[DDMR ROBOTIC]

Abstract

Nailatun Naja Nisrina

nailatunnaja@me.student.pens.ac.id

Differential Drive Mobile Robots (DDMRs) are two-wheeled robots with independently controlled wheels for versatile movement. Widely used in robotics, DDMRs offer simplicity, cost-effectiveness, and precise control. They navigate by adjusting wheel speeds, making them suitable for educational projects, material transport, and autonomous applications.

A Differential Drive Mobile Robot (DDMR) is a two-wheeled robotic system known for its simplicity and versatility. The design features independently controllable wheels, allowing the robot to move forward, backward, turn, and rotate in place by adjusting the speeds of each wheel separately. DDMRs are widely used in robotics for their cost-effectiveness, ease of control, and adaptability. They serve various purposes, including educational projects, exploration, and practical applications like material transport and automated cleaning. With the ability to integrate sensors and navigation algorithms, DDMRs can navigate environments autonomously, making them a popular choice in both educational settings and real-world scenarios due to their straightforward yet effective design.

Final System and Team Personnel (Insert Pictures)

Table of Contents

[1 Introduction and Initial Analysis 2](#_Toc149728741)

[1.1 Project Context 2](#_Toc149728742)

[1.2 Initial Thought Process 2](#_Toc149728743)

[2 Requirement Analysis and Specification 2](#_Toc149728744)

[2.1 User Requirements 2](#_Toc149728745)

[2.2 System Requirements 2](#_Toc149728746)

[2.3 Tools and Technologies 2](#_Toc149728747)

[3 Conceptual Design 2](#_Toc149728748)

[3.1 System Architecture 2](#_Toc149728749)

[3.2 Interface Design 2](#_Toc149728750)

[3.3 Control Algorithm Design 3](#_Toc149728751)

[4 Detailed Design and Development 3](#_Toc149728752)

[4.1 Component Design 3](#_Toc149728753)

[4.2 Coding and Implementation 3](#_Toc149728754)

[4.3 Integration 3](#_Toc149728755)

[4.4 Unique Features 3](#_Toc149728756)

[5 Testing, Evaluation, and Optimization 3](#_Toc149728757)

[5.1 Testing Strategy 3](#_Toc149728758)

[5.2 Performance Evaluation 3](#_Toc149728759)

[5.3 Optimization 3](#_Toc149728760)

[6 Collaboration and Project Management 3](#_Toc149728761)

[6.1 Teamwork Dynamics 3](#_Toc149728762)

[6.2 Project Management 3](#_Toc149728763)

[7 Conclusion and Reflection 3](#_Toc149728764)

[7.1 Project Summary 3](#_Toc149728765)

[7.2 Future Work 3](#_Toc149728766)

[7.3 Personal and Group Reflections 3](#_Toc149728767)

[8 Appendices 4](#_Toc149728768)

[8.1 Bill of Materials 4](#_Toc149728769)

[8.2 Electrical Wiring and System Layout 4](#_Toc149728770)

[8.3 Code Repository 4](#_Toc149728771)

[8.4 Additional Documentation 4](#_Toc149728772)

[9 References 4](#_Toc149728773)

# 1 Introduction and Initial Analysis

## 1.1 Project Context

## Differential Drive Mobile Robots (DDMRs) are a type of robotic platform widely used in various applications, such as logistics, warehouse automation, surveillance, and even education. DDMRs are characterized by their simplicity and maneuverability, making them suitable for a wide range of tasks. These robots typically consist of two motorized wheels, which can move independently, allowing them to navigate through different terrains and environments.

## The development and improvement of DDMRs are significant for several reasons: Efficiency and Automation: DDMRs can improve the efficiency of various industrial processes by automating tasks that would otherwise require human intervention. This can lead to cost savings and increased productivity, Safety: DDMRs are often used in scenarios where human presence could be risky or inconvenient. For example, they can be deployed in hazardous environments, reducing the risk to human workers, Education and Research: DDMRs serve as excellent educational tools, helping students and researchers understand the fundamentals of robotics, control systems, and navigation algorithms, Versatility: DDMRs are adaptable to different applications. They can be customized for specific tasks, such as delivering goods in warehouses, inspecting pipelines, or exploring unstructured environments.

## The problem statement for a DDMR project might be as follows: "Despite the versatility and potential of Differential Drive Mobile Robots (DDMRs), there are challenges in achieving precise navigation, obstacle avoidance, and efficient control. This project aims to address these challenges by developing a DDMR system that can autonomously navigate through complex environments while avoiding obstacles and reaching predefined targets."

## The objectives of the project can include the following: Design and Build: Design and build a differential drive mobile robot platform with appropriate sensors and actuators, Navigation and Control: Develop navigation and control algorithms that allow the robot to move accurately, avoid obstacles, and reach target locations, Sensing and Perception: Implement sensors, such as ultrasonic sensors, cameras, or lidar, to enable the robot to perceive its environment, Mapping: Create a mapping system that helps the robot understand its surroundings and build a map of the environment, Autonomous Operation: Enable the robot to operate autonomously and making decisions based on sensor data and navigation algorithms, Testing and Validation: Conduct extensive testing in various environments to validate the robot's performance and ensure it meets the project's goals, Documentation and Reporting: Document the design, algorithms, and results in a comprehensive report or research paper.

## In summary, a DDMR project aims to address the challenges related to precise navigation and control of mobile robots. Its significance lies in its potential to improve efficiency, safety, and versatility in various applications, while the problem statement and objectives guide the project toward achieving these goals.

