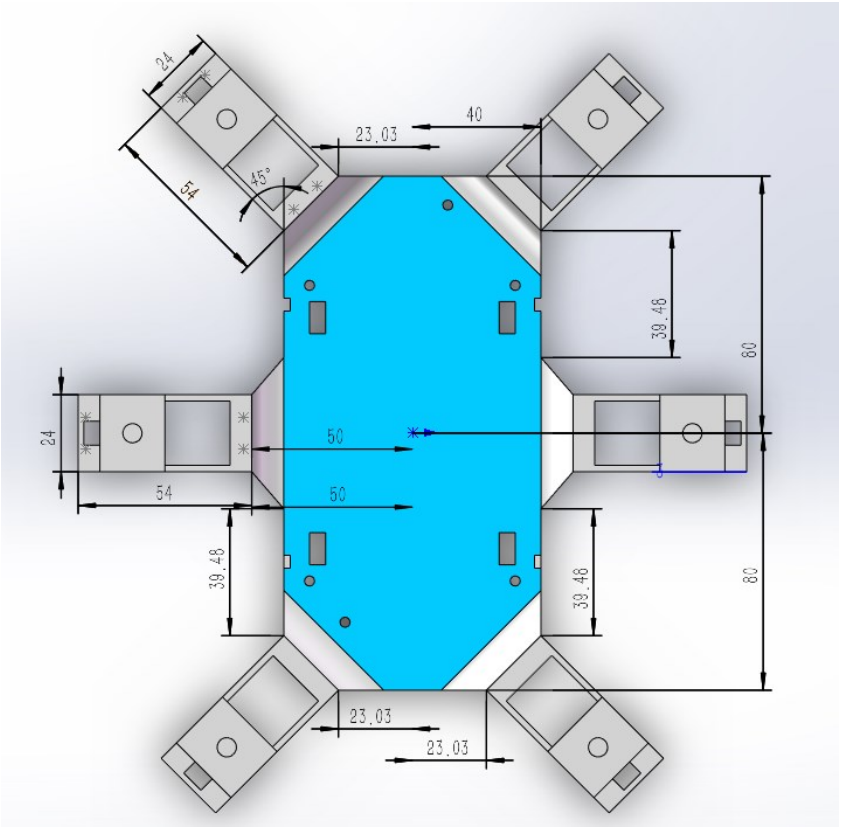

Figure 3

Combined with leg degrees of freedom, it can achieve translation and rotation in three directions. The overall degree of freedom is independent of the number of legs, totaling 6 degrees of freedom.

2.2. Gait analysis

The basic data of the robot is as follows table 1, due to accuracy issues, there is a deviation of about 1% in length and about 10% in quality:

Table 1

Component	Length (mm)	Width (mm)	Height (mm)	Mass (kg)
main body	160	80	5	106.31
Leg 1	39	51	50	12.67
Leg 2	110	24	31	23.84
Leg 3	187	49.7	40	37.72
Whole body	Maximum extension radius:360			180.54

Generally speaking, the movement of robots requires maintaining overall stability while meeting their own motion constraints. The common method for simulating insect gait is triangular gait. In a triangular gait, three legs that are not adjacent to each other serve as a group of legs, serving as swinging or supporting legs, and the state of each group of legs is completely consistent. When two sets of legs alternate for support and swing, they provide forward momentum. Numbering the six legs as shown in Figure 4 shows the forward pattern of the triangular gait, with the red link representing the supporting leg.

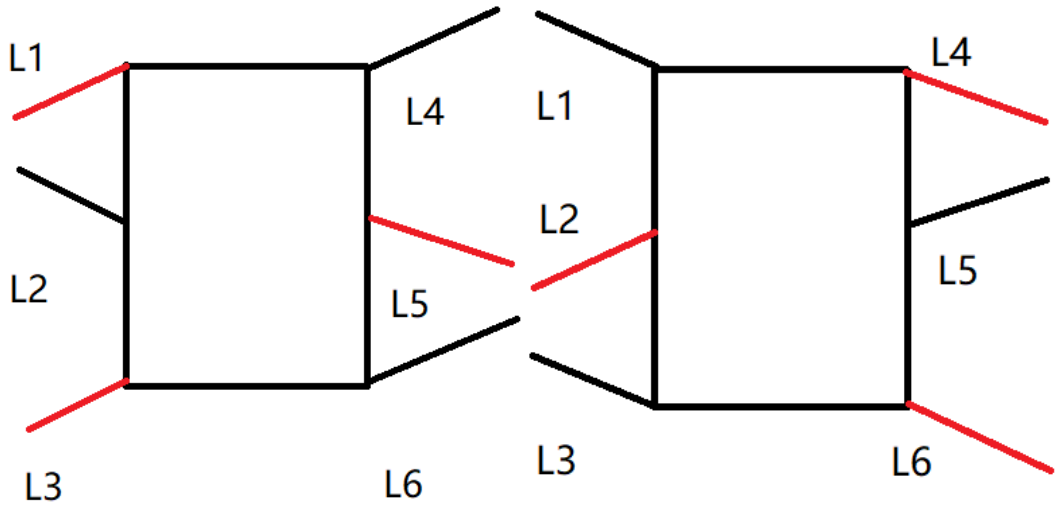

Figure 4

The body is a regular rectangular body, and the weight of the hexapod walking robot is concentrated on the body. Therefore, it can be approximately considered that the center of gravity of the robot, the center of gravity of the body, and the center of gravity of the body coincide [9]. According to the ZMP criterion, the robot is stable when the vertical projection of the robot's center of gravity on the horizontal ground is located within the support surface of the robot. At any time, the position relationship of the center of gravity on the supporting surface geometry is shown in Figure 5.

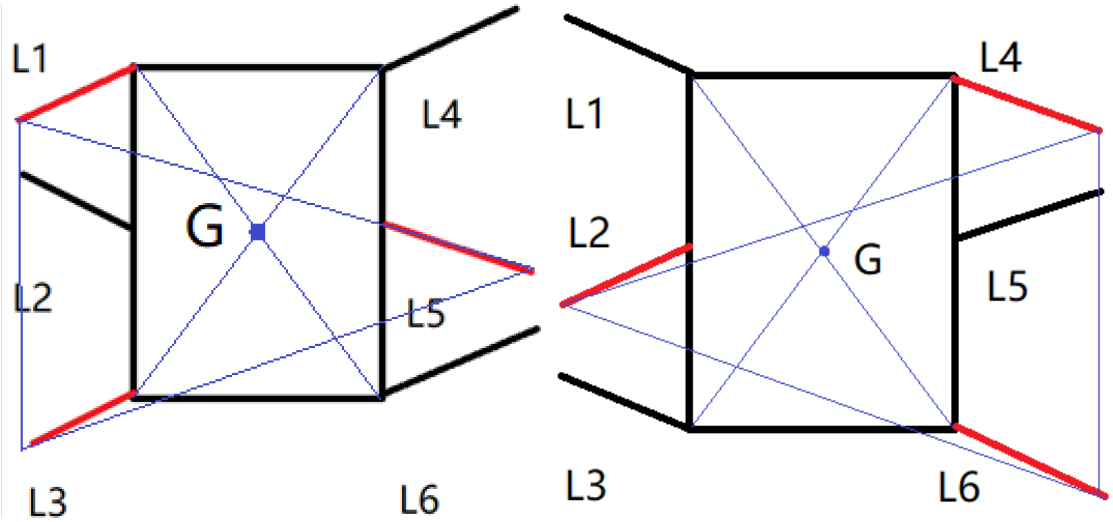

Figure 5

If the suspended state of the legs is defined as 1 and the support state is defined as 0, the duty factor can be defined as the ratio of each leg's support time to the total time within a single cycle. The phase relationship of the robot legs is shown in Figure 6. When the duty cycle is 0.5, assuming that the robot advances X meters in one gait cycle and takes t seconds, then a set of legs with supporting phases supports the robot to advance X/2 meters and takes t/2 seconds.

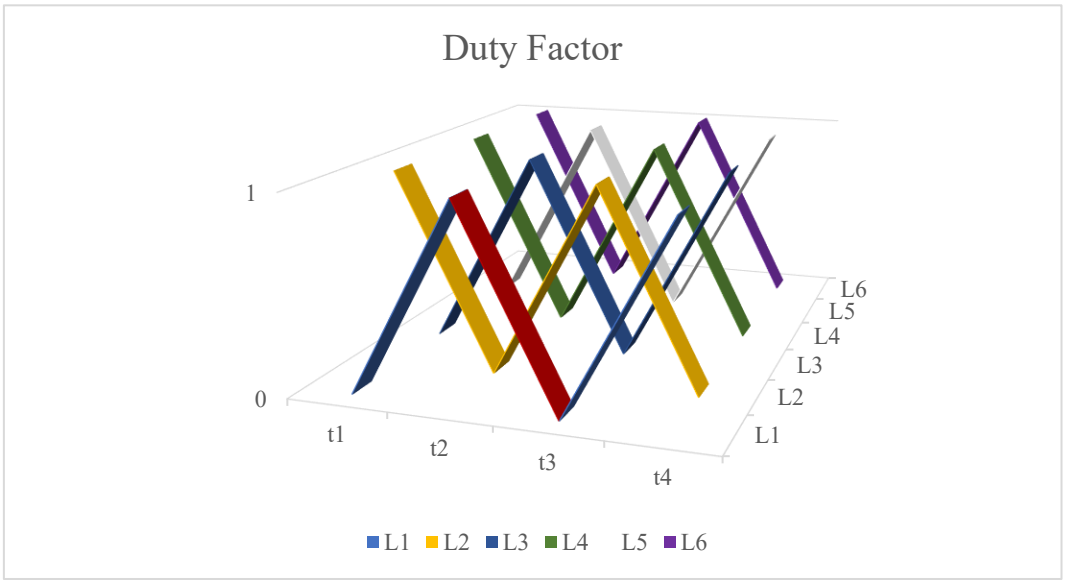

Figure 6

2.3. Kinematic Analysis

Forward kinematics is used to calculate the distance from the root to the end of each leg. The sufficient condition for a series robotic arm to have an inverse kinematic analytical solution is to meet the Pieper criterion. This method is based on solving the inverse kinematics equations of the robot, ie., determining such positions of individual members for which the effect is at a given point.[10] The establishment of a single leg coordinate system is shown in Figure 7, with the root of the leg as the initial coordinate system and calculated in the plantar coordinate system (end effector). Establish a coordinate system for each joint and use homogeneous coordinate transformations to describe the relative positions and postures between these coordinate systems.

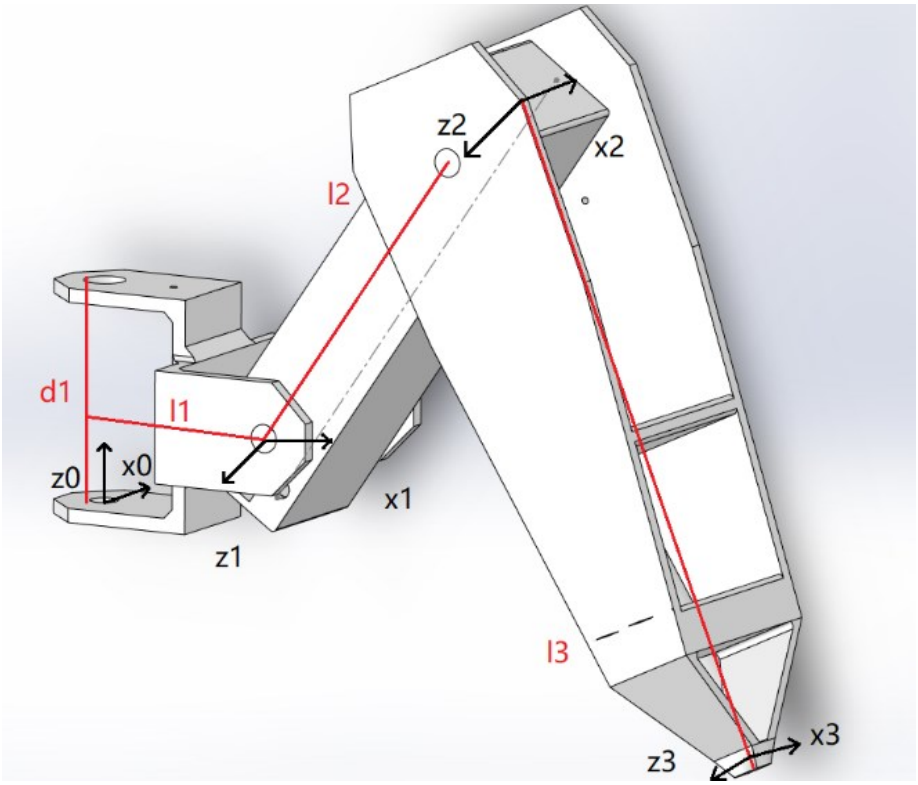

Figure 7

For a series mechanism, the DH parameter describes the connection relationship between each continuous joint through four parameters, including:

d: The distance between z-axis of the previous joint and the current joint, measured along the x-axis of the current joint.

θ: The angle between z-axis of the previous joint and the current joint, which is the angle of rotation around the z-axis of the current joint.

a: The distance between the x-axis of the previous joint and the current joint, measured along the z-axis of the current joint.

