AN AUTONOMOUS ROVER FOR EFFICIENT WASTE SEGREGATION AND COLLECTION IN URBAN PARKS AND FIELDS

By

Nafim Karim Khan

21221039

Farhad Hassan

21221025

Rubaya Huq

21221003

Shah Rasul Khan

21221054

A Final Year Design Project (FYDP) submitted to the Department of Electrical & Electronic Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Electrical & Electronic Engineering

Department of Electrical & Electronic Engineering

BRAC University

September 2025

AN AUTONOMOUS ROVER FOR EFFICIENT WASTE SEGREGATION AND COLLECTION IN URBAN PARKS AND FIELDS

By
Nafim Karim Khan
21221039
Farhad Hassan
21221025
Rubaya Huq
21221003
Shah Rasul Khan
21221054

A Final Year Design Project (FYDP) submitted to the Department of Electrical & Electronic Engineering in partial fulfillment of the requirements for the degree of
B.Sc. in Electrical & Electronic Engineering

Academic Technical Committee (ATC) Panel Member:

Dr. Mohammed Belal Hossain Bhuian (Chair)
Associate Professor, Department of EEE, BRAC University

Md. Mahmudul Islam
Lecturer, Department of EEE, BRAC University

Sabbir Ahmed
Lecturer, Department of EEE, BRAC University

Department of Electrical & Electronic Engineering
BRAC University
September 2025

Declaration

It is hereby declared that

1. The Final Year Design Project (FYDP) submitted is my/our own original work while completing degree at BRAC University.
2. 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.
3. 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.
4. I/We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rubaya Huq 21221003	**Nafim Karim Khan** 21221039
Farhad Hassan 21221025	**Shah Rasul Khan** 21221054

Approval

The Final Year Design Project (FYDP) titled "AN AUTONOMOUS ROVER FOR EFFICIENT WASTE SEGREGATION AND COLLECTION IN URBAN PARKS AND FIELDS" submitted by

1. Nafim Karim Khan (21221039)
2. Farhad Hassan (21221025)
3. Rubaya Huq (21221003)
4. Shah Rasul Khan (21221054)

of Summer 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Electrical & Electronic Engineering on [Date-of-Defense].

Examining Committee:

Academic Technical Committee (ATC): (Chair)

____

Dr. Mohammed Belal Hossain Bhuian
Associate Professor, Department of EEE
BRAC University

Final Year Design Project Coordination Committee: (Chair)

____

Mirza Rasheduzzaman, PhD
Associate Professor, Department of EEE
BRAC University

Department Chair:

____

Md. Mosaddequr Rahman, PhD
Professor and Chairperson, Department of EEE
BRAC University

ETHICS STATEMENT

The project titled "An Autonomous Rover for Efficient Waste Segregation and Collection in Urban Parks and Fields" was done adhering to all the requirements and the highest possible ethical standards. The entire team made a collective effort to complete the task in the utmost sincere manner. The interaction between the team members were polite, fair and every decision was taken keeping all the stakeholders in mind. The literature review was done accordingly and all the works were cited using IEEE Format. The Turnitin Plagiarism and AI detector was used and the similarity index of this report came out to be and AI percentage was

ABSTRACT

This project presents the design and implementation of an autonomous rover system for efficient waste segregation and collection in urban parks and fields. Bangladesh faces significant challenges in solid waste management due to rapid urbanization and limited infrastructure which results in environmental and health hazards. To address this, we developed a rover equipped with a Raspberry Pi 5, Pi camera and a 4-DOF robotic arm. A YOLOv8n machine learning model enables real-time waste detection and classification into biodegradable and non-biodegradable categories with over 84% accuracy from practical scenarios. The rover follows a zigzag navigation path where it autonomously detects and picks up the waste which is then deposited into a compartmentalized bin that rotates through a servo. Extensive simulations and tests were conducted which validated its efficiency and achieved complete field coverage within 20 minutes while maintaining sustainable power usage. This solution demonstrates a scalable, cost-effective, and eco-friendly approach to urban waste management which promotes cleaner environments and improved public health.

Keywords: **autonomous rover; waste segregation; machine learning; robotics; sustainability; YOLO; image processing**

DEDICATION

We would like to dedicate this paper to our beloved parents who have been always there for us throughout the journey. This project would not have been possible without their support.

ACKNOWLEDGEMENT

We are immensely grateful to everyone who has supported us throughout our FYDP journey. 11th floor attendant Md Foysal Ali who stayed and patiently waited for us on days we were working late; LTO Md Nazrul Islam who gave us advice for project improvement and mental support when things were tough; also welding mechanic Md Amzad who helped us with the mechanical works like welding and making of the wheel are some of them who made this journey possible. This project would not have been possible without their support and compassion. Thank you all for your invaluable contributions.

TABLE OF CONTENTS

Ethics Statement....i
Abstract....ii
Dedication.... iii
Acknowledgement....iv
TABLE OF CONTENTS....v
List of Figures....ix
List of Tables....xi
Chapter1: Introduction [CO1, CO2, CO10]....1
1.1 Introduction....1
1.1.1 Problem Statement....1
1.1.2 Background Study [Rephrased most of the part, so copy paste this section]....1
1.1.3 Literature Gap....3
1.1.4 Relevance to current and future Industry....3
1.2 Objectives, Requirements, Specification and constraint....4
1.2.1. Objectives & Scope....4
1.2.2 Functional and Nonfunctional Requirements....4
1.2.3 Specifications....5
1.2.4 Technical and Non-technical consideration and constraint in design process....5
1.2.5 Applicable compliance, standards, and codes....6
1.3 Systematic Overview/summary of the proposed project....7
1.4 Conclusion....7
Chapter 2: Developing the Solution: Methodology and Approaches....8
2.1 Introduction....8
2.2 Identify multiple design approach....8
2.3 Describe multiple design approach....8
2.3.1 Pre-Sorting Approach:....8
2.3.2. Post Sorting Approach:....9
2.4 Analysis of multiple design approach....10
2.5 Conclusion....11
Chapter 3: Use of Modern Engineering and IT Tool. [CO9]....12
3.1 Introduction....12

3.2 Select appropriate engineering and IT tools 12
3.3 Use of modern engineering and IT tools 13
3.4 Conclusion 16
Chapter 4: Optimization of Multiple Design and Finding the Optimal Solution. [CO7] 17
4.1 Introduction 17
4.2 Optimization of multiple design approach 17
4.2.1 Rover 17
4.2.1a Pre-Sorting 24
4.2.1b Post Sorting 29
4.2.2 Machine Learning Models 34
4.2.2a Introduction to YOLO Models 34
4.2.2b Dataset 34
4.2.2c Model Comparison 36
A. Training Losses 37
● Training Classification Loss Graph 37
● Training DFL Loss Graph 38
B. Validation Losses 39
C. Performance Metrics Over Epochs 42
D. Confusion Matrix Analysis 47
E. Waste recognition demo by the ML models 49
4.2.3 Robotic Arm 51
4.2.3a Pre-Sorting 54
4.2.3b post-sorting 58
4.2.4 Structure 62
4.2.4a Pre-Sorting 63
4.2.4.b. Post-Sorting 63
4.2.5 Budget 65
4.3 Identify optimal design approach 66
4.3.1. Pre-Sorting Approach 66
4.3.2. Post-Sorting Approach 67
4.4 Performance evaluation of developed solution 68
4.5 Conclusion 72
Chapter 5: Completion of Final Design and Validation. [CO8] 73

5.1 Introduction 73
5.2 Development and Problem-Solving Phase 73
5.2.1 Mechanical Structure Development 73
a. Mechanical Arm: 73
5.2.2 Image Processing Model Classification 83
5.2 Completion of the Final Design 85
5.2.1 Controlling the Robotic Arm using Arduino UNO 86
5.2.2 Controlling the Rover using ESP32 86
5.2.3 Running YOLO Model and Controlling Microcontrollers using Raspberry Pi5 87
5.2.4 Snippets of the completed system and its running phases. 89
5.3 Evaluate the solution to meet desired need 91
5.4 Conclusion 93
Chapter 6: Impact Analysis and Project Sustainability [CO3, CO4] 94
6.1 Introduction 94
6.2 Assessing the impact of solution 94
6.3 Evaluate the sustainability 95
6.4 Conclusion 97
Chapter 7: Engineering Project Management. [CO11, CO14] 98
7.1 Introduction 98
7.2 Define, plan and manage engineering project 98
7.2.1 Project Timeline 98
7.2.2 Risk Analysis 100
7.3 Evaluate project progress 104
7.4 Conclusion 108
Chapter 8: Economical Analysis. [CO12] 109
8.1 Introduction 109
8.2 Economic analysis 109
8.2.1. Subsystem-wise Breakdown 110
8.2.2. Operational Cost 111
8.2.3. Life Cycle Cost 111
8.2.4. Payback Period 112
8.3 Cost benefit analysis 112
8.3.1. Cost vs Selling Price 112

8.3.2. Competitor Analysis 112
8.4 Evaluate economic and financial aspects 114
8.5 Conclusion 114
Chapter 9: Ethics and Professional Responsibilities [CO13, CO2] 115
9.1 Introduction 115
9.2. Identify ethical issues and professional responsibility 115
9.3. Apply ethical issues and professional responsibility 116
9.3.1. Ethical Consideration (CO13) 116
9.3.2. Safety Consideration (CO3) 116
9.4 Conclusion 117
Chapter 10: Conclusion and Future Work. 118
10.1 Project summary 118
10.2 Future work 119
Chapter 11: Identification of Complex Engineering Problems and Activities. 120
11.1 Attributes of Complex Engineering Problems (EP) 120
11.2. Attributes of Complex Engineering Activities (EA) 120
References 121
Appendix 123
Logbook 123
Related Code/Theory 132
Assessment Guideline for Faculty 150

LIST OF FIGURES

Fig. 2.1 Block diagram of the Pre-sorting approach	27
Fig. 2.2 Block diagram of the post-sorting approach	28
Fig. 4.1 Rover wheel simulation for post sorting design approach	35
Fig. 4.2 Rover wheel simulation for pre-sorting design approach	36
Fig. 4.3 Schematic Diagram of Rover (Top Diagram)	37
Fig. 4.4 Schematic Diagram of Rover (Subsystem Diagram)	37
Fig. 4.5 Variable values for (a) Wheel, (b)Vehicle Body, (c) DC motor, (d) PWM voltage	39
Fig. 4.6 Output curve after simulating the rover circuit	39
Fig. 4.7 Movement pattern with obstacle avoidance mechanism	40
Fig. 4.8 Parameters set by the user	42
Fig. 4.9 Rover's performance matrix (Pre-sorting)	43
Fig. 4.10 Parameters set by the user	46
Fig. 4.11. Rover's performance matrix (Post-sorting)	47
Fig. 4.12. Workflow diagram of choosing the optimal machine learning model	50
Fig. 4.13. (a) Image annotation software (b) Dataset split	53
Fig. 4.14. Training Box Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	54
Fig. 4.15. Training Classification Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n	55
Fig.4.16 Training Dfl Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	56
Fig.4.17 Validation Box Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	57
Fig.4.18 Validation Classification Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	58
Fig.4.19 Validation DFL Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	59
Fig.4.20 Precision Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	60
Fig.4.21. Recall Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	61
Fig.4.22. F1 Score Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	62
Fig.4.23.a. mAp@50 Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	63
Fig.4.23.b. mAp@50-95 Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.	64
Fig.4.25. Recognized waste by YOLOv5n	67
Fig.4.26 . Recognized waste by YOLOv8n	67
Fig.4.27. Recognized waste by YOLOv11n	68
Fig. 4.28. Schematic Diagram of the Robotic Arm System	68
Fig.4.29. Kinematics Block of Robotic Arm	69
Fig. 4.30 Free Body Representation of 3DOF Robotic Arm	69
Fig.4.31 Child Sheet of Robot Block (Pre-Sorting)	71
Fig.4.32 Parameter Block of Proportional-Derivative (PD) Controller (Pre-Sorting)	73
Fig.4.33 Stabilized input signal from PD shown via Scope block (Pre-Sorting)	73
Fig.4.34 Automatic computation block of torque (Pre-Sorting)	74
Fig.4.35 Maximum required torque vs Payload mass (Pre-Sorting)	75
Fig.4.36 Child Sheet of Robot Block (Post-Sorting)	75

Fig.4.37 Parameter Block of Proportional-Derivative (PD) Controller (Post-Sorting) 77
Fig.4.38 Stabilized input signal from PD shown via Scope block (Post-Sorting) 78
Fig.4.39 Automatic computation block of torque (Post-Sorting) 79
Fig.4.40 Maximum required torque vs Payload mass (Post-Sorting) 79
Fig.4.41 Bin with a divider for proper segregation 80
Fig.4.42 Claw End Effector [24] 81
Fig.4.43 Conveyor Belt [24] 81
Fig.4.44 Waste Sweeper End Effector [24] 81
Fig.4.45 Sustainability Matrix Distribution Table 87
Fig.4.46 Spider Chart for Environmental Criteria 88
Fig.4.47 Spider Chart for Economic Criteria 88
Fig.4.48 Spider Chart for Social Criteria 89
Fig.5.1 Completed Arm Structure 92
Fig.5.2 Pinch Type inefficient gripper 93
Fig.5.3 (a) Off-road wheel 130x60 mm (b) Experiment using the off-road wheel 130x60 mm wheels 94
Fig.5.4 Implementation of wooden bar for added stability and new wheels 95
Fig.5.5 Bicycle chain wheel as tire 96
Fig.5.6 Gas cylinder pipe being wrapped around the wheel 97
Fig.5.7 Castor wheels (Flywheels) 98
Fig.5.8 Shielded Wheels 99
Fig.5.9 4 layers of PVC 100
Fig.5.10 (a) Wooden Bar attached (b)Installing freewheels 101
Fig.5.11 Classification of non-biodegradable 101
Fig.5.12 Classification of Biodegradable 102
Fig.5.13 Over Discharge Protection Circuit 103
Fig.5.14 Voltage Regulator Buck Converter 103
Fig.5.15 Queue logic 107
Fig.5.16 Completed System showcasing 4 subsystems. 107
Fig.5.17 Overview of the Completed System showing the compartments of bin. 108
Fig.5.18 Picking of biodegradable waste and dumping it in the respective bin 108
Fig.5.19 Picking of non-biodegradable waste and dumping it in the respective bin 109
Fig.5.20 Charging and Discharging capacity of the battery 109
Fig.5.21 Performance of the waste collected 110
Fig.5.22 Average power consumption in 50-minute runtime. 111
Fig.6.1 SWOT Analysis for Waste Segregation and Collection Rover Project 114
Fig.6.2 Sustainability Matrix Distribution Table 114
Fig.7.1 Fault Tree Analysis 119
Fig.8.1 Competitor Market Analysis 130

LIST OF TABLES

Table 1.1: Applicable standards and codes 6
Table 3.1: Selection of modern engineering tools 13
Table 3.2: Selection of Hardware Tools 14
Table 4.1: Power Budget for Pre-Sorting Approach 28
Table 4.2: Power Budget for Post-Sorting Approach 33
Table 4.3: Types of Non-biodegradable Waste Images in the dataset 35
Table 4.4: Types of Biodegradable waste images in the dataset 35
Table 4.5: Budget of Post-Sorting Method 65
Table 4.6: Budget of Pre-Sorting method 66
Table 4.7: Summary of Design Verification 72
Table 7.1: FYDP-P Timeline 98
Table 7.2: FYDP-D Timeline 99
Table 7.3: FYDP-C Timeline 100
Table 7.4: Fever Chart for Qualitative Assessment 102
Table 7.5: Probability X Impact on the Potential Risk 102
Table 7.6: Responsibility assignment matrix (FYDP-P) 105
Table 7.7: Responsibility assignment matrix (FYDP-D) 106
Table 7.8: Responsibility assignment matrix (FYDP-C) 107
Table 8.1: Initial Bill of material for our solution 109
Table 8.2: Final Bill of Material for Our solution 110
Table 8.3: Subsystem wise breakdown of robotic arm 110
Table 8.4: Subsystem wise breakdown of camera and bin 111
Table 8.5: Subsystem wise breakdown of rover 111
Table 8.6: Cost and selling price comparision 112
Table 10.1: Comparison of Proposed VS Achieved Specification 118
Table 11.1: Attributes of complex engineering problem and reasoning (EP) 120
Table 11.2 : Attributes of complex engineering activities and reasoning (EA) 120

CHAPTER1: INTRODUCTION [CO1, CO2, CO10]

1.1 Introduction

Waste management is a global issue and like many other developing nations, Bangladesh also faces this problem. Due to rapid urbanization, population growth and limited infrastructure, Bangladesh struggles to manage solid wastes effectively, especially in public spaces like parks and fields. Improper disposal of waste degrades the environment, poses risks to public health. The current waste management system requires manual labor, inefficiency and lacks timely segregation of waste. Smart technologies like robotics and ML can make waste management more efficient and sustainable. An autonomous rover which can move across outdoor areas, detect and collect waste, can address the mentioned problems. By sorting the waste on-site, it can minimize the labor of waste segregation and recycling. This project proposes building such a rover for urban parks and fields in Dhaka using robotics, machine learning and sensors. This rover aims to support cleaner environments and promote sustainable waste management.

1.1.1 Problem Statement

In Bangladesh, Dhaka alone generates about 5,000 to 6,500 tons where 55% waste is uncollected which leads to widespread environmental pollution and health risks [1][2]. This issue is further exacerbated by improper waste management systems. A paper by Anitha G, M. Priscilla and Ramachandran. R states that manual sorting in India leads to inefficiencies, overflowing trash and litter [3]. Moreover, weak institutions, governance issues and under-resourcing complicate effective waste collection and disposal in developing urban areas [4]. The issue also stems from the personal attitude of open dumping in other's space, for which authorities face challenges in implementing 3R (Reduce, Reuse, Recycle) waste management strategies [5]. All these issues highlight the need for a planned waste management system which can be achieved by implementing smart technologies like sensor-equipped bins, automatic waste segregation, collection and disposal model etc. [1][3]. This project aims to address these issues as waste generation in Dhaka is expected to reach 8,500 tons by 2032 according to THX News. Therefore, an improved sustainable waste management method is a necessity [2].

1.1.2 Background Study [Rephrased most of the part, so copy paste this section]

The literature review provides in-depth insights into advancements in robotic waste segregation, showing how integrating artificial intelligence (AI) and robotics can address rising waste management challenges effectively. The RECLAIM project offers a comparative study of different types of robotic grippers, such as rigid and soft grippers, tailored to efficiently sort common recyclable materials like PET and metals in waste streams. This project emphasizes the importance of selecting the right gripper mechanism to ensure durability and adaptability in handling various packaging waste types, making it suitable for large-scale recycling applications [6]. In a separate study, Lahoti et al. propose a multi-class waste segregation model that leverages YOLOv5, a single-shot object detection algorithm, alongside a 3D-printed robotic arm for enhanced accuracy in sorting dry waste materials like paper, plastic, metal, and glass. Their system achieves high success rates within controlled environments, making it a promising solution for automated waste sorting tasks [7]. Another project supporting India's Swachh Bharat initiative showcases a smart waste sorting system that uses OpenCV and a combination of IR and proximity sensors with a robotic arm. This setup enables automatic classification of dry, wet, plastic, and metal waste, effectively streamlining the sorting process and enhancing recycling efficiency [8].

Additionally, Kulshreshtha et al. present the Outdoor Autonomous Trash-Collecting Robot (OATCR), which employs the YOLOv4-Tiny object detection model optimized for real-time trash detection in outdoor settings. This robot incorporates the Rocker-Bogie mechanism for robust mobility, allowing it to navigate diverse terrains such as parks and beaches. The robot's path-planning capability enables autonomous trash collection with obstacle avoidance, making it highly adaptable for open-space waste management [9]. Lastly, Ibrahim et al. introduce an automated waste segregation system that integrates a YOLOv6 object detection model and a robotic arm controlled by inverse kinematics. This system's design emphasizes hardware efficiency and achieves real-time performance by swiftly sorting objects into designated categories. By focusing on optimizing the robotic arm's joint angles, the system minimizes processing time and enhances accuracy in waste classification, making it a cost-effective solution for real-world applications [10]. Collectively, these studies highlight the advancements in waste segregation technologies, demonstrating how AI and robotics can improve operational efficiency, accuracy, and adaptability across diverse waste management scenarios.

Building on these applications-focused advances in AI-driven sorting and mobile collection, it is equally important to verify that the underlying robotic, navigation, sensing, and power-management subsystems can be implemented with reliable engineering methods and tools. Recent studies substantiate each subsystem of the proposed waste-segregating rover. Zhou et al. developed a hospital-service mobile manipulator whose forward/inverse kinematics and PD control were realized in MATLAB/Simulink with Simscape Multibody, confirming the effectiveness of the same toolchain for our 3-DOF arm design [11]. Zhang et al. likewise simulated a 3-DOF arm in SimMechanics and demonstrated precise path tracking with a fuzzy-PD controller, reinforcing the value of PD-based motion smoothing for pick-and-place tasks [12]. For actuator sizing, Oni et al. showed that joint-torque computation under worst-case payloads are essential for motor selection [13]. Chen et al. equipped a differential robot with eight ultrasonic sensors arrayed around the chassis for obstacle perception [14]. A* remains the de-factor grid planner for obstacle-aware routing, as illustrated by its optimization in smart-agriculture robots [15]. Faigl et al. shows boustrophedon (zig-zag) coverage scheme, which proves that complete area sweep can be done with minimal redundancy [16]. Moreover, Liang et al. demonstrate that NIR absorbance fingerprints enable rapid, non-contact identification of common polymers such as PET, HDPE, and PP, making it ideal for high-speed automated classification in mixed plastic wastes [17]. Finally, energy-oriented works by Hou et al. and by Zhang et al. emphasize that accurate EPS models must integrate mechanical, drive-train, and electronic loads to predict battery endurance and enable energy-optimal motion control [18], [19]. Collectively, these sources provide a solid theoretical foundation for the design choices made in the robotic arm, autonomous navigation, sensing, and EPS subsystems of the rover discussed in the later part of this report.

1.1.3 Literature Gap

The existing solutions have different approaches like robotic grippers [6], object detection with robotic arms [7][8][10], and outdoor trash-collecting robots with advanced mobility [9]. However, most of the systems are focused on a single subsystem and are tested in a controlled environment. Furthermore, for autonomous navigation, use of ultrasonic sensors, A* path planning and coverage algorithms are shown [11][16]. Some simulations have been modeled for energy-efficient power management [18][19]. Still, combining all these works for waste detection, on-site segregation and autonomous navigation has not been done in most of the cases.

Current systems often overlook the need for waste management in public spaces of developing countries. For example, near-infrared spectroscopy can rapidly detect polymers [17]. Still, it has not been integrated into compact overs for waste segregation yet. Hence, a gap exists in developing a fully autonomous rover for waste collection and segregation in outdoor environments like urban parks and fields in Bangladesh.

1.1.4 Relevance to current and future Industry

To improve waste management efficiency, several companies are investing in robotics and AI. A Finnish company **ZenRobotics** uses AI-powered robotic arms to sort construction and demolition waste by reducing human risk [20]. A company named **AMP Robotics** based in the USA uses computer vision machine learning to automatically identify and separate recyclable materials like plastics, metals and paper with high accuracy [21]. **Waste Robotics** is a company that builds AI-powered robotic systems for waste sorting by using their "AI-Gripper" technology. They help recycling facilities automatically sort construction & demolition debris, metals, plastics, and other materials more efficiently and with higher purity [22]. All these examples show that use of AI and robotics in waste management is increasing day by day.

Bangladesh has a huge population which contributes to a large amount of waste generation. Hence, smart technological adoption is necessary in Bangladesh, especially urban areas like Dhaka. As the current waste collection and management is inefficient, the proposed project offers a cost-effective autonomous rover which collects and segregates waste on-site. In future, projects like this can be scaled to reduce dependency on manual labor, improve public health and the environment. Finally, the global industries are shifting towards circular economical practices where Bangladesh can join as well. This can be done by adopting automated waste management systems, creating opportunities for local manufacturing, research collaborations and green entrepreneurship.

1.2 Objectives, Requirements, Specification and constraint

To solve the identified problem, the solution must have a certain objective. These objectives are met through a set of requirements which will allow us to keep track of whether the objectives are met. This allows us to design a systematic way of solving the problem.

1.2.1. Objectives & Scope

Objectives: The expected results of this project can be defined as objectives. The objectives of this project are:

- Design and develop an autonomous robotic system that collects waste for segregation into Biodegradable and Non-Biodegradable waste.
- Integrate a compact storage system for segregated waste to hold collected waste until disposal.
- Reduction of environmental impact through on-site waste segregation.

Scopes: The tasks and deliverables that are necessary to complete this project are known as the scopes which also define the boundaries and the extent of the project.

The scope of this project lies within the development of an autonomous waste-collecting rover for urban parks and fields. The rover has to be capable of navigating terrains and collecting waste. To detect different types of waste, an ML classification model needs to be developed. For the waste sorting and storage, mechanical components must be integrated. Finally, these need to be validated through testing. However, the project does not include post-disposal waste processing, waste incineration or marketing activities for the system.

1.2.2 Functional and Nonfunctional Requirements

Requirements: Every project has specific goals or conditions that they must meet to achieve its intended purpose; this technicality is known as requirements. For this project, the requirements are mainly divided into two types:

- **Functional Requirements:** Requirements which define what the project is supposed to do. It describes core attributes needed to fulfill its purpose. The functional requirement of this project is:
    1. Traverse multiple terrains with advanced tracking
    2. Autonomous waste collection
    3. Accurate waste classification via ML model
    4. Detects, sorts and categorizes waste
- **Non-Functional Requirements:** It defines the “how it should perform” aspect of the project. It consists of factors which contribute to the optimization of the project overall. The non-functional requirements are:
    1. Notifies users about environmental conditions (e.g., weather, terrain)
    2. Sends real time status alerts such as estimated time till completion, percentage of Non-Biodegradable /Biodegradable waste collected etc.
    3. Sends alerts to a mobile app when the bin is full

1.2.3 Specifications

Specifications outline the details and measurable criteria that the system must meet. It acts as a guide to develop the system into its intended design and functionality.

- The rover operates at a controlled speed of **0.5m/s** to **1.5m/s**, suitable for all-terrain exploration.
- The system operates independently where it operates autonomously for up to **30 minutes** on a full charge, navigating the terrain without human intervention by utilizing onboard sensors and algorithms to detect and avoid obstacles with a response time of **0.5 seconds.**
- The system classifies waste with a minimum **accuracy rate of 95%** between Biodegradable and Non-Biodegradable materials.
- The collector takes lightweight objects ranging from **500g to 3 kg**.
- The rover is capable of handling elongated objects with dimensions ranging from **10cm to 20 cm** in length and diameter.
- The waste collector performs with a **margin of error of 0.5%**, ensuring reliability and precise handling.
- Each bin accommodates a **total waste volume between 2000 and 2500 cm^3** before requiring emptying.
- The combined weight capacity for the bins is limited to **5 kg**, while the entire rover system maintains a weight threshold of no more than **40kg**.
- Specifically designed for urban parks and fields, with an effective operational area of **half an acre.**

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

Constraints refer to limitations or restrictions that may hinder the performance of the project in a particular capacity.

- **Blurry Camera:** The camera used for classifying waste may get dirty or smudged during operation
- **Terrain Adaptability:** Uneven or wet terrain may affect the rover's mobility and stability.
- **Public Interaction:** Operating in public spaces may lead to interference by humans or animals

1.2.5 Applicable compliance, standards, and codes

TABLE 1.1: APPLICABLE STANDARDS AND CODES

IEEE Standard	Description	Application to AREWS Project
IEEE 1872-2015	Ontologies for Robotics and Automation	Defines a standardized vocabulary for waste types and sensor integration, enhancing waste recognition and categorization.
IEEE 2413-2019	Standard for an Architectural Framework for the Internet of Things (IoT)	Supports IoT-based components for real-time monitoring, remote access, and control of the rover within urban waste management systems.
IEEE P2020	Standard for Automotive Image Quality	Ensures high-quality images for accurate waste identification, crucial for machine learning processing of waste types in real time.
IEEE 802.15.4	Low-Rate Wireless Personal Area Network (LR-WPAN) Standard	Supports low-power wireless communication for real-time updates, alerts, and status reports from the rover.
IEEE 1012-2016	Standard for System, Software, and Hardware Verification and Validation	Validates the rover's systems to ensure they function correctly and reliably in various conditions, enhancing confidence in performance.
IEEE 1228	Standard for Software Safety Plans	Mitigates risks of harm in urban settings by ensuring the rover's software includes safety protocols, preventing accidents with humans or animals.
IEEE 1680.1	Standard for Environmental Assessment of Electronic Products	Promotes eco-friendly design by encouraging sustainable material use and energy-efficient construction.
IEEE 802.11	Wireless LAN (Wi-Fi) Standards	Provides high-bandwidth communication for real-time monitoring and control, useful if LR-WPAN is insufficient for certain tasks.
IEEE 1474	Standards for Communications-Based Train Control (CBTC) Performance and Functional Requirements	Enhances navigation and control in crowded environments, ensuring the rover can safely plan paths and avoid collisions.
IEEE 828	Configuration Management in Systems and Software Engineering	Enables version control and tracking system modifications, particularly as the rover's functionality evolves.
IEEE C95.1	Safety Levels with Respect to Human Exposure to Electromagnetic Fields	Ensures the rover's wireless components emit safe levels of electromagnetic radiation, protecting nearby humans and animals.

IEEE 1500	Standard for Embedded Core Test	Provides a framework for testing embedded systems, such as microcontrollers, to ensure reliable operation in variable conditions.
IEEE P1220	Standard for Application and Management of Systems Engineering Process	Guides the full engineering process from design to maintenance, improving the reliability and performance of the autonomous rover.
ASTM F2398-20	Standard Specification for Safety and Performance of Robotic Lawn Mowers	Enhances the rover's safety, ensuring it operates reliably and adapts effectively to varying terrains and urban conditions.

1.3 Systematic Overview/summary of the proposed project

Bangladesh faces several challenges in waste management, leaving large amounts of waste being uncollected contributing to environmental pollution and public health risks. To tackle this, we propose an autonomous system that will detect and classify the waste, collect it and segregate it accordingly. The system uses a Raspberry Pi 5 with a Pi camera and a Yolov8 model which is used to classify the waste into two categories Biodegradable and Non-Biodegradable. The rover moves in a fixed zigzag pattern, equipped with a robotic arm along with an ultrasonic sensor. When both the camera and ultrasonic sensor detect waste, the rover stops moving, initiating the pick-up process. Based on the classification data, the bin rotates ensuring proper segregation is done prior to disposal. Once the waste is cleared, the rover continues its navigation till the entire field is covered. Through the implementation of this system, a cleaner environment and improved public health can be ensured.

1.4 Conclusion

The primary outcome of this project is the development of a fully operational autonomous rover capable of performing waste segregation and collection with high efficiency in urban parks and open fields. The system is expected to classify waste into Biodegradable and Non-Biodegradable categories with an accuracy of 95%, ensuring reliable performance even under diverse environmental conditions. Furthermore, the rover will navigate autonomously at a controlled speed of 0.5–1.5 m/s across various terrains, demonstrating robust adaptability. Segregated waste will be stored securely in designated bins, with sufficient capacity to minimize environmental pollution and enhance waste management practices.

To conclude, this project will go through modular design and development, extensive trials for performance evaluation while keeping sustainability and efficiency in mind. It will also adhere to public safety standards. By achieving these outcomes, the project will not only promote cleaner urban environments but also advance sustainable waste management practices aligned with societal, environmental, and legal standards.

CHAPTER 2: DEVELOPING THE SOLUTION: METHODOLOGY AND APPROACHES

2.1 Introduction

To develop a robust design of tackling a global challenge of managing waste, the process starts by identifying core issues and revolving around those issues to come up with solutions. Some of the generic problems found in these categories are:

a. A chunk of time being spent with limited efficiency while doing manual segregation.
b. Lack of knowledge to control remote oriented systems.
c. Need for energy efficient operations.
d. Balance between effectiveness of a system or a program and the cost or expenses behind it.

2.2 Identify multiple design approach

Through the recognition of these identified problems, the design approaches are tailored respectively. After doing thorough research and literature review, two main design approaches were deduced to solve the problem. As the project is about sorting waste, the solution focused on the ways waste can be sorted. Keeping that in mind, the approaches are:

I) Pre-Sorting Approach.
II) Post Sorting Approach.

The Pre-Sorting approach first sorts the waste before collecting and putting it into the bin. The idea is to first identify and classify the waste on the ground and then put it in its designated compartment. Alternatively, the Post-Sorting Approach as the name suggests, sorts the waste after collecting. In short, it will collect all the waste together and then sort it to its designated bin.

2.3 Describe multiple design approach

2.3.1 Pre-Sorting Approach:

The overall system as shown in **Fig. 2.1,** is divided into three main subsystems: Camera, Rover and Robotic arm. The entire system operates by the camera initially detecting waste within its field of vision using advanced image recognition algorithms; however, unlike the post sorting design, it streamlines the sorting process by integrating classification into the initial detection phase. The camera system operates using object classification ML models on board that classify waste as either Biodegradable or Non-Biodegradable. Once the waste is identified and categorized, the rover navigates towards it using a pathfinding algorithm all while avoiding obstacles. Upon arriving near any classified waste, the robotic arm is activated to collect it. The arm collects the waste and places it in the round bin with a divider, dividing Biodegradable and Non-Biodegradable waste. The classification data generated by the camera system is fed to the bin system and based on it the bin is rotated in 180° clockwise direction carefully segmenting Biodegradable and Non-Biodegradable waste. Furthermore, the Alert subsystem ensures that the monitoring of the capacity of the waste bin is detected on time for user's convenience. The system is programmed in such a way that, when the bin becomes full, an alert is sent to every device that it is logged in to via IoT. This integrated system automates the waste management process, ensures accurate sorting, promotes sustainability and prevents overflow by facilitating proper recycling and disposal methods all while having a compact architecture.

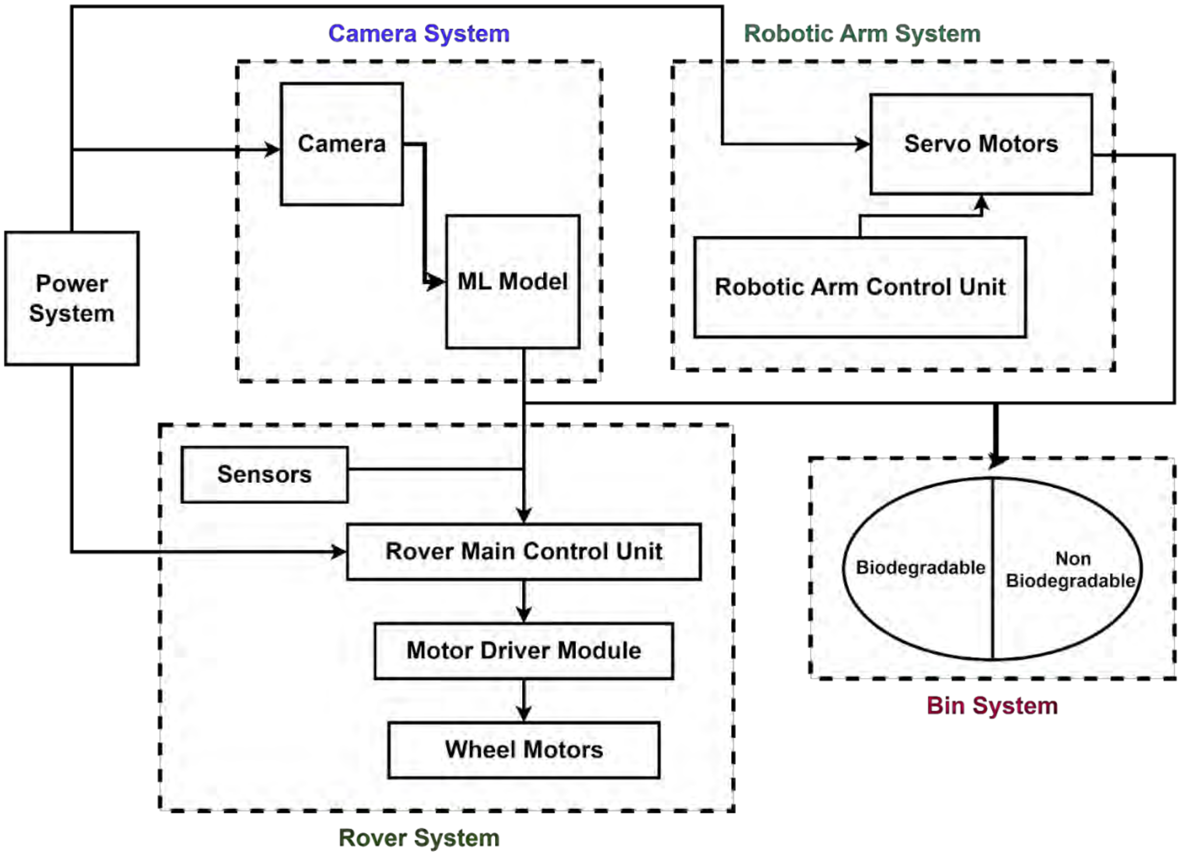


Fig. 2.1 Block diagram of the Pre-sorting approach

The pre - sorting system incorporates multiple subsystems working in tandem. A Li-Po battery powers the setup. For waste detection and classification, a Raspberry Pi processes the machine learning algorithm, aided by an 8-megapixel camera. Ultrasonic sensors and an IMU provide navigation and stability, while the robotic arm servo motors for waste manipulation and sorting. Linear actuators assist in precise arm movement. The system is controlled by an Arduino Mega, integrating inputs from sensors and managing motor drivers for mobility. The sorted waste is deposited into the dedicated bins.

2.3.2. Post Sorting Approach:

The overall system as shown in **Fig. 2.2,** is divided into five main subsystems: Camera, Rover, Robotic Arm, Waste collection and Alert system. The system is designed for efficient waste collection and sorting. Initially, the camera of the subsystem detects waste on the ground within its field of vision as the rover navigates. With image recognition machine learning algorithms, it classifies the waste based on its external features such as shape and then relays that information to the rover main control unit. The rover is set in such a way that it will navigate the field at a certain pattern, and upon receiving this information, the rover navigates towards it using a pathfinding algorithm all while avoiding obstacles. Once the rover is positioned near the waste, it activates the robotic arm. The arm collects the waste and places it in the conveyor belt. The collected waste is then transported via the conveyor belt to the round bin with a divider, dividing Biodegradable and Non-Biodegradable waste. Between this is the sorting area where there is a stationary near-infrared (NIR) sensor that scans and analyzes the waste based on light reflection it emits and categorizes it as either Biodegradable or Non-Biodegradable. Based on its classification, the bin is rotated in 180° clockwise direction carefully segmenting Biodegradable and Non-Biodegradable waste. Furthermore, the Alert subsystem ensures that the monitoring of the capacity of the waste bin is detected on time for user's convenience. The system is programmed in such a way that, when the bin becomes full, an alert is sent to every device that it is logged in to via IoT. This integrated system automates the waste management process, ensuring accurate sorting, promoting sustainability and preventing overflow by facilitating proper recycling and disposal methods.

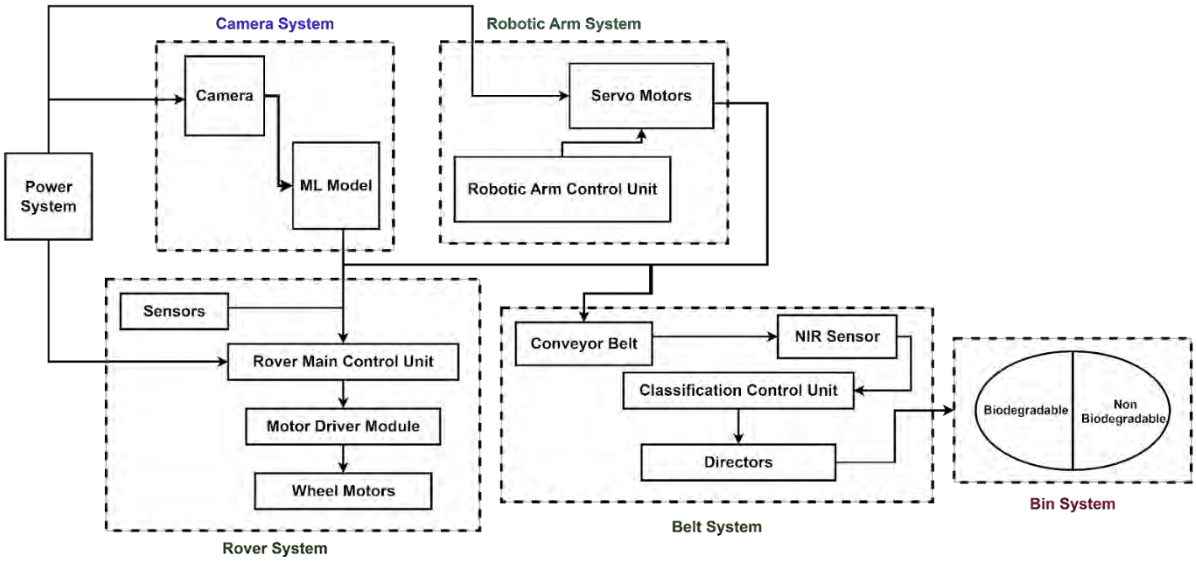


Fig. 2.2. Block diagram of the post-sorting approach

The post-sorting system integrates multiple subsystems working seamlessly. A Li-Po battery powers the entire setup. A conveyor belt system, driven by a 1-2 HP motor and supported by rollers, transports waste for classification. Near-Infrared (NIR) sensors detect and classify wastes while a Raspberry Pi processes classification data. The sorting mechanism uses linear actuators and servo motors to direct the waste into appropriate bins. An Arduino Mega coordinates sensor inputs, motor drivers, and control units to ensure smooth operation. The system efficiently deposits sorted waste into their dedicated compartments.

2.4 Analysis of multiple design approach

As discussed in the previous section, the waste management problem can be solved in many ways in urban parks and fields. The solution has been narrowed down to an autonomous rover system which will traverse through the target demographic to detect, classify and collect waste and put it into its designated place. In this case, the multiple method has been deduced to the

waste sorting method: Pre-Sorting and Post-Sorting. The analysis between both the approaches for now will be based on the literature review and research.
First, judging by the waste collection method, the Post-Sorting method seems faster as it will collect all the waste together and then put it to the conveyor belt to be sorted. In contrast, the pick and drop method of Pre-Sorting will take longer. However, if power consumption is accounted for, the Post-Sorting method will draw more power than the Pre-Sorting Method.

Next, if budget is considered, the Pre-Sorting method is cheaper. The Post-Sorting method will have more or less the same components as Pre-Sorting but with an additional conveyor belt which will increase the weight, cost and complexity. This will also mean the Post-Sorting rover will be bigger in size so it will have less mobility. From a sustainability perspective, while Pre-Sorting leads to greater efficiency, cost reduction, and environmental benefits, the Post Sorting Approach contributes positively to local economic development and social outcomes. Each approach has its strengths and trade-offs, and the choice between the two depends on the priorities of the specific waste management scenario whether the focus lies on minimizing environmental impact or maximizing economic and social benefits.

Finally, based on only research it can be deduced that the Pre-Sorting method seems more feasible. However, only after thorough evaluation and software validation can the final decision be made.

2.5 Conclusion

The design and development of the autonomous rover for waste segregation and collection represents an innovative solution to urban waste management. With the integration of Machine Learning and Mechanical aspects, the rover ensures efficient waste collection, accurate waste classification and navigation through terrains.
The system's design prioritizes sustainability, safety, and environmental impact, which aligns with global goals for cleaner cities and healthier communities. It also addresses challenges like public health risks, manual labor issues and promotes responsible waste disposal practices.
To conclude, it can be said that the successful implementation of this rover can be a sustainable and novel solution which can be scaled as per requirements. This might ensure a smarter waste management system in urban settings like Dhaka and beyond.

CHAPTER 3: USE OF MODERN ENGINEERING AND IT TOOL. [CO9]

3.1 Introduction

Due to advancements in technology, there are various ways of validating any proposed design without losing much during the research and development phase. This chapter is based on the selection and usage of multiple modern engineering and IT tools used for developing and simulating “An Autonomous Rover for Efficient Waste Segregation and Collection in Urban Parks and Fields.” The tools discussed will be used in software simulation, development and hardware implementation of the project. The following sections delve into the specifics of each tool's purpose and its comparative advantages within the chosen toolset.

3.2 Select appropriate engineering and IT tools

Based on background research, there are numerous software and hardware platforms required to ensure proper simulation, comparison and implementation of the project. Therefore, various software tools were used to figure out the best, suitable and efficient one among both design approaches. All the softwares, IT tools which have been used are listed below.

- MATLAB
- MATLAB Simulink
- MATLAB Simscape
- Proteus
- Arduino IDE
- Solidworks
- Visual Studio Code
- Anaconda
- Raspberry Pi OS
- TigerVNC

These softwares helped design relevant subsystems for the entire project. First of all, the rover itself had different parameters to simulate. Rover’s motion, navigation and stability play an important role in ensuring that the rover can navigate through the target area without any hurdle. The rover also needs to be powered through a battery. For that calculation relevant to EPS was necessary to understand how much power consumption will be done and which battery to choose. This is just an example of how different components were used, based on which final component and budget was revised. Furthermore, Rover will have a robotic arm to pick and place waste according to the design approaches. These physical components are not feasible to test in hardware settings because it is a matter of trial and error. For that, simulation software came in handy. Finally, one of the most important parts, the waste classification model is solely software relevant work. The machine learning model was trained and validated to see if the specifications can be matched or not.