## 1.2 Initial Thought Process

## Brainstorming and decision-making for a Differential Drive Mobile Robot (DDMR) project involve several stages, from initial idea generation to addressing potential challenges and opportunities. Here's how the process might unfold:

# **Stage 1: Idea Generation** - **Define the Problem**: Start by clearly defining the problem you want the DDMR to solve. For example, the problem could be efficient indoor navigation or automating material handling in a warehouse. - **Research and Literature Review**: Conduct a thorough literature review to understand existing solutions and technologies in the field of mobile robotics. Identify gaps and opportunities for improvement. - **Team Brainstorming**: Gather the project team to brainstorm initial ideas. Encourage open discussions and idea sharing. Consider questions like: What are the project's goals? What are the desired features and capabilities of the DDMR.

# **Stage 2: Preliminary Ideas - Conceptual Designs**: Create conceptual designs or sketches of the DDMR. Discuss potential shapes, sizes, and configurations of the robot. Consider factors like wheel design, chassis, sensors, and power supply. **- Sensor Selection**: Explore sensor options for perception. Common choices include ultrasonic sensors for obstacle detection, cameras for vision, or lidar for more advanced mapping and localization. **- Control and Navigation Algorithms**: Brainstorm possible control and navigation algorithms. Consider approaches like PID control, SLAM (Simultaneous Localization and Mapping), and obstacle avoidance techniques.

# **Stage 3: Decision-Making - Evaluation Criteria**: Define criteria for evaluating and prioritizing ideas. Criteria might include feasibility, cost, technical complexity, and alignment with project objectives. **- Idea Selection**: Evaluate preliminary ideas against the criteria and choose the most promising concepts. This might involve selecting a robot design, sensor configuration, and control strategy.- **Risk Assessment**: Identify potential challenges and risks associated with each idea. Consider factors like hardware limitations, software complexity, and budget constraints.

# **Stage 4: Analyzing Challenges and Opportunitie - Challenges:** Technical Challenges: Consider the complexity of implementing the selected ideas. Address challenges related to hardware and software integration. **-Cost Constraints**: Determine the project budget and evaluate how it might impact hardware and sensor choices. **- Software Development**: Developing robust navigation and control software can be a significant challenge. **- Testing and Validation**: Ensuring the DDMR operates safely and efficiently in real-world environments can be a challenge. **- Opportunities**:Learning and Skill Development: The project offers an opportunity for team members to gain valuable experience in robotics, control systems, and software development. **-** **Innovation**: DDMRs have a wide range of applications, presenting opportunities for innovation and marketable solutions. **- Collaboration**: Collaboration with experts in robotics, AI, and engineering can lead to valuable insights and partnerships.a

# **Research Contribution**: The project may contribute to the field of robotics and automation, leading to research opportunities.

# **Stage 5: Final Decision and Project Planning - Final Idea Selection**: Make a final decision on the DDMR's design, features, and control system. **- Project Planning**: Develop a detailed project plan with milestones, timelines, and resource allocation. **- Prototyping**: Begin prototyping and development. Address technical challenges as they arise and iterate on the design and control algorithms as needed.

# Throughout the project, it's essential to maintain open communication, document decisions, and adapt to unforeseen challenges. Regular team meetings and progress tracking will help ensure a successful DDMR project.2 Requirement Analysis and Specification

## 2.1 User Requirements

Designing a Graphic User Interface (GUI) and control system for a Differential Drive Mobile Robot (DDMR) involves understanding and incorporating various user requirements. These requirements can be categorized into functional and non-functional aspects. Here's an enumeration and elucidation of user requirements for the GUI and control system:

1. Intuitive User Interface:

Requirement: The GUI should be user-friendly and intuitive.

Elucidation: Users, including operators and administrators, should be able to interact with the system easily. The interface should have clear navigation, well-organized controls, and be visually appealing.

2. Real-time Robot Status:

Requirement: Provide real-time information on the robot's status.

Elucidation: The GUI should display critical information such as the robot's position, orientation, battery level, and sensor readings in real-time. This allows users to monitor the robot's behavior and respond to any issues promptly.

3. Manual Control Capability:

Requirement: Enable manual control of the robot.

Elucidation: Users should have the option to manually control the robot using the GUI. This can include controlling the robot's movement, adjusting its speed, and steering it in different directions.

4. Autonomous Navigation Settings:

Requirement: Allow users to set autonomous navigation parameters.

Elucidation: Users should be able to input navigation parameters, such as waypoints or destination points, for the robot to navigate autonomously. The GUI should provide an interface for setting up and modifying autonomous navigation tasks.

5. Sensor Data Visualization:

Requirement: Display sensor data in a comprehensible format.

Elucidation: Present sensor readings, such as proximity sensors, cameras, or other environmental sensors, in a visual and easily interpretable manner on the GUI. This helps users understand the robot's perception of its surroundings.

6. Error and Warning Alerts:

Requirement: Provide alerts for errors and warnings.

Elucidation: The GUI should notify users about any errors or warnings encountered by the robot, such as obstacle detection, communication issues, or low battery. Clear and actionable alerts enable users to address issues promptly.

7. Mapping and Localization Features:

Requirement: Include mapping and localization capabilities.

Elucidation: If the robot is equipped with mapping and localization sensors, the GUI should visualize the robot's map of the environment and its estimated localization on the map. This assists users in understanding the robot's spatial awareness.

8. Task Scheduling and Execution:

Requirement: Enable users to schedule and execute tasks.

Elucidation: Users should be able to schedule specific tasks for the robot, such as patrolling a certain area or performing inspections. The GUI should provide a scheduling interface and display the status of scheduled tasks.

9. Customizable Dashboard:

Requirement: Allow customization of the dashboard.

Elucidation: Users may have different preferences for the information displayed. The GUI should support customization, allowing users to configure the dashboard to show the metrics and controls most relevant to their tasks.

10. User Authentication and Access Control:

Requirement: Implement user authentication and access control.

