A QUADRUPED ROBOT FOR REAL-TIME INSPECTION AND RESPONSIVE OPERATIONS

By

Md. Shahariar Hossain

22221067

Rafin Md Abrar

22321024

Shadid Al Akib

22221050

Tazwar Ahmed

22321055

A Final Year Design Project (FYDP) submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical & Electronic Engineering (BSEEE)

Department of Electrical & Electronic Engineering

BRAC University

May, 2026

A QUADRUPED ROBOT FOR REAL-TIME INSPECTION AND RESPONSIVE OPERATIONS

By

Md. Shahariar Hossain

22221067

Rafin Md Abrar

22321024

Shadid Al Akib

22221050

Tazwar Ahmed

22321055

A Final Year Design Project (FYDP) submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering (BSEEE)

Academic Technical Committee (ATC) Panel Member:

Dr. Touhidur Rahman
Professor, Department of EEE, BRAC University

Shahed Alam
Senior Lecturer, Department of EEE, BRAC University

Atib Mohammad Oni
Lecturer, Department of EEE , Brac University

Department of Electrical & Electronic Engineering

BRAC University

May, 2026

Declaration:

It is hereby declared that

- The Final Year Design Project (FYDP) submitted is our own original work while completing a degree at Brac University.
- The Final Year Design Project (FYDP) does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
- The Final Year Design Project (FYDP) does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
- We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Shahariar Hossain 22221067	**Rafin Md Abrar** 22321024
Shadid Al Akib 22221050	**Tazwar Ahmed** 22321055

Approval

The Final Year Design Project (FYDP) titled "A Quadruped Robot for Real-Time Inspection and Responsive Operations" submitted by

1. Md. Shahariar Hossain (22221067)
2. Rafin Md Abrar (22321024)
3. Shadid Al Akib (22221050)
4. Tazwar Ahmed (22321055)

of Spring, 2026 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 30-04-26.

Examining Committee:

On behalf of the Academic Technical Committee (ATC): (Chair)

____

Shahed Alam
Senior Lecturer, Department of EEE,
BRAC University

Final Year Design Project (FYDP) Coordination Committee: (Chair)

____

Dr. Mirza Rasheduzzaman
Associate Professor, Department of EEE
BRAC University

Department Chair:

____

Dr. Md. Mosaddequr Rahman
Professor, Department of EEE
BRAC University

Ethics Statement

This Final Year Design Project (FYDP), titled “A Quadruped Robot for Real-Time Inspection and Responsive Operations” has been conducted and prepared in accordance with accepted academic, research, and ethical standards set by BRAC University and the Department of Electrical and Electronic Engineering.

All stages of this project, including literature review, system design, data collection, analysis, implementation, testing, and validation, were carried out responsibly, honestly, and transparently. No part of this work involves fabrication, falsification, or misrepresentation of data or results. The findings, designs, prototype details, and test results presented in this report are the original work of the project group, completed under the guidance of the assigned supervisors and ATC Panel.

Proper citation and referencing have been maintained throughout the report to acknowledge all relevant sources, literature, tools used in this study. A plagiarism check was also conducted through Turnitin to ensure academic integrity and transparency. The similarity index of this report was found to be 4.5%, which is well below the acceptable institutional limit.

The authors take full responsibility for the authenticity, integrity, and ethical compliance of this work and certify that it fulfills the requirements for the completion of the Final Year Design Project in partial fulfillment of the degree of Bachelor of Science (B.Sc.) in Electrical and Electronic Engineering.

Abstract

Industrial inspection in hazardous environments such as chemical plants, oil and gas facilities, power systems, and nuclear sites poses significant risks to human workers due to exposure to toxic gases, extreme temperatures, high voltage, and confined spaces. To address these challenges, this project presents the design and implementation of a quadrupedal robot capable of performing real-time inspection and responsive operations with minimal human intervention. The robot integrates a multi-sensor inspection framework for detecting structural and operational abnormalities, with particular focus on identifying surface cracks, microcracks or other early signs of equipment degradation. The system architecture combines a legged mobility platform, onboard computing unit, wireless communication module, and sensor suite including camera, ultrasonic, and IR-based sensing components. These subsystems enable the robot to collect and transmit real-time inspection data to operators for rapid monitoring and decision-making. In addition to routine inspection, the robot is also equipped with a manipulation mechanism for limited emergency operations such as operating switches or mechanical levers when direct human access is not feasible. The quadruped structure provides greater adaptability than conventional wheeled or tracked platforms in stairs, narrow passages, and uneven surfaces, making it more suitable for industrial inspection scenarios. This study addresses key limitations in existing inspection systems, including limited autonomy, infrastructure dependency, and high operational costs, by proposing a modular, cost-effective, and energy-efficient design. Experimental analysis demonstrates that the system can perform reliable inspections while reducing human exposure to hazardous environments. By enabling real-time crack and microcrack detection alongside general hazard inspection, the system supports predictive maintenance, reduces unplanned downtime, and improves worker safety. Overall, this work contributes a modular and practical robotic solution aligned with the growing industrial demand for intelligent, autonomous, and inspection-centric systems.

Keywords: Quadruped inspection robot; Autonomous industrial inspection; Structural crack and microcrack detection; Multi-sensor condition monitoring; Hazardous environment robotics; Remote inspection and monitoring.

Dedication

First and foremost, we express our deepest gratitude to Almighty Allah for granting us the strength, patience, and guidance to successfully complete this project. Without His blessings, this achievement would not have been possible. This project is also dedicated to our families, whose unconditional love, endless sacrifices, and unwavering support have been the foundation of our journey. Their encouragement, patience, and belief in our abilities have inspired us to overcome challenges and strive toward success. We remain profoundly grateful for their constant support in every moment of joy and difficulty. Finally, we dedicate this work to all those who have the courage to dream big and pursue their ambitions with determination, and to the people around them who continue to support and believe in those dreams without doubt. Their strength and encouragement serve as a powerful reminder that no achievement is accomplished alone.

Acknowledgement

We extend our sincere appreciation to our Academic Technical Coordinators (ATCs) for their valuable guidance, suggestions, and continuous feedback throughout this project. We are deeply grateful to Dr. Touhidur Rahman, Professor, Department of Electrical and Electronic Engineering, BRAC University, for his insightful vision, realistic guidance and constructive criticism, which laid a strong foundation for our work. We are especially thankful to Shahed Alam, Senior Lecturer, Department of Electrical and Electronic Engineering, BRAC University, for his continuous motivation, support, and thoughtful guidance that inspired us to remain ambitious and dedicated. We also sincerely thank Atib Mohammad Oni, Lecturer, Department of Electrical and Electronic Engineering, BRAC University, for his timely assistance and guidance in helping us address important requirements and maintain academic completeness. In addition, we would like to express our gratitude to Mr. Nazrul Islam, FYDP Lab LTO, for his kind cooperation and technical assistance whenever needed, and to Mr. Faruk Ahmed, Assistant Engineer of the BRAC University substation, for his guidance and permission during our substation visit. Finally, we thank our friends and project teammates for their collaboration, encouragement, and teamwork, which made this journey both successful and meaningful.

Table of Contents

List of Tables

List of Figures

List of Acronyms

3-DOF: Three Degrees of Freedom
3S2P: Three Series To Parallel
ADC: Analog-to-Digital Converter
API: Application Programming Interface
AQM: Air Quality Monitoring
AWG: American Wire Gauge
BCE: Binary Cross-Entropy
BCR: Benefit-Cost Ratio
BLDC: Brushless Direct Current
CAGR: Compound Annual Growth Rate
CFR: Code of Federal Regulations
CIoU: Complete Intersection over Union
CNC: Computer Numerical Control
CNN: Convolutional Neural Network
CUDA: Compute Unified Device Architecture
ECR: Environmental Conservation Rules
EMC: Electromagnetic Compatibility
EMI: Electromagnetic Interference
EPA: Environmental Protection Agency
FPN: Feature Pyramid Network
FPS: Frames Per Second
FPV: First-Person View
GDP: Gross Domestic Product
GDPR: General Data Protection Regulation
GPIO: General-Purpose Input/Output
GPS: Global Positioning System
GUI: Graphical User Interface
HAZWOPER: Hazardous Waste Operations and Emergency Response
HTTP: Hypertext Transfer Protocol
HVGA: Half Video Graphics Array
IMU: Inertial Measurement Unit
KPI: Key Performance Indicator
LiDAR: Light Detection and Ranging
LiPo: Lithium Polymer
LPG: Liquefied Petroleum Gas
mAP: Mean Average Precision
NPV: Net Present Value
YOLOv8: You Only Look Once Version 8
NRC: Nuclear Regulatory Commission
NSPE: National Society of Professional Engineers
ODE: Open Dynamics Engine
OSHA: Occupational Safety and Health Administration
PAN: Path Aggregation Network
PCB: Printed Circuit Board
PDB: Power Distribution Board
PLA: Polylactic Acid
PM: Particulate Matter
POST: Power-On Self-Test
POV: Point of View
PVC: Polyvinyl Chloride
PWM: Pulse Width Modulation
R&D: Research and Development
RGB: Red, Green, Blue
RISC-V: Reduced Instruction Set Computer - Five
RL: Reinforcement Learning
RMP: Risk Management Program
RoHS: Restriction of Hazardous Substances
ROI: Return on Investment
ROS: Robot Operating System
SCCB: Serial Camera Control Bus
SCL: Serial Clock Line
SDA: Serial Data Line
SDG: Sustainable Development Goal
SLAM: Simultaneous Localization and Mapping
SMD: Surface-Mount Device
SRAM: Static Random-Access Memory
STA: Station Mode
SWOT: Strengths, Weaknesses, Opportunities, and Threats
TCP: Transmission Control Protocol
ToF: Time of Flight
UDP: User Datagram Protocol
URL: Uniform Resource Locator
USB: Universal Serial Bus
VS Code: Visual Studio Code
OBC: On-Board Computer

Chapter 1: Introduction [CO1, CO2, CO10]

1.1 Introduction

Industrial worksites like the chemical plants, oil & gas facilities, power systems, and nuclear sites, all expose human inspectors to toxic gases, extreme heat, high voltage, and confined spaces. In the United States, 1,030 workers died in confined-space incidents between 2011 and 2018, many during routine inspections and maintenance [1]. Globally, the mining sector alone accounts for approximately 12,000 fatalities per year, exposing how dangerous manual inspections remain in high-risk environments [2].

Our project introduces an autonomous quadrupedal robot, designed to perform real-time, routine inspections with minimal human intervention. This robotic platform navigates complex industrial floors, acquires multi-modal data (e.g. visual/thermal imagery, vibration/acoustic signatures, corrosion and leak indicators), and streams results to remote operators for rapid decision-making. Object manipulation (e.g. flipping switches, turning valves) is included as a secondary, emergency capability when human entry is infeasible. The sections below discuss the motivation, survey current solutions, identify unresolved challenges, and show industry relevance and timing.

1.1.1 Problem Statement

In sectors like nuclear energy, oil and gas, and chemical manufacturing, maintenance and safety engineers need a reliable way to inspect facilities remotely because sending human workers into hazardous environments is dangerous, inefficient, and often impractical. Routine inspection tasks currently require personnel to enter confined spaces or active industrial areas where they face toxic gas leaks, high temperatures, high-voltage equipment, and other threats. These manual inspections not only put workers at risk of injury or exposure, but also tend to be labor-intensive and time-consuming, sometimes forcing operational downtime for safety precautions. The challenge is to develop an inspection solution that can operate autonomously in such environments, navigating complex terrain, gathering critical sensor data, and detecting early signs of equipment failure (Crack or Microcrack), without exposing humans to harm. By addressing this need, industries can improve worker safety and regulatory compliance while also reducing inspection costs and unplanned shutdowns due to safer, continuous monitoring.

1.1.2 Background Study

Legged inspection robots are emerging as a practical alternative to humanoid robots. Boston Dynamics' Spot shows robust mobility (stairs, uneven terrain) and modular sensing payloads (RGB, zoom, thermal, gas, LiDAR) with autonomous navigation and optional remote control system. Industrial deployments have used such platforms for regular patrols and data collection in refineries, utilities, and manufacturing [3]. ANYbotics' ANYmal adds 360° perception, autonomous mapping, and self-docking to extend patrol cycles. The company has demonstrated multi-story navigation and routine gauge/valve checks in harsh environments including explosive atmospheres with specialized variants [4]. Beyond legged robots, tracked/wheeled inspectors (e.g., duct, tank, or pipe crawlers) and aerial platforms are widely used for visual, thermal, acoustic, and ultrasonic checks where legged access is unnecessary [4].

Efficiency and safety impacts are increasingly documented in these studies. In a production scenario, a tracked inspection robot surveyed about 853 m of ductwork, reducing labor from 448 human-hours to 24 hours and compressing schedules from 1.5 weeks to 1.5 days, while removing personnel from high-elevation hazards [5]. A cross-sector analysis of construction robots found that automation reduced time workers spent on hazardous tasks by an average of 72%, illustrating direct exposure reduction through robotic substitution [7]. Although legged systems typically run on battery for around 90 minutes per pack (with hot-swap or docking to extend duty cycles), their ability to perform consistent, repeatable rounds supports predictive maintenance and earlier anomaly detection [6], [4].

Sensor assembly that are common to these platforms include: thermal/IR for hotspots; optical imaging for rust, cracks, and gauge reading; acoustic/vibration sensing for bearing/fan faults; LiDAR/SLAM for navigation; and chemical sensors for gas leaks [4], [6]. Depending on different industry specific needs, these sensors enable continuous, structured data capture that is difficult to maintain with human rounds, especially in hazardous areas or off-shift hours.

1.1.3 Literature Gap

Industrial inspection robots have become increasingly important in hazardous sectors such as oil and gas, power generation, mining, chemical processing, and infrastructure maintenance, where human inspection is costly, risky, and often inconsistent. Recent developments show a clear movement from fixed or wheeled monitoring systems toward agile legged robots that can traverse stairs, narrow passages, uneven terrain, and cluttered facilities. Commercial systems such as Boston Dynamics' Spot and ANYbotics' ANYmal demonstrate that quadruped platforms are now viable for industrial sensing tasks through the integration of thermal, visual, gas, LiDAR, and acoustic payloads, as well as autonomous mission execution and, in some cases, docking-based operation [8], [9]. At the same time, the literature also shows that inspection quality increasingly depends on combining mobility with perception, especially computer vision pipelines for structural defect detection such as cracks and corrosion. Deep-learning-based crack detection has advanced rapidly, with object-detection frameworks like YOLO being favored for real-time performance, but practical deployment still faces issues in robustness, false alarms, dataset generalization, and edge-side efficiency [14]. These trends strongly support the motivation of your project: a quadruped robot that combines mobility, real-time fault inspection, and limited responsive manipulation in hazardous environments.

TABLE 1.1: BACKGROUND RESEARCH AND LITERATURE REVIEW

No	Reference	Technology/Focus	Key Insight
[8]	Boston Dynamics	Commercial quadruped inspection robot	Payload-enabled multimodal industrial inspection
[9]	ANYbotics	Autonomous industrial quadruped	Autonomous navigation, docking, and repeatable plant inspection
[10]	Halder et al. (2023)	Quadruped-assisted inspection	Semi-autonomous inspection still requires human-in-the-loop support
[11]	Sánchez-Cubillo et al. (2024)	Automated critical-asset inspection	Full automation depends on reliable perception-localization-planning integration
[12]	Kaveh et al. (2024)	Crack detection survey	Small-defect segmentation remains sensitive to noise and texture complexity
[13]	Wu et al. (2025)	Deep-learning crack detection review	Accuracy gains remain limited by dataset bias and weak domain generalization
[14]	YOLOv8 Docs	Real-time object detection	One-stage detector suitable for edge-deployable defect recognition

Despite progress, the following gaps constrain current practice:

Limited inspection-centric autonomy:
Many existing quadruped robots are effective for patrol and data collection, but they often rely on predefined routes, structured environments, or operator supervision rather than adaptive, fault-driven inspection behavior [8], [10].

Weak integration of inspection and responsive manipulation:
In many current systems, sensing and locomotion are prioritized, while manipulation is either absent or treated as a secondary function. As a result, robots can detect faults but may not be able to respond physically when immediate intervention is needed [9].

Endurance and power management. Battery constraints limit continuous shifts; although docking/hot-swap help, long-duration autonomy with minimal operator touch is still an open engineering and operational challenge [10], [12].

Difficulty in small-crack and microcrack detection under real conditions:
Crack-detection studies show good progress, but real-time field-level microcrack detection

still suffers from false positives, weak generalization, and sensitivity to lighting, texture, and scale variation [12].

Accuracy–reliability tradeoff in real-time vision systems:
Fast detectors such as YOLO are attractive for robotics, but practical deployment can still show reduced localization robustness and unstable confidence under stricter performance criteria [13].

Gap between autonomous design and actual operational performance:
Although the literature promotes autonomous inspection, real-world and simulated systems often still struggle with obstacle handling, environmental variability, and dependable decision-making. This gap is also reflected in experimental findings where manual control can outperform autonomous systems in certain operational scenarios [8], [11].

High cost and deployment complexity:
Advanced quadrupeds with industrial sensor payloads can exceed $75k–$100k, and organizations face skills gaps in deployment/maintenance; this slows scaling beyond early adopters [15].

1.1.4 Relevance to current and future Industry

Our target sectors include nuclear facilities (radiation zones), oil & gas (pipelines, pressure vessels, flare infrastructure), chemical plants (tanks, process lines), electric utilities (switchyards, boilers), cement/mining (dusty and confined structures). Replacing manual human round trips with autonomous patrols reduces exposure, increases inspection frequency, and supports condition-based maintenance.

It should be noted that in oil & gas, unplanned downtime averages about 27 days per year per platform, costing almost US$38M. Even marginal improvements in early-fault discovery have multimillion-dollar impact [17]. Evidence shows robotic inspection can compress schedules dramatically and reallocate human labor to higher-value work [5]. Moreover, the inspection-robotics market is projected to grow rapidly over the next cycle, driven by safety regulation, labor shortages, and Industry 4.0 initiatives integrating robots with IoT data lakes and predictive analytics [15].

Despite decades of standards, hazardous working conditions still account for hundreds of daily worker deaths when acute and chronic causes are combined; shifting inspection from people to robots is an immediate, engineering-driven pathway to risk reduction [16]. So, by emphasizing autonomous navigation and sensing (with manipulation as a standby, emergency function), Our design solutions align squarely with operator needs like: frequent, reliable data; fewer hazardous entries; and faster, evidence-rich decisions. The design’s modular sensors, remote supervision, and docking-enabled duty cycles map directly to the pain points documented in current deployments and literature [15],[18].

1.2 Objectives, Requirements, Specification and constant

1.2.1 Objectives

The primary objective is to design, develop, a novel robot with human-level fine motor control and manipulation in environments too dangerous or inaccessible for human presence, bridging the gap between remote operation and required dexterity. The tentative objectives we aim to fulfill include:

- Perform scheduled, repeatable multisensor patrols, navigating human-designed corridors, stairs, and tight passages, to detect cracks in real time, keeping machines running without unnecessary shutdowns.

- Keep humans safe by conducting inspections remotely in hazardous areas, localizing issues from a distance and minimizing exposure, with escalation only for true emergencies.

- Wirelessly stream high-fidelity inspection data and alerts to operators in real time, enabling immediate decisions and feeding maintenance workflows for rapid follow-up.

1.2.2 Functional and Nonfunctional Requirements

Functional Requirements:

Table 1.2: Table of Functional and Nonfunctional Requirements

Requirement	Description
Mobility	The robot must be able to walk and pass by avoiding objects and move through industrial paths for inspection.
Inspection	Perform routine inspection tasks, detect hazards, and collect data from sensors at different locations.
Communication	Send and receive all kinds of wireless data in real time.
Object Manipulation	The robot must be able to manipulate switches and other mechanical levers for remote operation.

Non-Functional Requirements:

TABLE 1.3: TABLE OF NON-FUNCTIONAL REQUIREMENTS:

Requirement	Description
Reliability	The robot shall achieve at least 95% uptime during operation over its expected lifespan. [15]
Serviceability	The system shall allow for modular replacement of components using standard tools.
Power efficiency	The robot shall optimize motor control to extend battery life.
Environmental Tolerance	The robot shall operate properly in environments with various temperatures and humidity.

1.2.3 Specifications

TABLE 1.4: TABLE OF SPECIFICATIONS

Component	Specification
Mobility speed	Achievable walking speed of 0.5 m/s to 1.5 m/s over uneven terrain task. [14]
Payload Capability	Payloads up to 2- 3kg distributed evenly across its frame.
Power supply	Supports 30min **to** 1 hour of continuous operation per full charge. The recharging duration should be 2 to 4 hours.
Actuators	Actuators in each major elbow and shoulder joints with 100kg/cm torque. [13]
Environmental Tolerance	Operates reliably in ambient temperatures between -10°C to 45°C and withstands relative humidity levels of up to 90% (non-condensing) without malfunction. [14]

1.2.4 Technical and Non-technical consideration and constraint in design process

Technical Considerations in the Design Process:

1. Architectural integration of subsystems

The design required tight integration between locomotion, sensing, manipulation, computation, and power delivery within a single robotic platform. This was necessary to ensure coordinated operation without control conflict, instability, or resource overload.

2. Real-time sensing and control performance

The system had to support low-latency perception, localization, and decision-making for timely obstacle avoidance, hazard detection, and emergency response. This made computational responsiveness a major technical design factor.

3. Kinematic and actuation feasibility

The robot needed sufficient actuator torque, joint control, and kinematic stability to support both quadrupedal locomotion and arm-based responsive tasks. This required careful consideration of gait mechanics, inverse kinematics, and load-bearing capability.

4. Power distribution and energy management

The robot required a stable and efficient power architecture to support simultaneous operation of actuators, sensors, onboard computation, and wireless communication. Variations in load demand during locomotion and manipulation can affect voltage stability, thermal behavior, and overall endurance. Therefore, battery sizing, voltage regulation, and power allocation were important technical considerations in ensuring uninterrupted and safe operation.

5. Sensor fusion and perception reliability

The inspection task depends on the robot's ability to interpret heterogeneous sensory inputs, including visual and proximity data, in a consistent manner. A technical consideration was therefore the effective fusion of these sensing modalities to improve environmental awareness, reduce false detections, and support decision-making under uncertainty.

6. Fault tolerance and safety logic

Since the robot is intended for hazardous environments, the design had to account for sensor failure, communication loss, and control instability. Safe-stop behavior, fallback control, and fault monitoring were therefore important to ensure predictable and safe operation under abnormal conditions.

7. Human–machine control transition

Where manual and autonomous control coexist, transition between modes becomes a technical issue. The system must prevent command conflict, unstable switching, and unsafe override behavior, making control hierarchy and mode management important design considerations.

Non-Technical Considerations in the Design Process:

1. Industrial usability

The system needed to remain practical for operators, with interpretable feedback, manageable control, and suitability for real inspection workflows. In an industrial context, technical capability alone is insufficient if the system cannot be operated with confidence by end users. Accordingly, the design process considered the clarity of operator interaction, the interpretability of system outputs, and the extent to which the robot could support routine inspection activities without increasing operational complexity.

2. Deployment feasibility

The design had to consider use in legacy facilities where infrastructure, connectivity, and automation readiness may be limited. Since many target environments are not built for advanced robotic systems, successful deployment depends on the ability of the design to function under infrastructural and organizational limitations. This consideration therefore relates to the practical integration of the robot into existing facilities without requiring major changes to plant layout, digital infrastructure, or workforce structure.

3. Lifecycle viability

Adoption depends not only on performance, but also on maintainability, replacement simplicity, and long-term economic value. From an industrial perspective, the sustainability of a robotic system is closely tied to the ease with which it can be maintained, repaired, and economically justified over time. For that reason, lifecycle viability was considered as a non-technical issue influencing not only implementation but also the long-term acceptance and scalability of the design.

4. Regulatory and compliance constraints

The design is constrained by safety, data protection, and sector-specific regulatory requirements. Compliance with standards such as industrial robot safety, hazardous-environment operation, and secure handling of monitoring data must be considered from the design stage, since non-compliance would limit deployment feasibility.

5. Workforce acceptance and organizational readiness

Adoption may be constrained by operator resistance, fear of job displacement, and limited familiarity with robotic inspection systems. As a result, the design must support human oversight, clear operational roles, and practical integration into existing industrial workflows.

6. Economic and implementation pressure

Even if the system is technically viable, deployment is constrained by the need to justify cost through reduced downtime, safer inspection, and maintenance savings. This means the design must demonstrate practical value to industries before large-scale adoption becomes realistic.

Constraints in the Design Process:

1. Budgetary Constraint

The total project budget is limited to BDT 180,000–200,000, which restricts the selection of components, onboard computing capability, and sensing complexity [15].

2. Weight and Payload Constraint

The robot must remain under 15 kg while carrying approximately 2–3 kg of sensors and supporting modules. This limits material choice, structural design, and actuator loading [17].

3. Physical Dimensional Constraint

The system must be compact enough to pass through narrow industrial corridors and confined spaces. Therefore, chassis size and limb workspace are restricted by navigational requirements.

4. Environmental Tolerance Constraint

The robot must operate reliably under harsh conditions, including −10°C to +45°C, high humidity, dust, moisture, and corrosive environments. This constrains enclosure design and component selection.

5. Operational Endurance Constraint

The platform must support 30–60 minutes of continuous operation per charge, with a recharge time of 2–4 hours. This places limits on battery capacity, power management, and sensor usage [16].

6. Mobility and Speed Constraint

The robot must achieve 0.5–1.5 m/s walking speed on uneven terrain while maintaining stability. This constrains gait design, actuator torque, and motion control [18].

7. Indoor/Outdoor Versatility Constraint

The design must support operation in both indoor and outdoor environments, including exposure to sunlight, rain, and varying surface conditions. This affects sealing, traction, and communication reliability.

8. Communication and Connectivity Constraint

The robot must transmit control signals, sensor data, and alerts in real time over a wireless link. Since many target environments are GPS-denied, the system is constrained to rely on robust onboard localization and safe operation under communication loss.

1.2.5 Applicable Compliance, Standards, and Codes

TABLE 1.5: TABLE OF APPLICABLE COMPLIANCE, STANDARDS, AND CODES

Domain	Standard/Code	Purpose	Relevance to the Project
Electrical Safety	IEC 60204-1 (Safety of Machinery – Electrical Equipment) [19]	Defines requirements for safe design and wiring of electrical systems in machinery.	Ensures the robot's electrical components are safe, insulated, and protected from hazardous contact.
Robotics Safety	ISO 10218-1 / ISO 10218-2 (Robots and Robotic Devices – Safety Requirements) [20]	Specifies safety measures for the design and integration of industrial robots.	Guides the safe design and operation of the quadruped manipulator in hazardous environments.
Autonomous System Safety	ISO 13482 (Personal Care Robots Safety)	Establishes safety requirement for robots interacting with humans	Adapted for ensuring safe teleoperation and semi-autonomous functions in dangerous zones.
Environmental Protection	IEC 60529 (IP Codes) [21]	Classifies levels of protection against dust and water ingress.	Ensures the robot can operate in dusty, wet, or contaminated conditions without failure.
Radiation Safety	IAEA Safety Standards / NRC Guidelines	Sets safety measures for radiation exposure control and shielding.	Ensures the robot's components are protected and operators remain safe during radiation handling tasks.
Hazardous Materials Handling	OSHA HAZWOPER (29 CFR 1910.120) [22]	Provides regulations for safe work in biohazard and chemical environments.	Guides operational protocols for biohazard and chemical hazard zones.
Wireless Communication	IEEE 802.11 / ITU-R Standards [23]	Defines technical specifications for reliable wireless data transfer.	Ensures uninterrupted communication between the operator and the robot in mission environments.

Developing Custom Criteria for Ill-Defined Problems:

For our project, since quadruped robots are still developing, some areas may not have clear standards. In these cases, we can develop project-specific criteria while ensuring compliance with good engineering practice. This process includes:

- **Problem Analysis and Risk Assessment:** Break down the problem into smaller functional and operational components, identifying all potential risks to safety, performance, and environmental impact.

- **Benchmarking and Best Practices:** Review similar projects, even in unrelated industries, to identify proven design strategies and operational safety measures that can be adapted.

- **Stakeholder and Expert Consultation:** Engage with domain experts, regulatory bodies, and potential users to ensure that the developed criteria meet operational needs and safety expectations.

- **Iterative Design and Testing**: Refine the criteria through successive design–test–feedback cycles, incorporating improvements based on test results and user feedback.

- **Documentation for Traceability:** Maintain detailed records of the decision-making process, justifying each criterion to ensure transparency and replicability for future projects.

By following this systematic approach, the resulting criteria remain consistent with the principles of good engineering practice, even in the absence of formal standards, while ensuring safety, reliability, and performance.

1.3 Systematic Overview of the Proposed Project

The proposed project focuses on the design and development of a quadruped robot for real-time inspection and responsive operations in hazardous industrial environments. The main purpose of the system is to reduce human exposure to dangerous conditions such as toxic gas leakage, extreme heat, confined spaces, high-voltage zones, and structurally unsafe locations. The robot is intended to support inspection activities in sectors such as oil and gas, chemical plants, electric utilities, mining, and nuclear facilities, where manual inspection is risky, time-consuming, and often inefficient.

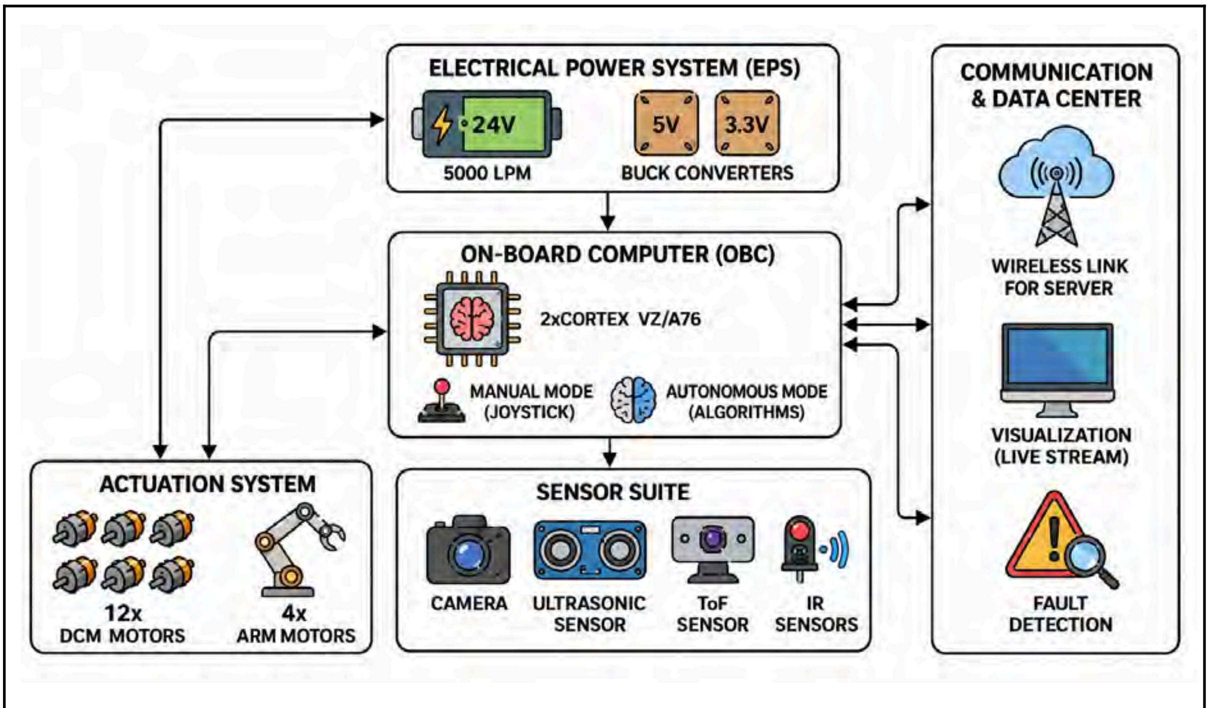


Figure 1.1: General System Architecture

Overall System Concept

The project is designed as an integrated robotic inspection platform that combines locomotion, sensing, actuation, wireless communication, and intelligent decision-making. The full system is divided into several connected subsystems: the Electrical Power System, On-Board Computer, Actuation System, Sensor Suite, and Communication and Data Center. Together, these units enable the robot to move through industrial environments, collect inspection data, detect faults, and report information to remote operators in real time.

Locomotion and Mobility Design

The robot uses a quadruped body structure to achieve stable movement over uneven terrain and narrow industrial pathways. Its body is constructed around a rectangular chassis with four legs placed symmetrically around the torso. Each leg has three degrees of freedom, allowing controlled motion in a vertical plane. The movement of the legs is governed through inverse kinematics and gait-planning logic, which enable the robot to perform walking, turning, and obstacle-aware navigation. Trotting and crawl-based motion strategies are considered to balance speed, stability, and terrain adaptability.

Power and Control Architecture

The robot is powered by a 3S2P 12.6V LiPo battery, while buck converters regulate the voltage to meet the needs of the onboard electronics. The On-Board Computer acts as the control center of the robot. In manual mode, it interprets commands from the operator and coordinates motion accordingly. In autonomous mode, it processes sensor data and executes navigation algorithms for path planning, obstacle avoidance, and system-level decision-making. This architecture ensures centralized and synchronized control of all robotic functions.

Sensing and Inspection Capability

The sensing subsystem consists of a camera, ultrasonic sensor, IR sensors, and related detection components. These sensors allow the robot to perceive its surroundings, avoid obstacles, and gather inspection-related data. The project specifically aims to detect visual and structural anomalies such as cracks and other fault indicators. By using multi-sensor input, the robot can support condition monitoring in industrial sites and generate real-time environmental awareness for both autonomous navigation and operator supervision.

Manual and Autonomous Operating Modes

A major feature of the project is its dual control approach. In the manual control system, the robot is operated remotely by a human using a joystick or keyboard, while live sensor and camera data are streamed to a display interface. In the autonomous control system, the robot independently interprets its sensor inputs and makes movement decisions without requiring continuous operator commands. The autonomous mode uses navigation logic based on sensor feedback, heading correction, and obstacle detection. This two-approach design allows the system to be evaluated from both usability and performance perspectives.

Robotic Arm and Responsive Operation

In addition to mobility, the proposed robot includes a 3-degree-of-freedom robotic arm for limited manipulation tasks. The arm is mounted on the robot and consists of a base rotation joint, shoulder joint, and elbow joint. Its operation is based on kinematic control and can be used in both manual and autonomous modes. This subsystem is designed for responsive operations such as pressing switches, interacting with simple control elements, or extending toward detected objects in emergency or inaccessible scenarios.

Fault Detection and Intelligent Vision Module

The project integrates a fault detection subsystem based on the YOLOv8 deep learning framework. This module processes live camera feeds to detect visible defects, particularly microcracks and related surface faults. The detection system includes initialization, real-time inference, frame annotation, confidence thresholding, and output display. It also incorporates an active learning mechanism, where user feedback can be used to refine the detection model over time. This makes the project not only a robotics system, but also an AI-assisted inspection solution.

Communication and Monitoring Framework

The robot transmits data wirelessly to a central monitoring interface. Through this communication framework, live video, environmental readings, robot status, and fault alerts can be observed remotely. This supports remote supervision, system safety, and immediate response in hazardous scenarios. The communication layer therefore plays an essential role in connecting the physical robot to the human decision-maker.

Overall, the proposed project is a multidisciplinary inspection robot that combines mechanical design, embedded systems, intelligent control, computer vision, and industrial safety considerations into one platform. Its significance lies in offering a safer and more efficient alternative to conventional inspection methods. By reducing human exposure, improving inspection frequency, and enabling remote monitoring, the system addresses an important real-world engineering need with strong industrial relevance.

1.4 Conclusion

Chapter one introduced the proposed quadruped robot for real-time inspection and responsive operations in hazardous industrial environments. It explained the problem of unsafe and inefficient manual inspection, reviewed the relevant background and literature, identified major research and practical gaps, and highlighted the project's relevance to current and future industry. The chapter also defined the project objectives, requirements, specifications, design considerations, constraints, and applicable standards. Finally, it provided a systematic overview of the proposed system, showing how locomotion, sensing, manipulation, communication, and intelligent fault detection are integrated into a single inspection platform. Overall, this chapter establishes the need, direction, and significance of the project, forming the basis for the detailed design and implementation discussed in the following chapters.

Chapter 2: Project Design Approach [CO5, CO6]

2.1 Introduction

Chapter 2 presents a comparative study of the two principal design approaches considered for the proposed quadruped industrial inspection robot: manual control and autonomous control. Both approaches are based on the same core hardware architecture, including the power system, onboard controller, actuation unit, sensor suite, and communication framework, but they differ in control logic, level of human involvement, and operational behavior. The purpose of this comparison is to evaluate their respective strengths, limitations, and practical suitability for hazardous inspection tasks, and thereby justify the most appropriate design choice for the current stage of the project.

2.2 Identify Multiple Design Approach

The quadruped robot proposes two distinct control architectures that address the challenge of performing inspections in hazardous industrial environments. These architectures: manual control and autonomous control correspond to different levels of human involvement, system complexity and operating autonomy.

2.2.1 Design 1. Manual Control Approach

A human operator directs the robot's movement and inspection tasks via a joystick. The hardware subsystems are arranged around an Electrical Power System (EPS), an on-board computer (OBC), actuation (eight servo motors for legged locomotion and a 3-DOF robotic arm) and a sensor suite. The OBC houses the microcontroller that interprets joystick commands and coordinates motor control, while environmental data from cameras, infrared and ultrasonic sensors are relayed to a central server for real-time feedback. The operator will remotely control the robot from a safe distance via Wi-Fi or other wireless communication protocols and will be able to stream the live view from the camera to see the feedback. The operator will have a joystick and keyboard to control the arm and the body walking movements.

2.2.2 Design 2. Autonomous Control Approach

The robot navigates without continuous human supervision. It uses the same EPS, actuation and sensing hardware as the manual system, but the OBC executes navigation algorithms that interpret sensor data based on pre-planned paths, avoid obstacles and make decisions. The camera, ultrasonic, and temperature sensors feed data to the autonomous algorithm for navigation and fault detection. A wireless data center monitors system health and provides a communication channel for remote supervision and fault handling.

2.3 Describe multiple design approach

2.3.1 Design Approach 1: Manual Control System

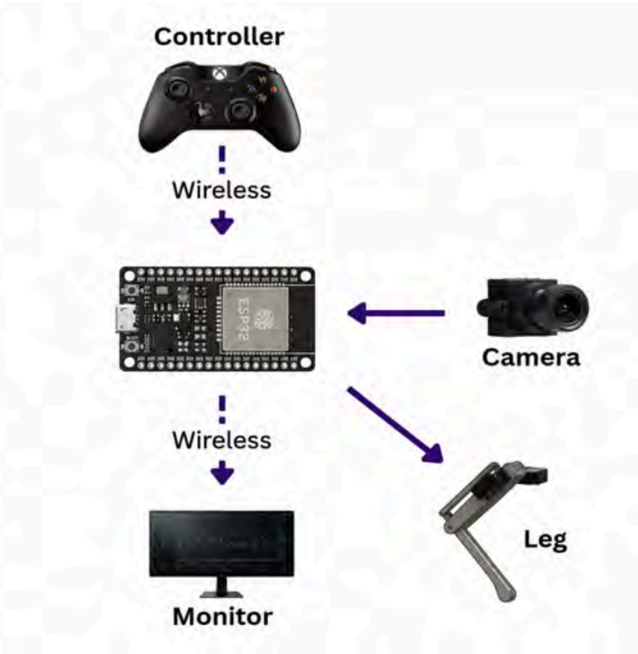


Figure 2.1**:** Manual Control System Overview

Methodology:

The manual control system is composed of three major subsystems. Firstly, the Electrical Power System (EPS) powers the entire circuitry and all peripherals. The primary power source is a 3S2P 12.6V LiPo battery. Separate batteries will be utilized to provide 5V and 3.3V to other components, ensuring operation remains within safe voltage margins. Secondly, the On-Board Computer (OBC), which houses the microcontroller, serves as the central brain. The servo control system incorporates a total of eight servo motors, providing fast mobility and a high degree of freedom. Additionally, a robotic arm will be included for object manipulation. Finally, these components are integrated with various sensors, such as a camera and an ultrasonic sensor. All acquired data will be relayed to a central server and then displayed on a visual system (i.e., a computer) to provide the operator with live surroundings and environmental feedback. The robot's movement and detailed environment assessment will be controlled by an operator using a joystick.

System Operation:

- The EPS delivers regulated power to all subsystems.
- The OBC processes operator commands and coordinates servo motors and the robotic arm.

- The sensors continuously capture environmental data, which is relayed to the server and displayed visually.
- The operator uses a joystick to control robot movement and assess surroundings in real time, ensuring precise navigation and effective task execution.

2.3.2 Design Approach 2: Autonomous Control System

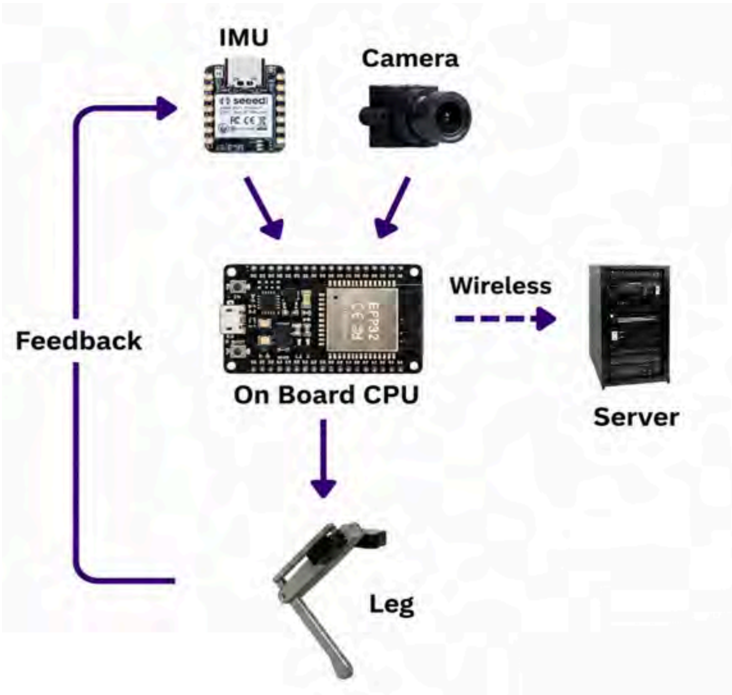


Figure 2.2**:** Autonomous Control System Overview

Methodology:

This system is fully autonomous and does not require a human operator to control its movement. The peripheral components and sensors are similar to the manual design approach and the only key difference is in the control system which is done autonomously. The system will utilize its camera sensor and other sensors such as ultrasonic, temperature and gas sensors to know about its environment and utilize that data to feed the autonomous navigation algorithm and take decisions accordingly for fault detection. The wireless data center is also present here to always have the room for checking the status of the robot's current situation and take necessary steps if any fault is detected and also it will act as a middle man to feed current information about navigation and fault systems to the autonomous algorithm.

System Operation:

- The EPS powers all subsystems with regulated voltage.

- The OBC runs autonomous navigation algorithms, interpreting sensor data to plan paths and avoid obstacles.
- The sensor suite provides continuous environmental input, enabling the robot to adapt to dynamic surroundings.
- The wireless data center monitors system health, ensuring fault detection and allowing corrective measures if required.
- Together, these subsystems enable the robot to move manipulate objects, and assess its environment without human control, while still offering remote monitoring for safety and reliability.

2.4 Analysis of multiple design approach

TABLE 2.1: TABLE OF ANALYSIS OF MULTIPLE DESIGN APPROACHES

Criteria	Manual Control System	Autonomous Control System
Power Consumption	Lower power usage as processing demand is minimal and mainly command-driven.	Higher power consumption due to continuous sensing, computation, and decision-making processes.
Data Transmission Speed	Requires continuous real-time data streaming for operator feedback and control [24].	Less dependent on constant high-rate transmission for control; mainly required for monitoring and reporting.
Ease of Setup	Easier to implement with straightforward control architecture and minimal algorithmic complexity.	More complex setup involving navigation algorithms, sensor fusion, and decision-making logic [25].
Coverage	Coverage depends on operator control and may vary between sessions.	Enables systematic and consistent coverage through programmed navigation and path planning.
Reliability	Dependent on operator skill, attention, and consistency during operation.	Dependent on sensor accuracy, algorithm robustness, and system stability.
Data Storage & Visualization	Provides real-time visualization for operator-based decision-making.	Supports remote monitoring, logging, and automated data handling through onboard and communication systems [25].

Scalability	Limited scalability due to continuous requirement of human supervision.	More scalable for large-area and repetitive inspections with reduced human involvement.
Latency	Includes human reaction delay in the control loop [26].	Lower response delay due to onboard real-time decision-making.
Security	Relies on human judgment, reducing algorithmic risks but prone to human error.	Reduces human exposure but depends on secure and reliable autonomous system behavior.
Suitability for Future Applications	Limited adaptability for future intelligent systems and automation trends.	Highly suitable for future development, including AI integration, smart inspection, and Industry 4.0 applications [27].

2.5 Conclusion

This chapter examined two alternative control approaches for the proposed quadruped industrial inspection robot: manual control and autonomous control. The comparison showed that both approaches are built on the same core hardware architecture, but they differ significantly in control strategy, operational dependency, system complexity, and future applicability. The manual control system offers a simpler and more direct solution, with lower processing demand, easier setup, and strong operator-guided decision-making. In contrast, the autonomous control system provides greater consistency, scalability, lower response delay, and stronger potential for future smart industrial applications through reduced human involvement and algorithm-based operation. Overall, the comparative analysis demonstrates that each approach has distinct advantages and limitations, and the selection between them depends on the required balance among practicality, reliability, coverage, latency, and long-term automation goals.

Chapter 3: Use of Modern Engineering and IT Tool. [CO9]

3.1 Introduction

The selection of engineering and IT tools for this project was guided by the technical demands of each development phase and the overall objective of building a reliable quadruped robot for real-time inspection and responsive operation. Since the project involves mechanical simulation, control development, embedded implementation, wireless monitoring, and intelligent fault detection, the tools were chosen to create an efficient and connected workflow from design and simulation to implementation and evaluation. Particular importance was given to ensuring that each tool could support the project's requirements in robotics, sensing, locomotion control, computer vision, and real-time system analysis.

3.2 Select Appropriate Engineering and IT tools

The selected tools were evaluated not only on general usability, but also on how effectively they addressed the specific needs of the project. For example, simulation tools needed to support quadruped locomotion, joint kinematics, and environment interaction; development tools needed to manage multi-module codebases; and vision tools needed to provide low-latency detection performance suitable for safety-critical inspection tasks. Compatibility between tools was also an important consideration so that simulation results, control logic, embedded deployment, and detection modules could be integrated into a unified development process.

Criteria for Tool Selection:

Technical relevance and fidelity: Each tool had to accurately represent or support the real system components and engineering processes involved in the project, such as quadruped motion, robotic arm control, sensor integration, and fault detection.

Simulation capability: Since the project required evaluation of multiple design approaches before physical implementation, the selected simulation tool had to support realistic modeling of walking gait, inverse kinematics, robotic arm motion, and obstacle-based environment testing.

Analytical and visualization support: Tools had to provide meaningful visualization, performance analysis, and feedback generation for validating system behavior, comparing design approaches, and interpreting detection results.

Embedded and hardware integration: The selected tools needed to support eventual transfer from simulation and software design to physical robotic implementation, including compatibility with embedded systems and microcontroller-based control.

Real-time performance: Because the robot is intended for inspection in hazardous environments, the tools had to support low-latency sensing, control, and detection, especially for navigation and crack/fault identification.

Industry relevance and standard practice: Preference was given to tools that are widely recognized in robotics, embedded systems, simulation, and computer vision, so that the project remains aligned with modern engineering practice.

Accessibility and learning curve: Since the project was developed within academic and budgetary constraints, tools with strong documentation, community support, cross-platform usability, and moderate learning complexity were preferred.

Cost-effectiveness: Tools with free, academic, open-source, or low-cost access were prioritized to ensure that the project remained feasible within available resources.

Based on the criteria a certain tools are selected:

- Webots for realistic robot simulation, including quadruped locomotion, gait analysis, and environment-based testing.
- Python for implementing control logic, inverse kinematics, autonomous navigation algorithms, and system integration.
- YOLOv8 (Ultralytics) for real-time object detection and microcrack/fault identification from camera input.
- VS Code for integrated development, debugging, and management of multi-module code (control, vision, and communication).
- Spyder IDE for reinforcement learning development and detailed numerical analysis of training processes.
- OpenCV for image processing, frame handling, and integration with the vision-based detection pipeline.
- PyTorch for deep learning model execution, training, and optimization of the fault detection system.

3.3 Use of Modern Engineering and IT Tools

Webots	
Webots is used as the main simulation platform because of its cross-platform stability and the powerful Supervisor API. The node Supervisor will be very important in our RL workflow since this will enable us, in the training script, to programmatically reset the robot's position and randomize the environment-for example, moving obstacles-without needing to restart the entire simulation. This greatly increased the speed with which data could be collected for the walking gait algorithm	

Alternative Considered: Gazebo is a common alternative in the robotics industry. However, we chose Webots because Gazebo has a higher learning curve and is often strictly coupled with specific Linux/ROS versions. Webots had better cross-platform stability and an easier-to-use GUI to rapidly prototype and allow focus on gait logic, not environment configuration.

Spyder IDE	
Spyder was utilized exclusively for the Reinforcement Learning phase. RL algorithms rely heavily on large multi-dimensional arrays (State-Action matrices). Spyder's unique "Variable Explorer" allowed us to inspect these matrices in real-time to debug mathematical errors in the reward function, a feature that provides deep insight into the learning process.	

Alternative Considered: Jupyter Notebooks are popular for data science and ML models. However, we chose Spyder because cell-based execution is exceptionally good with RL (Reinforcement Learning) loops where we can train it for hours of episodes(segments of epochs). Spyder provides a more cohesive debugging environment for continuous scripts, mimicking the workflow of MATLAB which is standard in our curriculum thus making Spyder familiar.

VSCode	
VS Code was adopted for the System Integration and Computer Vision phases. As the codebase grew to include multiple modules (vision, locomotion, communications), VS Code's robust project management and linting tools ensured code quality. Additionally, its integrated terminal and support for virtual environments made managing the YOLO dependencies (PyTorch, OpenCV) seamless.	

Alternative Considered: PyCharm is a robust Python IDE often used in industry. We selected VS Code because it is lighter weight and has a superior extension ecosystem for mixed-language development (Markdown, Python, C++), not to mention PlatformIO where we can even program our microcontrollers. This flexibility was necessary as we moved between writing the project report and coding the robot controllers.

YOLOv8	
We implemented the YOLOv8 model to process the robot's camera feed in real-time. This tool allows us to identify cracks with a single forward pass of the neural network. This single-stage architecture ensures the low latency required to trigger the "Emergency Stop" function immediately upon detecting a hazard, which will protect our industrial equipment	

Alternative Considered: Faster R-CNN offers arguably higher accuracy but utilizes a two-stage detection process, resulting in lower frame rates, that doesn't align with our inspection criteria. We chose YOLO V8 because industrial safety requires immediate reaction times (milliseconds matter), also being able to be deployed on edge platforms.

Table 3.1: Modern IT Use and Selection Criteria

Criteria	Webots (Robot Simulation)	YOLO Model (Fault Detection)
Ease of Use	Requires node hierarchy knowledge; Intuitive GUI; physics parameters are all easily adjustable for robust simulation	Extensive library support; Modern implementations (v5/v8) have simplified APIs, perfect for embedded systems
Flexibility	Multi-language & plugin support; Supports Python, C++, AI model deployment and custom physics plugins.	Efficient transfer learning; Easily retrainable on custom industrial fault datasets.
HW Integration	Native cross-compilation features; Direct code transfer to physical robots by using C++ code to Arduino IDE or using CircuitPy and micro-python	Edge device acceleration; Optimized for Edge devices (Jetson, Coral, Neural Sticks).
Visualization & Analysis	Integrated real-time 3D feedback; Built-in OpenGL rendering and sensor visualization (Lidar/Camera).	Clear visual overlays & metrics; Bounding boxes, confidence heatmaps, and confusion matrices.
Industry Standard	Academic & R&D benchmark; Widely used in automotive R&D and academic robotics.	Real-time vision benchmark; Standard for real-time object detection.
Reason for Selection	Selected for its robust physics engine (ODE), which is essential for simulating complex walking algorithms and robotic arm kinematics without hardware risks.	Selected for Emergency Stop functionality because its single-stage architecture offers the low latency required to detect risks (e.g., fire, cracks) instantly.

