AN-NAJAH NATIONAL UNIVERSITY

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

Computer Engineering Department

Hardware Graduation Project

Hexaplorer

Presented in partial fulfilment of the requirements
for Bachelor degree in Computer Engineering

Student Name(s): Asem Diab, Mohamad Salha
Supervisor Name: Dr. Aladdin Masri

June, 2025

Abstract

Hexaplorer is a six-legged robotic platform designed for navigating hazardous or inaccessible environments such as disaster zones and industrial infrastructures. Its hexapod design offers superior terrain adaptability compared to wheeled robots, utilizing forward/inverse kinematics and Bézier curve-based gait generation for lifelike, stable movement. Dynamic balance is maintained via a PID-controlled MPU6050 gyroscope, while dual IR and leg-mounted contact sensors enable responsive terrain interaction. Powered by 18 servo motors controlled through PCA modules and an Arduino Uno, Hexaplorer is remotely operated via a React Native mobile app and features live video streaming through an ESP32-CAM. This integration of intelligent software and robust hardware enables agile, adaptive exploration in complex environments.

Acknowledgment

We gratefully acknowledge the support and encouragement of Dr. Aladdin Masri throughout this project. We also thank our families and friends for their constant motivation.

Disclaimer

Contents

List of Figures

Chapter 1: Introduction

The Hexaplorer project was born from two core ideas: "Hexa," for its six-legged, spider-like structure, and "Explorer," reflecting its mission to navigate and investigate environments too hazardous or inaccessible for humans. This includes complex infrastructures like tunnels, HVAC systems in large buildings, and disaster zones. The inherent versatility of hexapod robots extends beyond basic locomotion, enabling them to perform various multitasking operations for diverse applications [1].

We opted for a hexapod design over traditional wheeled robots because of its superior terrain versatility. While wheeled robots are limited to smooth, predictable surfaces, Hexaplorer's six independently articulated legs — each with 180-degree motion — allow for agile and adaptive locomotion over uneven terrain and complex obstacles. This capability proved vital in navigating debris-laden areas. The choice of a hexapod design, inspired by resilient biological structures, has also led to advancements in robot locomotion, including novel bionic leg structures and body frames designed for adaptability [2]. Central to our movement strategy was the implementation of forward kinematics (FK), used to determine leg positions based on joint angles, and inverse kinematics (IK), which calculated the necessary joint angles to precisely position a leg at a given coordinate. These calculations enabled complex and smooth leg movements tailored to each step, allowing the robot to execute various gaits with improved stability and precision.

From a hardware perspective, Hexaplorer is driven by 18 servo motors, each chosen for its torque capabilities and current demands that can significantly increase when facing an obstacle it cannot overcome. These are controlled via PCA servo driver modules, interfaced with our primary controller, the Arduino Uno. A buck converter handles voltage regulation, supported by a separate 5V regulator for peripherals like IR sensors. Logic level shifters were integrated to ensure safe communication between 5V sensors and the Arduino Uno's 3.3V logic. A significant challenge emerged six days before our deadline when some of the initial 28 AWG wires failed under high current loads, burning out. We quickly reconnected and redesigned the system, successfully switching to more durable 24 AWG wiring and using solder sleeves for reliable, compact connections. The successful hardware implementation of autonomous hexapod spider robots, like Hexaplorer, relies on careful selection and integration of mechanical and electronic components, forming a robust foundation for intelligent behavior [3].

To enhance environmental awareness, we integrated dual IR sensors, mounted at different heights, to distinguish between low and high obstacles. Each leg is also equipped with a custom-built push-button contact sensor to detect terrain contact in real time, feeding into our adaptive climbing logic. These sensors significantly enhanced the robot's responsiveness and accuracy by providing real-time feedback during kinematic calculations. The chassis, based on designs by Sir KuhnHero, was 3D printed with strong, flexible material to withstand stress and reduce breakage. Friction and vibration in the leg joints were mitigated using ball bearings, which proved essential for accurate motion, especially when tuning our custom PID (Proportional-Integral-Derivative) control for balance stabilization.

The robot uses an MPU6050 gyroscope to track orientation, providing critical feedback for dynamic stability during walking or turning. We overcame early accuracy issues by adopting a specialized library, which was effective when the sensor IC faced upward—a hardware anomaly we accepted due to consistent results.

Our software, written in C++ via the Arduino IDE, communicates with a React Native mobile app through a Node.js backend hosted on Glitch. Commands are relayed via Firebase, with the ESP32-CAM providing live video streaming. This app allows remote control over Wi-Fi, forming the foundation for future autonomous operation. Modern hexapod robot designs increasingly incorporate IoT-based control systems, enabling remote operation and enhancing their versatility for a wide range of multipurpose applications [4]. Motion-wise, Hexaplorer supports multiple gaits, such as tripod and wave gaits, with each leg's motion path computed using Bezier curves to ensure fluid, lifelike movement. FK and IK calculations were vital in planning these curves and ensuring each leg accurately reached its intended position. Timing sequences were fine-tuned through iterative testing to achieve a natural pace while maintaining system performance.

PID control is further used to stabilize tilt during walking. Our codebase was modularized to streamline development, with dedicated classes like Leg, encapsulating behaviors such as push(), lift(), and getStatus(), enabling easy integration with movement routines. Hexaplorer's behavior is also shaped by intelligent decision-making algorithms. Based on sensor feedback, it adjusts posture, elevates itself, or seeks alternate paths depending on obstacle height and complexity. Thanks to its powerful servos and robust frame, it can carry loads heavier than its own body weight.

In summary, Hexaplorer is a testament to the power of intelligent software and robust hardware, built to explore challenging environments using advanced motion strategies powered by inverse and forward kinematics, intelligent gait planning, and real-time sensor feedback. Its live video feed, adaptive navigation, and PID-stabilized motion make it more than a prototype—it's a resilient foundation for future exploration and autonomy, with potential applications in disaster relief, inspection, or research in inaccessible terrains.

Chapter 2: Theoretical Background and Previous Work

Hexapod robots have been widely researched due to their inherent stability and adaptability across uneven terrain [5]. Early platforms such as *RHex* and *Lauron* introduced biologically inspired locomotion models, showcasing the advantages of legged systems over wheeled counterparts in unstructured environments. These early systems typically relied on predefined gait sequences or reactive control strategies, offering robust locomotion but limited adaptability to dynamic challenges. Some bio-inspired hexapod designs have even explored novel locomotion methods, such as rolling to enhance energy efficiency [6].

A fundamental component of legged robotics is kinematics. In particular, forward kinematics (FK) determines the end-effector (foot) position based on known joint angles, whereas inverse kinematics (IK) is used to compute the required joint angles to achieve a specific foot position. For robots with multi-jointed legs operating in 3D space, IK calculations are often computationally intensive. To address this, our design uses simplified geometric IK methods that maintain acceptable accuracy while reducing computational load, allowing real-time response during movement and climbing [7].

Control strategies like PID (Proportional-Integral-Derivative) control have been extensively employed to maintain balance and correct deviations due to external disturbances. In *Hexaplorer*, PID control was implemented with heuristic tuning specific to our terrain and leg configuration, ensuring responsive and stable motion. While PID control offers robust and widely adopted solutions for maintaining balance, research also explores alternative methods such as fuzzy logic, with comparative studies evaluating performance across various hexapod designs [8]. Furthermore, advanced hexapod systems often integrate sophisticated control methods like fuzzy logic to optimize gait patterns for specific terrain or performance goals [9]. While systems like Boston Dynamics' *Spot* or the *ANYmal* robot employ cutting-edge methods such as SLAM, dynamic path planning, and sensor fusion, our project draws on these theoretical principles within a more constrained, cost-effective setup — incorporating custom electronics, affordable microcontrollers, and an efficient gait engine.

