Republic of Yemen

Ministry of Higher Education and

Scientific research

University of Arab Academics University

Faculty of Engineering

The Intelligent Control of a Hexapod Spider Robot

الروبوت العنكبوتي سداسي الأرجل ذو التحكم الذكي

Done by:

Abdulrahman Mujahid Al-Qabsi	202201031
Jalal Hafidh Al-Suwaid	202201083
Jalal Mujahid Al-Hamzi	202201136
Sulaiman Mohammed Al-Shabibi	202201051
Yaseen Mohammed Al-Miqdad	202201099
Younis Fathi Al-Hammadi	202201046

Supervised by

Prof. Dr. Eng. Abdulrahman Mabkhot Al-thanebi

A graduation project document submitted to the department of Mechatronics Engineering in partial fulfillment to the requirements for Bachelor Degree in Mechatronics Engineering at the University of Arab Academics University for Science and Technology

Abstract

This project presents the design, fabrication, and control of a low-cost hexapod robot built on a Raspberry Pi platform for operation over irregular terrain. The mechatronic architecture integrates a lightweight chassis and six 3-DOF legs actuated by high-torque servomotors, enabling ripple and tripod gaits. Perception is provided by an IMU (MPU6050), HC-SR04 ultrasonic ranging, and a Raspberry Pi Camera. A modular software stack converts high-level commands into synchronized joint trajectories via PWM control and provides logging, calibration, and safety interlocks. Development followed a staged workflow—literature review, modeling, simulation, prototyping, and iterative field testing—under local constraints in materials and component availability. The resulting prototype demonstrates stable locomotion, in-place turning, and obstacle negotiation on uneven surfaces with responsive teleoperation. A proof-of-concept vision pipeline based on Haar cascades performs face detection to enable contextual interaction. The main contribution is a locally manufacturable, extensible legged-robot platform suitable for search-and-rescue training, inspection, and hazardous-site exploration, together with an engineering baseline for future work on autonomous navigation, AI-based perception, and multi-robot coordination.

Keywords: Hexapod; Legged locomotion; Raspberry Pi; Gait planning; Sensor fusion; IMU (MPU6050); HC-SR04; Raspberry Pi Camera; Ripple gait; Tripod gait; Haar cascade; PWM control.

Authorization

We authorize University of Arab Academics University for Science and Technology Faculty of Mechatronics Engineering to supply copies of our graduation project report to libraries, organizations or individuals on request.

Student Name	Student No.	Signature
Abdulrahman Mujahid Al-Qabsi	202201031	
Jalal Hafidh Al-Suwaidi	202201083	
Jalal Mujahid Al-Hamzi	202201136	
Sulaiman Mohammed Al-Shabibi	202201051	
Yaseen Mohammed Al-Miqdad	202201099	
Younis Fathi Al-Hammadi	202201046	

Date: …/ …/ ……

<u>Acknowledgment</u>

Before and above all, we would like to record our endless thanks to **Allah** for everything He gives us.

We wish to express our deepest gratitude and appreciation to **Prof. Dr. Eng. Abdulrahman Mabkhot Al-thanebi** for excellent guidance, kind encouragement, scientific advice, helpful supervision, honest cooperation and good wishes which instilled strength in us to make this work possible.

Last but not least, we owe a great deal of gratitude, thanks and appreciation to all members of our families, for their kind support, help and encouragement.

Examination Committee

Project Title: The Intelligent Control of a Hexapod Spider Robot

Supervisor

No	Name	Position	Signature
1	Prof. Dr. Eng.\ Abdulrahman Mabkhot Al-thanebi		

<u>Examination Committee</u>

No	Name	Position	Signature
1			
2			
3			

Department Head

……………………..

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Chapter 1: Introduction

1.1 Overview:

In the history of science and technology, important developments have taken place, the most important of which is the emergence of robotics and its applications, which became important as time progressed. Robots have contributed to many applications, including industrial and agricultural applications and many other applications, which contributed to increasing the efficiency of operations, increasing productivity and reducing the percentage of human errors.A robot is an electro-mechanical system capable of carrying out several operations independently, programmed or controlled by an operator.

Six-legged robots are used in several operations including search and rescue and exploration due to their small size and ability to move in all types of terrain according to appropriate algorithms.

The appropriate number of legs and its ability to move provides balance and fluidity in movement compared to the majority of multi-legged robots, as six-legged robots are considered the fastest among legged robots because they have the best and optimal number of legs to achieve walking speed.

Six-legged robots are superior to wheeled robots, as a six-legged robot can traverse uneven terrain, overcome obstacles, choose anchor points for maximum stability and traction, and the maneuverable legs allow it to rotate in place.

Compared to other multi-legged robots, hexapods offer a higher degree of stability, as they can keep five legs on the ground during movement, and the robot's center of mass remains within the triangle of support created by the movement of the legs, contributing to enhanced stability.Six-legged robots also have high durability if malfunctions or loss of legs can be dealt with by modifying the walking mechanism. This feature makes it possible to use one or more legs as arms to perform certain and sensitive tasks, and thanks to these features, the desire to use six-legged robots is increasing rapidly, and it is important to know the future modifications that will improve their form and function.

1.2 Motivation:

We sharpened our minds during our undergraduate studies, and we wanted to demonstrate this by choosing the Hexapod Spider Robot, and also benefit society. To achieve this, we realized that society, with all its knowledge, is a top priority. There are hidden treasures within us, and tools we seek to use. This is what shapes our society. We strived to alleviate suffering and prevent further disasters or destruction caused by the ongoing war. We also wanted to make its production extremely low-cost so that our government could deploy it widely. In any case, our country could save this robot some money to mass-produce it. Robotics has brought tremendous wealth to technology around the world. Our country strives to advance science to broader horizons, and we, as patriots, seek to share and realize these visions.

1.3 Problem Statement:

We took into consideration many of the problems that engineers faced in designing, manufacturing and programming the hexapod spider robot, and through our work on this project, we solved a number of them. The most important of these problems are:

- Saving energy with six-legged robots isn't easy. One of the problems is how the legs move. When the robot speeds up, the steps don’t always land right. That throws things off and drains more power. It's not just about how it’s programmed the parts themselves bend and flex, and even small wobbles in the system can mess things up. Heat makes it worse. If the robot walks for a while or carries weight, the motors start getting hot. Then they pull more power just to keep going. That keeps building until something gives. The motors could just stop. Even when it’s standing still, the system still burns power to stay on. Put all that together, and it’s clear: unless the motors are swapped for something better, the robot won’t get much more efficient.

- Robots typically run at a really low level of efficiency it can go as low as 1.1%, and it’s rare to see them above 2%. This slump mostly stems from old-school mechanical flaws and control hiccups you don’t find in animals or humans. For instance, parts can be a bit loose, joints may stick, motors leak energy, and the frame can bend or distort slightly. Even the robots’ paths get a bit off sometimes because their control system isn’t flawless.

- Walking robots with six legs often face a few recurring issues. One of the big ones is that they don’t move very fast, and their joints tend to shake or feel unstable while walking. This usually has to do with how their legs are timed and how they shift between steps. In many cases, the common walking patterns, like the tripod gait, don't handle these transitions very well. To improve performance, an alternative approach known as the “inverted gait” has been proposed. Instead of beginning with three legs in motion while the other three remain grounded, this method starts with all six legs in contact with the ground, providing a more stable launch posture. Shortly after initiating movement, a group of legs performs a slight backward swing while the rest provide support. This inverted start shifts the robot’s center of mass in a way that creates more room for forward hip movement. As the motion continues, the supporting legs alternate with the swinging ones in a rhythm that allows forward propulsion with less lateral drift. What makes this way of walking better is that the robot can stretch its steps a bit more, and it doesn’t waste as much time moving each leg group. That means it walks more smoothly, changes steps faster, and stays more steady even if the surface isn’t level.

- Working with deep learning models on tiny devices like the Raspberry Pi 3 isn’t exactly smooth sailing. They’re fine for basic jobs, but once you throw in model training, the cracks start to show. The hardware just doesn’t have the strength or space to manage that kind of demand. Things often slow to a crawl, and in some cases, the board struggles to keep up especially with images or heavier models. Some models are described as lightweight, but that doesn't mean they'll perform well memory runs out quickly. As a result, using the Raspberry Pi 3 for actual training becomes difficult, especially when using images or when higher power is required.

- Getting accurate angle readings on a hexapod robot isn’t always easy especially when the robot’s body is lifted higher off the ground. The further it gets from the surface, the more errors start to creep into the pitch, roll, and yaw measurements. This issue often comes down to two things: the IMU sensor might not be sitting flat, and the robot’s frame isn’t perfectly aligned. Even though the errors are fairly small, they still show up when you compare them to the robot’s actual movement. On average, pitch readings were off by about 0.67%, roll by 1.01%, and yaw by 0.79%.

1.4 Project Objectives:

Right now, we are working on a robot that moves using six legs kind of like how a spider walks. we chose this approach after seeing that many regular walking robots have trouble keeping their balance, especially when the ground isn't flat. To solve that, we came up with a simple leg structure that moves pairs of legs together for better control. we are also using a diagonal walking style to help it stay balanced whether it's going forward, turning, or avoiding stuff in the way. As we go through building and testing, we'll watch how the robot's body reacts how the legs move together and how its balance shifts. we'll also try it out on bumpy ground to see how well it holds up.

- My main aim is to build a robot that's steady, easy to use, and works well in real places whether that's for save life, exploration, or just trying out new concepts.

- The second part of the project is about helping the robot walk smoothly and stay steady, whether the ground is rough or flat. Some robots tend to shake or slow down on tricky surfaces, so we are using a different way of walking that lets the legs stretch more and switch faster. That should help reduce shaking and improve how it moves overall.

- The third goal is to let the robot recognize human faces using a camera. It works by picking out certain features from a face in the video and matching them with saved ones in its memory. This helps the robot ignore anything that isn't a real face. To handle this part, the robot uses something called Haar Cascade. It checks the camera feed frame by frame, looking for any faces. If it finds one, it puts a box around it and saves that frame for later use. This method uses four tools to make it work: Haar features, something called an integral image, Adaboost, and a chain of filters. This part is still being tested and developed.

- One of the main goals is to get the whole robot system working on a Raspberry Pi, without needing any fancy or expensive computer. Everything from walking to seeing faces should happen right on the Pi in real time. The idea is to keep the system simple, affordable, and smart enough to react instantly while the robot is moving.

- Making the Robot Strong and Ready for Any Ground. While building this robot, we really just wanted to make sure it wouldn't break if it moved around. Whether it was sand, rocks, or bumpy ground-we needed it to hold itself together and keep moving. we didn't use anything fancy. Just went with what felt solid. we played around with different metals, and honestly, AL6061-T6 just worked better than the others we messed with. So, we just kept using it. Putting it together took time. One little thing-like placing a sensor slightly off could make it walk weird. Same with the motors. They had to be just right, or the feet would land all over the place. So, we added some limits in the code to keep them from hitting other parts. Another thing we looked at was the type of ground it would move on. we didn't want it slipping or sinking. So, we made the feet wider to help spread the weight. That actually worked pretty well on soft stuff like loose dirt. After trying it in different places, we felt good about how it moved. No falling, no tripping. Even in tricky spots, it stayed steady. That was the goal.

- Letting the Robot Grow Later On. Back when we were putting the robot together, we didn't plan everything all at once. we figured, maybe down the road, we'd want to try new things with it like letting it move on its own, or carry something heavier. So, we left some wiggle room. Nothing fancy. we didn't box myself in. If something new comes up later, we can still make changes.

1.5 Project Scope and Limitations :

1.5.1Project Scope :

Letting the Robot Do More Than Just Walk. At the beginning, we weren't just trying to get it to move or run tests. My goal was to make something useful in tough spots where wheels just don't work-like ruins or bumpy ground. That's why we gave it six legs instead. It helped with balance and made it better at moving over rough surfaces. we thought about all the places this robot could go. It could help during rescue efforts, go into caves, or handle sandy ground. we also pictured it helping with army work-like checking an area or carrying gear or even being used on construction sites where people can't always reach. It might also be helpful at home or for collecting data. To make all that possible, we gave it different ways to walk so it could adjust to the ground. we made sure you can add stuff later-like smarter sensors or better code without tearing it apart. we used simple boards like Arduino and

Raspberry Pi, and it can be controlled through a phone or web too. I also tried to use cheap parts, so building more of it later wouldn't cost too much.

1.5.2Project Limitations :

As we worked on the robot, things didn't always go how we hoped. A few bugs were small and easy to fix, but others stopped me for days. I didn't expect the 3D-printed pieces to be so tricky. Most of them needed sanding or fixing just to fit the way they should. The steel we used made the frame stronger, but it also added weight-about 2.8 kg in total-which made it sink more on soft surfaces. Fitting the parts together took more effort than we thought. Some holes were off, and a couple of joints bent weirdly. Also, the motors aren't perfect. There's some looseness in the joints and friction that messes with how it walks. The electronics were a bit exposed too, which could be risky if the robot touches water. On the sensing side, the ultrasonic sensors were slow and not great at spotting small or shiny things. The camera didn't show much unless the robot turned its whole body. And even with all that, there were blind spots it just couldn't see. What we tested in simulation didn't always match real-life performance either. Programming was another challenge. The boards we used like Raspberry Pi can only handle so much. Writing smooth walking algorithms and making the robot avoid stuff without glitches took a lot of time. Some methods like PID worked okay, but weren't always the best for this kind of robot. Planning paths in complex spaces was slow and didn't always give the results we wanted. Time was also a factor. we couldn't test everything we wanted, and there's still a lot left to improve. It's just a prototype for now. Things like saving energy during movement or making it recover from a failed leg still need more work.

1.6 Project Methodology: Design and Development of a Hexapod Robot:

1.6.1Bio-inspired mechanical design :

When we set out to build a six-legged robot, we drew inspiration from the way insects and small crustaceans move through their surroundings. Those creatures spread their weight across multiple limbs, which allows them to scramble over obstacles without tipping over. The classic tripod stance two legs on one side and one on the other gives them a steady base as they lift and swing the remaining legs. Translating that idea into our own machine meant adding two extra legs compared with a quadruped so the body could stay balanced even while a leg is in mid-air.

1.6.1.1 Structural layout and materials :

To bring this concept to life, we designed each leg with three joints hip, thigh and shank so the robot can lift, extend and place its feet wherever needed. We paid close attention to the range of motion so that the legs would not clash with one another and could reach far enough to step over obstacles. Making the legs hollow reduced their weight, and broad, padded feet helped spread the load and prevent the robot from sinking into soft ground. Choosing materials was just as important. We opted for a mix of aluminum, ABS plastic, PVC sheets and some advanced composites to give us a chassis that was both light and stiff. Those choices also stand up well to corrosion, which is crucial if the robot is to be used outdoors. We sketched and refined the entire design in three-dimensional CAD software, then checked the stresses and clearances with multibody simulation tools. To decide on the best combination of link lengths and leg shapes, we compared different variants using design-for-quality methods and evolutionary algorithms, weighing up stability, payload capacity and ease of fabrication.

1.6.2 Kinematic and dynamic modelling:

After settling on the mechanical layout, we turned to mathematics to make sure the robot would move as intended. We used standard robotics techniques to derive equations that relate each joint angle to the position of the foot in space and vice versa. These kinematic models let us calculate how to place each leg tip wherever we need it while keeping within the limits of the joints. We also built a dynamic model that takes into account the mass of each link, their inertia and the loads they carry. That model allows us to estimate how much torque the motors will need to deliver to support the robot and its cargo. Running these models in simulation gave us confidence that the robot would remain stable as it walked and let us tune the design before building anything physical.

1.6.3Gait planning, stability analysis and control :

1.6.3.1 Gait patterns :

Walking with six legs gives us several different ways to coordinate the limbs. The fastest pattern keeps three legs on the ground at all times, with the remaining three swinging forward in unison; this is the same pattern used by many insects. For rough terrain, we can slow things down and move one leg at a time so that five legs always touch the ground, trading speed for extra security. Other patterns fall somewhere between these two extremes, adjusting how long each leg stays in contact with the ground to suit the terrain. We also experimented with an "inverted" version of the tripod pattern, which increases the hip's range of motion and allows longer steps when crossing very uneven surfaces.

1.6.3.2 Stability criteria and optimization:

To stay upright, the robot's weight must always be supported by the legs that are on the ground. In practice, we watch where the robot's center of mass falls relative to the area outlined by those legs and plan the timing of each step to keep that point well inside that polygon. We measure how close the center of mass gets to the edges of this area and use that margin as a safety indicator. By tweaking step length, the sequence of leg lifts and the fraction of each walking cycle that each leg spends on the ground, we can keep that margin comfortably positive even when walking across slopes or gaps. Widening the stance increases stability, but if we spread the legs too far, we make turning harder, so we balance these effects when choosing gait parameters.

1.6.3.3 Control Strategies :

We built the control system in layers, starting with the basics. At the lowest level, each motor uses a simple feedback loop to reach its target angle and reduce unwanted vibrations. On top of that, we added a gait controller that keeps all six legs moving in sync based on the walking pattern we choose. It also adjusts the timing of each step to make sure the robot stays balanced and centered. To make the movements feel smoother and more natural, we tested advanced methods like fuzzy logic and neural networks these helped correct small model errors and made the leg motion more fluid. We also used biologically inspired central pattern generators to produce natural, rhythmic movement patterns. Sitting at the top of the system is a planner, which turns user commands into actual walking paths. We even explored reinforcement learning letting the robot "learn" how to change its walking rhythm when it runs into obstacles or damage. To help with balance, we added sensors like IMUs and force sensors that track the robot’s orientation and how much force each leg is putting on the ground. That data lets the robot adjust its posture and joint stiffness in real time, which is especially useful on uneven ground.

1.6.4 Hardware Implementation :

1.6.4.1 Fabrication and Components:

Once our designs and control models were solid, we started building. The frame and legs were made using 3D printing and laser cutting a fast way to create detailed parts and test different ideas. Every joint has its own servo motor, carefully picked for its strength and compact size. The brain of the robot is a combination of an embedded microcontroller and a small onboard computer. These work together to run the control software and connect with the motors, batteries, and sensors. For sensing the environment, we added things like ultrasonic distance sensors, temperature sensors, and even microphones depending on the task the robot is doing.

1.6.4.2 Software and User Interface :

We wrote most of the control software in Python, choosing open-source tools to keep things lightweight and flexible. A Raspberry Pi sits at the core of the system, handling everything from movement to communication. To make the robot easy to operate, we connect over Bluetooth or Wi-Fi. With just a few taps, you can change the robot's walking pattern or steer it wherever you like, we also added a simple live video from the robot's camera, so whoever is controlling it can see exactly what the robot sees especially useful in tight spots or unfamiliar terrain.

1.6.5 Testing, Validation, and Performance Evaluation:

Before letting the robot loose in the real world, we spent plenty of time running simulations. These virtual tests made sure that the legs never collided, the motors could support the weight, and the robot stayed balanced even if its center of mass shifted while moving. Once we were confident with those results, we built an actual prototype and took it for a spin. We had it walk forward and backward, turn around in place, and tackle uneven ground. To find out how much it could carry, we added extra weights and measured how much power it used at different speeds. We even made it climb ramps to prepare it for potential tasks like search and rescue missions or exploring rugged terrain. During these tests, we kept track of how stable it remained, how accurately it followed commands, and how efficiently it used energy. If something didn't work-like a part bending too much or a motor getting too hot-we didn't ignore it. We went back, made the necessary fixes, and continued improving the design.

1.6.6 Document Organization:

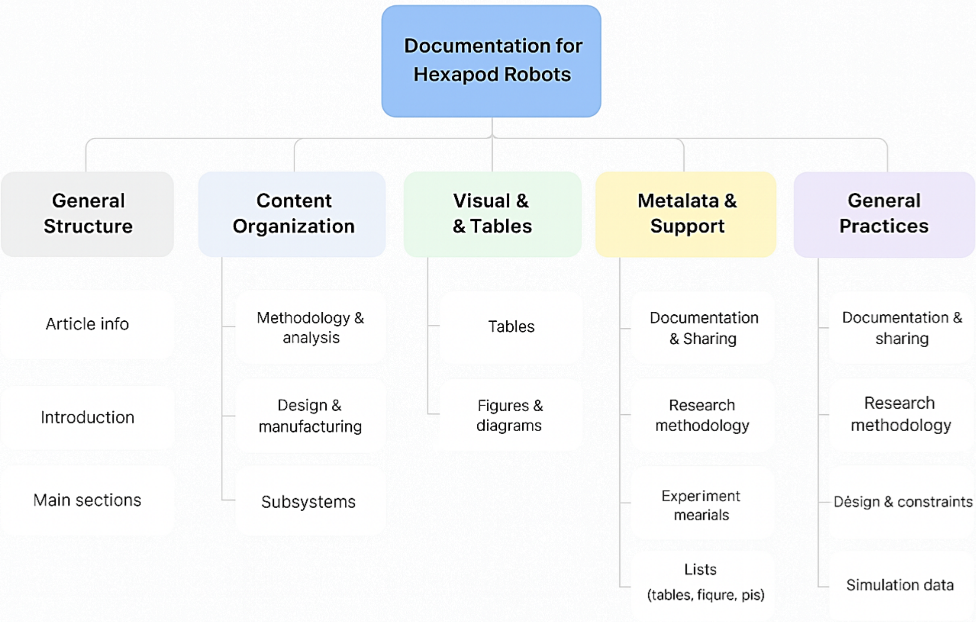


Figure 1. Documentation for Hexapod Robots

Chapter 2

Background and Literature Review

Chapter 2: Background and Literature Review

2.1 Introduction:

In this chapter, we will present a summary table of the previous studies that we relied on , mentioning the theoretical and practical aspects of each study.