3.4 Conclusion

In this chapter, a structured approach was followed to select appropriate modern engineering and IT tools for the development of the proposed quadruped robotic inspection system. The selection process considered multiple technical and practical criteria, including simulation capability, real-time performance, analytical support, hardware compatibility, and industry relevance. The chosen tools collectively enabled efficient system design, accurate simulation, effective algorithm development, and reliable fault detection, ensuring that each stage of the project, from concept to implementation, was properly supported. Overall, the tool selection was driven not merely by convenience, but by how effectively each tool supported the design, validation, analysis, and future deployment of the quadruped robotic inspection system. This approach ensured that the selected toolchain not only met the current project requirements but also provided a strong foundation for future enhancements and scalability.

Chapter 4: Optimization of Multiple Design and Finding the Optimal Solution. [CO7]

4.1 Introduction

Chapter 4 presents the optimization of the proposed design approaches through systematic performance evaluation under controlled simulation conditions. The primary objective is to assess how effectively each approach performs in terms of locomotion, manipulation, and overall system behavior, and to identify performance trade-offs between manual and autonomous control strategies. The evaluation is conducted separately for the body locomotion and arm manipulation subsystems to ensure a detailed and component-level understanding of system performance.

Key Performance Parameters Considered:

- **Completion Time:** Total time required to complete a given task or reach the target position.
- **Number of Steps:** Total motion steps taken by the robot to complete navigation tasks.
- **Positional Deviation:** Difference between the final robot position and the target position, indicating navigation accuracy.
- **Success Rate:** Percentage of successful task completions, particularly for manipulation tasks.
- **Execution Efficiency:** Overall performance in terms of time, steps, and control smoothness.
- **System Responsiveness:** Ability of the system to react to control inputs or environmental changes in real time.

4.2 Optimization of multiple design approach

This section focuses on the systematic evaluation and optimization of the multiple design approaches proposed for the quadruped robotic inspection system. First, here we describe the system and describe the optimization process we used. Then we analyze the performance of each approach under controlled simulation conditions and identify their strengths and limitations based on key operational metrics. By examining parameters such as execution time, number of steps, positional accuracy, and task success rate, the study aims to refine the design choices and establish a data-driven foundation for selecting the most efficient and reliable approach. The evaluation is carried out separately for the body locomotion and arm manipulation subsystems to ensure a comprehensive assessment of overall system performance.

4.2.1 General Overview of The Design:

4.2.1.1. Body of The Robot

The robot is a headless, quadrupedal platform designed with a symmetric, rectangular chassis. The torso is modeled as a three-dimensional rectangular box with dimensions 0.35 × 0.16 × 0.065 meters (Length × Width × Height), and it serves as the root node of the robot's kinematic chain.

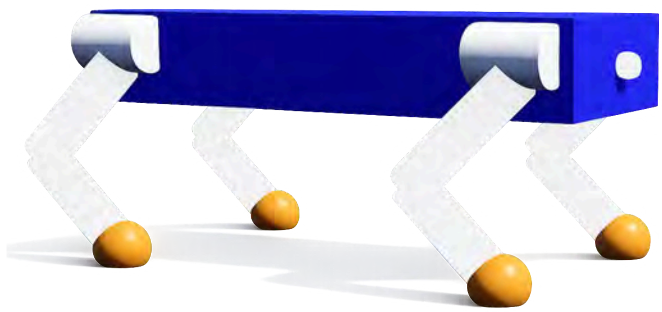


Figure 4.1: Representation of the body in Simulation.

Four limbs are attached to the torso at the shoulder coordinates (±0.14, ±0.10, −0.008) meters, defined relative to the center of the body frame. Each limb has two degrees of freedom (3-DOF) constrained to motion within a single vertical plane, resembling the leg structure of a digitigrade mammal.

The joint hierarchy for each leg is defined as follows:

- Shoulder Joint: A rotational motor rigidly attached to the torso.
- Upper Arm: A cylindrical segment with a length of 0.1 m and a radius of 0.015 m.
- Elbow Joint: A hinge joint connecting the upper arm to the forearm.
- Forearm: A cylindrical segment with a length of 0.1 m.
- End Effector: A spherical foot (Ball node) with a radius of 0.02 m, designed to minimize friction during ground contact.

The legs are configured in a bent-backward orientation. Left and right legs are mirror reflections across the sagittal plane, while front and rear legs are translated along the longitudinal axis of the torso.

Body Controller Code: The control architecture is implemented in Python, utilizing the Webots Robot API. The controller follows a state-machine structure, initializing all sensors (GPS, Compass, Distance Sensors, and Camera) and actuators at startup. The system supports three primary operational states:

- Calibration (rest) mode
- Manual teleoperation mode
- Autonomous navigation mode

A global control loop executes at a basic time step of 16 ms, during which sensor data acquisition, trajectory interpolation, and motor command dispatching are performed concurrently. Here we employed Inverse Kinematics Algorithm, which is explained in Chapter 5 of this report.

Walking Gaits and Cycle: The robot employs a trotting gait, providing a balance between locomotion speed and static stability. In this gait, diagonally opposite leg pairs (Front-Left with Back-Right, and Front-Right with Back-Left) move in synchrony while the opposing pair operates in an inverse phase. This guarantees at least two ground contact points at all times. Gait generation is based on linear interpolation between predefined Cartesian keyframes over fixed time segments. Each gait cycle consists of four distinct phases defined in the leg's local coordinate frame:

- Lift (1.4, 1.0): Leg lifts off the ground
- Plant (1.0, 0.0): Ground contact and load transfer
- Retract (1.4, −1.0): Backward push for propulsion
- Recover (1.4, 0.0): Return to neutral position

The controller smoothly interpolates the foot position between these points. Directional control is achieved by reversing the phase order for backward motion or inverting phase direction for lateral leg pairs to produce tank-style turning.

4.2.1.2 Robotic Arm

The robotic arm is a serial manipulator with a 3-degree-of-freedom (3-DOF) kinematic chain, mounted on a static cylindrical base. The base has a radius of 0.04 m and a height of 0.02m. The joint hierarchy proceeds as follows:

1. **Turret Joint (Base):** A vertical hinge joint (Z-axis) allowing 360-degree rotation of the entire upper assembly.
2. **Shoulder Joint:** A horizontal hinge joint (Y-axis) connecting the turret to the upper arm.
3. **Upper Arm:** A cylinder with a length of 0.1m and a radius of 0.01m.
4. **Elbow Joint:** A horizontal hinge joint (Y-axis) connecting the upper arm and forearm.
5. **Forearm:** A cylinder with a length of 0.1m.
6. **End Effector:** A laser distance sensor mounted at the tip of the forearm, used for object detection.

The arm operates in a spherical coordinate system where the base controls the azimuth angle, while the shoulder and elbow joints operate in a vertical plane to control elevation and radial distance. The base rotation is decoupled from this calculation and is controlled independently as a separate rotational axis.

Figure 4.2: Representation of the arm.

Arm Controller Code: The control system is implemented in Python using the Webots Robot API. Upon initialization, the controller enables the rotational motors (base_rm, hinge_rm, elbow_rm), their corresponding position sensors, and the distance sensor. The system operates via a finite state machine with three distinct modes: OFF (Calibration), AUTO (Autonomous Scanning), and MANUAL (Teleoperation).

4.2.2 Optimizing Design Approach 1: Manual Control

4.2.2.1 Body of The Robot

In Manual Mode, the operator directly controls the robot using keyboard inputs that dynamically modify the gait trajectory.

- The S key toggles between the calibration (rest) pose and active walking.
- **Forward / Backward Motion**: Controlled using arrow keys to drive synchronized gait cycles.
- **Turning:** The left and right arrow keys initiate tank turns, where the legs on one side move forward while the opposite side moves backward.
- **Visual Feedback**: An onboard camera provides a first-person view (FPV), allowing the operator to observe the environment from the robot's perspective.

4.2.2.2 Robotic Arm

Activated by pressing the B key, this mode allows the operator to control the manipulator using the keyboard:

- **Base Rotation:** Left & Right arrow keys actuate the base motor velocity at ± 1.2 rad/s.
- **Vertical Extension:** The Up and Down arrow keys increment or decrement the target Y coordinate (constrained between 1.0 and 1.8 in the normalized model space). The IK solver runs continuously to translate this coordinate change into shoulder and elbow joint angles.

4.2.3 Optimizing Design Approach 2: Autonomous Control

4.2.3.1 Body of The Robot

The autonomous behavior is governed by a Manhattan Navigation Algorithm, using onboard GPS and Compass sensors to determine position and heading.

- **Path Planning:** The controller computes the position error (Δx, Δy) relative to the target and moves strictly along the dominant axis, producing motion in discrete 90-degree segments.
- **Steering Logic:** If the heading error exceeds 20 degrees, the robot halts forward motion and executes a spot turn until alignment is achieved.
- **Obstacle Avoidance:** A front-facing distance sensor (ds1) continuously monitors the path. If an obstacle is detected within 0.06 m, the robot halts or reverses.

4.2.3.2 Robotic Arm

Activated by pressing the A key, the robot executes a "Search and Extend" algorithm:

- **Scanning State:** The base rotates continuously at 0.5 rad/s while monitoring the distance sensor.
- **Detection:** If the sensor detects an object within the threshold distance (< 500 units), the base rotation halts immediately to lock the target azimuth.

Extending State: The controller automatically increments the Y coordinate in the IK solver, extending the arm vertically/radially toward the target until the maximum reach is achieved.

4.2.3.3 Fault Detection System:

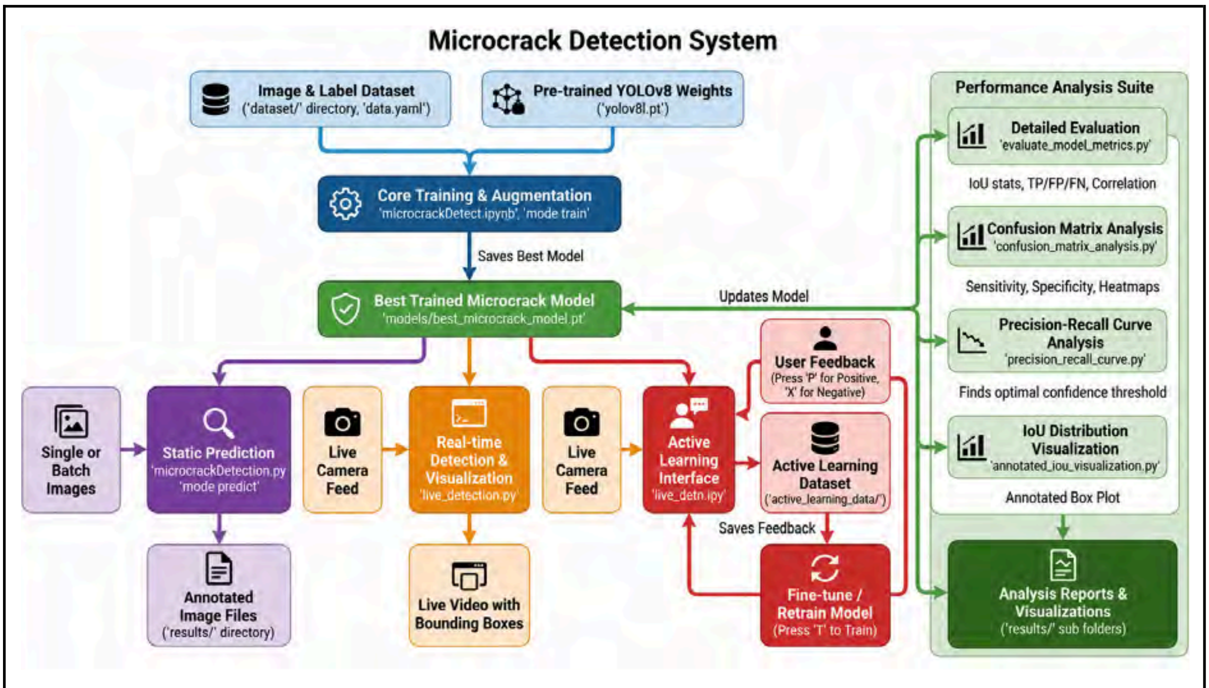


Figure 4.3: General Block Diagram of the detection process.

Object Detection: The object detection model has been done by utilizing the yolov8 model. The object detection subsystem detects objects. Since this is a pretrained model from ultralytics we are not providing any encapsulated data analysis here. But we are presenting the system architecture block diagram for object detection.

Microcrack Detection System: The quadruped bot will have a dedicated microcrack detection system. This system will enable the robot to detect microcracks using its camera. The whole system has been developed in python using the yolo model from ultralytics. Now below is the full microcrack detection architecture.

Training Process: The Microcrack Detection System Training Process using YOLOv8 is a robust, interactive pipeline designed to detect microcracks in real time while continuously improving model accuracy through user feedback. It begins with a dataset of raw images and corresponding labels, structured via a data.yaml file, alongside pre-trained YOLOv8 weights. These inputs feed into the training script, which applies advanced data augmentation techniques, such as mosaic, mixup, HSV adjustments, and rotation, to enhance generalization. Hyperparameters are optimized, and the model is trained on either CUDA or CPU, resulting in a saved best-performing model. This model is then deployed for live detection using live_detrl.py, which processes camera input and allows users to label detections as positive (P) or negative (N). These labeled samples are stored in an active learning dataset, enabling on-the-fly model refinement. When the user presses T, the system triggers a fine-tuning process for five epochs using the curated feedback data, updating the model while backing up the previous version. This closed-loop approach ensures the system evolves with real-world input, making it highly adaptive and accurate for microcrack detection tasks.

Dataset Source: https://www.kaggle.com/datasets/arunrk7/surface-crack-detectionBelow is the single iteration loss report for one iteration only:

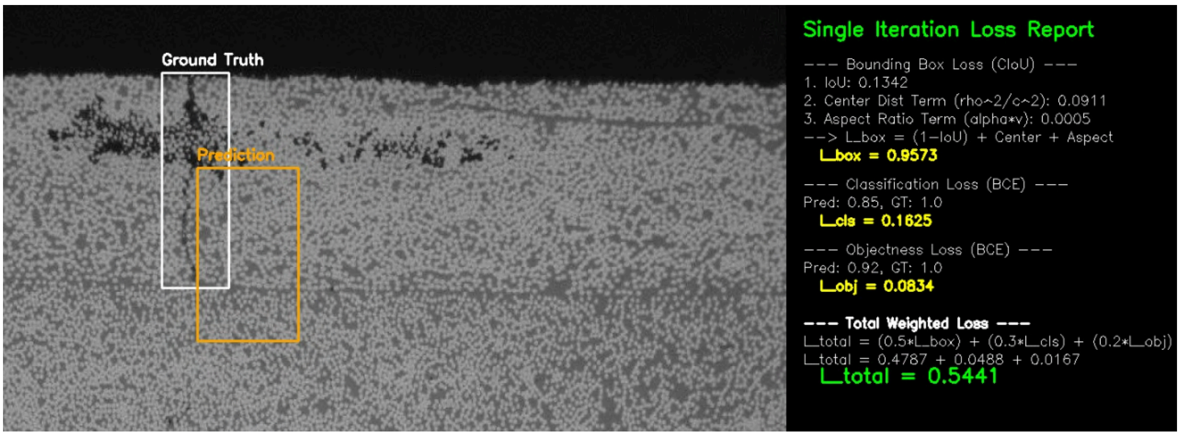


Figure 4.4: Single Iteration Loss Report For One Iteration

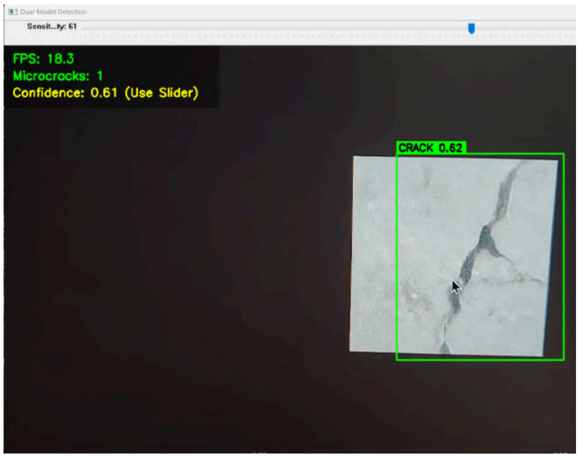


Figure 4.5: Real-Time Crack Detection Example

The above two images are using the mobile camera and a crack image on the desktop screen to detect the crack. As you can see that with 60% confidence we are able to detect the crack very proficiently. Rooms for hyperparameter tuning are still ongoing and by the end of the FYDP-C the whole system will have almost 95% accuracy.

4.3 Identify optimal design approach

4.3.1 Multiple Test Case Evaluation

We have performed a total of 16 test simulations on four arenas for both design approaches. The target positions for the four arenas are chosen randomly and run upon these. For comparison, the average values of "Execution Time", " Steps" and "Deviation" have been taken.

4.3.1.1 Body Locomotion Test Cases:

This study evaluates the navigational efficacy of a quadruped robot designed for hazardous industrial inspections. Simulations were conducted in Webots, comparing autonomous path-following against manual teleoperation across four distinct arenas containing randomized obstacles. The experimental framework consisted of 32 total trials:

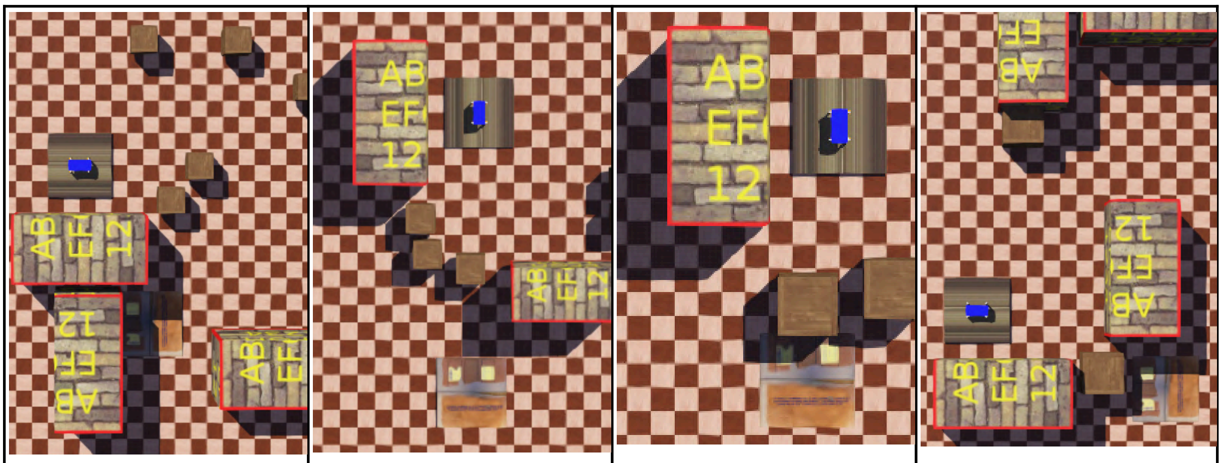
Autonomous Mode: The robot followed predefined trajectories; four trials were conducted per arena to ensure consistency.

Manual Mode: Four separate operators controlled the robot via a joystick/keyboard using a first-person visual feed. Each operator completed one trial per arena.

Performance was quantified using completion time, decision steps, and positional deviation from the goal. While the autonomous system achieved higher positional accuracy, the manual approach proved superior in overall efficiency and operational simplicity. Consequently, manual control is the recommended approach for the current system design.

Birds Eye View of Arina 2	Camera POV for Manual Mode

Figure 4.6: Example images from the test.



Arena 1	Arena 2	Arena 3	Arena 4

Figure 4.7: Example images for the arenas

Table for Body Test Cases:

TABLE 4.1: TABLE OF TEST RESULTS OF THE BODY

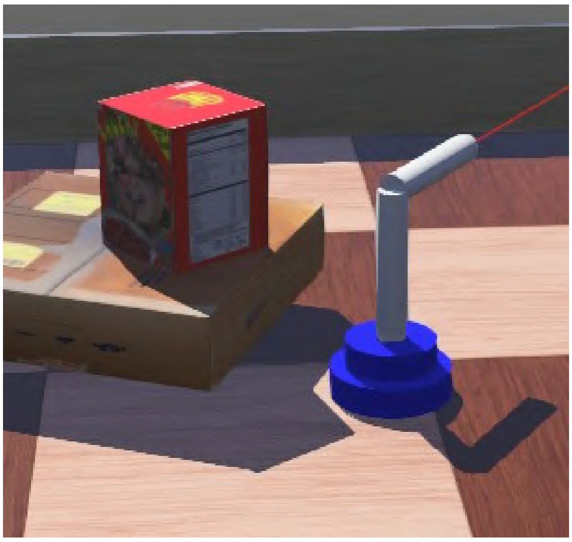
Mode	Arena	Trial / Operator	Time (s)	Steps	Bot Position (X, Y)	Goal Position (X, Y)	Deviation %
Auto	1	1	39.34	12	(0.81, -2.03)	(1.11, -2.46)	19.40%
		2	41.77	12	(0.81, -2.70)	(1.11, -2.46)	14.20%
		3	40.43	12	(0.81, -1.99)	(1.11, -2.46)	20.70%
		4	36.57	12	(0.71, -1.81)	(1.11, -2.46)	28.30%
	2	1	49.00	12	(3.800, 0.490)	(4.00, -0.11)	15.81%
		2	50.23	13	(3.705, 0.495)	(4.00, -0.11)	16.82%
		3	47.58	11	(3.910, 0.477)	(4.00, -0.11)	14.83%
		4	49.74	12	(3.770, 0.499)	(4.00, -0.11)	16.27%
	3	1	20.00	7	(2.400, 0.050)	(2.67, -0.34)	47.43%
		2	20.34	8	(2.441, 0.056)	(2.67, -0.34)	44.97%
		3	19.46	6	(2.368, 0.047)	(2.67, -0.34)	49.08%
		4	20.57	7	(2.417, 0.053)	(2.67, -0.34)	45.78%
	4	1	34.00	10	(3.200, -1.060)	(2.83, -1.20)	39.56%
		2	34.61	11	(3.216, -1.071)	(2.83, -1.20)	39.95%
		3	33.15	9	(3.172, -1.034)	(2.83, -1.20)	38.07%
		4	34.90	11	(3.238, -1.082)	(2.83, -1.20)	40.87%
Manual	1	Akib	31.00	9	(1.49, -2.18)	(1.11, -2.46)	47.20*
		Tazwar	32.00	11	(1.414, -2.41)	(1.11, -2.46)	30.41*
		Rafin	27.00	10	(1.46, -2.30)	(1.11, -2.46)	38.48*
		Shahariar	43.00	9	(1.09, -2.42)	(1.11, -2.46)	4.47*
	2	Akib	45.20	11	(3.73, 0.13)	(4.00, -0.11)	36.12*
		Tazwar	40.97	12	(3.59, 0.23)	(4.00, -0.11)	53.26*
		Rafin	36.20	9	(3.92, 0.32)	(4.00, -0.11)	43.73*
		Shahariar	44.30	10	(3.86, 0.36)	(4.00, -0.11)	49.04*
	3	Akib	34.60	9	(2.60, -0.04)	(2.67, -0.34)	38.64*
		Tazwar	24.70	9	(2.68, 0.056)	(2.67, -0.34)	39.61*
		Rafin	26.40	9	(2.70, 0.07)	(2.67, -0.34)	41.05*
		Shahariar	37.23	11	(2.67, 0.03)	(2.67, -0.34)	37.00*
	4	Akib	43.60	12	(3.20, -0.99)	(2.83, -1.20)	42.54*
		Tazwar	34.90	9	(3.07, -0.93)	(2.83, -1.20)	36.12*
		Rafin	28.26	11	(2.77, -1.12)	(2.83, -1.20)	10.10*
		Shahariar	33.10	8	(2.74, -0.86)	(2.83, -1.20)	35.17*

Here, Deviation %,

$$e = sqrt((X1 - X2)^2 + (Y1 - Y2)^2)$$

3.1.1.2 Arm Manipulation Test Cases:

The manipulation subsystem was evaluated through a "search-and-press" task within a randomized Webots simulation environment. For each of the 10 total trials, a target object was placed at unpredictable coordinates within the robot's workspace to test adaptability. We compared the performance of the Autonomous Mode against Manual Teleoperation (keyboard-driven via a first-person POV feed) using two primary metrics: Completion Time, representing the duration from initiation to contact, and Success Rate, a binary measure of whether the arm accurately pressed the target. Results demonstrated that while the autonomous system was functional, manual control achieved a superior 100% success rate with a faster average response time of 8.6 seconds.



Searching object in autonomous mode

Press initialized

Figure 4.8: Simulation of the manipulation task.

Table for Arm Test Cases:

TABLE 4.2: TABLE OF TEST RESULTS OF THE ARM

	Auto		Manual	
Trial No	Completion Time (s)	Success or not (1/0)	Completion Time (s)	Success or not (1/0)
1	15	1	8	1
2	9	0	9	1
3	5	1	10	1
4	10	1	6	1
5	12	1	10	1

Multiple Test Case Evaluation for the Simulation of Arm

TABLE 4.3: TABLE OF AUTONOMOUS AND MANUAL MODE EVALUATION

System	Parameters	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Comparison
Autonomous System	Completion Time (s)	15	9	5	10	12	Average Time = 10.2 sec
	Success or not (1/0)	1	0	1	1	1	Success Rate (Total Success/ Total trial) % = 80%
Manual System	Completion Time (s)	8	9	10	6	10	Average Time = 8.6 sec
	Success or not (1/0)	1	1	1	1	1	Success Rate (Total Success/ Total trial) % = 100%

4.3.2 Design Approach 1: Manual Control Test Results:

We will sum the values of all four trials for each parameter across all arenas and divide by the total number of trials (which is four) to calculate the average of each parameter for all the arenas.

TABLE 4.4: TABLE OF MANUAL MODE TRIAL RESULTS

Arena	Parameter	Trial 1	Trial 2	Trial 3	Trial 4	Sum	Average
Arena 1	Completion Time (s)	31.00	32.00	27.00	43.00	133.00	33.25
	Steps	9	11	10	9	39	9.75
	Deviation (%)	17.50	11.40	14.30	1.70	44.90	11.23
Arena 2	Completion Time (s)	45.20	40.97	36.20	44.30	166.67	41.67
	Steps	11	12	9	10	42	10.50
	Deviation (%)	9.00	13.30	10.90	12.30	45.52	11.38
Arena 3	Completion Time (s)	34.60	24.70	26.40	37.23	122.93	30.73
	Steps	9	9	9	11	38	9.50
	Deviation (%)	11.40	14.70	15.20	13.70	55.00	13.75

Arena 4	Completion Time (s)	43.60	34.90	28.26	33.10	139.86	34.97
	Steps	12	9	11	8	40	10
	Deviation (%)	14.00	11.90	3.30	11.60	40.80	10.20

Now, we will calculate the average of the four arenas for each of the three parameters to obtain the final average for the manual part, which can then be compared to the autonomous part.

TABLE 4.5: TABLE OF MANUAL MODE AVERAGE TRIAL RESULTS

Parameter	Arena 1	Arena 2	Arena 3	Arena 4	Sum	Average
Completion Time (s)	33.25	41.67	30.73	34.97	140.62	32.21
Steps	9.75	10.50	9.50	10	39.75	9.94
Deviation (%)	11.23	11.38	1.75	10.20	46.56	10.03

4.3.3 Design Approach 2: Autonomous Control Test Results:

We will sum the values of all four trials for each parameter across all arenas and divide by the total number of trials (which is four) to calculate the average of each parameter for all the arenas.

TABLE 4.6: AUTONOMOUS TRIAL RESULTS

Arena	Parameters	Trial 1	Trial 2	Trial 3	Trial 4	Sum	Average
Arena 1	Completion Time (s)	39.34	41.77	40.43	36.57	158.11	39.53
	Steps	12	12	12	12	48	12
	Deviation (%)	5.40	4.20	4.70	3.30	14.60	3.65
Arena 2	Completion Time (s)	49.00	50.23	47.58	49.74	196.55	49.14
	Steps	12	13	11	12	48	12
	Deviation (%)	5.81	6.82	4.83	6.27	22.73	5.93
	Completion Time (s)	20.00	20.34	19.46	20.57	80.37	20.09

Arena 3	Steps	7	8	6	7	28	7
	Deviation (%)	4.60	4.00	5.20	4.30	18.10	4.53
Arena 4	Completion Time (s)	34.00	34.61	33.15	34.90	136.66	34.17
	Steps	10	11	9	11	41	10.25
	Deviation (%)	3.00	3.40	2.40	4.00	13.80	3.30

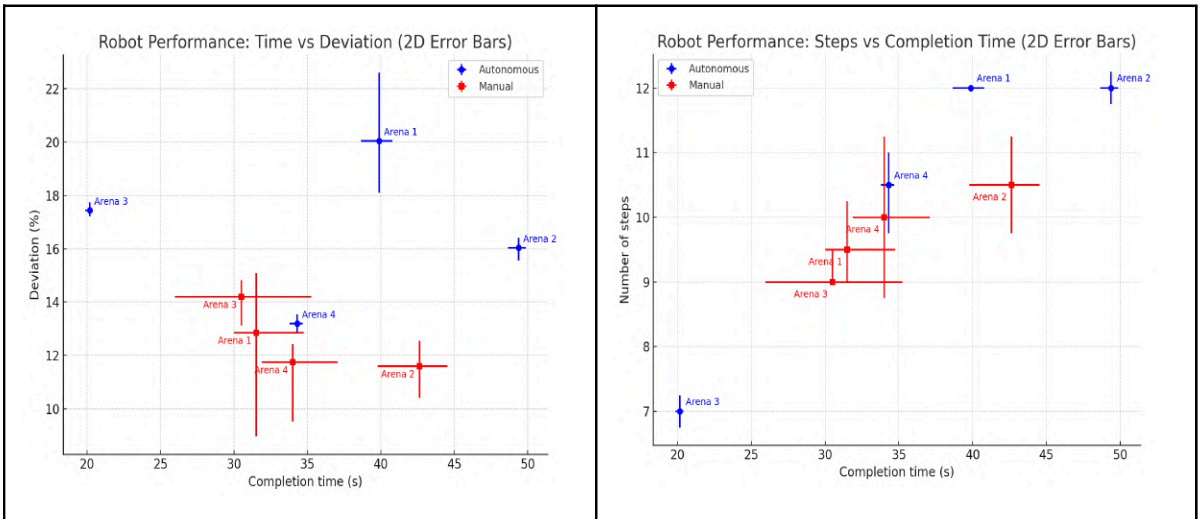
Now, we will calculate the average of the four arenas for each of the three parameters to obtain the final average for the autonomous part, which can then be compared to the manual part.

TABLE 4.7: TABLE OF AUTONOMOUS TRIAL AVERAGE RESULTS

Parameter	Arena 1	Arena 2	Arena 3	Arena 4	Sum	Average
Completion Time (s)	39.53	49.14	20.09	34.17	142.93	39.53
Steps	12	12	7	10.25	41.25	10.31
Deviation (%)	3.65	5.93	4.53	3.30	67.31	4.64

4.4 Performance evaluation of developed solution

Error Bar Plot of The Body Tests



Time vs Deviation Error Bar Plot for both Manual and Autonomous Approaches	Step vs Completion Time Error Bar Plot for both Manual and Autonomous approaches

Figure 4.9: Error Bar Plots of The Test Results

The two plots compare the performance of manual and autonomous robot control systems across different arenas. The left plot shows that the autonomous system (blue) has higher accuracy with lower deviation, but takes longer to complete tasks compared to the manual system (red), which is faster but less consistent. The right plot indicates that the manual approach requires fewer steps and completes tasks more quickly, while the autonomous system takes more steps and has a slightly longer completion time. These plots illustrate the trade-off between efficiency and accuracy, with manual control excelling in speed and the autonomous system offering better precision. Since the manual approach requires less time and fewer steps, the red bar is closer to the origin, indicating that the manual-controlled system can be considered better than the autonomous system.

Error Bar Plot of The Arm Tests:

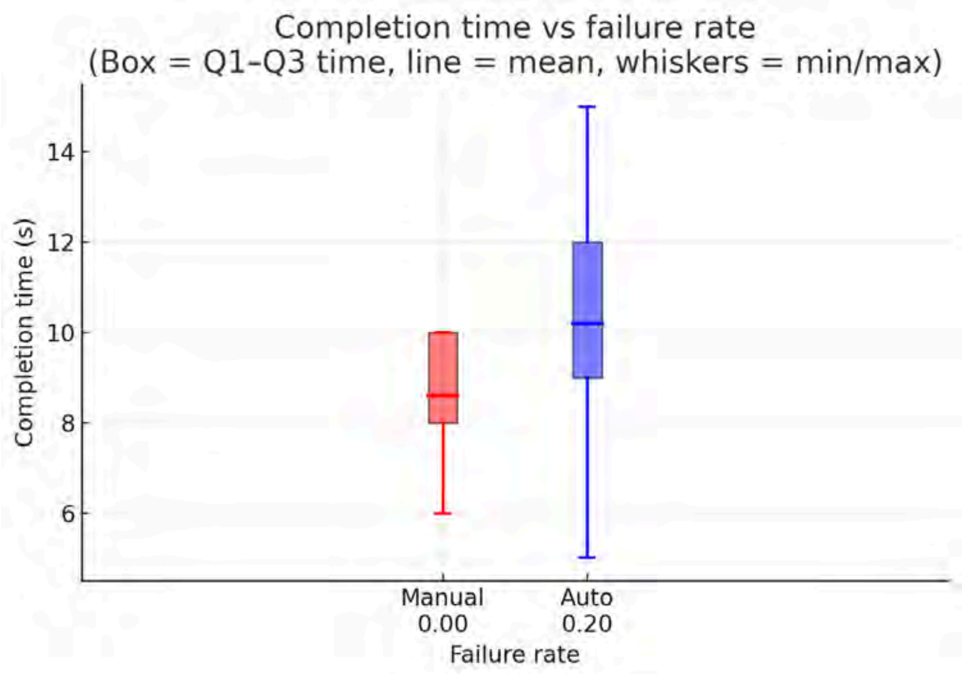


Figure 4.10: Box Plot to compare manual vs autonomous for Arm Simulation

This box plot compares the completion time against the failure rate for manual and autonomous control systems. The "Manual" system (in red) shows faster completion times with a 0% failure rate, whereas the "Auto" system (in blue) has slightly slower completion times with a 20% failure rate. The plot highlights that the manual approach is more efficient in terms of time, but the autonomous system tends to experience more variability due to failure rates.

Autonomous Crack Detection Performance Metrics:

```
============================================================
EVALUATION SUMMARY
============================================================
mAP@50       : 0.9435
mAP@50-95    : 0.5269
Precision    : 0.8838
Recall       : 0.7778
------------------------------------------------------------
Total images evaluated : 6
Total TP: 9
Total FP: 251
Total FN: 0
Mean IoU: 0.7667
Median IoU: 0.8147
IoU 25/75: 0.7410 / 0.8457
Confidence-IoU Correlation: 0.38525034525207513
Per-detection CSV: C:\Users\tazwa\OneDrive\Documents\FYDP\Microcrack Detection\results\eval_run_20251222_203028\detection_summary.csv
IoU boxplot: C:\Users\tazwa\OneDrive\Documents\FYDP\Microcrack Detection\results\eval_run_20251222_203028\iou_boxplot.png
============================================================
```

Figure 4.11: Performance Metrics

This evaluation summary provides a detailed snapshot of the microcrack detection model's performance across multiple dimensions. The mean Average Precision (mAP) at an IoU threshold of 0.5 is exceptionally high at 0.9435, indicating that the model is highly effective at detecting objects with moderate overlap. However, the mAP@50-95, which averages performance across stricter IoU thresholds, drops to 0.5269, revealing that the model's localization precision diminishes under tighter evaluation criteria. The precision of 0.8838 shows that most predicted detections are correct, while the recall of 0.7778 indicates that the model successfully identifies a large portion of actual microcracks. Despite this, the model produced 251 false positives and 0 false negatives across 6 evaluated images, meaning it detected all true objects but also flagged many incorrect ones, suggesting over-sensitivity. The mean IoU of 0.7667 and median IoU of 0.8147 reflect strong bounding box alignment with ground truth, and the IoU 25/75 range (0.7410 / 0.8457) confirms consistent localization quality. Lastly, the confidence-IoU correlation of 0.3825 implies a weak relationship between the model's confidence scores and actual localization accuracy, indicating that high-confidence predictions aren't always the most precise. Overall, the model is highly capable but requires tuning to reduce false alarms and improve reliability under stricter conditions.

Following multiple tests on the body and arm mechanisms, the performance of the automatic and manual systems was evaluated to identify the most suitable developed solution for further hardware implementation. Both simulated and non-simulated parameters were considered in the evaluation. The following sections present the simulated and non-simulated analyses used to justify and support the final selection decision.

4.4.1 Findings on Simulated Data:

TABLE 4.8: TABLE OF FINAL AUTO VS MANUAL TEST VERDICT

Subsystems	Paramet	Approach 1 (Manual Control)	Approach 2 (Autonomous Control)	Better Approach
Main Body	Accuracy	89.97%	95.36%	Approach 2
	Execution Time	32.21 seconds	39.53 seconds	Approach 1
	Steps	9.94 steps	10.31 steps	Approach 1
Arm	Execution Time	8.6 seconds	10.2 seconds	Approach 1
	Success Rate	100%	80%	Approach 1

4.4.2 Findings on Non-Simulated Data

TABLE 4.9: TABLE OF AUTO VS MANUAL QUALITATIVE ANALYSIS VERDICT

Criteria	Approach 1 (Manual Control)	Approach 2 (Autonomous Control)	Better Approach
Human Involvement	Required continuously	Not required	Approach 2
Control Type	Manual joystick	Algorithmic-based	Approach 2
Engineering Complexity	Low	High	Approach 1
Cost	143660 BDT	174620 BDT	Approach 1
Efficiency	Slower for repetitive tasks	Faster and more consistent	Approach 2
Error Handling	Prone to human error	Can use feedback to correct	Approach 2

Based on the tables, Approach 1 (Manual Control) outperforms Approach 2 (Autonomous Control) in terms of Accuracy, Execution Time for both the main body and arm, and Success Rate. Despite Approach 2 being more efficient and requiring no human involvement, Approach 1 shows better results in key parameters such as Accuracy and Success Rate. Thus, we can say, Approach 1 (Manual Control) is better.

4.5 Conclusion

This chapter presented the optimization of multiple design approaches through performance evaluation of the quadruped robotic system. Both manual and autonomous control strategies were analyzed using key parameters such as completion time, number of steps, positional deviation, and success rate. The results showed that the autonomous approach provides better accuracy, while the manual approach performs better in execution time, efficiency, and reliability. Based on the analysis, the manual control system is identified as the optimal design for the current stage due to its faster response, higher success rate, and lower complexity. Overall, the findings provide a clear, data-driven basis for design selection and highlight potential areas for future improvement, particularly in enhancing autonomy.

Chapter 5: Completion of Final Design and Validation. [CO8]

5.1 Introduction

The development of the robotic dog followed a step-by-step methodology, starting with design and gait calculation. In this stage, the robot structure, body dimensions, leg arrangement, and walking mechanism were planned using CAD software, while the basic gait motion was calculated to support stable locomotion.

After finalising the design, the required parts were produced through 3D printing and fabrication. The printed components were then assembled with the mechanical frame, servos, sensors, camera, robotic arm, claw, and other electronic components. This stage involved both mechanical and electronic system integration to ensure that all parts worked together as a complete robotic platform.

Next, the machine learning model was trained for perception tasks such as crack/fault detection. The robot was also calibrated for both manual and automatic control using a joystick/gamepad, allowing the user to control walking, arm movement, and claw operation. Finally, the complete system was tested and validated through different performance tests, including walking speed, turning ability, obstacle detection, fault detection, battery performance, payload capacity, and responsive emergency operation.

5.2 Completion of final design

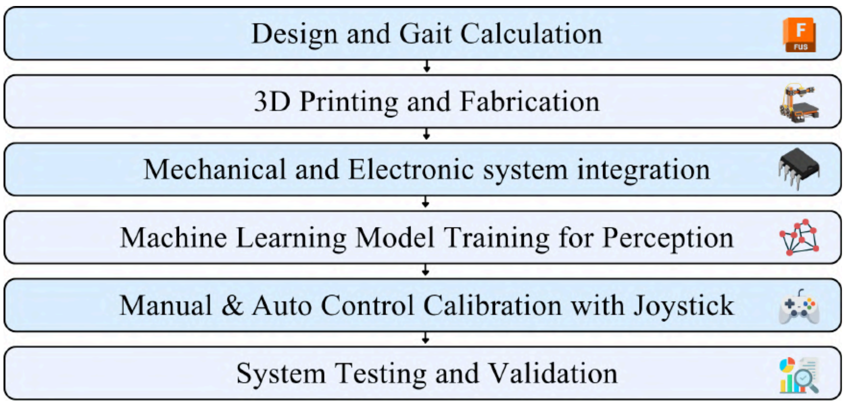


Figure 5.1: Steps of Project Development

For our quadrupedal robot, we completed the final design process using a step-by-step and sequential approach. The process followed is described below.

Working Process:

1. First, we designed the theoretical model based on our simulation work from the FYDP-D semester. We developed the initial model, torque requirement analysis, system requirements, and overall design requirements.

2. Next, we confirmed the required components based on the design requirements. We selected the necessary parts, ordered components online, and purchased some parts from local stores. Then, we designed the mechanical parts in Fusion 360 according to the required dimensions and selected components, such as servo motors, tubes, and other structural parts. After finalizing the CAD models, we 3D printed the parts.
3. After that, we assembled the mechanical structure, integrated the components, built the circuit, and placed the electronics inside the robot body.
4. Then, we implemented the control logic and inverse kinematics system that we developed during the FYDP-D design semester. We also tested the inverse kinematics through simulation.
5. Next, we developed and trained the machine learning model. Specifically, we trained the YOLO model according to our project requirements.
6. Finally, we calibrated the robot using the joystick and tested it in different scenarios. We evaluated whether the robot performed properly and met the expected functional and non-functional requirements of the project.

5.2.1 Theoretical Torque Calculation

To determine the required component ratings for building the robot structure, we performed a torque analysis based on the estimated size, geometry, and mass distribution of the quadruped robot. The total robot mass was estimated to be 7.3 kg, including the body, legs, and mounted robotic arm. Each leg was modeled with two 20 cm segments, giving a total leg length of approximately 40 cm, with each full leg weighing about 0.60 kg. The robot body was approximately 60 cm long and 40 cm wide. Using these assumptions, along with the robotic arm dimensions and payload requirements, we calculated the torque demands for the leg joints and arm joints under different operating conditions, including trotting, crawling, and turning. This torque analysis was used to select suitable servo motors with sufficient torque ratings and safety margins for reliable operation.

Part A. Leg Torque Calculations

Section I. Assumptions and Variable Definitions

The overall torque relies on the mass distribution and geometric properties of the robot:

- M: Total robot mass, including body, legs, and mounted arm ($M = 7.3\ kg$).
- g: Gravitational acceleration ($g = 9.80665\ m/s^2$).
- L_1, L_2: Lengths of the upper (hip-to-knee) and lower (knee-to-foot) leg segments ($L_1 = L_2 = 0.20\ m$).
- m_1, m_2: Masses of the upper and lower leg segments ($m_1 = m_2 = 0.30\ kg$). Total leg mass $m_{leg} = m_1 + m_2 = 0.60\ kg$.
- T: Interpolation segment time for gait motion ($T = 0.192\ s$).
- SF: Safety factor applied to the total torque ($SF = 1.2$).
- Unit Conversion: $1\ kg \cdot cm = 0.0980665\ N \cdot m$

Section II. General Equations

1. Support Force (F)

The force supported by one stance leg ($F = mg$) depends on the gait configuration:

- **Trot forward (2 legs grounded):** $F_{trot} = Mg/2 = 35.79\ N$
- **Crawl forward / Turn (3 legs grounded):** $F_{crawl} = F_{turn} = Mg/3 = 23.86\ N$

2. Gravity Torque (τ_g)

Gravity torque is the sum of the products of acting forces and their perpendicular moment arms (r):

- **Hip (Lateral):** $\tau_{g,hip} = Fz_f + m_1 g z_{c1} + m_2 g z_{c2}$ (where z denotes lateral moment arms).
- **Shoulder (Pitch):** $\tau_{g,shoulder} = Fr_f + m_1 g r_{c1} + m_2 g r_{c2}$ (where r denotes radial/horizontal moment arms).
- **Knee (Elbow):** $\tau_{g,knee} = Fr_{f/k} + m_2 g r_{c2/k}$ (moment arms measured relative to the knee).

3. Inertia Torque (τ_I)

Inertia torque is given by $\tau_I = I\alpha$, utilizing the moment of inertia for a rod rotating about one end ($I = \frac{1}{3}mL^2$) and an angular acceleration estimate ($\alpha = 2\Delta\theta/T^2$).

- **Hip/Shoulder (Combined $0.40\ m$ effective rod):**

  $I_{hip} = I_{shoulder} = \frac{1}{3}(0.60)(0.40)^2 = 0.032\ kg \cdot m^2$
- **Knee (Lower $0.20\ m$ segment only):** $I_{knee} = \frac{1}{3}(0.30)(0.20)^2 = 0.004\ kg \cdot m^2$
- **Total Torque:** $\tau_{total} = \tau_g + \tau_I$ (before safety factor) and $\tau_{final} = SF(\tau_g + \tau_I)$.

Section III. Trot Forward Calculation ($F_{trot} = 35.79\ N$)

Hip / Lateral Servo

Forward walking yields zero lateral coordinate changes ($z = 0, \Delta\theta_{hip} = 0$).

- **Result:** $\tau_{total,hip} = 0\ N \cdot m$

Shoulder / Pitch Servo

Utilizing maximum moment arms ($r_f = 0.36\ m, r_{c1} = 0.0996\ m, r_{c2} = 0.2796\ m$) and maximum angle change ($\Delta\theta = 0.6163\ rad$):

- $\tau_{g,shoulder} = 35.79(0.36) + 0.30(9.81)(0.0996) + 0.30(9.81)(0.2796) = 14.00\ N \cdot m$
- $\tau_{I,shoulder} = 0.032\left(\frac{2(0.6163)}{0.192^2}\right) = 0.032(33.44) = 1.07\ N \cdot m$
- **Total:** $\tau_{total,shoulder} = 14.00 + 1.07 = 15.07\ N \cdot m$

Knee / Elbow Servo

Utilizing maximum moment arms ($r_{f/k} = 0.20\ m, r_{c2/k} = 0.10\ m$) and angle change ($\Delta\theta = 1.0376\ rad$):

- $\tau_{g,knee} = 35.79(0.20) + 0.30(9.81)(0.10) = 7.45\ N \cdot m$
- $\tau_{I,knee} = 0.004\left(\frac{2(1.0376)}{0.192^2}\right) = 0.004(56.29) = 0.225\ N \cdot m$
- **Total:** $\tau_{total,knee} = 7.45 + 0.225 = 7.68\ N \cdot m$

Diagram from Python simulation

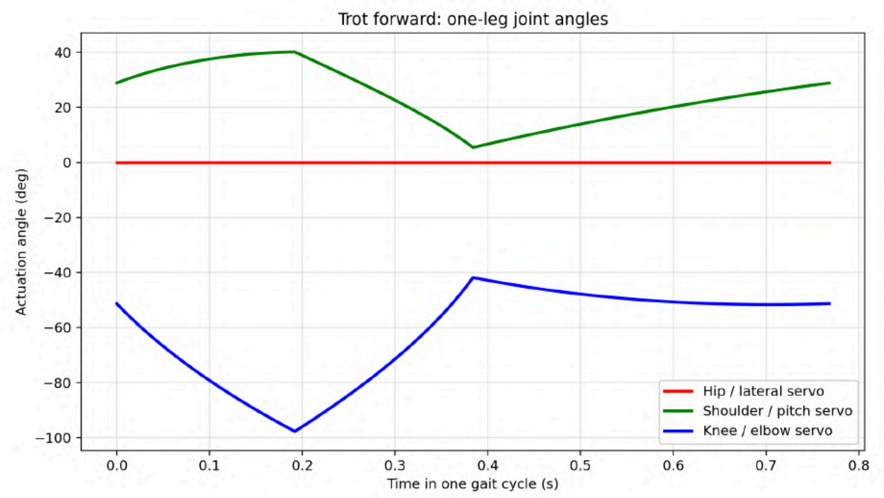


Figure 5.2: Tort Forward Angle Values with time

This figure depicts the angular trajectories for the trot gait, executed over a much faster 0.8-second cycle with a broader knee sweep (down to -100 degrees) for ground clearance. The key takeaway is that these rapid kinematic changes directly cause the higher angular accelerations and increased inertia torques defined in the theoretical calculations.

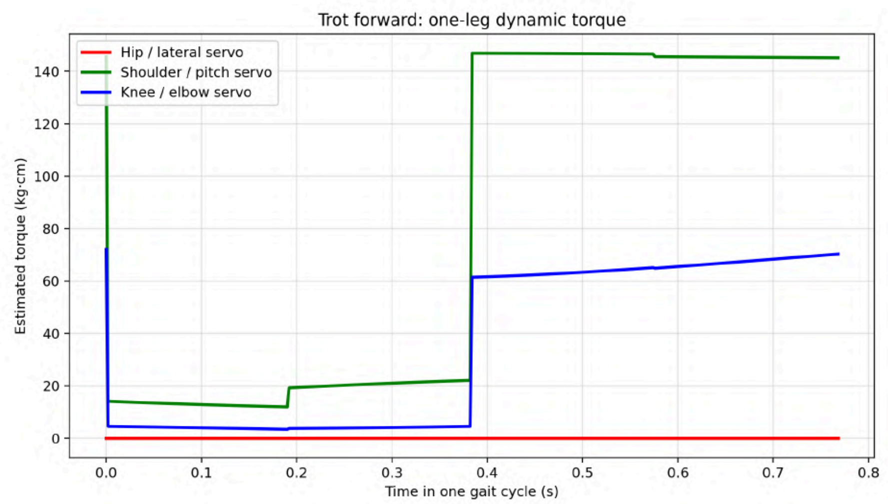


Figure 5.3: Trot Forward Torque Values

Highlighting the most demanding load conditions, this graph demonstrates the severe torque spikes that occur when the robot's mass is supported by only two legs. The shoulder servo withstands the system's absolute peak load of nearly 150 kg·cm during the stance phase, visually corroborating the prior conclusion that the shoulder/pitch joint dictates the maximum required motor sizing.

Section IV. Crawl Forward Calculation ($F_{crawl} = 23.86\ N$)

Hip / Lateral Servo

Forward crawl yields zero lateral coordinate changes.

- **Result:** $\tau_{total,hip} = 0\ N \cdot m$

Shoulder / Pitch Servo

Given maximum moment arms ($r_f = 0.36\ m, r_{c1} = 0.10\ m, r_{c2} = 0.28\ m$) and angle change ($\Delta\theta = 0.6435\ rad$):

- $\tau_{g,shoulder} = 23.86(0.36) + 0.30(9.81)(0.10) + 0.30(9.81)(0.28) = 9.71\ N \cdot m$
- $\tau_{I,shoulder} = 0.032\left(\frac{2(0.6435)}{0.192^2}\right) = 0.032(34.91) = 1.12\ N \cdot m$
- **Total:** $\tau_{total,shoulder} = 9.71 + 1.12 = 10.83\ N \cdot m$

Knee / Elbow Servo

Given maximum moment arms ($r_{f/k} = 0.20\ m, r_{c2/k} = 0.10\ m$) and angle change ($\Delta\theta = 1.2427\ rad$):

- $\tau_{g,knee} = 23.86(0.20) + 0.30(9.81)(0.10) = 5.07\ N \cdot m$
- $\tau_{I,knee} = 0.004\left(\frac{2(1.2427)}{0.192^2}\right) = 0.004(67.42) = 0.270\ N \cdot m$
- **Total:** $\tau_{total,knee} = 5.07 + 0.270 = 5.34\ N \cdot m$

Diagram from Python simulation

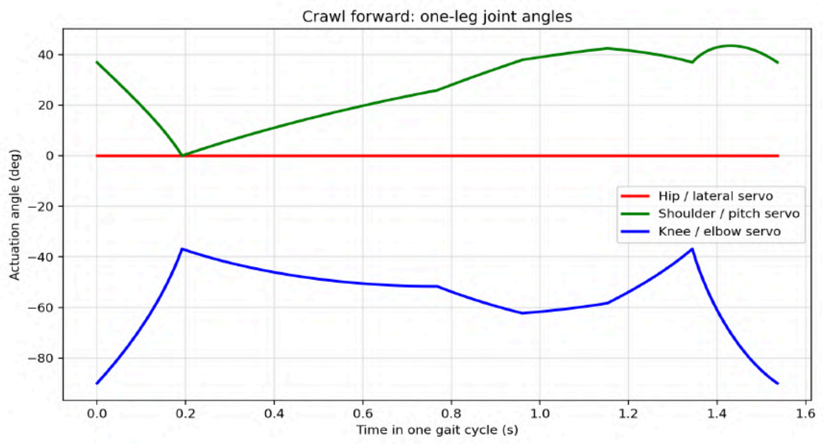


Figure 5.4: Trot Forward Torque Values

This graph illustrates the kinematic trajectories for a single leg during a 1.6-second crawl cycle, where the hip servo remains stationary at zero degrees while the shoulder and knee execute coordinated sweeping motions. The key takeaway is the smooth, extended actuation time required to maintain the statically stable three-point ground contact described in the analytical model.