α: The angle between the x-axis of the previous joint and the current joint, which is the angle of rotation around the x-axis of the previous joint.

By using these parameters, a mathematical model can be established between the rotating joint and the connecting rod of the robotic arm.

The DH parameter table is as follows:

Table 2

Joint	a_1	α	d	θ
1	*L1*	90°	0	*θ1*
2	*L2*	0°	0	*θ2*
3	*L3*	0°	0	*θ3*

Considering the limitations of the robot structure, which means that there cannot be overlapping areas between adjacent legs (as shown in red in Figure 8), The value range of θ1 generally does not exceed 45 °, which will be reflected in subsequent simulations and programs. At the same time, to ensure the normal walking of the robot, The values of θ 2 and θ3 are usually around 60 ° and 90 °.

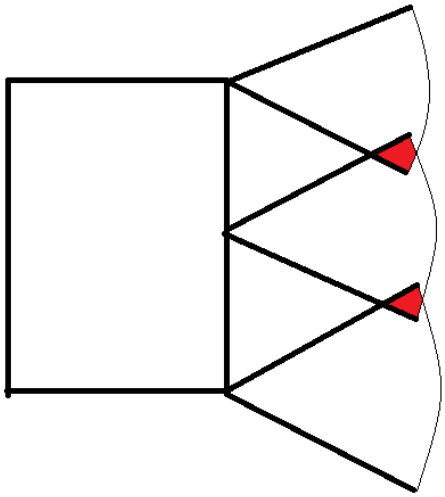

Figure 8

If the center of gravity of the robot is used as the reference coordinate system, where the main body is located in the X&Y plane, and the X&Y axis is parallel to the main body edge, and the Z-axis is perpendicular to the main body. Due to the fact that there is only movement on the X&Y plane and rotation around the Z-axis from the main body to the robot leg base (as shown in Figure 9), transformation matrix from the foot end coordinate system to the theme coordinate system (relative to the fixed coordinate system) is:

$$T_1 = T(x, y, 0) * R(z, \theta) * A_0^3$$

According to Table 1, detailed information can be obtained as shown in Table 3, the values of the T matrix and R matrix can be obtained by introducing them, shown as figure 9:

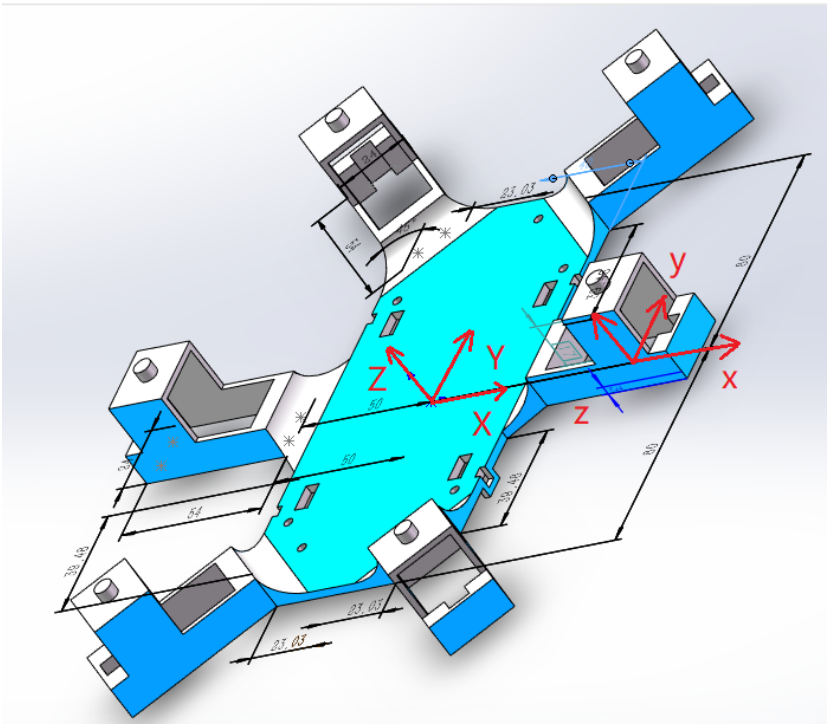

Figure 9

Table 3

Leg	X(mm)	Y(mm)	Z (°)
L1	-40	80	135
L2	-40	0	180
L3	-40	-80	-135
L4	40	80	45
L5	40	0	0
L6	40	-80	-45

Assuming A1 is the attitude and position of the first link of the base coordinate system, and A2 is the position and attitude of the second link of the first link, the matrix T1 of the second link to the base coordinate system can be calculated by the following relationship:

$$T_1 = A_1 * A_2$$

Similarly, by multiplying the transformation matrix of each leg, the pose and joint angle of the robot's end effector can be obtained.

$$A_1^0 = \begin{bmatrix} -\cos\theta_1 & 0 & \sin\theta_1 & l_1 * \cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & l_1 * \sin\theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_2 * \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_2 * \sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & l_3 * \cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & l_3 * \sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^0 = A_1^0 * A_2^1 * A_3^2$$

$$= \begin{bmatrix} \cos\theta_1 * \cos\theta_2 * \cos\theta_3 - \cos\theta_1 * \sin\theta_2 * \sin\theta_3 & -\cos\theta_1 * \cos\theta_2 * \sin\theta_3 - \cos\theta_1 * \sin\theta_2 * \cos\theta_3 & \sin\theta_1 & P_x \\ \sin\theta_1 * \cos\theta_2 * \cos\theta_3 - \sin\theta_1 * \sin\theta_2 * \sin\theta_3 & -\sin\theta_1 * \cos\theta_2 * \sin\theta_3 - \sin\theta_1 * \sin\theta_2 * \cos\theta_3 & -\cos\theta_1 & P_y \\ \sin\theta_2 * \cos\theta_3 + \cos\theta_2 * \sin\theta_3 & -\sin\theta_2 * \sin\theta_3 + \cos\theta_2 * \cos\theta_3 & 0 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From this, the forward kinematics formula for a single leg can be obtained:

$$P_x = l_1 * \cos\theta_1 + l_2 * \cos\theta_1 * \cos\theta_2 + l_3 * \cos\theta_1 * \cos(\theta_2 + \theta_3) \quad (1)$$

$$P_y = l_1 * \sin\theta_1 + l_2 * \sin\theta_1 * \cos\theta_2 + l_3 * \sin\theta_1 * \cos(\theta_2 + \theta_3) \quad (2)$$

$$P_z = l_2 * \sin\theta_2 + l_3 * \sin(\theta_2 + \theta_3) \quad (3)$$

In computer simulation, the angle to be reached is usually calculated based on the distance traveled. The forward kinematics solutions of equations (1), (2), and (3) above have been obtained, and the three formulas contain three unknown variables related to angles. Theoretically, numerical methods can be used to solve them. Firstly, since (1) and (2) contain the same terms regarding unknown variables, they can be simplified:

$$\frac{P_x - l_1 * \cos\theta_1 - l_2 * \cos\theta_1 * \cos\theta_2}{l_3 * \cos\theta_1} = \frac{P_y - l_1 * \sin\theta_1 - l_2 * \sin\theta_1 * \cos\theta_2}{l_3 * \sin\theta_1} \quad (4)$$

Since l3 is not 0, merging similar items yields:

$$P_x * \sin\theta_1 = P_y * \cos\theta_1$$

Obtaining information about Solution of θ1

$$\theta_1 = arctan\frac{P_y}{P_x} \tag{5}$$

Secondly, simplify equation (3) to:

$$sin(\theta_2 + \theta_3) = \frac{P_z - l_2 * sin\,\theta_2}{l_3} \tag{6}$$

According to the definition of trigonometric functions, trigonometric functions with the same angle conform to:

$$cos^2(\theta_2 + \theta_3) + sin^2(\theta_2 + \theta_3) = 1$$

Bringing (1) and (6) into the available information about θ The relationship of 2 can be simplified to obtain:

$$(2 * l_1 * l_2 * cos^2\theta_1 - 2 * P_x * l_2 * cos\,\theta_1) * cos\,\theta_2 - 2 * P_z * l_2 * cos^2\theta_1 * sin\,\theta_2$$
$$= (l_3^2 - l_2^2 - P_z^2) * cos^2\theta_1 + 2 * P_x * l_1 * cos\,\theta_1 - P_x^2$$
$$\text{Let } a = (2 * l_1 * l_2 * cos^2\theta_1 - 2 * P_x * l_2 * cos\,\theta_1),$$
$$b = -2 * P_z * l_2 * cos^2\theta_1,$$
$$\text{and } c = (l_3^2 - l_2^2 - P_z^2) * cos^2\theta_1 + 2 * P_x * l_1 * cos\,\theta_1 - P_x^2$$

The above equation can be converted to

$$a * cos\,\theta_2 + b * sin\,\theta_2 = c \tag{7}$$

One solution is to first shift the term and then square both sides of the equation simultaneously, transforming it into a quadratic equation of one variable. However, the number of solutions is not unique, and using discriminants to solve requires the following conditions:

$$a^2 + b^2 \geq c^2, where\ A\ and\ b\ are\ not\ all\ 0, and -1 \leq cos\,\theta_2 \leq 1$$

Another method is to construct auxiliary angle solutions. Construct a new angle A, so that

$$cos\,A = \frac{a}{\sqrt{a^2 + b^2}}, sin\,A = \frac{b}{\sqrt{a^2 + b^2}} (At\ this\ point, cos^2\,A + sin^2\,A = 1)$$

Carried into (7), with

$$(\sqrt{a^2 + b^2}) * (cos\,\theta_2 * sin\,A + sin\,\theta_2 * cos\,A) = c$$

Simplify and obtain

$$\theta_2 = arcsin\frac{c}{\sqrt{a^2 + b^2}} - A$$

The value of A is determined by a, b, c, and has

$$tan\,A = \frac{sin\,A}{cos\,A} = \frac{a}{b}$$

Bring in the above equation with

$$\theta_2 = arcsin\frac{c}{\sqrt{a^2 + b^2}} - arctan\frac{a}{b} \tag{8}$$

Bring the values of a, b, and c into equation (8) to obtain

$$\theta_2 = arcsin\frac{(l_3^2 - l_2^2 - P_z^2) * cos^2\theta_1 + 2 * P_x * l_1 * cos\,\theta_1 - P_x^2}{2 * l_2 * cos\,\theta_1\sqrt{l_1^2 * cos^2\theta_1 - 4 * l_1 * P_x * cos\,\theta_1 + P_x^2 + P_z^2 * cos^2\theta_1}} - arctan\frac{l_1 * cos\,\theta_1 - P_x}{P_z * cos\,\theta_1} \tag{9}$$

Finally, by taking the value obtained from (9) into equation (6), we can obtain the equation about θ Formula for 3:

$$\theta_3 = arcsin\frac{P_z - l_2 * sin\,\theta_2}{l_3} - \theta_2 \tag{10}$$

According to formulas (5), (9), and (10) in sequence, the required angle of the joint can be derived based on coordinate transformation. The formulas can be solved in program.

3. Electrical design

3.1. Hardware Design

The controller uses the Lynxmotion BotBoarduino Shield compatible robot controller from the Arduino platform, which is a hybrid of BotBoard II and Arduino Duemilanove and is very suitable for controlling small robot projects. As shown in Figure 10, it contains an onboard speaker, three buttons and LEDs, a reset button, logic&servo power input, an I/O bus with 20 pins, power and ground, 6 PWM output pins, and a 5V DC power supply. This controller supports a maximum of 15 servo motors to be directly inserted, perfectly meeting the 18 PWM outputs required in this design (the remaining three can be connected through PWM pins and connected to a common DC-5V and ground wire).

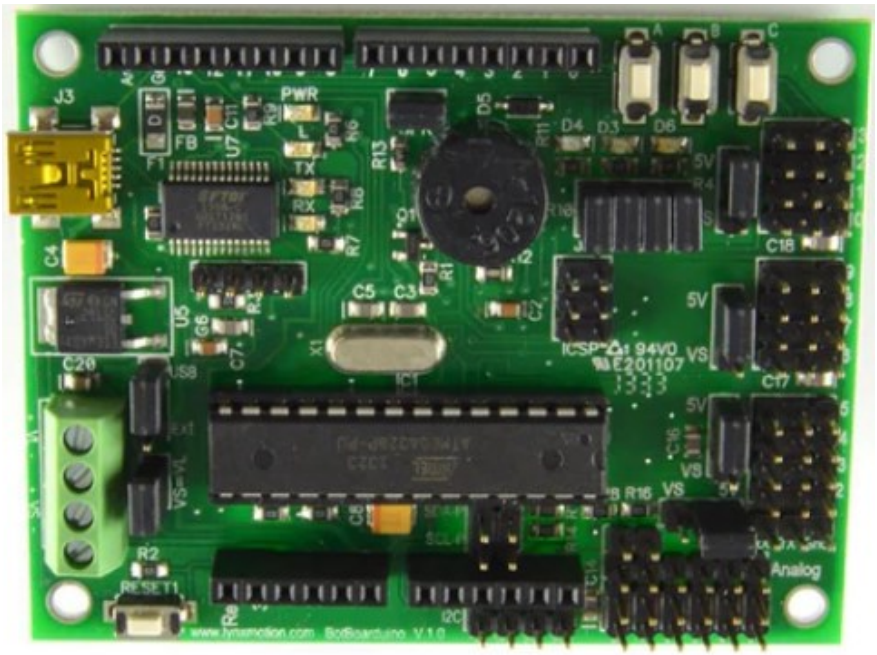

Figure 10

Compared to other microcontrollers, it has the following advantages:

Easy to learn and use: The software of the Arduino uses a simplified version of C++, making it easier to learn to program, and it provides an easier environment that bypass the functions of the micro-controller into a more accessible package.[11] Through simple code, users can control and interact with various sensors, actuators, and other peripheral devices.