2.2 Previous Study Schedule:

2.2.1 Rancang Bangun Kendali Robot Hexapod Menggunakan Smartphone (Bandung State Polytechnic of Electronics – 2023):

Table 1. Summary of Rancang Bangun Kendali Robot Hexapod Menggunakan Smartphone Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
he study aims to design a six-legged hexapod robot controlled remotely using a smartphone via Bluetooth or Wi-Fi. The robot was built using an ESP32 module and an IMU sensor to manage movement and balance. The walking mechanism adopted a tripod gait pattern to ensure stability and smooth directional control.	The study focused on the mechanical movement of the six legs using three degrees of freedom per leg. It applied inverse kinematics modeling and PID control algorithms. The robot's stability was monitored using a 6-DOF IMU sensor (pitch, roll, yaw).	The robot used 18 servo motors distributed over the six legs. Experiments were conducted for forward, backward, and turning motions over a 5-meter path. A mobile app was used to send directional commands and measure trajectory accuracy.	Results showed that the robot moved with good stability, with a minor deviation of 7% to 9% in straight-line walking over 5 meters at a speed of 20 cm/s. The PID controller improved movement accuracy and balance.	Recommended improvements include adding smart navigation features such as camera vision or ultrasonic sensors, optimizing servo response, reducing power consumption, and redesigning the structure for lighter weight and modular expansion.	– Power consumption was not deeply analyzed. – Terrain adaptation and slope performance were not tested. – No computer vision system was implemented.

2.2.2Spider Robot with Four Legs Using PIC16F877 Microcontroller

(Palestine Polytechnic University – 2022):

Table 2. Spider Robot with Four Legs Using PIC16F877 Microcontroller Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
The study focuses on designing and building a spider-like robot with four legs, simulating the real structure of spider joints and limbs. Each leg is powered by servo motors and controlled using the PIC16F877 microcontroller. The project primarily aims to physically construct the model and achieve basic walking motion.	The theoretical framework covered the classification of robots and their mechanical structures, especially those with articulated multi-leg designs. It analyzed movement through joint configuration and servo motor angles, targeting a realistic spider gait simulation.	The robot was built manually using a hexagonal body frame with four legs, each having three joints. All mechanical parts were designed digitally and assembled physically. A control circuit was implemented using MikroC PRO for PIC, with PCB printed and soldered manually. Servo motors were tested via parallel port, and motion was verified.	The robot successfully mimicked legged locomotion, representing one of the first multi-legged robots developed at the university. The use of the PIC16F877 microcontroller proved functional in basic walking commands. Initial tests confirmed servo response, with improved results after switching power supply from 9V battery to 12V adapter.	Future improvements include: – Adding antenna simulation. – Enabling wall/obstacle climbing. –Programming interactive leg movement like dancing or obstacle reaction.	– Limited to 4 legs (not full hexapod). – No use of wireless communication. – No sensor integration (e.g., IMU or camera). – Limited motion scenarios and power tests.

2.2.3 Design and Simulation Analysis of a Hexapod Bionic Spider Robot

(Nanjing University of Science and Technology – 2021**):**

Table 3. Design and Simulation Analysis of a Hexapod Bionic Spider Robot Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
This academic paper focuses on designing and simulating a bionic hexapod robot modeled after spider anatomy. It emphasizes the replication of biological movement and terrain adaptability through a software-only model. Using tools like ADAMS and SolidWorks, the researchers evaluated joint force, body stability, and motion accuracy under different surface conditions. No physical prototype was built; the study remained within simulation.	The study developed detailed 3D models to simulate motion mechanics under varying terrain inclinations. Each leg had six degrees of freedom for more precise adaptation. Mathematical modeling focused on inverse kinematics, center-of-gravity tracking, and force distribution during motion. Terrain slope and robot balance were analyzed computationally.	The robot was designed entirely in software using CAD and dynamic simulation tools. No actual motors or sensors were used. Simulated motion included stepping, climbing inclines, and adjusting to unstable platforms. Energy efficiency, torque requirements, and joint angles were all tested through simulations.	The simulations showed promising balance and terrain response. The robot was able to walk stably across inclined planes without losing alignment. Torque distribution was well-optimized, minimizing power spikes. However, as no real robot was built, real-world friction, terrain variation, and sensor noise were not accounted for.	The authors recommend building a physical prototype to validate simulation data. Brushless motors and real IMU sensors should be included for smoother gait transition. Terrain sensing should be added to adjust gait dynamically, and battery efficiency must be tested under real loads.	– Entirely virtual; lacks physical validation. – Excellent theoretical insight. – Software outputs may differ from reality. – High fidelity model but pending real-world proof.

2.2.4 Spider Robot (Palestine Polytechnic University – Approx. 2012):

Table 4. Spider Robot (Palestine Polytechnic University) Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
This study presents the complete design, construction, and basic operation of a four-legged walking robot inspired by spider locomotion. It explores how to simulate biological movement using simple components such as servo motors and a basic microcontroller. The focus was educational, to provide students with an accessible model for understanding motion mechanics without requiring complex electronics or control theory.	The project was theoretically grounded in studying the gait patterns of spiders and translating them into programmable sequences. It explored fixed-angle servo rotations to simulate tetrapod movement using simple timing logic, aiming to minimize algorithm complexity and cost while maintaining balance and coordination.	Practically, the robot was constructed using lightweight plastic and aluminum for stability and portability. Each leg had two degrees of freedom, with hard-coded servo angles. It lacked sensors or feedback systems, and all actions were manually programmed without adaptability to environmental changes.	The final prototype achieved stable motion on flat and uniform surfaces, capable of forward and backward movement. However, it lacked the ability to respond to obstacles or navigate uneven terrain. The mechanical structure proved durable, but overall functionality remained basic due to limited logic.	To enhance functionality, the study recommended adding sensor modules for obstacle detection, implementing more dynamic control algorithms, and increasing the number of legs to six for better balance and terrain adaptability. Wireless communication could also be explored.	– Fixed-angle gait logic limits adaptability. – Educational use only. – No remote control or autonomous behavior. – Simple but useful for foundational learning.

2.2.5 Low-Cost Hexapod Robot for Educational Purposes (Unspecified – 2021):

Table 5. Low-Cost Hexapod Robot for Educational Purposes Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
This research focuses on creating a budget-friendly hexapod robot aimed at educational use. The platform enables students to experiment with basic robotics principles including kinematics, servo control, and walking algorithms. It is designed to provide hands-on learning for understanding the fundamentals of robot mechanics and embedded systems without the need for expensive components.	The theoretical framework simplifies the concepts of inverse kinematics, basic gait patterns (such as tripod gait), and the logic behind servo actuation. It emphasizes educational delivery by breaking down complex ideas into modules suitable for beginners and students.	The robot is built using 6 legs and 12 servo motors (2 per leg), all managed through an Arduino-compatible board. Movements are controlled using pre-defined routines, with a user interface designed for ease of use. The chassis is modular, allowing easy component replacement and upgrading.	The prototype performed well in classroom settings, where it helped students grasp key robotics concepts. Its walking behavior was stable and consistent. Evaluation indicated a strong correlation between hands-on interaction and student understanding.	Suggestions include enhancing modularity for leg replacement, integrating Bluetooth or WiFi for remote control, and developing a drag-and-drop programming interface to support visual learning environments.	– Designed purely for educational contexts. – Not tested in rugged or outdoor terrain. – Gait control is fixed, with no learning feedback loop.

2.2.6 Design and Implementation of Spider Fire Detection Robot

(Politeknik Negeri Ujung Pandang – 2020**):**

Table 6. Design and Implementation of Spider Fire Detection Robot Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
Table This study focuses on building a hexapod robot capable of detecting fire sources using flame sensors. The robot is programmed with autonomous behavior and is controlled via an Arduino UNO. Its design mimics spider-like motion using a tripod gait to maintain balance while navigating toward fire. The robot targets early warning systems for indoor environments.	The research presents principles of gait control in multi-legged robots, fire detection mechanisms using flame sensors, and sensor-based automation. It also discusses real-time response behavior, the influence of sensor angles, and the calibration required for reliable detection.	The robot consists of 12 servo motors (2 per leg) connected to an Arduino UNO, and flame sensors for fire detection. Upon identifying a flame source, it adjusts its direction and walks toward the heat. The frame is built from acrylic for lightweight design, and power is provided by USB or a portable battery pack.	The robot efficiently recognized flame sources from a 50 cm distance and responded by turning and moving toward them. Its gait remained stable, and the sensors worked effectively in low-light conditions. Testing confirmed successful activation in simulated fire scenarios.	Recommendations include improving the detection range using higher-quality sensors, incorporating warning signals like sound or lights, and upgrading the microcontroller for more efficient processing. Integration with a central alarm system could increase utility.	– The study was limited to laboratory conditions. – No temperature or smoke detection was added. – Navigation was linear with no map awareness.

2.2.7 Design and Simulation of Hexapod Robot Using Arduino and WiFi

(IJSRD, India – 2020**):**

Table 7. Design and Simulation of Hexapod Robot Using Arduino and WiFi Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
This research explores the development of a hexapod robot using Arduino Uno, focusing on wireless control via WiFi. The robot is built to simulate six-legged walking motion using a tripod gait algorithm and to respond to nearby obstacles using ultrasonic sensors. The design aims to demonstrate cost-effective locomotion, making the robot ideal for educational and prototyping applications.	The theoretical foundation is based on the mechanics of hexapod movement, focusing on tripod gait control. It delves into servo positioning, kinematic coordination, and sensor-based path adaptation. Concepts such as DOF (Degrees of Freedom), sensor integration, and walking stability are key parts of the theory.	The system consists of 12 servo motors (2 per leg), an Arduino Uno microcontroller, and WiFi-enabled control. The robot is programmed to execute forward movement and turning, while detecting objects using ultrasonic sensors. Construction materials are lightweight, and the setup is powered via rechargeable battery.	The robot successfully demonstrated stable forward movement and basic obstacle avoidance. It was able to detect objects within a range of 10–15 cm and adjust its path accordingly. The WiFi control module functioned effectively, although response latency was slightly noticeable.	It is recommended to upgrade the servo motors to handle more load, enhance the obstacle detection system by integrating multiple sensors, and explore battery optimization for longer operational time. More advanced path-planning algorithms could also be added.	– No computer vision or camera was used for navigation. – Movement restricted to pre-defined gait paths. – Real-world terrain testing was minimal.

2.2.8Design and Development of Spider Hexapod Robot for Movement and Obstacle Avoidance (Udayana University, Indonesia – 2021):

Table 8. Design and Development of Spider Hexapod Robot for Movement and Obstacle Avoidance Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
The study explores the design and implementation of a spider-like hexapod robot using Arduino Mega and ultrasonic sensors. The robot aims to simulate spider movement using tripod gait and avoid obstacles in real time using sensor feedback.	The study is grounded in spider locomotion analysis, tripod gait theory, and ultrasonic sensing. It also addresses motor control logic and DOF structure (3 DOF per leg) necessary for stable motion and obstacle navigation.	The robot uses 18 servo motors (3 per leg), controlled by Arduino Mega. It includes three ultrasonic sensors for obstacle detection and avoidance. The legs were fabricated from light aluminum and PVC. The robot's gait is pre-programmed with basic directional movement.	The robot successfully demonstrated forward walking, turning, and obstacle avoidance. It reacted efficiently to objects within 20 cm, using left or right turns. Its motion was more stable on flat surfaces.	Suggested improvements include using more advanced sensors (e.g., IR, camera), implementing terrain adaptation, and using wireless remote control. Enhanced structural stability was also recommended.	– Basic prototype, limited testing. – Sensors limited to short-range detection. – No vision or feedback learning systems.

2.2.9 Design and Development of Hexapod Robot Using Smartphone

(Politeknik Elektronika Negeri Surabaya – 2023**):**

Table 9. Design and Development of Hexapod Robot Using Smartphone Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
This study focuses on developing a hexapod robot controlled remotely via smartphone using ESP32. The robot integrates an IMU sensor for balance and applies a tripod gait for walking. It targets improved stability and motion accuracy through sensor feedback and PID control.	Theoretical foundation includes inverse kinematics for six-legged motion, tripod gait theory, and implementation of PID control to correct deviations using IMU sensor readings (gyroscope and accelerometer).	Practically, the robot uses 18 servo motors (3 per leg) and is controlled through a custom mobile app over Bluetooth or Wi-Fi. IMU data ensures balanced locomotion. Movement was tested over 5 meters with a max speed of 20 cm/s.	The robot maintained walking stability with only 7–9% deviation from the intended path. The tripod gait provided consistent performance. IMU integration significantly improved balance in motion.	Future suggestions include integrating a vision-based navigation system, optimizing power efficiency, enhancing structural design for modularity, and using terrain-adaptive gait.	– Power consumption was not deeply studied. – Terrain variety not tested. – Vision systems not yet implemented.

2.2.10 Hexapod Robot Using Arduino Uno and Ultrasonic Sensor

(International Journal for Scientific Research and Development (IJSRD), India – 2021**):**

Table 10. Hexapod Robot Using Arduino Uno and Ultrasonic Sensor Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
The study aims to design and implement a hexapod robot that mimics insect locomotion, using Arduino Uno and ultrasonic sensors for obstacle detection. The robot adopts a tripod gait and emphasizes balance, forward/backward motion, and environmental interaction through real-time sensing.	The theoretical analysis focused on gait patterns (tripod gait), DOF considerations (2 DOF per leg), and ultrasonic sensing for distance measurement. It also reviewed servo control principles and basic inverse kinematics for movement sequencing.	The robot was constructed using acrylic material, with 12 servo motors (2 per leg), controlled by an Arduino Uno. An HC-SR04 ultrasonic sensor was mounted at the front for obstacle avoidance. The gait was programmed to respond to detected distances, stopping or turning when obstacles were near.	The robot successfully avoided obstacles using ultrasonic input and maintained balance while walking forward. It adjusted direction when objects were within a 10 cm range. The performance was consistent on flat surfaces and showcased effective coordination between sensors and motion.	Future improvements include increasing the number of sensors for 360° detection, integrating wireless control, and testing on uneven terrain. Better power management was also suggested for longer operation.	– Only one ultrasonic sensor was used. – No wireless features implemented. – Movement limited to basic directions (forward, stop, turn). – Tested only on smooth surfaces.

2.2.11 Design and Simulation Analysis of Hexapod Bionic Spider Robot

(Hohai University, China – Journal of Physics Conference Series, 2019):

Table 11. Design and Simulation Analysis of Hexapod Bionic Spider Robot Study

Abstract	Theoretical Aspect	Practical Aspect	Results and Findings	Recommendations	Notes
This study proposes a bionic hexapod robot inspired by insect physiology, aiming to design a spider-like robot that balances flexibility and stability. The structure of the body and six legs was modeled and optimized to reflect natural motion. The research includes an analysis of joint load, structural materials, and motion efficiency.	Focused on bionic principles, the study analyzed the physiology of spiders and applied it to mechanical structure design, including a trunk and three-joint legs (coxa, femur, tibia). It explored optimal ratios of leg segments and mass distribution to improve stability during locomotion.	The robot was simulated using SolidWorks. Each leg has three joints and was modeled to maintain realistic movement. Materials were selected to reduce weight (ABS for the body, aluminum alloy for joints). FEM analysis was performed to test joint stress and displacement, confirming suitability of materials.	The analysis showed that the robot design offers optimal stability with reduced weight and sufficient flexibility. Torque and stress on hip and knee joints were within acceptable limits, and design improvements like limb ratio (0.455:0.455:0.09) enhanced motion performance.	Suggested future steps include integrating sensors for terrain feedback and expanding simulation to dynamic walking on uneven surfaces. FEM optimization of all joints was recommended.	– Study was purely simulation-based; no physical prototype was built. – No use of electronic controllers or sensors. – Focused on structural and biomechanical modeling.

Chapter 3
Design Theory

Chapter 3: Design Theory and Robotic Equations

3.1 Introduction:

In this chapter, we will talk about the research we conducted, the purpose of this robot, the benefits of investing in it, and some additional tasks that it can perform.

Table 12 The parameters of the robot

Name	Length(mm)	Material	Weight(N)
Upper Body	182	Aluminum alloy	1.341
Lower Body	182	Aluminum alloy	1.321
Servo Bracke-1	33	Aluminum alloy	0.106
Servo Bracke-2	33	Aluminum alloy	0.109
Upper Leg	90	Aluminum alloy	0.127
Lower Leg	136	Aluminum alloy	0.124

3.2 The check of the key components:

The legs is the mainly key component of the bionic spider robot. Because the legs not only bear the weight of itself and trunk, but also bear the counterforce from the ground during the movement of the robot, the content of this section is mainly about the check analysis of the leg structure. The force analysis of one supporting leg of the robot is shown in (Fig 3.2-1).

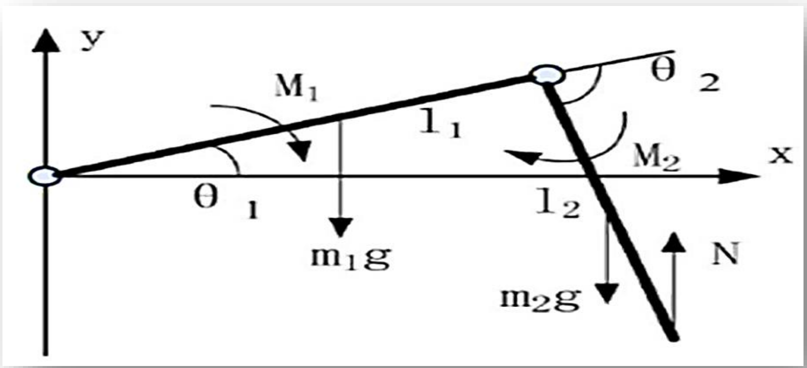


Figure 2. The force diagram of the supporting leg

Since the rotate axis of the base joint is perpendicular to that of the other joints, the bolt bears most of the torque of the joint. Therefore, only the torque analysis of the hip joint and the knee joint are required. According to the principle of virtual work, the generalized equilibrium equation of the support leg is established as follows:

$$\boldsymbol{M_1 + \frac{1}{2}\, m_1 g l_1 cos\theta_1 + m_2 g \left[l_1 cos\theta_1 + \frac{1}{2} l_2 \cos(\theta_2 - \theta_1)\right]}$$

$$\boldsymbol{= N[l_1 cos\theta_1 + l_2 \cos(\theta_2 - \theta_1)} \text{ (1)} \quad (3.1)$$

$$\boldsymbol{M_2 + \frac{1}{2}\, m_2 g l_1 \cos(\theta_2 - \theta_1) = N l_2 \cos(\theta_2 - \theta_1)} \text{(2)} \quad (3.2)$$

$$\boldsymbol{N = \frac{1}{3}\, mg} \text{ (3)} \quad (3.3)$$

Where:

$\boldsymbol{l_1}$& $\boldsymbol{l_2}$ = *the length of the Upper Leg and the Lower Leg respectively*

$\boldsymbol{m_1}$& $\boldsymbol{m_2}$ = *mass of the Upper Leg and the Lower Leg respectively*

$\boldsymbol{M_1}$&$\boldsymbol{M_2}$ = *is the torque received by the hip and knee joint respectively*

$\boldsymbol{\theta_1}$= *is the angle between the Upper Leg and the horizontal*

$\boldsymbol{\theta_2}$ = *is the angle between the Lower Leg the extension of the Upper Leg*

$\boldsymbol{N}$ = *is the counterforce of the ground to the foot end.*

m = *total mass equals* ***2.84*** *Kg.*

When $\boldsymbol{\theta_1}$= 30° Assuming the maximum torque at this angle and the range of rotation between the joints is $30 \le \boldsymbol{\theta_1} \le 60$.

$$\therefore \boldsymbol{\theta_2 - \theta_1 = 60°}.$$

From the previous equations it was found that:

$$\boldsymbol{N = 9.28\, N}$$

$\boldsymbol{M_1} = 0.935\, N.m \approx 9.54\, Kgf \cdot cm$ **and** $\boldsymbol{M_2} = 0.628\, N.m \approx 6.40\, Kgf \cdot cm$.