In developing *Hexaplorer*, we reviewed prior student projects and academic literature to understand common challenges and inform our design. For example, previous student-built hexapods reported difficulty climbing even minor obstacles like books. While these struggles were not thoroughly documented, they emphasized a key issue that *Hexaplorer* set out to address using improved leg articulation, optimized climbing algorithms, and real-time foot feedback mechanisms. Numerous projects also demonstrate the feasibility of controlling complex hexapod locomotion and gait generation using accessible microcontrollers like the Arduino, highlighting its utility for educational and research platforms [10].

A directly comparable project, the "Obstacle Course Hexapod Robot", aimed to create a lightweight robot capable of tunnel navigation. Their system featured 3D-printed PLA parts and carbon fiber rods for lightweight strength and used 18 micro servos (three per leg), plus a front-mounted time-of-flight sensor on a sweeping servo for basic perception. Although a switch to a laser-cut frame improved maneuverability, they faced persistent problems with long leg designs, such as servo backlash, overheating, and friction issues, particularly during climbing attempts. Their solution involved retrofitting leg ends with furniture bumpers for better traction — but the robot ultimately failed to complete the obstacle course.

Learning from their experience, *Hexaplorer* uses shorter, mechanically optimized leg segments and joint articulation constraints to minimize servo stress. Additionally, we introduced push-button feedback sensors on each foot to improve terrain sensing and balance during movement [11]. These features have already enabled *Hexaplorer* to successfully traverse low obstacles and demonstrate promising results on stair-like barriers — an improvement over their outcomes.

On the theoretical side, the survey "Trends in the Control of Hexapod Robots" offers a comprehensive overview of the state of the art, including adaptive gait algorithms, model-based trajectory planning, and AI-assisted movement strategies [12]. Although these techniques often require powerful processors and complex software architectures beyond our scope, they served as a benchmark for situating our simpler implementation — which relies on PID control, preprogrammed gait cycles, and sensor-triggered responses — within the broader development landscape. The survey also highlighted opportunities for future development, such as integrating sensor fusion or smarter path-planning algorithms.

Regarding leg control, the paper "Design of 3 DOF Hexapod Leg Movement Using Inverse Kinematics" provided practical mathematical insight into controlling three-degree-of-freedom legs using trigonometric calculations [7]. The authors' clear formulation using tangent and cosine functions reinforced our own IK modeling and ensured smooth and accurate foot placement — critical for successful walking and climbing. Another study also focused on the design and simulation analysis of hexapod bionic spider robots, providing theoretical insights into structural resilience and balance [13].

Finally, the study "Design and Simulation Analysis of Hexapod Bionic Spider Robot" (2019) inspired aspects of our mechanical design. It focused on stress distribution and joint flexibility, using simulation to evaluate structural resilience and balance in a biologically inspired hexapod [14]. While their work was mostly theoretical, it validated our choices around joint placement, leg geometry, and balance dynamics. Unlike their simulation-only approach, however, *Hexaplorer* has been tested in real environments, and those trials have directly influenced design refinements.

In summary, the development of *Hexaplorer* is grounded in both foundational robotics principles and lessons learned from related practical and academic efforts. Through careful design decisions and validation against prior work, our robot aims to push the boundaries of what affordable, accessible hexapods can achieve in real-world obstacle navigation.

Chapter 3: Methodology

This chapter describes the theoretical foundations used in the project and provides insight into the motion planning and control techniques implemented on the hexapod robot. The methodology is divided into three parts: theory, hardware implementation, and software integration. This section focuses on the theoretical basis.

0.1 Theory

0.1.1 Kinematics of the Hexapod

The motion of the hexapod robot is governed by both forward and inverse kinematics. Each leg has three degrees of freedom: **coxa (hip)**, **femur (thigh)**, and **tibia (shin)**. To control the leg's tip (end-effector), we calculate the angles for these joints based on a desired position using inverse kinematics.

Forward kinematics is used when the joint angles are known, and we want to determine the position of the leg tip in 3D space. This involves summing the rotations and translations of each joint starting from the base.

Inverse kinematics, on the other hand, is used when we specify a desired position of the leg tip and want to calculate the necessary joint angles to reach that position. It is the core algorithm that allows the robot to walk, shift weight, and recover balance.

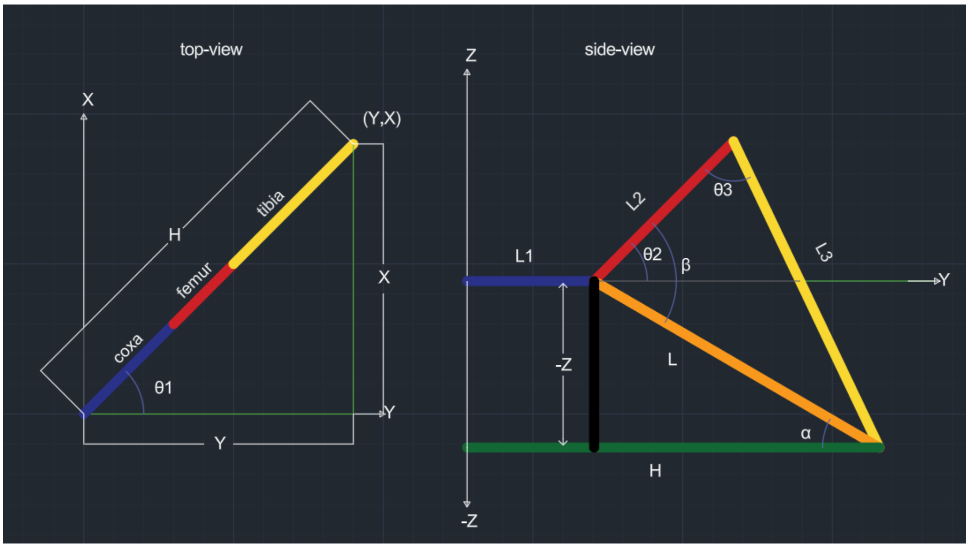

Figure 1: Side and top view of a single leg showing the joint structure

The basic inverse kinematics equations for a 2D planar arm (extendable to 3DOF) are as follows:

$$\theta_1 = \tan^{-1}\left(\frac{x}{y}\right) \tag{1}$$

$$H = \sqrt{x^2 + y^2} \tag{2}$$

$$\beta = \cos^{-1}\left(\frac{L^2 + L_2^2 - L_3^2}{2L_2L}\right) \tag{3}$$

$$\alpha = \tan^{-1}\left(\frac{\text{-}Z}{H - L_1}\right) \tag{4}$$

$$\theta_2 = \beta + \alpha \tag{5}$$

$$\theta_3 = \cos^{-1}\left(\frac{L_2^2 + L_3^2 - L^2}{2L_2L_3}\right) \tag{6}$$

0.1.2 Trajectory Generation using Bézier Curves

To generate smooth walking trajectories, Bézier curves were used. These curves allow for the creation of elegant, continuous paths for the leg tip, avoiding jerky motions and emulating natural walking behaviors.

The cubic Bézier curve equation is:

$$P(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3, \quad 0 \le t \le 1 \tag{7}$$

Where P_0 is the starting point, P_3 is the target point, and P_1, P_2 shape the curve. This allows for height control during leg lift and smooth movement over the terrain.

