



BeyondTech

Long Thin Hauler

Version 2.4.9

01.05.2025

Presented by:

Ahmet Eren Cengiz

Furkan Çitil

Furkan Karatoprak

Kamil Anıl Işık

Tolga Demirdal

Umut Gürbüz

Project Start Date:

04.10.2023

Project End Date:

10.05.2024

Table of Contents

Table of Contents	3
Executive Summary	4
Introduction	5
Background of the project.....	5
Problem Statement	5
Current Status of the Project	6
Widespread Application and Impacts on Society	6
Potential Effects of the Long Thin Hauler in Widespread Use.....	7
Scope and Organization	7
Design Description	9
Mechanical Subsystem	14
Chassis Design	14
Mobility and Drive Mechanism	16
Sensor and Actuator Placement.....	17
Parking Area Design	21
Power Subsystem	23
Visual Positioning Subsystem	23
Control Subsystem	29
State Machine	32
Path Generation.....	35
Communication Subsystem	37
Sensor Subsystem	39
Madgwick Filter.....	39
Kalman Filter	41
Design Requirements.....	44
System Requirements	44
Subsystem Requirements	45
Mechanical Subsystem	45
Power Subsystem	45
Visual Positioning Subsystem.....	45
Control Subsystem.....	46
Communication Subsystem.....	46
Sensor Subsystem	46
Compatibility Analysis.....	47
Transport Agnostic Communication Protocol	47
Packet Structure	47
Payload Organization	47
Modularity and Flexibility	47
Simulation and Testing	48

Power Management	50
Battery and Power Distribution	50
Voltage Regulators	51
Modularity and Robustness	51
Compliance with Standards	51
Compliance with Requirements	52
Detailed Compliance with Requirements Analysis	52
Robustness of the System Against Possible Error Sources	55
Safety Issues and Precautions	56
Test Results	57
Visual Positioning	57
Wireless	59
Camera Vibration & Exposure	63
IMU Attitude Test	65
Path planning	67
Power Test at Full Power	68
Battery Failsafe	68
Parking Maneuver	69
Final Mission	70
List of Deliverables	75
Resource Management	76
Cost Analysis	76
Power Management	78
Conclusion	83
Appendix 1: Gantt Chart	85
Appendix 2: Test Document: Visual Positioning	86
Appendix 3: Test Document: Wireless	113
Appendix 4: Test Document: Camera Vibration & Exposure	155
Appendix 5: Test Document: IMU Attitude	161
Appendix 6: Test Document: Path planning	167
Appendix 7: Test Document: Power Test at Full Power	170
Appendix 8: Test Document: Battery Failsafe Test	174
Appendix 9: Test Document: Parking Maneuver Test	178
Appendix 10: Test Document: Final Mission Test	181
Appendix 11: User Manual	186
Package Contents	186
Quick Start Guide	187

Vehicle Assembly	187
Assembled Vehicle	190
Vehicle Software	193
Vehicle Operation	194
Trouble Shooting	195
Appendix 12: Poster of the Project	199

Executive Summary

BeyondTech has started an innovative project to address the growing need for efficient parking solutions in urban settings. Our project, the "Long Thin Hauler," is a cutting-edge autonomous vehicle designed to navigate and self-park within a constrained environment. As final-year Electrical Electronics Engineering students, this project is a reflection of our academic proficiency and practical skills aimed at simplifying parking.

The objective of the Long Thin Hauler project is to create an autonomous ground vehicle optimized for accurate parking within a narrow designated area in a 3m by 3m space, regardless of its starting position and orientation.

Our vehicle is capable of autonomous forward motion and rear-wheel drive, utilizing wireless technologies, visual-based algorithms, and sensor fusion for precise self-parking maneuvers. It consists of a front section with the controller, camera, and communication modules, and a rear section with the power and mechanical subsystems, communicating wirelessly through a metal hollow tube.

The project consists of six main subsystems: Power, Mechanics, Sensors, Visual Positioning, Control, and Communication subsystems. The Power subsystem supplies power to all other subsystems. The Mechanics subsystem maintains the vehicle's structure and enables it to move within the design constraints. The Sensors subsystem integrates IMU data and localization feedback using fiducial markers. The Visual Positioning subsystem determines the vehicle's position and orientation within a designated driving area using a camera module. The Control subsystem precisely controls DC motors to ensure smooth operation. The Communication subsystem enables wireless communication between the front and rear sections of the vehicle, which are 30cm apart.

Our team paid attention to how the longthin vehicle aligns with predefined guidelines regarding its functionality, performance, and structural integrity. In that way, decisions and design choices are made, and trade-offs are taken into consideration. The main requirements of the include the ability to autonomously navigate and park within a 3m by 3m space in 60 seconds and communicate wirelessly between the front and rear sections through a metal hollow tube.

The design of our vehicle ensures optimal performance by promoting seamless integration and compatibility between subsystems, allowing for independent testing and validation of each subsystem and facilitating easy component replacement and various simulation tests. Additionally, the design of the vehicle complies with common IEEE standards like IEEE 802.11.

According to our test results, the vehicle incorporates a robust sensor fusion algorithm that combines IMU data with visual feedback to enhance navigation accuracy. Utilizing advanced 3D printing techniques and a sophisticated mechanical design, the vehicle is ready to navigate and park within the designated area. The visual positioning system, optimized to detect symmetrically positioned ArUco markers, enables the vehicle to determine its precise position and orientation. Integration of a camera sensor for marker detection, an IMU sensor, and the implementation of a Kalman filter has further enhanced the precision and reliability of the localization process. The rear-wheel system has been refined to ensure precise motor control, and the controller has been finalized, promising seamless maneuvering and parking execution. Whole system have been successfully tested and the project has achieved its vision of a fully autonomous parking solution, demonstrating strong awareness and accurate positioning and parking within the environment.

Team BeyondTech successfully managed to produce the Long Thin Hauler within the allocated budget of \$300, achieving high-quality performance while staying within financial constraints. The vehicle's front part consumes approximately 7 W of power, which is supplied by a 2S lithium-ion battery. Similarly, the rear part consumes around 7.7 W of power and is powered by a 4S lithium-ion battery pack.

Introduction

Background of the project

Self-driving vehicles have revolutionized transportation by enabling vehicles to navigate and deliver their loads autonomously, without human intervention. These vehicles have been designed to transport passengers, goods, and materials, and have been used in various industries, including logistics, agriculture, and construction, which require the transportation of heavy loads.

Autonomous vehicles designed for transporting goods or materials have the potential to reduce traffic congestion, eliminate driver errors, save time in cargo transportation, and increase parking efficiency by being able to park themselves in less space.

However, before achieving these benefits, long transportation vehicles require optimized solutions for autonomous localization, parking spot detection, and accurate parking, despite their limited movement capabilities due to heavy loads. Fortunately, designing a vehicle to meet these requirements is achievable.

Problem Statement

The objective of this project, namely "Long Thin Hauler," is to design and produce a model of an autonomous land vehicle tailored for a specific task: flawless and precise parking within a randomly defined narrow parking spot extruded from a 3m by 3m area. The vehicle must be able to locate itself and detect the parking spot from a randomized starting position and orientation. Then, it should park correctly in a predetermined spot. Moreover, this project requires the following conditions to be met:

- **Chassis Requirement:** The chassis of the designed vehicle should include a metallic or metal-covered hollow tube of at least 50 cm long.
- **Vehicle Dimension:** The width of the vehicle is limited between 3 cm and 10 cm.
- **Motion Limitation:** The vehicle should move flawlessly but only in the forward direction. Any kind of backward motion is considered a violation.
- **Sensors and Actuator Placement:** Sensors should be placed on the front side of the vehicle, and actuators should be on the rear side.
- **Distance Requirement For Sensors And Motion Units:** Between motion units and sensors, there should be at least a 30 cm distance.
- **Wireless Communication:** Communication between motion units and sensors should be done wirelessly through the chassis of the vehicle over a minimum distance of 30 cm.
- **Parking Spot Dimension:** The width of the parking spot, extruded from the 3 m x 3 m area, should not exceed 1.5 times the width of the vehicle.
- **Time Limitation:** The vehicle should reach the parking spot within 1 minute after it is randomly located in the drivable area.
- **Area Restriction:** There should be nothing inside a 3 m x 3 m area; if desired, only edges and corners are allowed to be used for placing external reference points for the vehicle.
- **Parking Spot Location:** The parking spot should be designed so that it can easily be relocated within any extrusion of the 3 m x 3 m area.

The development and production of an autonomous land vehicle that is capable of mapping, localization, and precise parking in limited spaces brought engineering challenges to compare the requirements mentioned above across multiple domains. From a technical point of view, several problems needed to be solved to achieve the successful creation of this product.

- **Mechanical Design:** Designing a compact chassis housing a metallic or metal-covered hollow tube of at least 50 cm in length while adhering to vehicle width constraints (between 3 cm and 10 cm) posed a considerable engineering challenge. Additionally, motor and sensor placement requirements caused non-flexible mechanical design options.
- **Power Subsystems:** To achieve flexible maneuverability, the vehicle required motors with high torque and batteries that are able to power these motors. Additionally, the power system required elaborate analysis.
- **Communication System:** Establishing reliable wireless communication through a metallic hollow tube between front sensors and rear actuators presented a technical problem. Since vehicles utilized real-time data, this communication process needed to be completed flawlessly or with minimal errors.
- **Sensor Fusion:** Integrating and effectively utilizing various sensors, including inertial sensors (IMU) and the Raspberry Pi Camera module v3, presented a challenge. The task involved fusing data from different sensors to achieve accurate localization, parking spot detection, and flawless motion.
- **Visual Positioning:** Achieving precise localization and navigation within a limited 3m x 3m area for the autonomous vehicle brought engineering challenges. The vehicle had to identify its initial position, map the environment, and generate a parking spot trajectory with high accuracy.
- **Area Design:** Designing a 3m x 3m area layout with a relocatable parking spot and markers had some design challenges. Additionally, the area itself had to be designed as flat and sufficiently illuminated. Marker locations needed to be fixed only at the edges and corners.

Current Status of the Project

With the project now completed, a significant milestone has been achieved in the realm of autonomous parking systems. The vehicle, constructed using advanced 3D printing techniques and incorporating a sophisticated mechanical design, stands ready to navigate and park within the designated area. Utilizing ArUco markers symmetrically positioned around the area for localization, along with a camera sensor for marker detection and location determination, the system has strong awareness capabilities. Integration of an IMU sensor and the implementation of a Kalman filter have further enhanced the precision and reliability of the localization process, ensuring accurate positioning within the environment. The controller has been finalized and it promises seamless maneuvering and parking execution. Through rigorous testing of individual subsystems and comprehensive integration efforts, the project has realized its vision of a fully autonomous parking solution.

Widespread Application and Impacts on Society

A possible widespread application of the designed autonomous long thin hauler is in automated warehouse logistics, particularly for narrow aisle navigation and precise inventory management. By streamlining parking, this technology can reduce the time drivers spend looking for parking spaces, easing traffic congestion and saving fuel, which benefits both users and the environment. Additionally, it can provide a convenient parking solution for those with mobility issues, making transportation more accessible. Automating parking processes can improve the utilization of available spaces, enhance safety by reducing parking-related accidents, and influence urban planning by reducing the need for extensive parking lots. On a larger scale, this project could create job opportunities in the tech and manufacturing sectors, ultimately contributing to smarter, more efficient transportation and positively impacting daily life and urban infrastructure.

Potential Effects of the Long Thin Hauler in Widespread Use

The widespread adoption of the Long Thin Hauler presents numerous potential benefits and challenges, fundamentally altering urban logistics and transportation dynamics. A primary advantage is the substantial improvement in parking efficiency for both personal and commercial vehicles. This autonomous system reduces the time spent searching for parking, thereby decreasing traffic congestion and lowering emissions from idling cars, contributing positively to environmental sustainability. Enhanced parking management can also lead to better utilization of urban spaces, potentially freeing up areas for green spaces or additional infrastructure. However, the integration of such advanced technology may require significant initial investments in infrastructure upgrades, widespread marker placements, and system maintenance. There is also the challenge of ensuring the system's resilience against cyber threats, as increased connectivity might expose vulnerabilities. Additionally, the shift towards autonomous systems could impact employment within the parking and vehicle management sectors, necessitating measures to support affected workers. On balance, while the deployment of the Long Thin Hauler promises to streamline urban transport logistics and environmental benefits, careful consideration of infrastructure demands, cybersecurity, and socio-economic impacts is essential for a smooth transition.

Scope and Organization

This critical design review report for the Long Thin Hauler project is structured to offer a comprehensive and clear understanding of our work and its implications. The report is designed to be accessible to both technical experts and stakeholders with a general interest in the project.

The report is organized into the following key sections:

1. **Executive Summary:** This section presents a brief and impactful overview of the entire project. It highlights the main objectives, innovative features, and potential impacts of the Long Thin Hauler, serving as a snapshot of the project's scope and significance.
2. **Introduction:** Here, we provide the context and background of the project. This section includes a clear statement of our project, the current status of the work, and the scope and organization of the report.
3. **System Design:** This section provides an in-depth look at the overall design of the "Long Thin Hauler," including the integration of the 2.4GHz communication band and visual-based algorithms. It covers the design approach and specifics of subsystems such as navigation, sensor integration, power management, and control mechanisms, explaining their individual contributions to the vehicle's functionality and their interaction for efficient and autonomous parking capabilities.
4. **Design Requirements:** This section presents the final system and subsystem requirements with justifications in terms of literature, test results, and engineering insights.
5. **Design Modifications:** This section clearly shows and justifies system and subsystem level modifications that have been made since the conceptual design report.
6. **Compatibility Analysis:** This section includes a detailed analysis of the compatibility of the subsystems.
7. **Compliance with Requirement:** This section explains reasons behind design decisions and modifications by providing justification according to requirements.
8. **Test Results:** This section is dedicated to presenting the findings from our tests. It discusses how these results have influenced the project's direction and any design modifications made in response to these outcomes.
9. **List of Deliverables:** This section gives list of deliverable materials of the project.
10. **Resource Management:** This section provides information about resource management in terms of cost, power management and schedule.

11. **User Manual:** This section provides a comprehensive guide for users of the Long Thin Hauler. It includes detailed instructions on the operation, maintenance, and troubleshooting of the vehicle. The user manual is designed to ensure that users can effectively and safely interact with the Long Thin Hauler, maximizing its functionality and longevity. It covers everything from initial setup and configuration to routine maintenance procedures, safety protocols, and common issues that may arise, along with their solutions. This section aims to be a thorough resource for both new users and experienced operators, ensuring a smooth and efficient experience with the Long Thin Hauler.
12. **Conclusion:** The concluding section summarizes the report, provides closing remarks related to analysis of "Long Thin Hauler" project.

Design Description

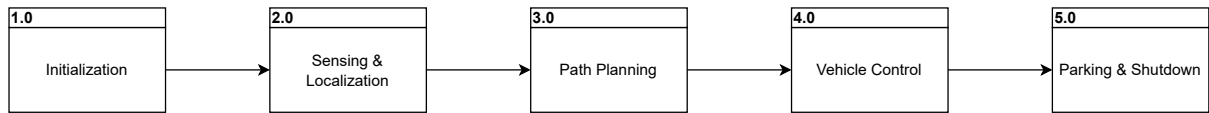
The Long Thin Hauler, designed by the BeyondTech team, is a new step in automated land vehicle design. Our system, intended for forward motion with rear-wheel drive and self-parking capabilities, is sectioned into six fundamental subsystems: Power, Mechanics, Sensor, Visual Positioning, Controller, and Communication.

The mechanical subsystem is designed to support the vehicle's structure and help it move, even with the strict limits of our design goal. We chose components carefully and designed a cost-effective chassis to make sure our vehicle works well, is light, and does not cost too much.

The dimensions of our vehicles are 9.5 cm in width, 60.6 cm in length, and 28.4 cm in height. Our vehicle weights 2.1 kg. The power system is made to spread energy efficiently to all parts of the vehicle. After conducting a weight analysis, we estimated the torque necessary to inform our choice of motors and subsequent power requirements. Calculations of drawn current and RPM justification were crucial in selecting a power source that would sustain the vehicle's functionality. The chosen batteries—two positioned at the front to power the Raspberry Pi 5, IMU, and an ESP8266 module, and four at the rear for the motors, an ESP8266 module, and Raspberry Pi Pico—provide a robust and reliable energy supply. We observe that the front of the vehicle consumes 7W, and the back consumes 7.7W. In total, the vehicle consumes 14.7W. Precision is paramount in our sensor subsystem, which integrates IMU data and, if viable, fuses it with localization feedback for enhanced accuracy in path planning. The subsystems' design ensures that, whether combined or separate, the data streams contribute to the vehicle's navigation. The vehicle uses the Raspberry PiCam 3 and Aruco markers together to figure out its exact location accurately, which helps it to park correctly. The controller subsystem stands as the brain of the operation, processing diverse inputs to navigate the vehicle efficiently. It takes the helm in converting data into actionable directives, steering the vehicle seamlessly into its designated spot. Our communication system links the front and rear of the vehicle using a metal hollow tube with ESP8266 Wi-Fi modules, ensuring information is shared perfectly and keeping everything working together smoothly.

For our solution procedure, we have developed five main steps: initialization, sensing & localization, path planning, vehicle control, parking & shutdown. The initialization phase includes system configuration, diagnostics, checking the integrity and readiness of all subsystems, calibrating sensors, and establishing initial positioning data for operation. Sensing & localization involve collecting raw sensor data, including visual and orientation sensors, and merging this data to form a cohesive environmental understanding. It also includes Aruco Marker detection, position and orientation estimation, and local mapping. Path planning involves route generation, maneuver planning, trajectory optimization, and predicting vehicle motion in response to control signals to anticipate adjustments. Color and line detection is also implemented for precise parking. The vehicle control phase involves executing fine control for accurate positioning, continuously checking distances to prevent collisions, and adjusting control signals based on real-time feedback. The parking & shutdown stage includes a sequential shutdown of systems once the vehicle is correctly parked, turning off active sensors to conserve power, and ensuring all systems are safely turned off. Each stage is detailed with specific sub-tasks, reflecting a comprehensive process from the initialization of the system to its shutdown, which can be seen in **Figure 1**.

Functional Flow - Top Level



Functional Flow - Second Level

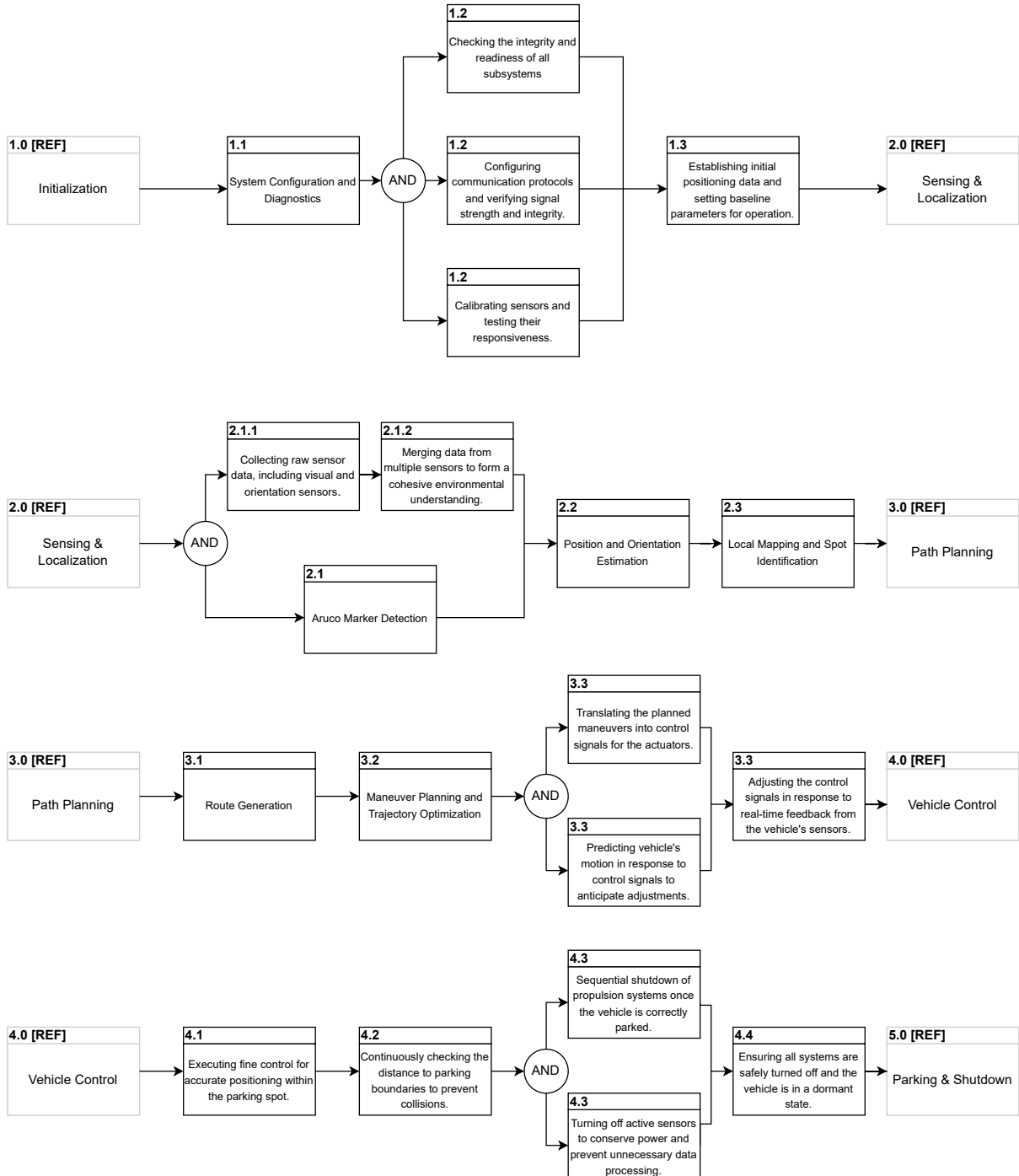


Figure 1: Functional Flow Block Diagram

Each subsystem in the vehicle communicates with each other through the integration of hardware and software components. The communication system, using ESP8266 modules within a metal hollow tube, connects the front and rear parts of the vehicle, ensuring seamless information transfer. This setup allows data from the sensor subsystem (like IMU data and localization feedback) and visual positioning data (from the Raspberry PiCam 3 and Aruco markers) to be relayed to the controller subsystem. The controller subsystem then processes this information to navigate and position the vehicle effectively. The power system supports all these operations by providing the necessary energy to the Raspberry Pi and Picos, sensors, motors, and communication modules. This interconnected network of subsystems ensures the vehicle operates as a cohesive, efficient unit. A detailed system operation diagram and the componentwise relationships with their communication protocols can be seen in **Figure 2** and **Figure 3**.

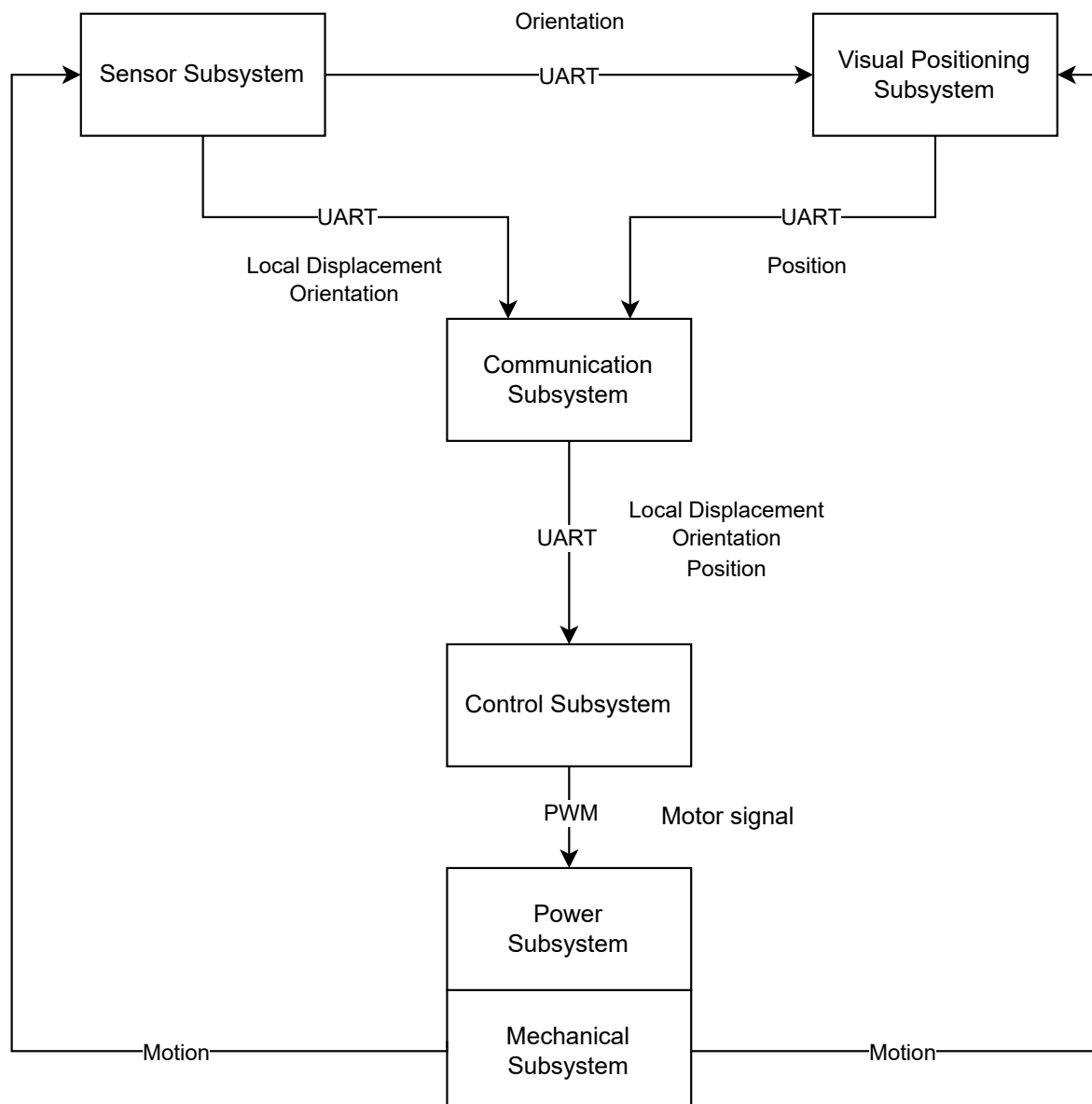


Figure 2: System Operation Block Diagram

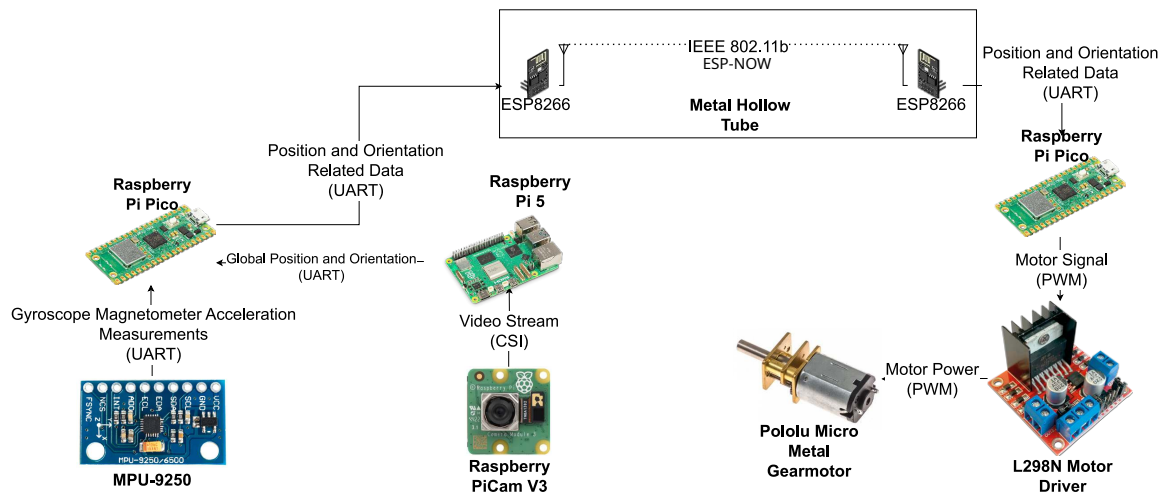
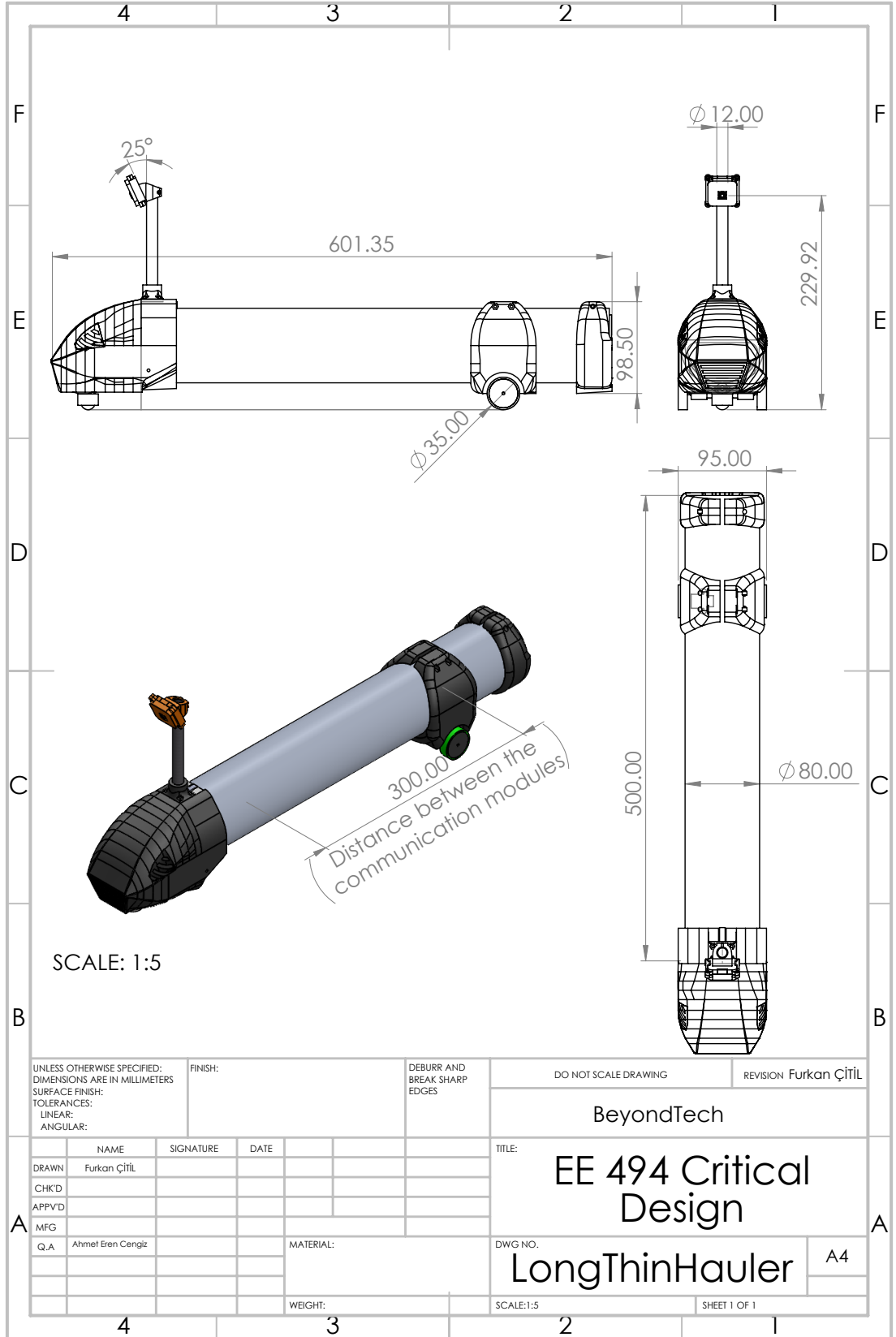


Figure 3: Relations Between Individual Components with Communication Protocols



Mechanical Subsystem

The mechanical subsystem of the long thin hauler project is meant to sustain the structure and permit vehicle mobility while complying with the project's rigorous design limitations. This subsystem is critical to the vehicle's stability, mobility, and longevity, especially given its unusually long and thin shape. The mechanical design parts were carefully picked and integrated to improve performance while keeping the structure small and economical.

Critical design dimensions can be found in the technical drawing above, which satisfies design requirement M-03.

Chassis Design

The chassis is built around a hollow metallic tube, as in the **Figure 5** and **Figure 7**, which serves as the spine of the vehicle. Also, transparent chassis can be seen in **Figure 4** and **Figure 6**. This tube, 50 cm in length and 8 cm in diameter, not only provides structural integrity but also works as a wave-guide for the communication subsystem. The material chosen for the tube is a lightweight yet sturdy aluminum alloy, offering an ideal balance between durability and weight while satisfying the metallic tube requirement, M-01.

Adhering to the width limitations of 7 cm to 10 cm, the vehicle's chassis is designed to be 10 cm wide, which is determined by the front body. This width ensures that the vehicle fits comfortably within the parking space, which is no more than 1.5 times the width of the vehicle.

To support various components, 3D-printed brackets, and plates are attached to the metallic tube. These parts are designed to be lightweight and are made from high-strength, heat-resistant PLA material. These components create spaces for electronic components to be placed in an organized manner.

Moreover, the center of gravity of the chassis is placed as low as possible to provide more stable movement during the operation. This mitigates the change in the angle of the vehicle with respect to the ground as much as possible so that the visual positioning system is not affected by the variation of the pose of the camera.

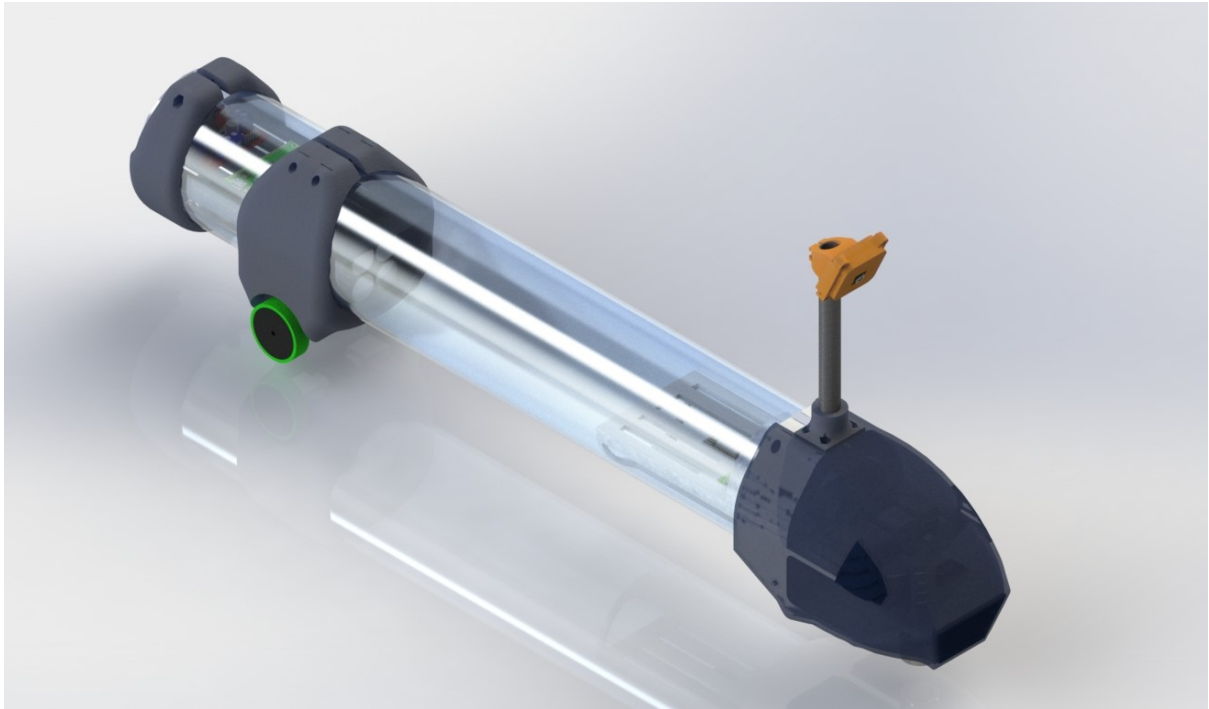


Figure 4: Long Thin Hauler CAD Design - Transparent View

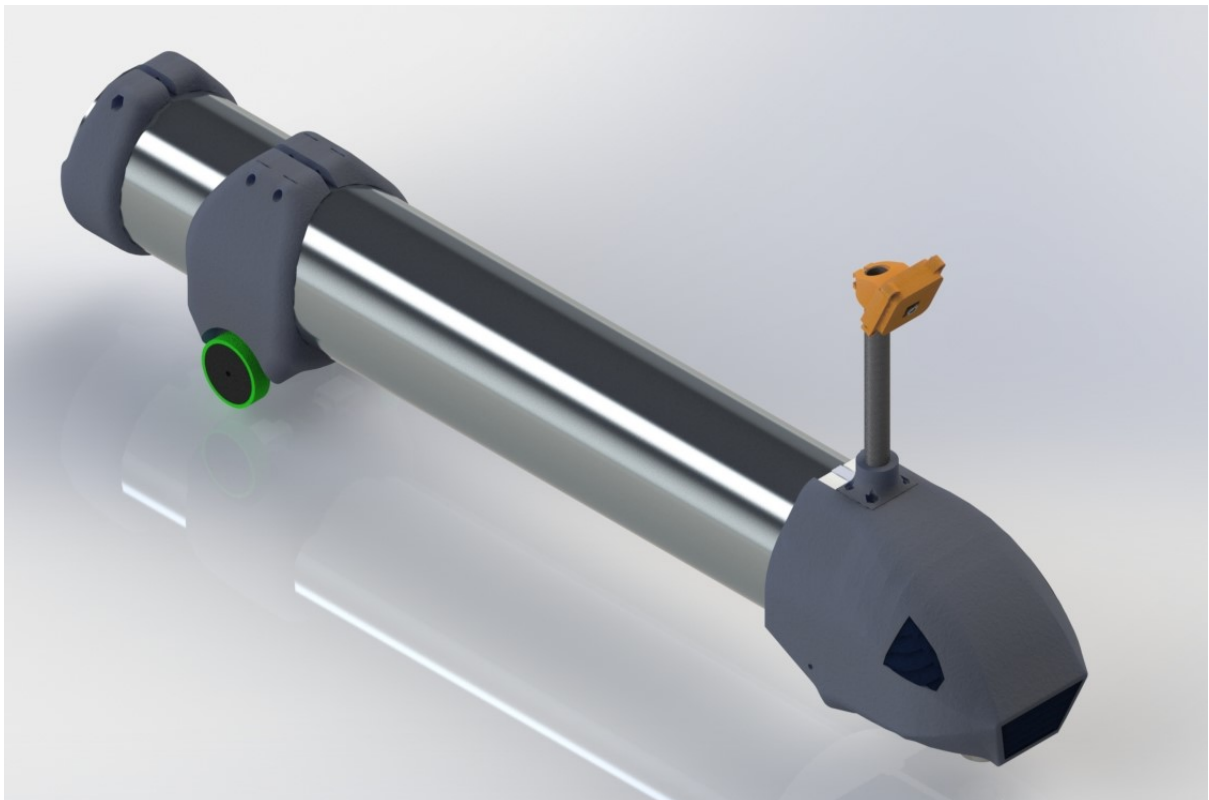


Figure 5: Long Thin Hauler CAD Design - Solid View

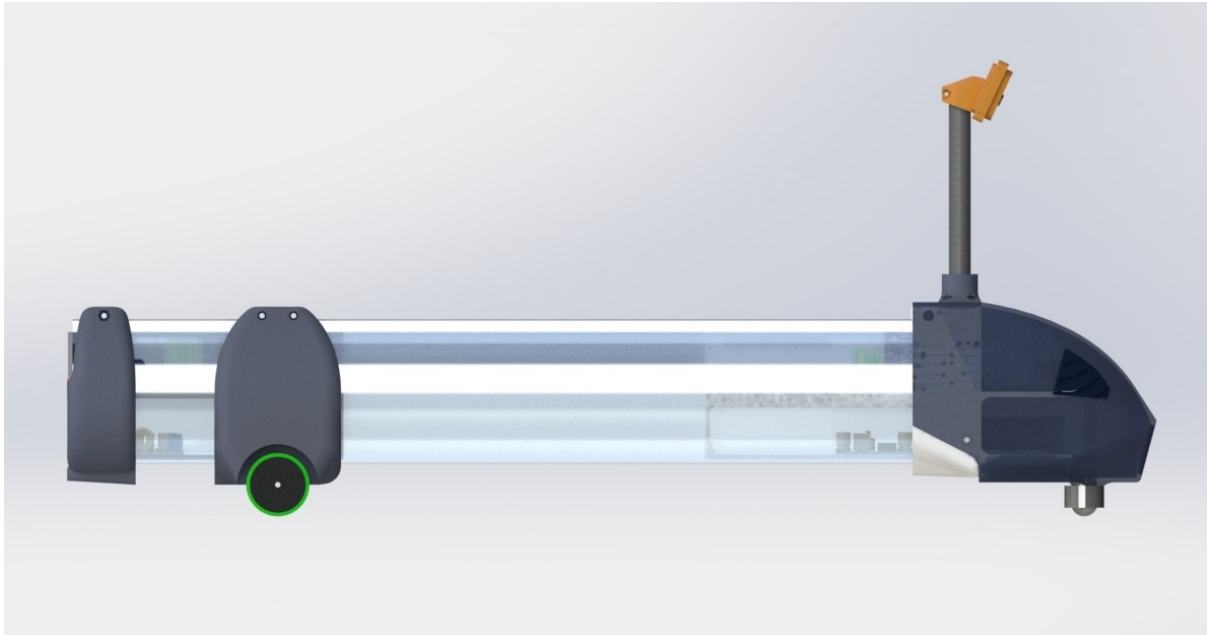


Figure 6: Long Thin Hauler CAD Design - Side Transparent View

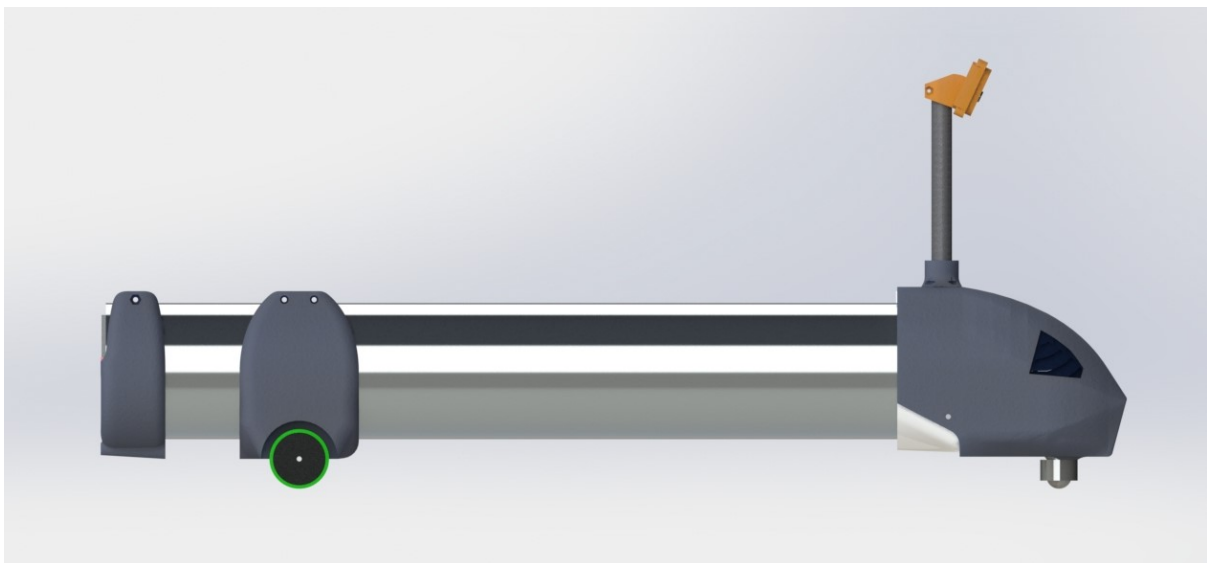


Figure 7: Long Thin Hauler CAD Design - Side Solid View

Mobility and Drive Mechanism

The vehicle features a two-wheel drive system located at the rear. The wheels, each 3.5 cm in diameter, are attached to high-torque motors capable of precise speed and directional control. This design enables forward-only movement, which is essential for the project's guidelines. Also, the rear-drive mechanism bracket is designed such that its position on the longitudinal axis can be adjusted with a compliant mechanism that attaches the metal tube to the rear-drive mechanism bracket.

The motors selected are DC-geared motors with a specification of 140 RPM at 12 Volts, providing sufficient torque to move the vehicle even on slight inclines. Their compact size allows them to fit within the narrow confines of the vehicle's rear. Also, the speed of the

motor enables approximately 15 meters of movement in 1 minute, which is more than enough for reaching the parking spot.

In order to maintain traction on the wheels, rubber wheels are preferred, and the motors are placed on the longitudinal axis as close to the center of gravity of the vehicle as possible to increase friction on the wheels. For the front side, single caster wheel with bearing inside to reduce friction.

The camera mount consist of hard-case for the camera board, carbon fiber rod and a body mount for the camera system. Use of carbon fiber in the front body, not only decreases the inertial moment of the vehicle with respect to the differential motor centers, but also contribute to the mitigation of vibration damping for the camera because of its low weight and vibration damping material properties. Following the new camera mount design, front body design is also changed to reduce the weight of the body and time to print the part with 3D printer. Tabulated component list and their weights can be found in **Table 1**.

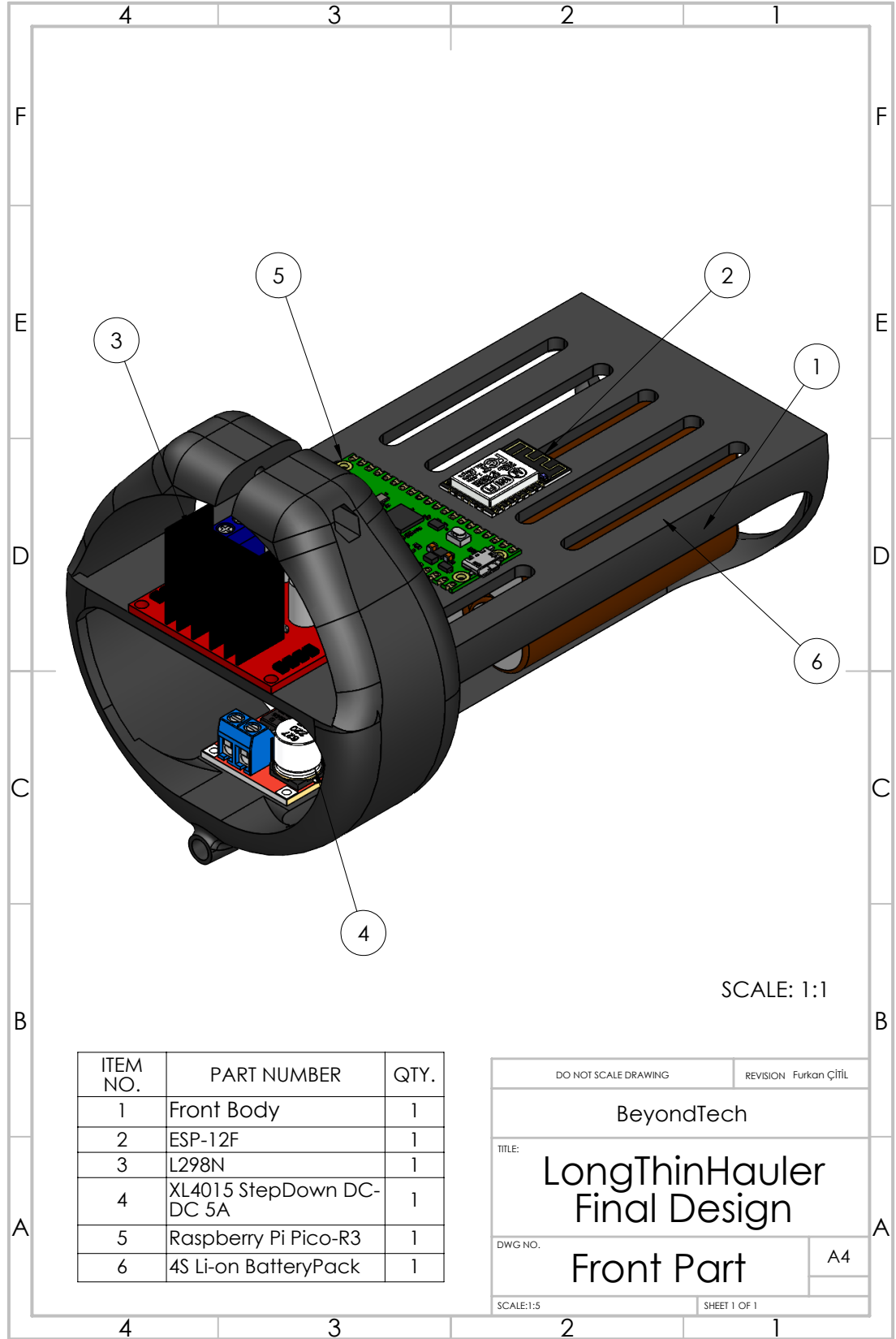
Table 1: Component List and Their Weights

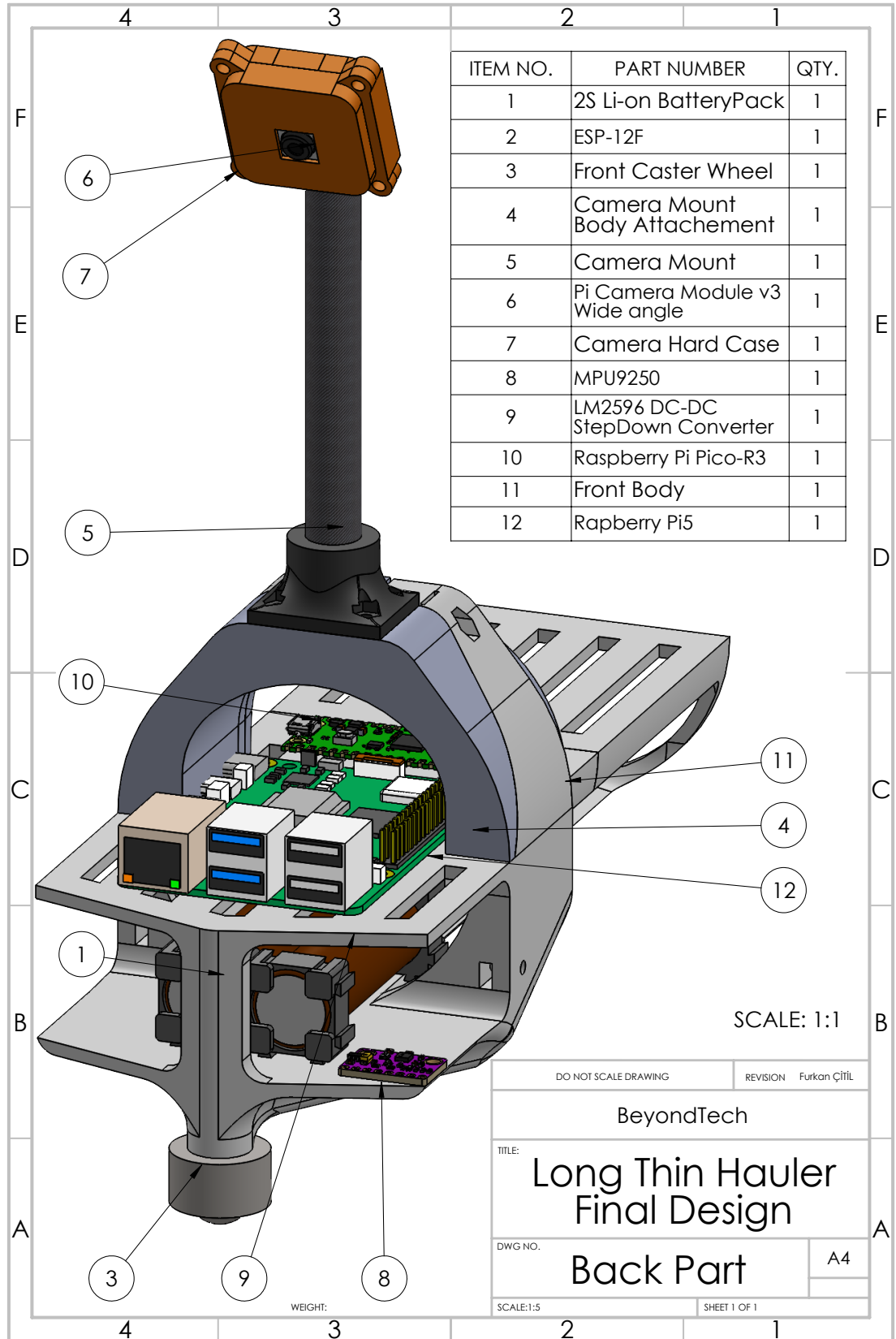
Component	Quantity	Total Weight
Raspberry Pi 5 + Cooler	1	46+25=71g
Raspberry Cam Module 3	1	4g
Motor	2	2x10=20g
Motor driver (2 channel)	1	26g
ESP8266	2	2x17=34g
Micro-controller	2	2x15=30g
IMU	1	10g
Voltage Regulator	2	40g
Wheel	2	10g
Battery	6	6x45=270g
LED	1	1g
Buzzer	1	1g
Metal Tube	1	700g
3D printed parts	1	800g
Camera Stick	1	20g
Caster Wheel	1	60g
Total		2097g

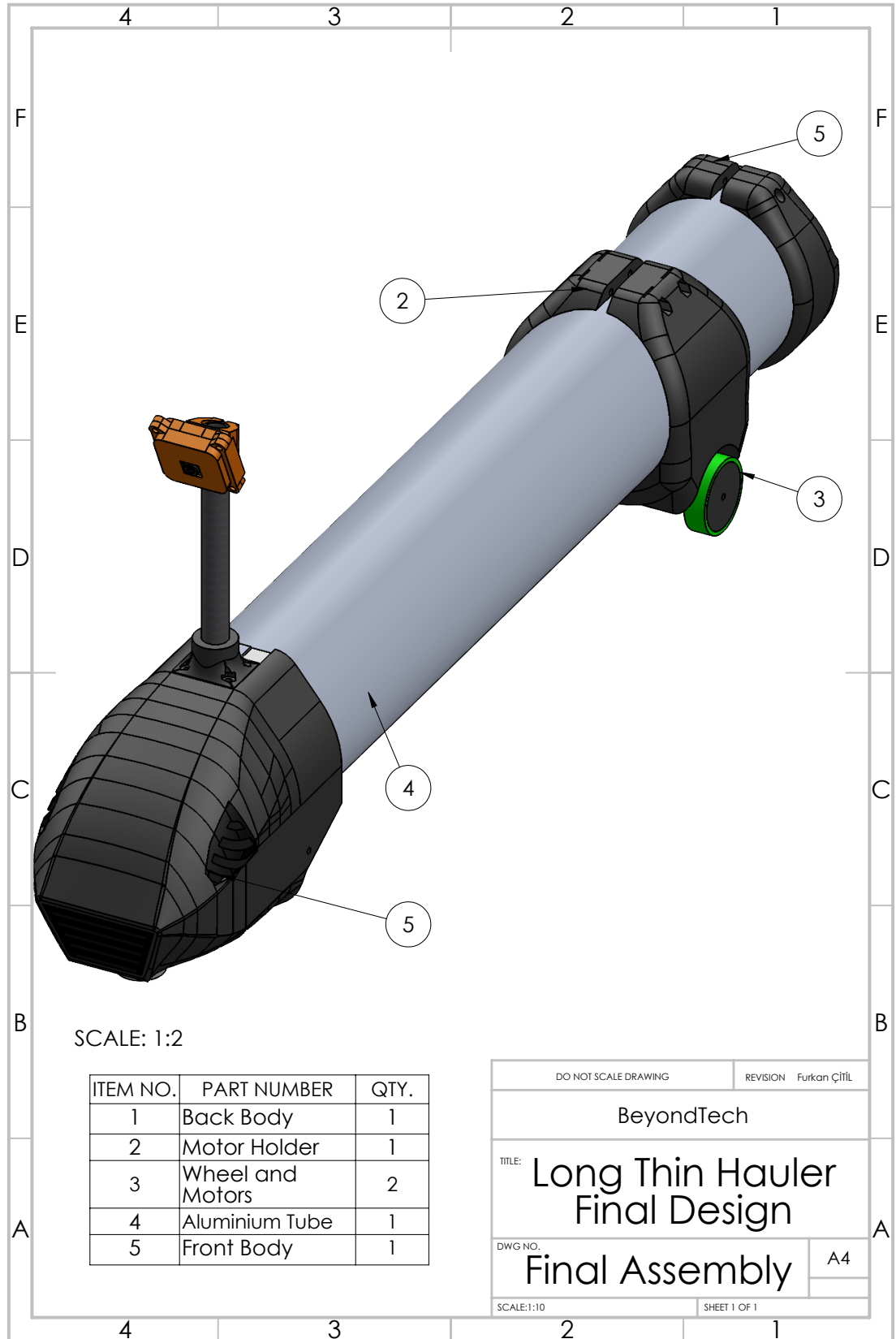
Sensor and Actuator Placement

All sensors are positioned in the vehicle's front. A custom-designed sensor enclosure is 3D printed and put at the front, ensuring that the sensors have clear and unimpeded vision. This positioning is critical for the visual positioning subsystem to detect the vehicle's position and orientation properly. Since the visual positioning system fully relies on the camera, the camera mount is tightly connected to the front body with four screws to ensure angle variation in the camera pose.

The actuators, primarily the motors and motor drivers, are situated at the rear of the vehicle. This placement provides direct power to the wheels and allows for a better separation between the vehicle's control and sensory mechanisms with a hollow metal tube. The placement of all components can be found in the following bill of materials. The total weight is 2097g, which is below 3kg. Hence, the requirement M-02 is satisfied.