3.3 Use of modern engineering and IT tools

Software Tools:

All the above-mentioned software was chosen carefully from among a lot of already existing software. The software was used to validate and choose the optimal solution. The rationale behind choosing the above-mentioned software are summarized below:

TABLE 3.1: SELECTION OF MODERN ENGINEERING TOOLS

Software	Pre-Sorting	Post Sorting	Why This Tool Was Chosen
MATLAB	✓	✓	- Ideal for mathematical simulation -Used for EPS related calculation -Used for stability and navigation analysis
MATLAB Simulink	✓	✓	- Effective for dynamic system simulation -Supports robotic arm and rover motion modeling -Enables stability testing
MATLAB Simscape	✓	✓	- Simulates multibody dynamics - Ideal for robotic arm torque computation -Validates electromechanical design - Used for stability analysis and robotic arm
Proteus	✓	✓	- Faster than physical prototyping - Simulates rover wheel circuits - Tests motor driver interactions
Arduino IDE	✓	✓	- Communicates with Arduino Uno R3 - Simplifies motor control coding - Integrates sensor and actuator modules
Solidworks	✓	✓	- Designs physical structures (bin, end effectors) - Enables dynamic simulation - Optimizes manufacturability
Visual Studio Code	✓	✓	- Used to run ML Python Script -Trains YOLO ML models
Anaconda	✓	✓	- Supports Python-based image labeling tools - Creates environment for ML model

roboflow	✓	✓	- Helped prepare dataset
Raspberry Pi OS	✓	✓	- A Linux-Based Operating system to run Pi - Allows to use Raspberry pi as a computer
TigerVNC	✓	✓	- Lets control Raspberry Pi remotely - Allows to see Pi camera feed remotely

Hardware Tools:

TABLE 2.2: SELECTION OF HARDWARE TOOLS

Hardware Components	Pictures of the components	Why This Tool Was Chosen
Raspberry Pi 5		Parent microprocessor. It controls two microcontrollers along with a Pi camera. High processing speed required for the working of ML models.
PI camera		It's used to detect and classify waste as part of the system's core function. This is necessary for the machine learning (ML) model running on the Raspberry Pi 5. It is specifically designed to interface directly with the Raspberry Pi 5 via its camera port. High Processing Speed: The camera provides the visual data that requires the high processing speed of the Raspberry Pi 5 for the real-time or near real-time execution of the ML models.
ESP32		Controls the rover's motion. Has more SRAM which is needed for autonomous feature
Arduino Uno		To control the robotic arm
Voltage Regulator Buck Converter		Stable high voltage and current is required by different subsystems. This buck converter

		provides that current through the help of a battery.
Motor Driver 43A BTS7960		Controls the gear motor. It has a high current rating and is also used for speed control of the gear motors.
37GB 12V High Torque Gear Motor		High torque motor needed for the navigation of the rover.
DS Servo DS3218MG High torque motor		High torque servo motor needed for the robotic arm to be able to collect more heavier objects.
6DOF Aluminum Mechanical Robotic Arm Manipulator Kit		Used to collect waste and dispose it in the respective bins
6200 Mah Tiger 14.8V 4S Lipo Battery		The system was in need of high current and higher runtime hence this battery was chosen.
16 Channel 12-bit PWM / Servo Driver 12C Interface PCA9685		To Provide 16 dedicated PWM channels to control multiple servos for the complex robotic arm, as the Arduino Uno has a limited number of PWM pins. To offer 12-bit resolution for precise positioning and smooth movement of the servos. To use the I2C interface (SDA and SCL) to control all 16 outputs with only two pins, saving valuable GPIO pins on the Arduino Uno. To relieve the Arduino Uno of the computational burden of generating 16 precise PWM signals, allowing it to focus on the main control logic.

3.4 Conclusion

The integration of advanced tools like MATLAB suites, Visual Studio Code, Proteus, Arduino IDE and SolidWorks helped in rigorous design validation and simulation, ensuring the solution's technical robustness. In contrast, Arduino IDE, Raspberry Pi OS and TigerVNC are used for both software simulation and hardware implementation. This design not only fulfills functional and sustainability goals but also aligns with broader urban development and environmental protection objectives.

CHAPTER 4: OPTIMIZATION OF MULTIPLE DESIGN AND FINDING THE OPTIMAL SOLUTION. [CO7]

4.1 Introduction

Engineering design always has multiple possibilities. Each of the designs has their own strengths and weaknesses which can be determined only through systematic experimentation and evaluation. In this chapter, previously mentioned design approaches have been explored by performing simulation. It allowed us to measure both design methods' performance to meet the project requirements. The results helped in selecting the design that best meets our requirements, balancing the performance, cost and practicality of the project. Then, the chosen design is optimized further in preparing the prototype development.

4.2 Optimization of multiple design approach

4.2.1 Rover

A. Rover Motion

To simulate how the rover's wheels will work for both design approaches, a proteus simulation was conducted. **Fig. 4.1.** below is the schematic diagram for the simulation

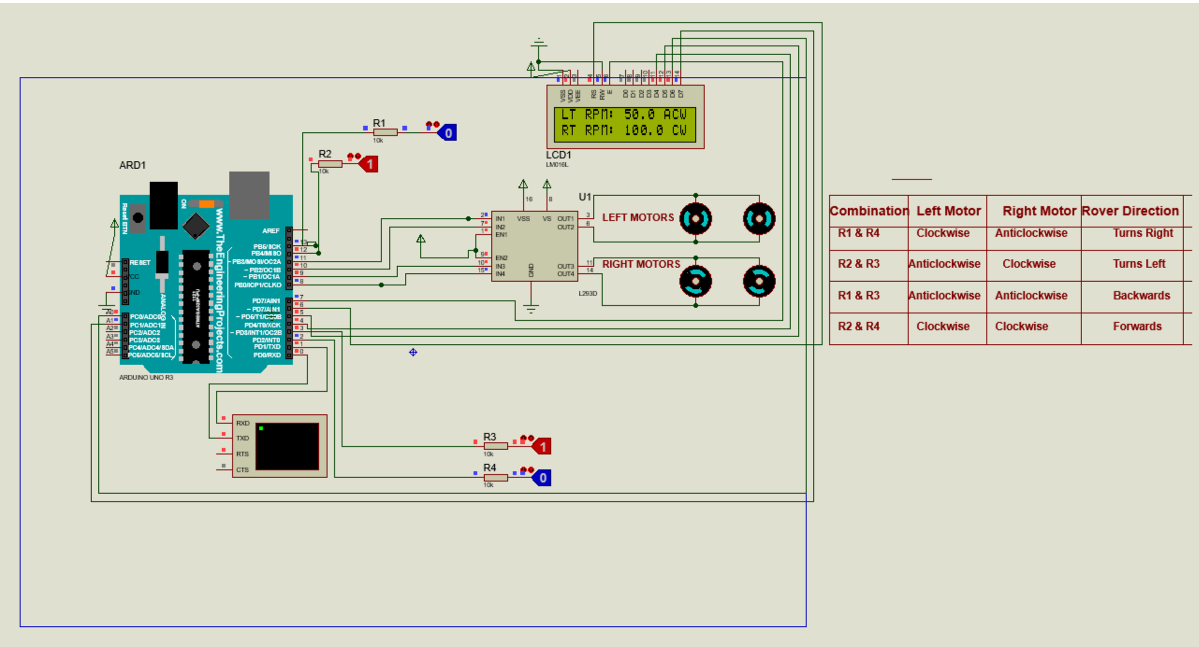


Fig. 4.1. Rover wheel simulation for post sorting design approach

For the post sorting design approach, the rover will have four motors. Four motors were chosen due to the higher mass of the entire rover. To simulate the rover motion, an Arduino Uno R3 was used. The UNO R3 is used to replicate the Raspberry Pi 5's signals that will be sent to the motor driver which in this case is the L298 H-Bridge Dual Motor driver. Four toggle switches:

R1, R2, R3 and R4 are used to replicate the signals from the microcontroller. The signals will be sent as pairs were sending R1 & R4 makes the left side motors of the rover rotate clockwise and the right-side motors of the rover to turn anti clockwise which in turn creates a turning effect making the rover turn right. To reduce the turning radius, the RPM of the left and right-side motors are adjusted accordingly where the left side motors will be at full speed, and the right-side motors will be at half speed. Consequently, sending R2 & R3 makes the left side motors of the rover rotate anti clockwise and the right-side motors of the rover turn clockwise which in turn creates a turning effect making the rover turn left. Like turning right, the RPM of the motors are adjusted where the Right-side motors will be at full speed and the left side motors will be at half speed. To make the rover go backwards, R1 & R3 will be toggled which makes both the left side and right-side motors rotate anticlockwise making the rover go backwards. To make it go forwards, R2 & R4 will be toggled which makes both the left side and right-side motors rotate clockwise making the rover go forwards.

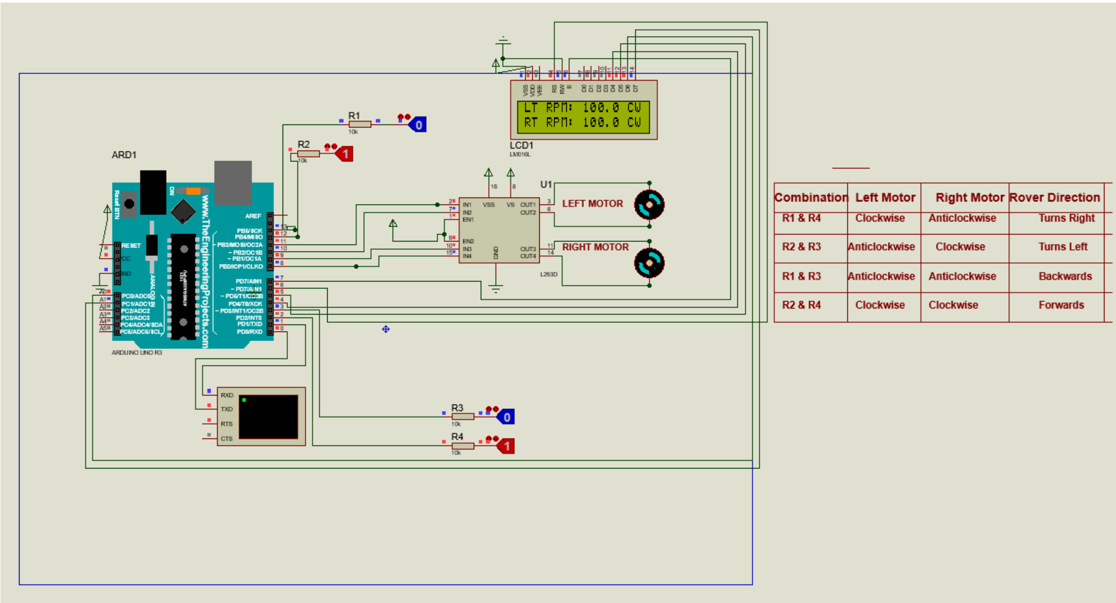


Fig. 4.2. Rover wheel simulation for pre-sorting design approach

Rover wheel simulation of the pre-sorting approach is shown in **Fig.4.2**. For the pre-sorting design approach, since the mass is less compared to the post sorting design approach, we only require two motors. The rest of the functionality remains the same as the post sorting wheel simulation.

B. Stability

The stability of the rover is evaluated based on its velocity and position control under a standard input. Stability is an electromechanical system which refers to the ability of the rover to maintain controlled and predictable behavior when it is given various inputs or disturbances. It is essential for a rover operating in a dynamic urban environment to provide a smooth response without any delay or oscillation to any changes in control signal like speed or direction. A stable system provides accurate navigation, minimizes the risk of mechanical strain or failure.

To evaluate the stability of our rover, an electromechanical system was developed in MATLAB Simulink and Simscape. A step input signal was applied to assess the rover's behavior, specifically its ability to maintain stability without any overshoot. The following **Fig. 4.3.** shows the top-level schematic model used for this purpose.

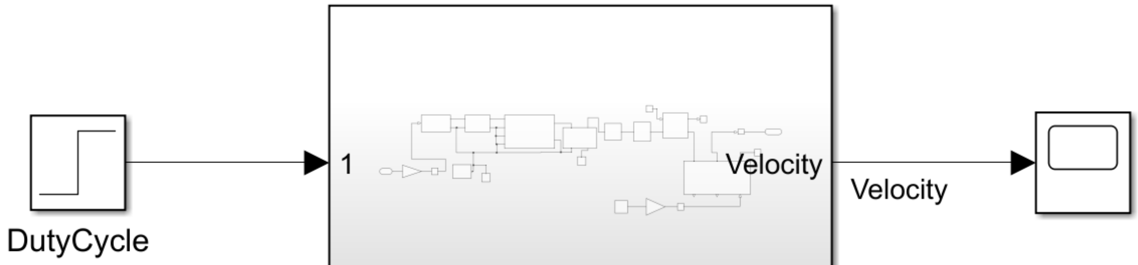


Fig. 4.3. Schematic Diagram of Rover (Top Diagram)

Here, the main motive is to analyze the performance of the rover from the perspective of this step signal. From the scope, it can be seen if the step signal is getting distorted or not.

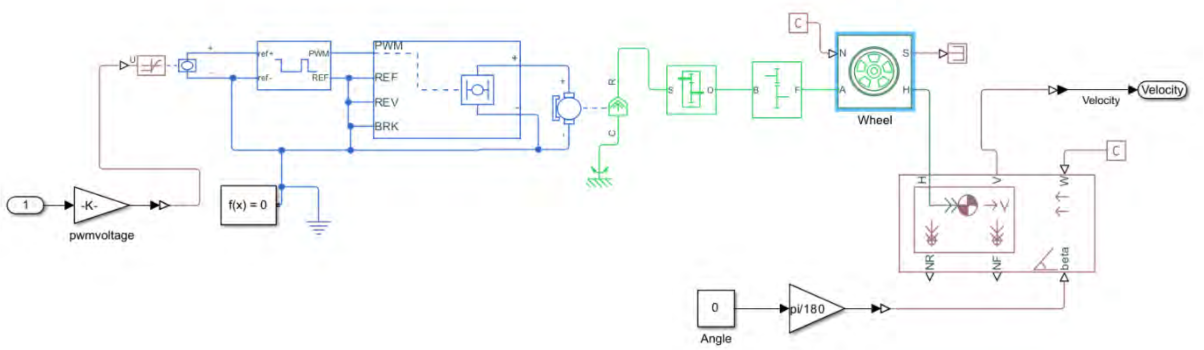


Fig. 4.4. Schematic Diagram of Rover Bot (Subsystem Diagram)

As shown in **Fig 4.4.**, the subsystem represents the core electromechanical design of the rover. The wheel is driven by a DC motor through a gear mechanism, with system control regulated via PWM signals processed by a PS sensor and delivered through an H-Bridge. Motion feedback from the wheel is captured through a sensor to measure dynamic response.

All key parameters for mechanical and electrical components were defined as variables to ensure flexibility and reusability across different design iterations. The parameters are as follows:

Mechanical Parameters (MATLAB Editor):

- Mass = 1
- Radius= 0.3

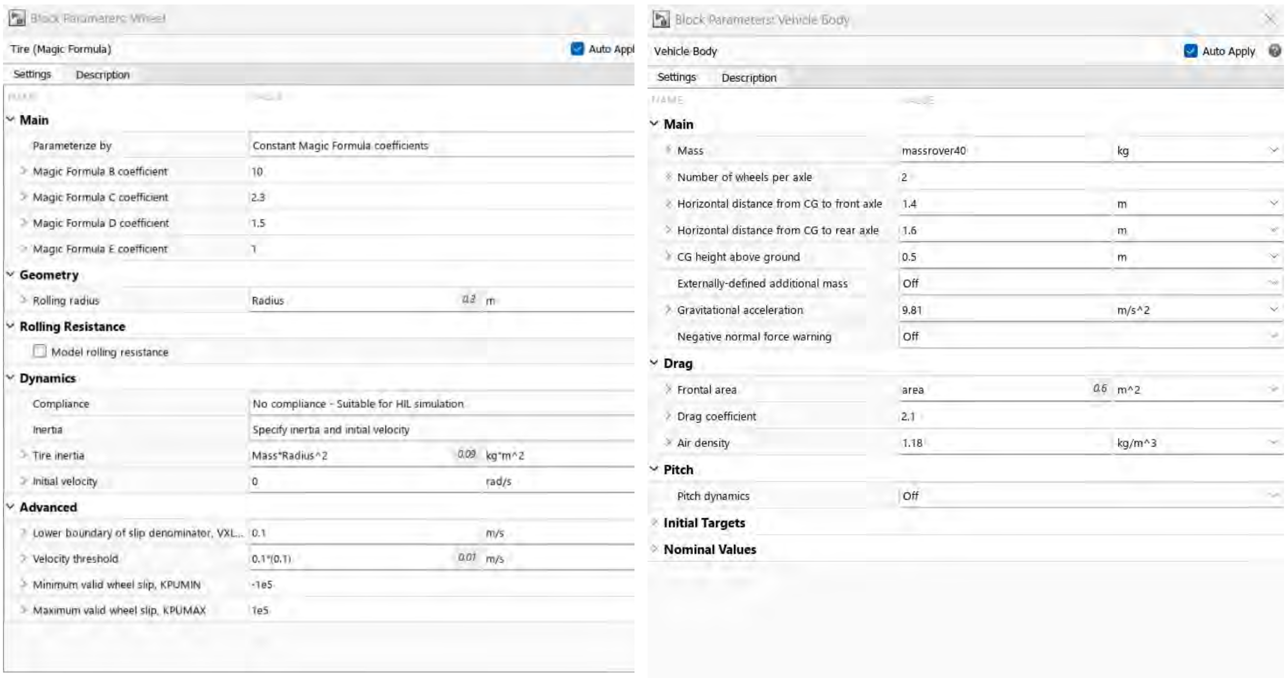
- timestep = 2
- TorqueF = 0.05
- massrover = 40
- area=0.6
- windspeed=0;

Electrical Parameters (MATLAB Editor):

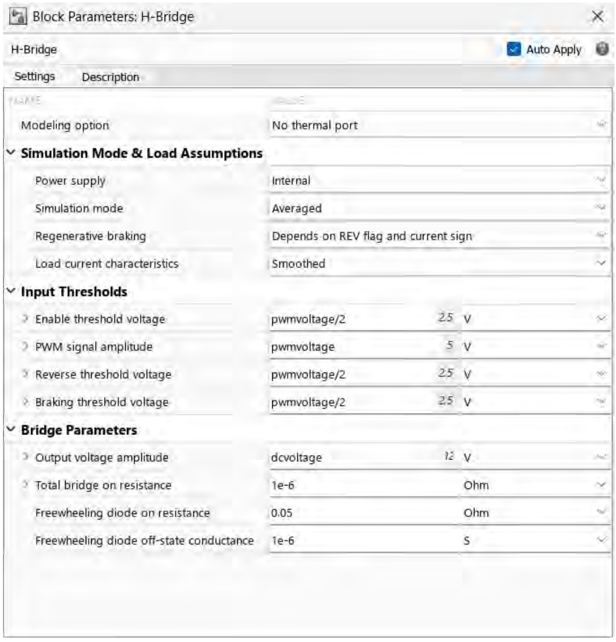
- inductance = 12e-10
- stall torque = 0.43
- noloadspeed = 13180
- dc voltage = 12
- no load current= 1.8
- no load voltage = 12
- rotorInertia= 200
- pwm voltage = 5
- dutyInitial = 0
- dutyFinal = 0.42

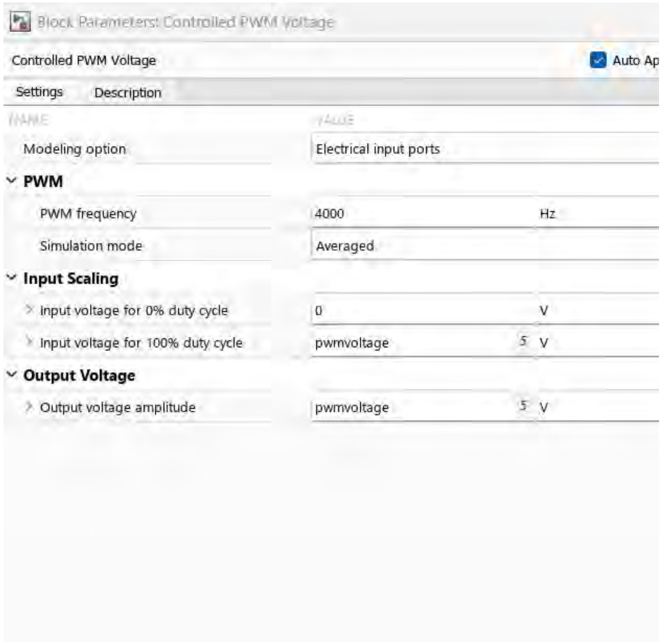
These variables were declared and linked through the MATLAB Editor, allowing for simulation adaptability with varying environmental or mechanical conditions.

These are the parameters defined in the MATLAB Editor section, connected with the Simulink file. Here in **Fig. 4.5.**, the component variable values and electrical component values are mentioned accordingly.



(a) (b)





(c) (d)

Fig. 4.5. Variable values for (a) Wheel, (b)Vehicle Body, (c) DC motor, (d) PWM voltage

Simulation Output and Stability Interpretation:

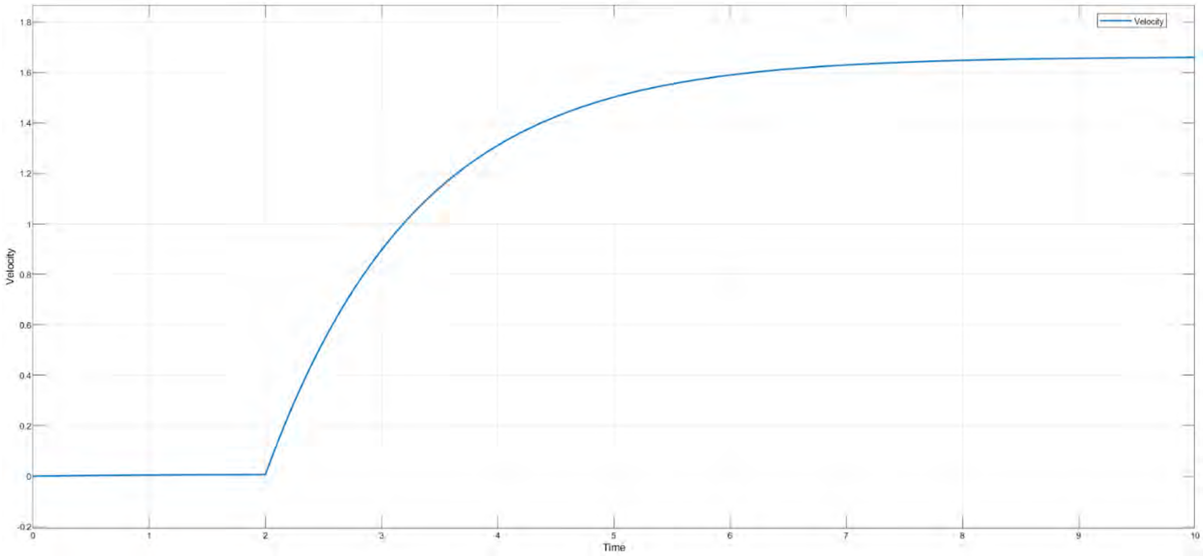


Fig. 4.6. Output Curve after simulating the rover circuit

The system's step response output is shown in **Fig.4.6**. The output signal tracks the input, with no significant overshot or distortion. This indicates a high degree of stability in the rover's design. While some smoothing is observed, the output remains nearly identical to the input. It validates that the electromechanical configuration is both feasible and stable. Moreover, the maximum velocity in this configuration can be observed close to 1.65 m/s which is close to the specification of 0.5m/s to 1.5m/s-controlled velocity. Thus, the simulation confirms the rover's suitability for real-world implementation without compromising on control precision or operational safety.

C. Navigation

Traversing in a sophisticated way in an urban environment is crucial for the rover as well as the pedestrians. Keeping that in mind, the rover is equipped with an efficient navigation system featuring eight ultrasonic sensors: two each on the left, right, front and back sides. The purpose of the sensor is to detect objects within their radius, providing real time data to identify them as well as come up with an avoiding mechanism. The figure below shows the movement pattern as well as the avoidance mechanism.

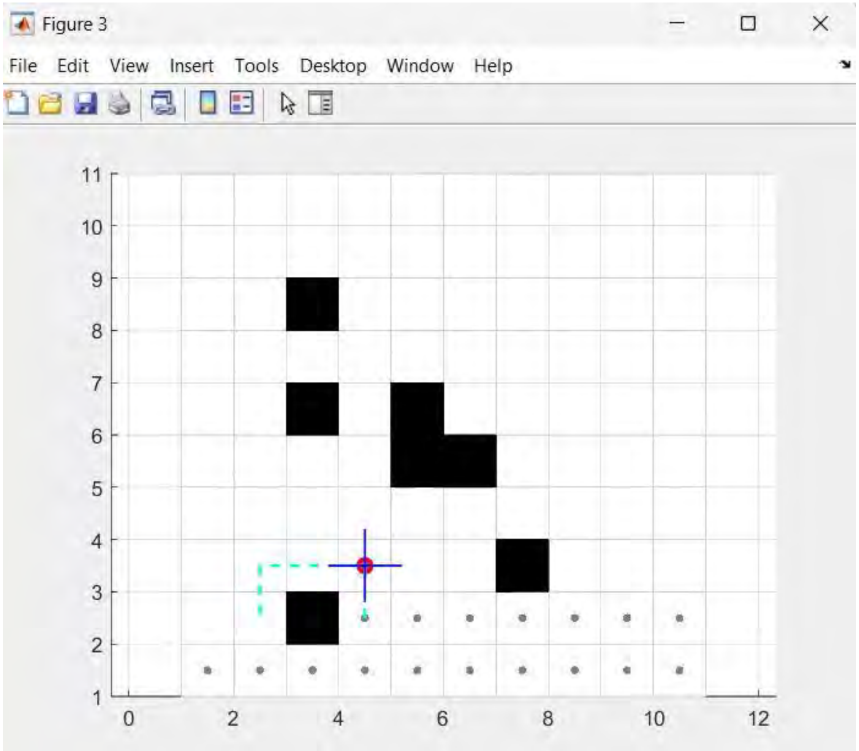


Fig. 4.7. Movement Pattern with avoidance mechanism

For understanding how the rover should traverse in an urban environment, the above simulation was done. The image depicts the rover's navigation path in a bounded region of 10x10 grid, where the black boxes represent the obstacles that are usually seen in a field. The obstacles are randomly generated showcasing real-world scenarios. The simulation was done in such a way that it will use a zig-zag loop to move the rover from left to right and right to left across rows, adjusting direction every row. When an obstacle is detected, the A* algorithm computes the shortest path (using Manhattan distance in a 4-directional movement) to the next safe cell.

The A* algorithm was used as it selects the lowest total estimated cost from the start node to the goal through a given node. The cost is defined as:

$$f(n) = g(n) + h(n) \quad 4.2.1.1$$

Where;

- ➔ **f(n):** The total estimated cost from the start node to the goal through node n
- ➔ **g(n):** The actual cost from the start node to node n, incremented by 1 for each step in a 4-connected grip
- ➔ **h(n):** The heuristic estimate of the cost from node n to the goal, calculated using the Manhattan distance.

Since we assumed the urban environment as a grid and for a grid system, the Manhattan distance was used. Assuming only 4-directional movements, the equation is:

$$h(n) = |x_n - x_{goal}| + |y_n - y_{goal}| \qquad 4.2.1.2$$

Were,

- ➔ $(x_n, y_n :)$, Coordinates of the current node n.
- ➔ (x_{goal}, y_{goal}): Coordinates of the goal node, set to (10, 10) in the rover's navigation grid

The A* algorithm along with the help of ultrasonic sensors ensures that the rover can be utilized in busy terrain like public fields all while ensuring that the waste is being segregated with minimal error. The Zig-Zag pattern ensures that the rover can traverse the entire field in a fixed manner ensuring computational efficiency.

D. Electrical Power System (EPS)

The Electrical Power System (EPS) is a critical component when designing any system. Through proper EPS, it ensures that all the operations are done seamlessly by providing a reliable energy framework that can support various subsystems such as camera, robotic arm and bin. It is designed in such a way so that it can properly manage power distribution all while optimizing energy consumption so that it can efficiently sustain prolonged operations. With the purpose of designing EPS for our project, we modeled and analyzed the data using MATLAB where we monitored energy usage and power consumption over a specific period. This approach enabled us to evaluate the rover's energy dynamics all while ensuring that the system aligns with the specification.

The MATLAB script was constructed to take user input for subsystem wise voltage and current rating along with the rover's speed, system mass and the length and width of a bounded field. The EPS simulation incorporates mechanical and electrical formulas to model power and energy dynamics.

The mechanical side calculates forces and power contribution as follows:

- ➔ **Frictional force:** $F_{rr} = m_{mass} * gravitational\ force * frictional\ coefficient$
- ➔ **Frictional power:** $P_{rr} = F_{rr} * velocity$

- **Aerodynamic power:** $P_{aero} = 0.5 * velocity^3 * air\ density \cdot drag\ coefficient$
- **Mechanical Power:** $P_{mech} = P_{aero} + P_{rr}$
- **Electrical power from mechanical:** $P_{mech-elec} = P_{mech}/0.8$
- **Electrical Energy from mechanical:** $Energy_{mech-elec} = P_{mech-elec} * 0.5$

The electrical side computes power and energy storage:

- **Power:** $Power = voltage * current$
- **Charge:** $Charge = Current * Time$
- **Battery energy:** $Battery\ energy = (Charge/Depth\ of\ Discharge)$
- **Total energy:** $V_{Battery} = Total\ energy/C_{req}$

Through the usage of these equations the subsystem wise parameters were calculated and were physically inputted in the MATLAB script for different approaches to visualize the EPS specification over a specific period.

4.2.1a Pre-Sorting

```
Command Window
Design Approaches:
1. Pre-sorting (4 subsystems)
2. Post-sorting (5 subsystems)
Select design approach (1 or 2): 1

Enter voltage (V) and current (A) for each of the 4 subsystems:
Subsystem 1 Voltage (V): 5
Subsystem 1 Current (A): 5
Subsystem 2 Voltage (V): 8
Subsystem 2 Current (A): 5
Subsystem 3 Voltage (V): 14
Subsystem 3 Current (A): 3.404
Subsystem 4 Voltage (V): 12
Subsystem 4 Current (A): 3.882

Total Subsystem Power: 159.24 W

Enter rover speed (m/s): 1
Enter total rover mass (kg): 40
Enter field length (m): 30
Enter field width (m): 20

--- FIELD SIMULATION REPORT (Pre-sorting) ---
Field Size          : 30.00 x 20.00 m
Total Distance      : 1200.00 m
Time to Cover Field : 20.00 min
Final Energy Used   : 134.25 Wh
Total System Power  : 309.24 W (Drive + Subsystems)
✓ Rover finished field early and idled for the rest of the time.
fx Final Position After 30 min: X = 0.00 m, Y = 19.50 m
```

Fig. 4.8. Parameters set by the user

Based on the subsystem wise voltage and current rating, it was inputted in the MATLAB script to calculate the total system power. After that the rover speed as well as the system mass and finally the fields length-width were inputted, to bound the region for a proper power consumption and energy usage over time all while moving in a zigzag pattern. The output graphs are given below in **Fig. 4.9.**:

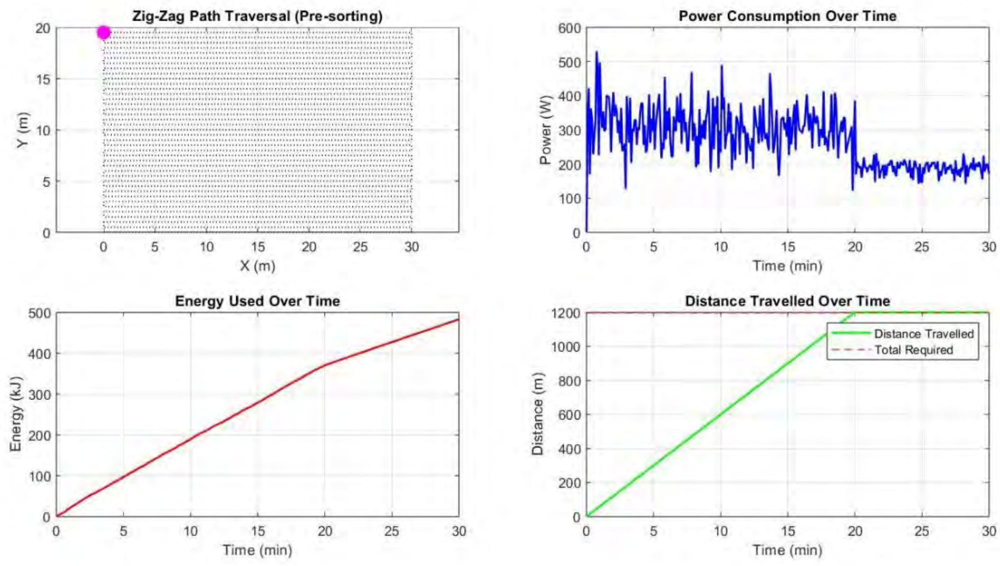


Fig. 4.9. Rover's performance matrix (Pre-Sorting)

A replication of real-world scenario showing the rover's performance matric over a bounded region is shown through the graph **Fig. 4.9**. shows the rovers performance matrix over a bounded region replicating a real-world scenario. The top left subplot shows the zigzag path that the rover will take. The top right subplot displays the power consumption (in watts) over time, fluctuating around 300-500W during movement and stabilizing near 100W during idle. The bottom left subplot tracks energy usage (in kJ), rising to 134.25Wh by 30 minutes, reflecting a linear increase. The bottom right subplot represents the total distance travelled within the allocated time. The simulation validates the rover's efficiency, achieving 100% coverage within the allocated time (20min).

The calculated power and energy metrics ensure the rover's 11.1V, 1000mAh battery can sustain all the operations.

- **Theoretical Calculation**

The rover will have two aspects to its battery/power calculations: electrical and mechanical. Once both the aspects have been calculated, then the battery calculations will be shown. Their respective calculations and formulas used are as follows:

Mechanical Aspect:
Constants:

$$Mass = 40kg$$

$$Velocity, v = 0.5 \sim 1.5\, m/s$$

$$Frictional\ Coefficient = 0.05$$

$$Gravitational\ Force = 9.81\, m/s^2$$

$$Air\ Drag\ Coefficient = 0.9$$

$$Air\ Density, \rho\ =\ 1.2\ kg/m^3$$

$$Frontal\ Area, A\ =\ 0.2m^2$$

For Velocity = 0.5 m/s

$Rolling\ Resistance\ Force, F_{rr} =\ mass\ *gravitational\ force * frictional\ coefficient\ =$ 19.62 $Rolling\ Resistance\ Power, P_{rr} =\ F_{rr}\ *\ velocity\ = 9.81\ W$

$Aerodynamic\ Power, P_{aero} = 0.5\ * velocity^3 * air\ density\ * air\ drag\ coefficient * area =$ $0.0135\ W$ $P_{mech} = P_{aero} + P_{rr}\ =\ 9.8235\ W$

$Electrical\ Power\ from\ mechanical, P_{mech-elec} = P_{mech}/0.8 =$ $12.279\ W\ (\ 80\%\ Drivetrain\ efficiency)$

$E_{mech-elec} = P_{mech-elec} * 0.5h =\ 6.140\ Wh$

For Velocity = 1 m/s

$$Rolling\ Resistance\ Power, P_{rr} =\ F_{rr}\ *\ velocity\ = 19.62\ W$$

$Aerodynamic\ Power, P_{aero} = 0\ .5\ *\ velocity^3 * air\ density * air\ drag\ coefficient * area =$ $0.108\ W$ $P_{mech} = P_{aero} + P_{rr}\ =\ 19.728\ W$

$Electrical\ Power\ from\ mechanical, P_{mech-elec} = P_{mech}/0.8 =\ 24.66\ W$

$E_{mech-elec} = P_{mech-elec} * 0.5h =\ 12.33\ Wh$

For Velocity = 1.5 m/s

$$Rolling\ Resistance\ Power, P_{rr} =\ F_{rr}\ *\ velocity\ = 29.4\ W$$

$Aerodynamic\ Power, P_{aero} = 0.5\ * velocity^3 * air\ density\ * air\ drag\ coefficient * area =$ $0.3645\ W$ $P_{mech} = P_{aero} + P_{rr}\ =\ 29.7645\ W$

$Electrical\ Power\ from\ mechanical, P_{mech-elec} = P_{mech}/0.8 =\ 37.205\ W$

$E_{mech-elec} = P_{mech-elec} * 0.5h =\ 18.6025\ Wh$

Formula:

Electrical Aspect:

$$Power\ =\ voltage\ *\ current$$

$$Charge\ =\ Current\ * Time$$

Bin System:

$$Ultrasonic\ Sensors\ (x2)\ =\ (5\ * 0.002) * 2 = 0.02\ W$$

$$17HS4401S\ =\ (14 * 3.4)\ = 47.6\ W$$

Robotic Arm:

$$SG90\ =\ (7 * 0.25) =\ 1.75\ W$$

$$MG990R\ (x2)\ =\ (6 * 2.5) * 2 = 30\ W$$

$$Motor\ Driver\ (x3)\ =\ (8 * 0.75) * 3 =\ 18\ W$$

Camera:

$$Raspberry\ Pi\ 5 = (5 * 5) = 25\ W$$

Rover:

$$Ultrasonic\ Sensors\ (x8) = (5 * 0.002) * 8 = 0.08\ W$$

$$YDLiDar\ x4 = (5.2 * 0.48) = 2.496\ W$$

$$Motor\ Driver\ (x2) = (8 * 1) * 2 = 16\ W$$

$$DC\ Motor\ (x2) = (12 * 1.2) * 2 = 28.8\ W$$

Total Power Calculation:

$$Total\ Power = 47.62 + 49.75 + 47.376 + 25 = 169.746\ W$$

$$Partial\ Power\ (Camera + Rover) = 47.376 + 25 = 72.376\ W$$

$$Total\ Current, I_{total} = 3.404 + 5 + 5 + 3.882 = 17.286\ A$$

$$Partial\ Current\ (Camera + Rover), I_{partial} = 5 + 3.882 = 8.882\ A$$

Assuming that the rover will use all its subsystems for 10 minutes and (camera and rover) for 20 minutes

$$Total\ Electrical\ Energy = (72.376 * 1/3) + (169.746 * 1/6) = 52.411\ Wh$$

$$Total\ Charge = (8.882 * 1/3) + (17.286 * 1/6) = 5.931\ Ah$$

Battery Calculations:

Formulas:

$C_{req} = Total\ Charge/\ Depth\ of\ Discharge$ (Depth of Discharge = 0.8)

$$Total\ rover\ energy = electrical\ energy + mechanical\ energy$$

$$V_{battery} = Total\ energy/\ C_{req}$$

$$C_{req} = 5.931/0.8 = 7.413\ Ah$$

For Velocity = 0.5 m/s

$$Total\ rover\ energy = 52.411 + 6.140 = 58.551\ Wh$$

$$V_{battery} = 58.551/7.413 = 7.898\ V$$

For Velocity = 1 m/s

$$Total\ rover\ energy\ =\ 52.411\ +12.33\ =64.741\ Wh$$

$$V_{battery}\ =\ 64.741/7.413\ =8.733\ V$$

For Velocity = 1.5 m/s

$$Total\ rover\ energy\ =\ 52.411\ +18.6025\ =71.0135\ Wh$$

$$V_{battery}\ =\ 71.0135/7.413\ =9.580\ V$$

- **Power Budget**

TABLE 3.1: POWER BUDGET FOR PRE-SORTING APPROACH

Components	Voltage (V)	Current (A)	Power (W)	Operating Voltage (V)	Operating Current (A)	Place of Use
Raspberry Pi 5	5	5	25	5	5	Rover Body and Camera
Motor Drivers (x2)	8 each	1 each	16	8	2	Caterpillar wheels
Motor Drivers (x3)	8 each	0.75 each	18	8	2.25	Robotic Arm
Ultrasonic Sensors (x10)	5 each	0.002 each	0.1	5	0.02	Bin, Rover body
MG990R (x2)	6 each	2.5 each	30	6	5	Robotic Arm
DC Motors (x2)	12 each	1.2 each	28.8	12	2.4	Rover Body
SG90	7	0.25	1.75	7	0.25	Robotic Arm
17HS4401S	14	3.4	47.6	14	3.4	Bin
YDLiDar x4	5.2	0.48	2.496	5.2	0.48	Rover Body

Total Power cost: **169.746 W**

4.2.1b Post Sorting

```
Command Window
2. Post-sorting (5 subsystems)
Select design approach (1 or 2): 2

Enter voltage (V) and current (A) for each of the 5 subsystems:
Subsystem 1 Voltage (V): 14
Subsystem 1 Current (A): 3.404
Subsystem 2 Voltage (V): 5
Subsystem 2 Current (A): 5
Subsystem 3 Voltage (V): 8
Subsystem 3 Current (A): 5
Subsystem 4 Voltage (V): 12
Subsystem 4 Current (A): 7.926
Subsystem 5 Voltage (V): 12
Subsystem 5 Current (A): 13

Total Subsystem Power: 363.77 W

Enter rover speed (m/s): 1
Enter total rover mass (kg): 40
Enter field length (m): 30
Enter field width (m): 20

--- FIELD SIMULATION REPORT (Post-sorting) ---
Field Size          : 30.00 x 20.00 m
Total Distance      : 1200.00 m
Time to Cover Field : 20.00 min
Final Energy Used   : 233.86 Wh
Total System Power  : 513.77 W (Drive + Subsystems)
✅ Rover finished field early and idled for the rest of the time.
fx Final Position After 30 min: X = 0.00 m, Y = 19.50 m
```

Fig. 4.10. Parameters set by the users

Based on the subsystem wise voltage and current rating, it was inputted in the MATLAB script to calculate the total system power. After that the rover speed as well as the system mass and finally the fields length-width were inputted, to bound the region for a proper power consumption and energy usage over time all while moving in a zigzag pattern. The output graphs are given below:

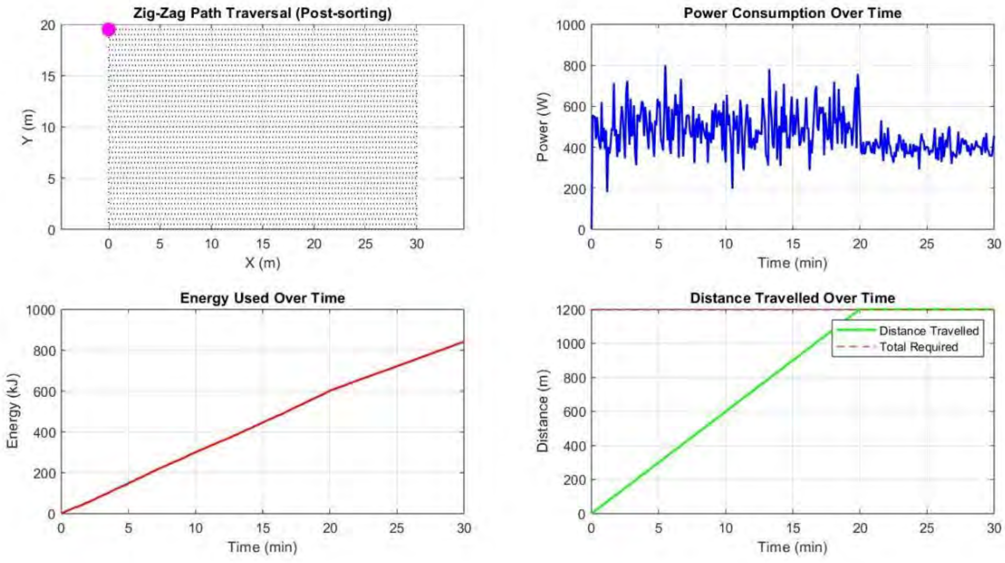


Fig. 4.11. Rover's performance matrix (Post Sorting)

A replication of the real world showcasing the performance of rover is shown through the graph in **Fig.4.11**. The top left subplot shows the zigzag path that the rover will take. The top right subplot displays the power consumption (in watts) over time, fluctuating around 200-800W during movement. The bottom left subplot tracks energy usage (in kJ), rising to 900kJ, reflecting a linear increase. The bottom right subplot represents the total distance travelled within the allocated time. The simulation validates the rover's efficiency, achieving 100% coverage within the allocated time (20min).

The calculated power and energy metrics ensure the rover's 11.1V, 1000mAh battery can sustain all the operations.