(1) (Sun et al., 2019)

(2) (Sun et al., 2019)

(3) (Sun et al., 2019)

From the above, the servo motors were selected and their torque was $\boxed{\boldsymbol{M} = 10\ Kgf \cdot cm}$ to get the right torque.

3.3 Simulation of analysis:

3.3.1Body design:

The spider's body is designed in a rectangular shape to ensure that the blocks are optimally distributed and to apply the triple gait pattern better than other designs.

The spider's body consists of an upper base and a lower base. The following figures show the upper and lower body design .

- **Lower body:**

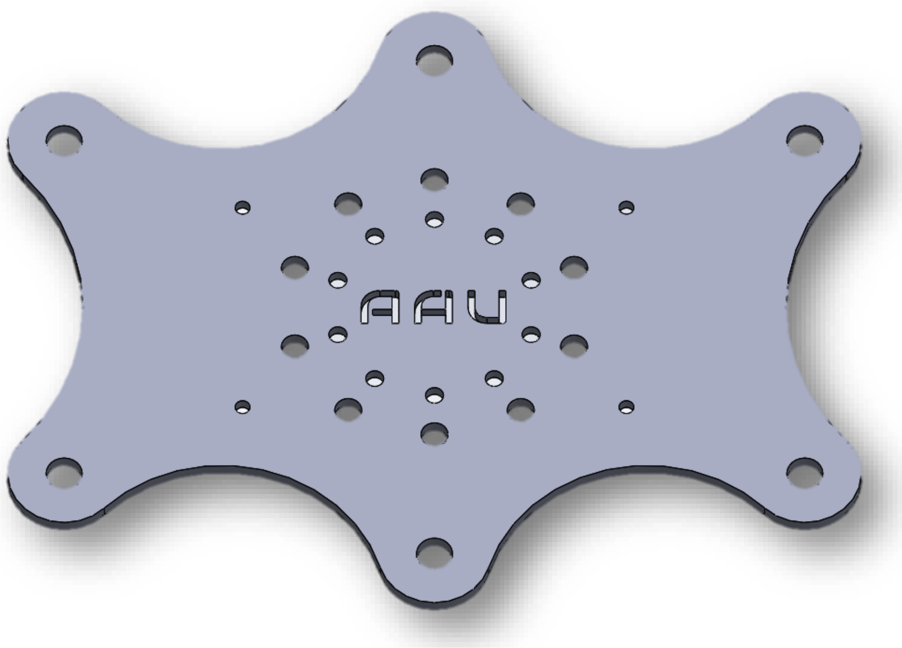


Figure 3. Lower body hexapod robot

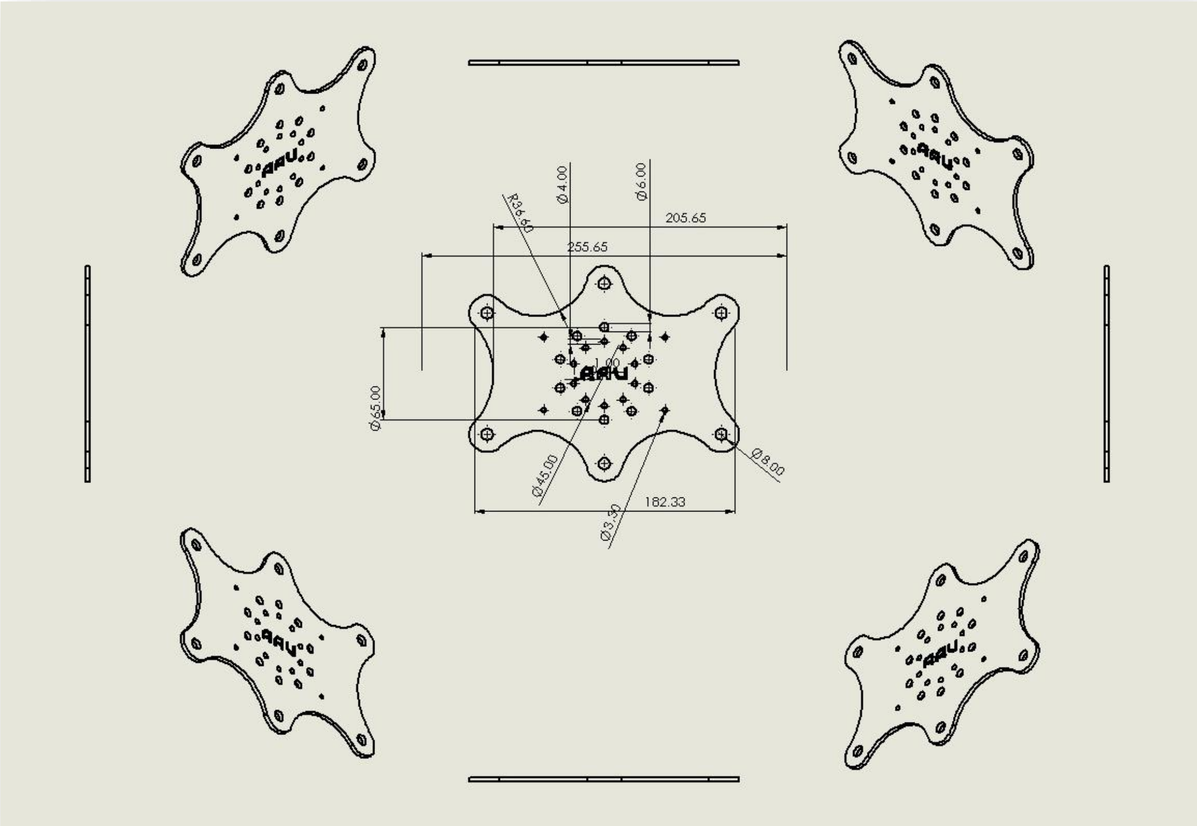


Figure 4. Hexapod robot lower body design

Table 13. Model Information lower body

Document Name and Reference	Treated As	Volumetric Properties
Boss-Extrude4 	Solid Body	Mass:0.134868 kg Volume:4.99512e-05 m^3 Density:2,700 kg/m^3 Weight:1.32171 N

➤ Upper body:

Figure 5. Upper body hexapod robot

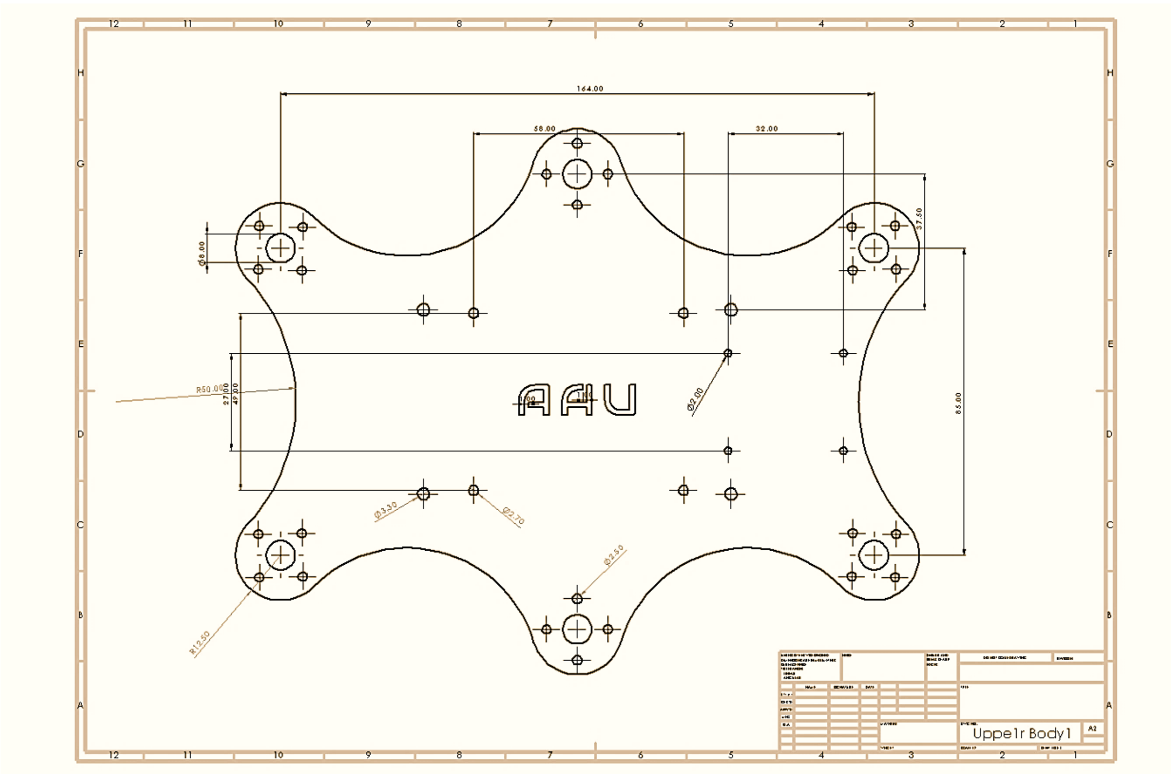


Figure 6. Upper body design

Table 14. Model Information upper body

Document Name and Reference	Treated As	Volumetric Properties
Boss-Extrude3 	Solid Body	Mass:0.136935 kg Volume:5.07166e-05 m^3 Density:2,700 kg/m^3 Weight:1.34196 N

3.3.2The design of coxa -1:

The connection between the coxa of leg structure and the trunk structure is set as the revolute pair of horizontal direction, which is called the base joint. The other end of the coxa is connected to the femur forming a revolute pair in a vertical direction, called a hip joint, fillet edge processing at the edge of the coxa . the following figure shows the coxa design.

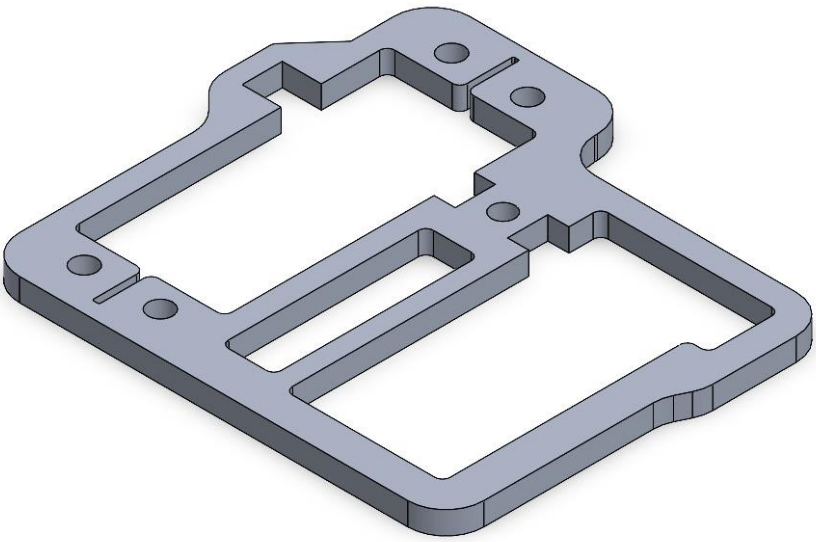


Figure 7. Coxa -1 hexapod robot

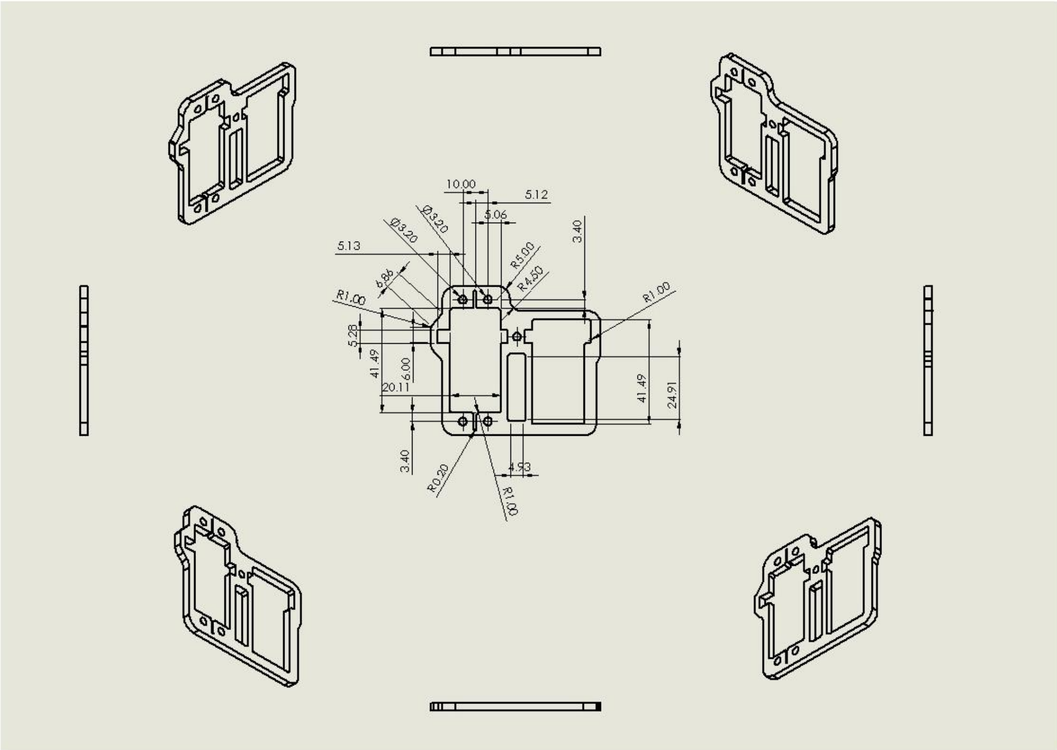


Figure 8. Hexapod robot coxa design

Table 15. Model Information coxa

Document Name and Reference	Treated As	Volumetric Properties
Boss-Extrude1 	Solid Body	Mass:0.0108573 kg Volume:4.02121e-06 m^3 Density:2,700 kg/m^3 Weight:0.106401 N

➢ The design of coxa -2:

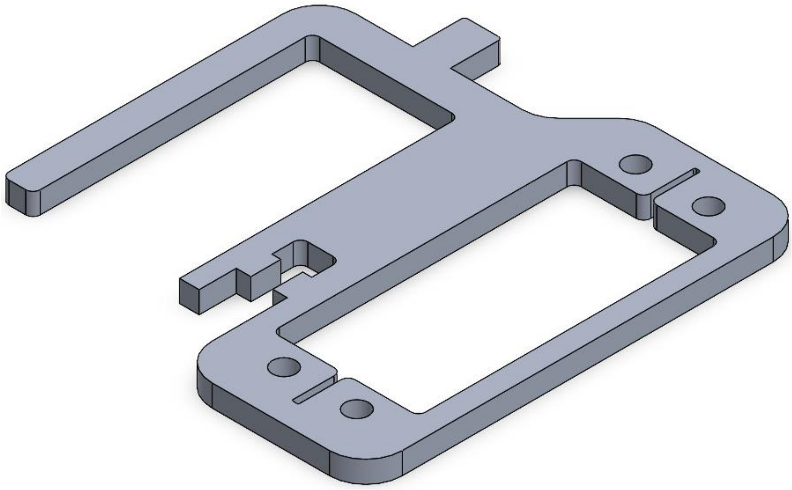


Figure 9. Coxa -2 hexapod robot

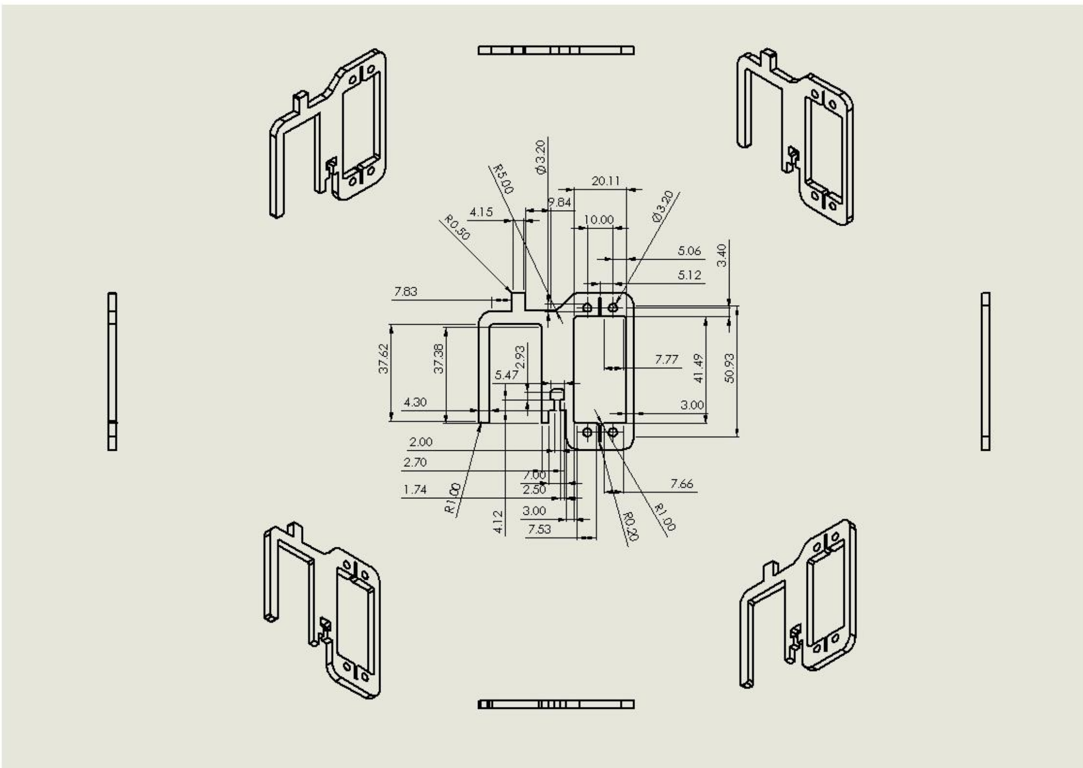
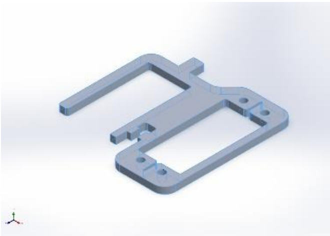


Figure 10. Hexapod robot coxa-2 design

Table 16. Model Information coxa-2

Document Name and Reference	Treated As	Volumetric Properties
Boss-Extrude1 	Solid Body	Mass:0.0111722 kg Volume:4.13784e-06 m^3 Density:2,700 kg/m^3 Weight:0.109487 N

3.3.3 The design of femur:

The femur is located in the middle part of the coxa and the tibia, which acts as a connection and rotation. It has mentioned the end of the femur is connected with the coxa to form the revolute pair of vertical direction, which is called the hip joint in the previous section. The other end of the coxa is connected with the tibia to form the revolute pair of vertical direction, called the knee joint. The three-dimensional model of the femur's model is following shown in Figure This design simplifies the structure of the femur, makes the structure of the assembly's model more concise, and meets the strength requirements.