Elucidation: Ensure that the GUI includes secure login mechanisms and assigns appropriate access levels to different users. This helps in controlling who can operate the robot and access specific functionalities.

11. Diagnostic and Maintenance Tools:

Requirement: Include diagnostic and maintenance features.

Elucidation: The GUI should offer tools for diagnosing issues, checking the health of components, and performing maintenance tasks. This contributes to the overall reliability and longevity of the robot.

12. Compatibility with Various Devices:

Requirement: Ensure compatibility with different devices.

Elucidation: The GUI should be accessible from various devices, such as desktop computers, tablets, or smartphones. This enhances the flexibility of control and monitoring options for users.

By addressing these user requirements, the GUI and control system for the Differential Drive Mobile Robot can provide a comprehensive and user-friendly platform for effective robot operation and management.

## 2.2 System Requirements

## 

## Designing a Differential Drive Mobile Robot (DDMR) involves specifying both technical and functional requirements to ensure the robot's optimal performance and the fulfillment of user needs. Here's a detailed breakdown of technical and functional requirements for a Differential Drive Mobile Robot:

## Technical Requirements:

## Motion Control System:

## Requirement: Implement a differential drive system for precise control.

## Elaboration: Design a robust motion control system that allows the robot to move forward, backward, and rotate about its axis by controlling the speed of individual wheels.

## Wheel Encoders:

## Requirement: Integrate wheel encoders for odometry.

## Elaboration: Use wheel encoders to measure the rotation of each wheel, enabling accurate odometry calculations for determining the robot's position and orientation.

## Motor Controllers:

## Requirement: Utilize motor controllers for precise motor control.

## Elaboration: Integrate motor controllers to regulate the speed and direction of the motors, allowing the robot to execute smooth and controlled movements.

## Sensors:

## Requirement: Incorporate sensors for environment perception.

## Elaboration: Integrate a suite of sensors, such as proximity sensors, cameras, or lidar, to enable the robot to perceive its surroundings and navigate autonomously.

## Communication Module:

## Requirement: Implement a reliable communication module.

## Elaboration: Use a communication module to facilitate data exchange between the robot and external systems, allowing for remote control, monitoring, and data transmission.

## Power Supply System:

## Requirement: Design an efficient and reliable power supply system.

## Elaboration: Incorporate a power supply system, such as rechargeable batteries, with sufficient capacity to support the robot's operations for extended periods.

## Charging Mechanism (if applicable):

## Requirement: Include a charging mechanism for autonomous charging.

## Elaboration: If the robot is expected to operate autonomously for extended periods, implement a charging mechanism that allows the robot to return to a charging station when necessary.

## Robust Mechanical Structure:

## Requirement: Design a durable and robust mechanical structure.

## Elaboration: Ensure that the robot's chassis, wheels, and other mechanical components are robust and capable of withstanding the stresses associated with different operating environments.

## Embedded Computing System:

## Requirement: Integrate an embedded computing system for onboard processing.

## Elaboration: Incorporate a computing system, such as a microcontroller or single-board computer, for processing sensor data, running control algorithms, and making autonomous decisions.

## User Interface (if applicable):

## Requirement: Include a user interface for manual control or monitoring.

## Elaboration: If the robot allows for manual control or requires monitoring by an operator, implement a user interface that provides a clear display of the robot's status and allows for user interaction.

## Functional Requirements:

## Autonomous Navigation:

## Requirement: Enable autonomous navigation capabilities.

## Elaboration: Implement algorithms that allow the robot to navigate autonomously, avoiding obstacles and following predefined paths or waypoints.

## Obstacle Detection and Avoidance:

## Requirement: Include obstacle detection and avoidance mechanisms.

## Elaboration: Integrate sensors to detect obstacles in the robot's path and develop algorithms to plan and execute movements to avoid collisions.

## Remote Control Capability:

## Requirement: Provide remote control capabilities.

## Elaboration: Allow users to remotely control the robot using a user interface, enabling teleoperation for specific tasks or scenarios.

## Path Planning:

## Requirement: Implement path planning algorithms.

## Elaboration: Develop algorithms for the robot to plan optimal paths considering its current position, destination, and the surrounding environment.

## Localization:

## Requirement: Implement localization mechanisms.

## Elaboration: Use sensor data and odometry information to determine the robot's precise location within its environment.

## Emergency Stop Mechanism:

## Requirement: Include an emergency stop mechanism.

## Elaboration: Implement a mechanism that allows for an immediate stop of all robot movements in case of emergencies or user intervention.

## Integration with External Systems:

## Requirement: Facilitate integration with external systems.

## Elaboration: Enable the robot to communicate with and receive commands from external systems, allowing for coordination with other devices or platforms.

## Dynamic Reconfiguration (if applicable):

## Requirement: Allow for dynamic reconfiguration of control parameters.

## Elaboration: Provide the ability to adjust control parameters, such as speed limits or sensor thresholds, dynamically based on the operating environment or user preferences.

## Task Execution:

## Requirement: Execute predefined tasks autonomously.

## Elaboration: Develop a system that allows users to define tasks for the robot, such as patrolling an area or inspecting specific locations, and ensure the robot can execute these tasks autonomously.

## User Feedback Mechanism:

## Requirement: Provide feedback to users.

## Elaboration: Implement mechanisms for providing feedback to users, whether through the user interface or other means, to keep operators informed about the robot's status and activities.

## By addressing these technical and functional requirements, the Differential Drive Mobile Robot can be designed to meet user expectations, perform reliably, and fulfill its intended tasks and functionalities.

## 2.3 Tools and Technologies

## Software:

## VSCode (Visual Studio Code):

## Justification: VSCode is a versatile and widely-used code editor with extensive support for multiple programming languages. It offers features like syntax highlighting, debugging, and extensions, making it suitable for programming the Arduino Mega 2560 and integrating with the other chosen software tools.