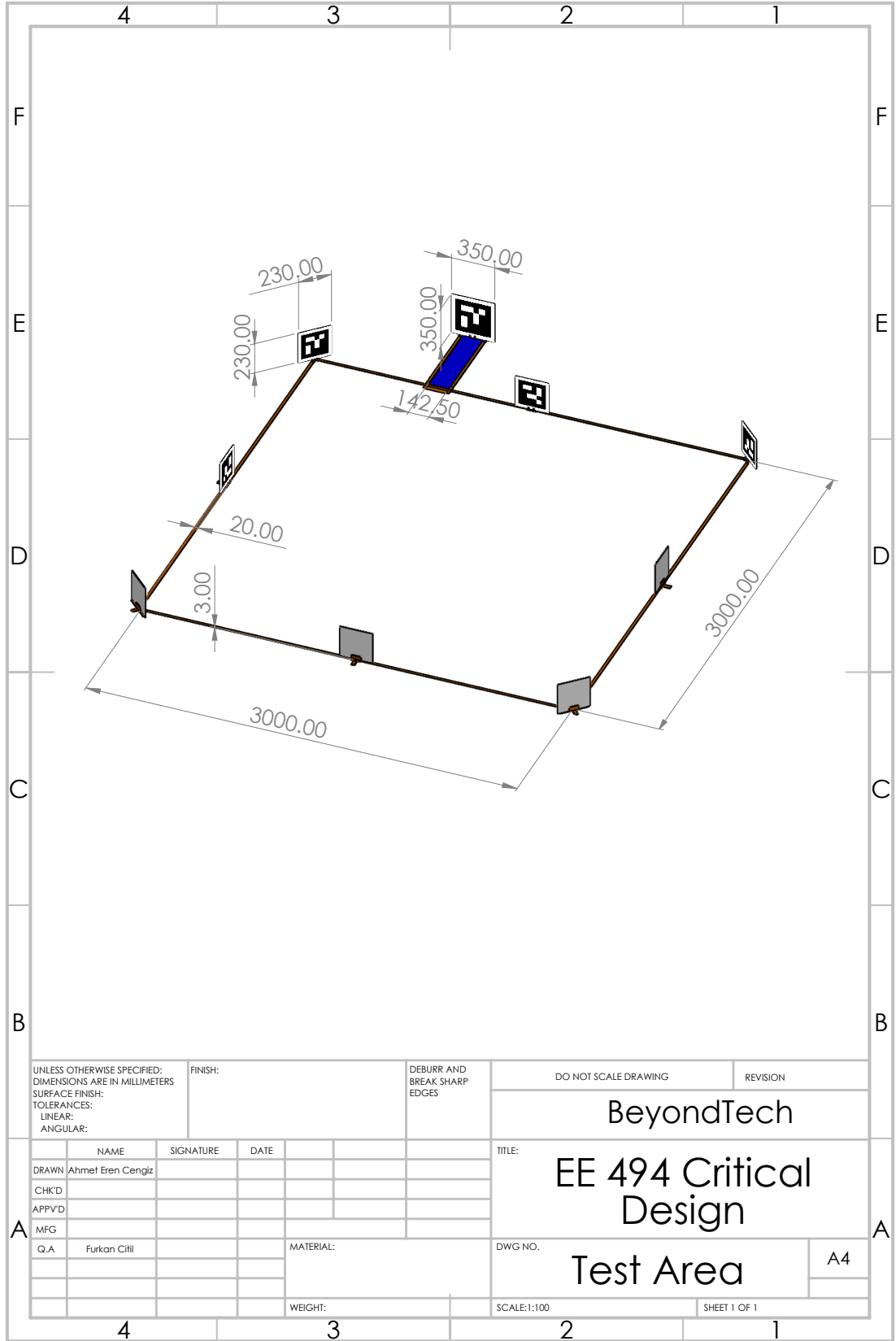






Parking Area Design

In order for our localization subsystem we proposed a solution with Aruco markers and inertial measurement sensor. The proposed solution relies on fusion of visual positioning with Aruco marker and IMU. Therefore, we needed a stand for holding Aruco marker plates and equipment for constructing parking spots and parking area lines. Following these requirements, we created Aruco marker plates with foam-boards and holders for them with PLA material using 3D printer. Also, we utilized these Aruco marker stands to join the area borders which are produced by polypropylene foam-board with thickness of 3 mm sliced to 20 mm of width lines. For localization, we used 8 Aruco markers, 4 of them for corners and 4 of them for mid-points of the edges. Apart from these, we used an additional, bigger sized Aruco marker for the parking spot, which would make parking spot easier to detect and allow our vehicle to continue visual positioning during final parking maneuvers if required. Moreover, we designed a slidable parking spot with a distinguishable color from the environment. This design aimed to make parking spot more detectable with color based algorithms, requiring lower computational power to ease burden on our main computer. Our parking area design with its dimensions can be seen in the following dimension figure.



Power Subsystem

This is a crucial subsystem to supply the required power to all other subsystems which ensures the operation of the overall system. Lithium-ion batteries are selected as the main power suppliers since they have high energy densities and can provide sufficient current for our applications. Furthermore, they are rechargeable (required by P-03). Moreover, they are safer to operate and not as expensive as compared to some other type of batteries such as lithium polymer.

The electronic components in the front side works with 5V and 3.3V, so series connection of two cells is enough with lithium-ion batteries which have a nominal voltage of 3.7 V. From the **Power Management**, it can be seen that 1.5 A is required from the battery, and it can be supplied by a single cell, so no parallel connection is required. Therefore, a 2S battery pack is used in the front.

The motors in the rear side work with 12 V, so four lithium-ion batteries in series are required. From the **Power Management**, it can be seen that 1 A required from the battery and it can be supplied by a single cell, so no parallel connection is required. Therefore, a 4S battery pack is used in the rear.

Lithium-ion batteries are connected to each other by nickel strips with 0.15 mm thickness and 8 mm width which has a current rating around 5A, which is suitable for our application. The nickel strips are spot welded to the batteries with a spot welding machine as in **Figure 8**.



Figure 8: 4S connection

Battery Management System (BMS) is connected to the battery packs by soldering cables. BMS ensures the voltage levels in different cells remain the same. Furthermore, it protects the batteries by preventing cell voltages to drop below 3.2 V and exceed 4.2 V as required by P-03.

The power is supplied to the rest of the vehicle by 14 AWG cables which have 5.9 A current rate, and XT30 connectors which have 30 A current rate are used to connect and disconnect the batteries easily as required by P-02. Also, the batteries are isolated by the heat shrink tubes.

Visual Positioning Subsystem

The visual positioning subsystem is one of the most critical subsystems which is evaluated during the module tests. This subsystem's primary function is to accurately determine the vehicle's position and orientation within a designated driving area using visual data from a camera module. This system depends on the Aruco markers, which are strategically

placed in the driving area. These markers, with their known positions and orientations, form the basis of our navigation algorithm. Our strategy includes positioning these markers at critical locations, particularly at the corners and the center of the edges of the driving area. This setup is anticipated to deliver adequate performance for the vehicle's navigation capabilities. In situations where markers are not visible, the vehicle is equipped to identify colored strips located at the boundaries of the drivable area. Even in those cases, the subsystem is continuously providing real-time data, which is required by V-02. To facilitate the detection of the parking spot, we place a large colored strip on the floor of the spot, accompanied by a fiducial marker. Our approach involves utilizing color and shape detection techniques to locate the parking spot in the captured images. Following this, we estimate the position of the vehicle in relation to the parking spot and subsequently determine its position relative to the entire drivable area. A fiducial marker is also planned to be installed at the end of the parking spot to aid in precise positioning.

The camera module on the vehicle captures images of these markers, and we utilize the OpenCV library to process these images. OpenCV is one of the most popular image processing libraries for its robust features in computer vision, including efficient detection of fiducial markers. Through this process, we can determine the vehicle's relative position and orientation based on the markers captured in the camera's field of view. The vehicle's pose calculation involves a detailed analysis of the spatial relationship between the known positions of the markers and their perceived positions in the camera's images. This process involves employing mathematical methods to accurately deduce the vehicle's exact location and orientation in the drivable area.

While integrating IMU feedback data to camera-based data, our system achieved additional robustness and reliable autonomous navigation and parking capabilities. The representative image illustrates the process given in **Figure 9**.

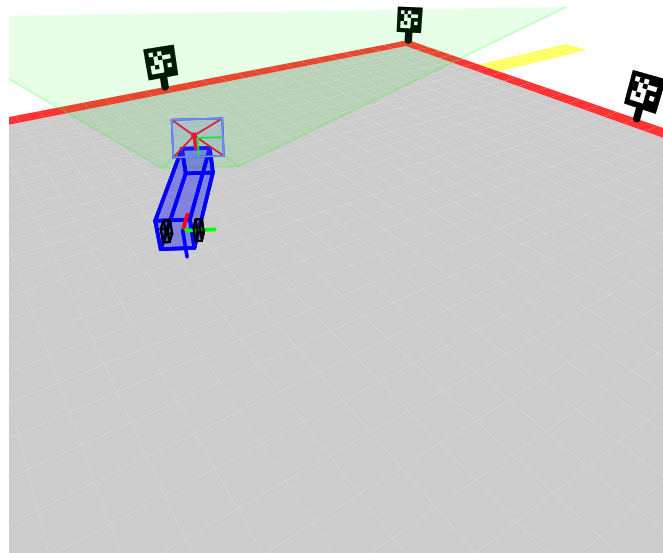


Figure 9: The vector rendering of the vehicle estimating its pose

As for fiducial markers, we have chosen to utilize Aruco markers, a specific type of 2D barcode, for their ease of detection and simplicity. Aruco markers are distinct, square-shaped patterns used extensively in computer vision and augmented reality for accurate positioning and orientation (**Figure 10**). Each marker has a black border surrounding a binary matrix, uniquely identifying the marker. This design allows for quick and reliable

detection by digital imaging devices. They are particularly advantageous due to their ease of production; they can be printed, offering a cost-effective solution for various applications. Commonly used in fields like robotic navigation, augmented reality, and 3D positioning, Aruco markers provide essential spatial data in real-time. Specialized libraries such as OpenCV facilitate their detection and decoding, which efficiently process images to identify and interpret these markers. The versatility and capability for precise spatial recognition make Aruco markers a preferred choice in applications requiring real-time, accurate spatial awareness. Since these markers are also resilient to lighting changes, this makes them suitable for the design (the requirement V-04).

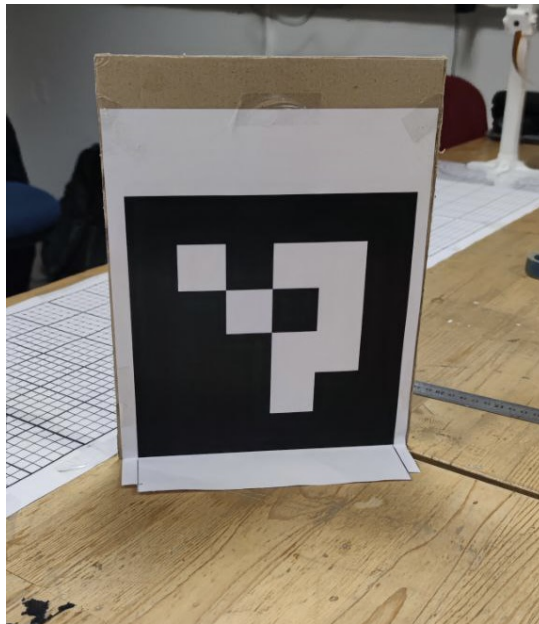


Figure 10: One of the Aruco markers used in the project

The structure of Aruco markers is straightforward and effective. We are using the 4x4 variant, which contains 16 bits of information. Of these, the first 12 bits are designated for the marker's unique identification, while the remaining 4 bits are used for error detection. This error detection capability is crucial for ensuring the accuracy and reliability of the marker recognition process.

The marker ID system is designed such that ID is divided into two parts: the first 6 bits represent the row number, and the subsequent 6 bits represent the column number. This bifurcation aids in simplifying the process of marker identification and tracking. The marker ID is then formed by concatenating these row and column numbers, providing a unique identifier for each marker. We have stored each marker's position and orientation data in the vehicle's memory system. This information is utilized to determine the vehicle's pose based on the perceived positions of the markers in the camera's images.

During our module testing phase, we observed a notable influence of the marker's immediate background on detection accuracy. Specifically, markers placed against a white background yielded more precise detection compared to those against a black background. This contrast is attributed to the tendency of a black background to be mistakenly interpreted as part of the marker itself, leading to potential errors in detection. To mitigate this issue, we have decided to use a white background for all our markers. This background extends approximately 2 cm around each marker, providing a clear and distinct border that enhances detection accuracy and consistency.

Our design incorporates a camera module mounted at the front of the vehicle, specifically targeting the driving area. We chose a fixed position and orientation for the camera,

utilizing a wide-angle lens to enhance its field of view. This wide field of view is essential for the camera to capture a large portion of the driving area, which is critical for accurate positioning. The camera is positioned approximately 25 cm above the ground. The field of view of the camera is about 100 degrees horizontally and 70 degrees vertically. This arrangement, we believe, provides a field of view to detect markers within the driving area accurately.

Since the camera used is a wide-angle camera, the lens distortion of the image is significant. Image distortion, particularly in wide-angle lenses, can significantly skew the data needed for accurate pose estimation. To overcome this problem, our approach involves the utilization of the OpenCV library for image undistortion. The process of undistorting the image requires a camera matrix and distortion coefficients, which are fundamental in rectifying the image. The camera intrinsics matrix is a 3x3 matrix, including information like the camera's focal length and its optical center. Meanwhile, the distortion coefficients are used to correct both radial and tangential distortions observed in the image.

These coefficients are derived through a camera calibration process. This process includes capturing multiple images of a checkerboard pattern, where the dimensions are known and precise. The MATLAB Camera Calibration tool can process these images to compute the camera matrix and the distortion coefficients (**Figure 11**). In our case, we used a 9x6 checkerboard pattern, which is the default pattern used by the tool. The tool also provides a visual representation of the distortion coefficients, which can be used to determine the accuracy of the calibration process. Then, the obtained camera matrix and distortion coefficients are used to undistort the image using the OpenCV library.

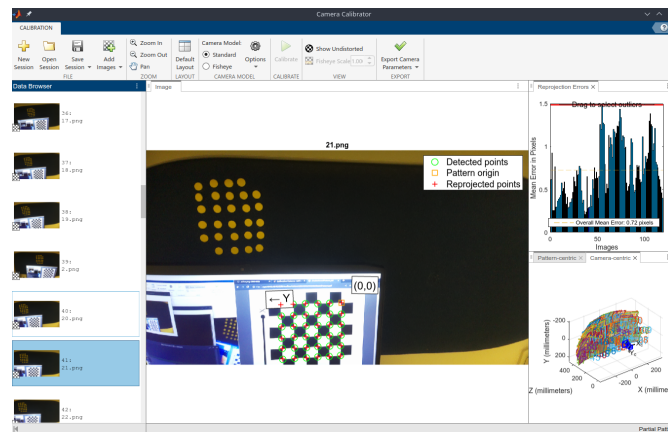


Figure 11: MATLAB Camera Calibration Tool in action

This model is known as the pinhole camera model, which is a simplified representation of the camera's optical system (**Figure 12**). This model assumes that the camera's lens is a pinhole, which allows light to pass through and form an inverted image on the image plane. This model is widely used in computer vision and image processing applications, including camera calibration and pose estimation. The pinhole camera model is represented as follows:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

$$\vec{p}_c = K[R|t]\vec{p}_w \quad (2)$$

Here, K represents the camera intrinsic matrix, a 3x3 matrix detailing the camera's focal length and optical center. R is the rotation matrix, another 3x3 matrix, which defines the camera's rotation in relation to the world coordinate frame. The translation vector, denoted as t , is a 3x1 vector indicating the camera's positional change relative to the world coordinate frame. Points in the world and camera coordinate frames are represented by p_w and p_c , respectively. Through this transformation, we can determine the image coordinates of a point using the equations $x = \frac{X_c}{Z_c}$ and $y = \frac{Y_c}{Z_c}$.

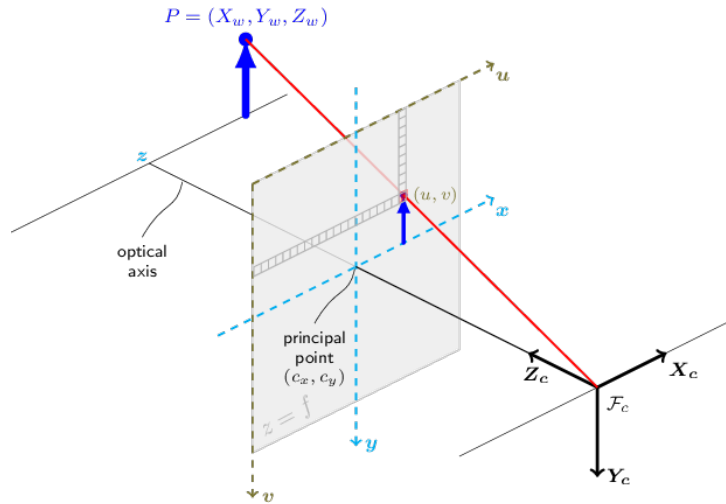


Figure 12: The pinhole camera model

Lens distortion is described using seven coefficients: k_1 , k_2 , p_1 , p_2 , k_3 , k_4 , and k_5 . The coefficients k_1 and k_2 are for radial distortion, while p_1 and p_2 address tangential distortion. The coefficients k_3 , k_4 , and k_5 are utilized for modeling thin prism distortion. The detailed formulation of these coefficients is not covered in this report.

A critical aspect of our system design involves the use of a homography matrix. This 3x3 matrix enables the transformation between the image and ground planes, which is essential for mapping camera observations to real-world coordinates. A homography matrix, in essence, is a perspective transformation tool that aligns two planes. This matrix is used to warp points seen by the camera on the image plane to their corresponding locations on the coordinate frame centered around the vehicle.

The homography matrix is represented as follows:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (3)$$

Generally, the parameter w is taken to be 1. Likewise, after applying the matrix, we find the transformed points as $x'' = x'/w'$ and $y'' = y'/w'$. This notation is known as the homogeneous coordinate system. This convention is quite useful since it allows any kind of transformation to be an affine transformation in a higher dimensional space. As explained, we are using the homography matrix to transform between the image plane and the ground plane.

Since the camera's angle and height are fixed, we expect the homography matrix to remain stable. After the construction of the vehicle, we calibrate this matrix to transform points between the image plane and ground plane relative to a vehicle-centered coordinate

frame. The calibration of the homography matrix employs the least-squares method. Given the fixed nature of the camera, we do not anticipate significant deviations from the initially calibrated matrix.

To estimate the position and the orientation of the vehicle, we employ homography calculations using the lower two corners of an image marker and their known positions within the driving area. The illustration of the area, including the markers, is given in **Figure 13**. The detection of the lower corners yields the ground plane coordinates of these corners relative to the vehicle. By comparing these coordinates with the known positions of the corners in the driving area, we can determine both the translation and rotation between these points. The rotation is used to determine the vehicle's orientation, while the translation vector is used to determine the vehicle's position with respect to the drivable area.

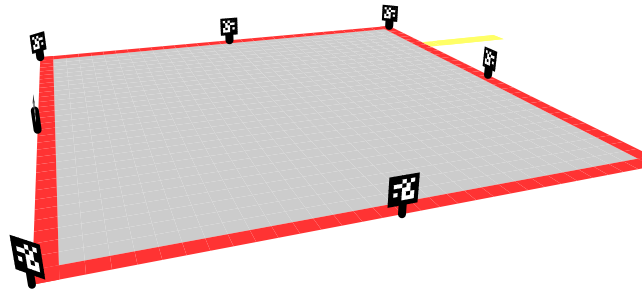


Figure 13: Vector rendering of the preliminary design of the area and the parking spot

The problem we address is known as a perspective-n-point (PnP) problem, with some modifications to improve accuracy. Unlike traditional PnP problems that involve six degrees of freedom, our scenario is simplified to three: X, Y, and yaw, as the positions of the markers and the camera's position and orientation are fixed. This restriction is expected to simplify the problem and improve the pose estimation accuracy. The highest possible precision is required by the V-05. The change of pitch and roll of the vehicle is expected to be negligible, and therefore, we are considering them as constants.

The inaccuracies in yaw measurement significantly impact positional accuracy. Although we have reduced the rotational freedom to only yaw, the errors in its estimation can still significantly affect position estimation. To mitigate this, we integrate an Inertial Measurement Unit (IMU) to refine yaw estimates. The IMU provides an initial yaw estimation, which is used to correct the camera-based yaw calculation. This fusion of data enhance the overall precision of the pose estimation. This aspect is included in the subsystem requirement V-03.

To locate the parking spot, our method employs a combination of color detection and shape finding algorithms. Once the parking spot is identified, we use a homography matrix

to ascertain the parking spot's pose in relation to the vehicle. As our vehicle is capable of estimating its own pose relative to the entire area, we can then accurately determine the absolute position of the parking spot, satisfying the V-01 requirement.

There are other considerable challenges in implementing this subsystem. One of the challenges is to ensure the accuracy of homography and camera calibration and the exact placement of fiducial markers within the driving area. Additionally, any shift in the camera's position relative to the vehicle could lead to significant pose estimation errors. Therefore, we ensured that the structure of the vehicle is rigid and that the camera's position remained fixed because any deviation in these parameters could lead to significant errors in the vehicle's pose estimation.

Control Subsystem

One of the critical parts of the Long Thin Hauler project is the control subsystem. It covers autonomous navigation and precise path tracking for automated parking. In collaboration with the sensor subsystem (IMU+Filter) and visual positioning subsystem, the control subsystem interfaces with the mechanical subsystem, driving DC motors to execute actions. The primary objective is to design a control structure that realizes autonomous parking maneuvers. The control subsystem aims to ensure a smooth drive of the mechanical subsystem through the precise actuation of DC motors. To address the control problem, four alternative approaches have been discussed, and one of them, given in **Figure 14**, is chosen as the final method.

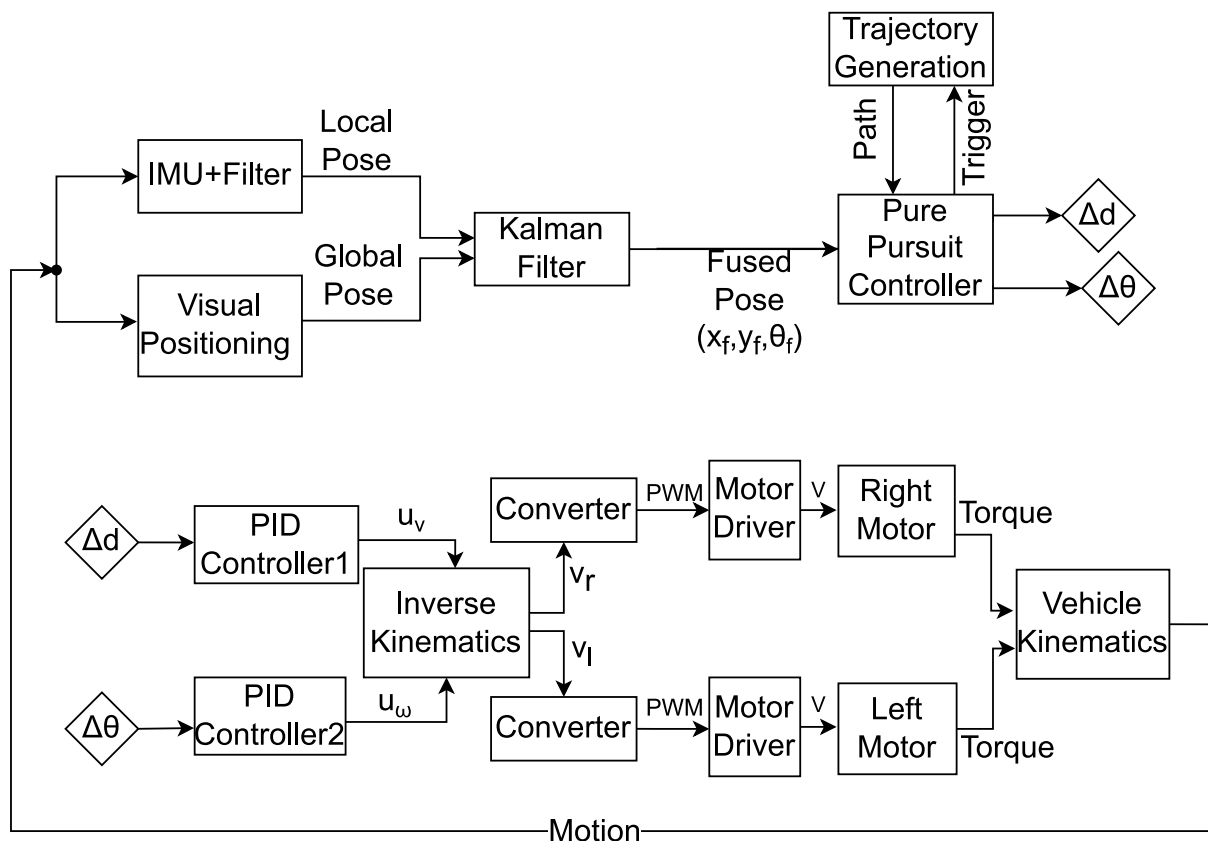


Figure 14: Chosen Controller Solution

To ensure the robustness and reliability of the control algorithm according to the subsystem requirement C-01, a closed-loop strategy relying on feedback from both the sensor

subsystem and visual positioning subsystem has been chosen. IMU+Filter block in the diagram refers to the orientation and velocity information coming from the sensor subsystem. Mainly, data is taken in two formats: one as raw orientation and velocity data and the other as odometry information, which provides a local pose estimate. In addition to this, a global pose estimation is obtained from the visual positioning subsystem.

The calculated global pose and local pose are combined using a Kalman Filter to obtain a more robust estimation that mitigates errors. The filter aims to reduce the high-frequency noise of the IMU data and the low-frequency drift of the visual positioning system. The output of the Kalman Filter, representing the fused pose data, is fed into the Pure Pursuit Controller. This controller block assesses these inputs by comparing information from the trajectory generation block and calculates the error in distance and orientation by using the following equation.

$$e_d = \sqrt{(x_{desired} - x_f)^2 + (y_{desired} - y_f)^2} \quad (4)$$

$$e_\theta = \theta_{desired} - \theta_f \quad (5)$$

More specifically, the Pure Pursuit Controller is a widely used path-tracking algorithm in the literature and automation industry. An illustration of the algorithm logic is provided in **Figure 15**. Essentially, it takes a predefined path and the current pose of the vehicle to calculate a desired orientation (heading) for guiding the vehicle along the path. Initially, it determines the closest point with the minimum lateral error to the vehicle. Then, based on the chosen lookahead distance, it identifies the next target point on the path. This lookahead distance is determined based on the cruise speed, which is determined based on the sub-system requirement C-03.

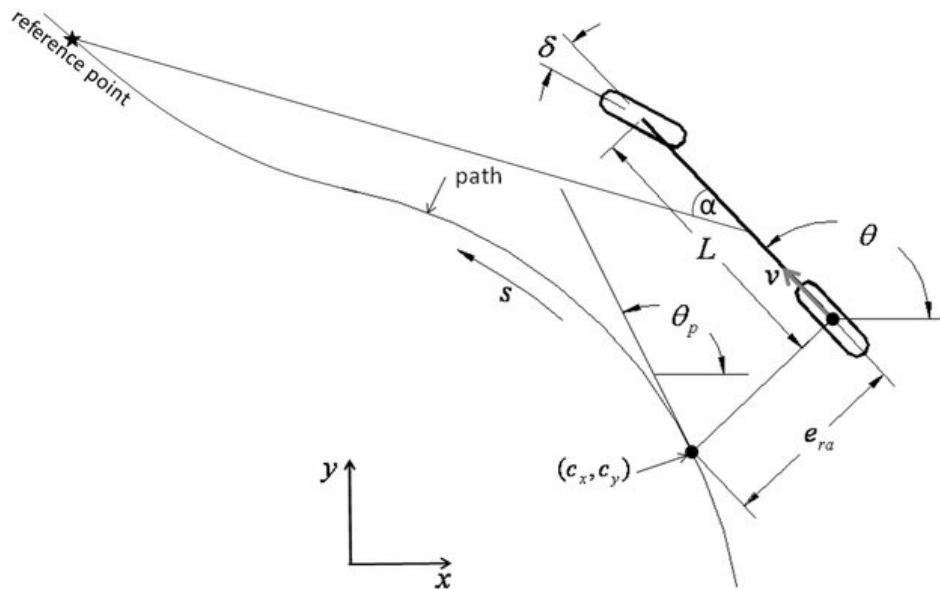


Figure 15: Pure Pursuit Controller Logic

It is crucial to indicate that the Pure Pursuit Controller has a feedback connection to the path generation block, which is triggered when the lateral error to the closest point exceeds a predetermined value for a duration of 5 seconds. The activation of this signal indicates that the current path is not proper to follow; consequently, a new path is generated and provided to the Pure Pursuit Controller based on the position and orientation of the current instance. After the Pure Pursuit Controller calculates the desired orientation value,

it is compared with the current orientation data coming from the IMU+Filter block, and the difference is sent to the PI controller, K_d parameter is set to zero.

In the chosen solution approach, the error between the current orientation and the desired orientation, and the distance to the target point calculated based on the Pure Pursuit algorithm are converted to angular and linear velocity values with two PI Controllers whose equations are given below.

$$u_v[k] = u_v[k-1] + k_{p1}(e_d[k] - e_d[k-1]) + k_{i1}e_d[k] \quad (6)$$

$$u_\omega[k] = u_\omega[k-1] + k_{p1}(e_\theta[k] - e_\theta[k-1]) + k_{i1}e_\theta[k] \quad (7)$$

K_p and K_i parameters are determined based on fine-tuning. These calculated linear and angular velocities are provided to an inverse kinematic block. The inverse kinematic model is used to estimate the wheel velocities necessary to achieve the desired linear and angular velocity inputs. This model is derived from the kinematic model of a differential wheel robot, as illustrated in the **Figure 16**.

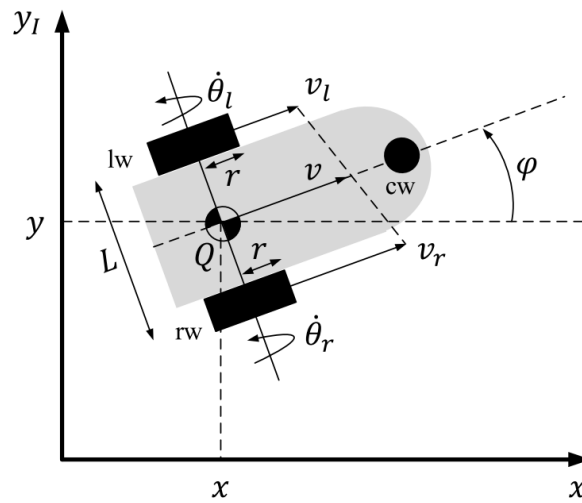


Figure 16: Parameters of The Differential Wheeled Robot

The kinematic model of the differential wheel robot calculates the pose change for a given set of angular and linear velocities. This model assumes that the robot is symmetric along the longitudinal axis and has a rigid body. Additionally, it is assumed that the wheels do not move in the lateral axes or slip. Based on these constraints, the equation of the kinematic model is obtained, as seen in the equation below.

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \frac{R}{2} \begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ \frac{1}{L} & \frac{-1}{L} \end{bmatrix} \begin{bmatrix} \phi_r' \\ \phi_l' \end{bmatrix} \quad (8)$$

Where R is the radius of the wheels, which is 1.75 cm, and L is the distance between the wheel centers, which is approximately 9.8 cm in our main solution approach.

What is required is to derive the inverse of this operation to calculate the necessary angular and linear velocities, specifically for the right and left wheel velocities, to achieve

a desired pose. This operation is known as the inverse kinematic, and the equation for it is as follows:

$$\begin{bmatrix} \phi'_r \\ \phi'_l \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & L \\ 1 & -L \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (9)$$

The Inverse Kinematics block in the diagram utilizes the formula mentioned above. Once the required right and left wheel velocities are obtained, these values are fed into a converter to transform them into PWM (Pulse Width Modulation) signals, controlling the voltage values supplied to the motors. Also, in the converter block, a clamping, which guarantees a non-zero velocity, is made to satisfy the sub-system requirement C-02. Additionally, motor drives are employed before the DC motors to enhance driving performance and efficiency.

The final part of the controller solution is related to the parking maneuver. For this part, the same controller is used. However, the source of the orientation value is changed from IMU to line detection algorithm to increase the accuracy since approaching the parking spot requires more precise motion.

To determine the switching time of the control subsystem between IMU and the line detection algorithm, a function is employed to check the position of the vehicle on the path based on information from the visual positioning and sensor subsystem.

State Machine

We have employed a state machine to manage the control subsystem's operation. The state machine is designed to switch between different states based on the vehicle's position and orientation. The state machine is responsible for determining the vehicle's current state and deciding the appropriate control strategy based on this state. The state machine is designed to switch between the following states:

- **IDLE:** The vehicle is in standby, monitoring for the activation command via the parking button to initiate the parking process.
- **TO CENTER:** The vehicle navigates to the center of the parking area, establishing a baseline position for initiating further maneuvers.
- **ORBIT:** The vehicle circles the central area, utilizing both onboard sensors and external cues to locate and align with the designated parking spot.
- **PINPOINT:** This state refines the vehicle's alignment and positioning relative to the parking spot to ensure an accurate approach.
- **ALIGNMENT:** Despite its name, this state requires the vehicle to reverse away from the parking spot, adjusting its position to enable a forward approach.
- **APPROACH:** As the vehicle proceeds along the planned path towards the parking spot, it continuously adjusts its trajectory based on real-time sensor feedback to maintain proper alignment.
- **APPROACH.NOEV:** Similar to the APPROACH state, this mode operates solely on internal sensor data, triggered when external vision systems are unavailable but the parking goal remains detectable.
- **PARK:** In this final stage, the vehicle executes controlled movements to accurately position itself within the parking spot.
- **RESET:** This state resets all operational parameters and sensor readings, preparing the vehicle for a new task by clearing any residual data.

The state machine diagram is given in **Figure 17**. The state machine is designed to ensure the vehicle follows a structured sequence of actions, transitioning between states based on the vehicle's position and orientation. This approach is essential for maintaining a

systematic and controlled operation, ensuring the vehicle executes the parking maneuver accurately and efficiently.

State: IDLE

Description	The vehicle remains inactive, ready to receive commands. It continuously monitors inputs, specifically waiting for the parking button to be pressed to initiate the parking process.
Transition	To TO CENTER: Occurs when the parking button is pressed, signaling the vehicle to start the parking maneuver by moving towards the center of the parking area.

State: TO CENTER

Description	The vehicle moves towards the designated center of the parking area. This central positioning is crucial as a starting point for the subsequent parking operations.
Transitions	To ORBIT: This transition occurs when the vehicle reaches the center, allowing it to start orbiting to scan for the parking spot. To ALIGNMENT: If sufficient data (like visual markers) is collected quickly, the vehicle might directly proceed to reposition itself for an optimal approach.

State: ORBIT

Description	During this state, the vehicle orbits around the center point to better locate and align itself with the parking spot using onboard sensors and external cues.
Transition	To PINPOINT: Initiates when the goal marker is detected within a close proximity that allows for precise parking adjustments.

State: PINPOINT

Description	This state involves fine-tuning the vehicle's position and orientation relative to the parking spot to ensure precise entry.
Transitions	To PARK: If the vehicle is correctly aligned with the parking markers and is in a position to execute the parking maneuver. To ALIGNMENT: If the vehicle requires further distance adjustments to create an optimal starting point for the approach.

State: ALIGNMENT

Description	This state involves the vehicle moving in reverse away from the detected parking spot to gain a better approach trajectory. This repositioning is key to aligning the vehicle for a forward entry into the parking spot.
Transition	To APPROACH: Once the vehicle is correctly positioned and set for a direct approach towards the parking spot.

State: APPROACH

Description	The vehicle advances along the defined path towards the parking spot. This state involves careful monitoring and real-time adjustments based on continuous sensor input to ensure alignment and correct positioning as the vehicle moves closer to the parking spot.
Transitions	To PARK: If the vehicle reaches the parking spot with accurate alignment. To APPROACH.NOEV: If external vision sources are compromised but the goal marker is still visible using internal sensors.

State: APPROACH.NOEV

Description	This state is similar to APPROACH but relies solely on the vehicle's onboard sensors. This mode is activated when external vision is unavailable, yet the goal is detectable, allowing the vehicle to continue its approach.
Transition	To PARK: If the vehicle remains aligned and the goal marker is visible and within the correct parameters for parking.

State: PARK

Description	The final maneuver where the vehicle precisely positions itself within the parking spot. This involves executing a series of controlled movements to safely park within the designated area.
Transition	To RESET: After successfully completing the parking.

State: RESET

Description	Resets all operational parameters, sensor readings, and state variables to ensure the vehicle is ready for a new parking task without any residual data affecting performance.
Transition	To IDLE: After ensuring all systems are reset, the vehicle returns to the IDLE state, awaiting new commands or to signify the end of the current parking operation.

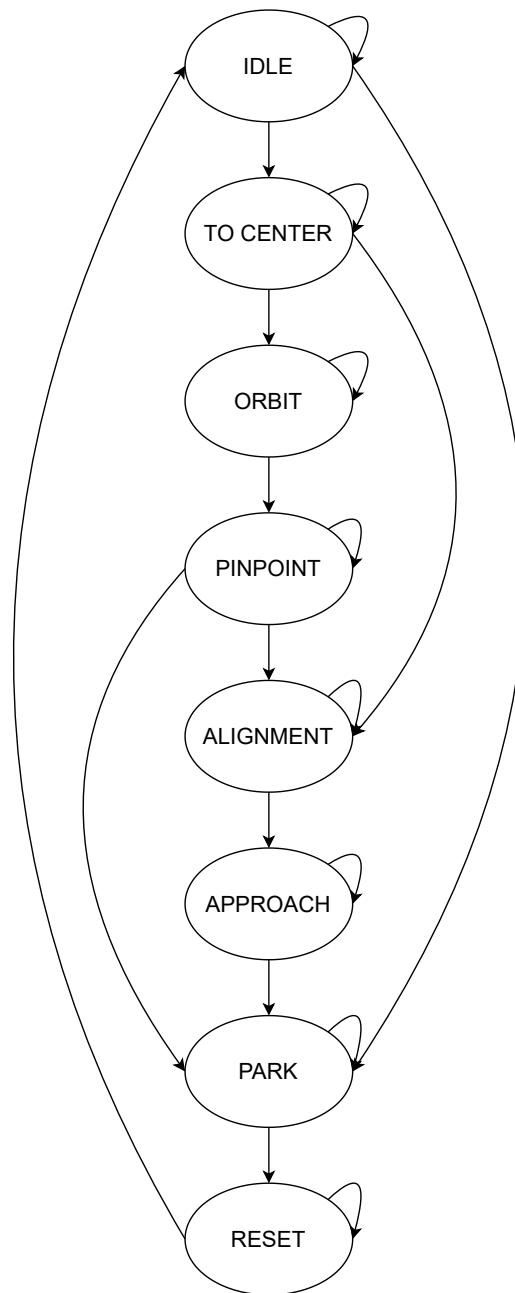


Figure 17: State Machine Diagram

Path Generation

For path planning, we are using Dubin's path generation related techniques after estimating the position of the vehicle and the parking spot with the visual positioning subsystem.

Dubin's paths are a fundamental concept in the field of robotics and autonomous vehicle navigation, particularly relevant in scenarios requiring efficient path planning. These paths

are named after Lester Dubin, who introduced them in the context of airplane navigation, where minimizing the path length is crucial. The paths consist of circular arcs and straight lines, reflecting the natural movements of vehicles with non-holonomic constraints, like cars and robots, which cannot move directly sideways. This constraint also applies to our case.

Dubin's theorem, which is central to this concept, assumes a minimum turning radius for the vehicle, reflecting its physical limitations in making turns. This theorem states that the shortest path connecting two points in space, each with a specified orientation, is one of six possible types: left-straight-left (LSL), left-straight-right (LSR), right-straight-right (RSR), right-straight-left (RSL), right-left-right (RLR), or left-right-left (LRL). Each segment of these paths represents either a straight line or an arc of a circle with a radius equal to the vehicle's minimum turning radius. The six scenarios can be seen in **Figure 18**. From those different scenarios, we determine the best possible combination, which is the minimum length within the drivable area, which ensures the path is the possible minimum length path.

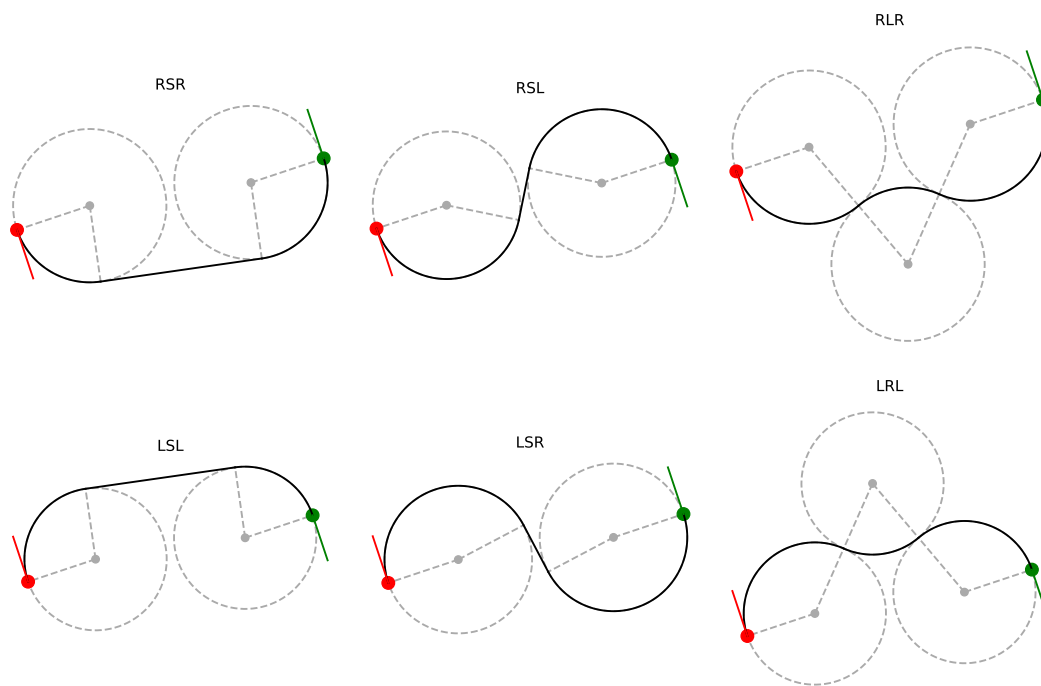


Figure 18: The different possible combination of Dubin's algorithm

This approach is particularly suitable for autonomous vehicles, as it balances computational efficiency with our vehicle's constraints. In our project, we have adapted the standard Dubin's path generation method. Instead of using a fixed radius equal to the vehicle's minimum turning radius, our algorithm increases the radii of the arcs. This increase continues until the path, with a set distance margin, nearly reaches the boundary of the drivable area.

The standard Dubin's path is illustrated in **Figure 19**, where it connects the starting pose to the goal pose using an LSR (left-straight-right) trajectory. Our proposed variation, featuring different turning radii, is demonstrated in **Figure 20**.

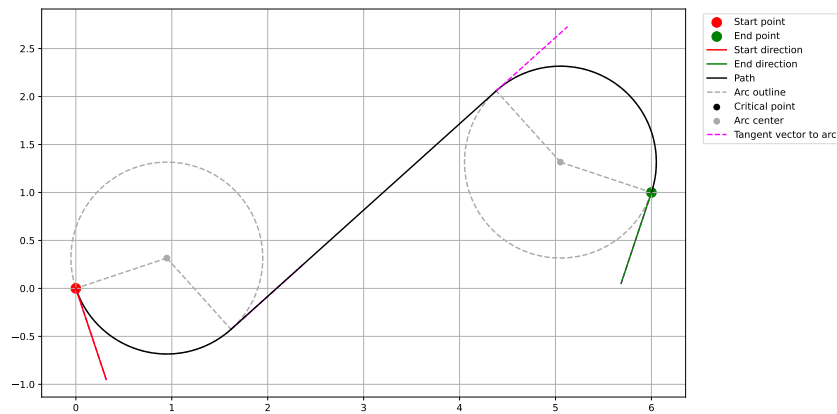


Figure 19: An example Dubin's path generated with same radii turns

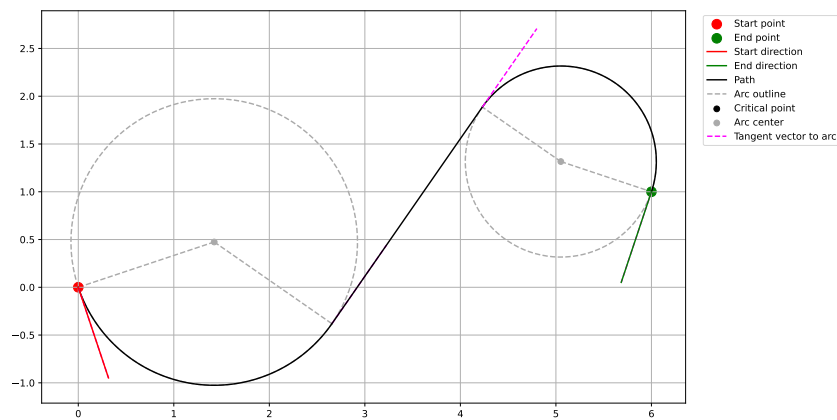


Figure 20: An example modified Dubin's path generated with different radii turns

This change is not an additional step, but a fundamental part of our revised algorithm. It's designed to suit our vehicle's limited ability to make tight turns. By doing this, we make sure the vehicle stays within the safe operational area. Although the area we can drive in is limited and relatively small, this method ensures the path length does not deviate from the optimal length path. Most importantly, it assures a safe and doable path, considering both the size of the parking area and the turning constraints of the vehicle.

Communication Subsystem

The position information for the vehicle and the parking spot obtained from the sensor subsystem needs to be fed into the control subsystem. According to the CO-01 requirement, this information transmission needs to be done wirelessly through a metallic hollow tube. CO-03 requirement obligates ESP8266 modules, which are used for wireless communication, to be placed at least 30 cm away from each other. Therefore, they are placed inside the tube with a 30 cm distance as given in **Figure 21**.

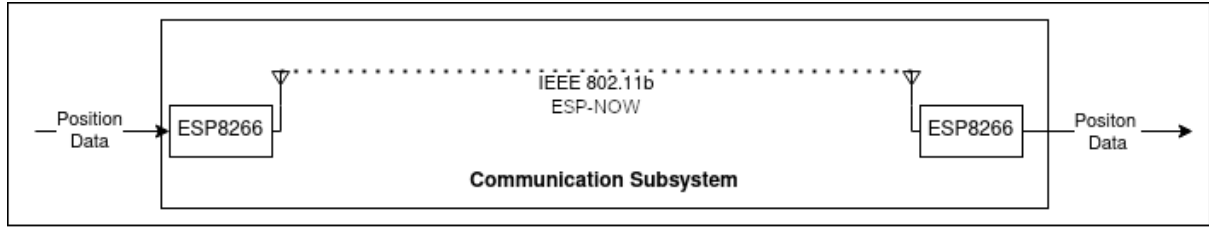


Figure 21: Communication subsystem diagram

Metal Tube: Metal tube works as a circular waveguide. Therefore, we have selected the diameter so that we can transmit 2.4 GHz RF signals used by standard communication protocols such as Wi-Fi and Bluetooth. The frequency formulas for circular waveguides are as follows.

For TE Mode:

$$f_{TE_{m,n}} = \frac{c\chi_{m,n}}{2\pi r}$$

For TM Mode:

$$f_{TM_{m,n}} = \frac{c\chi'_{m,n}}{2\pi r}$$

where c is the speed of light $\chi_{m,n}$ is the n th root of the m th Bessel Function, and $\chi'_{m,n}$ is the n th root of the derivative of the m th Bessel Function.

$f_{TE_{1,1}}$ and $f_{TM_{1,0}}$ act as lower and upper cut-off frequencies, respectively. When frequency is selected as 2.4 GHz, the diameter limits are found as $2r = 7.32$ cm for TE mode and $2r = 9.57$ cm for TM mode. Therefore, this puts an extra constraint on the mechanical design, which is then selected as a metal tube with an inner diameter of 7.8 cm according to the available tubes in the market.

ESP8266: ESP8266 is a low-cost micro-controller which supports IEEE802.11b/g/n standards. In this project, one of them is used as a transmitter in the front, and one of them is used as a receiver in the rear. The one on the front receives relative position according to the parking spot and global 2D position and orientation data with respect to the area boundaries as well as their derivatives for the vehicle from the serial port. Then, it transmits this data as ESP-NOW packets. Using ESP-NOW enables us to send data with low latency, as required in CO-04, since there is no flow control mechanism. It is a protocol built on IEEE802 by Espressif. The packet consists of 10 fields as given in **Table 2**.

The packet structure consists of a magic byte (0x78), which is chosen arbitrarily, followed by two bytes of length information, and a redundancy checksum in the form of CRC16. This packet layout, as shown in **Table 2**, allows us to send packets of varying sizes and transport layers. For detailed information about the actual data fields, please refer to the **Compatibility Analysis** section, as the communication subsystem does not interpret the data beyond the packet format and redundancy check.

Table 2: The packet format

Offset	0	1	2	3 - n+3	n+4	n+5
Field	Magic Byte	Length	Length	Payload	CRC16	CRC16

The ESP module on the rear would receive this signal and decode it according to the IEEE754 standard. Then, it would echo the received signal into the serial port so that the control subsystem could create a trajectory and make the vehicle follow this path accurately.

The receiver ESP, which is at the rear side of the vehicle, is used as an access point. A static IP is set for the module. Then, it continuously listens on a predefined port for ESP-

NOW packets. When it receives a packet, it decodes the data and writes it to the serial port.

The transmitter ESP, which is at the front side of the vehicle, is used as a station. It is connected to the access point ESP using the assigned MAC address and Password. It listens to the serial port continuously, and when it receives data, it encodes the data to a ESP-NOW packet and sends it to the access point ESP using its predefined MAC address.

When the communication tests are performed, it is observed that almost nothing changes with changing distances between ESP modules inside the 50cm tube, so no additional constraints are added to the mechanical design. Also, the error and latency results were good in our expected 40-80 bytes region, even though there were no significant changes in other regions.

Another approach would be using an Infrared (IR) transmitter and receiver. This would eliminate the possibility of interference with RF signals present in the air. On the other hand, this time, we cannot use well-established protocols such as Wi-Fi, which is our primary approach.

Sensor Subsystem

In order to have situational awareness, our robot long thin hauler needs to be equipped with sensors. Therefore, we used a camera, IMU, and magnetometer for sensing the environment. We utilized the camera for visual positioning and localization. Specifically, the camera is used to detect and track the ArUco markers placed within the designated driving area. By recognizing these markers and their known positions, the Visual Positioning subsystem can accurately determine the vehicle's location and orientation within the area.

The IMU (Inertial Measurement Unit) and magnetometer are used to complement the visual localization data from the camera. The IMU provides measurements of the vehicle's linear acceleration and angular velocity, which can be integrated to estimate its position and orientation over time. However, these estimates tend to drift due to sensor noise and biases. To mitigate this issue, we employ a Kalman filter to fuse the IMU data with the more accurate, but less frequent, position and orientation updates from the Visual Positioning subsystem based on the ArUco markers.

The Kalman filter optimally combines the measurements from the IMU and the Visual Positioning subsystem, taking into account their respective uncertainties. This fusion of sensor data provides a more robust and accurate estimate of the vehicle's pose (position and orientation) within the driving area. The filtered pose estimate is then used by the Control subsystem for path planning and maneuvering the vehicle towards the designated parking spot.

Madgwick Filter

The Madgwick filter, developed by Sebastian Madgwick, is a type of orientation filter that is widely used for estimating the orientation of a body in three-dimensional space. It combines data from multiple sensors, i.e., a gyroscope, and a magnetometer, to compute the orientation of the sensor with respect to the Earth's frame. This filter is particularly noted for its efficiency and performance, making it suitable for real-time applications in embedded systems.

It uses a quaternion-based formulation that allows for the representation of orientations without the singularity issues associated with Euler angles. The filter operates by estimating the gradient of the error between the measured orientation and the predicted orientation, and then using this gradient to correct the predicted orientation.

Let $\mathbf{q}_{\omega,t} = [q_w, q_x, q_y, q_z]^T$ be the quaternion representing the orientation of the sensor at time t can be calculated as:

$$q_{\omega,t} = q_{t-1} + \dot{q}_{\omega,t} \Delta t \quad (10)$$

$$= q_{t-1} + \frac{1}{2} q_{t-1} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \Delta t \quad (11)$$

where the $\boldsymbol{\omega}$ is the angular velocity of the sensor and $\otimes$ denotes quaternion multiplication.

The loss function for the accelerometer is given by:

$$f_g(\mathbf{q}, \mathbf{a}) = \begin{bmatrix} 2(q_2 q_4 - q_1 q_3) - a_x \\ 2(q_1 q_2 + q_3 q_4) - a_y \\ 2(0.5 - q_2^2 - q_3^2) - a_z \end{bmatrix} \quad (12)$$

The Jacobian of the loss function with respect to the quaternion is given by:

$$J_g(\mathbf{q}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \quad (13)$$

The gradient for f_g is defined as:

$$\nabla f_g(\mathbf{q}, \mathbf{a}) = J_g(\mathbf{q})^T f_g(\mathbf{q}, \mathbf{a}) \quad (14)$$

The loss function for the magnetometer is given by:

$$f_b(\mathbf{q}, \mathbf{b}, \mathbf{m}) = \begin{bmatrix} 2(q_2 q_4 - q_1 q_3) - m_x \\ 2(q_1 q_2 + q_3 q_4) - m_y \\ 2(0.5 - q_2^2 - q_3^2) - m_z \end{bmatrix} \quad (15)$$

The Jacobian of the loss function with respect to the quaternion is given by:

$$J_b(\mathbf{q}, \mathbf{b}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \quad (16)$$

The gradient for f_b is defined as:

$$\nabla f_b(\mathbf{q}, \mathbf{b}, \mathbf{m}) = J_b(\mathbf{q}, \mathbf{b})^T f_b(\mathbf{q}, \mathbf{b}, \mathbf{m}) \quad (17)$$

The filter combines the gravity and magnetic field measurements to estimate the orientation of the sensor.

$$f_{g,b}(\mathbf{q}, \mathbf{a}, \mathbf{b}, \mathbf{m}) = \begin{bmatrix} f_g(\mathbf{q}, \mathbf{a}) \\ f_b(\mathbf{q}, \mathbf{b}, \mathbf{m}) \end{bmatrix} \quad (18)$$

$$J_{g,b}(\mathbf{q}, \mathbf{b}) = \begin{bmatrix} J_g(\mathbf{q}) \\ J_b(\mathbf{q}, \mathbf{b}) \end{bmatrix} \quad (19)$$

Then the quaternion is updated using the following equation:

$$q_t = q_{t-1} + (\dot{q}_{\omega,t} - \beta \frac{\nabla f_{g,b}}{\|\nabla f_{g,b}\|}) \Delta t \quad (20)$$

$$= q_{t-1} + (\dot{q}_{\omega,t} - \beta \frac{J_{g,b}^T(\mathbf{q}, \mathbf{b}) f_{g,b}(\mathbf{q}, \mathbf{a}, \mathbf{b}, \mathbf{m})}{\|J_{g,b}^T(\mathbf{q}, \mathbf{b}) f_{g,b}(\mathbf{q}, \mathbf{a}, \mathbf{b}, \mathbf{m})\|}) \Delta t \quad (21)$$

Kalman Filter

The Kalman filter is a recursive algorithm that estimates the state of a dynamic system from a series of noisy measurements. It is widely used in various applications, including navigation, tracking, and signal processing. The Kalman filter is an optimal estimator in the sense that it minimizes the mean squared error of the estimated state. Consider a discrete-time linear dynamical system described by the following equations:

$$x_k = F_k x_{k-1} + B_k u_k + w_k \quad (22)$$

$$z_k = H_k x_k + v_k \quad (23)$$

where:

- x_k is the state vector at time k
- F_k is the state transition matrix
- B_k is the control input matrix
- u_k is the control input vector
- w_k is the process noise, assumed to be zero-mean Gaussian with covariance Q_k
- z_k is the measurement vector at time k
- H_k is the measurement matrix
- v_k is the measurement noise, assumed to be zero-mean Gaussian with covariance R_k

The Kalman filter consists of two main steps: prediction and update. In the prediction step, the Kalman filter uses the previous state estimate and the system dynamics to predict the current state:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \quad (24)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (25)$$

where $\hat{x}_{k|k-1}$ is the predicted state estimate at time k given observations up to time $k-1$, and $P_{k|k-1}$ is the predicted state covariance. In the update step, the Kalman filter incorporates the current measurement to correct the predicted state estimate using the following dynamic model of the long thin hauler:

$$p_{x,k} = p_{x,k-1} + v_{x,k-1} \Delta t \quad (26)$$

$$p_{y,k} = p_{y,k-1} + v_{y,k-1} \Delta t \quad (27)$$

$$v_{x,k} = v_{x,k-1} - \Delta v_{x,k}^{bias} + a_{k-1}^\theta \Delta v_x - b_{k-1}^\theta \Delta v_y \quad (28)$$

$$v_{y,k} = v_{y,k-1} - \Delta v_{y,k}^{bias} + b_{k-1}^\theta \Delta v_x + a_{k-1}^\theta \Delta v_y \quad (29)$$

$$\Delta v_{x,k}^{bias} = \Delta v_{x,k}^{bias} \quad (30)$$

$$\Delta v_{y,k}^{bias} = \Delta v_{y,k}^{bias} \quad (31)$$

$$a_k^\theta = a_{k-1}^\theta \quad (32)$$

$$b_k^\theta = b_{k-1}^\theta \quad (33)$$

where K_k is the Kalman gain, $\hat{x}_{k|k}$ is the updated state estimate at time k given observations up to time k , and $P_{k|k}$ is the updated state covariance.

The Kalman filter can be implemented recursively by repeating the prediction and update steps for each new measurement. The initial state estimate $\hat{x}_{0|0}$ and covariance $P_{0|0}$ need to be specified based on prior knowledge or set to reasonable values. In order incorporate with Kalman filter, dynamic motion of the long thin hauler is modeled with constant velocity. In this regard, we get the following matrices:

$$F_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & \Delta v_x & -\Delta v_y \\ 0 & 0 & 0 & 1 & 0 & -1 & \Delta v_x & \Delta v_x \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (34)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (35)$$

$$Q = \text{diag}([\sigma_x, \sigma_y, \sigma_{v_x}, \sigma_{v_y}, \sigma_{\Delta v_x^{bias}}, \sigma_{\Delta v_y^{bias}}, \sigma_{a_\theta}, \sigma_{b_\theta}]) \quad (36)$$

$$R = \text{diag}([\sigma_x^{meas}, \sigma_y^{meas}, \sigma_{a_\theta}^{meas}, \sigma_{b_\theta}^{meas}]) \quad (37)$$

Also, in order to eliminate the effect of outliers in the measurement, Mahalanobis distance metric has been employed as a gating mechanism. The Mahalanobis distance is a multi-dimensional generalization of the idea of measuring how many standard deviations away a point is from the mean of a distribution. In the context of the our kalman filter implementation for long thin hauler, the Mahalanobis distance is used to measure how far the current measurement z_k is from the predicted measurement $H_k \hat{x}_{k|k-1}$, taking into account the covariance of the predicted measurement. Specifically, the Mahalanobis distance d_k is calculated as:

$$\nu_k = z_k - H_k \hat{x}_{k|k-1} \quad (38)$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \quad (39)$$

$$d_k = \sqrt{\nu_k^T S_k^{-1} \nu_k} \quad (40)$$

where:

- ν_k is the measurement residual, which is the difference between the actual measurement z_k and the predicted measurement $H_k \hat{x}_{k|k-1}$.
- S_k is the covariance of the predicted measurement, which is calculated as $H_k P_{k|k-1} H_k^T + R_k$, where $P_{k|k-1}$ is the predicted state covariance, and R_k is the measurement noise covariance.
- S_k^{-1} is the inverse of the predicted measurement covariance matrix.