Strong compatibility and high scalability: The Arduino ecosystem is very rich, with many compatible sensors, actuators, and other peripheral devices. Users can choose appropriate hardware components based on project requirements to accelerate the development process.

Open source and rich resources: Arduino is open source, which means users can freely view, modify, and share their hardware design and software code. The community has abundant resources, including sample code, libraries, and technical support, making it convenient for users to solve problems.

Widely used: Due to its ease of use and scalability, Arduino has been widely used in many fields, including the Internet of Things (IoT), automation, robotics, and education.

HS-645MG

The servo is a common actuator in robot systems, commonly used in multi legged robots, cameras, and low-cost robotic arms. The essence of the servo system is a small closed-loop servo system, which receives the target signal, drives the internal DC motor, and passes through the reduction gear set to adjust the angle of the output shaft; The output angle is sampled by a potentiometer, and then the feedback control system adjusts its output angle to match the target value, thus forming a typical closed-loop feedback control system for the entire process.

Figure 11

The advantages of a servo are small mechanical wear, high-speed response, small size, and low price, but limited position accuracy, load capacity, and ability to maintain steady state. It is particularly suitable for situations where control performance is not high and volume requirements are small. When selecting a servo for a robot, the main factors and indicators to consider are:

Maximum rotation angle: The angle between the extreme position of one end and the extreme position of the other end of the motor; Common ones often include 90 °, 180 °, and 270 °. A small number of servo motors support 360 °, which generally cannot control specific angles and can only control the speed of rotation. The nature of this servo motor is a speed closed-loop (rather than a general position closed-loop), and it is commonly used as a small car programming platform for education.

Response speed: The response speed represents the required time to reach position A from position B, usually expressed in units of s/60 °. The higher the response speed, the better, but sometimes rapid response can easily cause significant control jitter.

Maximum torque: It will affects the maximum load that the servo can accept. In order to increase maximum torque, in addition to using metal gears for high torque servo motors, sometimes double servo motors are also used in parallel (some desktop level 6z degree of freedom robotic arms are used at the bottom and humanoid robot waist positions). However, this method is prone to damage to one of the servo motors due to subtle differences in servo parameters in practical use.

Power supply voltage: The typical voltage generally starts at 4.8V and reaches a maximum of 12V. Generally speaking, under the condition of ensuring sufficient current, the higher the voltage, the greater the maximum power that could be provided.

The detailed parameters of the steering gear are shown in Table 4:

Table 4

Torque	7.7kg.cm(4.8V),9.6kg.cm(6.0V)
Motor speed	0.24sec/60°(4.8V),0.20sec/60°(6.0V)

Rated voltage	DC 4.8-6V
Operating current	350mA(4.8V),450mA(6V)
Size	4.6 x 19.8 x 37.8mm
Operation temperature	-20˚C ~ +60˚C
Weight	55.2g

PWM is an abbreviation for pulse width modulation, used to adjust the PWM duty cycle to control the motor. It changes the pulse width to control the average power supply voltage of the motor, which is widely used in the field of motor control.The development of compact controllers such as MCU have its own PWM pins built, e.g. Arduino microcontrollers.[12]

There are three ways to use PWM to control the servo:

Simulate the timing of PWM through software I/O
Output PWM signal through hardware PCA (Programmable Counter Array) module
By using a dedicated PWM signal generation chip, such as PCA9685

Arduino's Servo library combines both methods (1) and (2), implementing hardware PWM on pins with PCA output, while using software PWM on other pins. The duty cycle refers to the percentage of time that a circuit is connected to the entire circuit cycle. Generally speaking, the PWM signal frequency received by the servo is 50Hz. During the use of the servo, a 20ms PWM square wave is used to control the rotation angle. Among them, the duration of the high-level determines the angle of the output shaft and shows a linear relationship. When the pulse width of the high-level is between 0.5ms and 2.5ms, the servo can rotate to different angles (linear relationship from 0 to 180 °). For example, rotating 45 ° corresponds to 1ms, as shown in Figure 12.

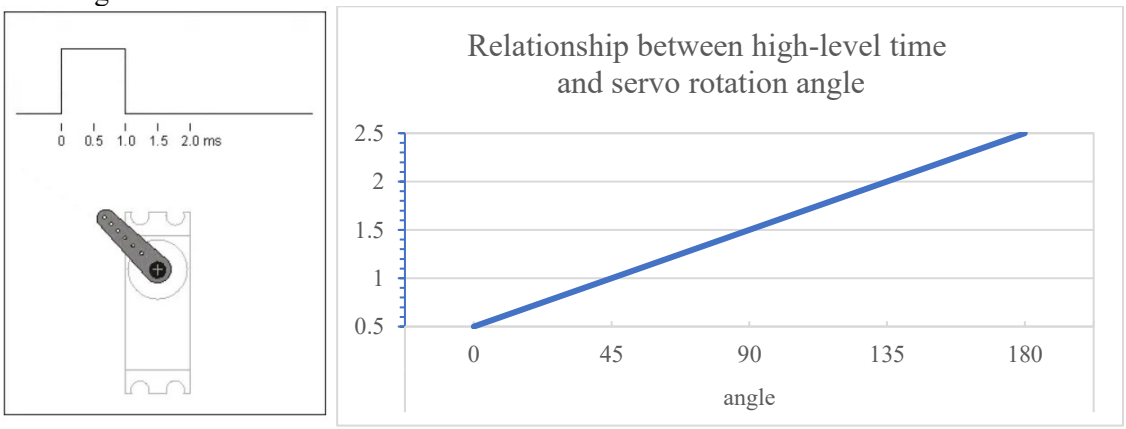

Figure 12

HC-SR04

The principle of ultrasonic ranging is to emit sound waves, which are reflected by objects and received by the module. The time difference is obtained, and then the distance between objects is calculated based on the speed of sound. This module has a total of 4 pins, including VCC, GND, Trig, and Echo. Arduino sends a digital signal of no less than 10us (high level) to the module through the Trig pin, triggering the module to emit 40KHz ultrasound. After receiving the returned-sound wave, the Echo outputs the high level. Based on the time difference between triggering and output, the distance can be calculated. The detailed parameters are shown in Table 5 (sound speed is 340m/s under standard environment):

Table 5

Operating Voltage	DC5V
Operating Current	15mA
Operating Frequency	40KHz

Range	20mm-4000mm
Dimension	45 x 20 x 15mm
Ranging Accuracy	3mm
Measuring Angle	15°
Trigger Input Signal	10μS TTL pulse

Figure 13

When using a common DC grounding terminal, a forward conducting diode can be connected in series to prevent backflow from damaging other devices. To measure whether the voltage or current at a certain point meets the requirements, a 0-ohm resistor can be connected in series. Among the 12 servo ports shown in the figure, the left side is the IO port, the middle is the power supply (5V), and the right side is the grounding terminal. Hardware connection diagram is shown in Figure 14.

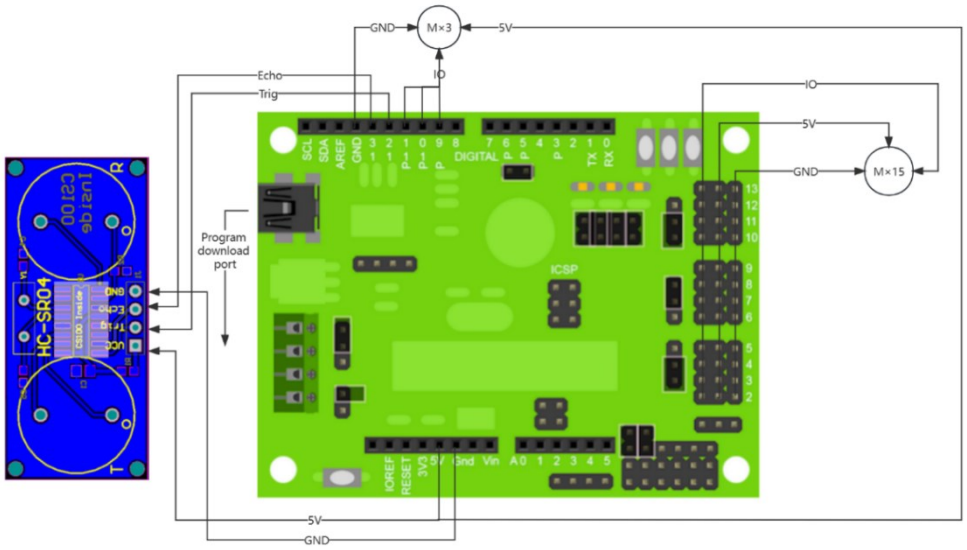

Figure 14

3.2. Software Design

In the Arduino IDE2.1.1 programming environment, select "Arduino Duemilanove" in the "Select motherboard or interface" column, as shown in Figure 15. Place the compiled program into it, click on debug, run, and burn it into the robot controller.

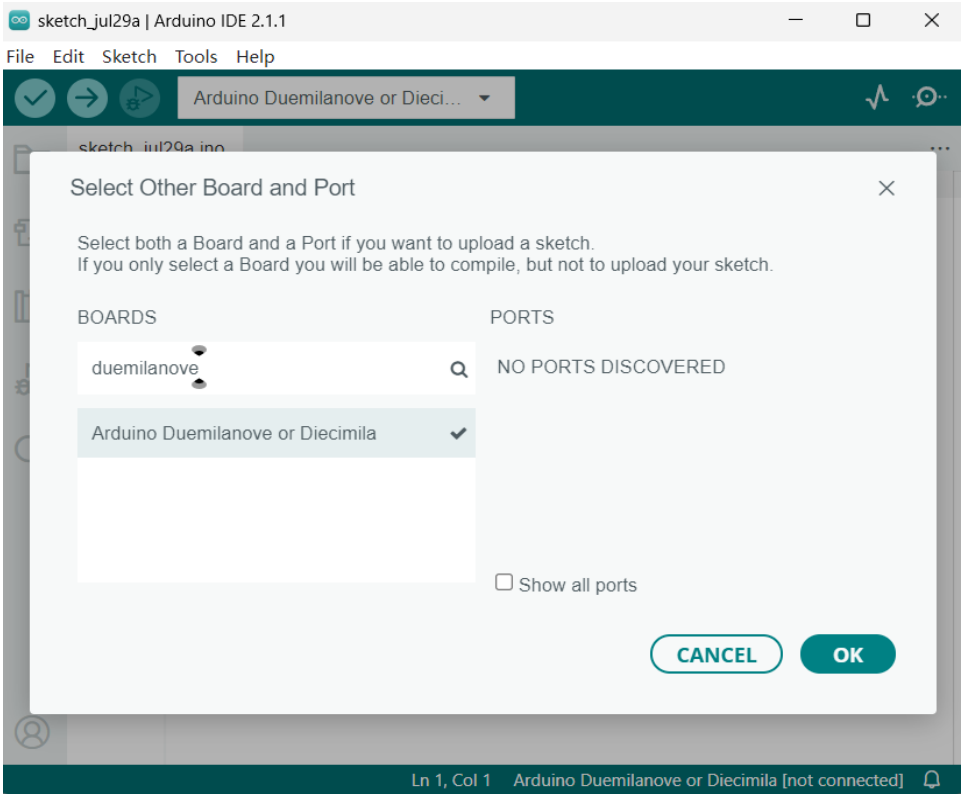

Figure 15

Similar to C language, the main function of the Arduino standard program entry is defined internally, and the setup function is called when the Arduino board starts. It can initialize variables, pin modes, start using a library, etc. This function only runs once during each power on and reset of the Arduino board. The loop function implements continuous loops, allowing programs to change state and respond to other events, and can be used to control the Arduino board in real life.

The application of hexapod robots in agriculture mainly includes sowing and watering. For example, in the program, it can be reflected as executing actions every certain distance (such as 1 meter) (such as rotating the main structure). When obstacles are detected, the robot can rotate in place to avoid them.

First, we need to import the basic library. In this design, it's # include<Servo. h>. Secondly, it is necessary to define the servo pins and ultrasonic module pins. Afterwards, it is necessary to initialize the aforementioned servo and define functions for different functions, including forward, rotation, etc. Finally, in the main loop, implement the call to the above functions.

Classic PID control

In this design, due to various circumstances, such as incomplete reflection of sound waves on the obstacle surface due to material reasons, or other environmental noise that may cause unstable detection values, there may be a certain deviation between the set distance and the actual moving distance. One method is to perform multiple sampling, taking the median of the sampled data to avoid the randomness of the measurement results. But this method may prolong the robot's response time and the effect is average. Another method is to establish a control system model with the set robot movement distance as the input and the difference in actual detection distance movement as the output, and use PID control for negative feedback.