0.1.3 PID Control for Balance

The robot uses a PID controller to maintain balance based on feedback from a gyroscope placed at its center. The gyroscope detects angular displacement in pitch (α) and roll (β), and PID controllers adjust the leg heights accordingly. Research consistently highlights the importance of robust posture control methods for hexapod robots navigating rugged terrain [5].

$$u(t) = K_pe(t) + K_i\int_0^t e(\tau)d\tau + K_d\frac{de(t)}{dt} \tag{8}$$

Where:

> $e(t)$ is the angular error (difference between actual and desired orientation), K_p, K_i, and K_d are the proportional, integral, and derivative gains.

Each leg's vertical offset is computed using:

$$z = x \cdot u_\alpha + y \cdot u_\beta \tag{9}$$

Where x and y are the leg's position relative to the center, and u_α, u_β are the PID outputs for pitch and roll.

0.2 Software

The system architecture of the hexapod robot is built around a modular interface structure, with each module responsible for a specific functionality. Numerous projects demonstrate the feasibility of controlling complex hexapod locomotion and gait generation using accessible microcontrollers like the Arduino, highlighting its utility for educational and research platforms [10, 11]. One of the core components is the `Leg` class, which encapsulates all operations and data structures related to a single robotic leg. Within this class, an `enum Status` is defined to represent the current action state of a leg. These states include: `Lift`, `Push`, `Ideal`, `Asc_z`, `Dec_z`, `Z_modified`, `HIGH_STEP`, and `Lift_Down`.

The `Lift` and `Push` states are typically used together to simulate human-like walking patterns and are essential for implementing gaits such as wave and tripod. These movements rely heavily on pre-implemented inverse and forward kinematics, along with Bézier curve interpolation, to generate smooth and realistic trajectories. The `Ideal` state indicates that the robot is stationary or in a resting posture. The `Asc_z` and `Dec_z` states allow for individual leg height adjustments, enabling dynamic terrain adaptation and posture control. The `Z_modified` state serves as a temporary z-position modification, typically following an `Asc_z` or `Dec_z` operation. The `HIGH_STEP` state was originally intended for stair climbing, but was not used in the final implementation; however, it is retained for future development. The `Lift_Down` state is responsible for lowering a leg onto the surface after being lifted.

To support leg movement tracking, a structure named `LegStatus` was implemented. This structure records both current and previous positions of the leg in 3D space (x, y, z), along with the current `Status`. It includes an `isBalanced` flag, which serves as a feedback mechanism indicating whether the robot's posture is stable. This flag plays a critical role in high-level decision-making, especially on uneven terrain or during transitional movements.

In the `.cpp` file corresponding to the `Leg` class, initial servo angles are calculated using inverse kinematics. Leg motion is divided into discrete frames—referred to as frames per motion—to minimize vibrations and increase precision by interpolating transitions over time.

The `XYAngleInterface` module handles data acquisition from the gyroscope. It communicates via a bus configured at 115200 bps to retrieve pitch, yaw, and roll values. This orientation data is passed to the main control system to assist in dynamically adjusting leg heights and positions for stability.

A PID control system was implemented in `PIDController.h`, consisting of two independent controllers: `alphaPID` for pitch and `betaPID` for roll. These are used in the `autoBalance()` function. Real-time tilt data from the MPU6050 sensor is processed by the controllers to compute corrective actions. Each controller includes proportional (Kp), integral (Ki), and derivative (Kd) terms to handle immediate, accumulated, and predicted errors. The resulting correction values—`factorAlpha` and `factorBeta`—are used to adjust the z-position of each leg relative to its spatial location. Each balance cycle begins with a call to `reset_error()` to clear prior accumulated errors.

Servo control is managed by the `ServoDriver` class, implemented in both `servoDriver.h` and `servoDriver.cpp`. This class interfaces with two PCA9685 I^2C-based PWM drivers, each controlling up to 16 servo channels. During initialization, the I^2C bus is configured using GPIO21 (SDA) and GPIO22 (SCL) on the ESP32, with a PWM frequency set to 60 Hz—suitable for standard servos. The `setServoAngle()` function provides fine-grained control over each servo by converting logical angles into PWM signals using a linear mapping defined in `angleToPWM()`. This abstraction layer allows for synchronized and safe control of all 18 servos without exposing low-level I^{2}C logic.

The `MotionManager` class is the main orchestrator for motion logic. It integrates forward and inverse kinematics with Bézier-based trajectory shaping to compute real-time motion paths. Each leg features three degrees of freedom, enabling full articulation for walking and climbing. The system uses a frame-based execution model, where movement is interpolated over discrete time steps. Quadratic Bézier curves are used for lift and place actions, while linear interpolation handles push phases. Each leg runs its own internal state machine to determine its phase—whether lifting, pushing, or idle. Contact sensors provide real-time feedback and trigger changes in internal flags such as `onland`, helping each leg adapt to terrain changes or balance loss. These decisions influence the calculated angles, which are passed through the `AngleConverter` and ultimately sent to the servos via the `ServoDriver`.

0.3 Hardware

The hardware system is composed of several essential components that enable the hexapod's functionality. Below are the main components used:

1) PCA9685 PCA9685 is a 16-channel, 12-bit PWM controller that can be used to drive LED drivers and servo motors by communicating with I^{2}C bus. PCA9685 is an IC that is integrated and can be used as I^{2}C-bus regulated PWM driver with on-chip permanent clock that can be utilized to drive servo motors, LEDs, or other PWM-driven devices. PCA9685 has 16 independent PWM channels that each offer 12-bit resolution (4096 steps) that ensure accurate control of devices connected.

The controller communicates via the I^{2}C protocol with just two of the host microcontroller pins (SDA and SCL) and is much less wasteful in terms of pin usage than direct PWM control. The controller features a supply voltage range of 2.3V to 5.5V and can generate PWM frequencies from 24 Hz up to 1526 Hz. The device has multiple I^{2}C addresses, and multiple PCA9685 controllers can exist on the same I^{2}C bus at the same time.

PCA9685 controllers are also frequently employed in robot uses, LED lighting control uses, and servo motor control uses where several PWM outputs are desired. PCA9685 provides benefits of microcontroller pin usage saving, hardware reserved for PWM creation, and it can even be possible to maintain PWM output accessible if the primary microcontroller is occupied with other tasks.

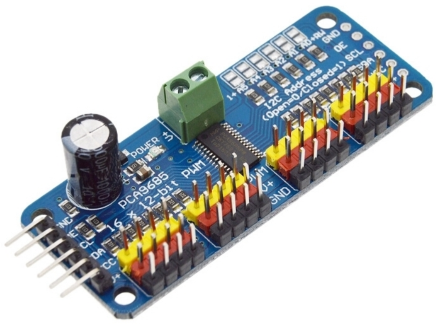

Figure 2: PCA9685 Module

The PCA9685 is employed for the control of the servo motors in the hexapod robot system. Since 18 servo motors are required to enable full leg movement control (3 servo motors per leg × 6 legs), and each PCA9685 provides 16 channels, it is utilized to power all the servo functions except those requiring more precise control. The PCA9685 is connected to the primary microcontroller through the I^2C bus, through which position commands are received and subsequently converted into accurate PWM signals for servo regulation. During gait execution, servo positions are transmitted from the microcontroller to the PCA9685, and appropriate PWM signals are generated to enable coordinated leg movement. This configuration ensures smooth and synchronized servo actions while also offloading the main microcontroller, allowing it to focus on tasks such as sensor processing and gait planning.