If the Mahalanobis distance d_k exceeds a predefined threshold β , it means that the current measurement is an outlier and deviates significantly from the expected value based on the predicted state and covariance. In such cases, the update step of the Kalman filter is skipped, and the algorithm proceeds to the next measurement. This helps to make the Kalman filter more robust to outliers and prevent them from corrupting the state estimates. The threshold β is typically chosen based on the desired trade-off between robustness to outliers and the ability to track sudden changes in the system dynamics. A larger value of β makes the filter more tolerant to outliers but may also make it less responsive to sudden changes in the system state. Finally, implementation of our kalman filter design is completed as in the following pseudo-code 1:

Algorithm 1 Kalman Filter Implementation

```

Initialize state estimate:  $\hat{x}_{0|0}$ 
Initialize state covariance:  $P_{0|0}$ 
Define state transition matrix:  $F_k$ 
Define measurement matrix:  $H_k$ 
Set process noise covariance:  $Q_k$ 
Set measurement noise covariance:  $R_k$ 
Set mahalanobis distance threshold:  $\beta$ 
for each new measurement  $z_k$  do
     $\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}$  ▷ Prediction step
     $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$ 
     $\nu_k = z_k - H_k \hat{x}_{k|k-1}$ 
     $S_k = H_k P_{k|k-1} H_k^T + R_k$ 
     $d_k = \sqrt{\nu_k^T S_k^{-1} \nu_k}$  ▷ Mahalanobis distance
    if  $d_k > \beta$  then
        Skip update step and proceed to next measurement
    else
         $K_k = P_{k|k-1} H_k^T S_k^{-1}$  ▷ Update step
         $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \nu_k$ 
         $P_{k|k} = (I - K_k H_k) P_{k|k-1}$ 
    end if
end for

```

Design Requirements

System Requirements

Table 3: System Requirements

	Code	Requirement
Functional Requirements	S-01	The vehicle must park autonomously without any human intervention.
	S-02	The vehicle must make continuous navigation without any stops.
	S-03	The vehicle must move only in the forward direction.
	S-04	The vehicle must be competent to find its relative location with respect to the operation area.
	S-05	The actuator system for navigation must be on the back, and the sensors system must be on the front of the vehicle.
	S-06	The sensor and actuator systems must communicate wirelessly through the metallic or metal-covered hollow tube.
	S-07	The chassis of the vehicle must be constructible from easily accessible materials.
	S-08	The power source of the vehicle must be rechargeable and replaceable.
	S-09	There must be a parking spot that can be placed on any side of the operation area.
Performance Requirements	S-10	The sensor and actuator compartments must be able to communicate at a distance greater than 30 cm.
	S-11	The communication system must be able to handle a transmission rate greater than 10kB/s.
	S-12	The vehicle must complete its parking under 1 minute.
	S-13	The vehicle must be able to operate continuously for 30 minutes.
	S-14	The vehicle must be able to park with a precision of at least 1 cm.
Physical Requirements	S-15	The body of the vehicle must possess at least 50 cm metallic or metal-covered hollow tube.
	S-16	The vehicle length must be between 50cm and 60cm, and the vehicle width must be between 7cm and 10cm.
	S-17	The weight of the vehicle must be less than 3kg.
	S-18	The operation area must be movable and 3m by 3m in size.
	S-19	Width of the parking area must be less than 150% of vehicle width.

Subsystem Requirements

Mechanical Subsystem

Table 4: Mechanical Subsystem Requirements

Code	Requirement	Related System Req.
M-01	The chassis must incorporate a metallic or metal-covered hollow tube of at least 50 cm long.	S-15
M-02	Overall vehicle weight must be kept under 3kg.	S-17
M-03	Vehicle dimensions must ensure a height between 50cm and 60cm and a width between 7cm and 10cm.	S-16

Power Subsystem

Table 5: Power Subsystem Requirements

Code	Requirement	Related System Req.
P-01	The power source must be a rechargeable battery capable of sustaining vehicle operation for a minimum of 60 minutes.	S-13
P-02	Battery must be easily replaceable without specialized tools.	S-08
P-03	The power system should include a mechanism to prevent overcharging of the battery.	S-08

Visual Positioning Subsystem

Table 6: Visual Positioning Subsystem Requirements

Code	Requirement	Related System Req.
V-01	The visual positioning system must accurately identify the parking spot within the operation area.	S-04, S-09
V-02	It should provide real-time data to the control subsystem to facilitate autonomous parking within 1 minute.	S-12
V-03	The system must be integrated with the sensor subsystem to enhance positioning accuracy.	S-04, S-05
V-04	The visual positioning system must be capable of operating in variable light conditions.	S-13
V-05	It should estimate the position of the vehicle with at least 10cm accuracy	S-14, S-19

Control Subsystem

Table 7: Control Subsystem Requirements

Code	Requirement	Related System Req.
C-01	The control subsystem must process sensor and positioning data to park the vehicle autonomously.	S-01
C-02	It must ensure continuous navigation without stops.	S-02
C-03	The control algorithm must enable the vehicle to complete parking maneuvers within 1 minute.	S-02, S-12

Communication Subsystem

Table 8: Communication Subsystem Requirements

Code	Requirement	Related System Req.
CO-01	Wireless communication between sensors and actuators must be maintained through the hollow tube.	S-05, S-06
CO-02	The system must support a minimum transmission rate of 10kb/s.	S-11
CO-03	The communication range must exceed 60 cm.	S-10
CO-04	Communication protocols must ensure data integrity and minimize latency.	S-02

Sensor Subsystem

Table 9: Sensor Subsystem Requirements

Code	Requirement	Related System Req.
SE-01	The IMU must provide accurate and stable measurements of the vehicle's orientation and acceleration.	S-04, S-13
SE-02	The IMU data should be filtered to reduce noise and improve accuracy.	S-12
SE-03	The sensor subsystem must be calibrated to ensure data accuracy over the 60 minute operational period.	S-13
SE-04	The IMU should be positioned to maximize its effectiveness and minimize interference from other subsystems.	S-13
SE-05	The IMU must be capable of operating within the environmental conditions expected in the operational area.	S-12

Compatibility Analysis

Our system is composed of multiple components, each responsible for a specific task. These components are designed to be modular, allowing for easy replacement or upgrade. To achieve this, we have strictly followed certain rules during the design and implementation of the subsystems.

Transport Agnostic Communication Protocol

In **Figure 3**, we present the block diagram of our system. The components of the system communicate with each other through various transport layers, including UART, I2C, ESP-NOW, Wi-Fi, TCP, and PWM. Each communication protocol serves a specific purpose within the system. For instance, the PWM protocol is utilized to control the speed of the motors, while the I2C protocol facilitates communication between the microcontroller and the IMU. Additionally, certain communication protocols are specific to particular components, while others are employed for general communication between all components.

Packet Structure

The ESP-NOW, Wi-Fi, TCP, and UART communication protocols in our system are based on a simple packet structure. Each packet consists of a header and a payload. The header contains information about redundancy and payload length, while the payload contains the actual data being transmitted. This packet structure ensures efficient and reliable communication between components by maintaining consistency and organization in data transmission. Additionally, the packet structure enables easy identification and processing of incoming data, as each packet contains all the necessary information for processing.

Payload Organization

Furthermore, each payload is structured as another packet that contains the message type and the corresponding data fields. This hierarchical packet structure allows for easy identification and extraction of the message type and data within the payload, ensuring efficient and accurate transmission of data between components. As a result, seamless communication and coordination among subsystems are facilitated.

Modularity and Flexibility

This design approach allows for easy integration of new components into the system. Each component is designed to be modular and independent, ensuring flexibility, scalability, and robustness. By adhering to these design principles, we have created a system that can accommodate new components and functionalities with minimal effort. This modular design approach ensures that our system is adaptable to changing requirements and can be easily upgraded or expanded in the future.

The modularity of our system also enables easy debugging and testing of individual components. Each component can be tested independently, allowing for efficient identification and resolution of issues. This modular testing approach ensures that the system is thoroughly tested and validated before integration, resulting in a reliable and robust system.

Simulation and Testing

We have employed Software In The Loop (SITL) and Hardware In The Loop (HITL) simulations to thoroughly test and validate our system. These simulations allow us to test the system in a controlled environment, enabling us to identify and resolve issues before deployment. Since our communication protocol is independent of the transport layer, we can use the same protocol for both simulations. For example, we can have the control firmware running on the Raspberry Pico controller while running the simulation and the sensor fusion algorithm on the computer. This setup allows us to test the control firmware without needing the actual vehicle itself.

The most important example of this has been the testing and validation of the EKF and sensor fusion algorithm. By isolating the EKF component and testing it independently, we were able to identify and resolve issues related to sensor fusion and data processing. Since we are able to record and replay sensor data, we can simulate various scenarios and test the EKF algorithm under different conditions. This approach has enabled us to fine-tune the EKF algorithm and ensure accurate and reliable estimation of the vehicle's state.

We believe that this communication protocol and highly compatible signal interfaces is one of the key factors that have contributed to the success of our system. By employing a diverse range of communication protocols and maintaining a consistent packet structure, we have achieved efficient and reliable inter-component communication. This has enabled seamless integration of components, easy testing and validation, and robust performance of the system. We are confident that our system is well-designed and well-implemented, and we are excited to see it in action during the final demonstration.

For the sake of completeness, the related packet types and their fields are listed in **Table .** Those packets are conveyed over the communication protocol described in **Design Description.**

Table 10: Packet Descriptions

Packet ID	0x0	
Description	Reserved	
Field	Type	Description
reserved	u8	Reserved field
Packet ID	0x1	
Description	Filtered IMU orientation and preintegrated acceleration data	
qw	f32	Quaternion component W
qx	f32	Quaternion component X
qy	f32	Quaternion component Y
qz	f32	Quaternion component Z
dvx	f32	Preintegrated acceleration X
dvy	f32	Preintegrated acceleration Y
dvz	f32	Preintegrated acceleration Z
dt	f32	Time delta
Packet ID	0x2	
Description	Raw IMU measurements	
accel_x	f32	Raw acceleration X
accel_y	f32	Raw acceleration Y
accel_z	f32	Raw acceleration Z
gyro_x	f32	Raw gyroscope X

gyro_y	f32	Raw gyroscope Y
gyro_z	f32	Raw gyroscope Z
mag_x	f32	Raw magnetometer X
mag_y	f32	Raw magnetometer Y
mag_z	f32	Raw magnetometer Z
Packet ID	0x6	
Description	Normalized motor setpoints	
left	f32	Normalized left motor setpoint
right	f32	Normalized right motor setpoint
Packet ID	0x7	
Description	Raw motor setpoints	
left	i16	Raw left motor setpoint
right	i16	Raw right motor setpoint
Packet ID	0x8	
Description	Desired velocity and yaw setpoints	
vel	f32	Desired velocity
yaw	f32	Desired yaw
Packet ID	0xa	
Description	Set a parameter to a float value	
param	i32	Parameter ID
value	f32	Float value to set
Packet ID	0xb	
Description	Set the state of the onboard LEDs	
index	u8	LED index
state	u8	LED state (on/off)
Packet ID	0xc	
Description	Debug information from the control subsystem	
current_d	f32	Current distance
current_yaw	f32	Current yaw
desired_d	f32	Desired distance
desired_yaw	f32	Desired yaw
current_vel	f32	Current velocity
current_w	f32	Current angular velocity
desired_vel	f32	Desired velocity
desired_w	f32	Desired angular velocity
u_v	f32	Control output for velocity
u_w	f32	Control output for angular velocity
u_r	f32	Control output for right motor
u_l	f32	Control output for left motor
Packet ID	0xd	
Description	Set a parameter to a unsigned int value	
param	i32	Parameter ID
value	u32	Unsigned int value to set
Packet ID	0xe	
Description	Set a parameter to a signed int value	

param	i32	Parameter ID
value	i32	Signed int value to set
Packet ID	0xf	
Description	Motor output values of the control subsystem	
left	f32	Left motor output value
right	f32	Right motor output value
Packet ID	0x10	
Description	Reboot the system	
reserved	u8	Reserved field
Packet ID	0x11	
Description	Estimated pose from the sensor fusion node	
x	f32	Estimated pose X coordinate
y	f32	Estimated pose Y coordinate
yaw	f32	Estimated pose yaw
Packet ID	0x12	
Description	State of the simulation	
w	f32	Angular velocity
v	f32	Linear velocity
theta	f32	Orientation
x	f32	X coordinate
y	f32	Y coordinate
rate	f32	Update rate

Power Management

The power management system of our system is designed to be compatible with a wide range of power sources and loads.

Battery and Power Distribution

The power management system consists of a battery, a power distribution board, and voltage regulators. The battery provides power to the entire system, while the power distribution board distributes power to the various components of the system. The voltage regulators ensure that each component receives the correct voltage and current, preventing damage and ensuring reliable operation.

The batteries used in our system are 4S and 2S lithium ion batteries. These batteries are commonly used in RC vehicles and drones, making them readily available and easy to replace. The batteries are connected to the power distribution board, which distributes power to the various components of the system. The power distribution board is designed to be compatible with different battery types and voltages, ensuring flexibility and compatibility with a wide range of power sources.

Voltage Regulators

The voltage regulators used in our system are designed to be compatible with a wide range of input voltages and loads. The voltage regulators are adjustable, allowing for precise control of the output voltage and current. This ensures that each component receives the correct voltage and current, preventing damage and ensuring reliable operation. Additionally, the voltage regulators are designed to be efficient and reliable, ensuring long battery life and stable operation.

Modularity and Robustness

The power management system is designed to be modular and independent, ensuring flexibility and scalability. Each component of the power management system can be easily replaced or upgraded, allowing for customization and optimization. The power management system is also designed to be robust and reliable, ensuring stable operation under various conditions. By adhering to these design principles, we have created a power management system that is compatible with a wide range of power sources and loads, ensuring reliable operation and long battery life.

Compliance with Standards

In engineering the Long Thin Hauler, we utilize ESP-NOW, a protocol designed for short-range, low-power communication between devices, for wireless communication. Although ESP-NOW is not an official IEEE standard like IEEE 802.11 (Wi-Fi), it operates on the 2.4 GHz Wi-Fi band and is optimized for efficient data transfer in scenarios where Wi-Fi networks might not be available.

In terms of electrical safety, our design conforms to the IEC 61010-1 standard. This standard outlines safety requirements for electrical equipment used for measurement, control, and laboratory use. By adhering to this standard, we ensure that our electrical components meet the necessary safety criteria.

For quality management, we follow to the ISO 9001 standard. This standard establishes requirements for a quality management system and helps us ensure that our processes and procedures consistently meet customer expectations. Additionally, for automotive safety, we follow the ISO 26262 standard. This standard addresses functional safety in automotive electronics and guides us in developing safe and reliable systems for our autonomous vehicle.

To mitigate hazards due to both intended and unintended behaviors of the autonomous vehicle, we also follow the guidelines of ISO 21448. This standard provides guidance on the safety of the intended functionality (SOTIF) of autonomous vehicles. By following to this standard, we ensure that our vehicle's behavior is predictable and safe for both passengers and other road users.

By following to these standards, we are able to develop a market-ready product that prioritizes safety, reliability, and quality.

Compliance with Requirements

Our team has put in significant effort to design a vehicle that meets and, where possible, exceeds the basic requirements. In this report, we outline the steps taken to ensure the vehicle's functionality and reliability. Using a variety of sensors and programming, we have anticipated and attempted to mitigate potential issues. Our goal is to demonstrate the vehicle's capabilities in achieving the set objectives. We present our findings with the acknowledgment that while the vehicle performs well in tests, real-world applications may present unforeseen challenges that we are prepared to tackle.

Detailed Compliance with Requirements Analysis

Our journey in creating the Long Thin Hauler has been challenging yet enlightening. As a team, we have put in our best efforts to build a self-driving vehicle that meets the project's criteria. In the following part, we explain how our vehicle aligns with the established guidelines for its function, performance, and structure. We discuss the design decisions we have made, the innovative approaches we have explored, and the testing processes we have undertaken. Our aim is to demonstrate the vehicle's compliance with the requirements and our commitment to delivering a high-quality product.

- **Functional Requirements Compliance**

- S-01: Autonomous parking without human intervention is verified by the final tests, confirming the vehicle's ability to accurately self-locate and navigate using Aruco Markers within the test area.
- S-02: Continuous navigation is validated by the integration of control and sensor subsystems tests.
- S-03: The actuator system's capacity for forward motion only is confirmed by integration of control and sensor subsystems tests as the controller is designed so that the motors cannot make backward motion, which verified in the tests, whose output should reflect accurate motion response to control inputs.
- S-04: High-precision location finding is demonstrated through visual positioning tests as it showed that under 1m, the error is less than 1 cm, affirming the effective use of Aruco Markers for spatial awareness.
- S-05: The sensor and actuator layout compliance is assured through system design checks trivially.
- S-06: The Communication test, which showed the packet loss is less than 1 percent, also verifies wireless communication over the 50 cm metal-covered hollow tube, using two ESP8266 modules, ensuring the system meets seamless integration requirements.
- S-07: Using 3D-printed components for the chassis is justified trivially by the design.
- S-08: The rechargeable and replaceable power source is justified by the positioning of the batteries in the vehicle and using the connectors during the production of the battery packs.
- S-09: Parking spot placement flexibility is achieved by area design as the parking spot is designed to be slidable through the area. Also, it is verified by the parking maneuver tests, which include different scenarios for the location of the parking spot.

- **Performance Requirements Compliance**

- S-10 to S-11: ESP8266 module communication capabilities exceed the stipulated range and data rate, as evidenced by successful communication tests, which showed that the communication is maintained even at 42cm with data rate of 100 kb/s, ensuring that the vehicle maintains robust data exchange.
- S-12: The final tests confirmed the swift and precise parking in under 1 minute, demonstrating consistent performance.

- S-13: The power test at full power validated that the vehicle's power system can support continuous operation for longer than 1 hour, surpassing the operational endurance requirement.
- S-14: Precision parking tests showed that the vehicle can park with a precision of 1 cm or less, attesting to the efficacy of the navigation and control algorithms.
- **Physical Requirements Compliance**
 - S-15 to S-19: The vehicle's physical attributes align with the project specifications, with tests validating the structural and dimensional integrity. The width deviation to 9.5 cm is acknowledged, and while it marginally exceeds S-16's upper limit, functionality is unaffected, as confirmed by the wireless communication tests, ensuring the deviation does not compromise the vehicle's capability to communicate within the 3x3 meter operation area effectively.

In the process of developing the Long Thin Hauler, we have systematically achieved the functional, performance, and physical requirements while considering necessary trade-offs.

- S-01: Achieved, as the vehicle can autonomously park without human input.
- S-02: Fulfilled, with the vehicle designed for continuous navigation without stops.
- S-03: Met, as the vehicle is restricted to forward motion only.
- S-04: Complied with, by enabling the vehicle to locate itself in the operation area.
- S-05: Achieved by placing the actuator system at the back and the sensor system at the front.
- S-06: The sensor and actuator systems' wireless communication is confirmed through the metallic or metal-covered tube.
- S-07: Realized, with the chassis constructed from easily accessible 3D printing materials.
- S-08: The vehicle's power source is both rechargeable and replaceable.
- S-09: The design allows for a parking spot to be placed on any side of the operation area.

Trade-offs and Engineering Decisions:

- S-16 and S-19: A slight increase in vehicle width (9.5 cm) was a trade-off to accommodate critical internal components, slightly exceeding the preferred range of 7 cm to 9 cm. This decision was made considering the trade-off between internal space requirements and the vehicle's physical footprint, which still allows for efficient parking and operation within the 3m x 3m operation area.
- S-14: To achieve parking with at least 1 cm precision, a trade-off was made in terms of processing power and sensor accuracy, ensuring the vehicle could reliably determine its precise position for parking.
- S-13: The requirement for 30-minute continuous operation necessitated a larger battery capacity, which in turn influenced the vehicle's weight and size. This is an example of balancing power needs with the physical design constraints.

Potential Conflicting Requirements:

- S-05 vs. S-06: The requirement for front-mounted sensors and rear-mounted actuators (S-05) needed to be balanced with the need for reliable communication through a metallic tube (S-06). The potential for signal interference was mitigated by choosing high-quality communication modules and positioning them to minimize loss.
- S-08 vs. S-13: The need for a replaceable and rechargeable power source (S-08) could potentially conflict with the demand for a 30-minute continuous operation (S-13). This was resolved by selecting high-capacity batteries that could be easily swapped out, ensuring both long operation times and the ability to quickly recharge or replace the power source.

These considerations and adjustments demonstrate our approach to address multiple requirements and deliver a vehicle that not only meets the criteria but also maintains high standards of functionality and performance.

Table 11: Requirements and Design Decisions

Requirements	Design Decision	Verification
S-01	Feedbacking the sensor output to the control system.	✓
S-02	Making sure that at least one motor is moving.	✓
S-03	Prohibiting backward motion of motors as the controller's output.	✓
S-04	The use of Aruco Markers to get position and orientation information.	✓
S-05	The mechanical system is designed so that the actuators and the sensors are placed at the rear and front, respectively.	✓
S-06	ESP8266 are used, and the modules are placed in the hollow tube, ensuring the rear and back are making wireless communication	✓
S-07	The vehicle is manufactured using the 3D printer.	✓
S-08	To ensure rechargeability, Lithium Ion batteries are used. Also, the mechanical design and battery pack cabling ensure replaceability.	✓
S-09	Parking spot is designed to be slid across the area's borders.	✓
S-10	The ESP8266 can communicate in the range of several meters.	✓
S-11	The ESP8266 can transfer information at our desired rate.	✓
S-12	The actuators are selected to be powerful so that the vehicle's motion is not bulky.	✓
S-13	The high-capacity batteries are selected as a power source.	✓
S-14	Localization with Aruco markers ensures the small error under 1 m, and the lane placement in the borders of the parking spot.	✓
S-15	The aluminum tube is bought with a length of 50 cm.	✓
S-16	The vehicle's width, which is 9.5cm, and the length exceeds the physical requirements, yet it was necessary to place the wide components of the vehicle.	X
S-17	The aluminum is selected as the metallic tube, which is both durable and lightweight. Also, during the design of 3D printed parts, their weights are carefully considered to avoid any additional burdens.	✓
S-18	The area is designed using lightweight material, i.e., polypropylene, and magnets enabled us to make smaller pieces of area borders. Those add up to a mobile, lightweight area.	✓
S-19	The parking spot is designed %150 of the width of the vehicle to compensate for parking errors.	✓

Robustness of the System Against Possible Error Sources

Our effort is to create a resilient vehicle against errors by integrating multiple sensors and employing advanced data processing, aiming to enhance reliability and precision in autonomous navigation and parking within the project's objectives and resources.

- **Enhanced Localization with IMU Integration** The primary method for vehicle localization involves using cameras to detect Aruco Markers. Recognizing the potential for errors due to motion blur and variable lighting conditions, an Inertial Measurement Unit (IMU) is also incorporated. The IMU provides additional data points, including acceleration and rotational rates, which are crucial for:
 - Compensating for temporal discrepancies in visual data, especially when the camera's view is obscured or blurred.
 - Reducing reliance on visual markers alone, thus mitigating errors in positioning due to visual obstructions or marker recognition failures.
- **Precise Parking with Color Detection** For the critical phase of parking, the area is distinguished with a blue color, surrounded by an orange border. This color-based differentiation serves multiple purposes:
 - It allows for a more refined detection of the parking boundary, enabling the vehicle to adjust its positioning with greater precision.
 - In scenarios where the vehicle's alignment with the parking spot is less than optimal, the contrast between the blue parking area and the orange border provides a clear visual cue for realignment.
 - The color detection algorithm acts as a secondary validation mechanism to confirm the vehicle's accurate placement within the parking spot, enhancing the precision beyond that achievable with marker-based localization alone.
- **Error Filtering with Extended Kalman Filter (EKF)** The use of an Extended Kalman Filter plays a crucial role in synthesizing data from multiple sensors and mitigating the impact of outliers, especially those arising from image processing errors. The EKF effectively:
 - Combines the strengths of visual localization with IMU data, producing a more accurate and stable estimate of the vehicle's state.
 - Filters out anomalous readings that deviate significantly from the predicted state, ensuring that temporary lapses in data quality do not derail the navigation process.
 - Adapts to the inherent uncertainties in sensor measurements, dynamically adjusting its estimates to reflect the most probable current state of the vehicle.
- **Controller Security Measures** Beyond the EKF, the vehicle's control system is reinforced with security measures designed to reject outliers and maintain stable operation under varying conditions. These measures ensure that the control commands issued to the vehicle are based on verified and reliable data, thereby preventing erratic behavior prompted by transient sensor errors or environmental disturbances.

By combining different systems and approaches, we have designed a strategy aimed at minimizing errors. The use of IMU sensors with visual markers for positioning, along with color detection for improved parking accuracy, helps the vehicle stay on track even in the presence of sensor errors or challenging environmental conditions. Incorporating an Extended Kalman Filter (EKF) into our data processing helps filter out inconsistent data and combines information from various sources to maintain the vehicle's understanding of its location and movement. We have also implemented additional precautions in the vehicle's control systems to handle unexpected situations, ensuring smooth operation and minimal deviation from the intended path. This comprehensive strategy for error reduction reflects our commitment to creating a precise and reliable vehicle. Through the careful integration of sensor data and advanced error filtering, our project demonstrates our dedication to high standards in self-driving vehicle development.

We have approached the vehicle requirements systematically, ensuring autonomous navigation and parking with reliability and precision. The vehicle's size was adjusted slightly to accommodate internal components while maintaining operational effectiveness. To enhance error resistance, we incorporated multiple sensors, including an IMU, to improve location determination. We also focused on the seamless integration and functionality of the vehicle's subsystems, supported by comprehensive testing. These tests validate the vehicle's adherence to functional, performance, and structural guidelines.

Safety Issues and Precautions

Our design has incorporated several safety measures to address potential safety issues associated with the autonomous vehicle. One of them was ensuring electrical safety and preventing any malfunctions that could lead to hazardous situations. To mitigate these risks, we have designed the power management system with modular and robust components, allowing for easy replacement or upgrades, thus ensuring stable operation under various conditions. Voltage regulators were used to provide precise control of output voltage and current, preventing damage to components and ensuring reliable operation.

Moreover, the inclusion of multiple sensors, such as cameras and IMUs, along with advanced data processing techniques like the Kalman Filter, enhances the system's robustness against errors. This setup helps in reducing reliance on visual markers alone and provides additional data points for accurate localization and navigation, thereby minimizing the chances of positioning errors due to visual obstructions or marker recognition failures.

Furthermore, our Battery Management System (BMS) includes fail-safes that promptly detect and respond to battery failures by halting the power supply to prevent potential risks. This is crucial for maintaining system integrity and preventing hazardous conditions that could arise from battery malfunctions.

Also, to prevent the vehicle from going outside the designated area and to minimize potential harm to people and the environment, we have implemented several critical safety features. Firstly, we have installed safety switches on both the front and rear parts of the vehicle. These switches act as emergency stop mechanisms, allowing for immediate halting of the vehicle's motion if it approaches the boundaries of its operational area. This ensures that the vehicle can be quickly and effectively stopped to prevent it from entering unauthorized zones .

Additionally, we have equipped the vehicle with LEDs and a buzzer system that provide visual and auditory signals indicating the vehicle's motion. The LEDs are strategically placed to be visible from all angles, and they change color to indicate different operational states, such as moving forward, reversing, or stopping. The buzzer emits distinct sounds to alert nearby individuals of the vehicle's movements, thereby enhancing awareness and reducing the risk of accidental collisions . These measures collectively contribute to a safer operational environment by preventing unintended excursions and providing clear indications of the vehicle's status to those nearby. By implementing these measures, we have ensured that our design prioritizes safety and reliability throughout its operation.

Test Results

Visual Positioning

We have completed our testing process of the visual positioning subsystem as planned, focusing on evaluating the performance of our visual positioning system under a variety of conditions. This involved altering the distance between the marker and the camera and adjusting the camera's yaw and pitch angles. Our objective was to gather data on how changes in camera height and orientation impact measurement accuracy and cause perspective distortion.

In our static testing, we collected data at different yaw angles, beginning with zero degrees and moving through 15 and 30 degrees. This step helped us understand how the yaw angle affects the system's performance. For dynamic testing, we moved the marker towards the camera, continuously tracking its position and the camera's orientation. By analyzing the video with Tracker software, we correlated the visual data with the marker's actual movement.

We observed a variation in position error based on marker distance during the testing phase. Specifically, when the marker is placed at a greater distance, the position error increases. Our tests showed that with the camera's yaw angle set to zero, the error in the x-coordinate remains within acceptable limits. Even at a distance of 2 meters, the x-coordinate error was less than 15 centimeters, which is satisfactory given the dimensions of our test area. This occurrence is relatively infrequent, considering the layout. Up to 1m distance, the errors at both coordinates are less than 2.5cm, satisfying the V-05 requirement.

However, it's important to note that the yaw error reached approximately 10 degrees, significantly impacting the y-coordinate error. This increase in error is due to the vehicle's method of solving its pose relative to the marker. Essentially, the vehicle determines the marker's position within its coordinate frame and then computes its own pose in relation to the marker. This process involves an implicit coordinate frame transformation.

The pitch angle has a minor impact on the accuracy of our measurements. A pitch angle of 30 degrees was found to be slightly more advantageous compared to 10 degrees. Therefore, we incorporated a 30-degree pitch angle into our vehicle's mechanical design. Those effects can be seen in **Figure 22**.

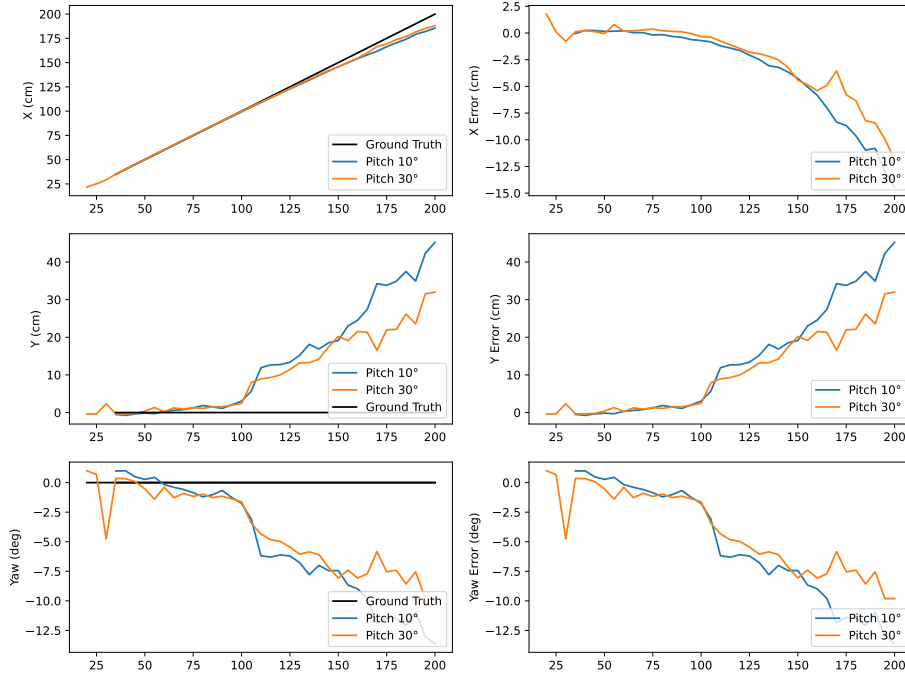


Figure 22: Measurement Comparisons and Ground Truths at Various Distances and Pitch Angles

The relationship between the y-coordinate error and the yaw error can be mathematically represented as $y_{error} \approx x \cdot \tan(\theta_{error})$. This formula aligns with our observations and the theoretical understanding of the problem. It's evident that minimizing yaw error is crucial for improving the overall accuracy of our visual positioning subsystem.

To mitigate this issue, we plan to incorporate IMU (Inertial Measurement Unit) heading measurements, as outlined in the 'Visual Positioning Subsystem' section. This strategy is effectively constrain the system to two spatial dimensions, thus avoiding ill-conditioned scenarios related to rotational errors. We incorporate this aspect of the solution to V-03.

According to the dynamic tests we performed, we observed that the vehicle's movement does not induce a significant level of distortion or motion blur in the images captured by our visual positioning system. This finding is crucial as it indicates that the quality of the visual data remains consistently high, even under motion. As a result, the effectiveness of the visual positioning system is not compromised by the vehicle's movement, as can be seen in **Figure 23**. Since the visual positioning subsystem is able to provide real-time measurements, it satisfies the subsystem-level requirement V-02. Finally, detailed test procedure and results can be found in **Appendix 2: Test Document: Visual Positioning**.

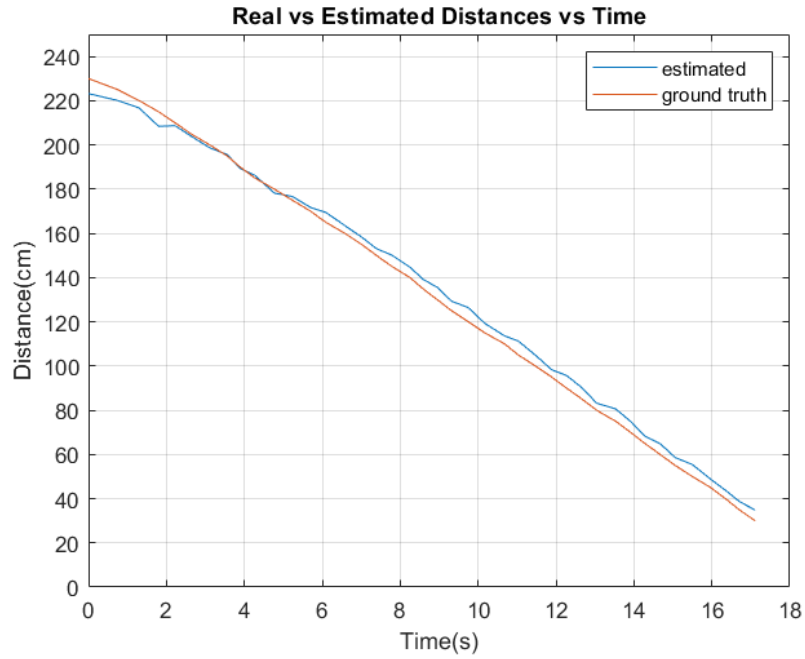


Figure 23: Comparison of Real and Estimated Distances Over Time During Vehicle Movement

Wireless

The testing environment consists of two ESP modules, a computer, a metallic hollow tube, and a ruler. The metallic hollow tube is placed on a flat surface. ESP8266 modules are fixed on their test points on the ruler. Then, the ruler is placed inside of the hollow tube, as 0 cm of the ruler coincides with one of the edges, so the communication through the hollow tube is tested as required in CO-01. Also, they are connected to a computer with Micro USB cables. The diagram for the test environment is given in **Figure 24**.

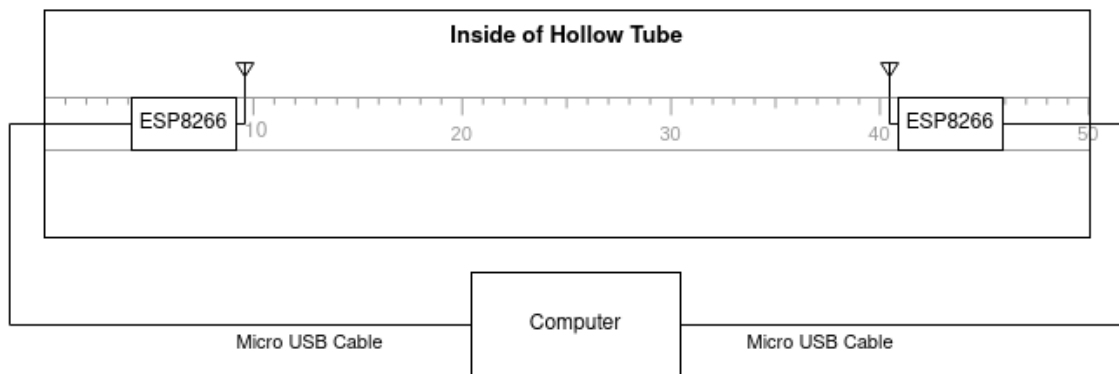


Figure 24: Test Configuration Diagram

The computer has two Python scripts, which are transmitter.py and receiver.py. The transmitter.py sends the parameters through the Serial Port, and the transmitting ESP sends ESP-NOW packets according to the data it got from the serial port. The receiving ESP counts

how many ESP-NOW packets it received and echos the timestamp of the packet it received. The transmitting ESP gets this echoed packet and calculates the latency as half of the round trip time. Transmitting ESP outputs the latency to the serial port. Also, the receiving ESP outputs the total number of packets it received to the serial port.

The test is done to determine how the distance, data rate, and packet size change the communication performance.

The distance between the ESP modules is required to be at least 30 cm, as required in CO-03, so the distances are changed to 30 cm, 36 cm, and 42 cm. After the tests, it is seen that the performance does not change much with these changes in the distance. The smallest error is obtained at a distance of 30 cm. Therefore, these ESP modules is placed as close to 30 cm distance as possible, as long as the mechanical design constraints permit.

Table 12: Communication System Performance for Different Distances

Distance	Average Lost Packet Number Percentage	Average Latency
30 cm	0.31%	1.15
36 cm	0.54%	1.15
42 cm	0.49%	1.17

The data rate is required to be at least 10 kb/s, as required in CO-03. For a 100-byte packet size, this would require the packet rate to be 100 packet/s. The packet rate is changed from 10 packet/s to 260 packet/s with 50 packet/s step size. After the test, it is observed that a peak latency occurs at the smallest data rate of 10 bytes/s, and then it drops. After that, although there is a slight increase in the average, there does not seem to be much correlation between them. The reason for the peak at 10 bytes/s can be that there might be a buffer in the ESP, and it may be waiting until that buffer is full.

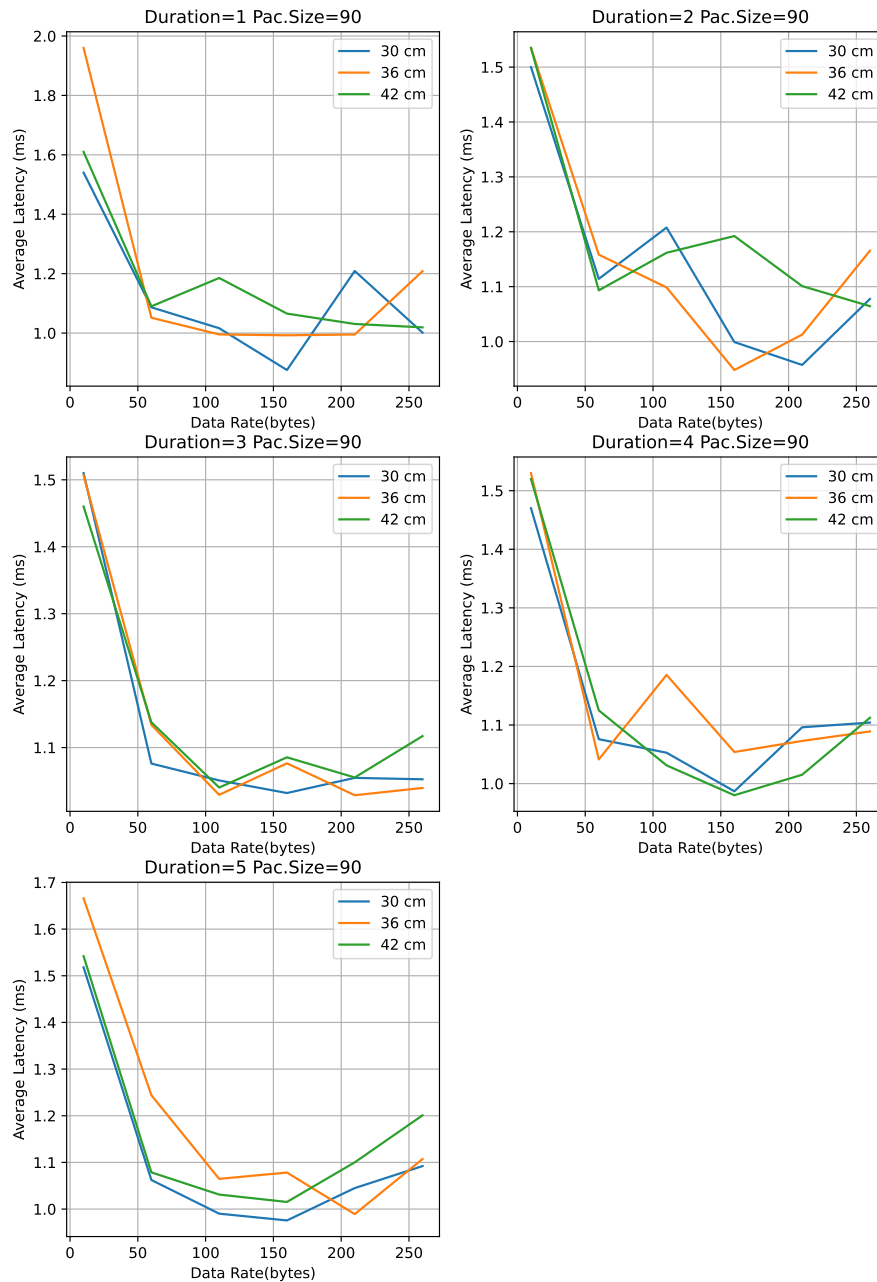


Figure 25: Graph of Data Rate vs Average Latency with same Packet Size and Different Durations

The packet size in the communication system is expected to be 42 bytes or 80 bytes. However, larger packet lengths are also tested in case any additional information is needed to be transmitted. Also, dividing a packet into smaller packets is also considered, so the packet length is changed from 30 bytes to 150 bytes with 20 bytes of step size. From the test results, it does not seem that there is a correlation between packet size and error or latency.

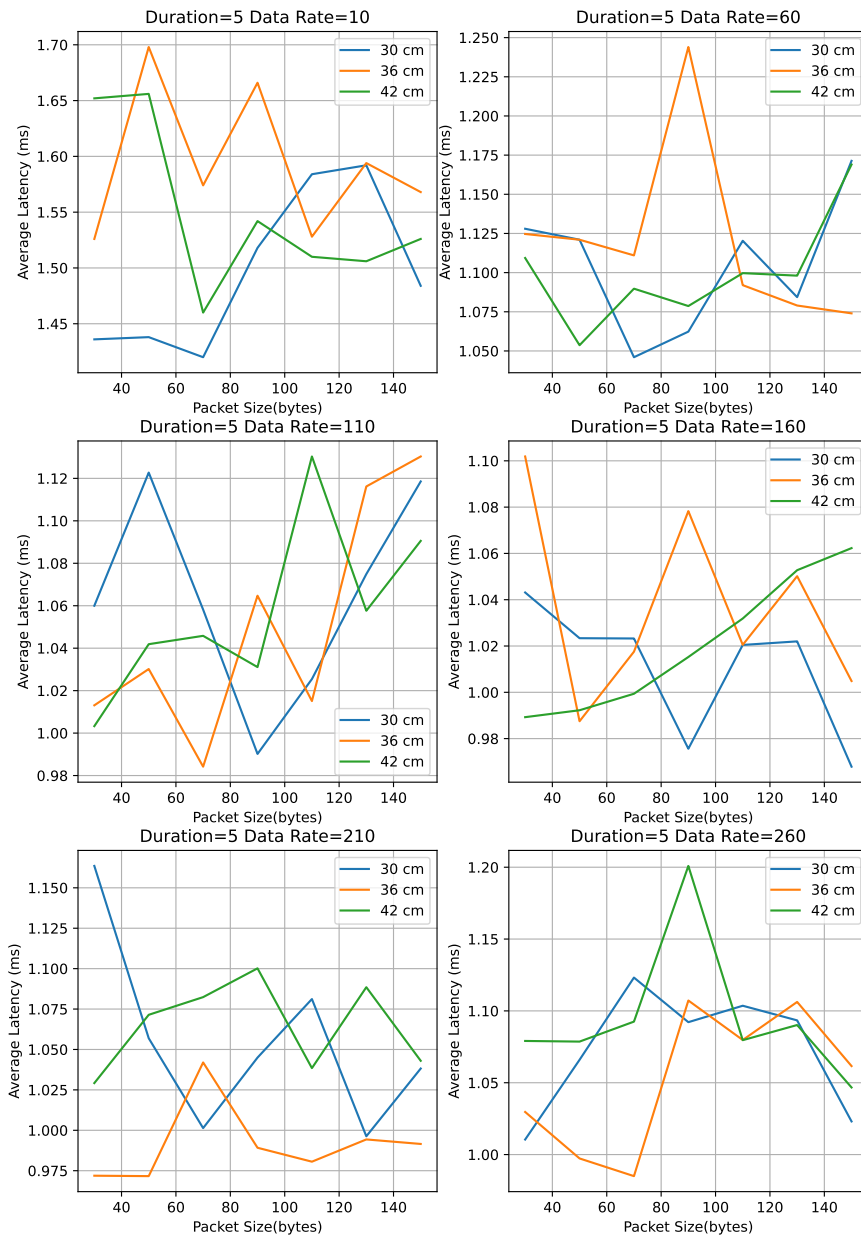


Figure 26: Packet Size vs Average Latency with Same Duration and Changing Data Rates

Also, the time duration is changed from 1s to 5s with 1s timestamps. With this test, we wanted to learn if the ESP buffers start to saturate at a steady state. In the test results, there does not seem to be any correlation between the results and the time duration.

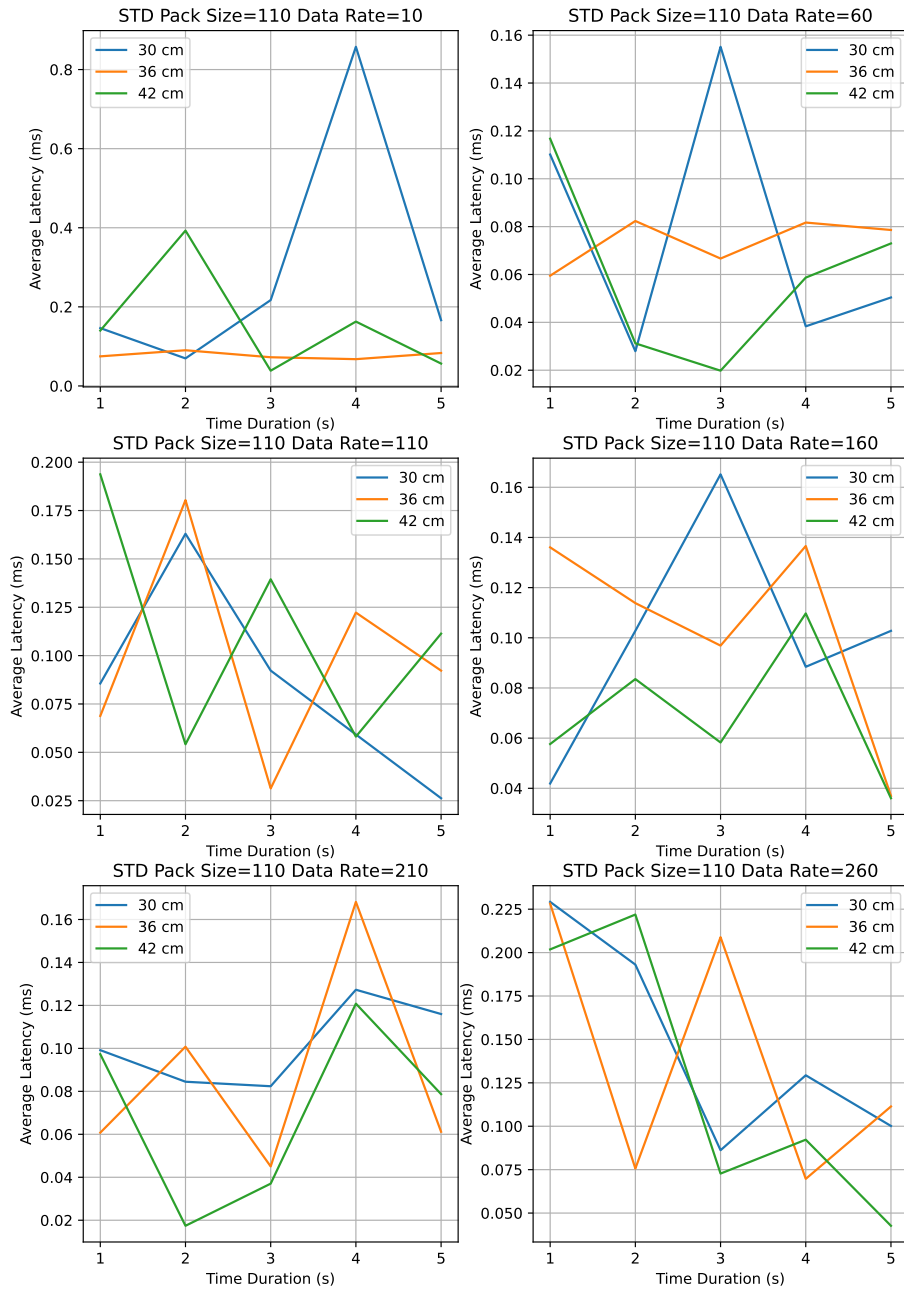


Figure 27: Standart Deviation Graph of Time Duration vs Average Latency with Same Packet Size and Changing Data Rate

From **Figure 25**, **Figure 26**, **Figure 27**, we observe that our metrics are uncorrelated from each other, so there is no effect on the transmission quality. Finally, detailed test procedure and results can be found in **Appendix 3: Test Document: Wireless**.

Camera Vibration & Exposure

In our exploration of the camera exposure settings for the vehicle's localization system, we conducted tests to observe the impact of exposure time on image quality, specifically look-

ing at the clarity of Aruco markers used for localization. We utilized two different exposure settings: short and long.

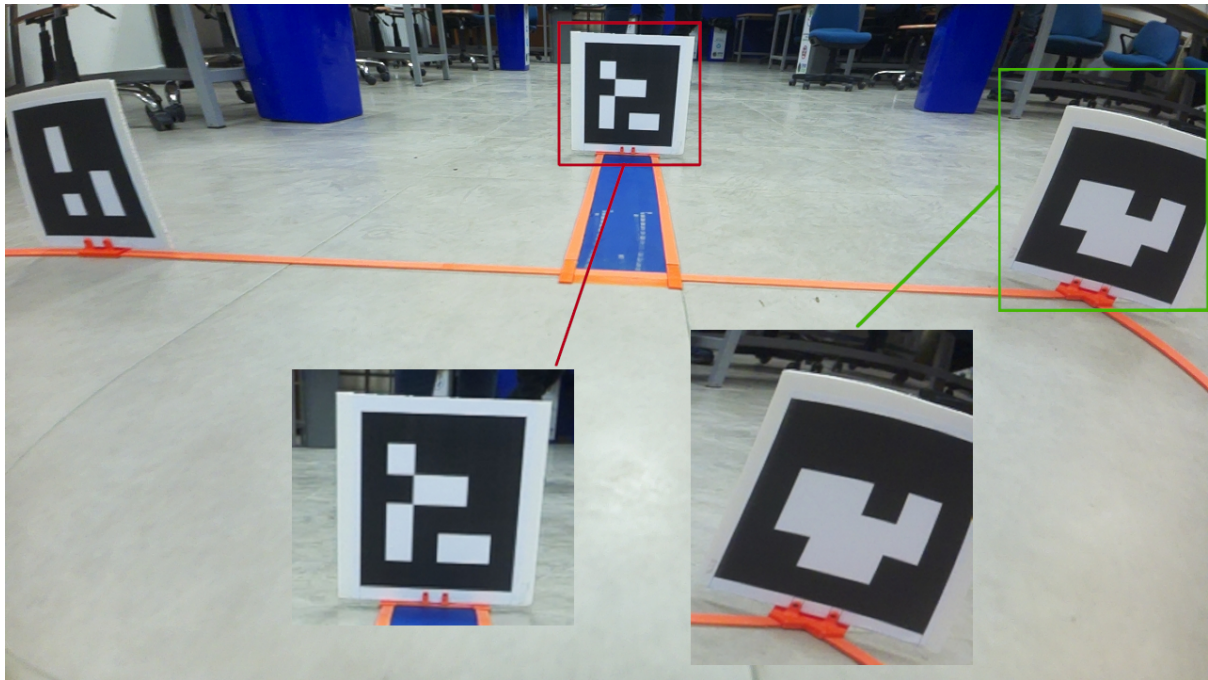


Figure 28: Blur When Exposure is Set to Short

Short Exposure Time: With the exposure time set to short, the images captured by the camera showed a marked reduction in motion blur (**Figure 28**). The Aruco markers are distinctly sharper and the details are more precise, facilitating accurate detection and localization by the system. The short exposure setting also helped mitigate the impact of any mechanical vibrations from the vehicle, as the quicker shutter speed reduced the time for such vibrations to affect the image stability. However, one potential downside noted in the lower-light segments of the test area was the inadequacy of ambient light to produce a well-exposed image. This resulted in some markers being less visible, which could pose challenges in consistently dark environments.

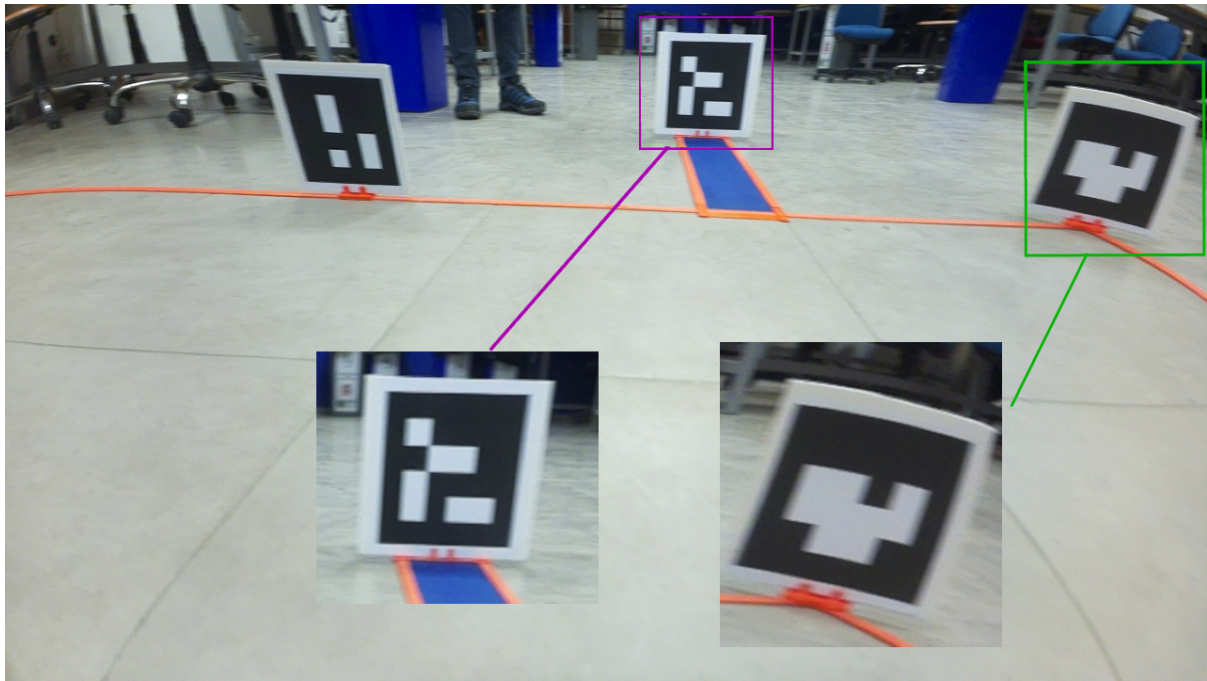


Figure 29: Blur When Exposure is Set to Long

Long Exposure Time: When the exposure time was set to long, the images showed increased motion blur (**Figure 29**). This was particularly noticeable in the details of the Aruco markers, where the clarity was reduced, leading to difficulties in precise localization. The long exposure allowed more light to reach the camera sensor, which is beneficial in low-light conditions, but the increased blur from the longer exposure countered this benefit by degrading the localization system's accuracy. The mechanical vibrations of the vehicle were more pronounced with long exposure settings, as the camera sensor's extended capture time increased the sensitivity to any movement.

The comparison between the two images demonstrates the trade-off between exposure time and image clarity, especially when mechanical vibrations are present. Short exposure times offer clearer images with less motion blur, enhancing the localization process's accuracy but may suffer in low-light conditions. Conversely, long exposure times may work better in darker settings but can introduce more motion blur, negatively impacting localization precision.

Based on these observations, it's essential to optimize the camera's exposure time for the specific lighting conditions in which the vehicle will operate, ensuring that the Aruco markers are always captured with sufficient clarity for accurate localization. Finally, detailed test procedure and results can be found in **Appendix 4: Test Document: Camera Vibration & Exposure**.

IMU Attitude Test

The Inertial Measurement Unit (IMU) was subjected to rigorous testing to evaluate its performance in determining the vehicle's attitude (**Figure 30, Figure 31**).

The data is analyzed by looking at the change in the IMU angle at each protractor angle. The expected change was 45 degrees since the protractor was rotated by 45 degrees each time. The data is compared with the expected change to determine the accuracy of the IMU.

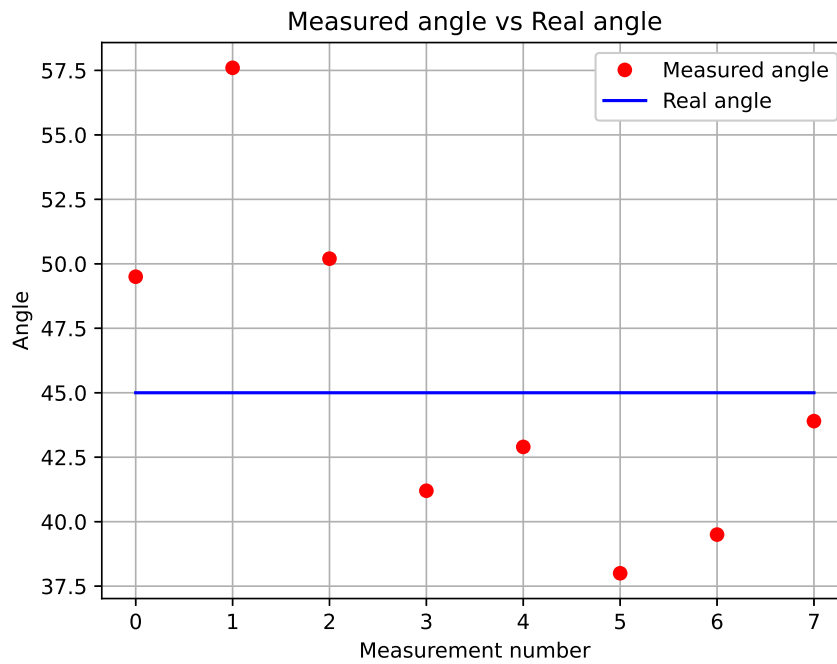


Figure 30: Changes in IMU angle at each protractor angle at steady state

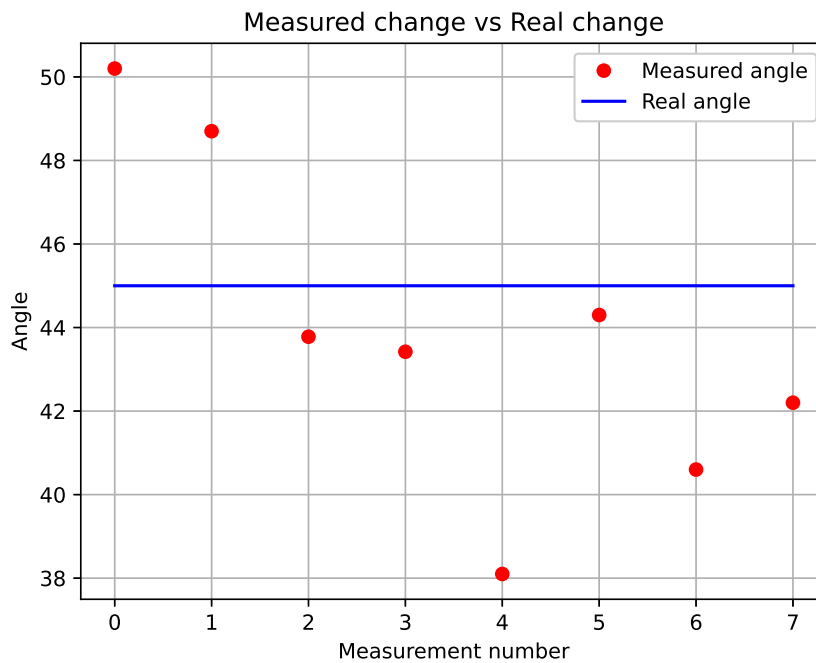


Figure 31: Changes in IMU angle at each protractor angle at transient state

It is seen that the non-uniformity of the magnetic field affects the IMU measurements. The IMU measurements are not linearly related to the protractor angle, and the worstly affected measurements have 180 degrees which shows the magnetic field affect is maximum at this angle.

This affect is experienced less at the transient state, which is expected since the magnetic field affect is less and the gyroscope measurements are also effective at the transient state. On the other hand, the yaw angle converges to the magnetometer measurements at the steady state.

In our system, the IMU will be consistently in motion since the vehicle will be moving. Therefore, the IMU will be at the transient state most of the time. Therefore, the IMU will be able to provide more accurate measurements. Also, the main position estimation will be done by the visual positioning system, and the IMU will be used to correct the drift in the yaw angle. Therefore, our system will be less sensitive to the errors in the IMU measurements. Finally, detailed test procedure and results can be found in **Appendix 5: Test Document: IMU Attitude**.