The proportional link P can improve the speed of system response, but using the proportional link alone cannot stabilize the system performance in an ideal state. When there is a surplus, a larger proportional coefficient will cause a larger controller output, leading to excessive overshoot, system oscillation, and poor system stability; Integral regulation I can reduce errors and improve the steady-state performance of the

system based on proportional regulation; Differential link D belongs to advanced regulation, which can improve the dynamic performance of the system, reduce the overshoot and increase the stability of the system. The system block diagram is shown in Figure 16:

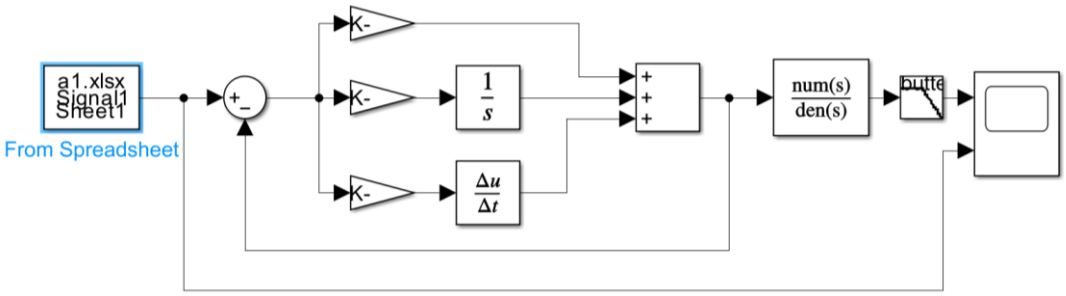

Figure 16

The waveform of the display is shown in Figure 17, where the blue line segment represents the input and the yellow line segment represents the output:

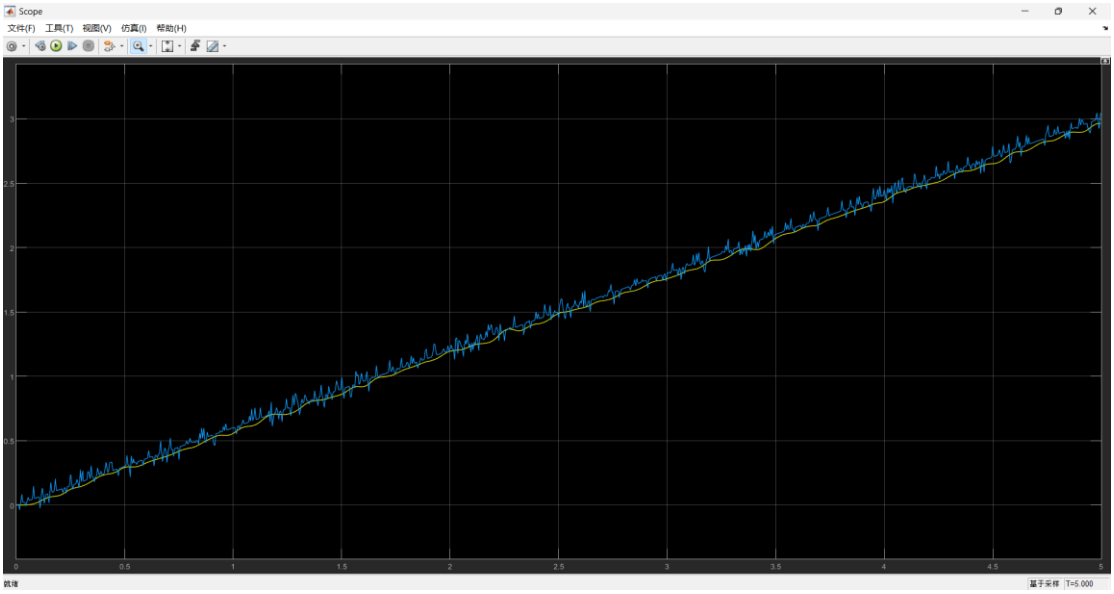

Figure 17

The overall flowchart is shown in the following figure 18:

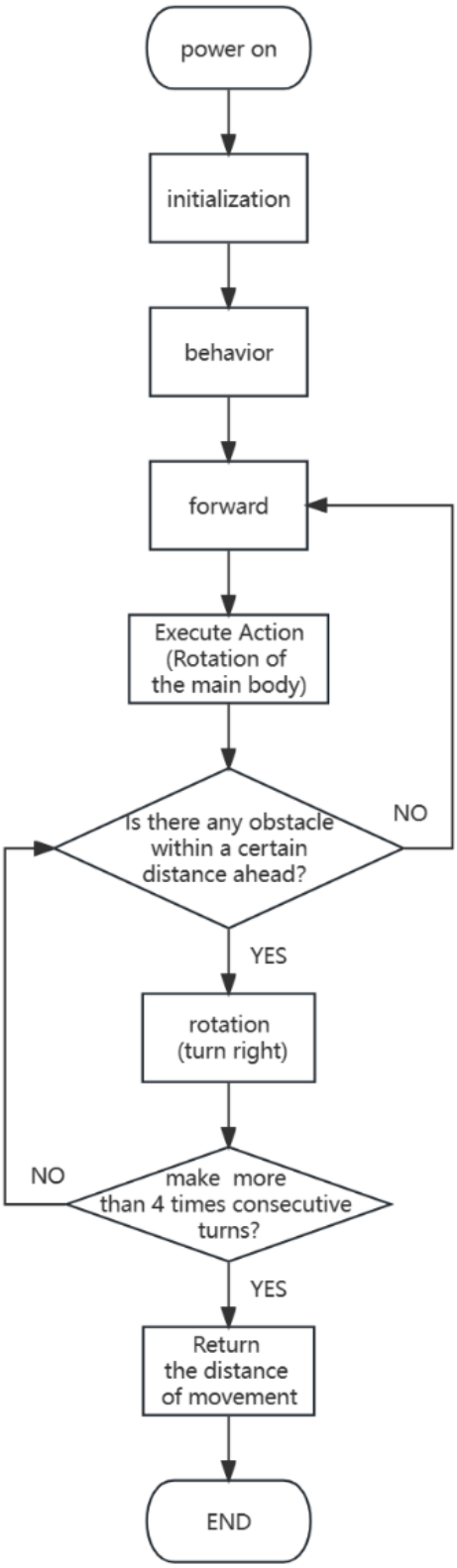
power on
initialization
behavior
forward
Execute Action (Rotation of the main body)
Is there any obstacle within a certain distance ahead?
NO
YES
rotation (turn right)
make more than 4 times consecutive turns?
NO
YES
Return the distance of movement
END

Figure 18

4. Design and Simulation

4.1. 3D Modeling

Solidworks2021 is a computer-aided design (CAD) software designed to provide engineers and designers with complete 3D design solutions, allowing users to create complex 3D models and supporting rapid design of multiple components and assemblies. It can import and export various CAD file formats, facilitating secondary design and cooperation with other software. In this design, first select the file, create a new, and create a new part, as shown in the figure 19:

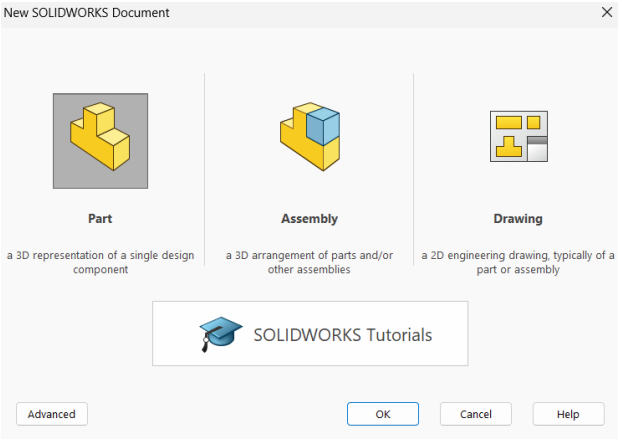

Figure 19

Select the top facing reference plane, click the command to sketch, select a straight line, first create a symmetry axis, and then create a closed polygon. Define the direction, size, and angle for each edge, as shown in the figure 20:

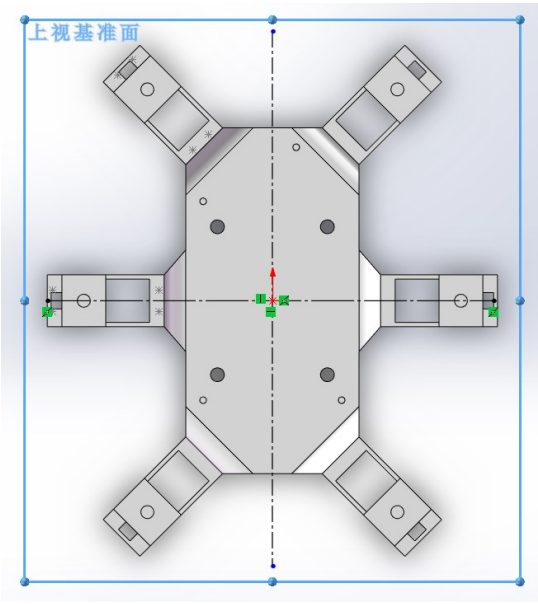

Figure 20

Secondly, select the polygon, click on the protrusion to stretch the main body on the upper reference plane, and then click on the protrusion to cut off each leg to generate the corresponding position of the servo, resulting in a three-dimensional model, as shown in the figure 21:

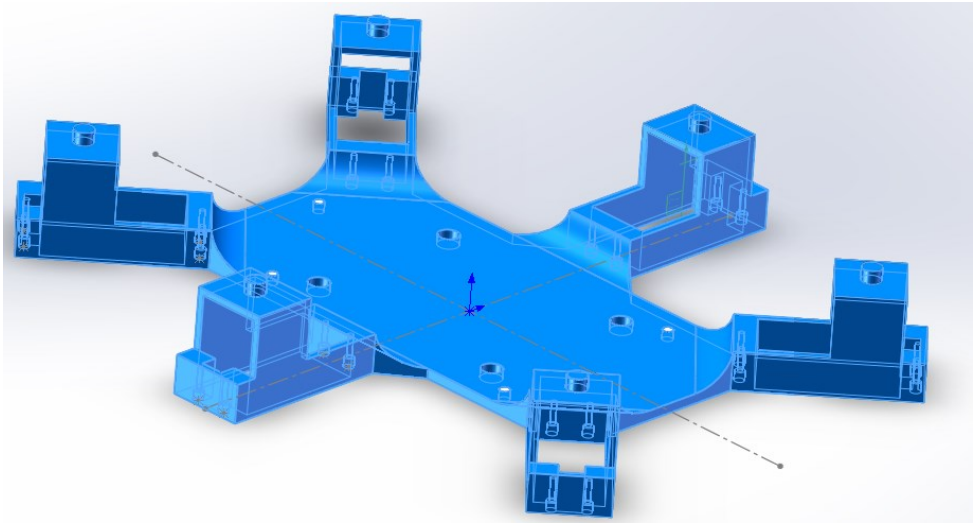

Figure 21

Finally, modify the screw holes and rotate the connection points, and cut them into three parts (limited by the size of the printer), as shown in the figure 22:

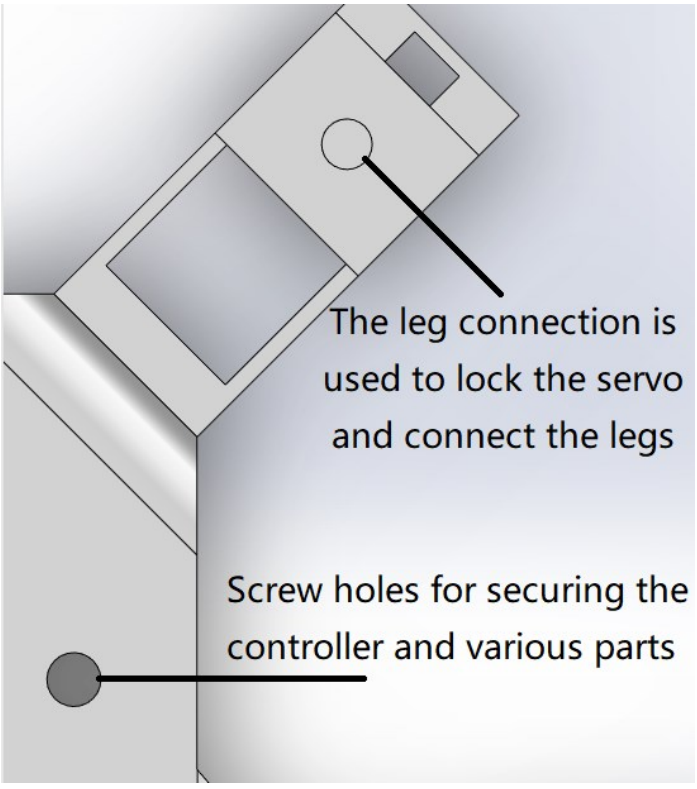

Figure 22

By analogy, the parts for each leg can be obtained. Select a file, create a new assembly, and import all components, as shown in the figure 23.