- **Theoretical Calculation**

The rover will have two aspects to its battery/power calculations: electrical and mechanical. Once both the aspects have been calculated, then the battery calculations will be shown. Their respective calculations and formulas used are as follows:

Mechanical Aspect:
Constants:

$$Mass = 45\ kg$$

$$Velocity, v = 0.5 \sim 1.5\ m/s$$

$$Frictional\ Coefficient = 0.05$$

$$Gravitational\ Force = 9.81\ m/s^2$$

$$Air\ Drag\ Coefficient = 0.9$$

$$Air\ Density, \rho = 1.2\ kg/m^3$$

$$Frontal\ Area, A = 0.2m^2$$

For Velocity = 0.5 m/s

$Rolling\ Resistance\ Force, F_{rr} = mass * gravitational\ force * frictional\ coefficient =$
$22.07N$ $Rolling\ Resistance\ Power, P_{rr} = F_{rr} * velocity = 11.03625\ W$

$Aerodynamic\ Power, P_{aero} = 0.5 * velocity^3 * air\ density * air\ drag\ coefficient * area$
$= 0.0135\ W$ $P_{mech} = P_{aero} + P_{rr} = 11.04975\ W$

$Electrical\ Power\ from\ mechanical, P_{mech-elec} = P_{mech}/0.8 =$
$13.812\ W$ (80% $Drivetrain\ efficiency$)

$E_{mech-elec} = P_{mech-elec} * 0.5h = 6.906\ Wh$

For Velocity = 1 m/s

$$Rolling\ Resistance\ Power, P_{rr} = F_{rr} * velocity = 22.0725\ W$$

$Aerodynamic\ Power, P_{aero} = 0.5 * velocity^3 * air\ density * air\ drag\ coefficient * area =$
$0.108\ W$ $P_{mech} = P_{aero} + P_{rr} = 22.1805\ W$

$Electrical\ Power\ from\ mechanical, P_{mech-elec} = P_{mech}/0.8 = 27.726\ W$

$E_{mech-elec} = P_{mech-elec} * 0.5h = 13.863\ Wh$

For Velocity = 1.5 m/s

$$Rolling\ Resistance\ Power, P_{rr} = F_{rr} * velocity = 33.109\ W$$

$Aerodynamic\ Power, P_{aero} = .5 * velocity^3 * air\ density * air\ drag\ coefficient * area =$
$0.3645\ W$ $P_{mech} = P_{aero} + P_{rr} = 33.47325\ W$
$Electrical\ Power\ from\ mechanical, P_{mech-elec} = P_{mech}/0.8 = 41.842\ W$
$E_{mech-elec} = P_{mech-elec} * 0.5h = 20.921\ Wh$

Formulas:

Electrical Aspect:

$$Power = voltage * current$$

$$Charge = Current * Time$$

Bin System:

$$Ultrasonic\ Sensors\ (x2) = (5 * 0.002) * 2 = 0.02\ W$$

$$17HS4401S = (14 * 3.4) = 47.6\ W$$

Robotic Arm:

$$SG90 = (7 * 0.25) = 1.75\ W$$

$$MG990R\ (x2) = (6 * 2.5) * 2 = 30\ W$$

$$Motor\ Driver\ (x3) = (8 * 0.75) * 3 = 18\ W$$

Rover:

$$Ultrasonic\ Sensors\ (x8) = (5 * 0.002) * 8 = 0.08\ W$$

$$YDLiDar\ x4 = (5.2 * 0.48) = 2.496\ W$$

$$Motor\ Driver\ (x2) = (8 * 1) * 2 = 16\ W$$

$$DC\ Motor\ (x4) = (12 * 1.2) * 4 = 57.6\ W$$

Camera:

$$Raspberry\ Pi\ 5 = (5 * 5) = 25\ W$$

Conveyor Belt:

$$Fielectric\ DC\ 12V\ 42\ RPM = (12 * 6) = 72\ W$$

$$L640 = (10 * 7) = 70\ W$$

Total Power Calculation:

$$Total\ Power = 47.62 + 49.75 + 76.176 + 25 + 142 = 340.546\ W$$

$$Partial\ Power\ (Camera + Rover) = 76.176 + 25 = 101.176\ W$$

$$Total\ Current, I_{total} = 3.404 + 5 + 5 + 13 + 7.296 = 33.7\ A$$

$$Partial\ Current\ (Camera + Rover), I_{partial} = 5 + 13 = 18\ A$$

Assuming that the rover will use all its subsystems for 10 minutes and (camera and rover) for 20 minutes

$$Total\ Electrical\ Energy = (101.176 * 1/3) + (340.546 * 1/6) = 90.483\ Wh$$

$$Total\ Charge = (18 * 1/3) + (33.7 * 1/6) = 11.62\ Ah$$

Battery Calculations:

Formula:

$C_{req} = Total\ Charge/\ Depth\ of\ Discharge$ (Depth of Discharge = 0.8)

$$Total\ rover\ energy = electrical\ energy + mechanical\ energy$$

$$V_{battery} = Total\ energy/\ C_{req}$$

$$C_{req} = 11.62/0.8 = 14.525\ Ah$$

For Velocity = 0.5 m/s

$$Total\ rover\ energy = 90.483 + 6.906 = 97.389\ Wh$$

$$V_{battery} = 97.389/14.525 = 6.705\ V$$

For Velocity = 1 m/s

$$Total\ rover\ energy\ =\ 90.483\ +13.863\ =104.346\ Wh$$

$$V_{battery}=\ 104.346/14.525\ =7.184V$$

For Velocity = 1.5 m/s

$$Total\ rover\ energy\ =\ 90.483\ +20.921\ =111.404\ Wh$$

$$V_{battery}=\ 111.404/14.525\ =7.669V$$

- **Power Budget**

TABLE 4.2: POWER BUDGET FOR POST-SORTING APPROACH

Components	Voltage (V)	Current (A)	Power (W)	Operating Voltage (V)	Operating Current (A)	Place of Use
Raspberry Pi 5	5	5	25	5	5	Rover Body and Camera
Motor Drivers (x2)	8 each	1 each	16	8	2	Caterpillar wheels
Motor Drivers (x3)	8 each	0.75 each	18	8	2.25	Robotic Arm
Ultrasonic Sensors (x10)	5 each	0.002 each	0.1	5	0.02	Bin, Rover body
MG990R (x2)	6 each	2.5 each	30	6	5	Robotic Arm
DC Motors (x4)	12 each	1.2 each	57.6	12	2.4	Rover Body
SG90	7	0.25	1.75	7	0.25	Robotic Arm
17HS4401S	14	3.4	47.6	14	3.4	Bin
YDLiDar x4	5.2	0.48	2.496	5.2	0.48	Rover Body
Fielectic DC 12V 42 RPM	12	6	72	12	6	Conveyor Belt
L640	10	7	70	10	7	Conveyor Belt

4.2.2 Machine Learning Models

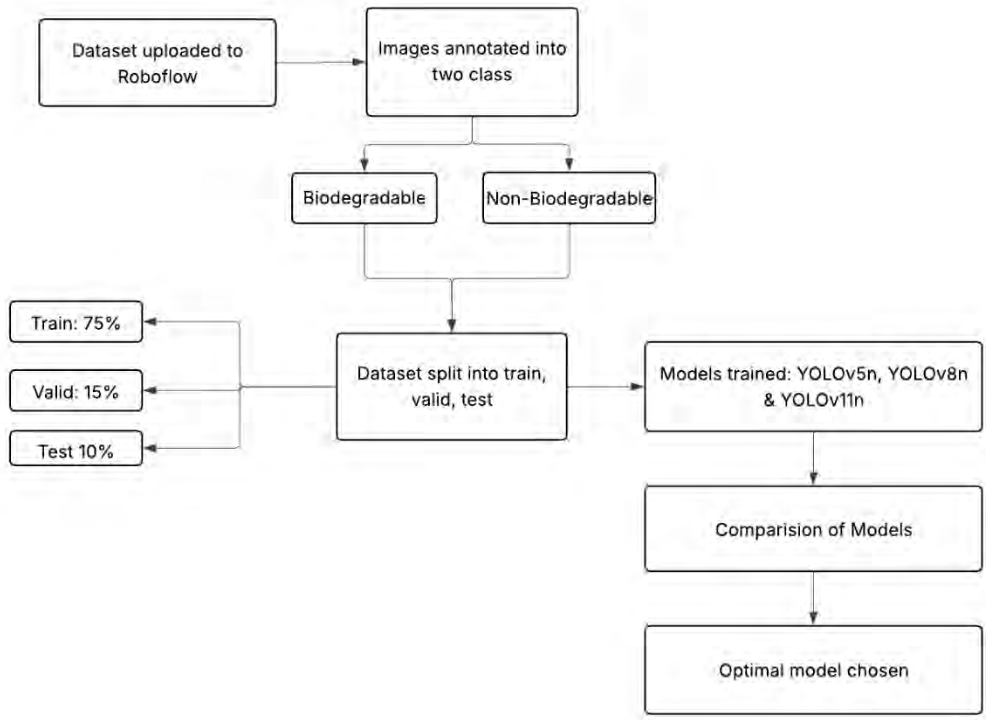


Fig. 4.12. Block diagram of the process of how the optimal model was chosen

4.2.2a Introduction to YOLO Models

YOLO (You Only Look Once) models are cutting-edge object detection systems renowned for their speed and accuracy as they process images in a single pass. This makes them perfect for real-time applications for projects such as our autonomous waste segregation rover which will classify and detect Biodegradable and Non-Biodegradable waste.

We evaluated the nano variants of YOLO models: Yolov5n, YOLOv8n, YOLOv11n as the nano variants are designed for resource constrained devices such as the Raspberry Pi 5. These lightweight models balance computational efficiency and detection performance which is critical for real-time waste segregation. By comparing these versions which each reflect their architectural improvements, we aim to determine the best one for the rover.

4.2.2b Dataset

When training a machine learning model, the first thing that comes to mind is the dataset. The dataset titled, "*BDWaste: A comprehensive image dataset of digestible and indigestible waste in Bangladesh*" was obtained from Mendeley [26]. It contains two main classes of waste: Biodegradable and Non-Biodegradable. To balance the dataset, more biodegradable images were added from a dataset titled, "*GARBAGE CLASSIFICATION 3*

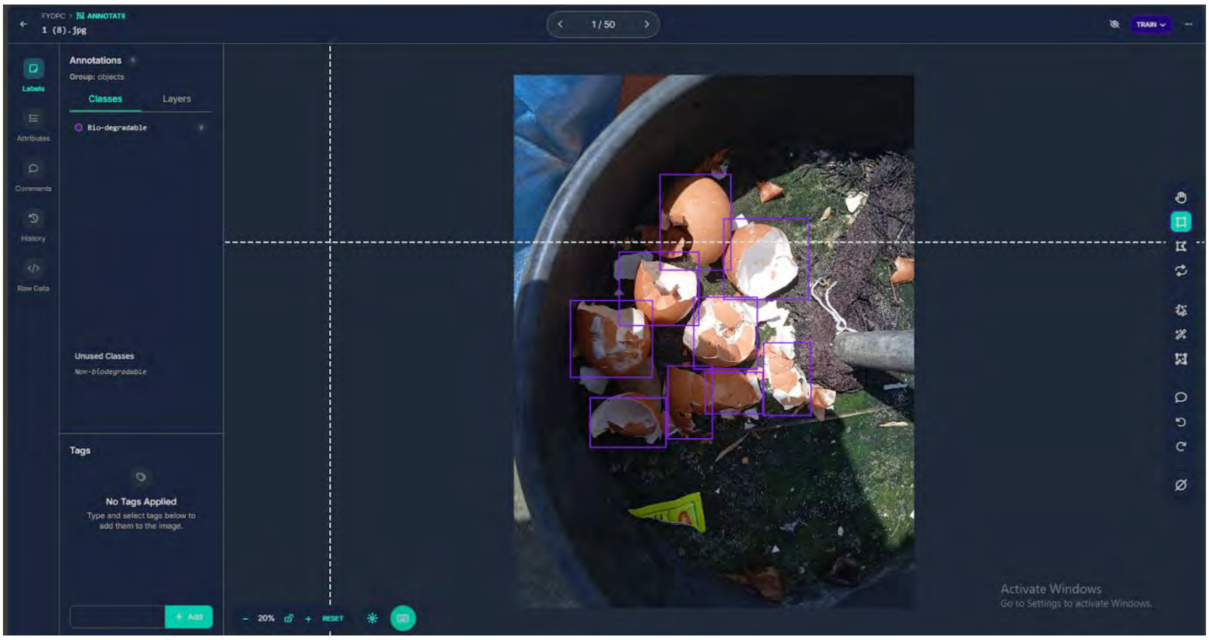
Computer Vision Project" from Roboflow which led to a total image count of 3190 where 1595 images were placed into each class [25]. This ensured that the dataset is balanced. The dataset contains 21 distinct kinds of waste where biodegradable waste contains potato peel, lemon peel, rice, paper, fish scale, malt shell, banana peel, sugarcane husk, and eggshell. Conversely non-biodegradable waste includes polythene, plastic, wire, glass, gloves, chip packets, empty medicine packets, masks, plastic bottles and disposable cups. Table 4.3 and Table 4.4 show the **21** distinct waste types that are in the dataset. Since we trained YOLO models, all the images had to be annotated manually. The images in this dataset were annotated using **Roboflow**, an open-source platform that simplifies dataset management and annotation for computer vision tasks [26]. **Fig.4.13(a)** shows the image annotation software that was used.

TABLE 4.3: TYPES OF NON-BIODEGRADABLE WASTE IMAGES IN THE DATASET

Waste Type: Inorganic	Pictures
Polythene	
Plastic	
Wire	
Glass	
Surgical Gloves	
Chip Packets	
Empty Medicine Packets	
Surgical Masks	
Bottles	

TABLE4.4: TYPES OF BIODEGRADABLE WASTE IMAGES IN THE DATASET

Waste Type: Organic	Pictures
Potato Peel	
Lemon Peel	
Rice	
Paper	
Fish Scale	
Malta Shell	
Banana Peel	
Sugarcane Husk	
Egg Shell	



(a)

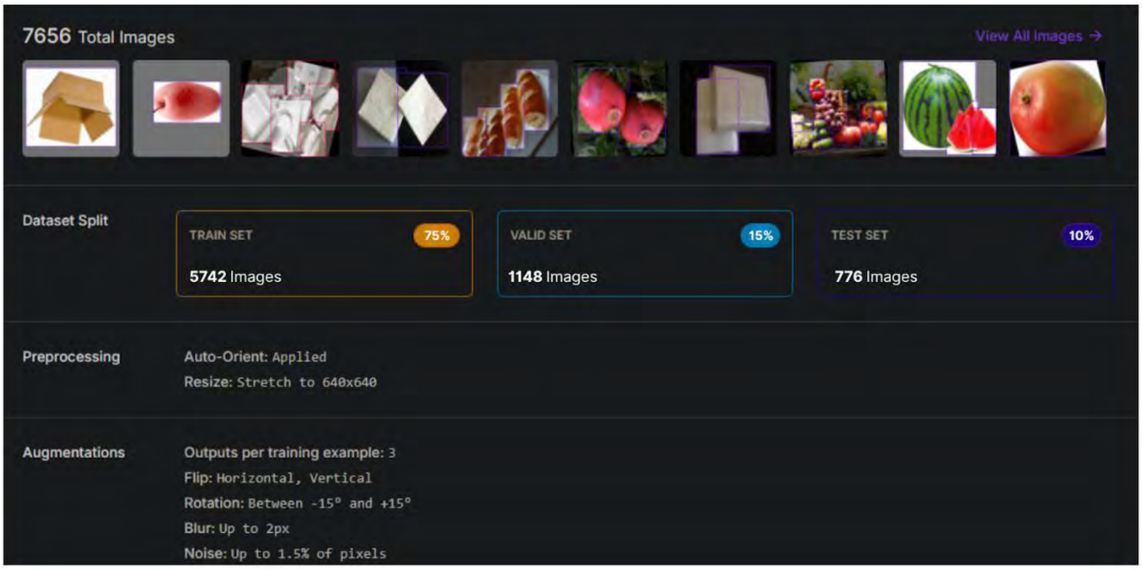


Fig. 4.13. (a) Image Annotation Software (b) Dataset Split

After all the images were annotated, the dataset was split into 75% for training, 15% for validation and 10% for testing. Fig.4.13. (b) shows the dataset split where in total before augmentation there were 3190 images and after data augmentations there was a total of 7656 images where 5742 images were in the training set, 1148 images were in the validation set and 766 images were in the test set.

4.2.2c Model Comparison

The comparison spans over multiple metrics such as training and validation losses, performance metrics and confusion matrix analysis.

A. **Training Losses**

Training losses measure how well a model learns from the training data.

- **Training Box Loss:**

Fig. 4.14. shows a graph plotting Training Box Loss over epochs for YOLOv5n, YOLOv8n, YOLOv11n. The training box loss graph shows how effectively each model learns to predict accurate bounding boxes around waste items over epochs. A steeper decline and lower final loss indicate faster and better learning for localization.

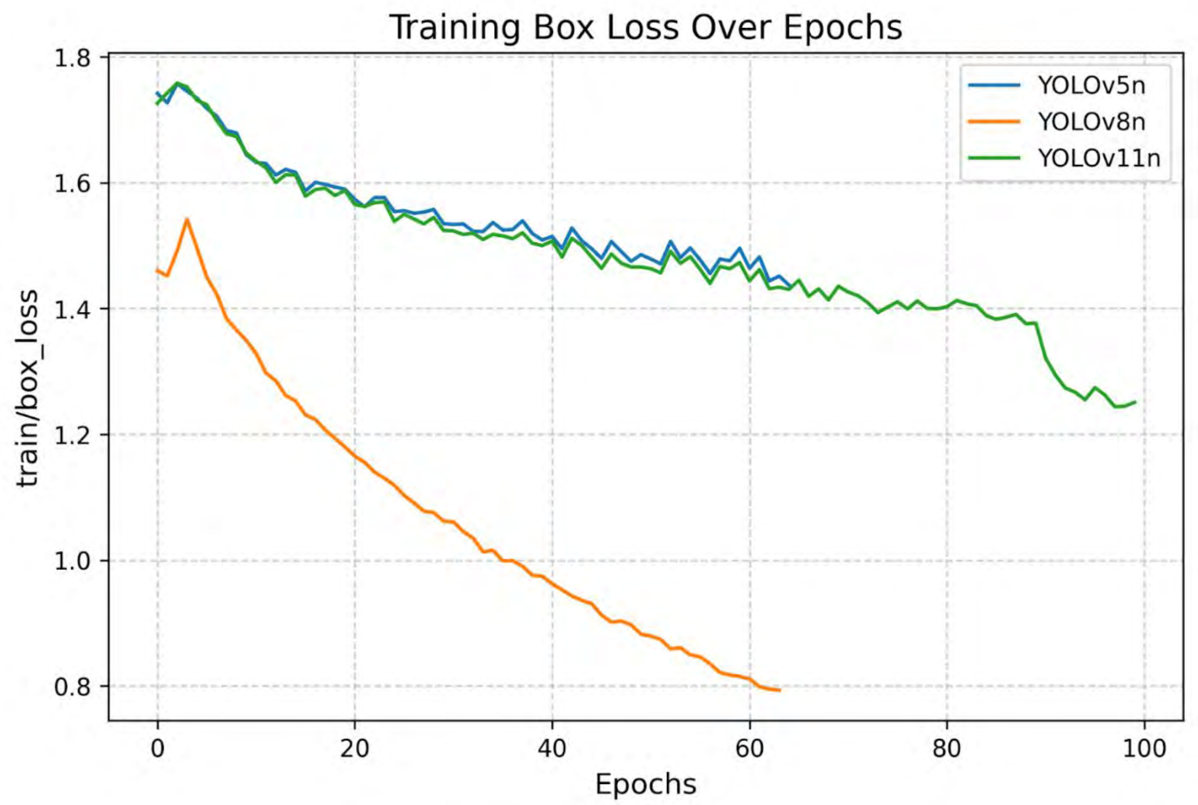


Fig. 4.14. Training Box Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

From Fig. **4.14.**, it can be observed that YOLOv8n shows the steepest decline and lowest final loss, suggesting it learns bounding box prediction most effectively. On the other hand, YOLOv11n declined moderately, stabilizing at a higher loss than YOLOv8n, and YOLOv5n had the slowest decline and highest final loss, indicating less efficient training. Hence YOLOv8n excels in learning to localize waste items accurately.

- **Training Classification Loss Graph**

Fig. 4.15. shows a graph plotting Training Classification Loss over epochs for YOLOv5n, YOLOv8n, YOLOv11n. This graph tracks how well each model learns to classify waste as Biodegradable or Non-Biodegradable during training. Lower loss means better classification learning.

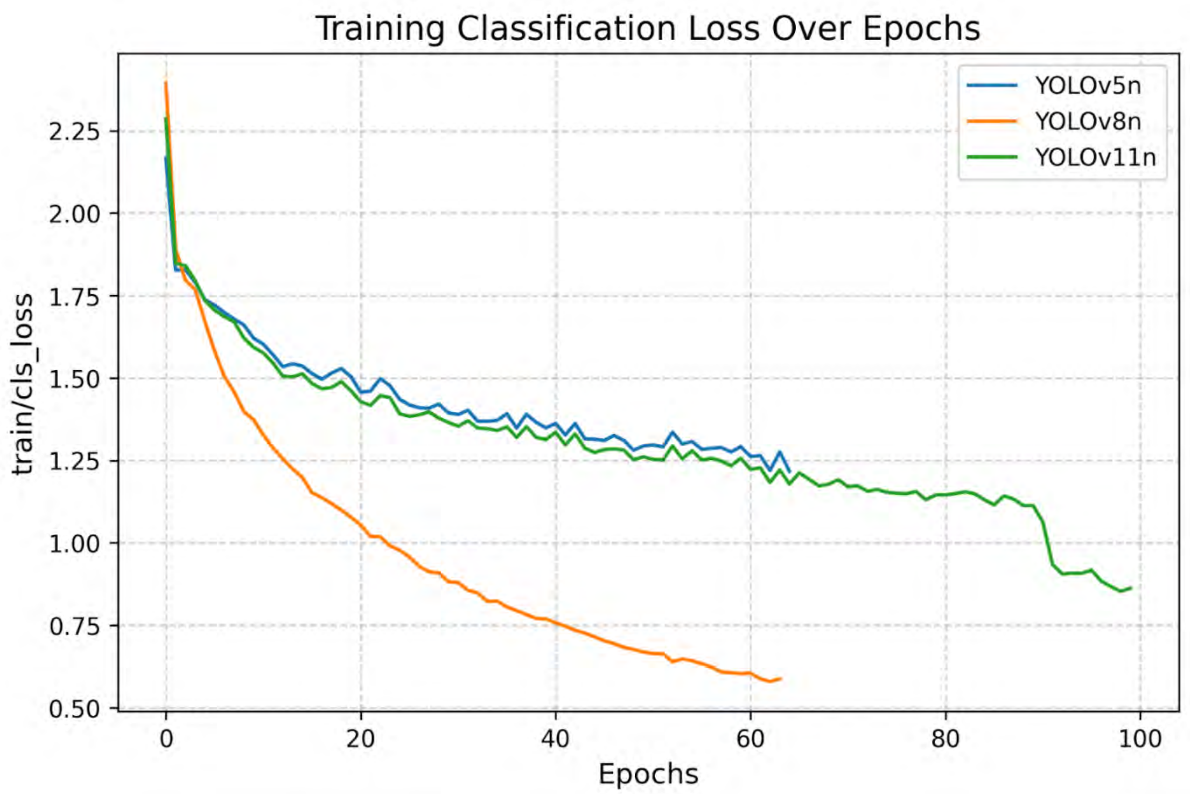


Fig. 4.15. Training Classification Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n

YOLOv8n exhibits the fastest reduction and lowest loss, which implies it is mastering classification quickly as shown through **Fig. 4.15**. On the other hand, YOLOv11n stabilizes at a competitive but higher loss than YOLOv8n, and YOLOv5n shows the highest loss, learning classification more slowly. Hence YOLOv8n is the best at learning waste classification.

- **Training DFL Loss Graph**

Fig. 4.16. shows a graph plotting Training DflLoss over epochs for YOLOv5n, YOLOv8n, YOLOv11n. The training DFL loss graph measures how well each model handles ambiguous detections (e.g., partially hidden waste). Lower loss indicates better performance in challenging scenarios.

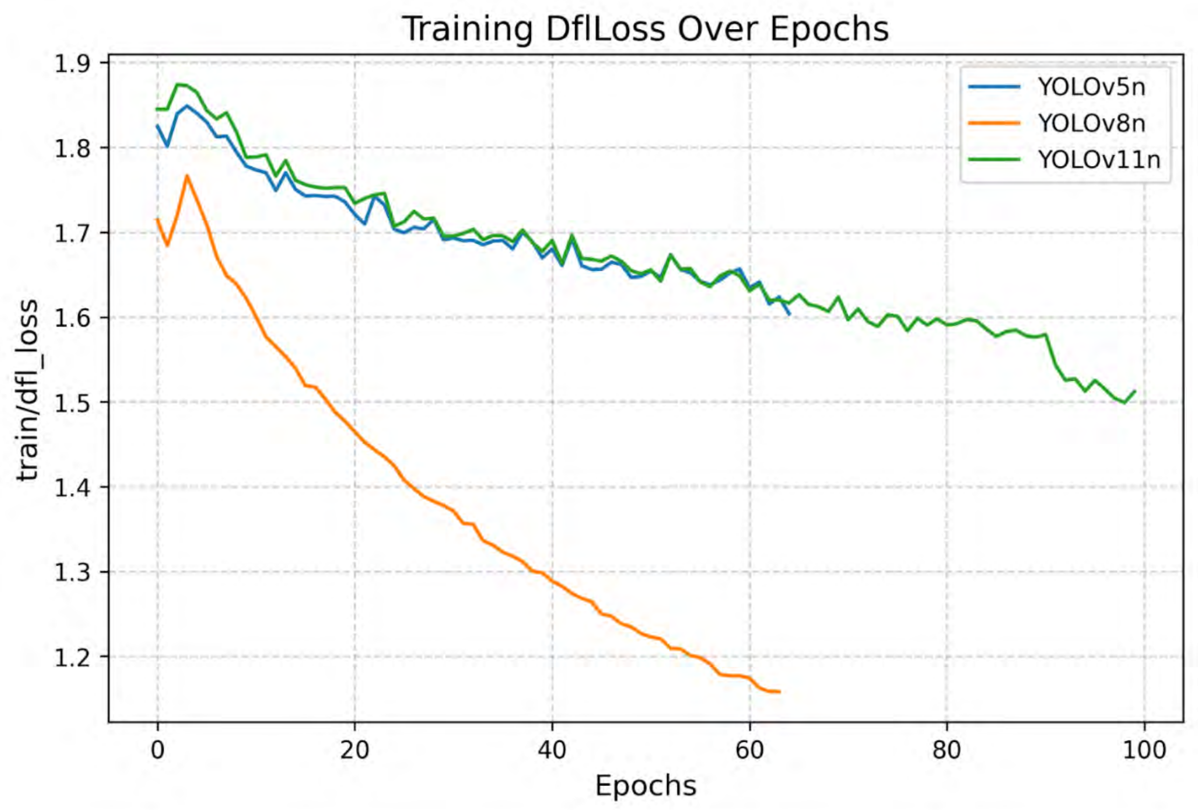


Fig.4.16. Training DflLoss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n**.**

From Fig. 4.16., it can be observed that YOLOv8n reaches the lowest DFL loss, excelling in ambiguous cases. On the other hand, YOLOv11n performs well but stabilizes slightly higher than YOLOv8n, and YOLOv5n has the highest loss, struggling with challenging detections. Hence YOLOv8n performs best in managing ambiguous waste detections.

B. Validation Losses

Validation losses evaluate how well models generalize to unseen data, with the same three types as training losses.

- **Validation Box Loss Graph**

Fig. 4.17. shows a graph plotting Validation Box Loss over epochs for YOLOv5n, YOLOv8n, YOLOv11n. This graph shows how accurately each model predicts bounding boxes on validation data. Lower, stable loss indicates better generalization for localization.

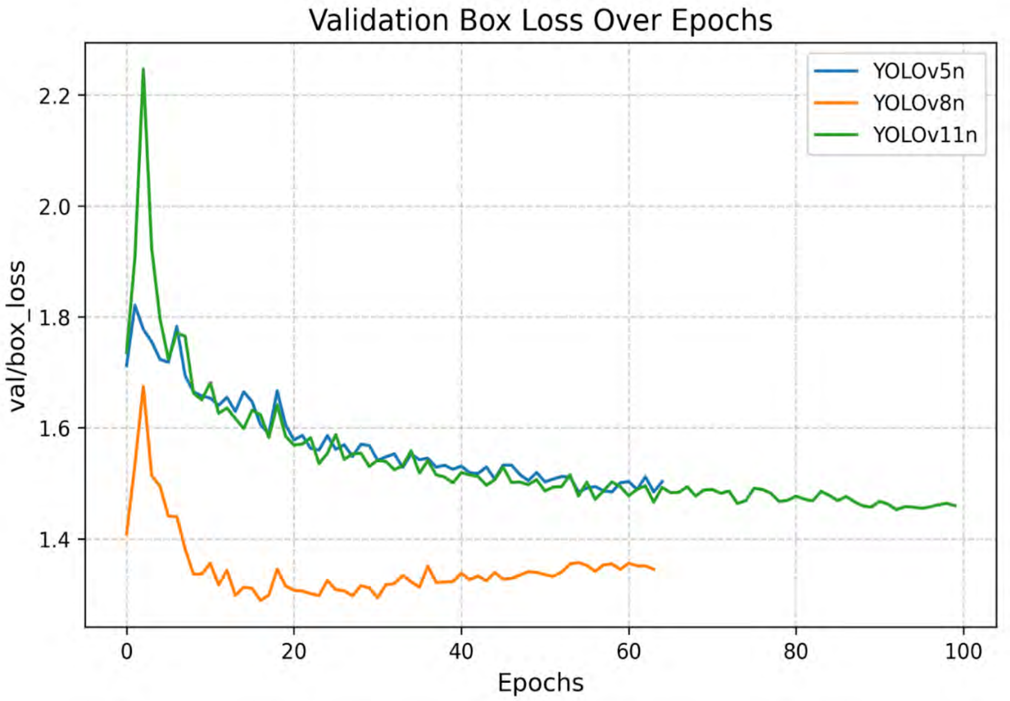


Fig.4.17. Validation Box Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

From **Fig. 4. 17.**, it can be observed that YOLOv8n achieves the lowest validation box loss, generalizing exceptionally well. On the other hand, YOLOv11n shows slightly higher loss, and YOLOv5n has the highest loss, indicating weaker generalization. Hence YOLOv8n generalizes best for bounding box prediction.

- **Validation Classification Loss Graph**

Fig. 4.18. shows a graph plotting Validation Classification Loss over epochs for YOLOv5n, YOLOv8n, YOLOv11n. This graph assesses how well each model classifies waste types on validation data. Lower loss means better generalization for classification.

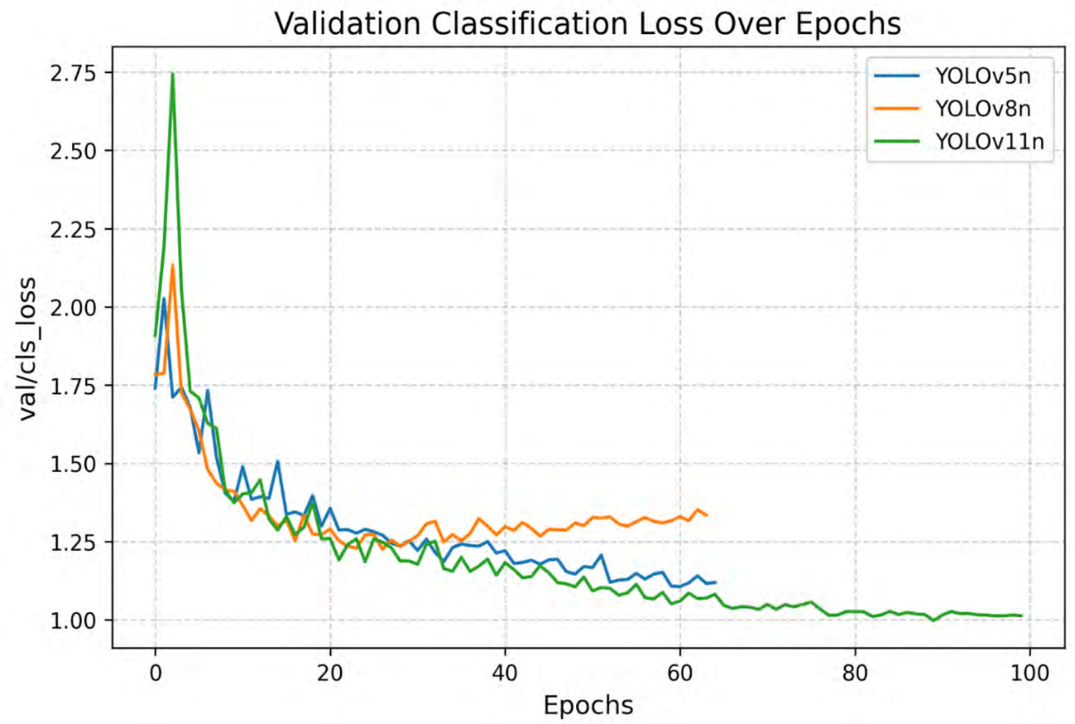


Fig.4.18. Validation Classification Loss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

From **Fig. 4. 18.**, it can be observed that while YOLOv11n has the lowest classification loss, YOLOv8n maintains a competitive and stable loss while outperforming YOLOv5n, making it highly reliable in real-world generalization. Since YOLOv8n shows stronger overall training–validation consistency, it is considered the best at practical classification performance.

- **Validation DFL Loss Graph**

Fig. 4.19. shows a graph plotting Validation DFL Loss over epochs for YOLOv5n, YOLOv8n, YOLOv11n. This graph measures how well each model handles ambiguous detections on validation data. Lower loss suggests better generalization in tough scenarios.

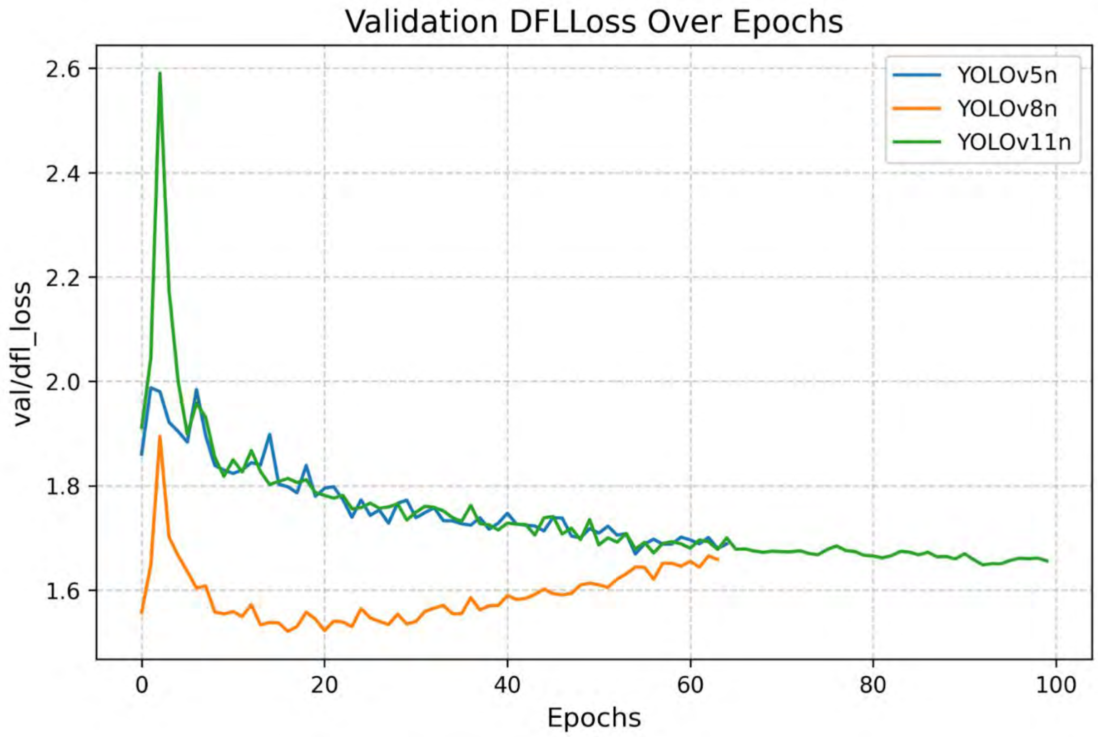


Fig.4.19. Validation DFLLoss Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

From **Fig. 4.19.**, it can be observed that YOLOv8n maintains one of the lowest DFL losses, excelling in ambiguous validation cases. YOLOv11n is close but slightly higher, and YOLOv5n struggles with the highest loss. Hence YOLOv8n performs the best in generalizing to ambiguous detections.

C. Performance Metrics Over Epochs

The following metrics assess the detection performance of the 3 YOLO models.

a. **Precision**: Correct positive detections among all positive predictions.
b. **Recall**: Actual positives correctly detected.
c. **F1-Score**: Balance of precision and recall.
d. **mAP@50**: Overall detection accuracy at IoU=0.5.

- **Precision Over Epochs Graph**

Fig. 4. 20. shows a graph plotting Precision over epochs for all three models where the graph shows how as precision improves, there are fewer false positives (e.g., detecting a rock as a waste). Hence higher values are better.

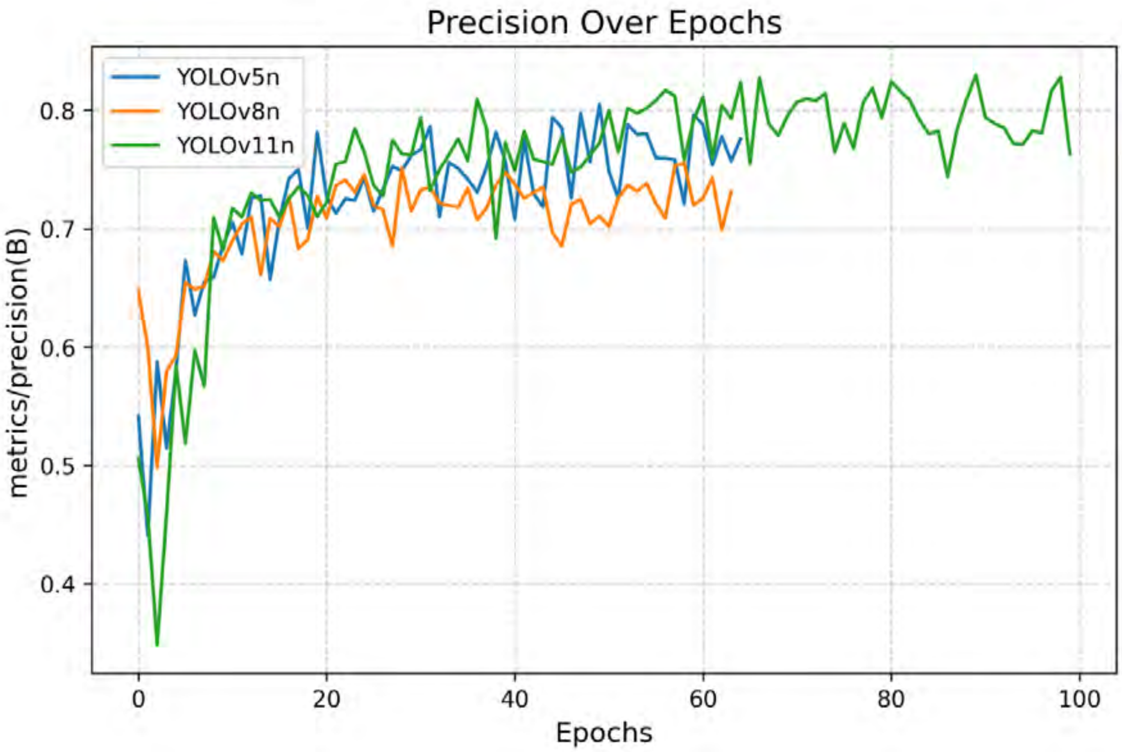


Fig.4.20. Precision Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

From **Fig. 4. 20.**, it is evident that YOLOv5n achieves the highest precision at 0.7759, followed by YOLOv11n at 0.7627, while YOLOv8n records 0.7314. This suggests that YOLOv5n is slightly more selective, producing fewer false positives. However, YOLOv8n still maintains a strong level of precision and, when combined with its lower training losses and stable generalization, it provides a more consistent trade-off between precision and recall compared to the other models.

- **Recall Over Epochs Graph**

Fig. 4.21. shows a graph of Recall over Epochs which tracks recall improvement, showing how well models detect all waste items. Higher values mean fewer missed detections.

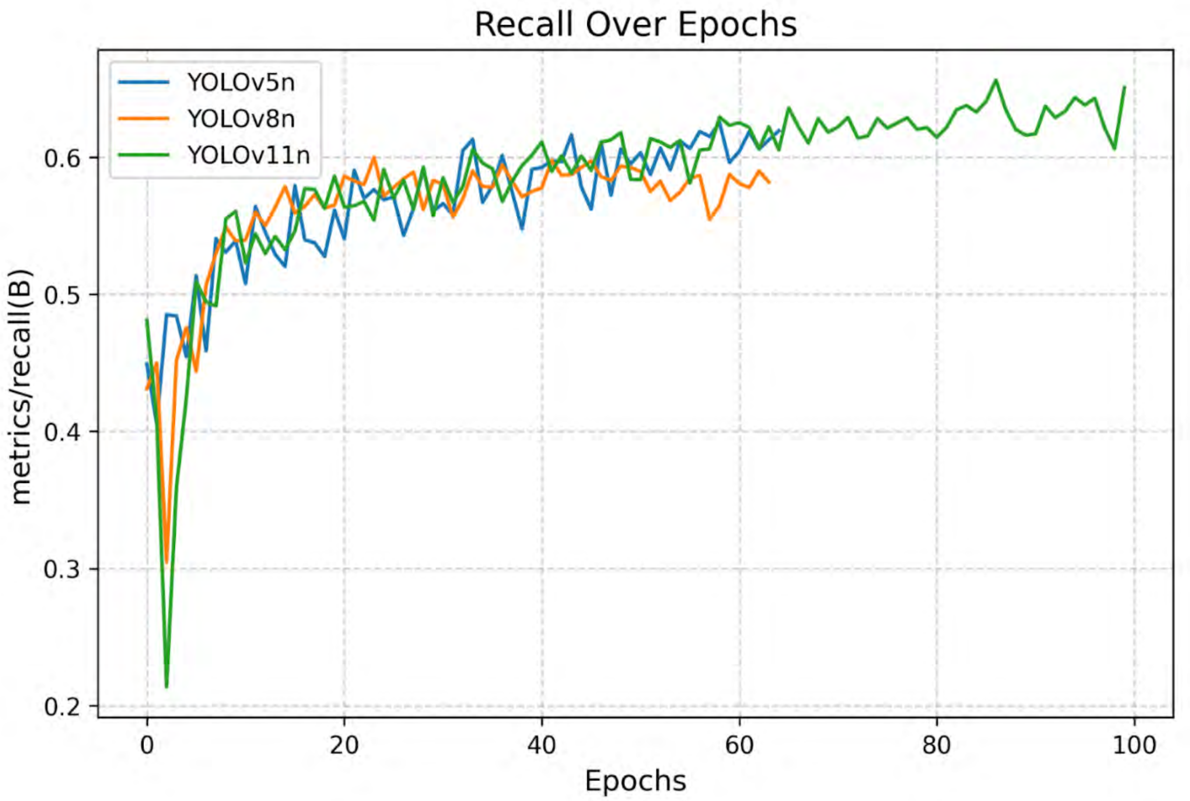


Fig.4.21. Recall Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

From **Fig. 4.21.**, it can be observed that YOLOv11n leads at 0.6508, YOLOv5n follows at 0.6193, and YOLOv8n records 0.5820. This indicates that YOLOv11n detects slightly more true objects overall. However, recall alone does not determine model effectiveness. Despite having a modest recall, YOLOv8n compensates through stronger performance in other areas such as precision stability, lower training losses, and higher mAP@50–95, which makes it more dependable for real-world applications.

- **F1-Score Over Epochs Graph**

Fig. 4. 22. shows a graph of F1-Score over Epochs graph where the graph shows how precision and recall are being balanced. Higher F1-scores indicate better overall performance.

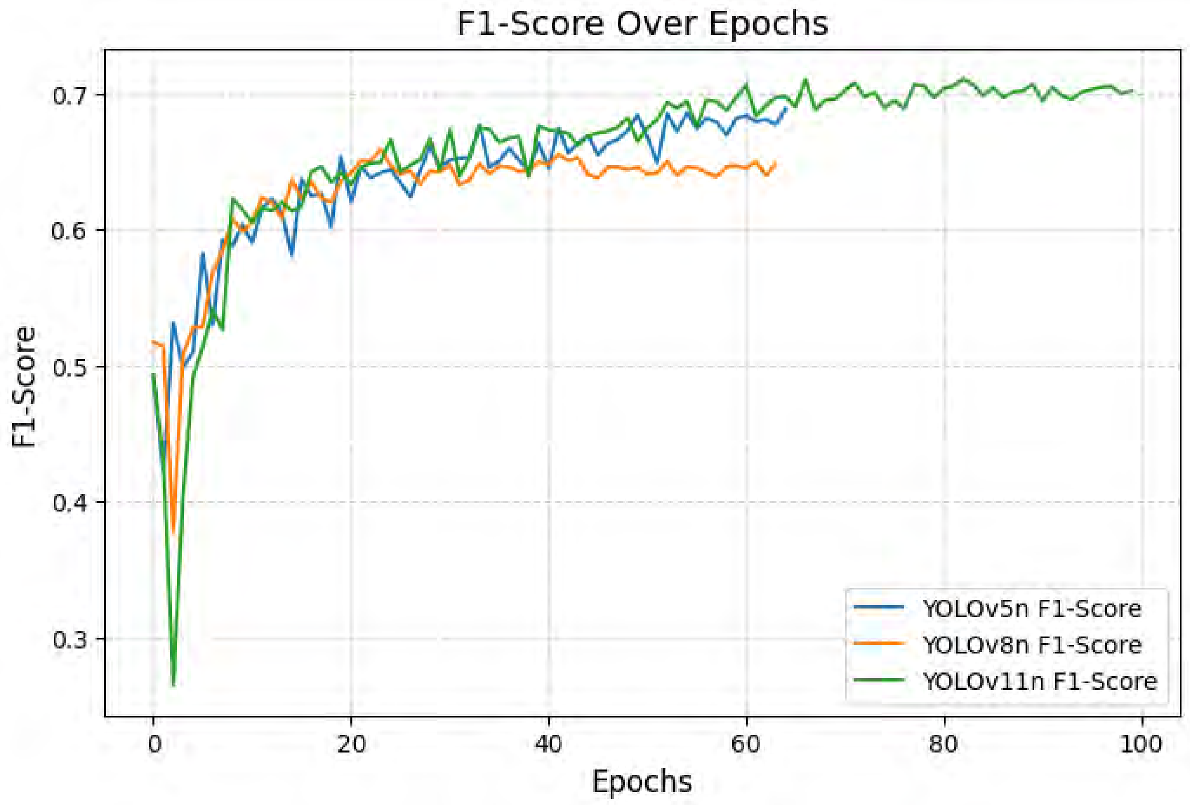


Fig.4.0.22. F1 Score Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

Fig. 4.22. plots F1-score, balancing precision and recall. YOLOv11n leads with 0.7023, YOLOv5n records 0.6888, while YOLOv8n scores 0.6482. Importantly, YOLOv8n shows consistency across epochs without sharp fluctuations, making it more dependable in practice despite slightly lower final values.

- **mAP@50 Over Epochs Graph**

Fig. 4.23. shows a graph plotting mAP@50 over epochs for all three models. This graph measures overall detection accuracy throughout their training process where a higher mAP@50 indicates better performance across thresholds.

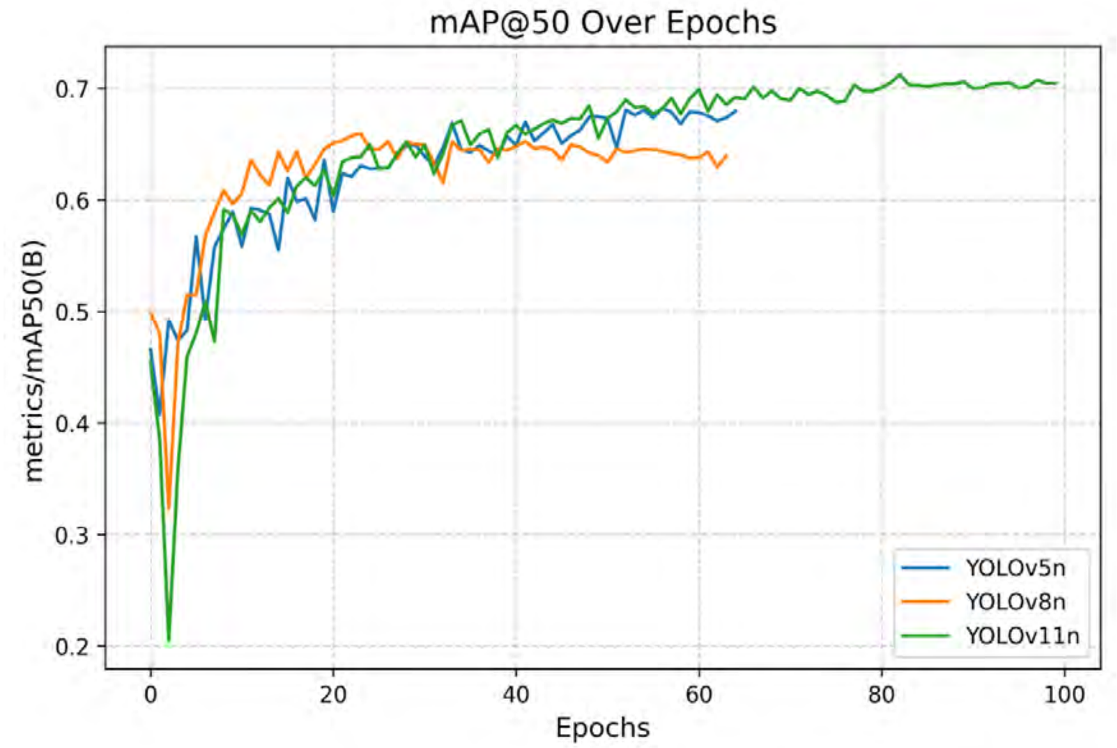


Fig.4.23.a. mAp@50 Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

Fig. 4. 23 shows mAP@50. YOLOv11n tops with 0.7046, YOLOv5n follows with 0.6796, and YOLOv8n records 0.6391. While YOLOv8n is slightly behind, its learning curve is smoother, ensuring reliable performance across the training process.