2) Servo DS558HV DS558HV is a digital servo motor used in robotics and remote control systems where accurate positioning control and high torque are needed. The digital control circuit of the servo motor gives a more precise, hold torque, and response time than analog servos. The DS558HV works in a voltage supply of 4.8V to 7.4V with a maximum torque of 14 kg.cm and is ideal for use in rotational force-demanding devices.

The servo features metal gears for added ruggedness and reduced backlash and a coreless motor for silent operation and fast response. For 180degree rotation, only simple PWM control with 900-2100 microseconds pulse width is supported. Benefits include the increased noise resistance and improved reliability(for varied loads) over a typical analog servo.

Digital servos like the DS558HV are popular inrobotics, radio-controlled models, robot arms and walking robots where the high torque is required along with the accurate joint control. Digital servos provide benefits such as higher positioning accuracy, improved response speed, higher torque holding, and an improved performance under various operational conditions.

Figure 3: DS558HV Servo Motor

We use 18 DS558HV servo motors to control every motion of our joint-based hexapod robot. The 6-legs spider iscomprised of three servo motors for coxa (hip), femur (thigh), and tibia (shin) joints. The 14 kg. cm torque rating ismore than enough to carry the weight of the robot and enough force to climb on different terrain. The servos are controlled with PWM control signals that come from the PCA9685 controllers, which convert movement commands from the main microcontroller into accurate angular positions. Whenever walking gaits are executed by the hexapod, all the servos react to position instructions in order to coordinate the leg movement in a smooth and stable manner. The digital control system ensures quality performance and accurate positioning during the cycle of operation of the robot with accurate joint control for complex walking patterns and terrain negotiation.

3) ESP32 WROOM The ESP32-WROOM is a high-performance wireless module designed for embedded systems that require both Wi-Fi (802.11 b/g/n) and Bluetooth (v4.2 BR/EDR and BLE) connectivity. At its core, the module integrates the ESP32-D0WDQ6 microcontroller, which features a dual-core 32-bit Xtensa LX6 processor running at up to 240 MHz, along with 520 KB of SRAM and 4 MB of external flash memory. It also incorporates essential hardware such as a crystal oscillator, antenna matching circuit, and RF shielding, making it a complete solution for compact and efficient wireless designs. The module supports a wide range of peripheral interfaces including GPIO, UART, SPI, I^{2}C, PWM, ADC, and DAC, enabling communication with various sensors, actuators, and external devices. It operates within a voltage range of 3.0V to 3.6V and includes multiple power management modes (modem-sleep, light-sleep, and deep-sleep) to optimize energy efficiency in battery-powered applications. These features make the ESP32-WROOM highly suitable for low-power IoT devices, real-time embedded systems, and smart electronics requiring reliable wireless performance and moderate processing capabilities. Thanks to its versatility and robust performance, the ESP32-WROOM is widely adopted in IoT systems, smart home devices, wearables, and industrial automation. It supports development through popular environments like ESP-IDF, Arduino IDE, and PlatformIO, with programming options in C/C++, MicroPython, or Lua. Its combination of wireless connectivity, peripheral support, and cost-effectiveness makes it ideal for both academic projects and scalable commercial deployments.

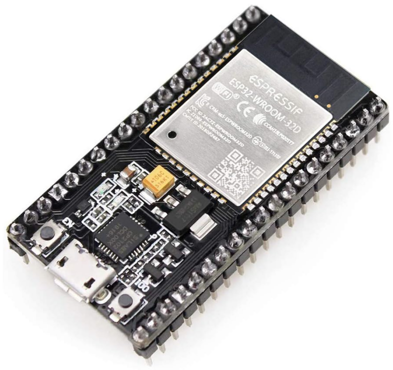

Figure 4: ESP32 WROOM Microcontroller

The ESP32 Wroom module was chosen as an improved alternative to the Arduino Uno due to various limitations experienced with the latter. Its lower clock speed and restricted memory in the Uno were found insufficient for the demands of the project. The limited number of pins on the Arduino Uno, each assigned specific functions, complicated the connection of multiple components, including two servo driver boards controlling 18 servo motors, a gyroscope, two IR sensors, and six push buttons (which were later disconnected to reduce computational complexity related to leg segment measurements). Furthermore, the Arduino Uno and its required USB-to-serial programmer occupied more physical space. These challenges were overcome by the ESP32 Wroom, which provided higher processing speed, expanded memory, and a more compact size. The built-in Wi-Fi capability of the ESP32 Wroom facilitated wireless communication and easier code deployment. Additionally, the use of a USB Type-C interface simplified programming and power delivery. Therefore, the ESP32 Wroom was effectively integrated to meet the performance and connectivity needs of the hexapod spider robot.

4) ESP32-CAM The ESP32-CAM is a specialized development board that combines the powerful ESP32-S microcontroller with an integrated camera module, creating a compact solution for computer vision and surveillance applications. Built around the ESP32-S chip with dual-core processing capabilities, the module features 4MB of PSRAM and supports external microSD card storage up to 4GB for image and video data. The board includes an OV2640 camera sensor capable of capturing images at resolutions up to 2 megapixels (1600x1200) and recording video in various formats including JPEG, BMP, and GRAYSCALE, making it suitable for both still photography and real-time video streaming applications. The ESP32-CAM maintains all the wireless connectivity features of standard ESP32 modules, supporting both Wi-Fi (802.11 b/g/n) and Bluetooth protocols for seamless data transmission and remote control capabilities. The module includes multiple GPIO pins for interfacing with external components such as LEDs, sensors, and servo motors, though some pins are reserved for the camera interface. It operates on a 5V power supply and includes an onboard antenna for wireless communication, though it also supports external antenna connections for improved range. The board features programmable flash LED functionality and supports various power management modes to optimize battery life in portable applications. Due to its integrated camera capabilities and wireless connectivity, the ESP32-CAM has become increasingly popular in security systems, home automation, robotics, and IoT projects requiring visual monitoring or image processing. The module can be programmed using familiar development environments like Arduino IDE and ESP-IDF, with extensive library support for camera functions, web server implementation, and image processing algorithms. Its applications range from simple surveillance cameras and doorbell systems to more complex projects involving facial recognition, motion detection, and remote monitoring solutions, making it an excellent choice for developers seeking an affordable yet capable camera-enabled microcontroller platform.

Figure 5: ESP32-CAM Module

The ESP32-CAM was utilized as one of the core components of the robot, functioning as its primary visual system. It was strategically placed at the front center of the robot's body to ensure optimal field of view while being physically protected within the frame. The module was programmed to stream live video to a Glitch-based server using Node.js, leveraging its Wi-Fi capabilities for wireless transmission. A React Native mobile application was developed to interface with the system, providing control buttons for robot movement while displaying the live video feed in the background. This integration enabled real-time remote monitoring and control, effectively turning the ESP32-CAM into the robot's "eye" and enhancing its interactive and autonomous functionalities.

5) DROK 300W DROK 300W is a high-efficient step-down (buck) DC-DC power conversion module with the feature to convert higher input voltage to lower output voltage with the potential for high current output. The switching power supply module can be capable of input voltages that are usually between 6V and 40V and offers adjustable output voltage between 1.2V and 36V with the maximum power output of 300 watts. The converter achieves high efficiency levels of up to 95

The module offers short circuit protection, thermal protection, and overcurrent protection for operation under all loads safely. The module has a high-frequency switching topology that minimizes the size of filtering components and maintains stable output voltage regulation. The converter combine potentiometer trimming for the accurate setting of output voltage and current limiting and LED indicating for the indication of operating status.