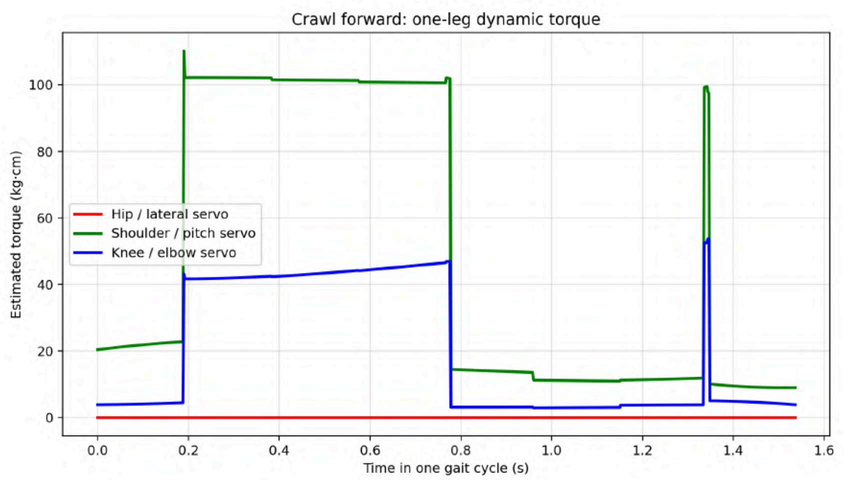


Figure 5.5: Trot Forward Torque Values

This plot displays the real-time torque loads during the crawl gait, characterized by sharp step changes that clearly distinguish the unloaded swing phase from the weight-bearing stance phase. The shoulder servo experiences the maximum load, plateauing near 100 kg·cm, which visually confirms the mathematical finding that three-leg support minimizes peak mechanical stress compared to faster gaits.

Section V. Turn-Left Calculation ($F_{turn} = 23.86\ N$)

Note: Right-turn logic is perfectly symmetric.

Hip / Lateral Servo

Given maximum lateral displacements ($z_f = 0.060\ m, z_{c1} = 0.015\ m, z_{c2} = 0.045\ m$) and angle change ($\Delta\theta = 0.2460\ rad$):

- $\tau_{g,hip} = 23.86(0.060) + 0.30(9.81)(0.015) + 0.30(9.81)(0.045) = 1.61\ N \cdot m$
- $\tau_{I,hip} = 0.032\left(\frac{2(0.2460)}{0.192^2}\right) = 0.032(13.35) = 0.427\ N \cdot m$
- **Total:** $\tau_{total,hip} = 1.61 + 0.427 = 2.04\ N \cdot m$

Shoulder / Pitch Servo

Given turning moment arms ($r_f = 0.365\ m, r_{c1} = 0.0912\ m, r_{c2} = 0.2737\ m$) and angle change ($\Delta\theta = 0.4071\ rad$):

- $\tau_{g,shoulder} = 23.86(0.365) + 0.30(9.81)(0.0912) + 0.30(9.81)(0.2737) = 9.78\ N \cdot m$
- $\tau_{I,shoulder} = 0.032\left(\frac{2(0.4071)}{0.192^2}\right) = 0.032(22.09) = 0.707\ N \cdot m$
- **Total:** $\tau_{total,shoulder} = 9.78 + 0.707 = 10.49\ N \cdot m$

Knee / Elbow Servo

Given turning moment arms ($r_{f/k} = 0.191\ m, r_{c2/k} = 0.0956\ m$) and angle change ($\Delta\theta = 0.7495\ rad$):

- $\tau_{g,knee} = 23.86(0.191) + 0.30(9.81)(0.0956) = 4.84\ N \cdot m$
- $\tau_{I,knee} = 0.004\left(\frac{2(0.7495)}{0.192^2}\right) = 0.004(40.67) = 0.163\ N \cdot m$
- **Total:** $\tau_{total,knee} = 4.84 + 0.163 = 5.00\ N \cdot m$

Diagram from Python simulation

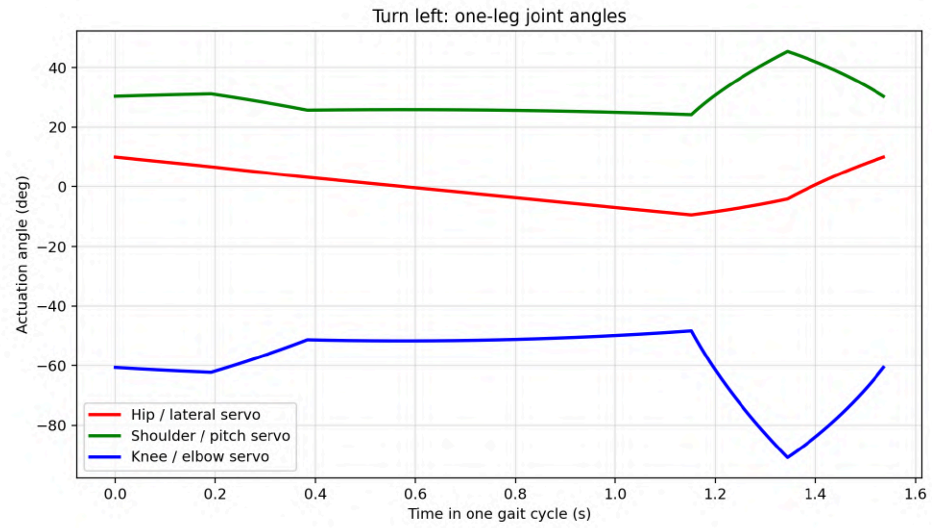


Figure 5.6: Trot Forward Torque Values

Unlike linear forward locomotion, this plot reveals active lateral kinematics, showing the hip servo oscillating continuously to facilitate the robot's rotational movement. Combined with the synchronized pitching of the shoulder and knee, these multi-axis kinematics illustrate how the robot effectively pivots while maintaining a stable tripod stance.

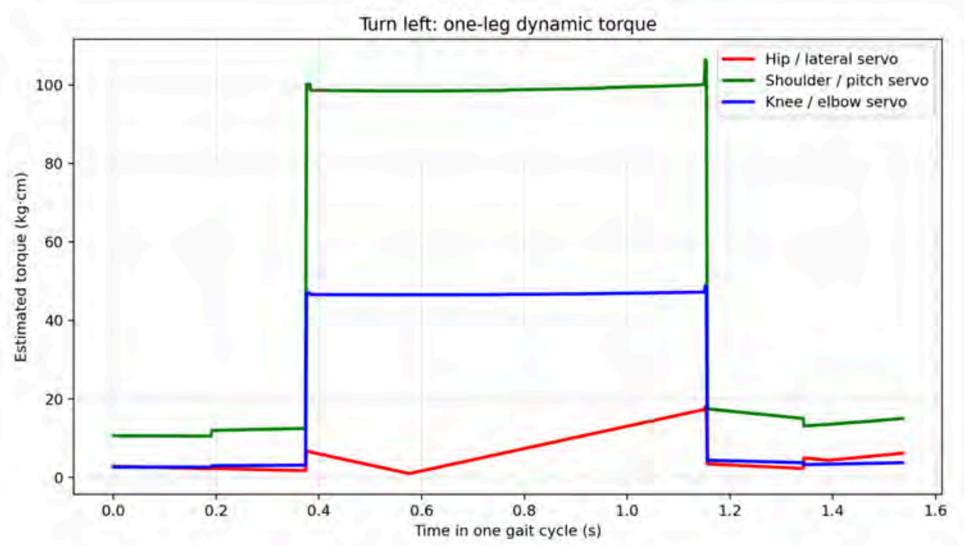


Figure 5.7: Trot Forward Torque Values

This plot outlines the mechanical loads during rotational maneuvering, uniquely showcasing the introduction of non-zero lateral torque on the hip servo that peaks near 20 kg·cm. While the shoulder and knee loads remain relatively moderate, this graph validates the earlier calculation that the hip/lateral servos must be specifically sized to handle turning forces.

Section VI. Full Leg Torque Summary & Motor Selection

TABLE 5.1: TABLE OF LEG TORQUE RATINGS

Gait Case	Joint	τ_g (N·m)	τ_I (N·m)	Total (N·m)	Total (kg·cm)	Final w/ SF (kg·cm)
Trot	Hip	0.00	0.00	0.00	0.00	0.00
	Shoulder	14.00	1.07	15.07	153.68	**184.42**
	Knee	7.45	0.23	7.68	78.30	**93.96**
Crawl	Hip	0.00	0.00	0.00	0.00	0.00
	Shoulder	9.71	1.12	10.83	110.39	132.47
	Knee	5.07	0.27	5.34	54.42	65.30
Turn	Hip	1.61	0.43	2.04	20.75	**24.91**
	Shoulder	9.78	0.71	10.49	106.97	128.36
	Knee	4.84	0.16	5.00	51.03	61.24

Taking the maximum required torques across all gaits gives the following recommended minimum motor ratings:

- $\tau_{hip/lateral} \geq 25\ kg \cdot cm$
- $\tau_{shoulder/pitch} \geq 185\ kg \cdot cm$
- $\tau_{knee/elbow} \geq 95\ kg \cdot cm$

The shoulder/pitch servo remains the limiting joint as it carries the largest body-load moment arm during the trot gait.

Part A. Robotic Arm Torque Calculations

Section I. Arm Geometry and Mass Model

The robotic arm consists of two equal segments and a payload capacity:

- **Segment 1 (Shoulder to Elbow):** $L_1 = 0.25\ m, m_1 = 0.25\ kg$
- **Segment 2 (Elbow to Claw):** $L_2 = 0.25\ m, m_2 = 0.25\ kg$
- **Payload at Claw:** $m_p = 0.30\ kg$
- **Operation Parameters:** $T = 1.20\ s$ (hard-limit full displacement), $SF = 1.2$.

Section II. General Equations

The fundamental torque equations are analogous to those of the legs ($\tau_g = \sum m_i g r_i$ and $\tau_I = I\alpha$). For base joint inertia, parallel axis theorem and point mass models are applied.

Section III. Lateral Base and Shoulder Servos Calculation

Both the lateral base servo and shoulder servo share the same origin point. For a conservative hard-limit sizing, both are calculated identically assuming the arm is fully extended: $r_{c1} = 0.125\,m, r_{c2} = 0.375\,m, r_p = 0.50\,m$.

- **Gravity Torque:**

  $\tau_{g,base} = 0.25(9.81)(0.125) + 0.25(9.81)(0.375) + 0.30(9.81)(0.50) = 2.697\,N \cdot m$

- **Inertia Torque:**

  Total inertia consists of Segment 1 (I_1), Segment 2 (I_2), and Payload (I_p).

  $$I_{base} = \left[\frac{1}{3} m_1 L_1^2\right] + \left[m_2\left(\frac{L_2}{2}\right)^2\right] + \left[m_p r_p^2\right]$$

  $I_{base} = 0.00521 + 0.03646 + 0.075 = 0.11667\,kg \cdot m^2$

  Assuming maximum angular displacement ($\Delta\theta = 180^\circ = \pi\,rad$):

  $$\tau_{I,base} = I_{base}\left(\frac{2\pi}{1.20^2}\right) = 0.11667(4.363) = 0.509\,N \cdot m$$

- **Total Required Torque (Lateral & Shoulder):**

  $\tau_{total,base} = 2.697 + 0.509 = 3.206\,N \cdot m$

  $\tau_{final,base} = 1.2(3.206) = 3.847\,N \cdot m = 39.23\,kg \cdot cm$

Section IV. Arm Elbow Servo Calculation

The elbow servo carries only the forearm segment and the payload ($r_{c2/e} = 0.125\,m, r_{p/e} = 0.25\,m$).

- **Gravity Torque:**

  $\tau_{g,elbow} = 0.25(9.81)(0.125) + 0.30(9.81)(0.25) = 1.042\,N \cdot m$

- **Inertia Torque:**

  $$I_{elbow} = \left[\frac{1}{3} m_2 L_2^2\right] + \left[m_p L_2^2\right] = 0.00521 + 0.01875 = 0.02396\,kg \cdot m^2$$

  Assuming code range angular displacement ($\Delta\theta = 135^\circ = 2.356\,rad$):

  $$\tau_{I,elbow} = I_{elbow}\left(\frac{2(2.356)}{1.20^2}\right) = 0.02396(3.272) = 0.078\,N \cdot m$$

- **Total Required Torque (Elbow):**

  $\tau_{total,elbow} = 1.042 + 0.078 = 1.120\, N \cdot m$

  $\tau_{final,elbow} = 1.2(1.120) = 1.344\, N \cdot m = 13.71\, kg \cdot cm$

Section V. Arm Torque Summary & Motor Recommendation

TABLE 5.2: TABLE OF ARM TORQUE RATINGS

Arm Joint	τ_g (N·m)	τ_I (N·m)	Total (N·m)	Total (kg·cm)	Final w/ SF (kg·cm)
Lateral Base	2.697	0.509	3.206	32.69	**39.23**
Shoulder	2.697	0.509	3.206	32.69	**39.23**
Elbow	1.042	0.078	1.120	11.42	**13.71**

After applying the $1.2\times$ safety factor, practical rounded-up target ratings for real hardware acquisition are:

- **Lateral base servo:** $\sim 45–50\ kg \cdot cm$ (Absolute minimum $40\ kg \cdot cm$)
- **Shoulder servo:** $\sim 45–50\ kg \cdot cm$ (Absolute minimum $40\ kg \cdot cm$)
- **Elbow servo:** $\geq 20\ kg \cdot cm$ (Absolute minimum $15\ kg \cdot cm$)

Section VI. Claw Servo Torque

The claw servo does not support the whole arm as a long lever. Its main requirement is gripping force. The payload capacity is approximately 300 g, but the actual claw servo requirement depends on finger geometry, friction coefficient, object shape, and contact distance from the claw pivot. Therefore, the 300 g payload value should be interpreted as the designed object-carrying capacity rather than a direct kg-cm torque value for the claw servo. So the minimum torque must be at least or greater than 300 g · cm.

5.2.2 Components sourcing

Based on the torque analysis, inverse kinematics requirements, structural design constraints, and overall functional objectives of the quadruped robotic system, a detailed component selection and sourcing process was carried out. The selected components were chosen to satisfy the mechanical load requirements, real-time control requirements, power delivery demands, wireless communication capability, sensing functionality, and structural stability necessary for stable locomotion and robotic manipulation. Both online electronics retailers and local physical hobbyist and hardware stores were used to procure the required components depending on availability, cost, and technical specifications.

The following list summarizes the major hardware components used in the system along with their specifications and short descriptions regarding their role within the robot architecture. Although several additional supporting items such as wiring, fasteners, connectors, regulators, adhesives, and assembly tools were also used during the fabrication process, the components listed below represent the primary subsystems responsible for locomotion, control, sensing, power distribution, and structural construction of the robot platform.

ESP32-S3 Microcontroller

<table>
<tr><td rowspan="6"></td><th>Specification</th><th>Rating / Value</th></tr>
<tr><td>Processor</td><td>Dual-core Xtensa LX7</td></tr>
<tr><td>Clock Speed</td><td>Up to 240 MHz</td></tr>
<tr><td>Wireless</td><td>2.4 GHz Wi-Fi + Bluetooth LE</td></tr>
<tr><td>Memory</td><td>512 KB SRAM, 384 KB ROM</td></tr>
<tr><td>Interface</td><td>USB Type-C</td></tr>
<tr><td colspan="3">The ESP32-S3 is used as the primary controller of the quadruped robot, handling locomotion, inverse kinematics calculations, gait generation, and wireless UDP communication for real-time robot control. Its onboard Wi-Fi capability allows the robot to receive motion commands and coordinate all leg and arm actuation systems simultaneously.</td></tr>
</table>

XIAO ESP32-C6

Specification	Rating / Value
Processor	Dual 32-bit RISC-V
Clock Speed	Up to 160 MHz
Wireless Connectivity	Wi-Fi 6, Bluetooth 5, Zigbee, Thread
Memory	512 KB SRAM, 4 MB Flash
Dimensions	21 × 17.8 mm

The XIAO ESP32-C6 is used as a secondary microcontroller dedicated to environmental sensing and auxiliary subsystem management. It interfaces with sensors such as gas, temperature, and ultrasonic modules while also supporting wireless IoT communication and real-time monitoring of the robot's surroundings.

XIAO ESP32-S3 Sense Camera Module

Specification	Rating / Value
Processor	ESP32-S3 Dual-core LX7
Camera	OV3660 Camera
Wireless	Wi-Fi + Bluetooth 5.0
Memory	8 MB PSRAM + 8 MB Flash
Storage	MicroSD Card Support

The XIAO ESP32S3 Sense module is used as the robot's onboard vision system for live video streaming and real-time inspection tasks. This allows the robot to visually monitor the environment and support remote observation during autonomous or manual operation

Fermion 10-DOF IMU Sensor

Specification	Rating / Value
Communication	I2C
Input Voltage	3–8 V
Sensors	Accelerometer, Gyroscope, Magnetometer, Barometer
Main ICs	ADXL345, ITG3205, VCM5883L, BMP280
Dimensions	26 × 18 mm

The 10-DOF IMU sensor provides orientation awareness by measuring roll, pitch, acceleration, and angular motion using the accelerometer and gyroscope modules. The controller uses this information to dynamically stabilize the robot body and compensate for tilt or uneven terrain during walking.

High Torque Servo (RDS51150)

Specification	Rating / Value
Torque	173 kg·cm @ 12.6 V
Operating Voltage	10 – 12.6 V
Speed	0.19 sec / 60° @ 12.6 V
Weight	165 g
Rotation Control	Digital PWM Servo

The RDS51150 high-torque servos are used as the primary actuators for the quadruped leg mechanism, with three servos assigned to each leg for lateral, shoulder, and elbow motion. These servos provide the torque necessary to support the robot's body weight, execute gait cycles, and maintain posture during locomotion.

DS3235 35KG Servo Motor

Specification	Rating / Value
Torque	35 kg·cm
Rotation Range	180°
Protection	IP66 Waterproof
Gear Type	Metal Gear
Control Method	PWM Servo Control

The DS3235 digital servos are used in the robotic arm and claw subsystem because they provide high torque with comparatively lightweight construction. These servos actuate the shoulder, elbow, and gripper joints of the arm, enabling object manipulation and payload handling.

PCA9685 PWM Driver

Specification	Rating / Value
PWM Channels	16 Channels
Communication	I2C
PWM Resolution	12-bit
Output Support	Up to 6 V
Board Size	62 × 26 mm

The PCA9685 PWM driver board is used to control all sixteen servos of the robot through the I2C communication protocol. Twelve channels are allocated to the quadruped legs and four channels are assigned to the robotic arm and claw, allowing synchronized multi-servo actuation from the microcontroller.

Li-Power 3S Battery Pack

Specification	Rating / Value
Battery Type	Li-Po 3S
Voltage	11.1 V
Capacity	2200 mAh
Discharge Rate	35C
Weight	240 g

The 3S LiPo battery is used as the main high-current power source for the twelve high-torque leg servos of the quadruped system. Its high discharge capability is necessary because multiple actuators operate simultaneously during walking, turning, and stabilization maneuvers.

Li-Power 2S Battery Pack

Specification	Rating / Value
Battery Type	Li-Po 2S
Voltage	7.4 V
Discharge Rate	25C
Application	Arm Servo Power
Rechargeable	Yes

The 2S LiPo battery is dedicated to powering the four servos used in the robotic arm and claw assembly. Separating the arm power system from the main locomotion actuators helps maintain stable voltage delivery and improves overall power reliability.

18650 Battery Holder Module

Specification	Rating / Value
Battery Support	2 × 18650 Cells
Charging Input	5 V Micro-USB
Output Voltage	5 V USB
Output Current	Up to 1 A
Dimensions	99.7 × 48.2 × 20 mm

The 18650 battery holder is used to supply portable power to the main controller boards, camera module, and sensor systems. It also includes an integrated charging and USB output circuit, simplifying onboard low-power electronics management.

Panasonic 18650 Li-ion Battery

Specification	Rating / Value
Battery Type	Li-ion 18650
Rated Voltage	3.7 V
Capacity	3400 mAh
Charging Voltage	4.2 V
Cell Weight	45 g

The Panasonic Li-ion cells are installed inside the battery holder to provide stable rechargeable power for the microcontrollers, camera system, and onboard sensing modules. These batteries support long-duration low-power operation independent of the high-current servo supply system.

PLA+ Filament

Specification	Rating / Value
Material Type	PLA+
Filament Diameter	1.75 mm
Spool Weight	1 kg
Tolerance	±0.02 mm
Features	High Strength & Low Warp

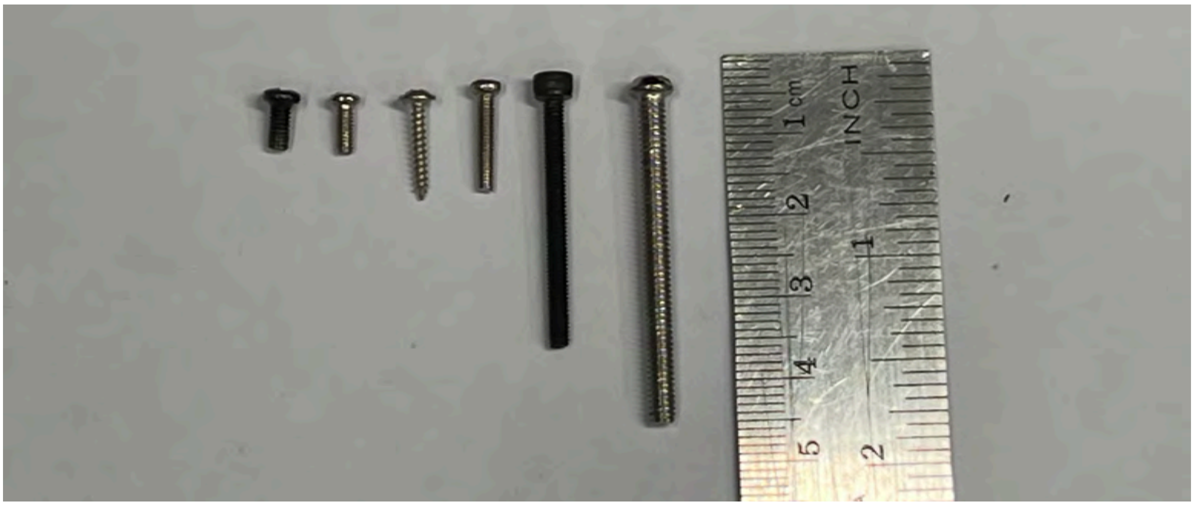
PLA filament is used for manufacturing the robot body shell, servo mounts, joint housings, and structural connection components through 3D printing. This allows rapid prototyping, lightweight construction, and easy customization of the mechanical structure.

Carbon Fiber Rod

Specification	Rating / Value
Length	1 m
Outer Diameter	25.5 mm
Wall Thickness	0.3 mm
Material	HM Carbon Fiber
Weight	26 g

The carbon fiber rods form the primary structural frame of the quadruped robot and provide high stiffness with low weight. This improves mechanical strength and reduces unnecessary load on the actuators during walking and manipulation tasks.

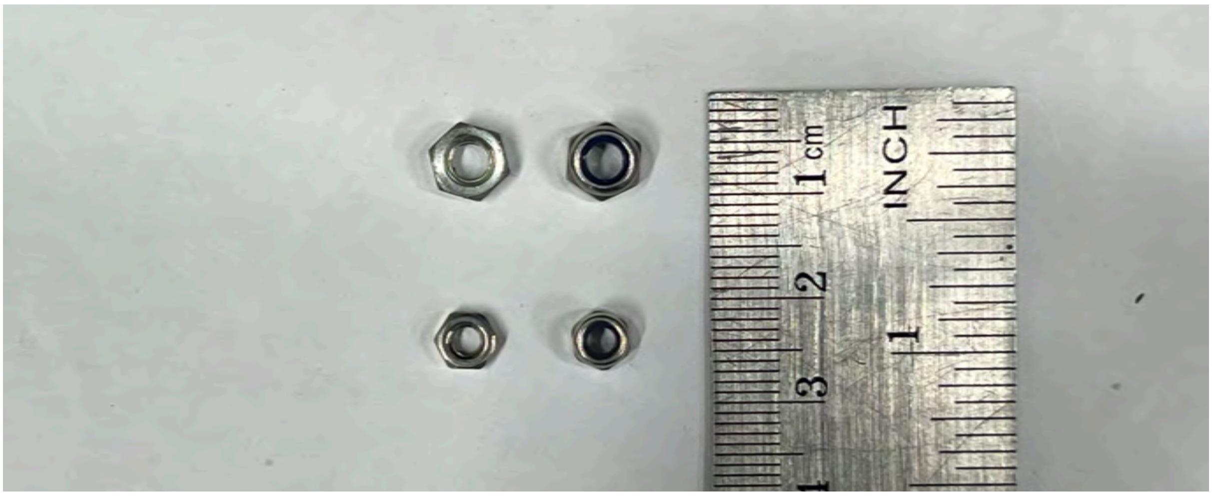
Screws



Different screw diameters and lengths were selected depending on the mechanical load, required fastening depth, and whether the connection was structural, actuator-related, or lightweight assembly. Larger M4 screws were mainly used for the carbon fiber structural frame together with locknuts, while M2.5 screws were extensively used for mounting the high-torque servos to the printed body because the servo housing itself contains threaded mounting holes. Smaller M2 self-tapping screws were used in lightweight arm structures where direct threading into the PLA body was required without additional nuts. Most screws used Philips heads for easier assembly, while some M3 structural screws used hex heads for improved tightening strength and reliability.

Screw Type	Diameter	Length	Head Type	Main Application
Structural Frame Screw	M4 (4 mm)	40 mm	Philips	Fastening carbon fiber tubes to 3D printed structural joints
Structural Interlocking Screw	M3 (3 mm)	15 mm	Hex	Connecting multiple 3D printed body and claw components
Servo Mounting Screw	M2.5 (2.5 mm)	10 mm	Philips	Mounting 150 kg·cm servos to the robot body and leg assemblies
Servo Bracket Screw	M2.5 (2.5 mm)	5 mm	Philips	Attaching metallic servo brackets and accessories
Servo Horn Locking Screw	M3 (3 mm)	5 mm	Hex	Securing servo horns to servo shafts and linkage structures
Self-Tapping Screw	M2 (2 mm)	10 mm	Philips	Lightweight robotic arm assembly and direct PLA fastening

M3 and M4 Locknuts



We used M3 and M4 locknuts, along with matching M3 and M4 screws, to secure the mechanical joints and structural components of the robotic dog. The locknuts helped prevent loosening caused by vibration and making the overall structure more mechanically stable during operation.

Nut Type	Supported Screw Size	Type	Main Application	Nut Type
Structural Locknut	M4	Nylon Insert Locknut	Locking carbon fiber frame joints and major structural connections	Structural Locknut
Structural Locknut	M3	Nylon Insert Locknut	Securing 3D printed structural assemblies and claw mechanisms	Structural Locknut

5.2.3 Iterative Development of 3D-Printed Parts and Walking Algorithm

The final system design was completed through three design iterations. In the first iteration, a crawling algorithm was used for locomotion. However, this approach was mechanically unstable and made the robot difficult to walk smoothly.

Therefore, in the second iteration, a trotting algorithm was introduced to reduce foot-ground friction and improve the smoothness of the walking mechanism. During this stage, the shoulder and body design were also modified to reduce mechanical wobble. A robotic arm with a metal claw was attached, and the 3D-printed feet were replaced with soft ball-type feet covered with car mat material to improve ground grip.

In the third iteration, further mechanical improvements were made to increase stability and functionality. Servo enclosures were designed for the shoulder servos, and 3D-printed servo brackets were used instead of metal brackets because the metal brackets caused additional wobbling. The inner body was redesigned as an enclosure to protect and organize the internal components. A 3D-printed claw was also developed to provide a stronger grip when pulling the lever during responsive operation. In addition, separate holders were designed for the camera, cooling fan, MQ-2 gas sensor, and DHT11 temperature sensor. Finally, a sliding latch mechanism was added to securely attach the enclosure to the main body of the robot.

Below, the CAD models and real-life images of the 3D-printed components are presented, with that a brief description is also provided to explain the design changes made during each iteration:

5.2.3.1 1st Iteration

Component 1. Knee Joint

Model on Fusion360	3D printed Component
	
This servo mount joint features a hollow cylindrical barrel centered on a rectangular base plate (35 × 48 mm footprint, 50 mm total height), with the open-top bore and side fastener holes designed to house a rotary servo motor. Four corner mounting holes on the rounded-corner base plate secure the joint to the robot's leg linkage.	

Component 2. Shoulder Joint

<table>
<tr><th>Model on Fusion360</th><th>3D printed Component</th></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">This quadruped servo joint is an L-shaped elbow connector comprising two perpendicular hollow cylinders, a vertical bore (40 mm diameter) and a horizontal barrel (35 mm diameter) joined by a smooth filleted knuckle. The overall envelope is approximately 40 × 55 × 80 mm, with bolt-circle fastener holes on both open faces for mounting servo motors at each axis.</td></tr>
</table>

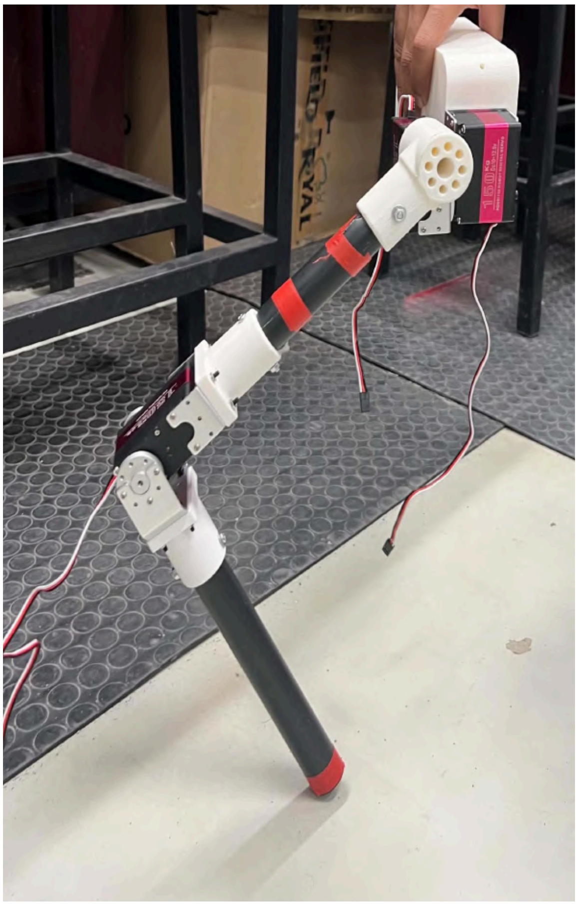
Component 3. Body

<table>
<tr><th>Model on Fusion360</th><th>3D printed Component</th></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">A rectangular box-shaped body enclosure (154 × 94 × 60 mm) with heavily rounded vertical edges forms the main chassis of this quadruped robot. One end face features a grid of bolt holes for servo motor attachment, with a small side fastener securing the joint assembly.</td></tr>
</table>

Component 4. Foot:

Model on Fusion360	3D printed Component
	
A spherical base (50 mm diameter) transitions upward into a short hollow cylindrical servo mount (30 mm diameter, 65 mm total height), forming a foot housing. Small side fastener holes on the cylinder secure the leg with the tube in place.	

Component 5. PVC Pipe Leg Segments

	As this was the first iteration of our leg design, we needed to confirm and verify whether the 3D-printed parts designed in Fusion 360 could hold up under pressure and fit properly with the robot body. We also needed to check whether the size, ratio, and overall structure were coherent before moving to the final material. Therefore, we needed a fast and cost-effective way to validate our assumptions. For the initial construction of the leg, we used PVC pipes instead of carbon fiber tubes because carbon fiber is harder to cut, more difficult to handle, and produces harmful micro-dust during fabrication. In this way, we developed the first prototype of the leg using PVC tubes to check the fit, alignment, structural proportions, and basic mechanical feasibility before finalizing the carbon fiber version.

5.2.3.2 2nd Iteration:

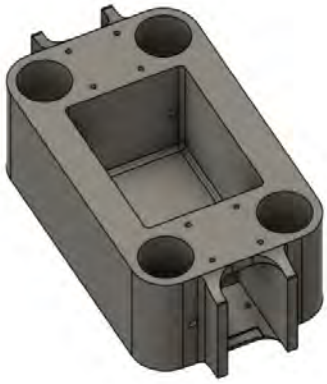
Component 1. Arm Claw Mount

Model on Fusion360	3D printed Component
	
An open U-shaped bracket (90 × 70 mm footprint) with two parallel side walls and a flat base houses a horizontal cylindrical servo motor mount (35 mm diameter) nestled into one corner. Countersunk bolt holes on the side walls and flanged base tabs secure both the servo and the bracket to the robot's leg frame.	

Component 2. Shoulder Version 2

Model on Fusion360	3D printed Component
	
The difference from version one of this shoulder joint is that this has now its lips removed so that metal brackets that come with the servo motor can also be placed before placing this connector to add extra support and reduce wobble.	

Component 3. Body Version 2

<table>
<tr><th>Fusion 360 Model</th><th>3D printed Component</th></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">The difference of this body with the version one is that it now has servo motor mounting space built in inside the body itself, which had mounting place outside. This gives superb rigidity and unprecedented stability. A rectangular open-top box (184 × 94 × 57 mm) with heavily rounded long edges features four large cylindrical servo mount bores at its corners and a recessed rectangular cavity in the top face. A protruding servo collar on one short end and small bolt holes throughout complete this quadruped body frame.</td></tr>
</table>

Component 4. Claw

This metal claw mechanism was initially used to grip and pull the emergency lever during responsive operation. However, testing showed that the metal tips were not strong enough to hold the lever firmly, which reduced the pulling reliability and led to further design improvement.	

5.2.3.3 3rd Iteration

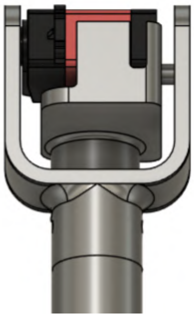
Component 1. Shoulder Servo Enclosure

Model on Fusion360	3D printed Component
	
A solid open-top rectangular block (79 × 46 × 52 mm) with lightly rounded edges features a large circular through-bore (30 mm diameter) on one side face for servo motor insertion, a bolt-circle pattern on the inner back wall, and a small notched clamp tab along the top edge for shaft retention.	

Component 2. Servo Bracket for 150 Kg servo

Model on Fusion360	3D printed Component
	
A T-shaped servo knuckle joint (98 × 72 × 45 mm) combines a horizontal cylindrical shaft bore (30 mm diameter) with a swept flange face bearing a bolt-circle pattern on one side and a small clamp yoke on top, all blended together with smooth organic fillets.	

Component 3. Servo Holder for 35 Kg servo

Fusion 360 Model	3D printed Component
	
A compact L-shaped servo bracket (61 × 53 × 47 mm) combines a hollow cylindrical shaft bore on one end with an open U-channel cradle on the other, joined by a flat triangular gusset wall. Small bolt holes on the cradle flanges and cylinder secure the servo motor and leg linkage.	

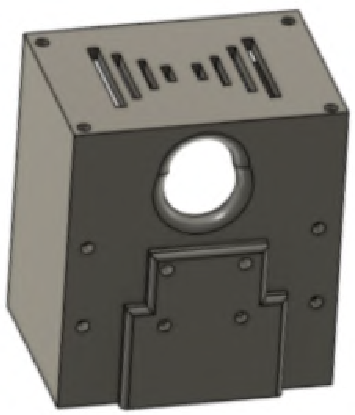
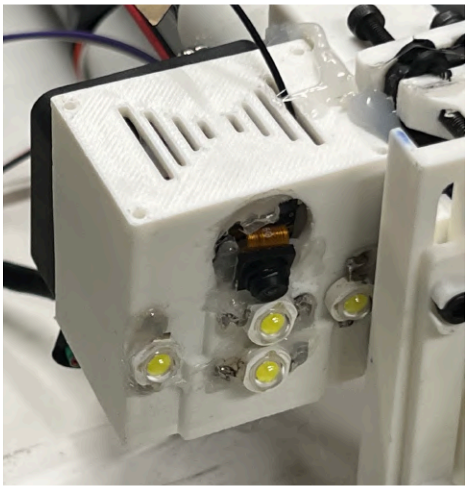
Component 4. Servo bracket for 35 Kg servo

Fusion 360 Model	3D printed Component
	
A U-shaped yoke bracket (90 × 61 × 30 mm) features two parallel rounded arms with a central shaft bore (15 mm) on the lower flange, a small cylindrical servo stub on one side, and a bolt-circle cluster on the outer arm face for servo motor attachment.	

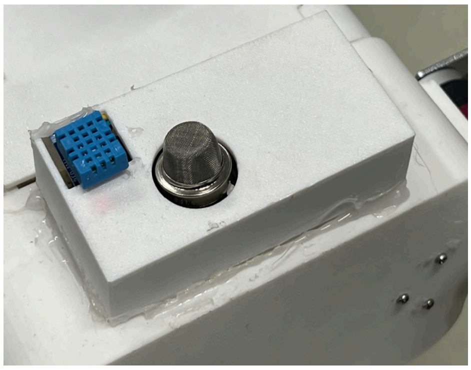
Component 5. Inner Body

Fusion 360 Model	**3D printed Component**
	
A tall rectangular body frame (149 × 96 × 180 mm) consists of four vertical hollow cylindrical columns (20 mm diameter) fused to flat connecting side walls, forming an open-bottom structural cage. Interlocking notched slots along the wall edges and open-top bores allow servo motor shafts and leg assemblies to be mounted at each corner.	

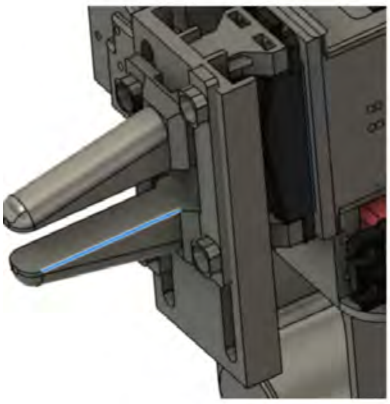
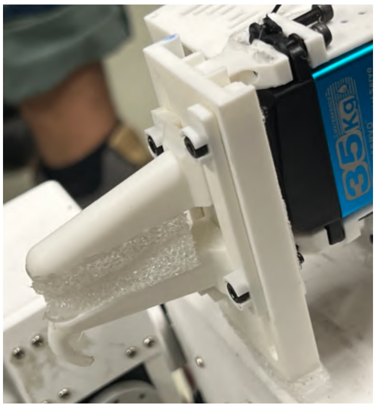
Component 6. Camera and Fan Holder

Fusion 360 Model	3D printed Component
	
A compact open-top rectangular enclosure (42 × 32 × 47 mm) serves as the robot's head, with ventilation slots on the front face, a small raised camera mount boss on top, and corner bolt holes on all walls for securing servo motor mounts and electronics inside.	

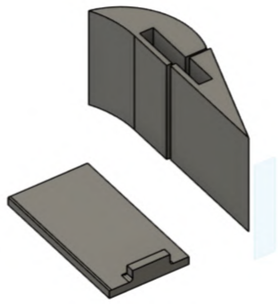
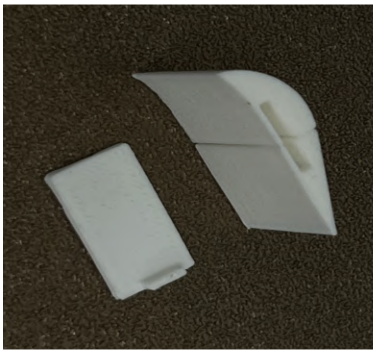
Component 7. Gas and Temperature Sensor Encloser

Fusion 360 Model	3D printed Component
	
A flat rectangular sensor mounting tray (89 × 46 × 22 mm) features a large circular through-hole (20 mm diameter) centered on the top face for a sensor lens, a small square pocket in one corner for a connector, and a thin open-ended U-channel lip along one short edge for clip-on attachment.	

Component 8. Claw

Fusion 360 Model	**3D printed Component**
	
An L-shaped claw jaw arm (56 × 45 × 24 mm) combines a flat rectangular upper rail with a tapered cylindrical finger (10 mm tip diameter) angled downward at roughly 45°. Two small square servo pivot bosses with through-bores sit at each end of the rail for joint attachment.	

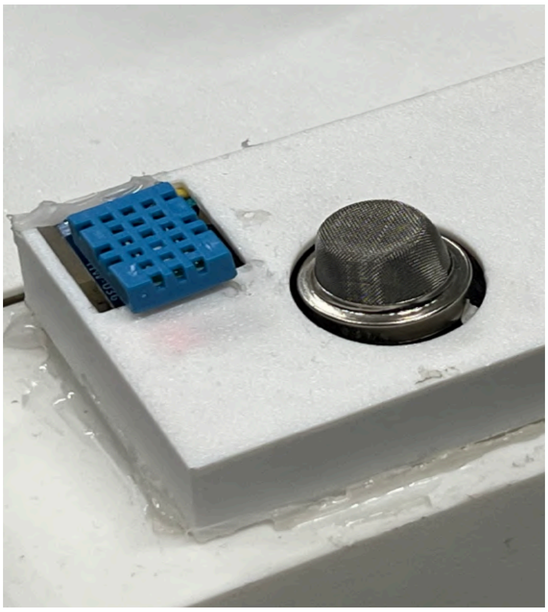
Component 9. Latch

Fusion 360 Model	3D printed Component
	
A set of three interlocking latch pieces, a flat rectangular base plate (49 × 20 mm), a D-shaped curved block, and a thin triangular wedge tab, each featuring a small rectangular tongue or slot (5 mm) on top for snap-fit body panel joining.	

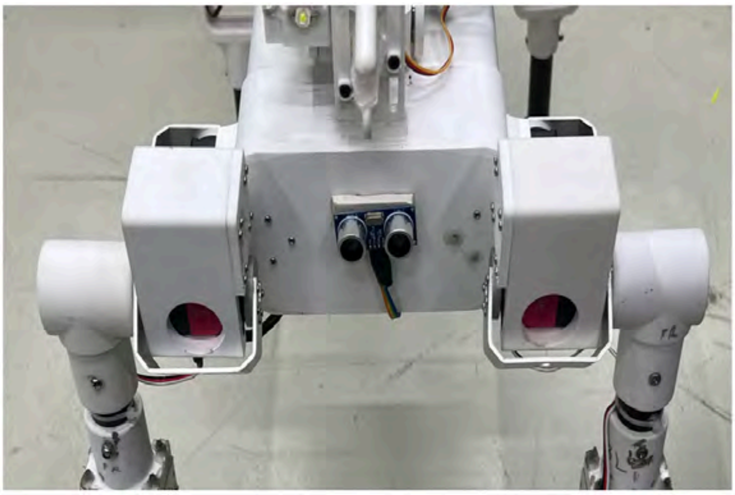
Component 10. Gamepad or Controller

The Logitech F710 Wireless Gamepad was used to control the robotic dog during operation. The gamepad allowed the user to control both the walking and navigation system for the robotic arm during responsive operation. Different buttons on the gamepad were assigned to control the legs, arm movement, and claw operation, enabling the complete robot system to be operated using a single controller.	

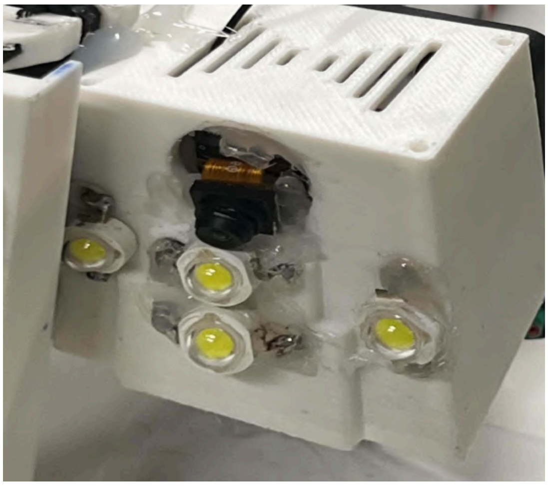
Component 11. Gas and Temperature sensor:

	A sensor holder was attached to the end edge of the robot body to mount the environmental sensing module. An MQ-2 gas sensor was used to detect the presence of gas and provide real-time gas-level output on the GUI interface. Alongside this, a DHT11 temperature sensor was used to measure the surrounding temperature, and the processed temperature reading was displayed on the GUI. The MQ-2 gas sensor and DHT11 temperature sensor were integrated into the robotic dog to support real-time environmental monitoring during operation.

Component 12. Ultrasonic Sensor:

An ultrasonic sensor was attached to the front side of the robot's face to detect obstacles in its path. This sensor helps the robotic dog measure the distance between itself and nearby objects, allowing it to stop or avoid obstacles during movement.	

Component 13. LED Light

LED lights were installed below the camera to improve visibility during operation in dark or low-light environments. This lighting arrangement allows the robotic dog to navigate more safely and helps the operator receive a clearer camera view for crack detection, obstacle observation, and emergency-response tasks.	

Final Full CAD model of the robotic dog

Fully assembled robotic dog structure	Exploded view with individual components

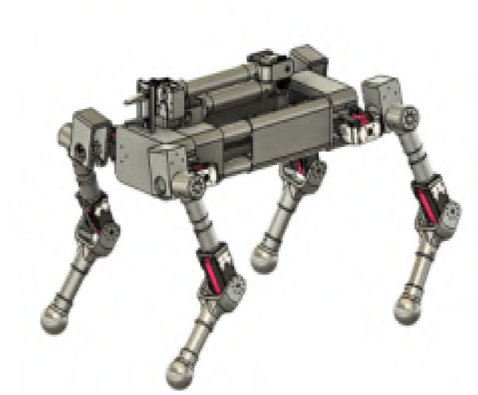
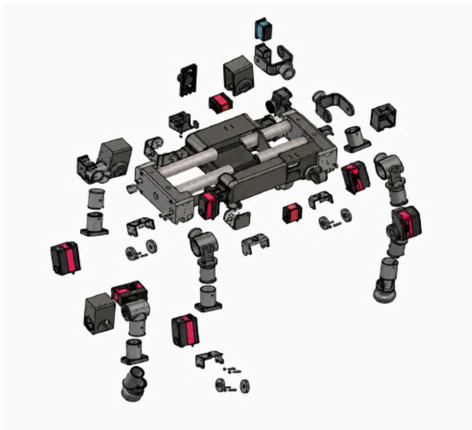

	

Figure 5.8: Exploded View of Full CAD	

We had after three interactions and several changes made this version 3 as the most stable version of the robot. We have put together a nice and assembled visualization of the robot in Fusion 360. And this is the version we show during the project showcasing, In between, some parts of previous iteration were reused and recycled. Therefore the previous versions were disassembled to make the new one.

5.2.4 Circuits and electrical component description

5.2.4.1 Leg and Arm Control Circuit Architecture and Pin Configuration

This section explains the complete electronic circuit architecture used to operate the quadrupedal robot. It includes the main control circuit based on the XIAO ESP32-C6, the PCA9685 servo driver connection, separate power supplies for the leg and arm servos, and the sensor connections through the I2C bus. The section also provides the pin configuration tables, I2C device addresses, PCA9685 channel mapping for 12 body motors and 4 arm motors, and PWM parameters used for servo control. Overall, this section shows how the controller, sensors, servo driver, actuators, and power sources are connected and coordinated to control the robot's walking, arm movement, obstacle detection, and wireless operation.

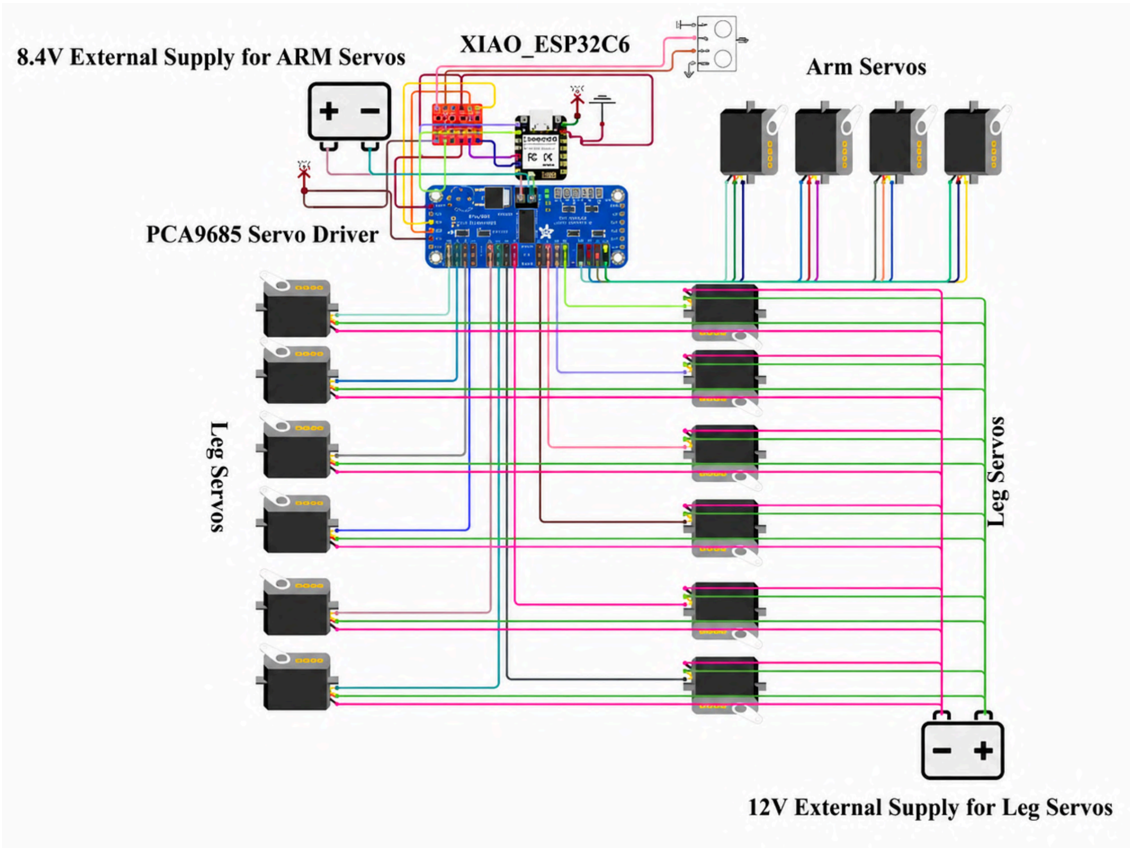


Figure 5.9: Leg-Actuator and Arm Control Circuit

We are using XIAO_ESP32C6 as our main control algorithm implementation microcontroller. The firmware was prepared via Arduino IDE. The Leg-Actuator and Arm Control Circuit uses an XIAO ESP32-C6 as its central controller, coordinating all motion and sensing over a shared I2C bus running at 400 kHz on pins D4 (SDA) and D5 (SCL). A PCA9685 16-channel PWM driver (0x40) sits on this bus and drives all 16 servos, twelve leg servos across four legs (Front-Right, Front-Left, Back-Left, Back-Right), each with a lateral, shoulder, and elbow joint on channels 0–11, plus a 4-DOF arm (lateral, shoulder, elbow, claw) on channels 12–15. All leg servos operate over a 0–270° range centered at 135° with per-joint mechanical offset corrections, while arm servos use a 0–180° range centered at 45°, and all PWM signals are bounded between 550 µs and 2500 µs at 50 Hz. An ADXL345 accelerometer (0x53) and ITG3205 gyroscope (0x68) also share the same I2C bus, feeding roll and pitch data into a complementary filter for active body-leveling that adjusts leg heights in real time. A front-facing HC-SR04 ultrasonic sensor, connected to D0 (trigger) and D1 (echo), provides obstacle detection, freezing all motion when an object comes within a configurable stop threshold, while the ESP32-C6's onboard Wi-Fi receives UDP commands wirelessly to control walking gaits, turning, arm poses, and demo sequences.