Path planning

The path generation algorithm underwent rigorous testing, producing a total of 600,000 paths across 100,000 scenarios, with each scenario generating six possible paths. The algorithm successfully generated all paths, demonstrating its reliability and robustness in creating feasible paths for the robot. The paths adhered to critical constraints such as the allowable drivable area and the minimum turning radii, confirming the algorithm's capability to produce viable paths within specified parameters. Additionally, the paths complied with geometric constraints, ensuring practical implementability.

The algorithm efficiently generates paths in real-time, with a failure rate of 4.56% out of 100,000 scenarios, making it suitable for the project's requirements.

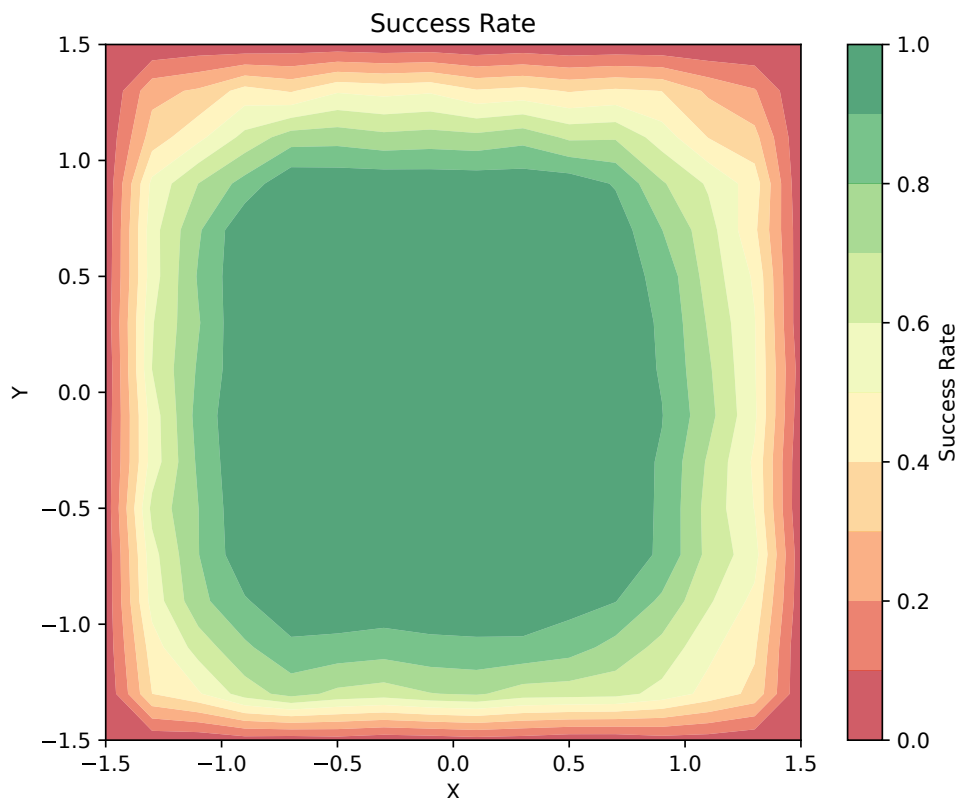


Figure 32: The heatmap of the success rate versus the starting points

According to **Figure 32**, the algorithm consistently achieves a high success rate across different starting points, demonstrating its robustness and adaptability to various scenarios. The failure rate is concentrated near the edges of the area, where constraints are more stringent.

The test results confirm that the path generation algorithm successfully meets the project's requirements by generating feasible paths that comply with specified constraints and demonstrate its reliability in navigating challenging scenarios.

The path generation algorithm performs efficiently, generating feasible paths while adhering to specified constraints, making it a valuable component of the overall system. Finally, detailed test procedure and results can be found in **Appendix 6: Test Document: Path planning**.

Power Test at Full Power

In our assessment of the vehicle's power system, a series of tests were carried out to evaluate the endurance of the battery packs when subjected to a full-power discharge. The focus was on maintaining a constant 6W load and observing the battery voltage behavior over a prolonged period.

Full Power Discharge: The test involved actively discharging both the active and backup battery packs using a smart charger programmed to simulate a high-power draw condition. This was done to evaluate the battery packs' ability to sustain long operational periods under significant power demands.

Voltage Behavior Under Load: Throughout the 240-minute test, the voltage of each cell was monitored and recorded every 10 minutes. The objective was to track the voltage decline under continuous load and to determine the operating time frame until the BMS safety protocols were triggered.

BMS Activation: The BMS demonstrated its capability by activating safety measures at approximately 3.3V for both active and backup battery cells, a higher voltage than the charger's setting, indicative of the BMS's conservative threshold for action.

Operational Endurance: The batteries displayed consistent performance, with the voltage readings confirming their ability to withstand extended durations of high-power discharge. The results from this test offer a quantitative measure of the battery packs' reliability and the BMS's protective efficiency.

The data collected serves to validate the battery packs' capability to handle real-world scenarios where sustained power output is critical. The test also highlights the integral role of the BMS in managing the power system's safety and ensuring the batteries' longevity. Finally, detailed test procedure and results can be found in **Appendix 7: Test Document: Power Test at Full Power**.

Battery Failsafe

In our evaluation of the Battery Management System (BMS) for the vehicle, we conducted a series of tests to assess the BMS's ability to detect and respond to simulated battery failures. The focus was on discharging the batteries to a predetermined threshold voltage and monitoring the BMS's reaction.

Simulated Battery Failures: For the test, both active and backup batteries were methodically discharged using a smart charger set to a 3.15V discharging mode. The objective

was to simulate a failing condition by reaching the threshold voltage and to observe the BMS's ability to accurately identify this decrease in voltage and activate the necessary safety measures.

BMS Response: Upon reaching the threshold voltage, the BMS promptly detected the condition and engaged its safety protocols, effectively halting the power supply to avert potential risks or damage. This response was consistent across all batteries tested. Furthermore, the BMS recorded the event and issued alerts, ensuring clear and timely communication for any required intervention.

Measured Battery Voltage: Post-test measurements revealed that the battery voltage after the BMS safety stop was within the range of 3.27-3.30V for all backup and active batteries. This indicates that the BMS initiated the safety stop slightly above the set discharging mode of the smart charger, providing a margin of safety.

System Reliability: The consistency in the BMS's response across different batteries underscores its reliability in monitoring the battery's condition and upholding the vehicle's safety standards. The ability of the BMS to detect and respond to simulated battery failures is a testament to its preparedness for real-world scenarios where battery health is paramount for the vehicle's operation.

The test results affirm the BMS's functionality in safeguarding the vehicle's electrical systems by effectively managing battery conditions. The system's successful detection and response to battery failures reinforce the vehicle's operational safety and reliability. Regular monitoring and testing of the BMS are essential to ensure its continued efficacy in protecting the vehicle. Finally, detailed test procedure and results can be found in **Appendix 8: Test Document: Battery Failsafe Test**.

Parking Maneuver

The parking maneuver is a critical aspect of the Longthin Hauler project, particularly due to the necessity of parking the vehicle in a narrow space. This test aimed to evaluate the vehicle's parking performance and its limitations, focusing on two key factors: the starting yaw angle and the distance to the parking spot.

We have limited the yaw angle to a maximum of ± 45 degrees. If the angle goes beyond this, it means there is a problem with the parking approach, and the vehicle's controller is not properly adjusting the orientation. Also, the parking process must start from a specific distance. The minimum distance is set at 30 cm, which is close enough to the parking spot. If the distance is less than 30 cm, it means the controller is not fixing poor approaches well. The maximum distance to start parking is 90 cm. This distance is fairly far, making sure the vehicle is in parking mode. If the distance is more than 90 cm, it gives too much space, making it too easy to park. Additionally, we cannot guarantee the vehicle is in parking mode due to the limitations of our image processing method.

Table 13: The test results

Starting distance	Starting yaw angle(ψ)	Completion
30	-45	Fail
30	0	Success
30	45	Fail
60	-45	Success
60	0	Success
60	45	Success

Starting distance	Starting yaw angle(ψ)	Completion
90	-45	Success
90	0	Success
90	45	Success

During the testing phase, we positioned the vehicle according to the specified parameters and observed its behavior. Each parking attempt took approximately 12 seconds, demonstrating a high level of efficiency in the vehicle's control system, which consistently executed the parking maneuver within a relatively short duration. The test results, as shown in **Table 13**, revealed that the vehicle struggled to park when the starting yaw angle was ± 45 degrees at a distance of 30 cm. This specific combination proved challenging for the vehicle's control system. However, in all other scenarios, the vehicle successfully parked in the designated spot. This overall success, except for the extreme yaw angle at close proximity, highlights the effectiveness and reliability of the control system under varied conditions.

The failure observed at ± 45 degrees yaw angle at a distance of 30 cm can be attributed to the implementation of our parking maneuver algorithm. Our algorithm leverages the high-contrast color of the parking spot, specifically utilizing the orange borders and dark blue areas. By detecting the contours and fitting lines to the orange borders, the algorithm is able to identify the parking space accurately. However, this method imposes certain limitations on the lookahead distance.

The lookahead distance, which is the distance ahead that the vehicle's control system considers for planning its movements, has limited adjustability in our current approach. When we try to extend the lookahead distance significantly, the parking spot can become undetectable by the vehicle's camera. This issue is aggravated by the relatively high yaw angle limit of 45 degrees.

With a yaw angle of 45 degrees, the vehicle's controller attempts to sharply turn the vehicle toward the parking spot to align it correctly. However, this increased turning effort can sometimes cause the parking spot to move out of the lookahead distance. When this happens, the vehicle's control system loses visual contact with the parking spot, which can lead to difficulties in accurately planning and executing the parking maneuver.

As the vehicle struggles to detect the parking spot within the extended lookahead distance, it may fail to align itself correctly, resulting in unsuccessful parking attempts. This scenario underscores the challenge of balancing the lookahead distance and yaw angle to ensure consistent detection and alignment with the parking spot.

To address this limitation, we have decided to implement a threshold-based solution. If the vehicle's initial yaw angle exceeds the threshold, indicating that it is too large to allow for successful parking, the vehicle first realigns itself. This involves reversing and adjusting its position and orientation to ensure it meets the required starting parameters. Once properly aligned, the vehicle can then proceed to park in the designated spot. This additional step ensures that the vehicle can handle a wider range of initial conditions while still achieving successful parking maneuvers. Finally, detailed test procedure and results can be found in **Appendix 9: Test Document: Parking Maneuver Test**.

Final Mission

The final mission test is a crucial component of the Longthin Hauler project, highlighting the vehicle's ability to autonomously park in a predesignated spot within one minute. This test is crucial as it directly assesses the car's precision, efficiency, and overall capability

in executing its primary mission. By focusing on this task, we aim to validate the vehicle's autonomous navigation systems and ensure it meets the time constraints, which are essential for operational success. The outcomes of this test provides valuable insights into the system's reliability and effectiveness, which are vital for the project's overall viability.

For the final mission test, we evaluated the vehicle to assess the vehicle's performance under different conditions. Specifically, we have varied three key factors: the starting yaw angle, and the starting x and y positions. These adjustments allow us to evaluate how well the vehicle can adapt to different initial setups and successfully complete the mission. We have recorded the time it takes for the vehicle to complete the mission under these different conditions. It's important to note that the vehicle must start randomly and park in the designated spot within one minute. This adds an extra layer of complexity as the vehicle must be capable of quickly and accurately orienting itself from an unpredictable starting point. The goal here is to ensure that the vehicle's autonomous systems can handle a variety of scenarios and still achieve the objective within the tight time constraint, which is crucial for the practical application of the Longthin Hauler project.

Table 14: The final mission test results

Position x (m)	Position y (m)	Orientation (degrees)	Mission Time(sec)
45	45	0	6.03
45	115	0	38.4
45	185	0	40.5
115	45	0	11.75
115	115	0	26.26
115	185	0	32.92
185	45	0	32.03
185	115	0	31.98
185	185	0	19.38
45	45	120	37.06
45	115	120	31.8
45	185	120	21.64
115	45	120	Fail
115	45	120	28.81
115	115	120	38.38
115	185	120	Fail
115	185	120	32.92
185	45	120	33.38
185	115	120	31.96
185	185	120	25.56
45	45	240	34.32
45	115	240	34.18
45	185	240	38.15
115	45	240	38.08
115	115	240	Fail
115	115	240	30.42
115	185	240	36.03
185	45	240	Multiple Fail
185	45	240	30.17

Position x (m)	Position y (m)	Orientation (degrees)	Mission Time(sec)
185	115	240	27.25
185	185	240	32.56
		Average	30.44

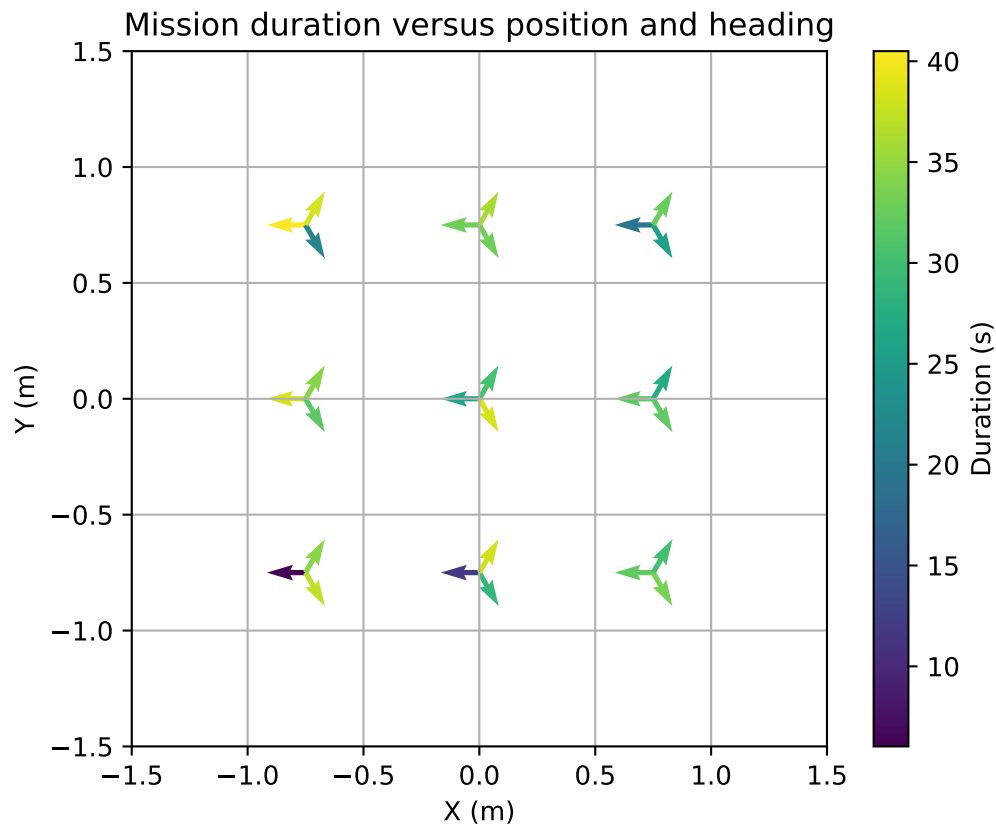


Figure 33: Parking Duration vs position graph

During the final mission test, we positioned the vehicle according to the various initial conditions and recorded its performance. The results shown in **Table 14**. On average, each mission attempt took about 30.44 seconds to complete. You can find a clear graph of the effect of initial position from **Figure 33**, the blue color indicates it took less time to complete the mission, and the yellow color indicates it took relatively long time to complete the mission. There were several instances where the vehicle failed to park within the required time frame. Specifically, the vehicle struggled in the following scenarios: $x = 115$ cm, $y = 45$ cm, angle = 120 degrees; $x = 115$ cm, $y = 185$ cm, angle = 120 degrees; and $x = 115$ cm, $y = 115$ cm, angle = 240 degrees. Additionally, the test scenario with $x = 185$ cm, $y = 45$ cm, and angle = 240 degrees resulted in multiple failures. Despite those failures, the vehicle managed to park successfully in other scenarios, demonstrating decent adaptability and efficiency overall.

Due to the numerous unsuccessful attempts that we have encountered, we undertook a thorough and meticulous investigation, followed by an extensive troubleshooting process, aimed at identifying the underlying issues within the code. Through this comprehensive examination, it became evident that the primary cause of these repeated failures was the inadequate calibration of the vehicle. This deficiency in calibration was directly responsible for the unsuccessful outcomes observed during our testing phases.

Following the implementation of precise and accurate calibration, the previous failures were successfully mitigated, and our vehicle consistently managed to navigate and park in the designated spot without encountering any issues. This proper calibration process ensured that the vehicle's performance was optimal, leading to seamless parking maneuvers each time. As a result, the vehicle was able to park effortlessly in the intended location, demonstrating the effectiveness of the calibration adjustments in eliminating the prior complications.

Finally, the results demonstrated that the integration of the various subsystems has been highly successful, enabling our vehicle to complete the mission exactly as specified. This indicates not only the efficiency of the vehicle's performance but also the seamless cooperation and functionality of its integrated components, fulfilling the mission's objectives as requested. The detailed test procedure and results can be found in **Appendix 10: Test Document: Final Mission Test**.

Table 15: Requirements and Tests

Requirements	Test Outputs	Verification
S-01	Integration of control and sensor subsystems test addresses the vehicle to follow a generated path autonomously and parking maneuver test checks if the vehicle can park autonomously	✓
S-02	Continuous motion is checked during the control and sensor subsystems test	✓
S-03	Path planning tests ensures a path for only forward motion is created, and control and sensor subsystems test checks whether the vehicle moves only in forward direction during its motion.	✓
S-04	Visual positioning tests are done to see if the vehicle can find its location relative to the area.	✓
S-05	It is a mechanical design constraint, not something to be tested.	X
S-06	Wireless test ensures the wireless communication through the metallic tube is successful.	✓
S-07	It is a mechanical design constraint, not something to be tested.	X
S-08	It is a design constraint, not something to be tested.	X
S-09	During parking maneuver test, the parking spot is placed at random places.	✓
S-10	Wireless test checks the communication is successful at distances greater than 30 cm.	✓
S-11	Wireless test checks the communication is successful at rates greater than 10 kB/s.	✓
S-12	The final tests considers the scenario where the parking should be completed under 1 minute.	✓
S-13	The power test at full power checks how long the system can operate at the worst case scenario.	✓
S-14	The parking maneuver test checks how precise the vehicle can park to the parking spot.	✓
S-15	It is checked after when the product is delivered, it was not something to create a test case	X
S-16	It is checked after the manufacturing, it was not something to create a test case	X
S-17	It is checked after the manufacturing, it was not something to create a test case	X
S-18	The moveability was a design constrained and the length of the area is checked after the manufacturing, it was not something to create a test case	X
S-19	The parking maneuver test checks if the vehicle can enter a parking spot with maximum width is 150 % of the vehicle width	✓

List of Deliverables

- **Vehicle:** An autonomous vehicle, the main product of the project, is now available. It is designed to move continuously and smoothly in the forward direction only. The vehicle measures over 50 centimeters in length and less than 10 centimeters in width. Sensors are installed in the front, while actuators are located at the back, with communication between them maintained wirelessly through a metal tube.
- **Drivable Area with Parking Spot:** The boundaries for the 9 m² square drivable area along with the parking spot markers have been set. These components are portable, allowing the system to be easily implemented in various locations.
- **Software:** The source codes have been provided with all necessary explanations. They are designed to support bug fixes and future updates, enabling customers to tailor the product to their changing and newly emerging needs, such as adjustments in area size or timeliness requirements.
- **Test Document:** A test document, outlining quantifiable measures of performance and the reliability of the final product, is now part of the delivery. This document verifies that the vehicle operates as required and details the conditions under which these performances are achieved. It also aids customers in understanding how performance varies with different conditions, providing test results for various scenarios and serving as a crucial resource for troubleshooting.
- **User Manual:** A user manual has been included to guide the product's use. It features step-by-step instructions for setting up the system, operating it, and troubleshooting common problems. The user manual can be found in **Appendix 11: User Manual**

Resource Management

Cost Analysis

Table 16: Component Costs

Component	Quantity	Total Price	Justifications
Raspberry Pi 5	1	\$100.7	It is the main component that process camera data which provides the main positioning information, so a high price is selected for better performance.
Raspberry Pi Cooler	1	\$6.6	Cooling is essential for maintaining the Raspberry Pi's performance under load, and the price is not high.
Raspberry Pi Camera Module 3	1	\$37.15	High POV and high resolution with low distortion is required for good positioning information, so the cost is justified by better performance.
Raspberry Pi Pico	2	\$9.82	It is a low price microcontroller that can run parallel processes such as sensor fusion and data relay to ESP modules
ESP8266	2	\$2.94	They are low price communication modules that has enough performance for our small range operations.
Pololu Motor 3.6 kg-cm 140 rpm	2	\$42.8	They provide the high torque we need with sufficient speed. Considering there is not much DC motor in the market, their high price is justified.
Metal Tube 50 cm, 8 cm diameter	1	\$9.32	A metallic hollow tube with 7-9 cm diameter and minimum of 50 cm length was required. In the market, it was the cheapest one.
Wheel	2	\$10	It can provide sufficient traction with its moderate price.
18650 Li-ion Battery	6	\$29.56	High-capacity batteries provide power for longer time. Their cost is justified by their energy density, rechargeability, and lifespan.
LN298N Motor Driver	1	\$2.08	It is capable of handling the needed motor control with low price.
MPU9250 IMU	1	\$8.59	It provides the sufficiently well attitude data for the navigation and control with its moderate price.
LM2596 Buck Converter	2	\$1.74	It is capable of regulating the voltage to provide stable power to the electronic components with low price.

Battery Management System	2	\$3.5	They provide a safety measure to power subsystem with a low price.
Nickel Strip	1m	\$0.5	They are essential for producing battery pack and they have low price.
Cables, connectors & bolts	1	\$5	These are the required items to assemble the components.
Filament	600g	\$5	It is used to print 3D printable components which enabled us to design custom components.
Ball bearing caster wheel	1	\$4	Since the actuators are placed in rear, caster wheel prevented the tube to touch the ground and did not affect the rotation of the vehicle with little cost.
Polypropylene Sheet	1	\$3	It is used to create the boundaries of the test area with low cost.
Aruco Marker	9	\$1	They are used for localization and they are printed to paper by low cost.
Spray Paint	1	\$4	It is used to paint the polypropylene sheet to increase the detectability of the boundaries.
Magnet	16	\$2	They provide a low cost solution to attach the Aruco markers to the area boundaries.
Camera Stick	1	\$4	It is used to attach the camera at a certain height to the vehicle with low cost.
Total	-	\$293.92	-

Table 17: Total Cost

Expenditure	Price
Cost of Final Product	\$293.92
Delivery Cost	\$9.62
Transportation	\$5
Stationery	\$6.22
Solder Wire	\$12.3
Backup Batteries & Connectors	\$38.06
Total	365.12

Power Management

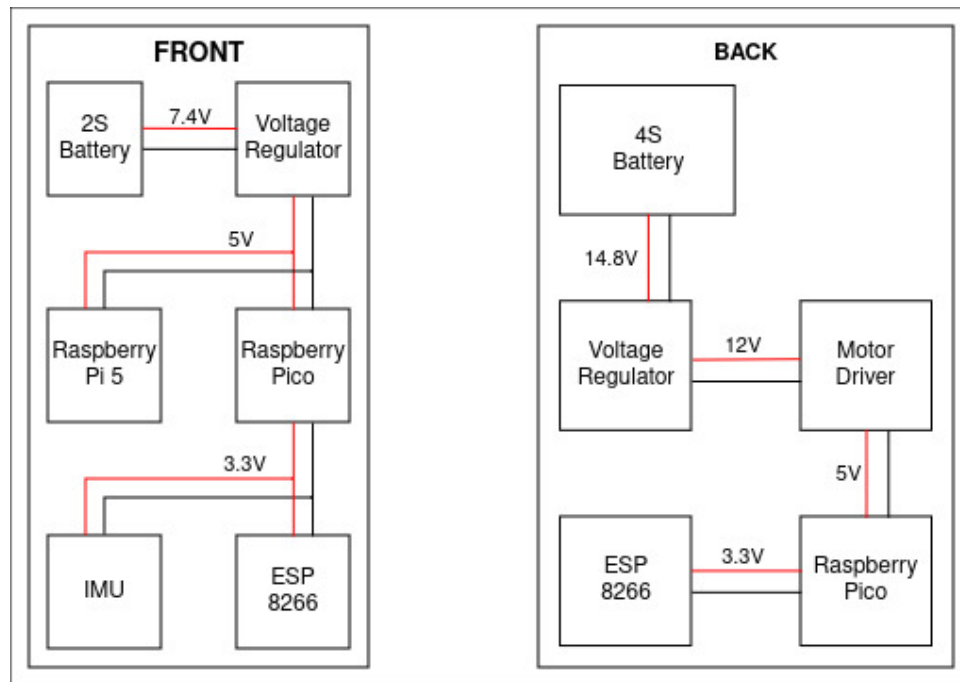


Figure 34: Power distribution diagram

The power distribution diagram of the vehicle is shown in **Figure 34**. Our motors have 800 mA stall current, so the two motors can draw a maximum of 1.6 A which means the maximum total power they can draw is $2 * 0.8A * 12V = 19.2W$. In our experiments, we saw that with our load and speed, the motors draw around 300 mA current each. Therefore, the motors together draw around $2 * 0.3A * 12V = 7.2W$ in average.

The Esp8266 draws around 50 mA current, so it consumes $3.3V * 50mA = 165mW$.

The Raspberry Pi Pico draws around 65 mA current, which means $5V * 65mA = 325mW$.

Therefore, the rear part of the vehicle consumes around 7.7 W which is supplied from the 2S lithium-ion battery.

In the front side, Raspberry Pi Camera could not be measured directly since it has a special cable, so it and Raspberry Pi 5 are connected and power measurements are made together. They draw around 1.3 A current, so they consume $5V * 1.3A = 6.5W$.

The Esp8266 draws around 50 mA current, so it consumes $3.3V * 50mA = 165mW$.

The Raspberry Pi Pico draws around 50 mA current, which means $5V * 50mA = 250mW$.

The IMU draws around 2.5 mA, so it consumes around $3.3V * 2.5mA = 8.25mW$.

Therefore, the front part of the vehicle consumes around 7 W which is supplied from 4S lithium-ion battery pack.

Schedule

This section details the comprehensive timeline of completed tasks for the Long Thin Hauler project, emphasizing the chronological progression and systematic integration of each component. This meticulous planning and execution ensure each subsystem and the overall project achieve optimal functionality and performance.

November 2023

- **Wireless Communication Subsystem: Literature Research** (November 4 - December 4): We embarked on an in-depth investigation into existing wireless communication technologies. This phase involved analyzing peer-reviewed articles, technical reports, and existing frameworks to understand the state-of-the-art methods and innovations. This extensive research laid the foundation for developing a robust, efficient, and secure wireless communication subsystem tailored to our project's unique requirements.
- **Localization Subsystem: Literature Research** (November 4 - November 20): Our team conducted thorough research to explore advanced algorithms and methodologies for localization, with a particular emphasis on visual positioning systems. This investigative period was focused on understanding the intricacies of visual localization techniques, including the utilization of ArUco markers. These markers are used for their robustness in various lighting conditions and their ability to provide accurate positional data. This stage of research was essential for us to determine the most suitable visual positioning system that aligns with the operational and environmental demands of our project, ensuring precise and reliable localization.

December 2023

- **Wireless Communication Subsystem: Transmission Tests and Working System** (Completed on December 18): Following our thorough literature review, we progressed to hands-on testing of the communication protocols. These tests validated the theoretical models, ensuring our wireless communication system could reliably transmit data across the required range without significant loss or interference.
- **Power Subsystem: Power Calculation** (November 20 - December 4): Detailed power calculations were performed to ascertain the exact power requirements of each component within the system. This analysis ensured that all subsystems could operate efficiently within the power constraints, providing a stable and continuous power supply.
- **Localization Subsystem: Algorithm Design and Localization Tests** (Completed on December 18): We designed and implemented localization algorithms, subsequently conducting rigorous tests to confirm their accuracy and reliability. These algorithms are crucial for the vehicle's ability to determine its position within the operational environment.
- **Power Subsystem: Battery and Actuator Selection** (December 10 - December 26): After finalizing power calculations, we selected the most suitable batteries and actuators. This selection process involved evaluating various options for their performance, longevity, and compatibility with our system requirements. We then acquired the chosen components to ensure timely integration into the project.

January 2024

- **Controller Design: Literature Research** (Completed by November 29): Preliminary research was undertaken to inform the design of our control system. This phase

involved studying existing controller architectures and techniques to derive insights that would guide our design process.

- **Hard Freeze: Finals Week** (January 5 - January 21): Due to academic commitments, project activities were significantly limited during this period. However, essential maintenance tasks and progress reviews were conducted to ensure project continuity.
- **Controller Design: Simulation Work and Simulink to Codebase** (January 21 - January 30): Post-academic hiatus, we resumed with the simulation of our controller design using Simulink. Successful simulations were then translated into actual code, ready for integration and testing on the physical system.
- **Mechanical Integration: Design Documentation** (January 21 - January 25): Detailed documentation of the mechanical design was prepared. This documentation included schematics, component specifications, and integration protocols, ensuring that all team members and stakeholders had a clear understanding of the mechanical aspects of the project.

February 2024

- **Controller Design: Testing on Vehicle and Working Controller** (Completed on February 3): The control system was rigorously tested on the actual vehicle. These tests confirmed that the controller could manage vehicle operations effectively, meeting all performance criteria.
- **Test Area Design: Area Border Holder Design** (February 3 - February 13): We designed and manufactured the border holder for the test area. This component is essential for physically defining the boundaries within which the vehicle operates during testing phases.
- **Power Subsystem: Battery Puncturing** (February 5 - February 12): Modifications to the battery setup were performed to enhance safety and performance. This involved puncturing the batteries to prevent hazardous failures and ensure they operated safely under various conditions.
- **Soft Freeze: Semester Break** (February 11 - February 20): Project activities were paused due to the academic calendar. This break allowed team members to recharge and prepare for the final stages of the project.
- **Mechanical Integration: Final Vehicle Integration** (Completed by February 10): All mechanical components were assembled into the vehicle. This integration ensured that the vehicle was fully functional and ready for subsequent subsystem tests and fine-tuning.

March 2024

- **Test Area Design: Parking Spot Design and Finalization** (February 24 - March 15): We completed the design and creation of the designated parking spot within the test area. This involved marking, measuring, and constructing the parking spot to ensure it met the specifications required for our testing protocols.
- **Sensor Subsystem: Kalman Filter Design** (Completed on March 29): Development of a Kalman filter was undertaken to enhance sensor accuracy. This filter processes noisy sensor data to produce more accurate and reliable readings, crucial for vehicle navigation and operation.

April 2024

- **Subsystem Integration: Integration of Finalized Subsystems and Fine Tuning** (March 10 - April 20): All subsystems were integrated into the vehicle, followed

by comprehensive fine-tuning. This phase ensured that each subsystem operated harmoniously within the complete system, optimizing overall performance.

- **Sensor Subsystem: Testing on the Vehicle** (February 23 - April 10): Extensive testing of the sensor subsystem was conducted on the vehicle to ensure its accuracy and reliability in real-world conditions. These tests verified that the sensors provided precise data necessary for navigation and operation.
- **Parking Spot Algorithm Design: Complete Design and Testing** (March 21 - April 10): We developed and thoroughly tested the algorithm for efficient vehicle parking. This algorithm enables the vehicle to autonomously identify and navigate to parking spots, a critical functionality for the project's success.

May 2024

- **Final Product and Documentation Towards Demo** (May 19 - May 26): The final phase involved compiling comprehensive documentation and preparing the project for the demonstration. This included final checks, polishing the presentation, and ensuring all aspects of the project were demo-ready.
- **Team BeyondTech Final Demo** (Scheduled for May 28): Our team is scheduled to present the final project demonstration, showcasing the operational excellence and innovative solutions developed throughout the project. This demo is a pivotal moment, highlighting the culmination of months of hard work and dedication.

June 2024

- **Project Fair** (Scheduled for June 13): The project will be presented at the Project Fair, providing an opportunity to share our achievements with a broader audience. This event will emphasize the technical prowess and innovative aspects of the Long Thin Hauler project, potentially attracting interest and feedback from industry experts and peers.

This methodical approach, along with the sequential and parallel execution of tasks, is visually represented in our Gantt chart, which is referenced as **Figure 35**. A more detailed version is available in the appendix under **Figure 36**. This chart encapsulates our commitment to a professionally managed project timeline, ensuring we meet our goals while addressing potential risks and coordinating team member responsibilities effectively.

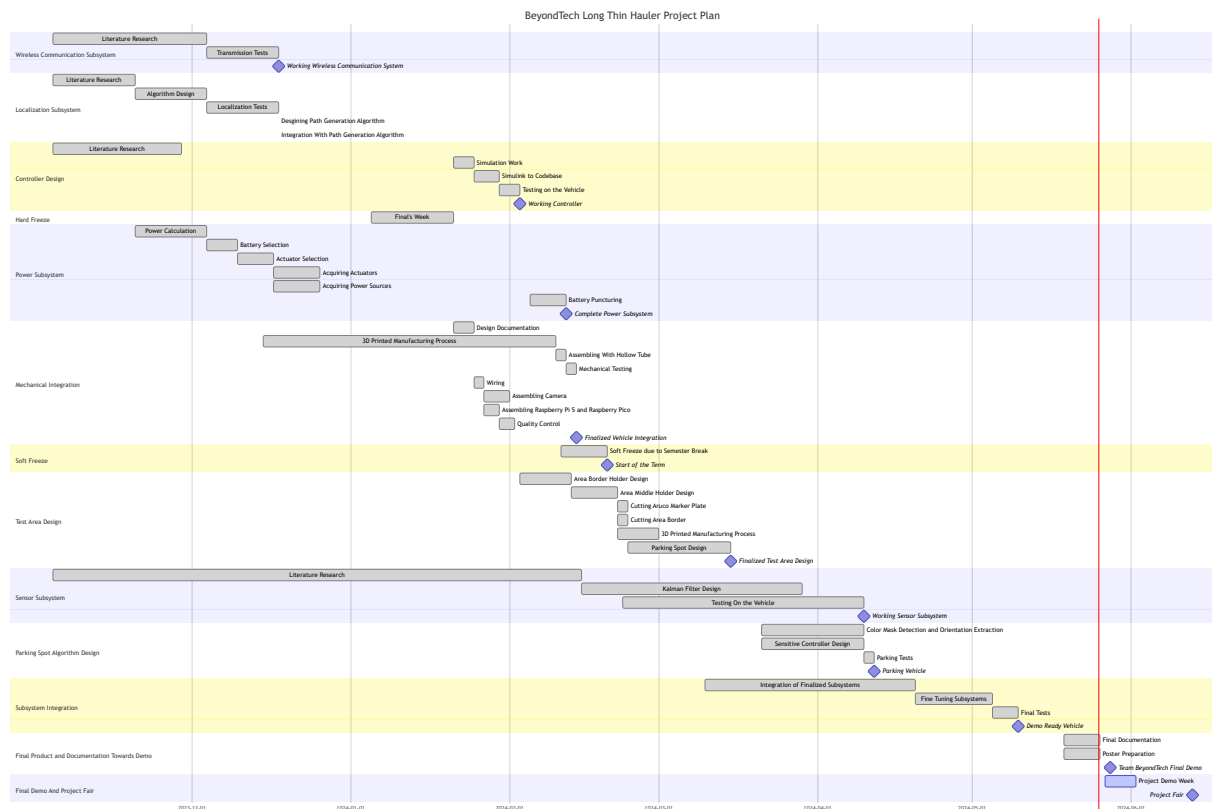


Figure 35: Gantt Chart for the duration of the project.

Conclusion

The "Long Thin Hauler" project stands as a testament to the innovative spirit and technical acumen of the BeyondTech team. Over the course of this extensive endeavor, the team has meticulously designed, developed, and tested an autonomous land vehicle that addresses contemporary challenges in urban parking solutions. This final document encapsulates the comprehensive journey from conceptualization to realization, highlighting the critical milestones, technical innovations, and collaborative efforts that have shaped this groundbreaking project.

The primary objective of the Long Thin Hauler project was to develop an autonomous vehicle capable of precise parking within a constrained urban environment. This objective necessitated a multifaceted approach, integrating advanced technologies in wireless communication, sensor fusion, and algorithmic control. The vehicle's design was driven by stringent requirements, including the necessity for forward-only motion, a compact chassis, and reliable wireless communication between subsystems separated by a substantial metallic tube. These design constraints posed significant engineering challenges, which the team addressed through innovative solutions and rigorous testing protocols.

The Long Thin Hauler's design is a marvel of modern engineering, combining multiple subsystems to achieve seamless operation. The vehicle's architecture comprises six fundamental subsystems: Power, Mechanics, Sensor, Visual Positioning, Controller, and Communication. Each subsystem was meticulously designed and integrated to ensure optimal performance. The mechanical subsystem supports the vehicle's structure and movement, while the power subsystem efficiently distributes energy to all components. The sensor subsystem integrates Inertial Measurement Unit (IMU) data with localization feedback, providing accurate path planning and navigation. The visual positioning subsystem, using a Raspberry Pi Camera and Aruco markers, ensures precise location tracking, critical for the vehicle's autonomous parking capabilities.

A significant technical achievement of the project is the robust communication system, which links the vehicle's front and rear sections using ESP8266 Wi-Fi modules through a metal hollow tube. This setup ensures reliable data transmission, vital for the synchronization of control commands and sensor feedback. The controller subsystem processes diverse inputs, translating them into actionable directives that guide the vehicle's movements. This subsystem's efficacy is evident in the vehicle's ability to perform precise parking maneuvers, even in constrained spaces.

The design phase involved overcoming several technical hurdles, including the integration of high-torque motors, compact chassis design, and reliable sensor placement. The team employed advanced 3D printing techniques to create a cost-effective and lightweight chassis, ensuring the vehicle's structural integrity and performance. The power subsystem, with its strategically placed batteries, ensures a stable energy supply, vital for the vehicle's continuous operation.

Extensive testing was a cornerstone of the project, ensuring that each subsystem functioned as intended under real-world conditions. The team conducted rigorous tests on the sensor subsystem, verifying its accuracy and reliability in various scenarios. The visual positioning system was optimized through dynamic testing, where the vehicle's camera tracked Aruco markers to determine precise location and orientation. The final mission tests demonstrated the vehicle's capability to autonomously park within a designated spot, consistently achieving the objective under diverse initial conditions.

The successful development of the Long Thin Hauler has significant implications for urban transportation and logistics. By automating the parking process, the vehicle reduces the time spent searching for parking spaces, thereby alleviating traffic congestion and lowering emissions from idling vehicles. This project not only enhances parking efficiency but

also contributes to environmental sustainability, aligning with broader goals of reducing urban carbon footprints.

Beyond the technical achievements, the Long Thin Hauler project offers substantial socio-economic benefits. It creates potential job opportunities in the tech and manufacturing sectors, driving innovation and economic growth. Furthermore, the project's success demonstrates the feasibility of deploying advanced autonomous systems in everyday applications, paving the way for future developments in smart transportation solutions.

In conclusion, the Long Thin Hauler project represents a significant leap forward in autonomous vehicle technology. The BeyondTech team's dedication, ingenuity, and collaborative efforts have culminated in a vehicle that not only meets but exceeds the initial project objectives. This project highlights the transformative potential of integrating advanced technologies in addressing contemporary urban challenges. As we look to the future, the insights and innovations from this project will undoubtedly inspire further advancements in autonomous systems, contributing to smarter, more efficient, and sustainable urban environments.

The comprehensive documentation, meticulous design, and successful implementation of the Long Thin Hauler underscore the team's commitment to excellence. This project stands as a beacon of what can be achieved through dedicated effort, innovative thinking, and collaborative teamwork. The Long Thin Hauler is not just a vehicle; it is a vision realized, a testament to the power of engineering to shape a better future.

Appendix 1: Gantt Chart

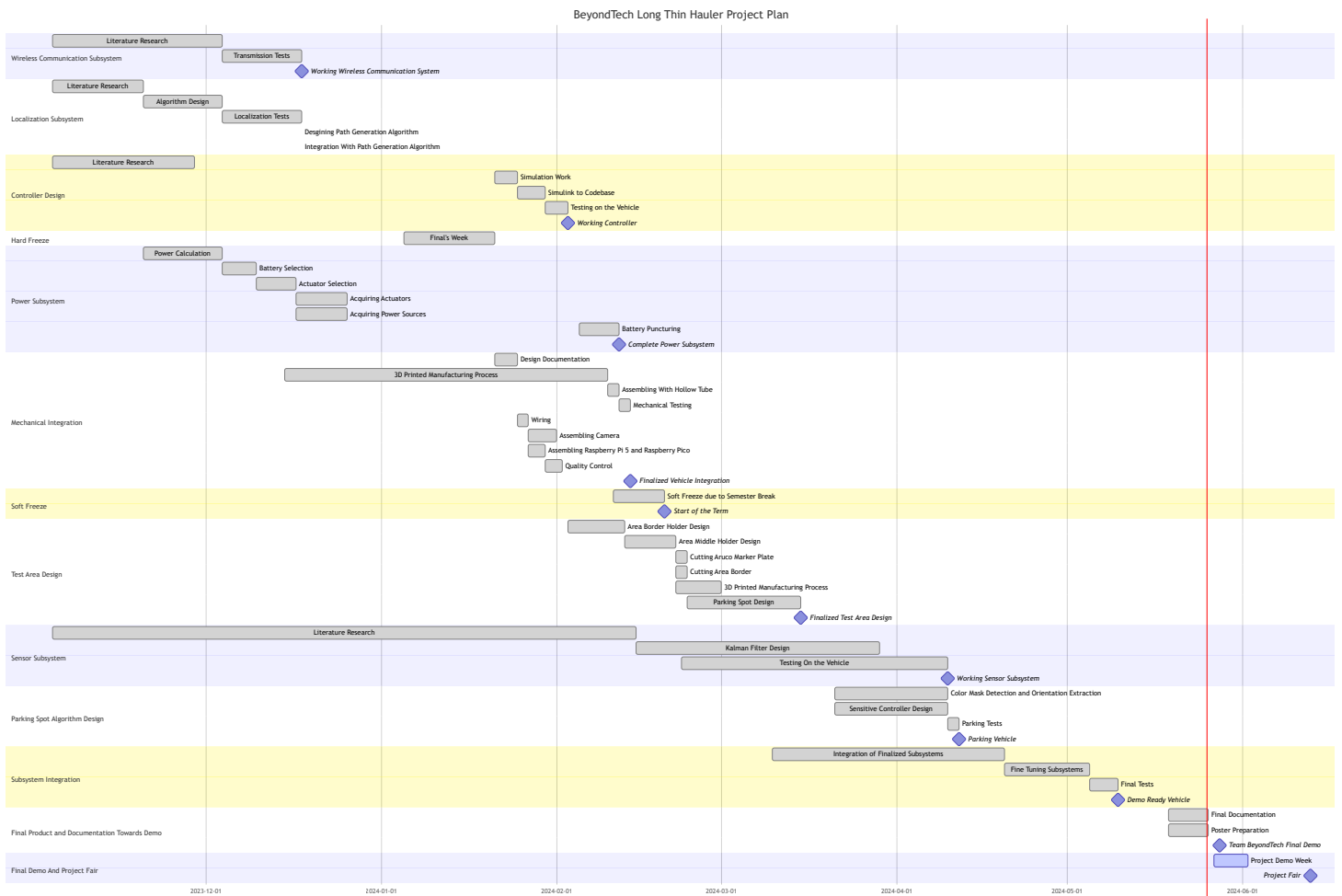


Figure 36: Gantt Chart for the duration of the project

Appendix 2: Test Document: Visual Positioning

Location	Electrical-Electronics Department Middle East Technical University
Date	December 20 2023
Time	15:00
Description	This test sheet explains the procedure for testing the localization.
Aim	The aim of this test is to verify the outputs of the localization
Expected Outcome	Limits of the localization system and accuracy of the measurements.
Participants	BeyondTech Team Members and Studio Coordinators

Test Devices and Tools

Required Test Equipments: Raspberry Pi 5, PiCam v3, CSI cable, 3D printed Camera Test Stand, Checkerboard, Laptop with required software installed, Ethernet cable, MATLAB Image Processing Toolbox, Aruca markers, Aruca marker stands, Gridlines on A3 paper, Ruler, Protractor, Chronometer, Tracker app, external video recording device (e.g. phone)

1. Tape measure:

The distances such as the position of marker relative to the camera stand are measured using tape measure. Also, some of the other measurement devices calibrated and verified using this device.

Ground truth/Calibration: The measuring tape is justified as a ground truth measurement tool due to its expected accuracy of less than one millimeter deviation over typical measuring lengths, aligning with our purposes in manual measurements. This level of accuracy is sufficient for most educational and practical applications.

2. Protractor:

The yaw angle of the camera stand is measured using a protractor. Also, the pitch angle control of the camera stand is validated using the protractor.

Ground truth/Calibration: The protractor is justified as a ground truth measurement tool for angular measurements due to its expected accuracy, typically within one degree. The measurement points for angles are sparse so the protractor is precise enough to perform this type of test procedure.

3. 3D printed Camera Test Stand:

The camera test stand is used to determine height and pitch angle of the camera and hold it in the given pose.

Ground truth/Calibration: The height of the camera test stand is adjusted to 10 cm according to the labels on the stand and then, it is checked by measuring the height with the measure tape. Similarly, the pitch angle of the camera test stand will be adjusted to minus 30 degrees using the labels and then, it will be verified with a protractor. Since the height nor pitch angle is directly measured and their affect on the measurement of the subsystem is indirect, the measure tapes accuracy and precision will be satisfactory.

4. A3 Paper with Gridlines:

This A3 paper is used to align the camera stand to predetermined yaw angles. Also, it is used to align the checkerboard with the origin of the camera.

Ground truth/Calibration:

The pattern on the A3 paper is verified using protractor and the measure tape.

5. Tracker App

Ground truth/Calibration:

For the calibration of tracker app, a ruler (or gridline) will be used. It will be placed into a place such that it will be visible for the video recording device. After the video is recorded, a calibration stick will be added on the ruler (or gridline) to complete the calibration process.

Test Environment

The testing environment consists of camera test stand and gridlines printed on A3 papers. Camera test stand and gridlines reside on a flat surface. The camera test stand is capable of positioning the camera at different heights (Z) and pitch angles(θ) in **Figure 37**.

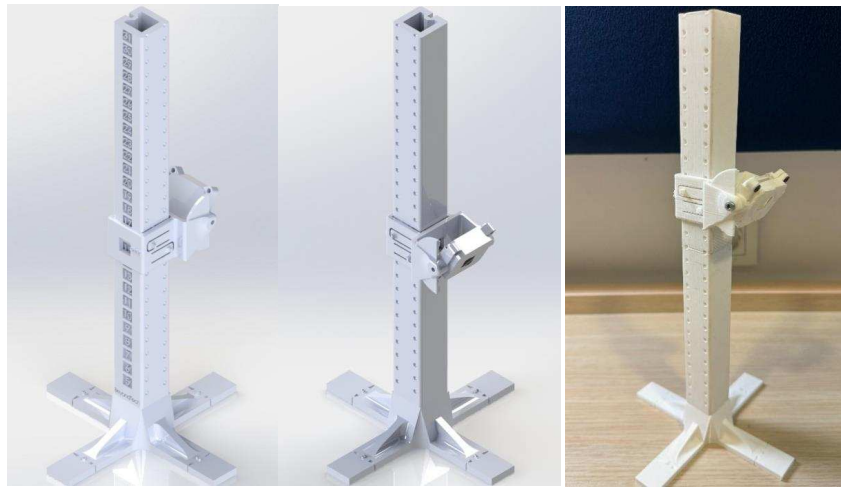


Figure 37: Camera test stand (CAD model and 3D printed model)

Thanks to the compliant design of the test stand with notch and void couples, height can be set to integer values from 5 to 31 centimeters and the pitch angle of the camera can be set to set to -90, - 70, -50, -30, -10, 10, 30, 50, 70, 90. Apart from discrete values, height and angle can also be adjusted using screws as in **Figure 38** to lock the camera at a desired position on the camera test stand.

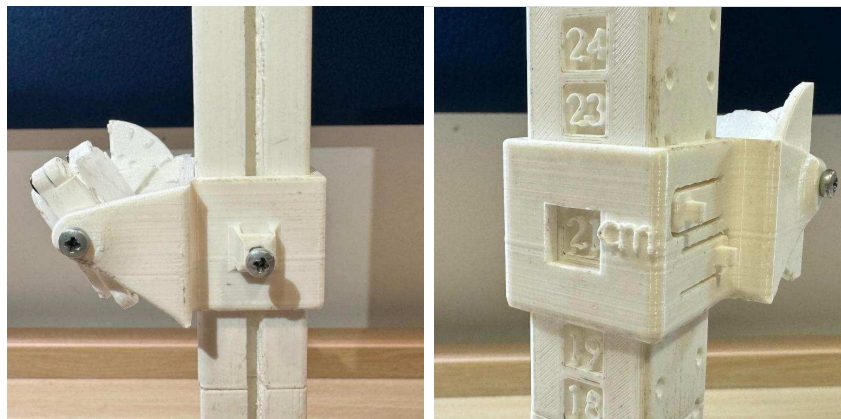


Figure 38: Adjustable angle and height

Gridlines allow us to position Aruco marker stand at different locations so that experiments would be done at different possible configurations **Figure 39**. Only a small portion of the area is marked using the designed pattern. The rest of the area is marked using colored tape.

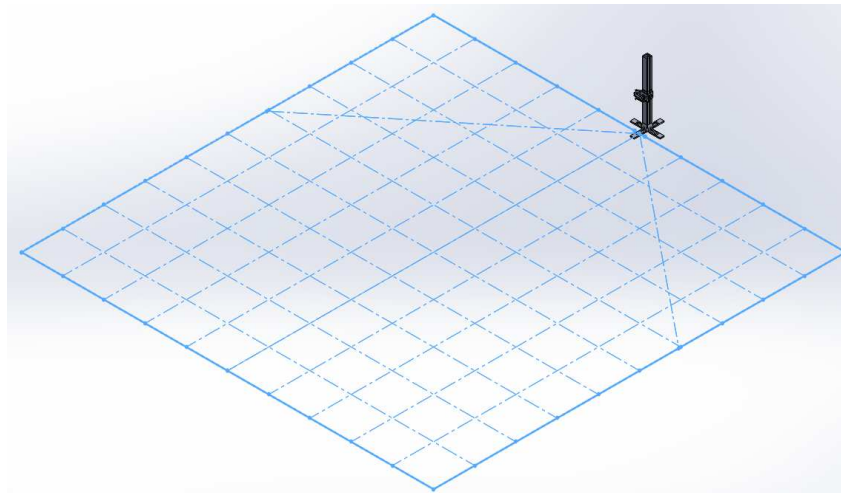


Figure 39: Gridlines for positioning aruco marker stand

The camera calibrator application is preferred to perform intrinsics and distortion calibration (**Figure 40**). It accepts the captured images of the known size checkerboard.

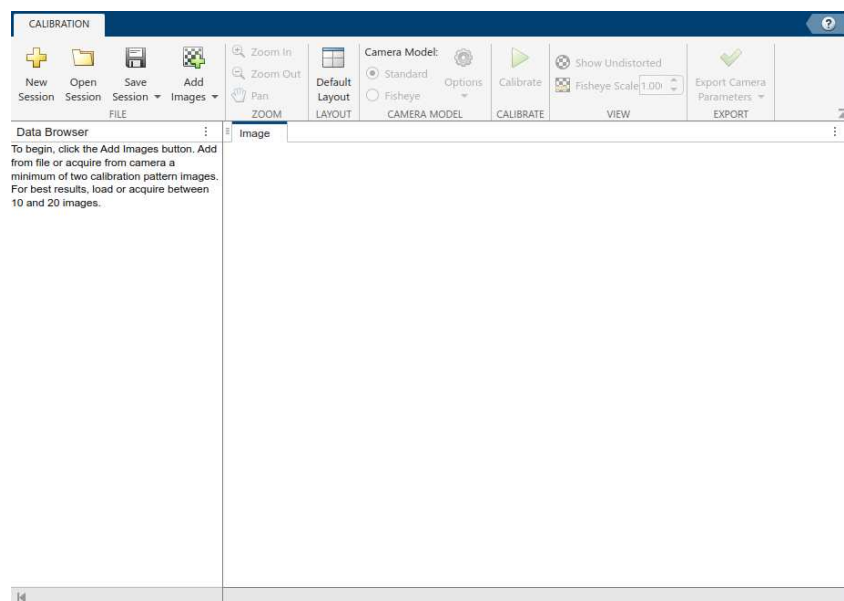


Figure 40: MATLAB Camera Calibration Toolbox

Through the test procedure, Raspberry Pi 5 single-board computer is used for processing of the images (**Figure 41**). It can be thought of as a general-purpose computing device with low power requirements. It also communicates with the camera module to capture images from the environment.

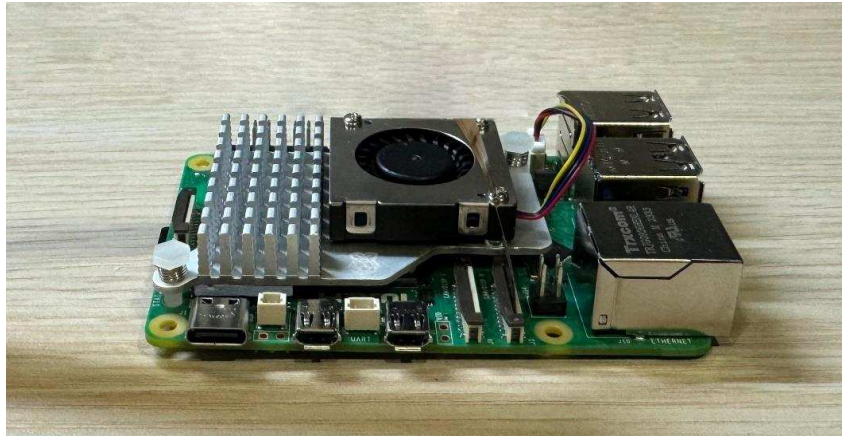


Figure 41: Raspberry Pi 5

The Raspberry Pi Camera Module V3 is used as the camera module **Figure 42**. It can be connected to Raspberry Pi 5 using the CSI connector. Different than general purpose USB cameras, it is tailored to used with Raspberry Pi boards. It allows setting many parameters such as autofocus mode, exposures time, and scaling crops.

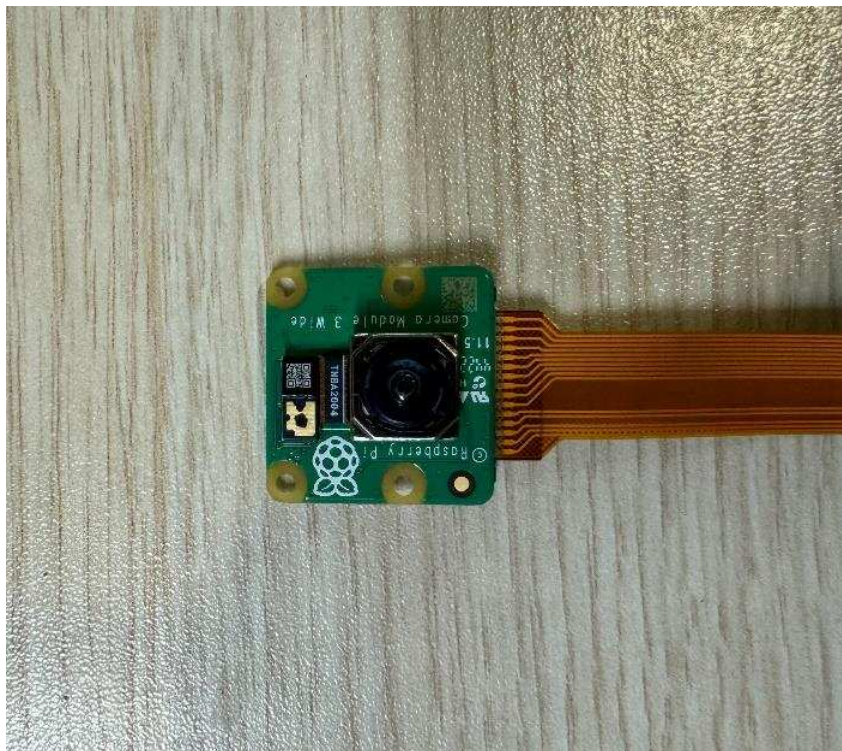


Figure 42: Raspberry camera module V3 wide

The Aruco marker is used as a visual marker throughout the test **Figure 43**. It is a 2D barcode that can be detected by the camera. It is used to determine the position and orientation of the camera relative to the marker.

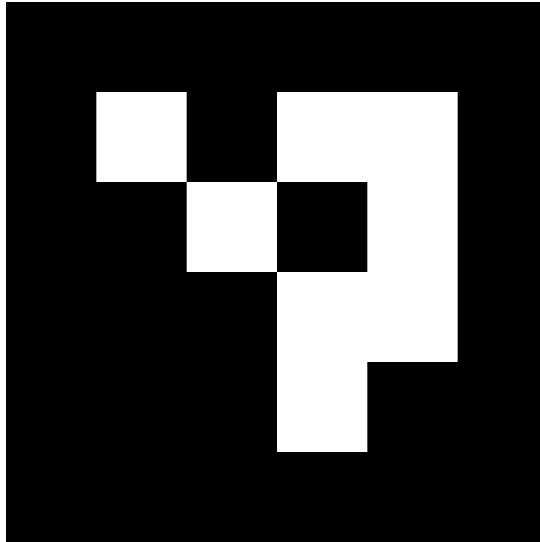


Figure 43: The Aruco marker is used as visual marker throughout the test

Test Parameters

With changing distance between the marker and the camera stand, the location of the camera relative to the marker also changes. Similarly, the yaw angle of the camera stand is also measured by the system. On the other hand, the height and the pitch angle of the camera affects the measurement results indirectly. For some combinations of them, the perspective distortion will induce errors in the estimated location and heading. They also change the area visible by the camera.