## Arduino IDE:

## Justification: Arduino IDE is the official integrated development environment for Arduino boards, including the Arduino Mega 2560. It simplifies the development and uploading of code to the microcontroller, making it accessible for users with varying levels of programming experience.

## Qt Designer:

## Justification: Qt Designer is a graphical user interface (GUI) design tool that integrates seamlessly with Qt, a popular C++ framework. It is used for designing and creating graphical interfaces, making it suitable for developing control interfaces for the differential drive mobile robot.

## Hardware:

## Arduino Mega 2560:

## Justification: The Arduino Mega 2560 is a microcontroller board that provides a sufficient number of digital and analog pins, making it suitable for controlling the differential drive mobile robot. Its ease of use and compatibility with Arduino IDE simplify the development process.

## MPU6050 (Accelerometer and Gyroscope):

## Justification: The MPU6050 is a sensor module that combines an accelerometer and gyroscope. It is suitable for measuring the robot's orientation and acceleration, providing essential data for navigation and control algorithms.

## Ultrasonic Sensor:

## Justification: Ultrasonic sensors are commonly used for distance measurement and obstacle avoidance. They provide reliable data for detecting objects in the robot's path, enabling effective navigation and collision avoidance.

## LiPo 3S Battery:

## Justification: A LiPo 3S (3-cell Lithium Polymer) battery is a lightweight and high-energy-density power source suitable for mobile robots. It provides a portable and rechargeable power supply for the Arduino Mega 2560, motors, and other electronic components.

## Motor and Encoder:

## Justification: Motors and encoders are essential for the differential drive system. Motors drive the wheels, and encoders provide feedback for precise control and odometry calculations. This combination enables accurate and controlled robot movement.

## Other Resources:

## Chassis and Wheels:

## Justification: The chassis and wheels form the physical structure of the robot. Choosing a well-designed chassis with suitable wheels contributes to stability, durability, and effective motion.

## Motor Driver (H-Bridge):

## Justification: An H-Bridge motor driver is necessary to control the direction and speed of the motors. It interfaces between the Arduino Mega 2560 and the motors, allowing for bi-directional control.

## Voltage Regulator:

## Justification: A voltage regulator ensures stable and regulated power supply to the Arduino Mega 2560 and other components. It prevents voltage fluctuations that could affect the performance of electronic components.

## Wireless Communication Module (e.g., Bluetooth or Wi-Fi Module):

## Justification: A wireless communication module facilitates remote control and communication with the robot. Bluetooth or Wi-Fi modules can be integrated for wireless data exchange and control.

## Breadboard and Jumper Wires:

## Justification: Breadboards and jumper wires are essential for prototyping and connecting electronic components during the development phase. They provide a convenient and flexible way to test and iterate on the robot's circuitry.

## By selecting these specific software tools and hardware components, you are leveraging a combination that is well-suited for the development of a differential drive mobile robot. The chosen microcontroller, sensors, and design tools align with the requirements of the project, offering a balance between functionality, ease of development, and compatibility.

## 2.4 Target specification

Tabel . Tabel caption.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Description | Measurement Metric | Target Value |
| Example | Detail of the feature | Units/Scale/Methodology | Specified Value |
| ... | ... | ... | ... |

# 3 Conceptual Design

## 3.1 System Architecture

## The high-level architecture of a Differential Drive Mobile Robot (DDMR) system encompasses various components, including the graphical user interface (GUI) and control logic. Below is an illustration of the high-level architecture:

## Differential Drive Mobile Robot Architecture

## Components and Description:

## User Interface (GUI):

## Description: The GUI serves as the user's interaction point with the differential drive mobile robot. It is responsible for displaying real-time information, control options, and feedback to the user. The GUI allows users to monitor the robot's status and issue commands.

## User Authentication and Authorization:

## Description: This component handles user authentication to ensure secure access to the system. It verifies user credentials and assigns appropriate roles and permissions. User authorization controls the level of access to robot control and monitoring features.

## Control Logic and Navigation Module:

## Description: The Control Logic and Navigation Module form the core of the DDMR system. The control logic interprets user commands from the GUI, processes sensor data, and generates control signals for the robot's motors. The navigation module is responsible for path planning, obstacle avoidance, and determining the robot's position.

## Communication Module:

## Description: The Communication Module facilitates communication between different system components. It manages data exchange between the GUI, control logic, and external devices. Communication may occur over wired or wireless connections, depending on the chosen communication protocols.

## Sensor Interface:

## Description: The Sensor Interface connects the various sensors used by the robot, such as the MPU6050 (accelerometer and gyroscope) and ultrasonic sensors. It collects sensor data, which is then processed by the control logic and used for navigation and control decisions.

## Motor Control Unit:

## Description: The Motor Control Unit regulates the speed and direction of the robot's motors based on commands received from the control logic. It interfaces with motor drivers and encoders to ensure accurate motor control and feedback.

## Encoder Feedback:

## Description: Encoder feedback provides information about the rotation of the robot's wheels. This feedback is crucial for odometry calculations, enabling the system to estimate the robot's position and make adjustments for precise movement.

## Robot Dynamics and Kinematics:

## Description: The Robot Dynamics and Kinematics component models the physical dynamics and kinematics of the differential drive mobile robot. It takes into account the wheel velocities, dimensions, and other parameters to predict the robot's motion.

## Mapping and Localization Module:

## Description: The Mapping and Localization Module creates and maintains a map of the robot's environment. It also estimates the robot's position within this map using sensor data and odometry information. Mapping and localization are crucial for autonomous navigation.

## Emergency Stop and Safety Systems:

## Description: The Emergency Stop and Safety Systems provide mechanisms for halting the robot's movement in emergency situations. This may include hardware or software-based emergency stop buttons and safety protocols to ensure the robot's safe operation.