DC-DC buck converters like the DROK 300W are used in robotics, automotive electrical systems, LED lighting and battery-powered appliances where voltage transformation is required. They offer high efficiency, small size, adjustable output parameters, as well as protection capabilities, and are used to supply power to a wide range of servo motors and electronic devices in robots.

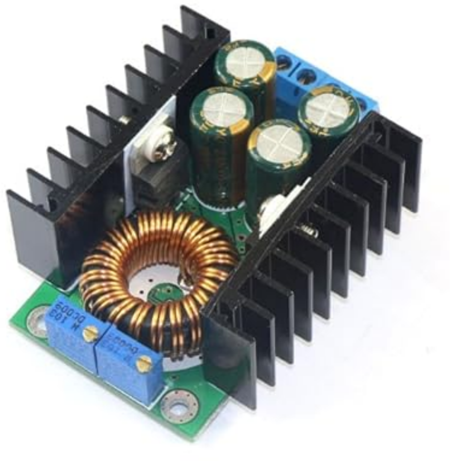

Figure 6: DROK 300W Power Converter

We use the DROK 300W DC-DC converter to transform the voltage source of the 12V battery to 8V in order to drive all 18 of our hexapod robot system servo motors. As the DS558HV servos are configured for optimal usage in their voltage range and a combined 18 servos will consume high current when operated in parallel, the 300W specification provides us with sufficient power delivery capability. The converter takes 12V from the main battery pack and supplies a consistent 8V that powers all of the servo motors via the distribution system. The voltage conversion is important because it enables us to employ a high voltage battery system in order to achieve increased energy density without giving the servos their correct operating voltage. High converter efficiency reduces the loss of power and heat generation while the system is in operation, which is important for the reliability of the system when in continuous operation as the hexapod executes complex walking patterns and maneuvering.

6) XL4015 The XL4015 is a high-efficiency step-down (buck) DC-DC converter module designed to provide stable voltage regulation for electronic circuits requiring lower voltages than their input power source. Based on the XL4015E1 switching regulator IC, this module can handle input voltages ranging from 4V to 38V and deliver adjustable output voltages from 1.25V to 36V with a maximum output current of 5A under optimal conditions. The converter operates at a fixed switching frequency of 180 kHz and incorporates a high-efficiency design that typically achieves 80-96The XL4015 module utilizes pulse-width modulation (PWM) switching technology combined with an external inductor and capacitive filtering to smooth the output voltage and minimize ripple. The board includes an adjustable potentiometer for precise output voltage control, along with LED indicators for power status monitoring. Key components on the module include the main switching MOSFET, Schottky diode for freewheeling current, input and output filter capacitors, and the feedback network for voltage regulation. The compact PCB design incorporates proper thermal management with adequate copper pour areas to dissipate heat generated during high-current operation, making it suitable for continuous duty applications requiring reliable power conversion. The XL4015 buck converter finds extensive use in various electronic projects including battery-powered systems, LED driver circuits, motor control applications, and voltage regulation for microcontroller-based projects. Its wide input voltage range makes it particularly valuable for automotive applications, solar charging systems, and industrial equipment requiring stable power supplies from variable input sources. The module's high current capability and efficiency make it ideal for powering high-performance microcontrollers, single-board computers like Raspberry Pi, and multi-servo robotics projects where multiple voltage rails are needed. Its cost-effectiveness and ease of integration have made it a popular choice among hobbyists, students, and professional engineers developing power management solutions for embedded systems and IoT devices.

Figure 7: XL4015 Voltage Regulator

In the project, the XL4015 buck converter was employed to step down a 12V power supply to a regulated 5V output, which was then used to power several components requiring 5V input, such as the IR sensors and the gyroscope. While these components operate at 5V, many of the underlying integrated circuits, particularly in modules like the gyroscope, contain onboard low-dropout (LDO) voltage regulators to internally convert the 5V supply to 3.3V, which is necessary for their internal logic. This configuration ensured stable and noise-free operation without overloading sensitive components. The use of the XL4015 allowed for efficient power distribution across the system, reducing heat dissipation and power loss compared to linear regulators, while maintaining the necessary current for reliable sensor performance.

7) IR Obstacle Avoidance Sensor Module The IR Obstacle Avoidance Sensor Module is a compact and cost-effective proximity detection device that uses infrared light reflection to identify nearby objects. It consists of an infrared emitter that sends out IR light and a receiver that detects this light when reflected off surfaces within the sensor's field of view. Operating at a voltage range of 3V to 5V, the sensor is highly compatible with popular microcontroller platforms such as Arduino, Raspberry Pi, and other embedded systems. Most modules offer a simple three-wire interface (VCC, GND, and digital OUT), making it easy to integrate into various electronic projects. The detection range of the module, typically spanning from 2 cm to 30 cm, can be adjusted using an onboard potentiometer, which also allows users to fine-tune the sensor's sensitivity. The digital output pin provides a LOW signal when an obstacle is detected and a HIGH signal when the path is clear, enabling straightforward integration with microcontroller input pins. Additional features include onboard indicator LEDs for power and detection status, which simplify troubleshooting and system monitoring. Despite variations in labeling—especially on modules manufactured in China—the core functionality remains consistent across suppliers. These sensors are widely adopted in robotics and automation due to their simplicity and effectiveness in short-range detection. Common applications include autonomous vehicles, obstacle-avoiding robots, line-following systems, and smart home devices. They perform best in controlled indoor lighting and are sensitive to surface reflectivity; shiny or light-colored objects are detected more easily than dark, matte surfaces. Their affordability, ease of use, and reliable functionality have made them a staple in educational kits and DIY electronics projects alike.

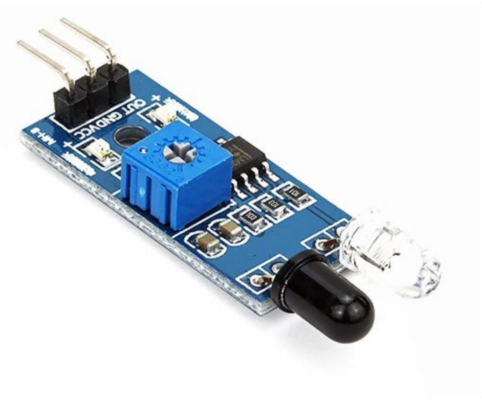

Figure 8: IR Obstacle Avoidance Sensor

For our hexapod spider robot, we used two IR obstacle avoidance sensors to help it detect obstacles and decide whether it could climb over them. We placed one sensor near the top of the robot's body and the other near the bottom, both facing forward. This setup let the robot figure out how tall an obstacle was. The way it worked was simple: the robot would lift itself until both sensors no longer detected anything, meaning the path ahead was clear. If the top sensor detected something but the bottom one didn't, it meant the robot was facing stairs or a low object it could probably climb. But if both sensors detected something, that meant the obstacle was too tall, so the robot would stop and look for another way around. This system gave our robot a smarter way to handle tricky terrain and make better climbing decisions.