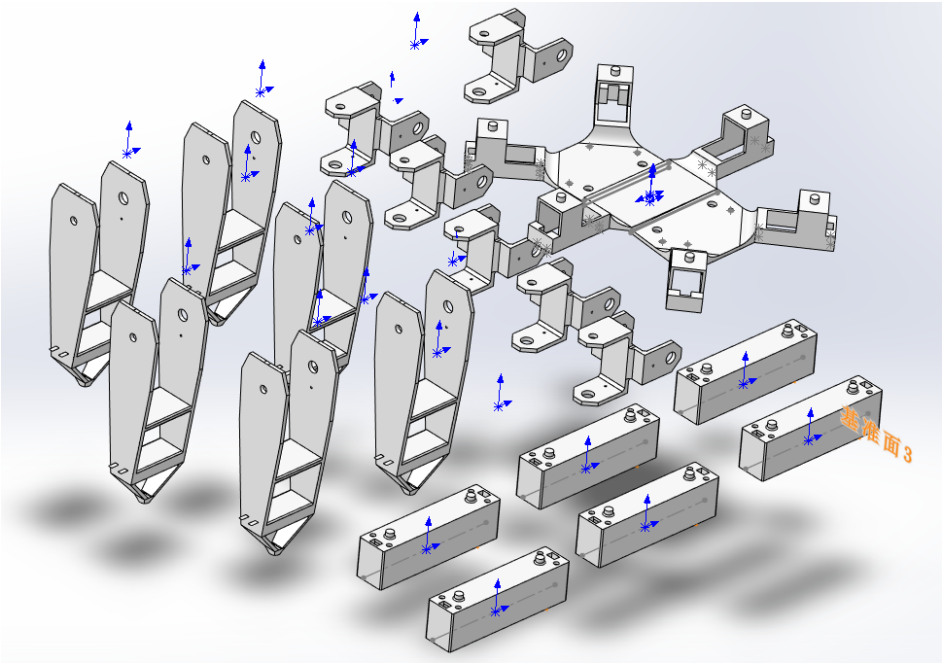

Figure 23

Using the fit option, define the connection relationships of all parts, use contact relationships for the main structure, and use rotation relationships for the legs. After the above steps, a robot model with motion relationships can be obtained, as shown in the figure 24.

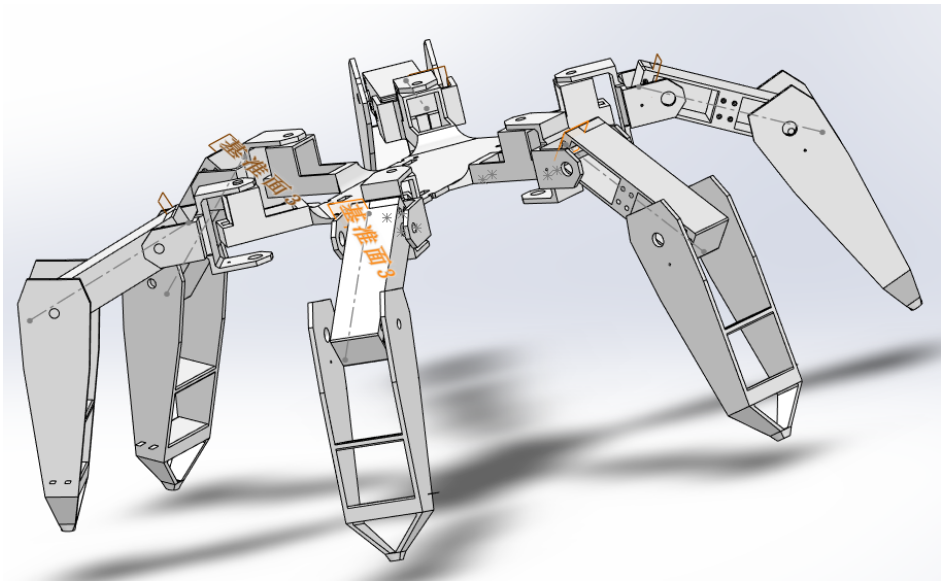

Figure 24

If the file is directly exported as an STL file and imported into simulation software, there may be a mismatch between the robot size and the main interface. A better solution is to use the SW2URDF plugin for structural definition (such as defining a leg as a sub part of the main structure).

Firstly, download the SW2URDF plugin and install it in the installation directory of solidworks2021, then restart the software. Afterwards, open the settings, select the add in option, and check the option to use the SW2URDF plugin in the other add ins column, as shown in the figure 25.

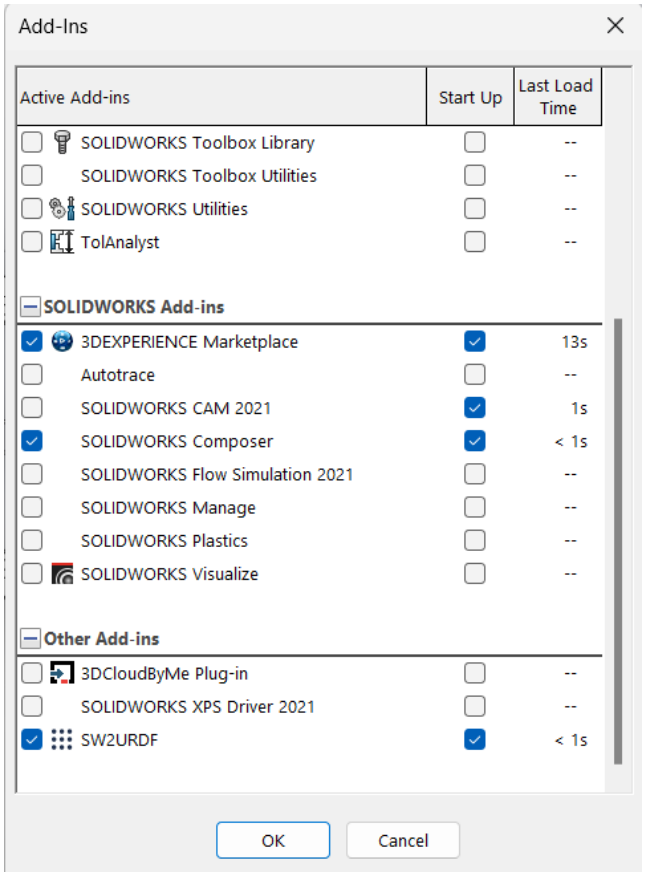

Figure 25

Afterwards, select the assembly body, open the tool, select "Export as URDF", define the maximum number of connections for each part and the corresponding joint names (usually cannot be duplicate and cannot be empty, otherwise an error will be reported), and then click to continue, as shown in the figure 26.

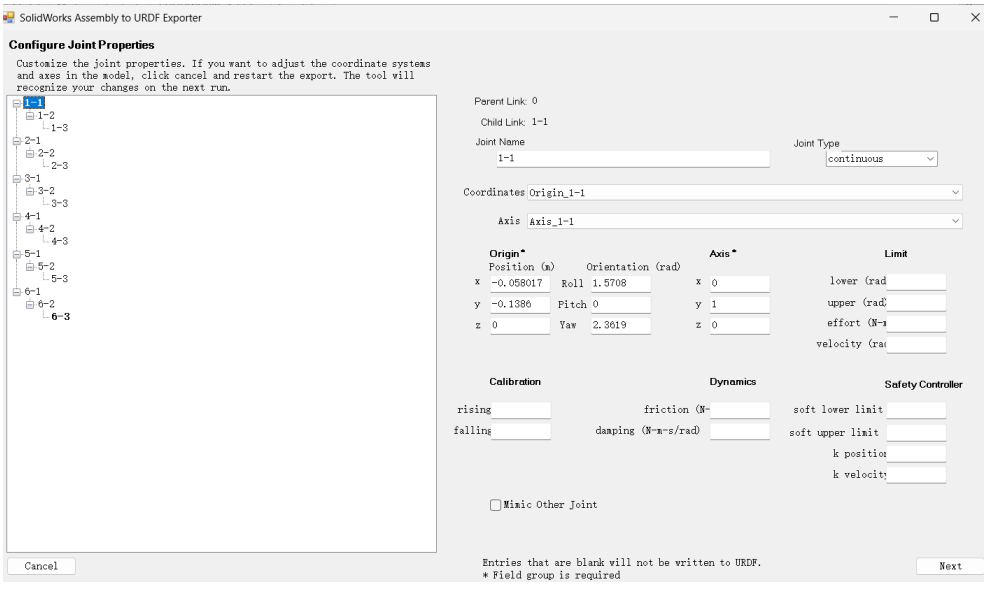

Figure 26

In the export interface, it can be seen that the coordinates, torque, and color information corresponding to each joint. After confirmation, click Export as URDF (the file suffix must be in URDF format) and MESHES (if not selected, it will cause subsequent simulation import failure), as shown in the figure 27.

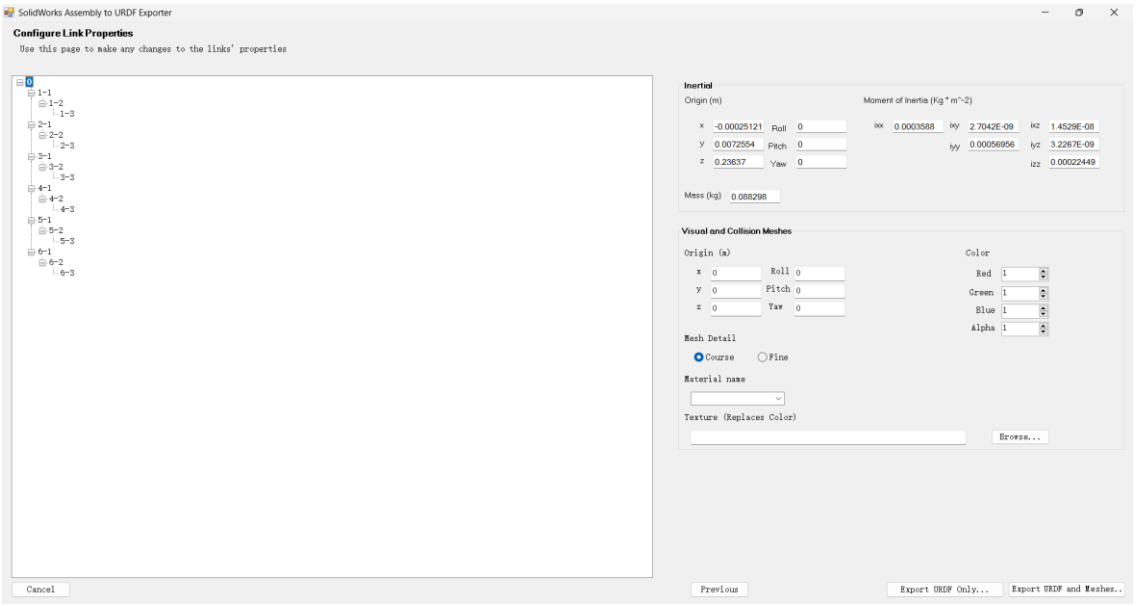

Figure 27

4.2. Robot simulation

CoppeliaSim (formerly known as V-REP, Virtual Robot Experimentation Platform) is an advanced robot simulation software developed by Coppelia Robotics, widely used in academic research, educational training, and industrial applications. It aims to provide researchers, engineers, and students with a powerful tool for designing, simulating, and validating various robot systems, control algorithms, and interactive behavior designs. At the same time, it provides various preset robot models and sensor simulators, covering various types of robots, drones, robotic arms, etc., which can quickly build complex robot systems. In terms of software, it provides rich language support, allowing users to use programming languages such as C/C++, Python, Java, etc. to control simulation scenarios and achieve custom control algorithms. The built-in physics engine makes the simulation of robots and the environment more realistic and accurate, and can simulate physical phenomena such as collisions and gravity effects between objects.

Open Copliansim 4.1.0 (EDU), click on the 'URDF Import' option, and select the saved file mentioned above (the URDF file must be placed in the same subfolder as the MESHES file, otherwise there will be only joint definitions without corresponding models), as shown in the figure 28.

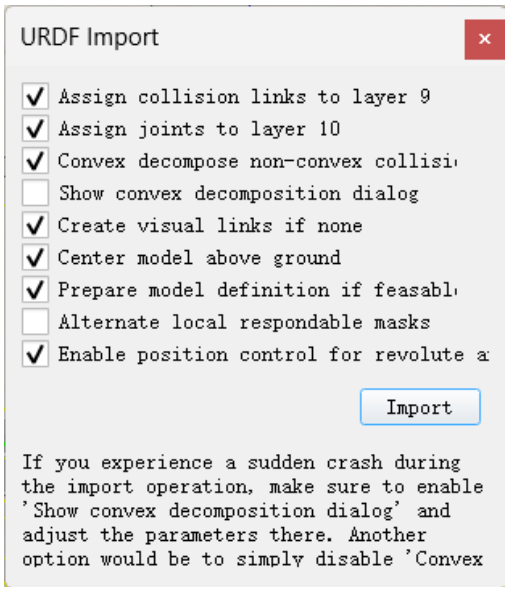

Figure 28

After importing, all parts are selected by default, and it can be seen that the parent-child attributes between joints have been defined in the left scene bar, as shown in figure 29.