Below is the PIN Configuration Table:

1. XIAO ESP32-C6 Direct Pins

TABLE 5.3: TABLE OF XIAO ESP32-C6 DIRECT PINS

Pin	Role	Device	Notes
D0	TRIG (OUTPUT)	HC-SR04 Sonar	Front obstacle trigger
D1	ECHO (INPUT)	HC-SR04 Sonar	Front obstacle echo
D4	I2C SDA	PCA9685 / ADXL345 / ITG3205	Shared I2C bus
D5	I2C SCL	PCA9685 / ADXL345 / ITG3205	Shared I2C bus, 400 kHz

2. PCA9685 Channel Map: Leg Servos

TABLE 5.4: TABLE OF PCA9685 CHANNEL MAP: LEG SERVOS

PCA9685 CH	Leg	Joint	Sign	Offset (deg)
0	FR (Front-Right)	Lateral	+1	+13.0
1	FR	Shoulder	−1	−3.0
2	FR	Elbow	+1	−10.0
3	FL (Front-Left)	Lateral	−1	−32.0
4	FL	Shoulder	+1	−10.0
5	FL	Elbow	−1	−13.0
6	BL (Back-Left)	Lateral	−1	−30.0

7	BL	Shoulder	+1	−2.0
8	BL	Elbow	−1	−6.0
9	BR (Back-Right)	Lateral	+1	+13.0
10	BR	Shoulder	−1	−18.0
11	BR	Elbow	+1	−13.0

3. PCA9685 Channel Map: Arm Servos

TABLE 5.5: TABLE OF PCA9685 CHANNEL MAP: ARM SERVOS

PCA9685 CH	Joint	Offset (deg)	Rel. Range (deg)
12	Arm Lateral	+79.0	−45 to +135
13	Arm Shoulder	+102.0	−180 to 0
14	Arm Elbow	−43.0	0 to +135
15	Arm Claw	+23.0	−40 to +64

4. Servo PWM Parameters

TABLE 5.6: TABLE OF SERVO PWM PARAMETERS

Parameter	Value
PWM Frequency	50 Hz
Min pulse width	550 µs
Max pulse width	2500 µs
Leg servo center	135° (0–270° range)
Arm servo center	45° (0–180° range)

5.2.4.2 Camera Feed System Circuit and Configuration

This section explains the wireless camera feed system built around the XIAO ESP32S3 Sense camera module. It includes the camera circuit connection, cooling fan setup, stream port, server URL, UDP port, Wi-Fi mode, and camera configuration parameters. Overall, this section shows how the robot captures live video and streams it wirelessly to the remote operator for real-time monitoring.

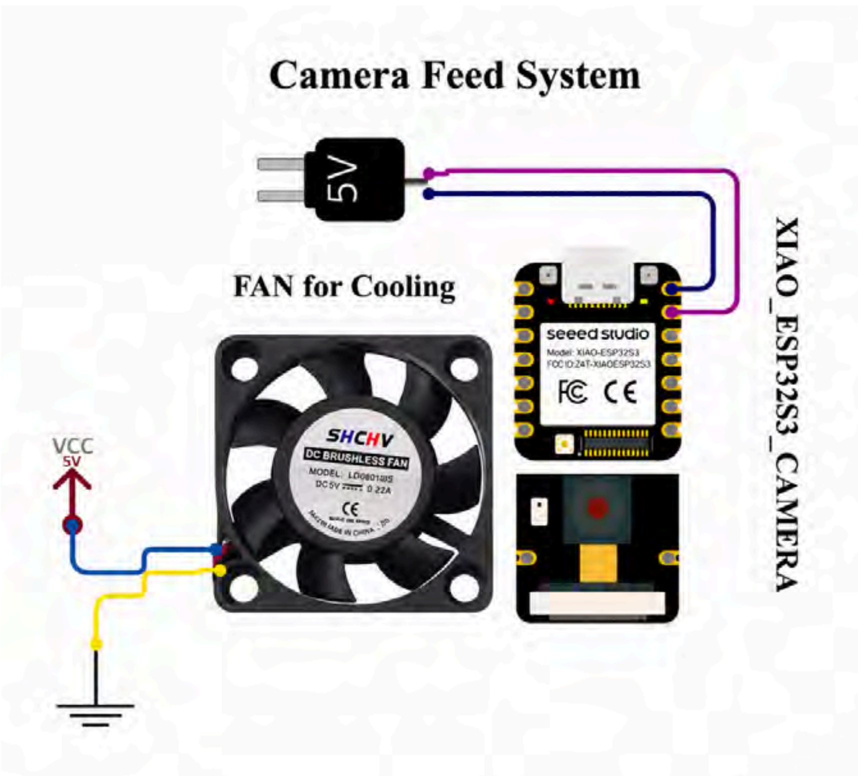


Figure 5.10: Camera Feed System

The circuit represents a wireless video streaming system built around the XIAO ESP32S3 Sense, which hosts an onboard camera module connected via a dedicated parallel interface using 8 data lines (GPIO 15–18 and GPIO 11–12, 14, 48 for pixel data), along with synchronization signals, VSYNC on GPIO 38, HREF on GPIO 47, and PCLK on GPIO 13, and a 24 MHz external clock on GPIO 10, with camera configuration handled over SCCB (I2C-like) on GPIO 40 (SDA) and GPIO 39 (SCL). The camera is configured to capture HVGA resolution frames in JPEG format with a quality factor of 6, storing frames in PSRAM using a double-buffer scheme and always grabbing the latest frame to minimize latency. These frames are served as a continuous MJPEG stream over HTTP on port 81 at the endpoint /stream, using TCP_NODELAY and an enlarged 128 KB send buffer to push frames out with minimal delay, targeting approximately 20 FPS. The ESP32S3 connects to the network using WiFiMulti, automatically selecting the strongest known access point, and monitors connection health every 10 seconds with automatic reconnection. A live FPS counter is printed to the serial monitor every 2 seconds, giving real-time feedback on streaming performance. In the context of the quadruped robot, this circuit acts as the onboard camera head, providing a live first-person video feed to a remote operator over Wi-Fi.

1. Network / Server Config

TABLE 5.7: TABLE OF NETWORK / SERVER CONFIG

Parameter	Value
Stream port	81
Stream URL	`http://<IP>:81/stream`
UDP port	5555
WiFi mode	STA (WiFiMulti)

2. Camera Configuration

TABLE 5.8: TABLE OF CAMERA CONFIGURATION

Parameter	Value
Frame size	HVGA
Pixel format	JPEG
JPEG quality	6
Frame buffers	2
XCLK frequency	24 MHz
Frame buffer location	PSRAM
Grab mode	CAMERA_GRAB_LATEST

5.2.4.3 Air Quality Monitoring Circuit and Environmental Sensor Configuration

This section explains the air quality monitoring circuit built around the ESP32-C6 environmental sensor system. It includes the DHT11 temperature and humidity sensor, gas sensor analog/digital outputs, relay module, toggleable LED indicator, GPIO pin assignments, wiring notes, server configuration, and timing intervals. Overall, this section shows how the robot monitors environmental conditions and sends live air quality data wirelessly for remote inspection and safety awareness.

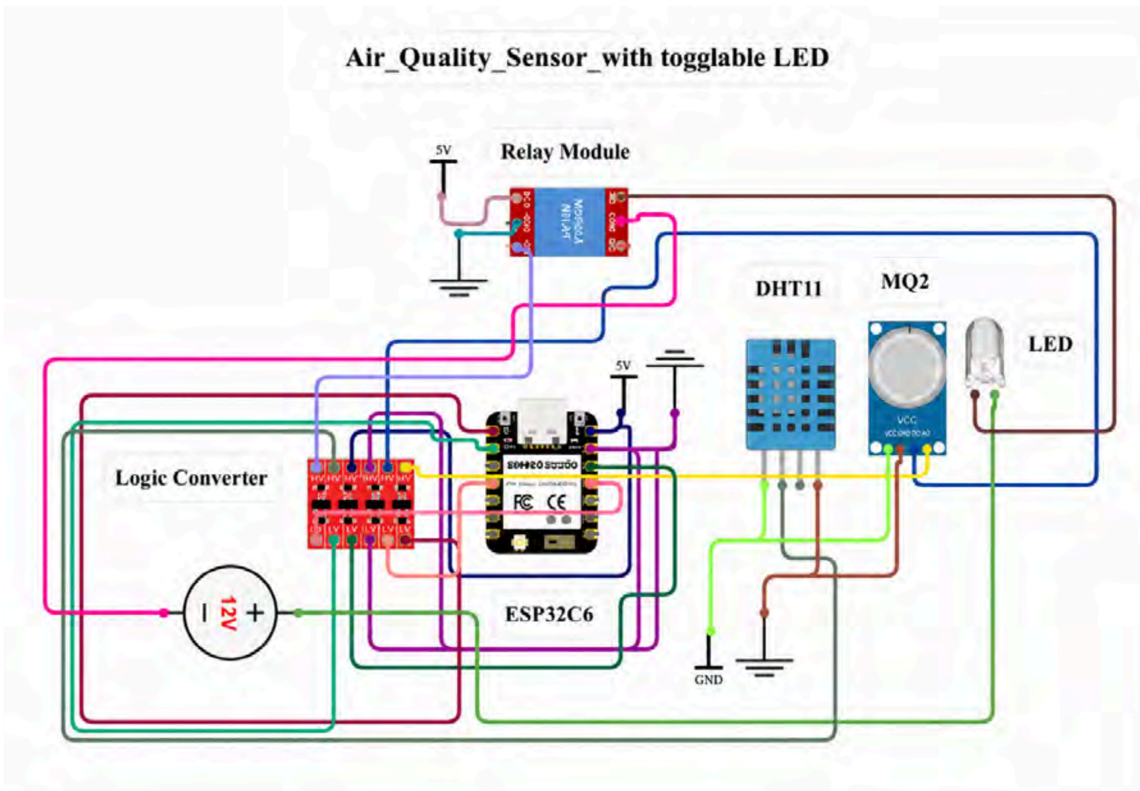


Figure 5.11: Air Quality Monitoring Circuit

The Air Quality Monitoring (AQM) circuit is built around an XIAO ESP32-C6 and continuously reads two environmental sensors, a gas sensor and a DHT11, at regular intervals. The gas sensor outputs an analog voltage on GPIO 0 that is scaled into estimated readings for oxygen (O_2), carbon monoxide (CO), hydrogen sulfide (H_2S), and lower explosive limit (LEL) concentrations, while a digital threshold output on GPIO 3 provides a simple over-limit alert. The DHT11 on pin D2 supplies ambient temperature (in both Celsius and Fahrenheit) and relative humidity every two seconds. All sensor readings are packaged into a JSON payload and streamed at 5 Hz over a TCP server on port 5555, allowing a connected Python client or any TCP listener on the same network to receive live environmental data. An active-low LED on pin D6 serves as a local visual indicator and can be toggled remotely by sending the character 'L' over the TCP connection or via the serial monitor. The ESP32-C6 uses WiFiMulti to connect to the strongest known access point automatically, monitors connection health every 30 seconds, and restarts the TCP server automatically upon reconnection if the link drops.

Pin Connection Table: ESP32-C6 Environmental Sensor

1. GPIO Pin Assignments

TABLE 5.9: TABLE OF GPIO PIN ASSIGNMENTS

Pin	GPIO Role	Device	Notes
D2	Digital Input	DHT11	Temperature & humidity data pin
D6	Digital Output	LED	Active-low; D6 LOW = ON, D6 HIGH = OFF
GPIO 0	Analog Input	Gas Sensor	Analog read (0-4095), mapped to GAS_ANALOG
GPIO 3	Digital Input	Gas Sensor	Digital threshold output, mapped to GAS_DIGITAL

2. Sensor Wiring Notes

TABLE 5.10: TABLE OF SENSOR WIRING NOTES

Device	Connection Detail
DHT11	Data → D2; VCC → 3.3V; GND → GND
Gas Sensor (analog)	AOUT → GPIO 0 (ADC); VCC → 3.3V/5V; GND → GND
Gas Sensor (digital)	DOUT → GPIO 3; threshold set by onboard potentiometer
LED	Anode (+) → 5V; Cathode (−) → D6 (active-low drive)

3. Network / Server Config

TABLE 5.11: TABLE OF NETWORK / SERVER CONFIG

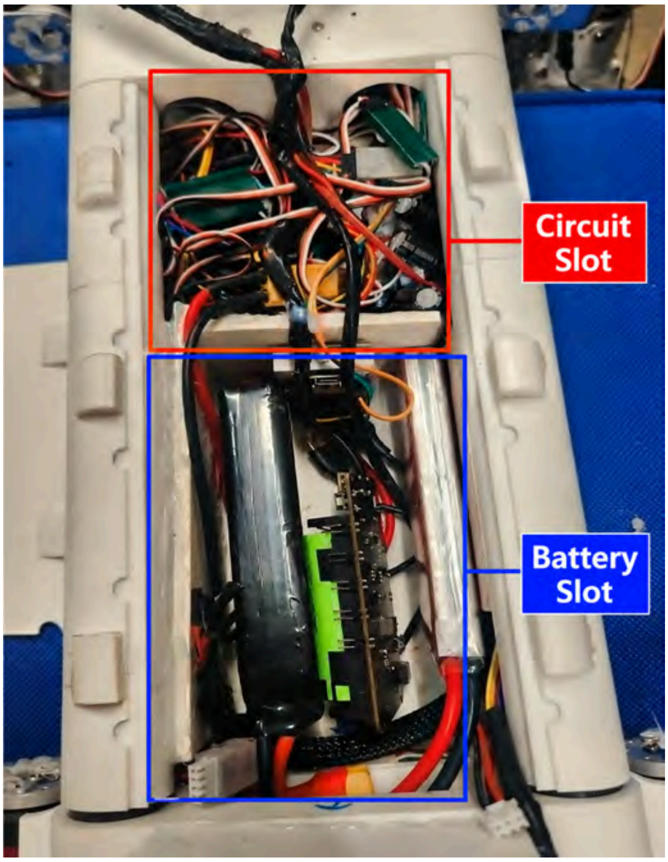
Parameter	Value
TCP server port	5555
WiFi mode	STA (WiFiMulti)
JSON stream interval	Every 200 ms (tied to gas task)
LED toggle command	Send `L` or `l` via TCP or Serial

4. Timing Intervals

TABLE 5.12: TABLE OF TIMING INTERVALS

Task	Interval
Gas sensor read + TCP send	200 ms
DHT11 read	2000 ms
Serial dashboard print	1000 ms
WiFi health check	30000 ms

Robot Belly Compartment with Separate Circuit and Battery Slots:



This image shows the actual circuitry used inside our robot. Specifically, it shows the battery compartment, which contains three batteries, the main ESP32-C6 module, and the main power and signal distribution board located on the upper side. The internal layout is divided into two main sections. The upper section contains the power distribution and signal wiring, while the larger lower compartment houses the batteries, main logic power distribution board, and main logic control boards.The remaining two circuits are positioned separately. The camera circuit is mounted on the gripper claw, while the sensor circuitry is located inside the rear sensor box.

Figure 5.12: Visualization of circuit and battery slot inside the body

5.2.5 Algorithm:

Part I. Overview of the Quadruped Robot Functional Workflow Algorithm

The quadruped robot operates as a synchronized real-time control system. The main controller receives motion commands through Wi-Fi UDP, updates the active operating mode, generates leg trajectories, applies stabilization and safety logic, converts Cartesian foot targets into servo angles using inverse kinematics, and finally sends PWM commands to the servo driver. The robot does not directly walk by manually assigning every servo angle. Instead, it defines foot positions in Cartesian keyframes, interpolates between those keyframes, and uses inverse kinematics to calculate the lateral, shoulder, and elbow angles for each leg at every control step.

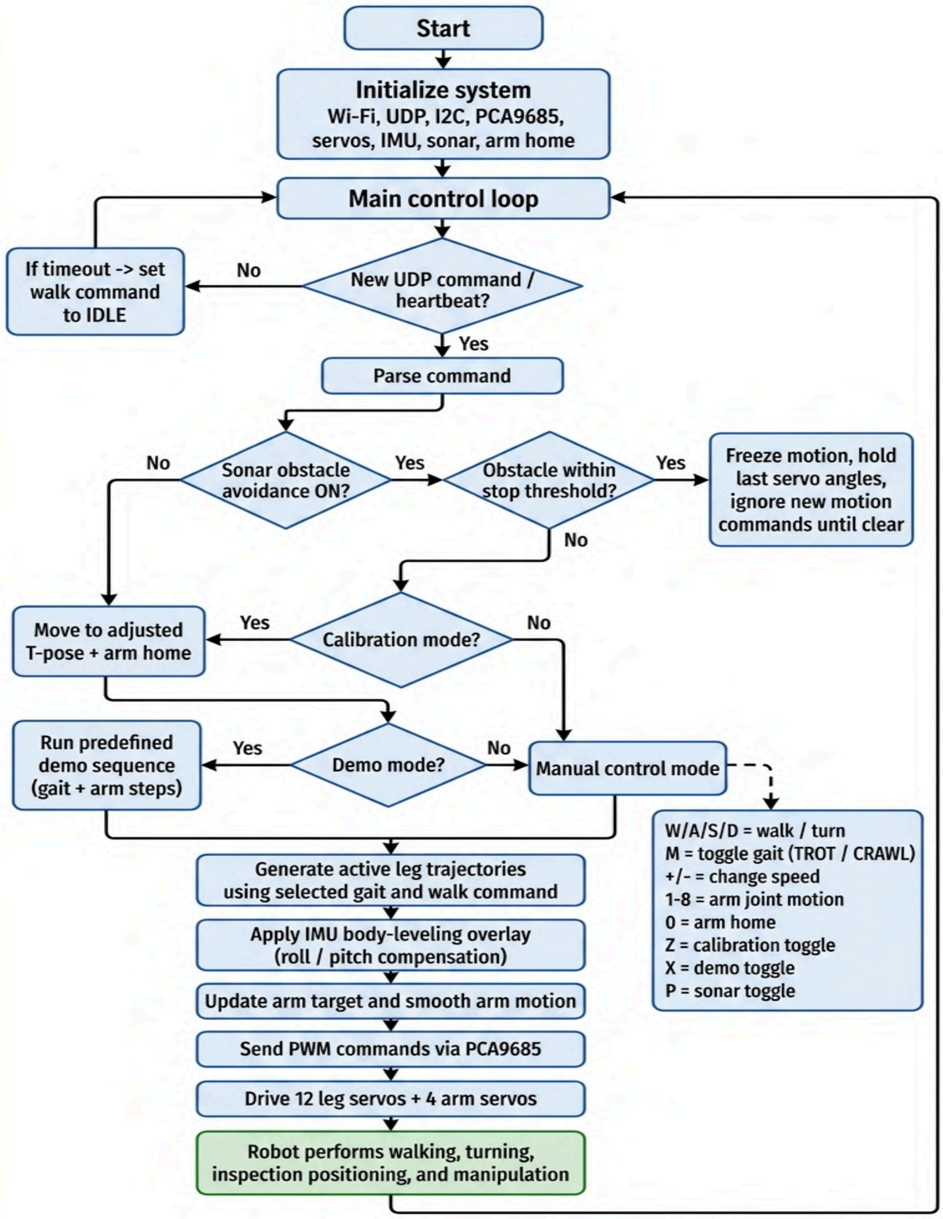


Figure 5.13: Full Working Algorithm

Functionally, the robot follows this loop:

- Initialize communication, servo driver, IMU, sonar, arm home position, and default leg pose.
- Listen for UDP control packets from the Python teleoperation interface.
- Decide the current mode: calibration, idle, manual control, or autonomous/demo.
- Check obstacle distance and decide whether motion should continue, slow down, or freeze.
- Update each leg's active trajectory based on the walking command and gait mode.
- Apply IMU roll/pitch leveling correction to the leg targets.
- Solve inverse kinematics for each leg.
- Convert IK angles into physical servo commands using sign conventions and offsets.
- Update the robotic arm either from manual key input, home transition, or autonomous/demo pose.
- Run camera streaming and environmental monitoring in parallel for operator feedback and inspection.

This produces a modular control architecture where locomotion, stabilization, obstacle safety, arm manipulation, camera vision, and environmental monitoring operate together without being merged into one single blocking process.

Part II. Operating Modes of the Robot

1. Initialization Stage: Initialization is the first stage after power-on. The controller prepares the complete motion system before accepting normal operation. It initializes the sonar pins, starts the I2C bus, starts the PCA9685 servo driver, moves the robot to an adjusted T-pose, initializes the IMU, calibrates the gyroscope bias, sets each leg to the rest target, sets the arm to its home pose, connects to Wi-Fi, and starts listening for UDP commands. The controller then enters the main loop. Algorithmically, this stage is important because it gives the robot a known starting condition. Without this step, the controller would not know whether the legs, arm, IMU, and communication system are in a valid state. The initialization stage therefore acts as the baseline from which calibration, manual control, demo motion, and safety logic can operate reliably.

2. Calibration Mode: Calibration mode places the robot into an adjusted T-pose. In this mode, the controller does not execute walking trajectories. Instead, each leg servo is commanded to its center position plus its own calibration offset. The purpose is to verify that the four legs are mechanically aligned and that the lateral, shoulder, and elbow joints are positioned symmetrically.

Functionally, calibration mode does three things: First, it stops active walking by setting the command state to idle. Second, it returns the arm toward the home pose so that the arm does not interfere with leg calibration. Third, it moves the legs to their calibrated neutral positions using smooth transition logic rather than sudden angle jumps.

The calibration mode is toggled from the operator side using the calibration command. In the provided teleoperation logic, the Z key sends the calibration toggle. In the robot firmware, receiving this command switches calibration mode on or off, disables demo mode, resets walking commands to idle, and starts a pose transition. This mode is not meant for locomotion. It is mainly a checking and preparation mode used before testing, tuning, and operation.

3. Idle / Rest Mode: Idle mode is the safe non-walking mode. In this state, the robot is active, powered, and responsive, but it is not performing forward, backward, or turning motion. The legs are moved to the default inverse-kinematics rest target:

$$REST = [1.8\ 0.0\ 0.0\ ]$$

Because one normalized unit corresponds approximately to 20 cm, this rest position places the foot around 36 cm from the hip/shoulder axis. This creates a slightly bent, stable leg configuration instead of fully straightening the legs. Idle mode is entered when the operator releases the walking keys, when the command times out, when calibration begins, when demo mode stops, or when the sonar safety logic forces the robot to stop after an obstacle is detected.

4. Manual Control Mode: Manual mode is the main teleoperation mode. The Python keyboard program sends UDP packets to the ESP32 controller. Holding W, A, S, or D selects the body motion command:

$$W \rightarrow forward$$
$$S \rightarrow backward$$
$$A \rightarrow turn\ left$$
$$D \rightarrow turn\ right$$

When no walking key is held, the Python program sends IDLE. It also sends heartbeat packets so that the controller knows the operator is still connected. If the controller does not receive a valid command within the timeout period, it automatically changes the manual command to idle to prevent runaway motion. Manual control also includes arm operation. Keys 1 to 8 directly increment or decrement the arm lateral, shoulder, elbow, and claw servos. The arm is not controlled by the leg IK solver; instead, it is controlled by direct servo angle increments. This makes the arm easier to operate during teleoperation because the operator can independently rotate, raise, bend, and open/close the claw. The 0 key sends the arm to the calibrated home pose through a smooth transition.

Manual mode is therefore a human-in-the-loop control system. The operator closes the loop by watching the live video stream and environmental data, then sending the next command based on what the robot sees.

5. Autonomous Mode: The current autonomous behavior in the controller is implemented as a predefined sequence. It is an open loop system that does not use environmental feedback, making it much simpler in terms of execution. Instead of continuously planning a new path from sensor data, the robot follows a time-based sequence of body commands and arm poses. Each autonomous step stores:

{cmd, armLatRelDeg, armShoulderRelDeg, armElbowRelDeg, clawRelDeg, durationSec}

The cmd part defines the body movement, such as forward, turn, or idle. The arm values define the desired arm and claw pose for that step. The duration tells the controller how long that combined behavior should continue. This allows the robot to demonstrate coordinated walking and manipulation without continuous manual input. The Auto mode is toggled using the X command. The firmware only starts demo mode when the robot is not in calibration and is not already moving manually. This prevents unsafe mode switching. When demo mode begins, the controller loads the first demo step, sets the corresponding leg trajectory, and applies the arm target for that step. When demo mode ends, the robot returns to idle and becomes ready for manual control again.

Although this mode is called autonomous, it should be interpreted as scripted autonomy in the current implementation. Full autonomy would require path planning, mapping, and sensor-based decision-making, while the provided code mainly executes predefined motion steps with sonar-based safety interruption. We compare the performance of autonomous and manual modes in the validation testing section of this chapter.

Part III. Main Control Loop Logic

The controller loop is the central algorithmic layer. Every loop cycle performs the same ordered decision process.

First, the sonar state is updated. This checks whether obstacle avoidance is enabled and whether the robot should move normally, slow down, or freeze. Next, Wi-Fi health is checked and incoming UDP packets are read. If a packet is received, it is sanitized and interpreted as a walking command, mode toggle, arm command, speed command, gait toggle, sonar toggle, calibration toggle, or demo toggle. After command handling, the controller checks whether the manual command has timed out. If there is no recent command, and the robot is not in demo, calibration, or sonar freeze, the manual command is set to idle. The controller then advances only when the microstep timing interval has passed. This creates a regular control update rate instead of updating the servos at random intervals.

Inside the active control update, the controller determines the active command. If demo mode is active, the active command comes from the current demo step. If manual mode is active, the active command comes from the latest operator input. If calibration is active, walking is bypassed and the adjusted T-pose is maintained. If sonar freeze is active, leg and arm progression is paused or ignored depending on the state.

This loop structure makes the robot deterministic: commands are received asynchronously, but actual motor updates happen through a timed control loop.

Part IV. Gait Selection and Walking Logic

The walking algorithm is keyframe-based. Each walking command selects a set of foot trajectories. The controller contains separate trajectory tables for forward, backward, turn-left, and turn-right motion. It also contains two gait families:

$$TROT$$
$$CRAWL$$

The gait mode can be toggled by the operator. In trot gait, diagonal leg pairs move together. This gives faster movement but less static stability because only two legs support the body during the most critical part of the gait. In crawl gait, the robot moves more slowly and keeps three legs in support for better stability.

For each leg, a trajectory is defined as a sequence of keyframes:

$$K_i = [x_i, y_i, z_i, scale_i]$$

Here, $\left(z_i\right)$ define the desired foot target, and $scale_i$ defines the relative duration of that segment. The controller linearly interpolates from the current foot position to the next keyframe. At the end of a segment, it advances to the next keyframe and eventually loops back to the beginning. The interpolation is implemented in the leg state machine, where the current foot target is updated smoothly over time before IK is calculated.

Forward and backward motion are produced by using different keyframe sequences. Turning is produced by changing the lateral (z)-coordinate the foot targets so that the left and right leg groups move in opposite effective directions. This produces an in-place tank turn rather than a wide circular turn.

Part V. Inverse Kinematics Logic

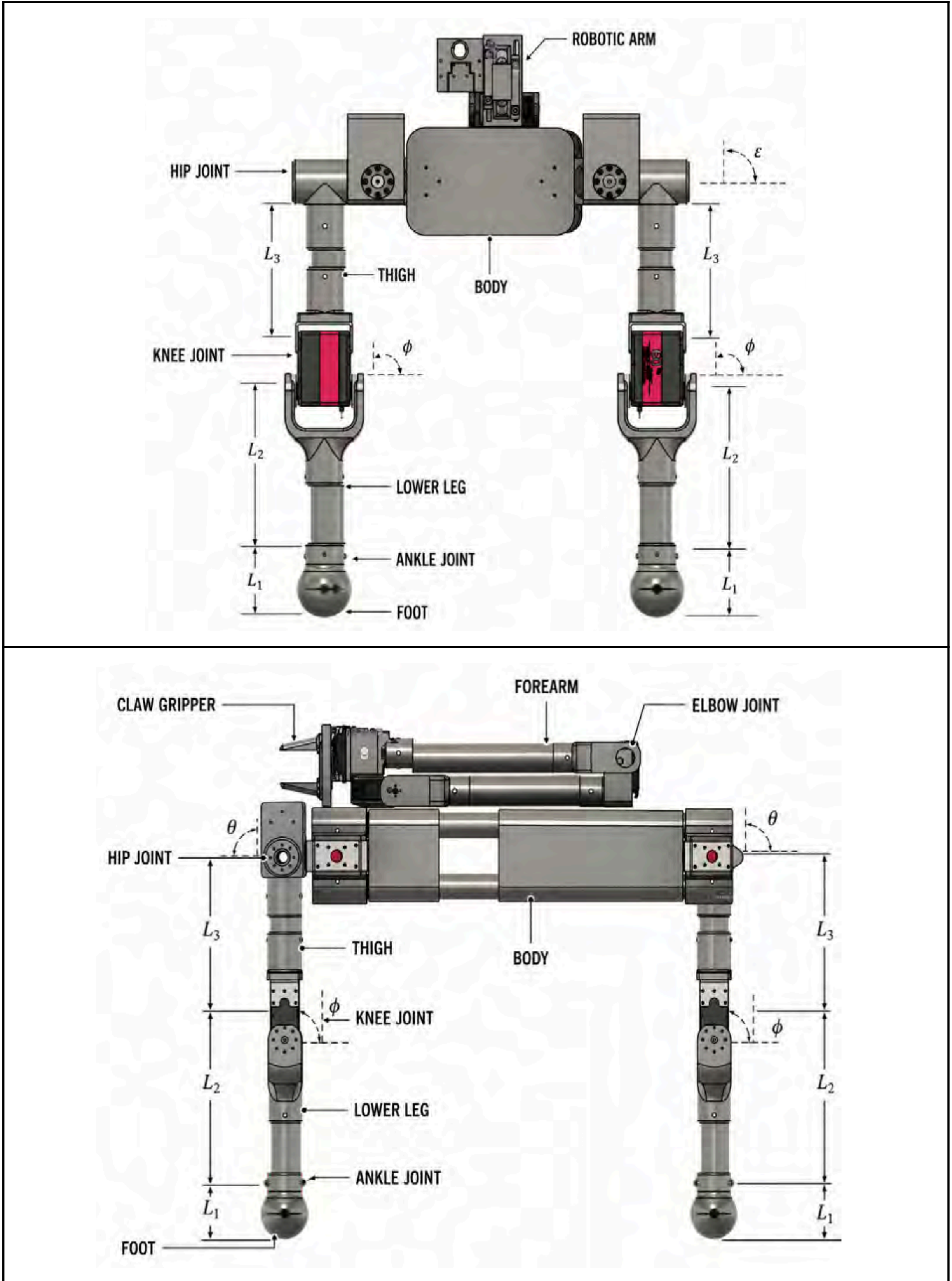


Figure 5.14: Inverse Kinematics Variables Visualization

The inverse kinematics part is the main mathematical layer of the robot's walking algorithm. The controller does not directly specify the lateral, shoulder, and elbow servo angles for every step. Instead, it first defines where each foot should be in a local Cartesian coordinate frame. Then the IK solver converts that foot target into joint angles.

Each leg has three controlled joint angles:

$$\epsilon = lateral\ angle$$
$$\theta = shoulder\ angle$$
$$\phi = elbow\ angle$$

The target foot position is written as:

$$p = [x\ y\ z\ ]$$

This is a 3D target, so the IK should not be described as purely 2D. It is more accurate to describe it as 3D-to-planar two-link inverse kinematics. The solver first computes the lateral angle from the 3D target, then reduces the remaining problem into a planar two-link IK problem for the shoulder and elbow. We explain the process in the following steps.

1. Lateral Angle Calculation: The lateral angle is calculated from the horizontal x- and z -components:

$$\epsilon = atan2(z, x)$$

The radial distance in the horizontal plane is:

$$r = \sqrt{x^2 + z^2}$$

After this step, the original 3D target becomes a 2D target in the radial-vertical plane:

$$(r,\ y)$$

This means the lateral servo first aims the leg plane toward the target, and then the shoulder and elbow solve the two-link reach problem inside that plane.

2. Planar Two-Link IK

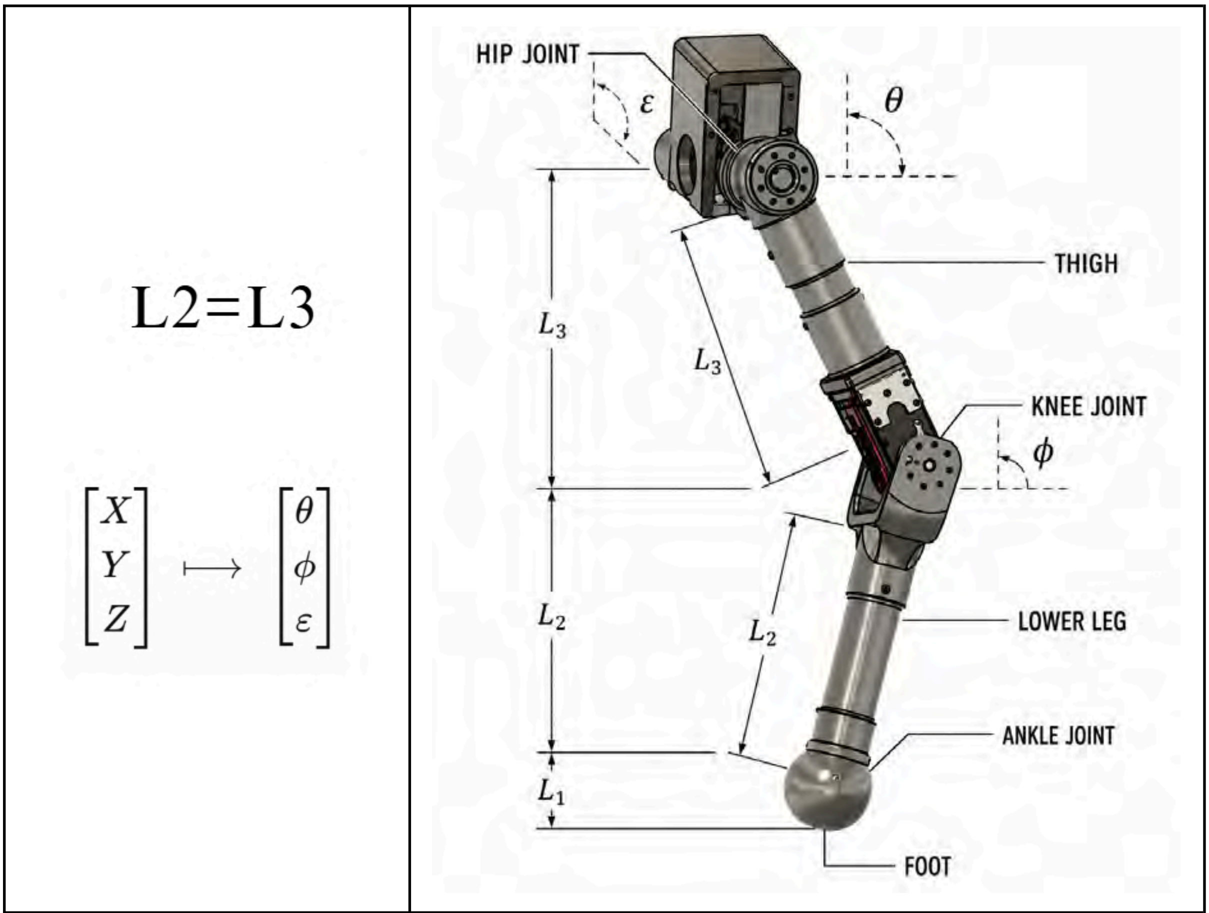


Figure 5.15: Leg Coordinates to Angle Mappings

The physical leg has two equal segments:

$$L_1 = 0.20\, m$$
$$L_2 = 0.20\, m$$

In the controller, the same model is normalized as:

$$R_1 = 1.0, R_2 = 1.0$$

The distance from the shoulder to the target in the 2D IK plane is:

$$d = \sqrt{r^2 + y^2}$$

Because the robot cannot reach targets outside the leg workspace, the distance is clamped:

$$d_c = clamp(d,\ |R_1 - R_2| + \epsilon_{small},\ R_1 + R_2 - \epsilon_{small})$$

The elbow angle is computed using the Law of Cosines:

$$\phi = -\cos^{-1}\left(\frac{d_c^2 - R_1^2 - R_2^2}{2R_1R_2}\right)$$

The negative sign chooses the bent-leg configuration used by the robot.

Then the shoulder angle is calculated using:

$$k_1 = R_1 + R_2 cos(\phi)$$

$$k_2 = R_2 sin(\phi)$$

$$\theta = atan2(y, r) - atan2(k_2, k_1)$$

Therefore, the IK output for each leg is:

$$\Theta = [\epsilon\ \theta\ \phi]$$

The provided controller follows this same logic in the ik3dEqualLinks() and planar2LinkIK(). It calculates the lateral angle using atan2(z,x), computes the radial distance, solves the planar two-link IK, and marks the target as out of reach if the distance had to be clamped.

3. Conversion from IK Angles to Servo Commands: The mathematical IK angles cannot be sent directly to the servos because each leg is mounted in a mirrored orientation. The controller therefore applies leg-specific sign conventions and calibration offsets. The general mapping is:

$$servo_{lat,i} = 135^{\circ} + LAT\_SIGN[i]\epsilon + LAT\_OFFSET[i]$$

$$servo_{shoulder,i} = 135^{\circ} + SHOULDER\_SIGN[i]\theta + SHOULDER\_OFFSET[i]$$

$$servo_{elbow,i} = 135^{\circ} + ELBOW\_SIGN[i]\phi + ELBOW\_OFFSET[i]$$

The center value of 135° corresponds to the middle of the 0–270° leg servo range. The sign terms mirror the motion between front-right, front-left, back-left, and back-right legs. The offset terms correct real mechanical mounting differences. This is what allows the same IK solver and same style of Cartesian foot target to be used for all four legs.

4. IK Role in Different Walking Modes: In forward trot, diagonal legs move in paired phases. The foot targets mostly change in the x- and y-directions, while z remains near zero. This means the lateral angle remains small and most of the motion is handled by the shoulder and elbow.

In crawl gait, the same IK solver is used, but the keyframes are more gradual and the phase distribution is more stable. Since three legs are usually supporting the robot, the crawl mode prioritizes balance over speed. In turning mode, the z-coordinate changes significantly. This creates nonzero lateral angles, so the lateral servo becomes more active. The shoulder and elbow still maintain the radial extension and height of the foot, while the lateral joint helps create the in-place turning motion. Thus, the IK solver is the common mathematical engine behind forward walking, backward walking, crawling, trotting, and turning.

5. Gait Keyframe Matrices: The gait is generated from explicit Cartesian keyframes. Each gait table defines the end-effector positions for each leg. The controller has separate tables for trot forward, trot backward, crawl forward, crawl backward, turn left, and turn right.

The keyframe matrix for one leg is:

$$K = \begin{bmatrix} x_0 & y_0 & z_0 & scale_0 \\ x_1 & y_1 & z_1 & scale_1 \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

At runtime, the active trajectory is selected depending on the active gait mode.

6. Trot Forward Keyframe Matrix for FR Leg: For the front-right leg in forward trot mode, the controller defines:

$$K_{TROT\_FORWARD\_FR} = \begin{bmatrix} 1.8 & 0.1 & 0.0 & 1.0 \\ 1.3 & -0.2 & 0.0 & 1.0 \\ 1.8 & -0.5 & 0.0 & 1.0 \\ 1.8 & -0.2 & 0.0 & 1.0 \end{bmatrix}$$

This four-point cycle produces a compact trot pattern. The leg moves through a repeated sequence of forward placement, lift/recovery, ground contact, and return. The physical interpretation of the foot-contact state is based mainly on the keyframe x-coordinate. When the leg target uses the shortened extension value, nominally x = 1.4, the leg is treated as lifted from the ground and moving through the air during the swing/recovery phase. In some trot-forward entries, the controller uses x = 1.3, which represents an even slightly more retracted version of the same swing action. In contrast, when x = 1.8 or x = 1.7, the leg is interpreted as remaining in ground contact. The value x = 1.8 corresponds to a fully extended support/stance condition, while x = 1.7 represents a slightly shortened support condition used to adjust body balance or gently slant the robot body while another leg is being lifted.

Therefore, the swing/stance rule can be summarized as:

- Swing/recovery: x ≈ 1.4, or the more retracted x = 1.3 value in some trot keyframes.
- Ground contact/stance: x = 1.7 or x = 1.8.

During the swing phase, the leg moves quickly to reposition the foot for the next stride. During the ground-contact phase, the foot remains loaded and drifts more slowly relative to the body, producing propulsion, balance correction, or tank-turn rotation depending on the active gait command.

Results from Python 3D angle and torque simulation:

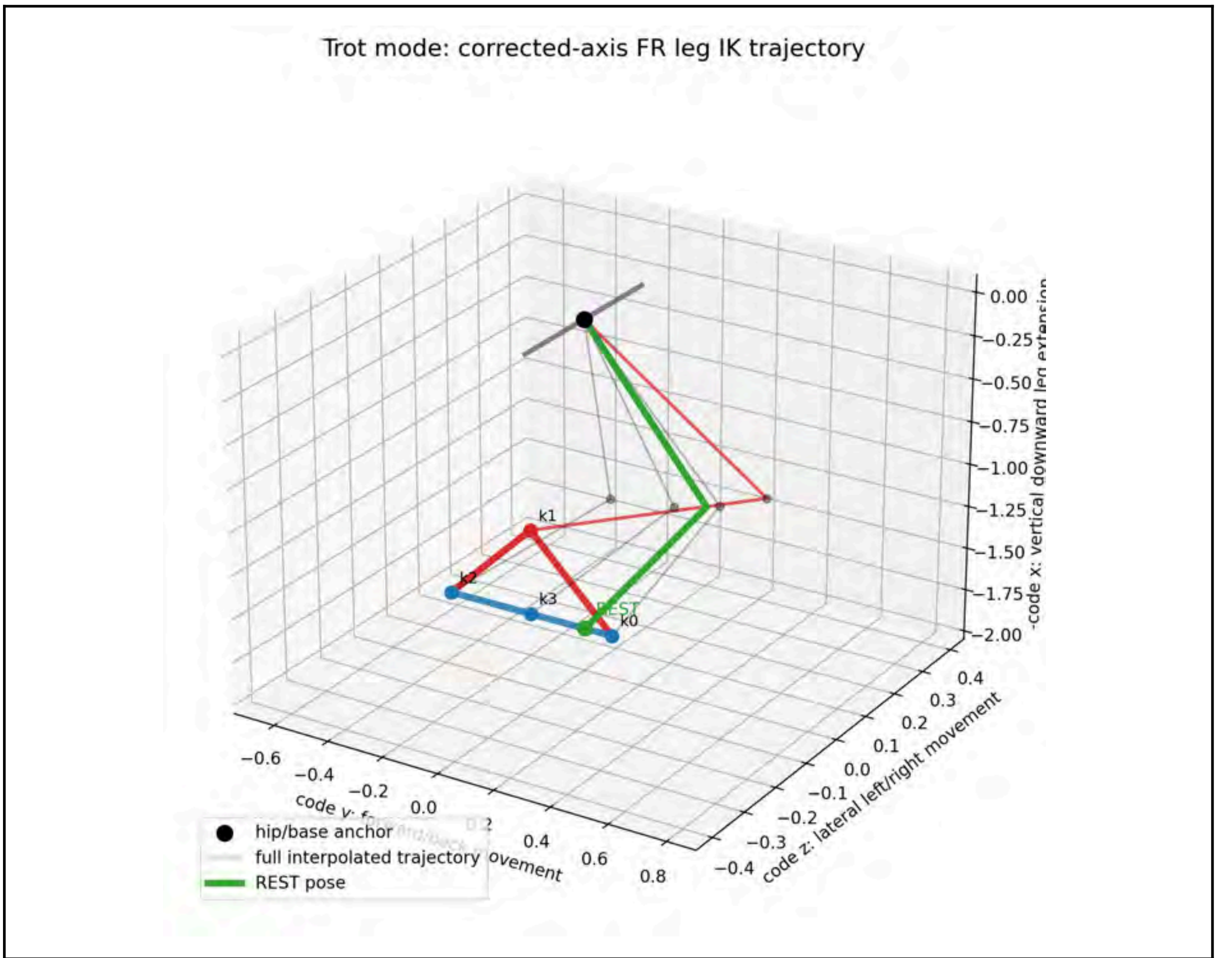


Figure 5.16: Forward Trotting gait in 3D coordinate

7. Crawl Forward Keyframe Matrix for FR Leg: For the front-right leg in forward crawl mode, the controller defines:

$$K_{CRAWL\_FORWARD\_FR} = \begin{bmatrix} 1.4 & -0.2 & 0.0 & 1.0 \\ 1.8 & -0.6 & 0.0 & 1.0 \\ 1.8 & -0.4 & 0.0 & 1.0 \\ 1.8 & -0.2 & 0.0 & 1.0 \\ 1.8 & 0.0 & 0.0 & 1.0 \\ 1.7 & 0.2 & 0.0 & 1.0 \\ 1.7 & 0.4 & 0.0 & 1.0 \\ 1.8 & 0.6 & 0.0 & 1.0 \end{bmatrix}$$

The crawl gait uses more keyframes than trot gait. This makes the leg motion more gradual and produces a slower, more stable walking pattern. In the crawl gait, three legs are assumed to be on the ground at most times while one leg is in the swing/recovery phase. The keyframe with x = 1.4 represents the airborne swing step, where the foot is lifted and repositioned. The keyframes with x = 1.7 and x = 1.8 represent ground-contact support states. The x = 1.8 points are the more fully extended stance points, while the x = 1.7 points are slightly shortened stance points used for balance adjustment and body slanting during the gait cycle.

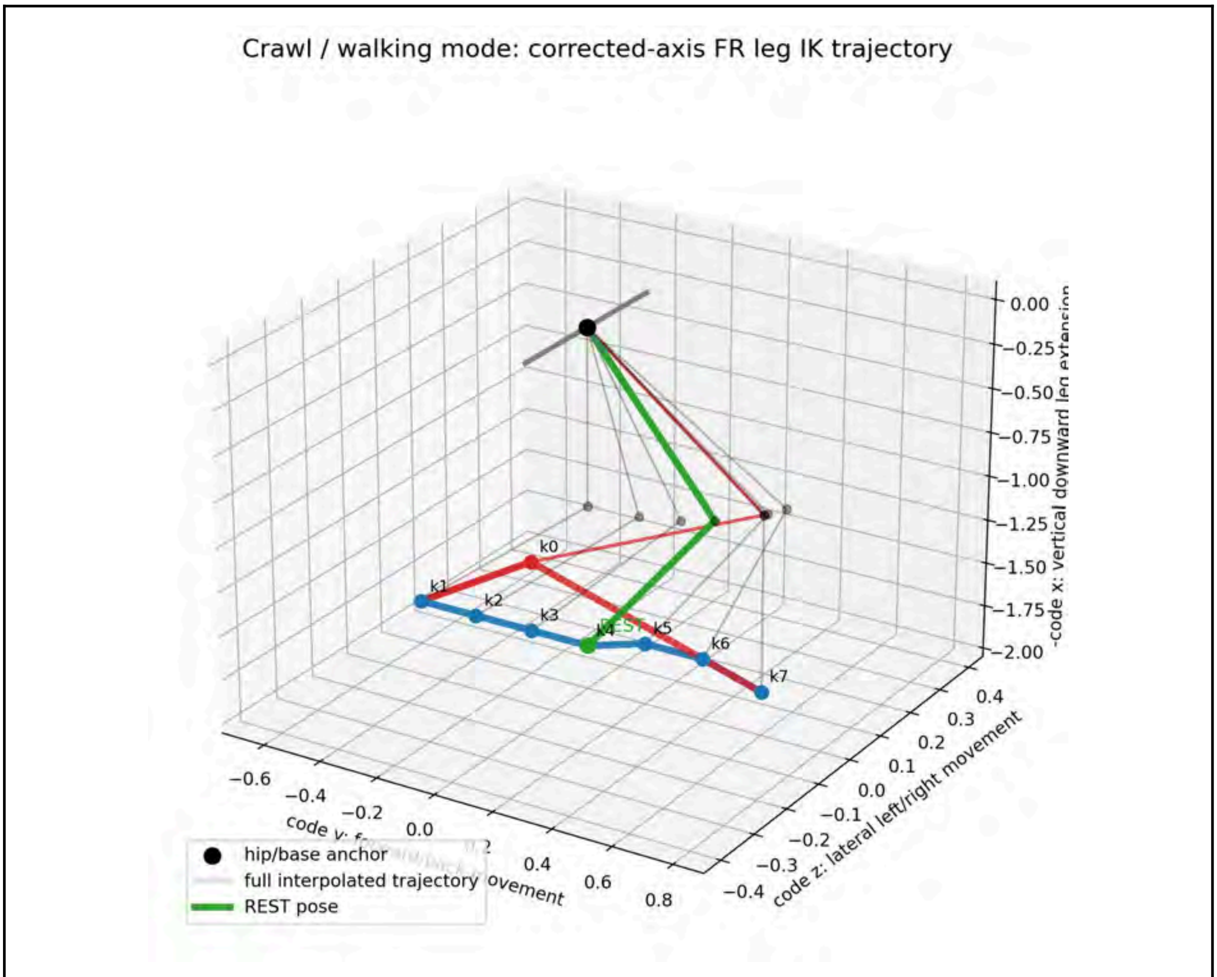


Figure 5.17: Forward Crawling gait in 3D coordinate

8. Turn-Left Keyframe Matrix for FR Leg: For turning left, the FR leg follows a lateral trajectory in the z-direction:

$$K_{TROT\_TURN\_LEFT\_FR} = \begin{bmatrix} 1.7 & 0.0 & 0.3 & 1.0 \\ 1.7 & 0.0 & 0.2 & 1.0 \\ 1.8 & 0.0 & 0.1 & 1.0 \\ 1.8 & 0.0 & 0.0 & 1.0 \\ 1.8 & 0.0 & -0.1 & 1.0 \\ 1.8 & 0.0 & -0.2 & 1.0 \\ 1.8 & 0.0 & -0.3 & 1.0 \\ 1.4 & 0.0 & -0.1 & 1.0 \end{bmatrix}$$

The turn-right matrix is the mirrored version of the turn-left motion. During tank-style turning, legs on one side of the robot move in the opposite effective direction from the legs on the other side. This allows the robot to rotate about its own center rather than following a large turning radius. The same foot-contact interpretation applies during turning: keyframes with x = 1.4 correspond to lifted swing/repositioning motion, while keyframes with x = 1.7 or x = 1.8 remain in ground contact. In the turning trajectories, the changing z-coordinate provides the lateral foot placement required for yaw rotation, while the x-coordinate determines whether the foot is in the air or supporting the robot body.