8) MPU6050 (GY-521 Module) The MPU6050, commonly available on the GY-521 breakout board, is an advanced 6-axis motion tracking sensor that combines a 3-axis accelerometer and a 3-axis gyroscope in a single chip. It also includes an onboard Digital Motion Processor (DMP) that helps process motion data efficiently. With 16-bit analog-to-digital converters on each channel, the MPU6050 captures highly accurate measurements along the x, y, and z axes simultaneously. This Micro Electro-Mechanical Systems (MEMS) device measures acceleration, angular velocity, orientation, displacement, and other motion-related parameters. Operating on a supply voltage from 3.3V to 5V, the module features built-in pull-up resistors for I2C communication, making it easy to connect to microcontrollers like Arduino. Communication with the MPU6050 occurs primarily over the I2C bus, with support for serial communication protocols as well. The accelerometer measures forces up to ±16g, while the gyroscope detects angular velocities up to ±2000°/s. Both sensors offer programmable full-scale ranges to adapt to different needs. The integrated Digital Motion Processor fuses accelerometer and gyroscope data to calculate orientation angles such as yaw, pitch, and roll, significantly reducing the processing burden on the host microcontroller. The GY-521 breakout board simplifies wiring with clearly labeled pins: VCC, GND, SDA, SCL, along with interrupt and auxiliary I2C pins. The MPU6050 module is widely used in robotics, drone stabilization, virtual reality controllers, and gesture recognition projects where precise motion sensing is critical. Thanks to its high accuracy, low power consumption, and onboard processing, it is especially useful in battery-powered devices, wearable electronics, and autonomous navigation systems. Its ability to provide real-time orientation and motion data has made it a staple component in quadcopter flight controllers, self-balancing robots, and motion-controlled smartphone applications.

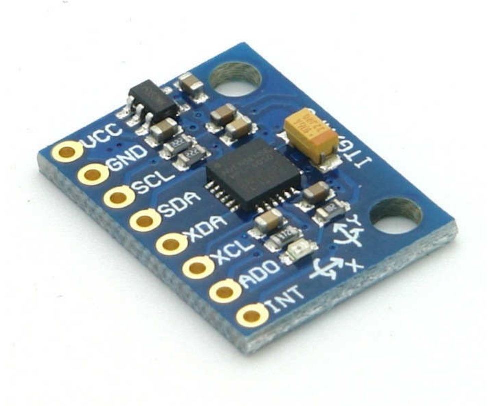

Figure 9: MPU6050 Gyroscope and Accelerometer

The MPU6050 sensor was used to measure the pitch and roll angles of the hexapod spider robot, which were referred to as alpha and beta. High-speed angle readings were provided by the sensor's built-in Digital Motion Processor, allowing smooth and accurate data collection. These angles were continuously read to enable the robot's posture to be automatically adjusted, keeping it balanced and level on uneven surfaces. Although a small margin of error exists in the measurements, it is considered negligible and does not significantly affect performance. Additionally, the temperature is measured by the MPU6050 as an extra feature. Multiple rapid samples are taken and averaged to improve stability and precision when controlling the robot's movements.

9) Ball Bearing Ball bearings are small mechanical components used to reduce friction in rotating parts by replacing sliding motion with rolling. They consist of two rings (inner and outer) with small steel or ceramic balls in between, held in place by a cage. This setup helps parts spin smoothly and efficiently, with less resistance and wear. In robotics, they're commonly used in wheels, joints, motors, and any moving part that needs reliable rotation. Ball bearings not only improve motion but also help reduce power use and heat buildup. Depending on the type of load, different bearings are used—like deep groove for radial loads, angular contact for both radial and axial, and thrust bearings for axial-only loads. Choosing the right size, material, and lubrication is important for performance and lifespan. Proper installation and care also play a big role. Whether in a DIY robot or an industrial system, ball bearings help ensure smoother, longer-lasting, and more efficient movement.

Figure 10: Ball Bearing

18 ball bearings were used—one for each motor in the robot. They were essential for reducing friction between the joints and motors, minimizing vibrations in the legs, and improving overall stability. This allowed the robot to move more smoothly and deliver more accurate and consistent performance.

10) Arduino Uno The Arduino Uno is a microcontroller board based on the ATmega328P and is widely used in embedded system and robotics applications due to its simplicity, affordability, and robust community support. It features 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. It operates at 5V and provides sufficient GPIO access for controlling various sensors, actuators, and modules in basic robotic systems.

The Arduino Uno communicates with peripherals using standard interfaces such as I^2C, SPI, and UART. Although it supports PWM signals, its limited number of PWM-enabled pins (6) and relatively low processing power (16 MHz clock speed, 2 KB SRAM) restrict its ability to control a large number of actuators simultaneously. It also relies on a separate USB-to-serial converter for programming, which adds to its physical size and wiring complexity in multi-component systems.

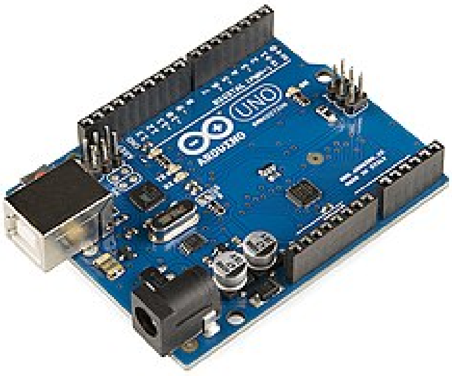

Figure 11: Arduino Uno Board

The Arduino Uno was initially employed in the hexapod robot project for managing servo motor control and sensor integration. It served as the first microcontroller platform to coordinate the robot's locomotion and sensor feedback. However, due to several limitations—including insufficient memory, restricted processing speed, and a limited number of usable pins—it became challenging to integrate and manage all required peripherals such as two servo driver boards for 18 servos, a gyroscope, IR sensors, and push buttons. These constraints led to the temporary replacement of the Arduino Uno with the ESP32 Wroom module, which offered enhanced performance, more GPIOs, built-in Wi-Fi, and a compact form factor suitable for the robot's physical layout.

Despite the switch, the Arduino Uno was reconsidered for specific roles later in development, particularly for offloading simpler tasks or acting as a co-processor for peripheral management, owing to its ease of use and reliable performance in simpler control loops. This strategic reuse helped in modularizing control logic and simplifying debugging without overburdening the ESP32.

Chapter 4: Results and Analysis

Our hexapod spider robot project successfully demonstrated several practical outcomes, even though there were no direct numeric performance metrics. The main achievements and observations are as follows:

1. Walking Gaits and Mobility:

The robot effectively achieved a triple walking gait using both inverse and forward kinematics. This enabled it not only to walk forward but also to move in four directions, showcasing advanced maneuverability. Implementing the inverse kinematics method allowed precise control of each leg's position and orientation, which was crucial for stable multi-directional walking.

2. Balancing and Stability:

A significant challenge was developing the balancing algorithm, which relied on data from a gyroscope sensor combined with a PID controller to correct angular deviations. The controller adjusted the height of each leg in real time to maintain balance. Since each leg has three joints, coordinating the correct angles for all joints simultaneously was complex. However, by sequentially balancing each leg, the robot maintained stability .

3. Environmental Exploration:
The robot was equipped with a camera for exploration purposes. To ensure visibility in dark environments such as tunnels, an LED light was mounted alongside the camera. This setup allowed the robot to "see" in low-light conditions, enhancing its operational versatility.

4. Motion Smoothness:
To further improve the robot's movement fluidity, we integrated Bezier curves with the inverse kinematics approach. This resulted in smoother, more natural leg trajectories during walking, reducing abrupt motions and mechanical strain.

5. Hardware Challenges:
Despite these successes, one of the motors burned out due to excessive load, even though it was rated to handle 2 Amps. This incident highlighted the need for better current management and possibly selecting more robust components. We also identified that upgrading to the ESP32 microcontroller from the Arduino significantly improved system performance, module capacity, and space efficiency.

6. Obstacle Detection and Navigation:
The IR sensors performed effectively in detecting obstacles in the robot's path. We constructed a prototype demo stair set to test climbing capabilities. Although initial tests were promising, the motor failure set back progress, and currently, the robot can move forward and backward with some difficulty. We continue to troubleshoot and improve this functionality.