## Power Management and Battery Monitoring:

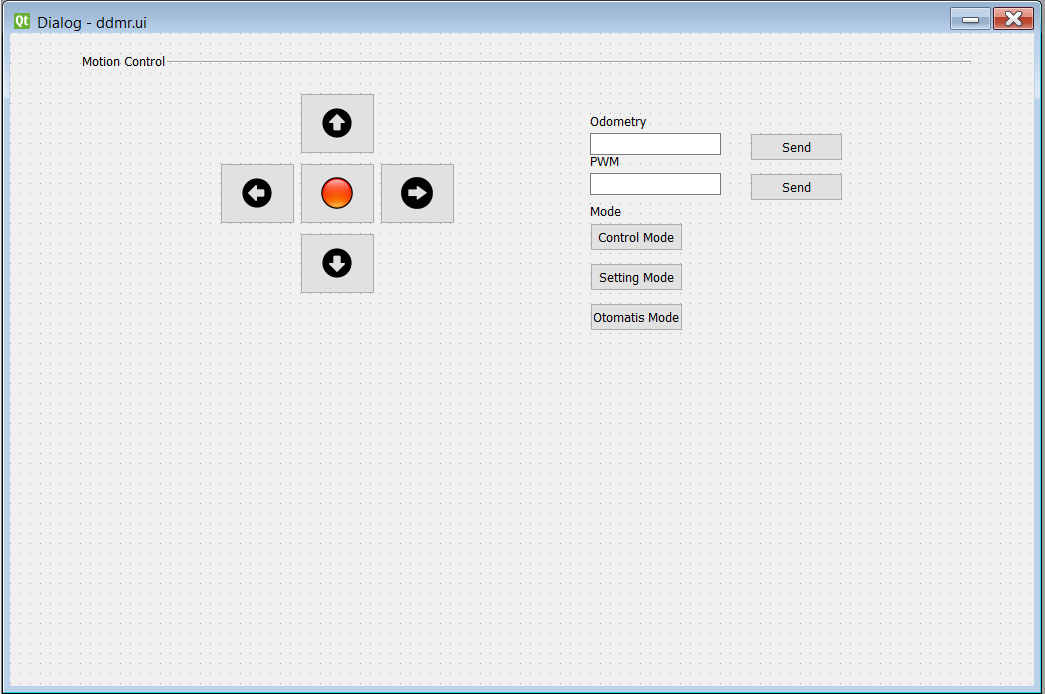
## Description: The Power Management and Battery Monitoring component oversees the robot's power supply. It manages battery usage, monitors the LiPo 3S battery's state, and may implement power-saving strategies to optimize energy consumption.

## Logging and Diagnostics:

## Description: Logging and Diagnostics record system activities, errors, and warnings. This component aids in troubleshooting, system analysis, and performance monitoring. Logging may include information about sensor readings, control decisions, and user interactions.

## By integrating these components, the high-level architecture ensures a well-coordinated and efficient operation of the differential drive mobile robot. The GUI provides a user-friendly interface for monitoring and control, while the control logic, navigation module, and associated components work together to enable accurate and responsive robotic movements. Communication, sensors, and safety features contribute to the overall reliability and functionality of the DDMR system.

## 3.2 Interface Design



## 3.3 Control Algorithm Design

# Designing control algorithms and data processing workflows for a differential drive mobile robot involves creating a system that enables the robot to navigate, avoid obstacles, and execute tasks efficiently. Below is an outline of the design considerations for control algorithms and data processing workflows:

# Control Algorithms:

# Motor Control:

# Objective: Regulate the speed and direction of the robot's motors.

# Design Considerations: Implement a proportional-integral-derivative (PID) controller or other motor control algorithms to translate high-level commands from the navigation module into motor control signals. Ensure smooth acceleration and deceleration to prevent jerky movements.

# Odometry and Dead Reckoning:

# Objective: Estimate the robot's position based on wheel encoder feedback.

# Design Considerations: Use odometry calculations to estimate the robot's displacement and heading based on wheel encoder data. Implement dead reckoning to update the robot's position as it moves, incorporating changes in wheel distances and angles.

# Navigation and Path Planning:

# Objective: Plan paths and navigate the robot autonomously.

# Design Considerations: Utilize algorithms such as A\* or Dijkstra's to plan optimal paths from the robot's current position to a target location. Implement obstacle avoidance mechanisms, considering sensor inputs to dynamically adjust the path and avoid collisions.

# PID Control for Angular Velocity:

# Objective: Control the robot's angular velocity for accurate turns.

# Design Considerations: Implement a PID controller to regulate the angular velocity of the robot during turns. Fine-tune the PID parameters to achieve smooth and precise rotational control.

# Emergency Stop Mechanism:

# Objective: Provide a mechanism to halt the robot's movement in emergency situations.

# Design Considerations: Implement an emergency stop button or protocol that immediately disables motor control and brings the robot to a safe stop. Ensure fail-safes are in place to handle critical situations.

# Proximity-Based Speed Adjustment:

# Objective: Adjust robot speed based on proximity to obstacles.

# Design Considerations: Use data from proximity sensors (e.g., ultrasonic sensors) to dynamically adjust the robot's speed when approaching obstacles. Slow down or stop when obstacles are detected in close proximity.

# Data Processing Workflows:

# Sensor Data Preprocessing:

# Objective: Prepare raw sensor data for further analysis.

# Design Considerations: Implement filtering, noise reduction, and calibration steps to preprocess sensor data. Ensure the data is accurate and reliable for subsequent processing.

# Real-time Sensor Fusion:

# Objective: Combine data from multiple sensors for a comprehensive perception.

# Design Considerations: Employ sensor fusion techniques to integrate data from sensors like the MPU6050 (accelerometer and gyroscope) and ultrasonic sensors. This enhances the robot's perception capabilities by considering multiple sensor inputs.