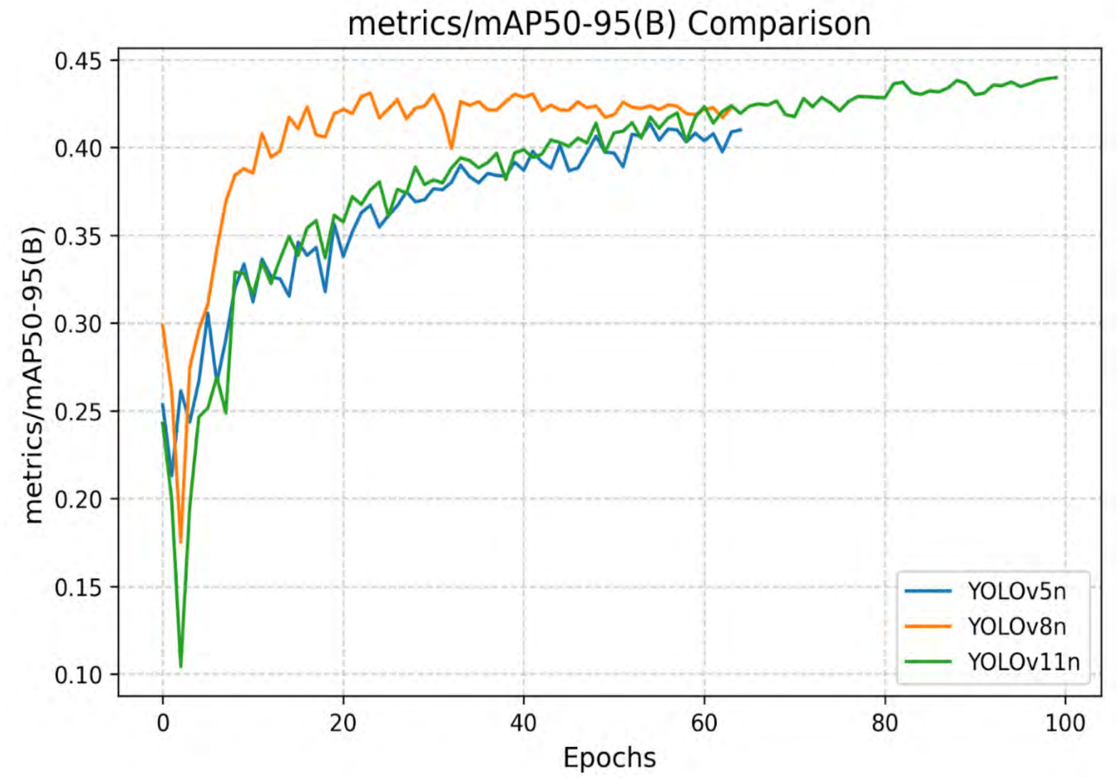
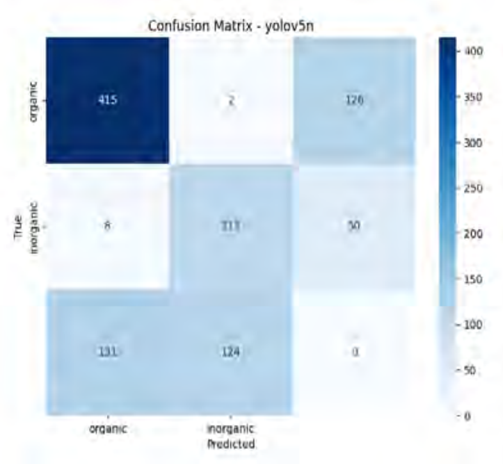


Fig.4.0.24.b. mAp@50-95 Over Epochs for YOLOv5n, YOLOv8n, YOLOv11n.

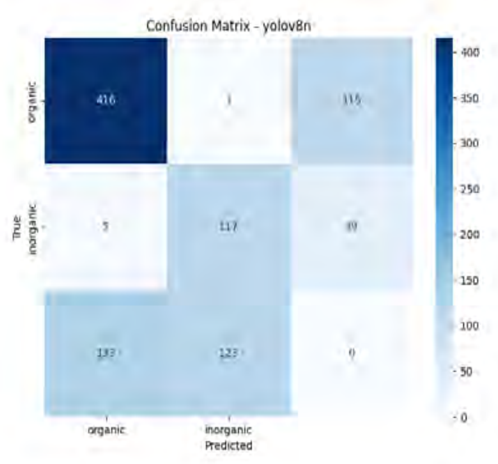
On stricter IoU thresholds, YOLOv11n scores 0.4398, YOLOv8n 0.4226, and YOLOv5n 0.4099. Here, YOLOv8n surpasses YOLOv5n and comes close to YOLOv11n, highlighting its generalization strength at more demanding evaluation levels.

D. Confusion Matrix Analysis

The confusion matrix gives insight into the classification performance of the three models for Biodegradable and Non-Biodegradable waste. **Fig. 4.24.** shows the confusion matrix for (a) YOLOv5n, (b)YOLOv8n and (c)YOLOv11n respectively.



(a)



(b)

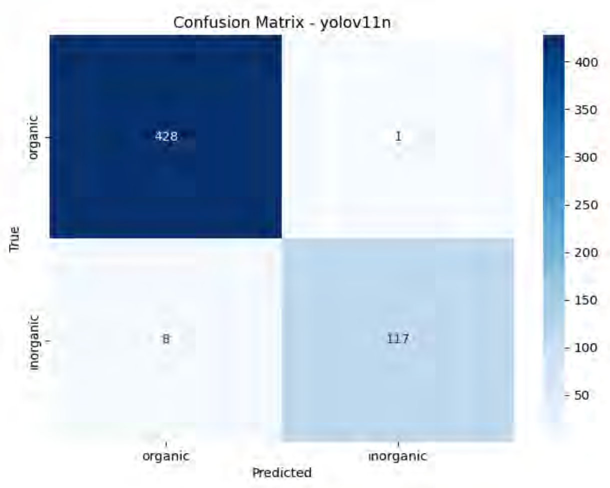


Fig.4.24. Confusion Matrix of (a) YOLOv5n, (b) YOLOv8n, (c) YOLOv11n

The confusion matrix gives insight into the true positives, false positives, and false negatives of the three models. True positives represent cases where waste was correctly identified. For example, in **Fig. 4. 24** (c), YOLOv11n correctly detected 433 instances of Biodegradable waste like banana peels and 117 instances of non-biodegradable waste such as plastic bottles. False positives indicate incorrect detections or misclassifications. From **Fig. 4.24** (a), YOLOv5n incorrectly classified 8 non-biodegradable items such as plastic bags as Biodegradable, while also misclassifying 126 Biodegradable items such as leaves as non-biodegradable. True negatives are not directly represented in this matrix because the background is not a predicted class. This can be thought of as a rock not being classified as waste. False negatives occur when waste is missed or misclassified. For example, YOLOv5n misclassified 126 Biodegradable food scraps as non-biodegradable and 8 non-biodegradable cans as Biodegradable.

Examining the per class metrics highlights the differences. For Biodegradable waste, YOLOv5n achieved a true positive count of 415 with 8 false positives and 126 false negatives. This gave a precision of 0.9811, a recall of 0.7671, and an F1 score of 0.8611. YOLOv8n showed 426 true positives, 5 false positives, and 118 false negatives. This resulted in a precision of 0.9881, a recall of 0.7834, and an F1 score of 0.8741. YOLOv11n had the highest Biodegradable waste performance with 433 true positives, 8 false positives, and only 1 false negative. This led to a precision of 0.9817, a recall of 0.9977, and an F1 score of 0.9896.

For non-biodegradable waste, YOLOv5n had 115 true positives, 126 false positives, and 8 false negatives. This gave a precision of 0.4772, a recall of 0.9350, and an F1 score of 0.6323. YOLOv8n achieved 117 true positives, 115 false positives, and 5 false negatives. This produced a precision of 0.5043, a recall of 0.9590, and an F1 score of 0.6605. YOLOv11n showed strong performance with 117 true positives, only 1 false positive, and 8 false negatives, giving a precision of 0.9915, a recall of 0.9360, and an F1 score of 0.9630.

Looking at the confusion matrix alone, YOLOv11n performs best, especially with very high recall for Biodegradable waste and precision for non-biodegradable waste. However, when considering all evaluation criteria together, the conclusion is different. YOLOv8n shows the most balanced results across training and validation losses, F1-scores over epochs, and detection metrics such as precision and mAP. It achieves reliable performance without large fluctuations and delivers consistent results for both Biodegradable and Non-Biodegradable waste.

While YOLOv11n has the strongest confusion matrix results, YOLOv8n performs more consistently across all metrics. Its stable learning curves, competitive mAP values, and balanced F1-scores make it the most practical choice for deployment. Therefore, YOLOv8n is selected as the best model for the waste segregation rover because it provides the best overall balance of accuracy, stability, and generalization in real-world conditions.

E. Waste recognition demo by the ML models

Fig. 4.25., Fig. 4.26. and **Fig. 4.27**. shows the first few rows of the test dataset being recognized by the YOLO models.

Fig.4.25. Recognized waste by YOLOv5n

Fig.4.26. Recognized waste by YOLOv8n

Fig.4.27. Recognized waste by YOLOv11n

4.2.3 Robotic Arm

A robotic arm is used to segregate waste. To sum up the process, the rover will move in a zig-zag pattern and whenever it detects waste, the robotic arm will carefully pick it up and place the waste sorting system. The arm is being used as a "Pick and Place" bot and to do the respective movement, 3-DOF is chosen. The schematic diagram below illustrates its key components.

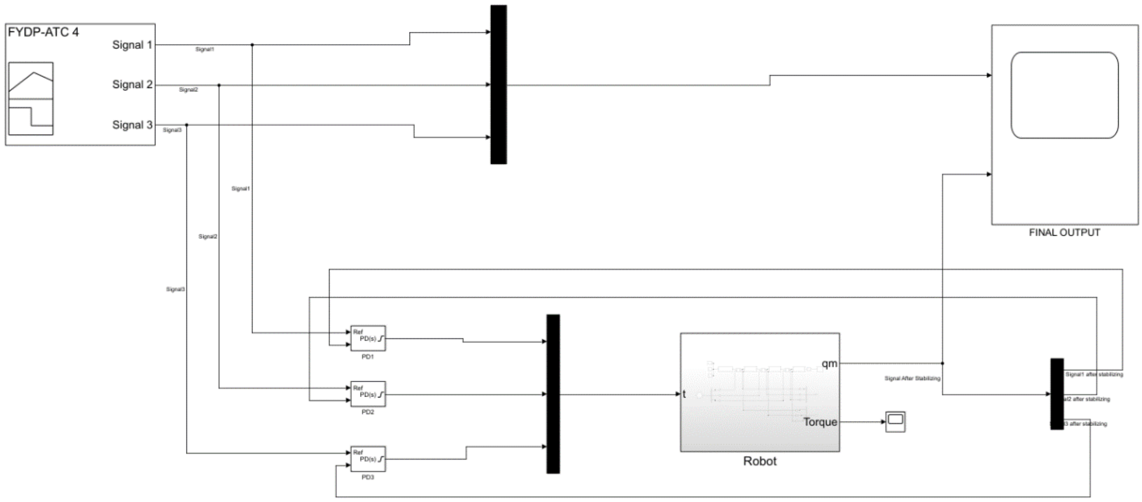


Fig. 4.28. Schematic Diagram of the Robotic Arm System

The Schematic Diagram above represents the control architecture for a robotic system. The simulation of a 3 Degrees-of-Freedom (3DOF) robotic arm was modeled in MATLAB Simulink using Simscape Multibody. It shows how three input signals labelled as Signal 1, Signal 2 and Signal 3 are processed through Proportional - Derivative (PD) controllers labelled as PD1, PD2 and PD3 to generate control commands. These processed signals are then fed into the robot block, which responds by producing the proper positional movements of the arm. The robot's performance is continuously monitored, and the resulting signals are compared to ensure accurate motion and control. The comparison is shown in the Final Output "Scope" block.

A. Kinematics

While working on the robotic arm, considering kinematics is crucial as it focuses on how objects move without considering the forces or torques that cause that motion. In robotics, the analytical examination of a manipulator's motion is known as robot kinematics. It is crucial to accurately consider the kinematics for understanding the functionality.

Kinematics approaches are primarily divided into two parts: Cartesian space and Quaternion space. They represent the manipulator's configuration with transformations between coordinate systems consisting of both translation and rotation.

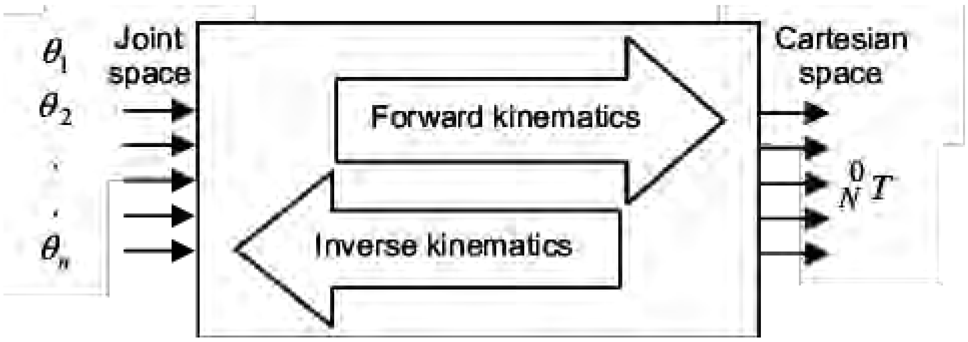


Fig.4.29. Kinematics Block

According to the block, forward kinematics used joint space variables to determine the position of the end-effector in Cartesian space. On the other hand, inverse kinematics takes a desired end-effector position in Cartesian space and calculates the joint variables needed to achieve the position.

B. Inverse Kinematics

The robotic arm is designed in such a way so that it can position its claw for a precise waste pickup mechanism.

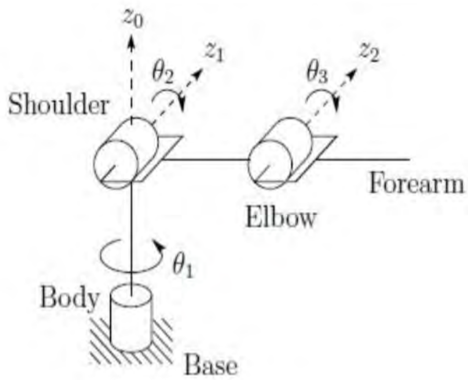


Fig. 4.30 Free Body Representation of 3DOF

The arm will consist of three servo motors: θ_1 will be at the base, controlling horizontal plane rotation whereas, θ_2 and θ_3 will oversee the vertical plane positioning. The following formulas that are derived using inverse kinematics are the ones that determine the joint angles that will be needed to reach a target point (x, y. z) in Cartesian space. The lengths of the arm segments are denoted as M1 (Base to Joint 1), M2 (First joint to Claw).

- **Base Servo Angle (θ_1)**

The base servo angle rotates the arm in a horizontal motion aligning with the targets (x, y) coordinates. The angle is calculated as:

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

The angle controls the horizontal axis of the vertical plane, denoted as x′

$$x' = \sqrt{x^2 + y^2}$$

- **Vertical Plane angles (θ_2 and θ_3)**

The angles θ_2 and θ_3 are computed to the position the claw at the target (x′, z) in the vertical plane. For simplicity purposes it is assumed that the claw is not equipped with an additional degree of freedom, making the arm a two - link planar manipulator. The distance to the target in the vertical place

$$P = \sqrt{x'^2 + z^2} \quad (4.2.3.1)$$

The angle θ_3, between the second arm segment and the line connecting the first joint to the target, is calculated to ensure the arm extends upward, selecting the positive solution among possible configurations:

$$\theta_3 = \boldsymbol{\pi}\text{-}cos^{-1}\left(\frac{-P^2 + M^2{}_1 + M^2{}_2}{2M_1M_2}\right) \quad (4.2.3.2)$$

The angle θ_2, the orientation of the first arm segment relative to the horizontal axis x′, is determined as:

$$\theta_2 = tan^{-1}\left(\frac{z}{x'}\right) + tan^{-1}\left(\frac{M_2 sin(A_2)}{M_1 + M_2 cos(A_2)}\right)$$

(4.2.3.3)

Next the Input joint signal is further processed through a PD controller, all to smoothen the movement replicating a real-world scenario [23].

C. Proportional - Derivative (PD) controller:

PD control is a combination of feedforward and feedback control, because it operates on both the current process conditions and predicted process conditions. In PD - control, the control output is a linear combination of the error signal and its derivative. It controls the output in a linear combination of the error signal and its derivative. In PD, proportional controls the damping of the fluctuation where the derivative predicts the process error. PD control behavior is mathematically illustrated using the following equation: **[23]**

$$c(t) = K_c\left(e(t) + T_d\frac{de}{dt}\right) + C \quad (4.2.3.4)$$

Were,

- **c(t)** = Controller Output
- **Kc** = Proportional Gain
- **e** = error
- **C** = initial value of controller

The equation acts as a combination of the P-only and D-only control factors. Through these factors, the purpose is divided where the D-only control is to predict the error to increase stability of the closed loop system. PD control is not commonly used because of the lack of the integral term. However, for the robotic arm, the steady state operation or the purpose of integral term (I) is being minimized through the schematic diagram

4.2.3a Pre-Sorting

A. Configuring the robotic arm (Robot-Child Sheet**)**

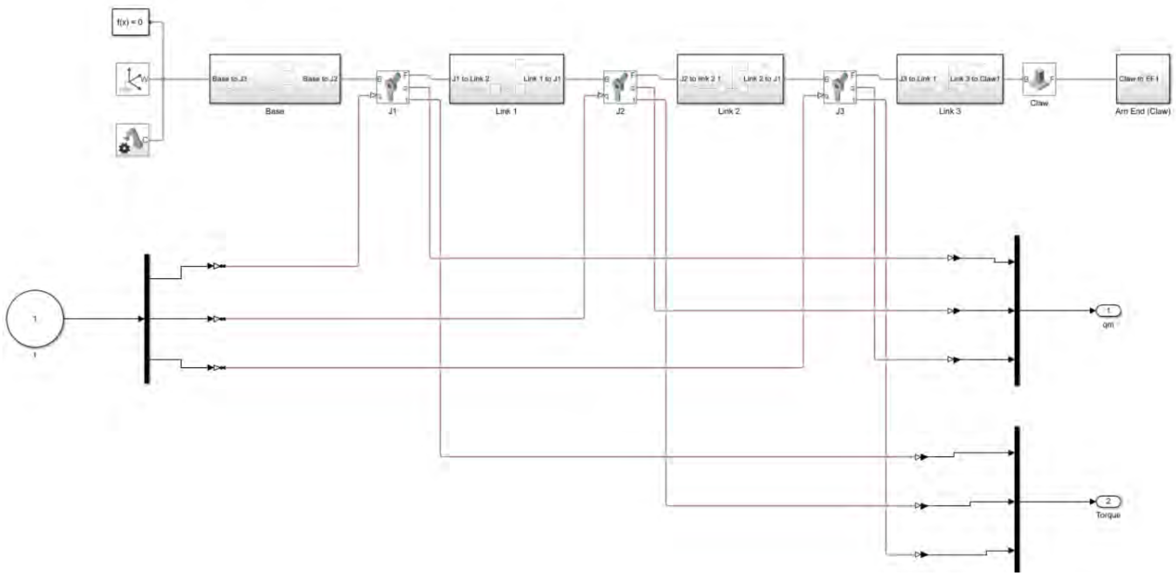


Fig.4.31. Child Sheet of Robot Block (Pre-Sorting)

The child sheet above comprises the main specification, link lengths and the toque workaround. The arm consists of a fixed base, three revolute joints (J1, J2, J3), and a claw-shaped end effector. The physical dimensions are as follows:

- Base: **4 cm** diameter and **5 cm** length
- Link 1: **10 cm**
- Link 2: **20 cm**
- Link 3: **15 cm**
- End Effector length: **1.5cm**
- Mass of Link 1: **0.085kg**
- Mass of Link 2: **0.17kg**
- Mass of Link 3: **0.1kg**
- Mass of End Factor: **0.1kg**

The link lengths are chosen in such a way so that it can fulfil its purpose in the approach, which is being a pick and place robot,

A constant time input drives the simulation. Each revolute joint outputs angular position (**qm**) and torque, which are routed through multiplexers for analysis. The links are modeled as rigid bodies with accurate lengths, allowing for realistic motion. The model successfully simulates joint movements and torque requirements, offering a solid foundation for future control design and dynamic analysis.

Overall, the model provides a simplified yet accurate representation of a 3DOF robotic manipulator, suitable for studying motion behavior and developing control strategies.

B. Inverse Kinematics

- **Applied Input Signal:**

Using the formulas (**4.2.3.1 - 3**), the input signal / joint angles are calculated over a time vector of t = {0,1,2,3,4,5,6,7,8,9,0}. The joint angles for the servos are as followed:

- θ_1 : Signal1 = [0.3 0.3 0.3 0.5 -0.4363 -0.1309 0.1745 0.48 0.7854 0.7854 0.7854];
- θ_2 : Signal2 = [-1 -1 -1 -2 0.1745 0 0.1745 0.48 0.7854 0.7854 0.7854];
- θ_3 : Signal3 = [-2 -2 -2 -0.22 0.2618 0.1309 0.1745 0.48 0.7854 0.7854 0.7854];

The Joint angles are further processed and smoothened using a PD controller.

C. Proportional - Derivative (PD) Controller

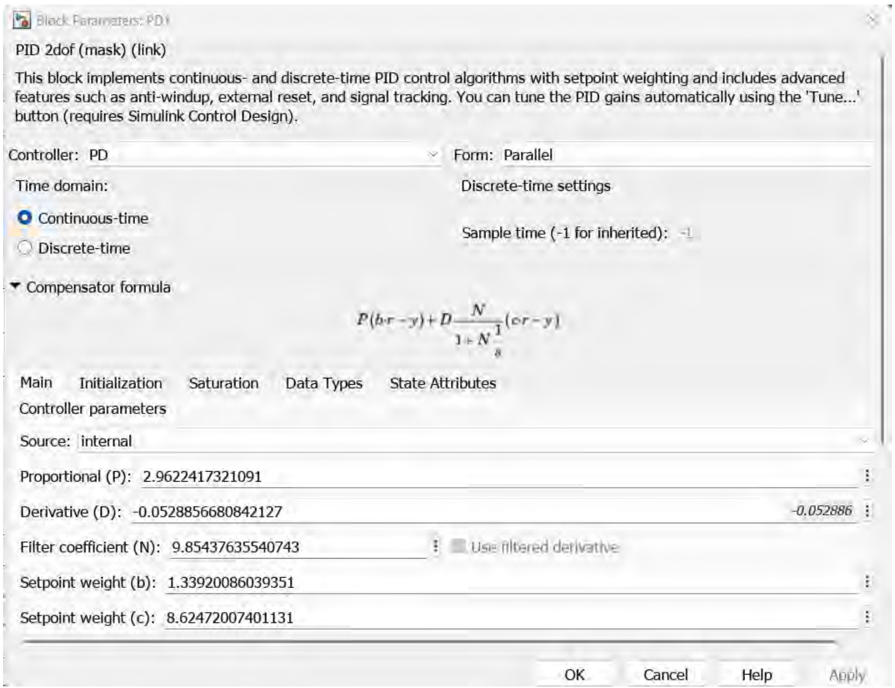


Fig.4.32. Parameter Block of Proportional-Derivative (PD) Controller

The parameter block represents a PD controller configured in continuous time and parallel form; it is primarily used for motion control in the robotic arm. It regulates joint position by minimizing the error between the desired setpoint and the actual joint output. These values scale the error for immediate correction, as well as predicting it to reduce overshoot all while smoothing the derivative action. Ultimately modifying the reference signal that influences proportional and derivative terms. These values were automatically tuned using an in-built tuning function available within the Simulink model.

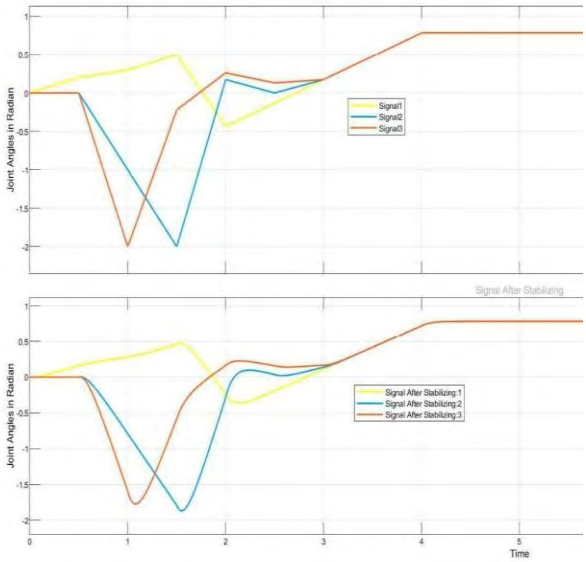


Fig.4.33. Stabilized input signal from PD shown via Scope block

The graph shows the before and after image of applying a PD controller to the input signal. It can be observed that the initial sharp turns or edges of the signals are now smoothed due to the addition of a PD controller. This means that the movement of the robotic arm will be smoother now representing a real-world scenario.

D. Torque Configuration (Choosing the actuator motors required to drive the robotic arm**)**

The selection process of actuator motors is a crucial part of designing robotic arms. The motors need to be chosen wisely so that it can effectively handle waste segregation tasks for a varying payload condition. By analyzing the maximum required torque for each joint as a function of payload, we can determine the appropriate motor specifications to meet the design requirements, such as handling from 0.5kg to 3kg with a collector error margin of ≤0.5%. As the robotic arm was modeled in MATLAB Simulink using Simscape Multibody, there is an option in the Robot>Revolute Joints that automatically computes the torque of the motor required to do movement on the provided input angle and specification of the arm. The option block is given below:

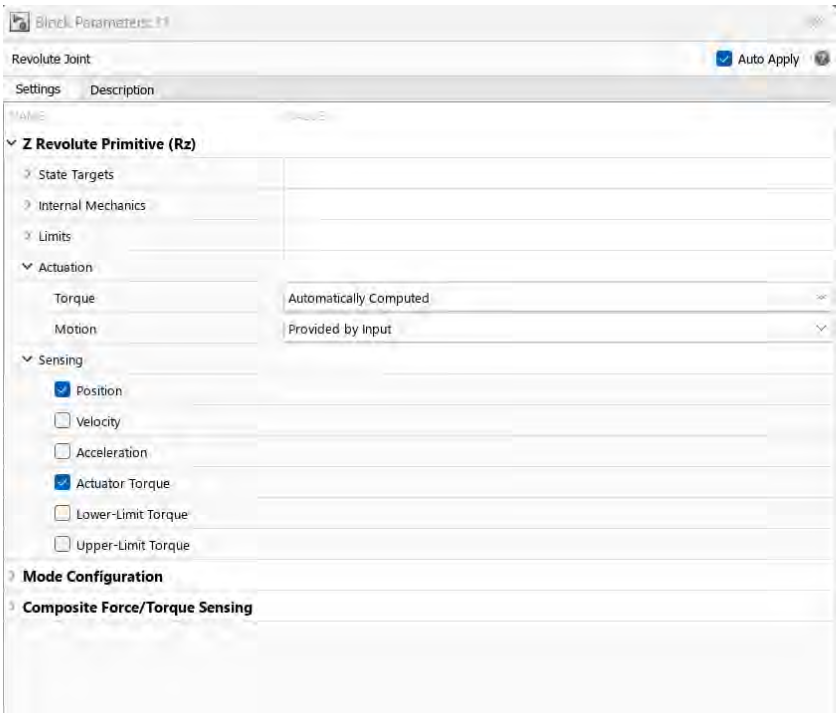


Fig.4.34. Automatic computation block of torque

Through the configuration torque values are automatically computed. Now using a matlab script and through the script it is seen what the torque values are for a varying range of payload.

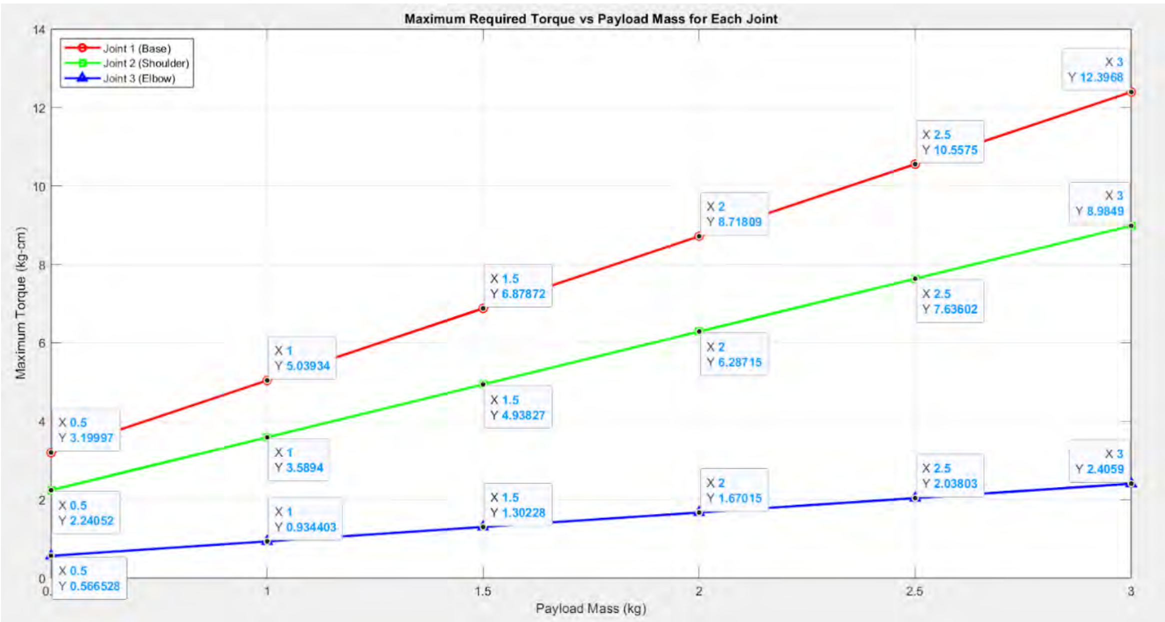


Fig.4.35. Maximum required torque vs Payload mass

The graph illustrates the maximum required torque vs payload mass for each joint for a 3DOF robotic arm, with torque increasing linearly from 0.5kg to 3kg payloads. Based on the relation, for Joint 1 (Base) the highest torque is at 12.3968 kg-cm, followed by Joint 2 (shoulder) at

8.9849 kg-cm, and Joint 3 (elbow) at 2.4059 kg-cm. Based on the torque requirement the motors were chosen. Along with the motor, the power required to drive the entire robotic arm all while picking the waste of varying mass is being computed and sent to the EPS system.

4.2.3b post-sorting

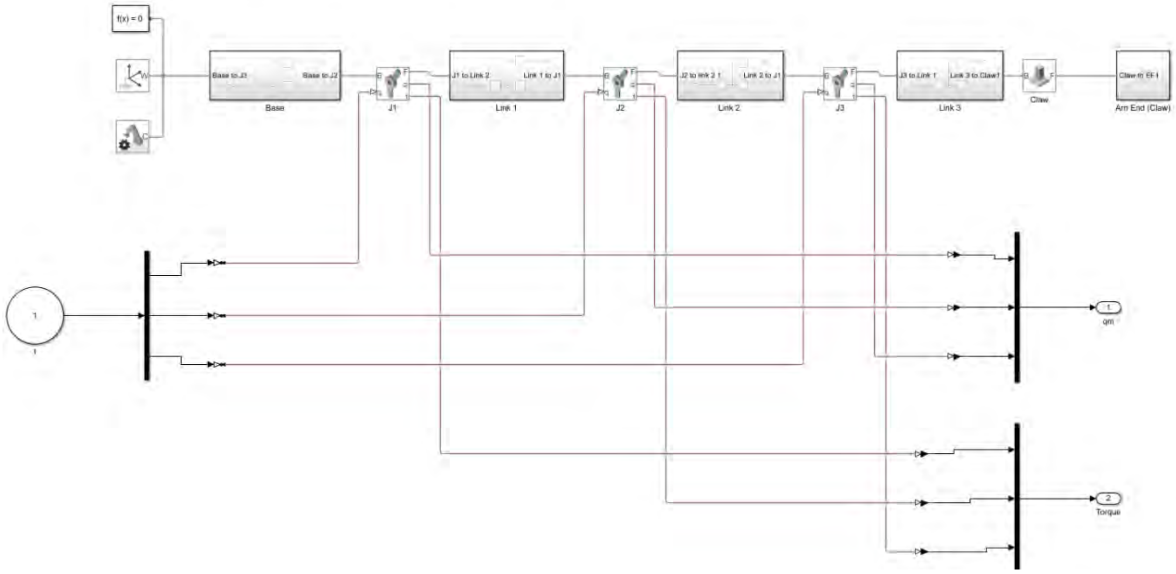


Fig.4.36. Child Sheet of Robot Block (Post-Sorting)

The child sheet above comprises the main specification, link lengths and the toque workaround. The arm consists of a fixed base, three revolute joints (J1, J2, J3), and a claw-shaped end effector. The physical dimensions are as follows:

- Base: **4 cm** diameter and **5 cm** length
- Link 1 = **5 cm**
- Link 2: **25 cm**
- Link 3 = **15 cm**
- End Effector length = **5 cm**
- Mass of Link 1: **0.2kg**
- Mass of Link 2: **0.3kg**
- Mass of Link 3: **0.3kg**
- Mass of End Factor: **0.5kg**

The link lengths are chosen in such a way so that it can fulfil its purpose in the approach, which is being a waste sweeper

A constant time input drives the simulation. Each revolute joint outputs angular position (**qm**) and torque, which are routed through multiplexers for analysis. The links are modeled as rigid bodies with accurate lengths, allowing for realistic motion. The model successfully simulates joint movements and torque requirements, offering a solid foundation for future control design and dynamic analysis.

Overall, the model provides a simplified yet accurate representation of a 3DOF robotic manipulator, suitable for studying motion behavior and developing control strategies.

A. Inverse Kinematics

- **Applied Input Signal:**

Using the formulas (4.2.3.1 - 3), the input signal / joint angles are calculated over a time vector of t = {0,1,2,3,4,5,6,7,8,9,0}. The joint angles for the servos are as followed:

- θ_1 **:** Signal1 = [-0.5, -0.5, -0.5, -0.5, 1, 1, 1, 0.8, 0.8, 0, 0];
- θ_2 **:** Signal2 = [-1.5, -1.5, -1.5, -1.5, -0.5, -0.5, -0.5, -0.5, -0.5, 0, 0];
- θ_3 : Signal3 = [-0.3, -0.3, -0.3, 0, 0, 0, 0, 1, 1, 0, 0];

The Joint angles are further processed and smoothened using a PD controller.

B. Proportional - Derivative (PD) Controller

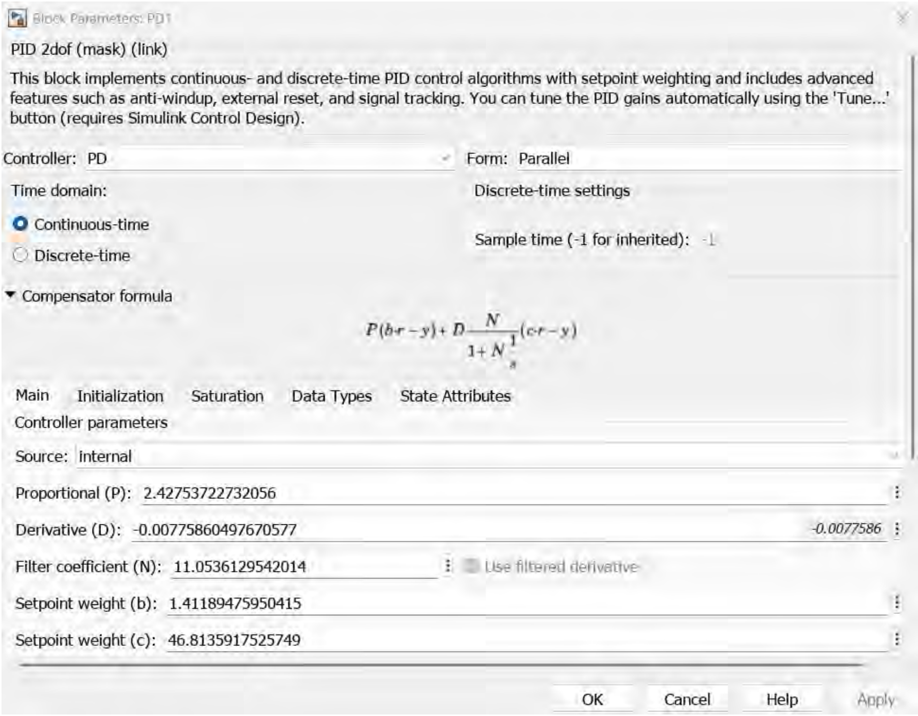


Fig.4.37. Parameter Block of Proportional-Derivative (PD) Controller

The parameter block represents a PD controller configured in continuous time and parallel form; it is primarily used for motion control in the robotic arm. It regulates joint position by minimizing the error between the desired setpoint and the actual joint output. These values scale the error for immediate correction, as well as predicting it to reduce overshoot all while smoothing the derivative action. Ultimately modifying the reference signal that influences proportional and derivative terms. These values were automatically tuned using an in-built tuning function available within the Simulink model PD.

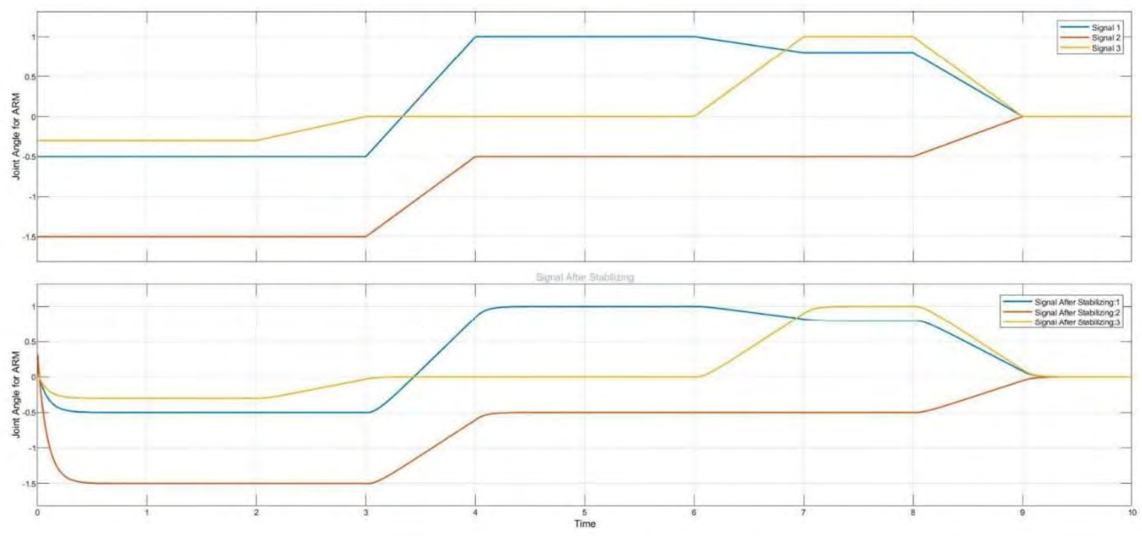


Fig.4.38. Stabilized input signal from PD shown via Scope block

The graph shows the before and after image of applying a PD controller to the input signal. It can be observed that the initial sharp turns or edges of the signals are now smoothed due to the addition of a PD controller. This means that the movement of the robotic arm will be smoother now representing a real-world scenario.

C. **Torque Configuration (**Choosing the actuator motors required to drive the robotic arm**)**

The selection process of actuator motors is a crucial part of designing robotic arms. The motors need to be chosen wisely so that it can effectively handle waste segregation tasks for a varying payload condition. By analyzing the maximum required torque for each joint as a function of payload, we can determine the appropriate motor specifications to meet the design requirements, such as handling from 0.5kg to 3kg with a collector error margin of ≤0.5%. As the robotic arm was modeled in MATLAB Simulink using Simscape Multibody, there is an option in the Robot>Revolute Joints that automatically computes the torque of the motor required to do movement on the provided input angle and specification of the arm. The option block is given below:

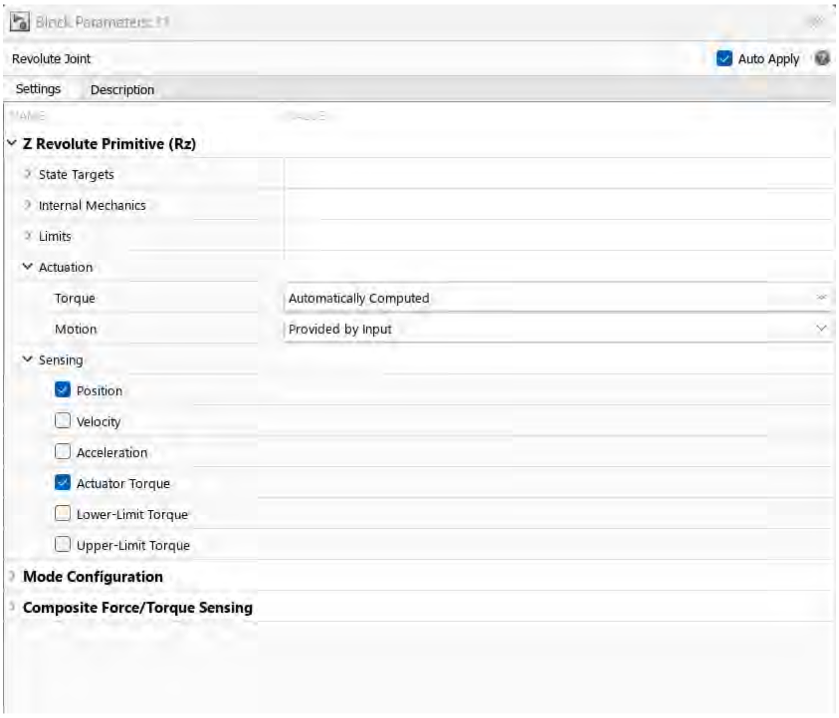


Fig.4.39. Automatic computation block of torque

Through this block torque values are automatically computed. Now using a MATLAB script the torque values for a varying payload mass is seen for each joint

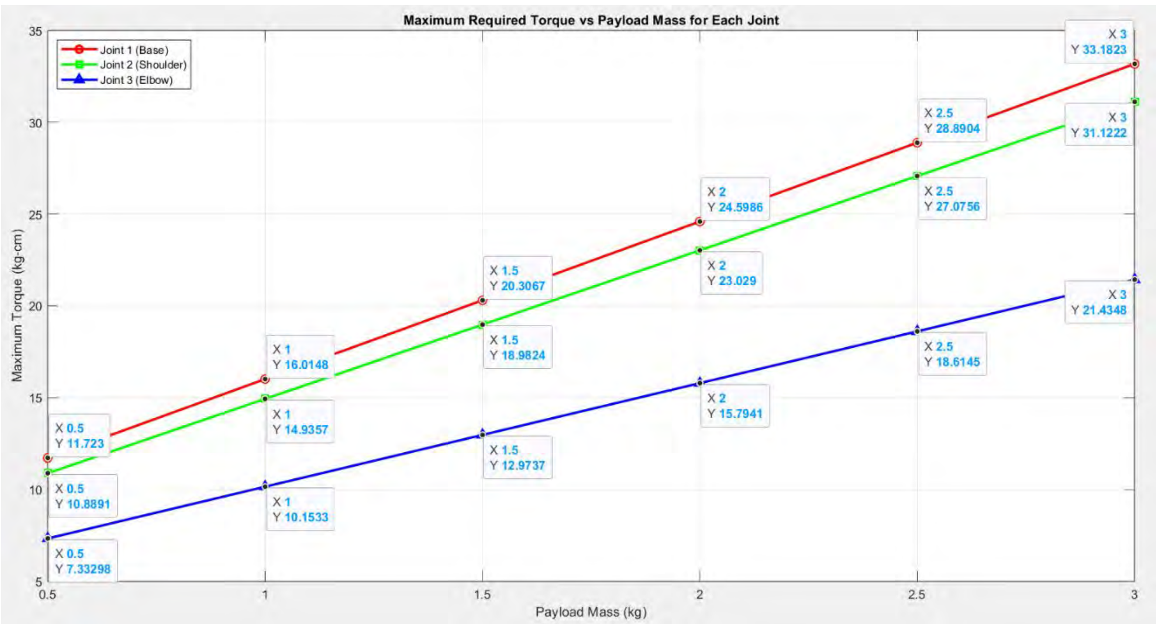


Fig.4.40. Maximum required torque vs Payload mass

The graph illustrates the maximum required torque vs payload mass for each joint for a 3DOF robotic arm, with torque increasing linearly from 0.5kg to 3kg payloads. Based on the relation, for Joint 1 (Base) the highest torque is at 33.1823 kg-cm, followed by Joint 2 (shoulder) at 31.1222 kg-cm, and Joint 3 (elbow) at 21.4348 kg-cm. Based on the torque requirement the motors were chosen. Along with the motor, the power required to drive the entire robotic arm all while picking the waste of varying mass is being computed and sent to the EPS system.

4.2.4 Structure

During the design phase, several notable components were modeled using SOLIDWORKS, so that it can provide a clear and accurate visualization of the physical structure and their potential functionalities. The structure of the rover's robotic arm and conveyer belt was learnt and implemented with the help of GrabCAD Library [27]. The models that were designed are as follows:

Bin:

Fig.4.41. Bin with a divider for proper segregation

For both approaches this representation of bin will be used. The dual compartment waste bin ensures a seamless segregation process. The divider divides the waste dump into two equal sections - Biodegradable and Non-Biodegradable. There will be a servo motor placed at the bottom of the bin for dynamic rotation. A signal from the Raspberry Pi 5 will prompt the motor to turn in such a way so that the bin categorized compartment is just below the arms drop point will trigger the motor movement upon waste detection and classification. There will also be two ultrasonic sensors on each side of the compartment with the sole purpose of sending alerts if the bin becomes full. Keeping the specification in mind the bin will be chosen with the following parameters:

- Length = **15 cm**
- Width = **30 cm**

Some other designs were modelled specifically for pre-sorting and post-sorting approaches.

4.2.4a Pre-Sorting

Fig.4.0.42 Claw End Effector [27]

For the Pre-Sorting approach the effector is supposed to work as a “Pick and Place” system, hence the claw was chosen. The claw was selected due to its compact, servo-actuated design that enhances the robotic arm’s grasping capability, suitable for lightweight and elongated items.