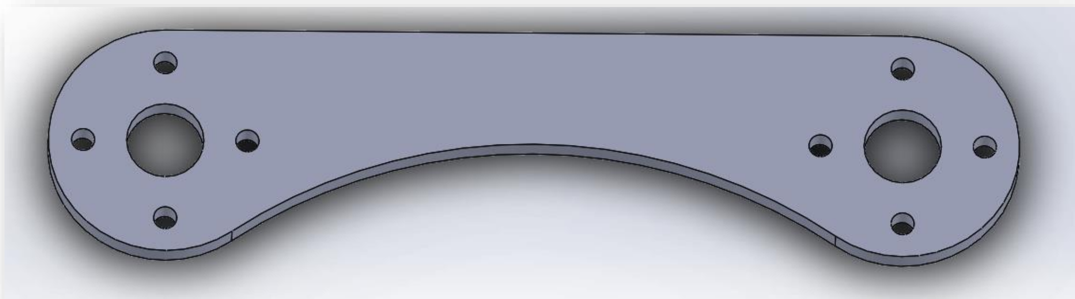


Figure 11. Femur hexapod robot

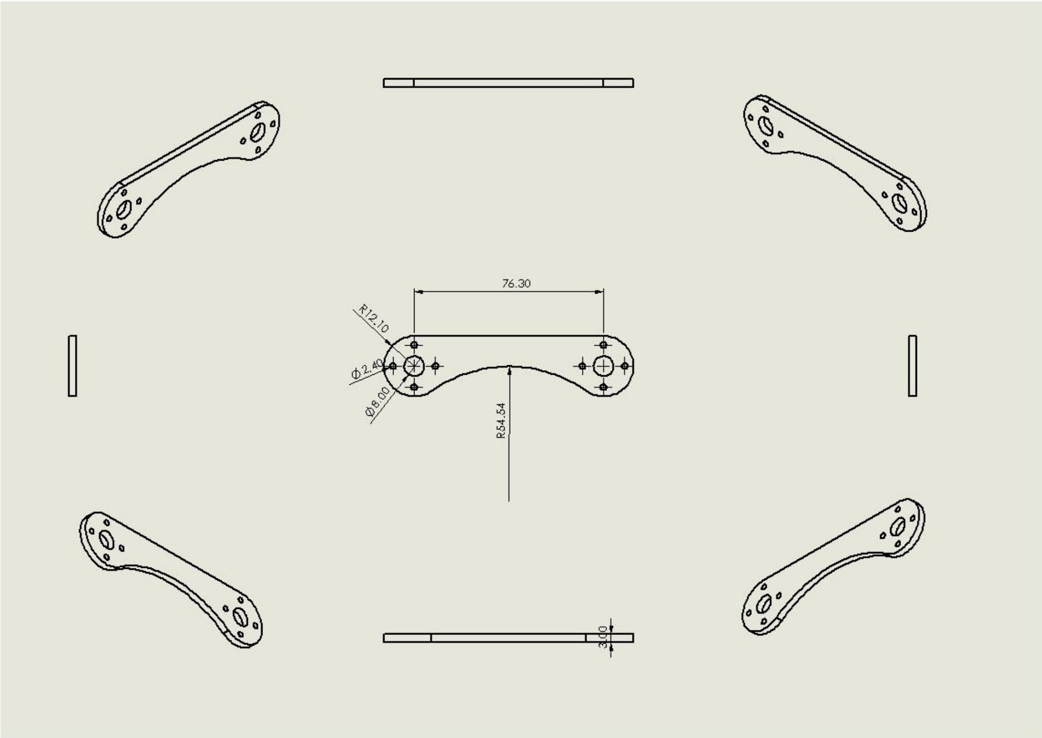
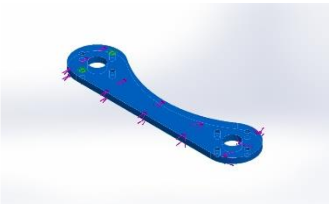


Figure 12. Hexapod robot femur design

Table 17. Model Information femur

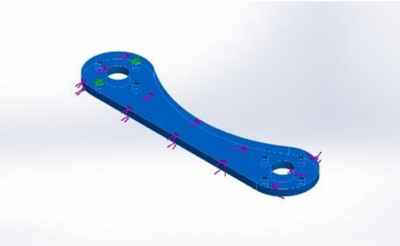
Document Name and Reference	Treated As	Volumetric Properties
Boss-Extrude1 	Solid Body	Mass:0.0130177 kg Volume:4.82138e-06 m^3 Density:2,700 kg/m^3 Weight:0.127574 N

❖ Several simulations were conducted to verify the effectiveness of the design and the results were as follows:

Table 18. Study properties

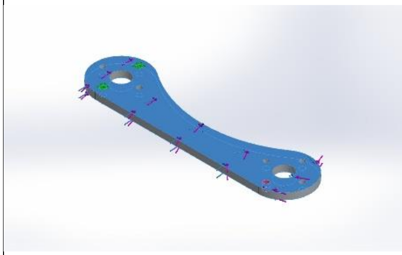
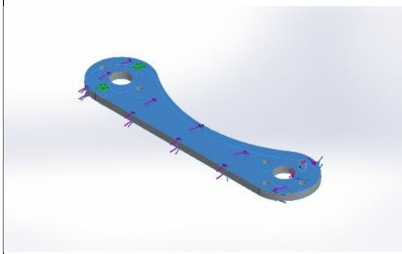
Study name	Static 1
ANALYSIS TYPE	Static
Mesh type	Solid Mesh
THERMAL EFFECT:	On
Thermal option	Include temperature loads
ZERO STRAIN TEMPERATURE	298 Kelvin
Include fluid pressure effects from SOLIDWORKS Flow Simulation	Off
SOLVER TYPE	Automatic
Inplane Effect:	Off
SOFT SPRING:	Off
Inertial Relief:	Off
INCOMPATIBLE BONDING OPTIONS	Automatic
Large displacement	Off
COMPUTE FREE BODY FORCES	On
Friction	Off
USE ADAPTIVE METHOD:	Off

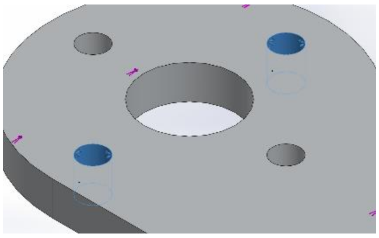
Table 19. Material properties

<table>
<tr><th>Model Reference</th><th colspan="2">Properties</th></tr>
<tr><td rowspan="10"></td><td>Name:</td><td>6061-T6 (SS)</td></tr>
<tr><td>Model type:</td><td>Linear Elastic Isotropic</td></tr>
<tr><td>Default failure criterion:</td><td>Max von Mises Stress</td></tr>
<tr><td>Yield strength:</td><td>2.75e+08 N/m^2</td></tr>
<tr><td>Tensile strength:</td><td>3.1e+08 N/m^2</td></tr>
<tr><td>Elastic modulus:</td><td>6.9e+10 N/m^2</td></tr>
<tr><td>Poisson's ratio:</td><td>0.33</td></tr>
<tr><td>Mass density:</td><td>2,700 kg/m^3</td></tr>
<tr><td>Shear modulus:</td><td>2.6e+10 N/m^2</td></tr>
<tr><td>Thermal expansion coefficient:</td><td>2.4e-05 /Kelvin</td></tr>
</table>

➤ Loads and Fixtures:

Table 20. Load and Fixture femur for two nails

Load name	Load Image	Load Details	
Torque-1		Entities:	**2 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**5 kgf.cm**
Torque-2		Entities:	**2 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**5 kgf.cm**

Fixture name	Fixture Image	Fixture Details	
Fixed-1		Entities:	***2 face(s)***
		Type:	***Fixed Geometry***
		Entities:	***2 face(s)***

Resultant Forces

Components	X	Y	Z	Resultant
Reaction force(N)	**5.78075**	**6.0536e-08**	**24.8711**	**25.5341**

Selection set	Units	Sum X	Sum Y	Sum Z	Resultant
Entire Model	N	5.78075	6.0536e-08	24.8711	25.5341

Selection set	Units	Sum X	Sum Y	Sum Z	Resultant
Entire Model	N	9.94038e-06	1.20141e-07	6.91891e-06	1.21119e-05

Name	Type	Min	Max
Stress	VON: von Mises Stress	2.997e+03N/m^2 Node: 112	4.178e+06N/m^2 Node: 30

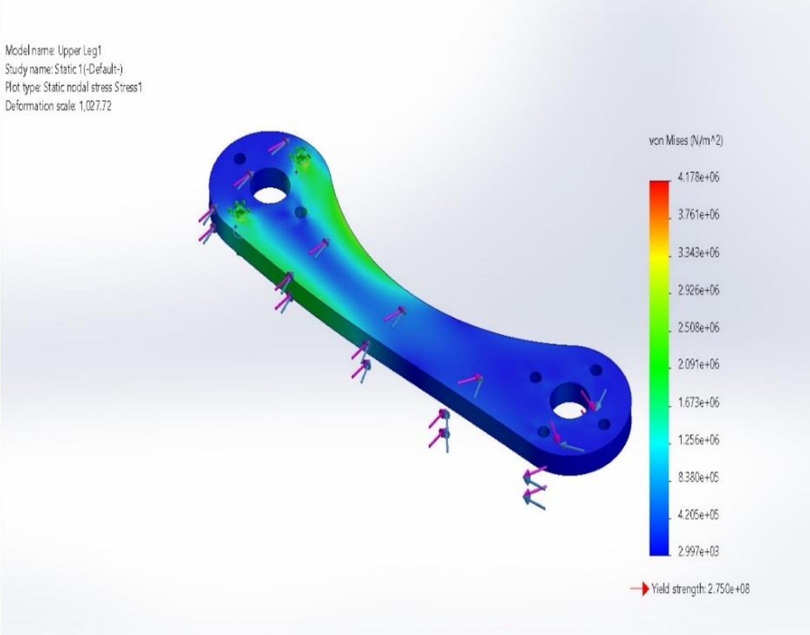


Figure 13. Upper Leg Stress (two nails)

Name	Type	Min	Max
Displacement	URES: Resultant Displacement	0.000e+00mm Node: 9	9.807e-03mm Node: 16160

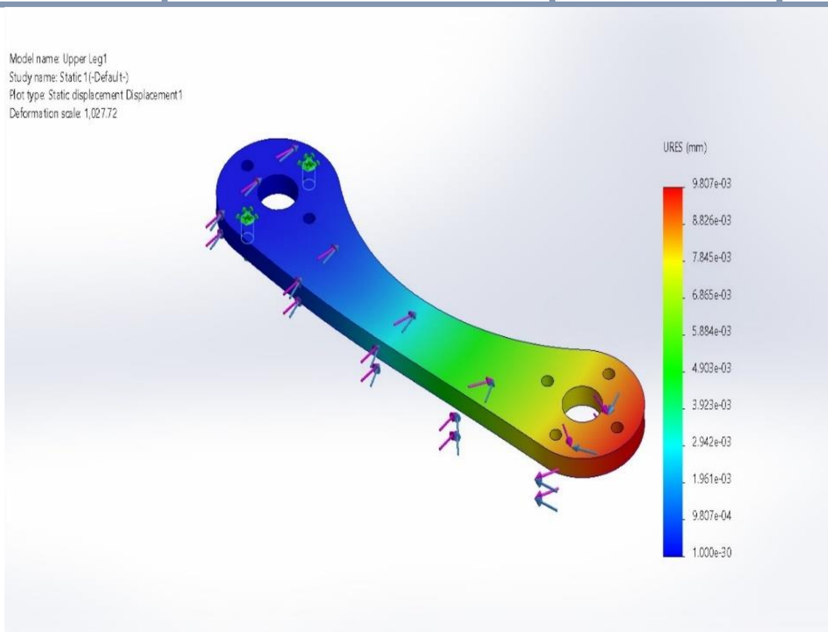


Figure 14. Upper Leg Displacement (two nails)

Name	Type	Min	Max
Strain	ESTRN: Equivalent Strain	5.765e-08 Element: 4154	4.392e-05 Element: 4664

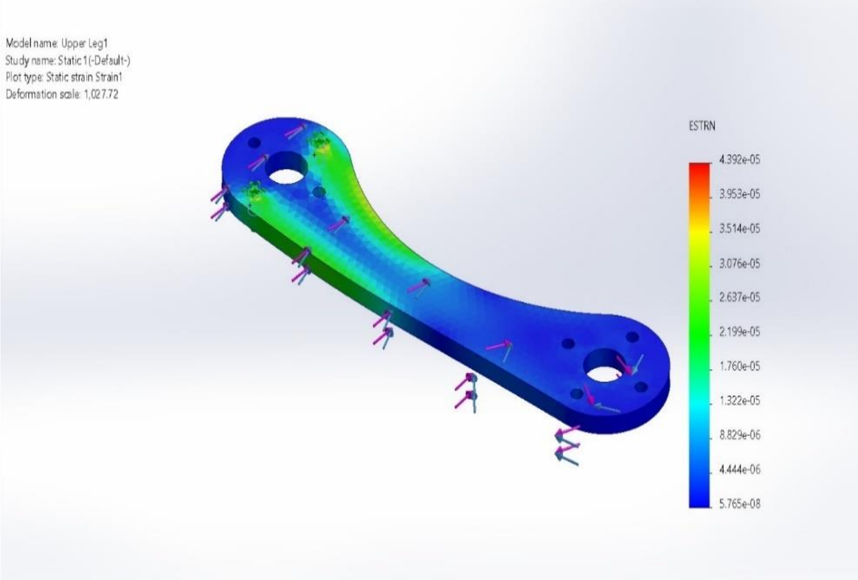


Figure 15. Upper Leg Strain (two nails)

➢ **Resultant Forces :**

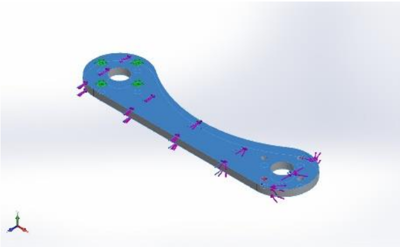
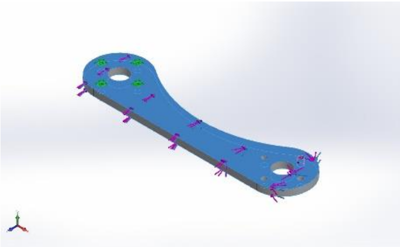
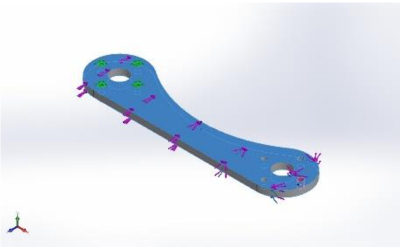
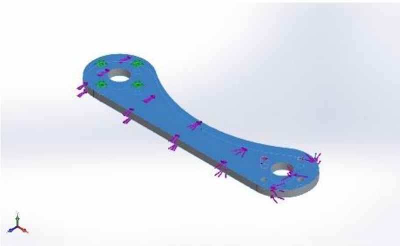
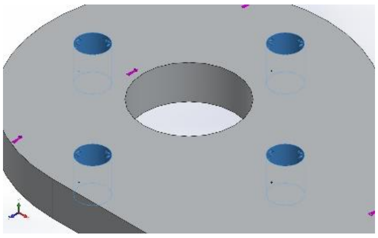
Table 21. Reaction forces

Selection set	Units	Sum X	Sum Y	Sum Z	Resultant
Entire Model	N	5.02777	1.30385e-08	29.4207	29.8472

Table 22. Free body forces

Selection set	Units	Sum X	Sum Y	Sum Z	Resultant
Entire Model	N	1.86842e-05	1.02445e-07	6.37956e-06	1.97436e-05

Table 23. Load and Fixture femur for (four nails)

Load name	Load Image	Load Details	
Torque-1		Entities:	**2 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**2.5 kgf.cm**
Torque-2		Entities:	**2 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**2.5 kgf.cm**
Torque-3		Entities:	**2 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**2.5 kgf.cm**
Torque-4		Entities:	**2 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**2.5 kgf.cm**
Fixture name	Fixture Image	Fixture Details	
Fixed		*Entities:*	***4 face(s)***
		Type:	***Fixed Geometry***
		Entities:	***4 face(s)***

Resultant Forces

Components	X	Y	Z	Resultant
Reaction force(N)	**5.02777**	**1.30385e-08**	**29.4207**	**29.8472**

Name	Type	Min	Max
Stress	VON: von Mises Stress	2.269e+03N/m^2 Node: 14190	4.772e+06N/m^2 Node: 38

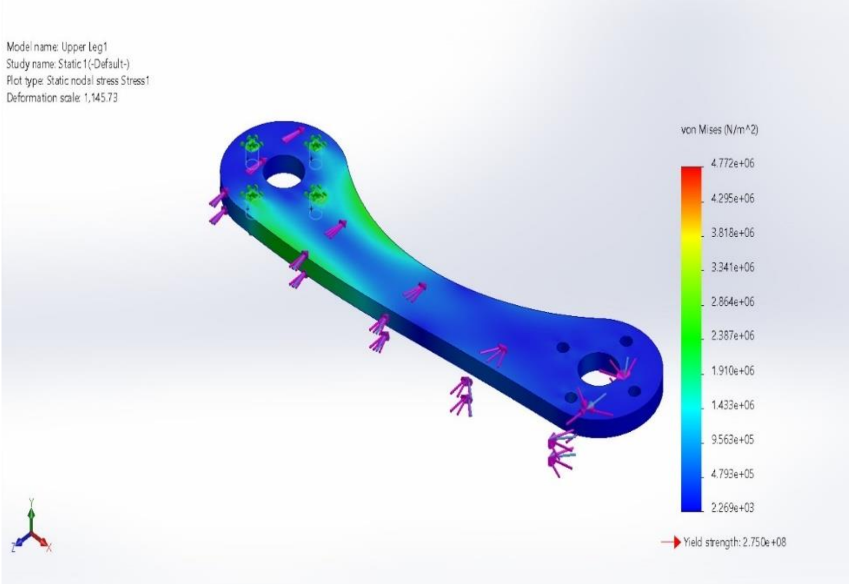


Figure 16. Upper Leg Stress (four nails)

Name	Type	Min	Max
Displacement	URES: Resultant Displacement	0.000e+00mm Node: 9	8.789e-03mm Node: 15963

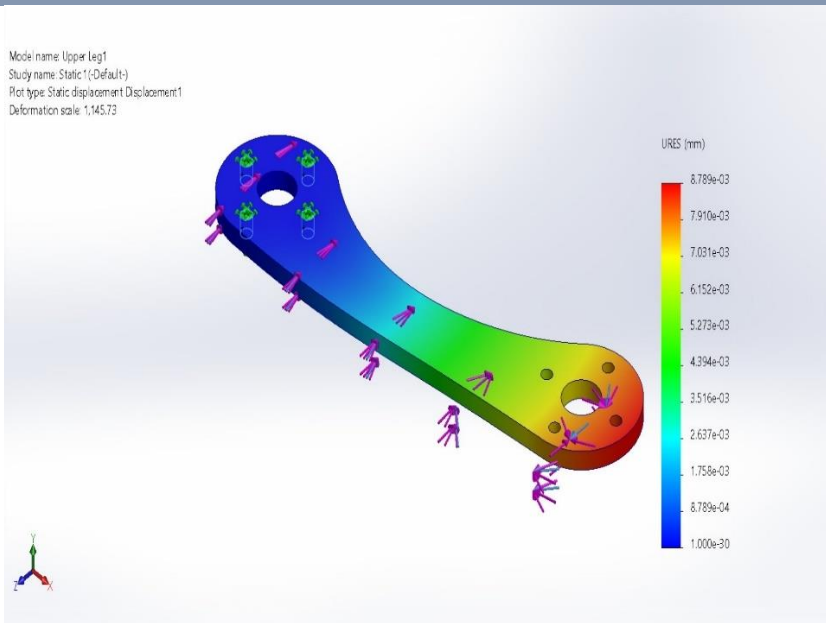


Figure 17. Upper Leg Displacement (four nails)

Name	Type	Min	Max
Strain	ESTRN: Equivalent Strain	4.563e-08 Element: 4925	5.134e-05 Element: 6677

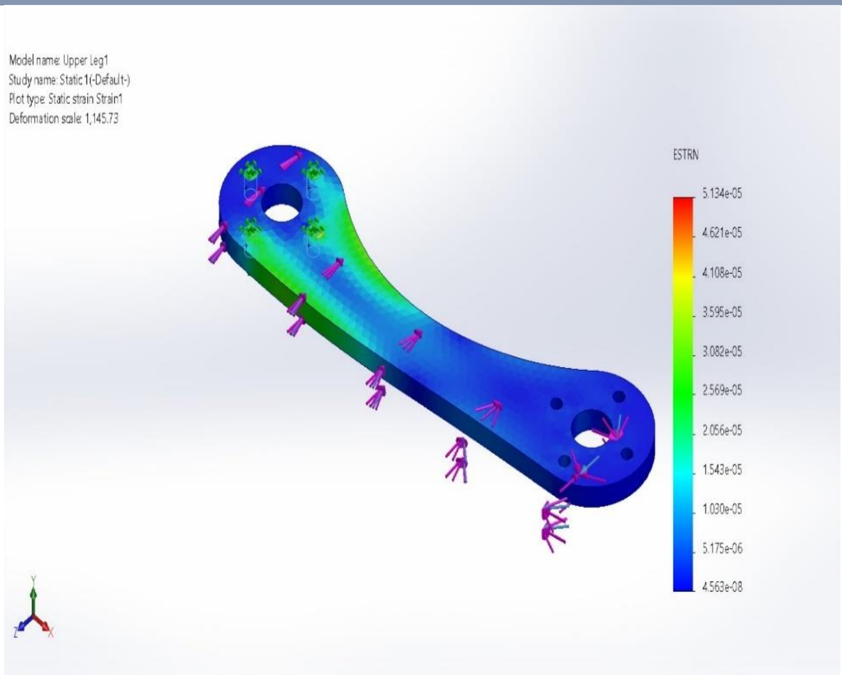


Figure 18. Upper Leg Strain (four nails)

3.3.4The Design of Tibia :

The tibia is 110mm and its thickness is 6 mm.