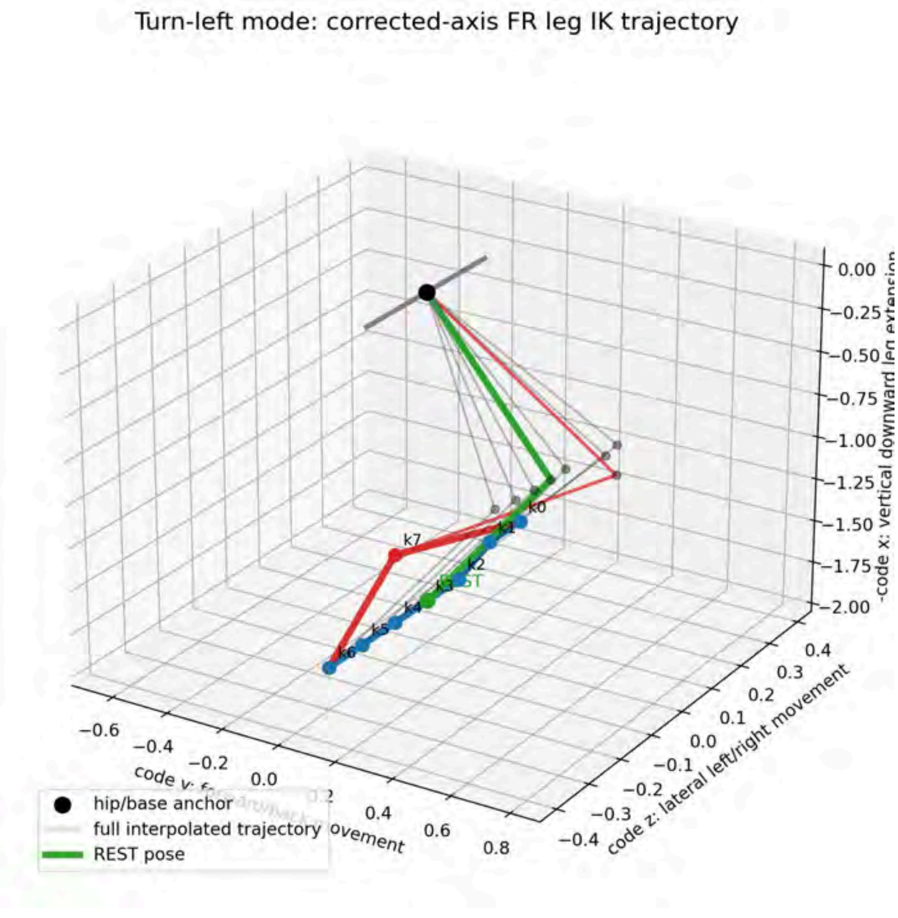


Figure 5.18: Turn-Left gait in 3D coordinate

9. Timing and Interpolation: The controller defines:

$$INTERP\_POINTS\_BASE = 48$$

$$MICROSTEP\_MS = 10\ ms$$

Therefore, the base segment duration is:

$$T_{base} = 48\times10\ ms = 480\ ms = 0.48\ s$$

The default gait speed multiplier is:

$$GAIT\_SPEED = 2.50$$

Thus, the effective segment duration is approximately:

$$T_{seg} = \frac{0.48}{2.50} = 0.192\ s$$

The controller also allows the speed to be increased or decreased between minimum and maximum values:

$$0.25\leq GAIT\_SPEED\leq 3.00$$

The interpolated foot motion therefore depends on both the keyframe scale value and the active command speed multiplier.

Part VI. IMU Body-Leveling Logic

The IMU leveling logic adds a stabilization overlay on top of the gait. The accelerometer and gyroscope estimate roll and pitch. A complementary filter combines the short-term responsiveness of the gyroscope with the long-term stability of the accelerometer. The result is a smoother estimate of body tilt.

The leveling algorithm does not replace the gait. Instead, it modifies the leg target before IK is solved. For each leg, a small vertical correction is added to the y-coordinate:

$$y_{corrected} = y + dy_{applied}$$

The correction depends on the measured roll and pitch, the leg's front/back position, and the leg's left/right position. For example, if the robot leans to one side, the legs on that side can be extended slightly while the opposite side is adjusted in the opposite direction. If the robot pitches forward or backward, the front and rear legs receive different corrections.

The controller also limits the maximum correction, applies deadband to ignore small noise, and uses smoothing so that the compensation does not jerk the servos. The IMU leveling overlay is applied to the legs only, not to the arm.

Functionally, this logic improves walking stability on uneven or slightly tilted surfaces. It does not make the robot fully dynamically balanced, but it helps the body remain more level while the predefined gait is running.

Part VII. Obstacle Avoidance and Fuzzy Freeze Logic

The robot uses the front ultrasonic sensor as a safety layer. The code defines two obstacle thresholds:

$$SONAR\_SLOW\_START = 60\ cm$$
$$SONAR\_STOP = 30\ cm$$

The obstacle response works like a simple fuzzy or graded logic system. Instead of only using a hard stop, the robot creates a motion scale from the measured distance:

$$s(d) = \begin{cases} 1, & d \geq 60 \text{ cm} \\ \dfrac{d - 30}{60 - 30}, & 30 < d < 60 \text{ cm} \\ 0, & d \leq 30 \text{ cm} \end{cases}$$

If the obstacle is farther than 60 cm, the robot moves normally. If the obstacle is between 60 cm and 30 cm, the motion is reduced gradually. If the obstacle is at or below 30 cm, the motion scale becomes zero and the robot freezes. The sonar function measures distance using the trigger/echo pulse timing and then computes this motion scale.

The freeze state is a safety override. When freeze is active, the robot holds its last commanded servo angles instead of continuing to walk into the obstacle. Demo time is paused, so the demo resumes from the same point after the obstacle is cleared. UDP commands received during freeze are ignored rather than queued. This prevents the robot from suddenly executing old commands after the obstacle disappears. The sonar avoidance system can also be enabled or disabled by command. In the supplied teleoperation mapping, the P key toggles sonar avoidance. If sonar avoidance is disabled, the obstacle state is cleared, the motion scale returns to 1, and any active freeze state is released. If it is enabled, the controller begins sampling the sonar periodically and applies the slow/freeze behavior. This gives the operator two layers of control: normal teleoperation and an optional obstacle safety shield. The freeze logic has three important states.

In the normal free state, sonar avoidance is either disabled or the measured distance is safely above the slow-start threshold. Commands are accepted normally, gait timers run normally, and the arm can be controlled normally. In the slow zone, sonar avoidance is enabled and the obstacle is closer than 60 cm but farther than 30 cm. The robot does not fully stop, but the motion scale is reduced. This creates a cautious approach behavior. In the freeze state, the obstacle is at or below the stop threshold. The robot holds its current pose and ignores movement commands until the obstacle clears. After unfreezing, the robot does not execute previously ignored packets. In manual mode, the robot waits for fresh operator input. This prevents unsafe delayed motion.

This logic is important because quadruped walking has momentum and multi-servo coordination. A sudden obstacle should not be handled by simply continuing the current gait cycle. Holding the last safe servo angles is a safer response than trying to complete the current step.

Part VIII. Robotic Arm Functional Logic

The arm is controlled separately from the leg IK system. The legs use Cartesian foot targets and inverse kinematics, but the arm uses direct relative servo angle commands. This is simpler and more practical for manual manipulation.

The arm has four controlled functions:

$$lateral\ rotation$$
$$shoulder\ movement$$
$$elbow\ movement$$
$$claw\ opening/closing$$

The arm is primarily controlled manually through UDP keyboard commands. Unlike the leg subsystem, where a gait command selects a full cyclic foot trajectory, the arm is driven directly joint-by-joint. Each key command changes one arm's degree of freedom by a small angular step.
The controller defines the manual arm step size as:

$$ARM\_STEP\_DEG = 2^{\circ}$$

The manual key mapping implemented in the controller is:

- Key 1: decrease arm lateral angle
- Key 2: increase arm lateral angle
- Key 3: decrease arm shoulder angle
- Key 4: increase arm shoulder angle
- Key 5: decrease arm elbow angle
- Key 6: increase arm elbow angle
- Key 7: decrease claw angle, closing or opening depending on the mechanical orientation
- Key 8: increase claw angle, opening or closing depending on the mechanical orientation
- Key 0: return the arm to the calibrated home position

The supplied controller code uses keys 1 through 8 and 0 for the arm. No active arm command is assigned to key 9 in the current code. When a manual command is received, the controller first copies the current active arm state into the manual state. Then it increments or decrements the selected joint angle by 2 degrees. After the update, the controller clamps the result to the allowed angle range. This prevents the servo from being commanded beyond the physical limits of the arm. This direct-drive method is useful during teleoperation because the operator can independently adjust the arm base, shoulder, elbow, and claw. For example, the operator can first rotate the arm toward an object using the lateral servo, then lower the shoulder, then bend the elbow, and finally close the claw to grasp the object.

Arm Home Position and Smooth Retraction: Pressing key 0 commands the arm to return to its calibrated home pose. This is important because the arm should be tucked back toward its safe initial position before or during walking so that it does not swing freely, collide with the environment, or destabilize the quadruped body. The home pose is defined by all positions as zeros "0". The controller does not snap the arm instantly to this pose. Instead, it performs a smooth transition using an interpolation routine. The transition duration is:

$$POSE\_INTERP\_DURATION\_SEC = 1.20\ s$$

The interpolation uses a smooth ease-in/ease-out function:

$$q(t) = t^2(3 - 2t), for\ 0 \le t \le 1$$

The interpolated angle for each arm joint can be written as:

$$angle(t) = angle\_start + q(t)(angle\_target - angle\_start)$$

This prevents sudden jumps in servo angle, reduces shock loading, and avoids rapid uncontrolled servo motion. The interpolated arm angle is computed between the starting pose and the target home pose. Therefore, the arm gradually retracts to the initial tucked configuration instead of moving abruptly.
This smooth retraction is especially important for the physical robot because sudden servo motion can create large inertial loads, cause mechanical vibration, or disturb the balance of the quadruped while it is standing or walking.

Part IX. Auxiliary Environmental Sensor Logic

The environmental monitoring system operates in parallel with locomotion. It does not directly drive the leg IK loop, but it provides inspection data to the operator or GUI.

The air quality monitoring circuit reads a gas sensor and a DHT11 temperature/humidity sensor. The gas sensor provides an analog value that is mapped into estimated gas-related readings such as oxygen, carbon monoxide, hydrogen sulfide, and combustible gas level. It also provides a digital threshold output for simple over-limit detection. The DHT11 provides temperature and humidity readings. These values are packaged into JSON and streamed over TCP so that the operator interface can display the surrounding environmental condition in real time.

This subsystem is important because the robot is intended for inspection. While the main controller handles movement, the auxiliary sensor circuit continuously reports environmental status. As a result, the operator can remotely observe temperature, humidity, gas conditions, and warning states while the robot is moving or standing.

Part X. Camera Streaming and Vision Logic

The camera subsystem provides real-time visual feedback. The ESP32S3 Sense camera captures JPEG frames, stores them in PSRAM using a double-buffer scheme, and streams the latest frame as MJPEG over HTTP. The use of “grab latest” behavior reduces latency because the operator receives the newest available frame instead of waiting for old buffered frames. The camera stream functions as the robot’s first-person visual feed.

This video stream supports manual driving, inspection, and vision-based detection. In manual mode, the operator uses the live feed to decide where to walk, when to turn, and when to move the arm. In inspection mode, the video frames can be processed by the object detection pipeline.

The YOLO-based detection system works as a frame-by-frame perception loop. It initializes the model and camera source, captures a frame, runs inference, filters low-confidence detections, draws bounding boxes and labels, computes performance information such as FPS or latency, and displays the annotated result. If the user changes the confidence threshold, the detection sensitivity can be adjusted during operation.

For crack or microcrack detection, the same pipeline is adapted to detect surface defects from the robot camera feed. The trained YOLO model identifies visual fault regions and overlays bounding boxes on the live image. The report also describes an active learning interface, where user feedback can be used to refine the model over time.

Functionally, the camera and YOLO system do not replace the locomotion controller. Instead, they provide perception and inspection output. The detection result can be used by the operator to stop the robot, inspect an area more closely, or trigger a higher-level emergency response. If future autonomy is added, these detections could become direct inputs to path planning or fault-localization logic.

Part XI. Crack Detection with Yolo V8 Model

We already discussed the training of this YOLO model in Chapter 4, as we had completed our machine learning software preparation during FYDP-D when we did the simulations. In this section, we discuss the integration of our existing model into the Graphical User Interface (GUI) that we developed to control the robot.

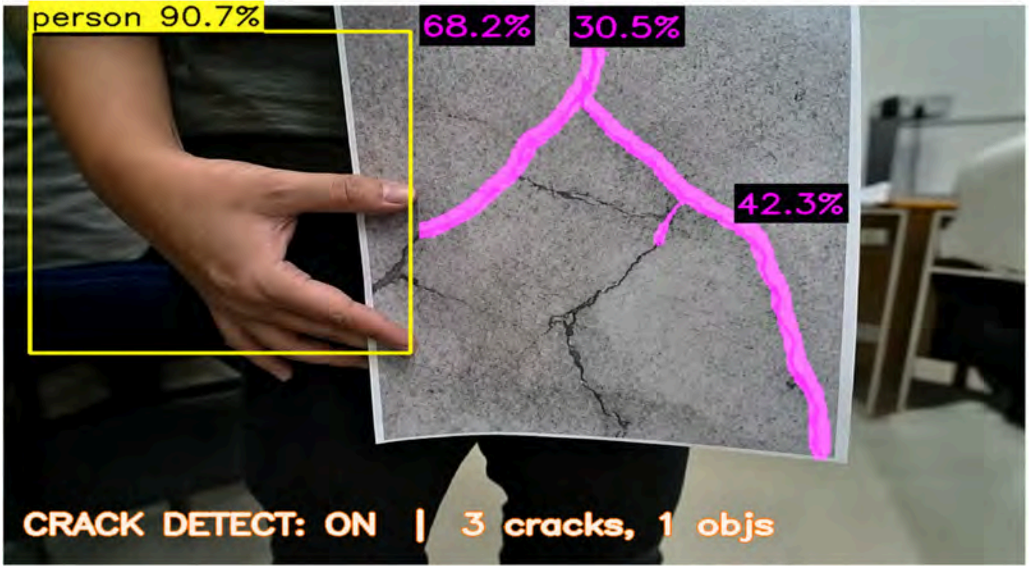


Figure 5.19: Crack Detection with Yolo V8 Model

Graphical User Interface or GUI:

Figure 5.20: Complete overview of the GUI

Connection Manager:

Figure 5.21: Connection Manager

The Connection Manager section is used to establish and monitor communication between the GUI and the robotic dog system. It displays the connection status of the main robot controller, camera stream, and air quality monitoring module using their respective IP addresses and port numbers. This panel allows the user to confirm whether the robot, ESP32 camera, and AQM module are online and actively connected. The camera source can also be selected between laptop camera and ESP32 camera, while the Disconnect All button provides a quick way to terminate all active connections safely.

Gamepad State and Manual Control Panel:

The Gamepad State panel displays the real-time input status of the Logitech F710 wireless gamepad used to control the robotic dog. It shows the condition of different buttons, triggers, D-pad, and left/right joystick positions, allowing the operator to confirm whether the controller is properly connected and responding.

Figure 5.22: Gamepad State and Manual Control Panel

System Log and CLI Debug Terminal

The System Log panel displays real-time status messages from the robotic dog control system. It shows whether the main robot controller, camera, gamepad, air quality module, and crack detection model are successfully connected and running. This section is useful for debugging and monitoring because it allows the operator to quickly identify connection errors, sensor issues, camera streaming problems, or model-loading status.

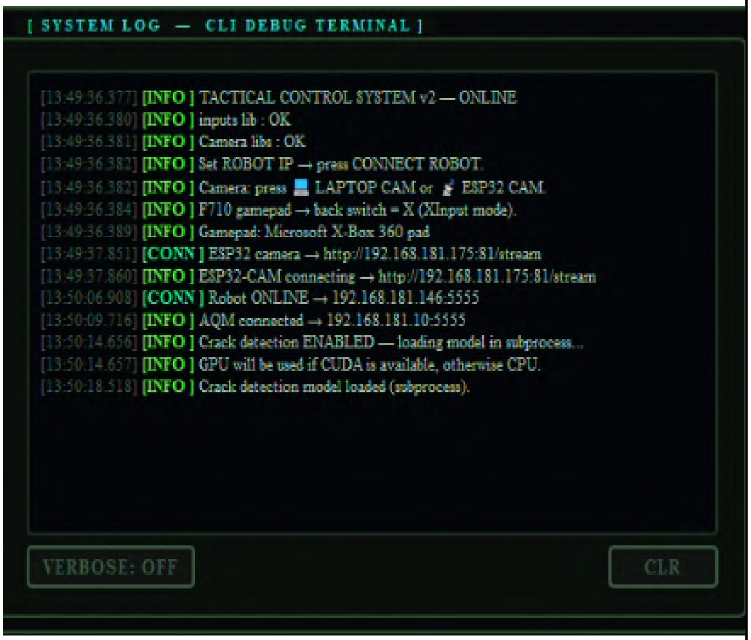


Figure 5.23: System Log and CLI Debug Terminal

Air Quality and Climate Monitoring Panel

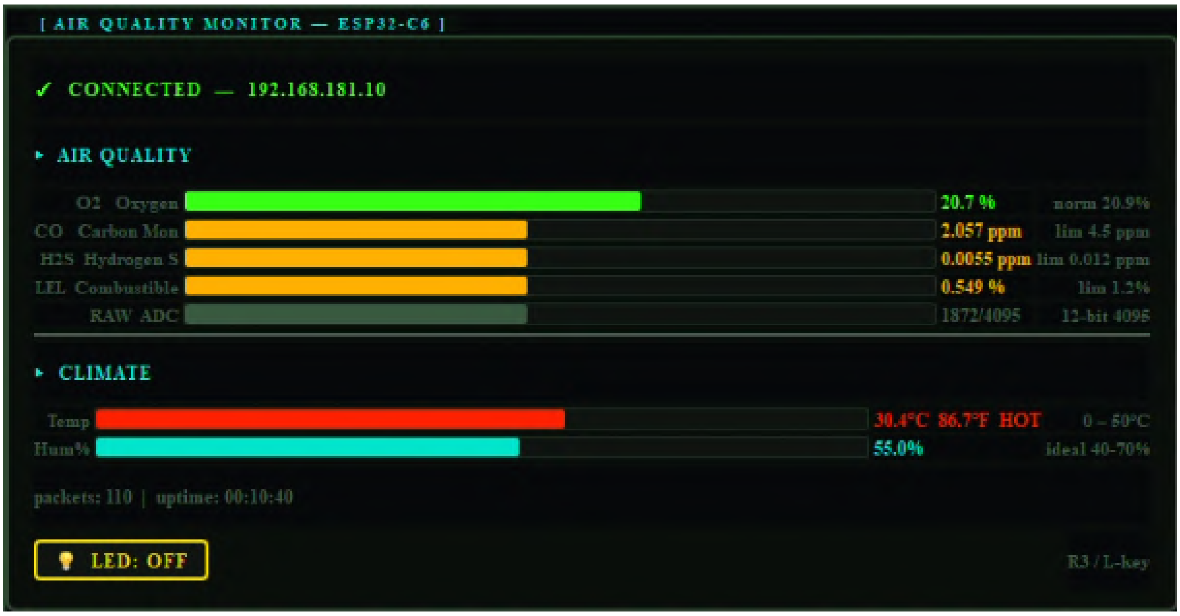


Figure 5.24: Air Quality and Climate Monitoring Panel

The Air Quality Monitor(AQM) panel displays real-time environmental data collected from the sensor module connected to the robotic dog. In the Air Quality section, the system displays different gas-related readings, including oxygen level, carbon monoxide, hydrogen sulphide, combustible gas level, and raw ADC value from the gas sensor. These readings help the operator monitor the surrounding air condition and detect potentially unsafe gas levels during robot operation. The Climate section shows the environmental temperature and humidity. The temperature reading is displayed in both °C and °F, while humidity is shown as a percentage. In this example, the temperature is marked as HOT, indicating that the detected temperature is above the normal range. The humidity value is also shown with its ideal operating range for comparison.

5.3 Evaluate the solution to meet desired need

The Physical Dimensions:

The physical dimensions of the robotic dog were manually measured using a measuring tape.

TABLE 5.13: TABLE OF THE PHYSICAL DIMENSIONS OF THE ROBOT

Measurement Parameter	Recorded Measurement
Body vertical length in T-pose	52 cm
Inner body length	44 cm
Inner body width	14.5 cm
Shoulder-to-shoulder width	35.5 cm
Front shoulder to back shoulder length	58 cm
Left leg edge to right leg edge width	38 cm
Front leg edge to back leg edge length	54 cm
Leg height (shoulder to end effector)	41 cm
Arm horizontal extension	58 cm

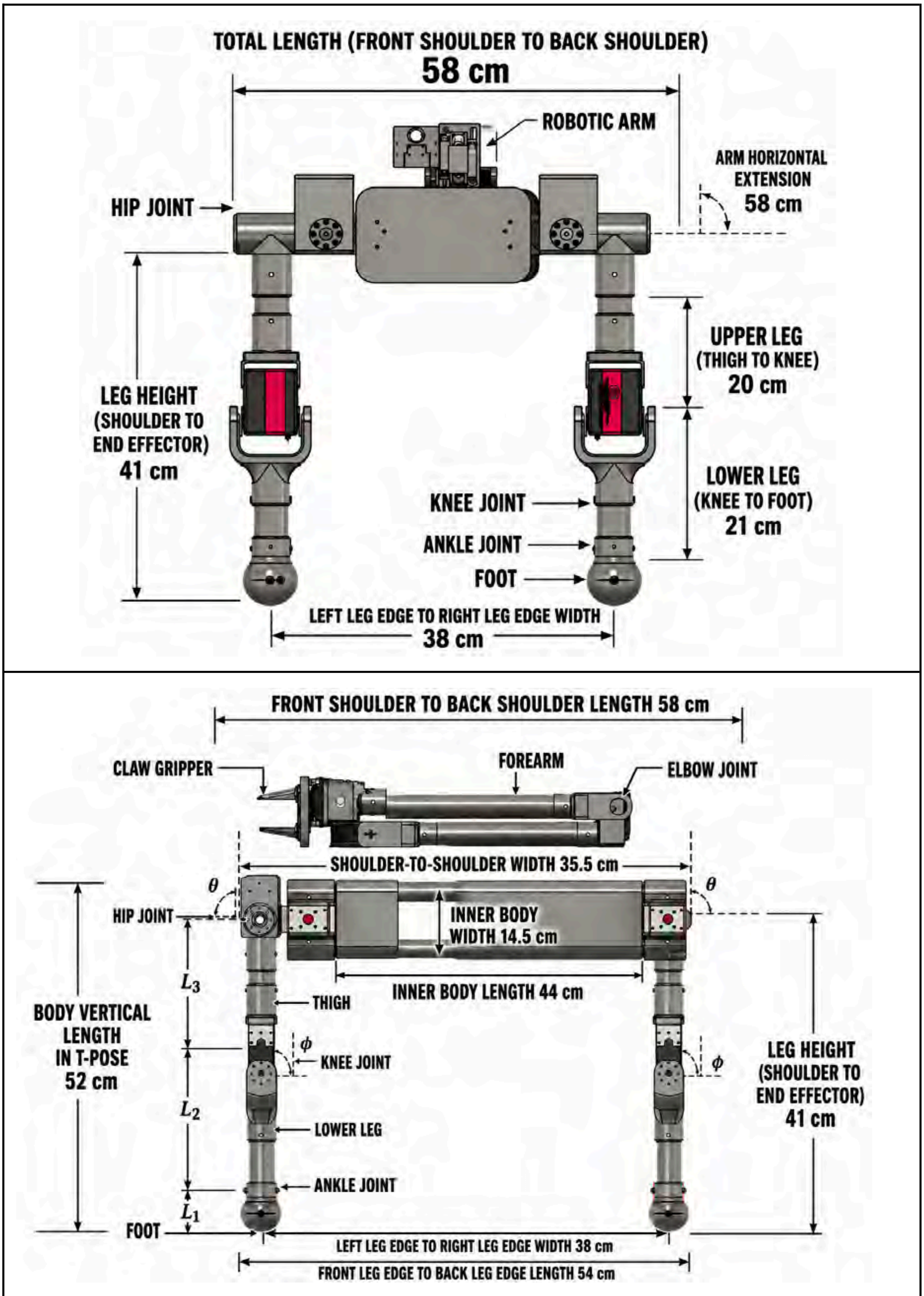


Figure 5.25: The Physical Dimensions of The Robot

Weight Measurement:

The robotic dog was weighed using the weight measurement machine at the BRACU Gym. The measured weight was approximately 7.30 kg.

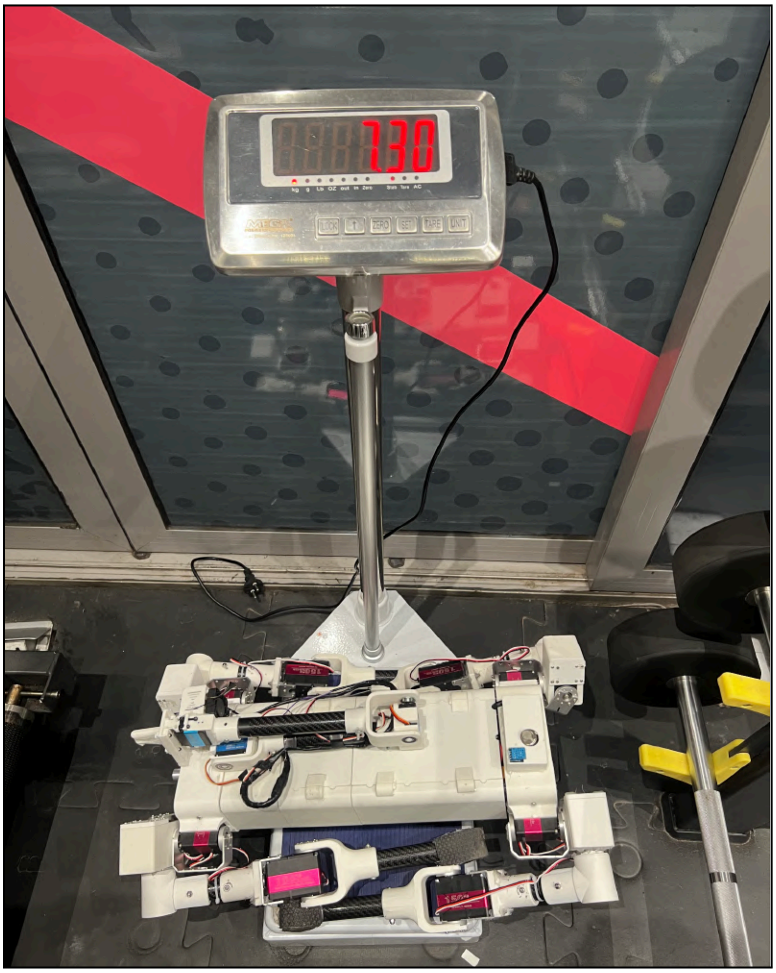


Figure 5.26: Weight Measurement

Payload:

TABLE 5.14: TABLE OF THE PAYLOAD CAPACITY

Maximum Payload Capacity	5 kg
Test Surface	Concrete floor
Average Speed with Payload	0.20 m/s

5.3.1 Locomotion Test on Different Arenas:

The locomotion performance of the robotic dog was tested on three different surface conditions: concrete, grass/turf, and tiles. For each surface, the robot was allowed to travel a fixed distance, and the travel time, walking speed, and path deviation from a straight line were recorded. This test was conducted to evaluate how different ground surfaces affect the robot's movement stability, speed, and straight-line tracking accuracy.

1. **Concrete**

TABLE 5.15: TABLE OF THE WALKING TEST RESULTS ON CONCRETE

Trial	Surface	Distance (m)	Travel Time (sec)	Speed (m/s)	Path Deviation from Straight Line (cm)
1	concrete	5m	22.85	0.2188	-12cm
2	concrete	5m	23.96	0.2087	-13cm
3	concrete	5m	23.87	0.2095	+27cm
4	concrete	5m	23.92	0.2090	+30cm
5	concrete	5m	23.81	0.2100	-26cm

Picture of Testing:

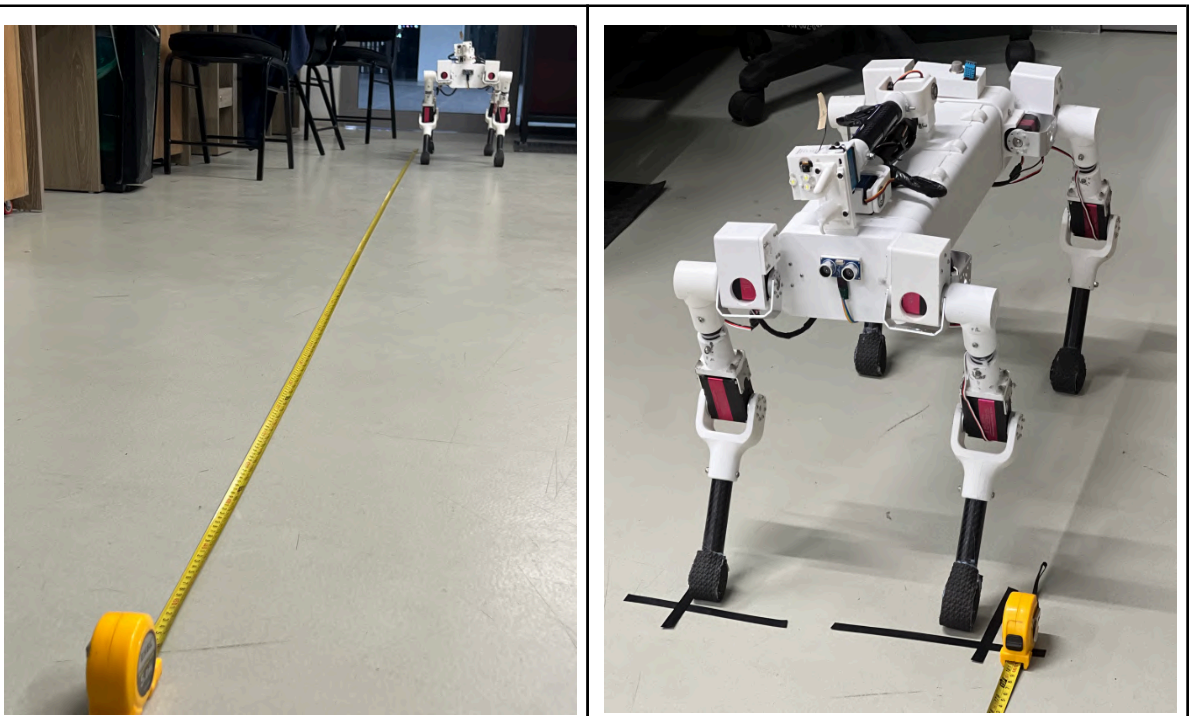


Figure 5.27: Picture of Walking on Concrete

2. **Grass/Turf**

TABLE 5.16: TABLE OF THE WALKING TEST RESULTS ON GRASS/TURF

Trial	Surface	Distance (m)	Travel Time (sec)	Speed (m/s)	Path Deviation from Straight Line (cm)
1	grass/turf	1.5m	17.05	0.0880	-12cm
2	grass/turf	1.5m	17.43	0.0861	-13cm
3	grass/turf	1.5m	16.73	0.0897	+27cm
4	grass/turf	1.5m	17.21	0.0872	+30cm
5	grass/turf	1.5m	17.69	0.0848	-26cm

Picture of Testing:

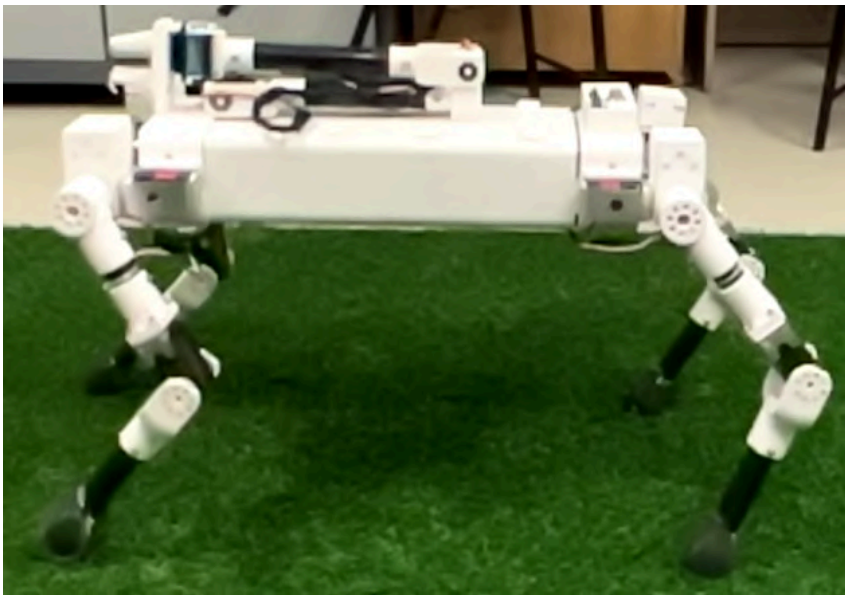


Figure 5.28: Picture of Walking on Grass/Turf

3. Tiles

TABLE 5.17: TABLE OF THE WALKING TEST RESULTS ON TILES

Trial	Surface	Distance (m)	Travel Time (sec)	Speed (m/s)	Path Deviation from Straight Line (cm)
1	Tiles	3	19.37	0.1549	-10
2	Tiles	3	18.81	0.1595	+35
3	Tiles	3	19.67	0.1525	+140
4	Tiles	3	18.83	0.1593	+44
5	Tiles	3	21.27	0.1410	+21

Picture of Testing:

Figure 5.29: Picture of Walking on Tiles

5.3.2 Turning Performance Test on different Arenas

Turning performance was measured on the same surfaces by recording the time required to complete 90°, 180°, 270°, and 360° turns in both right and left directions. The minimum turning radius was also recorded during the 360° turning test. The purpose of this test was to assess the robot's maneuverability and turning control in different arena conditions. Overall, these tests helped to understand the robotic dog's movement capability, surface adaptability, and control performance.

1. **Concrete**

TABLE 5.18: TABLE OF THE TURNING TEST RESULTS ON CONCRETE

Turning Direction	90° Turn Time (s)	180° Turn Time (s)	270° Turn Time (s)	360° Turn Time (s)	Min Turning Radius (cm) (360 deg)
Right	12.78	22.89	34.12	45.60	82cm
Right	14.65	24.34	37.23	46.25	80cm
Right	12.67	22.44	34.06	46.64	86cm
Left	10.71	21.51	31.77	43.02	77cm
Left	18.11	26.3	33.03	39.76	75com
Left	10.55	19.36	30.08	42.55	80cm

Picture of Testing:

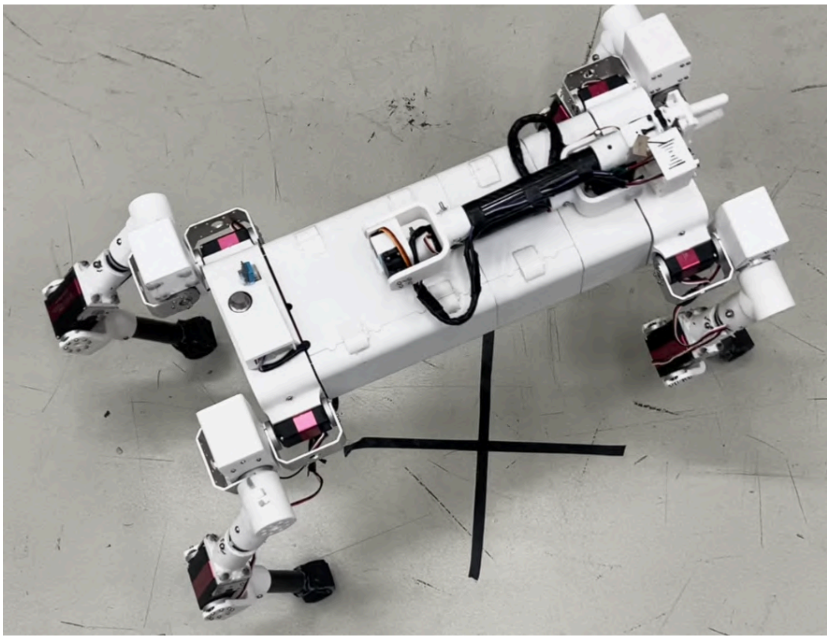


Figure 5.30: Picture of Turning on Concrete

2. **Grass/Turf**

TABLE 5.19: TABLE OF THE TURNING TEST RESULTS ON GRASS/TURF

Turning Direction	90° Turn Time (s)	180° Turn Time (s)	270° Turn Time	360° Turn Time (s)	Min Turning Radius (cm) (360 deg)
Right	12.01	25.08	38.36	54.84	104.5
Right	12.29	26.47	40.68	55.74	90
Right	16.12	27.29	42.49	57.88	85
Left	11.20	24.20	37.50	50.08	90
Left	12.28	27.06	42.06	52.01	80
Left	14.17	26.43	41.97	52.99	83

Picture of Testing:

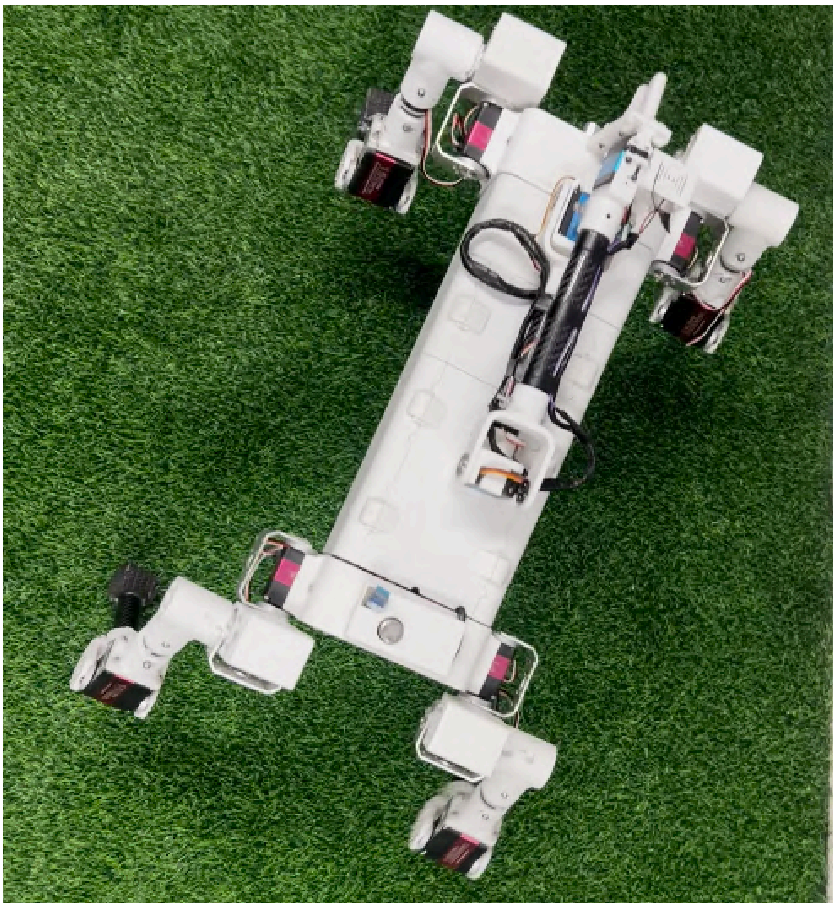


Figure 5.31: Picture of Turning on Grass/Turf

3. **Tiles**

TABLE 5.20: TABLE OF THE TURNING TEST RESULTS ON TILES

Turning Direction	90° Turn Time (s)	180° Turn Time (s)	270° Turn Time	360° Turn Time (s)	Min Turning Radius (cm) (360 deg)
Right	12.00	21.90	32.04	41.72	76vm
Right	11.09	21.55	31.51	42.70	78cm
Right	09.60	19.93	28.20	39.07	70cm
Left	12.64	26.42	38.42	53.39	104cm
Left	12.87	28.44	40.44	53.06	100cm
Left	12.54	25.66	37.32	50.41	98cm

Picture of Testing:

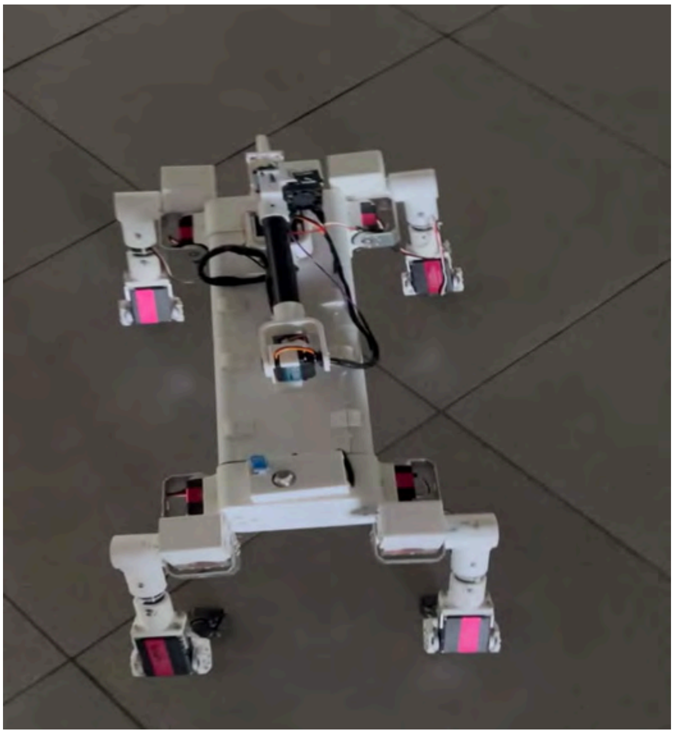
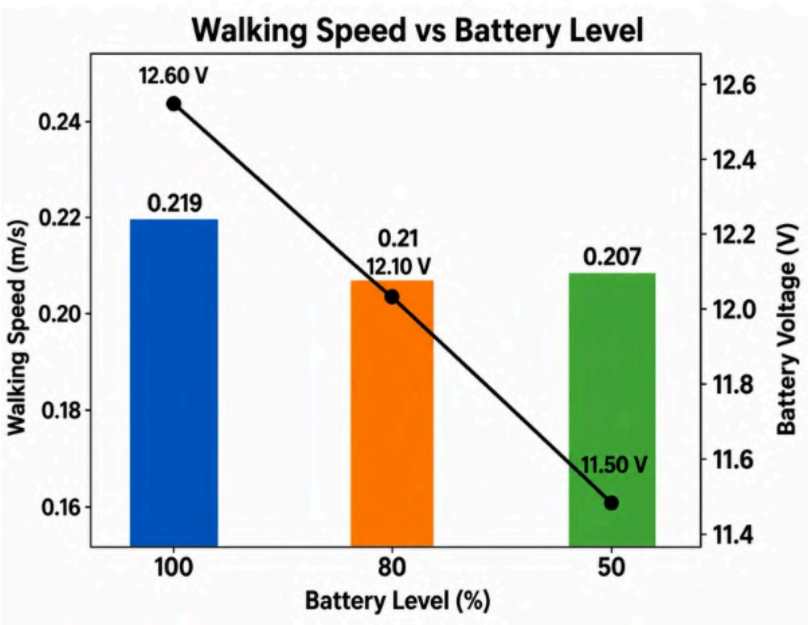


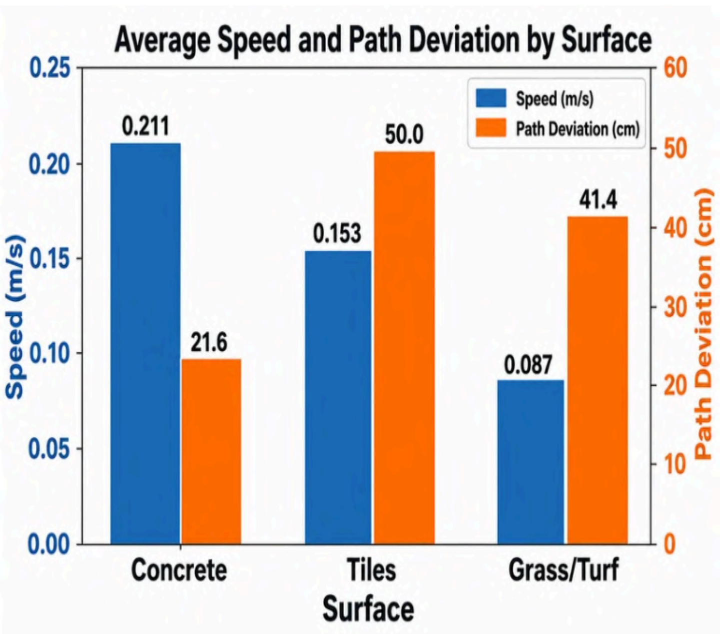
Figure 5.32: Picture of Turning on Tiles

5.3.3 Result and Discussion:

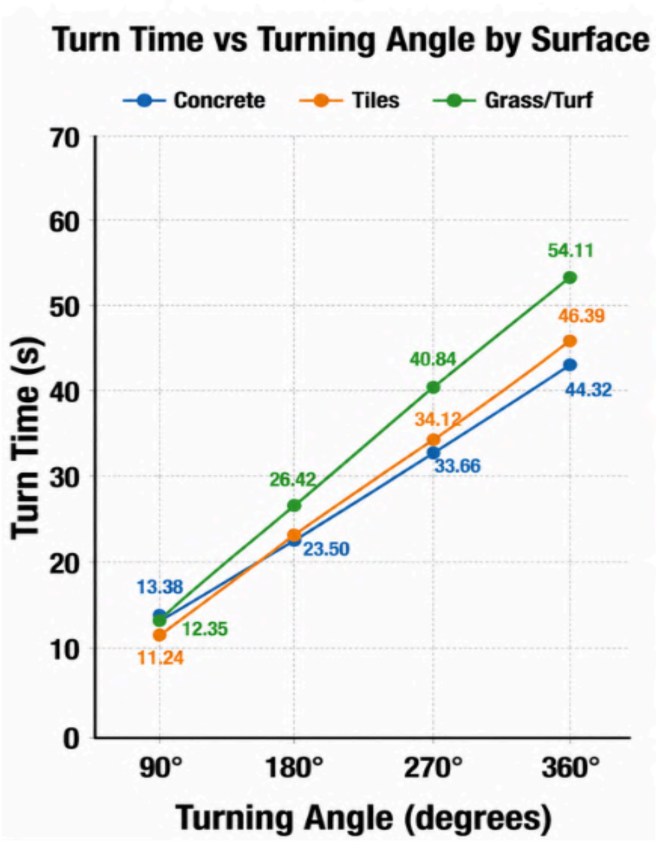
1. **Battery-Level-Based Walking Performance Analysis**

<table>
<tr>
<td>
</td>
<td>The graph shows the effect of battery level on the robotic dog's walking speed. As the battery level decreased from 100% to 50%, the voltage dropped from 12.60 V to 11.50 V. However, the walking speed changed only slightly, from about 0.219 m/s at full charge to 0.207 m/s at 50% battery. This indicates that the robotic dog maintained a relatively stable walking speed within the tested battery range. The small speed reduction suggests that the motor system was still able to provide sufficient power, although performance may decrease more noticeably below 50% battery.</td>
</tr>
<tr>
<td colspan="2">Figure 5.33: Battery-Level-Based Walking Performance Analysis</td>
</tr>
</table>

2. **Speed and Path Deviation Evaluation Across Different Surface Conditions**

<table>
<tr>
<td>
</td>
<td>The graph shows that the robotic dog performed best on the concrete surface, where it achieved the highest average speed of 0.211 m/s and the lowest path deviation of 21.6 cm. This indicates that concrete provided better stability, traction, and straight-line movement control. On tiles, the average speed decreased to 0.153 m/s, while the path deviation increased to 50.0 cm. This suggests that the robot experienced more slipping or directional instability on the smooth tiled surface. The grass/turf surface produced the lowest average speed of 0.087 m/s, showing that the surface created higher resistance against movement. However, its path deviation was 41.4 cm, which was lower than tiles but still higher than concrete.</td>
</tr>
<tr>
<td colspan="2">Figure 5.35: Speed and Path Deviation</td>
</tr>
</table>

3. Surface-Dependent Turning Time and Maneuverability Analysis



The graph compares the turning time of the robotic dog at different turning angles on concrete, tiles, and grass surfaces. The turn time increased as the turning angle increased from 90° to 360° for all surfaces. Sowing that larger turning angles required more movement adjustment and longer execution time. Among the three surfaces, grass/turf required the highest turning time, especially at larger angles. E.g., at 360°, the robot took approximately 54.11 s on grass/turf, compared with 46.39 s on tiles and 44.32 s on concrete. This suggests that grass created more resistance and reduced efficiency. Concrete showed comparatively better performance, while tiles also produced stable turning performance but required slightly more time than concrete at higher angles.

Figure 5.34: Surface-Dependent Turning Time and Maneuverability

Key Findings:

Concrete proved to be the most suitable surface for the robotic dog, showing the highest average walking speed of 0.211 m/s, the lowest path deviation of 21.6 cm, and the quickest turning performance. Battery level had only a minor effect on walking speed, as the robot maintained stable movement even when voltage dropped from 12.60 V to 11.50 V. Grass/turf reduced speed the most, while tiles caused the highest path deviation.

5.3.4 Battery Performance and Power Consumption Analysis

This section evaluates the battery performance of the robotic dog during operation. The battery voltage and elapsed time were recorded at different charge levels to observe discharge behaviour. In addition, walking performance was tested at different battery levels, and the approximate power consumption was calculated for both walking and idle modes.

1. **Battery Recharge or Elapsed Time**

TABLE 5.21: TABLE OF THE BATTERY RECHARGE OR ELAPSED TIME

Battery Level	Recorded Time (PM)	Voltage (V)	Time Difference from Previous Reading	Elapsed Time from Initial Reading
100%	2:14	12.60 V	0 min	0 min
80%	2:35	12.14 V	20 min	20 min
50%	2:46	11.50 V	12 min	32 min

2. **Walking Performance based on the battery level**

TABLE 5.22: TABLE OF THE WALKING PERFORMANCE BASED ON THE BATTERY LEVEL

Battery Level at Start	Battery Voltage (V)	Distance (m)	Travel Time (s)	Corrected Walking Speed (m/s)	Runtime / Elapsed Time
100%	12.60	5	22.80	0.2193	0 min
80%	12.10	5	24.10	0.2075	20 min
50%	11.50	5	23.81	0.2100	32 min

3. **Battery Power Consumption in Walking and Idle Modes**

We know, Energy = Voltage × Ampere-hour

The Lipo battery we used is 5000 mAh, which is equivalent to 5 Ah. Since the battery voltage is 12 V, the total energy capacity is: Energy = 12 V × 5 Ah = 60 Wh

In walking mode, the battery was depleted in approximately 30 minutes, which is equal to 0.5 hours. Therefore, the estimated power draw is:

$$\text{Power} = 60 \text{ Wh} \div 0.5 \text{ h} = 120 \text{ W}$$

So, it consumes approximately 120 W in walking mode.

In idle mode, the battery was depleted in approximately 3 hours. Therefore, the estimated power draw is:

$$\text{Power} = 60 \text{ Wh} \div 3 \text{ h} = 20 \text{ W}$$

So, it consumes approximately 20 W in idle mode.

5.3.5 Specification of Arm

The robotic arm was tested to measure its horizontal reach, vertical reach, lifting capacity, and load-holding time. This test was conducted to evaluate whether the arm could effectively reach, lift, and hold small objects during robot operation.

TABLE 5.23: TABLE OF THE SPECIFICATION OF ARM

Max Horizontal Reach (cm)	Max Vertical Reach (cm)	Max Lift at Short Reach	Max Lift at Full Extension	Hold Time with Load (s)
38cm	96cm	30gm	300gm	27.37s (full extended)

Picture of Testing:

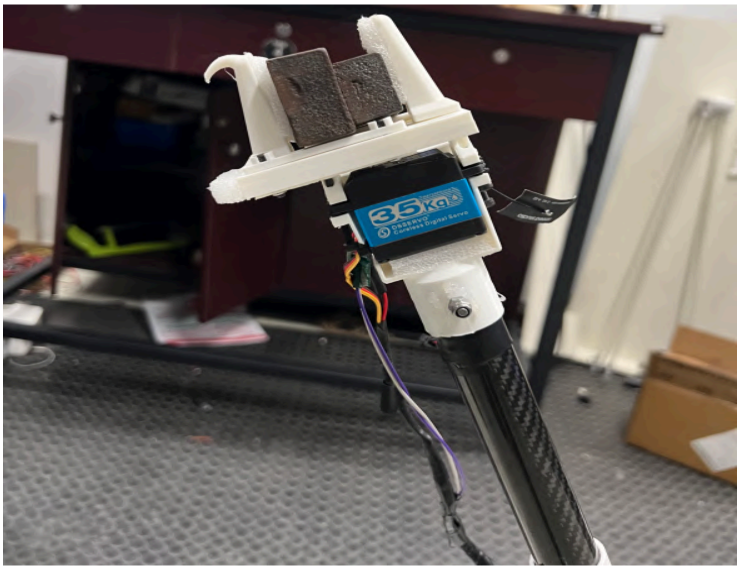


Figure 5.35: Arm Weight Lifting Test

5.3.6 Fault Detection Test:

The fault detection test was conducted to evaluate the robotic dog's ability to identify cracks using its vision system. A real crack was manually made on a PVC board, while the artificial cracks were printed on paper to test detection range, response time, and accuracy under different crack conditions.