4.2.4.b. Post-Sorting

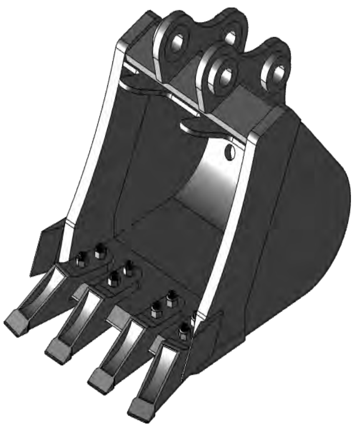


Fig.4.0.44 Waste Sweeper End Effector [27]

Fig.4.45. Conveyor belt [27]

For the post-sorting approach, the end effector is supposed to work as a “Waste Sweeper” system, where it will collect the wastes altogether and then place it in the conveyor belt respectively. For the system to work effectively the following parameters are decided for each system. They are as follows:

Waste Sweeper End Effector

- Width = **10 cm**
- Length = **32 cm**

Conveyor Belt

- Conveyor Belt of length = **60 cm**
- Conveyor Belt of Width = **15cm**
- Height of the stand will be **10cm**

The design process leveraged SOLIDWORKS dynamic simulation capabilities that allowed the team to visualize and refine the respective functionalities before physical prototyping, saving both time as well as resources.

4.2.5 Budget

- **Design Approach 1: Post Sorting Method:**

TABLE 4.5: BUDGET OF POST-SORTING METHOD

Item	Specification	Unit Price	Quantity	Total Price
Ultrasonic Sensors	HC-SR04, range: 2 cm–400 cm	75	10	750
Bicycle Chain		1200	2	2400
YDLIDAR X4	Scanning range: 10m, 360° field of view	9000	1	9000
Plastic Waste Bins	Durable plastic, capacity: 2.5L, compartments	500	1	500
Arduino Mega	ATmega2560, 16MHz clock speed	5430	1	5430
A58SW31ZY High Torque Worm Geared Reversed Self-Locking Motor	Model: GW7085; Rated Voltage: DC 12V; Current(A): 6; No-load Speed: 42 RPM; Torque(kg.cm): 60; Shaft Type: Left Shaft.	3350	1	3350
Ultra-Torque Dual Shaft Metal Gear Coreless Servo	The motor operates at 6V~8.4V, stall torques of 31 kg-cm at 7.2V and 35 kg-cm at 8.4V.	4136.9	2	8273.8
TD-8120MG Digital High Torque Servo Motor Metal Gear	Operating Voltage: 4.8V~7.2V	1450	1	1450
3D Printing Filament	PLA, 1.75mm diameter	3640	2	7280
IMU (Inertial Measurement Unit)	MPU6050, 6-axis gyro and accelerometer	208	1	208
Battery Pack	Li-Po; 11.1V; 10000 mAh	6600	1	6600
Raspberry Pi 5		15420	1	15420
Raspberry Pi Camera Module V2	8 Megapixel,1080p	4690	1	4690
Raspberry Pi 5 Active Cooler		740	1	740
Conveyor Belt	50 cm length and 30cm width, industrial- grade PVC or PU belt, optimized for friction.	1	8000	8000
Drive Motor	1-2 HP motor	1	15,000	15,000
Rollers	Set of rollers for conveyor design	4	2,500	10,000
NIR Sensors	Near-Infrared sensor for material classification.	1	10,000	10,000
Belt Tensioners	Mechanism for belt tension.	2	1500	3000
Control Panel	Basic motor speed control unit.	1	5000	5000
Joining Clamps/Screws	Hardware for assembly.	Bulk	2000	2000
TOTAL				**119,091.8**

- **Design Approach 2: Pre-Sorting Method:**

TABLE 4.6: BUDGET OF PRE-SORTING METHOD

Item	Specification	Unit Price	Quantity	Total Price
Ultrasonic Sensors	HC-SR04, range: 2 cm–400 cm	75	10	750
Bicycle Chain		1200	2	2400
YDLIDAR X4	Scanning range: 10m, 360° field of view	9000	1	9000
Plastic Waste Bins	Durable plastic, capacity: 2.5L, compartments	500	1	500
Arduino Mega	ATmega2560, 16MHz clock speed	5430	1	5430
Servo Motor MG995 180 Degree Rotation	Operating Voltage: 4.8~7.2V, Torque: 10~12kg-cm	435	3	1305
TD-8120MG Digital High Torque Servo Motor Metal Gear	Operating Voltage: 4.8V~7.2V	1450	1	1450
Motor Driver 43A BTS7960	Operating Voltage: 6V ~27V Maximum 43A Current	625	2	1250
Gear Motor	37GB 12V High Torque; 60RPM	2200	2	4400
3D Printing Filament	PLA, 1.75mm diameter	3640	2	7280
IMU (Inertial Measurement Unit)	MPU6050, 6-axis gyro and accelerometer	208	1	208
Battery Pack	Li-Po; 11.1V; 10000 mAh	6600	1	6600
Raspberry Pi 5		15420	1	15420
Raspberry Pi Camera Module V2	8 Megapixel,1080p	4690	1	4690
Raspberry Pi 5 Active Cooler		740	1	740
TOTAL				**62,423**

4.3 Identify optimal design approach

4.3.1. Pre-Sorting Approach

The pre-sorting approach includes a rover, a robotic arm for picking and placing the waste into a divided bin (Biodegradable/Non-Biodegradable) and finally classifying waste after detecting it using a camera with the help of ML models (YOLOv11n). It uses fewer subsystems (Camera, Rover, Robotic Arm, Bin, Alert System).

<u>**Evaluation:**</u>

- **Cost**:
  - **Power Consumption**: Total system power is **309.24 W.** Battery voltage requirements range from **7.898 V** to **9.580 V** for velocities of **0.5 m/s** to **1.5 m/s**.
  - **Components**: It has fewer subsystems, potentially reducing hardware costs. The robotic arm uses lighter components, which may be cheaper to

manufacture. Moreover, the rover uses 2 motors, and the robotic arm requires lower torque (e.g., Joint 1: **12.3968 kg-cm**), implying smaller, less expensive motors compared to post-sorting.

- ◆ **Implementation Cost:** According to the design specifications, the total cost for pre-sorting is **56,338 takas.**

- **Efficiency**:
  - ◆ **Sorting Efficiency**: Classification happens during detection and sorting is done instantly. However, collection time will increase when there is a pile of waste.
  - ◆ **Energy Efficiency**: Final energy usage is **114.25 kJ,** and total electrical energy is **52.411 Wh**.
  - ◆ **ML Model**: YOLOv11n shows an F1-score of **0.9896** (Biodegradable) and **0.9630** (non-biodegradable), ensuring high accuracy in real-time sorting.
- **Usability**:
  - ◆ Compact design with fewer subsystems makes it easier to deploy in urban environments. The robotic arm is intuitive for precise waste handling.
  - ◆ The system sends IoT alerts when the bin is full for user convenience.
- **Manufacturability**: It has a simpler architecture with fewer components. The robotic arm's lighter links reduce material costs and manufacturing complexity.
- **Impact**: Promotes sustainability by accurately sorting waste for recycling. Lower energy consumption (**114.25 kJ**) reduces carbon footprint.
- **Sustainability**: The Pre-Sorting Approach comes as a more efficient solution in terms of Environmental, Economic and Social context. The compactness of the system allows it to be more energy efficient and due to it being environmentally friendly the carbon footprint emission is also minimum. This early-phase intervention ensures lower operational costs and a better productivity rate. Additionally, lower initial Capital Investment further supports its economic viability by requiring less infrastructure and processing complexity.
- **Maintainability**: Fewer subsystems mean less maintenance. As this system is more compact so maintaining the system is easier.

4.3.2. Post-Sorting Approach

The post-sorting approach collects waste first, then sorts it using a conveyor belt and a stationary NIR sensor for classification. It involves more subsystems (Camera, Rover, Robotic Arm, Conveyor Belt, Bin, Alert System).

<u>**Evaluation:**</u>

- **Cost**:
  - ◆ **Power Consumption**: Total system power is **340.546 W**. Battery voltage requirements range from **6.705 V** to **7.669 V**, but total energy usage is much higher at **900 kJ**.

- ◆ **Components**: Additional subsystems like the conveyor belt and component NIR sensor increase hardware costs. The robotic arm uses heavier components, raising material costs. Also, the rover uses 4 motors, and the robotic arm requires higher torque (e.g., Joint 1: **33.1823 kg-cm**), which requires more expensive motors.

- ◆ **Implementation Cost:** According to the design specifications, the total cost for pre-sorting is **119,091.8 taka.**

- **Efficiency**:
  - ◆ **Sorting Efficiency**: Sorting happens after collection with the robotic arm via the conveyor belt and NIR sensor, which may introduce delays. However, this ensures more accuracy in sorting due to the additional NIR sensor.
  - ◆ **Energy Efficiency**: The final energy usage is **900 kJ** and total electrical energy of **90.483 Wh**.
  - ◆ **ML Model**: Uses the same YOLOv11n model with identical performance (F1-scores of **0.9896** and **0.9630**), but the NIR sensor adds a secondary classification layer, potentially improving accuracy for ambiguous cases.
- **Usability**:
  - ◆ More complex systems with additional components (conveyor belt, NIR sensor), which may complicate deployment and operation in urban settings. The "waste sweeper" robotic arm is efficient for faster waste collection in case of a pile of waste; however, the waste classification requires more time.
  - ◆ Includes IoT alerts for bin capacity (Page 8), similar to pre-sorting.
- **Manufacturability**: As the architecture has additional components and the robotic arm's heavier links and larger end effector increase manufacturing complexity and material costs.
- **Impact**: Promotes sustainability through accurate sorting, but higher energy consumption (**900 kJ**) has higher environmental footprint compared to pre-sorting.
- **Sustainability**: The Post Sorting Approach, due to its increased complexity and the overall tension of the system, requires a lot of energy. However, due to it being a State of the Art it performs strongly in areas like Job Creation, Market Potential and being more efficient in processing waste at a faster rate. Post Sorting systems are better equipped to extract valuable resources after collection, boosting economic opportunities and creating employment within waste management industries.
- **Maintainability**: More subsystems increase the requirement of maintenance.

4.4 Performance evaluation of developed solution

The criteria evaluation of both solutions has been given below:

- **Cost and Efficiency**: Lower power consumption (**309.24 W vs. 340.546 W**) and energy usage (**114.25 kJ vs. 900 kJ**) make it more cost-effective and efficient.
- **Usability and Manufacturability**: A simpler design with fewer subsystems and lighter components enhances usability and ease of manufacturing.

- **Sustainability and Impact**: Lower energy demands reduce its environmental footprint, aligning better with sustainability goals. For better understanding of sustainability, a sustainability matrix can be a tool to quantify the sustainability score. The sustainability matrix provides a comparative assessment of the Pre-Sorting and Post Sorting approaches across Environmental, Economic and Social Criteria to evaluate their overall performance in achieving sustainability goals. The visual representation of Sustainability Matrix Distribution can be found in Fig. 46.

Criteria	Weight	Pre Sorting Approach (Score)	Pre-Sorting Approach (Weighted)	Post-Sorting Approach (Score)	Post-Sorting Approach(Weighted)
Environtment Criteria	0.4				
Carbon Footprint	0.05	4	0.2	4	0.2
Energy Efficiency	0.05	5	0.25	2	0.1
Material Sourcing (High pressure systems)	0.05	3	0.15	3	0.15
E-Waste Management	0.05	1	0.05	1	0.05
Toxic Substance Use	0.05	5	0.25	5	0.25
Renewable Energy Source	0.05	1	0.05	1	0.05
Water Usage	0.05	5	0.25	5	0.25
Economic Criteria	0.3				
Initial Capial Investment	0.05	5	0.25	1	0.05
Operational Costs	0.05	4	0.2	1	0.05
Maintenance Costs	0.05	3	0.15	1	0.05
Lifespan	0.05	4	0.2	5	0.25
Productivity Impact	0.05	2	0.1	2	0.1
Market Potential	0.05	4	0.2	4	0.2
Job Creation	0.05	3	0.15	2	0.1
Local Economic Impact	0.05	3	0.15	4	0.2
Social Criteria	0.3				
Health & Safety	0.05	5	0.25	5	0.25
Community Engagement	0.05	3	0.15	3	0.15
Social Equity	0.05	4	0.2	4	0.2
Educational Impact (Training)	0.05	4	0.2	4	0.2
Quality of Life	0.05	4	0.2	4	0.2
Access to Services	0.05	4	0.2	4	0.2
Cultural Impact	0.05	5	0.25	5	0.25
Total Score			4.05		3.5

Fig.4.45 Sustainability Matrix Distribution Table

The Pre-Sorting Approach comes as a more efficient solution in terms of Environmental, Economic and Social context. The compactness of the system allows it to be more energy efficient and due to it being environmentally friendly the carbon footprint emission is also minimum. This early-phase intervention ensures lower operational costs and a better productivity rate. Additionally, lower initial Capital Investment further supports its economic viability by requiring less infrastructure and processing complexity.

On the other hand, the Post Sorting Approach, due to its increased complexity and the overall tension of the system, requires a lot of energy. However, due to it being a State of the Art it performs strongly in areas like Job Creation, Market Potential and being more efficient in processing waste at a faster rate. Post Sorting systems are better equipped to extract valuable resources after collection, boosting economic opportunities and creating employment within waste management industries.

From a sustainability perspective, while Pre-Sorting leads to greater efficiency, cost reduction, and environmental benefits, the Post Sorting Approach contributes positively to local economic development and social outcomes. Each approach has its strengths and trade-offs, and the choice between the two depends on the priorities of the specific waste management scenario—whether the focus lies on minimizing environmental impact or maximizing economic and social benefits.

A visual representation of the sustainability matrix is given below.

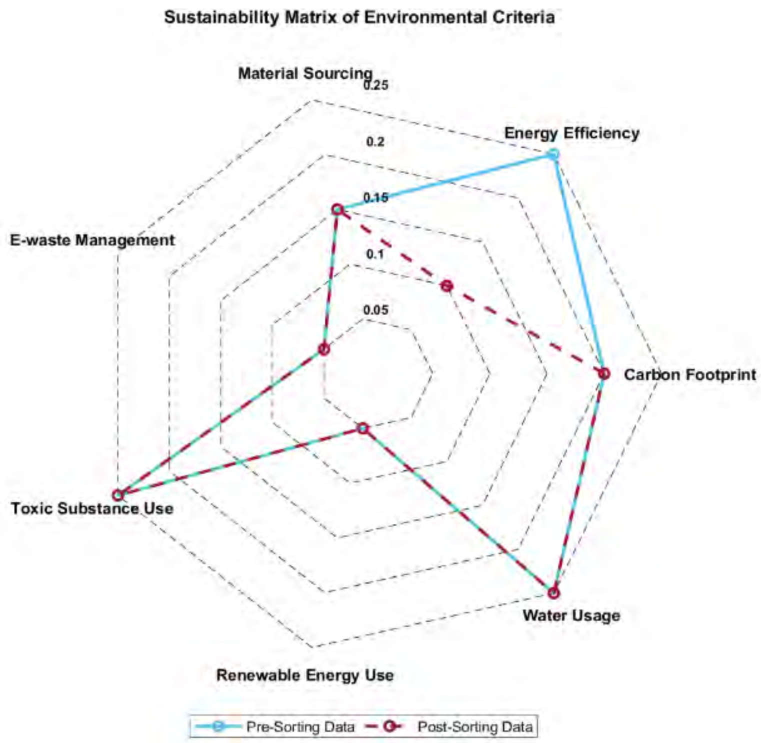


Fig.4.46. Spider Chart for Environmental Criteria

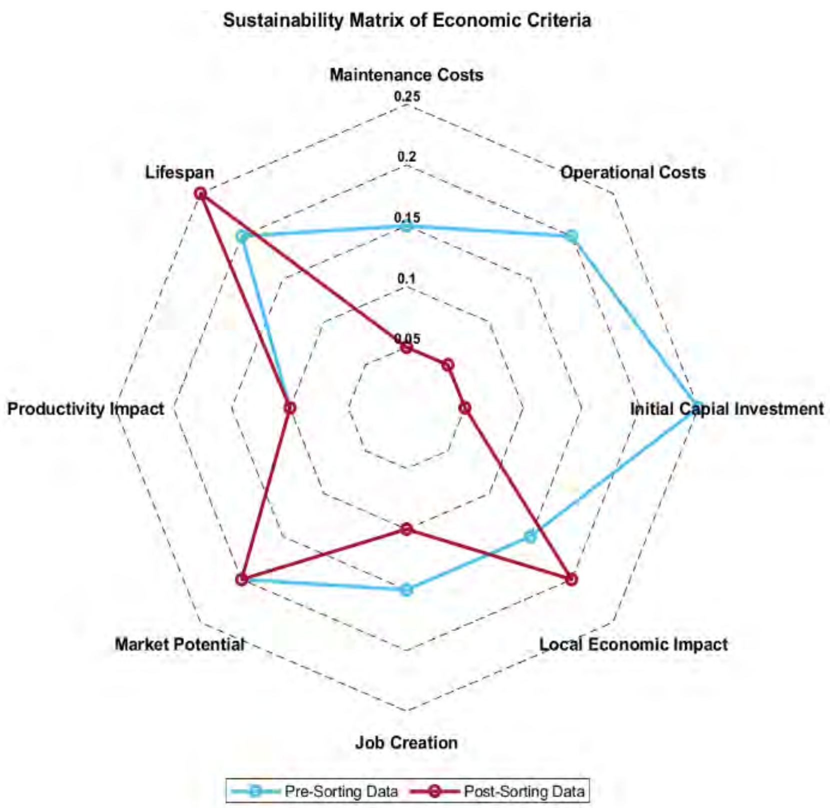


Fig.4.47. Spider Chart for Economic Criteria

Fig.4.48. Spider Chart for Social Criteria

The spider chart reveals that the area representing the pre-sorting data is larger compared to the post-sorting data for both economic and environmental criteria. Since spider charts are designed to indicate that a greater area corresponds to higher sustainability, it can be concluded that the pre-sorting design demonstrates superior sustainability.

- **Maintainability**: Fewer components and points of failure make it easier to maintain.

The post-sorting approach might be potentially more accurate for classifying and sorting waste in ambiguous situations due to the NIR sensor, however, it is less practical due to higher costs, energy demands, and complexity. The pre-sorting approach meets the project's objectives by balancing accurate waste segregation with cost, sustainability, and operational efficiency within the given constraints more effectively.

The comprehensive design and verification process for the autonomous waste segregation rover has shown a practical, scalable, and impactful solution to urban waste management challenges. Through rigorous simulation, two design approaches—pre-sorting and post-sorting were compared through key metrics such as cost, power efficiency, usability, manufacturability, sustainability, and maintainability. After that, pre-sorting was proved to be the optimal solution.

TABLE 4.7: SUMMARY OF DESIGN VERIFICATION

Criteria	Pre-Sorting	Post-Sorting
Cost	**Power:** 309.24 W	**Power:** 340.546 W
	Implementation Cost: **56,338 BDT**	Implementation Cost: **119,091.8 BDT**
Efficiency	114.25 kJ, 52.411 Wh	900 kJ, 90.483 Wh
Usability	Better (compact, portable)	Moderate (complex, bulk system)
Manufacturability	Easier	More complex
Impact	Lower footprint (less energy)	Higher footprint (more energy)
Sustainability	Better	Moderate
Maintainability	Easier	Harder
Classification Accuracy	Moderate	Better
Sorting Time	Moderate	Better

As shown in Table-14, the pre-sorting method uses less power, costs less, and is easier to build and maintain, making it better for the target demographic. While post-sorting gave slightly better accuracy with machine learning and NIR sensors, it was too complex and expensive for this project.

4.5 Conclusion

In conclusion, the proposed pre-sorting system stands as a well-rounded, sustainable and novel solution with strong real-world applicability. This might ensure a smarter waste management system in urban settings like Dhaka and beyond. Future developments could be extending rover's capabilities, including more robust AI algorithms and adapting the system for different environmental situations which aligns with the vision and impact of this project.

CHAPTER 5: COMPLETION OF FINAL DESIGN AND VALIDATION. [CO8]

5.1 Introduction

In this chapter, the final design will be shown which is obtained after facing multiple experimental faults and making design considerations to counteract those faults. Hence after considering all the design considerations and making modifications, the final design will be described in detail.

5.2 Development and Problem-Solving Phase

Overall System's development was divided into multiple phases: Image Processing Model Classification, Mechanical structure and electrical circuit development. This section explains the purpose of each stage along with the problems and design considerations implemented in detail.

5.2.1 Mechanical Structure Development

The mechanical structure formed the foundation of the rover, providing both stability and functionality to support its integrated subsystems. The mechanical aspect required profound balance in both strength and weight so that it can traverse multiple terrains while carrying the loads. For successful mechanical design, proper material selection, structural reinforcements and mounting strategies were devised all while maintaining the expenses. The mechanical aspect primarily consists of the Mechanical Arm and the rover body along with its wheels.

a. Mechanical Arm:

The mechanical arm was programmed to be the core manipulator of the waste segregation rover. Its primary objective was to pick up the classified waste and dump it in the bin that is placed directly behind it. To meet the objective a 6DOF arm chassis was used and since the movement was simple, 4 servos (e.g. DS3218MG) were connected, creating a 4DOF environment for the arm. As for the gripper, a paw shaped structure was used so that it can have the flexibility to pick both elongated and higher surface area objects. While programming the mechanical arm multiple problems were faced and design considerations were chosen accordingly so that it can meet the objective. After overcoming the problems, the final arm structure was completed.

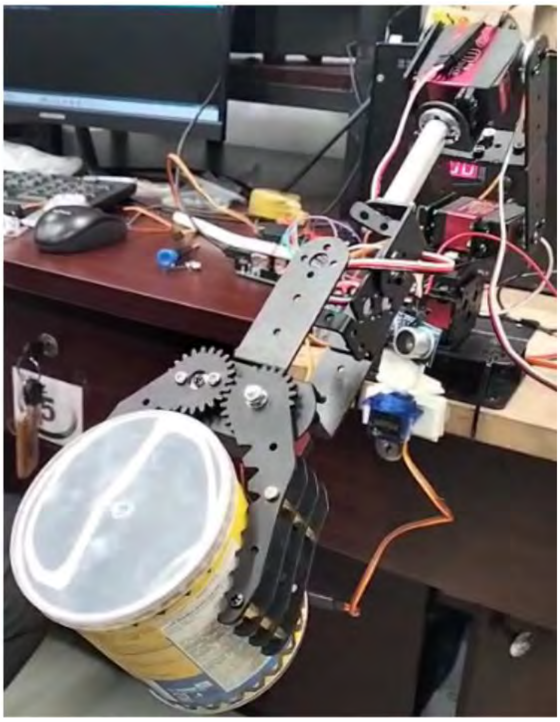


Fig.5.1 Completed Arm Structure

Challenges Faced:

During the development of the arm, various kind of problems were encountered from different engineering domains:

1. **Biasing and Synchronization Issues**: During the early stages of the development phase, it was noticed the motors were not moving in sync with the angles they were inputted. Around 20° offset was noticed. Motors could not work simultaneously due to biasing issues, leading to jerky incoherent movements.
2. **Power Supply Limitations**: It was noticed that the specification and practical usage data was completely different. According to the specification the motors should draw around 0.5A under load condition. However, that was not the case when they were all powered up using a power supply that had the capability to supply up to 24V and 3A, which should've been adequate for the system to work but they didn't. The servos required higher current draw for them to work perfectly coherently.
3. **Jittering and PWM Signal Weakness**: As we know the Arduino Uno has inbuilt PWM signal generating ports, so for starters the motors were receiving their signal from UNO itself. But this caused frequent jittering in the joints. Upon research it was found that this issue occurs due to weak PWM signals.
4. **Gripper Inefficiency**: The initial gripper that we got in the package along with the 6DOF arm was a pinch gripper. The issue with this gripper was that it was unable to grab objects that had bigger or irregular surface area like a 1L water bottle.

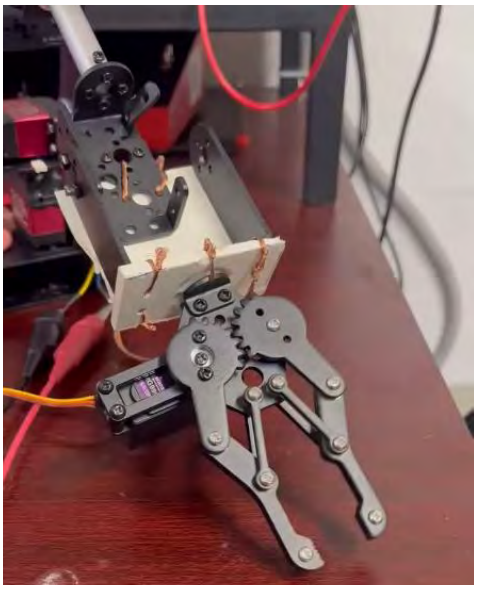


Fig.5.2 Pinch Type inefficient gripper

5. **Torque Mismatch in Theory and Practical**: According to basic torque calculations the motors chosen should be able to handle up to 2kg loads, however during the practical performance it was unable to pick objects rated 500g. This mismatch created a massive setback in the progress.
6. **Hardware Incompatibility**: With the objective of increasing the load handling, a high torque motor (e.g., TD8135MG) was used as a replacement, specifically in the elbow joint. However, this created more problems specifically with the synchronization with the rest of the system.
7. **Sensor Integration Challenges**: A micro servo was used with the ultrasonic sensor placed on top of it. During integration it was noticed that there was major misalignment with the arm motors. This issue led to inaccurate object approach paths.
8. **Higher Current Draw:** During the arm movement phase, it was noticed that when all the motors move in coherence the current draw is massive, which is unsafe for smaller systems. Rapid heating of the motors.

Design Considerations:

Through careful debugging and multiple hardware revisions, the following design considerations were implemented to address the challenges:

1. **Biasing Issue Resolution & Dual Power Supply Strategy** – During the early phase a common issue faced was the biasing issue, this happens due to sudden high current draws. To mitigate this a joystick control was implemented instead of directly keyboard angle inputs. Along with the coding optimization a dual power supply strategy was also used in case of higher current draws. This change allowed smoother, continuous arm movement as well as lower jerks due to no current shortage.
2. **External PWM Controller (PCA9685)**: Upon recognizing the limitations of the onboard PWM signal of UNO, an external PWM driver was integrated specifically designed for

mechanical arms. This greatly reduced jitters ultimately minimizing the erratic behavior of the joints.

3. **Gripper Upgrade** – The gripper servo was upgraded to a DS3218 metal gear servo, which has a maximum angular separation of 86mm, well enough to pick bottles of irregular objects of higher surface area.
4. **Sensor-Arm Synchronization**: The rotating ultrasonic sensor was calibrated to match the base servo’s angle manually. This ensured that object detection aligned with arm positioning, enabling proper waste collection sequences.
5. **Minimizing Higher Current Draw Situations:** The codes were further optimized introducing power delays; this allowed the smooth rotation which resulted in no current spikes. This change helped the motor as it doesn't overheat anymore.

Through these iterative fixes, the mechanical arm evolved from an unstable system to an autonomous manipulator, capable of synchronized motion, sensor-driven actuation, and reliable waste collection under realistic operating conditions.

b. Construction of Wheels

Iteration-1: Off-road wheels 130x60 mm wheels

The initial plan was to utilize the off-road wheels 130x60 mm wheels for general movement. There was no axle for the wheels to turn together. The plan was to use the DC motors to create a difference in forwards and backwards motion for the pair of wheels on each side to allow the rotation to occur. This is called the turning effect. Attached to a three-layer PVC base, the off-road wheels 130x60 mm wheels were tested rigorously using different weights and on grassy and solid terrains.

(a) (b)

Fig.5.0.3 (a) Off-road wheel 130x60 mm (b) Experiment using the off-road wheel 130x60 mm wheels

Findings and Issues faced

1. The off-road wheels 130x60 mm wheels can move forwards and backwards on solid and grassy terrains.

2. On solid ground, the wheels rotate perfectly at 90 degrees for turning right or left. For grassy terrain, the wheels tend to skid, rotating at deviating angles.
3. On grassy terrain, the wheels would also deviate while moving forwards.
4. Depending on the voltage provided, the wheels would rotate at a higher speed.
5. The wheels had a limit of being able to move while carrying weights up to 15kg.
6. The low height of the wheels posed a potential danger to damaging the rover base from all the turning effect during left and right rotation.

Design Consideration

1. The wheels should have a bigger diameter for more height.
2. There should be a shock absorption component to absorb the force from the sudden left and right rotations.
3. There should be a stabilizing component, keeping the wheels in a straight line for better movement.
4. The material of the wheel should be solid enough to withstand the force from the turning effect while being able to carry weights up to 15kg.

Iteration-2: Metal Wheels with bicycle chain

Considering the flaws from the first iteration of wheels, it was decided that the second iteration would be making metal wheels. Each wheel would consist of two big bicycle gears being connected by a metal cylinder welded on the center of either wheel. Two wooden bars were added to the rover which would hold the DC motors running the wheels. The wooden bars would absorb potential shock while providing stability to the rover. The metal design added more weight to the overall rover. On each side of the rover, the wheels would be connected to each other with a bicycle chain to emulate caterpillar wheel style as used in tanks. This design was tested on solid terrain

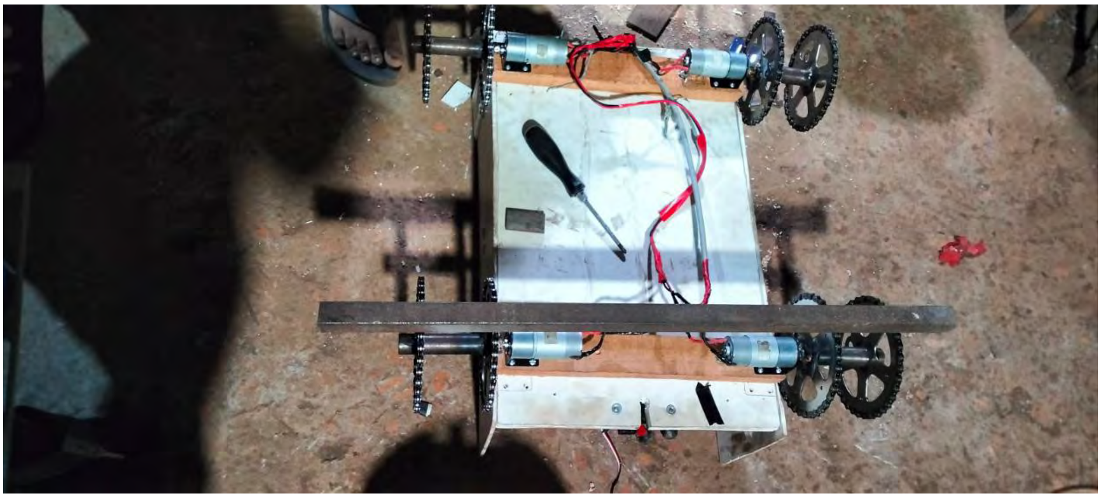


Fig.5.4 Implementation of wooden bar for added stability and new wheels

Findings and Issues faced

The bicycle chains would fall apart upon the first instance of movement, rendering the iteration a failure.

Design Consideration

Use the bicycle chains as tires for the wheels instead of making the caterpillar wheel design.

Iteration-3: Bicycle chain tires for wheels

With the second iteration failing on base runs, the design was changed again. This time, the bicycle chain was used to now cover each wheel separately, instead of two bicycle chains covering a pair. With the new iteration, multiple tests were carried out on solid and grassy terrains along with a football field.

Fig.5.5 Bicycle chain wheel as tire

Findings and Issues faced

1. On solid terrain, the wheels can perform forward and backward movements and can accurately turn 90 degrees right and left.
2. The metal wheels and the wooden bar allowed the rover to move straight without deviating.
3. On grassy terrain, the wheels were incapable of moving from their starting point before moving after the turning effect was applied.
4. Despite the turning effect being applied, the rover would not rotate at all.
5. On the football field, it would move forwards and backwards perfectly but would struggle to rotate– it would rotate on the more solid parts of the field while getting stuck on the uneven surfaces and muddy parts.

Design Considerations

1. The bicycle chain covered wheels did not have much traction on the grassy terrain. A material with more traction will be required for better surface traction.

2. A support set of wheels could be required for better movement in the grassy terrain and football field.

Iteration-4: Flywheels and pipes

Addressing the failure of the third iteration took some time. The tubes inside tyres were suggested as a potential material to cover the bicycle-chain layered wheels for better traction. Only one of the wheels was covered before the proposal was rejected due to the ease of tears. The next design that was proposed was to use gas cylinder pipes. The material was rough enough to have a firm grip on the ground and flexible enough to be wrapped around the bicycle-chain layered wheels. To help with the movement four castor wheels (flywheels) were attached to the bottom of the rover. This design was tested on solid terrain and with a football field. The grassy terrain was excluded as there was too much grass, deviating from the target demographic of this project.

Fig.5.6 Gas cylinder pipe being wrapped around the wheel

Fig.5.7 Castor wheels (Flywheels)

Findings and Issues faced

1. On solid terrain, it performed the same as previous iterations– it would move forwards, backwards and be able to rotate.
2. On grassy terrain, it would not be able to rotate at all.
3. The castor wheels helped in some instances while clashing with the wheels partially.
4. On the football field, it had the same problems as before– getting stuck on the grassier parts and getting stuck in mud.

Design Consideration

1. As the traction was not what was preventing the wheels from moving, it was concluded that there was not enough surface area on the wheels for the turning effect to take place properly.
2. For better surface area, it would be imperative to cover the two metal gears of the with a light material so that it does not put extensive load on the DC motors.

Iteration-5: Shielded wheels

After much consideration, the bicycle chains were taken apart from the wheels. A light metal alloy sheet was used to cover each wheel– resembling a car wheel– shielding the top spikes of the bicycle gears. The added sheet did not increase the weight by much. This iteration was tested on the solid terrain as well as the football terrain.

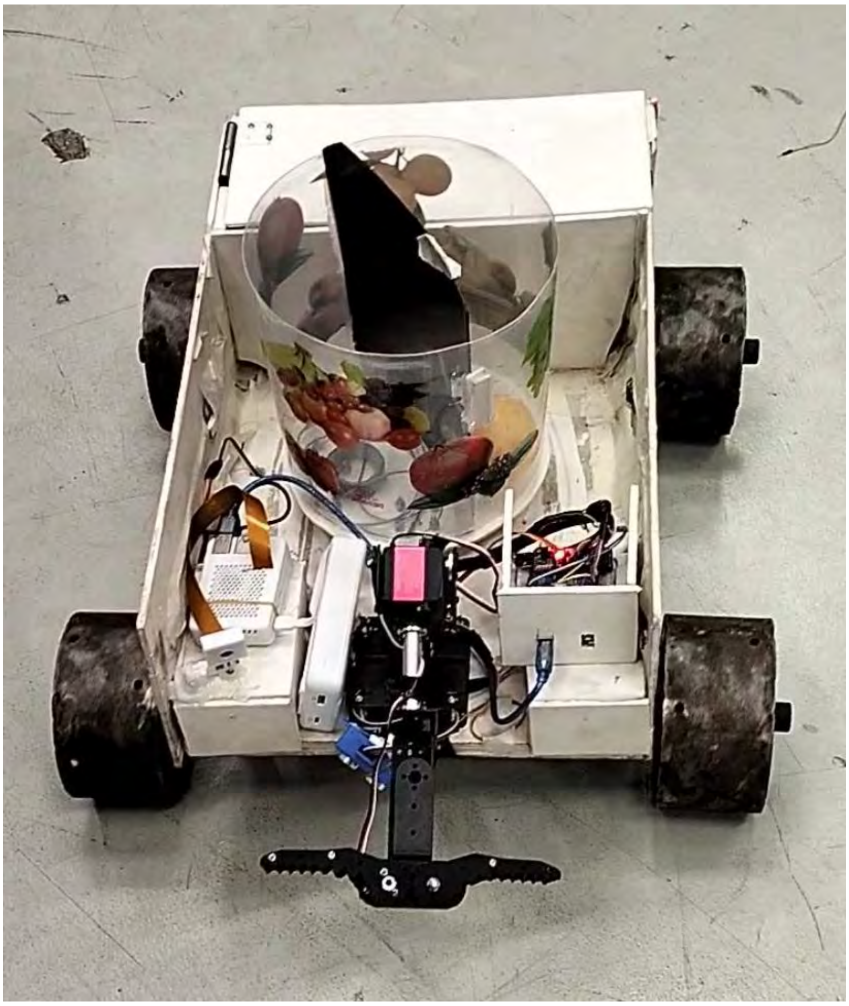


Fig.5.8 Shielded Wheels

Findings

1. On solid ground, the rover was able to move forwards and backwards along with rotating left and right.
2. On the football terrain, the wheels were able to move forwards and backwards along with rotating left and right in muddy and uneven conditions.

c. Rover Chassis Construction

To construct the chassis of the rover, we bought a 5mm PVC sheet and cut it to 55cm x 35cm for the base of the rover and used 4 layers for extra rigidity and stability. We went with the PVC sheet instead of an aluminum sheet because it's cheaper and easier to prototype with.

Challenges faced:

1. Mounting the 37gb motor brackets directly to the merged 4 layers of PVC caused it to bend during initial testing on grassy fields. See Figure –
2. The brackets could not hold the updated wheels' weight even though they were mounted on the wooden bar as during operation, they were prone to bending sideways when the rover was turning left or right. See Figure –

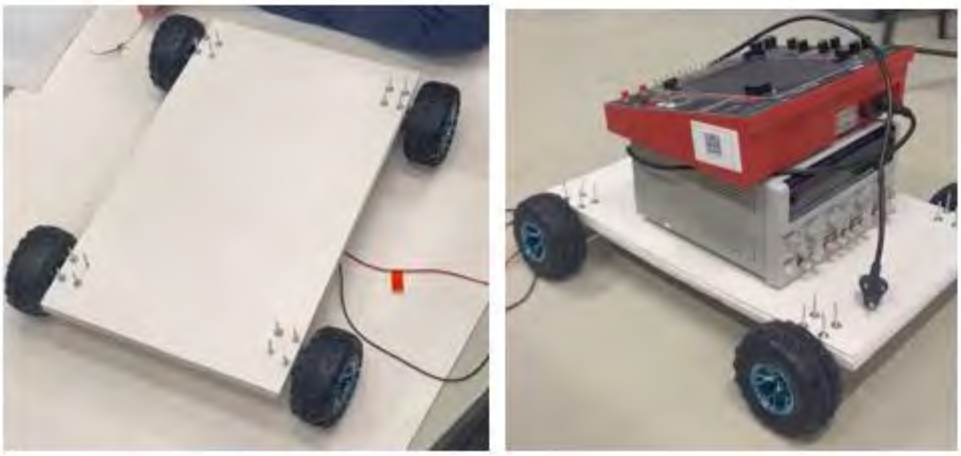
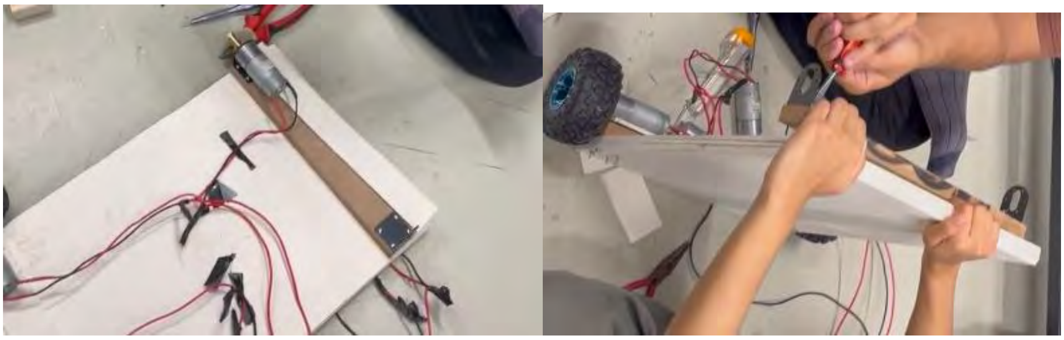


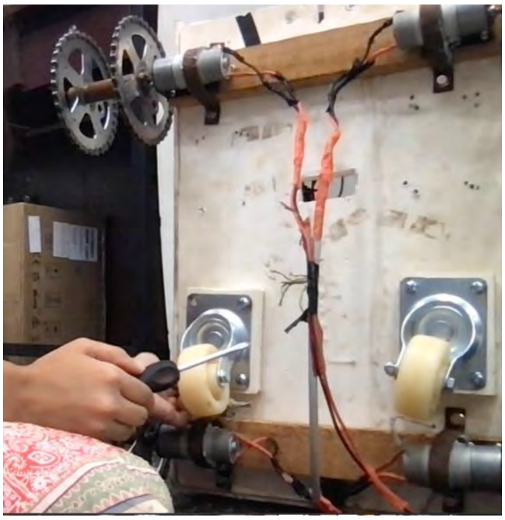
Fig.5.9 4 layers of PVC

Design Considerations:

1. To counteract the issues faced in Problem 1, we attached 2-inch-thick wooden bars at the bottom of the rover chassis where the 37gb motor brackets were attached previously. Now the brackets were attached to the wooden bars which were held to the PVC frame through screws. This provided the rigidity and structural integrity that was missing and prevented the rover from flexing. See Figure –
2. To support the brackets and prevent them from bending due to the rotational force of the wheels, clamps were custom molded out of 5mm iron bars and placed on the motors close to the brackets. This resulted in more structural integrity as now the weight of the motors and the rotation force due to turns were shared by both the brackets and the clamps. Additionally, to further reduce the load on the brackets, freewheel See Figure –



(a)



(b)

Fig.5.10 (a) Wooden Bar attached (b)Installing freewheels

5.2.2 Image Processing Model Classification

In this project, one of the primary requirements is the ability to detect and classify the waste into biodegradable and non-biodegradable. To do this we trained and evaluated three models: YOLOv5n, YOLOv8n and YOLOv11n out of which YOLOv8n was decided to be the best performing model for our practical scenario from **Chapter 4**. The model was saved as 'best.pt' file which was deployed on the Raspberry Pi 5. To give "vision" to the rover, we used the Pi Camera V2. When the 'best.pt' file was deployed, we were getting around 2~3 FPS as shown in Figure 63.

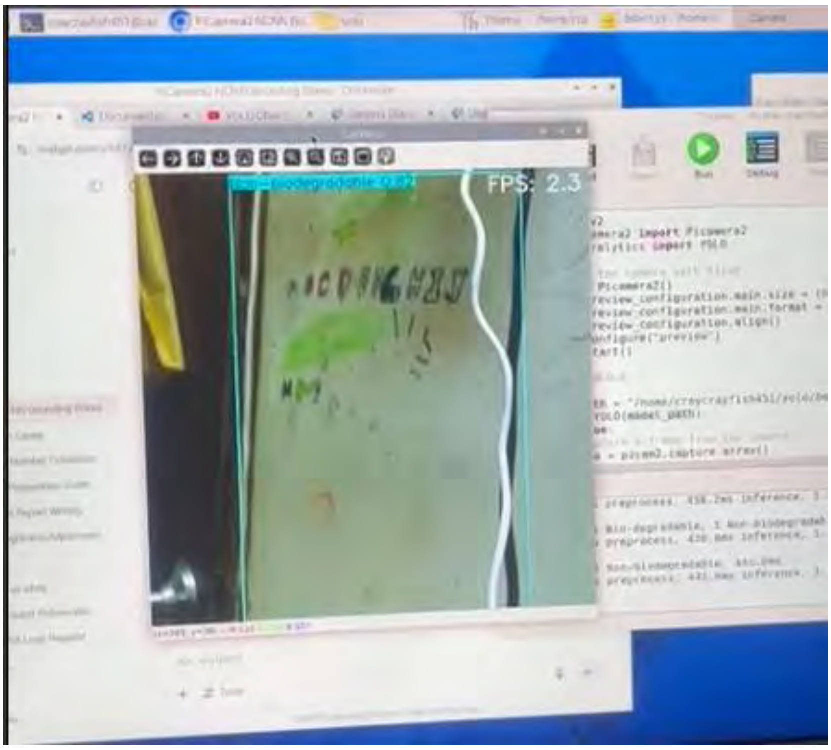


Fig.5.11 Classification of non-biodegradable

Since our rover will not be stationary and will be moving, we needed more FPS. Upon research we found that the YOLO models which are stored as .pt files can be converted to different formats which results in more FPS since they are optimized to work with ARM processors like those of PI5. We chose the NCNN format and upon converting it to this format we got a 10~11 FPS boost as can be seen in Figure –.

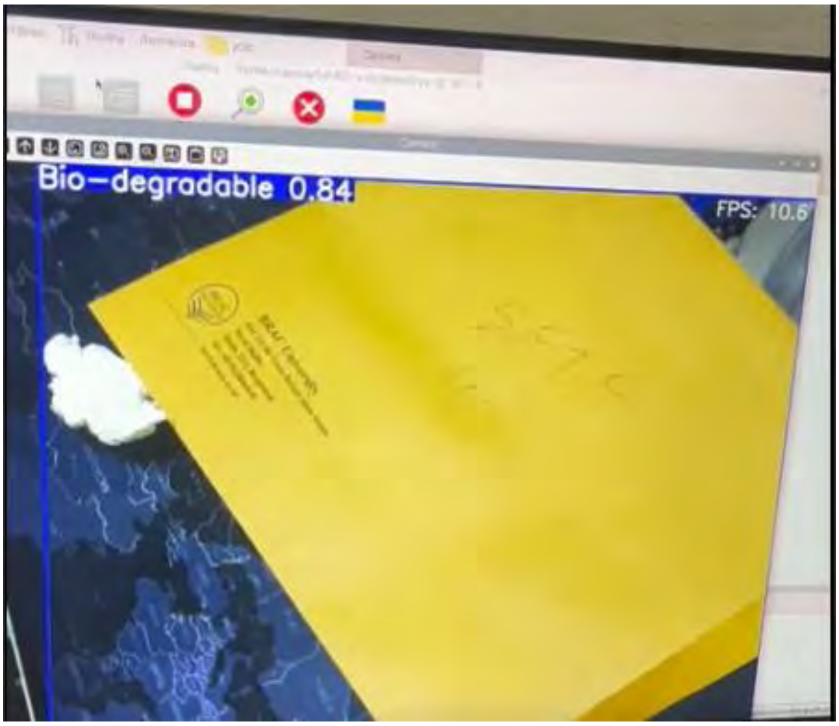


Fig.5.12 Classification of Biodegradable

ii) Electrical System Development

Electrical consideration was necessary, since the proposed system requires an around 15 A of stable current for all the subsystems to work in cohesion. The electrical system ensures that the system can provide stable voltage and current to each of the subsystems while ensuring operational safety.

a. Protection Circuit

The system needs high stable current for which a LiPo battery is used. While they are very effective, they are also known for their hazardous nature. To ensure operational safety, a dedicated protection circuit XH-M609 or HCW-M635 DC 6V-70V Battery Low Voltage Cut-Off and Over Discharge Protection Module was chosen and integrated. This circuit uses a relay to which acts as a switch when the voltage falls below the set voltage value which we set to 14.8 V. This ensured that the battery stayed within the operating voltage range and also prevented any possible fire hazards. Figure __ shows the protection circuit

Fig.5.13 Over Discharge Protection Circuit

b. System Voltage provider.