Table 18: The control parameters

Parameter	Range	Step Size	Number of Measurements
Vertical distance of the aruco marker (X)	[+0.2m +2m]	20cm	10
Horizontal distance of the aruco marker(Y)	[-2m +2m]	40cm	11
Height of the camera(Z)	[+15 +25cm]	10cm	2
Yaw angle of the camera(ψ)	[0° 45°]	15°	4
Pitch angle of the camera(θ)	[-30° -15°]	15°	2

Table 19: The control parameters

Parameter	Range	Step Size	Number of Measurement
Vertical distance of the aruco marker (X)	[+0.2m +2m]	20cm	10
Horizontal distance of the aruco marker(Y)	[-2m +2m]	40cm	11
Height of the camera(Z)	[+15 +25cm]	10cm	2
Yaw angle of the camera(ψ)	[0° 45°]	15°	4
Pitch angle of the camera(θ)	[-30° -15°]	15°	2

Test Procedure

1. If it is the first time Raspberry Pi 5 is used, please follow the instructions in the following link to flash the image to the SD card:
<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>
2. Connect the Raspberry Pi 5 to Raspberry Pi Camera Module V3 by the ribbon cable.
3. Power up the Raspberry Pi 5 using the 27W USB-C Power Adapter and wait for the system to boot up.
4. Use another computer in the same network to SSH into the Raspberry Pi 5. Use the following command;


```
ssh -X [HOSTNAME]@[IP_ADDRESS]
```

5. Check whether the camera is working. Run the following command to start the camera preview:

6. After verifying the camera works correctly, press Ctrl+C to stop the preview.

7. Now, the camera calibration will be performed. Run the provided Python script to take the calibration images. This script will take pictures every time you press the Enter key. The images will be saved in the current directory as calibrationimage.png. The number will be increased by one every time you press the Enter key. The script will stop when you press the Ctrl+C key combination.

```
python3 camera_calibration.py --capture
```

8. After taking the calibration images, you can run the following command to calibrate the camera. After the calibration is done the calibration parameters will be saved in the "calibration.txt" file.

```
python3 camera_calibration.py --calibrate
```

9. Alternatively, you can use the MATLAB to calibrate the camera. First, you need to copy the calibration images to your computer. You can use the following command to copy the images to your computer. Replace the [HOSTNAME] and [IPADDRESS] with the hostname and IP address of your Raspberry Pi 5. Then, run the "cameraCalibrator" app in MATLAB and follow the instructions in the app to calibrate the camera. After the calibration, open the calibration.txt file and write the calibration parameters to the file. The first line contains f_x , f_y , c_x , and c_y . The second line contains k_1 , k_2 , p_1 , p_2 , and k_3 . This is the same format as the calibration.txt file generated by the Python script. Refer to <https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html> for more information about the cameraCalibrator app.

```
scp [HOSTNAME]@[IP_ADDRESS]:~/folder ./
```

10. Power off the Raspberry Pi 5, mount the camera, and adjust the height and angle of the camera using the camera stand. The height can be adjusted by loosening the screw on the side of the camera stand and sliding the camera stand up and down. The angle can be adjusted by loosening the screw on the top of the camera stand and rotating the camera stand. The camera stand can be rotated vertically, changing its pitch angle.
11. Now, you will calibrate the homography matrix. After changing the height and angle of the camera using the camera stand, the homography matrix must be recalculated. Align the checkerboard with the camera. This alignment is critical since it will determine the world coordinate system's origin and orientation.
12. Run the following command to calculate the homography matrix. The homography matrix will be saved in the "homography.txt" file.

```
python3 camera_calibration.py --homography
```

13. Now, you can start estimating the camera's position and orientation. The center of the bottom edge of the checkerboard is the origin of the world coordinate system. Run the following command to start the camera pose estimation. It will output the camera's position and orientation in the world coordinate system. The position is in meters, and the orientation is in degrees.

```
python3 camera_calibration.py --estimate
```

14. For each static measurement case, place the marker in the desired location and take note of the position and the camera's orientation. Do not move the camera or the marker during the measurement. Essentially, you are changing the origin of the world coordinate system. Do not forget to calibrate the homography matrix after adjusting the height and angle of the camera using the camera stand.
15. Repeat the step 14 for each static measurement case.

16. After the static measurements are performed, you can start the dynamic measurement. This step is more elaborate than the static measurement. Prepare the external camera to record and make sure that it is recording.
17. While both estimation and recording are running, capture both the computer screen and the marker's position. Do not forget to start chronometer.
18. Slowly move the marker forward to the camera. Make sure that your hand is not in front of the marker. The marker must be always visible to the camera.
19. Stop the estimation and recording after the marker is moved to the desired location.
20. Use the Tracker app to track the marker in the video. The Tracker app can be downloaded from the following link:
<https://physlets.org/tracker/>
21. Prepare the measurement for the axis of movement. For each frame, the marker is visible in the video; record the position of the marker in the video and the position of the marker in the real world.
22. In this part of the experiment, use a video capturing device to estimate the position of the marker at each frame. Put your camera devices to a fixed position at which the test scene is visible while performing the experiment. Set the pitch angle of the camera to 30 degrees, height of the camera to 25 centimeters. Then slide a marker along the given x and y directions in the measurement table. Then, plot your position- time graph using Tracker software, then compare with your measurements with camera.

Test Data

Perform measurements in static test grouped by yaw angles. Start testing with yaw angle of zeros for the camera test stand as in **Figure 39**. And continue with yaw angle of 15 and 30 degrees as in the figures below (**Figure 44** and **Figure 45**).

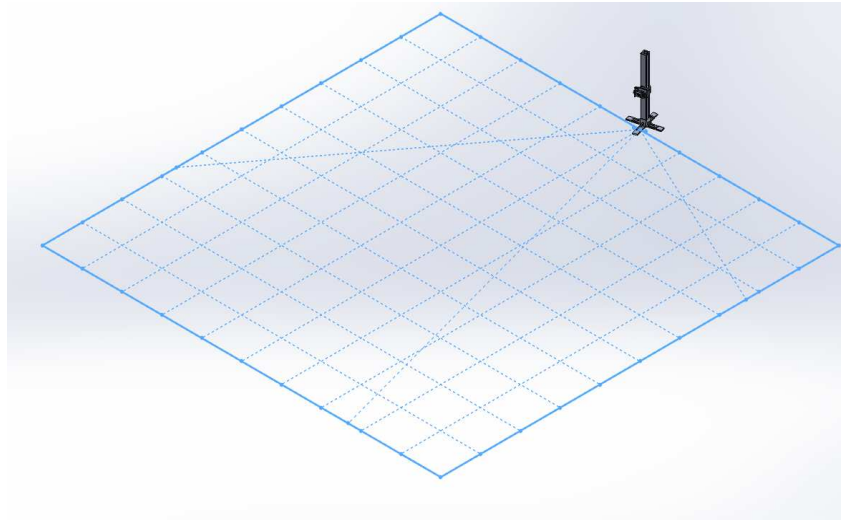


Figure 44: Figure 9: 15 Degrees of Yaw Angle

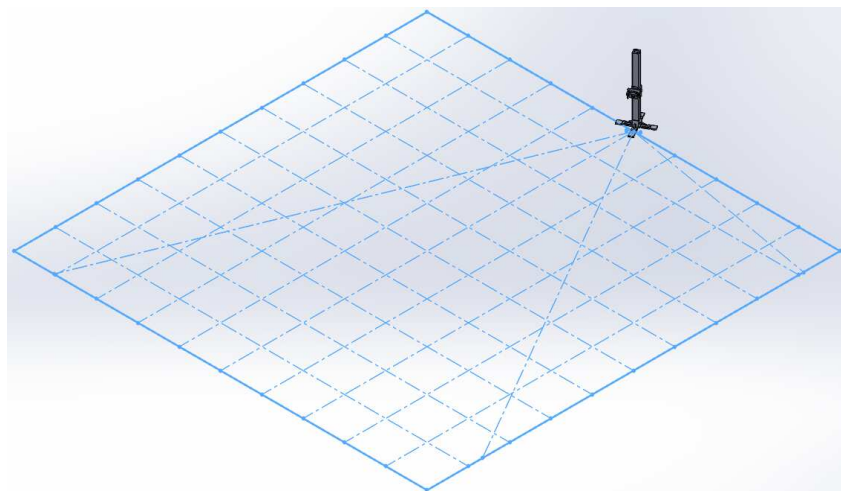


Figure 45: Figure 10: 30 Degrees of Yaw Angle

Table 20: Vertical positioning test with 10 degree pitch with no yaw

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
35.0	0.0	0.0	-34.953	0.497	-0.98
40.0	0.0	0.0	-40.24	0.746	-0.994
45.0	0.0	0.0	-45.236	0.357	-0.489
50.0	0.0	0.0	-50.147	0.157	-0.277
55.0	0.0	0.0	-55.171	0.3	-0.443
60.0	0.0	0.0	-60.199	-0.32	0.163
65.0	0.0	0.0	-65.034	-0.579	0.398
70.0	0.0	0.0	-70.031	-0.794	0.583
75.0	0.0	0.0	-74.813	-1.237	0.852
80.0	0.0	0.0	-79.846	-1.865	1.204
85.0	0.0	0.0	-84.678	-1.506	1.023
90.0	0.0	0.0	-89.598	-1.105	0.666
95.0	0.0	0.0	-94.381	-2.076	1.264
100.0	0.0	0.0	-99.294	-3.042	1.757
105.0	0.0	0.0	-104.167	-5.652	3.079
110.0	0.0	0.0	-108.798	-11.902	6.18
115.0	0.0	0.0	-113.589	-12.66	6.306
120.0	0.0	0.0	-118.367	-12.732	6.107
125.0	0.0	0.0	-122.923	-13.385	6.204
130.0	0.0	0.0	-127.52	-15.192	6.76
135.0	0.0	0.0	-131.933	-18.106	7.785
140.0	0.0	0.0	-136.782	-16.865	7.0
145.0	0.0	0.0	-141.338	-18.543	7.446
150.0	0.0	0.0	-145.747	-19.144	7.435
155.0	0.0	0.0	-149.949	-22.996	8.658

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
160.0	0.0	0.0	-154.176	-24.535	8.988
165.0	0.0	0.0	-157.987	-27.434	9.792
170.0	0.0	0.0	-161.663	-34.227	11.869
175.0	0.0	0.0	-166.33	-33.782	11.406
180.0	0.0	0.0	-170.369	-34.872	11.489
185.0	0.0	0.0	-174.034	-37.473	12.086
190.0	0.0	0.0	-179.196	-34.909	10.951
195.0	0.0	0.0	-182.026	-42.313	13.022
200.0	0.0	0.0	-185.572	-45.195	13.621

Table 21: Vertical positioning test with 10 degree pitch with 15 degree yaw

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
40.0	0.0	15.0	-39.849	0.703	-15.716
50.0	0.0	15.0	-50.224	-0.183	-14.361
60.0	0.0	15.0	-60.182	-0.964	-13.619
70.0	0.0	15.0	-70.365	-1.677	-13.168
80.0	0.0	15.0	-80.566	-2.62	-12.605
90.0	0.0	15.0	-90.62	-3.808	-12.023
100.0	0.0	15.0	-100.649	-4.737	-11.714
110.0	0.0	15.0	-110.783	-11.789	-8.335
120.0	0.0	15.0	-120.518	-14.174	-7.674
130.0	0.0	15.0	-130.143	-16.579	-7.112
140.0	0.0	15.0	-139.69	-19.062	-6.569
150.0	0.0	15.0	-148.944	-23.132	-5.546
160.0	0.0	15.0	-158.359	-25.963	-5.033
170.0	0.0	15.0	-166.897	-31.036	-3.788

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
180.0	0.0	15.0	-175.19	-37.607	-2.224
190.0	0.0	15.0	-184.208	-40.46	-1.921
200.0	0.0	15.0	-191.217	-52.535	1.024

Table 22: Vertical positioning test with 10 degree pitch with 30 degree yaw

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
45.0	0.0	30.0	-45.115	0.13	-30.069
50.0	0.0	30.0	-50.276	0.107	-29.931
60.0	0.0	30.0	-60.706	0.142	-29.797
70.0	0.0	30.0	-70.861	-0.878	-28.822
80.0	0.0	30.0	-81.408	-1.832	-28.058
90.0	0.0	30.0	-91.8	-1.794	-28.149
100.0	0.0	30.0	-102.029	-3.708	-27.11
110.0	0.0	30.0	-112.532	-11.185	-23.487
120.0	0.0	30.0	-122.703	-13.415	-22.903
130.0	0.0	30.0	-132.569	-16.566	-21.918
140.0	0.0	30.0	-142.682	-18.67	-21.607
150.0	0.0	30.0	-152.656	-20.714	-21.298
160.0	0.0	30.0	-161.855	-27.657	-19.336
170.0	0.0	30.0	-171.422	-31.082	-18.733
180.0	0.0	30.0	-180.033	-14.097	-25.177
190.0	0.0	30.0	-189.485	-42.442	-16.353
200.0	0.0	30.0	-198.527	-49.316	-15.048

Table 23: Horizontal positioning test with 10 degree pitch at 80 cm distance

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
80.0	0.0	0.0	-79.932	-2.065	1.393
80.0	20.0	0.0	-80.694	-26.532	4.12
80.0	40.0	0.0	-80.911	-47.911	3.916
80.0	60.0	0.0	-80.922	-47.891	3.899
80.0	80.0	0.0	-81.683	-69.519	3.01

Table 24: Horizontal positioning test with 10 degree pitch at 100 cm distance

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
100.0	0.0	0.0	-99.397	-2.689	1.516
100.0	20.0	0.0	-99.565	-31.313	6.163
100.0	40.0	0.0	-99.821	-52.145	5.783
100.0	60.0	0.0	-100.061	-74.111	5.45
100.0	80.0	0.0	-100.873	-97.12	4.838
100.0	100.0	0.0	-97.993	-125.087	6.275

Table 25: Horizontal positioning test with 10 degree pitch at 180 cm distance

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
180.0	0.0	0.0	-170.283	-35.13	11.55
180.0	20.0	0.0	-164.969	-65.48	15.283
180.0	40.0	0.0	-167.469	-78.553	12.637
180.0	60.0	0.0	-165.409	-101.258	13.099
180.0	80.0	0.0	-169.312	-116.925	10.788
180.0	100.0	0.0	-171.91	-136.707	9.508
180.0	120.0	0.0	-171.318	-162.281	9.732

Table 26: Horizontal positioning test with 10 degree pitch at 200 cm distance

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
200.0	20.0	0.0	-185.89	-43.388	13.053
200.0	40.0	0.0	-184.163	-65.075	13.733
200.0	60.0	0.0	-187.478	-76.136	10.837
200.0	80.0	0.0	-189.987	-92.656	9.398
200.0	100.0	0.0	-190.499	-115.608	9.563
200.0	120.0	0.0	-192.317	-138.384	9.289
200.0	140.0	0.0	-190.75	-140.895	10.002
200.0	160.0	0.0	-193.096	-163.115	9.255

Table 27: Vertical positioning test with 30 degree pitch with no yaw

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
20.0	0.0	0.0	-21.772	0.412	-1.006
25.0	0.0	0.0	-25.095	0.347	-0.673
30.0	0.0	0.0	-29.217	-2.332	4.76
35.0	0.0	0.0	-35.119	0.322	-0.359
40.0	0.0	0.0	-40.223	0.341	-0.335
45.0	0.0	0.0	-45.124	0.225	-0.081
50.0	0.0	0.0	-49.942	-0.375	0.531
55.0	0.0	0.0	-55.796	-1.338	1.401
60.0	0.0	0.0	-60.167	-0.246	0.387
65.0	0.0	0.0	-65.201	-1.249	1.28
70.0	0.0	0.0	-70.289	-0.909	0.904
75.0	0.0	0.0	-75.361	-1.322	1.17
80.0	0.0	0.0	-80.215	-1.09	0.974

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
85.0	0.0	0.0	-85.151	-1.519	1.264
90.0	0.0	0.0	-90.117	-1.559	1.151
95.0	0.0	0.0	-94.941	-1.943	1.38
100.0	0.0	0.0	-99.663	-2.464	1.621
105.0	0.0	0.0	-104.611	-7.938	3.432
110.0	0.0	0.0	-109.242	-8.956	4.339
115.0	0.0	0.0	-113.896	-9.298	4.826
120.0	0.0	0.0	-118.555	-9.992	4.975
125.0	0.0	0.0	-123.207	-11.433	5.45
130.0	0.0	0.0	-128.053	-13.195	6.048
135.0	0.0	0.0	-132.819	-13.245	5.853
140.0	0.0	0.0	-137.478	-14.219	6.092
145.0	0.0	0.0	-141.756	-17.448	7.173
150.0	0.0	0.0	-145.577	-20.239	8.064
155.0	0.0	0.0	-150.162	-19.125	7.401
160.0	0.0	0.0	-154.595	-21.521	8.07
165.0	0.0	0.0	-160.107	-21.306	7.713
170.0	0.0	0.0	-166.448	-16.553	5.832
175.0	0.0	0.0	-169.225	-21.969	7.542
180.0	0.0	0.0	-173.657	-22.105	7.405
185.0	0.0	0.0	-176.792	-26.167	8.573
190.0	0.0	0.0	-181.574	-23.539	7.542
195.0	0.0	0.0	-185.059	-31.542	9.809
200.0	0.0	0.0	-188.109	-31.999	9.797

Table 28: Vertical positioning test with 30 degree pitch with 15 degree yaw

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
25.0	0.0	15.0	-25.171	0.092	-15.553
30.0	0.0	15.0	-30.502	0.649	-16.02
40.0	0.0	15.0	-40.034	0.115	-14.887
50.0	0.0	15.0	-50.244	-0.394	-14.264
60.0	0.0	15.0	-60.317	-0.117	-14.575
70.0	0.0	15.0	-70.545	-0.339	-14.37
80.0	0.0	15.0	-80.856	-0.366	-14.29
90.0	0.0	15.0	-91.016	-0.162	-14.45
100.0	0.0	15.0	-101.124	-0.385	-14.306
110.0	0.0	15.0	-111.218	-5.228	-11.833
120.0	0.0	15.0	-121.276	-6.269	-11.541
130.0	0.0	15.0	-130.951	-8.559	-10.717
140.0	0.0	15.0	-140.553	-12.135	-9.552
150.0	0.0	15.0	-150.376	-11.108	-10.262
160.0	0.0	15.0	-159.397	-13.716	-9.52
170.0	0.0	15.0	-169.578	-11.251	-10.684
180.0	0.0	15.0	-178.944	-6.862	-12.237
190.0	0.0	15.0	-186.823	-15.241	-9.797
200.0	0.0	15.0	-195.413	-24.59	-7.266

Table 29: Vertical positioning test with 30 degree pitch with 30 degree yaw

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
35.0	0.0	30.0	-34.895	-0.007	-29.683
40.0	0.0	30.0	-40.046	-0.076	-29.444

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
50.0	0.0	30.0	-50.384	0.137	-29.504
60.0	0.0	30.0	-60.665	0.384	-29.57
70.0	0.0	30.0	-70.427	-0.399	-28.863
80.0	0.0	30.0	-79.565	-0.01	-29.089
90.0	0.0	30.0	-90.576	-0.821	-28.499
100.0	0.0	30.0	-100.505	-0.892	-28.471
110.0	0.0	30.0	-110.412	-6.692	-25.477
120.0	0.0	30.0	-120.023	-7.475	-25.364
130.0	0.0	30.0	-129.642	-11.309	-23.912
140.0	0.0	30.0	-138.932	-12.147	-23.875
150.0	0.0	30.0	-148.328	-16.799	-22.459
160.0	0.0	30.0	-157.718	-19.679	-21.757
170.0	0.0	30.0	-166.422	-25.002	-20.337
180.0	0.0	30.0	-175.083	-29.522	-19.312
190.0	0.0	30.0	-183.253	-35.467	-17.919
200.0	0.0	30.0	-192.133	-37.786	-17.742

Table 30: Horizontal positioning test with 30 degree pitch at 80 cm distance

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
80.0	0.0	0.0	-80.367	-1.803	1.428
80.0	20.0	0.0	-80.852	-23.927	2.63
80.0	40.0	0.0	-80.854	-44.96	2.707
80.0	60.0	0.0	-80.334	-69.041	3.706

Table 31: Horizontal positioning test with 30 degree pitch at 100 cm distance

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
100.0	0.0	0.0	-100.204	-2.802	1.752
100.0	20.0	0.0	-100.404	-28.126	4.572
100.0	40.0	0.0	-100.539	-27.83	4.394
100.0	60.0	0.0	-101.394	-47.458	3.492
100.0	80.0	0.0	-100.718	-70.98	4.373
100.0	100.0	0.0	-98.278	-95.84	5.689

Table 32: Horizontal positioning test with 30 degree pitch at 180 cm distance

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
180.0	0.0	0.0	-171.305	-29.812	10.01
180.0	20.0	0.0	-171.777	-40.95	7.327
180.0	40.0	0.0	-170.413	-61.251	7.621
180.0	60.0	0.0	-168.995	-85.342	8.799
180.0	80.0	0.0	-170.464	-102.369	7.528
180.0	100.0	0.0	-171.303	-120.395	6.596
180.0	120.0	0.0	-160.168	-157.834	11.39

Table 33: Horizontal positioning test with 30 degree pitch at 200 cm distance

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
180.0	0.0	0.0	-171.305	-29.812	10.01
180.0	20.0	0.0	-171.777	-40.95	7.327
180.0	40.0	0.0	-170.413	-61.251	7.621
180.0	60.0	0.0	-168.995	-85.342	8.799
180.0	80.0	0.0	-170.464	-102.369	7.528

$X(\text{cm})$	$Y(\text{cm})$	$\phi(\text{cm})$	$X_{est}(\text{cm})$	$Y_{est}(\text{cm})$	$\phi_{est}(\text{cm})$
180.0	100.0	0.0	-171.303	-120.395	6.596
180.0	120.0	0.0	-160.168	-157.834	11.39

Data Analysis

Static positioning Tests

In the static test, we performed the tests only in the one half of the FOV, assuming the characteristic of the camera would be symmetric in both halves.

In the first test, we aimed to find the accuracy of the positioning output at multiple points in the field of view of the camera. The result of this test is as follows in **Figure 46**.

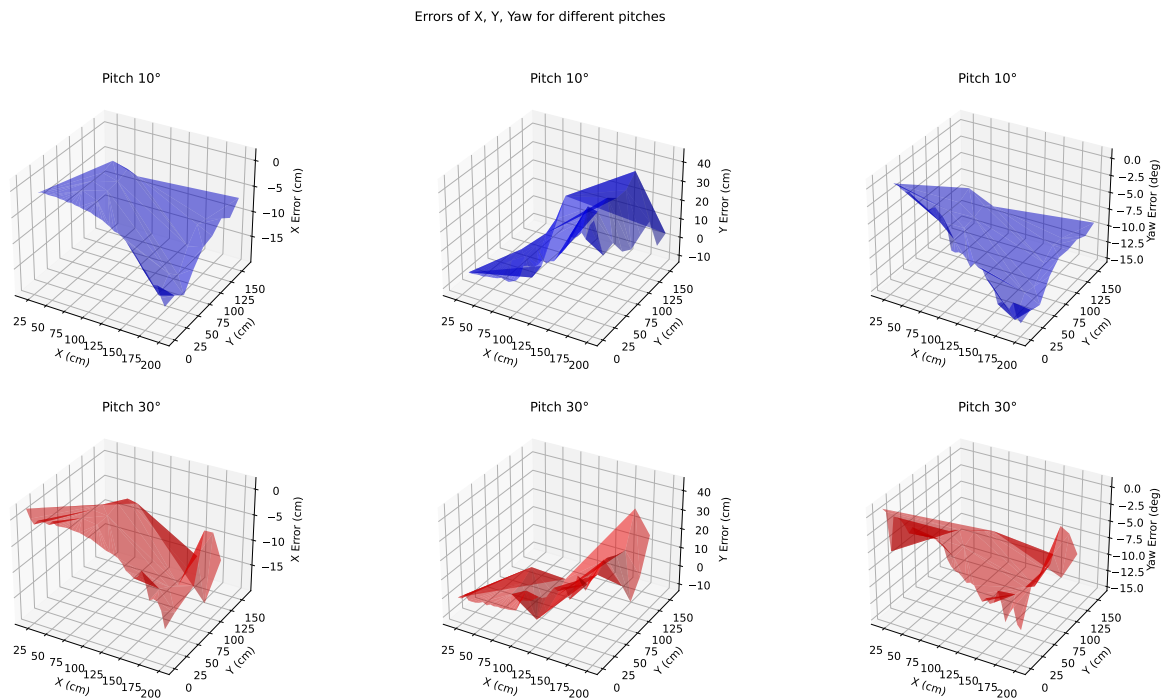


Figure 46: Error of the positioning output in FOV

Also, these results can be individually examined from the following plots (**Figure 47** and **Figure 48**).

Camera with 10° pitch angle

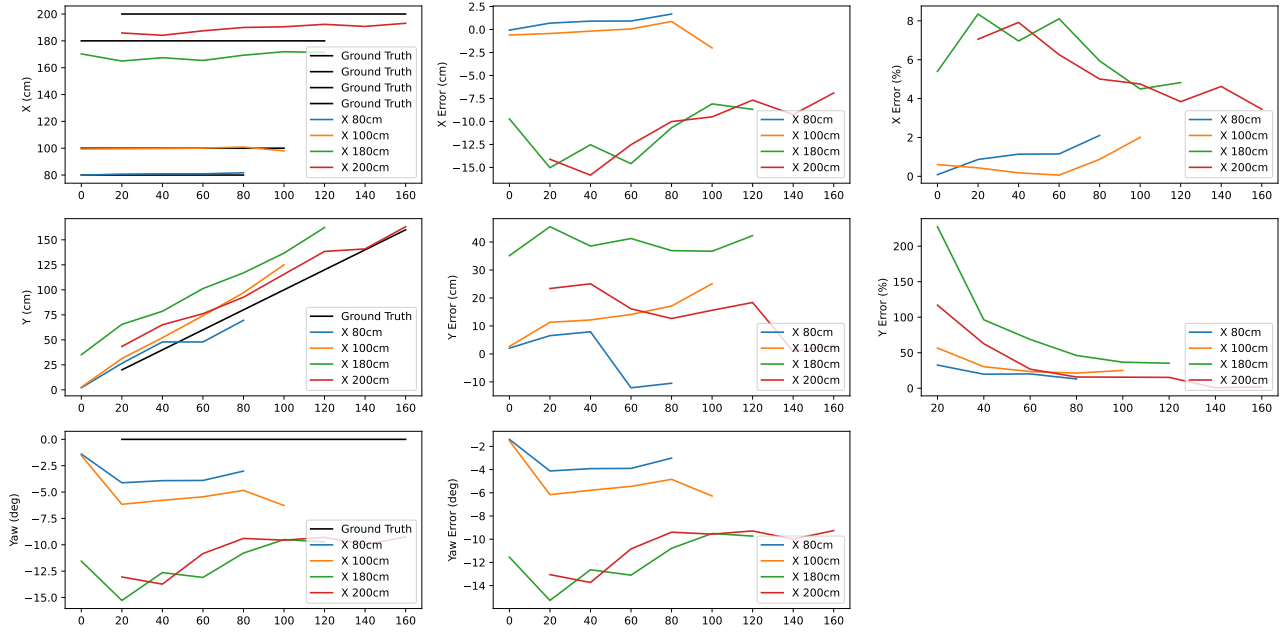


Figure 47: Error of the positioning output in FOV with pitch angle of 10 degrees

Camera with 30° pitch angle

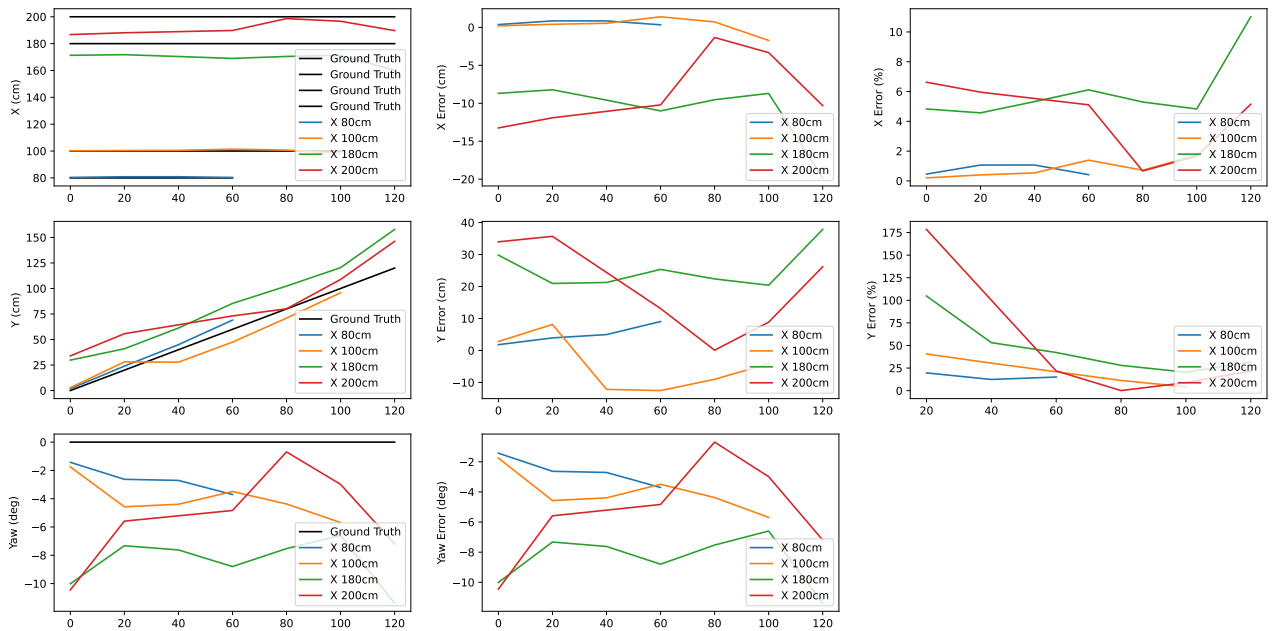


Figure 48: Error of the positioning output in FOV with pitch angle of 30 degree

Then, for 10 and 30 degrees of pitch angles, and 0, 15, and 30 degrees of yaw angles measurements are made the results are examined in two configurations:

When pitch angle is kept constant and the effect of yaw angles on positioning is examined, the following results are obtained in **Figure 49**, **Figure 50**, and **Figure 51**:

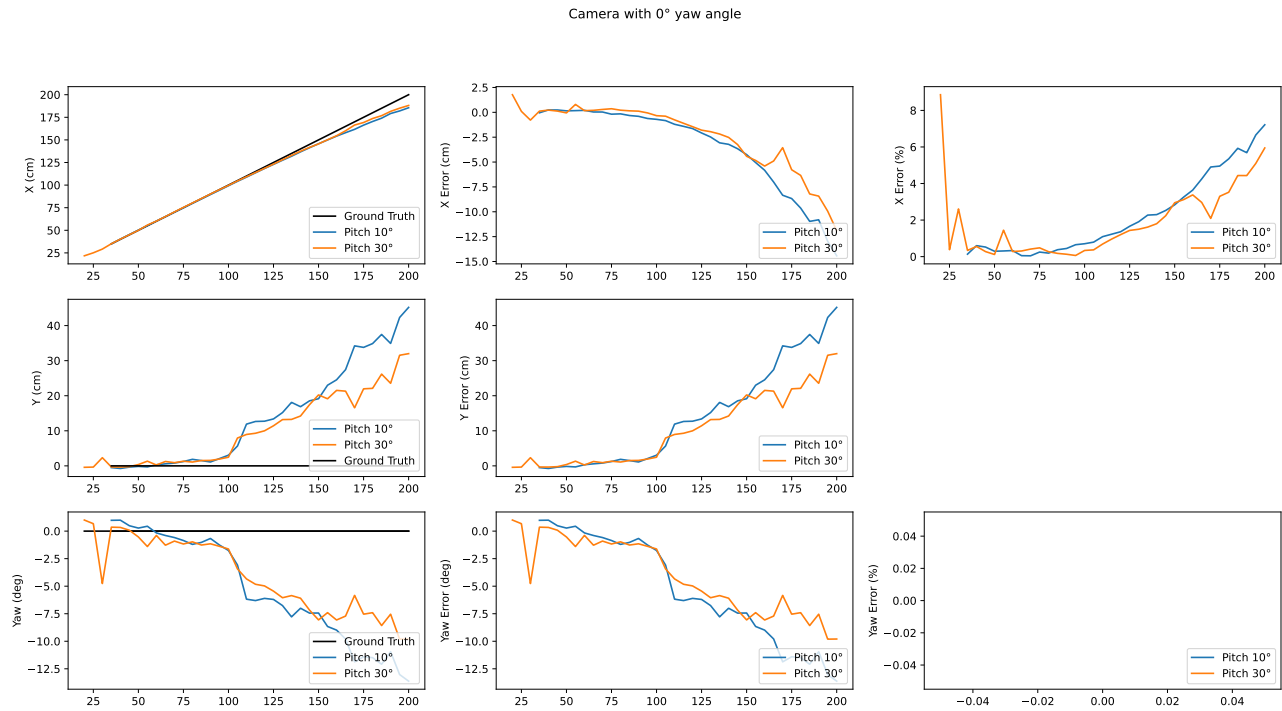


Figure 49: Error of the positioning with 0 degree of yaw angle

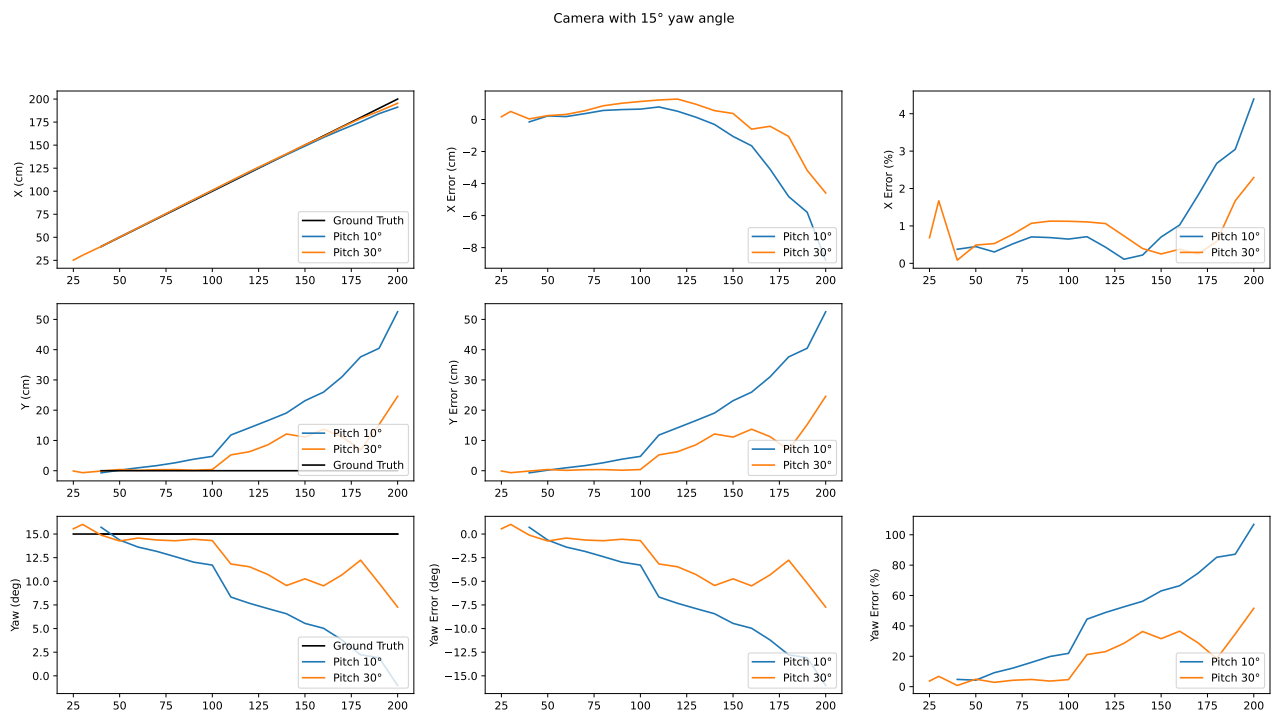


Figure 50: Error of the positioning with 15 degrees yaw angle

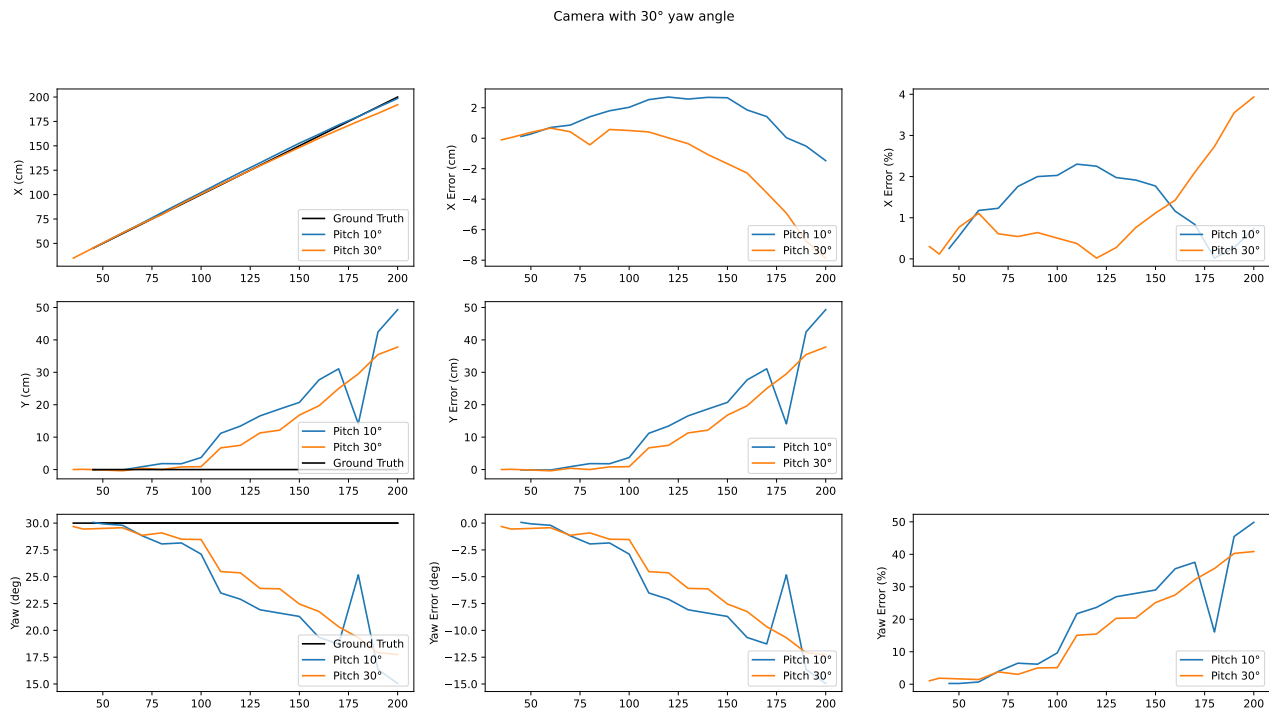


Figure 51: Error of the positioning with 30 degrees yaw angle

When the yaw angle is kept constant and the effect of pitch angles on positioning is also examined, the following results are obtained in ?? and ??:

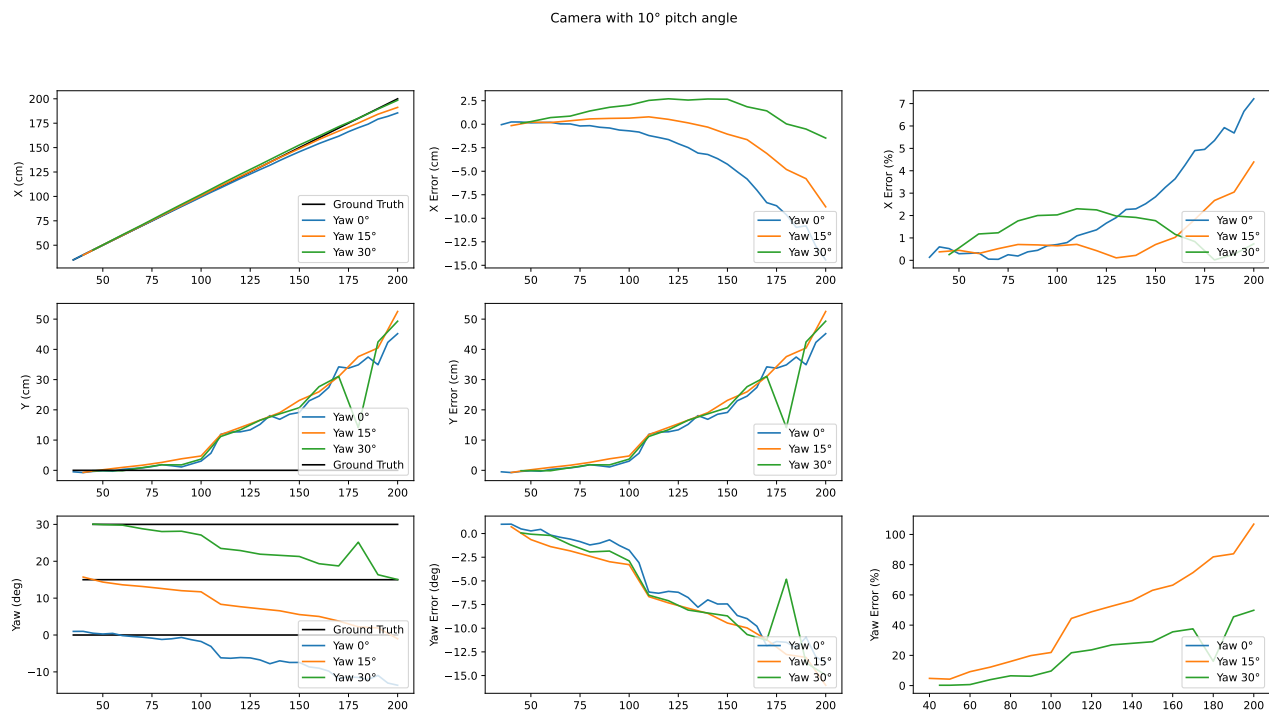


Figure 52: Error of the positioning with 10 degrees of pitch angle

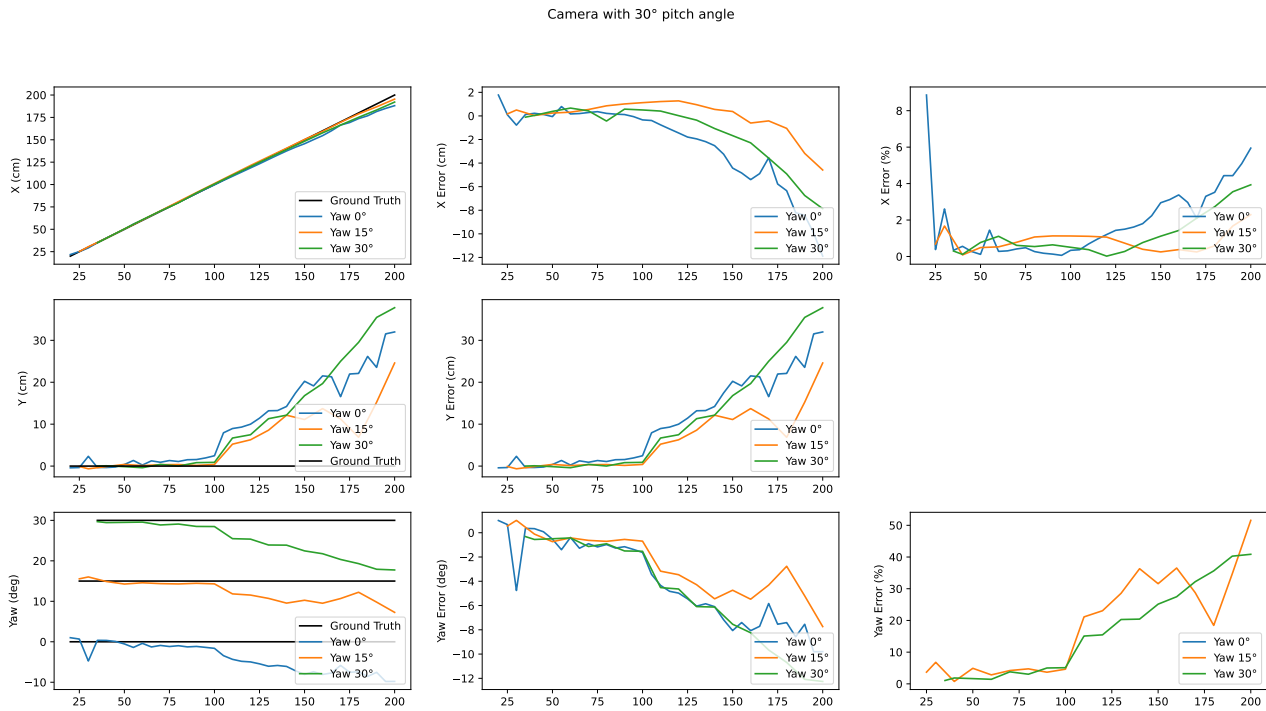


Figure 53: Error of the positioning with 30 degrees of pitch angle

Dynamics positioning Tests

For the dynamic case, two tests were done. First one, **Figure 54**, was done when pitch of the camera equals to 10 degrees and the second one, **Figure 55**, when it equals to 30 degrees.

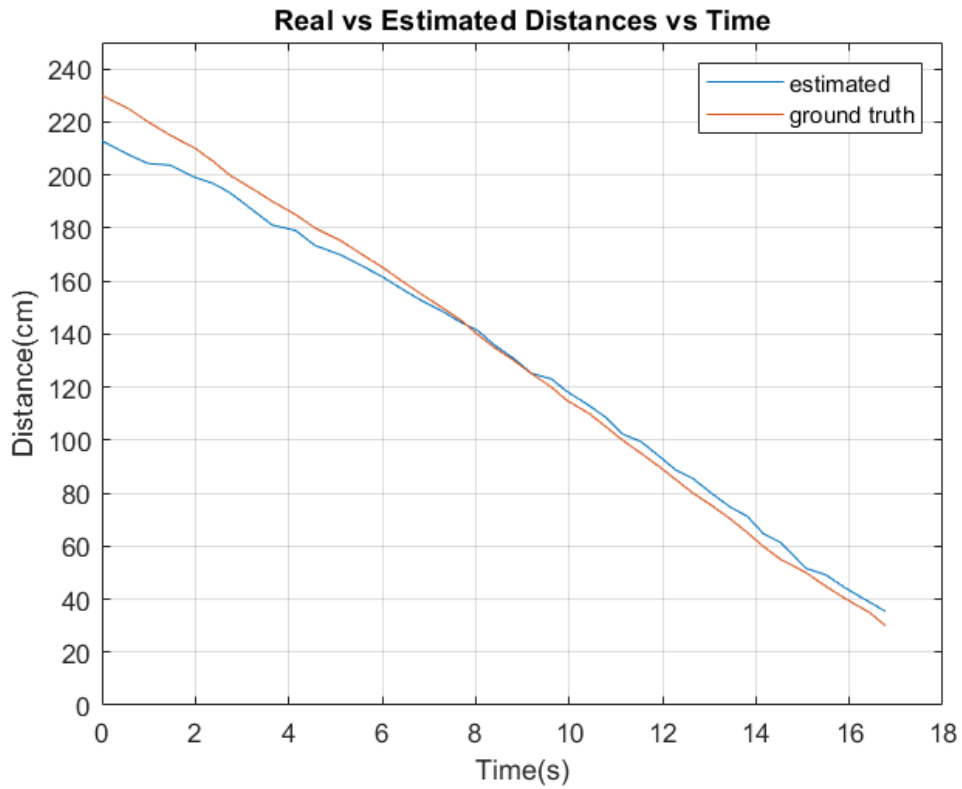


Figure 54: Real and Estimated Distances vs Time For 10 Degree Pitch

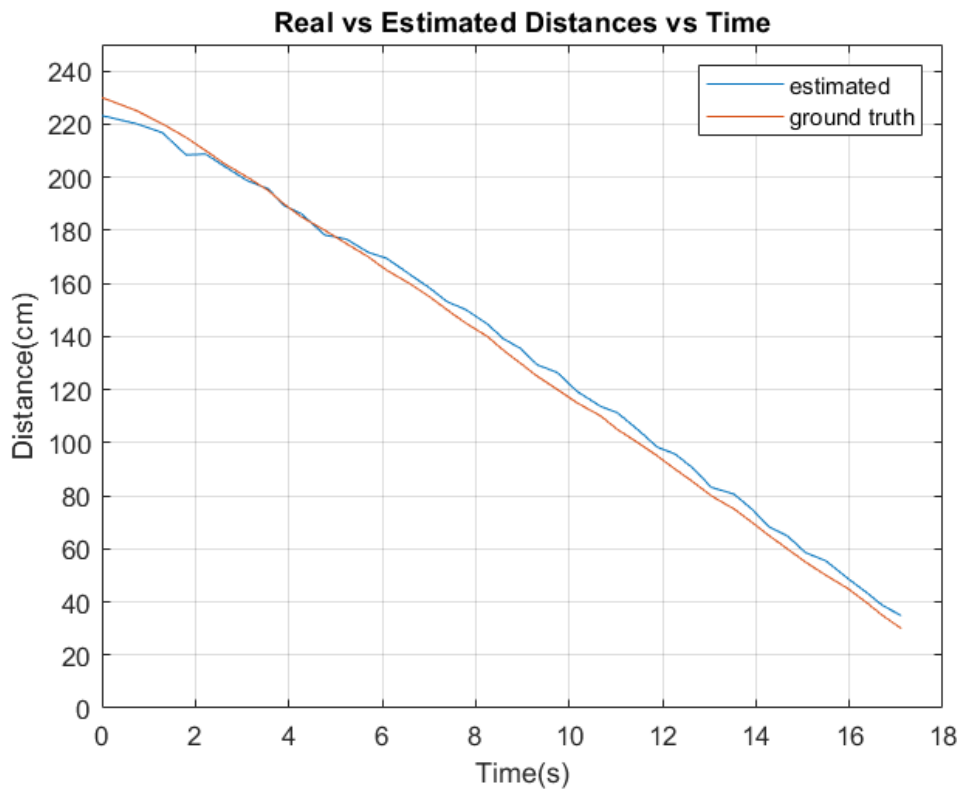


Figure 55: Real and Estimated Distances vs Time For 30 Degree Pitch

Also, error percentages of each case vs the distances are given on the same plot, **Figure 56**.

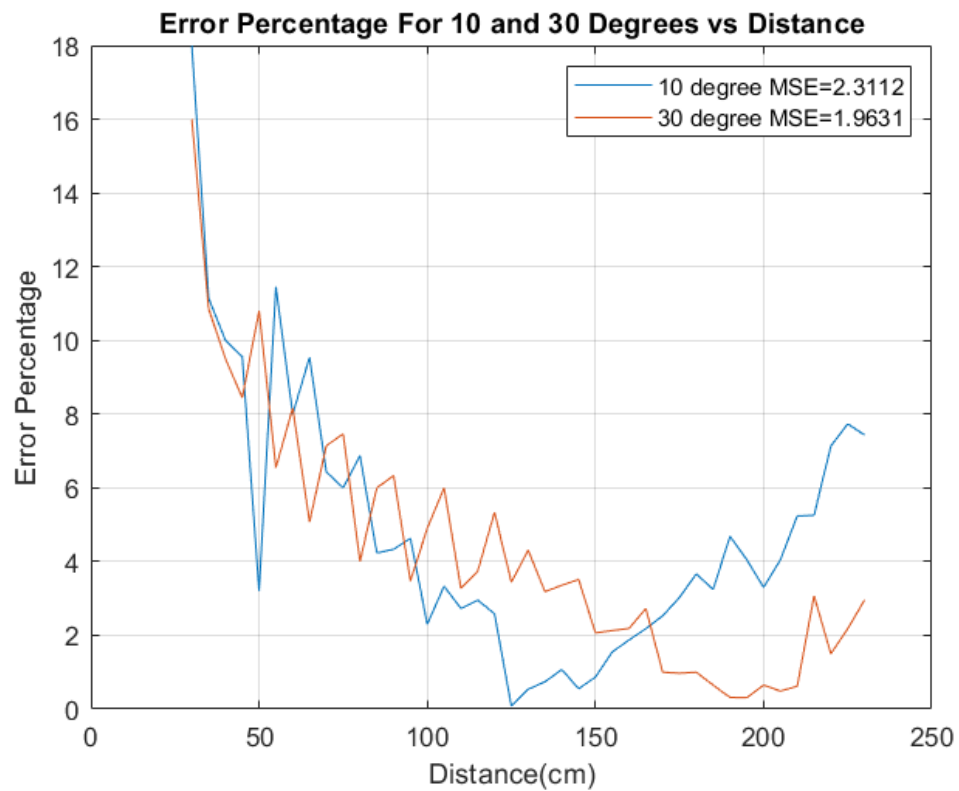


Figure 56: Error Percentages for 10 Degrees and 30 Degrees vs Distances

Finally, tabulated results can be found in **Table 34**

Test	RMSE Error XY	RMSE Error X	RMSE Error Y	Standard Deviation
Vertical Test at Pitch of 10°	21.11	5.50	20.38	14.79
Vertical Test at Pitch of 30°	14.83	4.02	14.27	10.33
Horizontal Test at Pitch of 10°	8.61	0.99	8.56	3.50
Horizontal Test at Pitch of 30°	5.63	0.65	5.59	2.61
Vertical Test at Pitch of 10° and Yaw of 15°	23.21	2.97	23.02	15.80
Vertical Test at Pitch of 30° and Yaw of 15°	9.32	1.47	9.21	6.77
Vertical Test at Pitch of 10° and Yaw of 30°	21.14	1.77	21.06	14.72
Vertical Test at Pitch of 30° and Yaw of 30°	17.30	2.94	17.05	12.84
Tests at Pitch of 10°	26.30	7.05	25.34	16.47
Tests at Pitch of 30°	18.16	6.32	17.03	11.64

Table 34: Tabulated Test Results

Appendix 3: Test Document: Wireless

Location	Electrical-Electronics Department Middle East Technical University
Date	December 20 2023
Time	15:00
Description	Wireless communication through metallic hollow tube by using ESP8266 Wifi
Aim	Determining optimum parameters to establish communication network.
Expected Outcome	Packet errors and latency under different parameter combinations.
Participants	BeyondTech Members and Studio Coordinators

Test Devices and Tools

Required Test Equipments: Hollow Tube, ESP 8266 Module (x2), Micro Usb Cable (x2), Minimum 50 cm Ruler, Computer with Python Interpreter (x2), Attacher (x2)

1. Number of Packets

Ground truth/Calibration: The parameter in the Python script to adjust the number of times transmission command is executed.

2. Packet Content

Ground truth/Calibration: The transmitted data using the Python script which is also printed in the computer connected to the transmitter ESP8266.

3. Distance between Wi-Fi modules

Ground truth/Calibration: The distance between the closest points of the two ESP8266 which is measured with the ruler.

Test Environment

The testing environment consists of two ESP modules, a computer, and a metallic hollow tube. Metallic hollow tube, and ESPs reside on a flat surface.

Environmental Conditions:

1. Temperature Control: The environment is maintained at $25^{\circ}\text{C} \pm 10^{\circ}\text{C}$.
2. Lighting: Standard lighting, with adjustable intensity to simulate various time-of-day conditions.

Test Benches and Setup:

1. Computer will be the device that send the transmission command and output the received packet. The computer require a Python interpreter to be installed. If the computer do not have the Python interpreter, please refer to this link for installation: <https://www.python.org/downloads/release/python-3121/>
2. ESP8266 modules (**Figure 57**) are simple, all-in-one chip that lets microcontrollers use WiFi. It can run programs itself or help another processor with WiFi tasks. In the testing they will be the communication devices that send or receive the packet.

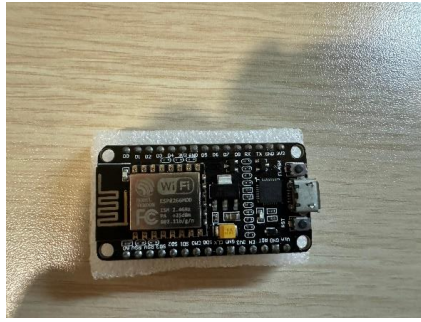


Figure 57: ESP8266 Module

3. Micro USB Cables (**Figure 58**) are miniaturized version of the Universal Serial Bus interface developed for connecting compact and mobile devices. In the testing, it will be used to ensure communication between ESP8266 modules and the computer. Also, it will be powering up the ESP8266 modules.



Figure 58: Micro USB Cable

4. Metallic Hollow Tube (**Figure 59**) has the dimensions of 80x2x500mm. It is the instrument where communication is happening.



Figure 59: Metallic Hollow Tube

5. Ruler (**Figure 60**) is the device for positioning the ESP8266 modules. The minimum length for the ruler is 50cm.



Figure 60: 100cm Ruler

6. The desired test configuration is given in **Figure 61**.

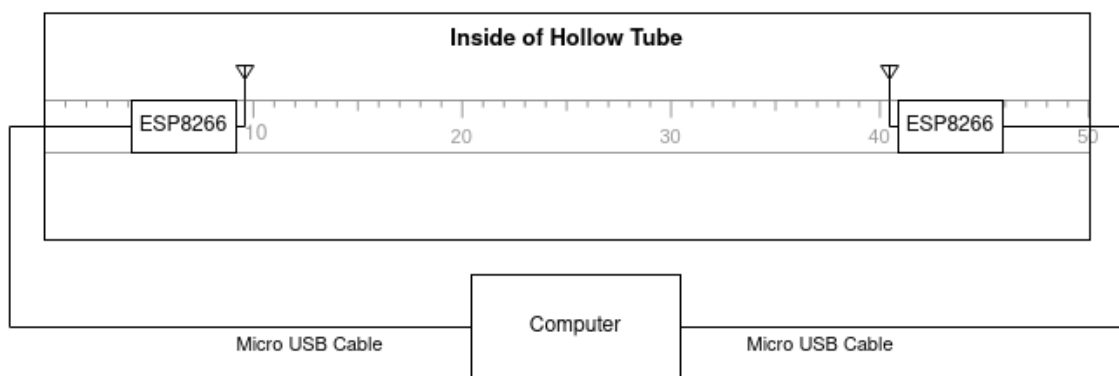


Figure 61: Desired Test Configuration

Test Parameters

Table 35: The control parameters

Parameter	Range	Step Size	Number of Measurements
Distance	30-42 cm	6 cm	3
Time Duration	1-5s	1 s	5
Data Rate	10-260 Packets/s	50 Packets/s	6
Packet Size	30-150 byte	20 bytes	7

The distance does not change much, so we do not expect much change in the output. On the other hand, with the increase in the data rate, we may fill the buffer of transmitter, and some packets may be dropped. Also, some packets may wait longer in the buffer queue which would increase the latency. Similarly, with high packet sizes, buffer can be filled with less packet, so increasing the packet size may cause increase in packet loss and latency. For the time duration, we do not expect much change, but if time ends before the system reaches the steady state, increasing it may help to see the steady state case, and in some situations, buffer may saturate and increase the packet loss and latency.

Test Procedure

1. Establishing Connection: ESP modules are connected to the computer. One terminal runs the receiver script, and the other terminal runs the transmitter script. When a successful transmission is observed, it can be proceeded to the next step.
2. One ESP module is placed on a ruler at the $(25\text{cm} - (\text{Test distance}/2))$ and the other ESP module is placed at the $(25\text{cm} + (\text{Test distance}/2))$ with the help of attachers. Do not contact the metal tube with the ESP modules.
3. The ruler is placed inside of the hollow tube as 0 cm of the ruler coincides with one of the edges of the hollow tube.
4. The ESP modules are connected to the computer with Micro USB cables.
5. The time duration, data rate, and packet size parameters are adjusted in the “transmitter.py” script. This script calculates the time between consecutive transmissions using the data rate as $\text{Inter Message Time} = \frac{1}{\text{Data rate}}$. Using the time duration parameter, this script adjusts the number of packets as $N \text{ of packets} = \text{Data Rate} \cdot \text{Time Duration}$. According to those parameters “transmitter.py” script sends commands to ESP8266 for it to transmit packets. Also, it calculates the latency as the half of the round trip time and outputs it
6. The “receiver.py” script listens to the serial port and when ESP8266 module receives a packet, it extracts the timestamp and echos it back. Also, it counts the received packets.
7. The “receiver.py” script is run. Then the “transmitter.py” script is run. The received packets and the latency is observed on the computer.
8. Step 7 is repeated 10 times and the average received packet number and the average latencies with their standard deviations are calculated.
9. Steps 7 and 8 are repeated with different parameters mentioned in Step 5.
10. Steps 5-9 is repeated for different test distance length.

Delay Measurement

1. The distance between the ESP modules are set to 30 cm.
2. The Python script where each ESP is used as both access point and station is run. This script allows the second module to send acknowledgement when it receives a packet. In the first computer, the transmission time, acknowledgment reception time, and half of the difference between those times are shown.
3. The delays are taken to be equal for the transmissions in both directions, so half of the difference is taken as the delay between transmission and reception.
4. This procedure is repeated 20 times, and the average delay is calculated.

