RMIT University

Final Project

COSC3070 – Programming Autonomous Robots

Final Project

Student Name: ______________________________

Student ID: ______________________________

Submission Date: __________________________

Lecturer: ______________________________

School of Science, Engineering & Technology

RMIT University

24-5-2025

Abstract

This report presents the development and evaluation of an autonomous robot for obstacle avoidance and navigation tasks, as part of Final Project for COSC3070 – Programming Autonomous Robots at RMIT University. The project involved designing a control algorithm using sensor feedback, implementing the logic on embedded hardware, and testing the robot in a simulated environment. The methodology combined reactive behavior with a simple rule-based system for path planning. Results demonstrate successful autonomous operation in controlled scenarios, with room for improvement in real-time adaptability. The findings highlight the importance of sensor calibration and software-hardware integration in robotics.

Contents

1 Introduction

Autonomous robots play a critical role in the advancement of smart systems across industries such as logistics, transportation, and manufacturing. These systems rely on embedded intelligence to perceive their environment, make decisions, and act without human input. A fundamental capability in such systems is the ability to follow predetermined paths while adapting to dynamic environments through obstacle avoidance and signal interpretation.

This project aims to replicate these core functionalities using the mBot2 Neo robot, a programmable platform equipped with sensors. The robot was tasked with navigating a complex maze environment by accurately following either a white or yellow line, detecting and responding to visual stop/go signs, and avoiding static and dynamic obstacles. The environment mimics real-world traffic scenarios where autonomous systems must balance goal-directed behavior with real-time responsiveness.

The report documents the approach taken to achieve these tasks, the challenges encountered, and the evaluation of the robot's performance within the given constraints.

2 Related Work

Discuss existing research, previous projects, or technologies relevant to your robot or programming approach. Highlight differences between your work and existing solutions.

3 Methodology

The autonomous navigation behavior of the mBot2 Neo robot in this project is driven by a combination of sensor inputs and control algorithms designed for line following, obstacle avoidance, and color-based decision making.

3.1 mBot2 Neo Robot Overview

The mBot2 Neo is a programmable mobile robot equipped with various sensors and actuators suitable for autonomous navigation tasks. It includes a quad RGB sensor for color detection, an ultrasonic distance sensor for obstacle avoidance, and motor encoders for precise movement control. The robot also features a built-in camera, an LED matrix, and is powered by a CyberPi microcontroller, which supports Python programming and real-time sensor integration.

This platform offers flexibility and simplicity for implementing sensor-based decision-making, path following algorithms, and reactive behaviors required for this project.

3.2 Line Following using Proportional Control Algorithm (PCA)

To ensure smooth and accurate path following, a Proportional Control Algorithm (PCA) was implemented. This algorithm adjusts the robot's steering proportionally to the error detected between the center of the robot and the center of the path. The error is calculated based on the intensity readings from the quad RGB sensor mounted on the front of the robot.

When the robot deviates from the path, the PCA applies a corrective steering adjustment proportional to the deviation. This allows for responsive and stable line following, even through curves and slight shifts in path alignment.

3.3 Color Detection using Quad RGB Sensor

A quad RGB sensor is used to detect both the path color and visual indicators (traffic signs) embedded in the maze. The sensor differentiates between multiple colors (white, yellow, red, green), and corresponding behaviors are triggered based on the detected color:

- **White:** Normal speed — this is the default behavior when no special signs are detected.

- **Yellow:** The robot slows down to ensure cautious navigation.
- **Red:** The robot stops immediately and waits.
- **Green:** The robot resumes movement and increases speed on second detection.

This color-coded system simulates real-world traffic control mechanisms and ensures the robot can adapt dynamically to the environment.

3.4 Obstacle Detection using Ultrasonic Sensor

An ultrasonic sensor is mounted on the front of the robot to detect obstacles. It continuously measures the distance to objects ahead in real-time. When an object is detected within a predefined safety threshold, the robot halts and initiates an obstacle handling routine.

The avoidance behavior follows a multi-stage decision process:

- The robot first comes to a complete stop and waits for the obstacle to be removed.
- If the obstacle remains after 10 seconds, the robot assumes the path is blocked and performs a U-turn to find an alternative route.
- If the obstacle only partially covers the path and the remaining space is sufficient, the robot attempts to **drift or steer around** the obstacle and continue following the line.

This logic ensures the robot behaves intelligently in dynamic environments, minimizing manual intervention while maintaining a consistent navigation flow. The timing and spatial thresholds used in this process were calibrated through iterative testing to balance responsiveness with stability.

4 Implementation

Provide details about how the methodology was implemented. Mention programming language, tools, libraries, and hardware (e.g., sensors, actuators, microcontroller). Include code snippets or diagrams if needed.

5 Results, Evaluation & Discussion

Present the outcomes of your implementation. Evaluate how well your robot performed. Use tables, graphs, or figures as needed. Discuss what worked well, challenges faced, and potential improvements.

6 Conclusion

Summarize the main findings of your work. Restate the importance of your approach and suggest future improvements or applications.