7. Power and Wiring Considerations:
The project underscored the importance of appropriate wiring capable of handling higher currents and the potential need for a better buck converter to stabilize the power supply. However, time constraints due to the exam period limited the implementation of some hardware upgrades.

Chapter 5: Discussion

The Hexaplorer project offered a comprehensive practical exploration into the design, implementation, and challenges of a multi-legged robotic system. This discussion section interprets the results presented in Chapter 4, critically evaluates the effectiveness of our design choices, elaborates on the limitations encountered, and contextualizes our work within the broader field of hexapod robotics, setting the stage for future improvements.

5.1 Performance Interpretation and Design Effectiveness

Our practical trials unequivocally validated the core design philosophy of Hexaplorer, demonstrating its capability for adaptive locomotion in unstructured environments. The successful implementation of a **triple walking gait** and multi-directional movement highlighted the advantages of a hexapod design over wheeled platforms for navigating complex terrain. The efficacy of our **simplified geometric Forward and Inverse Kinematics (FK/IK)** was paramount, enabling precise leg positioning and facilitating smooth, lifelike movements even with the Arduino Uno's computational constraints. This precision was crucial for agile navigation over low obstacles and debris. The integration of **Bezier curves** further enhanced motion fluidity, minimizing abrupt transitions and promoting smoother, more natural leg trajectories.

The **PID controller** effectively maintained the robot's dynamic stability. Despite requiring iterative, heuristic tuning in real-world scenarios, it consistently corrected angular deviations detected by the MPU6050 gyroscope, contributing significantly to its ability to traverse slightly uneven surfaces. Critical real-time feedback from the dual-height **IR sensors** and custom **push-button contact sensors** on each foot enabled effective obstacle differentiation and terrain contact detection, directly enhancing the robot's responsiveness and accuracy during adaptive movements. The integration of the **ESP32-CAM with an LED light** also successfully fulfilled the robot's exploration mandate, providing essential visual feedback in low-light environments like simulated tunnels.

—

5.2 Critical Analysis of Challenges and System Limitations

The development of Hexaplorer was marked by several significant challenges, each providing crucial learning experiences and highlighting inherent limitations that inform future development.

- **Hardware Resilience and Power Management:** A major challenge arose from a servo motor burning out due to excessive load, despite its 2A rating. This incident critically underscored that **peak current draws under stress can significantly exceed rated continuous currents**, necessitating more robust current management and motor selection. The subsequent failure of the initial **28 AWG wiring** during high-current demands further emphasized the absolute necessity of selecting appropriate wire gauges. The upgrade to **24 AWG wiring** and solder sleeves was a crucial, albeit reactive, solution that drastically improved system reliability. Additionally, observations indicated a need for a more robust buck converter to ensure stable power delivery under fluctuating servo loads.

- **Microcontroller Selection and Impact:** The pragmatic decision to revert from the more powerful ESP32 to the **Arduino Uno** due to project deadlines introduced computational constraints. While the Uno managed current tasks, its comparatively slower processing speed, reduced memory, and fewer GPIO pins limited the scope for implementing more sophisticated algorithms, such as advanced sensor fusion or complex path planning, highlighting a current ceiling on our system's computational complexity.

- **Obstacle Navigation and Stair Climbing Limitation:** Despite success with low obstacles, Hexaplorer was **unable to successfully navigate stair-like structures**. This limitation primarily stems from the current leg design's insufficient vertical reach for typical step heights and the absence of specialized gait algorithms for stair ascent/descent. The complexities of real-time perception of step geometry further complicated this challenge. The motor failure during initial stair-climbing attempts underscored the intense mechanical demands of such maneuvers.

- **MPU6050 Practicality:** An unexpected hardware anomaly required the **MPU6050 gyroscope** to be mounted with its IC facing upward for accurate readings. While a specialized library provided consistent results, this unconventional mounting highlights a design constraint and the importance of thorough sensor calibration for reliable orientation data.

—

5.3 Comparison with Previous Work

Our project's design and outcomes were significantly informed by insights from previous work in legged robotics. Analysis of student-built hexapods, particularly the "Obstacle Course Hexapod Robot," provided valuable lessons. Their reported issues with long leg designs, servo backlash, overheating, and failure to complete obstacle courses directly influenced our mechanical design. Hexaplorer employed **shorter, mechanically optimized leg segments** and judicious **joint articulation constraints** to minimize servo stress and improve mechanical efficiency. Crucially, the integration of real-time **push-button contact sensors** on each foot provided direct, real-time terrain feedback that was absent in their approach. This enhanced sensory input allowed Hexaplorer to achieve demonstrably better performance in navigating varied low obstacles and showing promising results on simulated stair-like barriers, distinguishing our solution by directly addressing and overcoming previously documented practical failures. While high-end systems like Boston Dynamics' Spot operate on a much larger scale with advanced techniques like SLAM, Hexaplorer effectively demonstrates how core principles of robust locomotion and adaptive behavior can be achieved within a more constrained and cost-effective setup, making it a valuable platform for accessible robotics research.

—

5.4 Implications and Future Directions

The Hexaplorer project has established a solid foundation for further research into resilient and intelligent legged robotics. The successful implementation of kinematics, balance control, and basic environmental sensing within a student-level budget demonstrates the potential for accessible platforms in complex environments. Future work should directly address the identified limitations and expand upon current capabilities:

- **Advanced Stair Climbing:** This remains a key target, requiring redesigned legs for increased vertical reach, integration of depth sensors (e.g., LiDAR or a small depth camera) for precise step detection, and development of dedicated stair-climbing gaits and control algorithms.

- **Enhanced Autonomous Navigation:** Transitioning beyond remote control necessitates implementing Simultaneous Localization and Mapping (SLAM) for real-time environment mapping and self-localization, coupled with sophisticated path planning algorithms and advanced sensor fusion techniques.

- **Hardware and Power System Optimization:** A full transition to a more powerful microcontroller like the ESP32 would support computationally intensive algorithms. Further investment in more robust motors and a redesigned, stable, and efficient power distribution system with appropriate overcurrent protection is also crucial.

- **Vision-Based Interaction:** Utilizing the ESP32-CAM beyond live streaming for on-board image processing tasks, such as object recognition or visual landmark detection, would significantly enhance navigational capabilities.

Chapter 6: Conclusions and Recommendations

The Hexaplorer project represents a successful endeavor in developing a resilient and adaptable hexapod robot capable of navigating challenging environments. This chapter synthesizes the key achievements of the project and outlines essential recommendations for future work, building upon the insights gained from its design, implementation, and rigorous testing.