Test Data

Table 36: Wireless test at 30cm distance between modules

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
1.0	10.0	30.0	1.38	10.0	0.133	0.0
1.0	10.0	50.0	1.53	10.0	0.133	0.0
1.0	10.0	70.0	1.47	10.0	0.081	0.0
1.0	10.0	90.0	1.54	10.0	0.229	0.0
1.0	10.0	110.0	1.41	10.0	0.146	0.0
1.0	10.0	130.0	1.4	10.0	0.114	0.0
1.0	10.0	150.0	1.49	10.0	0.066	0.0
1.0	60.0	30.0	1.028	60.0	0.066	0.0
1.0	60.0	50.0	1.047	60.0	0.038	0.0
1.0	60.0	70.0	1.093	60.0	0.117	0.0
1.0	60.0	90.0	1.087	60.0	0.162	0.0
1.0	60.0	110.0	1.095	60.0	0.11	0.0
1.0	60.0	130.0	1.333	60.0	0.408	0.0
1.0	60.0	150.0	1.165	60.0	0.084	0.0
1.0	110.0	30.0	0.993	110.0	0.124	0.0
1.0	110.0	50.0	1.089	110.0	0.156	0.0
1.0	110.0	70.0	0.98	110.0	0.028	0.0
1.0	110.0	90.0	1.016	110.0	0.113	0.0
1.0	110.0	110.0	1.026	110.0	0.086	0.0
1.0	110.0	130.0	1.074	110.0	0.108	0.0
1.0	110.0	150.0	1.09	110.0	0.132	0.0
1.0	160.0	30.0	1.004	160.0	0.146	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
1.0	160.0	50.0	0.993	160.0	0.111	0.0
1.0	160.0	70.0	1.01	160.0	0.105	0.0
1.0	160.0	90.0	0.876	160.0	0.039	0.0
1.0	160.0	110.0	1.011	160.0	0.042	0.0
1.0	160.0	130.0	0.932	160.0	0.069	0.0
1.0	160.0	150.0	1.157	160.0	0.125	0.0
1.0	210.0	30.0	0.945	210.0	0.069	0.0
1.0	210.0	50.0	1.143	209.6	0.259	0.8
1.0	210.0	70.0	1.02	210.0	0.098	0.0
1.0	210.0	90.0	1.209	210.0	0.264	0.0
1.0	210.0	110.0	1.013	210.0	0.099	0.0
1.0	210.0	130.0	1.067	210.0	0.232	0.0
1.0	210.0	150.0	1.115	210.0	0.223	0.0
1.0	260.0	30.0	1.167	258.8	0.291	1.47
1.0	260.0	50.0	1.114	259.6	0.185	0.8
1.0	260.0	70.0	0.979	260.0	0.085	0.0
1.0	260.0	90.0	1.002	260.0	0.038	0.0
1.0	260.0	110.0	1.143	260.0	0.229	0.0
1.0	260.0	130.0	1.294	257.6	0.435	2.939
1.0	260.0	150.0	1.083	260.0	0.099	0.0
2.0	10.0	30.0	1.585	20.0	0.222	0.0
2.0	10.0	50.0	1.41	20.0	0.082	0.0
2.0	10.0	70.0	1.43	20.0	0.113	0.0
2.0	10.0	90.0	1.5	20.0	0.15	0.0
2.0	10.0	110.0	1.455	20.0	0.07	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
2.0	10.0	130.0	1.54	20.0	0.265	0.0
2.0	10.0	150.0	1.57	20.0	0.058	0.0
2.0	60.0	30.0	1.062	120.0	0.096	0.0
2.0	60.0	50.0	1.011	120.0	0.05	0.0
2.0	60.0	70.0	1.127	120.0	0.111	0.0
2.0	60.0	90.0	1.114	120.0	0.056	0.0
2.0	60.0	110.0	1.076	120.0	0.028	0.0
2.0	60.0	130.0	1.096	120.0	0.067	0.0
2.0	60.0	150.0	1.072	120.0	0.052	0.0
2.0	110.0	30.0	1.107	220.0	0.327	0.0
2.0	110.0	50.0	1.102	220.0	0.054	0.0
2.0	110.0	70.0	1.034	220.0	0.112	0.0
2.0	110.0	90.0	1.208	220.0	0.204	0.0
2.0	110.0	110.0	1.121	220.0	0.163	0.0
2.0	110.0	130.0	1.227	220.0	0.328	0.0
2.0	110.0	150.0	1.13	220.0	0.122	0.0
2.0	160.0	30.0	1.049	320.0	0.148	0.0
2.0	160.0	50.0	0.999	320.0	0.095	0.0
2.0	160.0	70.0	1.043	320.0	0.073	0.0
2.0	160.0	90.0	0.999	320.0	0.074	0.0
2.0	160.0	110.0	1.016	320.0	0.103	0.0
2.0	160.0	130.0	1.124	320.0	0.148	0.0
2.0	160.0	150.0	1.049	320.0	0.062	0.0
2.0	210.0	30.0	1.118	420.0	0.191	0.0
2.0	210.0	50.0	1.029	420.0	0.089	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
2.0	210.0	70.0	1.077	419.8	0.182	0.4
2.0	210.0	90.0	0.957	420.0	0.029	0.0
2.0	210.0	110.0	1.033	419.8	0.084	0.4
2.0	210.0	130.0	1.081	420.0	0.095	0.0
2.0	210.0	150.0	1.014	420.0	0.061	0.0
2.0	260.0	30.0	1.089	520.0	0.078	0.0
2.0	260.0	50.0	1.125	519.6	0.039	0.49
2.0	260.0	70.0	1.046	520.0	0.122	0.0
2.0	260.0	90.0	1.077	520.0	0.086	0.0
2.0	260.0	110.0	1.325	517.8	0.193	3.487
2.0	260.0	130.0	1.137	518.6	0.153	2.8
2.0	260.0	150.0	1.329	515.6	0.162	2.417
3.0	10.0	30.0	1.843	30.0	0.281	0.0
3.0	10.0	50.0	1.74	30.0	0.268	0.0
3.0	10.0	70.0	1.63	30.0	0.216	0.0
3.0	10.0	90.0	1.51	30.0	0.12	0.0
3.0	10.0	110.0	1.597	30.0	0.217	0.0
3.0	10.0	130.0	1.48	30.0	0.059	0.0
3.0	10.0	150.0	1.607	30.0	0.294	0.0
3.0	60.0	30.0	1.081	180.0	0.021	0.0
3.0	60.0	50.0	1.067	180.0	0.051	0.0
3.0	60.0	70.0	1.08	180.0	0.032	0.0
3.0	60.0	90.0	1.076	180.0	0.027	0.0
3.0	60.0	110.0	1.167	180.0	0.155	0.0
3.0	60.0	130.0	1.123	180.0	0.097	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
3.0	60.0	150.0	1.122	180.0	0.031	0.0
3.0	110.0	30.0	1.02	330.0	0.045	0.0
3.0	110.0	50.0	1.056	330.0	0.081	0.0
3.0	110.0	70.0	1.021	330.0	0.041	0.0
3.0	110.0	90.0	1.051	330.0	0.058	0.0
3.0	110.0	110.0	1.148	330.0	0.092	0.0
3.0	110.0	130.0	1.049	330.0	0.046	0.0
3.0	110.0	150.0	1.136	329.8	0.108	0.4
3.0	160.0	30.0	1.001	480.0	0.1	0.0
3.0	160.0	50.0	1.012	480.0	0.088	0.0
3.0	160.0	70.0	1.015	480.0	0.107	0.0
3.0	160.0	90.0	1.032	480.0	0.066	0.0
3.0	160.0	110.0	1.097	480.0	0.165	0.0
3.0	160.0	130.0	1.04	480.0	0.078	0.0
3.0	160.0	150.0	0.994	480.0	0.053	0.0
3.0	210.0	30.0	0.998	630.0	0.071	0.0
3.0	210.0	50.0	0.985	630.0	0.058	0.0
3.0	210.0	70.0	1.038	629.6	0.083	0.8
3.0	210.0	90.0	1.055	630.0	0.101	0.0
3.0	210.0	110.0	1.07	630.0	0.082	0.0
3.0	210.0	130.0	1.139	629.8	0.125	0.4
3.0	210.0	150.0	1.055	630.0	0.058	0.0
3.0	260.0	30.0	1.045	779.6	0.042	0.8
3.0	260.0	50.0	1.087	780.0	0.149	0.0
3.0	260.0	70.0	1.103	779.0	0.103	1.549

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
3.0	260.0	90.0	1.053	779.8	0.061	0.4
3.0	260.0	110.0	1.134	779.8	0.086	0.4
3.0	260.0	130.0	1.136	779.4	0.143	0.8
3.0	260.0	150.0	1.088	779.0	0.052	1.265
4.0	10.0	30.0	1.615	40.0	0.244	0.0
4.0	10.0	50.0	1.76	40.0	0.238	0.0
4.0	10.0	70.0	1.5	40.0	0.084	0.0
4.0	10.0	90.0	1.47	40.0	0.08	0.0
4.0	10.0	110.0	2.013	40.0	0.858	0.0
4.0	10.0	130.0	1.512	40.0	0.073	0.0
4.0	10.0	150.0	1.478	40.0	0.086	0.0
4.0	60.0	30.0	1.105	240.0	0.118	0.0
4.0	60.0	50.0	1.102	240.0	0.088	0.0
4.0	60.0	70.0	1.124	240.0	0.095	0.0
4.0	60.0	90.0	1.076	240.0	0.027	0.0
4.0	60.0	110.0	1.064	240.0	0.038	0.0
4.0	60.0	130.0	1.077	240.0	0.036	0.0
4.0	60.0	150.0	1.115	240.0	0.056	0.0
4.0	110.0	30.0	1.0	440.0	0.061	0.0
4.0	110.0	50.0	1.045	440.0	0.079	0.0
4.0	110.0	70.0	1.028	440.0	0.051	0.0
4.0	110.0	90.0	1.053	440.0	0.045	0.0
4.0	110.0	110.0	1.012	440.0	0.059	0.0
4.0	110.0	130.0	1.029	440.0	0.037	0.0
4.0	110.0	150.0	1.043	440.0	0.073	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
4.0	160.0	30.0	0.963	640.0	0.075	0.0
4.0	160.0	50.0	0.992	640.0	0.091	0.0
4.0	160.0	70.0	1.026	640.0	0.036	0.0
4.0	160.0	90.0	0.987	640.0	0.074	0.0
4.0	160.0	110.0	1.049	640.0	0.088	0.0
4.0	160.0	130.0	1.0	640.0	0.067	0.0
4.0	160.0	150.0	1.059	640.0	0.085	0.0
4.0	210.0	30.0	1.055	839.8	0.209	0.4
4.0	210.0	50.0	1.084	839.8	0.211	0.4
4.0	210.0	70.0	1.057	840.0	0.104	0.0
4.0	210.0	90.0	1.096	838.6	0.123	1.96
4.0	210.0	110.0	1.101	838.4	0.127	1.625
4.0	210.0	130.0	1.061	840.0	0.095	0.0
4.0	210.0	150.0	1.087	839.6	0.076	0.8
4.0	260.0	30.0	1.114	1036.8	0.114	3.25
4.0	260.0	50.0	1.133	1036.4	0.074	2.059
4.0	260.0	70.0	1.06	1039.6	0.131	0.8
4.0	260.0	90.0	1.104	1038.8	0.107	1.939
4.0	260.0	110.0	1.113	1037.6	0.129	3.007
4.0	260.0	130.0	1.119	1038.8	0.108	1.6
4.0	260.0	150.0	1.026	1039.8	0.073	0.4
5.0	10.0	30.0	1.436	50.0	0.094	0.0
5.0	10.0	50.0	1.438	50.0	0.083	0.0
5.0	10.0	70.0	1.42	50.0	0.026	0.0
5.0	10.0	90.0	1.518	50.0	0.119	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
5.0	10.0	110.0	1.584	50.0	0.166	0.0
5.0	10.0	130.0	1.592	50.0	0.17	0.0
5.0	10.0	150.0	1.484	50.0	0.04	0.0
5.0	60.0	30.0	1.128	300.0	0.126	0.0
5.0	60.0	50.0	1.121	300.0	0.04	0.0
5.0	60.0	70.0	1.046	300.0	0.046	0.0
5.0	60.0	90.0	1.062	300.0	0.031	0.0
5.0	60.0	110.0	1.12	300.0	0.05	0.0
5.0	60.0	130.0	1.084	300.0	0.033	0.0
5.0	60.0	150.0	1.171	300.0	0.073	0.0
5.0	110.0	30.0	1.06	550.0	0.04	0.0
5.0	110.0	50.0	1.123	550.0	0.13	0.0
5.0	110.0	70.0	1.058	550.0	0.08	0.0
5.0	110.0	90.0	0.99	550.0	0.027	0.0
5.0	110.0	110.0	1.026	550.0	0.026	0.0
5.0	110.0	130.0	1.075	550.0	0.053	0.0
5.0	110.0	150.0	1.119	550.0	0.078	0.0
5.0	160.0	30.0	1.043	800.0	0.096	0.0
5.0	160.0	50.0	1.023	800.0	0.05	0.0
5.0	160.0	70.0	1.023	800.0	0.104	0.0
5.0	160.0	90.0	0.976	800.0	0.073	0.0
5.0	160.0	110.0	1.02	799.6	0.103	0.8
5.0	160.0	130.0	1.022	800.0	0.07	0.0
5.0	160.0	150.0	0.968	800.0	0.027	0.0
5.0	210.0	30.0	1.163	1048.4	0.061	1.357

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
5.0	210.0	50.0	1.057	1049.4	0.105	0.8
5.0	210.0	70.0	1.001	1049.8	0.035	0.4
5.0	210.0	90.0	1.045	1049.8	0.051	0.4
5.0	210.0	110.0	1.081	1049.4	0.116	1.2
5.0	210.0	130.0	0.996	1050.0	0.065	0.0
5.0	210.0	150.0	1.038	1050.0	0.105	0.0
5.0	260.0	30.0	1.01	1300.0	0.06	0.0
5.0	260.0	50.0	1.066	1298.2	0.132	2.713
5.0	260.0	70.0	1.123	1298.6	0.093	2.8
5.0	260.0	90.0	1.092	1299.8	0.078	0.4
5.0	260.0	110.0	1.104	1299.2	0.1	1.166
5.0	260.0	130.0	1.093	1298.6	0.09	1.855
5.0	260.0	150.0	1.023	1298.8	0.072	1.939

Table 37: Wireless test at 36cm distance between modules

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
1.0	10.0	30.0	1.6	10.0	0.283	0.0
1.0	10.0	50.0	1.59	10.0	0.107	0.0
1.0	10.0	70.0	1.41	10.0	0.116	0.0
1.0	10.0	90.0	1.96	10.0	0.748	0.0
1.0	10.0	110.0	1.43	10.0	0.075	0.0
1.0	10.0	130.0	1.5	10.0	0.176	0.0
1.0	10.0	150.0	1.5	10.0	0.089	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
1.0	60.0	30.0	1.068	60.0	0.067	0.0
1.0	60.0	50.0	1.015	60.0	0.059	0.0
1.0	60.0	70.0	1.013	60.0	0.036	0.0
1.0	60.0	90.0	1.052	60.0	0.048	0.0
1.0	60.0	110.0	1.032	60.0	0.059	0.0
1.0	60.0	130.0	1.057	60.0	0.049	0.0
1.0	60.0	150.0	1.072	60.0	0.029	0.0
1.0	110.0	30.0	1.011	110.0	0.087	0.0
1.0	110.0	50.0	1.115	110.0	0.243	0.0
1.0	110.0	70.0	1.281	110.0	0.435	0.0
1.0	110.0	90.0	0.995	110.0	0.07	0.0
1.0	110.0	110.0	0.977	110.0	0.069	0.0
1.0	110.0	130.0	1.042	110.0	0.078	0.0
1.0	110.0	150.0	1.108	110.0	0.12	0.0
1.0	160.0	30.0	0.881	160.0	0.023	0.0
1.0	160.0	50.0	1.063	160.0	0.203	0.0
1.0	160.0	70.0	0.981	160.0	0.18	0.0
1.0	160.0	90.0	0.993	160.0	0.056	0.0
1.0	160.0	110.0	1.014	160.0	0.136	0.0
1.0	160.0	130.0	0.996	160.0	0.088	0.0
1.0	160.0	150.0	1.227	159.8	0.513	0.4
1.0	210.0	30.0	1.11	209.8	0.176	0.4
1.0	210.0	50.0	0.969	210.0	0.091	0.0
1.0	210.0	70.0	1.016	210.0	0.225	0.0
1.0	210.0	90.0	0.995	210.0	0.095	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
1.0	210.0	110.0	0.987	210.0	0.061	0.0
1.0	210.0	130.0	1.162	210.0	0.186	0.0
1.0	210.0	150.0	1.0	210.0	0.101	0.0
1.0	260.0	30.0	1.201	258.4	0.336	3.2
1.0	260.0	50.0	1.146	260.0	0.316	0.0
1.0	260.0	70.0	1.229	258.4	0.277	2.332
1.0	260.0	90.0	1.208	259.6	0.242	0.8
1.0	260.0	110.0	1.161	260.0	0.228	0.0
1.0	260.0	130.0	1.008	260.0	0.084	0.0
1.0	260.0	150.0	1.126	260.0	0.252	0.0
2.0	10.0	30.0	1.56	20.0	0.102	0.0
2.0	10.0	50.0	1.56	20.0	0.243	0.0
2.0	10.0	70.0	1.405	20.0	0.076	0.0
2.0	10.0	90.0	1.535	20.0	0.041	0.0
2.0	10.0	110.0	1.51	20.0	0.09	0.0
2.0	10.0	130.0	1.415	20.0	0.086	0.0
2.0	10.0	150.0	1.595	20.0	0.17	0.0
2.0	60.0	30.0	1.076	120.0	0.098	0.0
2.0	60.0	50.0	1.12	120.0	0.107	0.0
2.0	60.0	70.0	1.064	120.0	0.063	0.0
2.0	60.0	90.0	1.158	120.0	0.203	0.0
2.0	60.0	110.0	1.056	120.0	0.082	0.0
2.0	60.0	130.0	1.087	120.0	0.067	0.0
2.0	60.0	150.0	1.08	120.0	0.119	0.0
2.0	110.0	30.0	0.977	220.0	0.145	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
2.0	110.0	50.0	0.985	220.0	0.09	0.0
2.0	110.0	70.0	0.975	220.0	0.06	0.0
2.0	110.0	90.0	1.099	220.0	0.178	0.0
2.0	110.0	110.0	1.065	220.0	0.18	0.0
2.0	110.0	130.0	1.224	220.0	0.35	0.0
2.0	110.0	150.0	1.088	220.0	0.201	0.0
2.0	160.0	30.0	0.966	319.8	0.097	0.4
2.0	160.0	50.0	0.88	320.0	0.068	0.0
2.0	160.0	70.0	0.92	320.0	0.083	0.0
2.0	160.0	90.0	0.948	320.0	0.148	0.0
2.0	160.0	110.0	0.959	320.0	0.114	0.0
2.0	160.0	130.0	0.986	320.0	0.045	0.0
2.0	160.0	150.0	0.98	320.0	0.099	0.0
2.0	210.0	30.0	1.018	420.0	0.117	0.0
2.0	210.0	50.0	0.992	419.6	0.121	0.8
2.0	210.0	70.0	0.996	420.0	0.096	0.0
2.0	210.0	90.0	1.013	420.0	0.069	0.0
2.0	210.0	110.0	1.04	420.0	0.101	0.0
2.0	210.0	130.0	1.099	419.6	0.122	0.49
2.0	210.0	150.0	0.998	420.0	0.059	0.0
2.0	260.0	30.0	1.041	520.0	0.065	0.0
2.0	260.0	50.0	1.008	520.0	0.036	0.0
2.0	260.0	70.0	1.091	517.8	0.147	2.713
2.0	260.0	90.0	1.165	517.0	0.205	3.347
2.0	260.0	110.0	1.077	520.0	0.076	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
2.0	260.0	130.0	1.219	516.6	0.224	2.498
2.0	260.0	150.0	1.149	517.4	0.276	5.2
3.0	10.0	30.0	1.503	30.0	0.11	0.0
3.0	10.0	50.0	1.637	30.0	0.3	0.0
3.0	10.0	70.0	1.56	30.0	0.085	0.0
3.0	10.0	90.0	1.507	30.0	0.023	0.0
3.0	10.0	110.0	1.537	30.0	0.073	0.0
3.0	10.0	130.0	1.61	30.0	0.267	0.0
3.0	10.0	150.0	1.607	30.0	0.1	0.0
3.0	60.0	30.0	1.063	180.0	0.071	0.0
3.0	60.0	50.0	1.244	180.0	0.372	0.0
3.0	60.0	70.0	1.108	180.0	0.093	0.0
3.0	60.0	90.0	1.134	180.0	0.106	0.0
3.0	60.0	110.0	1.109	180.0	0.067	0.0
3.0	60.0	130.0	1.154	180.0	0.035	0.0
3.0	60.0	150.0	1.163	180.0	0.125	0.0
3.0	110.0	30.0	1.037	330.0	0.052	0.0
3.0	110.0	50.0	1.145	329.8	0.146	0.4
3.0	110.0	70.0	1.051	330.0	0.064	0.0
3.0	110.0	90.0	1.029	330.0	0.052	0.0
3.0	110.0	110.0	1.008	330.0	0.031	0.0
3.0	110.0	130.0	1.152	330.0	0.087	0.0
3.0	110.0	150.0	1.117	330.0	0.113	0.0
3.0	160.0	30.0	1.004	480.0	0.075	0.0
3.0	160.0	50.0	0.927	480.0	0.033	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
3.0	160.0	70.0	0.949	480.0	0.027	0.0
3.0	160.0	90.0	1.077	480.0	0.158	0.0
3.0	160.0	110.0	1.072	479.8	0.097	0.4
3.0	160.0	130.0	1.212	479.6	0.255	0.49
3.0	160.0	150.0	1.126	479.8	0.217	0.4
3.0	210.0	30.0	1.085	629.8	0.159	0.4
3.0	210.0	50.0	1.051	629.4	0.081	0.8
3.0	210.0	70.0	1.007	630.0	0.051	0.0
3.0	210.0	90.0	1.029	630.0	0.094	0.0
3.0	210.0	110.0	0.985	630.0	0.045	0.0
3.0	210.0	130.0	1.029	629.4	0.12	1.2
3.0	210.0	150.0	1.042	630.0	0.09	0.0
3.0	260.0	30.0	1.022	778.4	0.116	3.2
3.0	260.0	50.0	1.189	778.0	0.209	2.53
3.0	260.0	70.0	1.042	779.4	0.071	1.2
3.0	260.0	90.0	1.04	780.0	0.066	0.0
3.0	260.0	110.0	1.149	778.8	0.209	2.4
3.0	260.0	130.0	1.063	778.8	0.096	1.939
3.0	260.0	150.0	1.131	776.4	0.145	4.8
4.0	10.0	30.0	1.627	40.0	0.287	0.0
4.0	10.0	50.0	1.633	40.0	0.157	0.0
4.0	10.0	70.0	1.68	40.0	0.197	0.0
4.0	10.0	90.0	1.53	40.0	0.062	0.0
4.0	10.0	110.0	1.465	40.0	0.068	0.0
4.0	10.0	130.0	1.455	40.0	0.128	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
4.0	10.0	150.0	1.45	40.0	0.022	0.0
4.0	60.0	30.0	1.092	240.0	0.081	0.0
4.0	60.0	50.0	1.108	240.0	0.116	0.0
4.0	60.0	70.0	1.121	240.0	0.1	0.0
4.0	60.0	90.0	1.041	240.0	0.041	0.0
4.0	60.0	110.0	1.084	240.0	0.082	0.0
4.0	60.0	130.0	1.16	240.0	0.038	0.0
4.0	60.0	150.0	1.16	240.0	0.083	0.0
4.0	110.0	30.0	1.056	440.0	0.151	0.0
4.0	110.0	50.0	1.112	440.0	0.161	0.0
4.0	110.0	70.0	1.072	440.0	0.188	0.0
4.0	110.0	90.0	1.185	440.0	0.178	0.0
4.0	110.0	110.0	1.22	439.8	0.122	0.4
4.0	110.0	130.0	1.077	440.0	0.056	0.0
4.0	110.0	150.0	1.146	440.0	0.063	0.0
4.0	160.0	30.0	0.949	640.0	0.042	0.0
4.0	160.0	50.0	0.965	640.0	0.048	0.0
4.0	160.0	70.0	1.062	639.8	0.096	0.4
4.0	160.0	90.0	1.054	639.8	0.109	0.4
4.0	160.0	110.0	1.107	638.8	0.137	1.47
4.0	160.0	130.0	1.049	640.0	0.061	0.0
4.0	160.0	150.0	1.079	640.0	0.148	0.0
4.0	210.0	30.0	1.178	837.6	0.208	3.2
4.0	210.0	50.0	1.113	838.0	0.083	1.897
4.0	210.0	70.0	1.056	838.4	0.092	1.96

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
4.0	210.0	90.0	1.073	838.4	0.107	2.728
4.0	210.0	110.0	1.153	838.4	0.168	2.332
4.0	210.0	130.0	1.102	839.4	0.146	0.8
4.0	210.0	150.0	1.129	838.8	0.116	1.939
4.0	260.0	30.0	1.12	1035.2	0.13	5.913
4.0	260.0	50.0	1.102	1035.8	0.103	4.261
4.0	260.0	70.0	1.141	1036.6	0.147	4.716
4.0	260.0	90.0	1.089	1037.8	0.104	2.857
4.0	260.0	110.0	1.032	1039.6	0.07	0.8
4.0	260.0	130.0	1.119	1035.8	0.125	5.231
4.0	260.0	150.0	1.103	1040.0	0.078	0.0
5.0	10.0	30.0	1.526	50.0	0.084	0.0
5.0	10.0	50.0	1.698	50.0	0.192	0.0
5.0	10.0	70.0	1.574	50.0	0.119	0.0
5.0	10.0	90.0	1.666	50.0	0.255	0.0
5.0	10.0	110.0	1.528	50.0	0.083	0.0
5.0	10.0	130.0	1.594	50.0	0.175	0.0
5.0	10.0	150.0	1.568	50.0	0.142	0.0
5.0	60.0	30.0	1.125	300.0	0.134	0.0
5.0	60.0	50.0	1.121	300.0	0.102	0.0
5.0	60.0	70.0	1.111	300.0	0.097	0.0
5.0	60.0	90.0	1.244	300.0	0.057	0.0
5.0	60.0	110.0	1.092	300.0	0.079	0.0
5.0	60.0	130.0	1.079	300.0	0.066	0.0
5.0	60.0	150.0	1.074	300.0	0.045	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
5.0	110.0	30.0	1.013	550.0	0.081	0.0
5.0	110.0	50.0	1.03	550.0	0.076	0.0
5.0	110.0	70.0	0.984	550.0	0.056	0.0
5.0	110.0	90.0	1.065	550.0	0.047	0.0
5.0	110.0	110.0	1.015	550.0	0.092	0.0
5.0	110.0	130.0	1.116	550.0	0.153	0.0
5.0	110.0	150.0	1.13	549.8	0.112	0.4
5.0	160.0	30.0	1.102	799.6	0.108	0.8
5.0	160.0	50.0	0.988	800.0	0.062	0.0
5.0	160.0	70.0	1.018	799.6	0.109	0.49
5.0	160.0	90.0	1.078	799.4	0.142	0.8
5.0	160.0	110.0	1.02	800.0	0.037	0.0
5.0	160.0	130.0	1.05	799.8	0.071	0.4
5.0	160.0	150.0	1.005	800.0	0.039	0.0
5.0	210.0	30.0	0.972	1049.2	0.104	1.6
5.0	210.0	50.0	0.972	1048.6	0.105	2.8
5.0	210.0	70.0	1.042	1049.8	0.098	0.4
5.0	210.0	90.0	0.989	1048.6	0.121	2.332
5.0	210.0	110.0	0.981	1048.8	0.061	2.4
5.0	210.0	130.0	0.994	1048.8	0.085	2.4
5.0	210.0	150.0	0.992	1050.0	0.065	0.0
5.0	260.0	30.0	1.03	1299.6	0.125	0.8
5.0	260.0	50.0	0.997	1299.8	0.103	0.4
5.0	260.0	70.0	0.985	1297.6	0.071	4.317
5.0	260.0	90.0	1.107	1296.0	0.109	3.033

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
5.0	260.0	110.0	1.08	1298.0	0.111	2.53
5.0	260.0	130.0	1.106	1296.2	0.096	4.75
5.0	260.0	150.0	1.062	1297.0	0.175	6.0

Table 38: Wireless test at 42cm distance between modules

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
1.0	10.0	30.0	1.58	10.0	0.238	0.0
1.0	10.0	50.0	1.78	10.0	0.391	0.0
1.0	10.0	70.0	1.54	10.0	0.073	0.0
1.0	10.0	90.0	1.61	10.0	0.086	0.0
1.0	10.0	110.0	1.53	10.0	0.14	0.0
1.0	10.0	130.0	1.58	10.0	0.225	0.0
1.0	10.0	150.0	2.01	10.0	0.811	0.0
1.0	60.0	30.0	1.047	60.0	0.042	0.0
1.0	60.0	50.0	1.052	60.0	0.031	0.0
1.0	60.0	70.0	1.178	60.0	0.294	0.0
1.0	60.0	90.0	1.09	60.0	0.054	0.0
1.0	60.0	110.0	1.17	60.0	0.117	0.0
1.0	60.0	130.0	1.092	60.0	0.085	0.0
1.0	60.0	150.0	1.267	60.0	0.195	0.0
1.0	110.0	30.0	1.02	110.0	0.11	0.0
1.0	110.0	50.0	1.034	110.0	0.091	0.0
1.0	110.0	70.0	1.06	110.0	0.042	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
1.0	110.0	90.0	1.185	110.0	0.184	0.0
1.0	110.0	110.0	1.171	110.0	0.194	0.0
1.0	110.0	130.0	1.014	110.0	0.035	0.0
1.0	110.0	150.0	1.081	110.0	0.059	0.0
1.0	160.0	30.0	1.017	160.0	0.16	0.0
1.0	160.0	50.0	1.014	160.0	0.087	0.0
1.0	160.0	70.0	1.049	159.8	0.222	0.4
1.0	160.0	90.0	1.066	159.8	0.189	0.4
1.0	160.0	110.0	1.005	160.0	0.058	0.0
1.0	160.0	130.0	1.262	160.0	0.223	0.0
1.0	160.0	150.0	1.101	160.0	0.109	0.0
1.0	210.0	30.0	0.994	210.0	0.182	0.0
1.0	210.0	50.0	1.169	210.0	0.248	0.0
1.0	210.0	70.0	1.129	209.6	0.292	0.8
1.0	210.0	90.0	1.031	209.8	0.088	0.4
1.0	210.0	110.0	1.062	209.8	0.097	0.4
1.0	210.0	130.0	1.017	210.0	0.059	0.0
1.0	210.0	150.0	1.008	210.0	0.082	0.0
1.0	260.0	30.0	1.353	255.8	0.412	5.879
1.0	260.0	50.0	1.086	260.0	0.162	0.0
1.0	260.0	70.0	1.067	259.8	0.157	0.4
1.0	260.0	90.0	1.019	260.0	0.052	0.0
1.0	260.0	110.0	1.241	259.2	0.202	1.6
1.0	260.0	130.0	1.033	260.0	0.021	0.0
1.0	260.0	150.0	1.036	260.0	0.052	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
2.0	10.0	30.0	1.54	20.0	0.094	0.0
2.0	10.0	50.0	1.445	20.0	0.066	0.0
2.0	10.0	70.0	1.595	20.0	0.135	0.0
2.0	10.0	90.0	1.535	20.0	0.163	0.0
2.0	10.0	110.0	1.695	20.0	0.393	0.0
2.0	10.0	130.0	1.55	20.0	0.154	0.0
2.0	10.0	150.0	1.56	20.0	0.143	0.0
2.0	60.0	30.0	1.042	120.0	0.034	0.0
2.0	60.0	50.0	1.147	120.0	0.156	0.0
2.0	60.0	70.0	1.091	120.0	0.099	0.0
2.0	60.0	90.0	1.093	120.0	0.093	0.0
2.0	60.0	110.0	1.08	120.0	0.031	0.0
2.0	60.0	130.0	1.099	120.0	0.062	0.0
2.0	60.0	150.0	1.163	120.0	0.144	0.0
2.0	110.0	30.0	1.027	220.0	0.066	0.0
2.0	110.0	50.0	1.052	220.0	0.12	0.0
2.0	110.0	70.0	1.008	220.0	0.036	0.0
2.0	110.0	90.0	1.162	219.8	0.131	0.4
2.0	110.0	110.0	1.045	220.0	0.054	0.0
2.0	110.0	130.0	1.064	220.0	0.098	0.0
2.0	110.0	150.0	1.141	220.0	0.064	0.0
2.0	160.0	30.0	1.048	320.0	0.167	0.0
2.0	160.0	50.0	0.954	320.0	0.041	0.0
2.0	160.0	70.0	0.952	320.0	0.03	0.0
2.0	160.0	90.0	1.192	319.6	0.201	0.8

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
2.0	160.0	110.0	1.019	320.0	0.084	0.0
2.0	160.0	130.0	1.21	320.0	0.161	0.0
2.0	160.0	150.0	1.011	320.0	0.051	0.0
2.0	210.0	30.0	1.107	419.2	0.139	1.166
2.0	210.0	50.0	1.027	420.0	0.098	0.0
2.0	210.0	70.0	0.969	420.0	0.018	0.0
2.0	210.0	90.0	1.101	419.8	0.128	0.4
2.0	210.0	110.0	1.002	420.0	0.017	0.0
2.0	210.0	130.0	1.175	420.0	0.077	0.0
2.0	210.0	150.0	1.143	419.6	0.153	0.8
2.0	260.0	30.0	1.133	518.2	0.207	3.6
2.0	260.0	50.0	1.084	518.6	0.159	2.8
2.0	260.0	70.0	1.094	519.2	0.166	1.6
2.0	260.0	90.0	1.064	518.0	0.1	2.757
2.0	260.0	110.0	1.12	519.2	0.222	1.166
2.0	260.0	130.0	1.108	518.6	0.156	2.332
2.0	260.0	150.0	1.105	519.4	0.126	1.2
3.0	10.0	30.0	1.533	30.0	0.099	0.0
3.0	10.0	50.0	1.577	30.0	0.093	0.0
3.0	10.0	70.0	1.487	30.0	0.176	0.0
3.0	10.0	90.0	1.46	30.0	0.095	0.0
3.0	10.0	110.0	1.53	30.0	0.039	0.0
3.0	10.0	130.0	1.61	30.0	0.175	0.0
3.0	10.0	150.0	1.45	30.0	0.048	0.0
3.0	60.0	30.0	1.045	180.0	0.047	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
3.0	60.0	50.0	1.124	180.0	0.157	0.0
3.0	60.0	70.0	1.063	180.0	0.055	0.0
3.0	60.0	90.0	1.138	180.0	0.149	0.0
3.0	60.0	110.0	1.106	180.0	0.02	0.0
3.0	60.0	130.0	1.087	180.0	0.072	0.0
3.0	60.0	150.0	1.065	180.0	0.016	0.0
3.0	110.0	30.0	1.012	330.0	0.067	0.0
3.0	110.0	50.0	1.034	330.0	0.06	0.0
3.0	110.0	70.0	1.006	330.0	0.026	0.0
3.0	110.0	90.0	1.04	330.0	0.095	0.0
3.0	110.0	110.0	1.116	329.8	0.139	0.4
3.0	110.0	130.0	1.113	330.0	0.073	0.0
3.0	110.0	150.0	1.071	330.0	0.055	0.0
3.0	160.0	30.0	0.997	480.0	0.056	0.0
3.0	160.0	50.0	0.927	480.0	0.018	0.0
3.0	160.0	70.0	1.077	480.0	0.192	0.0
3.0	160.0	90.0	1.085	480.0	0.117	0.0
3.0	160.0	110.0	1.008	480.0	0.058	0.0
3.0	160.0	130.0	1.073	480.0	0.091	0.0
3.0	160.0	150.0	1.075	480.0	0.104	0.0
3.0	210.0	30.0	0.986	630.0	0.072	0.0
3.0	210.0	50.0	1.032	629.8	0.082	0.4
3.0	210.0	70.0	1.012	630.0	0.052	0.0
3.0	210.0	90.0	1.055	628.2	0.156	3.124
3.0	210.0	110.0	0.985	630.0	0.037	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
3.0	210.0	130.0	1.101	628.8	0.16	2.4
3.0	210.0	150.0	1.033	629.8	0.097	0.4
3.0	260.0	30.0	1.1	777.0	0.039	2.53
3.0	260.0	50.0	1.043	779.4	0.068	0.8
3.0	260.0	70.0	1.05	779.6	0.059	0.49
3.0	260.0	90.0	1.117	779.6	0.1	0.49
3.0	260.0	110.0	1.034	779.2	0.073	1.6
3.0	260.0	130.0	1.053	778.8	0.055	1.6
3.0	260.0	150.0	1.069	779.6	0.111	0.8
4.0	10.0	30.0	1.45	40.0	0.128	0.0
4.0	10.0	50.0	1.52	40.0	0.044	0.0
4.0	10.0	70.0	1.603	40.0	0.149	0.0
4.0	10.0	90.0	1.52	40.0	0.074	0.0
4.0	10.0	110.0	1.6	40.0	0.163	0.0
4.0	10.0	130.0	1.52	40.0	0.029	0.0
4.0	10.0	150.0	1.512	40.0	0.064	0.0
4.0	60.0	30.0	1.119	240.0	0.056	0.0
4.0	60.0	50.0	1.112	240.0	0.063	0.0
4.0	60.0	70.0	1.101	240.0	0.135	0.0
4.0	60.0	90.0	1.125	240.0	0.054	0.0
4.0	60.0	110.0	1.09	240.0	0.059	0.0
4.0	60.0	130.0	1.074	240.0	0.044	0.0
4.0	60.0	150.0	1.184	240.0	0.151	0.0
4.0	110.0	30.0	1.0	440.0	0.033	0.0
4.0	110.0	50.0	1.086	440.0	0.059	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
4.0	110.0	70.0	1.055	440.0	0.038	0.0
4.0	110.0	90.0	1.031	440.0	0.032	0.0
4.0	110.0	110.0	1.028	440.0	0.058	0.0
4.0	110.0	130.0	1.093	440.0	0.029	0.0
4.0	110.0	150.0	1.054	440.0	0.039	0.0
4.0	160.0	30.0	0.99	640.0	0.094	0.0
4.0	160.0	50.0	1.065	639.4	0.1	1.2
4.0	160.0	70.0	0.999	640.0	0.076	0.0
4.0	160.0	90.0	0.98	640.0	0.053	0.0
4.0	160.0	110.0	1.05	639.8	0.11	0.4
4.0	160.0	130.0	1.047	639.8	0.087	0.4
4.0	160.0	150.0	1.048	640.0	0.06	0.0
4.0	210.0	30.0	1.005	839.6	0.07	0.8
4.0	210.0	50.0	1.06	838.4	0.075	2.059
4.0	210.0	70.0	1.015	839.4	0.068	0.8
4.0	210.0	90.0	1.015	839.4	0.06	1.2
4.0	210.0	110.0	1.069	840.0	0.121	0.0
4.0	210.0	130.0	1.058	839.2	0.085	0.748
4.0	210.0	150.0	1.04	840.0	0.037	0.0
4.0	260.0	30.0	1.044	1038.6	0.124	1.744
4.0	260.0	50.0	1.078	1036.8	0.078	3.059
4.0	260.0	70.0	1.107	1036.6	0.05	1.744
4.0	260.0	90.0	1.112	1038.4	0.052	3.2
4.0	260.0	110.0	1.115	1038.2	0.092	3.6
4.0	260.0	130.0	1.157	1037.0	0.076	4.29

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
4.0	260.0	150.0	1.085	1039.0	0.068	1.549
5.0	10.0	30.0	1.652	50.0	0.173	0.0
5.0	10.0	50.0	1.656	50.0	0.202	0.0
5.0	10.0	70.0	1.46	50.0	0.082	0.0
5.0	10.0	90.0	1.542	50.0	0.087	0.0
5.0	10.0	110.0	1.51	50.0	0.057	0.0
5.0	10.0	130.0	1.506	50.0	0.05	0.0
5.0	10.0	150.0	1.526	50.0	0.142	0.0
5.0	60.0	30.0	1.109	300.0	0.103	0.0
5.0	60.0	50.0	1.054	300.0	0.051	0.0
5.0	60.0	70.0	1.09	299.8	0.046	0.4
5.0	60.0	90.0	1.079	300.0	0.034	0.0
5.0	60.0	110.0	1.1	300.0	0.073	0.0
5.0	60.0	130.0	1.098	299.8	0.05	0.4
5.0	60.0	150.0	1.169	300.0	0.124	0.0
5.0	110.0	30.0	1.003	550.0	0.108	0.0
5.0	110.0	50.0	1.042	549.6	0.093	0.49
5.0	110.0	70.0	1.046	550.0	0.033	0.0
5.0	110.0	90.0	1.031	550.0	0.056	0.0
5.0	110.0	110.0	1.13	549.4	0.111	1.2
5.0	110.0	130.0	1.058	550.0	0.045	0.0
5.0	110.0	150.0	1.091	550.0	0.075	0.0
5.0	160.0	30.0	0.989	799.8	0.064	0.4
5.0	160.0	50.0	0.992	800.0	0.04	0.0
5.0	160.0	70.0	0.999	800.0	0.078	0.0

Duration (s)	Rate (1/s)	Length (bytes)	Average Latency (ms)	Average Received Packets	Standard Deviation Latency (ms)	Standard Deviation Received Packets
5.0	160.0	90.0	1.015	800.0	0.047	0.0
5.0	160.0	110.0	1.032	800.0	0.036	0.0
5.0	160.0	130.0	1.053	800.0	0.059	0.0
5.0	160.0	150.0	1.062	800.0	0.065	0.0
5.0	210.0	30.0	1.029	1047.8	0.074	2.857
5.0	210.0	50.0	1.071	1049.6	0.083	0.49
5.0	210.0	70.0	1.082	1049.2	0.076	0.748
5.0	210.0	90.0	1.1	1048.6	0.057	0.49
5.0	210.0	110.0	1.038	1049.4	0.079	0.8
5.0	210.0	130.0	1.089	1049.2	0.084	0.4
5.0	210.0	150.0	1.043	1049.6	0.074	0.49
5.0	260.0	30.0	1.079	1296.4	0.089	4.03
5.0	260.0	50.0	1.079	1300.0	0.151	0.0
5.0	260.0	70.0	1.093	1298.0	0.044	2.098
5.0	260.0	90.0	1.201	1296.4	0.168	3.72
5.0	260.0	110.0	1.08	1297.6	0.043	3.2
5.0	260.0	130.0	1.09	1298.2	0.09	2.4
5.0	260.0	150.0	1.047	1299.8	0.044	0.4

Data Analysis

Wireless tests are performed with 2 ESP8266 modules located on a 50 cm ruler which is placed inside of an aluminum hollow tube. Those ESP modules are programmed to send and receive packets using ESP-NOW with parameters written in the serial port via the computer.

First, ESP modules are placed 30 cm apart. Then, a Python script which initiates the transmission of the data is started. This code writes the test parameters on the serial port with pyserial module. One ESP reads the serial and sends ESP-NOW packets to the other ESP. The second ESP echos the timestamp part of the received data, so the latency can be measured with round trip time in the first ESP module. Also, when it receives all data, it writes how many packets it received on another serial port.

Error in packet reception and the latency of the transmission are tested with different packet rate and packet sizes. Also, the time duration of the data transmission is changed. These test are repeated for 36 cm and 42 cm distances. The result are plotted in **Figure 62, Figure 63, Figure 64, Figure 65, Figure 66, Figure 67, Figure 68, Figure 69, Figure 70, Figure 71** respectively.

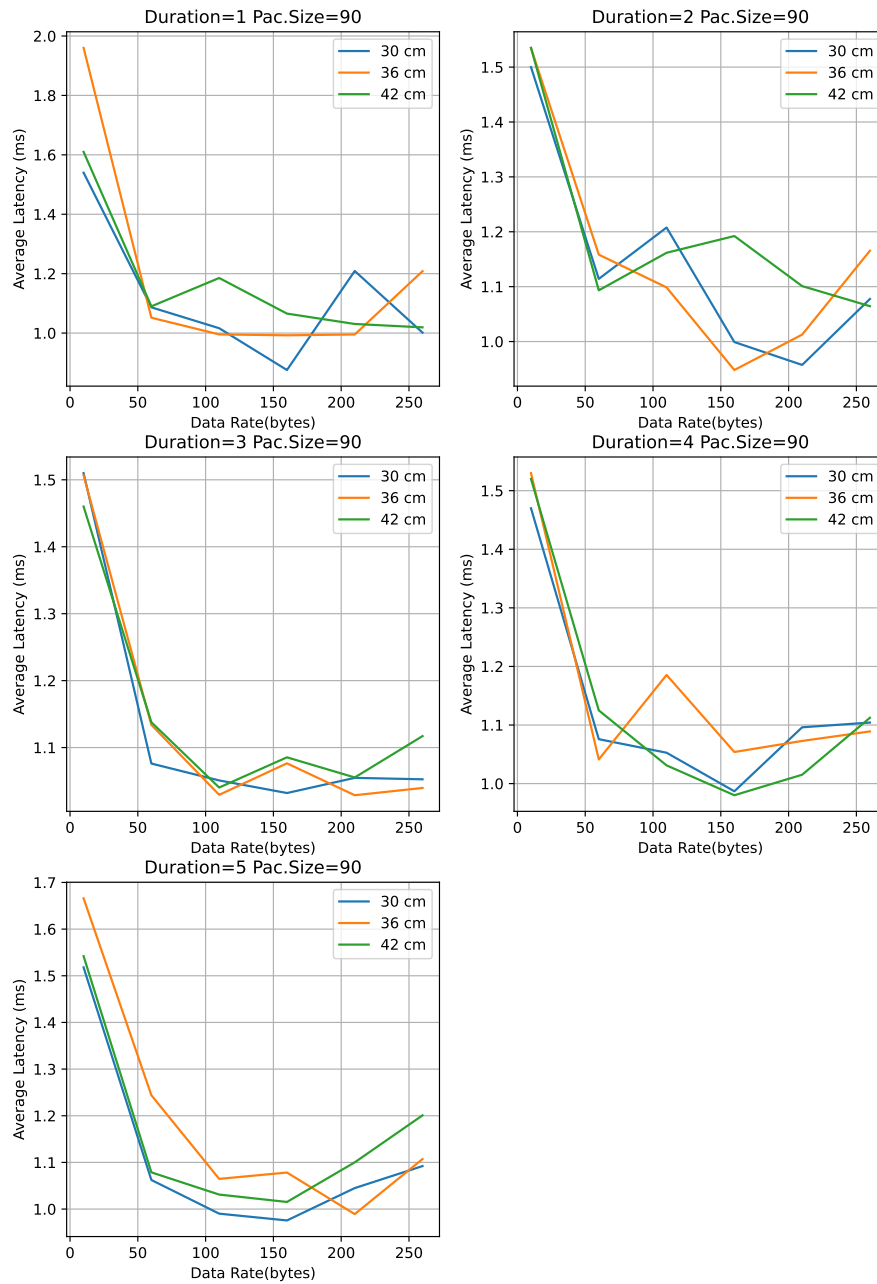


Figure 62: Graph of Data Rate vs Average Latency with same Packet Size and Different Durations

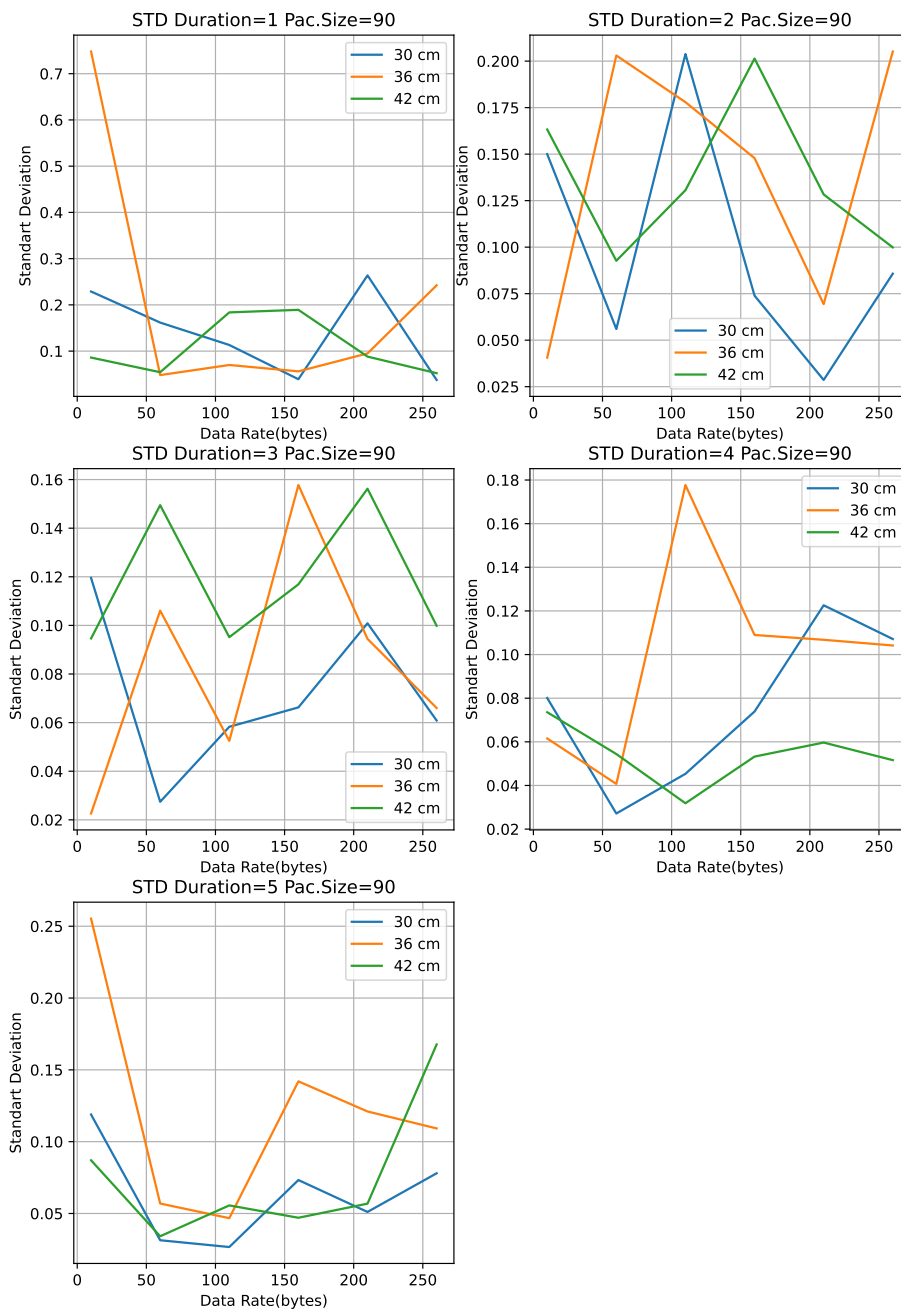


Figure 63: Standart Deviation Graph of Data Rate vs Average Latency with same Packet Size and Different Durations

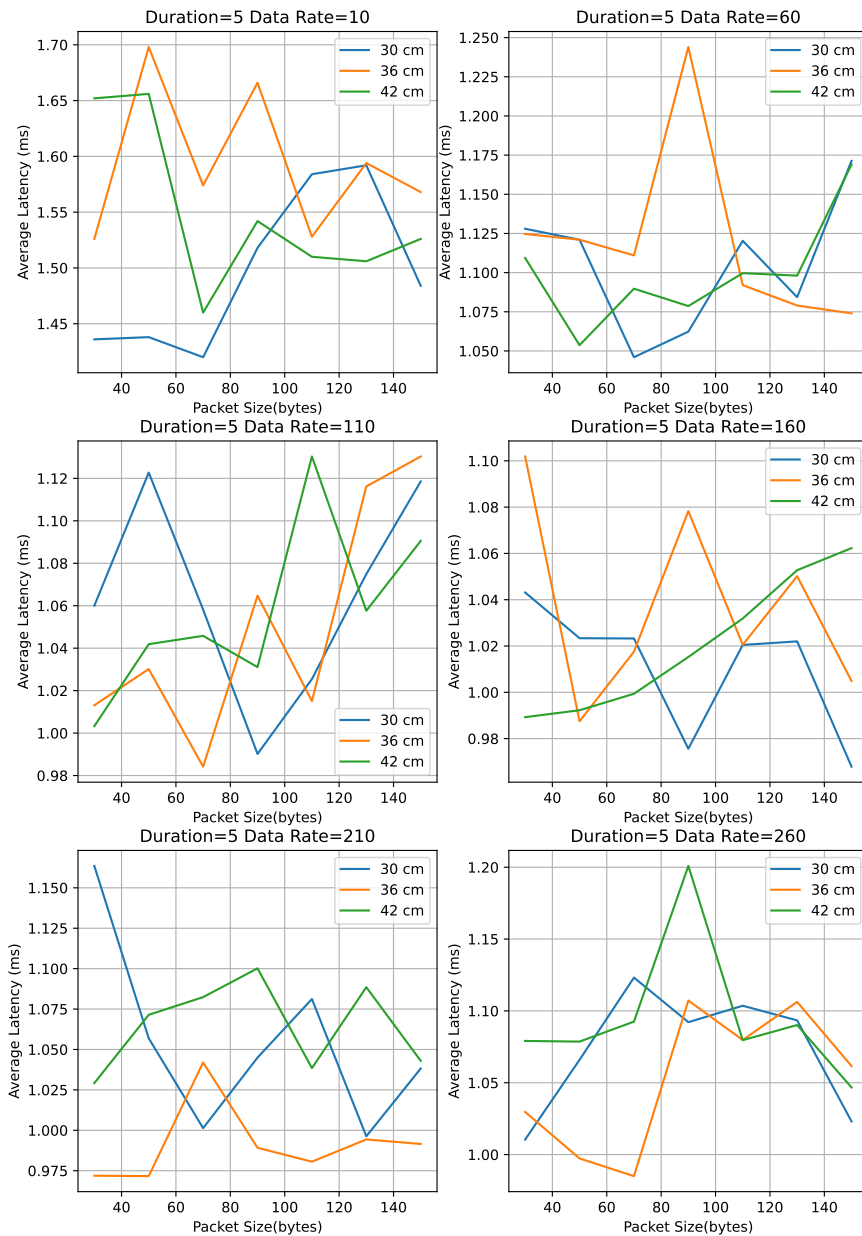


Figure 64: Packet Size vs Average Latency with Same Duration and Changing Data Rates

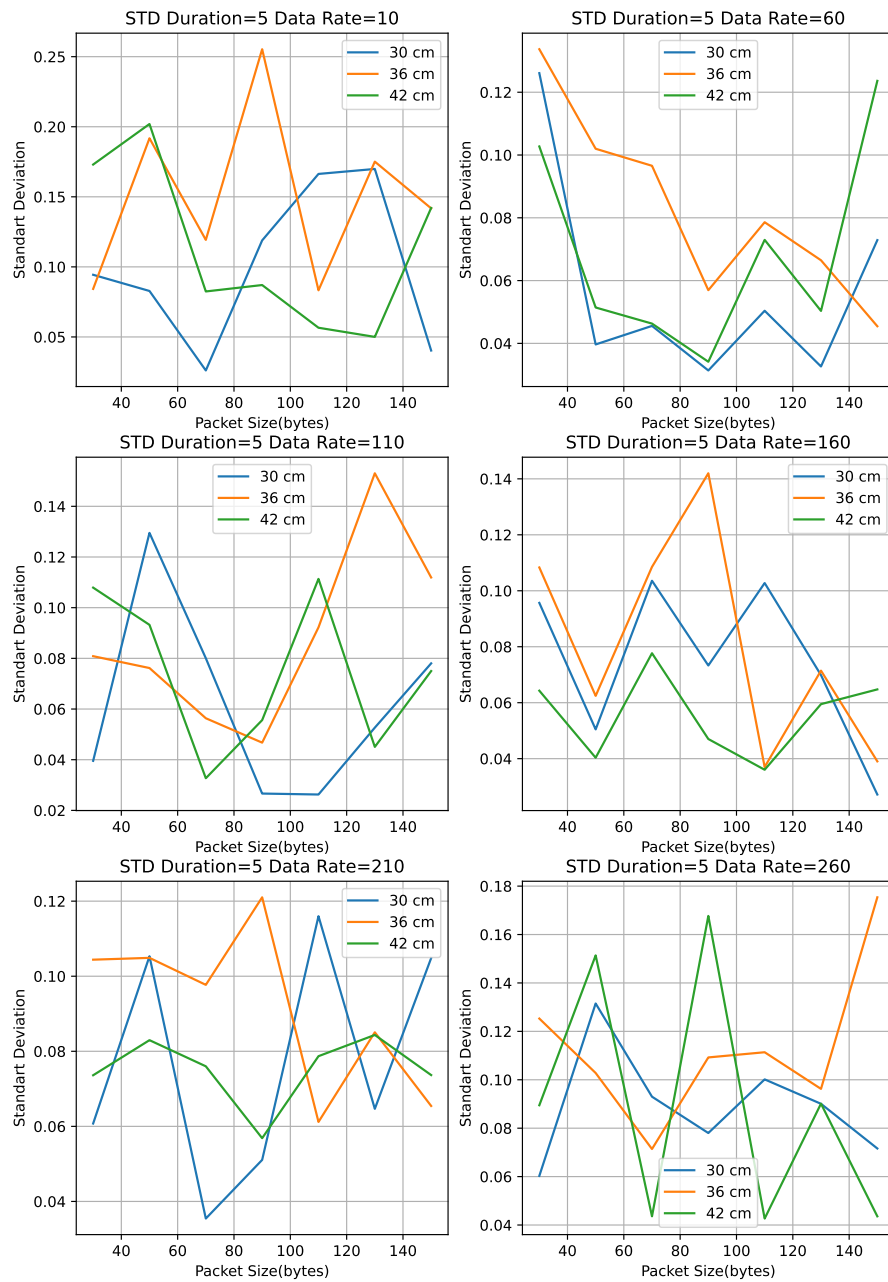


Figure 65: Standart Deviation Graph of Packet Size vs Average Latency with Same Duration and Changing Data Rates

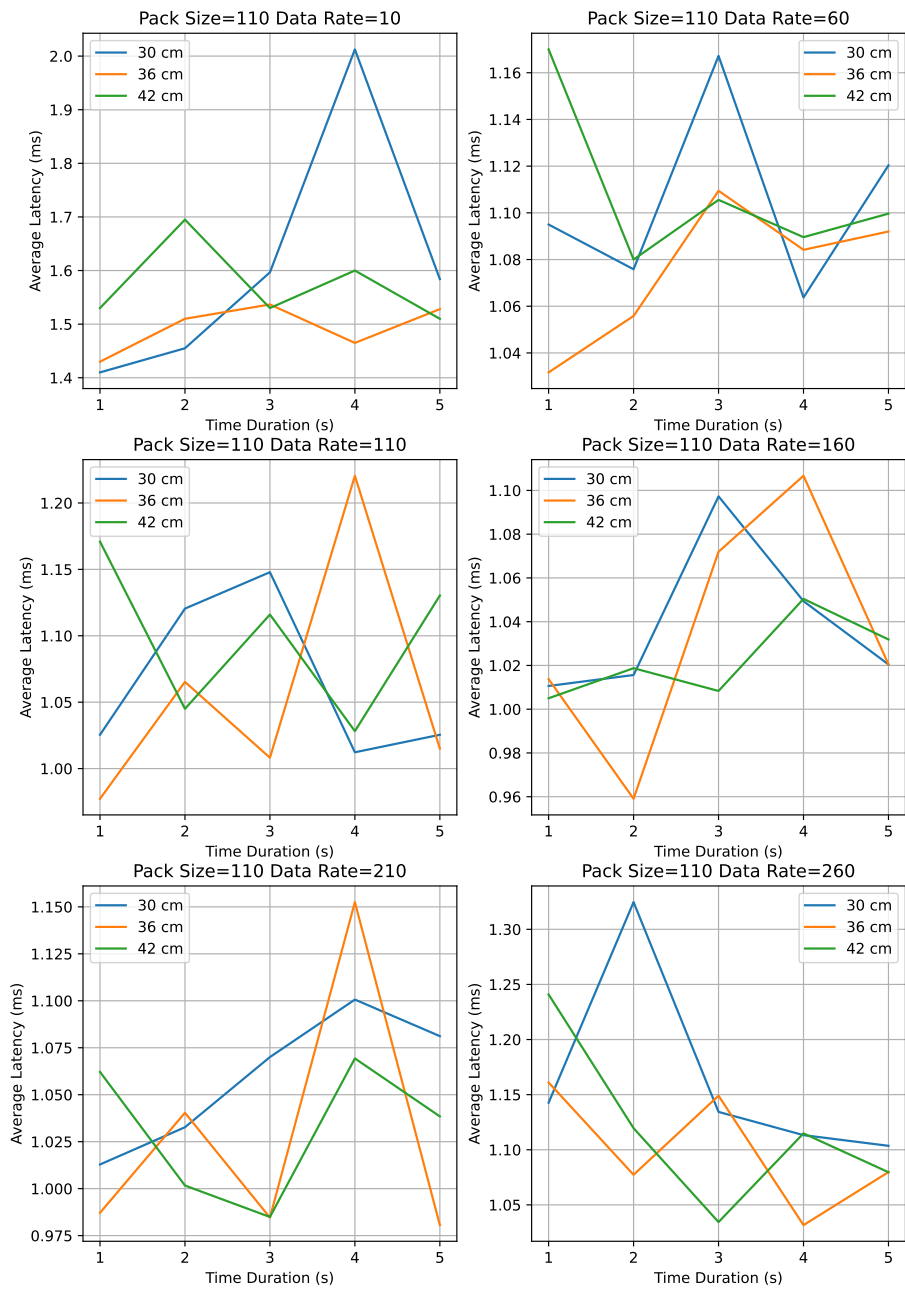


Figure 66: Time Duration vs Average Latency with Same Packet Size and Changing Data Rate