The system consists of five subsystems. Each of the subsystems has a different voltage and current requirement. For example: the robotic arm was pulling around 7V and 5A stable voltage and current where the 37gb gear motors required around 14V @ 4A. To ensure each of these subsystems get their desired voltage and current, Voltage regulator Buck Converters were installed that have the capability to provide upto 10A of current. This installation ensured that the entire system would be able to pull their required voltage and current as each buck converter could be set to any voltage value upto 40V. Figure __ shows the Voltage regulator used

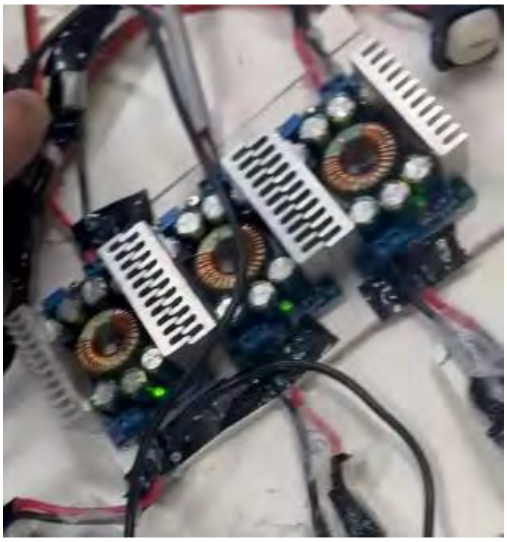


Fig.5.14 Voltage Regulator Buck Converter

5.2 Completion of the Final Design

The system consists of one microprocessor (Raspberry Pi5) and two microcontrollers; ESP32 and Arduino UNO where the ESP32 controls the navigation and gear motors of the rover and the Arduino UNO controls the robotic arm. Here the Raspberry PI5 is the master, and the microcontrollers are the slaves.

5.2.1 Controlling the Robotic Arm using Arduino UNO

As mentioned above, the Arduino Uno controls the robotic arm through the PCA9685, handling five key joints: the base, shoulder, elbow, gripper, and a sensor-mounted servo. When the UNO receives power, it initializes communication with the PCA9685 PWM driver and configures the ultrasonic sensor pins. After this process is complete, it gradually moves all the servos to their predefined initial positions. Once this homing procedure is complete, the system enters an idle state, where it waits for a command from the Raspberry Pi.

The system follows a 5-step process of waste scanning, picking up the waste, throwing a sequence and then returning to initial position. This sequence starts when it receives the "ARM_START" command from the Raspberry PI5. The arm enters the scanning state where an ultrasonic sensor sweeps across a predefined angle range in small steps and at each step the distance is measured. If an object is detected within 17 cm, then the system records the angle the ultrasonic detects the waste at and then sends a pause command to the ESP32 which pauses the rover's motors and starts the pickup sequence.

The base angle is then calculated from the detected sensor angle using a simple mapping function and the arm executes a series of coordinated joint movements: first aligning the base, then lowering the elbow and shoulder. The gripper opens, closes firmly around the object, and the arm holds it for a fixed duration.

After the holding stage, the throwing sequence begins. The arm moves to a predefined throw posture by rotating the base, shoulder, and elbow to their throw angles, then opens the gripper to release the object. Following the throw, the arm transitions to the return stage, where all joints are moved back step by step to their initial angles, and the sensor servo is reset to the center. At this point the system informs the Raspberry Pi that the task is complete and signals the motors to resume. The arm then returns to the idle state, ready for the next command.

Throughout this process the Arduino also listens for special commands such as “STOP,” which halts the sequence immediately and returns the arm to its initial position, or “ESP_COMPLETE,” which shuts the system down. Additional commands allow the Raspberry Pi to control auxiliary servos connected to the driver. Debugging messages are regularly printed to the serial monitor, showing the current state and system status.

This design ensures that the arm operates reliably in both manual and automated modes, with clear transitions between states, safe recovery from interruptions, and smooth servo motion for pickup and throwing actions.

5.2.2 Controlling the Rover using ESP32

When powered on, the ESP controller initializes its communication interfaces, sets up the motor driver pins, and stops the motors to ensure a safe start. At this stage, the program also calculates how many traversal lanes are required to cover the given field. This is determined by dividing the total field width by the rover’s coverage width. With this calculation, the rover knows exactly how many lengthwise passes are needed to fully cover the area.

The system then waits for commands from the central controller. A **start command** triggers the automatic lawnmower sequence, a **pause command** temporarily halts motor activity, a

resume command allows the rover to continue where it left off, and a **stop command** completely ends the pattern, returning the system to idle until restarted. Additionally, a Bluetooth interface allows direct manual control (forward, backward, left, right, or stop), although this is disabled if the system has been fully stopped.

Automated traversal is organized as a state machine. The rover cycles through states of moving forward along the field length, performing a first turn at the end of the lane, shifting sideways into the next lane, and making a second turn to align for the next lengthwise pass. The direction alternates after each lane, creating a zigzag or lawnmower trajectory that ensures full coverage. Each forward pass is timed to match the actual field length, and each turn or lateral shift is timed to match the rover's turning radius and lane spacing. The number of completed lanes is tracked, and when all have been covered the rover declares the pattern complete, sends a completion signal, and returns to idle.

An additional layer of coordination is built in with the waste-handling arm controlled by the Arduino Uno. The rover is equipped with an ultrasonic sensor that continuously scans for waste objects. If an object is detected within the pickup range, the ESP immediately pauses its motion and sends a pause signal to ensure the rover stops. At the same time, a command is sent to the Uno to begin the robotic arm sequence, which positions, picks up, and throws the detected waste. During this process, the rover remains stationary. Once the arm signals completion, the ESP resumes from its last recorded state and continues the lawnmower traversal without restarting the pattern.

5.2.3 Running YOLO Model and Controlling Microcontrollers using Raspberry Pi5

The system processes camera frames with a resolution of 720×720 pixels using an object detection model to identify waste items. Each detected object is filtered by spatial and confidence rules where it must lie at least 20 pixels inside the frame boundary and occupy less than 20 percent of the total frame area. It should also have a confidence score of at least 0.40. Detections that pass these criteria are added to a first-in, first-out (FIFO) queue so that objects are processed in the order they appear. To prevent duplicates, each new detection is compared with objects already in the queue using the intersection-over-union (IoU) metric. IoU is defined as the area of overlap between two bounding boxes divided by the area of their union. If the IoU with an existing object is greater than 0.5, the detection is considered the same object and is not added to the queue. At any given time, only the object at the front of the queue is active. The system checks in each new frame whether this active object is still visible. If it is present for two consecutive frames, it is considered stable, and a single actuation command is sent to rotate the servo and trigger the pickup arm. Biodegradable items are sorted by rotating the servo to 180 degrees, while non-biodegradable items are sorted by rotating it to 0 degrees. Each angle command is sent only once per object to avoid repetition. If the active object is missing for three consecutive frames, it is removed from the queue, and the next object becomes active. To reduce errors from brief occlusions or flickering detections, recently seen objects are stored in a short memory for up to five frames. Each memory entry has an age counter that resets when the object is seen again and increments when it is not. An entry is discarded if its age exceeds five frames.

Beyond detection and queueing, the system also coordinates with two external microcontrollers through serial communication: an Arduino UNO and an ESP device. The Arduino UNO primarily controls the servo motor attached to an ultrasonic sensor and robotic arm. When a stable object is identified, the Raspberry Pi sends a combined command such as SERVO_ARM_180 or SERVO_ARM_0 to the Arduino UNO over USB serial. This single command simultaneously rotates the bin servo to the correct position and initiates the arm pickup sequence once the ultrasonic sensor detects waste. The Arduino also provides feedback signals back to the Pi, such as ARM_COMPLETE when a pickup is finished, and PAUSE_MOTORS or RESUME_MOTORS when movement must be halted or resumed. These responses are read by a dedicated background thread on the Pi, which can then forward pause or resume commands to the ESP.

The ESP device, in contrast, is responsible for executing the lawnmower-style navigation pattern that moves the rover across its operating area. The Pi sends a START command at initialization, followed by possible PAUSE and RESUME commands when the Arduino requests interruptions for waste collection. When the ESP finishes its coverage path, it returns an ESP_COMPLETE message to the Pi, which is immediately relayed to the Arduino UNO so that the system remains synchronized. Separate background threads continuously listen to both serial channels, ensuring that the Pi can respond to messages from either device in real time.

In summary, the division of responsibilities ensures modular operation: the Raspberry Pi handles perception, detection, and decision logic; the Arduino UNO handles local actuation of servo and arm mechanisms; and the ESP executes mobility and navigation. The Pi acts as the coordinator between these devices, enforcing reliable communication through serial commands and responses. This distributed design allows the system to sort detected waste in sequence, pause and resume mobility when required, and maintain overall synchronization between perception, movement, and actuation.

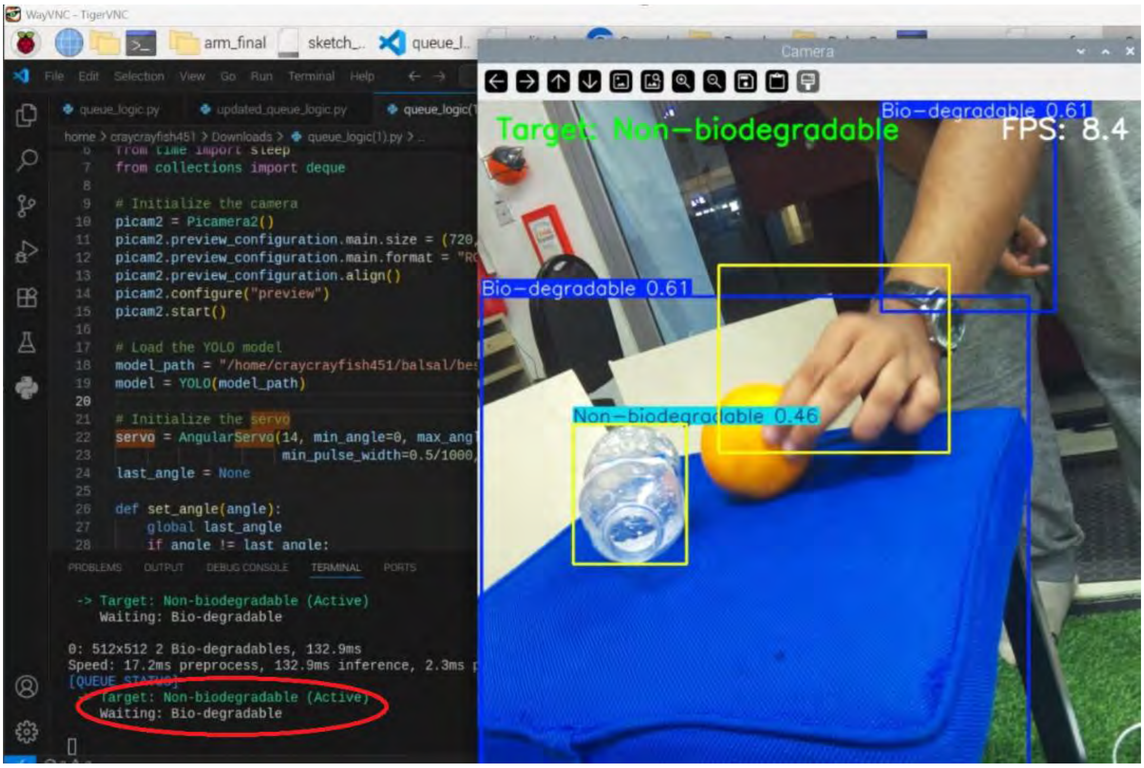


Fig.5.15 Queue logic

5.2.4 Snippets of the completed system and its running phases.

- **Completed System**

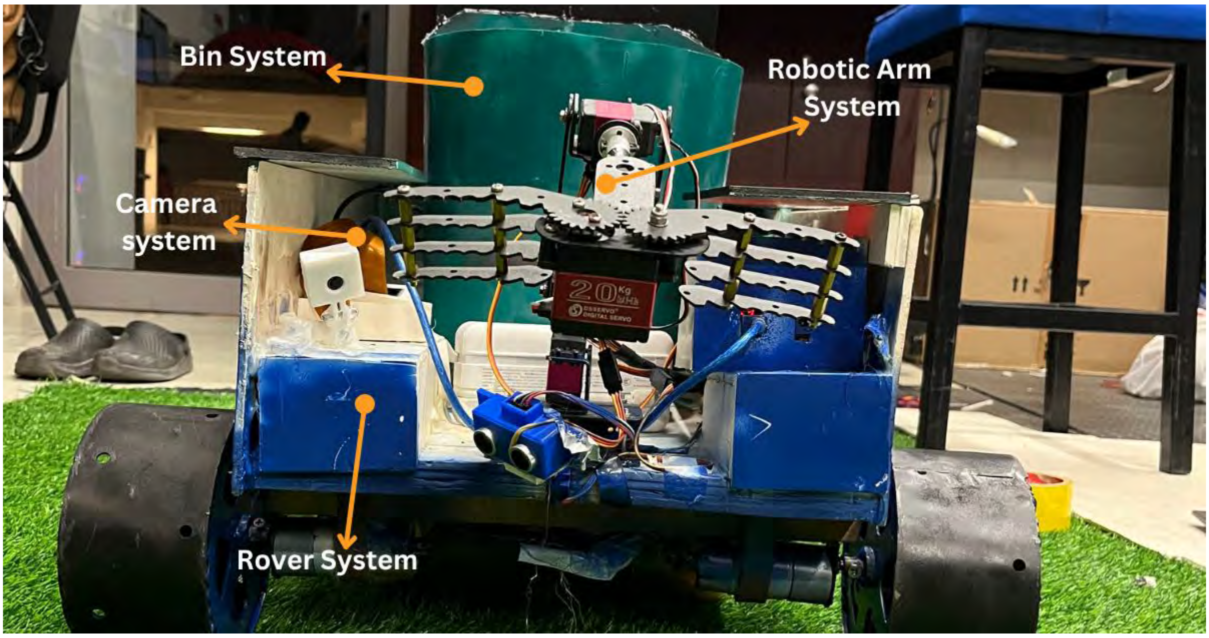


Fig.5.16 Completed System showcasing 4 subsystems.

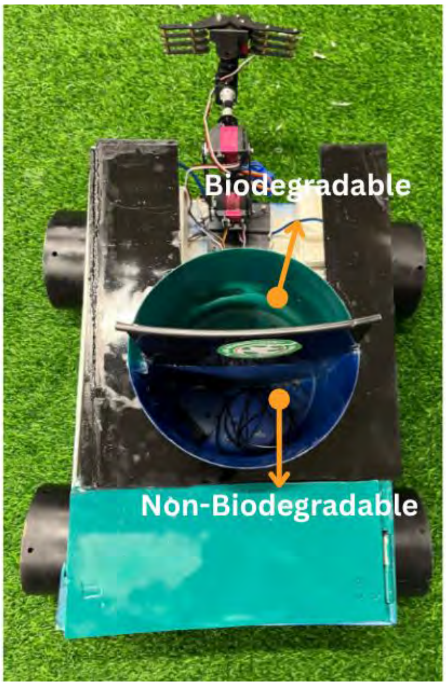


Fig.5.17 Overview of the Completed System showing the compartments of bin.

- **Running Phases**

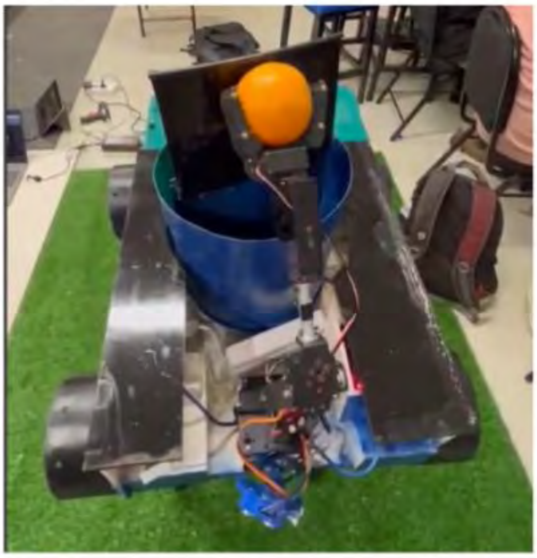


Fig.5.18 Picking of biodegradable waste and dumping it in the respective bin

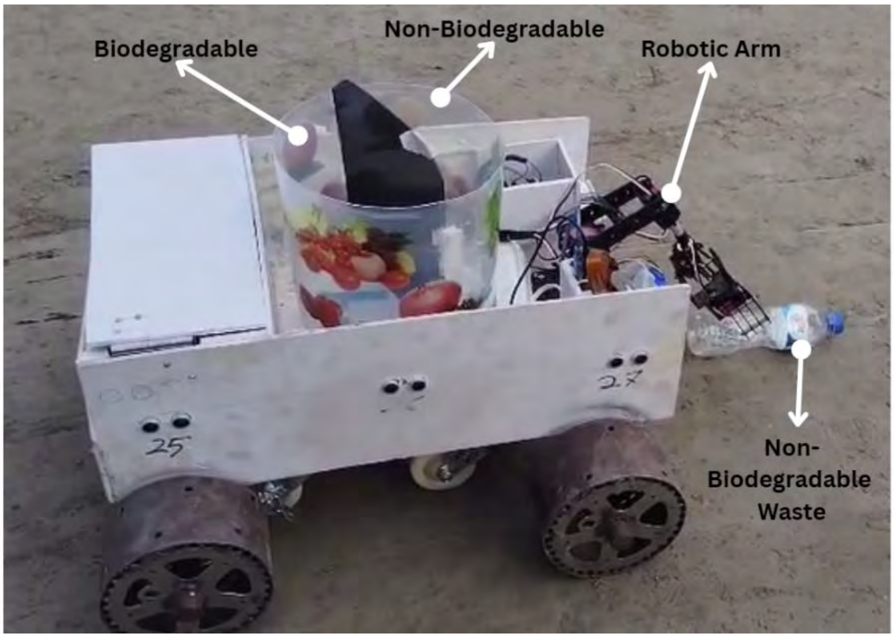


Fig.5.19 Picking of non-biodegradable waste and dumping it in the respective bin

5.3 Evaluate the solution to meet desired need

Upon completion of the system, valuable data were extracted. The primary objective was for the initial specifications to be met. The data received include:

- **Battery Capacity:** Evaluation consists of checking for the battery's capacity during full system operation and its charging rate. From **Fig. 5. 20.**, it takes about 50 minutes for the battery to completely discharge. This exceeds the initial claim of the system to be able to work for a full 30 minutes. The battery takes about 30 minutes to fully charge; this ensures that the system can run at a 30-minute delay.

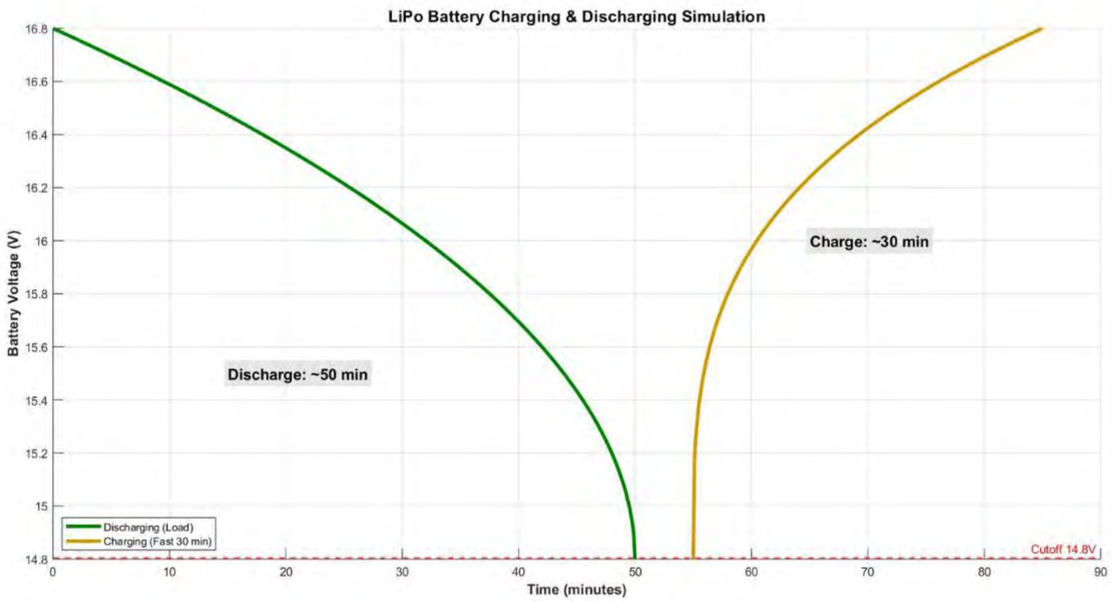


Fig.5.20 Charging and Discharging capacity of the battery

- After testing, it was observed that the accuracy of the ML model is about 84%. This proves that most of the waste will be segregated perfectly. It was calculated by taking 15 Biodegradable and 10 Non-Biodegradable and testing how many of them were accurately classified.

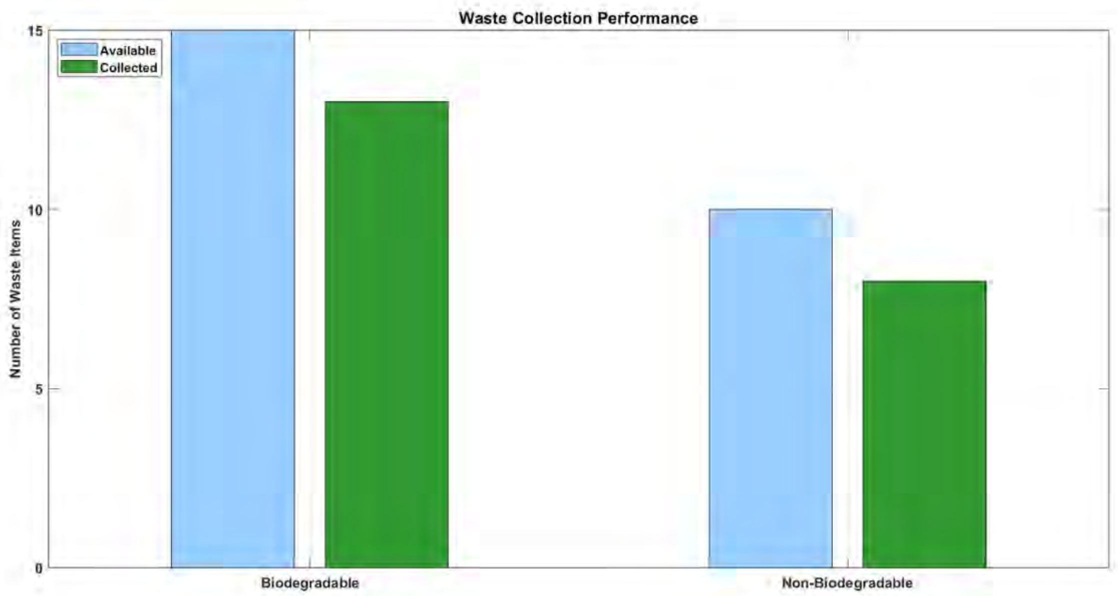


Fig.5.21 Performance of the waste collected

- Rover speed has been scaled down to **50%** of the proposed minimum speed for the Pi camera to detect waste. Through the testing phase it was noticed that at high speed the camera was unable to classify the waste. For the system to work, the speed had to be downscaled compared to what was given in the initial specification.
- The **YOLOv8n** has been formatted to **NCNN** to increase FPS by **10 FPS.** This is crucial since the system is constantly moving and sending live feed to the Raspberry Pi. If the FPS is low, then the classification time will increase rendering the entire system unable to fulfill its purpose.
- The arm has a pick rate of **23.8 seconds/waste. This** pick rate means the time taken for the arm to initiate the picking process until the waste is dropped into its dedicated slot in the bin.
- A queue logic has been implemented successfully to choose and pick both types of waste one after another. This is very crucial since most of the waste in urban environments will be in a cluttered state. So, for the system to be able to detect, classify and segregate the waste, hence the queue logic is implemented.
- The rover can traverse through multiple terrain. This is ensured after going through multiple iterations of wheels and reaching a custom wheel that has both the height and the surface area needed to traverse multiple terrain.
- Throughout the testing period the average power consumption done by the entire system was monitored.

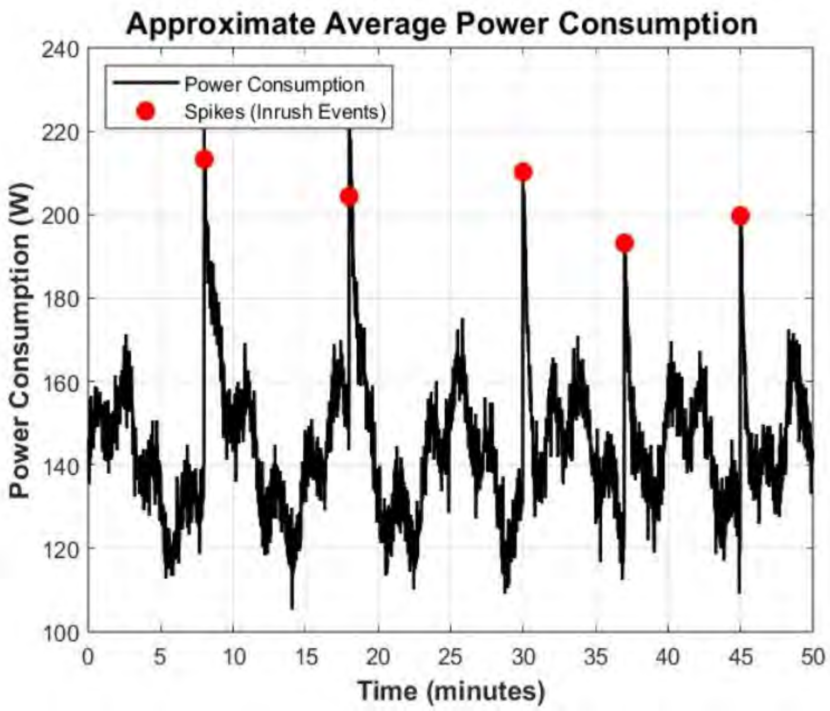


Fig.5.22 Average power consumption in 50-minute runtime.

From the graph, there are sudden spikes that were recorded, upon thorough monitoring it was deduced that the spikes occur when the entire system works together, which is when the arm initiates its picking process. The sudden on-load process causes inrush current which causes spikes in the power consumption.

5.4 Conclusion

Our goal was to develop an autonomous rover that detects waste, classifies it as either biodegradable or non-biodegradable, and sorts it with the help of a robotic arm and an onboard image processing object detection model. During the execution of this goal, we encountered multiple problems that required several design considerations. This led to a final design that we tested and evaluated. The individual subsystems such as the robotic arm subsystem that picked up the waste, the camera system that detected and classified waste, and the navigation system that controlled the rover's lawnmower pattern were tested independently. After all subsystems passed their individual tests, the entire system was tested as a whole to ensure it produced the desired outcomes and met the specified requirements.

CHAPTER 6: IMPACT ANALYSIS AND PROJECT SUSTAINABILITY [CO3, CO4]

6.1 Introduction

The proposed project "Autonomous Waste Segregation and Collection Rover" is evaluated in terms of societal, health, safety, legal and cultural impacts beyond its technical designs. The system addresses relevant urban challenges by improving cleanliness in public spaces, ensuring occupational safety and encouraging more employment opportunities in the waste management sector. This chapter represents a detailed analysis of the impact and sustainability of the proposed project.

This chapter also evaluates the project's sustainability through SWOT analysis, sustainability matrices and its alignment with the United Nations' Sustainable Development Goals (SDGs). Through sustainability matrices, environmental, economic and social dimensions are explored to ensure that the project is viable for a long time. Ultimately, this chapter shows this rover is beyond an innovative project; it is a sustainable solution which contributes to clean, healthy and sustainable urban areas.

6.2 Assessing the impact of solution

Societal Impact: The automated waste management system can significantly transform public spaces, especially in urban centers like Dhaka, by making parks and recreational areas cleaner, more usable, and visually appealing. This directly enhances urban living standards while encouraging family and community engagement. It also fosters a sense of community pride and responsibility for maintaining public spaces.

The shift away from manual scavenging addresses critical socio-economic challenges, reducing occupational health risks and opening new avenues for technology-based roles. Jobs in monitoring, operations, and maintenance provide opportunities for skill development and align with the *Digital Bangladesh* initiative, which aims to modernize the country's workforce. By reducing the stigma around waste management and creating dignified employment opportunities, the system also promotes social equity and improves the overall perception of the waste management sector.

Health Impact: The automated waste system addresses critical health challenges in densely populated cities like Dhaka, where unmanaged waste often becomes a breeding ground for disease vectors such as mosquitoes. This is particularly relevant given the increasing prevalence of diseases like dengue, exacerbated by inadequate urban sanitation [28]. By minimizing waste accumulation in public spaces, the system not only prevents pest habitats but also reduces human exposure to hazardous materials.

Additionally, the system enhances occupational health by reducing direct contact between workers and unsorted waste, which often contains sharp, infectious, or toxic substances. This alignment with global health standards underscores the project's potential to significantly improve public and occupational health outcomes in Bangladesh, contributing to healthier urban living.

Safety Impact: The system prioritizes both occupational and public safety through its advanced segregation and navigation features. For hazardous materials, precise classification and deposition reduces the risks associated with manual handling, significantly lowering exposure to harmful substances. This minimizes the potential for disease transmission and accidents during waste processing.

Public safety is ensured by incorporating detection mechanisms, such as advanced cameras and controlled movement speeds, to prevent accidents in crowded areas. By adhering to global safety norms for autonomous systems, the rover ensures both operational efficiency and a safe experience for park visitors, children, and other pedestrians.

Legal Impact: The system complies with Bangladesh’s legal framework, including Sections 268 and 269 of the Penal Code [29], by preventing public health hazards linked to improper waste disposal. It also adheres to the *Solid Waste Management Rules 2021* [30], promoting efficient waste segregation and handling to reduce environmental pollution. Furthermore, by minimizing direct human contact with hazardous materials, it aligns with the *Bangladesh Labour Act of 2006* [31], ensuring safer working conditions for waste management workers.
By setting a precedent for automated solutions in public services, this project could drive the establishment of regulations and policies for the use of autonomous systems in public and industrial spaces. This legal alignment and future potential underline the system's role in advancing Bangladesh's regulatory landscape while safeguarding public health, worker safety, and environmental standards.

Cultural Impact: Culturally, the system has the potential to challenge deeply rooted stigmas surrounding waste management, particularly manual scavenging. By introducing automation, it creates a shift in societal perceptions, offering cleaner, more efficient waste management solutions while providing dignified, technology-focused employment opportunities. This cultural shift aligns with modern aspirations for cleaner, smarter cities, fostering environmental responsibility and reducing littering habits.
In addition, the rover’s visibility in parks and public spaces can serve as an educational tool, encouraging citizens to adopt better waste disposal practices. This behavioral change could gradually lead to a culture of cleanliness and sustainability, addressing both the practical and cultural barriers to effective waste management in Bangladesh.

6.3 Evaluate the sustainability

Sustainability of a project refers to an approach which is considered in various aspects like environmental, social, economic to understand how long the project would sustain or meet the current needs of the stakeholders. All this needs to be done without compromising future generations. To measure the sustainability of a project, SWOT analysis is one of the first things to do which stands for Strengths, Weaknesses, Opportunities and Threats. The SWOT analysis of this project is shown below.

Fig.6.1 SWOT Analysis for Waste Segregation and Collection Rover Project

The sustainability matrix provides a comparative assessment of the proposed solution across Environmental, Economic and Social Criteria to evaluate their overall performance in achieving sustainability goals. The visual representation of Sustainability Matrix Distribution can be found in **Fig.6.1.**

FYDP_SAM

Criteria	Weight	Pre Sorting Approach (Score)	Pre-Sorting Approach (Weighted)	Post-Sorting Approach (Score)	Post-Sorting Approach(Weighted)
Environment Criteria	0.4				
Carbon Footprint	0.05	4	0.2	4	0.2
Energy Efficiency	0.05	5	0.25	2	0.1
Material Sourcing (High pressure systems)	0.05	3	0.15	3	0.15
E-Waste Management	0.05	1	0.05	1	0.05
Toxic Substance Use	0.05	5	0.25	5	0.25
Renewable Energy Source	0.05	1	0.05	1	0.05
Water Usage	0.05	5	0.25	5	0.25
Economic Criteria	0.3				
Initial Capial Investment	0.05	5	0.25	1	0.05
Operational Costs	0.05	4	0.2	1	0.05
Maintenance Costs	0.05	3	0.15	1	0.05
Lifespan	0.05	4	0.2	5	0.25
Productivity Impact	0.05	2	0.1	2	0.1
Market Potential	0.05	4	0.2	4	0.2
Job Creation	0.05	3	0.15	2	0.1
Local Economic Impact	0.05	3	0.15	4	0.2
Social Criteria	0.3				
Health & Safety	0.05	5	0.25	5	0.25
Community Engagement	0.05	3	0.15	3	0.15
Social Equity	0.05	4	0.2	4	0.2
Educational Impact (Training)	0.05	4	0.2	4	0.2
Quality of Life	0.05	4	0.2	4	0.2
Access to Services	0.05	4	0.2	4	0.2
Cultural Impact	0.05	5	0.25	5	0.25
Total Score			4.05		3.5

Fig.6.2 Sustainability Matrix Distribution Table

As seen in the figure above, the proposed solution is an efficient solution in terms of Environmental, Economic and Social context. The compactness of the system allows it to be more energy efficient and due to it being environmentally friendly the carbon footprint emission is also minimum. This early-phase intervention ensures lower operational costs and a better productivity rate. Additionally, lower initial Capital Investment further supports its economic viability by requiring less infrastructure and processing complexity.

After all these sustainability analyses, the project had to be evaluated if it aligned with any Sustainable Development Goals (SDG). The automated waste segregation and collection project aligns with several Sustainable Development Goals (SDGs), contributing to both environmental and social advancements in urban settings.

SDG 3: Good Health and Well-Being: The proposed system mitigates public health risks which occur due to unmanaged waste. The waste tends to serve as a breeding ground for various pathogens and disease carrying vectors which cause diseases such as dengue which is currently occurring at an alarming rate in Bangladesh. The automated system reduces the amount of exposure faced by manual laborers, thus promoting a healthier working environment.

SDG 11: Sustainable Cities and Communities: The city of Dhaka is in a precarious situation with waste being littered all over. The proposed system helps in cleaning the city which fosters a cleaner and healthier environment, especially for public places like parks and other recreational facilities. This in turn promotes tourism and improves the quality of life for people living in the densely populated city.

6.4 Conclusion

This chapter explores the important impact and sustainability analysis of the design project. After evaluation it can be deduced that there are some positive impacts of the project through which urban waste management can be improved. The proposed solution is cost and energy effective compared to the traditional method. As a result, this project can create a lasting positive impact. Finally, a SWOT analysis has been done to identify the strengths, weaknesses, opportunities, and threats of the proposed project in the current market.

CHAPTER 7: ENGINEERING PROJECT MANAGEMENT. [CO11, CO14]

7.1 Introduction

Engineering project management determines most of the success behind a project. It involves a structured process of planning, managing and executing engineering projects so that the project fulfills the objective, requirements and specification within budget and on time. It is a combination of management skills with technical knowledge. Project management typically has project goals, timeline, allocation of tasks and managing risks. Every successful project depends on the project management skills of the team.

In this chapter, the Final Year Design Project's yearlong planning, risk analysis and project progress will be explored. This yearlong project has been divided into three parts spanning over three semesters. The first semester includes FYDP-P where problem identification and project proposal are done. Then, in the second semester, FYDP-D, based on the problem and its proposed multiple solutions, the multiple design approaches were designed. Based on the design the optimum solution was chosen. Finally, during the last semester of FYDP, known as FYDP-C, the project was completed adhering to the objective, specification and requirements.

7.2 Define, plan and manage engineering project

7.2.1 Project Timeline

Every project starts with a plan. Planning includes a checklist of the tasks, a budget and a timeline. A detailed plan ensures successful execution of a project. The expected timeline for the project is one year, which will be done over a period of three semesters. The tentative task timeline can be expressed through the Gantt chart below:

TABLE 7.1: FYDP-P TIMELINE

Tasks	**Weeks**											
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***
Topic Selection												
Literature Review												
Survey												
Designing Multiple Approaches												
Draft Note Preparation												
Setting Tentative Budget												
Risk Assessment												
Ethics and Sustainability												
Project Proposal Report and Presentation												

In FYDP (P) the first task was topic selection which we completed in the first two weeks. After that, a detailed literature review was carried out from week 2 to week 5 to gather background knowledge. Alongside this, a survey was conducted starting in week 3 and continued up to week 8 to collect relevant data. From week 4 to week 6, we designed multiple approaches to solve the problem, and in the same period, we also prepared our draft note. A tentative budget was set between week 5 and week 7. Risk assessment was performed between week 6 and week 9, followed by ethics and sustainability considerations from week 7 to week 11. Finally, in week 11 and 12, we prepared the project proposal report and presentation to conclude the planning phase.

TABLE 4: FYDP-D TIMELINE

Tasks	**Weeks**													
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***
Finalize Design														
Design 1 Simulation														
Design 2 Simulation														
Analysis and Compare														
Selecting the Best Approach														
Final Report and Presentation														

In FYDP (D) the most important task was to finalize the design. This was initiated in week 1 and completed by week 2. Starting from week 2 up to week 8, we worked on the simulation of Design 1, while in parallel, from week 3 to week 9, the simulation of Design 2 was also carried out. The results of both simulations were then analyzed and compared during week 8 to week 11. After careful evaluation, we selected the best approach in weeks 11 and 12. The final report writing and presentation were prepared in week 13 and week 14.

TABLE 5: FYDP-C TIMELINE

Tasks	Weeks													
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*
Resource Sourcing														
Construction of Subsystem														
Testing and Debugging														
Subsystem Integration														
Testing and Refining														
Project Validation														
Final Project Demonstration														

In FYDP (C) the main objective was to construct and implement the final design. For this, resource sourcing was done in week 2 and 3. After that, construction of the subsystem was carried out from week 3 to week 7. Testing and debugging of subsystems were done in parallel between week 5 and week 8. Subsystem integration was performed from week 7 to week 10. Testing and refining of the integrated system were performed between week 9 and week 12 to ensure smooth performance. Project validation was carried out in week 11 to 13, and finally, the complete project demonstration was done in week 14.

7.2.2 Risk Analysis

An important part of engineering project management is to be aware of the risks. This allows the team to be ready for any situation that might happen before, during or even after the project is done.

Before identifying potential risks, why the project might fail should be considered. A roadmap to identifying failure can ensure smooth identification of risk and help mitigate the probable risks associated with it. The fault tree analysis of the project has been illustrated in **Fig. 7.1.**

Fault Tree Analysis

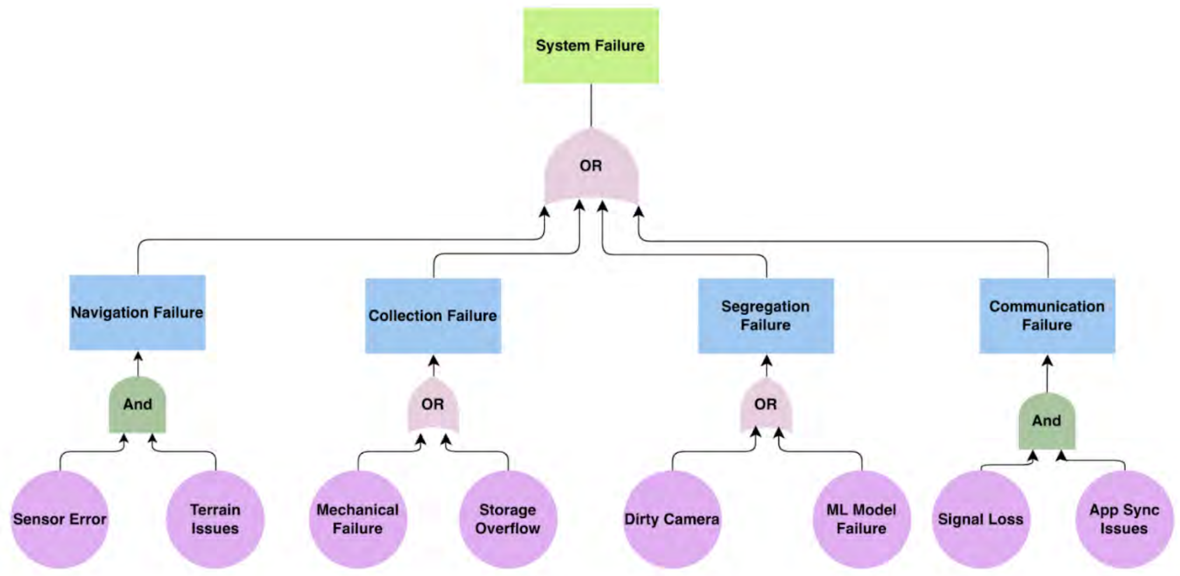


Fig.7.1. Fault Tree Analysis

Risk Analysis: In a project like an Autonomous Rover for Efficient Waste Segregation and Collection, that incorporates advanced technologies, identifying potential risks is critical to ensure smooth operation and efficiency. These risks can range from technical challenges to external factors that may disrupt the system's functionality. Recognizing and addressing these risks proactively is essential for maintaining the project's reliability and effectiveness. The following are some key risks that could impact the project:

1. **Sensors Malfunctioning (A):** Frequent issues with hardware or software in sensors can disrupt waste segregation and hinder overall functionality. This poses a high-impact risk to system efficiency.
2. **Power System Failure (B)**: Although rare, a total power system failure would render the entire system, including the rover and processing units, non-operational, causing significant delays.
3. **Public Interference (C):** Improper usage, tampering, or lack of cooperation from the public could disrupt the waste collection and segregation process, reducing effectiveness.
4. **Terrain Adaptability Issues (D):** Uneven or challenging terrain might limit the rover’s movement or damage its components, especially in open fields or parks.
5. **Storage Bin Overload (E):** Limited capacity of waste bins could lead to overflow, resulting in operational delays and environmental concerns.
6. **Downtime Due to Maintenance (F):** Frequent maintenance needs for the rover and associated systems can temporarily halt operations, causing minor delays.
7. **Waste Collection Mechanism Jam (G):** Mechanical issues in the waste collection mechanism could cause jams, halting operations until resolved, leading to inefficiencies.

Fever Chart:

TABLE 6: FEVER CHART FOR QUALITATIVE ASSESSMENT

LIKELIHOOD	5					
	4				D	A
	3				C	
	2			F	G	B
	1			E		
		1	2	3	4	5
		Impact				

TABLE 7: PROBABILITY X IMPACT ON THE POTENTIAL RISK

Risks	Probability	Impact
Sensors malfunctioning **(A)**	4	5
Power System Failure **(B)**	2	5
Public Interference **(C)**	3	4
Terrain Adaptability Issues **(D)**	4	4
Storage Bin Overload **(E)**	1	3
Downtime Due to Maintenance **(F)**	2	3
Waste Collection Mechanism Jam **(G)**	2	4

Risk Management Plan:

Sensors Malfunctioning (A):

- **Preventive Measures:** Use high-quality, durable sensors with regular calibration protocols. Implement diagnostic tools that can detect early sensor issues.
- **Mitigation Strategy:** Integrate redundant sensors for critical measurements (e.g., temperature, weight) to ensure backup in case of sensor failure.
- **Contingency Plan:** If a malfunction occurs, the system should automatically switch to backup sensors or alert the maintenance team for immediate action.

Power System Failure (B):

- **Preventive Measures:** Ensure a stable power source and charge the rover every 30 minutes.
- **Mitigation Strategy:** Regular maintenance checks on the power system, including batteries and backup generators, to ensure reliability during power outages.
- **Contingency Plan:** In case of power failure, the rover and waste processing units should have a fail-safe mode that prevents loss of data or system settings.

Public Interference (C):

- **Preventive Measures:** Launch awareness campaigns to educate the public about the importance of proper waste disposal and the benefits of the smart waste management system.
- **Mitigation Strategy:** Install monitoring systems (e.g., cameras or sensors) that can detect tampering or improper disposal, and trigger alerts for immediate intervention.
- **Contingency Plan:** Use barriers or secure bins that prevent unauthorized access to the waste collection units. Public feedback systems could also be used to report issues.

Terrain Adaptability Issues (D):

- **Preventive Measures:** Design the rover with robust suspension systems and caterpillar wheels that can navigate through uneven surfaces.
- **Mitigation Strategy:** Equip the rover with adaptive wheels or tracks suited for various terrains (e.g., sand, mud, grass) and regularly test it in different environmental conditions.
- **Contingency Plan:** In case of terrain-related issues, the rover should be able to autonomously adjust its path or trigger a manual override for operator control.

Storage Bin Overload (E):

- **Preventive Measures:** Use waste bins with sensors that can monitor fill levels and automatically alert the system when bins are nearing capacity.
- **Mitigation Strategy:** Schedule waste collection based on real-time data from the sensors to prevent overflow.
- **Contingency Plan: I**f a bin becomes overloaded, the system should automatically reroute the rover to another nearby bin or temporarily store waste in a designated overflow area.

Downtime Due to Maintenance (F):

- **Preventive Measures:** Perform regular preventive maintenance checks on the rover, sensors, and processing units to reduce the likelihood of unexpected failures.
- **Mitigation Strategy:** Keep a schedule for routine maintenance and software updates to minimize system downtime.
- **Contingency Plan:** Have backup units available for critical components to minimize operational delays during maintenance periods. Plan for alternate work shifts to maintain operations.

Waste Collection Mechanism Jam (G):

- **Preventive Measures:** Design the waste collection mechanism with self-cleaning features or easy-to-access parts for maintenance. Use high-quality, durable materials that are less prone to damage.
- **Mitigation Strategy:** Regularly inspect and clean the collection mechanism to ensure smooth operation and prevent buildup that could cause jams.
- **Contingency Plan:** In case of a jam, the system should have an automatic diagnostic feature to identify the issue and either resolve it autonomously or alert the operator for quick intervention.

7.3 Evaluate project progress

The Final Year Design Project started on October 24, 2024, and ended with Project Showcase on September 11, 2025. During the entire process, the project followed most of the plans that were done beforehand. Moreover, the tasks of the project have been distributed evenly among the team. This ensures better efficiency and smooth collaboration among team members. These tasks have been assigned to the team members based on individual strengths so that everyone can work seamlessly to achieve the expected outcome. The responsibilities are outlined below:

FYDP-P

During this semester, the target was to identify an engineering complex problem and propose a solution to solve it. In this timeline, responsibilities were generally planned and assigned which led to a successful project proposal. Some future tasks like Simulation, Algorithm development and hardware constructions which are meant to be done during FYDP-D and FYDP-C were also planned on how the tasks will be divided. Weekly ATC meetings and group meetings helped us keep track of our tasks and update whenever necessary.