# Environment Mapping:

# Objective: Build and update a map of the robot's environment.

# Design Considerations: Implement Simultaneous Localization and Mapping (SLAM) algorithms or use pre-built SLAM libraries to create and update a map of the robot's surroundings. Incorporate sensor data and odometry information for accurate mapping.

# Data Integration for Localization:

# Objective: Use sensor data to estimate the robot's position.

# Design Considerations: Integrate data from wheel encoders, IMU, and other sensors to improve localization accuracy. Implement algorithms such as Extended Kalman Filters (EKF) for robust sensor fusion.

# Real-time Feedback to GUI:

# Objective: Provide real-time information to the user interface.

# Design Considerations: Establish a communication channel between the control logic and GUI to relay real-time information. Display robot status, sensor readings, and navigation updates on the GUI for user monitoring.

# Event Logging and Diagnostics:

# Objective: Record system events and diagnostics for analysis.

# Design Considerations: Implement a logging mechanism to record important system events, errors, and warnings. This aids in post-analysis, debugging, and system improvement.

# Task-Specific Data Processing:

# Objective: Process data for specific robot tasks or applications.

# Design Considerations: Tailor data processing workflows to the robot's intended tasks. For example, if the robot has computer vision capabilities, implement image processing algorithms for object recognition or localization.

# Wireless Communication Handling:

# Objective: Manage wireless communication with external devices.

# Design Considerations: Implement protocols for wireless communication (e.g., Bluetooth or Wi-Fi) to receive commands from external sources. Ensure secure and reliable data exchange between the robot and external systems.

# Battery Monitoring and Power Management:

# Objective: Monitor battery status and manage power consumption.

# Design Considerations: Integrate algorithms to monitor the LiPo 3S battery voltage and current. Implement power management strategies to optimize energy usage and provide warnings or take actions when the battery level is low.

# By addressing these design considerations, the control algorithms and data processing workflows form a robust foundation for the operation of a differential drive mobile robot. These elements contribute to accurate navigation, obstacle avoidance, and overall system reliability.

# 4 Detailed Design and Development

## 4.1 Component Design

Delving into the design of individual components, modules, and functionalities for a Differential Drive Mobile Robot (DDMR) involves breaking down the system into specific elements. Below is an exploration of key components and functionalities:

1. Motor Control Module:

Objective: Regulate the speed and direction of the robot's motors.

Design:

Implement a PID control algorithm for each motor to achieve smooth and precise control.

Interface with the Arduino Mega 2560's PWM outputs for motor speed control.

Employ an H-Bridge motor driver for bidirectional control.

2. Wheel Encoders:

Objective: Provide feedback for odometry and precise control.

Design:

Connect wheel encoders to the Arduino Mega 2560 to measure wheel rotations.

Implement an interrupt-based system to capture encoder pulses.

Use encoder feedback for odometry calculations to estimate the robot's position and heading.

3. Sensor Integration:

Objective: Gather data for navigation, obstacle avoidance, and environment perception.

Design:

Integrate the MPU6050 (accelerometer and gyroscope) to measure acceleration and rotation.

Connect ultrasonic sensors for obstacle detection and distance measurement.

Implement software to read sensor data and convert it into usable information for control algorithms.

4. Navigation and Path Planning:

Objective: Enable the robot to navigate autonomously and plan optimal paths.

Design:

Implement path planning algorithms (e.g., A\* or Dijkstra's) to find optimal routes.

Utilize sensor data (e.g., ultrasonic) for obstacle detection and dynamic path adjustment.

Develop algorithms for waypoint navigation and following predefined paths.

5. Bluetooth Communication Module:

Objective: Facilitate wireless communication for remote control and data exchange.

Design:

Integrate a Bluetooth module (e.g., HC-05) with the Arduino Mega 2560.

Develop a communication protocol for sending/receiving control commands and receiving status updates.

Enable secure and reliable data exchange between the robot and external devices.

6. Emergency Stop Mechanism:

Objective: Provide a quick and reliable mechanism to halt the robot's movement.

Design:

Implement an emergency stop button as a physical or software-based switch.

Ensure immediate cessation of motor control upon activation.

Include fail-safes to handle critical situations and prevent unintended stops.

7. Mapping and Localization Module:

Objective: Create and maintain a map of the robot's environment and estimate its position.

Design:

Utilize SLAM algorithms (e.g., GMapping or Cartographer) for mapping.

Combine sensor data and odometry information for accurate localization.

Implement a map update mechanism to adapt to changes in the environment.

8. Power Management System:

Objective: Monitor battery status and manage power consumption.

Design:

Integrate voltage and current sensors to monitor the LiPo 3S battery.

Implement algorithms for efficient power management, considering sleep modes and low-power states.

Provide real-time battery status updates to the user interface.

9. User Interface (GUI):

Objective: Enable user interaction, monitoring, and control.

Design:

Develop a graphical user interface using Qt Designer.

Display real-time data, including sensor readings, battery status, and robot position.

Include control elements for manual operation and task definition.

10. Logging and Diagnostics:

Objective: Record system events, errors, and warnings for analysis.

Design:

Implement a logging mechanism to record critical events, sensor readings, and control decisions.

Create a diagnostic system that logs errors and provides insights for troubleshooting.

Include timestamped logs for post-analysis.

11. Wireless Communication Handling:

Objective: Manage wireless communication with external devices.

Design:

Develop a robust communication protocol to handle data exchange.

Include error-checking mechanisms to ensure data integrity.

Implement secure and encrypted communication, especially for critical commands.

12. Chassis and Mechanical Design:

Objective: Provide a stable and durable physical structure.

Design:

Select or design a chassis that accommodates all components.