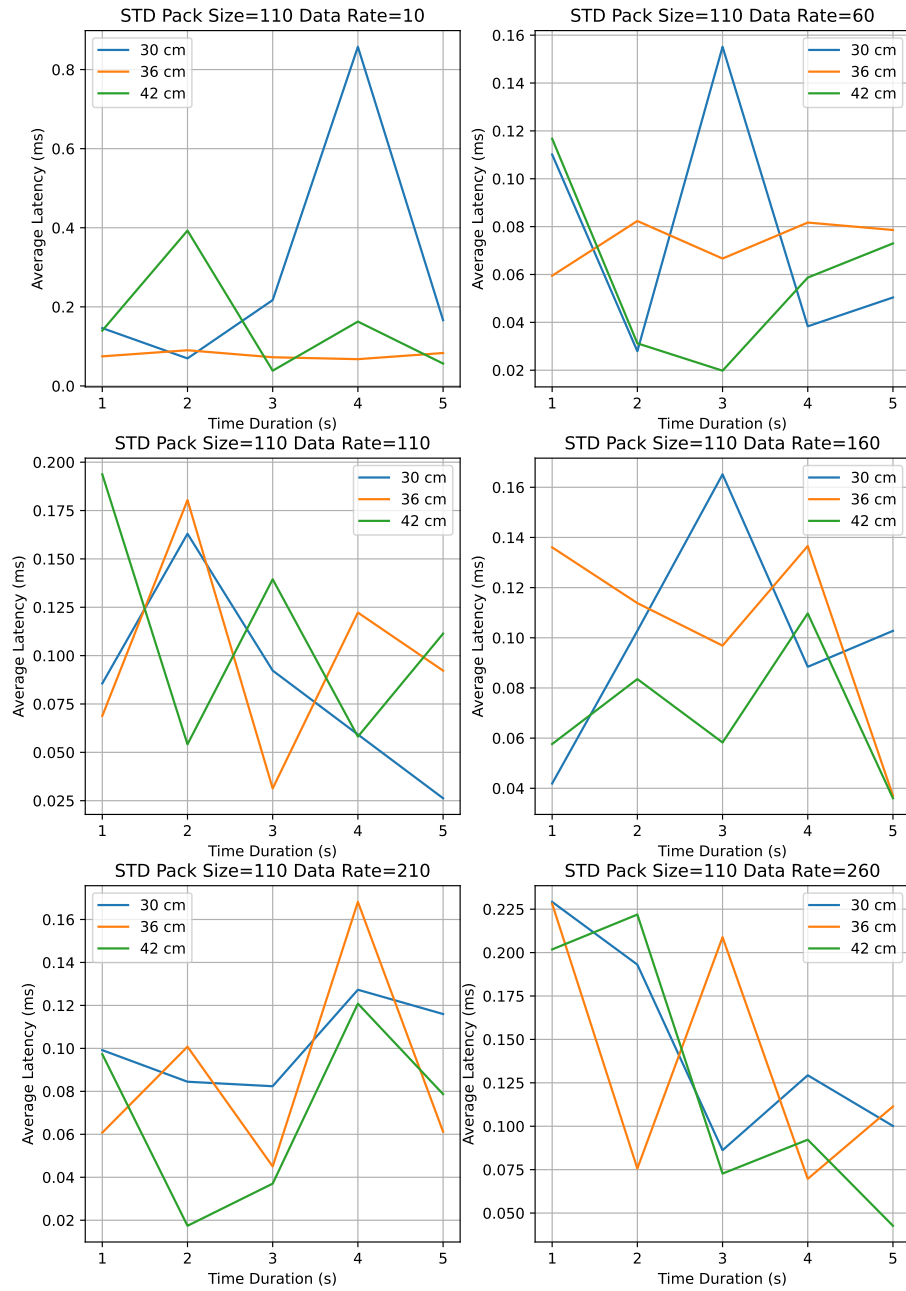


Figure 67: Standart Deviation Graph of Time Duration vs Average Latency with Same Packet Size and Changing Data Rate

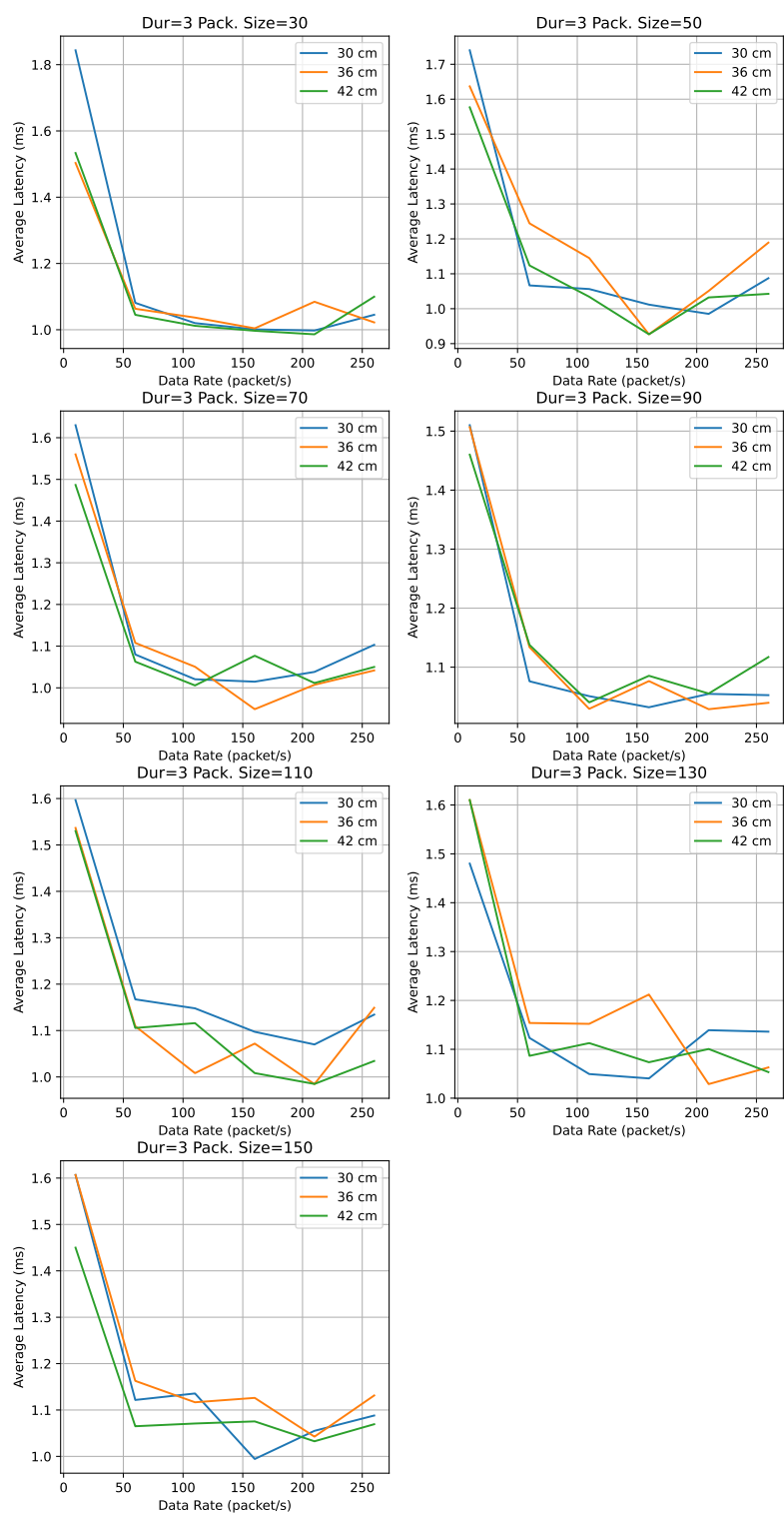


Figure 68: Data Rate vs Average Latency with Duration is Fixed and Size is Changing

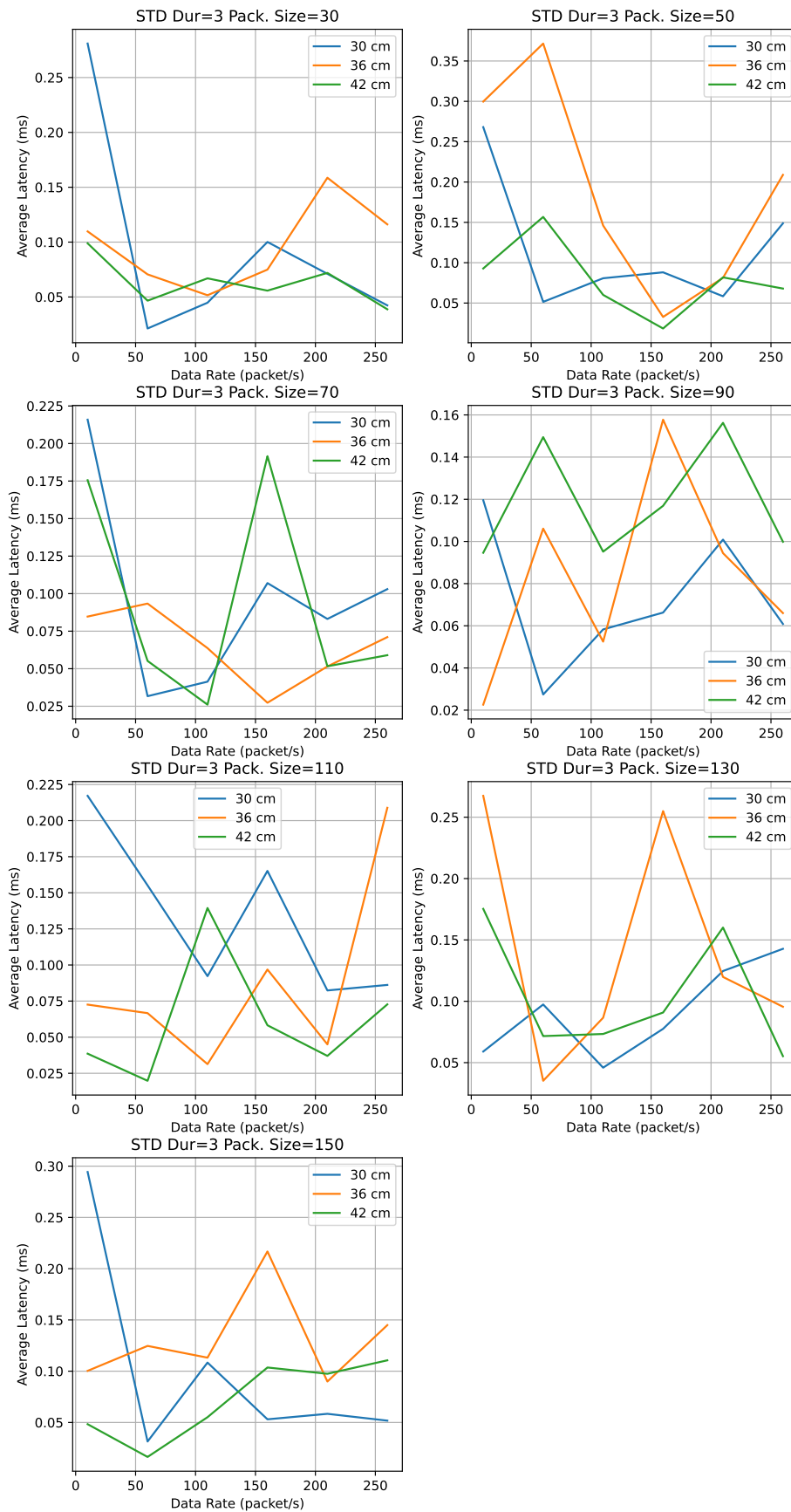


Figure 69: Standart Deviation Graph of Data Rate vs Average Latency with Duration is Fixed and Size is Changing

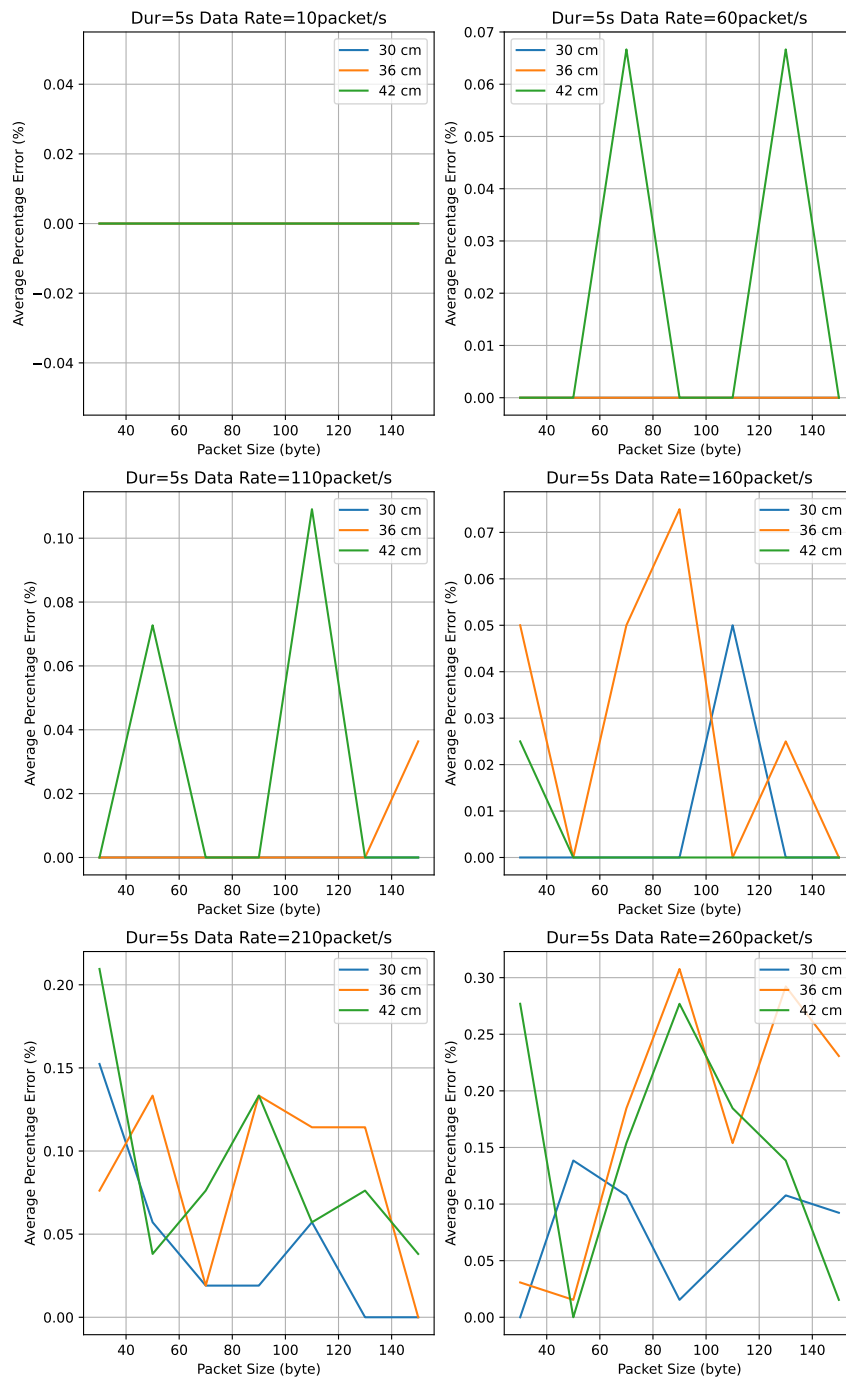


Figure 70: Percentage for Packet Size with Different Data Rates and Same Duration

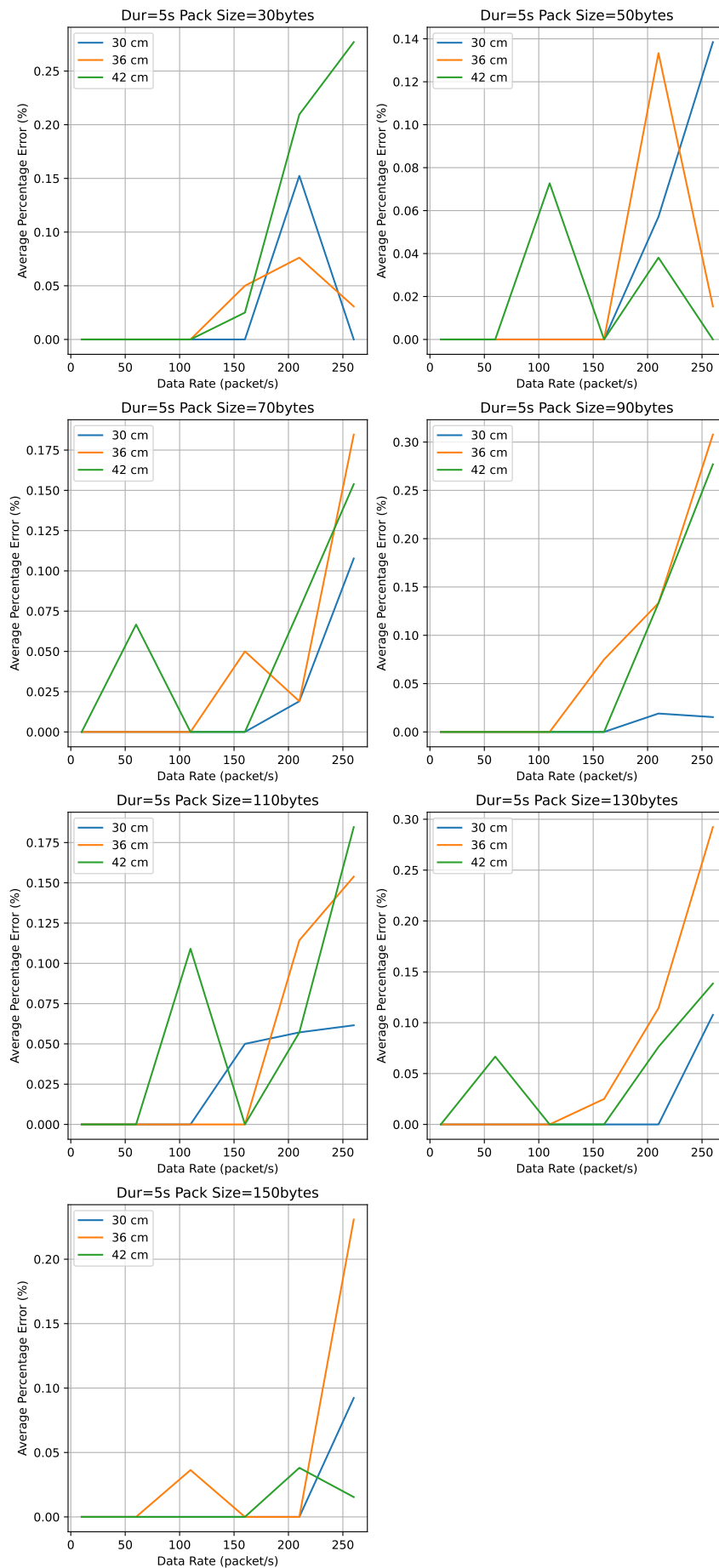


Figure 71: Percentage Error for Data Rate at same duration with Changing Package Size

Appendix 4: Test Document: Camera Vibration & Exposure

Location	Electrical-Electronics Department Middle East Technical University
Date	March 16 2024
Time	15:00
Description	Camera vibration and exposure time check on the vehicle
Aim	Observing the effect of vehicle movement on the camera output
Expected Outcome	Minor distortion and blurring on the camera output
Participants	BeyondTech Members and Studio Coordinators

Test Devices and Tools

Required Test Equipment: The Metal Chassis, 3x3m Area, The Raspberry PiCam3, The Camera Grip and Related Connection Parts

1. Camera

Ground truth/Calibration: The camera should have a resolution of 1280x720 pixels, be capable of capturing images at a minimum frame rate of 50Hz, and have adjustable exposure settings to adapt to varying lighting conditions.

2. Camera Mount

Ground truth/Calibration: The camera mount should be stable enough to hold the camera securely, allow for easy adjustment of the camera's position and orientation, and be durable enough to withstand the vibrations and movements of the vehicle during the tests.

3. Data Logging Tools (Raspberry Pi 5 with SD Card)

Ground truth/Calibration: The SD card should have sufficient storage capacity to save all frames captured by the camera at a 50Hz frame rate, the Raspberry Pi 5 should have enough processing power to handle image capture and saving without significant delays, and the data logging system should be reliable, ensuring no frame loss or corruption and accurate timestamps.

Test Environment

- **Area:** The test will be conducted in a 3x3 meter confined area, which will serve as the test arena for the vehicle's movements and parking maneuvers.
- **Vehicle:** The Long Thin Hauler vehicle, equipped with the camera and camera mount, will be used for the tests. The vehicle is designed for forward motion only and will be required to park in a random spot within the test area.
- **Computer:** A computer will be used for processing the images captured by the camera, analyzing the data, and controlling the vehicle's movements if necessary.

- **Lighting:** The test will be conducted under standard lighting conditions to simulate a typical indoor environment. The lighting should be consistent and evenly distributed across the test area to ensure accurate camera exposure settings.
- **Temperature:** The test will be conducted at room temperature, ensuring that the environmental conditions do not adversely affect the performance of the vehicle or the camera.
- **Aruco Markers:** Aruco markers will be placed at the borders and corners of the test area to assist with the vehicle's localization and to evaluate the camera's ability to capture these markers accurately during motion.

Test Benches and Setup

- **Camera and Mount Setup:**
 - Mount the camera securely on the vehicle using the camera mount.
 - Adjust the camera's position and orientation to ensure it has a clear view of the entire test area and all Aruco markers.
 - Set the camera to its default exposure settings to start with a baseline for the tests.
- **Vehicle Preparation:**
 - Place the vehicle at the starting position within the 3x3 meter test area.
 - Ensure that the vehicle is charged and functioning properly before beginning the tests.
 - Check that the camera and Raspberry Pi 5 are properly connected and ready to capture and store images.
- **Lighting and Environmental Setup:**
 - Set up the standard lighting conditions in the test area to ensure consistent illumination.
 - Verify that the room temperature is stable and within the normal range for indoor environments.
- **Aruco Marker Placement:**
 - Place Aruco markers at the designated borders and corners of the test area for localization purposes.
 - Ensure that the markers are visible and unobstructed from the camera's perspective.
- **Data Logging and Image Capture Configuration:**
 - Set up the Raspberry Pi 5 to capture images from the camera at a 50Hz frame rate.
 - Configure the data logging system to save the captured frames to the SD card with accurate timestamps.
- **Test Execution:**
 - Start the vehicle's forward motion and observe its movement through the test area.
 - Monitor the camera's image capture and exposure adjustments in response to the vehicle's motion and changes in lighting.
 - Record the output images to the SD card on Raspberry Pi 5, mounted on the vehicle.
- **Data Analysis:**
 - After the test, analyze the captured images to assess the camera's exposure settings and the impact of vibrations on image quality.
 - Evaluate the effectiveness of the camera's exposure adjustments in maintaining clear and consistent images of the Aruco markers.

Test Parameters

Table 39: The control parameters

Parameter	Possible Selections	Number of Measurements
AeExposureMod	Long-Short-Normal	3
AeFlickerMode	FlickerOff-FlickerManual	2
AeEnable	True-False	2

1. **AeExposureMod (Auto Exposure Exposure Mode):** This parameter controls the exposure mode of the camera. Changing this parameter will affect how the camera adjusts its exposure in response to varying lighting conditions in the test area.
Expected Outcome: Adjusting the AeExposureMod should result in changes to the brightness and clarity of the captured images. For example, setting the exposure mode to 'night' might result in brighter images in low-light conditions, while 'sports' mode might result in faster shutter speeds to capture clearer images of the moving vehicle.
2. **AeEnable (Auto Exposure Enable):** This parameter enables or disables the auto exposure feature of the camera. When enabled, the camera automatically adjusts its exposure settings based on the lighting conditions.
Expected Outcome: Enabling auto exposure (AeEnable=1) should lead to the camera automatically adjusting its exposure to maintain consistent image quality. Disabling auto exposure (AeEnable=0) means the exposure settings will remain fixed, which could result in overexposed or underexposed images if the lighting conditions change.
3. **AeFlickerMode (Auto Exposure Flicker Mode):** This parameter is used to reduce flickering in the captured images caused by artificial lighting, such as fluorescent lights. It can be set to match the frequency of the lighting (e.g., 50Hz or 60Hz).
Expected Outcome: Setting the AeFlickerMode correctly should minimize flickering in the images, resulting in clearer and more stable visuals. If the flicker mode is not matched to the lighting frequency, flickering may be more pronounced in the captured images.

Test Procedure

1. **Set Up Test Environment:** Arrange the 3x3 meter test area with appropriate lighting and place the Aruco markers at designated positions. Mount the camera on the vehicle and ensure it is securely attached with the camera mount. Position the vehicle at the starting point within the test area.
2. **Configure Camera Settings:** Set the camera to its default settings as a baseline. Enable auto exposure (AeEnable) and set the auto exposure flicker mode (AeFlickerMode) to match the frequency of the room's lighting.
3. **Begin Data Logging:** Start the Raspberry Pi 5 to begin capturing images at a 50Hz frame rate and saving them to the SD card.
4. **Conduct Initial Test Run:** Activate the vehicle's forward motion and allow it to move through the test area. Observe the camera's performance in capturing images of the Aruco markers and the effects of any vibrations on image quality.
5. **Adjust AeExposureMod Settings:** Repeat the test with different AeExposureMod settings (e.g., 'Long', 'Short', 'Normal') to observe how changes in exposure mode impact image quality under the same lighting conditions.

6. **Disable Auto Exposure:** Disable the auto exposure feature ($AeEnable=0$) and manually set exposure settings. Conduct another test run to observe the effects of fixed exposure settings on image quality, especially in varying lighting conditions.
7. **Analyze Captured Images:** Review the images captured during each test run, focusing on the clarity of the Aruco markers and the presence of any motion blur or flickering. Assess how well the camera maintained consistent exposure and image quality across different settings and conditions.

Test Data

Figure 73 illustrates the result of setting the camera to a short exposure time during the testing phase. The captured image presents the Aruco markers with well-defined edges and a high level of detail, evidencing the reduced motion blur associated with shorter exposure times. This condition is ideal for the localization algorithm, which relies on the crispness of these markers to accurately determine the vehicle's position. The effectiveness of the short exposure setting in minimizing blur while maintaining marker visibility is clearly demonstrated.

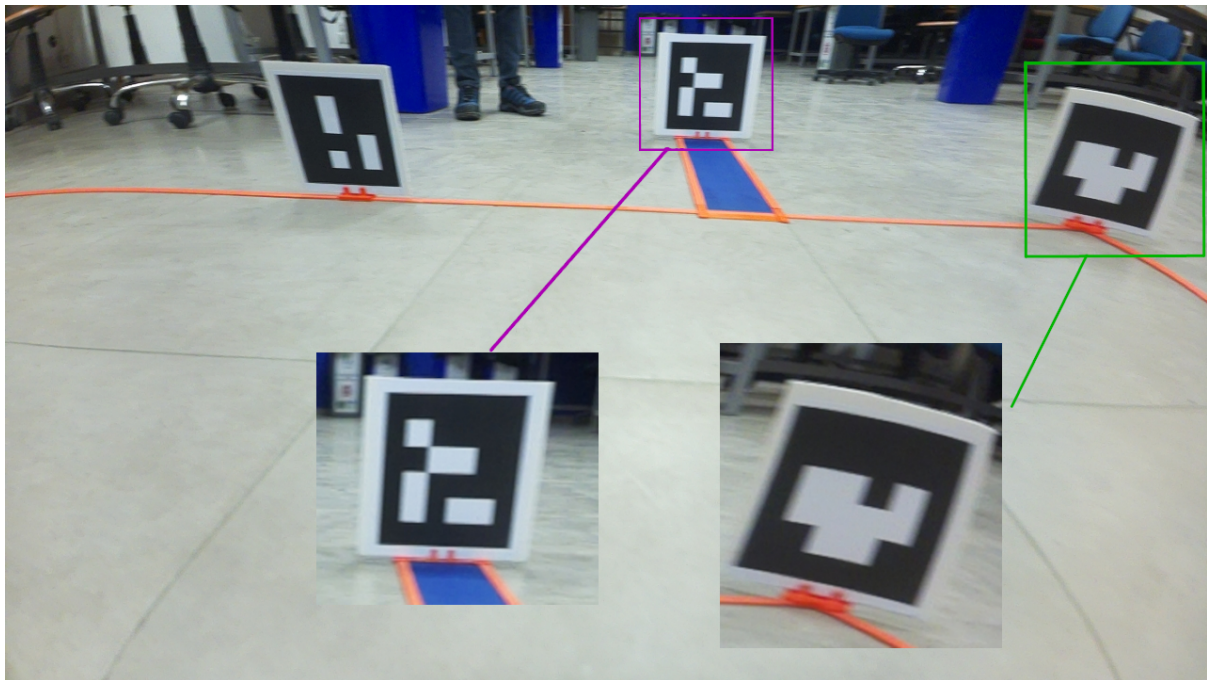


Figure 72: Blur When Exposure is Set to Long

As shown in the accompanying **Figure 72**, the long exposure setting resulted in a noticeable blur in the image of the Aruco markers, highlighting the impact of extended exposure times. This effect of motion blur, more evident in the elongated edges and corners, is a consequence of the camera's increased sensitivity to vehicle movement over the prolonged exposure period. Although this setting allows more ambient light to be captured, which can be advantageous in low-light environments, it also poses challenges for the localization algorithm due to the reduced image clarity.

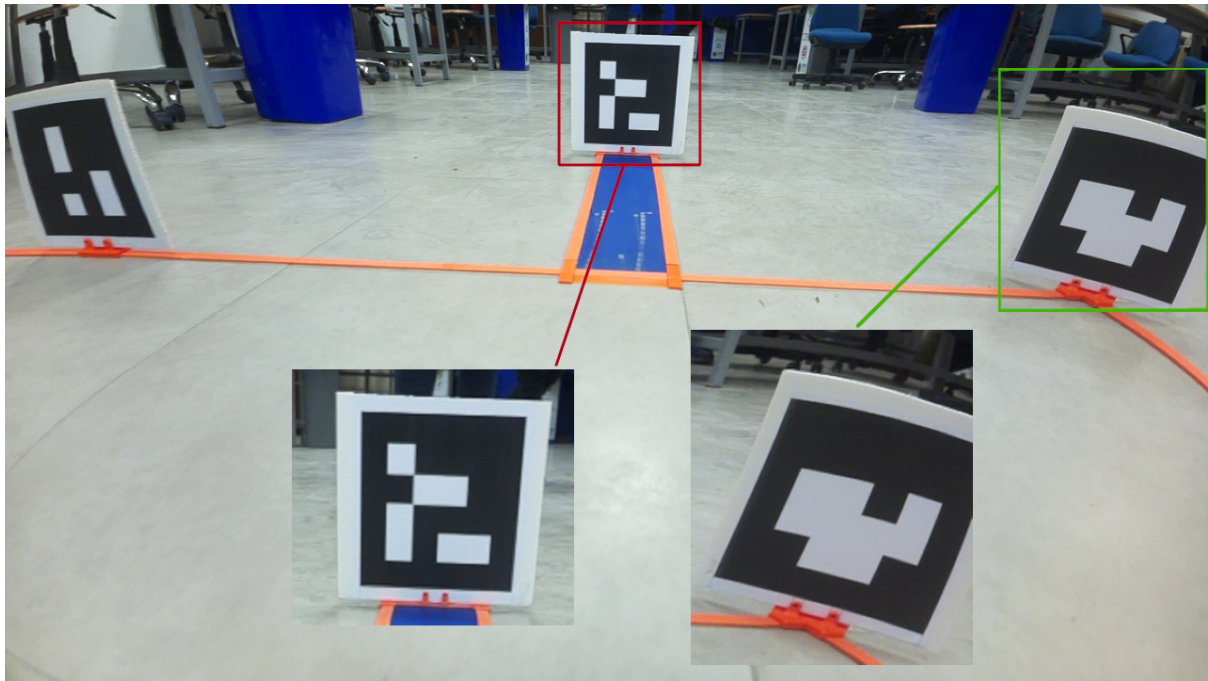


Figure 73: Blur When Exposure is Set to Short

These images serve as test data, showcasing the direct impact of exposure settings on the image quality captured by the camera. The comparison provides insight into how each setting may influence the vehicle's ability to accurately localize itself using visual markers under varying conditions.

Data Analysis

- Exposure Time:** Short exposure times were found to be effective in reducing motion blur in the images, as they limit the amount of light captured by the camera sensor, thus minimizing the blurring effect caused by the vehicle's movement. However, the experiment also highlighted that mechanical vibrations can impact image clarity, independent of the exposure time settings. These vibrations can cause minute movements of the camera lens, leading to a loss of focus and resulting in blurred images. To address this issue, it is recommended to further stabilize the camera mount to minimize the impact of mechanical vibrations on the lens. This could involve using more rigid mounting materials or adding vibration-damping elements to the camera mount.

The tradeoff in Low Light Conditions: While short exposure times improve image clarity by reducing motion blur, they can lead to underexposed images in low light conditions. In such environments, the markers may not be sufficiently illuminated for accurate detection by the localization algorithm. Therefore, it is important to balance the exposure settings with the lighting conditions of the operating environment. In darker settings, adjusting the exposure time to allow more light may be necessary, while also considering measures to mitigate the effects of mechanical vibrations on image clarity.
- Lens Vibration Issue:** In the experiment, even when the exposure time was set to short, there were still instances of blurry images. To further investigate this issue, the autofocus feature was tested in both enabled and disabled states. It was observed that disabling autofocus resulted in better image quality, with reduced blurring compared to when autofocus was enabled. However, despite disabling autofocus, some blurring persisted in the images. Further investigations revealed that the remaining blur was caused by the movement of the lens coil due to mechanical vibrations. These

vibrations led to slight shifts in the lens position, affecting the focus and resulting in blurred images.

Solution - Immobilizing the Camera Lens: To counteract the motion blur caused by lens vibrations, it was decided to immobilize the camera lens. By fixing the focus of the lens, the camera is no longer susceptible to the fluctuations caused by the vehicle's movement and autofocus mechanism. With a static lens, the camera can maintain a consistent focus on the markers, even as the vehicle moves. This stability in focus is expected to result in sharper images, where the Aruco markers are more distinctly visible, thereby enhancing the reliability of the localization process.

Based on these findings, it is recommended to use a fixed camera lens and short exposure times for optimal image clarity during vehicle movement. However, it is important to consider the lighting conditions of the operating environment, as adjustments to the exposure settings may be necessary to ensure the vehicle can effectively localize in varying light conditions.

Appendix 5: Test Document: IMU Attitude

Location	Electrical-Electronics Department Middle East Technical University
Date	March 22 2024
Time	11:00
Description	This test sheet explains the procedure for testing the sensor subsystem.
Aim	The aim of this test is to verify the outputs of the sensor system.
Expected Outcome	Verification of the sensor system outputs.
Participants	BeyondTech Team Members and Studio Coordinators

Test Devices and Tools

Required Test Equipments: MPU9250 IMU, Breadboard, 4 jumper cable, 360° protractor, Microcontroller, Computer

1. Protractor The yaw angle is measured using a protractor.

Ground truth/Calibration: The protractor is justified as a ground truth measurement tool for angular measurements due to its expected accuracy, typically within one degree. The measurement points for angles are sparse so the protractor is precise enough to perform this type of test procedure.

2. IMU Data

Ground truth/Calibration: The ground truth measurement tool for angular measurements is the protractor, which is expected to have an accuracy of within one degree and is precise enough for this type of test procedure.

Test Environment

The testing environment consists of 360° protractor that resides on a flat surface with a microcontroller and a computer to collect the data from IMU.

Environmental Conditions:

1. Temperature Control: The environment is maintained at $25^{\circ}\text{C} \pm 10^{\circ}\text{C}$.
2. Lighting: Standard lighting, with adjustable intensity to simulate various time-of-day conditions.
3. Magnetic Field: No electromagnetic devices that would alter the magnetic field significantly in a non-uniform way.

Test Benches and Setup:

1. Microcontroller is a small IC that enables people to control electronic systems. During the tests, it will read IMU data from I2C interface, and send it to the computer by applying filter to obtain yaw value. Also, it will power the IMU.
2. The computer will be used to obtain the orientation data from the microcontroller.

3. MPU9250 (**Figure 74**) is a 9 Degree of Freedom (DoF) IMU that consists of 3 axes accelerometer, gyroscope, and magnetometer.

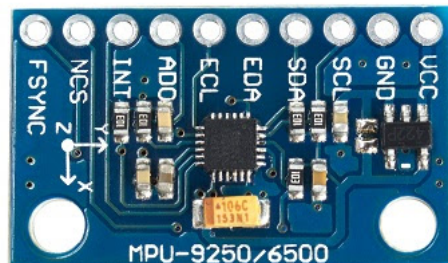


Figure 74: 9 DoF IMU - MPU9250

4. Breadboard (**Figure 75**) is a tool to enable people build prototype circuits. During the tests, it will be used to fix the IMU orientation.

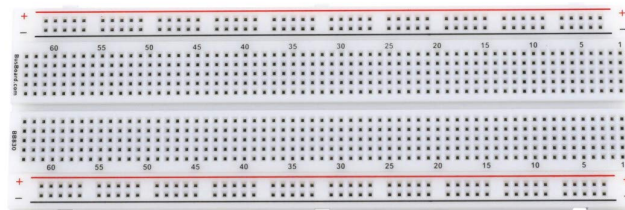


Figure 75: Breadboard

5. Jumper cables (**Figure 76**) are cables that enable people make connections fast and easy, especially in prototypes. During the tests, they will be used to power the IMU from microcontroller and transmit the IMU data to microcontroller with I2C interface.

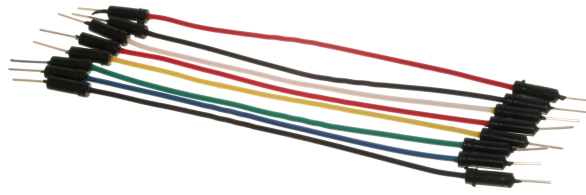


Figure 76: Jumper Cables

6. "Beyond Tech LongThin Software Suite" (**Figure 77**) includes a program that will be used to read the orientation data and visualize it in real-time.



Figure 77: IMU Data Visualization interface

Test Parameters

Table 40: The control parameters

Parameter	Range	Step Size	Number of Measurements
Yaw angle of the IMU(ψ)	[0° 315°]	45°	8

Test Procedure

1. IMU VCC and GND pins are connected to the 3.3V and GND pins of the microcontroller. The SCL and SDA pins of the IMU are connected to the SCL and SDA pins of the microcontroller.
2. The microcontroller is connected to the computer with a USB cable.
3. "longthin-imu" program is run on the computer.
4. Breadboard which has IMU on it placed on the protractor with a flat surface. The IMU direction is adjusted to show 0 degree.
5. The breadboard direction is changed by 30 degree steps and the IMU measurements are read.

Test Data

Table 41: Test Data at Steady State

Protractor Angle	Measured IMU Angle
0	102.3
45	151.8
90	-150.6
135	-100.4
180	-59.2
225	16.3
270	21.7
315	61.2
360	105.1

Table 42: Test Data Transient State

Protractor Angle	Measured IMU Angle
0	113.2
45	163.2
90	-148.1
135	-104.32
180	-60.9
225	-22.8
270	21.5
315	62.1
360	104.3

Data Analysis

The data is analyzed by looking at the change in the IMU angle at each protractor angle. The expected change was 45 degrees since the protractor was rotated by 45 degrees each

time. The data is compared with the expected change to determine the accuracy of the IMU. Two different cases are analyzed: steady state and transient state (**Figure 78**, **Figure 79** respectively).

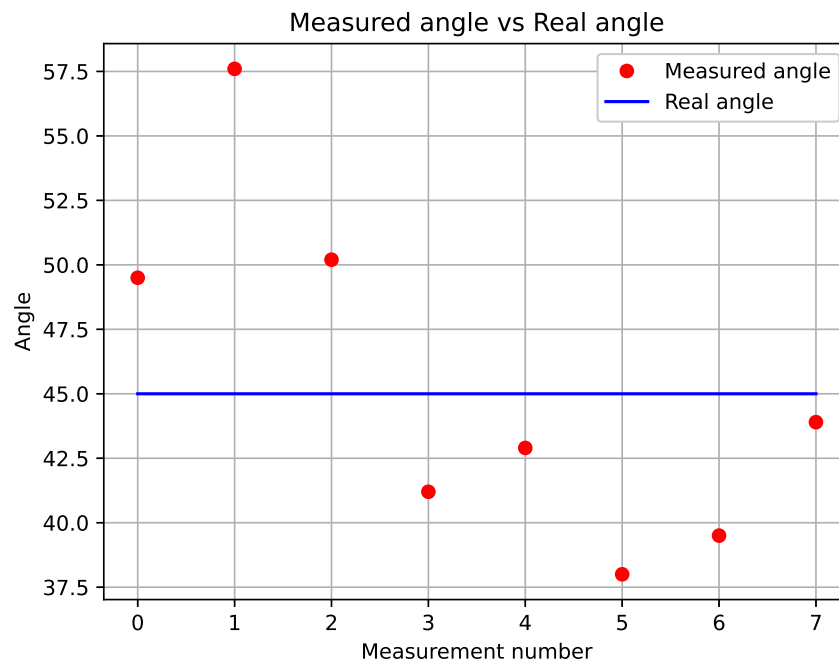


Figure 78: Changes in IMU angle at each protractor angle at steady state

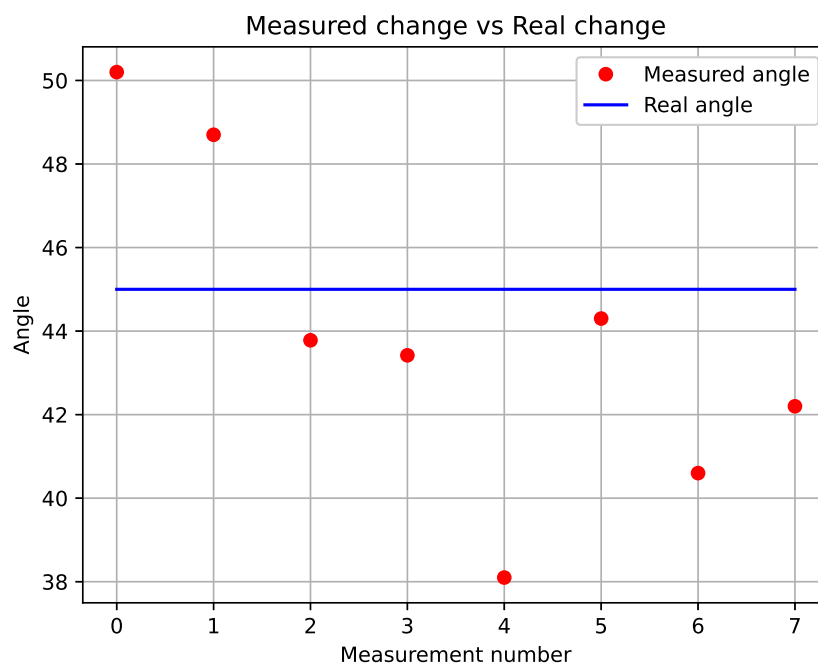


Figure 79: Changes in IMU angle at each protractor angle at transient state

It is seen that the non-uniformity of the magnetic field affects the IMU measurements. The IMU measurements are not linearly related to the protractor angle, and the worstly af-

affected measurements have 180 degrees which shows the magnetic field affect is maximum at this angle.

This affect is experienced less at the transient state, which is expected since the magnetic field affect is less and the gyroscope measurements are also effective at the transient state. On the other hand, the yaw angle converges to the magnetometer measurements at the steady state.

Appendix 6: Test Document: Path planning

Location	Electrical-Electronics Department Middle East Technical University
Date	March 28 2024
Time	15:30
Description	This test sheet explains the procedure for testing the path planning
Aim	The aim of this test is to test the path planning method.
Expected Outcome	Generated path is collision free and reaches the goal.
Participants	BeyondTech Team Members and Studio Coordinators

Test Devices and Tools

Required Test Equipment: The required test equipment includes the onboard computer Raspberry Pi 5, the Longthin Software Suite, an Ethernet cable or equivalent, a power adaptor, and a computer with SSH capabilities.

1. Path Generation Algorithm: The path generation algorithm is a software algorithm that generates a path for the robot to follow. It will be tested to ensure that it generates a feasible path for the robot, taking into account constraints such as the allowable drivable area and the minimum turning radii.

Ground Truth/Calibration: The ground truth for the path generation algorithm includes the starting and ending points of the path, as well as additional constraints such as the allowable drivable area and the minimum turning radii.

Test Environment

The test environment is a controlled software environment, conducted without external disturbances, and does not require a physical test environment as the main goal is to evaluate the performance of the path generation algorithm.

As for the software environment, the test will be conducted on the Raspberry Pi 5, which is the onboard computer of the robot. The software suite will be used to run the path generation algorithm. The test will be conducted via a computer with ssh capabilities.

The test procedure involves supplying necessary parameters, such as the starting and ending pose of the robot, which are provided as input to the path generation algorithm. The output of the algorithm is then evaluated by the software suite, taking into account constraints such as the allowable drivable area and the minimum turning radii.

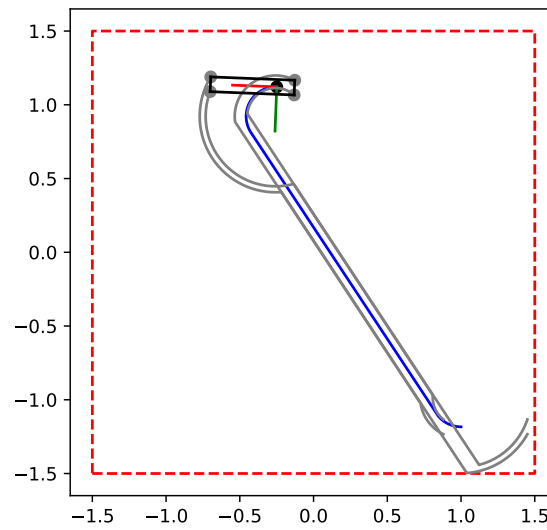


Figure 80: The output of the path generation algorithm

Test Parameters

Table 43: The control parameters

Parameter	Range	Step Size	Number of Measurements
Start point x	[-1.5m 1.5m]	0.1m	31
Start point y	[-1.5m 1.5m]	0.1m	31
End point x	[-1.5m 1.5m]	0.1m	31
End point y	[-1.5m 1.5m]	0.1m	31
Start angle	[-180° 180°]	10°	37
End angle	[-180° 180°]	10°	37
Min turning radius	[0.1m 1m]	0.1m	10

Test Procedure

1. Connect the Raspberry Pi 5 to the computer via an Ethernet cable.
2. Connect to the Raspberry Pi 5 via SSH.
3. Run the Longthin Software Suite.
4. Run "longthin-pathtest" command to test the path generation algorithm.
5. (Optional) If needed, set the parameters for the path generation algorithm using command line arguments.
6. Observe and evaluate the output of the path generation algorithm.

Test Results

After generating 100000 possible scenarios using the path generation algorithm. Every scenario generates 6 possible paths, resulting in a total of 600000 paths.

- The path generation algorithm successfully generated 600000 paths.
- The paths generated by the algorithm were feasible and took into account constraints such as the allowable drivable area and the minimum turning radii.
- Geometric constraints were also taken into account. They are verified by the numerical values by fully realizing the paths.
- The path generation algorithm was able to generate paths in a reasonable amount of time.
- The heatmap of the success rate versus the starting points is shown in Figure 81.
- The 4561 out of 100000 scenarios failed to generate a path. The failure rate is 4.56%. This failure rate is acceptable since there are many possible scenarios that are not feasible to satisfy the constraints.

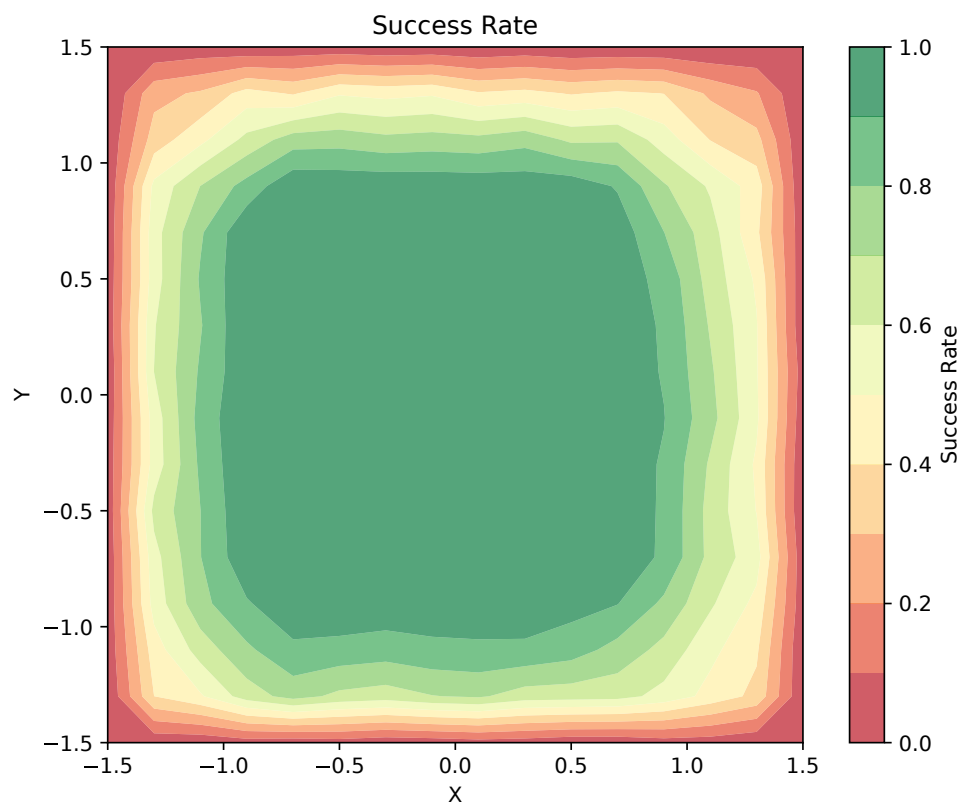


Figure 81: The heatmap of the success rate versus the starting points

Appendix 7: Test Document: Power Test at Full Power

Location	Electrical-Electronics Department Middle East Technical University
Date	March 15 2024
Time	15:00
Description	This test sheet explains the procedure for testing the battery drain time.
Aim	The aim of this test is to obtain the voltage characteristics of the battery.
Expected Outcome	Limits and the characteristics of the battery system.
Participants	BeyondTech Team Members and Studio Coordinators

Test Devices and Tools

Required Test Equipment: Vehicle's Batteries, Smart Li-Ion Battery Charger, Power Supply.

1. Maximum Discharge Rate: The highest rate at which the batteries can safely discharge without causing damage or significantly reducing their lifespan. This rate is crucial for determining the maximum power output the vehicle can achieve without compromising battery health.

Ground Truth/Calibration: The measurement of the battery capacity indicator is assumed to be accurate.

2. Voltage Range Under Load

Ground truth/Calibration: The acceptable voltage range for the batteries when operating at full power. This range is essential for ensuring that the vehicle's electrical systems receive a stable and sufficient power supply during high-demand situations.

Test Environment

- **Controlled Laboratory Setting:** The test is conducted in a lab environment where temperature and humidity are regulated to ensure consistent conditions.
- **Smart Charger Setup:** The smart charger is set to discharge the batteries to 3.15V with a constant drain of 6W, simulating full power operation of the vehicle.
- **Safety Equipment:** Proper safety equipment, such as fire extinguishers and protective gear, is available in case of battery overheating or other safety hazards.
- **Monitoring Equipment:** Voltage, current, and temperature sensors are set up to monitor the batteries' conditions in real-time during the test.
- **Data Logging System:** A data logging system is in place to record the performance parameters of the batteries and the vehicle at a rate of every 10 minutes during the full power test for subsequent analysis.

Test Benches and Setup:

- **Battery Preparation:** Ensure all batteries are fully charged and in good condition before the test. Label each battery for easy identification during and after the test.
- **Smart Charger Configuration:** Set up the smart charger to discharge the batteries at a constant drain of 6W, with a target voltage of 3.15V. Verify the charger's settings to ensure accurate simulation of full power operation.

- **Safety Equipment Setup:** Place fire extinguishers, protective gear, and other safety equipment near the test bench for quick access in case of emergency.
- **Monitoring Equipment Installation:** Install voltage, current, and temperature sensors on each battery and connect them to the data logging system. Ensure that the sensors are calibrated and functioning correctly.
- **Data Logging System Configuration:** Set up the data logging system to record battery performance parameters every 10 minutes. Confirm that the system is capturing data accurately and storing it securely for analysis.
- **Test Execution:** Begin the test by activating the smart charger to discharge the batteries. Monitor the batteries and the charger throughout the test to ensure they are operating as expected.
- **Test Initiation:** Begin the test by activating the smart charger to discharge the battery, simulating a failure scenario, while closely monitoring the BMS's response.
- **Post-Test Procedures:** After the test, safely disconnect the batteries from the charger and monitoring equipment. Recharge the batteries to full capacity and perform a visual inspection for any signs of damage or wear.

Test Parameters

Table 44: The control parameters

Parameter	Possible Range	Number of Measurements
Voltage	4.15-3.15	25

Voltage Measurement: The primary parameter measured during this test is the voltage of each battery. The voltage is monitored continuously as the batteries are discharged by the smart charger to simulate full power operation. The voltage readings are recorded every 10 minutes to track the batteries' performance under load.

Expected Outcome: The expected outcome for this test is that all batteries will maintain a voltage level close to their nominal value, with minimal drop, throughout the discharge process. A successful test will demonstrate the batteries' capability to sustain the required power output without significant voltage fluctuations, ensuring the reliability of the vehicle's power system during full power operation.

Test Procedure

1. **Prepare Batteries:** Ensure all batteries are fully charged and in good condition. Label each battery for easy identification.
2. **Set Up Smart Charger:** Configure the smart charger to discharge the batteries at a constant drain of 6W, targeting a voltage of 3.15V.
3. **Install Monitoring Equipment:** Attach voltage, current, and temperature sensors to each battery, connecting them to the data logging system.
4. **Configure Data Logging:** Set the data logging system to record battery performance parameters every 10 minutes.
5. **Initiate Test:** Start the test by activating the smart charger to discharge the batteries.
6. **Monitor Test:** Throughout the test, observe the batteries and charger to ensure they operate as expected.
7. **Conclude Test:** Once the test is complete, safely disconnect the batteries from the charger and monitoring equipment.

8. **Post-Test Analysis:** Recharge the batteries, inspect them for any signs of damage or wear, and analyze the recorded data to evaluate the batteries' performance under full power conditions.

Test Data

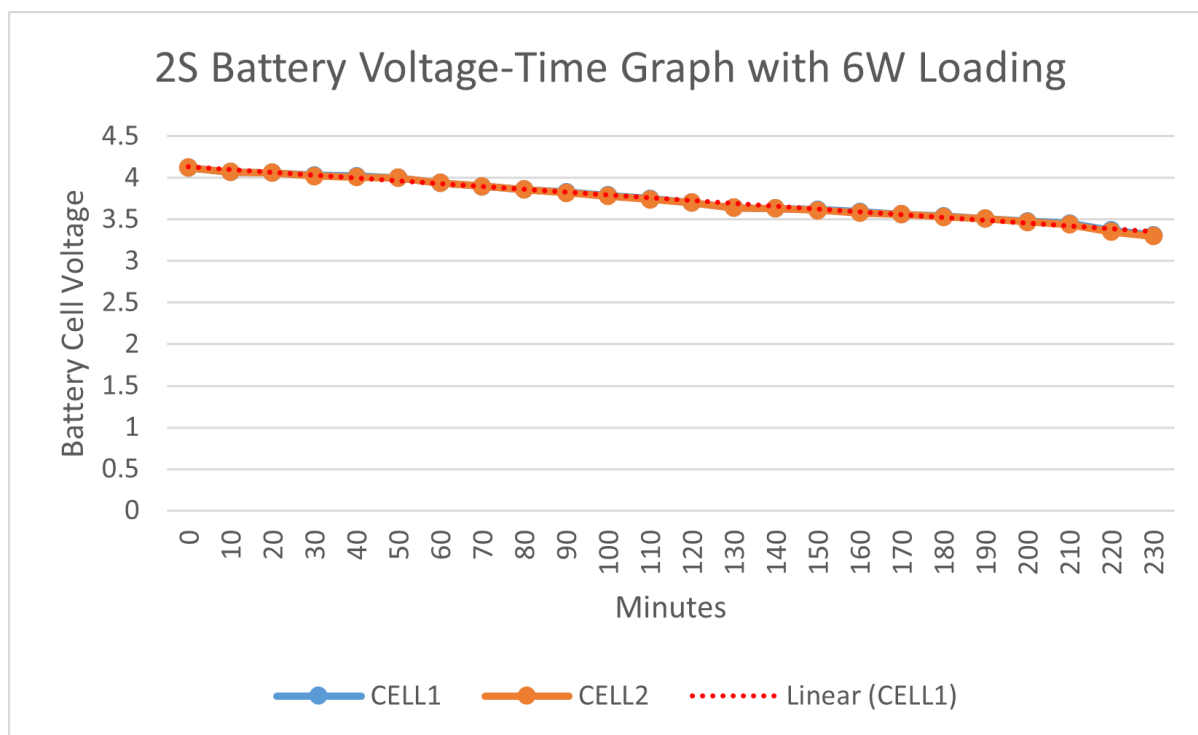


Figure 82: Voltage Discharge Profile for 2S Battery Pack Under 6W Load Over 240 Minutes Highlighting BMS Activation Threshold

Figure 82 presented visualizes the voltage of two individual battery cells (Cell 1 and Cell 2) over the course of a 240-minute full-power discharge test. The following observations are made from the plotted data:

- The voltage of both cells starts above 4V, which is within the expected operational range for a fully charged state.
- Over time, as the test progresses, there is a gradual decline in voltage, depicting the discharge curve of each cell under a constant 6W load.
- The voltage remains above the 3.15V threshold as specified for the discharging mode; however, the safety stop initiated by the BMS is observed at approximately 3.3V for both cells, slightly higher than the discharge setting of the smart charger.
- The test duration totals 240 minutes, during which the voltages of both cells exhibit a similar pattern of decline, showcasing the battery pack's balanced discharge behavior.