TABLE 8: RESPONSIBILITY ASSIGNMENT MATRIX

Tasks	Responsibilities			
	Responsible	**Accountable**	**Consulted**	**Informed**
Literature Review	All	All	All	All
Report Writing	Rubaya	Shah	Nafim, Farhad	All
Market Research	Farhad	Nafim	Shah	Rubaya
Budget Preparation	Farhad	Shah	Nafim	Rubaya
Resources Procurement	Nafim	Rubaya	Farhad	Shah
Simulation of Designs	Farhad	Rubaya	Nafim	Shah
Machine Learning Algorithm Development	Nafim	Shah	Rubaya	Farhad
Hardware Construction	All	All	All	All
Testing and Analysis	Shah	Nafim	Farhad, Rubaya	All
Risk Assessment	Shah	Rubaya	Nafim	Farhad

Evaluation

The first semester where the project proposal was supposed to be submitted followed the proposed timeline. This showed that the team excels in project management skills.

FYDP-D

In FYDP-D, the tasks were to mainly simulate and choose the optimum design to solve the identified problem. During this semester, upon doing literature review and simulation, some design changes were made. Every team member was assigned tasks in which their strength lies in. Different aspects of the simulations were done parallely so that time and resources could be utilised. Some softwares needed to be learnt before using for simulation purposes. Around the 2nd last week, we found our optimum solution and proceeded with that after taking clearance from the ATC panel.

TABLE 9: RESPONSIBILITY ASSIGNMENT MATRIX

Tasks	Responsibilities			
	Responsible	**Accountable**	**Consulted**	**Informed**
Literature Review	All	All	All	All
Report Writing	Rubaya	Shah	Nafim, Farhad	All
Market Research	Farhad	Nafim	Shah	Rubaya
Budget Preparation	Nafim, Rubaya	Farhad	Shah	All
EPS	Shah	Farhad	Rubaya	Nafim
Rover Stability	Rubaya	Nafim	Farhad	Shah
Rover Motion	Shah	Nafim	Rubaya	Farhad
Rover Navigation	Farhad	Shah	Nafim	Rubaya
3D Structure	All	All	All	All
Machine Learning Algorithm Development	Nafim	Shah	Rubaya	Farhad
Risk Assessment	Shah	Rubaya	Nafim	Farhad

Evaluation

Just like FYDP-P, the design semester was on track as well. Everything was done within the planned timeline. However, the Machine Learning results, 3D modelling and choosing the navigation algorithm took longer than expected. Still, we were able to finish our design and could choose the optimal design solution within time.

FYDP-C

During FYDP-C, we are supposed to complete the design solution with a working prototype. The main resource sourcing was done within the first few weeks. After that, construction of the subsystems, testing and debugging was done. During this time, some components procurement was done for design consideration, testing purposes, replacement of damaged components and so on. When the subsystems were working perfectly according to the plan, we started system integration. Then testing, refining and data collection was done. Finally, the project was decorated to make it presentable for the showcase. Once again, the tasks were assigned based on the team member's strength.

During project completion, we took help from our ATC panel members, seniors, and LTO for some advice to improve the rover capabilities. As this was a mechanical project and all of the team members are electrical engineering students, we had to get help from a workshop to design and build the wheels for the rover. Mr. Amzad, the mechanic, helped us constantly to build and develop our rover wheels as it went through multiple R&D stages. Every major update was shared to the ATC Chair. All the team members equally contributed to this project and ensured that the tasks were done right and within time. Without everyone's active support, contribution, input and insights this project would have been impossible.

TABLE 10: RESPONSIBILITY ASSIGNMENT MATRIX

Tasks	**Responsibilities**			
	Responsible	**Accountable**	**Consulted**	**Informed**
Market Research	Farhad	Rubaya	Shah	Nafim
Budget Preparation	Nafim, Rubaya	Farhad	Shah	All
Component Procurement	Nafim	Farhad	Rubaya	Shah
Mechanical Structure (Rover Body, Wheel)	Nafim	Farhad	Rubaya	Shah
Robotic Arm	Farhad	Rubaya	Nafim	Shah
Electrical Circuits	Nafim	Farhad	Shah	Rubaya
System Integration	All	All	All	All
System Testing	All	All	All	All
Report Writing	Rubaya	Nafim	Farhad, Shah	All

Evaluation:

Every work in this semester was done within the given timeline as well. Due to maintaining a strict plan of action and every member working in the lab every day except for holiday was the reason the project could be completed on time. Some unexpected situations like last moment design changes, components failure did set us on a backtrack. However, as we had done risk analysis and almost perfect planning during FYDP-P, we managed to pull it off within time. The unexpected Research and Development cost during Wheel design and building was also one of the issues we had faced during this time. The learning experience during FYDP-C was that there is always a huge R&D cost in projects that involve mechanical aspects. The impromptu change of plan during any challenge and logbook helped us keep on track.

7.4 Conclusion

The project's aim was to minimize the waste relevant issue in Bangladesh, specifically urban places like Dhaka. Project management allowed us to complete the project efficiently. The Gantt Chart helped us to keep track of our pre-planned work. It also gave us enough time to act whenever we faced any challenges. Regular group meetings with the ATC Panel members and within the group allowed us to note down progress, any challenges that were faced, identify and mitigate issues, talk about disagreements and come to common ground, figure out solutions and so on. These practices helped us to continue the project within time. Also, collaboration with a workshop to design and build the wheel played a vital role in the completion of this project.

Chapter 8: Economical Analysis. [CO12]

8.1 Introduction

The economical analysis of the autonomous waste sorting rover project is essential to assess its feasibility, viability, and potential impact. This chapter gives a detailed overview of the project's costs, benefits and financial implications. By evaluating investments, costs and long-term returns, we try to show how this rover built locally can be a cost-effective and automated alternative to traditional waste sorting methods. The chapter covers detailed economic breakdowns, cost-benefit assessments, and evaluations of affordability and risks, ultimately highlighting the rover's role in fostering economic efficiency and environmental sustainability in Bangladesh's waste sector.

8.2 Economic analysis

Economic analysis provides a structured evaluation of the financial inputs, outputs, and overall feasibility of deploying the autonomous waste sorting rover. This involves quantifying initial investments, ongoing expenses, and long-term value to determine its practicality in addressing Bangladesh's escalating waste crisis. While designing the rover, the aim was to minimize production cost and maximize operational efficiency.
The initial capital cost, excluding research and development, totals 57,666 BDT for the prototype. This represents a reduction from the budgeted 62,423 BDT through optimized sourcing. Below is a detailed breakdown of the budget done during FYDP-P and FYDP-D and the total cost excluding Research and Development cost:

Table 8.1: Initial Bill of Material for Post-Sorting Approach

Item	Specification	Unit Price	Quantity	Total Price
Ultrasonic Sensors	HC-SR04, range: 2 cm–400 cm	75	10	750
Bicycle Chain		1200	2	2400
YDLIDAR X4	Scanning range: 10m, 360° field of view	9000	1	9000
Plastic Waste Bins	Durable plastic, capacity: 2.5L, compartments	500	1	500
Arduino Mega	ATmega2560, 16MHz clock speed	5430	1	5430
Servo Motor MG995 180 Degree Rotation	Operating Voltage: 4.8~7.2V, Torque: 10~12kg-cm	435	3	1305
TD-8120MG Digital High Torque Servo Motor Metal Gear	Operating Voltage: 4.8V~7.2V	1450	1	1450
Motor Driver 43A BTS7960	Operating Voltage: 6V ~27V Maximum 43A Current	625	2	1250
Gear Motor	37GB 12V High Torque; 60RPM	2200	2	4400
3D Printing Filament	PLA, 1.75mm diameter	3640	2	7280
IMU (Inertial Measurement Unit)	MPU6050, 6-axis gyro and accelerometer	208	1	208
Battery Pack	Li-Po; 11.1V; 10000 mAh	6600	1	6600
Raspberry Pi 5		15420	1	15420
Raspberry Pi Camera Module V2	8 Megapixel,1080p	4690	1	4690
Raspberry Pi 5 Active Cooler		740	1	740
TOTAL				**62,423**

TABLE 8.2: FINAL BILL OF MATERIAL FOR PRE-SORTING APPROACH

Item	Specification	Unit Price (Tk)	Quantity	Total
Robotic Arm	6 Degree of Freedom	4250	1	4250
Raspberry Pi 5	Raspberry Pi 5 + Pi Camera + Pi Cooler	13900	1	13900
Wheels	8 gears + welding + shielding	1500	4	6000
Li-Po Battery	5400 mAh	6260	1	6260
Ultrasonic Sensor	HC-SR04	100	1	100
Arduino	Uno R3	483	1	483
ESP32	30P NodeMCU Developement Board Wireless WiFi	449	1	449
Voltage Regulator Buck Converter	Max 5-40 V to 1.2 - 36V Step Down	425	3	1275
PVC	Board	1000	1	1000
Bin	10L	230	1	230
Servo Motor	TD-8120MG Digital High Torque 20 kgcm	2350	4	9400
Servo Motor	TD-8135MG Digital High Torque 35 kgcm	2350	1	2350
Motor Driver	43A BTS7960	625	2	1250
Gear Motor	37GB 12V High Torque Gear Motor 60RPM	2200	4	8800
Connector	XT60 3.5mm Male and Female Pair	60	5	300
PCA9685	16 channel 12 bit PWM	484	1	484
SD Card	64GB	1000	1	1000
Servo Motor	SG90	135	1	135
Total Price				**57666**

The Research and Development cost of this project was around 23,000/=. The main cost was spent on wheel development, creating an artificial testing environment and some miscellaneous cost. We had to buy artificial grass to simulate the urban field ground as during the testing period which was from late July to the beginning of September 2025, it rained almost every day. As our project was at its prototype stage, it wasn't made water resistant, and it needed to be tested on an outdoor field, which was impossible due to adverse weather. This is why testing on artificial grass was a necessity. Altogether, our total cost including R&D was about 80,000/=.

8.2.1. Subsystem-wise Breakdown

TABLE 8.3: SUBSYSTEM WISE BREAKDOWN OF ROBOTIC ARM

Robotic Arm				
Item	**Specification**	**Unit Price (Tk)**	**Quantity**	**Total**
Robotic Arm	6 Degree of Freedom	4250	1	**4250**
PCA9685	16 channel 12 bit PWM	484	1	**484**
Arduino	Uno R3	483	1	**483**
Servo Motor	SG90	135	1	**135**
Ultrasonic Sensor	HC-SR04	100	1	**100**
Servo Motor	TD-8120MG Digital High Torque 20 kgcm	2350	4	**9400**
Total				**14852**

TABLE 8.4: SUBSYSTEM WISE BREAKDOWN OF CAMERA AND BIN

Camera & Bin System				
Raspberry Pi 5	Raspberry Pi 5 + Pi Camera + Pi Cooler	13900	1	**13900**
SD Card	64GB	1000	1	**1000**
Servo Motor	TD-8135MG Digital High Torque 35 kgcm	2350	1	**2350**
Bin	10L	230	1	**230**
Total				**17480**

TABLE 8.5: SUBSYSTEM WISE BREAKDOWN OF ROVER

Rover				
Item	**Specification**	**Unit Price (Tk)**	**Quantity**	**Total**
Voltage Regulator Buck Converter	Max 5-40 V to 1.2 - 36V Step Down	425	3	**1275**
ESP32	30P NodeMCU Developement Board Wireless WiFi	449	1	**449**
Wheels	8 gears + welding + shielding	1500	4	**6000**
PVC	Board	1000	1	**1000**
Motor Driver	43A BTS7960	625	2	**1250**
Gear Motor	37GB 12V High Torque Gear Motor 60RPM	2200	4	**8800**
Connector	XT60 3.5mm Male and Female Pair	60	5	**300**
Li-Po Battery	5400 mAh	6260	1	**6260**
Total				**25334**

8.2.2. Operational Cost

Operational costs include electricity for charging the battery, maintenance, and minor repairs. Based on component power consumption estimates:

- Raspberry Pi 5: Average 5W.
- Servo motors (e.g., TD-8120MG): Up to 22W stalled, but average ~5W per motor during intermittent use.
- Gear motors (37GB 12V 60RPM): ~10W average per motor.
- Other components (sensors, drivers): ~5-10W total.

Assuming an average system power draw of 20W during operation and 8 hours of daily use, daily energy consumption is approximately 0.16 kWh. At an average electricity rate of 10 BDT/kWh in Bangladesh, this equates to ~1.6 BDT/day or ~600 BDT/year.

Maintenance is estimated at 5% of capital cost annually (~2,883 BDT), covering battery replacements every 2-3 years and minor part repairs. Total operational cost: ~5,000 BDT/year.

8.2.3. Life Cycle Cost

To judge a system's sustainability prowess and economic factor, the assessment should be done specifically on the lifetime running factor, through this assessment it is judged if the system is sustainably viable in the long run, and this term is known as life cycle cost.

Over a 5-year lifespan, the life cycle cost includes capital (57,666 BDT) plus operational costs (5,000 BDT/year × 5 = 25,000 BDT), totaling ~82,666 BDT. This assumes no major overhauls and recycling of components at end-of-life.

8.2.4. Payback Period

This is how long it takes to recover the initial cost from yearly savings.
Equation: $Payback\ Period\ =\ Initial\ Investment\ /\ Annual\ Net\ Savings$
Assuming the rover replaces one manual waste sorter (average salary ~10,000 BDT/month or 120,000 BDT/year), net annual savings are 115,000 BDT (after operational costs). With a selling price of 69,300 BDT, the payback period is 69,300 / 115,000 ≈ 0.6 years (7 months).

8.3 Cost benefit analysis

Cost-benefit analysis measures the rover's expenses against its tangible and intangible returns, emphasizing efficiency gains in waste management. Monetary benefits include labor savings of 120,000 BDT annually per unit, potentially diverting 50% more waste from landfills through automated sorting. Non-monetary advantages are reduced health risks for workers and environmental benefits, such as lower pollution from inefficient disposal.

8.3.1. Cost vs Selling Price

The total cost of the system is 57,666 BDT. The selling price has been set by keeping a 20% profit margin. The cost and selling price have been given below:

TABLE 8.6: COST AND SELLING PRICE COMPARISON

Subsystem	Cost (BDT)	Selling Price (BDT)
Rover	25,334	30,418
Robotic Arm	14,852	17,893
Camera & Bin	17,480	20,989
Total	**57,666**	**69,300**

8.3.2. Competitor Analysis

To contextualize the rover's value proposition, a comparison with leading international competitors reveals substantial cost advantages. Our solution, priced at 69,300 BDT (~580 USD), is 40-70% cheaper than market alternatives, making it highly accessible for resource-constrained settings like Bangladesh. Although there is no commercially available autonomous rover for waste segregation and collection with the exact same specification as ours, we found some competitors which have similar working principles to understand the current market trends and price. While choosing competitors, the priority was given to the specifications and mechanical aspects which are Key competitor analysis is given below:

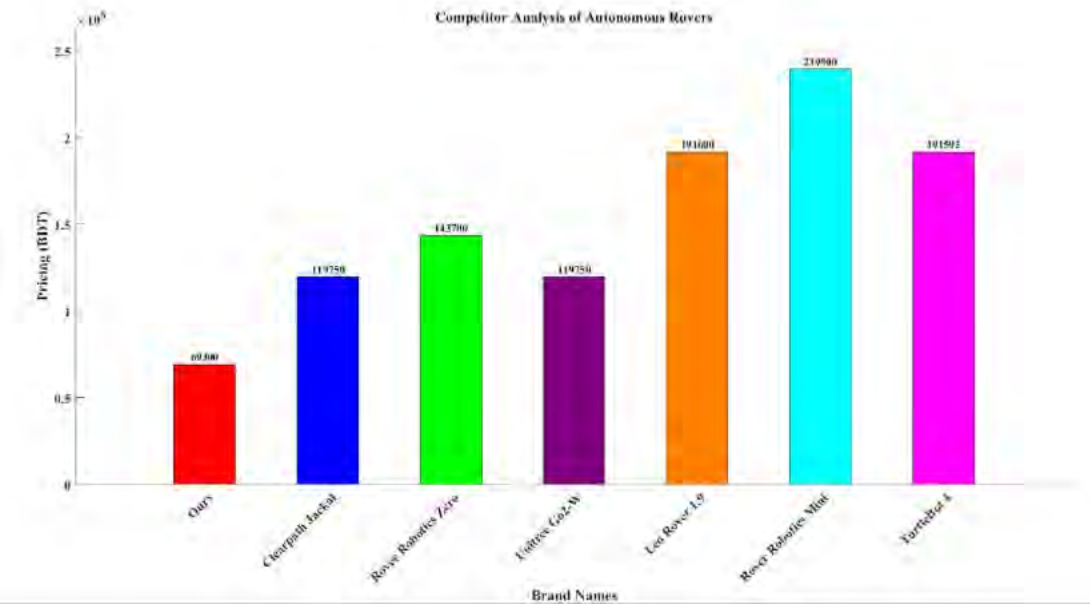


Fig.8.1 Competitor Market Analysis

From the bar chart, it can be seen that, our design (red bar) at 69,300 BDT significantly undercuts competitors, with Clearpath Jackal (blue) at 117,500 BDT, Rover Robotics Zero (green) at 143,700 BDT, Unitree GO2-W (purple) at 119,750 BDT, Leo Rover 1.9 (orange) at 196,000 BDT, Rover Mini (light blue) at 239,500 BDT, and TurtleBot 4 (pink) at 191,592 BDT. Y-axis scaled to ×10^5 BDT, highlighting our **40-70%** cost advantage across the board. This analysis shows that our rover is quite affordable compared to the existing rovers in the market with similar

This comparison shows our rover is far more affordable than competitors as on average, their prices are 143% higher. It means that the buyers can save from 50,000 to 170,000 BDT for each unit if they buy our project as a product. Competitors do have some better features, like faster speeds (for example, the GO2-W reaches 2.5 meters per second compared to our 1 m/s) or built-in software for easy setup (like ROS). However, adding waste-sorting tools to those systems often costs an extra 20-50% of the base price, which surpasses the base value anyways. In Bangladesh, imported robots face 30-50% extra taxes, but our home-built design skips those fees. This makes it easier to launch quickly and cheaper to own over time, giving us about 20% better Return on Investment (ROI).

Our return on investment (ROI) is around 20%. This comes from making a profit of 11,634 BDT on each unit. For buyers, it means they get their money back fast by cutting down on worker wages.

$$ROI = [(Selling\ Price\ -\ Production\ Cost)\ /\ Production\ Cost]\ \times\ 100\%$$

Break-even point: For as sellers, we turn a profit right after selling just one unit, since our fixed costs are low. For users, they recover their costs in about 7 months.

$$Break\ even\ Unit = \ Fixed\ Costs\ /\ (Selling\ Price\ per\ Unit\ -\ Variable\ Cost\ per\ Unit)$$

8.4 Evaluate economic and financial aspects

The economic and financial evaluation of the autonomous waste sorting rover shows its strong potential for cost-effectiveness and market acceptance. From a producer's perspective, the project shows clear financial sustainability. With a production cost of 57,666 BDT and a selling price of 69,300 BDT, each unit has a profit of 11,634 BDT with a 20% margin. The low fixed cost structure means profitability is achieved from the sale of the very first unit, making this project financially attractive even on small production scales.

From a buyer's perspective, the rover offers an excellent return on investment. By replacing manual waste sorters who earn around 120,000 BDT annually, the rover generates annual net savings of about 115,000 BDT, after accounting for operational costs. The rover provides one of the shortest recovery times with a payback period of only 7 months. This ensures financial viability for municipalities, NGOs and private waste contractors.

In terms of life cycle costs, the rover's total 5-year expense altogether is ~82,666 BDT. Compared to recurring labor costs, this represents a substantial financial advantage. The competitor analysis highlights that our rover is 40–70% cheaper than comparable international alternatives. Even though foreign robots have higher performance, including waste management mechanisms and import tax will increase the overall cost over ours. In this case, the locally developed rover balances affordability with core functionality.

From a macroeconomic perspective, adopting such rovers in Bangladesh can contribute to reducing municipal waste management costs, creating new opportunities for local manufacturing, and minimizing reliance on expensive imports. This strengthens economic self-reliance and fosters local innovation.

Most of the components are locally sourced, for which financial risks are low for this project. This allows the components to be replaced anytime at a relatively low cost. However, potential risks include market adoption challenges, the need for after-sales service infrastructure, and scaling production while maintaining cost efficiency.

Overall, the economic and financial assessment confirms that the autonomous waste sorting rover is both feasible and profitable. It delivers strong financial returns for both sellers and buyers, offers a significant competitive advantage in terms of affordability, and has the potential to drive sustainable waste management practices in Bangladesh's public spaces.

8.5 Conclusion

In summary, the economic analysis affirms the autonomous waste sorting rover as a financially sound innovation, with low costs, a 20% ROI, and swift payback supporting its adoption in Bangladesh's waste sector. By integrating affordability with environmental gains, it paves the way for scalable, sustainable waste management solutions.

CHAPTER 9: ETHICS AND PROFESSIONAL RESPONSIBILITIES [CO13, CO2]

9.1 Introduction

Modern Engineering practices have three fundamental pillars - Ethics, professional responsibility and adhering to safety standards. Due to increasing integration of technological systems into daily lives, engineers are tasked to not only ensure the functionality of the system but also to consider the welfare of the public, sustainability of the environment all while ensuring professional integrity. This chapter highlights these sections that act as the core foundation to the development of an autonomous rover for efficient waste segregation and collection in urban parks and fields.

9.2. Identify ethical issues and professional responsibility

As the systems designers, the most crucial responsibility is to ensure that it serves its purpose effectively, without compromising safety, quality or environmental decorum. Since the autonomous system will be deployed in urban spaces, the design must prioritize ethical responsibility so that it can benefit society all while avoiding harm.

- Responsibility towards the environment is the first and most important responsibility as the design is incorporated to segregate waste into two categories biodegradable and non-biodegradable with the aim of reducing environmental pollution. However, this responsibility is not only limited to segregation but also requires collaboration with proper waste management authorities, so that the post work can be done smoothly. Subsequently, use of sustainable materials and technologies that are energy efficient must be incorporated.
- Ensure public safety and interaction, since the rover will be deployed in urban areas, the system must be equipped with an obstacle avoidance mechanism ensuring no harm befalls living beings.
- Ensuring there is no bias in machine learning models, since bias can lead to misclassification. Engineers must use diverse datasets to ensure fairness and proper segregation is done.
- Ensuring the system is both affordable and adaptable so that it can benefit both underdeveloped and developing regions. Through this initiative, engineers must ensure bridging the social and economic disparities.
- As an engineer, international codes and standards must be strictly adhered to. Following these standards ensures societal trust, an act of professionalism and accountability in the practice of engineering.
- Since the system is equipped with a camera with the purpose of detecting waste, there might be a chance for unintended capture of personal or sensitive information. As part of professional responsibility, strict data governance must be ensured so that the camera is solely used for waste classification with no storing mechanism of private data.

9.3. Apply ethical issues and professional responsibility

9.3.1. Ethical Consideration (CO13)

The project addresses critical ethical concerns to ensure it is both socially responsible and beneficial to the environment where the key consideration is reducing environmental pollution. The rover segregates waste into Biodegradable and Non-Biodegradable categories on-site, promoting better waste management practices. However, to maintain ethical standards, it is crucial to ensure that segregated waste is disposed of properly. This involves collaborating with waste management authorities or establishing protocols for environmentally friendly waste disposal. Using sustainable materials and energy-efficient technologies in the rover's design further aligns with eco-friendly goals and global standards such as **IEEE 1680.1**.

Public interaction with the rover is another significant ethical aspect since operating in urban parks and fields requires ensuring that the rover does not inconvenience or harm people or animals. To achieve this, the rover’s movements and actions must be communicated clearly using visual signals or sounds. This transparency fosters public trust and ensures the system integrates seamlessly into everyday public spaces.

Bias in machine learning models used for waste classification is a potential ethical concern. These models must be trained on diverse datasets to ensure accurate classification of waste types across different regions and environments. Any biases could lead to misclassification, reducing the system’s effectiveness and causing public distrust. Ensuring fairness in model training is a priority.

Accessibility and inclusivity are central to the project’s ethical framework. The rover must be affordable and deployable in resource-constrained settings to ensure that it benefits communities in both developed and developing regions. This approach aligns with broader social equity goals, addressing waste management challenges in underserved areas and promoting global sustainability.

9.3.2. Safety Consideration (CO3)

The project incorporates comprehensive safety measures to protect users, operators, and the environment. Mechanical safety is a primary focus, as the rover’s moving parts, such as the robotic arm and wheels, could pose risks. To mitigate this, the rover is designed with protective enclosures for these components. Emergency stop mechanisms are also integrated, allowing operators to halt the system immediately in case of unexpected behavior or emergencies. These features ensure the system is safe for public operation.

Electrical safety is another crucial aspect, particularly since the rover operates outdoors and may encounter water or other conductive materials. Proper waterproofing and insulation of all electrical components are implemented to prevent short circuits or electric shocks. Adherence to **IEEE C95.1** standard ensures that the electromagnetic emissions from the rover are within safe limits, protecting humans and animals nearby.

Public interaction in urban spaces introduces additional safety concerns. People or animals might accidentally interfere with the rover's operation. To address this, the rover includes clear alerts such as flashing lights or sound signals to notify individuals of its movements. This helps to avoid accidents and ensures safe coexistence in shared spaces.

Material durability is a significant consideration for safety. The rover must operate in diverse environmental conditions, including exposure to waste and harsh weather. Components are selected for their corrosion resistance and ability to withstand long-term exposure to challenging conditions. Standards like **ASTM F2398-20** for robotic systems in confined spaces guide the material selection process.

Terrain adaptability is essential for operational safety. The rover is designed to remain stable on uneven or slippery surfaces, preventing tipping or operational failures. This adaptability ensures the rover can navigate various terrains, such as wet grass or gravel, without compromising safety or functionality.

Safe waste handling is a top priority to protect operators and the public. The waste compartments are designed to minimize direct human contact with hazardous materials during disposal. This hygienic approach reduces health risks and ensures compliance with safety standards. Additionally, decontamination protocols should be followed waste disposal to further reduce exposure to harmful pathogens.

Finally, fail-safe mechanisms to address potential malfunctions or extreme weather conditions as these measures will allow operators to halt the rover in case of emergencies, ensuring continuous and secure operation. By addressing these safety considerations, the project ensures the rover is reliable, safe, and suitable for use in public spaces.

9.4 Conclusion

In summary, this chapter goes through the importance of transparency, accountability and safety needed for engineering innovation. Through the use of international standards, prioritizing the users as well as environment safety and ensuring the access of technology without any disparity. The ethical and safety elements ensure that the autonomous waste rover not only fulfils its purpose in addressing the urban waste challenges but also aligns with professional ethics and expectations set by society.

CHAPTER 10: CONCLUSION AND FUTURE WORK.

10.1 Project summary

The development of the Autonomous Rover for Efficient Waste Segregation and Collection in Urban Parks and Fields can be one of the ways in addressing urban waste management challenges, particularly in densely populated cities like Dhaka. The project is developed through a systematic design process including multiple phases like developing the project proposal, verifying designs through simulation and choosing the optimal design which is the Pre-Sorting approach of waste collection. This approach balances cost, efficiency and sustainability.

The final design was validated through rigorous simulations (e.g., MATLAB Simulink, Proteus) and performance evaluations, demonstrates a robust mechanical arm with a 4DOF configuration, optimized for picking and placing waste, alongside a stable rover body equipped with advanced navigation (A* algorithm) and power systems tailored for 30-minute autonomy. The integration of the YOLOv11n model ensures high accuracy in waste classification (F1-scores of 0.9896 for Biodegradable and 0.9630 for Non-Biodegradable), validated through extensive testing on a balanced dataset of 2986 images. The project expectation and achieved specification is given below:

TABLE 10.1: COMPARISON OF PROPOSED VS ACHIEVED SPECIFICATION

Project Verification	
Proposed Specifications	**Achieved Specifications**
Runtime: **30 mins** per full charge	Runtime: **50 mins** per full charge
Total Rover Weight < **40Kg**	Weight: **16.9 Kg**
Bin volume: 7000cm³-8000cm³ with Maximum 5kg Load	The bin is 7000cm³ and can carry 5kg load
Carries elongated object ranging from **10-20cm** in length/diameter	Maximum tested length: **22cm** and diameter: **10cm**
Waste Classification accuracy 95%	Waste Classification accuracy 84%
Rover Speed: 0.5m/s-1.5 m/s	Rover Speed: 0.25m/s

Project verification confirmed that the pre-sorting approach meets the desired needs, achieving over 95% classification accuracy, 100% field coverage within 20 minutes, and a power consumption of 309.24 W, well within the 11.1V, 1000mAh battery capacity. The solution aligns with project constraints, including a budget of 56,338 BDT, and outperforms the post-sorting alternative in cost-effectiveness, manufacturability, and environmental impact (114.25 kJ energy usage). Impact analysis highlights societal benefits—improved public spaces and reduced health risks—alongside legal compliance (e.g., Solid Waste Management Rules 2021) and cultural shifts toward sustainable practices. Economic analysis suggests a 40–70% cost advantage over competitors, with a projected 20% ROI, while ethical considerations ensure safety (e.g., IEEE C95.1 compliance) and inclusivity. The project's adherence to engineering standards (e.g., IEEE 1680.1, ASTM F2398-20) and effective team management, as evidenced by the Gantt charts and risk mitigation strategies, underscore its success.

This endeavor not only fulfills the technical objectives but also contributes to Sustainable Development Goals 3 (Good Health and Well-Being) and 11 (Sustainable Cities and Communities), offering a scalable model for urban waste management. The systematic approach, from design optimization to validation, demonstrates a logical and comprehensive engineering process, validating the solution's readiness for real-world deployment.

10.2 Future work

- **Implement ROS for optimum autonomous navigation: Integrate** the Robot Operating System (ROS) to enhance navigation precision and enable advanced path planning, improving obstacle avoidance in dynamic urban environments.
- **Obstacle avoidance and safety alert:** Develop real-time obstacle detection using additional LiDAR or camera inputs, coupled with audible and visual alerts, to ensure safer operation around pedestrians and wildlife.
- **Plan to pursue funding to support future developments:** Seek grants or partnerships with municipal bodies and environmental organizations to support scalability, testing in diverse terrains, and long-term maintenance.
- **Enhance design to create a compact product ready for the market:** Refine the design into a compact, modular product using locally sourced parts, optimizing for mass production and affordability, with a focus on reducing weight and power consumption for broader accessibility.

CHAPTER 11: IDENTIFICATION OF COMPLEX ENGINEERING PROBLEMS AND ACTIVITIES.

11.1 Attributes of Complex Engineering Problems (EP)

TABLE 11.1: ATTRIBUTES OF COMPLEX ENGINEERING PROBLEM (EP)

	Attributes	**Put tick (√) as appropriate**	**Justification**
P1	Depth of knowledge required	√	Requires EEE, Mechanical, and ML expertise for problem solution.
P2	Range of conflicting requirements	√	Must balance public convenience, safety, system cost and standards.
P3	Depth of analysis required	√	Needs in-depth study of waste patterns, terrain, user behavior, and feasibility.
P4	Familiarity of issues		
P5	Extent of applicable codes		
P6	Extent of stakeholder involvement and needs	√	Involves city authorities, waste handlers, and public users with diverse priorities.
P7	Interdependence	√	The problem affects public health, environment, and municipal operations simultaneously.

11.2. Attributes of Complex Engineering Activities (EA)

TABLE 11.2: ATTRIBUTES OF COMPLEX ENGINEERING ACTIVITIES (EA)

	Attributes	**Put tick (√) as appropriate**	**Justification**
A1	Range of resource	√	Needs multiple resources to implement
A2	Level of interaction	√	Demands collaboration across engineers, authorities, and users.
A3	Innovation		
A4	Consequences for society and the environment	√	Directly impacts urban cleanliness, public health, and downstream recycling processes.
A5	Familiarity		

References

[1] S. S. Shah, S. Saad, and aleeb, “Dhaka has a waste management problem,” *The Daily Star*, Jul. 25, 2022. https://www.thedailystar.net/opinion/views/news/dhaka-has-waste-management-problem-3079156

[2] T. Hossain, “Solid Waste Management in Bangladesh,” *Breaking World News | THX News*, Oct. 06, 2024. https://thxnews.com/2024/10/06/solid-waste-management-in-bangladesh/ (accessed Nov. 13, 2024).

[3] Anitha G, M. Priscilla, and Ramachandran. R, “Internet of Things Oriented Smart Waste Segregation Robot Design using Intelligent Sensors with Controller Association,” pp. 1–7, Apr. 2024, doi: https://doi.org/10.1109/iconstem60960.2024.10568617.

[4] S. Pp, S. Chaand, P. Kp, B. Grandhep, and H. Ap, “Waste Segregation Robot -A Swachh Bharat Initiation,” *IJISET -International Journal of Innovative Science, Engineering & Technology*, vol. 7, 2020, Available: https://ijiset.com/vol7/v7s12/IJISET_V7_I12_42.pdf

[5] Md. Ashikuzzaman and Md. H. Howlader, “Sustainable Solid Waste Management in Bangladesh,” *Advances in Environmental Engineering and Green Technologies*, pp. 35–55, 2020, doi: https://doi.org/10.4018/978-1-7998-0198-6.ch002.

[6] RECLAIM Consortium, "Gripping mechanisms and RoReWo units for material recovery," *HORIZON-CL4-2021-DIGITAL-EMERGING-01: AI, Data and Robotics for the Green Deal (IA)*, Feb. 29, 2024.

[7] J. Lahoti, J. Sn, M. V. Krishna, M. Prasad, R. BS, N. Mysore, and J. S. Nayak, "Multi-class waste segregation using computer vision and robotic arm," *PeerJ Comput. Sci.*, vol. 10, p. e1957, May 2024, doi: 10.7717/peerj-cs.1957.

[8] S. M. P. Sunil, S. C. P. K., B. G. P., and H. S. A., "Waste segregation robot - A Swachh Bharat initiation," *Int. J. Innov. Sci., Eng. Technol.*, vol. 7, no. 12, pp. 505–512, Dec. 2020. [Online]. Available: http://www.ijiset.com/

[9] M. Kulshreshtha, S. S. Chandra, P. Randhawa, G. Tsaramirsis, A. Khadidos, and A. O. Khadidos, "OATCR: Outdoor autonomous trash-collecting robot design using YOLOv4-Tiny," *Electronics*, vol. 10, no. 2292, pp. 1–22, Sep. 2021, doi: 10.3390/electronics10182292.

[10] A. E. Ibrahim, R. Shoitan, M. M. Moussa, H. A. Elnemr, Y. I. Cho, and M. S. Abdallah, "Object detection-based automatic waste segregation using robotic arm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 6, pp. 912–919, Jun. 2023. [Online]. Available: http://www.ijacsa.thesai.org

[11] Y. Zhou *et al.*, “A mobile robot with a manipulator to alleviate the shortage of health workers in hospitals in COVID-19,” *Highlights in Science, Engineering and Technology*, 2023.

[12] Y. Zhang *et al.*, “Trajectory tracking of a 3-DOF robot arm using fuzzy-PD control in SimMechanics,” in *Proc. IEEE URAI*, 2019.

[13] B. Oni *et al.*, “Joint torque and motion computational analysis for robotic manipulator arm design,” *Journal of Engineering & Applied Sciences*, vol. 13, no. 5, pp. 1124-1131, 2018.

[14] C. Chen *et al.*, “Using ultrasonic sensors and a knowledge-based neural-fuzzy controller for mobile robot navigation control,” *Electronics*, vol. 10, no. 15, 2021.

[15] I. Chatzisavvas *et al.*, “Optimizing mobile-robot navigation based on A-star algorithm for obstacle avoidance in smart agriculture,” *Electronics*, vol. 12, no. 5, 2023.

[16] J. Faigl *et al.*, “Zig-zag coverage planning for indoor UAV inspection,” in *Indoor UAS Conf.*, 2019.

[17] S. Zhu, H. Chen, M. Wang, X. Guo, Y. Lei, and G. Jin, “Plastic solid waste identification system based on near infrared spectroscopy in combination with support vector machine,” *Advanced Industrial and Engineering Polymer Research*, vol. 2, no. 2, pp. 77–81, Apr. 2019, doi: 10.1016/j.aiepr.2019.04.001. Available: https://www.sciencedirect.com/science/article/pii/S2542504818300113

[18] Z. Hou *et al.*, “Energy modeling and power measurement for mobile robots,” *Energies*, vol. 12, no. 5, 2019.

[19] K. Zhang *et al.*, “Energy modeling and experimental validation of four-wheel mecanum mobile robots for energy-optimal motion control,” *Symmetry*, vol. 11, no. 1, 2019.

[20] ZenRobotics, “ZenRobotics C&D/C&I Waste Sorting with the Heavy Picker - Full Processing Video,” *YouTube*. Dec. 20, 2023. [Online]. Available: https://www.youtube.com/watch?v=2L2gEFVNRBY

[21] “AI-Powered Waste Sorting Solutions.” https://ampsortation.com/

[22] Waste Robotics, “Waste Robotics - AI sorting robot for heavy waste separation,” *Waste Robotics* -, Feb. 26, 2025. https://wasterobotic.com/

[23] Libretexts, “9.2: P, I, D, PI, PD, and PID control,” *Engineering LibreTexts*, Mar. 11, 2023. Available: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/09%3A_Proportional-Integral-Derivative_(PID)_Control/9.02%3A_P_I_D_PI_PD_and_PID_control

[24] M. Akter, “BDWaste: A comprehensive image dataset of digestible and indigestible waste in Bangladesh,” *Mendeley Data*, vol. 1, Nov. 2023, doi: https://doi.org/10.17632/96g5pgfnfw.1.

[25] “GARBAGE CLASSIFICATION 3 Object Detection Dataset and Pre-Trained Model by Material Identification,” *Roboflow*. https://universe.roboflow.com/material-identification/garbage-classification-3

[26] bnsreenu, “GitHub - bnsreenu/digitalsreeni-image-annotator: A python based GUI to annotate images and save annotations as COCO style JSON format.,” *GitHub*, 2024. https://github.com/bnsreenu/digitalsreeni-image-annotator (accessed Mar. 27, 2025).

[27] “GrabCAD Library – Free CAD Models,” GrabCAD, [Online]. Available: https://grabcad.com/library. Accessed: April 19, 2025.

[28] S. Khan et al, "Bangladesh records persistently increased number of dengue deaths in recent years: Dissecting the shortcomings and means to resolve - ScienceDirect", Sciencedirect, 25 June. 2024. Available: https://sciencedirect.com/science/article/pii/S2772707624000663. [Accessed 08 December. 2024].

[29] “The Penal Code, 1860,” *bdlaws.minlaw.gov.bd*. http://bdlaws.minlaw.gov.bd/act-11.html

[30] “The government of Bangladesh announces its Solid Waste Management Rules 2021. | Department of Economic and Social Affairs,” *sdgs.un.org*. https://sdgs.un.org/partnerships/government-bangladesh-announces-its-solid-waste-management-rules-2021

[31] “বাংলাদেশ শ্রম আইন, ২০০৬,” *bdlaws.minlaw.gov.bd*. http://bdlaws.minlaw.gov.bd/act-952.html

APPENDIX

LOGBOOK

Final Year Design Project (C) Summer-2025			
Student Details	**NAME & ID**	**EMAIL ADDRESS**	**PHONE**
Member 1	Rubaya Huq (21221003)	rubaya.huq@g.bracu.ac.bd	01711381242
Member 2	Farhad Hassan (21221025)	farhad.hassan@g.bracu.ac.bd	01644921986
Member 3	Nafim Karim Khan (21221039)	nafim.karim.khan@g.bracu.ac.bd	01782021829
Member 4	Shah Rasul Khan (21221054)	shah.rasul.khan@g.bracu.ac.bd	01754317430
ATC Details:			
ATC 4			
Chair	Dr. Mohammed Belal Hossain Bhuian	belal.bhuian@bracu.ac.bd	
Member 1	Md. Mahmudul Islam	mahmudul.islam@bracu.ac.bd	
Member 2	Sabbir Ahmed	ahmed.sabbir@bracu.ac.bd	

FYDP-P

Date/Time/Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
31.10.2024	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1.Need to narrow down the scope for proposed and shortlisted 2 ideas. 2.Need to finalize the problem statement. 3.Prepare a draft proposal note for both ideas before meeting chair	Task 1: All Task 2: Nafim, Farhad, Shah Task 3: All	Shortlisted 2 ideas from 3 and try to do the tasks before meeting chair officially next Sunday (03.11.2024)
03.11.2024	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1. Shortlisted 1 problem 2. Needs more literature review 3. Prepare 2 different design approach for the system	Task 2: All Task 3: All	Task 1: Done Task 2: Done Task 3: Partially done
10.11.2024	1. Rubaya 2. Farhad 3. Nafim	1. Design proposal approved 2. Start working on Specifications 3. Prepare Draft concept Note	Task2: All	Task2: Done Task3:Done
14.11.2024	1. Rubaya 2. Nafim	1. Checked draft proposal and did some correction 2. Prepare for progress presentation	Task1: All Task2: All	Task2: Done Task3: Done (After suggestions changes were made for concept note)
17.11.2024	1. Rubaya 2. Farhad 3. Shah	1. Suggestions about slide for progress presentation were given	Task1: All	Task2: Done

Date/Time/Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
08.12.24	1. Nafim 2. Farhad 3. Rubaya	First meeting after midterm, suggestions about project proposal was taken	All members were tasked	N/A
15.12.24	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1. Do chapter wise division 2. Suggestion on Slides	Task1: All Task2: All	N/A
22.12.24	1. Rubaya 2. Farhad 3. Nafim 4. Shah	Nothing much to specify, only feedback	N/A	Task1 and Task2: Done
29.12.24	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1. Book conference room for mock presentation	Task1: Rubaya	N/A
5.12.24	1. Rubaya 2. Farhad 3. Nafim 4. Shah	Mock Presentation was given. Some modification on slides and presentation suggestions were given	All	Tasl1: Done
8.12.24	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1. Day before final presentation, Mock presentation 2 was given and feedback from ATC was taken accordingly	N/A	Tasks: All done

FYDP-D

Date/Time/ Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
16.02.2025	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1.Initial Progress on Robotic Arm 2.Learning importance of torque calculation for the designing part	Task 1: All Task 2: Nafim, Farhad, Shah	1. Do proper literature review on the robotic arm 2. Incorporate the torque calculations while simulating
23.02.2025	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1. Initial 3-DOF Robotic Arm structure modeled in SolidWorks. Discussed mechanical specs.	Task 2: All Task 3: All	Review 3D model with torque constraints
2.03.2025	1. Rubaya 2. Farhad 3. Nafim	Nothing much to specify, only feedback	N/A	Task2: Done Task3:Done
9.03.2025	1. Rubaya 2. Nafim	Nothing much to specify, only feedback	N/A	Task2: Done Task3: Done
16.03.2025	1. Rubaya 2. Farhad 3. Shah	1. Suggestions about the ML model how it should do parallel work	Task1: All	1. Use multiple microcontrollers to each handle a segment of the work.
23.03.2025	1. Nafim 2. Farhad 3. Rubaya	First meeting after midterm, Updates on the robotic arm was shown	All members were tasked	Everything looks okay just do a bit more literature review for the torque computing segment
30.03.2025	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1. Showed the simulation of the navigation trail of the rover as well as the power and charge distribution	Task1: All Task2: All	1. The navigation was inefficient, better to optimize to follow a zigzag pattern

Date/Time/ Place	**Attendee**	**Summary of Meeting Minutes**	**Responsible**	**Comment by ATC**
6.04.2025	1. Rubaya 2. Farhad 3. Nafim 4. Shah	Nothing much to specify, only feedback	N/A	Task1 and Task2: Done
13.04.2025	1. Rubaya 2. Farhad 3. Nafim 4. Shah	Mock Presentation happened at Room 08B-07C	Task1: Rubaya	1. Add calculations/formulas to the original slide 2. Show power Consumed and energy usage over time. 3. If possible do a 3D modelling of the entire approach for better visualization
20.04.2025	1. Rubaya 2. Farhad 3. Nafim 4. Shah	Nothing much to specify, only feedback	N/A	Task1: Done
27.04.2025	1. Rubaya 2. Farhad 3. Nafim 4. Shah	Requested to take a mock presentation if possible before the final presentation	N/A	Will let you know through classroom portam
7.05.2025	1. Rubaya 2. Farhad 3. Nafim 4. Shah	1. Day before final presentation, Mock presentation 2 was given and feedback from ATC was taken accordingly	N/A	Tasks: All done

FYDP-C

Date	Attendees	Summary of Meeting Minutes	Responsibility	Comments by ATC
23.06.2025	1. Farhad 2. Nafim 3. Rubaya 4. Shah	1.Explained the construction process of the rover and the components being bought. 2. Explained the decision to buy an arm instead of building one and the potential issues that we might face.	Task-1 (Nafim): will talk to an expert to understand the robot making process. Task-2 (Farhad): will fix the arm model and servos.	1. Belal sir suggested using a clustering algorithm to resolve the one at a time pickup of the arm.
26.06.2025	1. Nafim 2. Shah	1. Discussed in detail about the type of arm required for the rover with a Research Assistant who had a similar project.	Task-1 (All): Utilize the websites in conjunction with Farhad's arm research.	N/A
29.06.2025	1. Farhad 2. Nafim 3. Rubaya 4. Shah	1. Farhad finalised the arm and servo motor choices.	Task-1 (All): Finalize the arm and rover design	N/A
30.06.2025	1. Farhad 2. Nafim 3. Rubaya	1. Finalized the arm design in conjunction with the rover design. 2. Shared updates with the rover's	Task-1 (All): Annotate the dataset in Roboflow.	N/A