Ensure proper mounting for sensors, motors, and the power system.

Consider factors like weight distribution and ground clearance for stability.

13. Real-time Feedback and Data Processing:

Objective: Provide real-time information to the user interface and process data.

Design:

Implement data processing workflows to handle sensor data.

Develop algorithms for real-time feedback to the GUI.

Use multithreading or parallel processing to ensure responsiveness.

14. Task Execution and Autonomous Behavior:

Objective: Enable the robot to execute predefined tasks autonomously.

Design:

Develop a task execution module that interprets high-level commands.

Implement algorithms for dynamic reconfiguration based on the environment.

Include routines for returning to a charging station or responding to specific events.

15. Dynamic Reconfiguration (if applicable):

Objective: Allow for dynamic adjustments in control parameters.

Design:

Develop a mechanism for adjusting control parameters based on environmental changes or user preferences.

Include user-configurable settings for speed limits, obstacle detection thresholds, etc.

16. User Feedback Mechanism:

Objective: Provide feedback to users about the robot's status and activities.

Design:

Implement visual and/or auditory feedback mechanisms on the GUI.

Include status indicators, progress bars, or alerts to convey information.

Ensure user-friendly messages and notifications.

17. Safety Features:

Objective: Integrate features to enhance overall system safety.

Design:

Implement fail-safes for critical components.

Include collision detection and response mechanisms.

Consider incorporating computer vision for advanced safety features.

18. Documentation and Manuals:

Objective: Provide comprehensive documentation for system replication and maintenance.

Design:

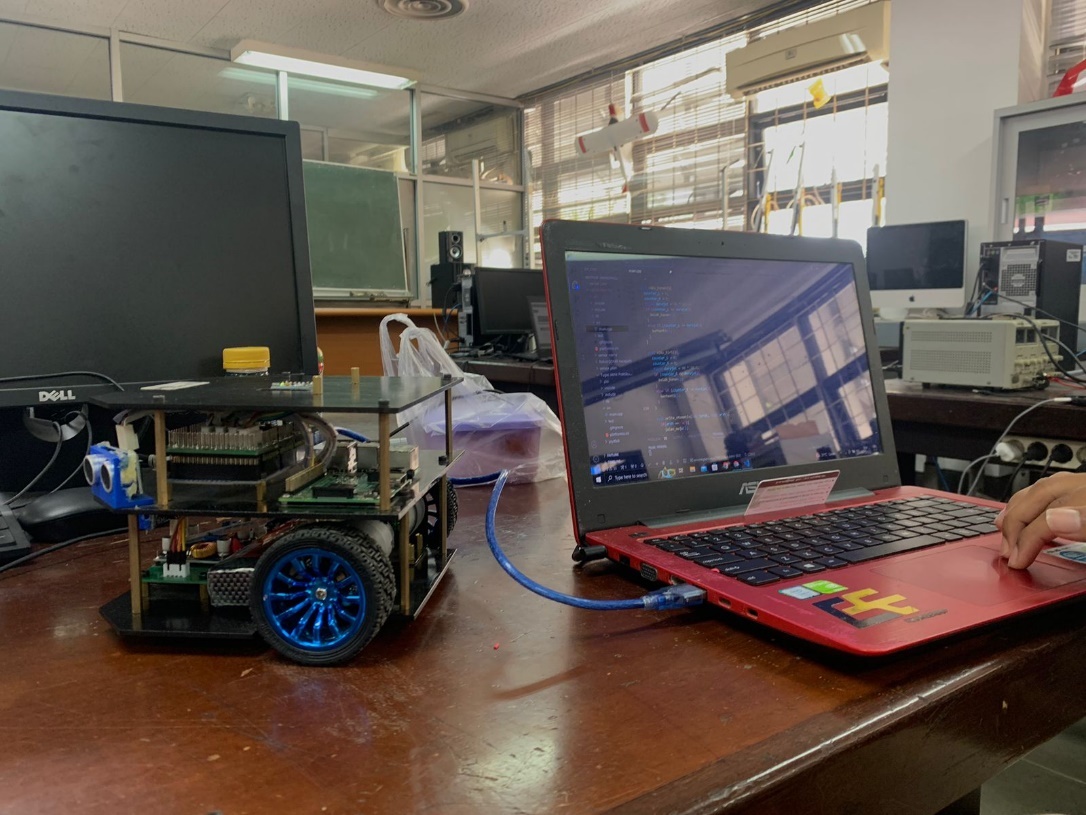
Create detailed schematics, wiring diagrams, and assembly instructions.

Develop user manuals covering system operation, troubleshooting, and maintenance.

Include comments in the code for ease of understanding.

This detailed breakdown covers various aspects of the design for a Differential Drive Mobile Robot, encompassing both hardware and software components. Each section can be expanded further with specific algorithms, code snippets, and detailed technical documentation as per the needs of the project

## 4.2 Coding and Implementation



## 4.3 Integration

Discuss the integration of GUI with the control system, and among different system components.

## 4.4 Unique Features

Highlight any novel features, optimizations or technologies employed.

# 5 Testing, Evaluation, and Optimization

## 5.1 Testing Strategy

Describe the testing methodologies, cases, and tools used. Emphasize on how the testing validates the targets specified in Section 2.4.

## 5.2 Performance Evaluation

Evaluate the system performance against the defined requirements and objectives. Include a comparative analysis with the targets specified in Section 2.4, illustrating how well the system meets or exceeds these targets.

## 5.3 Optimization

Discuss any optimizations made to enhance system performance and user experience.

# 6 Collaboration and Project Management

## 6.1 Teamwork Dynamics

Reflect on the collaborative endeavor, roles, and contributions of team members.

## 6.2 Project Management

Document the project timeline, milestones, and management practices adopted.