This data provides a quantitative foundation for understanding the battery discharge characteristics and the BMS's role in managing the discharge process under full-power conditions.

Data Analysis

The 240-minute full-power discharge test reveals significant findings about the battery pack's capacity and the effectiveness of the Battery Management System (BMS). Both cells within the 2S battery pack displayed a controlled and gradual voltage decrease under a 6W load, without any abrupt voltage drops. This behavior indicates a resilient battery capacity, capable of sustaining long durations of operation under high power demands.

Interestingly, the BMS initiated a safety stop at around 3.3V for both cells, which is higher than the smart charger's discharge setting of 3.15V. This preemptive action by the BMS underscores its conservative strategy for preserving battery life and preventing over-discharge. By halting the discharge before reaching the critical threshold, the BMS adds a margin of safety, thus avoiding the risks associated with deep discharging.

The consistency in the discharge curves between Cell 1 and Cell 2 suggests that the battery cells are well-matched and operate cohesively, which is vital for the longevity and balanced performance of the battery pack. The BMS's intervention at 3.3V, rather than allowing the voltage to drop to the set threshold, indicates its crucial role in safeguarding the battery's health.

The ability of the battery pack to endure a high-power load for a substantial period is a testament to its robustness, and when coupled with the diligent monitoring by the BMS, it forms a reliable power system for applications requiring extended operational capabilities.

Appendix 8: Test Document: Battery Failsafe Test

Location	Electrical-Electronics Department Middle East Technical University
Date	March 15 2024
Time	15:00
Description	Assessment of BMS response to battery failure.
Aim	Ensure BMS detects and manages battery failures.
Expected Outcome	Successful BMS activation of safety protocols.
Participants	BeyondTech Team Members and Studio Coordinators

Test Devices and Tools

Required Test Equipment: Active Vehicle Batteries (x2), Backup Batteries (x2), Smart Li-Ion Battery Charger, Power Supply.

1. Battery Voltage Thresholds

Ground truth/Calibration: The BMS recognizes a failing battery at specific voltage levels, typically set below the nominal cell voltage, which is essential for determining the point of failure.

2. Response Time

Ground truth/Calibration: Upon failure detection, the BMS’s reaction must occur within a defined time frame, ensuring timely activation of protective measures.

3. Battery Temperature Limits

Ground truth/Calibration: The BMS has set temperature thresholds to identify over-heating, crucial for preventing thermal runaway and ensuring safe operation.

4. Charger Discharge Rate

Ground truth/Calibration: The discharge rate of the smart charger, used for simulating battery failure, is consistent and verified, vital for test accuracy. For this test, 3.15V will be the rate.

5. Power Supply Stability

Ground truth/Calibration: The power supply must maintain a stable output during the test to avoid influencing the BMS’s performance and the test outcomes.

Test Environment:

- **Controlled Laboratory Setting:** The test is conducted in a lab environment where temperature and humidity are regulated to ensure consistent conditions.
- **Isolated Area:** The setup is placed in an isolated area to prevent external factors from influencing the test results.

- **Safety Equipment:** Proper safety equipment, such as fire extinguishers and protective gear, is available in case of battery failure leading to thermal events.
- **Monitoring Equipment:** Temperature sensors and voltage meters are set up to monitor the batteries' conditions in real-time during the test.
- **Emergency Shutdown:** An emergency shutdown mechanism is in place to immediately stop the test in case of unexpected or hazardous occurrences.

Test Benches and Setup



Figure 83: Battery Failsafe Test Setup

- **Battery Setup:** Arrange the active and backup batteries on the test bench, ensuring they are securely mounted and have clear labels to distinguish between them.
- **Charger Connection:** Connect the smart Li-Ion battery charger to the first battery, ensuring a stable and secure connection for controlled discharging.
- **Power Supply Setup:** Attach the power supply to the battery charger, verifying that it delivers the required voltage and current for the discharging process.
- **Monitoring System:** Set up the monitoring equipment, including voltage meters and temperature sensors, to track the battery's condition throughout the test.
- **BMS Integration:** Ensure the Battery Management System is properly integrated with the test setup, allowing it to monitor and respond to the battery's status in real-time.
- **Safety Protocols:** Implement safety protocols, such as placing fire extinguishers and protective gear nearby, and establish an emergency shutdown procedure.
- **Test Initiation:** Begin the test by activating the smart charger to discharge the battery, simulating a failure scenario, while closely monitoring the BMS's response.
- **Data Recording:** Record the BMS's actions, the battery's voltage and temperature data, and any other relevant parameters throughout the test for later analysis.

Test Parameters

Table 45: The control parameters

Parameter	Possible Selections	Number of Measurements
BMS Response Check	Active-Inactive	4

BMS Response Check: This parameter assesses the functionality of the Battery Management System (BMS) when the battery reaches its threshold voltage. It checks whether the BMS detects this condition and activates its safety protocols accordingly. The response is categorized as "Active" if the BMS successfully initiates protective measures, such as shutting down the power supply or issuing an alert. Conversely, the response is deemed "Inactive" if the BMS fails to recognize the battery's critical state and does not activate any safety protocols. This parameter is crucial for ensuring the reliability and safety of the BMS in real-world scenarios.

Expected Outcome: For the BMS Response Check, the expected outcome is that the BMS will be "Active" upon the battery reaching its threshold voltage. This indicates that the BMS has correctly identified the battery's critical condition and has initiated appropriate safety protocols to prevent potential hazards and protect the system. An "Active" response confirms the BMS's effectiveness in monitoring and managing battery health, which is essential for the overall safety and reliability of the vehicle.

Test Procedure

1. **Set Up Test Environment:** Ensure that the BMS and all monitoring equipment are correctly configured and operational. Charge all active and backup batteries to full capacity before starting the test.
2. **Testing Active Batteries:** Connect the first active battery to the smart Li-Ion battery charger. Set the charger to discharge the battery at the 3.15V. Monitor the battery voltage as it discharges and record the data. Once the battery reaches the threshold voltage, observe and record the BMS response. Repeat the process for the second active battery.
3. **Testing Backup Batteries:** Connect the first backup battery to the smart Li-Ion battery charger. Follow the same procedure as for the active batteries: discharge, monitor voltage, and record the BMS response. Repeat the process for the second backup battery.
4. **Data Analysis:** Analyze the recorded data to evaluate the BMS's performance across all batteries. Assess the BMS's ability to detect the threshold voltage and activate safety protocols.
5. **Safety Measures:** Throughout the test, ensure that safety protocols are in place to address any unexpected situations, such as thermal runaway or battery leakage. Keep fire extinguishers and protective gear readily available in the test area.
6. **Post-Test:** Recharge all batteries to full capacity and perform a visual inspection for any signs of damage or wear.

Test Data

During the battery fail test, voltage measurements were taken for each battery in both active and backup battery packs after the BMS safety stop was activated. The results are as follows:

First Active Battery Pack: The first battery registered a voltage of 3.27V, while the second battery showed a slightly higher voltage of 3.29V. These readings indicate a consistent response from the BMS in managing the discharge process and activating the safety stop.

Second Active Battery Pack: Voltages in this pack varied from 3.26V to 3.29V, with the first and fourth batteries showing the highest voltage of 3.29V. The second battery had the lowest voltage at 3.26V, and the third battery was at 3.28V. The slight variations in voltage demonstrate the BMS's ability to maintain control over multiple batteries within a pack.

First Backup Battery Pack: The first battery in this pack had a voltage of 3.26V, and the second battery showed a slightly higher voltage of 3.28V. These values suggest that the BMS effectively manages backup batteries, ensuring they are also protected in case of a simulated failure.

Second Backup Battery Pack: Voltages in this pack were more uniform, with the first and second batteries both at 3.27V and the third and fourth batteries at 3.28V. This uniformity is indicative of the BMS's consistent performance across different battery packs.

Overall, the voltage readings across all batteries after the BMS safety stop ranged from 3.26V to 3.29V. These results highlight the BMS's reliability in detecting the threshold voltage and executing the safety protocols, ensuring the vehicle's electrical systems are safeguarded against potential battery failures.

Data Analysis

The battery fail test was designed to evaluate the Battery Management System (BMS) in terms of its ability to detect and manage simulated battery failures. Throughout the test, batteries were gradually discharged to a predetermined threshold voltage using a smart charger set to a 3.15V discharging mode. The BMS's reaction to this scenario was carefully observed. The BMS demonstrated its effectiveness by accurately identifying the voltage drop across all batteries, both active and backup. It activated its safety protocols as anticipated, initiating a shutdown procedure when the voltage reached the threshold. Notably, the measured battery voltage after the BMS safety stop was found to be in the range of 3.27-3.30V for all batteries, indicating a prompt and precise response slightly above the set discharging mode of the smart charger.

These results highlight the BMS's reliability in monitoring battery health and ensuring the safety of the vehicle's electrical systems. The system's successful detection and response to the simulated battery failures provide assurance of its readiness to handle real-world scenarios, maintaining the operational safety and longevity of the vehicle. The consistent performance of the BMS across all batteries reinforces confidence in its ability to safeguard the vehicle's performance.

Appendix 9: Test Document: Parking Maneuver Test

Location	Electrical-Electronics Department Middle East Technical University
Date	April 23 2024
Time	20:30
Description	This test sheet explains the procedure for testing the parking maneuver.
Aim	The aim of this test is to test the parking maneuver.
Expected Outcome	The final position of the vehicle is completely inside the parking spot.
Participants	BeyondTech Team Members and Design Studio Coordinators

Test Devices and Tools

Required Test Equipment: The test equipment required for this experiment includes the Raspberry Pi 5 onboard computer, the Longthin Software Suite, a computer with SSH capabilities, ESP8266 dongle, 3x3 meter test area with parking spot, and all subsystems of the vehicle such as the sensor, power, mechanical, and wireless communication subsystems.

Test Environment

The test environment of the test is the 3x3 meter parking area with the parking spot. During the test, the vehicle will try to park into the parking spot by starting at a point close to it. The main purpose is to observe the behavior of the controller when the orientation information coming from the visual positioning subsystem based on the parking spot boundary lines is used during the parking approach.

Test Parameters

Table 46: The control parameters

Parameter	Range	Step Size	Number of Measurements
Starting yaw angle(ψ)	$[-45^\circ \ 45^\circ]$	45°	3
Starting distance	$[30\text{cm} \ 90\text{cm}]$	30cm	3

Test Procedure

1. Place the vehicle into a starting point based on yaw angle and distance from the parking spot.
2. Connect the ESP8266 dongle to external computer.
3. Start the software suite on the external computer.
4. Turn on both the front and rear compartments of the robot using the power switches.

5. Connect to the Raspberry Pi 5 onboard computer using SSH.
6. Start the control system on the Raspberry Pi 5 onboard computer by executing the "longthin-bridge" command to establish communication with the robot's subsystems and the "longthin-parking" command to initiate the control algorithm.
7. Monitor the robot's yaw angle and position on the external computer, and save the data for analysis.
8. Verify that all parts of the robot is inside the parking spot by checking the final yaw angle and position.

Test Data

The primary objective of the data collection was to determine whether the vehicle could successfully park within a designated blue region under various starting conditions, categorized by yaw angles and distances. The outcome of each test was observed visually to assess parking success. Additionally, the time duration for each parking attempt was recorded to gauge the efficiency of the parking maneuver. The results are summarized in the **Table 47** below.

Table 47: The test results

Starting distance	Starting yaw angle(ψ)	Completion
30	-45	Fail
30	0	Success
30	45	Fail
60	-45	Success
60	0	Success
60	45	Success
90	-45	Success
90	0	Success
90	45	Success

Each parking attempt, regardless of the outcome, took approximately 12 seconds. This consistent timing across various tests indicates a high level of efficiency in the vehicle's control system for executing the parking maneuver within a relatively short duration. This information is crucial for understanding both the performance capabilities of the autonomous system and areas where improvements may be necessary, particularly in scenarios where the vehicle failed to park correctly.

Data Analysis

The autonomous parking system underwent rigorous testing to evaluate its ability to autonomously park within a designated blue region under various conditions. Below are the comprehensive results, including specific instances where adjustments were required:

- **Accuracy of Parking:** The vehicle demonstrated high reliability in parking successfully when starting from a straight angle (0 degrees) at any distance. Challenges were observed at extreme yaw angles (± 45 degrees) from the shortest distance (30 cm), where the vehicle failed to park within the designated area.
- **Efficiency of Maneuver:** The tests showed that each parking attempt took approximately 12 seconds, indicating a high level of operational efficiency. However, the scenarios with initial conditions of ± 45 degrees yaw angle at 30 cm starting distance showed that while the maneuver time remained consistent, these cases resulted in parking failures.
- **Impact of Yaw Angle and Distance:** The vehicle's performance at 60 cm and 90 cm distances was consistently successful across all yaw angles, suggesting that the system handles longer approach distances more effectively. The failures at ± 45 degrees from the closest distance (30 cm) highlight limitations in the system's ability to correct extreme angles over short distances.
- **Adaptability and Safety Enhancements:** The system showed robust performance in various environmental conditions except when approaching with sharp angles as specified. To enhance safety and ensure reliability, a new protocol will be implemented: if the vehicle approaches with an angle greater than a predetermined threshold at a specific distance, it must retreat and reattempt parking. This measure aims to address the detected issue of misalignment during sharp angle approaches and ensures the vehicle parks accurately within the safety margins.
- **Proposed System Improvements:** Further development will focus on optimizing sensor integration and control algorithms to better handle edge cases like the ± 45 degrees, 30 cm scenario. Continuous testing will be conducted to refine the reattempt mechanism, ensuring it functions seamlessly under all conditions, further reinforcing the system's reliability and safety.

Based on these observations, it is recommended to implement a safety protocol where the vehicle assesses its yaw angle and distance upon approach. If the angle exceeds a predetermined threshold at a close distance, the vehicle will retreat and attempt to re-align before parking. This approach aims to reduce failures by providing the system with an additional opportunity to adjust its position under challenging initial conditions.

Appendix 10: Test Document: Final Mission Test

Location	Electrical-Electronics Department Middle East Technical University
Date	April 23 2024
Time	21:30
Description	Procedure explanation for testing the full autonomous parking mission.
Aim	The aim of this test is to verify the autonomous parking.
Expected Outcome	Successful and precise parking into the parking spot.
Participants	BeyondTech Team Members and Design Studio Coordinators

Test Devices and Tools

Required Test Equipment: The test equipment required for this experiment includes the Raspberry Pi 5 onboard computer, the Longthin Software Suite, a computer with SSH capabilities, ESP8266 dongle, 3x3 meter test area with parking spot, and all subsystems of the vehicle such as the sensor, power, mechanical, and wireless communication subsystems.

Test Environment

The test environment of the test is the 3x3 meter parking area with the parking spot. This is the final test of the vehicle and it includes all the subsystems of the vehicle. The vehicle will be started at a random position with a random orientation and it will be expected to find the parking spot and park itself autonomously. Successful completion of this test will verify the successful completion of the autonomous parking mission and the satisfaction of all the system requirements.

Test Parameters

Table 48: The control parameters

Parameter	Range	Step Size	Number of Measurements
Starting yaw angle(ψ)	[0° 360°]	120°	3
Starting x	[75cm 225cm]	75cm	3
Starting y	[75cm 225cm]	75cm	3

Test Procedure

1. Place the vehicle into a starting point based on yaw angle and x-y position parameters.

2. Connect the ESP8266 dongle to external computer.
3. Start the software suite on the external computer.
4. Turn on both the front and rear compartments of the robot using the power switches.
5. Connect to the Raspberry Pi 5 onboard computer using SSH.
6. Start the control system on the Raspberry Pi 5 onboard computer by executing the "longthin-bridge" command to establish communication with the robot's subsystems and the "longthin-mission" command to initiate the control algorithm.
7. Monitor the robot's yaw angle and position on the external computer, and save the data for analysis.
8. When the mission is finished, check whether all parts of the robot is inside the parking spot by recording the final yaw angle and position and the total mission duration and the total mission duration.

Test Data

The final mission test was structured to rigorously evaluate the autonomous parking system's capability to perform under varied and random initial conditions within a controlled 3x3 meter test environment. The test was designed to simulate real-world scenarios, testing the vehicle's navigation and parking precision from different starting positions and orientations. Each trial was meticulously recorded, capturing the vehicle's position, orientation, time to complete the parking maneuver, and any instances of failure. The data was meticulously tabulated in **Table 49**. Also, in the **Figure 84**, you can find the parking duration vs starting position graph.

Table 49: The final mission test results

Position x (m)	Position y (m)	Orientation (degrees)	Mission Time(sec)
45	45	0	6.03
45	115	0	38.4
45	185	0	40.5
115	45	0	11.75
115	115	0	26.26
115	185	0	32.92
185	45	0	32.03
185	115	0	31.98
185	185	0	19.38
45	45	120	37.06
45	115	120	31.8
45	185	120	21.64

Position x (m)	Position y (m)	Orientation (degrees)	Mission Time(sec)
115	45	120	Fail
115	45	120	28.81
115	115	120	38.38
115	185	120	Fail
115	185	120	32.92
185	45	120	33.38
185	115	120	31.96
185	185	120	25.56
45	45	240	34.32
45	115	240	34.18
45	185	240	38.15
115	45	240	38.08
115	115	240	Fail
115	115	240	30.42
115	185	240	36.03
185	45	240	Multiple Fail
185	45	240	30.17
185	115	240	27.25
185	185	240	32.56
		Average	30.44

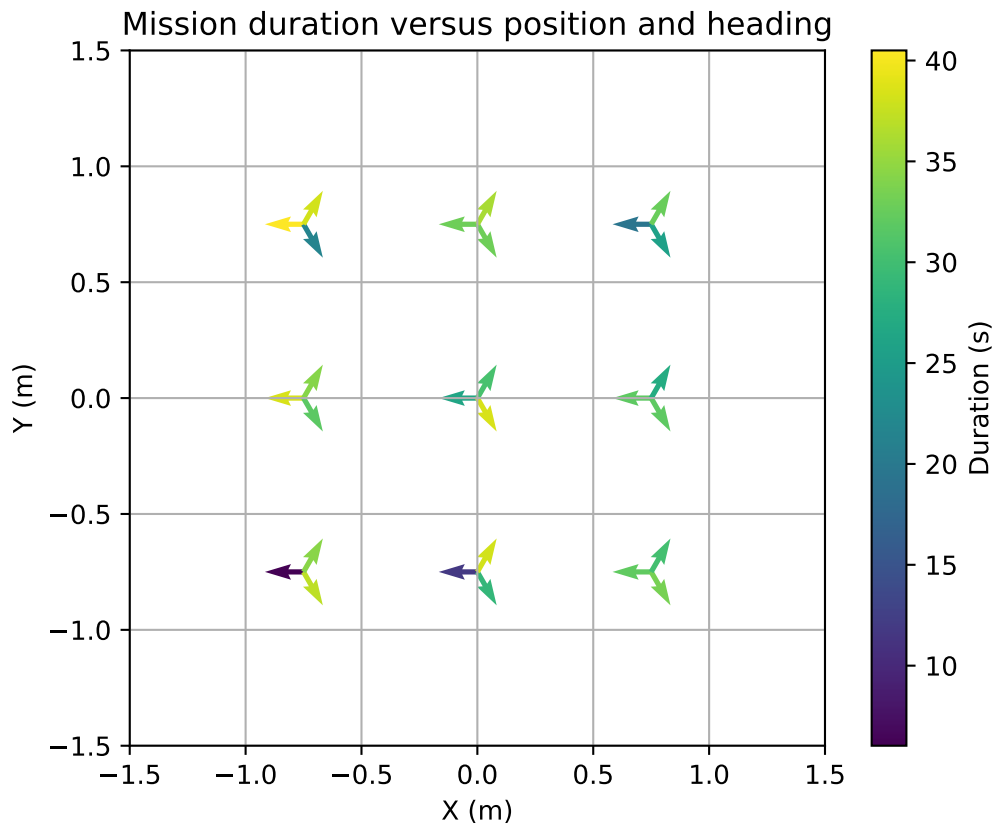


Figure 84: Parking duration vs position graph

The vehicle successfully parked within the stipulated time in the majority of test scenarios. Failures were specifically noted in tests with certain orientations and when the vehicle started from the farthest distance of the parking spot. These were linked to a known software bug, which is under active investigation.

Data Analysis

The final mission test was a critical component of the project, aimed at verifying the autonomous system's effectiveness in achieving a reliable parking maneuver from any random starting point under strict time constraints.

- **Overall Performance and Efficiency:** The autonomous parking system proved highly effective, with an average successful mission time of 30.44 seconds across all trials, comfortably below the one-minute requirement with a success rate of 87%. This performance underscores the system's capability to quickly and efficiently navigate and position itself within the parking spot with higher accuracy.
- **Analysis of Failures:** Failures occurred during tests with specific challenges related to initial orientation and positional extremes. These failures were attributed to a recurrent software issue, which caused instability in the vehicle's navigation algorithm under complex conditions. Immediate system reboots temporarily resolved the issues, confirming the need for a permanent software fix.
- **Verification of the System Requirements:** With this final mission test, it is once more verified that all subsystems work properly satisfying their respective functional requirements, and their integration is also successful.

- **System Reliability and Adaptability:** Despite some failures, the tests demonstrated the system's robust adaptability to a variety of starting positions and orientations. The vehicle maintained wireless communication integrity and met all predefined movement constraints, confirming its readiness for real-world application pending resolution of identified issues.
- **Recommendations for Further Development:** It is imperative to address the software bugs that led to the observed failures to enhance the system's reliability. The development team should prioritize refining the navigation algorithms to better handle extreme orientations and close-proximity starts. Ongoing testing should include more varied scenarios to rigorously challenge the system's limits and ensure its operational stability across all possible real-world conditions.

This comprehensive analysis provides a clear view of the system's capabilities and areas for improvement. The data collected from this final mission test forms a solid foundation for the subsequent optimization phase, aiming to fully prepare the autonomous parking system for a dependable final demo.

Appendix 11: User Manual

The BeyondTech Longthin Hauler is an innovative, miniaturized autonomous vehicle that includes a deployable 3x3m parking area and a designated parking spot. Designed for precision and efficiency, this vehicle showcases cutting-edge technology suitable for both hobbyists and technology enthusiasts interested in autonomous navigation systems.

The vehicle is composed of three main components: the front electronics, rear electronics, and a hollow aluminium tube. It is powered by two types of lithium-ion batteries, each connected to the corresponding front or rear electronics.

The parking setup comprises eight orange colored plastic strips and eight aruco marker stands with preconfigured IDs. These elements are designed to be assembled quickly into a 3x3m area. The parking spot itself is a 60cm region highlighted by a single aruco marker stand at the end, facilitating precise vehicle parking.

Package Contents

The package includes all necessary components and tools required for full assembly and operation:

- User manual
- Aluminium tube, 50cm in length and 8cm in diameter
- Front electronics compartment
- Rear electronics compartment
- Two silicone tires
- Two-cell lithium-ion battery pack
- Four-cell lithium-ion battery pack
- Eight 20cm aruco marker stands
- Eight colored plastic strips
- One 30cm aruco marker stand
- Software suite for Windows, Linux, and MacOS
- ESP8266 dongle compatible with USB2.0 and USB3.0
- Test result documents

The package contents are shown in **Figure 85**.

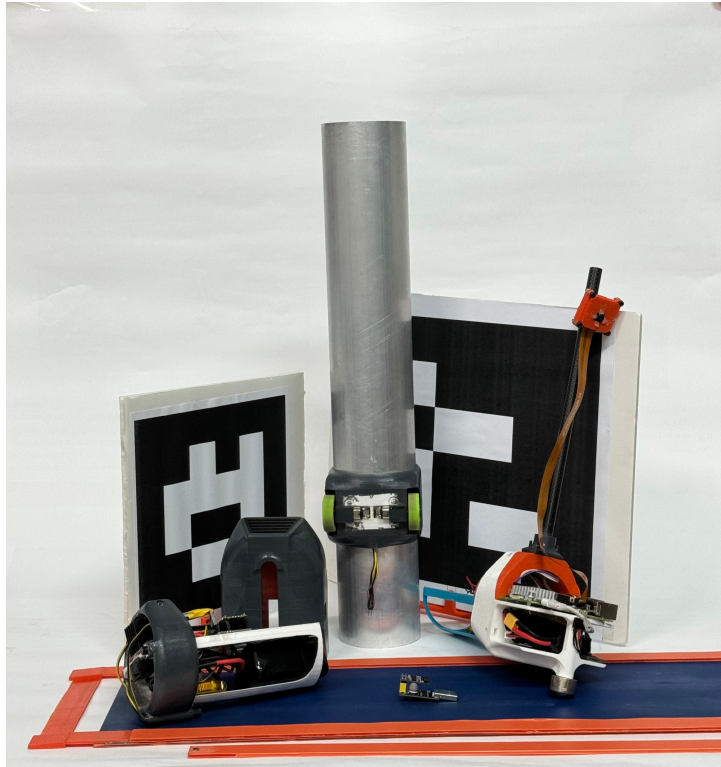


Figure 85: Package contents

Quick Start Guide

This guide provides a step-by-step approach to get your vehicle up and running:

1. Connect the 2-cell lithium-ion battery pack to the front electronics using the provided cable.
2. Connect the 4-cell lithium-ion battery pack to the rear electronics using the provided cable.
3. Securely attach the front and rear electronics to the aluminium tube using the provided screws.
4. Fit the silicone tires onto the motors located in the rear electronics compartment.
5. Assemble the parking area and spot according to the detailed instructions in the assembly guide.
6. Activate the front electronics using the power switch, ensuring the vehicle is stable and stationary during this initial power-up.
7. Power on the rear electronics with its dedicated switch.
8. Allow the vehicle a few minutes to calibrate; during this time, it should remain completely stationary.
9. Observe the green LED on the front electronics; its illumination signifies that the vehicle is ready to operate.
10. Position the vehicle within the parking area and initiate operation by pressing the start button on the rear.
11. Watch as the vehicle expertly moves and parks itself within the designated parking spot.

Vehicle Assembly

Detailed instructions for assembling the vehicle components:

1) Front Electronics

The front electronics include the main control unit and sensors. Before assembly, ensure all parts are undamaged and present. Connect the 2-cell lithium-ion battery pack to the front electronics using the provided cable shown in **Figure 86**.

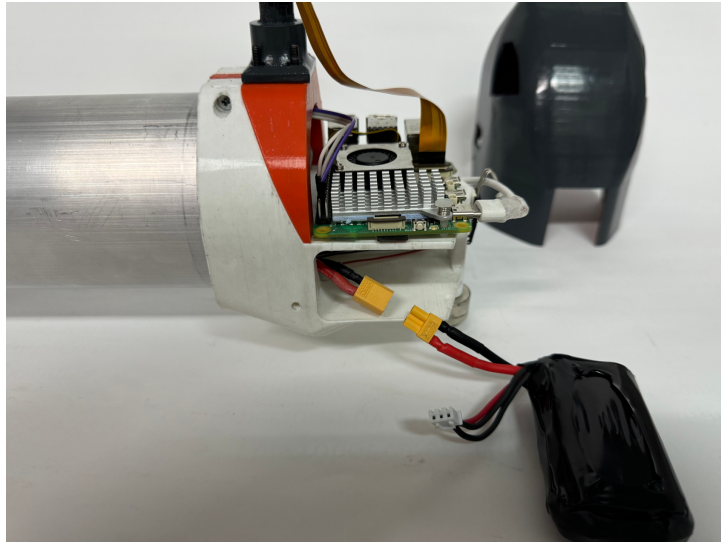


Figure 86: Front electronics with power connection

2) Rear Electronics

The rear electronics contain the propulsion system, including motors and motor drivers. Inspect for any signs of damage or defects before connecting the 4-cell lithium-ion battery pack using the provided cable shown in **Figure 87**.

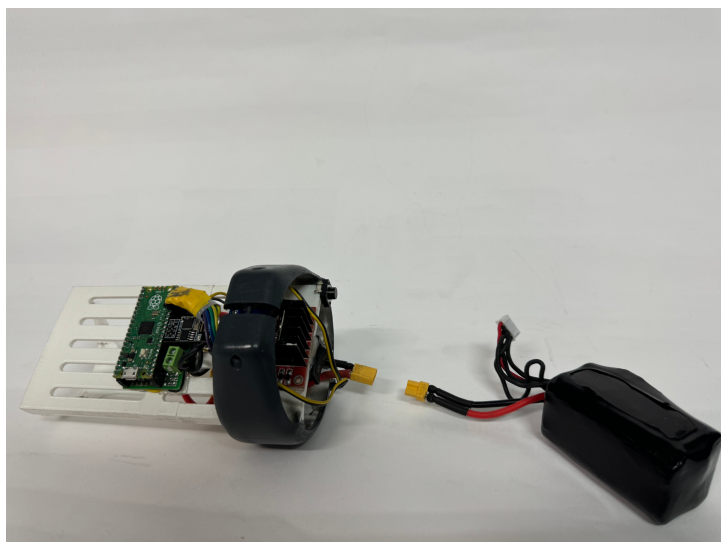


Figure 87: Rear electronics with power connection

3) Aluminium Tube

The aluminium tube serves as the chassis connecting the front to the rear electronics. Attach both electronic units to the tube and secure them with the provided screws, ensuring a tight and stable assembly. The screw holes are located on the front and rear electronics compartments, as shown in **Figure 88**.

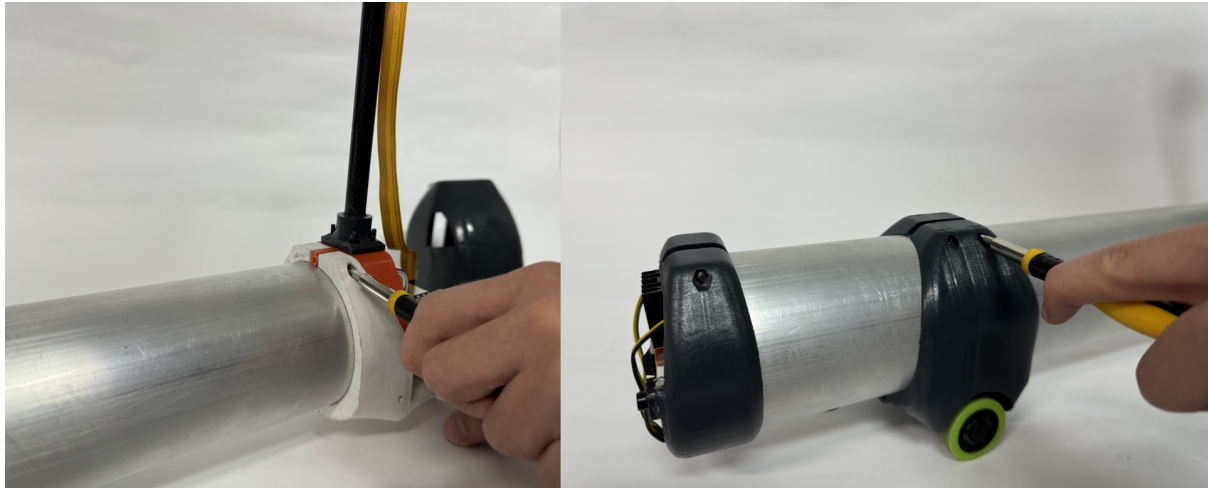


Figure 88: Screw holes on the front and rear electronics compartments

4) Silicone Tires

The silicone tires provide traction and mobility. Begin by using the special tool to separate the tire from its central hub and remove the tires as shown in **Figure 89**. Attach the hub to the motor shaft and secure it with screws. Finally, reattach the tire to the hub.



Figure 89: Silicone tire removal

Assembled Vehicle

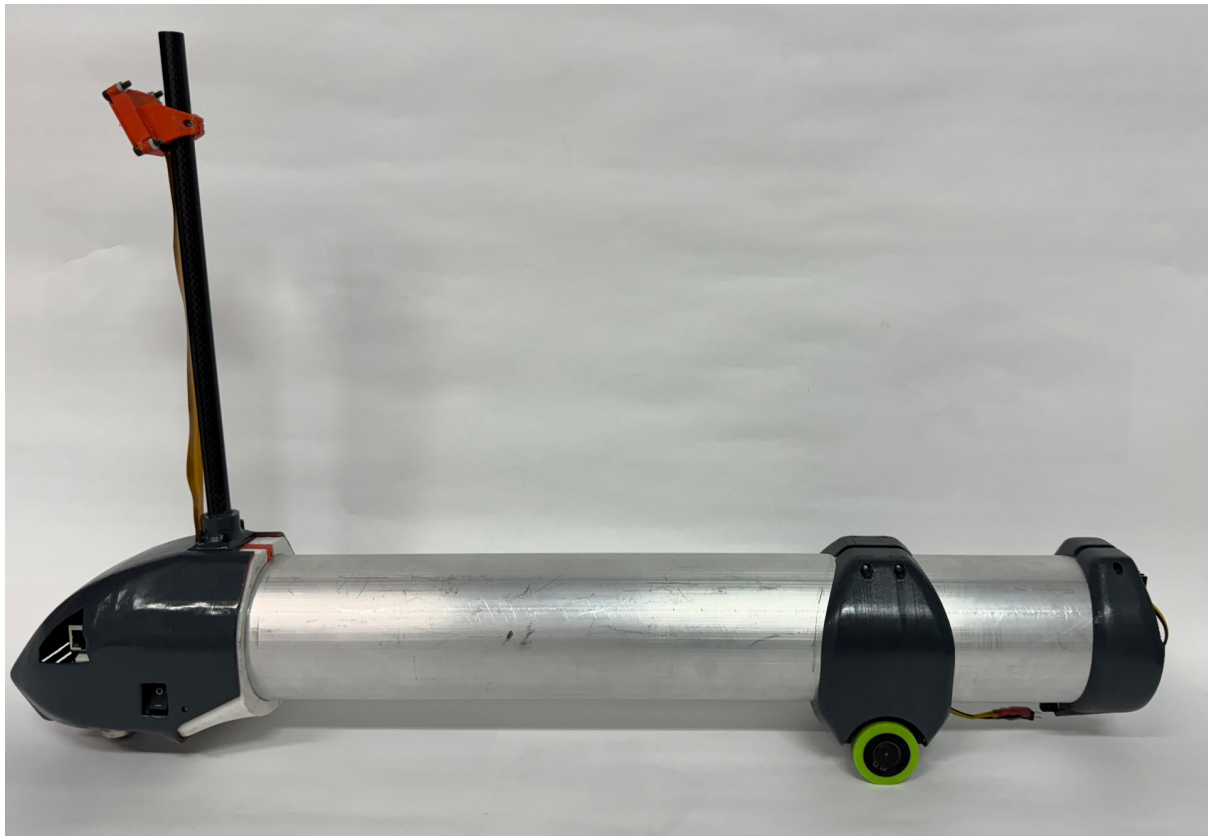


Figure 90: Assembled Vehicle

Figure 90 showcases the fully assembled autonomous underwater vehicle, equipped with advanced navigational and operational components. The Aruco Marker holder is prominently positioned on the vertical support to enable precise localization and tracking during deployment. Note the integration of silicone tires on the hubs to enhance maneuverability and traction in underwater environments.

5) Aruco Marker Installation

To ensure accurate placement and functionality of the Aruco Marker, follow these detailed steps: Begin by identifying the magnets located beneath the Aruco Holder and the designated area on the installation surface. Carefully align the magnet on the Aruco Holder with the corresponding magnet on the installation area to ensure that the magnets are precisely aligned to secure the holder in place. Once aligned, press the Aruco Holder gently towards the surface to engage the magnets fully. Confirm that the holder is firmly attached and stable.

By following these steps, the Aruco Marker will be properly installed, ensuring it remains securely in place during operation. You can find the clear steps in **Figure 91**.

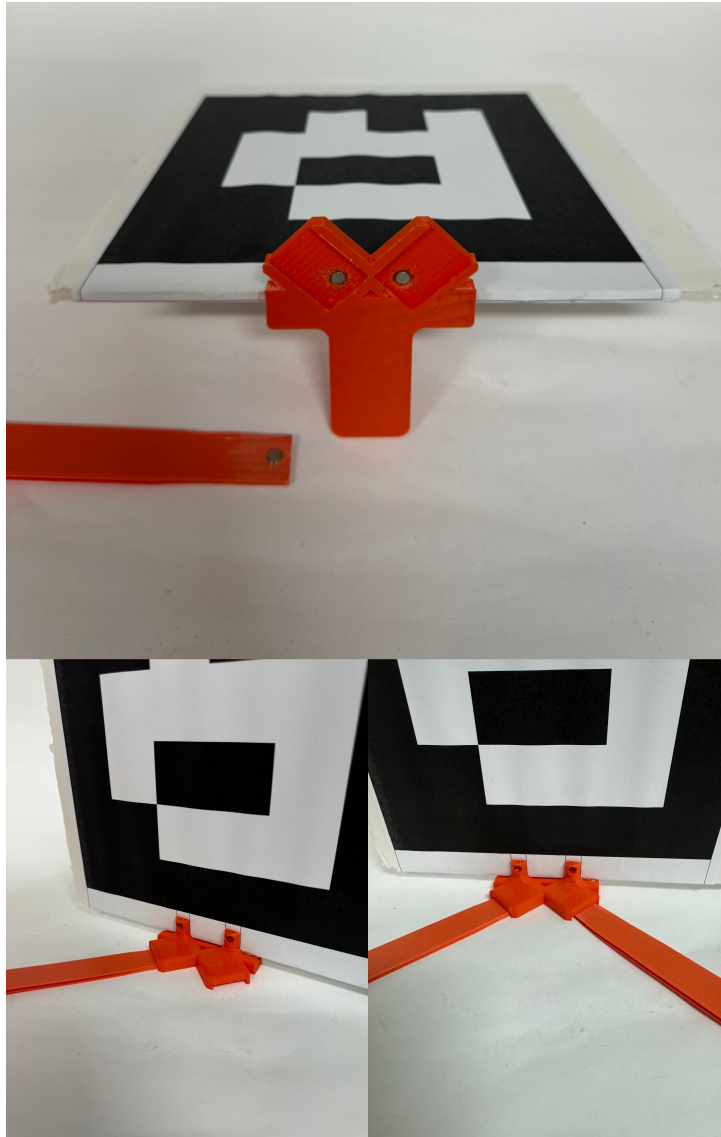


Figure 91: Aruco Marker Installation Steps

6) Parking Area

Layout the eight orange-colored plastic strips to form a square 3x3m area. Place the Aruco marker stands in numerical order in a clockwise fashion at each corner and along the edges to define the perimeter clearly as shown in **Figure 92**.

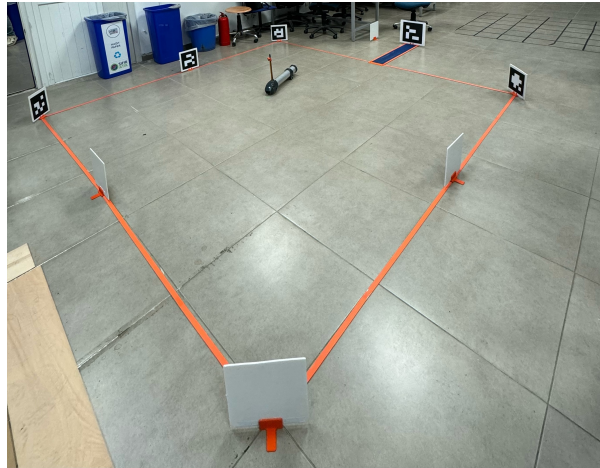


Figure 92: Parking area layout

7) Parking Spot

The parking spot can be installed at any edge or within the parking area as needed. Place the 95cm parking spot in the desired location and install the 30cm aruco marker stand at its end, ensuring the marker faces towards the driving area as shown in **Figure 93**. If situating the parking spot centrally, remove one of the 20cm aruco stands from the perimeter to accommodate this setup.



Figure 93: Ready to use parking spot

Vehicle Software

The vehicle is equipped with sophisticated preinstalled firmware designed for autonomous operation. This includes capabilities for self-calibration, navigating the parking area, and executing precise parking maneuvers. The software suite provided allows for firmware updates, vehicle calibration, system diagnostics, real-time monitoring, and manual control overrides. It is compatible across Windows, Linux, and MacOS platforms. The software suite also includes a ground control station for monitoring vehicle status and performance. The user interface is intuitive and user-friendly, providing a comprehensive overview of the vehicle's operational parameters, sensor data, and battery status. The ground station user interface is shown in **Figure 94**. You can also view the plots tab in the ground control station in **Figure 95**. The simulation environment for the vehicle is shown in **Figure 96**.

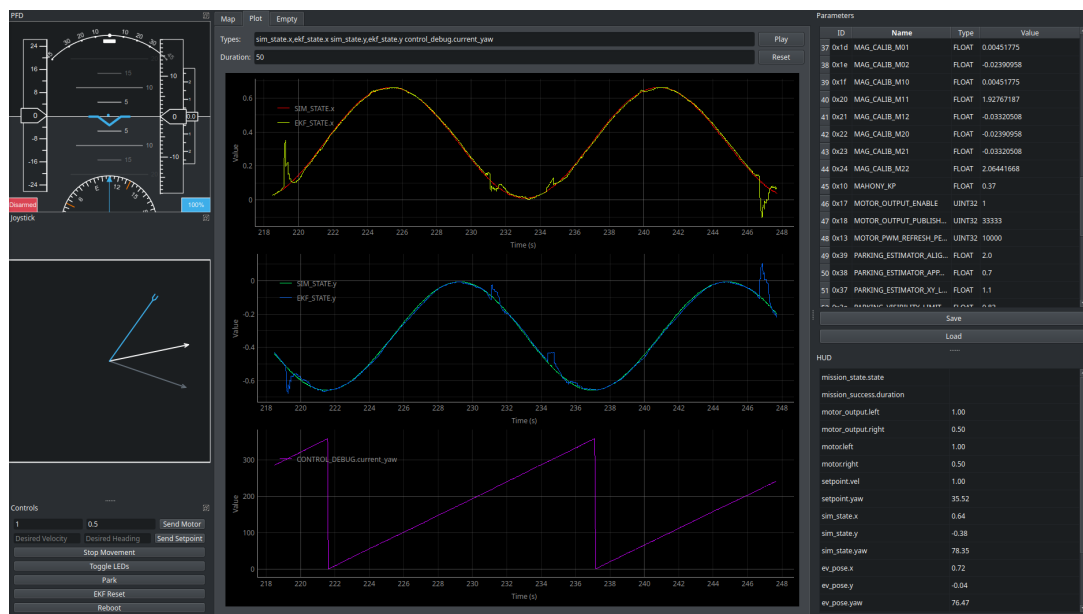


Figure 94: Ground control station user interface

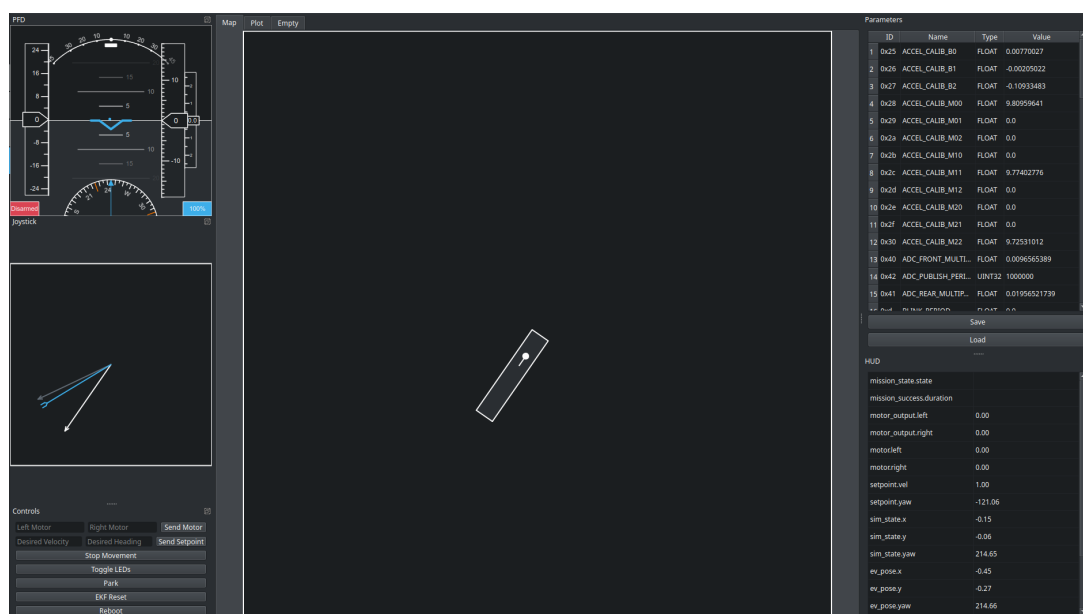


Figure 95: Ground control station user interface plots tab

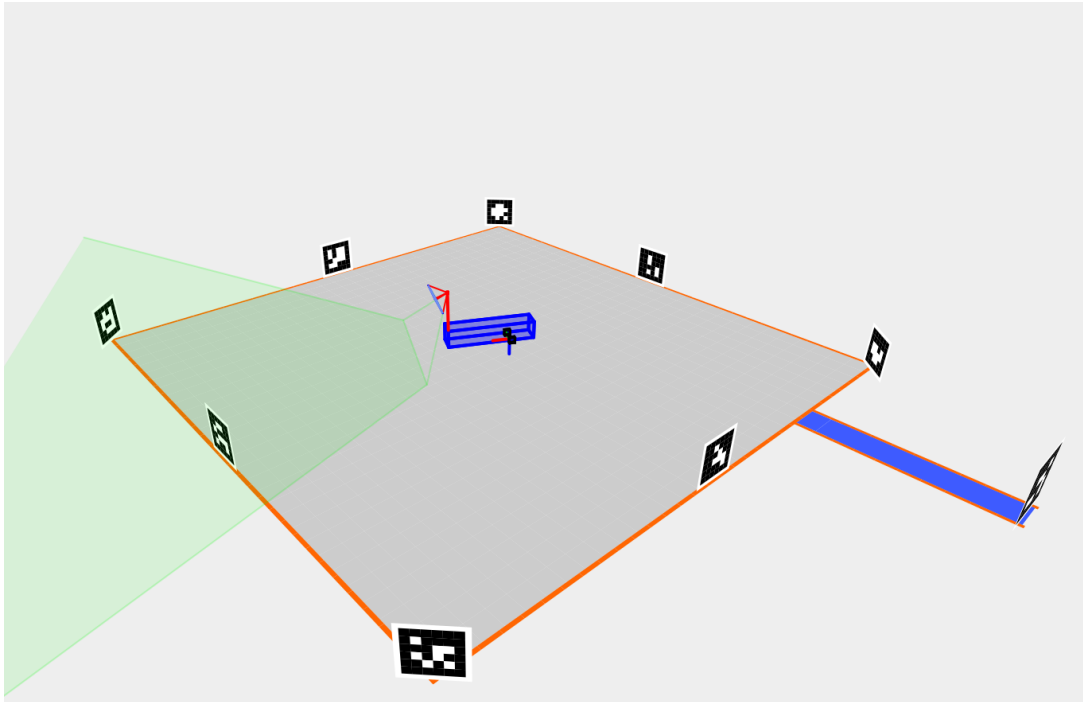


Figure 96: The simulation environment for the vehicle

Software Installation

The software suite is distributed as a Python wheel package, ensuring easy installation on systems running Python 3.6 or later. Begin by ensuring Python is installed on your system, available at <https://www.python.org/downloads/>. Install the software suite with the following commands, depending on your need for the stable release or the latest development build:

```
pip install beyondtech-longthin-hauler
```

or for the latest version:

```
pip install git+https://gitlab.com/beyondtech/longthin-hauler.git
```

Vehicle Operation

Instructions for basic operations and modes available for the vehicle:

Starting the vehicle

1. Complete the vehicle assembly as described above.
2. Position the vehicle at the start point within the parking area.
3. Activate the front electronics by turning on the power switch, ensuring the vehicle remains stationary while powering up.
4. Turn on the rear electronics using its switch.
5. Allow the vehicle to perform self-calibration; it will remain stationary until calibration is complete.
6. Once the green LED on the front electronics lights up, the vehicle is ready for operations.

(Optional) Starting the Ground Control Station

1. Connect the ESP8266 dongle to your computer.
2. Launch the software suite using the command:

```
longthin-runner --config gcs.yaml
```

3. The software will connect to the vehicle and start monitoring various parameters such as attitude and battery status.

Parking mode

1. Follow the assembly and initial start-up instructions.
2. Place the vehicle at the designated starting point within the parking area.
3. Press the start button on the rear to engage parking mode.
4. The vehicle will signal its operational status by blinking LEDs and emitting beeps.
5. Watch as the vehicle moves toward the parking spot.
6. The vehicle will execute its parking maneuver and automatically stop upon successful parking, indicated by the cessation of LED blinking and a final beep.

Manual mode

1. Ensure the vehicle is fully assembled and operational as per the setup guide.
2. Position the vehicle within the parking area.
3. Use the arrow keys on a connected keyboard or the mouse interface on the software to manually direct the vehicle.
4. The vehicle will move according to the input commands and stop when input ceases.

Trouble Shooting

Comprehensive troubleshooting guide to assist in resolving common issues:

The vehicle cannot be powered on

1. Verify the battery levels. The vehicle will not operate if the batteries are depleted.
2. Ensure that all power switches are in the ON position. You can find the figure positions from **Figure 97**
3. Check that all battery connections are secure and properly connected.
4. Consider that the vehicle might be in calibration mode. Allow time for it to complete this process.

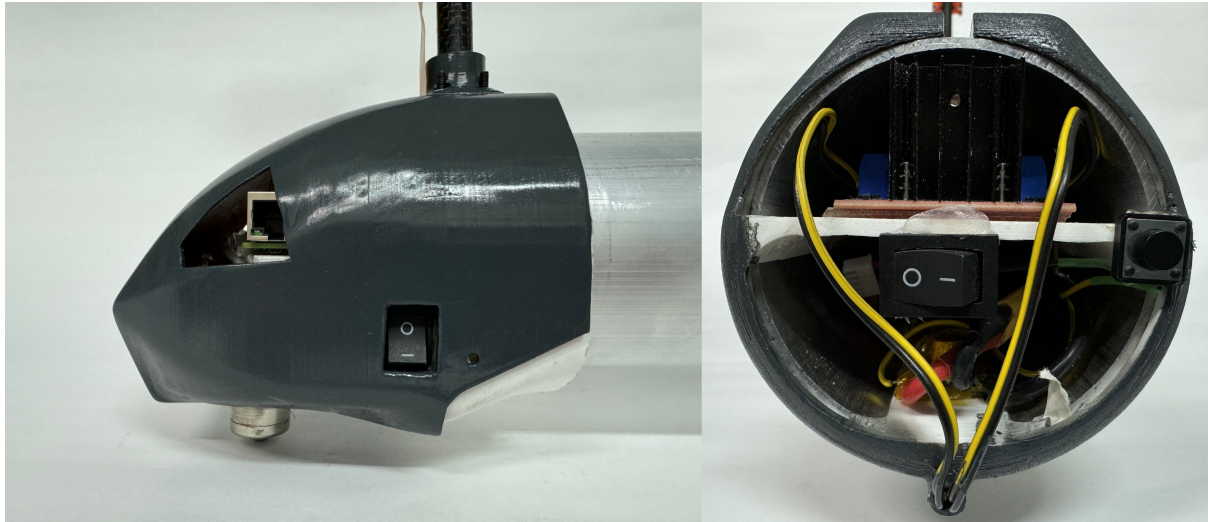


Figure 97: Front and Rear Switch Positions

The vehicle is powered on but the green LED won't turn on

1. Be patient as the vehicle may still be calibrating. Wait for the calibration to complete.
2. Ensure that the vehicle is correctly positioned within the designated parking area; it cannot operate outside of this zone.
3. If the issue persists, try powering off the vehicle, wait a few seconds, then power it on again, ensuring you allow time for calibration.

The vehicle is powered on but won't move

1. Confirm that the vehicle is within the parking area; it will not operate outside this designated zone.
2. Check that the vehicle has completed its calibration process. Wait for it to finish if it has not.
3. Make sure the vehicle is in the correct operational mode. Activate the parking mode by pressing the start button on the rear.
4. Inspect the motor connections. They must be securely connected for the vehicle to move. You can find the motor connections **Figure 98**.

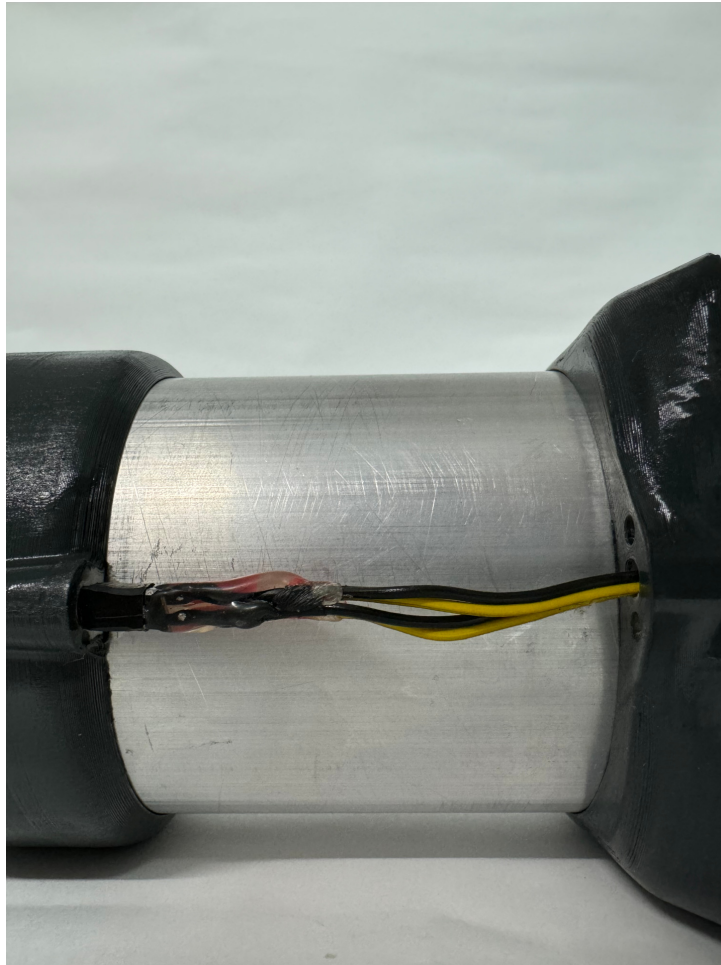


Figure 98: Motor Cable Connections

The vehicle is moving but not parking correctly

1. Verify that the vehicle is within the parking area and is set to the parking mode by pressing the start button on the rear.
2. Check the correct placement and orientation of the parking spot within the parking area.
3. If the vehicle approaches the parking spot but fails to park, inspect the aruco marker stand at the end of the parking spot for proper alignment and visibility.

The vehicle is not responding to manual controls

1. Check the connection of the ESP8266 dongle; it must be properly connected to the computer for manual control.
2. Look for any error messages in the software suite which might indicate what the issue is.
3. Ensure that the vehicle is set to manual mode. If it has commenced parking, you can override this by pressing the start/stop button on the vehicle or through the ground control station.
4. Confirm that the motor connections are intact and securely fastened.

The software suite is not functioning properly

1. Verify that the Python version installed on your computer is 3.6 or higher, as this is necessary for the software suite's compatibility.
2. Ensure that the software suite has been installed correctly. If in doubt, reinstall using the provided installation commands.
3. Check the ESP8266 dongle connection to ensure that it is properly established for the software suite to function effectively.

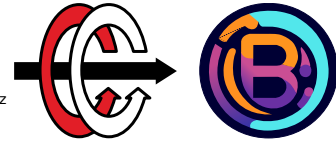
Appendix 12: Poster of the Project

Longthin Hauler

BeyondTech

Shareholders: Ahmet Eren Cengiz, Furkan Çitil, Furkan Karatoprak, Kamil Anıl Işık, Tolga Demirdal, Umut Gürbüz
{eren.cengiz, furkan.citil, furkan.karatoprak, kamil.isik, tolga.demiral, gurbuz.umut}@metu.edu.tr

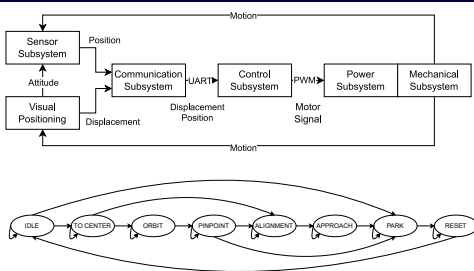
Supervisor: Assoc. Prof. Ayşe Melda Yüksel Turgut



Project Description

The Long Thin Hauler is an autonomous vehicle project developed by BeyondTech, designed to address the challenges of parking in restricted urban spaces. This project, created by final-year Electrical and Electronics Engineering students, demonstrates the application of autonomous navigation technologies, enabling the vehicle to efficiently park within a compact 3m by 3m area.

System Design



Features

- Modular design for mechanical components
- Portable and easily deployable parking area and spot
- Complete software stack for autonomous navigation
- A ground station for monitoring and live tuning of the vehicle
- The software suite available as a PyPI package
- Use of LED and buzzer to indicate the status of the motion of the vehicle

Compliance with Requirements

Requirements	Test Result	Compliance
Autonomous park of the vehicle	87% success	✓
Communication at a distance greater than 30 cm.	99.5% success	✓
Parking under 1 minute.	Parking in 30.44s in average	✓
Operating time longer than 60 minutes	Continuous operation for 120 minutes	✓

Requirements

- The chassis should be metallic.
- The length should be more than 50cm, and the width should be between 3cm and 10cm.
- The sensors should be on the front, and the actuators should be on the rear with at least 30cm apart from the sensors.
- The communication between sensors and actuators should be wireless through the hollow tube.
- The parking spot width can be at most 1.5 times the width of the vehicle.
- The parking should be completed under 1 minute with forward only motion.
- The vehicle should operate at least 60 minutes.

Energy Budget

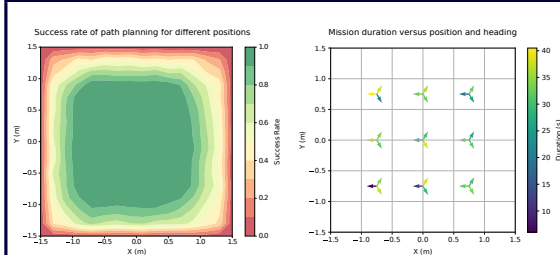
Front:

Component	Power
Raspberry Pi 5	6.5 W
Raspberry Pi Pico	250 mW
ESP8266	165 mW
Total	7 W

Rear:

Component	Power
Motors	7.2 W
Raspberry Pi Pico	325 mW
ESP8266	165 mW
Total	7.7 W

Test Results and Performance



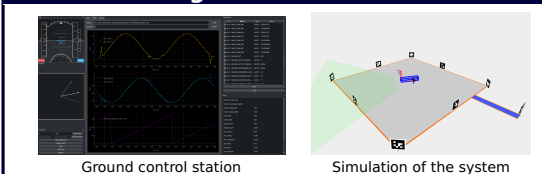
Cost Breakdown

Component	Quantity	Total Price
Raspberry Pi 5 + Cooler	1	\$107.3
Raspberry Pi Camera Module 3	1	\$37.15
Raspberry Pi Pico	2	\$9.82
ESP8266	2	\$2.94
Pololu Motor 3.6 kg-cm 140 rpm	2	\$42.8
Metal Tube 50 cm, 8 cm diameter	1	\$9.32
Wheels	2+1	\$14
18650 Li-ion Battery + BMS	6+2	\$29.56+\$3.5
LN298N Motor Driver	1	\$2.08
MPU9250 IMU	1	\$8.59
LM2596 Buck Converter	2	\$1.74
Others	-	\$24.5
Total	-	\$293.92

Mechanical Design



Software Design



Deliverables

- Autonomous Vehicle
- Parking Area and Spot
- Software Suit as PyPI Package
- Test Document and Results
- User Manual and Assembly Instructions