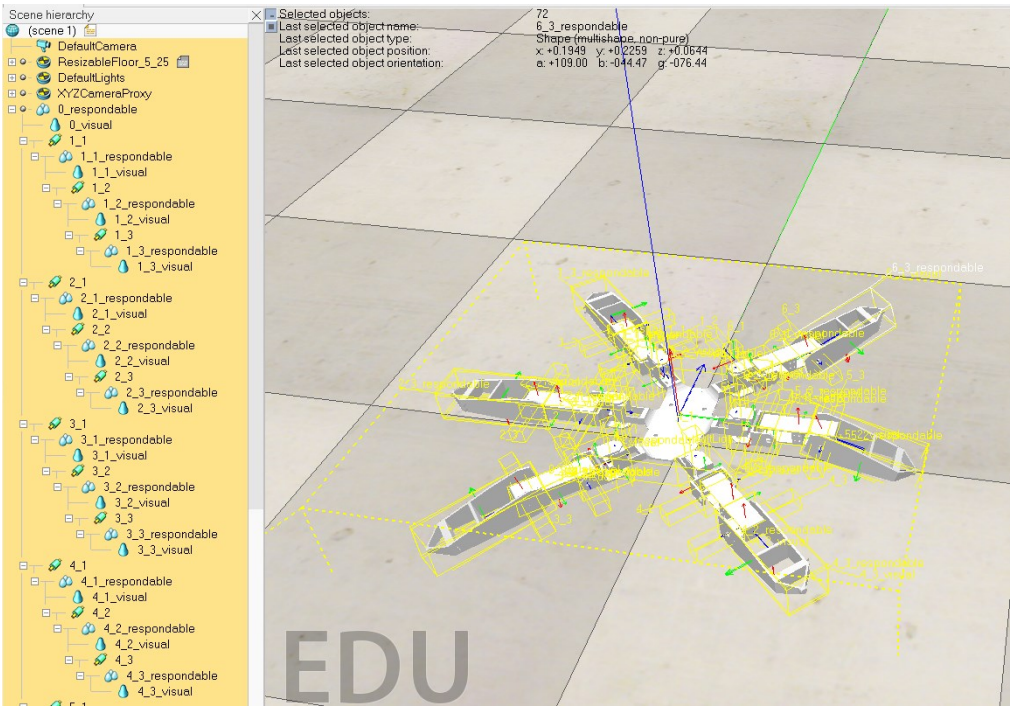

Figure 29

Open the options for each leg (such as leg 1-1 responsive in the picture), click on the show dynamic properties dialog option, and uncheck the options for body is dynamic and body is responsive. Afterwards, open each leg option again (such as leg 1-1) and select the Passive Move option from the mode drop-down menu. This can avoid threading between different legs or crossing the ground, as shown in figure 30.

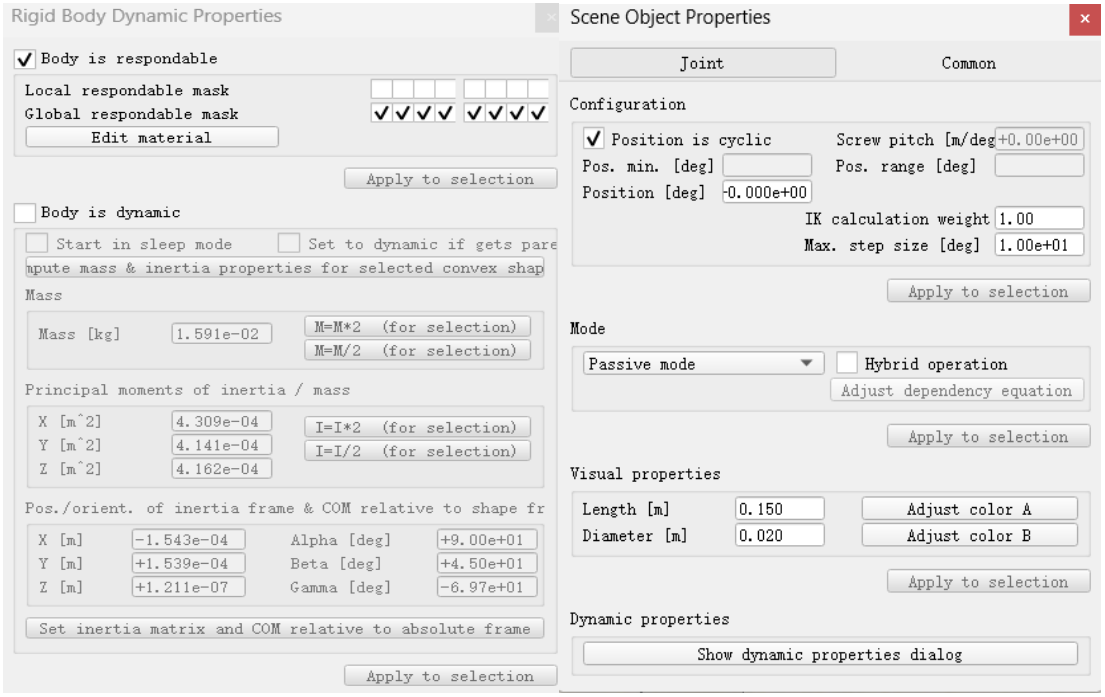

Figure 30

Right click on the robot and the main structure (i.e. 0-responsive), select add, associated child script, threaded to add program files to the robot, as shown in figure 31.

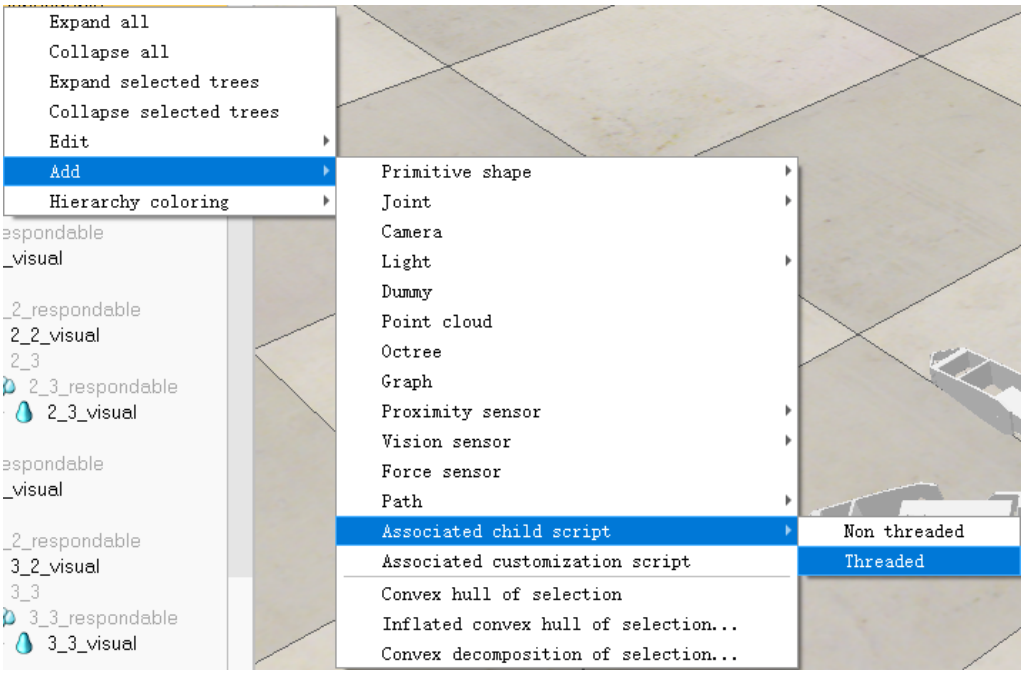

Figure 31

Create the end effector for each leg, click on the create handle (i.e. sphere), set it to be responsive (so that the force can act on the ground), and connect it to the corresponding force sensor. The force sensor, as a sub object of the leg, can connect the leg and handle, as shown in figure 32.

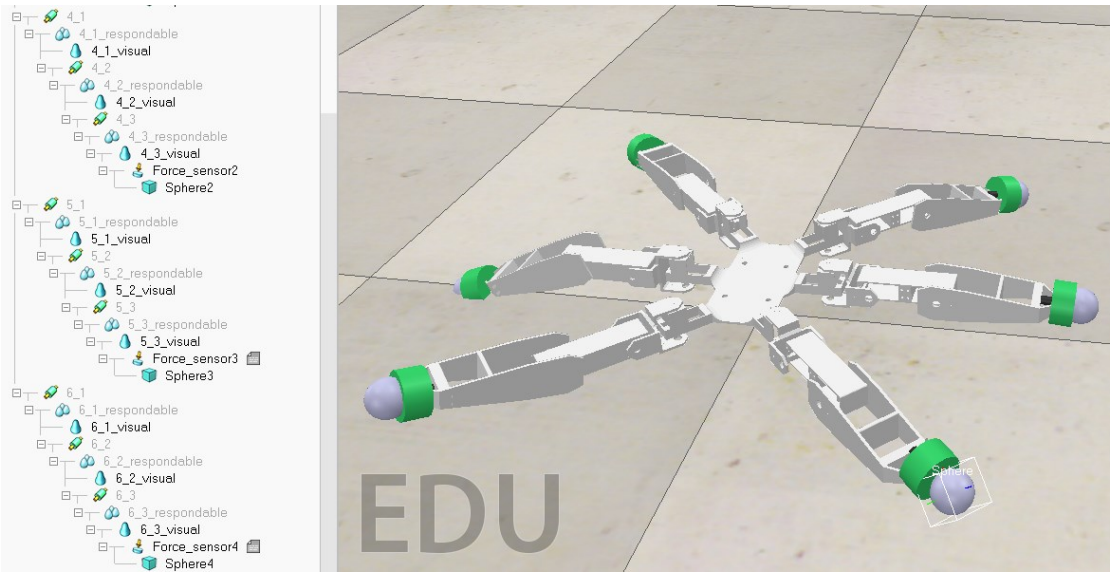

Figure 32

Write the Lua script program, click Run, and you can see the robot's swinging posture, as shown in figure 33, which is the posture in the 5th second of the 10 second simulation.

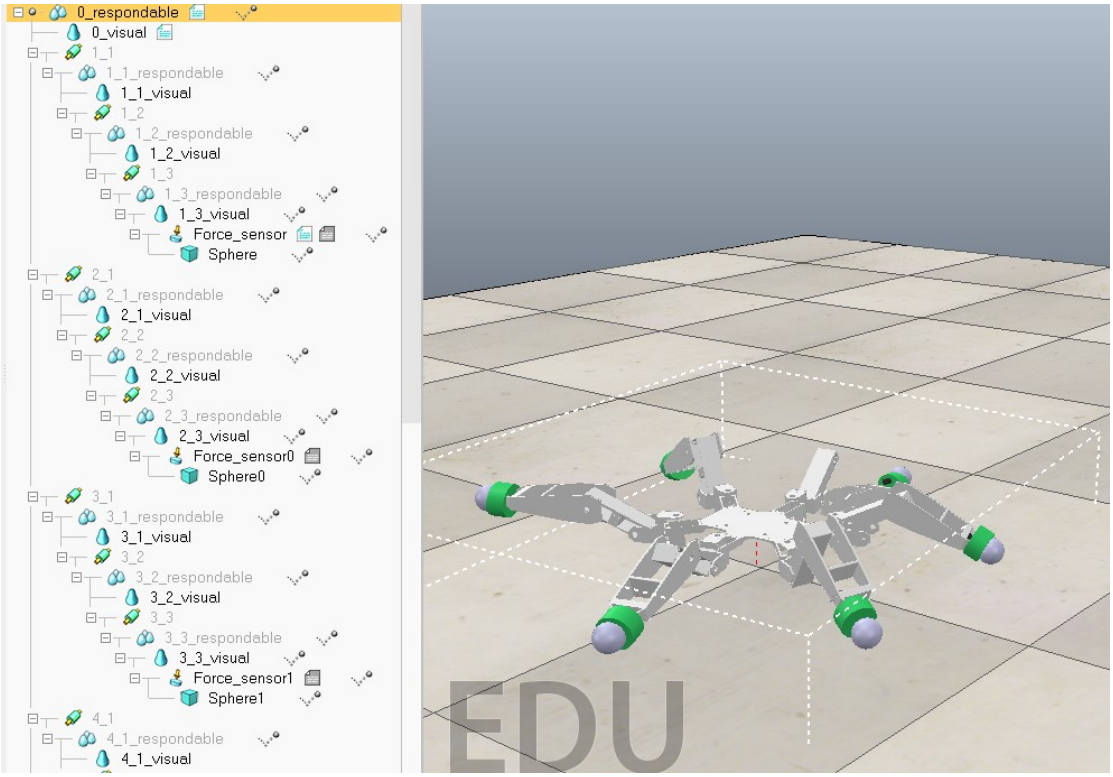

Figure 33

The program mainly includes the following three aspects:

Set Handle Call (Get End Executor)
Set gait parameters (set stride, step height)
Set Gait Motion Cycle (Set Triangular Gait)

4.3. 3D printing

PLA (polylactic acid): It is a biodegradable plastic that is easy to print and use. It has a low melting point and is suitable for most desktop 3D printers. PLA does not produce strong odors during printing and the printing process is relatively quiet. In addition, it has good detail performance, and the printed products have high accuracy, smooth surface, and basically no burrs or layering.