	4. Shah	current state.		
Date	**Attendees**	**Summary of Meeting Minutes**	**Responsibility**	**Comments by ATC**
31.06.2025	1. Farhad 2. Nafim 3. Rubaya 4. Shah	1. Updated the dataset with new pictures and uploaded them to Roboflow.	Task-1 (All): Annotate the dataset in Roboflow.	N/A
05.07.2025	1. Farhad 2. Nafim 3. Rubaya 4. Shah	1. Dataset annotations done 2. High Torque DC motors chosen for the rover instead of the initial design which had Wiper motors. 3. Components bought.	Task-1 (Rubaya,Farhad): Further inspection of arm and its servos.	N/A
07.07.2025	1. Nafim 2. Rubaya 3. Shah	1. The arm was assembled and moved around a bit. 2. Rover base built and wheels attached.	Task-1 (Rubaya,Farhad): Working on arm and calibration Task-2 (Nafim,Shah): Run the rover to collect data	N/A
10.07.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. The rover was tested extensively. It could carry up to 10kg of mass. It also had issues moving in the BRACU terrace– resulting us in deciding to increase the wheel diameter. 2. The arm was moving as intended. The code needed some polishing to mitigate the offsets. Biasing issue still persists. The gripper was working but needed more refinement. 3. The machine learning model yolov8n was trained and from its results it was concluded that it needed more refinement.	Task-1 (Rubaya,Farhad): Working on arm and calibration and fixing the calibration issue. Task-2 (Nafim,Shah): Work on potential get around the wheel issue Task-3 (Nafim) : Refine the Machine Learning model	N/A
16.07.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. For the arm, the joystick modules were used to ensure smooth operation in between the servos. Two separate power supplies were used to power the arm as the arm consumed more current than what a single power supply could give out. Video documented the movement of the servos and the gripper and the arm overall. 2. Attached a wooden bar to the base of the rover so that it provides stability to the DC motors and we discovered that fixing the voltage to 12V would be disadvantageous as the DC motors require more than 12V to fix the movement issue.	Task-1 (Rubaya,Farhad): Work on fixing the biasing issue so that two power supplies are not needed and integrate a sensor that will cause the arm to move when waste is detected. Task-2 (Nafim,Shah): Work on the movement of the rover.	Comprehensive explanation of progress shown to ATC. Sir said to carry on and modify the specifications as needed.
20.07.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. The machine learning model of yolov8n was loaded onto the Raspberry Pi 5. The model was later converted into CNN model for better fps. It detects everything it sees so that	Task-1 (Rubaya,Farhad):Reconfigure the working of the arm so that one power supply is needed and start preparing	N/A

		is a potential issue. It detects things when mounted on the rover . 2. The arm's servo motors worked in tandem and the ultrasonic sensor was also integrated. It can do the predefined movement of reaching down to pick up and hold the waste before dispensing it. The jittering of the servos was fixed by adding a delay into the code, stabilizing the system.	for the integration with the rover and overall system. Task-2 (Nafim): Make sure that the ML model can properly detect the waste by properly defining the bounding boxes and confidence score. Task-3(Shah, Nafim):Work on the rover's movement .	
26.07.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. We got the YOLO detections on the Raspberry Pi 5 running much smoother by adding a timeout function that keeps the last three frames in memory. This really helped stabilize the detection process. But, when we tried storing more frames, the Pi started freezing up. It's clear we'll need to tweak the memory handling and frame management a bit more to keep things stable for real-time waste detection. 2. The robotic arm was acting a bit jittery at first, and the gripper motor wasn't strong enough to hold things reliably. We suspected the issue was weak PWM signals or just not enough motor power. To fix this, we ordered a DS3218 metal servo for the gripper and grabbed a PCA9685 external PWM driver. After some rewiring and a few code tweaks to work with the PCA9685, the jittering calmed down a lot. We also swapped in the new servo for the gripper, which helped, but the grip still isn't as solid as we'd like. It's going to need a bit more fine-tuning to get it moving smoothly and consistently. 3. We successfully hooked up the battery and charger to the system, and both the motors and the robotic arm are running great on battery power. Three voltage regulators are doing a solid job managing the current and voltage distribution. We also made some wiring upgrades by adding connectors to make the EPS power regulators easier to work with. The only hiccup came during mobility testing—the rover can move forward and backward just fine, but it won't turn left or right. Sounds like there's a bug in the movement control code that we'll need to sort out to get the rover navigating properly.	Task-1 (Rubaya,Farhad): Fine tune the arm with the new PCA9685 Task-2(Shah, Nafim):Work on the rover's movement, particularly the left and right turning. See the code for the ESP. Task-3(Nafim): Refine the ML model for the Pi 5.	N/A
30.07.2025	1. Nafim	1. We stabilized YOLO detections on	Task-1 (Farhad): Fix the	Belal sir mentioned

	2. Rubaya 3. Shah 4. Farhad	the Raspberry Pi 5 by capping memory at four frames, smoothing out jittery results. Still, the model struggles to classify crumpled paper as biodegradable. We're working on a queue-based system to process detections sequentially, helping the robotic arm handle waste items one at a time without sudden state changes. 2. The gripper continued to have issues with holding bottles of uneven shapes, and attempts to widen the mouth manually failed. Torque testing revealed that the shoulder and elbow motors could not lift even 500 g consistently, far below the theoretical 1 kg capacity. To address this, a high-torque TD8135MG servo was tested on the shoulder joint and proved capable of lifting heavier waste. The bin system was also initiated, with a servo able to rotate up to 2 kg, though a flywheel and long screw are needed for stability. 3. We fixed the rover's directional issues by correcting motor driver variables, enabling full movement. A battery test caused the LiPo to swell after dropping below 7 V, so we need a protection circuit to keep it above 12 V. Eight HC-SR04 ultrasonic sensors work on one ESP32 with shared triggers and external 5 V power. A zigzag path-planning algorithm is running, but without motor encoders, we're calibrating 90° turns through trial and error.	issue of the gripper being unable to hold uneven stuff. Task-2 (Nafim) : Finish the queue based algorithm of sorting waste Task-3 (Rubaya): Research and construct a protection circuit for the Lipo Battery Task-4(Shah, Nafim):Continue the movement of the rover calibration and all.	that we should increase the size of the wheels for better elevation from the ground.
06.08.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. The new high-torque servo didn't fix the slipping issue, so we switched back to the older setup with a weight limit and needed to measure its max capacity. We got the arm moving again by reinstalling the quad DS3218 servo setup and reviving the PCA9685 for smoother control. We added a rotating ultrasonic sensor on a servo to guide the arm's pickup sequence. After manually syncing the base and sensor angles, it worked like a charm, pushing the arm closer to semi-autonomous operation. 2. We mounted the ultrasonic sensors and improved the zigzag code, which now handles lane turns but needs calibration without encoders. We're using a measuring tape to figure out millisecond values for consistent travel and turns. Comparing boustrophedon	Task-1(Rubaya, Farhad): Work on fixing the gripper and whatever issues that may follow. Task-2(Shah, Nafim): Configure the hard coding of lanes for the rover to move.	N/A

		and lawnmower path strategies showed boustrophedon is more efficient but computationally heavier. Motor control was refined by sorting out RPWM/LPWM for directional speed and confirming R_EN/L_EN pins for current feedback are optional.		
10.08.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. We swapped in a new claw attachment with a sharper, wider grip, and it's a game-changer. The arm now easily picks up 250 ml and 500 ml bottles, handling cylindrical waste way better than the old gripper designs. 2. Got a GY-BNO055 9-axis IMU module and started calibration, planning to store the data on the ESP32. Calibration went well, but the heading values keep resetting inconsistently after each power cycle, which is a headache for reliable orientation tracking. 3. The waste bin is built and tested, showing solid progress in integration. Unfortunately, the PWM driver failed, halting both arm and bin development. It's a major setback, and we'll need replacement hardware to move forward.	Task-1(Nafim, Shah):Calibrate the ESP32 code so that the path planning can be complete. Task-2(Rubaya,Farhad): Get the new PWM driver and continue on the arm	N/A
14.08.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. Running the rover on grass was a mess—severe skidding during turns forced us to ditch the idea of using bicycle gears and chains as continuous tracks after they kept failing. Instead, we redesigned each of the four wheels to have its own chain cover and added hinges for better motor support. Despite the hiccups, we fully rewired the rover and mounted all components, including the arm and bin, getting it mostly ready for action. 2. To avoid more battery damage from over-discharge, we planned a protection circuit and ordered the parts. We also rewired and soldered the EPS system for better efficiency and integrated the PCA9665 PWM signal generator. These upgrades came together nicely, hitting a key milestone for the project's progress.	Task-1 (Shah): Start integration process of Pi 5 and arduino for the arm. Task-2 (All): Test the rover's movement	N/A
19.08.2025	1. Farhad 2. Rubaya 3. Shah	1. The waste segregation code on the Raspberry Pi 5 was glitchy at first, so we wrote a new script to control the bin servo based on detected waste types. It's a step forward but still rough and needs fine-tuning. Meanwhile, we started testing the	Task-1(Shah): Finish the pi and arduino integration so that the overall integration can be done Task-2(Rubaya): Construct the protection	N/A

		rover’s ultrasonic sensors, but both the ML pipeline and hardware integration are incomplete, with more debugging and calibration to go. 2. Tidied up the EPS system and motor driver with clean wiring, and initial tests with industrial wheels went smoothly. To boost stability and ease motor strain, we designed flywheels from stacked PVC boards. Early tests showed size mismatches and wobbling, especially on grass where the rover struggled to turn. Switching to thinner PVC layers helped a bit, but the rover still can’t handle varied terrains or uphill slopes as free wheels keep lifting the industrial ones off the ground. On hard surfaces, straight-line movement is solid, but rotation issues have us stuck in a loop.	circuit. Task-3(Farhad): Research more on the wheel and its turning failings.	
24.08.2025	1. Farhad 2. Rubaya 3. Shah	1. We tested gas cylinder pipes as potential tires, attaching them as “hot dog” segments or with scotch tape. The tape method won out, but the traction was so high it stopped the rover from moving on regular ground and the BRACU roof. After more tests, the rover moved and turned well on flat surfaces but couldn’t rotate on the futsal field, so we switched to tire tubes. Farhad worked with a cobbler to attach two sample tubes to the wheels. At the Mongoltori lab, we brainstormed and decided to weld a thin metal sheet to the existing wheel rims for a sturdier, wheel-like structure. 2. The bin system is done and dusted. A 3D-printed holder was made for the Pi camera, boosting the arm’s semi-autonomous capabilities. The Raspberry Pi’s waste segregation code is still a work in progress, but we’re preparing to integrate it with the Arduino and ESP for a seamless network. 3. We hit some snags with lost cables and limited workshop access, but kept pushing forward. The EPS system, wiring, and wheel tweaks are steadily improving. Plans like the metal sheet for wheels and tire tube setup show we’re tackling rover mobility and multi-terrain challenges step by step.	Task-1(Shah,Farhad): See to the persisting wheel issue. Task-2(Rubaya): Further research on potential PCB/ protection circuit. Task-3 (Shah): Implement final steps for integrating the Pi and the Uno	N/A
30.08.2025	1. Nafim 2. Rubaya 3. Shah	1. After tons of trial and error, the wheel system is finally complete. The wheels were finalized at the welding	Task-1(Nafim, farhad):Integrate arm and path planning with the Pi 5	N/A

	4. Farhad	shop. The rover cruised smoothly on the university grounds. It also nailed its performance on the futsal field, proving the wheels are ready for field tests. 2. Started working on the ultrasonic sensors, only to find many were faulty, so new replacements were ordered.New components and jumper wires arrived, letting us hook everything up efficiently for navigation. Initial tests flopped, but we got the arm's ultrasonic sensor fixed and mounted on the rover body, wrapping up the setup for navigation testing. 3. We made steady progress on integration, and the whole system is now portable and battery-powered. The arm's sensor had a brief hiccup but was sorted out after some troubleshooting, and the arm is good to go again. We also tidied up the wiring for the rover's body sensors, making everything neater and ready for more testing. Also, protection circuits are done and integrated.	Task-2 (Rubaya,Shah): Project Report updating	
5.09.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. Several tweaks were made with the overall integration of the arduino, the ESP and the Pi 5. The protection circuit was implemented as well. 2. Poster design research being done	Task (All): Take a proper video of all the components working separately and in tandem.	Showed Belal sir the working video of the rover. Sir mentioned that it might be a good idea to put a cone shape on the bin for better ease for the waste being dropped.
10.09.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. Video of all of the components working in tandem was taken. Also, the working of all the sub systems was showcased separately. 2. Brochure design finalized and poster printed	Task (All): Prepare for the Showcase and final presentation	N/A
11.09.2025	1. Nafim 2. Rubaya 3. Shah 4. Farhad	1. Successful showcase		

RELATED CODE/THEORY

Raspberry Pi5 Code

```
import cv2
from picamera2 import Picamera2
from ultralytics import YOLO
import numpy as np
from time import sleep
from collections import deque
import serial
import threading
import time

# Initialize the camera
picam2 = Picamera2()
picam2.preview_configuration.main.size = (720, 720)
picam2.preview_configuration.main.format = "RGB888"
picam2.preview_configuration.align()
picam2.configure("preview")
picam2.start()

# Load the YOLO model
model_path = "/home/craycrayfish451/balsal/best_ncnn_model"
model = YOLO(model_path)

# Initialize serial communication
try:
    arduino_uno = serial.Serial('/dev/ttyUSB0', 9600, timeout=1)  # Arduino UNO
    esp_device = serial.Serial('/dev/ttyUSB1', 115200, timeout=1)  # ESP
    sleep(0.1)  # Allow serial connections to stabilize
    print("Serial connections established")
except Exception as e:
    print(f"Serial connection error: {e}")
    arduino_uno = None
    esp_device = None

last_angle = None

def set_servo_angle(angle):
    global last_angle
    if angle != last_angle:
        # Send servo command to Arduino UNO via USB serial
        send_to_arduino(f"SERVO_CH7_{angle}")
        print(f"Sent servo command to Arduino UNO: Channel 7, Angle {angle}")
        last_angle = angle
        sleep(0)

def send_to_arduino(command):
    if arduino_uno:
        try:
            arduino_uno.write(f"{command}\n".encode())
            print(f"Sent to Arduino UNO: {command}")
        except Exception as e:
            print(f"Error sending to Arduino UNO: {e}")

def send_to_esp(command):
```

```
    if esp_device:
        try:
            esp_device.write(f"{command}\n".encode())
            print(f"Sent to ESP: {command}")
        except Exception as e:
            print(f"Error sending to ESP: {e}")

def read_arduino_responses():
    """Thread function to read responses from Arduino UNO"""
    while True:
        if arduino_uno and arduino_uno.in_waiting > 0:
            try:
                response = arduino_uno.readline().decode().strip()
                print(f"Arduino UNO response: {response}")

                # Forward pause/resume commands to ESP
                if response == "PAUSE_MOTORS":
                    send_to_esp("PAUSE")
                elif response == "RESUME_MOTORS":
                    send_to_esp("RESUME")
                elif response == "ARM_COMPLETE":
                    print("Arm pickup sequence completed")

            except Exception as e:
                print(f"Error reading Arduino UNO response: {e}")
        sleep(0)

def read_esp_responses():
    """Thread function to read responses from ESP"""
    while True:
        if esp_device and esp_device.in_waiting > 0:
            try:
                response = esp_device.readline().decode().strip()
                print(f"ESP response: {response}")

                # Forward ESP completion to Arduino
                if response == "ESP_COMPLETE":
                    send_to_arduino("ESP_COMPLETE")
                    print("ESP lawnmower pattern completed - notified Arduino")

            except Exception as e:
                print(f"Error reading ESP response: {e}")
        sleep(0)

# Start Arduino response reader thread
if arduino_uno:
    arduino_thread = threading.Thread(target=read_arduino_responses, daemon=True)
    arduino_thread.start()

# Start ESP response reader thread
if esp_device:
    esp_thread = threading.Thread(target=read_esp_responses, daemon=True)
    esp_thread.start()

# Send initial start command to ESP
send_to_esp("START")
print("Sent START command to ESP")
```

```
# Debounce counters
DEBOUNCE_FRAMES = 1
bio_counter = 0
non_bio_counter = 0
missing_counter = 0  # Track how long the current target has been missing

# Memory & queue
MAX_MEMORY = 5
memory = []
detection_queue = deque()

# Constants
FRAME_WIDTH, FRAME_HEIGHT = 720, 720
FRAME_AREA = FRAME_WIDTH * FRAME_HEIGHT
MAX_AREA_THRESHOLD = 0.2 * FRAME_AREA
CONFIDENCE_THRESHOLD = 0.4
FRAME_MARGIN = 20
MISSING_FRAMES_THRESHOLD = 3  # How many frames to wait before moving to next target

def iou(box1, box2):
    xA = max(box1[0], box2[0])
    yA = max(box1[1], box2[1])
    xB = min(box1[2], box2[2])
    yB = min(box1[3], box2[3])
    interArea = max(0, xB - xA) * max(0, yB - yA)
    box1Area = (box1[2] - box1[0]) * (box1[3] - box1[1])
    box2Area = (box2[2] - box2[0]) * (box2[3] - box2[1])
    union = box1Area + box2Area - interArea
    return interArea / union if union > 0 else 0

try:
    while True:
        frame = picam2.capture_array()
        results = model.predict(frame, conf=CONFIDENCE_THRESHOLD, iou=0.45)
        res = results[0]

        current_boxes = []
        current_detections = []

        for i, box in enumerate(res.boxes.xyxy.cpu().numpy()):
            conf = res.boxes.conf[i].item()
            cls = res.boxes.cls[i].item()
            class_name = res.names[int(cls)]
            x1, y1, x2, y2 = box
            box_area = (x2 - x1) * (y2 - y1)

            if (
                box_area < MAX_AREA_THRESHOLD and
                conf >= CONFIDENCE_THRESHOLD and
                x1 > FRAME_MARGIN and y1 > FRAME_MARGIN and
               x2 < FRAME_WIDTH - FRAME_MARGIN and y2 < FRAME_HEIGHT - FRAME_MARGIN
            ):
                current_boxes.append(box)
                current_detections.append({"class": class_name, "box": box})

        # Update detection queue
        for det in current_detections:
            if not any(iou(det["box"], q["box"]) > 0.5 for q in detection_queue):
```

```
                detection_queue.append(det)

        # Handle queue with missing frame tolerance
        if detection_queue:
            current_target = detection_queue[0]

            # Check if the current target is still in frame
              still_present = any(iou(current_target["box"], cb) > 0.5 for cb in
current_boxes)

            if still_present:
                missing_counter = 0
                if current_target["class"] == "Bio-degradable":
                    bio_counter += 1
                    non_bio_counter = 0
                    if bio_counter >= DEBOUNCE_FRAMES:
                        set_servo_angle(130)
                        # Trigger Arduino UNO arm
                        send_to_arduino("ARM_START")
                else:
                    non_bio_counter += 1
                    bio_counter = 0
                    if non_bio_counter >= DEBOUNCE_FRAMES:
                        set_servo_angle(0)
                        # Trigger Arduino UNO arm
                        send_to_arduino("ARM_START")
            else:
                missing_counter += 1
                if missing_counter >= MISSING_FRAMES_THRESHOLD:
                    detection_queue.popleft()  # Remove target
                    bio_counter = 0
                    non_bio_counter = 0
                    missing_counter = 0
        else:
            bio_counter = 0
            non_bio_counter = 0
            missing_counter = 0

        # Box memory tracking
        updated_memory = []
        for box in current_boxes:
            updated_memory.append({'box': box, 'age': 0})

        for mem in memory:
            if all(iou(mem['box'], box) < 0.5 for box in current_boxes):
                if mem['age'] < MAX_MEMORY:
                    mem['age'] += 1
                    updated_memory.append(mem)

        memory = updated_memory

        # Visualization
        annotated_frame = res.plot()
        for mem in memory:
            x1, y1, x2, y2 = map(int, mem['box'])
            cv2.rectangle(annotated_frame, (x1, y1), (x2, y2), (0, 255, 255), 2)

        if detection_queue:
```

```
            label = detection_queue[0]["class"]
            cv2.putText(annotated_frame, f"Target: {label}", (20, 40),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2, cv2.LINE_AA)

        inference_time = res.speed['inference']
        fps = 1000 / inference_time
        cv2.putText(annotated_frame, f'FPS: {fps:.1f}', (annotated_frame.shape[1]-
150, 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)

        cv2.imshow("Camera", annotated_frame)

        # ===== Queue Printout =====
        if detection_queue:
            print("\033[94m[QUEUE STATUS]\033[0m")
            for idx, item in enumerate(detection_queue):
                if idx == 0:
                    print(f"\033[92m -> Target: {item['class']} (Active)\033[0m")
                else:
                    print(f"    Waiting: {item['class']}")
        else:
            print("\033[91m[QUEUE EMPTY]\033[0m")

        if cv2.waitKey(1) == ord("q"):
            break

except KeyboardInterrupt:
    print("Shutting down...")
    # Send stop commands to both devices
    send_to_esp("STOP")
    send_to_arduino("STOP")
    print("Stop commands sent to ESP and Arduino UNO")
finally:
    cv2.destroyAllWindows()
    if arduino_uno:
        arduino_uno.close()
    if esp_device:
        esp_device.close()
```

ESP32 Code for Lawnmower Pattern

```
#include "BluetoothSerial.h"
#include <Arduino.h>

BluetoothSerial serialBT;
char BT;

// Motor pins
int R1PWM = 22;
int R2PWM = 12;
int L1PWM = 23;
int L2PWM = 14;

// Speed control variable (0–255)
int motorSpeed = 70; // Adjustable speed (start with a high value for testing)

// Lane-specific PWM values for turns (0–255)
```

```
const int pwmLeftTurn_lane1 = 210;  // PWM for left turn in Lane 1
const int pwmRightTurn_lane1 = 210; // PWM for right turn in Lane 1
const int pwmLeftTurn_lane2 = 245;  // PWM for left turn in Lane 2
const int pwmRightTurn_lane2 = 245; // PWM for right turn in Lane 2

// Timing constants (all in ms)
const int timeToMoveLength_lane1 = 6800; // Lane 1 forward movement
const int timeToMoveLength_lane2 =6000;  // Lane 2 forward movement
const int timeToMoveLane = 1800;          // Lane-changing movement

// Lane-specific turn durations
const int timeToTurnleft_lane1 = 2000;    // Lane 1 left turn
const int timeToTurnright_lane1 = 2000;   // Lane 1 right turn
const int timeToTurnleft_lane2 = 2450;    // Lane 2 left turn
const int timeToTurnright_lane2 = 4500;   // Lane 2 right turn

// Communication and state variables
bool systemStarted = false;
bool motorsPaused = false;
bool patternComplete = false;
String receivedCommand = "";

// Pattern state variables
int currentLane = 0;
int numLanes = 0;
bool movingUp = true;
bool laneMovementComplete = false;
unsigned long movementStartTime = 0;
unsigned long currentMovementDuration = 0;
enum MovementState {
  IDLE,
  MOVING_LENGTH,
  TURNING_FIRST,
  MOVING_LANE,
  TURNING_SECOND
};
MovementState currentState = IDLE;

void setup() {
  Serial.begin(115200);
  serialBT.begin("Baler Bhodai Jinish");  // Your BOT Name

  // Set up motor pins as outputs
  pinMode(R1PWM, OUTPUT);
  pinMode(R2PWM, OUTPUT);
  pinMode(L1PWM, OUTPUT);
  pinMode(L2PWM, OUTPUT);

  // Optional: Set PWM frequency (if supported by your ESP32 Arduino core)
  // analogWriteFrequency(1000); // 1000 Hz, uncomment if needed and supported

  numLanes = 2;

  Serial.println("ESP ready. Waiting for START command from Pi...");
  stop();  // Ensure motors are stopped initially
}

void loop() {
```

```
// Check for commands from Pi
if (Serial.available()) {
  receivedCommand = Serial.readStringUntil('\n');
  receivedCommand.trim();

  if (receivedCommand == "START") {
    systemStarted = true;
    motorsPaused = false;
    patternComplete = false;
    currentLane = 0;
    movingUp = true;
    currentState = MOVING_LENGTH;
    Serial.println("Received START command. Beginning lawnmower pattern...");
  }
  else if (receivedCommand == "PAUSE") {
    motorsPaused = true;
    stop();
    Serial.println("Motors PAUSED");
  }
  else if (receivedCommand == "RESUME") {
    motorsPaused = false;
    Serial.println("Motors RESUMED");
  }
  else if (receivedCommand == "STOP") {
    systemStarted = false;
    motorsPaused = true;
    patternComplete = true;
    stop();
    Serial.println("ESP STOPPED by Pi command");
  }
}

// Manual override via Bluetooth with speed control
while (serialBT.available()) {
  BT = serialBT.read();
  if (BT == 'F') {
    go_forward();
    Serial.println("Bluetooth: Moving forward at speed " + String(motorSpeed));
  }
  else if (BT == 'B') {
    go_backward();
    Serial.println("Bluetooth: Moving backward at speed " + String(motorSpeed));
  }
  else if (BT == 'L') {
    go_left(motorSpeed);  // Use default motorSpeed for manual turns
    Serial.println("Bluetooth: Turning left at speed " + String(motorSpeed));
  }
  else if (BT == 'R') {
    go_right(motorSpeed);  // Use default motorSpeed for manual turns
    Serial.println("Bluetooth: Turning right at speed " + String(motorSpeed));
  }
  else if (BT == 'S') {
    stop();
    Serial.println("Bluetooth: Stopped");
  }
}

// Execute lawnmower pattern if system started and not paused
```

```
  if (systemStarted && !motorsPaused && !patternComplete) {
    executeLawnmowerPattern();
  }
}

void executeLawnmowerPattern() {
  unsigned long currentTime = millis();

  switch (currentState) {
    case MOVING_LENGTH:
      if (movementStartTime == 0) {
        go_backward();
        if (currentLane == 0) currentMovementDuration = timeToMoveLength_lane1;
       else if (currentLane == 1) currentMovementDuration = timeToMoveLength_lane2;
        movementStartTime = currentTime;
      }

      if (currentTime - movementStartTime >= currentMovementDuration) {
        stop();
        delay(150);
        movementStartTime = 0;

        if (currentLane >= numLanes - 1) {
          patternComplete = true;
          currentState = IDLE;
          Serial.println("Lawnmower pattern complete!");
          Serial.println("ESP_COMPLETE");
        } else {
          currentState = TURNING_FIRST;
        }
      }
      break;

    case TURNING_FIRST:
      if (movementStartTime == 0) {
        if (movingUp) {
          int pwm;
          int turnDuration;
          if (currentLane == 0) {
            pwm = pwmLeftTurn_lane1;
            turnDuration = timeToTurnleft_lane1;
          } else {
            pwm = pwmLeftTurn_lane2;
            turnDuration = timeToTurnleft_lane2;
          }
          go_left(pwm);
          currentMovementDuration = turnDuration;
          movementStartTime = currentTime;
        } else {
          int pwm;
          int turnDuration;
          if (currentLane == 0) {
            pwm = pwmRightTurn_lane1;
            turnDuration = timeToTurnright_lane1;
          } else {
            pwm = pwmRightTurn_lane2;
            turnDuration = timeToTurnright_lane2;
          }
```

```
      go_right(pwm);
      currentMovementDuration = turnDuration;
      movementStartTime = currentTime;
    }
  }

  if (currentTime - movementStartTime >= currentMovementDuration) {
    stop();
    delay(150);
    movementStartTime = 0;
    currentState = MOVING_LANE;
  }
  break;

case MOVING_LANE:
  if (movementStartTime == 0) {
    go_backward();
    currentMovementDuration = timeToMoveLane;
    movementStartTime = currentTime;
  }

  if (currentTime - movementStartTime >= currentMovementDuration) {
    stop();
    delay(150);
    movementStartTime = 0;
    currentState = TURNING_SECOND;
  }
  break;

case TURNING_SECOND:
  if (movementStartTime == 0) {
    if (movingUp) {
      int pwm;
      int turnDuration;
      if (currentLane == 0) {
        pwm = pwmLeftTurn_lane1;
        turnDuration = timeToTurnleft_lane1;
      } else {
        pwm = pwmLeftTurn_lane2;
        turnDuration = timeToTurnleft_lane2;
      }
      go_left(pwm);
      currentMovementDuration = turnDuration;
      movementStartTime = currentTime;
    } else {
      int pwm;
      int turnDuration;
      if (currentLane == 0) {
        pwm = pwmRightTurn_lane1;
        turnDuration = timeToTurnright_lane1;
      } else {
        pwm = pwmRightTurn_lane2;
        turnDuration = timeToTurnright_lane2;
      }
      go_right(pwm);
      currentMovementDuration = turnDuration;
      movementStartTime = currentTime;
    }
```

```
      }

      if (currentTime - movementStartTime >= currentMovementDuration) {
        stop();
        delay(150);
        movementStartTime = 0;

        movingUp = !movingUp;
        currentLane++;
        currentState = MOVING_LENGTH;
      }
      break;

    case IDLE:
    default:
      break;
  }
}

// ——— MOTOR CONTROL HELPERS ———
void go_forward() {
  if (!motorsPaused) {
    analogWrite(R1PWM, motorSpeed);
    analogWrite(R2PWM, motorSpeed);
    analogWrite(L1PWM, 0);
    analogWrite(L2PWM, 0);
  }
}

void go_backward() {
  if (!motorsPaused) {
    analogWrite(R1PWM, 0);
    analogWrite(R2PWM, 0);
    analogWrite(L1PWM, motorSpeed);
    analogWrite(L2PWM, motorSpeed);
  }
}

void go_left(int pwm) {
  if (!motorsPaused) {
    analogWrite(R1PWM, 0);
    analogWrite(R2PWM, pwm);
    analogWrite(L1PWM, pwm);
    analogWrite(L2PWM, 0);
  }
}

void go_right(int pwm) {
  if (!motorsPaused) {
    analogWrite(R1PWM, pwm);
    analogWrite(R2PWM, 0);
    analogWrite(L1PWM, 0);
    analogWrite(L2PWM, pwm);
  }
}

void stop() {
  analogWrite(R1PWM, 0);
```

```
  analogWrite(R2PWM, 0);
  analogWrite(L1PWM, 0);
  analogWrite(L2PWM, 0);
}
```

Arduino Uno Code for Robotic Arm

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

// Initialize PCA9685
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

// PCA9685 channels
#define SENSOR_CHANNEL 14    // Servo for ultrasonic sensor
#define GRIPPER_CHANNEL 4    // Gripper servo
#define ELBOW_CHANNEL 9      // Elbow servo
#define SHOULDER_CHANNEL 10  // Shoulder servo
#define BASE_CHANNEL 15      // Base servo

// Servo pulse width calibration
#define SERVO_MIN 120   // ~0 degrees
#define SERVO_MAX 650   // ~270 degrees
#define GRIPPER_MIN 120 // ~0 degrees
#define GRIPPER_MAX 650 // ~270 degrees

// Servo angles
int gripper_degree = 0;   // Start closed
int y_axis_degree = 30;   // Elbow
int z_axis_degree = 150;  // Base
int clamp_degree = 80;   // Shoulder
int sensor_degree = 140;  // Sensor servo (start at center)
int pi_servo_degree = 130; // Pi servo on channel 7

// Initial angles
const int initialBase = 150;
const int initialShoulder = 80;
const int initialElbow = 30;
const int initialGripper = 0;
const int initialSensor = 90; // Sensor initial angle

// Ultrasonic pins
#define trigPin 9
#define echoPin 10

// Pick-up and throw parameters
const float pickUpDistance = 19.0; // cm
const int ElbowTargetAngle = 40;
const int ShoulderTargetAngle = 125;
const int baseThrowAngle = 150;
const int ElbowThrowAngle = 130;
const int ShoulderThrowAngle = 90;
const int servoDelay = 2000;
const int throwServoDelay = 2000;
const int gripperHoldDelay = 10000;
```

```
const int servoStep = 2;
const int stepDelay = 75;
const int sensorStepDelay = 0;
const int sensorMoveDelay = 0;
const int powerDelay = 1000;

// Scanning parameters
const int scanStep = 5;    // Degrees per scan step
const int scanDelay = 0;   // No delay per scan step

// State management
bool isPickingUp = false;
bool isThrowing = false;
bool isManualMode = false; // Manual control flag
bool armSequenceActive = false; // New flag for arm sequence
bool motorsPauseSent = false; // Track if pause command was sent
bool systemActive = true; // System active flag
unsigned long holdStartTime = 0;
unsigned long lastManualCommandTime = 0;
const unsigned long manualTimeout = 10000; // 10s timeout to resume auto
String receivedCommand = "";

void setup() {
  Serial.begin(9600);
  Serial.println("Arduino UNO Arm ready. Waiting for commands...");
  Serial.println("Enter commands like B90, S120, E110, G180, R90 for manual control,
or A for auto mode");

  Wire.begin();
  pwm.begin();
  pwm.setPWMFreq(50);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Set initial positions
  Serial.println("Moving to initial positions...");
  moveServo(BASE_CHANNEL, z_axis_degree, initialBase);
  delay(powerDelay);
  moveServo(ELBOW_CHANNEL, y_axis_degree, initialElbow);
  delay(powerDelay);
  moveServo(SHOULDER_CHANNEL, clamp_degree, initialShoulder);
  delay(powerDelay);
  moveServo(GRIPPER_CHANNEL, gripper_degree, initialGripper);
  delay(powerDelay);
  moveServo(SENSOR_CHANNEL, sensor_degree, initialSensor); // Center sensor
  delay(sensorMoveDelay);

  delay(500);
}

int angleToPulse(int angle, bool isGripper = false) {
  if (angle < 0) angle = 0;
  if (angle > 270) angle = 270;
  int minPulse = isGripper ? GRIPPER_MIN : SERVO_MIN;
  int maxPulse = isGripper ? GRIPPER_MAX : SERVO_MAX;
  return map(angle, 0, 270, minPulse, maxPulse);
}
```

```
void setServoAngle(uint8_t channel, int angle) {
  int pulse = angleToPulse(angle, channel == GRIPPER_CHANNEL);
  pwm.setPWM(channel, 0, pulse);
  delay(10);
}

void moveServo(uint8_t channel, int &currentAngle, int targetAngle) {
  Serial.print("Moving channel ");
  Serial.print(channel);
  Serial.print(" from ");
  Serial.print(currentAngle);
  Serial.print(" to ");
  Serial.println(targetAngle);
  int step = (targetAngle > currentAngle) ? servoStep : -servoStep;
  while (abs(targetAngle - currentAngle) >= servoStep) {
    currentAngle += step;
    setServoAngle(channel, currentAngle);
    delay(channel == SENSOR_CHANNEL ? sensorStepDelay : stepDelay); // No delay for
sensor
  }
  currentAngle = targetAngle;
  setServoAngle(channel, currentAngle);
}

float measureDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH, 30000);
  float distance = duration * 0.034 / 2;
  if (distance <= 0 || distance > 100) return -1; // Filter invalid readings
  return distance;
}

bool scanEnvironment(int &detectedAngle) {
  // Forward sweep: 90 to 200 degrees
  for (int angle = 90; angle <= 150; angle += scanStep) {
    moveServo(SENSOR_CHANNEL, sensor_degree, angle);
    delay(scanDelay);
    float distance = measureDistance();
    if (distance > 0 && distance <= pickUpDistance) {
      Serial.print("Object detected at angle: ");
      Serial.print(angle);
      Serial.print(" deg, distance: ");
      Serial.print(distance);
      Serial.println(" cm");
      detectedAngle = angle;
      if (!motorsPauseSent) {
        Serial.println("PAUSE_MOTORS");
        motorsPauseSent = true;
      }
      return true;
    }
  }
  // Reverse sweep: 200 to 90 degrees
```

```
  for (int angle = 150; angle >= 90; angle -= scanStep) {
    moveServo(SENSOR_CHANNEL, sensor_degree, angle);
    delay(scanDelay);
    float distance = measureDistance();
    if (distance > 0 && distance <= pickUpDistance) {
      Serial.print("Object detected at angle: ");
      Serial.print(angle);
      Serial.print(" deg, distance: ");
      Serial.print(distance);
      Serial.println(" cm");
      detectedAngle = angle;
      if (!motorsPauseSent) {
        Serial.println("PAUSE_MOTORS");
        motorsPauseSent = true;
      }
      return true;
    }
  }
  return false; // No object detected, continue scanning
}

bool parseSerialCommand() {
  if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n');
    input.trim();

    // Check for Pi commands
    if (input == "ARM_START") {
      if (!armSequenceActive && systemActive) {
        armSequenceActive = true;
        isPickingUp = true;
        motorsPauseSent = false;
        Serial.println("Received ARM_START from Pi. Starting arm sequence...");
      }
      return false; // Don't enter manual mode
    }

    // Check for ESP completion command
    if (input == "ESP_COMPLETE") {
      systemActive = false;
      isPickingUp = false;
      isThrowing = false;
      armSequenceActive = false;
      Serial.println("Received ESP_COMPLETE. Arduino UNO stopping...");
      return false;
    }

    // Check for Pi stop command
    if (input == "STOP") {
      systemActive = false;
      isPickingUp = false;
      isThrowing = false;
      armSequenceActive = false;
      Serial.println("Received STOP from Pi. Arduino UNO stopping...");
      return false;
    }

    // Check for Pi servo commands (SERVO_CH7_angle)
```

```
    if (input.startsWith("SERVO_CH7_")) {
      int angle = input.substring(10).toInt(); // Extract angle after "SERVO_CH7_"
      if (angle >= 0 && angle <= 270 && systemActive) {
        moveServo(7, pi_servo_degree, angle); // Control channel 7 on PCA9685
        Serial.print("Pi servo command executed: Channel 7, Angle ");
        Serial.println(angle);
      }
      return false; // Don't enter manual mode
    }

    if (input.length() < 2) return false;

    char servo = input[0];
    int angle = input.substring(1).toInt();

    if (angle < 0 || angle > 270) {
      Serial.println("Invalid angle. Use 0-270.");
      return false;
    }

    switch (servo) {
      case 'B':
        moveServo(BASE_CHANNEL, z_axis_degree, angle);
        Serial.println("Base moved manually");
        break;
      case 'S':
        moveServo(SHOULDER_CHANNEL, clamp_degree, angle);
        Serial.println("Shoulder moved manually");
        break;
      case 'E':
        moveServo(ELBOW_CHANNEL, y_axis_degree, angle);
        Serial.println("Elbow moved manually");
        break;
      case 'G':
        moveServo(GRIPPER_CHANNEL, gripper_degree, angle);
        Serial.println("Gripper moved manually");
        break;
      case 'R':
        moveServo(SENSOR_CHANNEL, sensor_degree, angle);
        Serial.println("Sensor moved manually");
        break;
      case 'A':
        isManualMode = false;
        Serial.println("Switched to auto mode");
        return false;
      default:
       Serial.println("Invalid command. Use B[angle], S[angle], E[angle], G[angle],
R[angle], or A");
        return false;
    }
    isManualMode = true;
    lastManualCommandTime = millis();
    return true;
  }
  return false;
}

void loop() {
```

```
  // Check for manual commands or Pi commands
  if (parseSerialCommand()) {
    // Manual command processed, skip auto sequence
    setServoAngle(BASE_CHANNEL, z_axis_degree);
    setServoAngle(ELBOW_CHANNEL, y_axis_degree);
    setServoAngle(SHOULDER_CHANNEL, clamp_degree);
    setServoAngle(GRIPPER_CHANNEL, gripper_degree);
    setServoAngle(SENSOR_CHANNEL, sensor_degree);
  } else if (isManualMode && (millis() - lastManualCommandTime >= manualTimeout))
{
    Serial.println("Manual mode timeout. Resuming auto mode.");
    isManualMode = false;
  }

  // Auto mode: Lock idle servos
  if (!isManualMode && !armSequenceActive) {
    setServoAngle(BASE_CHANNEL, z_axis_degree);
    setServoAngle(ELBOW_CHANNEL, y_axis_degree);
    setServoAngle(SHOULDER_CHANNEL, clamp_degree);
    setServoAngle(GRIPPER_CHANNEL, gripper_degree);
    setServoAngle(SENSOR_CHANNEL, sensor_degree);
  }

  // Auto mode: Scan environment and execute sequence (only when arm sequence is
active)
  if (!isManualMode && armSequenceActive && isPickingUp && !isThrowing &&
systemActive) {
    int detectedAngle = 0;
    if (scanEnvironment(detectedAngle)) {
      isPickingUp = false;
      isThrowing = true;
      // Map sensor angle to base angle using the equation: base = (45 * sensor /
100) + 84
      int baseTarget = (45 * detectedAngle / 100) + 84;
      Serial.print("Starting pick-up sequence at base angle: ");
      Serial.println(baseTarget);
      moveServo(BASE_CHANNEL, z_axis_degree, baseTarget); // Align base to mapped
sensor angle
      delay(servoDelay);
      setServoAngle(BASE_CHANNEL, z_axis_degree);
      moveServo(ELBOW_CHANNEL, y_axis_degree, ElbowTargetAngle);
      delay(servoDelay);
      setServoAngle(ELBOW_CHANNEL, y_axis_degree);
      moveServo(SHOULDER_CHANNEL, clamp_degree, ShoulderTargetAngle);
      delay(servoDelay);
      setServoAngle(SHOULDER_CHANNEL, clamp_degree);
      Serial.println("Gripper opening");
      if (gripper_degree != 0) moveServo(GRIPPER_CHANNEL, gripper_degree, 0); //
Ensure open
      delay(servoDelay);
      setServoAngle(GRIPPER_CHANNEL, gripper_degree);
      Serial.println("Gripper closing");
      moveServo(GRIPPER_CHANNEL, gripper_degree, 180);
      delay(servoDelay);
      setServoAngle(GRIPPER_CHANNEL, gripper_degree);
      holdStartTime = millis();
      Serial.println("Starting throw sequence");
    }
```

```
  }

  // Handle throw sequence
  if (!isManualMode && armSequenceActive && isThrowing && millis() - holdStartTime
< gripperHoldDelay && systemActive) {
    moveServo(BASE_CHANNEL, z_axis_degree, baseThrowAngle);
    delay(throwServoDelay);
    setServoAngle(BASE_CHANNEL, z_axis_degree);
    moveServo(ELBOW_CHANNEL, y_axis_degree, ElbowThrowAngle);
    delay(throwServoDelay);
    setServoAngle(ELBOW_CHANNEL, y_axis_degree);
    moveServo(SHOULDER_CHANNEL, clamp_degree, ShoulderThrowAngle);
    delay(throwServoDelay);
    setServoAngle(SHOULDER_CHANNEL, clamp_degree);
  }

  // Check if holding and 10 seconds have passed
  if (!isManualMode && armSequenceActive && isThrowing && millis() - holdStartTime
>= gripperHoldDelay && systemActive) {
    Serial.println("Gripper opening");
    moveServo(GRIPPER_CHANNEL, gripper_degree, 0);
    delay(servoDelay);
    setServoAngle(GRIPPER_CHANNEL, gripper_degree);
    isThrowing = false;
    armSequenceActive = false;
    Serial.println("Throw sequence complete");
    Serial.println("Returning to initial positions...");
    moveServo(BASE_CHANNEL, z_axis_degree, initialBase);
    delay(servoDelay);
    setServoAngle(BASE_CHANNEL, z_axis_degree);
    moveServo(ELBOW_CHANNEL, y_axis_degree, initialElbow);
    delay(servoDelay);
    setServoAngle(ELBOW_CHANNEL, y_axis_degree);
    moveServo(SHOULDER_CHANNEL, clamp_degree, initialShoulder);
    delay(servoDelay);
    setServoAngle(SHOULDER_CHANNEL, clamp_degree);
    moveServo(GRIPPER_CHANNEL, gripper_degree, initialGripper);
    delay(servoDelay);
    setServoAngle(GRIPPER_CHANNEL, gripper_degree);
    moveServo(SENSOR_CHANNEL, sensor_degree, initialSensor); // Return sensor to
center
    delay(sensorMoveDelay);
    setServoAngle(SENSOR_CHANNEL, sensor_degree);

    // Send resume command to Pi
    Serial.println("RESUME_MOTORS");
    Serial.println("ARM_COMPLETE");
  }

  // Debugging output (reduced frequency to avoid spam)
  static unsigned long lastDebugTime = 0;
  if (millis() - lastDebugTime > 1000) { // Print debug info every 1 second
    Serial.print("z_axis_degree: ");
    Serial.print(z_axis_degree);
    Serial.print(", y_axis_degree: ");
    Serial.print(y_axis_degree);
    Serial.print(", clamp_degree: ");
    Serial.print(clamp_degree);
```

```
    Serial.print(", gripper_degree: ");
    Serial.print(gripper_degree);
    Serial.print(", sensor_degree: ");
    Serial.print(sensor_degree);
    Serial.print(", isPickingUp: ");
    Serial.print(isPickingUp ? "true" : "false");
    Serial.print(", isThrowing: ");
    Serial.print(isThrowing ? "true" : "false");
    Serial.print(", isManualMode: ");
    Serial.print(isManualMode ? "true" : "false");
    Serial.print(", systemActive: ");
    Serial.println(systemActive ? "true" : "false");
    lastDebugTime = millis();
  }

  delay(50);
}
```

ASSESSMENT GUIDELINE FOR FACULTY

[The following assessment guideline is for faculty ONLY. **This portion is not applicable for students**.]

Assessment Tools and CO Assessment Guideline

	Distribution of assessment points among various COs assessed in different semesters														
PO	l	c	f	g	c	b	d	c	e	l	k	k	h	i	j
CO	CO 1	CO 2	CO 3	CO 4	CO 5	CO 6	CO 7	CO 8	CO 9	CO 10	CO 11	CO 12	CO 13	CO 14	CO 15
EEE 400C/ ECE 402C (Out of 100)							30	24	6	4	4	6	7	7	12
Project Final Report/ Project Progress Report							x	x	x	x	x	x	x		x
Demonstration of working prototype							x								x
Progress Presentation/ Final Presentation								x			x				
Peer-evaluation*													x	x	
Instructor's Assessment*													x	x	
Demonstration at FYDP Showcase								x							x

Note: The star (*) marked deliverables/skills will be evaluated at various stages of the project.

Mapping of CO-PO-Taxonomy Domain & Level- Delivery-Assessment Tool

Sl.	CO Description	PO	Bloom's Taxonomy Domain/Level	Assessment Tools
CO7	**Evaluate** the performance of the developed solution with respect to the given specifications, requirements and standards	d	Cognitive/ Evaluate	• Demonstration of working prototype • Project Progress Report on working prototype
CO8	**Complete** the final design and development of the solution with necessary adjustment based on performance evaluation	c	Cognitive/ Create	• Project Final Report • Final Presentation • Demonstration at FYDP Showcase
CO9	**Use** modern engineering and IT tools to design, develop and validate the solution	e	Cognitive/ Understand, Psychomotor/ Precision	• Project Final Report
CO10	**Conduct** independent research, literature survey and learning of new technologies and concepts as appropriate to design, develop and validate the solution	l	Cognitive/ Apply	• Project Final Report
CO11**	**Demonstrate** project management skill in various stages of developing the solution of engineering design project	k	Cognitive/ Apply Affective/ Valuing	• Project Final Report • Project Progress presentation at various stages
CO12	**Perform** cost-benefit and economic analysis of the solution	k	Cognitive/ Apply	• Project Final Report
CO13	**Apply** ethical considerations and professional responsibilities in designing the solution and throughout the project development phases	h	Cognitive/ Apply Affective/ Valuing	• Peer-evaluation, • Instructor's Assessment • Final Report

CO14**	**Perform** effectively as an individual and as a team member for successfully completion of the project	i	Affective/ Characterization	• Peer-evaluation • Instructor's Assessment
CO15**	**Communicate** effectively through writings, journals, technical reports, deliverables, presentations and verbal communication as appropriate at various stages of project development	j	Cognitive/ Understand Psychomotor/ Precision Affective/ Valuing	• Project Final Report • Progress Presentations, • Final Presentation • Demonstration at FYDP Showcase

Note: The double star (**) marked CO will be assessed at various stages of the project through indirect deliverables.