The tibia is the last joint of the leg's design. The end of the tibia is connected with the femur to form the revolute pair in the vertical direction, called knee joint. The other end of the tibia is in contact with the ground, forming a friction pair to assist the moving, turning and other movement of the bionic spider robot. The three-dimensional model of the tibia is shown in Figure:

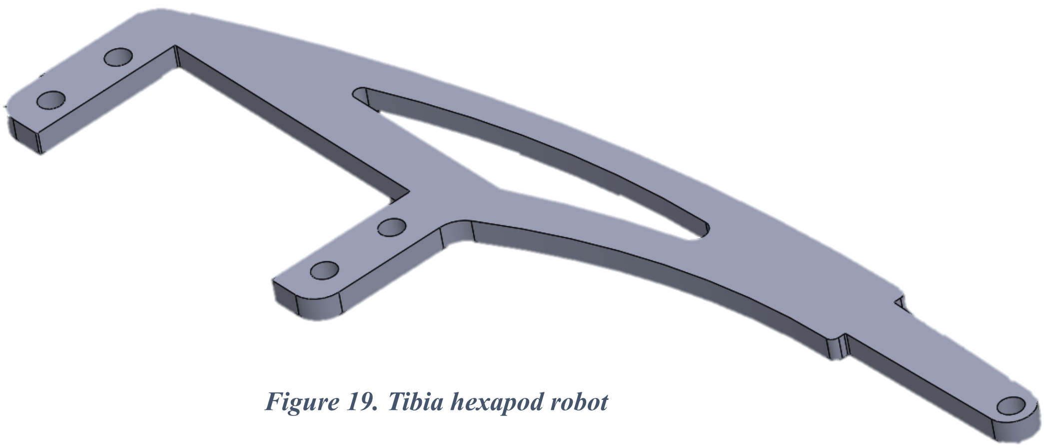


Figure 19. Tibia hexapod robot

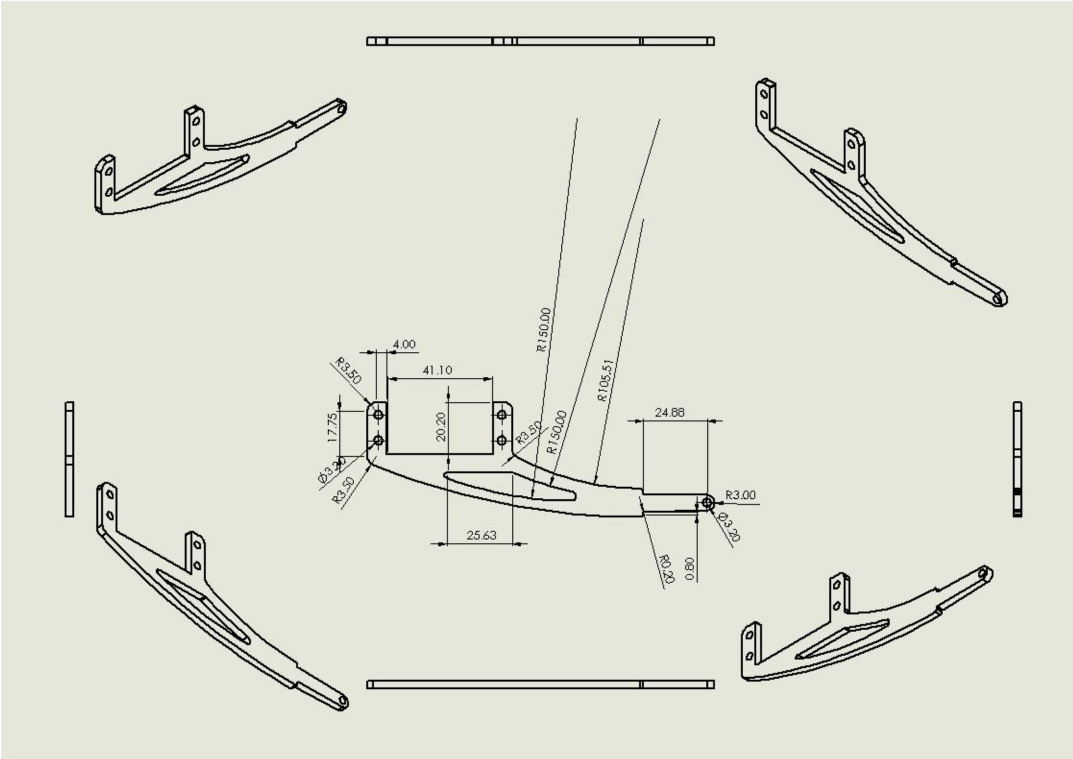
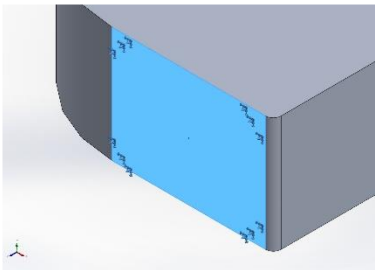


Figure 20. Hexapod robot tibia design

- **Loads and Fixtures:**

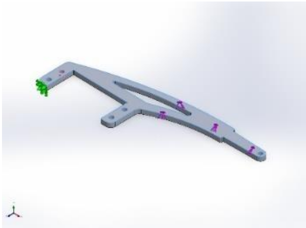
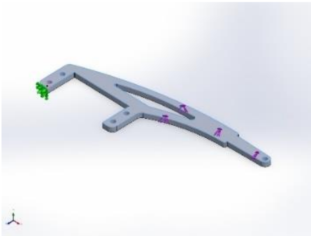
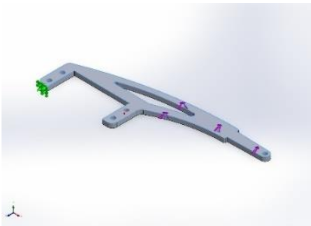
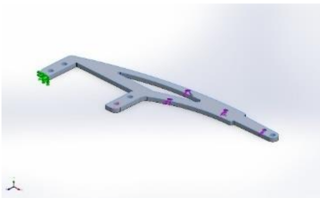
Table 24. load Tibia

Fixture name	Fixture Image	Fixture Details	
Fixed-2		*Entities:*	***1 face(s)***
		Type:	***Fixed Geometry***

Resultant Forces:

Components	X	Y	Z	Resultant
Reaction force(N)	**9.24986**	**1.66893e-06**	**7.09865**	**11.6598**

Table 25. Fixture Tibia

Load name	Load Image	Load Details	
Torque-1		Entities:	**1 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**-2.5 kgf.cm**
Torque-2		Entities:	**1 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**-2.5 kgf.cm**
Torque-3		Entities:	**1 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**-2.5 kgf.cm**
Torque-4		Entities:	**1 face(s)**
		Reference:	**Face< 1 >**
		Type:	**Apply torque**
		Value:	**-2.5 kgf.cm**

➢ Study Results :

Name	Type	Min	Max
Displacement1	URES: Resultant Displacement	0.000e+00mm Node: 15	9.093e-01mm Node: 1177

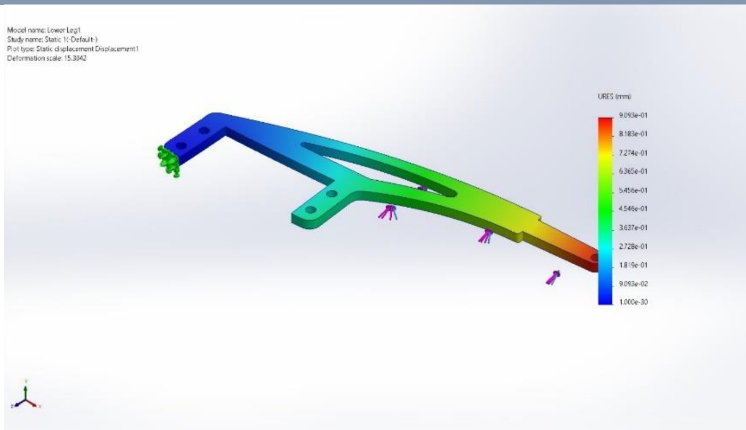


Figure 21. Lower Leg Displacement

Name	Type	Min	Max
Strain1	ESTRN: Equivalent Strain	1.287e-09 Element: 2074	2.221e-03 Element: 6887

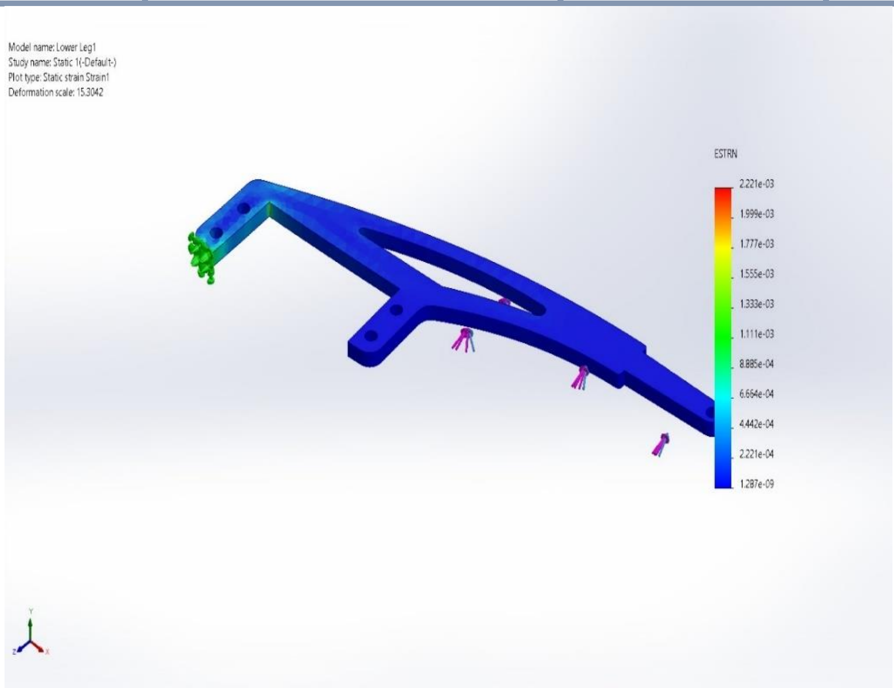


Figure 22. Lower Leg Strain

Table 26. study properties

Model Reference	Properties	
	Name:	**6063-T6**
	Model type:	**Linear Elastic Isotropic**
	Default failure criterion:	**Unknown**
	Yield strength:	**2.15e+08 N/m^2**
	Tensile strength:	**2.4e+08 N/m^2**
	Elastic modulus:	**6.9e+10 N/m^2**
	Poisson's ratio:	**0.33**
	Mass density:	**2,700 kg/m^3**
	Shear modulus:	**2.58e+10 N/m^2**
	Thermal expansion coefficient:	**2.34e-05 /Kelvin**

3.4 Robotic Equations and Control Systems :

3.4.1 Introduction:

This chapter formalizes the mathematical and physical foundations that govern the motion of a six-legged (hexapod) robot and explains how digital commands are transformed into precise physical movements. The treatment integrates:

- **Kinematics:** forward/inverse mapping between joint space and foot space, and differential kinematics via the Jacobian and its time derivative.
- **Frame transformations:** from the robot body to each leg frame, consistent with the implementation in your code.
- **Gaits:** generation of leg trajectories and footfall timing for Tripod and Ripple patterns, including speed-to-subdivision mapping.
- **Control and sensor fusion:** position-form PID using IMU feedback, with complementary and Kalman-based fusion options and quaternion attitude propagation.
- **Actuation and sensing equations:** PWM mapping on PCA9685, ultrasonic time-of-flight, and servo calibration.
- **Validation and stability:** numerical consistency checks (FK↔IK), Jacobian verification, workspace and singularity analysis, stability margins in the support polygon.

Implementation alignment. The first joint (Coxa) rotates about the x-**axis**; joints 2–3 (Femur, Tibia) rotate about y. Link lengths (from code): $L_1 = 33$ mm, $L_2 = 90$ mm,

$L_3 = 110$ mm. Internal angles are in radians; servo outputs are in degrees. Forces are in newtons [N]; torques in N·mm (divide by 1000 for N·m).

Figure 4-1 (suggested): Leg schematic with link lengths and axes (a, b, c).

3.5 Robot Kinematics:

3.5.1 Frames, Notation, and Body-to-Leg Transform:

Let {B} denote the body frame and $\{L_i\}$ the frame attached to leg $i \in \{1, \ldots, 6\}$. The body attitude is applied to six nominal foot anchor points, then each point is mapped to the corresponding leg frame using a fixed yaw rotation ϕ_i and base offsets.

- **Leg base yaw angles (degrees):**

$$\boldsymbol{\phi_i} \in \{\mathbf{+54},\ \mathbf{0},\ \mathbf{-54},\ \mathbf{-126},\ \mathbf{180},\ \mathbf{+126}\}.$$

- **Base offsets (mm):**

$$\boldsymbol{\Delta x} = \{\mathbf{-94}, \mathbf{-85}, \mathbf{-94}, \mathbf{-94}, \mathbf{-85}, \mathbf{-94}\}, \quad \boldsymbol{\Delta z} = \mathbf{-14}.$$

Body → leg mapping (as in your code):

$$\begin{aligned} \mathbf{x_i^{(L)}} &= \mathbf{x_i^{(B)} cos\phi_i + y_i^{(B)} sin\phi_i + \Delta x_i}, \\ \mathbf{y_i^{(L)}} &= \mathbf{-x_i^{(B)} sin\phi_i + y_i^{(B)} cos\phi_i}, \\ \mathbf{z_i^{(L)}} &= \mathbf{z_i^{(B)} + \Delta z}. \end{aligned} \qquad (4\text{-}1)$$

Permutation before IK (implementation):

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{P} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

then *coordinate_to_angle(x',y',z')* is called. This preserves the mathematical model while matching code conventions.

Figure 4-2 (suggested): Transform pipeline and the permutation matrix $\mathbf{P}$.

3.5.2 Forward Kinematics (FK):

For a single leg with joint angles a (about x) and b, c (about y), foot location in {L}:

$$\boxed{\begin{aligned} x &= L_2 \sin b + L_3 \sin(b+c), \\ y &= \sin a \left(L_1 + L_2 \cos b + L_3 \cos(b+c)\right), \\ z &= \cos a \left(L_1 + L_2 \cos b + L_3 \cos(b+c)\right). \end{aligned}} \tag{4.2}$$

Worked FK examples.

Using $L_1 = 33$, $L_2 = 90$, $L_3 = 110$ (mm):

- **FK-1:** $(a, b, c) = (150°,\ 20°,\ 100°)$
  $(x, y, z) \approx (126.05,\ 31.29,\ -54.19)$ mm.

- **FK-2:** $(a, b, c) = (120°,\ 10°,\ -20°)$
  $(x, y, z) \approx (32.68,\ 129.98,\ -75.08)$ mm.

- **FK-3 (near-straight femur+tibia):** $(a, b, c) = (135°,\ 5°,\ 0°)$
  $(x, y, z) \approx (19.85,\ 101.98,\ -101.98)$ mm.

(Rounding here is for readability; analytically, no rounding is part of the formula.)

3.5.3 Inverse Kinematics (IK):

Given target (x, y, z) in {L}:

1. **First joint (about** x**):**

$$\mathbf{a = \frac{\pi}{2} - atan2(z,\ y).} \tag{4.3}$$

2. **Auxiliary definitions:**

$$\mathbf{x_3 = 0, \quad x_4 = L_1 \sin a, \quad x_5 = L_1 \cos a, \quad \ell_{23} = \sqrt{(z - x_5)^2 + (y - x_4)^2 + (x - x_3)^2}.} \tag{4.4}$$

3. **Femur/Tibia angles:**

$$\boxed{\begin{aligned} \mathbf{w} &= \frac{\mathbf{x}}{\boldsymbol{\ell}_{23}}, \qquad \mathbf{v} = \frac{\mathbf{L_2^2 + \ell_{23}^2 - L_3^2}}{\mathbf{2 L_2 \ell_{23}}}, \qquad \mathbf{u} = \frac{\mathbf{L_2^2 + L_3^2 - \ell_{23}^2}}{\mathbf{2 L_3 L_2}}, \\ \mathbf{b} &= \mathbf{\arcsin(w) - \arccos(v)}, \qquad \mathbf{c = \pi - \arccos(u)}. \end{aligned}} \tag{4.5}$$

Reachability (as used in your project):

$$\mathbf{70\ mm \le d = \sqrt{x^2 + y^2 + z^2} \le 248\ mm}. \tag{4.6}$$

Execution note (not part of the math): clip arguments of $\arcsin(\cdot)$, $\arccos(\cdot)$ to $[-1,1]$ and optionally round results to match servo resolution.

Worked IK examples (multiple targets).

- **IK-A (nominal stance):** $(x, y, z) = (130,\ 25,\ -60)$ mm
  $a \approx 157.38°,\ b \approx 21.47°,\ c \approx 96.60°$.
  FK back-check with (4-2) reproduces the target to numerical precision.

- **IK-B (forward reach):** $(160,\ 20,\ -40)$ mm
  $(a, b, c) \approx (153.44°,\ 44.64°,\ 73.76°)$.

- **IK-C (lateral out):** $(100,\ 60,\ -50)$ mm
  $(a, b, c) \approx (129.81°,\ -0.27°,\ 114.36°)$.

- **IK-D (downward):** $(120,\ 10,\ -90)$ mm
  $(a, b, c) \approx (173.66°,\ 9.30°,\ 97.22°)$.

- **IK-E (in-front compact):** $(110,\ 30,\ -50)$ mm
  $(a, b, c) \approx (149.04°,\ 12.52°,\ 112.13°)$.

3.5.4 Workspace and Singularity Analysis:

The planar distance from joint-2 to the foot lies in $[|L_2 - L_3|,\ L_2 + L_3] = [20,\ 200]$ mm. Accounting for L_1 (projected along the a-dependent direction) yields a toroidal workspace in $\{L\}$ truncated by (4-6).

Singularities (loss of rank in J):

- When $\sin b \approx 0$ and $\sin(b + c) \approx 0$ (leg nearly folded straight), the Jacobian loses manipulability in directions orthogonal to the active axes.

- Practical rule: avoid $|b| < 3°$ with $|b + c| < 3°$ during motion planning; bias foot placements away from straight-leg configurations.

3.5.5 Differential Kinematics: Jacobian and Accelerations:

Let:

$$\mathbf{S} = \mathbf{L_1} + \mathbf{L_2 cosb} + \mathbf{L_3 cos(b + c)}, \qquad \mathbf{T} = \mathbf{L_2 sinb} + \mathbf{L_3 sin(b + c)}.$$

Then the linear Jacobian is

$$\boxed{\mathbf{J(q)} = \begin{bmatrix} \mathbf{0} & \mathbf{L_2 cosb + L_3 cos(b+c)} & \mathbf{L_3 cos(b+c)} \\ \mathbf{Scosa} & \mathbf{-sina\,T} & \mathbf{-sina\,L_3 sin(b+c)} \\ \mathbf{-\,Ssina} & \mathbf{-cosa\,T} & \mathbf{-cosa\,L_3 sin(b+c)} \end{bmatrix}.} \tag{4.7}$$

Kinematic/dynamic relations:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}, \qquad \boldsymbol{\tau} = \mathbf{J}^\top \mathbf{f}, \qquad \ddot{\mathbf{x}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}. \tag{4.8}$$

Time derivative $\dot{\mathbf{J}}$. Define

$$\dot{\mathbf{S}} = -\mathbf{L_2 sinb}\,\dot{\mathbf{b}} - \mathbf{L_3 sin(b+c)}(\dot{\mathbf{b}} + \dot{\mathbf{c}}), \quad \dot{\mathbf{T}} = \mathbf{L_2 cosb}\,\dot{\mathbf{b}} + \mathbf{L_3 cos(b+c)}(\dot{\mathbf{b}} + \dot{\mathbf{c}}),$$

and differentiate (4-7) element-wise (full expression omitted for brevity but obtainable directly). This is needed for operational-space acceleration control and feedforward compensation.

Worked Jacobian/torque (IK-A).
At $(a, b, c) \approx (157.38°,\ 21.47°,\ 96.60°)$, a vertical downward foot force $\mathbf{f} = [0,0,-5]^\top$ N yields:

$$\boldsymbol{\tau} = \mathbf{J}^\top \mathbf{f} \approx \begin{bmatrix} 0.125 \\ -0.600 \\ -0.448 \end{bmatrix} \text{ N·m.}$$

This informs actuator sizing and per-joint load limits.

Dexterity snapshot. Singular values σ of **J** (mm/rad) at the IK cases:

Case	σ (mm/rad)	Condition $\kappa = \sigma_{max}/\sigma_{min}$
A	(162.3, 65.0, 60.6)	2.68
B	(187.8, 50.6, 44.7)	4.20
C	(141.7, 78.1, 63.6)	2.23
D	(161.6, 90.6, 60.8)	2.66

Case	σ (mm/rad)	Condition $\kappa = \sigma_{max}/\sigma_{min}$
E	(144.2, 63.6, 58.3)	2.47

Lower κ indicates better isotropy; Case B is comparatively less dexterous.

3.6 Control and Balancing Systems:

3.6.1 Position-Form PID Controller:

We use a **position-form** PID whose output is an absolute corrective offset in attitude:

$$K_p = 0.500, \quad K_i = 0.00, \quad K_d = 0.0025, \quad |I| \le 10. \tag{4.9}$$

$$\boxed{u = K_p\, e + K_i \sum e + K_d \left(e - e_{prev}\right)} \tag{4.10}$$

where e is roll/pitch error from IMU. The correction u modifies (roll,pitch) prior to generating foot references.