Select a 3D printer produced by FlashForge and download the corresponding printing software.Open the software, in the bottom left corner of the interface, select the machine type as Creator Pro Normal mode, then click on File, load the file, and select the STL format model file. For some models that exceed the maximum size of the printer, first rotate them completely inside. Usually, the bottom layer is printed before the supporting structure and upper layer to ensure the stability of the bottom layer, as shown in figure 34.

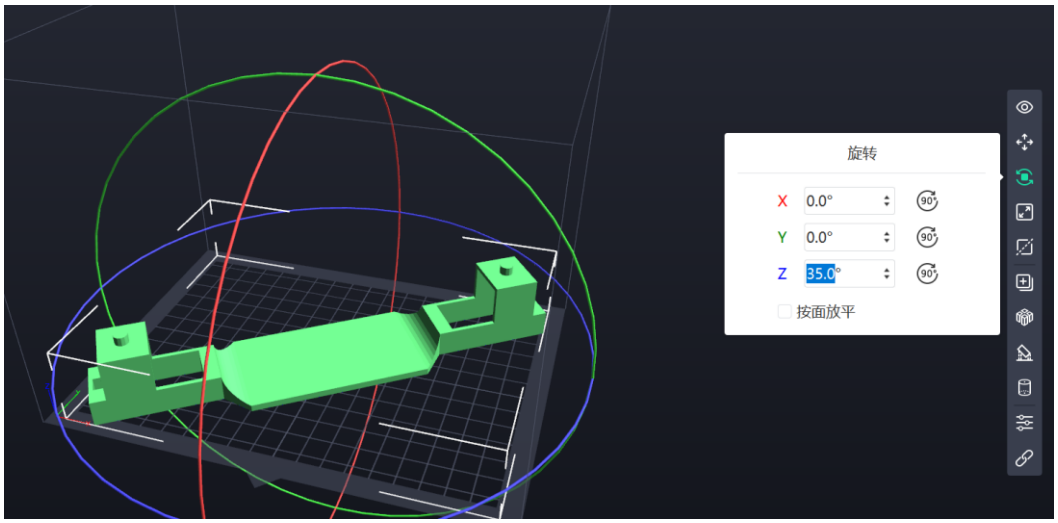

Figure 34

For hollow areas, if the inclination angle is greater than 45 ° (such as a vertical plane), a filling term must be added, otherwise it will cause the model to collapse or tilt. The use of a support model with branch cones can effectively reduce material consumption for subsequent removal while avoiding the impact on the printed product. The default setting is shown in the following figure. Select automatic filling, and then manually fill the unfilled area, as shown in figure 35.

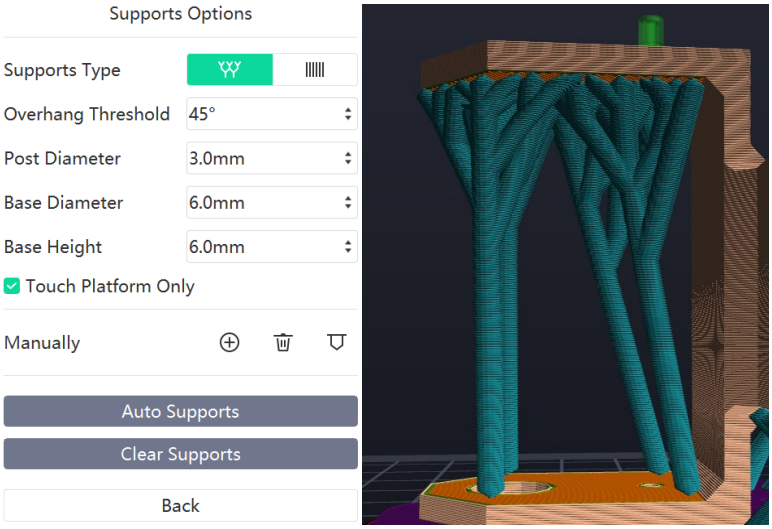

Figure 35

Click the Start Slicing button on the interface to configure printing parameters. Different materials require different printing temperatures, and high or low temperatures can lead to printing problems such as poor adhesion, deformation, or blockage. A good non working area temperature can limited improve the problem of model warping. Printing speed affects printing time and quality. Excessive printing speed may lead to blurry details, while low printing speed can increase printing time. Limiting the printing speed of the first layer can help the model better adhere to the platform. A smaller layer height can improve printing quality, but it will increase printing time. The nozzle diameter directly affects the extrusion amount and detail performance during printing, with common nozzle diameters of 0.4mm or 0.6mm. A higher filling density can increase the strength of the printed product, but it also increases printing time and material consumption. The obtained results indicated that infill density has important role in deformation and strength, but low density and high density infills are cost effective compared to solid components. [13] Although hexagonal filling reduces the adhesion between layers compared to triangular filling, it achieves higher filling strength and shorter printing time. The detailed data is shown in Table 6:

Table 6

Printing temperature (same for left and right nozzles)	210 ° C (non-work area 40 ° C)
PRINT SPEED	Standard mode reference value 60mm/s (first layer 10mm/s)
floor height	0.18mm (first layer 0.27mm)
Nozzle Diameter	0.4mm
packed density	15% (Hexagonal filling)

Setting a smaller model gap can facilitate the stripping of the bottom surface from the model, as shown in figure 36, it can be changed from 0.20mm to 0.15mm:

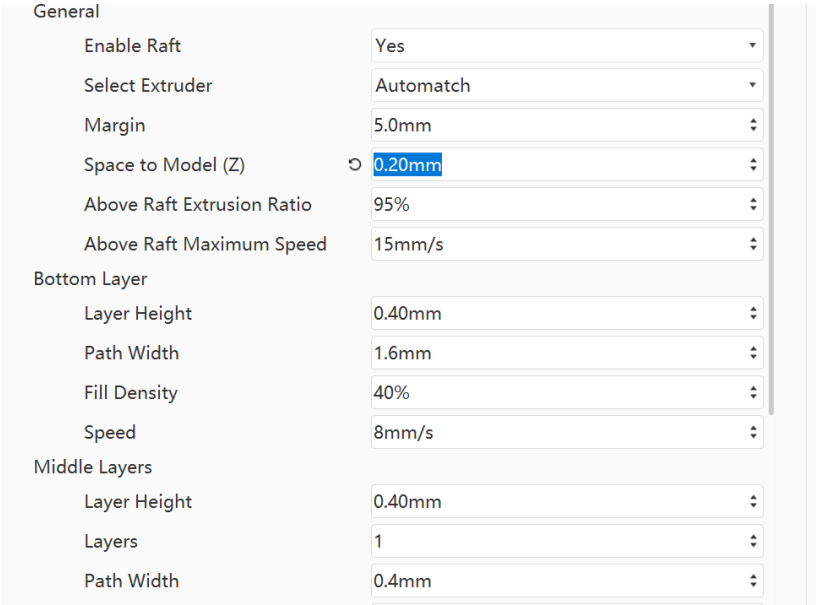

Figure 36

Click Slice, then click Slice Preview to see the detailed operation process. For example, the model file has a total of 199 layers, as shown in figure 37, showing the printing status of the 150th layer and step 700.

Connect the printer and repeat the above steps to print all parts.

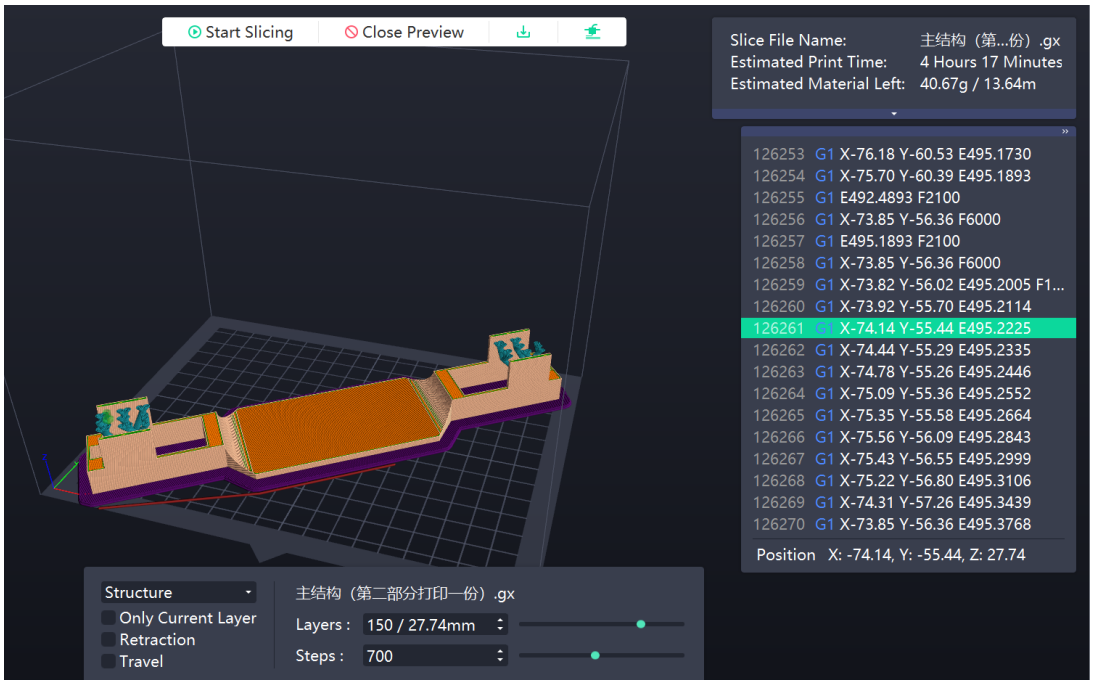

Figure 37

5. Summary and outlook

5.1. Design difficulties encountered

Firstly, the lack of design experience in 3D models leads to a heavier overall quality of the model; And the lack of necessary quality analysis may lead to a slight deviation of the overall center of gravity from the geometric center of the robot, and the support plate can also be replaced by a lighter and stronger buckle design. Secondly, I am not proficient in robot software kinematic simulation and spent a lot of time designing and importing models; Finally, Arduino programming has been debugged multiple times, but there is a lack of experience in optimizing PID parameters, and the trial value method can only be used to ensure the normal operation of the robot.

5.2. Possible error terms

Applying theory to practice always leads to certain deviations. The sources of error may include:

Sensor error: Hexapod robots rely on sensors to perceive the surrounding environment and posture during movement. The accuracy and accuracy of sensors may have certain errors, leading to inaccurate perception data and affecting the motion control and decision-making of robots.

Mechanical structure error: During the manufacturing and assembly process of mechanical structures, there may inevitably be certain machining and assembly errors, resulting in small deviations in the size and position of mechanical components, which affects the stability and accuracy of robot motion.

Control Algorithm Error: Although advanced control algorithms may be used in theoretical design, in practical implementation, the execution of algorithms may be affected by factors such as computational power and sampling rate, resulting in control errors.

Energy limitations: Due to the hexapod robot's reliance on battery supply, insufficient energy or unstable voltage may limit the robot's running time and movement ability.

External interference: External factors such as electromagnetic interference and wireless interference may cause interference to the robot, leading to errors or temporary loss of control.

Environmental factors: Agricultural operations are greatly affected by weather factors, such as rain, snow, strong winds, etc., which may affect the stability of robot movement, while high temperatures and sunlight may cause performance degradation of robots and sensors.

5.3. Further Improvement Areas

a. Due to time constraints, the drawing of solidworks files did not perform stress analysis on the main body and robot legs. Establishing hollow areas on the model can not only reduce the use of materials, but also reduce the overall quality. The crawling motion is driven by a sinusoidal function, and the leg tip rotating posteriorly contacts the ground and generates a frictional force.[14] Conducting friction analysis on the sole of the foot can limit the increase in system load and enhance walking ability.

b. Due to the coupling of dynamics and kinematics in this design, the impact of dynamics can be ignored. But dynamic analysis is an important means to comprehensively understand and optimize the motion and mechanical behavior of robots. Force and moment are important measures that determine body stability, particularly when there is a disturbance.[15] Through dynamic analysis, such as joint torque, inertia force, inertia matrix, etc., it can help optimize motion trajectory, acceleration, and speed control, thereby achieving more accurate and efficient motion. A common method is to use the Lagrangian-Euler method to derive the dynamic model and obtain the Jacobian matrix. One possible matrix is as follows:

$$J = \begin{bmatrix} -(l_1 + l_2 * cos\,\theta_2 + l_3 * cos(\theta_2 + \theta_3) * sin\,\theta_1 & -[l_2 * sin\,\theta_2 + l_3 * sin(\theta_2 + \theta_3)] * cos\,\theta_1 & -l_3 * sin(\theta_2 + \theta_3) * cos\,\theta_1 \\ (l_1 + l_2 * cos\,\theta_2 + l_3 * cos(\theta_2 + \theta_3) * sin\,\theta_1 & -[l_2 * sin\,\theta_2 + l_3 * sin(\theta_2 + \theta_3)] * sin\,\theta_1 & -l_3 * sin(\theta_2 + \theta_3) * sin\,\theta_1 \\ 0 & l_2 * cos\,\theta_2 + l_3 * cos(\theta_2 + \theta_3) & l_3 * cos(\theta_2 + \theta_3) \end{bmatrix} \quad (11)$$

c. Although triangular gait is simple and easy to operate, it has advantages in energy consumption.But, as some studies show, using of one gait is not always effective, and under different environmental conditions, it is, more advantageous, using different methods of movement. [16] Studies have shown that the hexapod robot runs fastest on the flat under tripod locomotion while it runs slowest under quadruped locomotion on the up-slope. [17] In addition, there are differences in contact surface and friction between soil and sand, and changing gait can better adapt to the environment. A more advanced approach is to let the robot determine the terrain on its own and choose the appropriate gait. The gait here is not just about calling and switching the gait on the program, but also the progression over a period of time, which can be the result of the combined action or superposition of multiple gaits.