6.1 Conclusions

The Hexaplorer project successfully demonstrated the feasibility and advantages of a hexapod design for versatile locomotion over uneven and debris-laden terrains, affirming its potential for exploration in human-inaccessible areas. Key conclusions drawn from this project include:

- **Effective Kinematic Control:** The implementation of simplified geometric Forward and Inverse Kinematics proved highly effective in achieving precise, real-time control over each leg's position, enabling stable multi-directional walking and agile navigation over low obstacles. The integration of Bezier curves further refined leg trajectories, resulting in smooth and natural motion.
- **Robust Stability Management:** The heuristically tuned PID control system, combined with feedback from the MPU6050 gyroscope, successfully maintained the robot's balance and stability across various movements and slightly uneven surfaces, validating its importance for dynamic legged locomotion.
- **Adaptive Environmental Interaction:** The dual-height IR sensors and custom push-button contact sensors provided critical real-time environmental feedback, enabling Hexaplorer to adapt its posture and gait based on obstacle presence and terrain contact. This demonstrated a foundational capability for intelligent navigation.
- **Resilience in Hardware Design:** The project underscored the critical importance of robust electrical and mechanical design, particularly highlighted by the incident of motor overload and inadequate wiring. The swift adaptation to more robust wiring and components ensured project continuity and provided invaluable learning experiences in practical engineering problem-solving.
- **Functional Remote Operation:** The integrated software architecture, comprising the Arduino Uno, React Native mobile app, Node.js backend, and Firebase, established a reliable remote control system with live video streaming, providing essential human oversight and a platform for future autonomous development.
- **Validation Against Prior Work:** Hexaplorer's design choices, particularly the optimized leg segments and use of contact sensors, directly addressed and significantly improved upon limitations observed in comparable student-built hexapod projects, demonstrating a clear advancement in practical obstacle negotiation.

Overall, Hexaplorer, despite operating within resource constraints and overcoming significant challenges, stands as a testament to the power of integrating fundamental robotics principles with adaptive design to create a functional and capable exploration platform.

6.2 Recommendations for Future Work

Building upon the current capabilities and addressing the identified limitations, the following recommendations are proposed to enhance Hexaplorer's performance and expand its operational scope:

- **Achieve Comprehensive Stair Climbing Capability:** This is the foremost recommendation. It requires:
  - Redesigning leg mechanisms to increase vertical reach and provide greater mechanical advantage for lifting the robot's body.
  - Integrating advanced perception sensors, such as a small LiDAR unit or stereo cameras, for precise 3D mapping of stair geometry.
  - Developing and implementing sophisticated gait algorithms specifically optimized for stair ascent and descent, incorporating dynamic balance control for multi-step transitions.
- **Transition to a More Powerful Microcontroller:** Fully migrating the control system to the ESP32 microcontroller is essential. This upgrade will provide:
  - Increased processing power and memory to support more complex kinematic calculations, advanced sensor fusion, and higher-level decision-making algorithms.
  - Additional GPIO pins and integrated features for easier expansion with new sensors and communication modules.
- **Enhance Autonomous Navigation and Intelligence:**
  - Implement Simultaneous Localization and Mapping (SLAM) for real-time environment mapping and self-localization, enabling the robot to navigate unknown terrains.
  - Develop and integrate sophisticated path planning algorithms that can generate optimal routes while autonomously avoiding complex obstacles.
  - Explore advanced sensor fusion techniques to combine data from the gyroscope, IR sensors, and new perception systems for a more robust understanding of the environment.
- **Improve Hardware Robustness and Power Management:**
  - Invest in more robust servo motors with higher continuous torque and better overload tolerance for sustained demanding operations.
  - Design and implement a more stable and efficient power distribution system, including proper overcurrent protection and potentially higher-capacity battery solutions for extended mission durations.
- **Develop Vision-Based Interaction:** Utilize the ESP32-CAM beyond live streaming for on-board image processing capabilities, such as:
  - Basic object recognition for identifying specific targets or landmarks.

  - Visual line following or edge detection for structured environments.
  - Real-time visual feedback for human operators beyond simple streaming.

By systematically pursuing these recommendations, the Hexaplorer project can evolve into an even more capable, intelligent, and autonomous robotic platform, expanding its potential for critical applications in hazardous and inaccessible environments.

References

[1] M. I. Uddin, M. S. Alamgir, J. Chakrabarty, M. I. Hossain, and M. A. Abdulla Samy, “Multitasking spider hexapod robot,” in *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, 2019, pp. 135–140.

[2] Y. Nie, D. Zhu, Y. Chen, X. Hu, and L. Wang, “Design of a bionic spider robot with a two-degrees-of-freedom leg structure and body frame,” *Applied Sciences*, vol. 14, no. 21, p. 9809, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/21/9809

[3] G. Verma, P. Rai, B. Chauhan, N. Dutt, and D. Agarwal, “Hardware implementation of autonomous hexapod spider robot,” *International Journal of Information Technology*, vol. 9, no. 4, pp. 395–398, 2017.

[4] G. Jayanth, G. Abinash, and M. Rajmohan, “Multipurpose hexapod robot with integrated iot control,” *International Research Journal of Modern Engineering and Technology Sciences (IRJMETS)*, vol. 05, no. 11, pp. 1969–1972, 2023.

[5] Y. Liu, C. Wang, H. Zhang, and J. Zhao, “Research on the posture control method of hexapod robot for rugged terrain,” *Applied Sciences*, vol. 10, no. 19, p. 6725, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/19/6725

[6] T. Nemoto, R. E. Mohan, and M. Iwase, “Energy-based control for a biologically inspired hexapod robot with rolling locomotion,” *Digital Communications and Networks*, vol. 1, no. 2, pp. 125–133, 2015.

[7] M. S. L. M. Rani and D. U. S. A. Reddy, “Design of 3 dof hexapod leg movement using inverse kinematics,” *Sinergi*, vol. 23, no. 1, pp. 29–34, 2019.

[8] L. Gökrem, M. S. Can, and S. Aydın, “Hexapod robot design and performance comparison of fuzzy and pid control methods,” *Balkan Journal of Electrical and Computer Engineering*, vol. 8, no. 1, pp. 88–97, 2020.

[9] S. Sarker, K. M. Masuk, S. Halder, S. Sarker, R. Bhuiyan, and M. Hossain, “Investigation of gait optimization of a hexapod robot using fuzzy logic controller,” in *AIP Conference Proceedings*, vol. 2596, 2022, p. 020005.

[10] A. A. Nugroho, D. Y. Adji, and D. A. Puspandari, “Mechanical design and gait generation of a hexapod robot using arduino,” *Journal of Robotics and Control (JRC)*, vol. 4, no. 6, pp. 863–870, 2023.

[11] V. B. Jagtap, A. B. Jagdale, and P. S. Kadu, “Hexapod: A spider for terrain and obstacle shunning by arduino,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 573–578.

[12] H. H. Khalili, F. Arvin, and P. Green, “Trends in the control of hexapod robots: A survey,” *Robotics*, vol. 10, no. 3, p. 100, 2021.

[13] C. Sun, Y. Wang, M. Yuan, F. Li, Z. Yang, and X. Ding, “Design and simulation analysis of a hexapod robot based on biomimetic principle,” in *Journal of Physics: Conference Series*, vol. 1168, no. 2. IOP Publishing, 2019, p. 022094.

[14] C. Sun, M. Yuan, F. Li, Z. Yang, and X. Ding, "Design and simulation analysis of hexapod bionic spider robot," in *2019 4th International Conference on Robotics and Vision (ICRV)*, vol. 1168. IOP Publishing, 2019, p. 022094.

Appendix A: Project Disclaimer

This disclaimer clarifies the context and responsibility for the content presented in this graduation project report.

DISCLAIMER

This report was prepared by student(s) of the Computer Engineering Department, Faculty of Engineering, An-Najah National University, as part of their graduation project requirements. While every effort has been made to ensure accuracy and thoroughness, the views, conclusions, and recommendations expressed herein are solely those of the student author(s). An-Najah National University and the Computer Engineering Department accept no responsibility or liability for the consequences of this report being used for any purpose other than that for which it was originally commissioned.

Appendix B: Citation Style

All references in this report adhere strictly to the Institute of Electrical and Electronics Engineers (IEEE) citation style. This standard ensures consistent formatting for in-text citations and the comprehensive reference list, encompassing academic journals, books, technical documents, and open-source software utilized in this project.