Discrete-time details. With sampling period Δt,

$I_k = \mathrm{sat}(I_{k-1} + e_k)$, $\quad D_k = \frac{e_k - e_{k-1}}{1}$ (code uses a unitless diff), $u_k = K_p e_k + K_i I_k + K_d D_k$.

Anti-windup is implemented via integral clamping $|I_k| \le 10$.

Tuning guidance. Start with $K_i = 0$, increase K_p until acceptable stiffness without oscillation; add small K_d to damp overshoot. Introduce K_i only if steady-state tilt persists.

Numeric illustration. $e = 2.0°$, $e_{prev} = 1.0° \Rightarrow u = 1.0025°$.

3.6.2 Sensor Fusion for Attitude:

Let $\mathbf{z}_g = \boldsymbol{\omega}$ (gyro), $\mathbf{z}_a = \mathbf{a}$ (accelerometer).

Complementary filter (lightweight).

$$\boldsymbol{\hat{\theta}_k = \alpha\left(\hat{\theta}_{k-1} + \omega_k \Delta t\right) + (1 - \alpha)\, \theta_{a,k},}$$

where $\boldsymbol{\theta}_{a,k}$ is the tilt from accelerometer and $\alpha \in [0,1]$. This is effective and cheap to implement.

Kalman filter (roll/pitch summary).

Predict: $\mathbf{x}^- = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{P}^- = \mathbf{F}\mathbf{P}\mathbf{F}^\top + \mathbf{Q};$ Update: $\mathbf{K} = \mathbf{P}^-\mathbf{H}^\top(\mathbf{H}\mathbf{P}^-\mathbf{H}^\top + \mathbf{R})^{-1},$
$$\mathbf{x} = \mathbf{x}^- + \mathbf{K}(\mathbf{z} - \mathbf{H}\mathbf{x}^-), \quad \mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^-. \tag{4.10}$$

Quaternion propagation (for large angles).

$$\dot{\mathbf{q}} = \frac{1}{2}\,\mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}, \qquad \mathbf{q}_k = \mathrm{normalize}(\mathbf{q}_{k-1} + \dot{\mathbf{q}}\Delta t). \tag{4.12}$$

Design rationale. Complementary filters are robust and simple; Kalman fusion yields higher fidelity by balancing gyro drift with accelerometer noise; quaternions avoid Euler singularities.

3.7 Gaits and Foot Trajectories:

3.7.1 Speed–Subdivision Mapping and Vertical Lift:

Let user speed $s \in [2,10]$. Your code maps s to **subdivisions** F per gait cycle:

$$F = \begin{cases} \mathrm{round}(13\,s - 4), & \text{Tripod} \\ \mathrm{round}(15.75\,s + 13.5), & \text{Ripple/Wave}' \end{cases} \qquad Z = 80\ \mathrm{mm}, \quad \Delta z = \frac{Z}{F}.$$

Sample values.

s	F_{Tripod}	F_{Ripple}	Δz Tripod (mm/it)	Δz Ripple (mm/it)
2	22	45	3.636	1.778
6	74	108	1.081	0.741
10	126	171	0.635	0.468

3.7.2 Horizontal Increments and Turning:

For turning by α and translation (x_c, y_c), each leg's planar reference is updated by

$$\delta\mathbf{p}_i = \frac{(R_\alpha \mathbf{p}_i - \mathbf{p}_i) + [x_c, y_c]^\top}{F}, \qquad R_\alpha = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}. \tag{4.14}$$

Worked increment (Tripod, $s = 6$**).**

$\alpha = 10°$, $x_c = 15$ mm, $y_c = 5$ mm, $\mathbf{p}_1 = (110{,}192)$ mm, $F = 74$:

$\delta x \approx 0.631$ mm/it, $\delta y \approx -0.230$ mm/it; $\Delta z \approx 1.081$ mm/it.

4.4.3 Tripod vs Ripple: Timing and Footfall

- **Tripod:** two synchronous triplets (3+3) alternate swing/stance; high static stability and good throughput.
- **Ripple/Wave:** feet lift in sequence (e.g., order {5,2,1,0,3,4}), extending ground contact for smooth low-speed motion.

3.7.3 Swing Trajectories (Smoother Option):

Your implementation uses piecewise-linear lift/settle. A smoother alternative is a C^2 fifth-order polynomial $z(\xi) = \sum_{k=0}^{5} a_k\, \xi^k$, $\xi \in [0,1]$, with boundary conditions on position/velocity/acceleration at lift-off and touch-down. The horizontal motion uses (4-14); vertical follows $z(\xi)$ reaching height Z.

3.8 Sensor and Actuation Equations:

3.8.1 Ultrasonic Time-of-Flight:

$$\boxed{\mathbf{d} = \frac{\mathbf{v_{sound}\, t_{echo}}}{\mathbf{2}}}, \qquad \mathbf{v_{sound} \approx 331 + 0.6\, T\ (m/s)}. \tag{4.15}$$

Example. $T = 25°C \Rightarrow v \approx 346$ m/s; $t_{echo} = 12$ ms $\Rightarrow d \approx 2.076$ m.

Practical filtering. Median-of-5, minimum hold, and rejection of echoes with t_{echo} outside plausible bounds.

3.8.2 PWM Servos on PCA9685:

Prescale (12-bit, $f_{osc} = 25$ **MHz):**

$$\boxed{\text{prescale} \approx \left\lfloor \frac{f_{osc}}{4096\, f_{PWM}} - 1 \right\rceil} \tag{4.16}$$

For $f_{PWM} = 50$ Hz: prescale ≈ 121.

Angle → pulse width → counter.

$$t_{pulse}(\theta) = t_{min} + \frac{\theta}{180°}(t_{max} - t_{min}), \tag{4.17}$$

$$\boxed{\text{count} = \frac{t_{pulse}}{T_{PWM}} \cdot 4096 = t_{pulse}\, f_{PWM} \cdot \frac{4096}{10^6}} \tag{4.18}$$

with $t_{min} \approx 1000$ µs, $t_{max} \approx 2000$ µs, $T_{PWM} = 1/50 = 20$ ms.

Counts at typical pulses.

t_{pulse} (µs)	Count
1000	$1000 \cdot 50 \cdot 4096/10^6 \approx 204.8$
1500	≈ 307.2
2000	≈ 409.6

Calibration maps (as implemented).

Legs 1–3:

$$\begin{gathered} s_{i1} = \text{clip}(a_i + \delta a_i), \\ s_{i2} = \text{clip}\big(90 + (b_i + \delta b_i)\big), \\ s_{i3} = \text{clip}(-(c_i) + \delta c_i) \end{gathered} \tag{4.19}$$

Legs 4–6:

$$\begin{gathered} s_{i1} = \text{clip}(a_i + \delta a_i), \\ s_{i2} = \text{clip}(90 - (b_i) + \delta b_i), \\ s_{i3} = \text{clip}\big(180 + (c_i + \delta c_i)\big) \end{gathered} \tag{4.20}$$

Values are clipped to $[0°, 180°]$ then converted via (4-17) (4-18).

Worked actuation example (IK-A, legs 1–3, zero offsets).

$(a, b, c) \approx (157.38°, 21.47°, 96.60°) \Rightarrow$

$s_1 = 157.38°,\ s_2 = 111.47°,\ s_3 = -96.60°$ (tibia typically needs a per-leg offset δc to land inside range).

For $s_2 = 111.47°$: $t_{pulse} \approx 1619.3$ µs $\Rightarrow$ count ≈ 331.6.

3.9 Body Attitude and Foot Anchors:

3.9.1 Rotation Composition and Body Height:

The body rotation (consistent with your code) is

$$\mathbf{R} = \mathbf{R}_x(\text{pitch})\,\mathbf{R}_y(\text{roll})\,\mathbf{R}_z(\text{yaw}). \tag{4.21}$$

Nominal body-fixed foot anchors (mm):

$\{(110,192,0),\ (220,0,0),\ (110,-192,0),\ (-110,-192,0),\ (-220,0,0),\ (-110,192,0)\}.$

Body height $H = -25$ mm. Each anchor maps to

$${}^{B}\mathbf{p}_{F_i} = \begin{bmatrix} 0 \\ 0 \\ H \end{bmatrix} + \mathbf{R}\,{}^{B}\mathbf{p}^{0}_{F_i}. \tag{4.22}$$

Worked substitution.

pitch $= -3°$, roll $= 5°$, yaw $= 15°$:

${}^{B}\mathbf{p}_{F_1} \approx (56.34,\ 213.89,\ -31.27)$ mm.

Mapping to $\{L_1\}$ via (4-1): $\left(x^{(L)}, y^{(L)}, z^{(L)}\right) \approx (112.16,\ 80.14,\ -45.27)$ mm $\Rightarrow$ IK yields $(a, b, c) \approx (119.46°,\ 4.17°,\ 102.05°)$.

3.10 Stability and Support Polygon:

3.10.1 Quasi-Static Stability:

Let Π be the vertical projection and $\mathcal{H}$ the convex hull of feet in stance:

$$\Pi({}^{W}\mathbf{p}_{\text{COM}}) \in \text{ConvHull}\left(\{{}^{W}\mathbf{p}_{F_i}\}_{\text{stance}}\right). \tag{4.23}$$

Stability margin:

$$\text{SM} = \min_{e \in \partial\mathcal{H}} \text{dist}(\Pi({}^{W}\mathbf{p}_{\text{COM}}), e) > 0. \tag{4.24}$$

Tripod typically yields larger SM than Ripple at the same step length.

3.10.2 Friction Constraint:

For each stance foot with normal reaction F_n and tangential force $\mathbf{F}_t$:

$$\| \mathbf{F}_t \| \leq \mu F_n, \quad F_n > 0, \tag{4.25}$$

with μ depending on the terrain. Foot trajectories and duty factors must respect (4-24)–(4-26).

3.11 Implementation Architecture and Safety:

- **Loop structure:** attitude fusion → PID corrections → body rotation → foot anchor update → body-to-leg mapping → IK per leg → calibration → PWM output.
- **Sampling:** keep IMU fusion at higher rate (e.g., 100–200 Hz), gait updates at 50–100 Hz, PWM at 50 Hz.
- **Safety checks:** apply reachability (4-6); saturate joint angles; reject IK if $\kappa(\mathbf{J})$ exceeds a threshold (e.g., > 10).
- **Slew limiting:** rate-limit servo angle changes to reduce jerk and power spikes.

3.12 Validation and Tests:

1. **FK↔IK consistency:** sample $N \geq 1000$ foot targets within (4-6); run IK then FK; report mean and max position error $\| \Delta\mathbf{p} \|$. Target ≤ 0.5 mm with double precision.
2. **Jacobian check by finite differences:** perturb each joint q_j by ϵ, compute $\Delta\mathbf{x}/\epsilon$; compare with column j of $\mathbf{J}$. Relative error $< 1\%$ for $\epsilon \approx 10^{-4}$ rad.
3. **Load-to-torque mapping:** apply known payload/contact force estimates; verify $\boldsymbol{\tau} = \mathbf{J}^{\top}\mathbf{f}$ trends and ensure actuator limits are respected.
4. **Gait metrics:** measure step length, body speed, power draw, and stability margin for Tripod vs Ripple at $s \in \{3,6,9\}$.
5. **Control response:** command a 5° step in pitch; record settling time, overshoot, and steady error; tune K_p, K_d accordingly.

3.13 Design Rationale (Consolidated):

- **Tripod** chosen for **maximum stability** with reasonable throughput on uneven terrain; **Ripple** for smoother low-speed motion and extended ground contact.
- **Position-form PID** aligns with producing absolute attitude corrections; it is simple to tune and consistent with your code.
- **Kalman/Complementary fusion** balances gyro drift and accelerometer noise; **quaternions** avoid Euler singularities for larger maneuvers.
- **Workspace/Manipulability** analysis feeds safe foot placements and helps avoid near-singular postures that inflate joint torques.

3.14 Electronic circuit design :

In this paragraph , we will talk about the electronic pieces used in the design and why these pieces were chosen . the electronic components used will be mentioned in detail in another chapter.

1- **Channel 12-bit PWM/Servo Driver (using I2C):** The servo driver control up to 16 motor and connects via I2C protocol to microcontrollers that have few pwm pins , so this piece is used .

2- **servo motor :** Servo motor is used in applications that require precise control of the rotation angle . The torque of this motor is 10kg /cm at 5v , and its rotating angle is 180.

3- **Raspberry Pi 4 Model B (4GB RAM):** Advanced Performance: Features a quad-core processor running up to 1.5GHz. High Memory Capacity: 8GB supports simultaneous multitasking. Multiple Ports: Includes Micro HDMI, USB 3.0/2.0, Gigabit Ethernet, and built-in Wi-Fi/Bluetooth. Versatile Applications: Suitable for mini-servers, media centers, smart systems, and more.

4- **Raspberry Pi Camera Module V1.3:** High-Resolution Imaging: Enables high-quality still images and video recording . Computer Vision Projects: Useful for facial and object recognition applications. Surveillance and Security: Can be integrated into simple monitoring systems. Educational Projects: Ideal for learning camera programming with languages like Python.

5- **MPU6050 GY-521 Sensor Module:** Determine the direction and location of the robot by measuring the angular change rate relative to the axons and the acceleration in the three axes. It is also used to maintain the robot's balance when walking and prevent it from stumbling by monitoring movement and vibration.

6- **HC-SR04 Ultrasonic Sensor:** Used for obstacle avoidance.

The following figure shows the electronic design of the six – legged robot:

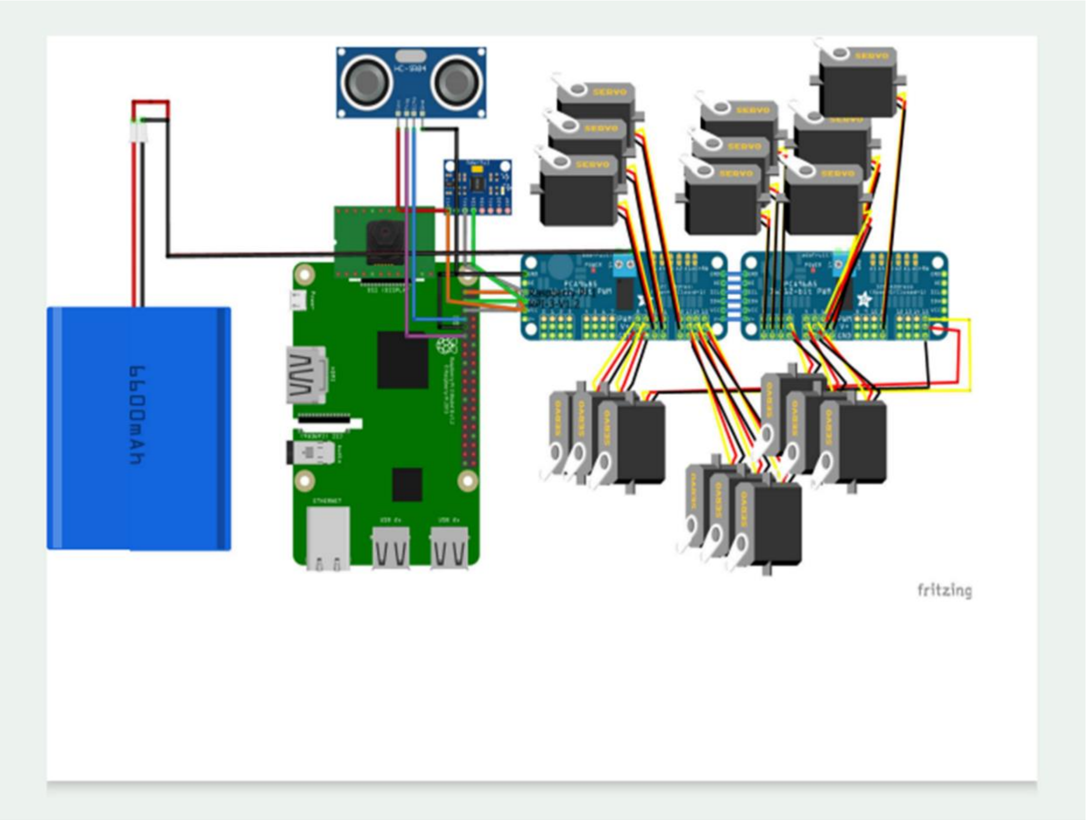


Figure 23. Electronic circuit design

Chapter 4
Installation
Electrical

Chapter 4: Installation Electrical

Introduction:

In this chapter, we implemented the electrical components that are used in the project such as the power supply, actuators, sensors, and other components that will be explained below.

4.1 Lithium-Ion Battery 2200mAh 3.7V:

Component Description:

This is a Lithium-Ion (Li-ion) battery with a capacity of 2200mAh and a voltage of 3.7V. It is widely used in portable devices and electronic projects requiring a reliable, rechargeable power source.

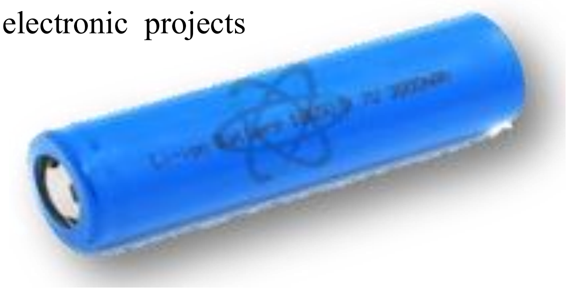


Figure 24. Lithium-Ion Battery

Importance:

High Capacity: Provides 2200mAh, ensuring extended operation before recharging

Rechargeable: Maintains performance over many charging cycles.

Efficient Performance: Delivers a steady voltage of 3.7V, ideal for applications needing stable current.

Lightweight: Suitable for portable devices due to its light weight compared to other battery types.

Main Components:

1. **Electrochemical Cells:** Store energy through chemical reactions.
2. **Outer Casing:** Protects the battery from moisture
3. **Contact Terminals:** Positive and negative terminals for connection.

4.2 16-Channel 12-bit PWM/Servo Driver (using I2C):

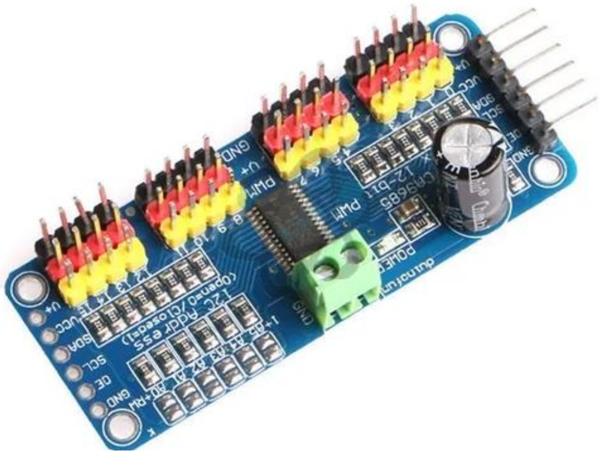


Figure 25. PWM/Servo Driver

Component Description:

The component shown in the image is a 16-Channel 12-bit PWM/Servo Driver using I2C, specifically the model with reference number RSA241-321-07-A.

Importance:

The servo driver control up to 16 motor and connects via I2C protocol to microcontrollers that have few pwm pins , so this piece is used .

Main Components:

1. **GND:** ground.
2. **VCC:** power outlet.
3. **V+:** it is the power selector pin that provides the distributed power to the servos.
4. **SDA:** this port connects to the microcontroller's I2C data line.
5. **SCL:** used to synchronize data transmission and reception.
6. **OE:** this port controls all outputs.
7. **OUTPUT PINS:** 16 output pins

4.3 MG996r Servo Motor:

Servo Motor :

It is ads motor equipped with a gearbox and electrical circuits to precisely control the rotation angle.
Servo motor is used in applications that require precise control of the rotation angle .
The torque of this motor is 10kg /cm at 5v , and its rotating angle is 180.

Figure 29. MG996r Servo Motor

Figure 27. MG996r Servo Motor

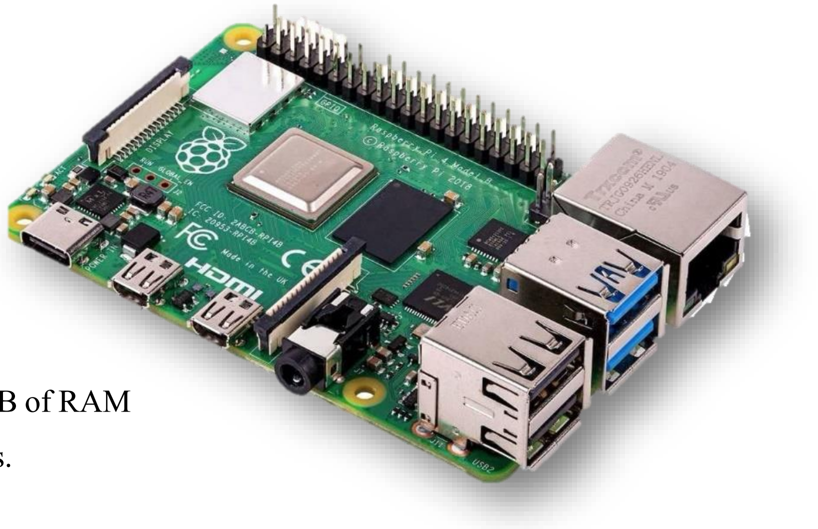
Main components:

1. **Dc motor :**the part that provides movement and rotation.
2. **Metal gear assembly :** transferring motion form the motor to the servo arm.
3. **Internal control circuit:** used to precisely control the rotation angle.
4. **Servo horns:** fixed on the axis of rotation to perform the required movement .
5. **Wiring:** to connect the servo to the power source and control.