# 7 Conclusion and Reflection

## 7.1 Project Summary

## The DDMR robot movement development project has successfully overcome a number of key challenges in an effort to improve its movement capabilities. Improvements in motion quality were achieved through algorithm optimization, hardware adjustments, and increased system responsiveness. Through systematic testing steps, we ensure that the robot is able to consistently and optimally perform movements in various contexts. The results of this project mark important progress in the development of more adaptive and responsive humanoid robots.

## 7.2 Future Work

## Future developments could focus on the integration of the latest sensor technology, the development of smarter control algorithms, and a deeper understanding of how we want mobile robots to operate. The application of DDMR robots can also be extended to various important sectors in daily life, such as health care, education, and exploration of hazardous environments.

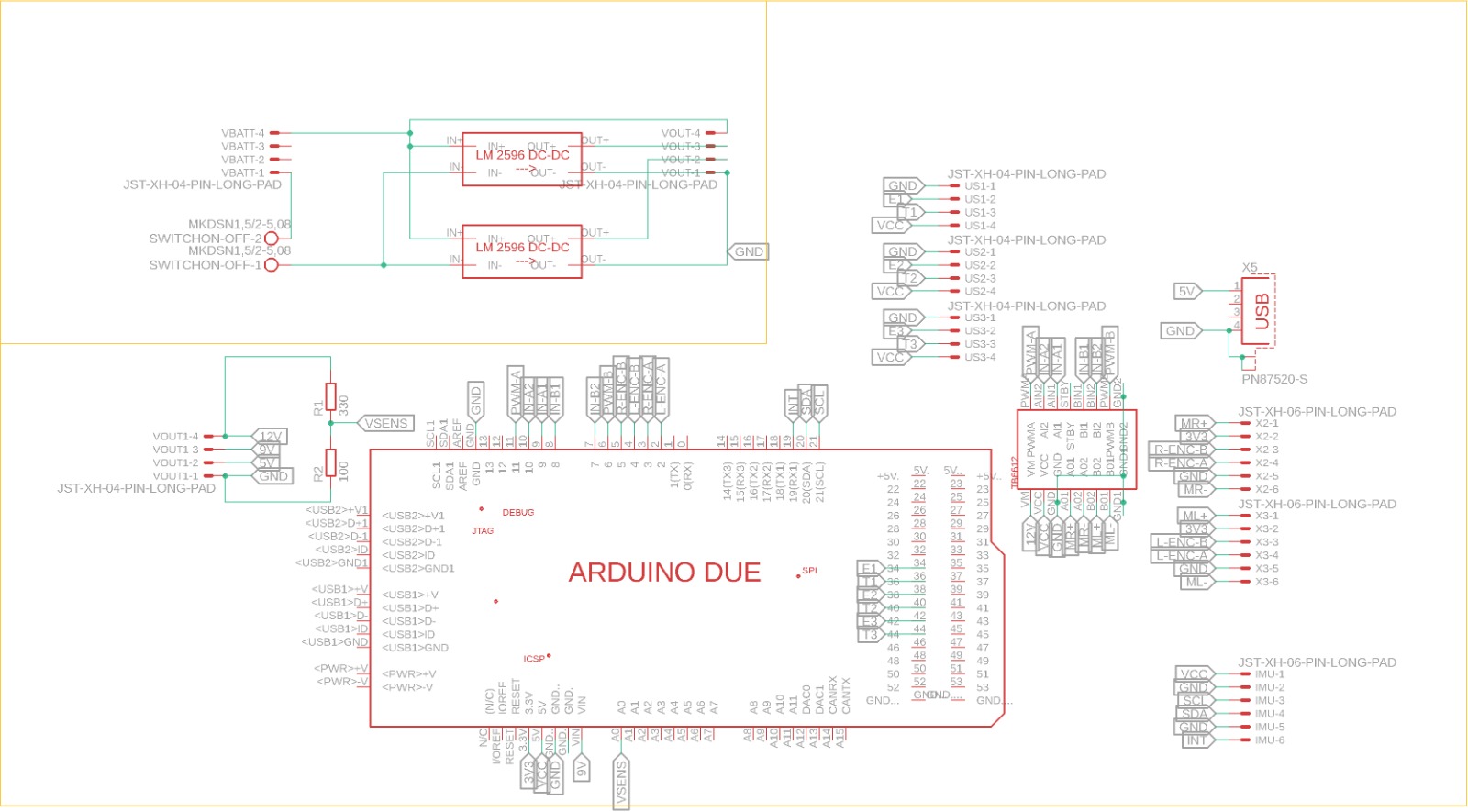
## 7.3 Personal and Group Reflections

## During the course of this project, we were faced with a number of significant challenges. However, through solid teamwork and effective project management, we managed to overcome every obstacle. This experience taught us a lot about the complexity of developing DDMR robots, both from a technical and non-technical perspective. In our personal reflections, we emphasize how important cooperation, patience and dedication are in achieving common goals. Taken together, these projects have not only broadened our horizons, but also prepared us to face future challenges in the development of robotics technology.8 Appendices

## 8.1 Bill of Materials

The detailed GUI display for controlling the movement of a DDMR robot using Python involves several key elements. The main part includes the graphical interface itself, control buttons for various gestures and functions. In the context of cost, components such as ultrasonic sensors and motors as actuators. Additional investment may be required for Python GUI development learning resources, including books, online courses, and official documentation. By integrating all these elements, the GUI display can provide intuitive and informative control over the movement of the DDMR robot, while keeping cost efficiency under control.

## 8.2 Electrical Wiring and System Layout



## 8.3 Code Repository

## <https://github.com/anh0001/Robot-DDMR-GUI-Control/tree/main>

## 8.4 Additional Documentation

# 9 References

Cite all references, tools, libraries, and external resources used in the project.