TABLE 5.24: TABLE OF THE FAULT DETECTION TEST

Trial	Fault Type	Detected Correctly? (Y/N)	Longest range to detect	Time to Identify (s)	Accuracy (%)
1	Real Crack	Y	50cm	0.7s	70%
2	Artificial Crack	Y	60cm	1.3s	70%
3	Artificial Crack	Y	30cm	1s	70%

Picture of Testing:

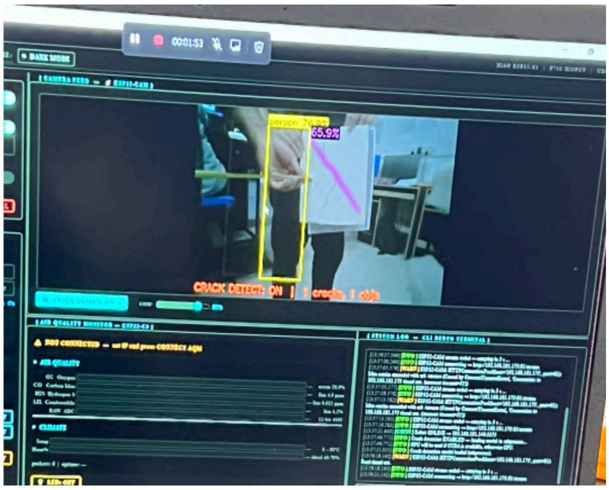
Measuring distance from Crack	Real-time crack detection with accuracy (%)
	

Figure 5.36: Measuring distance from Crack With Accuracy

5.3.7 Obstacle Avoidance Test

The obstacle avoidance test was conducted to evaluate the robotic dog's ability to detect nearby obstacles and stop before collision. An ultrasonic sensor was used to measure the distance between the robot and the obstacle, and the test was performed using both a cardboard box and a wall to check detection accuracy in different obstacle conditions.

TABLE 5.25: TABLE OF THE OBSTACLE AVOIDANCE TEST

Trial	Obstacle Type	Stopping Distance from Obstacle (cm)	Detected Correctly? (Y/N)
1	Cardboard Box	42cm	Y
2	Cardboard Box	46cm	Y
3	Cardboard Box	43cm	Y
4	Wall	45cm	Y
5	Wall	37cm	Y

Picture of Testing:

Figure 5.37: Measuring Distance of Obstacle Avoidance

5.3.8 Control and Video Communication Performance

The control and video communication test was conducted to evaluate the robot's remote operation performance. The gamepad control range and input latency were measured to check how quickly the robot responded to user commands, while the video range and video latency were recorded to assess the reliability of real-time visual feedback during operation.

TABLE 5.26: TABLE OF THE CONTROL AND VIDEO COMMUNICATION PERFORMANCE

Max Control Range (m)	Latency: Gamepad Input to Robot Movement (ms)	Max Video Range (m)	Video Latency (ms)
6.53m	350ms	5.1m	280ms

Responsive Emergency Shutdown Operation

For the responsive operation test, a small emergency-control setup was developed using a PVC board wall. A 3D-printed lever was mounted on the board, with a red LED indicator placed above it. The red LED was connected to a buzzer, so when the red light turned on, the buzzer also activated to indicate an emergency condition.

During the test, the robotic dog was required to respond to the emergency signal by moving toward the wall, reaching the lever, and pulling it downward. Once the lever was pulled down, the emergency system was deactivated, the buzzer stopped, and the green LED turned on to indicate that the system had been successfully shut down.

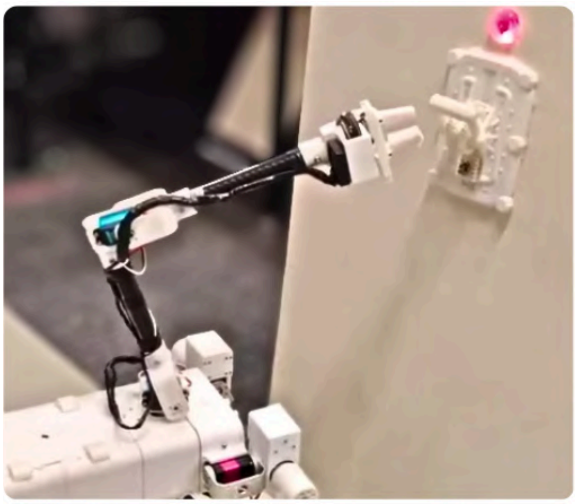
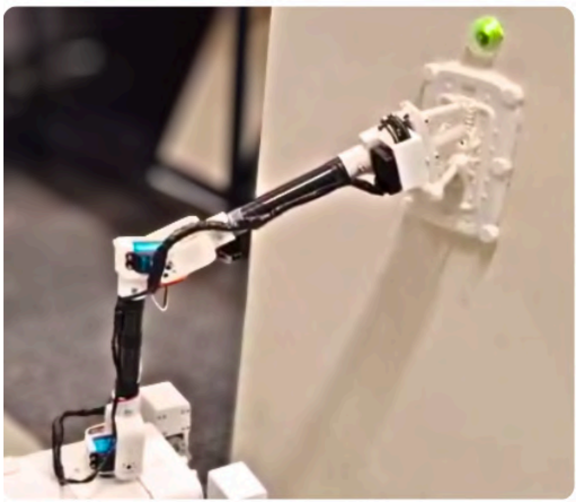
Red bulb is on in Emergency condition	Green LED after the lever was pulled down
	

Figure 5.38: Liver Pulling Test

Idle Mode (T Pose) of the Robot:

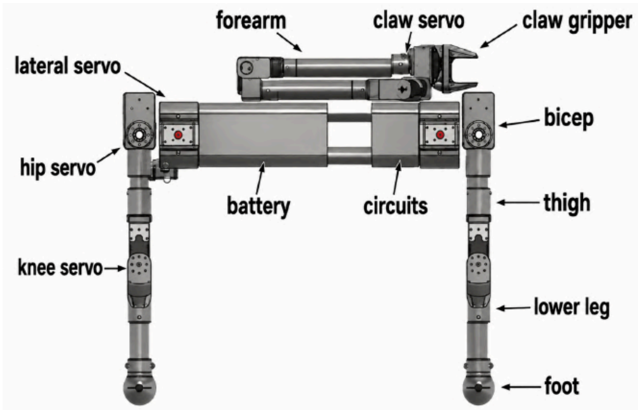
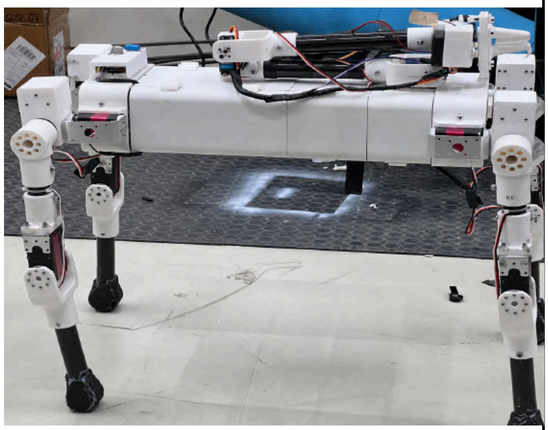
Model on Fusion360	Hardware Implementation
	

Figure 5.39: Idle Mode (T Pose) of the Robot

Walking Mode of the Robot:

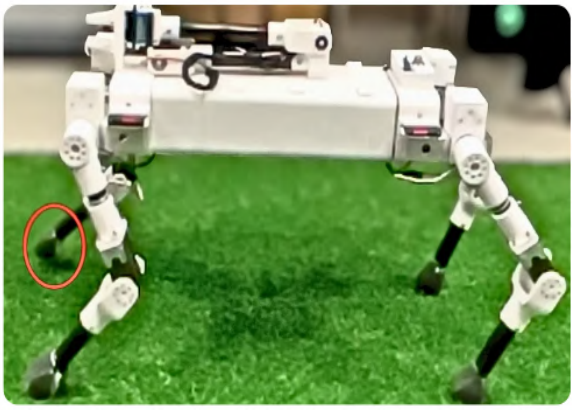
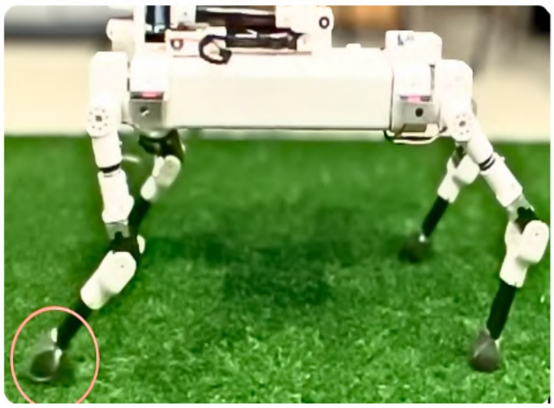
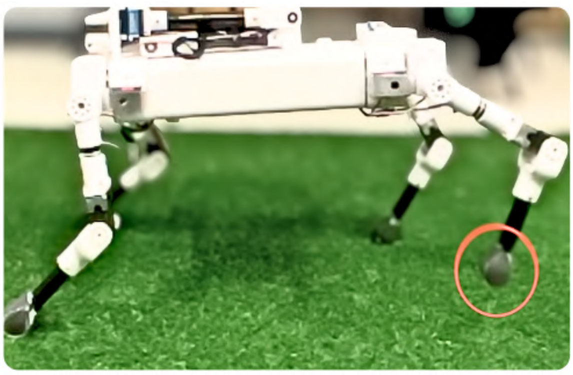
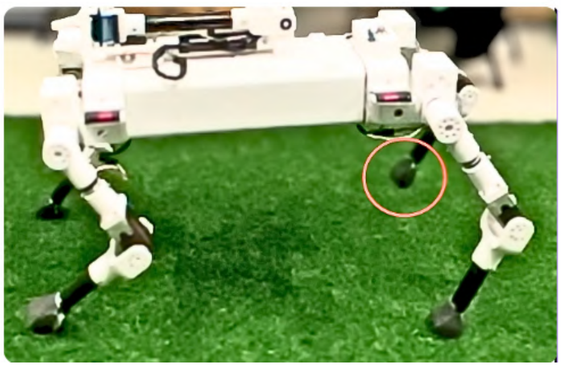
	
Initial leg lift during walking	**Forward swing of the lifted leg**
	
Foot placement after forward movement	**Body support and next step preparation**

Figure 5.40: Walking Mode of the Robot

Overall System Workflow:

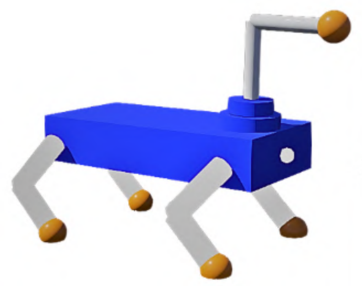
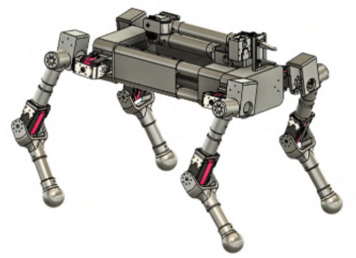
Software Simulation	Fusion360 Design	Hardware Implementation
		

Figure 5.41: Overall System Workflow

5.4 Conclusion

Chapter 5 presented the complete development and validation of the final quadruped robot design. In this chapter, we moved from the theoretical design stage to the physical implementation stage by completing torque analysis, component selection, CAD modelling, 3D printing, mechanical fabrication, circuit integration, and full system assembly. We also implemented the control logic, inverse kinematics, joystick-based manual operation, robotic arm control, sensor integration, and YOLO-based fault detection system. After completing the prototype, we tested the robot through different validation activities, including walking, turning, obstacle detection, battery performance, payload handling, fault detection, and responsive operation. These tests helped us verify that the major subsystems were working together as an integrated platform. Overall, Chapter 5 showed that the proposed design was successfully converted into a functional prototype and validated through practical testing against the project's main technical requirements.

Chapter 6: Impact Analysis and Project Sustainability. [CO3, CO4]

6.1 Introduction

Chapter 6 evaluates the broader impact and sustainability aspects of the proposed quadruped robotic inspection system. Beyond technical performance, it is essential to assess how the solution influences societal, economic, health, safety, legal, and cultural domains. The chapter further examines the environmental implications of the system across different lifecycle stages, including pre-construction, construction, and operational phases. Additionally, sustainability challenges and opportunities are analyzed through application-based impact assessment and SWOT analysis. This comprehensive evaluation ensures that the proposed system is not only technically feasible but also socially responsible, environmentally sustainable, and aligned with modern industrial requirements.

6.2 Assess the impact of solution

The impact of this project can be demonstrated in social, health, economic, cultural, and legal impacts.

6.2.1 Societal Impact:

TABLE 6.1: TABLE OF THE SOCIETAL IMPACT

Impact Area	Evidence	Description
Global economic loss due to corrosion	Corrosion costs ~US $2.5 trillion annually (~3.4% of global GDP) [28]	Early detection reduces infrastructure maintenance costs significantly.
National burden (example: USA)	US $276 billion/year (~3.1% GDP) lost due to corrosion [29]	Lower repair/replacement expenses in industries like oil & power.
Savings potential	Preventive monitoring can cut costs by 15–35% [30]	Billions saved globally; increased asset lifespan and efficiency.

6.2.2 Health Impact:

Table 6.2: Table of The Health Impact

Impact Area	Evidence	Description
Workplace hazards in heavy industries	2.78 million deaths and 374 million non-fatal injuries yearly due to workplace conditions [31]	Robots reduce direct human exposure to toxic gas, heat, and high-risk areas.
Exposure to gases/oil leaks	Common in petrochemical & refinery environments	Continuous environmental monitoring via sensors (temp, humidity, ultrasonic) prevents health hazards.
Long-term occupational illnesses	Chronic respiratory & heat-related diseases in workers	Automated early warning systems ensure a healthier workforce and reduced healthcare costs.

6.2.3 Safety Impact

Table 6.3: Table of The Safety Impact

Impact Area	Evidence	Description
Industrial robot accidents	61 fatalities (1992–2015) from robot-related incidents [31]	Remote-controlled & autonomous scouting reduces direct human–robot contact.
High-risk environments	Maintenance in HV substations, nuclear plants, and refineries is highly dangerous	The robot's retractable arm manipulates valves/switches, eliminating human exposure.
Safety standards	ISO 10218 and ISO/TS 15066 provide robotics safety frameworks [32]	Compliance ensures reduced accident probability and industry acceptance.

6.2.4 Legal Impact

Table 6.4: Table of The Legal Impact

Impact Area	Evidence	Description
Regulatory compliance in hazardous industries	Oil, gas, and chemical industries are regulated under frameworks such as OSHA Process Safety Management (PSM) and EPA Risk Management Program (RMP) [33]	Robots ensure continuous monitoring that helps industries comply with strict safety/environmental laws.
Liability and worker protection	Companies face heavy fines for workplace accidents; OSHA penalties can exceed $15,000 per violation and $156,000 for repeated violations [33]	By reducing worker exposure, robots minimize legal liability and protect companies from penalties.
Nuclear and high-voltage regulations	Nuclear plants require compliance with IAEA Safety Standards and national laws on radiation exposure [34]	Robots can perform remote tasks (valve/switch operations) which reduces human exposure and ensures compliance.
Data and operational transparency	EU's General Data Protection Regulation (GDPR) and similar laws require secure handling of monitoring data	Robot's data logs create audit trails for compliance, increasing legal accountability and operational transparency.

6.2.5 Cultural Impact

Table 6.5: Table of The Cultural Impact

Impact Area	Evidence	Description
Workforce anxiety about automation	Workers fear robots may replace jobs, especially among older groups	Framing a robot as a safety-assistant (not replacement) builds acceptance.
Demand for safer workplaces	Workers & unions push for reduced exposure to hazardous conditions	Robots satisfy this demand for a positive cultural shift toward robotics.
Need for training & adaptation	Industrial transformation requires upskilling	Training programs align workers with robotic operations, ensuring smoother cultural adoption.

6.3 Evaluate the sustainability

6.3.1 Environmental and Materials Impacts during Pre-Construction Stage:

Table 6.6: The Environmental and Materials Impacts during Pre-Construction Stage

IECs/Issues	Potential Impacts from Various Activities	Magnitude	Significance Prior to Mitigation	Compliance (laws/regulations)
Material Selection	Poor selection of PLA, PVC, carbon fiber, or electronics may increase material waste during later redesign.	Moderate	Moderate adverse	ISO 14001; SDG 12
Component Procurement	Importing servos, batteries, sensors, and microcontrollers creates packaging waste and transport-related emissions.	Moderate	Moderate adverse	ISO 14001; Green procurement guidelines
Battery Planning	Early selection of LiPo/Li-ion batteries affects future battery safety, storage, disposal, and recycling needs.	Moderate	Moderate adverse	Basel Convention; EU Battery Directive
Digital Design & Simulation	CAD, Webots simulation, and ML model preparation require computer energy use before physical construction.	Minor	Minimal adverse	ISO 50001; Green IT guidelines
Positive Design Planning	Modular design, reusable components, and simulation before fabrication reduce trial-and-error material waste.	Positive	Beneficial	SDG 12; ISO 14006

6.3.2 Environmental and Materials Impacts during Construction Stage:

Table 6.7: Table of The Environmental and Materials Impacts during Construction Stage

IECs/Issues	Potential Impacts from Various Activities	Magnitude	Significance Prior to Mitigation	Compliance (laws/regulations)
3D Printing Waste	Failed prints, support material, and excess PLA from body parts, joints, and mounts create plastic waste.	Moderate	Moderate adverse	ISO 14001; SDG 12
Carbon Fiber/PVC Cutting	Cutting rods and prototype pipe sections may produce dust, offcuts, and small particles during fabrication.	Moderate	Moderate adverse	OSHA PPE guidance; ISO 45001
Soldering & PCB Assembly	Soldering veroboards, wiring, and connectors can produce fumes, lead residue, and electronic scrap.	Moderate	Moderate adverse	RoHS; E-waste Rules 2011
Battery & Power Integration	LiPo battery installation and high-current wiring may create short-circuit, fire, or damaged-cell waste risks.	Major	Major adverse	IEC 62133; Basel Convention
Positive Reuse & Repairability	Reusing available parts, careful soldering, and modular assembly reduce unnecessary component waste.	Positive	Beneficial	SDG 12; ISO 14001

6.3.3 Environmental and Materials Impacts during Operational Stage:

Table 6.8: Table of The Environmental and Materials Impacts during Operational Stage

IECs/Issues	Potential Impacts from Various Activities	Magnitude	Significance Prior to Mitigation	Compliance (laws/regulations)
Battery Lifecycle	Continuous operation and frequent charging cycles increases battery disposal and recycling demand.	Major	Major adverse	Basel Convention; EU Battery Directive
Sensor Degradation & Replacement	Cameras, ultrasonic sensors, and humidity probes degrade in harsh petrochemical/nuclear sites, creating high-tech e-waste streams.	Moderate	Moderate adverse	E-waste Rules 2011
Data Transmission Energy Use	Cloud-based monitoring (AI-inference, data storage) adds hidden carbon footprint from server energy use.	Moderate	Moderate adverse	ISO 50001 (Energy Management); Green IT guidelines
Electromagnetic Exposure	Continuous robot patrols in HV substations may add to localized EMI affecting precision grid control systems.	Minor	Minimal adverse	IEEE/IEC EMC Standards
Positive Safety Benefit	Robots reduce human entry into high-risk zones (radiation, toxic gas, HV), preventing accidents and fatalities.	Positive	Beneficial	IAEA, OSHA, ISO 10218

6.3.4 Environment & Sustainability: Alignment with SDGs

Throughout the project, we tried to maintain relevant Sustainable Development Goals (SDGs), especially during the operation and development phases. The robot was designed to improve worker safety by reducing direct human exposure to hazardous inspection areas. We also simulated the design and control system in software before physical fabrication, which helped us avoid repeated rebuilding and reduce unnecessary waste of materials such as carbon fiber and PLA. In this way, the project supported sustainability through safer operation, efficient material use, rechargeable power, modular repair, and responsible engineering practice.

SDG 3: Good Health and Well-Being

The project supports SDG 3 by reducing health risks during both development and future industrial use. During the fabrication stage, we followed safer working practices while soldering, cutting carbon fiber, and handling electronic components. These activities can produce fumes, dust, sharp waste, and hazardous residues if not managed properly. Therefore, we maintained careful handling, controlled cutting, and safe soldering practices to reduce direct exposure to harmful fumes, dust, and waste. In the final application, the robot also improves worker safety by allowing remote inspection in hazardous environments such as substations, confined spaces, high-voltage areas, and industrial zones, reducing the need for human workers to enter dangerous locations directly.

SDG 8: Decent Work and Economic Growth

The project also supports SDG 8 by promoting safer, more efficient, and technology-assisted working conditions in industrial environments. In many hazardous workplaces, inspection tasks require workers to enter high-risk areas such as substations, confined spaces, high-voltage zones, and chemically exposed industrial sites. By using the quadruped robot for remote inspection and responsive operation, the project helps reduce direct human exposure to dangerous conditions while allowing workers to monitor faults from a safer distance. This improves occupational safety and supports decent working conditions. The robot can also reduce inspection time, support early fault detection, and help prevent unexpected shutdowns through predictive maintenance. As a result, the project contributes to productivity, safer industrial operations, and long-term economic value by combining human supervision with robotic assistance rather than fully replacing human workers.

SDG 9: Industry, Innovation and Infrastructure

The strongest SDG connection of this project is SDG 9, because the project develops an innovative low-cost quadruped robot for industrial inspection and responsive operation. The system combines mechanical design, embedded control, wireless communication, sensor-based inspection, YOLOv8-based crack detection, obstacle detection, and a robotic arm into one integrated platform. Chapter 5 validates the final prototype through practical testing, including walking, fault detection, battery performance, arm operation, and system response. This supports industrial innovation by offering a practical academic prototype that can inspect infrastructure, identify cracks or faults, and support predictive maintenance before serious failure occurs.

SDG 11: Sustainable Cities and Communities

The project supports SDG 11 by contributing to safer and more resilient infrastructure. Industrial facilities, substations, power systems, and hazardous plants are important parts of modern communities, and failures in these systems can affect public safety, energy supply, and environmental conditions. The robot's ability to inspect cracks, corrosion, faults, and unsafe areas helps identify risks before they become major incidents. The real-life testing plan at the BRACU substation also strengthens this connection because it shows that the system is not only a laboratory prototype, but also intended for practical inspection in real infrastructure environments.

SDG 12: Responsible Consumption and Production

The project supports SDG 12 through responsible material use, reuse of available components, and reduction of unnecessary waste. We reduced waste by using rechargeable batteries instead of disposable power sources. We also reused functional components from broken or unused devices where possible, which helped reduce electronic waste and lower project cost. During development, the project also used 3D-printed PLA joints, carbon fiber rods, selected cost-effective actuators, reused available laboratory resources, and avoided unnecessary CNC metal cutting. These decisions helped reduce cost, material waste, and overuse of expensive manufacturing processes. The sustainability section also identifies e-waste as a challenge and proposes modular design and recycling compliance as mitigation strategies. Therefore, the project maintains SDG 12 by encouraging repairability, efficient material use, and responsible handling of electronic components.

SDG 13: Climate Action

The project supports SDG 13 by reducing material impact and encouraging lower-emission fabrication choices. Some of the mechanical parts were produced using PLA, a bio-based material commonly used in 3D printing. Compared with more energy-intensive manufacturing methods, 3D printing with PLA helped reduce material wastage during prototyping because parts could be printed according to the required shape and size. This also reduced the need for excessive machining and unnecessary raw material use. By combining lightweight design, rechargeable batteries, reusable components, and bio-based PLA parts, the project contributed to a lower environmental footprint during development.

6.3.5 SWOT Analysis:

Table 6.9: Table of The SWOT Analysis

	Beneficial	Harmful
Internal	**Strengths** • Reduces human risk in hazardous zones. • Multi-sensor integration for real-time monitoring. • Cost-saving by preventing failures.	**Weaknesses** • High initial development cost. • Requires skilled operators and maintenance. • Limited payload compared to heavy-duty robots.
External	**Opportunities** • Expanding into Industry 4.0 and smart factories. • Integration with AI-driven predictive analytics. • Growing demand for industrial automation post-COVID.	**Threats** • Worker resistance due to job replacement fears. • Cybersecurity risks in data transmission. • Legal barriers in regulated industries.

6.3.6 Challenges in Sustainability:

Table 6.10: Table of Challenges in Sustainability

Challenge	Explanation	Mitigation Strategy
Energy consumption	Robots require frequent charging; batteries have limited lifecycle.	Use renewable-powered charging stations; explore solid-state batteries.
E-waste management	Sensors and electronics contribute to rising e-waste (53.6 Mt globally in 2019) [36]	Modular design for easy part replacement; adherence to recycling regulations.
Industry adoption	Resistance due to cultural/job loss fears.	Training and awareness programs; frame robots as “co-worker” not “replacement.”
Regulatory Bodies	Compliance with OSHA, EPA, and IAEA standards needed for industrial acceptance [33],[36].	Design aligned with safety standards; provide audit-friendly monitoring logs.

6.4 Conclusion

This chapter demonstrated that the proposed quadruped robotic system has significant positive impact across multiple domains, particularly in improving safety, reducing human exposure to hazardous environments, and enhancing operational efficiency in industrial applications. The analysis also highlighted its potential economic benefits through reduced maintenance costs and improved asset reliability. At the same time, the sustainability assessment identified key environmental and operational challenges, including energy consumption, e-waste generation, and regulatory compliance requirements. However, with proper mitigation strategies such as modular design, efficient energy management, and adherence to safety and environmental standards, these challenges can be effectively addressed. Overall, the findings confirm that the proposed system is not only technically viable but also contributes positively toward sustainable industrial practices, making it a valuable solution for future smart and safe industrial environments.

Chapter 7: Engineering Project Management. [CO11, CO14]

7.1 Introduction

This chapter presents the engineering project management process followed for the development of the quadruped robotic dog for real-time inspection and responsive operation. The project was carried out from June 2025 to May 2026, covering proposal development, design planning, simulation, fabrication, integration, testing, validation, and final documentation.

Since the project involved multiple technical areas, including mechanical design, walking algorithm development, electronics integration, GUI communication, machine learning-based crack detection, robotic arm operation, and system testing, proper planning was necessary to distribute the work effectively among team members. The project was divided into sequential phases so that each stage could build on the previous one without creating major delays in the final implementation.

Our project management approach was mainly maintained in three areas: time management, cost management, and risk management with contingency planning. Time management helped us complete each phase within the academic timeline, while cost management helped us reduce the actual project cost from the planned budget through careful procurement, component selection, reuse of resources, and fabrication decisions. Risk management was also included to identify possible hazards, assess their severity, apply mitigation strategies, and prepare contingency plans for issues such as battery failure, robot tipping, communication loss, motor overheating, collision, and electrical faults.

Although the project faced schedule interruptions due to university closures, national election-related vacation, Eid-ul-Fitr vacation, and extended closure due to power shortage, the team managed the time by starting practical work early, working remotely on software-related tasks, and using available resources efficiently.

7.2 Define, Plan, and Manage Engineering Project

7.2.1 Project Timeline and Phase Structure

The project was planned as a multidisciplinary engineering task that required coordination between mechanical, electrical, software, control, and testing activities. The overall aim was to build a functional quadruped robot that could walk on different surfaces, detect faults, avoid obstacles, monitor environmental conditions, and perform limited responsive operations using a robotic arm and claw.

The project was divided into five major phases:

1. Planning and Literature Review
2. CAD Design, Gait Calculation and Simulation
3. 3D Printing, Fabrication and Electronics Integration

4. GUI, Machine Learning Model and Control Calibration
5. Testing, Validation and Final Documentation

During the initial phase from June 2025 to August 2025, the team focused mainly on project proposal preparation, background study, problem identification, literature review, and requirement analysis. At this stage, no major physical construction was done. Instead, the main focus was to define the project scope and understand the technical requirements of quadruped locomotion, inspection robotics, and responsive operation.

In the next phase, during the Fall 2025 semester, the team worked on simulation and design validation. Webots simulation was used to test walking behaviour, inverse kinematics, and manual versus autonomous control approaches. This phase helped the team understand the expected motion behaviour before moving into physical construction. During the vacation after Fall 2025, the team also worked on Fusion 360 CAD modelling, preparing the 3D models required for fabrication.

The physical development phase began earlier than the official semester start. Although classes were scheduled to begin on January 31, the team started laboratory work on January 26 with special permission. This early start was important because the project required a large amount of 3D printing, mechanical assembly, wiring, testing, redesign, and validation. Starting early reduced the risk of final-stage delays.

7.2.2 Team Organization and Responsibility Distribution

The project management structure enforces equitable contribution through specialization-based task assignment while maintaining cross-functional collaboration at all major integration points. Each team member holds a defined primary domain of responsibility while providing active support across all project areas.

1. Shadid Al Akib (Mechanical Design Lead): Leads all mechanical design, CAD modelling, and 3D printing activities — including leg geometry design, joint bracket iterations, PLA part fabrication, and the transition from PVC to carbon fiber structural components.
2. Tazwar Ahmed (Electronics, GUI and Communication Lead): Responsible for all electronics assembly and wiring, including servo wiring harnesses, veroboard soldering, PCB layout, power distribution, as well as the graphical user interface (GUI) and wireless communication stack.
3. Md. Shahariar Hossain (Walking Algorithm, IK, ML and Simulation Lead): Develops the walking algorithm, inverse kinematics engine, and machine learning components for gait optimization, and conducts full robot simulation in the Webots robotics simulator to validate locomotion before physical trials.
4. Rafin Md Abrar (Literature Review, Body Assembly, Testing, Validation and Documentation Lead): Conducts and coordinates the literature review, oversees physical body assembly, leads testing and validation campaigns across all subsystems, and compiles the complete project report and technical documentation.

The detailed task distribution across all five phases is presented below.

TABLE 7.1: TASK DISTRIBUTION AMONG TEAM MEMBERS

Task	Md. Shahariar Hossain (Walking Algorithm, IK, ML & Simulation)	Rafin Md Abrar (Literature Review, Assembly, Testing & Docs)	Shadid Al Akib (Mechanical Design, CAD & 3D Printing)	Tazwar Ahmed (Electronics, Wiring, GUI & Communication)
Phase 1: Planning and Initialization				
Literature Review	Review locomotion algorithms, IK theory, ML for gait, Webots simulation	Lead literature review across all domains; compile annotated bibliography	Review legged robot mechanical design, materials, DOF layouts	Review electronics architectures, GUI frameworks, wireless protocols
Problem Statement & Objectives	Define algorithmic and simulation objectives	Draft problem statement and overall project scope	Define mechanical constraints, material goals, and DOF targets	Define electronics, GUI, and communication requirements
Design Approach Selection	Evaluate control architectures and simulation environments	Evaluate feasibility of design choices; consolidate team decisions	Evaluate structural materials (PVC vs carbon fiber vs metal)	Evaluate wiring topologies and communication protocols (Wi-Fi, BT)
Initial Concept Note	Write algorithm, IK and simulation section	Write literature summary and project management section	Write mechanical design section	Write electronics, GUI and communication section
Phase 2: Design and Component Selection				
System Architecture Design	Design IK solver and gait planning framework; set up Webots simulation environment	Coordinate architecture across all subsystems; define integration interfaces	Design leg geometry and DOF layout in CAD; select materials	Design electronics architecture, wiring topology, and GUI framework

Component Procurement	Identify simulation software requirements	Assist with procurement coordination; manage delivery timelines	Procure carbon fiber rods, PLA filament, fasteners, servo hardware	Procure ESP32 boards, PWM drivers, batteries, wiring components
Software Framework	Build Webots simulation prototype; implement preliminary IK model	Document design decisions; review and validate component choices	Prototype 3D-printed joint brackets; check fit against servo dimensions	Develop preliminary wiring diagrams; set up ESP32 development environment
Phase 3: Development and Fabrication				
Iteration 1: PVC Frame Assembly	Validate Webots gait model against PVC prototype behaviour; log discrepancies	Assist with body assembly; document iteration 1 design decisions	Fabricate PVC pipe frame; 3D print initial joint brackets; assemble leg structure	Wire servo motors and ESP32; connect power distribution; test basic joint actuation
Iteration 2: Carbon Fiber Upgrade	Update Webots model geometry to match carbon fiber dimensions; re-validate IK	Document redesign rationale; assist with assembly and fit-checking	Replace PVC with carbon fiber rods; redesign and reprint PLA joints to new geometry	Re-route and tighten wiring harness; solder veroboards carefully for clean layout
Electronics & PCB Assembly	Integrate IMU data into ML model and Webots feedback loop	Test and verify each electronic subsystem as assembled	Assist with mounting hardware; print enclosures and cable management parts	Solder veroboards; integrate PCA9685 PWM drivers; build and test GUI interface
Full System Integration	Deploy IK + gait algorithm on ESP32; verify Webots predictions match real motion	Lead full system assembly; integrate all subsystems; coordinate integration testing	Ensure all mechanical components are correctly assembled and torqued	Verify wiring under full load; integrate wireless communication; test GUI end-to-end

Phase 4: Testing and Validation				
Lab Testing	Validate walking algorithm performance; compare against Webots simulation predictions	Lead testing sessions; log all results; identify failure modes	Inspect mechanical joints under load; identify and reprint failed parts	Test electronics reliability; verify GUI and wireless communication stability
Locomotion Validation	Tune gait timing, step height, and IK parameters based on test data	Document locomotion test results; compile performance metrics	Refine leg structure to eliminate flex or backlash at joints	Verify power draw during extended gait; confirm wiring stability under vibration
Responsive Operation Testing	Test ML-based adaptive responses; validate against real-world disturbances	Document responsive operation test results	Inspect and reinforce mechanical attachments for manipulation/claw tasks	Test GUI command latency; verify wireless control under motion
Compliance and Safety Validation	Add software limit checks and fail-safe conditions in firmware	Compile all test data; verify all KPIs are met	Inspect all mechanical stress points; confirm safe joint travel limits	Add fuse and over-current protection; finalize wiring safety provisions
Phase 5: Documentation and Finalization				
Final Performance Review	Final algorithm review; validate Webots vs real-world correlation	Coordinate final review; confirm all deliverables are complete	Final mechanical inspection; tighten all fasteners; clean printed parts	Final electronics audit; inspect all solder joints and connectors
Documentation	Write walking algorithm, IK, ML and simulation chapter	Write literature review, testing and validation chapter; compile full report	Write mechanical design, CAD and 3D printing chapter	Write electronics, GUI and communication chapter
Presentation Preparation	Prepare algorithm and simulation demonstration	Compile full presentation; coordinate submission; present project overview	Prepare mechanical design and fabrication walkthrough	Prepare electronics, GUI and communication demonstration

7.2.3 Effort Distribution and Workload Balance

The project management framework ensures equitable workload distribution through task allocation proportional to the complexity and volume of each domain. Akib's 25% effort spans all mechanical design, CAD modelling and 3D printing — including both PVC and carbon fiber structural iterations and all printed joint components. Tazwar's 25% covers electronics assembly, wiring harnesses, veroboard soldering, power distribution circuitry, the GUI interface, and the wireless communication stack. Shahariar's 25% is the most computationally intensive workstream, encompassing the walking algorithm, inverse kinematics engine, machine learning components, and full simulation of the robot in the Webots simulator. Rafin's 25% underpins the project's intellectual and quality foundations: leading the literature review, coordinating body assembly, running and documenting all testing and validation campaigns, and compiling the complete project report.

This balanced distribution prevents bottlenecks, enables parallel workflow execution across hardware, software, power, and sensing subsystems, and ensures that no single team member carries disproportionate responsibility during high-intensity phases such as Development and Testing.

7.2.4 Time Management

Phase-Level Planning and Weekly Review

Time management across the twelve-month project was governed by phase-level milestones and weekly internal reviews. Each phase boundary was treated as a formal checkpoint, where we reviewed the deliverables collectively before initiating the next phase in full. This structured approach helped us prevent scope creep and ensured that incomplete work from one phase did not silently accumulate into later phases.

Early Procurement Planning

A key strength of our time management was early procurement planning during the FYDP-D phase. We procured the required components before FYDP-C had officially started. For example, we ordered servo motors from China approximately one month before the beginning of FYDP-C. As a result, when FYDP-C began, we did not have to wait for the components to arrive in the country or become available in the local market. This allowed us to start working on the project immediately without unnecessary delays, supporting smoother progress during the development and implementation stage.

Mechanical Redesign Challenge

The most significant time management challenge occurred during Phase 3, when we decided to transition from a PVC frame to a carbon fiber structure, which required a partial rebuild of the mechanical assembly. This introduced an estimated two to three week delay to the development phase. We mitigated the impact by parallelising the structural rebuild with continued firmware development and electronics assembly, ensuring that no phase was fully blocked while the mechanical redesign was in progress.

Early Completion and Real-Life Field Testing

Despite these challenges, we completed all five phases within the twelve-month window. We also had to finish the project at least two to three weeks earlier than the final submission deadline because we needed enough time to test and evaluate the robot at the BRACU substation. Therefore, we completed the main development work early so that we could conduct real-life industrial inspection testing in an actual substation environment. This early completion helped us evaluate the robot's practical performance, identify any final issues, and validate whether the system could operate effectively in a real inspection scenario. Final documentation and the full project report were then submitted in May 2026 as planned, with all technical objectives achieved.

7.2.5 Cost Management

Cost management was one of the most critical aspects of this project. A quadruped robot is inherently expensive to build: it requires multiple high-torque actuators, a rigid and lightweight frame, a high-current power system, real-time control hardware, and various sensors, all of which carry significant individual costs. The project's initial planned budget was approximately BDT 160,000. Through a systematic cost minimization strategy applied at every layer of the design and procurement process, the team brought the final total project cost down to BDT 144,906 which is a saving of approximately BDT 15,094 (9.4%) against the plan. This was achieved through deliberate component selection, iterative fabrication, overseas procurement, laboratory resource reuse, and careful manufacturing practices.

The following subsections describe each cost management decision in detail.

Structural Frame: Iterative Material Strategy

The structural frame of the robot underwent two design iterations, each motivated by cost and performance considerations. In the first iteration, standard PVC pipes were used as the primary structural members. PVC is extremely low in cost, widely available locally, and easy to cut and assemble. This allowed the team to rapidly prototype the leg geometry, validate the joint layout, and confirm the servo mounting positions without investing in expensive materials at an early stage where design changes were likely.

After the zero iteration or initial testing stage, it was revealed that PVC lacked the stiffness required for stable dynamic loading during gait cycles, particularly at the hip and knee joints so we shifted to carbon fiber rods. Carbon fiber provided the necessary structural rigidity and significantly reduced unwanted bending and vibration during locomotion, while remaining lighter than metallic alternatives. The cost of the carbon fiber rods (BDT 9,600, or 6.63% of the total budget) was substantially lower than the cost of CNC-machined aluminium or steel structural components, which would have required external manufacturing services and incurred machining labor costs in addition to material costs. The use of 3D-printed PLA joints (BDT 7,650) complemented the carbon fiber rods, allowing custom brackets and connectors to be fabricated in-house at low cost with rapid design iteration. CNC metal cutting was specifically avoided for this reason.

Actuation System: Servo Motors Over BLDC and Harmonic Drives

The most cost-intensive subsystem in any quadruped robot is the actuation system. Industrial and research-grade quadruped robots typically use BLDC (Brushless DC) motors with planetary gearboxes, or synchronized harmonic drive actuators, both of which provide high torque density, low backlash, and excellent control precision. However, these actuators are prohibitively expensive for an academic prototype. A single harmonic drive actuator suitable for quadruped locomotion can cost several thousand BDT or more, and a full twelve-degree-of-freedom robot would require twelve such units.

The team instead selected high-torque servo motors, specifically the RDS51150 (150 kg·cm) and DS3235 as the primary actuators. These motors provide sufficient torque for the robot's weight and locomotion requirements at a fraction of the cost of BLDC or harmonic drive alternatives. Critically, the team identified that purchasing these servos from local Bangladeshi suppliers would cost approximately BDT 7,000 per unit. By placing an international order, the same servo motors were procured for approximately BDT 3,500 per unit, which was a saving of roughly 50% per servo. Across the full set of actuators required for the robot, this procurement decision produced a substantial reduction in total cost. The servo and actuation category ultimately accounted for BDT 67,810, or 46.80% of the total budget but one that is technically justified by the functional demands of legged locomotion.

Power System: Overseas Battery Procurement

The robot requires multiple LiPo battery packs (3S and 2S configurations) to power the twelve servo motors and onboard electronics simultaneously under high-current conditions. Locally available LiPo batteries of the required capacity were priced at approximately BDT 4,500 per pack. By ordering the batteries from a supplier in China, the team procured the equivalent batteries for approximately BDT 2,500 per pack which is a reduction of approximately 44%. While this involved a longer delivery lead time, the savings were significant enough to justify the wait, and the procurement was timed during the Design phase to ensure arrival before the Development phase began. The total battery and power category accounted for BDT 18,200, or 12.56% of the budget.

Control Platform: ESP32 Over Raspberry Pi

The central control platform of the robot was a key cost decision point. A Raspberry Pi single-board computer is commonly used in robotics research and would have provided a Linux-based computing environment with greater processing power. However, a Raspberry Pi suitable for real-time robotics control costs approximately BDT 6,000 to BDT 8,000 or more, and would have introduced software complexity related to operating system latency that is undesirable for hard real-time servo control.

The team instead selected the ESP32S3 microcontroller as the primary control unit, supplemented by XIAO ESP32C6 and XIAO ESP32S3 Sense modules for wireless communication and camera-based inspection. These microcontrollers provide real-time performance, integrated Wi-Fi connectivity, sufficient computational capacity for inverse kinematics and gait scheduling at the prototype level, and significantly lower cost. The full microcontroller and computing category totaled BDT 12,630, or 8.72% of the total budget which is far below what a Raspberry Pi-based solution would have cost.

Laboratory Resource Reuse: Relay Modules

Where possible, the team reused components already available in the FyDP laboratory rather than purchasing new ones. Relay modules required for power switching in the electrical system were sourced from the laboratory stock rather than purchased commercially. This eliminated the purchase cost of these components entirely and reflects a broader principle of making use of available institutional resources before incurring procurement costs.

Careful Veroboard Soldering

Custom PCB manufacturing was avoided in favor of veroboard (stripboard) construction. Veroboard allows point-to-point circuit assembly at low material cost and does not require external fabrication services. The risk of veroboard construction is that soldering errors can render boards unusable, increasing material waste. The team specifically prioritized careful, methodical soldering practices by planning each circuit layout on paper before soldering, using solder wicks for corrections, and testing each circuit section before committing to final connections. This approach ensured that veroboard panels were not wasted and that the prototype PCB category remained at BDT 1,260, or 0.87% of the total budget.

A summary of the key cost management decisions and the savings achieved in each case is presented in Table 7.2.

TABLE 7.2: COST REDUCTION DECISIONS AND SAVINGS SUMMARY

Component / Decision	Alternative (Avoided)	Chosen Option	Saving Achieved
High-torque servo motors (per unit)	BDT 7,000 (local)	BDT 3,500 (import)	BDT 3,500/unit
LiPo battery packs	BDT 4,500 (local)	BDT 2,500 (import, China)	BDT 2,000/pack
Structural frame	CNC-cut aluminium	PVC pipe to Carbon fiber rods	Significant reduction
Actuation type	BLDC / Harmonic drive motors	High-torque servo motors (150 kg·cm)	Substantial savings
Main control platform	Raspberry Pi 4 (BDT 19000)	ESP32S3 (BDT 2500)	BDT 16,500

PCB fabrication	Custom PCB manufacturing	Veroboard (careful soldering)	Significant reduction
Relay modules	New purchase	Reused from FyDP laboratory	Full cost avoided

The complete budget breakdown, showing all seventeen cost categories and their respective shares of the total project expenditure, is presented in Table 7.3.

TABLE 7.3: COMPLETE BUDGET BREAKDOWN

Cost Category	Included Items	Cost (BDT)	Share (%)
Servo motors and actuation	RDS51150, DS3235, generic 35 kg servo, MG996R	67,810	46.80%
Battery and portable power	3S LiPo, 2S LiPo, 18650 holder, Li-ion cells	18,200	12.56%
Microcontroller and computing	ESP32S3, XIAO ESP32C6, XIAO ESP32S3 Sense	12,630	8.72%
Carbon fiber body structure	25 mm carbon fiber rods	9,600	6.63%
3D printing filament	PLA filament	7,650	5.28%
Mechanical fasteners	Screws, nuts, lock nuts, washers	7,325	5.06%
IMU and balancing sensors	10-DOF IMU modules	4,700	3.24%
Servo extension cables	500 mm servo extension cables	4,500	3.11%
Tools and assembly supplies	Screwdriver set, soldering lead, paste, hacksaw, glue gun	3,150	2.17%
Wiring and connectors	Silicone wires, jumper wires, XT60 connectors	2,163	1.49%
Miscellaneous parts	Enclosure, claw, foot grip materials, cleaning items	1,730	1.19%
Adhesives and mounting	Nanotape, masking tape, electrical tape, glue	1,260	0.87%
Prototype PCB / Veroboard	Veroboard / PCB	1,260	0.87%

Servo drivers	PCA9685 PWM drivers	1,200	0.83%
General electronics	Resistors, capacitors, LEDs, voltage regulators	700	0.48%
Buck converters	LM2596 step-down modules	300	0.21%
Safety and insulation	Fuse, heat-shrink tubing	290	0.20%
Cooling	40 mm fans	238	0.16%
Power distribution board	100A PDB	200	0.14%
Total Project Cost		144,906	100%

Budget utilization analysis, presented in Table 7.4, confirms that 96.87% of the total project cost was spent on components that directly contribute to the robot's functionality. Only 3.13% of the budget was allocated to tools, adhesives, and consumables. This high core-functional spending ratio indicates rational and efficient budget utilization.

TABLE 7.4: BUDGET UTILIZATION METRICS

Metric	Value
Planned Budget	BDT 160,000 (approx.)
Total Actual Project Cost	BDT 144,906
Budget Saving Achieved	BDT 15,094 (~9.4%)
Core Functional Cost	BDT 140,366
Support and Implementation Cost	BDT 4,540
Core Functional Cost Percentage	96.87%
Support Cost Percentage	3.13%
Highest Cost Category	Servo motors and actuation
Highest Category Share	46.80%

7.2.6 Risk Management

7.2.6.1 Risk Assessment with Risk Scoring Method

	Consequence				
Likelihood / Severity	Insignificant(1)	Minor(2)	Moderate(3)	Major(4)	Severe(5)
Almost Certain (5)	M(5)	H(10)	E(15)	E(20)	E(25)
Likely (4)	M(4)	H(8)	H(12)	H(16)	E(20)
Possible (3)	L(3)	M(6)	H(9)	H(12)	E(15)
Unlikely (2)	L(2)	M(4)	M(6)	M(8)	H(10)
Rare (1)	L(1)	L(2)	L(3)	M(4)	H(5)

Legend: L = Low, M = Medium, H = High, E = Extreme

7.2.6.2 Hazard Register with Root Causes and Pre-/Post-Mitigation Scores

Result = Frequency × Consequences. Mitigations are written in plain English for engineers, operators, and supervisors.

TABLE 7.5: THE HAZARD REGISTER WITH ROOT CAUSES AND PRE-/POST-MITIGATION SCORES

Risk Identify	**Hazard**	**Pre-Mitigation Score**			**What We Should Do (mitigation and management strategy)**	**Post-Mitigation Score**		
		Likelihood	**Consequences**	**Result**		**Likelihood**	**Consequences**	**Result**
Risks to people during interaction	Collision with a person	3	4	**12**	Mark robot routes, limit speed, use safety sensors, and train staff to report near-misses.	2	3	**6**
	Robot tips or falls onto a person	2	5	**10**	Avoid risky surfaces, reduce speed on ramps/stairs, and define no-go zones near edges.	1	4	**4**
	Pinch, crush, or shear from the arm	3	4	**12**	Use soft covers, rounded tools, force limits, keep-out zones, and hold-to-run control.	2	3	**6**

	Unintended or unstable motion	2	5	**10**	Add emergency stop, safe-stop braking, watchdogs, and test changes before site release.	1	4	**4**
	Ignition in a hazardous area	2	5	**10**	Enter only certified zones, monitor gas levels, control temperature, and follow work permits.	1	4	**4**
	Nuisance alarms that distract staff	2	2	**4**	Set minimum alarm levels, define quiet hours, and silence alarms only when safe.	1	2	**2**
Operational and asset risks	Battery fire or thermal runaway	2	5	**10**	Use certified batteries, charge only within limits, and keep fire protection nearby.	1	4	**4**
	Interference causes missed defects	3	4	**12**	Route cables properly, use shielding, add sensor redundancy, and rescan doubtful readings.	2	3	**6**
	Sensor failure or drift	4	4	**16**	Calibrate regularly, cross-check sensors, replace aged sensors, and validate detections.	2	3	**6**
	Loss of communications or mapping	3	3	**9**	Use safe-stop, return-to-home, dual links if possible, and refresh maps regularly.	2	2	**4**
	Hazards at the docking station	3	3	**9**	Keep docking area clear, manage cables, use guarded charging, and add warning signs.	2	2	**4**
	Slips on floors or cables	3	3	**9**	Detect slippery surfaces, slow down, use proper foot pads, and avoid unsafe routes.	2	2	**4**
	Delay causes contact or drops	3	3	**9**	Set latency limits, slow motion near obstacles, and ensure stable network quality.	2	2	**4**
	Misuse of tool or payload upsets a process	2	4	**8**	Use checklists, limit torque/speed, require confirmation, and interlock with controls.	1	3	**3**

Cybersecurity, data, and privacy	Unauthorized control or data tampering	3	5	**15**	Use protected networks, multi-factor login, access logs, updates, and restricted remote access.	2	3	**6**
	Privacy or data leakage	3	3	**9**	Minimize data collection, blur IDs/faces, store data securely, and delete on schedule.	2	2	**4**
Organizational, financial, and social	Job displacement without support	3	3	**9**	Provide reskilling, define new roles, involve workers, and share productivity results.	2	2	**4**
	Higher total cost than expected	3	3	**9**	Run a pilot, set ROI targets, budget spares, and plan battery lifecycle costs.	2	2	**4**
	Reputational or regulatory harm	2	4	**8**	Maintain safety records, perform audits, run drills, and prepare incident communication.	1	3	**3**

Interpretation

Explanation	Average	Risk Score
Before mitigation, the sum of risk values for 18 hazards was 186, resulting in an average pre-mitigation risk (R_pre) of 10.33. This value falls into the orange band, indicating a high-risk state. After applying the described design, operational, and cybersecurity measures, the sum of the post-mitigation risk values was reduced to 80, bringing the average residual risk (R_post) down to 4.44. This reduced risk level is acceptable, as it falls between the green and yellow bands. The next step is to proceed with a bounded pilot, collecting near-miss data, and updating the risk register and safety case accordingly	$\bar{R} = \frac{\sum_i R_i}{N}$	$R_i = F_i \times C_i$

7.2.6.3 Contingency Plan for Hazards

TABLE 7.6: TABLE OF THE CONTINGENCY PLAN FOR HAZARDS

Risk	Mitigation Plan (Prevention)	Contingency Plan: Trigger	Contingency Plan: Response Actions
Crush or Pinch (Arm/Joints)	Soft padding on joints; "Hold-to-run" dead-man switch; force-limiting motors.	Resistance detected or a person yells "Stop."	Immediate E-Stop; manually back-drive the joint to release; check for injury.
Robot Tipping / Falling	Training on center-of-gravity; mechanical outriggers; speed limits on turns.	Chassis tilt > 20° or visual "wheel lift."	Drop payload/arm to ground instantly to lower center of mass; cut drive power.
Battery Fire / Thermal Runaway	Certified battery casing; charging in a fire-safe zone; pre-shift visual inspection.	Smoke, hissing, or acrid smell from the battery box.	Use Class D extinguisher; evacuate area; notify fire marshal; do not use water.
Loss of Control (Signal Lag)	5GHz dedicated Wi-Fi; heartbeat signal check; shielded cables for tethered units.	Latency > 200ms or "Control Lost" alert on screen.	The robot enters "Brake Lock"; operator waits for signal recovery or uses physical tether.
Collision (Operator Error)	High-visibility paint; rearview cameras; spotter required for tight spaces.	Audible impact or camera feed shows contact.	Hit E-Stop; inspect robot and environment for damage; file incident report.
Hazardous Area Ignition	Grounding straps for static; sealed electronics; gas sniffer on the robot.	External Gas Alarm or spark observed at robot.	Kill main breaker; manually tow the robot out of the hazard zone using a tow-bar.
Electrical Failure / Short	Fused circuits; wire loom protection; moisture-resistant connectors.	Burning smell or erratic motor "twitching."	Disconnect battery/power supply immediately; tag out the robot for maintenance.
Motor Burn	Overload protection circuits; thermal sensors; regular motor maintenance checks.	High motor temperature detected or "Motor Overheat" alert.	Immediately shut down the motor; allow it to cool; inspect motor for damage; replace or repair motor if necessary.