How it works

A control signal is applied to the motor in the form of a pulse width modulation signal. The width of this pulse is what controls the rotation angle. Then this pulse is converted into an equivalent voltage by a PWM converter. The variable resistance provides a voltage proportional to the location of the output handle. The comparator compares the voltage generated by the converter and the variable resistance. The difference between the two signals is called the error signal, which is used to drive the DC motor. When the DC motor and the output handles move, the voltage generated by the variable resistance is reduced or deleted, and then the output handle has reached the required location or angle.

4.4 Raspberry Pi 4 Model B (4GB RAM):



Component Description:

It is a small computer with 4 GB of RAM and a number of different ports.

Figure 30. Raspberry Pi 4 Model B (4GB RAM)

Importance:

Advanced Performance: Features a quad-core processor running up to 1.5GHz.

High Memory Capacity: 8GB supports simultaneous multitasking.

Multiple Ports: Includes Micro HDMI, USB 3.0/2.0, Gigabit Ethernet, and built-in Wi-Fi/Bluetooth.

Versatile Applications: Suitable for mini-servers, media centers, smart systems, and more.

Main Components:

1. **Broadcom SoC chip :** Contains the CPU, RAM, and GPU.
2. **HDMI Output :** Display output for connecting the Raspberry Pi to a high-quality display.
3. **Audio Jack :** The audio output is used to connect speakers.
4. **Ethernet Input :** Used to connect the Raspberry Pi to a network.
5. **USB Inputs :**Used to connect USB devices such as the keyboard and mouse.
6. **Micro USB Power Input :** A port to power the Raspberry Pi.
7. **SD Card Input :**For installing a memory card that contains the operating system and user data.
8. **GPIO Pin:** Multiple ports for controlling and communicating with electronic devices.
9. **CSI Camera Input:** Place for connecting the Raspberry Pi's high-resolution cameras.
10. **DSI Display Input:** Used to connect the Raspberry Pi's touch screens.

4.5 Raspberry Pi Camera Module V1.3 (5 Megapixel):

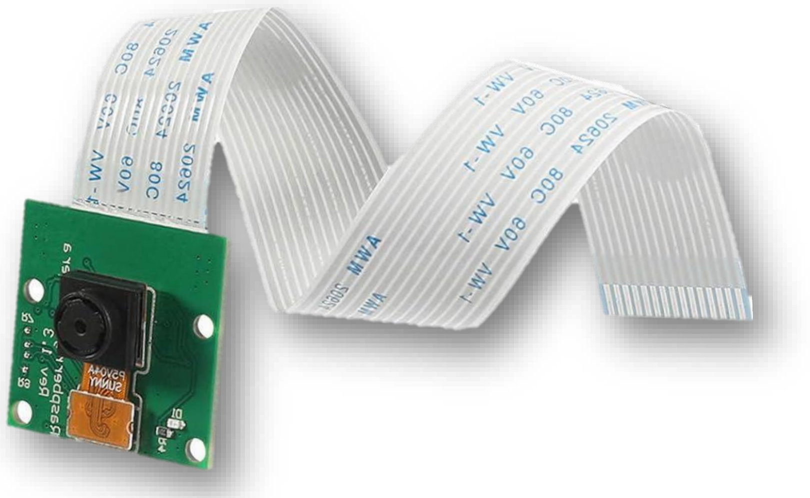


Figure 31. Camera Module V1.3 (5 Megapixel)

Component Description:

This is a 5-megapixel V1.3 camera designed
specifically for the Raspberry Pi. It captures images and records video at resolutions up to 5 megapixels and connects via the CSI (Camera Serial Interface).

Importance:

High-Resolution Imaging: Enables high-quality still images and video recording.
Computer Vision Projects: Useful for facial and object recognition applications.
Surveillance and Security: Can be integrated into simple monitoring systems.
Educational Projects: Ideal for learning camera programming with languages like Python.

Main Components:

1. **Sensor:** Typically, an Omni Vision OV5647 sensor (5 megapixels).
2. **Lens:** Focuses light onto the sensor, determining image quality.
3. **PCB:** Houses the sensor and necessary circuitry.
4. **IR Filter:** Improves image quality by filtering out non-visible light.
5. **Ribbon Cable:** Connects the camera to the Raspberry Pi’s CSI port.

.

4.6 MPU6050 GY-521 Sensor Module:

Figure 32. MPU6050 GY-521 Sensor

Component Description:

It is a sensor that combines the gyroscope sensor that measures the angle velocity around the X, Y, Z axles and the three-axis accelerometer sensor.

Importance:

Determine the direction and location of the robot by measuring the angular change rate relative to the axons and the acceleration in the three axes.
It is also used to maintain the robot's balance when walking and prevent it from stumbling by monitoring movement and vibration.

Main Components:

1. MPU6050 Chip: Contains the 3-axis gyroscope, 3-axis accelerometer, and digital motion processor.
2. Voltage Regulator: Supplies the proper voltage (typically 3.3V) while allowing for 3.3V or 5V input.
3. Pull-up Resistors: For the I2C lines (SDA and SCL).
4. Connection Pins: Includes SDA, SCL, VCC, GND, and sometimes additional pins like INT.
5. Digital Motion Processor Unit: Processes sensor data before transmitting it to the main controller.

How it works

- **Accelerometer:** when the device accelerates, a mass moves inside the chip ,which leads to a change in the electrical capacitance. This change is converted into digital values that represent the acceleration along the axis .
- **Gyroscope:** when the sensor rotates particles move inside the sensor , generating forces . these forces are measured as the rate of angular change.
- **Digital motion processor:** this module integrates gyroscope and Accelerometer data and gives the pitch angles , roll angles and yaw angles .

4.7 HC-SR04 Ultrasonic Sensor:

Figure 33. HC-SR04 Ultrasonic Sensor

Component Description:

It is sensor that measures the distance between objects using high -frequency sound wave technology (40 kHz)

Importance:

- **Accurate Distance Measurement:** Typically measures distances from approximately 2 cm to 400 cm.
- **Ease of Use:** Requires only Trigger and Echo pins to interface with a microcontroller.

- **Versatile Applications:** Used for obstacle avoidance, security systems, and automated controls.
- **Low Cost:** Provides reliable performance at an affordable price.

Main Components:

1. Ultrasonic Transmitter (T): Emits ultrasonic waves at a frequency of 40 KHz.

2. Ultrasonic Receiver (R): Receives the echo and converts it to an electrical signal.

3. Internal Control Circuit: Calculates the travel time and produces a measurable pulse.

4. Connection Pins: VCC (typically 5V), Trig, Echo, and GND.

How it works

- The controller sends a pulse to the TRIG terminal, then the transmitter sends 8 pulses at a frequency of 40 KHz into the air. When these waves collide with an object in front of the sensor , they return to the receiver.
- When the wave is received , the ECHO gives a pulse . the magnitude of this pulse represents the time of the wave's travel and return . to calculate the distance between the sensor and the body we use the following law:

$$d=\frac{t*0.0343}{2}$$

where:

t: time to go and come back unit second

0.0343: the speed of sound unit m/s

4.8 MG90S Servo:

Figure 34. MG90S Servo

Component Description:

The component shown is the MG90S servo, a compact servo motor equipped with metal gears for increased durability compared to plastic-gear servos.

Importance:

This motor is used to move the camera to obtain multiple viewing angles.

Chapter 5
The practical side

Chapter 5: The practical side

5.1 Introduction:

In this chapter we will review the work we have done and how to assemble the electronic components and the robot structure.

5.2 The structure :

The structure is made of aluminum and cut by CNC machine . The following figure shows a picture of the printed frame:

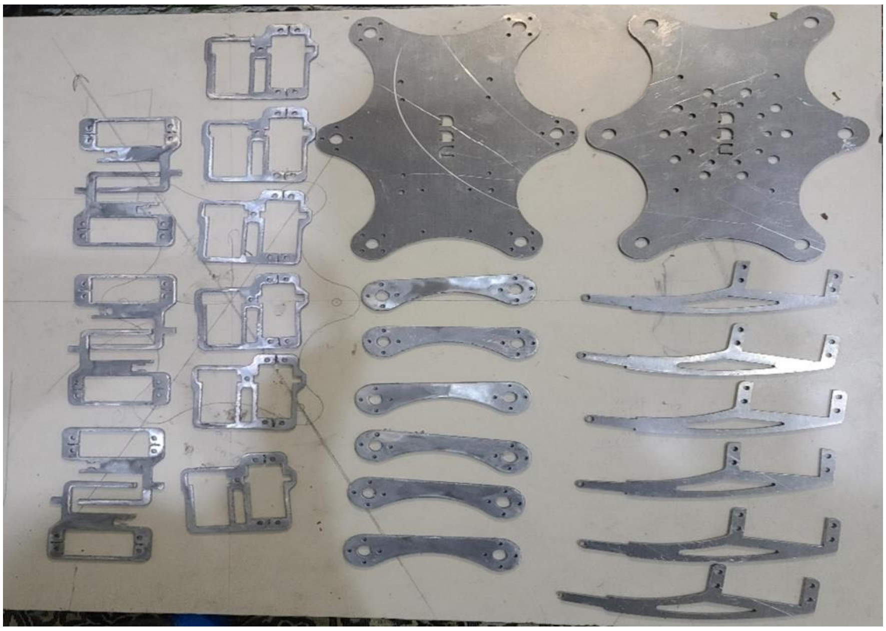


Figure 35. Hexapod robot structure

5.2.1 Leg Assembly:

The leg frame was assembled from the coxa , femur and tibia . three servo motors were installed in each leg and secured with screws . The following figure shows the stages of assembling the legs:

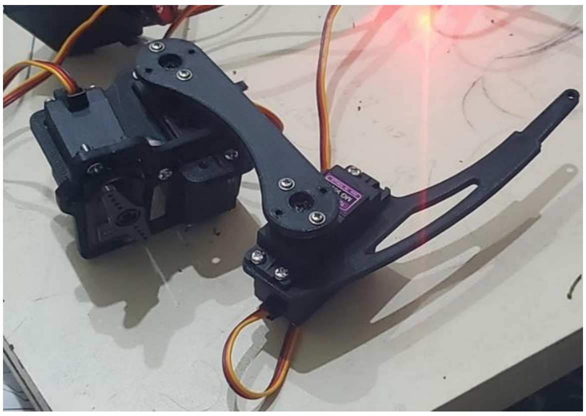


Figure 36. Hexapod Leg

Figure 37. Hexapod left Legs

5.2.2 Body Assemble:

The hexapod body of the robot is designed to carry the electronic components and to connect the legs to each other . The robot body consists of an upper body and a lower body , with fixing screws between them.

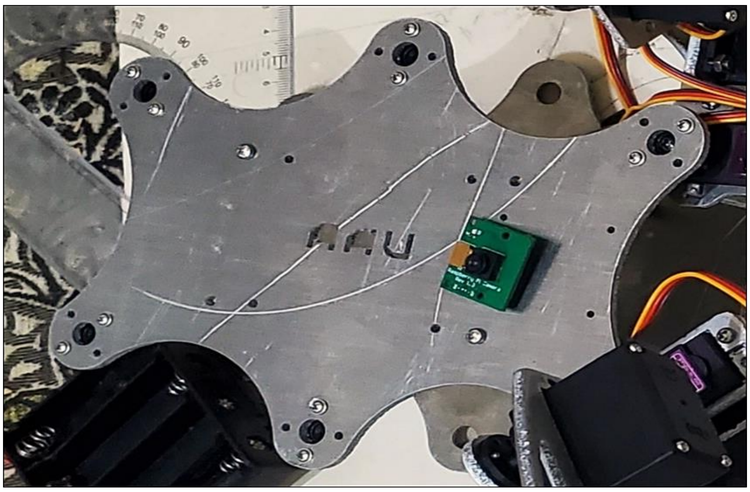


Figure 38. Hexapod Body

5.3 Electronic Assemble:

The electronic parts were connected to each other .as 18 servo motors were connected to the shield. We used two shield and connected them to each other , and then connected them to the raspberry pi 4.

As for the ultrasonic and the gyroscope, they are connected to the control ports on the raspberry pi 4.

Connect the camera to the raspberry pi through the CSI port and install the camera in a mobile base containing tow servo motors to move the camera and provide multiple shooting angles.

As for the power source for the raspberry pi, it is capacity for a 10000 mAh power bank with a voltage of 5V. As for the other electronic components , they are fed from three batteries , each with a capacity of 3200 mAh and voltage of 3.7v.

The following figure shows how to install electronic parts:

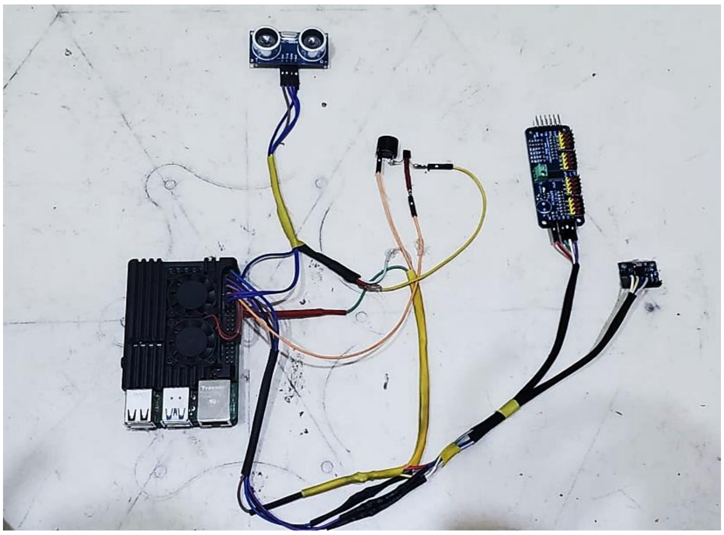


Figure 39. Electronic parts

5.4 Difficulties we faced in implementing the project:

1) The robot's body was made of "PLA" but this type of plastic could not withstand the high temperatures, so the robot's body was replaced with aluminum which resulted in additional financial costs . We made several attempts to print plastic. Our first attempt was with a 3D printer project of one of the university graduates, but the printer was not of the required accuracy. Then we tried to print the frame with acrylic using the university’s CNC machine, but the price of acrylic was not affordable. Then we printed PLA using one of the printers available in the local market, but the quality of the PLA was not good and it was damaged under the sun. Then, after a long search and the lack of availability of CNC laser machines for aluminum in abundance in the market, we found someone who owned this machine and we formed the aluminum according to the required design.

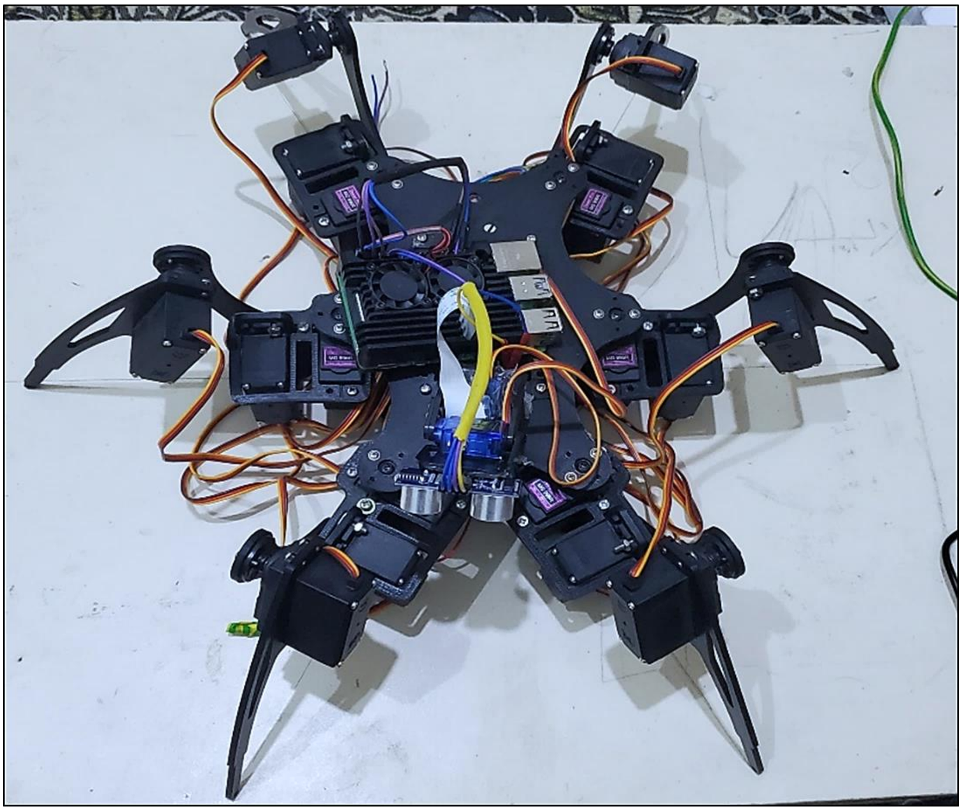


Figure 40. Previous PLA body

2) Lack of screw of the required size to install the servo motors and the robot chassis.
3) Some electronic devices are damaged due to their low quality.
4) The previous coxa design was prone to fracture , due to this issue, the design has been modified to a more dynamic and efficient model .

Figure 41. Previous coxa

Figure 44. New coxa

5) One of the problems we faced was the exposed wires , which caused the wires to be pulled out of place , which forced us to make insulators for these wires . the following figure shows a picture of the robot before and after the wires were insulated.

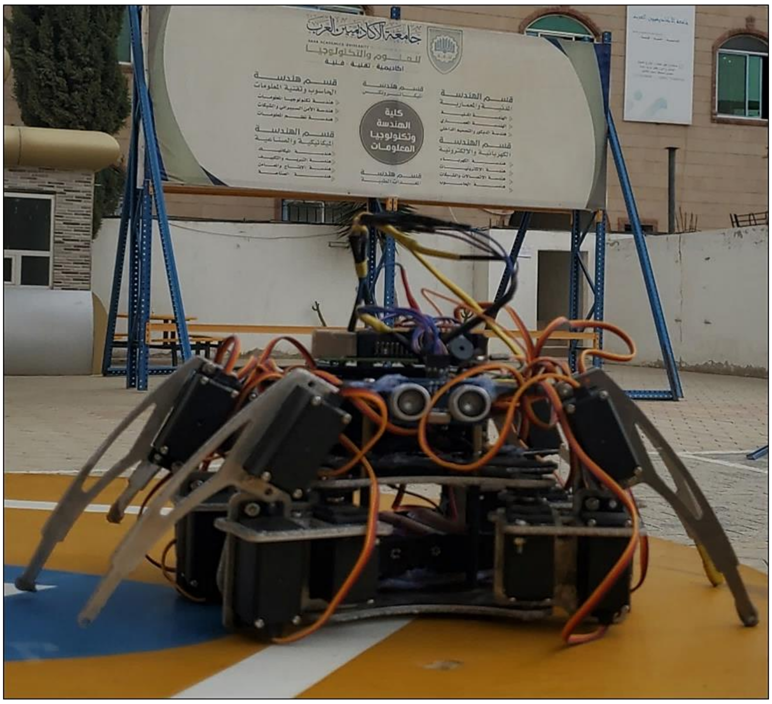


Figure 47. Hexapod shape before wire insulation

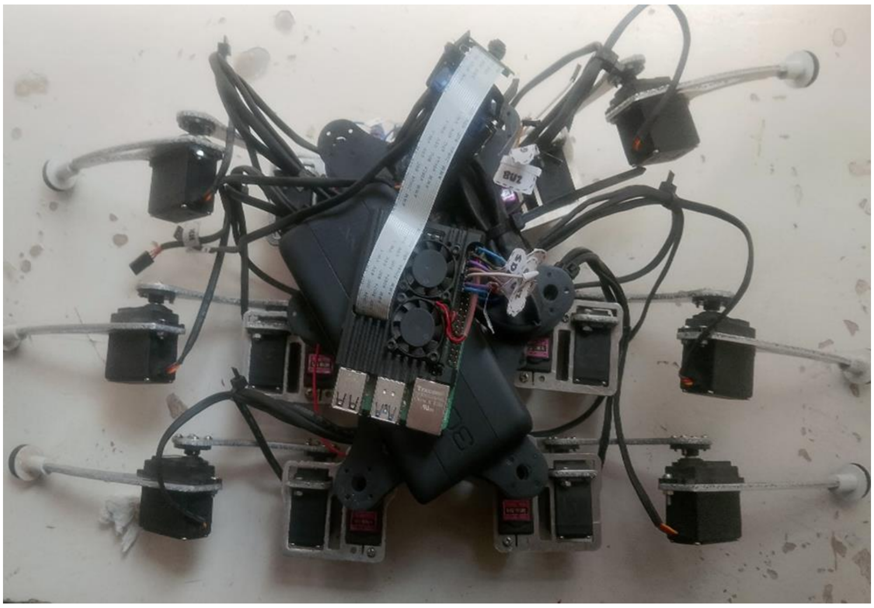


Figure 48. Hexapod shape after wire insulation

6) There was an issue with the hexapod robot slipping on smooth surfaces . after consulting with the project supervisor and the team a plastic component was designed and attached to the end of each leg to increase friction and prevent the robot from slipping .