d. For 3D printing, if conditions permit, higher precision printers and better materials can be used, but the corresponding printing time and cost will also increase.

e. In terms of circuit boards, although botboararduino is a mature integrated hardware, not all functions will be used, and some functions to be implemented may not be possible, such as tracking the rotation angle of each leg using sensors. One method is to customize the PCB board, and one possible solution is shown in Table 7:

Table 7

Size	60*60mm
Number of layers (Including copper coating, window-opening)	2 (Including silk screen printing, and green oil layer)
Thickness	1.6mm
Material quality	FR-4
Line width/line spacing	5mil(Power and ground wires 10mil)
Drill Size	(M3)133.86mil
Number of PWM inputs/outputs	18 (for servo)
Number of analog inputs/outputs	18(for sensors)

f. In the assembly of the entity, the battery and servo wires should be fully considered, that is, in the design of the model, a specific location should be left for packaging and wiring. In practice, agricultural production may come into contact with liquids in large quantities, which requires high waterproof performance. In addition, insulation treatment can also be carried out on the surface of the controller and steering gear without affecting system heat dissipation, to improve security.

g. The automatic discrimination of robots may have errors, leading to missing certain areas or making incorrect choices. A common approach is to add an external interrupt (priority higher than automatic mode), which allows the operator to manually adjust through wireless remote control when the robot makes a

mistake. Common wireless remote controls include zigbee protocol and RF radio frequency, suitable for data transmission of sensors and remote controls. A possible setting scheme is shown in Figure 38:

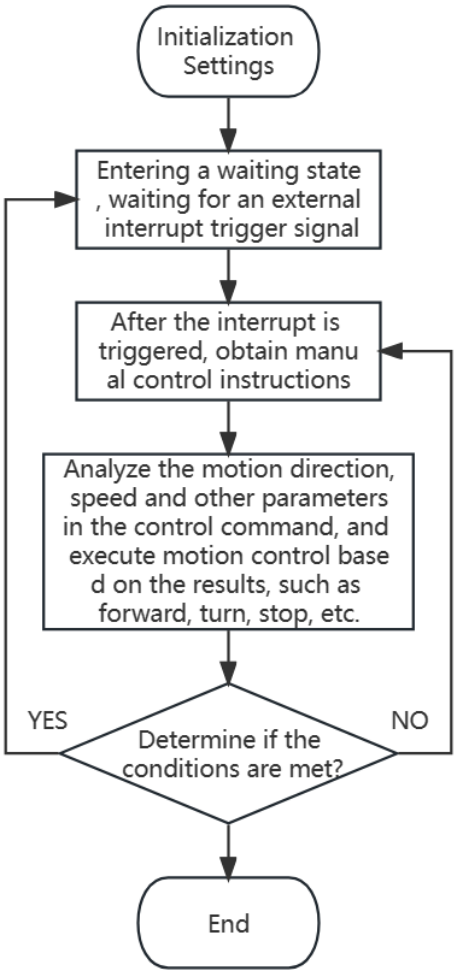

Figure 38

h. In practical agricultural applications, the terrain is not always flat, so it is important to incorporate slope based balance control. This type of control requires fast response speed and high control accuracy, and one possible method is to use reinforcement learning. Locomotion of a hexapod robot can be considered as a Markov Decision Process (MDP), which is described as an agent interacting with the environment in discrete time steps.[17] In a real environment, hexapod robots may fall or collide with other objects. In order to solve these problems, reward functions and probability distribution functions can be set up to make the actions of the hexapod robot constrained by various conditions such as acceleration, speed, and torque, ensuring that the robot safely executes the actions.

i. In agricultural production, the efficiency of robot execution is crucial. Path planning is an important part of autonomous navigation for robots, which can effectively shorten the working time of robots. When there are irregular paths in the generated path, the unnecessary working distance will increase, the number of turns and the turning angle will also increase. [18] Classic algorithms include Dijkstra algorithm, A * algorithm, etc. Currently, common algorithms in the field of robotics include particle swarm optimization algorithm, RRT algorithm, etc. Taking the A * algorithm as an example, f (n) is the comprehensive priority of node n. When selecting the next node to traverse, the node with the highest comprehensive priority (i.e. the lowest value) is always selected.

$$f(n) = g(n) + h(n) \tag{12}$$

Where g (n) is the cost of node n's distance from the starting point, and h (n) is the expected cost of node n's distance from the endpoint, which is the heuristic function of the algorithm. During the operation process, A * algorithm selects the node with the lowest f (n)(highest priority) from the priority queue as next node to be traversed. It comprehensively considers the actual cost and estimated cost of the node through a heuristic function, and advances towards the target direction as quickly as possible during the search process, finding the shortest path. For path planning of hexapod robots, Algorithm A can be applied to graphical representations of maps or environments to help robots find the optimal path, avoid obstacles, and achieve efficient motion planning. It should be noted that heuristic algorithms need to choose appropriate heuristic functions and parameters, otherwise it may lead to results tending towards local optima.

5.4. Development prospects

Overall, hexapod robots will continue to expand their application fields and technological level in future development, gradually meeting the needs of humans for more intelligence and autonomy. There are several possible directions for future development:

Application of precision agriculture: With the continuous promotion of agricultural intelligence in the future, hexapod robots can be used for directional planting, automatic fertilization, pesticide spraying, and other work to improve crop yield, reduce resource waste and environmental pollution.

Disaster rescue and survey: After disasters such as earthquakes, fires, and floods occur, hexapod robots can enter the affected area to help rescue personnel better understand the disaster situation, perform search, rescue, and survey tasks, and reduce casualties.

Extreme environment exploration: In planetary exploration missions, hexapod robots can overcome complex terrain, complete exploration and sampling tasks in steep areas, and provide valuable data for scientific research.

Intelligent logistics and warehousing: In places such as warehouses and factories, high degree of freedom hexapod robots can achieve automated handling, sorting, and storage, improving logistics efficiency and reducing labor costs.

Basic Science and Education: As a representative of complex multi legged robots, hexapod robots are of great significance for the research of robot dynamics, control algorithms, perception technology, and other aspects. These studies can also have a negative impact on biology, helping scientists better understand the behavior patterns of insects and other organisms. Future research can further expand the application of hexapod robots in science popularization for teenagers or programming for children.

5.5. Summary

The kinematics of a hexapod robot is the foundation for subsequent work. By solving forward and inverse kinematics, attitude control, and constraints, it can help understand the motion characteristics of robots, optimize their motion control, and improve their motion performance and response level.

The hardware design of a hexapod robot is the core of the entire design process. In hardware design, it is necessary to consider the structure, materials, sensors, actuators, and other aspects of the robot. Hexapod robots typically use lightweight materials to improve their motion flexibility and load capacity. In the hardware design process, it is also necessary to fully consider the stability, durability, and ability to adapt to complex environments of the robot.

In the design of hexapod robots, 3D modeling and simulation play an important auxiliary role. By using professional 3D modeling software, the structure and components of robots can be accurately drawn, helping designers visualize and optimize their designs. Simulation can simulate the motion performance of robots in different scenarios, identify potential problems before actual manufacturing, and evaluate and improve performance.

The software development of robot is the key to achieving autonomous motion and control of it. Software development involves aspects such as motion planning, path planning, and motion control. With the help of traditional algorithms and programming techniques, hexapod robots can achieve navigation, action execution, and decision-making, thereby completing various tasks in complex environments.

Physical demonstration is the final form of designing a hexapod robot into a real physical entity, showcasing its true motion and function. Through physical demonstrations, it is possible to verify whether the design of hardware and software meets expectations, test the performance of robots in actual environments, and analyze the sources of errors and improvement measures.

Based on the above, the hardware, software, 3D modeling, simulation, and physical demonstration of the hexapod robot design are intertwined, forming a complete design process. The effective combination and optimization of these aspects is of great significance for achieving efficient motion, intelligent control, and widespread application of hexapod robots. Driven by continuous innovation and development, hexapod robots will continue to demonstrate their unique potential and value in multiple fields.

6. References

[1] Meyer, JA., Guillot, A. (2008). Biologically Inspired Robots. In: Siciliano, B., Khatib, O. (eds) *Springer Handbook of Robotics.* Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_61

[2] Escobar, MJ., Alexandre, F., Viéville, T., Palacios, A. (2022). Bio-inspired Robotics. In: Auat, F., Prieto, P., Fantoni, G. (eds) *Rapid Roboting. Intelligent Systems, Control and Automation: Science and Engineering,* vol 82. Springer, Cham. https://doi.org/10.1007/978-3-319-40003-7_8

[3] SHI Zhaoyao, DING Hongyu, WANG Wenguang, LIU Yizhang, HU Yisen, JU Xiaozhu, ZHANG Pan. Design and Experimental Research on the Novel Leg Configuration of Bipedal Robot[J]. Journal of Mechanical Engineering, 2023, 59(1): 103-12.

[4] ZHU Xiaoqing, CHEN Jiangtao, ZHANG Siyuan, LIU Xinyuan, RUAN Xiaogang. Gait Learning of Quadruped Robot Based on Deep Arbitration Strategy[J]. Transactions of Beijing institute of Technology. doi: 10.15918/j.tbit1001-0645.2022.213

[5] Zubairuddin M, Akash T, Prabhakar S, et al. Eight-Legged Robot using Theo Jansen Mechanism[J].

[6] MA Jiliang, PENG Jun, GUO Yanjie, CHEN Xuefeng. Research Status and Development Trend of Wall Climbing Robot[J]. Journal of Mechanical Engineering, 2023, 59(5): 11-28.

[7] JIANG Shuhai, SUN Pei, TANG Jingjing, CHEN Bo. Structural design and gait analysis of hexapod bionic robot[J]. Journal of Nanjing Forestry University (Natural Science Edition), 2012, 55(06): 115-120. doi: 10.3969/j.jssn.1000-2006.2012.06.023.

[8] Claveau, D. (2017). System of 3-D Printed Components for the Rapid Prototyping of Legged Robots. In: Kim, JH., Karray, F., Jo, J., Sincak, P., Myung, H. (eds) Robot Intelligence Technology and Applications 4. Advances in Intelligent Systems and Computing, vol 447. Springer, Cham. https://doi.org/10.1007/978-3-319-31293-4_35

[9] Chen Gang, Jin Bo, Chen Ying Closed loop pose control of hexapod walking robot based on velocity Inverse kinematics [J] Journal of Agricultural Machinery, 2014, 45 (5): 265-270

[10] [Karol Borowski, Andrzej Wojtulewicz, Implementation of Robotic Kinematics Algorithm for Industrial Robot Model Using Microcontrollers, IFAC-Papers OnLine, Volume 55, Issue 4,2022, Pages 248-253, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2022.06.041.

[11] Y. A. Badamasi, "The working principle of an Arduino," 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, 2014, pp. 1-4, doi: 10.1109/ICECCO.2014.6997578.

[12] C. L. C. Bual, R. D. Cunanan, R. A. R. Bedruz, A. A. Bandala, R. R. P. Vicerra and E. P. Dadios, "Design of Controller and PWM-enabled DC Motor Simulation using Proteus 8 for Flipper Track Robot," 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Laoag, Philippines, 2019, pp. 1-5, doi: 10.1109/HNICEM48295.2019.9072736.

[13] Khosravani, M.R., Berto, F., Ayatollahi, M.R. et al. Characterization of 3D-printed PLA parts with different raster orientations and printing speeds. Sci Rep 12, 1016 (2022). https://doi.org/10.1038/s41598-022-05005-4

[14] Miyamoto, N., Kinugasa, T., Amasaki, T. et al. Analysis of body undulation using dynamic model with frictional force for myriapod robot. Artif Life Robotics 26, 29–34 (2021). https://doi.org/10.1007/s10015-020-00610-w

[15] Kao, C. Robert et al. "Reactive Force Analysis and Modulation of the Individual Legs in a Running Hexapod Robot." *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (2019): 370-375.

[16] Gumeniuk, V.A., Chepin, E.V., Voznenko, T.I., Gridnev, A.A. (2020). Reconfigurable Locomotion of Hexapod Robot Based on Inverse Kinematics. In: Samsonovich, A. (eds) Biologically Inspired Cognitive Architectures 2019. BICA 2019. Advances in Intelligent Systems and Computing, vol 948. Springer, Cham. https://doi.org/10.1007/978-3-030-25719-4_16

[17] Ouyang W, Chi H, Pang J, Liang W, Ren Q. Adaptive Locomotion Control of a Hexapod Robot via Bio-Inspired Learning. Front Neurorobot. 2021 Jan 26;15:627157. doi: 10.3389/fnbot.2021.627157. PMID: 33574748; PMCID: PMC7870720.

[18] Tang, Gang et al. "Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment." IEEE Access 9 (2021): 59196-210.