7.3 Evaluate Project Progress

The overall project progress can be grouped into three broader phases: the Planning phase (P), the Design and Development phase (D), and the Construction and Implementation phase (C). These phases provide a clear overview of how the team's effort was distributed throughout the project.

P Phase: Planning (15–20%)

During the Planning phase, the team mainly focused on selecting the project topic, reviewing relevant literature, defining the problem statement, and preparing the initial proposal. No major construction, fabrication, or coding was performed at this stage. For the quadruped robotic dog, this phase included studying existing quadruped platforms, understanding legged locomotion, identifying inspection-related requirements, and setting realistic objectives based on the available budget, timeline, and technical capability. Although this phase required the least physical workload, it was essential because it established the project direction and created the foundation for the later design and implementation stages.

D Phase : Design and Development (Approximately 30%)

The Design and Development phase involved more technical work, including CAD modelling, simulation, inverse kinematics development, gait planning, and control strategy evaluation. Webots was used to simulate the quadruped model and test the walking behaviour before physical construction. The team also developed and compared manual and autonomous control approaches, along with gait patterns such as crawling and trotting. This phase helped identify practical design requirements and performance limitations before moving to hardware implementation. Since this stage involved simulation, algorithm development, design validation, and machine learning preparation, it accounted for approximately 30% of the total project effort.

C Phase : Construction and Implementation (Approximately 50%)

The Construction and Implementation phase required the highest workload because the simulated design had to be converted into a functional physical prototype. This phase included 3D printing, mechanical assembly, electronics integration, wiring, servo calibration, sensor installation, GUI connection, and full system testing. The team also carried out several design improvements, such as replacing the initial PVC-based structure with stronger materials, using carbon fiber rods, redesigning PLA joints, improving the shoulder structure, and modifying the foot and claw mechanisms. Cost-control decisions were also made during this phase, including reusing available components, sourcing servos and batteries at lower prices, using ESP32-based controllers instead of more expensive alternatives, and carefully soldering veroboards to reduce material waste. Due to the amount of fabrication, redesign, testing, troubleshooting, and validation involved, this phase represented approximately 50% of the total project effort.

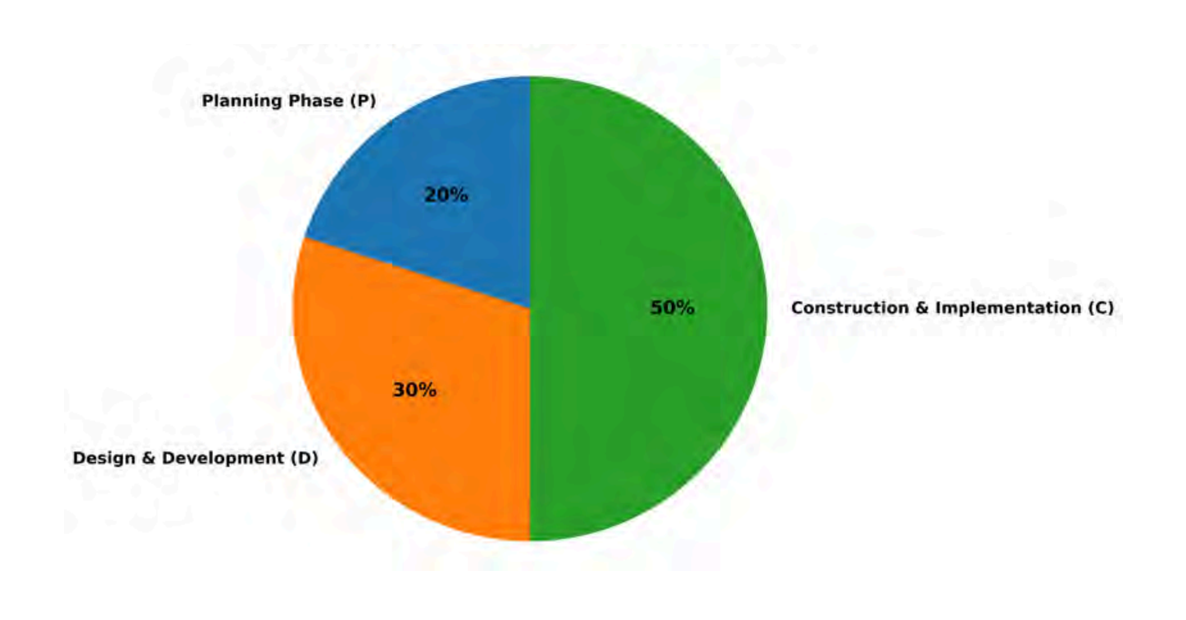


Figure 7.1: Project Progress Pie Chart for each phases

7.3.2 Project Progress Gantt Chart for FYDP-C:

The project progress was successfully maintained according to the planned weekly schedule due to effective project management practices carried out during the FYDP-D phase. This structured planning helped the team complete each activity on time and keep the overall project development aligned with the intended timeline.

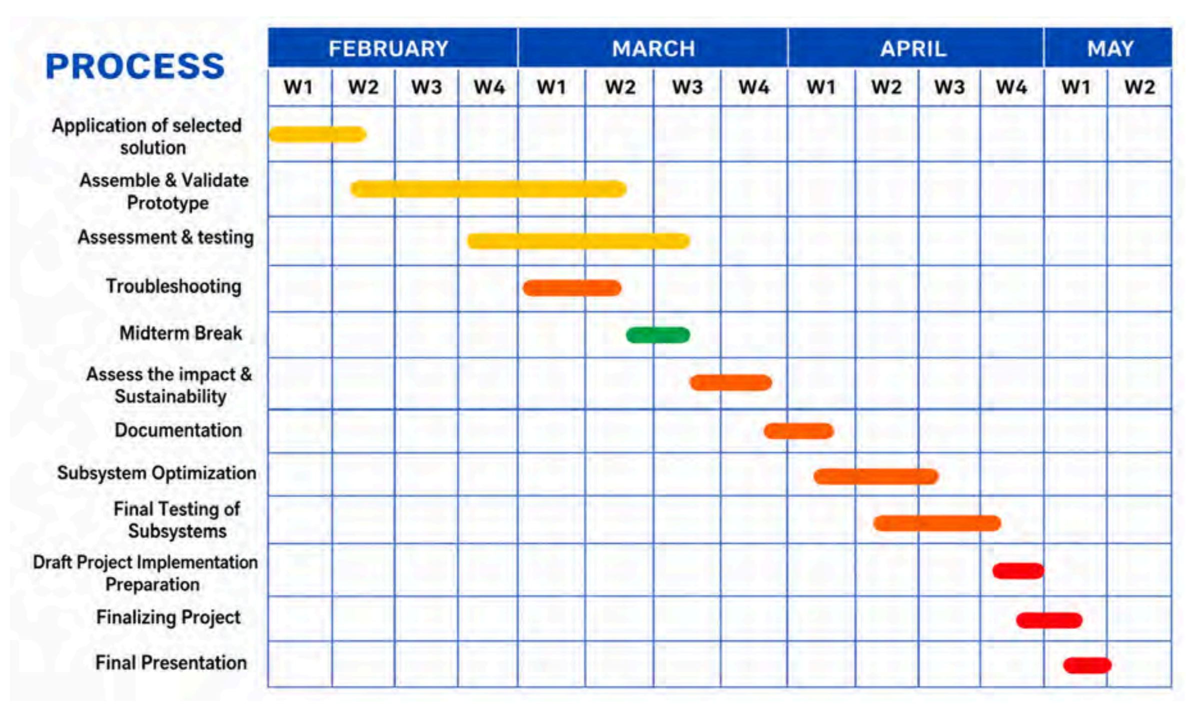


Figure 7.2: Gantt Chart

7.3.3 Milestone Achievement and Deliverable Quality

Each phase concluded with a formal milestone review assessing deliverable completeness and quality against predefined acceptance criteria before the next phase commenced.

Phase 1 Milestone: Confirmed problem statement clarity, selected system architecture, completed initial concept note. Acceptance criterion: all team members aligned on design approach and scope.

Phase 2 Milestone: System architecture finalized, all major components ordered, software framework initiated. Acceptance criterion: procurement confirmed, no major design ambiguities outstanding.

Phase 3 Milestone: Fully assembled quadruped robot (second iteration, carbon fiber structure) with all twelve servos installed, firmware operational, electronics wired and tested. Acceptance criterion: robot can execute basic joint movement under software control.

Phase 4 Milestone: Gait stability, power endurance, wireless latency, IMU feedback, and camera stream all confirmed within target ranges. Acceptance criterion: robot walks continuously for at least 30 seconds without failure.

Phase 5 Milestone: Complete project report compiled, all chapters reviewed and finalized, formal presentation prepared. Acceptance criterion: report submitted by May 2026 deadline.

All five milestones were achieved within the respective phase timelines. The structural redesign in Phase 3 was the most significant challenge, but it was resolved internally without escalating to a phase-level failure. The final delivered system met all nine KPIs listed in Table 7.5, and the total project cost of BDT 144,906 remained within the planned budget ceiling of BDT 150,000.

7.3 Conclusion

This chapter presented the project management framework for the quadruped robotic dog project, covering planning, team organisation, time, cost, risk, contingency, and progress evaluation from June 2025 to May 2026. The project was managed through three main areas: time, cost, and risk management. Time management divided the work into five phases, supported regular milestone reviews, and helped complete major development before final substation testing. Cost management reduced the planned budget from about BDT 160,000 to an actual cost of BDT 144,906, saving BDT 15,094 through careful procurement, material selection, component reuse, ESP32-based control, and efficient fabrication. Risk management addressed hazards such as collision, tipping, pinch points, LiPo battery fire, motor overheating, communication loss, and electrical faults. Mitigation measures included emergency stop, safe shutdown, operator training, speed limits, wiring checks, and battery safety procedures. Overall, the five-phase plan, milestone tracking, cost control, and risk management strategy helped complete the project within the academic timeline and budget.

Chapter 8: Economical Analysis. [CO12]

8.1 Introduction

This chapter presents the economic analysis of the proposed quadruped robotic dog project by evaluating its market relevance, budget structure, cost-benefit potential, and long-term financial feasibility. The analysis considers both global and domestic market growth, showing how inspection robotics and quadruped robots are becoming increasingly important for industrial automation, hazardous-environment inspection, predictive maintenance, and worker safety. It also examines the updated project cost of BDT 144,906, budget utilization, annual benefits, ROI, payback period, NPV, BCR, sensitivity analysis, and risk-adjusted performance to determine whether the prototype is economically reasonable for academic development and future industrial adaptation.

8.2 Economic analysis

8.2.1 Global Market Overview

The proposed quadruped robotic dog project is positioned within the rapidly growing global market for inspection robots, service robotics, and legged robotic platforms. The demand for such systems is increasing because industries are moving toward remote inspection, predictive maintenance, worker safety improvement, and automation of hazardous tasks.

The global inspection robots market was valued at approximately USD 2.8 billion in 2024 and is projected to grow at a 13.9% CAGR (Compound Annual Growth Rate) from 2025 to 2034 [37]. Another market estimate reports the global inspection robots market at USD 3.33 billion in 2024, increasing to USD 13.58 billion by 2033, with a projected 16.9% CAGR (Compound Annual Growth Rate) [28]. Although these values differ slightly because of differences in market scope and methodology, both sources show strong growth in inspection robotics. This is important for the proposed project because the robot's core purpose—real-time inspection in hazardous industrial environments—falls directly within this expanding market segment.

The quadruped robot market itself is also expanding at a high rate. Strategic Market Research estimates that the global quadruped robot market will increase from USD 0.7 billion in 2024 to USD 4.1 billion by 2030, representing a projected 32.5% CAGR (Compound Annual Growth Rate) [39]. Another estimate values the quadruped robot market at USD 1.371 billion in 2023 and projects it to reach USD 4.419 billion by 2030, with an expected 18.2% CAGR (Compound Annual Growth Rate) from 2024 to 2030 [40]. These projections indicate that legged robots are moving from research prototypes toward practical use in industry, defense, rescue, inspection, and security applications. For this project, the high growth of the quadruped robot market strengthens the justification for developing a low-cost academic prototype that can demonstrate legged mobility, inspection capability, wireless control, and limited responsive manipulation.

The broader professional service robotics market also supports the project's industrial relevance. The inspection and maintenance segment of professional service robotics generated approximately USD 2.7959 billion in 2024 and is expected to reach USD 5.485 billion by 2030, growing at a 11.7% CAGR (Compound Annual Growth Rate). The same source identifies Asia Pacific as the largest revenue-generating region for this segment in 2024 [41]. Since Bangladesh is located within the Asia-Pacific region and is gradually increasing industrial automation adoption, this regional trend is relevant for the future domestic adoption of inspection robots.

From a market perspective, the global growth trend shows that inspection robots are becoming economically and operationally important. Industries are increasingly adopting robots for infrastructure inspection, plant monitoring, predictive maintenance, hazardous-area inspection, and remote data collection. Therefore, the proposed robotic dog is aligned with a global shift toward intelligent, mobile, and safety-oriented inspection systems.

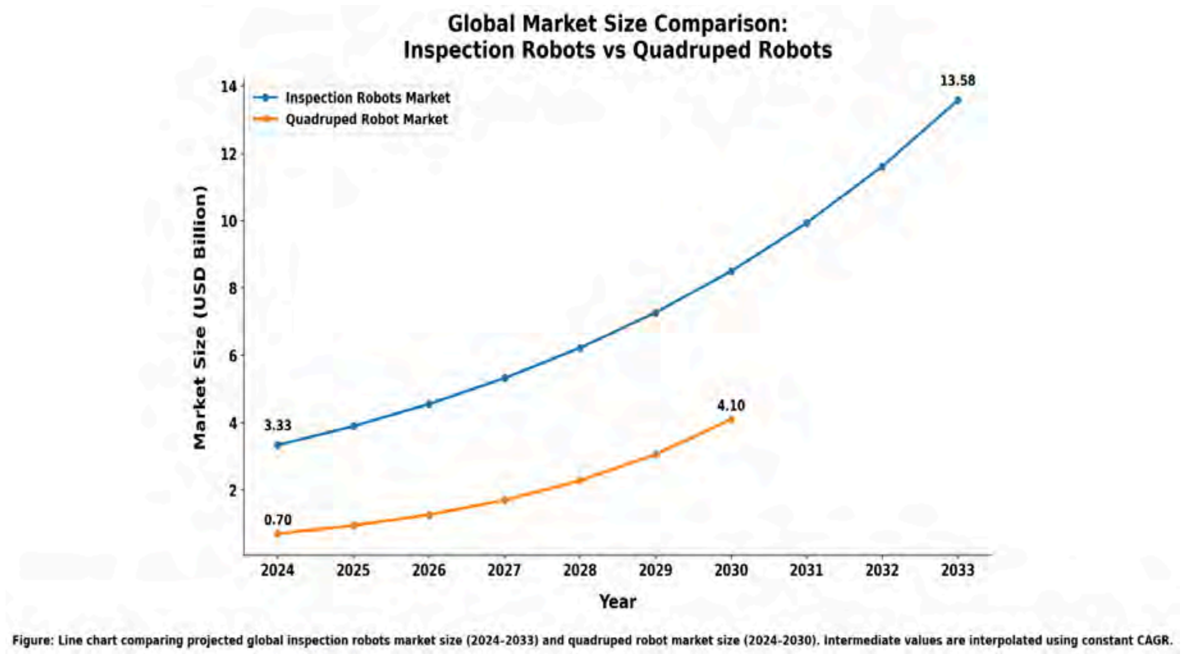


Figure 8.1: Inspection Vs Quadruped Robot Market

8.2.2 Domestic Market Overview: Bangladesh Context

Although Bangladesh does not yet have a mature commercial market specifically for quadruped inspection robots, the domestic market environment is becoming increasingly favorable for robotics and automation. The strongest evidence comes from the growth of factory and industrial automation. The Bangladesh factory and industrial automation market is projected to grow at a 10.2% CAGR from 2022 to 2028 [42]. This growth is driven by increasing factory development in economic zones, rising demand for industrial productivity, and the adoption of automation technologies in manufacturing sectors.

The textile sector is especially important for Bangladesh's automation market. Industry reports identify textiles as the highest revenue-share segment within Bangladesh's factory and industrial automation market [42]. This is significant because Bangladesh's industrial economy is highly dependent on manufacturing and export-oriented production. As factories modernize, demand will increase not only for production automation but also for inspection, monitoring, safety, and maintenance technologies. A quadruped robotic dog can serve this future need by providing mobility in factory floors, substations, warehouses, utility areas, and hazardous zones where wheeled robots may face limitations.

Automation adoption in Bangladesh's garment sector further indicates domestic readiness for robotics-based solutions. According to a report cited by Vogue Business, 80% of factories in Bangladesh planned to purchase automated machines within two years, compared with 28% in 2021. The same report states that Bangladesh's garment industry employs more than 4 million workers and accounts for more than 80% of national export revenue [43]. These figures show that automation is no longer only a future concept in Bangladesh; it is already becoming part of industrial planning. However, most automation adoption is currently focused on production machines rather than mobile inspection robots. This creates a potential market gap for low-cost inspection and safety robotics.

In the domestic context, the proposed robotic dog is economically relevant because it provides a lower-cost pathway toward robotic inspection. The prototype cost is only BDT 144,906, which is much more accessible for academic research, laboratory testing, and small-scale industrial demonstration than imported commercial quadruped systems. This cost advantage is important for Bangladesh, where many factories and utility facilities may not be ready to invest in expensive advanced robotic platforms. A locally developed, modular, and repairable quadruped robot can therefore act as a bridge between conventional manual inspection and fully commercial robotic inspection systems.

The domestic opportunity is strongest in sectors where inspection involves safety risks, repeated manual effort, or difficult access. These include electrical substations, factories, warehouses, chemical storage areas, construction sites, power plants, and industrial utility zones. Since Bangladesh's automation market is growing and industries are gradually adopting smart manufacturing practices, the proposed project has practical relevance as an early-stage prototype for local industrial inspection robotics.

8.2.3 Market Relevance to the Proposed Robotic Dog

The market data indicate that the proposed robotic dog is relevant from both global and domestic perspectives. Globally, inspection robots and quadruped robots are growing rapidly because industries require safer, faster, and more intelligent inspection systems. Domestically, Bangladesh's industrial automation market is expanding, and factories are increasingly open to automation technologies. However, there remains a gap in affordable mobile inspection robots suitable for local industrial conditions.

The proposed project addresses this gap through a cost-effective quadruped design. Its main market advantage is not that it immediately competes with premium commercial robots, but that it offers a practical low-cost platform for research, testing, and future industrial adaptation. The robot can be improved gradually by upgrading sensors, adding stronger actuators, improving autonomous navigation, and integrating AI-based fault detection. Therefore, the project has both immediate academic value and long-term market potential.

From an economic analysis perspective, the growing market supports the feasibility of the project in three ways. First, global demand confirms that inspection robotics is a valid and expanding engineering field. Second, domestic automation growth suggests that Bangladesh is moving toward greater use of robotic and automated systems. Third, the low prototype cost makes the system more suitable for local experimentation and future commercialization than high-cost imported alternatives.

8.2.4 Complete Budget Breakdown

The economic feasibility of the proposed quadruped robotic dog depends on whether the system can deliver reliable inspection, mobility, sensing, control, and limited responsive operation at a cost that is significantly lower than commercial quadruped inspection robots. The updated budget shows that the prototype was developed using modular electronic components, high-torque servo motors, ESP32-based control boards, IMU sensors, LiPo batteries, 3D-printed structural parts, carbon-fiber reinforcement, and locally sourced mechanical and assembly materials.

According to the updated final budget, the total project cost is BDT 144,906. This value is used as the baseline capital investment for the economic analysis. The cost is distributed across actuation, power, control, sensing, structural fabrication, wiring, assembly, safety, and support materials. Among these, the largest cost is associated with the servo motor and actuation system, which is expected because a quadruped robot requires multiple load-bearing actuators for walking, posture control, and manipulation.

Full Itemized Bill of Materials (BOM)

TABLE 8.1: TABLE OF THE FULL ITEMIZED BILL OF MATERIALS (BOM)

Item Category	Specific Item	Estimated Cost (BDT)	Quantity	Total Cost (BDT)
Microcontroller	Waveshare ESP32S3-NXR8 Development Board	1460	4	5840
	XIAO-ESP32C6	1300	2	2600
	XIAO-ESP32S3 Sense with Camera Kit	4190	1	4190
IMU Sensor	Fermion: 10 DOF IMU Sensor - ADXL345+ITG3205+VCM5883L+BMP280 Breakout	2350	2	4700
Power Distribution Board	100A PDB	100	2	200
Screw	M4	5	250	1250
Nuts	M4	3.5	250	875
	M4 Lock NUT	12	100	1200
	M3	5	100	500
	M3 Lock Nut	10	100	1000
Washer	M4	5	250	1250
	M3	5	250	1250
Servo Motors	High-Torque Standard Servo (RDS51150)	3500	17	59500
	DS3235 35KG Digital Servo Metal Gear Waterproof High Torque	2930	2	5860
	Generic 35KG Servo	1750	1	1750
	MG996R Metal Servo	350	2	700
Servo Extension Cable	500mm Servo Extension Cable	90	50	4500
Buck Converter	DC-DC Buck LM2596 DC-DC Buck Converter Step Down Module	75	4	300
Wiring/Cables	Silicone Hook-up Wire (22 AWG)	42	4	168
	Silicone Hook-up Wire (12 AWG)	160	2	320
	Assorted Jumper Wires (M/F, M/M, F/F)	2.5	250	625
	Jumper Wire Box	190	3	570
	XT60 Connector	60	8	480
Servo Drivers	PCA9685 16-Channel PWM Driver	600	2	1200
Battery	Li-Power 3S 11.1V 35C	3500	4	14000
	Li-Power 2S 7.4 25C	2750	1	2750
	18650 2 Slot Battery Holder with BMS Charger	350	1	350

	Panasonic High Grade Battery Li-ion Battery 3.7V 3400 mAh	550	2	1100
Cooling	40mm Fan	119	2	238
Adhesive	Nanotape	180	1	180
	Masking Tape	100	2	200
	Electrical Tape	30	4	120
	Double sided tape	150	1	150
	Hunter Tape	250	1	250
	SuperGlue	40	4	160
	Glue Stick	30	15	450
Electronics (General)	Resistor/Capacitor Assortment Kit	15	20	300
	3W SMD LED High Power	30	10	300
	Voltage Regulator (LM7805)	25	4	100
Veroboard/PCB	FR4 Single-Sided Prototype Board	630	2	1260
3D Printer Filament	Sunlu PLA Filament (1kg Spool, White)	1500	4	6000
	Elego PLA Filament (1kg Spool, White)	1650	1	1650
Body Structure	25mm Diameter Carbon Fiber Rod (1m)	2400	4	9600
Tools/Supplies	Precision Screwdriver Set	1100	1	1100
	Soldering Lead	350	2	700
	Soldering Paste	250	2	500
	Hacksaw	500	1	500
	Glue Gun	350	1	350
Protection	Car Fuse	10	25	250
	Heat Shrink Tubing Assortment	20	2	40
Miscellaneous	Project Enclosure/Case/PVC Board	500	1	500
	Claw	750	1	750
	Car Mat	75	2	150
	Kitty Ball	25	8	200
	Hexisol	55	1	55
	Nail polish remover (Acetone)	55	1	55
	Brush	20	1	20
Total Amount in BDT				**144906**

Pie Chart of Budget Distribution for the Quadruped Robotic Dog Project:

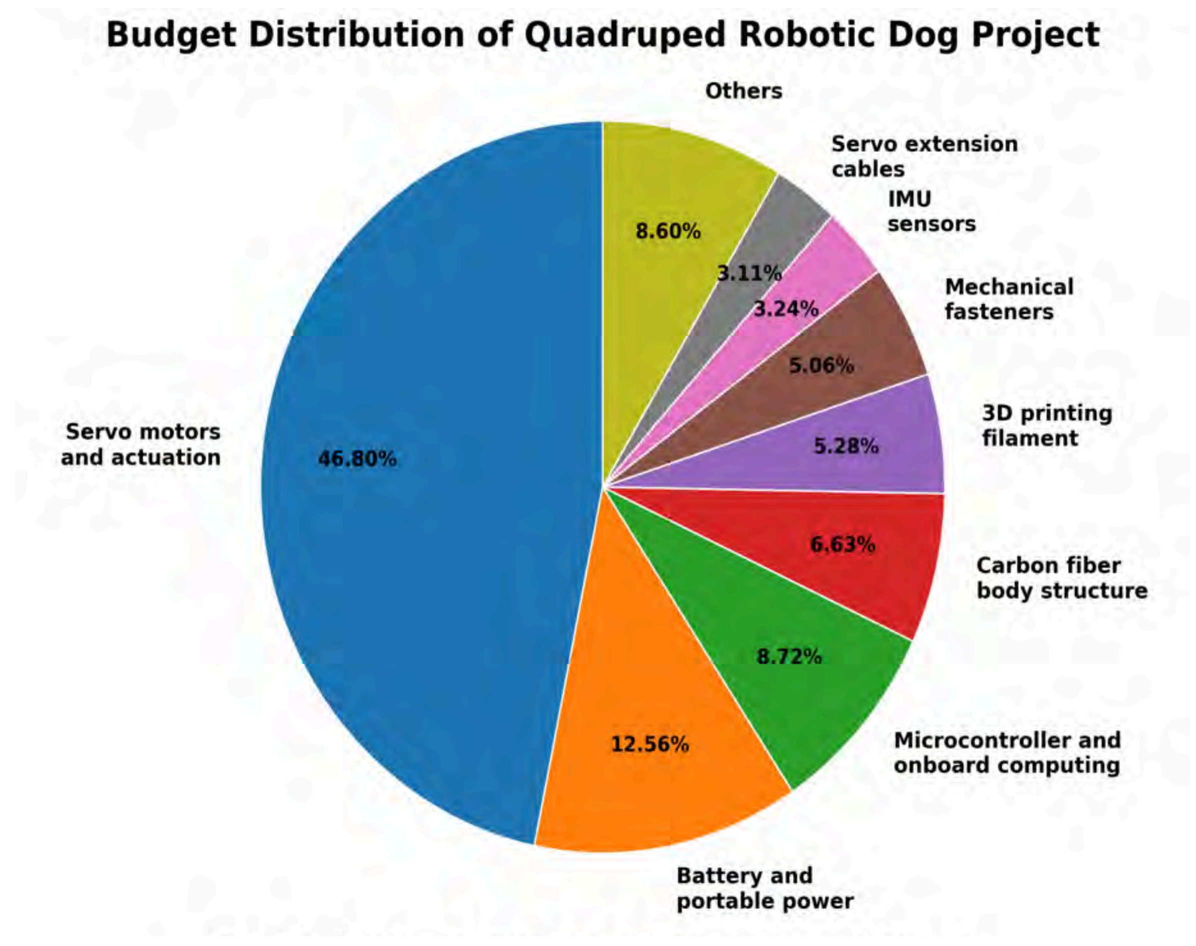


Figure 8.2: Budget As Categorical Percentage Distribution

The budget distribution shows that the project is primarily actuation-intensive. The servo motor and actuation category alone accounts for 46.80% of the total budget. This is technically reasonable because quadruped locomotion requires several high-torque motors to support the robot's weight, execute gait cycles, maintain posture, and handle dynamic loading during walking. Unlike wheeled robots, a quadruped platform cannot rely on simple rotational drive alone; each leg must generate coordinated joint movement. Therefore, the high allocation to servo motors is justified by the mechanical and functional requirements of the system.

The second largest category is the battery and portable power system, which accounts for 12.56% of the total cost. This is also technically justified because the robot must operate several servo motors, control boards, sensors, and communication modules simultaneously. During walking and manipulation, the motors may draw high current, so the use of LiPo batteries and supporting power components is necessary to maintain stable operation.

The microcontroller and onboard computing category represents 8.72% of the budget. This includes ESP32-based boards and the XIAO ESP32S3 Sense module with camera capability.

This cost is rational because the robot requires real-time control, firmware development, wireless communication, sensor processing, and camera-based inspection support. Instead of using a high-cost embedded computer, the project uses affordable microcontroller-based platforms, which improves cost efficiency while maintaining sufficient functionality for prototype-level development.

The structural portion of the budget, including carbon fiber rods and PLA filament, accounts for 11.91% of the total cost. This reflects a balanced mechanical design strategy. Carbon fiber improves stiffness and reduces unwanted bending, while PLA-based 3D printing allows low-cost custom fabrication and rapid design iteration. This combination is appropriate for an academic quadruped robot because it reduces manufacturing cost while preserving structural flexibility and repairability.

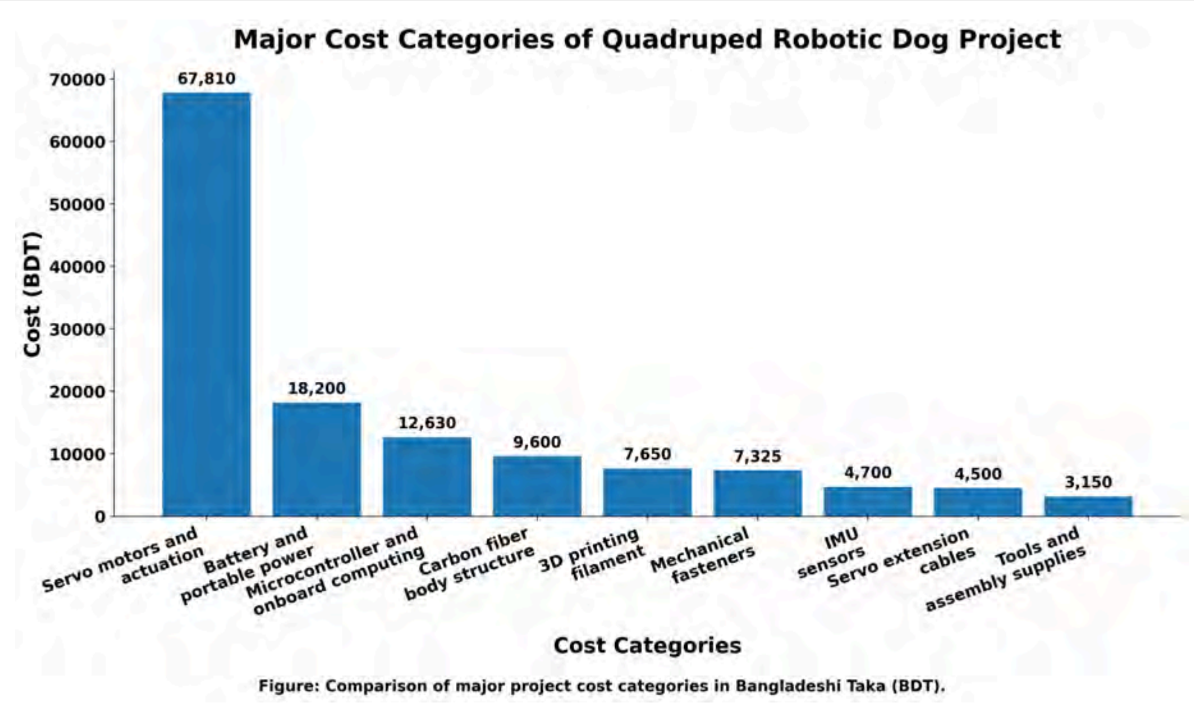


Figure 8.3: Major Cost Categories of Project

8.2.5 Budget Utilization Metrics

Budget utilization analysis is necessary to determine whether the project expenditure was concentrated on technically meaningful components or whether excessive spending occurred in non-functional areas. For a robotics project, the most important spending areas should be actuation, control, power, sensing, structural fabrication, and connectivity. In the updated budget, most of the total expenditure directly supports these essential engineering functions.

For this analysis, the budget is divided into two categories:

Core functional cost includes components that directly contribute to robot operation, such as servo motors, batteries, microcontrollers, IMU sensors, servo drivers, wiring, connectors, structural materials, fasteners, cooling, safety components, and power electronics. Support and implementation cost includes tools, adhesives, cleaning materials, and assembly-related consumables that are required for fabrication and maintenance but are not permanent functional subsystems of the robot.

The support and implementation cost is calculated as:

Here, cleaning consumables include Hexisol, acetone, and brush:

$$\text{Cleaning Consumables} = 55+55+20 = 130 \text{ BDT}$$

$$\text{Support Cost} = 3{,}150+1{,}260+130 = 4{,}540 \text{ BDT}$$

Therefore, the core functional cost is:

$$\text{Core Functional Cost} = 144{,}906-4{,}540 = 140{,}366 \text{ BDT}$$

The percentage of the total budget used for core functional components is:

$$\text{Core Functional Cost Percentage} = \frac{140{,}366}{144{,}906} \times 100 = 96.87\%$$

The support cost percentage is:

$$\text{Support Cost Percentage} = \frac{4{,}540}{144{,}906} \times 100 = 3.13\%$$

Table for Budget Utilization Metrics

TABLE 8.2: TABLE FOR BUDGET UTILIZATION METRICS

Metric	Value
Total project cost	BDT 144,906
Core functional cost	BDT 140,366
Support and implementation cost	BDT 4,540
Core functional cost percentage	96.87%
Support cost percentage	3.13%
Highest cost category	Servo motors and actuation
Highest category share	46.80%

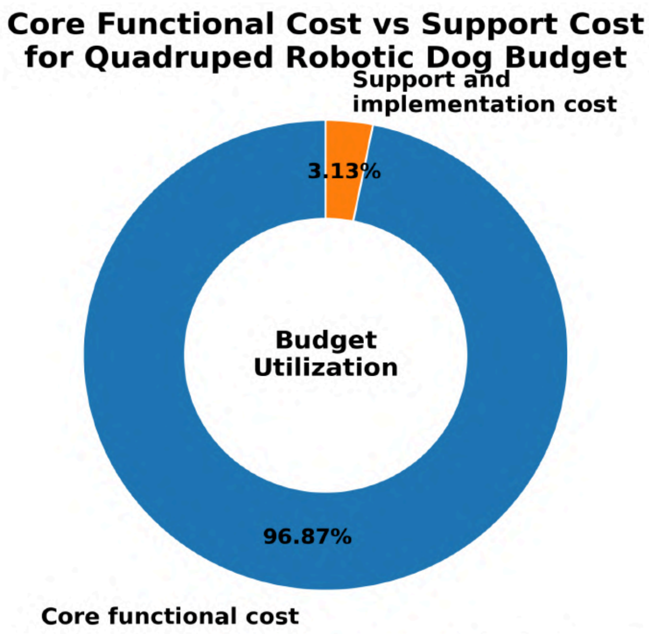


Figure 8.4: Core Functional Cost vs Support Cost

The result shows that the budget was used efficiently because 96.87% of the total cost was spent on components that directly contribute to the robot's functionality. This is a strong indicator of rational budget utilization. The spending is not concentrated on decorative, unnecessary, or non-engineering items; instead, the majority of the budget is allocated to actuation, power, sensing, control, structure, and connectivity.

This budget pattern is technically appropriate for a quadruped robot because the mechanical and electrical demands of legged locomotion are much higher than those of simple wheeled platforms. The robot requires multiple high-torque motors, stable power delivery, rigid but lightweight structural components, and real-time control hardware. Therefore, the budget allocation supports the project's functional goals rather than merely increasing cost.

8.3 Cost-Benefit Analysis

8.3.1 Benefit Estimation Framework

The economic value of the quadruped robotic dog is not limited to direct financial revenue. Its value comes from reducing inspection risk, improving operational efficiency, minimizing human exposure to hazardous environments, supporting early fault detection, and providing a low-cost alternative to commercial quadruped inspection systems.

The project is intended for real-time inspection and responsive operation in hazardous or difficult industrial environments. In such environments, manual inspection may expose workers to high voltage, toxic gases, heat, confined spaces, uneven surfaces, and structurally unsafe areas. Therefore, the economic benefit of the robot must be evaluated through both direct and indirect benefits.

The main benefit categories are:

Labor efficiency benefit:
The robot can reduce the number of workers required to physically enter inspection zones. Human operators may still supervise the system remotely, but the robot reduces repetitive manual inspection effort.

Downtime reduction benefit:
Manual inspection may require partial shutdowns, safety clearance, or restricted access. A remotely operated or semi-autonomous quadruped robot can inspect certain areas with less interruption.

Safety and risk reduction benefit:
The robot reduces direct human exposure to hazardous industrial conditions. This benefit is economically important because workplace incidents can result in medical cost, compensation, investigation delays, productivity loss, and reputational damage.

Preventive maintenance and early fault detection benefit:
The robot's camera and sensor-based inspection capability can help detect cracks, corrosion, loose fittings, overheating, or other early warning signs. Early identification of such problems can reduce future repair cost and unplanned failure.

Since this is an academic prototype, the benefit estimation uses conservative assumptions. The goal is not to claim exact industrial profit, but to demonstrate that the project can be economically reasonable if deployed in a small industrial facility, substation, workshop, or plant inspection environment.

8.3.2 Annual Benefits Calculation

The following assumptions are used for base-case annual benefit calculation:

TABLE 8.3: TABLE FOR THE ANNUAL BENEFITS CALCULATION

Parameter	Assumption
Number of inspection sessions per month	4
Manual inspection team size	2 workers
Time required per manual inspection	4 hours
Effective skilled labor cost per worker-hour	BDT 250
Avoided operational downtime	2 hours per month

Estimated downtime cost per hour	BDT 2,000
Annual safety/risk reduction value	BDT 30,000
Annual preventive maintenance value	BDT 40,000
Annual operation and maintenance cost	BDT 20,000

A. Labor Efficiency Benefit

$$\text{Annual Labor Benefit} = \text{Workers} \times \text{Hours per Inspection} \times \text{Inspections per Month}$$

$$= 2 \times 4 \times 4 \times 250 \times 12$$

$$= 96,000 \text{ BDT/year}$$

Therefore, the estimated annual labor efficiency benefit is BDT 96,000.

B. Downtime Reduction Benefit

$$\text{Annual Downtime Benefit} = \text{Avoided Downtime per Month} \times \text{Downtime Cost per Hour} \times 12$$

$$= 2 \times 2,000 \times 12$$

$$= 48,000 \text{ BDT/year}$$

Therefore, the estimated annual downtime reduction benefit is BDT 48,000.

C. Safety and Risk Reduction Benefit

A conservative annual value of BDT 30,000 is assigned for safety and risk reduction. This represents the economic value of reducing worker exposure to hazardous inspection areas. Although safety improvement is difficult to quantify precisely, it is highly relevant for this project because the robot is designed specifically to reduce human entry into risky industrial environments.

D. Preventive Maintenance Benefit

A conservative annual value of BDT 40,000 is assigned for preventive maintenance and early fault detection. If the robot helps identify one small equipment fault, structural crack, overheating issue, or loose mechanical connection before it develops into a larger failure, this value can be justified

E. Gross Annual Benefit

$$\text{Gross Annual Benefit} = 96,000 + 48,000 + 30,000 + 40,000$$

$$= 214,000 \text{ BDT/year}$$

F. Net Annual Benefit

The robot requires an annual operation and maintenance cost. This includes servo wear, battery degradation, wire replacement, structural repair, fastener replacement, and general servicing. A conservative value of BDT 20,000 per year is assumed.

$$\text{Net Annual Benefit} = \text{Gross Annual Benefit} - \text{Annual O\&M Cost}$$
$$= 214,000 - 20,000$$
$$= 194,000 \text{ BDT/year}$$

Annual Benefit Estimation:

TABLE 8.4: TABLE FOR THE ANNUAL BENEFIT ESTIMATION

Benefit Category	Estimated Annual Value
Labor efficiency benefit	BDT 96,000
Downtime reduction benefit	BDT 48,000
Safety and risk reduction benefit	BDT 30,000
Preventive maintenance benefit	BDT 40,000
Gross annual benefit	BDT 214,000
Annual operation and maintenance cost	BDT 20,000
Net annual benefit	BDT 194,000

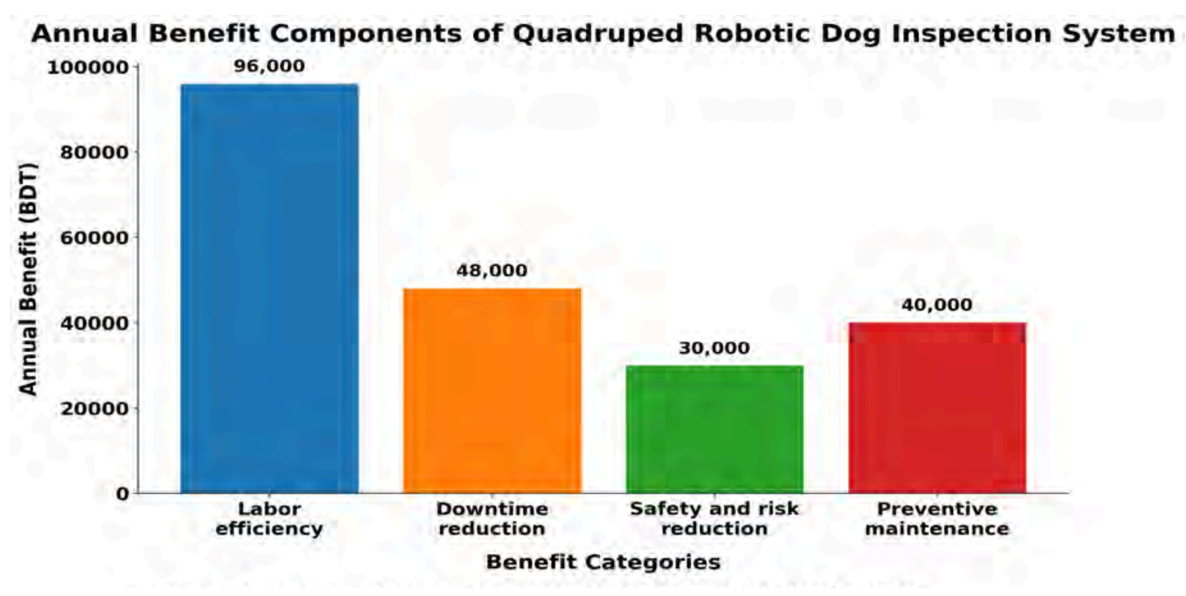


Figure 8.5: Annual Benefit Components

8.3.3 Return on Investment Calculation

Return on Investment measures whether the annual benefit generated by the project justifies the initial development cost. The updated total project cost is used as the initial investment.

$$\text{Initial Investment} = 144,906 \text{ BDT}$$

$$\text{Net Annual Benefit} = 194,000 \text{ BDT/year}$$

The annual ROI is calculated as:

$$\begin{aligned}\text{ROI} &= \frac{\text{Net Annual Benefit}}{\text{Initial Investment}} \times 100 \\ &= \frac{194,000}{144,906} \times 100 \\ &= 133.88\%\end{aligned}$$

Therefore, the project provides an estimated annual ROI of 133.88% under the base-case scenario.

A first-year net ROI can also be calculated after recovering the full initial investment:

$$\begin{aligned}\text{First-Year Net ROI} &= \frac{\text{Net Annual Benefit} - \text{Initial Investment}}{\text{Initial Investment}} \times 100 \\ &= \frac{194,000 - 144,906}{144,906} \times 100 \\ &= 33.88\%\end{aligned}$$

This means that after recovering the full initial investment, the project can still generate an estimated 33.88% net positive return within the first year.

Payback Period

The payback period measures how long it takes to recover the initial investment.

$$\begin{aligned}\text{Payback Period} &= \frac{\text{Initial Investment}}{\text{Net Annual Benefit}} \\ &= \frac{144,906}{194,000} \\ &= 0.747 \text{ years} \\ &= 0.747 \times 12 \\ &= 8.96 \text{ months}\end{aligned}$$

Therefore, the estimated payback period is approximately 9 months.

ROI and Payback Summary

TABLE 8.5: TABLE FOR THE ROI AND PAYBACK SUMMARY

Metric	Value
Initial investment	BDT 144,906
Gross annual benefit	BDT 214,000
Annual O&M cost	BDT 20,000
Net annual benefit	BDT 194,000
Annual ROI	133.88%
First-year net ROI after capital recovery	33.88%
Payback period	8.96 months

Cumulative Cash Flow Analysis of the Quadruped Robotic Dog Project:

This graph shows the monthly cumulative cash flow of the project over a 12-month period. It indicates that the project reaches its break-even point at approximately 8.96 months, after which the cash flow becomes positive.

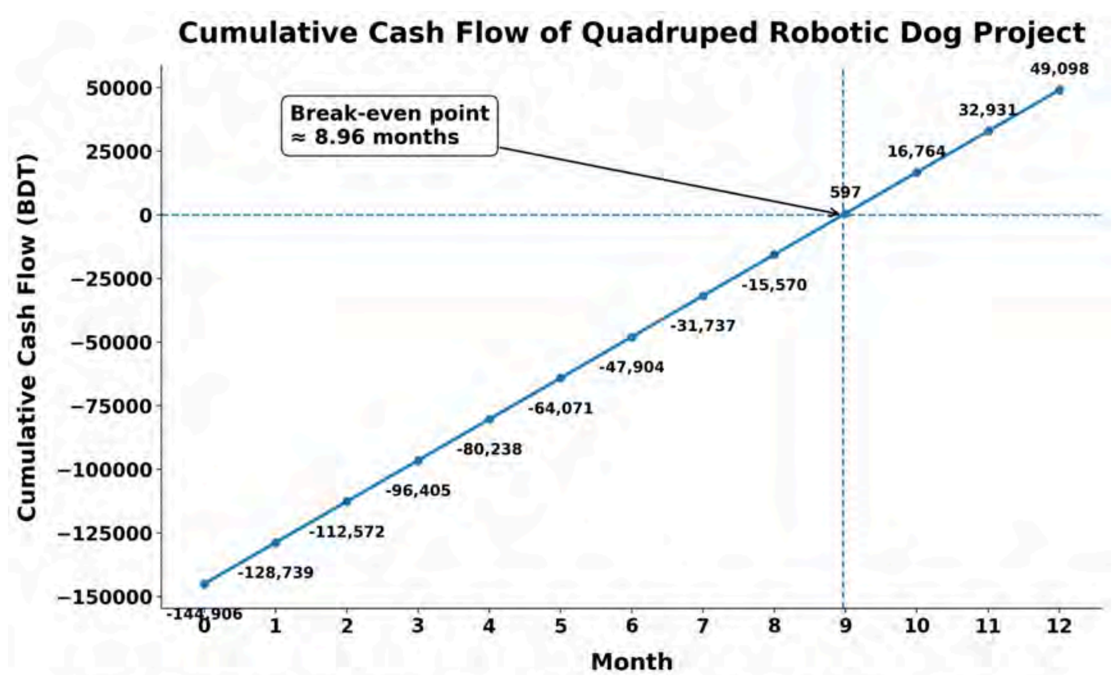


Figure 8.6: Cumulative Cash Flow of Project

8.4 Evaluate economic and financial aspects

8.4.1 Net Present Value Analysis

Net Present Value evaluates the long-term economic feasibility of the project by discounting future benefits into present value. A 5-year useful life is assumed because the robot's core structure, controllers, sensors, and mechanical platform can remain usable with periodic maintenance. Batteries, servos, wires, and small mechanical parts may require replacement, which is included in the annual operation and maintenance cost.

The following assumptions are used:

TABLE 8.6: TABLE FOR THE ASSUMPTIONS FOR PRESENT VALUE ANALYSIS

Parameter	Value
Initial investment	BDT 144,906
Gross annual benefit	BDT 214,000
Annual O&M cost	BDT 20,000
Net annual benefit	BDT 194,000
Project life	5 years
Discount rate	10%

The NPV formula is:

$$\text{NPV} = \sum_{t=1}^{n} \frac{\text{Net Annual Benefit}}{(1+r)^t} - \text{Initial Investment}$$

Here, $r = 10\% = 0.10$ and $n = 5$

The 5-year present value annuity factor at 10% is:

$$\sum_{t=1}^{5} \frac{1}{(1.10)^t} = 3.7908$$

Therefore,

$$\text{PV of Net Benefits} = 194,000 \times 3.7908$$
$$= 735,412.63 \text{ BDT}$$
$$\text{NPV} = 735,412.63 - 144,906$$
$$= 590,506.63 \text{ BDT}$$

The positive NPV of approximately BDT 590,507 indicates that the project is economically feasible over a five-year period. Since the NPV is positive, the project generates value even after considering the time value of money and annual maintenance cost.

TABLE 8.7: TABLE FOR THE NET PRESENT VALUE ANALYSIS

Year	Net Benefit	Discount Factor at 10%	Present Value
1	BDT 194,000	0.9091	BDT 176,363.64
2	BDT 194,000	0.8264	BDT 160,330.58
3	BDT 194,000	0.7513	BDT 145,755.07
4	BDT 194,000	0.6830	BDT 132,504.61
5	BDT 194,000	0.6209	BDT 120,458.73
Total PV of benefits			BDT 735,412.63
Initial investment			BDT 144,906.00
NPV			BDT 590,506.63

Discounted Cash Flow Analysis of the Quadruped Robotic Dog Project

This graph shows the project's present value cash flow from the initial investment to Year 5. It indicates that the project produces positive discounted returns after the initial cost, with cash flow gradually decreasing over time.

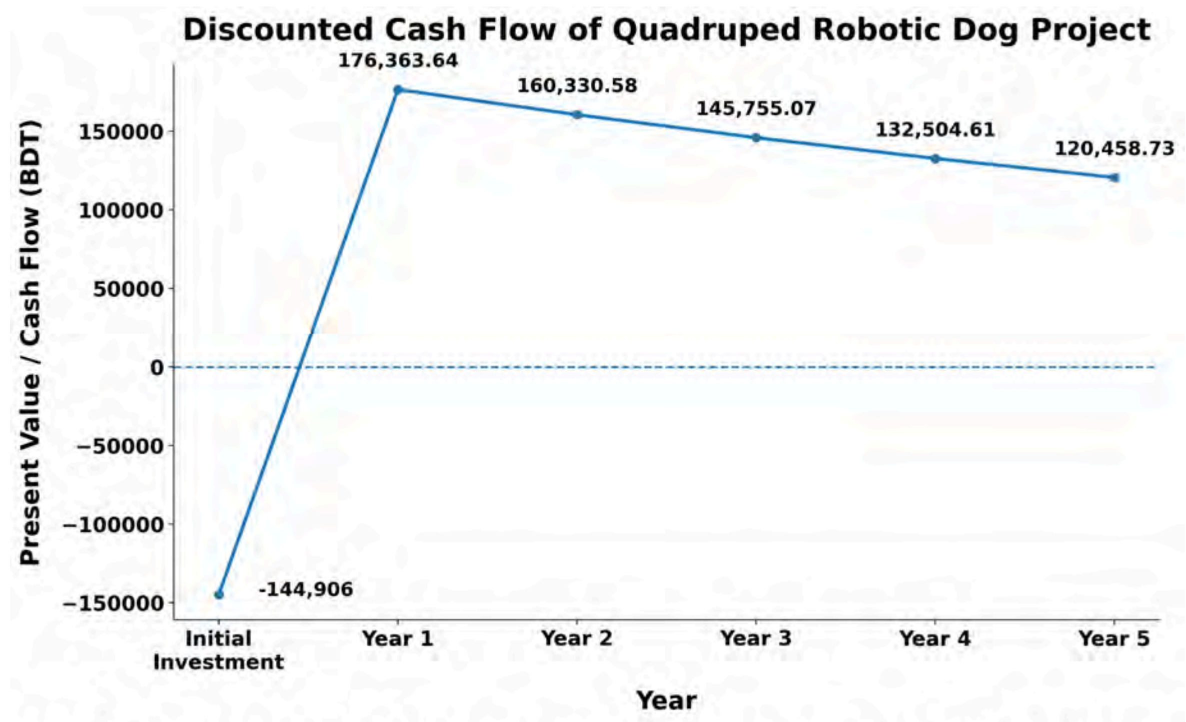


Figure 8.7: Cumulative Cash Flow of Project

8.4.2 Benefit-Cost Ratio

The Benefit-Cost Ratio compares the present value of total benefits with the present value of total costs. It is calculated as:

$$\text{BCR} = \frac{\text{PV of Gross Benefits}}{\text{Initial Investment} + \text{PV of O\&M Costs}}$$

The present value of gross benefits is:

$$\text{PV of Gross Benefits} = 214,000 \times 3.7908$$

$$= 811,228.37 \text{ BDT}$$

The present value of annual O&M cost is:

$$\text{PV of O\&M Costs} = 20,000 \times 3.7908$$
$$= 75,815.74 \text{ BDT}$$

Total present cost is:

$$\text{Total Present Cost} = 144,906 + 75,815.74$$
$$= 220,721.74 \text{ BDT}$$

This is why, Benefit-Cost Ratio will be:

$$\text{BCR} = \frac{811,228.37}{220,721.74}$$
$$= 3.68$$

A BCR of 3.68 means that every BDT 1.00 spent on the project generates approximately BDT 3.68 in present-value benefits over five years. Since the BCR is greater than 1, the project is economically justified.