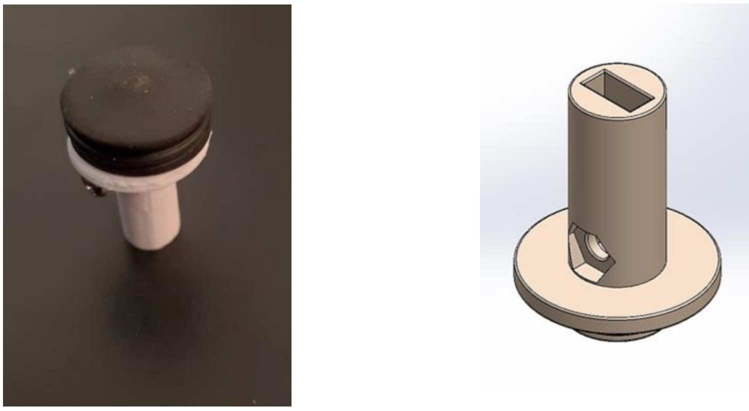


Figure 49. Shape of the piece

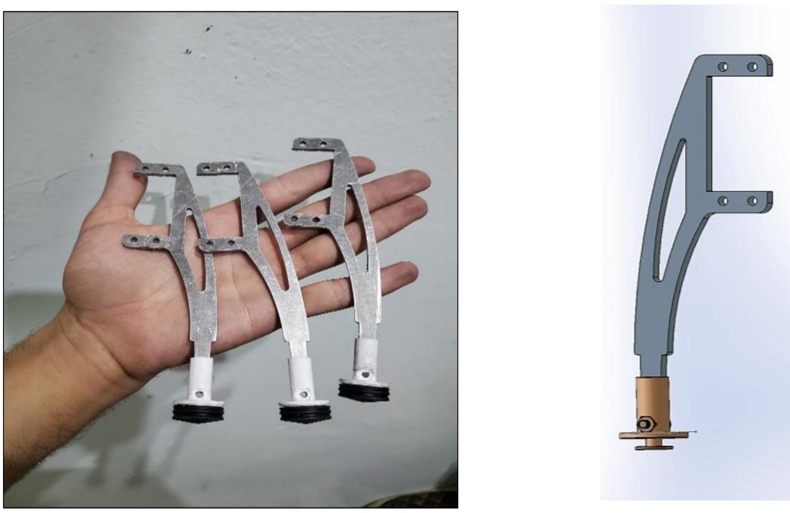


Figure 50. Piece location

5.5 Final product :

After a long search process , after addressing the errors that occurred during the implementation of the project, and after consulting the project supervisor and a number of academics and experts, we made this robot according to the specifications indicated in advance , and the following figure displays a picture of the implemented project.

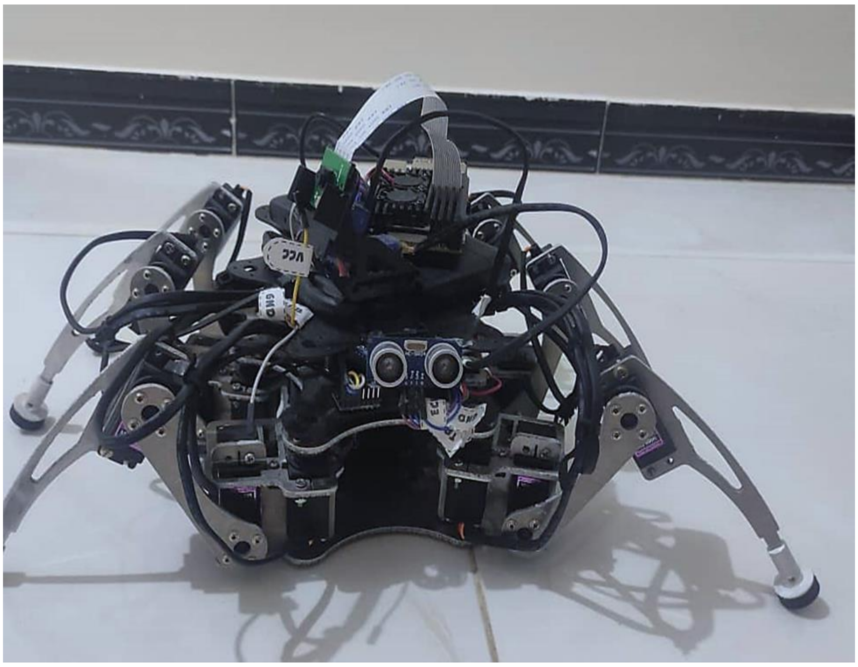


Figure 51. Final product

Chapter 6

The Results

and Discussion

Chapter 6: The Results and Discussion

6.1 Introduction:

In the chapter , we will talk about the results of the practical aspect that we did and the extent of their impact on the design of the robot . we will also talk about how the robot moves and the energy consumption diagram with the electrical study of the robot and we will discuss these results .

6.2 Spider walking methods:

There are two types of walking motions used in this robot which are :

6.2.1 Ripple gait hexapod robot:

The legs move sequentially , so that one leg moves at a time. This method is used in walking when bearing a certain weight and walking in rough places.
The following figure shows the ripple gait:

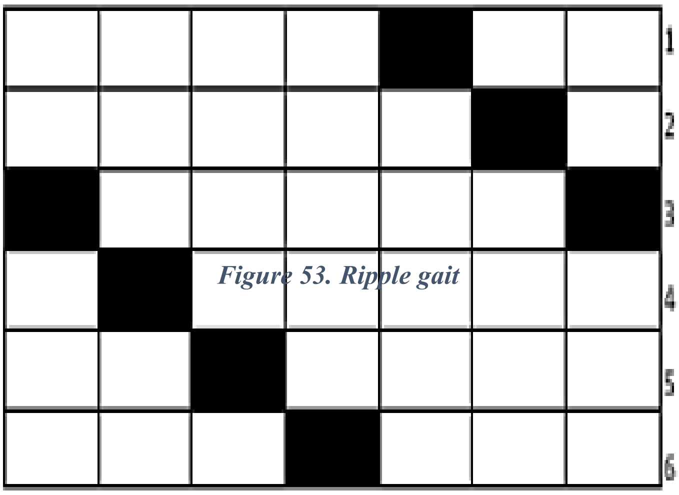


Figure 53. Ripple gait

Figure 53. Ripple gait

6.2.2 Tripod gait:

This gait is used in flat areas that require high speed and energy efficiency , as the front , back and middle legs move in the opposite direction with each step.

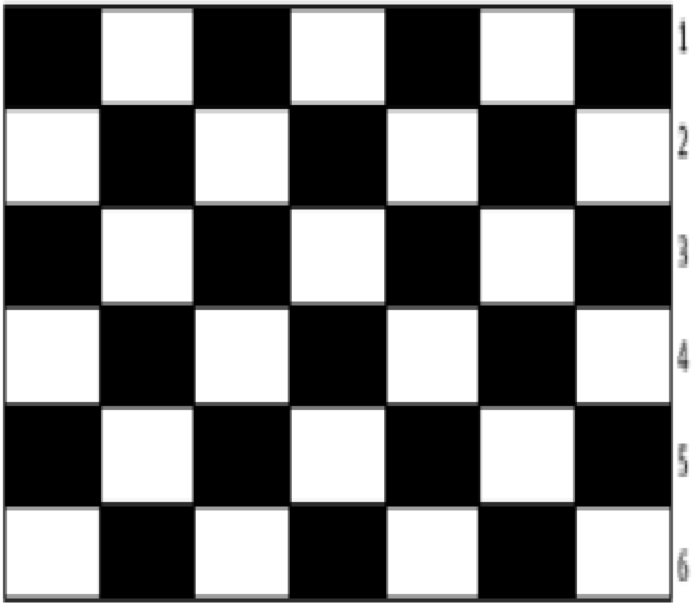


Figure 54. Tripod gait

The following table shows the most important differences between the walking styles used.

Table 27. comparison between types of steps

Comparison	Ripple gait	Tripod gait
speed	Lowest speed	faster
Energy use	Less efficient	More efficient
Usage	In the rugged places	In the flat places

6.3 Results Electrical study:

6.3.1 Factors contributing to increased energy consumption:

- kinematic errors: represents at least 13% of the total energy loss when using a three-legged gait pattern .
- energy consumption increases with speed , load and continuous operation.
- servo motors consume energy even in stationary mode , which leads to a constant rise in temperature .
- stride length : increasing it leads to greater speed but also higher energy consumption.

6.3.2 Electrical energy equations:

Average servo motor current = 0.66 [(4)]

- calculating the electrical power of servo motor
- first , we calculate the total current for 18 servo motor

$I_{18\,motor}$ = 18 × I_{ava}= 18× 0.66 =11.88 A [(5)]

- we calculate the electrical power of servo motor
- P_{18}=V×$I_{18\,motor}$ = 6× 11.88= 71.28 W

Where :

$\boldsymbol{P_{18}}$:electrical power of servo motor

$\boldsymbol{I_{18\,motor}}$:total current for 18 servo motor

V : voltage

We need several steps to find the appropriate battery capacity for the robot to operate . we will detail them below.

1- <u>We calculate the sum of the currents for all electronic components :</u>

I_{SD}10mA **I_{MPU}=5mA** **I_{HC}=15mA**

I_{TOTAL}= 11.88 +0.010+0.005+0.015=11.91 A

(4) We take it from datasheet.
(5) Fundamentals of Electric circuits.

Where:

I_{SD} **:** servo driver current

I_{MPU} **:** MPU6050 current

I_{HC} **:** Ultrasonic Sensor: current

I_{TOTAL} **:** total current

2- Total power calculation :

$$p_{total} = V \times I_{TOTAL} = 5 \times 11.91 = 59.55 \text{ watt}$$

Where :

p_{total} **:** Total power

V : voltage

3- Calculating the battery capacity required for a specific operating time :

Time = 0.5h

$$Ah = I_{TOTAL} \times T = 11.91 \times 0.5 = 5.955 \text{ Ah} = 5.995 \times 1000 = 5955\text{mAh}$$ [6]

Where :

T: Time

Ah : battery capacity

[6] Conventional current version.

Chapter 7
Conclusion and recommendations

Chapter 7: Conclusion and recommendations

7.1 Introduction:

we will discuss in this chapter the findings of this study and the recommendations we propose for further research development .

7.2 Recommendations:

This research began with the selection of the project ; A market survey was then conducted to determine the availability of the necessary electronic components . The project cost was determined . simulations were then conducted , components were purchased , the structure was designed, and the robot was tested in locations with varying terrain . the robot proved effective in traversing complex terrain . overall ,

This study aims to create a robot that can overcome difficult terrain and buy used in rescue operations . this robot could be developed in the future to be able to climb roofs and other ideas that contribute to improving the robot.

7.3 Possibility to develop:

This robot can be developed and improved in several aspects , the most important of which are the following :

- **Climbing capability :**

This robot is capable of climbing over any obstacle or surface and maintaining balance on rough terrains by adding suction cups its climbing ability can be further enhanced allowing it to scale vertical structures such as buildings.

- **Integration of sensors in hazardous areas :**

By equipping the robot with infrared sensors, it can detect dark and bright surfaces Additionally by incorporating sound sensors, the robot can detect acoustic signals using a sound detection module this enables it to navigate toward the source for human safety purposes in the event of earthquakes or fire incidents.

- **Addition of a robotic arm:**

By mounting a small robotic arm equipped with a drill on the upper surface of the robot its functionality can be significantly enhanced This arm can assist in disaster-stricken areas by manipulating or removing objects providing support in rescue and recovery operations.

- **Drone integration:**

A drone can be mounted on this robot to accelerate its mobility enhance rapid response to hazards and enable the execution of various aerial and ground operations.

- **Body wrap:**

Creating an external shell for the body will significantly enhance the visual appearance of the hexapod . it will also help protect the exposed electronics from damage.

7.4 Conclusion:

The final year project addressed various engineering concepts, including design, kinematics, materials science, CAD, and manufacturing. The project began with a literature review, which was the most critical step in designing the robot.

The literature review provided conceptual ideas that were later used in the robot's design. Research papers, websites, and various books were studied prior to the design phase (referenced in the References section). The design was carried out using the SolidWorks platform. Following the design, a basic speed analysis was conducted. This led to a market survey, which resulted in several modifications to the robot's design. Eventually, the robot had to be completely redesigned. After these steps, the parts were manufactured and printed using a CNC machine. The process is explained in the design and manufacturing chapter of this thesis.

Following fabrication, the control mechanism was developed. The robot's motion was tested and compared to theoretical calculations. The robot was then assembled and completed to a satisfactory degree, with room for future enhancements. We look forward to exploring additional improvements and further developing our mechanical knowledge and skills.

Overall, this thesis highlights the key stages in the development of a control system using model-based design. Due to the broad scope of this work, knowledge from multiple engineering and scientific disciplines was required, including electrical engineering, control systems, mathematics, physics, and programming. One of the main takeaways from this project is the importance of having a comprehensive understanding of the system to be designed when working with CAD tools.

Navigating terrain using a hexapod was a challenging task and could, in itself, be the subject of a full thesis. The enormous complexity of generating dynamic motion required far more time than was available during this project. However, despite the major effort invested in developing a model-based design platform, a functional walking pattern was successfully implemented. This demonstrates that model-based design and code generation can be effectively applied to a precise system such as a 6-legged robot with 18 degrees of freedom.

References:

1) “(PDF) RANCANG BANGUN KENDALI ROBOT HEXAPOD MENGGUNAKAN SMARTPHONE,” ResearchGate, May 2025, doi: 10.24912/tesla.v25i2.26714.
2) “(PDF) Multitasking Spider Hexapod Robot,” in ResearchGate, doi: 10.1109/RAAICON48939.2019.58.
3) “(PDF) MAKING OF A SPIDERBOT (HEXAPOD),” ResearchGate. Accessed: Aug. 09, 2025. [Online]. Available: https://www.researchgate.net/publication/376453564_MAKING_OF_A_SPIDERBOT_HEXAPOD
4) “(PDF) Direct Kinematic Analysis of a Hexapod Spider-Like Mobile Robot,” ResearchGate, Mar. 2025, doi: 10.4028/www.scientific.net/AMR.403-408.5053.
5) S. M. Khandaker and R. Acharjee, “An intelligent hexapod rescue robot,” Thesis, BRAC University, 2018. Accessed: Aug. 09, 2025. [Online]. Available: https://dspace.bracu.ac.bd:8443/xmlui/handle/10361/10171
6) C. Sun, M. Yuan, F. Li, Z. Yang, and X. Ding, “Design and Simulation Analysis of Hexapod Bionic Spider Robot,” J. Phys.: Conf. Ser., vol. 1168, no. 2, p. 022094, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022094.
7) M. H. Furqon, P. W. Ciptadi, R. H. Hardyanto, and E. H. Pratisto, “Design Build a Spider Robot Detector Fire Arduino based,” APPLIED SCIENCE AND TECHNOLOGY REASERCH JOURNAL, vol. 1, no. 2, pp. 39–45, 2022, doi: 10.31316/astro.v1i2.4644.
8) A. Naser and J. Hajaj, “Design, Implementation & Control of Autonomous Hexapod Robot for Search and Rescue,” 2023, Accessed: Aug. 09, 2025. [Online]. Available: https://hdl.handle.net/20.500.11888/18543
9) S. Nag, M. Chavan, R. Chogle, M. Solanki, and S. Shigwan, “Four Leg Spider Robot Using Arduino,” International Journal for Scientific Research and Development, vol. 9, no. 1, pp. 322–326, Apr. 2021, Accessed: Aug. 09, 2025. [Online]. Available: https://ijsrd.com/ http://www.ijsrd.com/
10) J. Hirnšal, “Návrh řídícího systému pro Hexapod s Raspberry Pi,” Dec. 2018, Accessed: Aug. 09, 2025. [Online]. Available: https://dspace.tul.cz/handle/15240/49734
11) “(PDF) Adaptive Intelligent Spider Robot,” ResearchGate. Accessed: Aug. 09, 2025. [Online]. Available:

https://www.researchgate.net/publication/386863905_Adaptive_Intelligent_Spider_Robot

Appendix A — Symbols and Units (Quick Reference):

Symbol	Meaning	Unit
L_1, L_2, L_3	Link lengths (Coxa/Femur/Tibia)	mm
a, b, c	Joint angles (about x, y, z)	rad (internal) / deg (servo)
$\mathbf{x} = [x, y, z]^\top$	Foot position in leg frame {L}	mm
$\mathbf{q} = [a, b, c]^\top$	Joint vector	rad
$\mathbf{J}$	Linear Jacobian $\partial\mathbf{x}/\partial\mathbf{q}$	mm/rad
$\mathbf{f}$	Foot wrench (force)	N
$\boldsymbol{\tau}$	Joint torques	N·mm (N·m/1000)
ϕ_i	Leg base yaw (deg)	deg
$\Delta x_i, \Delta z$	Leg base offsets	mm
F	Subdivisions per gait cycle	–
Z	Swing height	mm
s	User speed input	–
α	Body yaw command for turning	deg
H	Body height offset	mm

Appendix B — Detailed Derivations:

B.1 FK derivation.

Take joint-2 and joint-3 in a plane orthogonal to joint-1 axis; the in-plane projection has radius $r = L_1 + L_2 \cos b + L_3 \cos(b + c)$ and axial offset $x = L_2 \sin b + L_3 \sin(b + c)$. Rotating this radius by joint-1 angle a about x distributes r into $y = \sin a\, r$, $z = \cos a\, r$, giving (4-2).

B.2 IK derivation.

Given (x, y, z), first recover a from the yz-plane (4-3). Subtract the L_1 contribution along $(\sin a, \cos a)$ to form the triangle between joints 2–3–foot with side lengths L_2, L_3, ℓ_{23}. The law of cosines and sines give (4-5).

B.3 $\dot{\mathbf{J}}$.

Differentiate (4-2) to get (4-7); then differentiate each entry w.r.t. time using $\dot{S}, \dot{T}$ as given. The final $\dot{\mathbf{J}}$ follows by product rule.

Appendix C — Additional Worked Tables:

C.1 PWM counts vs angle (50 Hz, 1–2 ms).

θ (deg)	t_{pulse} (μs)	Count
0	1000	204.8
45	1250	256.0
90	1500	307.2
135	1750	358.4
180	2000	409.6

C.2 Example foot references for turning ($\alpha = 15°$, $x_c = 10$ mm, $y_c = 0$ mm, $F = 74$):

Leg	$\mathbf{p}_i$ (mm)	(δx, δy) mm/it
1	(110, 192)	(0.61, −0.29)
2	(220, 0)	(−0.08, −0.32)
3	(110, −192)	(−0.61, −0.29)
4	(−110, −192)	(−0.61, 0.29)
5	(−220, 0)	(0.08, 0.32)
6	(−110, 192)	(0.61, 0.29)

(Values rounded; intended for intuition.)

Appendix D — Validation Protocol Templates:

1. **FK↔IK sweep:** uniformly sample 3D points in (4-6); compute IK then FK; log RMS and max errors.
2. **Jacobian finite-difference:** perturb each joint angle by 10^{-4} rad; compare columns.
3. **Gait consistency:** verify no commanded foot leaves the reachable set; monitor stability margin (4-25).
4. **Torque headroom:** for a grid of postures and $F_n \in [0,10]$ N, compute $\boldsymbol{\tau}$; ensure margins to servo torque limits.
5. **PID step tests:** ±5° tilt commands; measure overshoot and settling vs K_p, K_d.

Appendix E — Calibration Checklists:

- **Servo neutral:** set each joint to its midpoint pulse (e.g., 1500 μs) and mechanically align links; record per-leg offsets $(\delta a_i, \delta b_i, \delta c_i)$.
- **Angle limits:** enforce mechanical stops in software (e.g., $[0°, 180°]$ at the servo; narrower if needed).
- **IMU alignment:** place robot level; calibrate gyro biases; compute tilt from accelerometer; set reference attitude.
- **Ultrasonic:** measure against a tape at 0.5–2.0 m; fit a linear correction if needed.
- **Gait tuning:** start with small Z and large F; increase step length gradually while tracking current draw and stability margin.