8.4.3 Cost-Benefit Summary

The economic analysis shows that the quadruped robotic dog is financially feasible under realistic assumptions. The updated project cost of BDT 144,906 is reasonable for a robotics prototype that includes multiple high-torque servos, onboard controllers, IMU sensors, batteries, carbon-fiber structure, and 3D-printed components.

The highest investment is in the actuation system, which is technically justified because quadruped locomotion requires many high-torque joints. The financial results also show that the system can recover its development cost in approximately 8.96 months. The annual ROI is 133.88%, and the five-year NPV is BDT 590,507, which supports the long-term economic logic of the project.

Economic Feasibility Summary

TABLE 8.8: TABLE FOR THE ECONOMIC FEASIBILITY SUMMARY

Financial Indicator	Result	Interpretation
Initial investment	BDT 144,906	Updated final project cost
Gross annual benefit	BDT 214,000	Total estimated annual value
Net annual benefit	BDT 194,000	Benefit after O&M cost
Annual ROI	133.88%	Strong annual return
First-year net ROI	33.88%	Positive return after full cost recovery
Payback period	8.96 months	Investment recovered within first year
5-year NPV	BDT 590,507	Strong long-term economic value
BCR	3.68	Economically justified

Sensitivity Analysis

Sensitivity analysis is required because real-world benefits may vary based on inspection frequency, labor cost, downtime cost, industrial environment, maintenance requirements, and actual field reliability. Therefore, three financial scenarios are considered: conservative, base-case, and optimistic.

TABLE 8.9: TABLE FOR THE ASSUMPTIONS OF SENSITIVITY ANALYSIS

Scenario	Gross Annual Benefit	Annual O&M Cost	Net Annual Benefit
Conservative	BDT 150,000	BDT 25,000	BDT 125,000
Base-case	BDT 214,000	BDT 20,000	BDT 194,000
Optimistic	BDT 280,000	BDT 20,000	BDT 260,000

Conservative Scenario:

$$\text{ROI} = \frac{125,000}{144,906} \times 100$$

$$= 86.26\%$$

$$\text{Payback Period} = \frac{144,906}{125,000} \times 12$$

$$= 13.91 \text{ months}$$

$$\text{NPV} = (125,000 \times 3.7908) - 144,906$$

$$= 328,942.35 \text{ BDT}$$

Base-Case Scenario

$$\text{ROI} = \frac{194,000}{144,906} \times 100$$

$$= 133.88\%$$

$$\text{Payback Period} = 8.96 \text{ months}$$

$$\text{NPV} = 590,506.63 \text{ BDT}$$

Optimistic Scenario

$$\text{ROI} = \frac{260,000}{144,906} \times 100$$

$$= 179.43\%$$

$$\text{Payback Period} = \frac{144,906}{260,000} \times 12$$

$$= 6.69 \text{ months}$$

$$\text{NPV} = (260,000 \times 3.7908) - 144,906$$

$$= 840,698.56 \text{ BDT}$$

TABLE 8.10: TABLE FOR THE FINAL SENSITIVITY ANALYSIS

Scenario	Net Annual Benefit	ROI	Payback Period	5-Year NPV	BCR
Conservative	BDT 125,000	86.26%	13.91 months	BDT 328,942	2.37
Base-case	BDT 194,000	133.88%	8.96 months	BDT 590,507	3.68
Optimistic	BDT 260,000	179.43%	6.69 months	BDT 840,699	4.81

The sensitivity analysis shows that the project remains financially viable even in the conservative case. Even when the annual net benefit is reduced to BDT 125,000 and annual maintenance cost increases to BDT 25,000, the project still has a positive NPV of BDT 328,942 and a BCR greater than 1. This indicates that the financial feasibility of the project is not dependent on overly optimistic assumptions.

8.4.5 Risk-Adjusted Financial Evaluation

Although the financial results are positive, the project includes practical risks. Quadruped robots experience repeated mechanical stress during walking, turning, and load transfer. Servo motors, batteries, wiring, joints, feet, and structural parts may degrade with repeated testing. Therefore, a risk-adjusted evaluation is included by increasing the annual maintenance cost from BDT 20,000 to BDT 30,000.

$$\text{Risk-Adjusted Net Benefit} = 214,000 - 30,000$$

$$= 184,000\ \text{BDT/year}$$

Risk-adjusted ROI:

$$\text{Risk-Adjusted ROI} = \frac{184,000}{144,906} \times 100$$

$$= 126.98\%$$

Risk-adjusted payback period:

$$\text{Risk-Adjusted NPV} = (184,000 \times 3.7908) - 144,906$$
$$= 552,598.66 \text{ BDT}$$

Risk-adjusted NPV:

$$\text{Risk-Adjusted NPV} = (184,000 \times 3.7908) - 144,906$$
$$= 552,598.66 \text{ BDT}$$

Risk-Adjusted Financial Results:

TABLE 8.11: TABLE FOR THE RISK-ADJUSTED FINANCIAL EVALUATION

Metric	Base Case	Risk-Adjusted Case
Annual O&M cost	BDT 20,000	BDT 30,000
Net annual benefit	BDT 194,000	BDT 184,000
ROI	133.88%	126.98%
Payback period	8.96 months	9.45 months
NPV	BDT 590,507	BDT 552,599

Even after increasing annual maintenance cost by 50%, the project remains economically feasible. The payback period increases only from 8.96 months to 9.45 months, while the NPV remains strongly positive. This suggests that the project has a reasonable financial safety margin.

8.4.6 Performance Benchmarking and Economic Justification

The budget demonstrates that the robotic dog prototype was developed at a relatively low cost compared with commercial quadruped inspection platforms. Commercial quadrupeds usually include industrial-grade actuators, proprietary control systems, advanced perception sensors, ruggedized protection, autonomous docking, and certified safety systems. As a result, they are often financially inaccessible for small organizations, student researchers, and early-stage industrial experimentation.

The proposed project does not attempt to directly compete with high-end commercial platforms. Instead, it provides a low-cost academic and experimental alternative that demonstrates the essential principles of quadruped inspection robotics: legged locomotion, real-time control, sensor integration, wireless communication, structural inspection, and limited responsive manipulation.

The economic justification can be explained from five perspectives.

Technical-cost justification:

The largest cost is assigned to servo motors because they are the most critical components for quadruped locomotion. Without sufficient motor torque, the robot would not be able to walk reliably, support its body, or carry the required inspection payload. Therefore, the high actuation cost is technically justified.

Operational-cost justification:

The robot can reduce the need for direct human entry into hazardous inspection areas. Even if a human operator remains involved, the robot shifts the worker from physical inspection to remote supervision, which improves safety and reduces operational burden.

Maintenance-cost justification:

The project uses modular and locally available components such as ESP32 boards, PCA9685 servo drivers, LiPo batteries, standard connectors, 3D-printed parts, and conventional fasteners. This improves repairability and reduces dependence on expensive proprietary replacement parts.

Scalability justification:

The system can be improved in future versions without redesigning the entire platform. Better cameras, stronger motors, improved batteries, AI-based fault detection, stronger frame materials, and autonomous navigation algorithms can be added gradually

Academic and research justification:

Even before commercial deployment, the project has strong educational and research value. It can serve as a platform for studying gait control, embedded systems, sensor fusion, power management, mechanical design, computer vision, and industrial inspection robotics.

8.5 Conclusion

The economic analysis demonstrates that the proposed quadruped robotic dog is financially feasible and technically justified as a low-cost inspection platform. The project aligns with global growth in inspection robotics and the increasing automation trend in Bangladesh, while its updated cost of BDT 144,906 makes it more accessible than commercial quadruped systems. The financial indicators are also favorable, including a net annual benefit of BDT 194,000, ROI of 133.88%, payback period of 8.96 months, five-year NPV of BDT 590,507, and BCR of 3.68. Even under conservative and risk-adjusted scenarios, the project remains economically viable, confirming its potential as a practical academic prototype for safer, modular, and scalable industrial inspection robotics.

Chapter 9: Ethics and Professional Responsibilities [CO13, CO2]

9.1 Introduction

The ethical dimension of this project extends beyond technical performance and must be examined as a core part of responsible engineering practice. Since the proposed quadruped robot is intended for operation in hazardous industrial environments, its design, implementation, and deployment involve important ethical issues related to safety, accountability, privacy, sustainability, human oversight, and social impact. In line with the NSPE Code of Ethics, the IEEE Code of Ethics, and relevant safety and cybersecurity standards, engineers are required not only to develop a functional system but also to ensure that the technology serves the public good in a safe, transparent, and responsible manner. Therefore, this section identifies the major ethical issues and professional responsibilities associated with the project and explains how these responsibilities are applied through practical design choices, operational safeguards, stakeholder engagement, and long-term engineering judgment.

9.2 Identify ethical issues and professional responsibility

Public Safety, Health, and Welfare

The most fundamental ethical issue in this project is the protection of public safety, health, and welfare, which is the highest duty of engineers under the NSPE Code of Ethics (NSPE I.1). Although the quadruped robot is intended to reduce human exposure to hazardous industrial environments, any malfunction in locomotion, sensing, manipulation, or communication could create new risks. For example, actuator failure, unstable movement, inaccurate fault detection, or delayed emergency response could endanger nearby personnel, damage industrial assets, or worsen unsafe site conditions. This creates a direct professional responsibility to design and test the system in a way that prioritizes safety over cost, speed, or convenience.

Competence and Professional Responsibility

A second major ethical issue is the need for engineers to work within their areas of competence and maintain professional responsibility throughout the project (NSPE II.2; IEEE). This system combines multiple technical domains, including robotics, embedded control, wireless communication, artificial intelligence, and cybersecurity. Since weaknesses in any one of these areas can affect the reliability and safety of the overall system, the engineering team has a responsibility to build the project on sound technical knowledge, validated methods, and continuous learning. Ethical responsibility here includes seeking supervision when needed, following accepted engineering practices, and avoiding decisions beyond one’s technical capability.

Integrity, Honesty, and Transparency

The project also raises issues related to integrity, honesty, and transparency (NSPE I.4; IEEE). Engineering design often requires balancing conflicting requirements, such as low weight versus structural robustness, or high actuator torque versus precise and delicate manipulation. These trade-offs may introduce limitations that affect system performance. If such limitations are hidden, exaggerated, or poorly documented, stakeholders may develop unrealistic expectations about the robot's capabilities. Therefore, engineers have a professional responsibility to report system limitations truthfully, disclose risks clearly, and communicate technical uncertainty honestly.

Cybersecurity and Data Privacy

Cybersecurity and data privacy are also important ethical issues in this project. Since the robot depends on wireless communication, remote monitoring, and sensor-based data collection, it may be vulnerable to unauthorized access, hacking, or interception of sensitive information. In industrial settings, this could affect not only data privacy but also physical safety if a malicious actor interferes with robot behavior. Moreover, onboard cameras and sensors may capture operationally sensitive images or environmental data. Engineers therefore have a responsibility to protect both digital and physical systems using recognized frameworks such as IEC 62443 and the NIST Cybersecurity Framework.

Dual-Use Risk and Misuse Potential

Another ethical concern is the possibility of dual-use. Although the system is designed for safe inspection and emergency-responsive operation in industrial environments, its mobility, perception, and manipulation capabilities could potentially be adapted for surveillance or other harmful purposes. This creates a professional responsibility to acknowledge possible misuse and define the intended scope of the technology clearly. In line with broader IEEE ethical principles, engineers should anticipate how a system may be repurposed and take reasonable steps to reduce such risks.

Human Oversight and Operator Well-Being

The robot also introduces ethical issues related to human oversight and operator well-being. Since the system may involve teleoperation or human supervision, poor interface design, communication latency, or excessive operator workload could increase fatigue, stress, and decision errors. This means the engineering responsibility is not only to make the robot function technically, but also to ensure that human operators can use it safely, confidently, and effectively. Ethical design must therefore consider usability, trust, and human accountability.

Socioeconomic Impact of Automation

A broader ethical issue concerns the long-term socioeconomic effect of automation. While the robot is developed to improve safety and reduce hazardous human exposure, its deployment may reduce the need for certain routine manual inspection roles. This creates a professional responsibility to consider the wider social consequences of engineering decisions. Ethical practice requires that automation be framed as a tool to support and protect human workers rather than simply replace them, with attention given to retraining, role transition, and inclusive technological adoption.

Environmental Stewardship and Sustainability

An additional ethical issue is the project's environmental responsibility. Even though the robot is designed to improve safety and operational efficiency, its development and deployment still involve material consumption, battery dependence, electronic components, and long-term maintenance requirements. From an ethical standpoint, engineers should not only consider whether the robot works effectively, but also whether its lifecycle is environmentally responsible. This includes concerns related to energy efficiency, battery disposal, component replacement, and the sustainability of materials used in the robot's structure and electronics. In the broader spirit of professional engineering ethics, acting responsibly in society also implies minimizing environmental harm and promoting sustainable engineering choices wherever possible.

Informed Consent and Stakeholder Engagement

Another important ethical issue is the need for informed consent and meaningful stakeholder engagement. The robot may be deployed in industrial spaces where workers, supervisors, plant managers, safety teams, and maintenance personnel are affected by its operation. Ethical responsibility requires that these stakeholders understand the purpose, limitations, and implications of the system before deployment. If workers are monitored by the robot's sensors or if their workflows are altered by automation, they should not be treated merely as passive subjects of technological change. Instead, engineers have a responsibility to engage stakeholders, communicate clearly, and consider their concerns in the design and implementation process. This supports accountability, trust, and professional respect for those affected by the system.

Social Justice and Equitable Access

The project also raises ethical concerns related to social justice and equitable access. Advanced robotic systems are often more accessible to well-funded organizations, while smaller facilities or resource-limited sectors may be unable to adopt them despite having similar safety risks. This can create inequalities in who benefits from safer inspection technologies. Ethical engineering requires awareness that technological progress should not widen existing gaps between institutions, industries, or worker groups. In a broader professional sense, engineers should consider how their designs can be made more modular, affordable, and adaptable so that the benefits of safety-focused innovation are not restricted only to a privileged few. This aligns with the social responsibility dimension of ethical engineering and with broader human-centered technology principles.

Overall Professional Responsibility

Taken together, these ethical issues show that the responsibilities of the engineering team go beyond technical performance alone. The project must be guided by professional duties related to safety, competence, honesty, accountability, privacy, sustainability, fairness, and social impact. Identifying these issues early is essential because it allows ethical responsibility to shape the system from the beginning rather than being added only after technical development is complete.

9.3 Apply ethical issues and professional responsibility

Apply ethical issues and professional responsibility

The ethical principle of protecting public safety, health, and welfare (NSPE I.1) is applied in this project through a safety-centered design approach. The robot is specifically intended to reduce direct human exposure in dangerous environments such as chemical plants, radioactive zones, and confined industrial spaces. To support this responsibility, safety measures such as emergency stop mechanisms, fail-safe operation, and redundancy in critical control paths are integrated into the design. In addition, relevant safety standards such as ISO 10218-1 and ISO 13482 guide the project's operational and design considerations. This ensures that the project does not merely aim for performance, but actively embeds protection of human life and property into the engineering process.

Applying Competence Through Research and Validation

Professional competence (NSPE II.2; IEEE) is applied by grounding the project in prior research, engineering analysis, and structured validation. The system design is informed by literature on advanced inspection quadrupeds and modern robotic platforms, which helps ensure that the proposed solution is not based on guesswork or unsupported assumptions. Simulations, subsystem evaluations, and iterative design refinements are used to verify performance before deployment. This reflects the ethical duty to practice only with adequate preparation and to improve technical understanding throughout the project lifecycle.

Keeping Integrity and Transparency in Reporting

Integrity, honesty, and transparency (NSPE I.4; IEEE) are applied through the way the project documents its methods, limitations, and trade-offs. Rather than presenting the system as flawless, the design report acknowledges areas of difficulty such as conflicting constraints, subsystem weaknesses, and the current limitations of autonomous control. This honest reporting supports ethical accountability by allowing evaluators, supervisors, and future users to make informed judgments about system capability. Transparent documentation of risks and decisions ensures that engineering claims remain truthful and professionally responsible.

Ensuring Cybersecurity and Privacy Protections

The ethical responsibility to protect systems and data is applied by incorporating cybersecurity and privacy protections into the project framework. Since the robot uses wireless communication and sensor-based monitoring, it must be protected from unauthorized access, manipulation, and data leakage. For this reason, the design aligns with IEC 62443 and relevant NIST cybersecurity guidance. Secure communication, controlled access, and responsible data handling help protect both operational safety and information privacy. A privacy-by-design perspective is also important, especially where camera and sensor data may contain sensitive industrial details.

Applying Ethical Limits to Dual-Use Possibilities

The issue of dual-use is addressed by clearly defining the project's intended application as a civilian industrial inspection and responsive support system. The project does not frame the robot as a general-purpose autonomous platform, but as a safety-focused engineering tool meant to reduce human risk. The use of a human-in-the-loop model further limits misuse by ensuring that major operational decisions remain subject to human supervision and accountability. This is an important application of professional responsibility because it places ethical boundaries around how the technology should be used.

Applying Human Oversight and Operator Protection

Human oversight is applied by ensuring that the system is not designed to remove responsibility from the operator, but to support better human decision-making. The operator interface, feedback systems, and control structure must be designed to minimize confusion, reduce stress, and maintain awareness during operation. By emphasizing reliable teleoperation, real-time feedback, and manageable control logic, the project reflects the ethical duty to safeguard operator well-being as well as technical performance. Training and usability considerations are also part of this responsibility, since a technically capable system can still be ethically weak if it places excessive strain on its users.

Applying Responsibility to Socioeconomic Impact

The project applies ethical responsibility to socioeconomic concerns by positioning the robot as a complement to human labor rather than a substitute for it. Its purpose is to take over inspection activities in dangerous environments while allowing human workers to move toward safer supervisory, analytical, and maintenance-related roles. This approach reflects responsible engineering because it recognizes that technological progress should not ignore workforce consequences. Supporting retraining and emphasizing collaboration between humans and machines aligns the project with broader social responsibility principles, including those promoted in UNESCO AI ethics guidance.

Ethics-Driven Design Decisions

Ethical responsibility is also applied by allowing ethical concerns to directly influence technical and design decisions rather than treating ethics as a separate discussion. In this project, decisions related to locomotion stability, emergency intervention capability, human supervision, subsystem redundancy, and operational limits are not only technical choices but also ethical ones. For example, selecting a design approach that favors safer and more reliable human-guided control at the current development stage reflects an ethics-driven decision because it prioritizes accountability and risk reduction over premature autonomy. This demonstrates that professional responsibility is embedded in design judgment itself.

Implementation of Safety and Privacy Safeguards

Beyond general ethical commitment, the project applies responsibility through concrete safeguards. Safety measures include controlled operation, fault monitoring, emergency stop capabilities, and structured testing under simulated hazardous conditions. Privacy and data protection are addressed through secure system architecture, limited data collection, protected communications, and careful handling of sensor outputs. By implementing these measures in practice, the project moves from ethical intention to operational responsibility. This is consistent with both professional engineering duty and the project's own commitment to safety-by-design and privacy-by-design principles.

Stakeholder Communication and Transparency

Professional responsibility is also applied through transparent communication with stakeholders. Supervisors, evaluators, operators, and potentially affected workers must be given a realistic understanding of the robot's purpose, benefits, limitations, and risks. Clear documentation, honest progress reporting, and open acknowledgment of unresolved challenges are essential to maintain trust. This reflects the ethical requirements of honesty and integrity under NSPE and IEEE principles. In practice, stakeholder communication ensures that no group is misled about the maturity, safety, or expected impact of the system.

Addressing Conflicting Ethical Demands

A particularly important application of professional responsibility is the careful management of conflicting ethical demands. In this project, several ethical goals may compete with one another, such as maximizing safety while keeping the robot lightweight, increasing sensing capability while preserving privacy, or improving autonomy while maintaining human accountability. Ethical engineering does not eliminate these conflicts, but requires them to be recognized, justified, and balanced responsibly. The project applies this principle by documenting trade-offs and favoring decisions that reduce harm, preserve oversight, and support reliable operation. This approach reflects mature professional judgment rather than purely performance-driven design.

Reflection on Limitations and Future Responsibilities

The application of ethical responsibility also requires reflection on what the project cannot yet fully achieve. At the current stage, the robot may still face limitations in autonomy, robustness, cybersecurity maturity, long-term deployment readiness, or affordability. A responsible engineering approach does not ignore these gaps; instead, it openly recognizes them and treats them as future obligations. This means that ethical responsibility continues beyond the present report and includes future work on stronger safeguards, broader accessibility, improved stakeholder inclusion, and more sustainable deployment strategies. Such reflection is essential because professional responsibility does not end when a prototype is completed.

Applying Ethics to Sustainability and Environmental Responsibility

Environmental stewardship is applied by encouraging design choices that improve energy efficiency, reduce unnecessary waste, and support maintainability. In practical terms, this includes selecting durable components, designing for modular repair instead of frequent replacement, and being mindful of battery usage and disposal implications. Although the project's main focus is safety in hazardous environments, responsible application of ethics requires that environmental impact also be considered during material selection, operational planning, and long-term lifecycle thinking. This expands the project from being only technically effective to being more responsibly engineered.

Applying Ethics Through Stakeholder Inclusion and Fairness

The project also applies ethical responsibility by recognizing that engineering decisions affect different groups in different ways. Workers, safety officers, plant operators, and institutions with fewer resources may not all experience the technology equally. For this reason, ethical application involves listening to stakeholder concerns, designing with usability and affordability in mind, and avoiding the assumption that one technological model fits all contexts equally well. This helps support fairness, inclusivity, and more equitable access to the benefits of the system. In a practical sense, it means the project should be communicated and developed not only for technical success, but also for broader social acceptability and fairness.

Applying Ethics Across the Project Lifecycle

Ethical responsibility in this project is not limited to the final design stage; it applies throughout planning, prototyping, testing, deployment, and long-term use. During planning, ethical concerns shape design priorities and intended use. During implementation, testing and subsystem validation help detect risks early. During deployment, operator guidance, safety rules, and controlled use become essential. Even after deployment, long-term responsibility remains important in monitoring misuse potential, updating protections, and ensuring that the system continues to serve the public good. This lifecycle-based approach reflects the idea that ethics must remain part of engineering practice at every stage.

Overall Application of Professional Responsibility

Overall, the project applies ethical issues and professional responsibility by translating broad principles into concrete engineering decisions. Safety standards, honest reporting, secure operation, human oversight, environmental awareness, stakeholder inclusion, and social responsibility are not treated as optional additions, but as core design requirements. In this way, the project demonstrates that responsible engineering is achieved not only by building a working system, but by ensuring that the system serves society safely, transparently, sustainably, and ethically.

9.4 Conclusion

In conclusion, the quadruped inspection robot presents not only a technical challenge but also a significant ethical responsibility. The project must address a broad set of concerns, including public safety, professional competence, honesty, cybersecurity, privacy, environmental responsibility, stakeholder inclusion, fairness, and the wider social effects of automation. By identifying these issues early and applying ethical principles throughout the design and development process, the project demonstrates a commitment to responsible engineering rather than purely technical achievement. The integration of safety-by-design, human-in-the-loop control, transparent reporting, privacy safeguards, and attention to long-term societal impact shows that ethical responsibility has been treated as a central design requirement. Ultimately, the success of the project depends not only on whether the robot performs effectively, but also on whether it is developed and deployed in a way that is safe, accountable, sustainable, and aligned with professional engineering values.

Chapter 10: Conclusion and Future Work

10.1 Project summary/Conclusion

This section summarizes the overall outcome of the quadruped robotic dog project with clarity, honesty, and academic integrity. It presents the main achievements of the project and also identifies the key limitations observed during design, development, testing, and evaluation. Through this balanced discussion, the section shows both the success of the prototype and the areas that require further improvement in future work.

10.1.1 Project Achievements

The project successfully developed a functional quadruped robotic dog capable of performing remote inspection and limited responsive operation. The final system combined mechanical locomotion, wireless control, camera feedback, environmental sensing, obstacle detection, crack detection, and robotic arm manipulation into one integrated platform. The robot was also supported by a GUI interface, allowing the operator to monitor connection status, camera feed, air quality data, system logs, and control inputs in real time.

A major achievement of the project was the successful design and validation of the manual control architecture. The manual system provided more practical control for the current prototype because it gave the operator direct command over walking, turning, arm movement, and claw operation. This was especially useful during responsive tasks where the robot had to approach a lever and physically interact with it. The report also identifies manual control as the more suitable approach for the current stage due to faster response, lower complexity, and stronger task reliability compared with full autonomy.

The final prototype also demonstrated meaningful inspection capability. The vision system was able to detect cracks through camera-based processing, while the ultrasonic sensor supported obstacle detection and stopping behaviour. The MQ-2 and DHT11 sensors added environmental monitoring capability, allowing the robot to collect gas and temperature-related information during operation. Together, these features show that the robot is not only a mobile platform, but also a small-scale inspection system with sensing, monitoring, and response functions.

The mechanical design improved through multiple iterations. Changes in gait strategy, shoulder structure, servo brackets, feet, enclosure, claw design, and sensor holders helped reduce instability and improve functional reliability. The final design therefore reflects an iterative engineering process where early mechanical issues were identified through testing and corrected through practical redesign. This supports the project’s aim of creating a modular and low-cost robotic inspection platform.

10.1.2 Project Limitations

Despite achieving the main prototype objectives, the system still has several limitations. The robot currently depends heavily on manual operation, which means its performance is affected by operator skill, attention, and control accuracy. Although manual control worked well for the prototype stage, it limits scalability for long-duration or large-area inspection tasks. The report also notes that the absence of full autonomy restricts continuous operation and independent decision-making in complex environments.

The locomotion performance was also affected by surface condition. The robot performed better on stable surfaces such as concrete, while grass/turf and tiles introduced more movement resistance or path deviation. This shows that the current gait and foot design are functional, but still not fully optimized for all terrain conditions. Mechanical wobble, grip variation, and surface friction remain important factors for future refinement.

Battery endurance and power consumption were another limitation. Since the robot uses multiple high-torque servos, sensors, wireless modules, and a camera system, power demand increases significantly during walking and manipulation. The report also identifies endurance and power management as a continuing challenge for quadruped inspection robots.

The fault detection system also requires further improvement before it can be considered fully reliable for real industrial use. Although crack detection was successfully demonstrated, detection accuracy can still be affected by lighting, camera angle, surface texture, crack size, and model confidence. The report highlights that real-time crack and microcrack detection can suffer from false positives, weak generalization, and sensitivity to environmental conditions.

Overall, the project achieved a working proof-of-concept for a quadruped robot capable of inspection, monitoring, and responsive action. However, the current prototype should be viewed as a foundation for further development rather than a final industrial-ready system. These limitations provide a clear direction for the improvement strategies and future work section.

10.2 Future work

1. Semi-Autonomous Navigation with LiDAR and SLAM

A major future improvement would be to integrate LiDAR and SLAM for real-time mapping and localization. This would allow the robot to first map the inspection environment and then mark the exact location of detected faults, such as cracks, gas leakage, or abnormal temperature zones. This improvement aligns with the project goal of real-time inspection, remote monitoring, and fault localization in hazardous environments.

2. Reinforcement Learning-Based Path Planning

Reinforcement learning can be added so that the robotic dog can learn how to move through unknown environments using a reward and penalty system. For example, the robot can receive rewards for reaching the target, avoiding obstacles, and maintaining stability, while penalties can be given for collision, slipping, excessive deviation, or unstable walking. This would help the robot improve its path selection and movement behaviour over time.

3. Foot Sensor and Closed-Loop Walking Feedback

Foot sensors can be added to detect whether each foot has properly touched the ground during walking. This would create a closed-loop feedback system where the robot can identify walking errors, such as missed steps, slipping, or unstable ground contact, and automatically correct its gait. This improvement would make locomotion more stable on uneven surfaces, stairs, and industrial floors.

4. Stair-Climbing Algorithm

A dedicated stair-climbing algorithm can be developed to improve the robot's mobility in real industrial environments. Since many hazardous inspection areas include stairs, platforms, and uneven walkways, stair-climbing capability would make the robot more practical for substations, factories, oil and gas facilities, and other complex sites.

5. Multi-Sensor Fusion and Radiation Detection

The robot can be upgraded with a stronger multi-sensor fusion system. A Geiger-Müller radiation sensor can be added for nuclear or radiation-prone environments. Other useful sensors may include a thermal camera for heat detection, flame sensor, smoke sensor, CO sensor, H_2S sensor, methane/LPG gas sensor, vibration sensor, acoustic sensor, pressure sensor, and improved IMU. Combining these sensors would help the robot detect multiple hazards more accurately and support safer inspection in nuclear, chemical, oil and gas, and power-system environments.

6. Battery Health Monitoring System

A battery health monitoring module can be integrated to continuously check battery voltage, current, temperature, discharge rate, and remaining capacity. The GUI can show warning messages when the battery is low, half-drained, or close to unsafe discharge level. This would improve power safety and prevent sudden shutdown during inspection or emergency-response tasks.

7. Real Industrial Environment Testing

Future testing should be conducted in real industrial environments such as nuclear facilities, oil and gas plants, chemical plants, substations, and confined inspection areas. This would help validate the robot's actual performance under real hazards, rough surfaces, low light, dust, heat, gas exposure, and communication limitations.

8. Claw and Soft Gripper Improvement

The current claw mechanism can be improved by adding a proper soft gripper. A soft gripper would provide better contact with the lever or object, reduce slipping, and increase gripping reliability during responsive operation. This would improve the robot's ability to pull levers, press switches, or handle fragile objects.

9. Voice-Based Control Using ChatGPT API

A voice-based control system can be added using the ChatGPT API or a similar speech-command interface. This would allow the operator to control the robot through spoken commands such as "move forward," "stop," "turn left," or "activate crack detection." For safety, critical commands such as emergency shutdown or lever pulling should require confirmation before execution.

10. Motion-Mirroring Teleoperation Using a Small Robot Model

A small-scale version of the robotic dog can be developed as a motion-mirroring controller. When the operator moves the small robot manually, its joint movements can be transmitted to the full-sized robotic dog, allowing the larger robot to copy the same motion. This master-slave control method could make teleoperation more intuitive, especially for complex movements such as arm positioning, claw control, and special walking postures.

Chapter 11: Identification of Complex Engineering Problems and Activities

11.1 Identify the attribute of complex engineering problem (EP)

Complex Engineering Problem (EP) attributes are necessary to show that the project meets the required level of engineering complexity and professional standard. By identifying these attributes, the project can be assessed against factors such as depth of knowledge, conflicting requirements, analysis, unfamiliar problems, applicable codes, stakeholder needs, and system interdependence. Maintaining these attributes is important because it ensures the project is not treated as a simple design task, but as a real engineering problem that requires systematic thinking, technical judgement, safety consideration, and evidence-based decision-making.

TABLE 11.1: ACHIEVEMENT OF THE COMPLEX ENGINEERING PROBLEM MILESTONES

EP	Attributes	Put tick (√) as appropriate
P1	Depth of knowledge required	√
P2	Range of conflicting requirements	√
P3	Depth of analysis required	√
P4	Familiarity of issues	√
P5	Extent of applicable codes	√
P6	Extent of stakeholder involvement and needs	
P7	Interdependence	√

11.2 Provide reasoning how the project address selected attribute (EP)

P1: Depth of Knowledge Required

The project required a high depth of engineering knowledge because it combined several technical areas into one integrated robotic system. The design involved quadruped locomotion, gait planning, inverse kinematics, actuator torque calculation, embedded control, sensor integration, wireless communication, robotic arm manipulation, and computer vision-based crack detection. The project also used YOLOv8 for real-time microcrack and fault detection, which required knowledge of machine learning, image processing, model training, and performance evaluation. Therefore, the project clearly satisfies P1 because it required advanced multidisciplinary engineering knowledge rather than basic or routine design work.

P2: Range of Conflicting Requirements

The project involved several conflicting requirements that had to be balanced during design and implementation. The robot needed high actuator torque for walking and manipulation, but the total weight had to remain low for stable locomotion and energy efficiency. The system also needed real-time sensing and wireless monitoring, but this increased power consumption and computational demand. Another conflict appeared between manual and autonomous control, where the autonomous system offered better accuracy and future scalability, while the manual system showed better completion time, lower complexity, and higher arm manipulation success. These trade-offs show that the project addressed P2 by requiring careful engineering judgement between performance, cost, safety, weight, power, and reliability.

P3: Depth of Analysis Required

The project required detailed analysis before selecting the final design approach. The team compared manual and autonomous control systems through simulation and testing. The body locomotion subsystem was evaluated using completion time, number of steps, positional deviation, and accuracy. The arm subsystem was evaluated using completion time and success rate. The report also included torque calculation, gait analysis, inverse kinematics, microcrack detection performance, battery and power considerations, cost analysis, and sustainability evaluation. This shows that the project satisfies P3 because the final design decision was based on systematic analysis and evidence rather than simple assumption.

P4: Familiarity of Issues

The project addressed unfamiliar and partially emerging engineering issues because quadruped inspection robots are still developing technologies, especially for hazardous industrial environments. The robot had to deal with uncertain terrain, obstacle detection, real-time camera feedback, fault detection, wireless operation, and responsive manipulation. The report also notes that some areas of quadruped robot design do not have fully clear standards, so project-specific criteria, benchmarking, risk assessment, and iterative testing were needed. This satisfies P4 because the problem was not a familiar routine engineering task.

P5: Extent of Applicable Codes

The project satisfies P5 because it considers multiple engineering standards, safety codes, and operational guidelines. These included IEC 60204-1 for machinery electrical safety, ISO 10218-1 and ISO 10218-2 for industrial robot safety, ISO 13482 for robot interaction safety, IEC 60529 for environmental protection, OSHA HAZWOPER for hazardous material handling, IAEA/NRC guidance for radiation-related safety, and IEEE/ITU standards for wireless communication. These codes were relevant because the robot is intended for hazardous environments such as chemical plants, oil and gas facilities, power systems, confined spaces, and high-risk industrial zones.

P7: Interdependence

The project satisfies P7 because the robot's subsystems were strongly interdependent. The locomotion system depended on actuator torque, power supply, inverse kinematics, gait control, and operator commands. The inspection system depended on camera input, sensors, YOLOv8 detection, wireless data transmission, and operator monitoring. The robotic arm depended on the control system, power system, kinematic model, and real-time user input. If one subsystem failed, such as communication, power regulation, sensing, or actuation, the whole inspection operation could be affected. This high level of subsystem dependency shows that the project clearly meets the interdependence attribute.

11.3 Identify the attribute of complex engineering activities (EA)

Complex Engineering Activities (EA) attributes are necessary to show that the project involved more than basic technical execution. These attributes help explain the level of resources, interaction, innovation, social impact, environmental responsibility, and practical complexity involved in completing the project. Maintaining these attributes is important because an engineering project should be planned and carried out with proper use of resources, teamwork, safety awareness, sustainability, and real-world application in mind. Therefore, Section 11.3 helps demonstrate that the project activities were complex, multidisciplinary, and relevant to professional engineering practice.

TABLE 11.2: ACHIEVEMENT OF THE COMPLEX ENGINEERING ACTIVITIES MILESTONES

EA	Attributes	Put tick (√) as appropriate
A1	Range of resource	√
A2	Level of interaction	√
A3	Innovation	√
A4	Consequences for society and the environment	√
A5	Familiarity	

11.4 Provide reasoning how the project address selected attribute (EA)

A1: Range of Resources

The project required a wide range of resources to complete the engineering activity. Hardware resources included servo motors, ESP32-based control components, sensors, camera, LiPo battery, buck converters, robotic arm parts, claw mechanism, carbon fiber rods, 3D-printed mechanical parts, and other electronic modules. Software and IT resources included Webots, Python, YOLOv8, PyTorch, VS Code, and Spyder. The project also required laboratory facilities, fabrication tools, simulation environments, testing setups, and team collaboration. Therefore, it satisfies A1 because the activity depended on multiple technical, physical, software, financial, and human resources.

A2: Level of Interaction

The project involved a high level of interaction between hardware, software, users, and engineering subsystems. The operator controlled the robot through joystick/manual control, live camera feed, wireless communication, and a monitoring interface. The internal systems also interacted continuously, including sensors, onboard computer, servo motors, robotic arm, power system, and fault detection module. Team members coordinated mechanical design, electronics integration, machine learning development, testing, report writing, and validation, satisfying A2 through strong interaction between people, tools, components, and subsystems.

A3: Innovation

The project satisfies A3 because it developed an innovative low-cost quadruped robot for real-time inspection and responsive operation. The system integrated live monitoring, sensor-based inspection, YOLOv8-based crack and microcrack detection, obstacle detection, wireless communication, and a 3-DOF robotic arm for limited emergency operations. The project attempted to provide a practical alternative to expensive commercial quadruped inspection robots by developing a modular academic prototype. This makes the activity innovative in terms of design integration, cost-effectiveness, and application in hazardous industrial inspection.

A4: Consequences for Society and the Environment

The project has important social and environmental consequences because it aims to reduce human exposure to hazardous inspection environments. By using the robot for remote inspection, workers can avoid direct entry into areas with toxic gases, high voltage, extreme temperature, confined spaces, radiation risk, or structural danger. This improves worker safety and supports predictive maintenance by detecting cracks and early signs of equipment failure. The project also considered environmental and sustainability issues such as battery use, energy efficiency, electronic waste, modular repair, and long-term maintainability. Therefore, the project satisfies A4 because it affects both society and the environment.

References

[1] Bureau of Labor Statistics, “Fatal occupational injuries involving confined spaces, 2011–2018,” Fact Sheet, Jul. 2020. [Online]. Available: https://www.bls.gov/iif/factsheets/fatal-occupational-injuries-confined-spaces-2011-19.htm

[2] International Labour Organization, “Chemical hazards in the mining sector: An industry case,” Instructional Material, 2023. [Online]. Available: https://www.ilo.org/sites/default/files/wcmsp5/groups/public/%40ed_protect/%40protrav/%40safework/documents/instructionalmaterial/wcms_871779.pdf

[3] Ghost Robotics, “Autonomous Robots for Industrial Inspection in Hazardous Zones,” Case Study, Sept. 2025. [Online]. Available: |https://www.ghostrobotics.io/case-studies/autonomous-robots-for-industrial-inspection-in-hazardous-zones

[4] ANYbotics, “Inspection Robots: Definitions, Applications and Advantages,” Technical Article, Mar. 2023. [Online]. Available: https://www.anybotics.com/news/inspection-robot-definitions-applications-and-advantages/

[5] National Safety Council (Work to Zero), “Improving Workplace Safety with Robotics,” White Paper, 2022. [Online]. Available: https://www.nsc.org/getmedia/25023964-33a8-4c93-a906-d29702a6d931/wtz-robotics-wp.pdf

[6] Standard Bots, “How much does the Boston Dynamics Spot robot cost in 2025?” Technical Brief, 2025. [Online]. Available: https://standardbots.com/blog/spot-robot

[7] C. Brosque and M. Fischer, “Safety, quality, schedule, and cost impacts of ten construction robots,” Construction Robotics, vol. 6, no. 2, pp. 163–186, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s41693-022-00072-5.

[8] ANYbotics, “ANYmal: Autonomous robotic inspection solution,” *ANYbotics*, 2026. Accessed: Apr. 13, 2026.

[9] Boston Dynamics, “Spot®: The agile mobile robot,” *Boston Dynamics*, 2026. Accessed: Apr. 13, 2026.

[10] J. Sánchez-Cubillo, J. Del Ser, and J. L. Martín, “Toward fully automated inspection of critical assets supported by autonomous mobile robots, vision sensors, and artificial intelligence,” *Sensors*, vol. 24, no. 12, p. 3721, 2024.

[11] S. Halder, K. Afsari, E. Chiou, R. Patrick, and K. A. Hamed, “Construction inspection and monitoring with quadruped robots in future human–robot teaming: A preliminary study,” *Journal of Building Engineering*, vol. 65, p. 105814, 2023.

[12] H. K. Kaveh and R. A. Alhajj, “Recent advances in crack detection technologies for structures: A survey of 2022–2023 literature,” *Frontiers in Built Environment*, vol. 10, Art. no. 1321634, 2024.

[13] Ultralytics, “YOLOv8,” Ultralytics Docs, 2026. Accessed: Apr. 13, 2026

[14] LeoTronics Robotics, “Service Robots for Inspection and Maintenance-Challenges and Trends,” Industry Brief, 2024. [Online]. Available: https://leotronics.eu/en/blog/service-robots-for-inspection-and-maintenance

[15] The Business Research Company, “Inspection Robots—Global Market Report 2025,” Market Analysis, 2025. [Online]. Available: https://www.thebusinessresearchcompany.com/report/inspection-robots-global-market-report

[16] 16 Kimberlite Research (as cited by Hint Global), “Cost of Unplanned Downtime in Oil & Gas,” Industry Study, 2019. [Online]. Available: https://www.hint-global.com/blog/cost-of-unplanned-downtime/

[17] AFL-CIO, “Death on the Job: The Toll of Neglect,” Annual Report, 2025. [Online]. Available: https://aflcio.org/sites/default/files/2025-04/2512%20AFL-CIO%20DOTJ%202025%20N-BUG_FINAL.pdf

[18] Bureau of Labor Statistics, Census of Fatal Occupational Injuries, “Confined Spaces-Fatality Data,” Statistical Series, 2018. [Online]. Available: https://www.bls.gov/iif/fatal-injuries-tables/archive.htm

[19] ISO 10218-1, "Robots and robotic devices — Safety requirements for industrial robots - Part 1: Robots," International Organization for Standardization, 2011.

[20] National Society of Professional Engineers (NSPE), "Code of Ethics for Engineers," 2020.

[21] IEEE, "IEEE Code of Ethics," IEEE, 2020.

[22] IEEE, "Ethical considerations for autonomous systems," IEEE Spectrum, 2019.

[23] UNESCO, "AI Ethics," United Nations Educational, Scientific and Cultural

[24] International Electrotechnical Commission, *IEC 60529: Degrees of Protection Provided by Enclosures (IP Code)*, IEC, Geneva, Switzerland, 2013.

[25] International Atomic Energy Agency, *IAEA Safety Standards Series No. GSR Part 3: Radiation Protection and Safety of Radiation Sources*, IAEA, Vienna, Austria, 2014.

[26] U.S. Nuclear Regulatory Commission, *Radiation Protection Regulations*, NRC, Washington D.C., USA, 2020.

[27] Occupational Safety and Health Administration, *Hazardous Waste Operations and Emergency Response (HAZWOPER), 29 CFR 1910.120*, U.S. Department of Labor, Washington D.C., USA, 2022.

[28] Institute of Electrical and Electronics Engineers, *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, New York, USA, 2021

[29] 16 Kimberlite Research (as cited by Hint Global), "Cost of Unplanned Downtime in Oil & Gas," Industry Study, 2019. [Online]. Available: https://www.hint-global.com/blog/cost-of-unplanned-downtime/

[30] AFL-CIO, "Death on the Job: The Toll of Neglect," Annual Report, 2025. [Online]. Available: https://aflcio.org/sites/default/files/2025-04/2512%20AFL-CIO%20DOTJ%202025%20N-BUG_FINAL.pdf

[31] Bureau of Labor Statistics, Census of Fatal Occupational Injuries, "Confined Spaces-Fatality Data," Statistical Series, 2018. [Online]. Available: https://www.bls.gov/iif/fatal-injuries-tables/archive.htm

[32] D. Vogel, R. Baines, J. Church, and M. Hutter, "Robust Ladder Climbing with a Quadrupedal Robot," arXiv preprint arXiv:2409.17731, 2024.

[33] A. A. Saputra, N. Takesue, K. Wada, A. J. Ijspeert, and N. Kubota, "AQuRo: A Cat-like Adaptive Quadruped Robot With Novel Bio-Inspired Capabilities," Frontiers in Robotics and AI, vol. 8, p. 562524, 2021.

[34] Z. Liu et al., "LocoMan: Advancing Versatile Quadrupedal Dexterity with Lightweight Loco-Manipulators," arXiv preprint arXiv:2403.18197, 2024.

[35] G. Shi, W. Zhang, Z. Yang, and J. Pan, "Dexterous Manipulation with a Quadrupedal Robot," arXiv preprint arXiv:2011.08811, 2020.

[36] A. Rudin, D. Hoeller, P. M. Wensing, and M. Hutter, "Pedipulate: Enabling Manipulation Skills using a Quadruped Robot's Legs," in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 2023, pp. 1-8.

37] Global Market Insights, "Inspection robots market size and share, 2025 to 2034," *Global Market Insights*, 2025.

[38] SkyQuest Technology, "Inspection robots market size, share, and growth analysis," *SkyQuest Technology*, 2025.

[39] Strategic Market Research, "Quadruped robot market size, share, and forecast," *Strategic Market Research*, 2024.

[40] Virtue Market Research, "Quadruped robot market size, share, growth, 2024–2030," *Virtue Market Research*, 2024.

[41] Grand View Research, "Professional services robotics market outlook: Inspection and maintenance segment," *Grand View Research*, 2024.

[42] 6Wresearch, "Bangladesh factory and industrial automation market," *6Wresearch*, 2023.

[43] Vogue Business, "Garment factories are ramping up automation. What will it do to jobs?" *Vogue Business*, 2023.

Appendix

FYDP-P/Summer 2025 Summary of Team Log Book Journal

Date/Time/ Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
19-06-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Shadid Al Akib 4. Tazwar Ahmed	1.Need to specify the objective 2. Need to find three design approaches.	Equal Contribution	
26-06-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed	1.Need to do literature review 2. Need to first fix what would be its application	Equal Contribution	
03-07-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain	1.Need to talk to stakeholders or industrial expertees 2. Need to find the current and future industries' relevance to our project.	Equal Contribution	
10-07-2025	1.Shadid Al Akib 2. MD Shahariar Hossain 3. Tazwar Ahmed	1.Need to make block diagram of design approaches 2. Need to check the feasibility of our project	Equal Contribution	
16-07-2025	1.Rafin Md Abrar 2. Tazwar Ahmed	1.Need to start writing our concept note with proper instructions 2. Need to summarize all the literature gap that we found	Equal Contribution	
24-07-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	1.Need to fix the design approaches’ block diagram 2. Need to start working on MID progress presentation slides	Equal Contribution	

05-08-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Shadid Al Akib	1.Need to use white background on slides 2. Need to narrow down our objectives and applications.	Equal Contribution	
14-08-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed	1.Need to reduce the amount of hands on top of our design approaches 2 and 3, remove the claws and climbing mechanism 2. Need to remove three alternative design and come up with two solid design approaches	Equal Contribution	

FYDP-D/Fall 2025 Summary of Team Log Book Journal

Date/Time/ Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
15-10-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Shadid Al Akib 4. Tazwar Ahmed	1. Informed ATC members about our simulation on Webots 2. Fixed the parameters on which we want to simulate 3. Provided a tentative budget and plan to order servo motor from China	Equal Contribution	
23-10-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	1. Provide clear list of Parameter matrics 2. Table for which parameter needs which tools to simulate 3. There should be one final design visualization at beginning our slides	Equal Contribution	

06-11-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	1. Comparison of cost matrix need to be added 2. Remove all the parameters that can not provide any quantitative value. 3. Fix the formula for deviation.	Equal Contribution	
20-11-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	1. Fix path finding algorithm. 2. Give a first person POV camera on the manual controlled robot. 3. Give a separate task for object manipulation.	Equal Contribution	
03-12-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	1. Add an error bar chart for comparison. 2. Remove the DQN controller since it is not completed yet. 3. Fix the keywords of Problem statement and objective to make it more clear.	Equal Contribution	
10-12-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	1. Distribute the whole poster into 3x3 matrices for 9 headlines 2. Make one slot for Background Research and Problem Statement together. 3. Fix the logo of key parameters (Cost, Accuracy, Complexity). 4. Give a picture of one arena instead of four. 5. Fix the project plan gantt chart since it's not visible.	Equal Contribution	

15-12-2025	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	1. Provide a picture of a robot operating in an industry. 2. Picture of Background research, problem statement looks messy. 3. Fix the picture of design approaches since it looks too sci-fi. 4. Remove the radar chart. 5. Increase the size of the table for optimal solution.	Equal Contribution	

FYDP-C/Spring 2026 Summary of Team Log Book Journal

Date/Time /Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
05-02-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Shadid Al Akib 4. Tazwar Ahmed	• Completed the 3D design and modelling of the major robot parts. • Designed the body frame, shoulder joint, knee joint, and end tool/foot. • Confirmed that the designs were ready for 3D printing. • Requested permission from the respected faculty members to begin the 3D printing phase.	Equal Contribution	
12-02-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Started the body assembly phase of the project. • Operated two 3D printers simultaneously to speed up fabrication. • Completed the assembly of the first quadruped leg using PVC pipe.	Equal Contribution	

19-02-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Completed the full mechanical assembly of the robot. • Rebuilt the structure using carbon fiber for improved strength and stability. • Finished the mechanical phase of the project. • Planned to begin the electronics integration phase next.	Equal Contribution	
26-02-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Completed the power distribution system. • Integrated the servo control shield with the robot. • Calibrated the servo motors to ensure correct movement orientation. • Rechecked the carbon fiber body structure after rebuilding.	Equal Contribution	
05-03-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Started calibrating the robot one leg at a time. • Tested the inverse kinematics walking gait with the servo motors. • Tested wireless control functionality for robot movement. • Identified areas that required further tuning before full walking tests.	Equal Contribution	
19-03-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Identified mechanical issues such as back-leg collision with the ground, excessive vibration, and structural instability. • One servo motor was damaged during PDB testing, so backup servos were used. • Added 10A automotive fuses to the PDB for protection. • Added lock nuts, screws, 3D-printed washers, and brackets to improve stability.	Equal Contribution	

		• Reduced overall dimensions and redesigned the structure for better balance. • Upgraded the control system from open-loop to PID-based closed-loop control. • Integrated an IMU sensor for tilt, pitch, and roll feedback.		
26-03-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Completed Version 2 of the robot. • Observed improvement in mechanical stability and balance control compared to Iteration 1. • Planned to begin Iteration 3 based on performance observations..	Equal Contribution	
16-04-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Started redesigning the robot body structure for Iteration 3. • Worked on improving robustness and mechanical stability. • Planned to reprint several parts and components.. • Decided to proceed with fault detection implementation after structural improvements.	Equal Contribution	
21-05-2026 BRACU Substation	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Successfully deployed the robotic dog at the BRACU substation. • Tested the emergency lever-pulling operation in a real-life inspection environment. • Developed and tested a 3D-printed lever mechanism. • Demonstrated the emergency condition using a red LED and buzzer. • Controlled the robot using a gamepad during the substation test. • Planned to integrate object detection and crack detection into the GUI.	Equal Contribution	

		• Planned to attach an LED light and implement sonar-based obstacle avoidance.		
28-05-2026 FYDP Lab, BRAC University	1.Rafin Md Abrar 2. MD Shahariar Hossain 3. Tazwar Ahmed 4. Shadid Al Akib	• Finalized the project poster. • Reviewed and selected testing videos for the final presentation. • Practiced the final presentation as a group. • Divided speaking parts among group members. • Prepared explanations for the final design, prototype development, testing results, limitations, and future improvements.	Equal Contribution	

All the project files

Simulation Files:

https://drive.google.com/drive/folders/1xYp_u9Bx-DFlmgdPY5isLakWikVq6koZ?usp=sharing

CAD Files:

https://drive.google.com/drive/folders/1uqg8P-FRBTOFVpR0fiMW0Cc9u35mIrkT?usp=sharing

Codes:

https://drive.google.com/drive/folders/1eOpjZ9VDF2FPkb1kIbxMsBPeoQ2NbCnX?usp=sharing

GUI:

https://github.com/Seyam554/Quadruped-Robot-Control-System-and-GUI-interface

Poster:

https://drive.google.com/file/d/15VykGXQJiSfH7RRQtEYFLAwFpvVy6qDS/view?usp=sharing

Demonstration Video:

https://www.youtube.com/watch?v=Nx-Y6TwPJRM