
An-Najah National University

Faculty of Engineering

Computer Engineering Department

Graduation project

MazeMaster

Students:

Mohammad Salman – 11941339

Mohammad Najjar – 11941357

Supervisors:

D. Mohunnad Al-jabi

D.Sufyan Samara

Presented in partial fulfilment of the requirements for Bachelor degree in Computer Engineering.

Septemper 6,2024

Acknowledgment:

We would like to thank everyone who helped us to complete this project. First, we are very grateful to our supervisors, Dr. Muhannad Al-Jabi and Dr. Sufyan Samara, for their continuous support, advice and feedback. Without their help, the project would not have been completed.

We would also like to thank our family and friends for their continuous support.

Finally, we appreciate everyone who provided us with any kind of assistance, which enabled us to complete our work successfully.

DISCLAIMER:

This report was written by students at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Contents

List of Figures

Abstract:

This project aims to design a maze solving robot that operates with left wall-following algorithm . The robot is built with a combination of hardware components, including DC motors, ultrasonic sensors, a microcontroller, and some other sensors and electrical components to navigate and solve mazes .

The left wall-following algorithm ensure that the robot is continuously follows the left wall of the maze until it reaches the goal . This method is efficient for structured maze environments where the goal is to find the black carpet at the end of maze.

A significant feature of the robot is the integration of the web interface that allows the user to control the robot remotely . The web interface enables the user to start , stop the robot with simple commands , providing flexibility in operation.

the interface displays the temperature around the robot . This temperature monitoring capability can be useful for some applications , such as informing the user of the situation inside the maze (for rescue applications) , detecting heat sources or ensuring that the robot is operating within safe temperature limits.

The robot is equipped with an ultrasonic sensor array to detect walls and obstacles, ensuring to navigation through the maze without hitting the walls . The left wall-following algorithm is implemented in the robot's microcontroller , allowing it to make decisions on turns and movements based on the distance to the walls . The web interface is designed for ease of use , with responsive controls and ambient temperature display.

This project demonstrates the integration of autonomous robotics with the remote control and monitoring capabilities , making it a versatile platform for maze- solving tasks and other applications where autonomous navigation and environmental sensing are very important.

We reach a point that we design this robot and apply the left wall follower algorithm and the robot performance was accepted for it's tasks.

1. Introduction:

1.1 General Background

Robotics has grown to encompass the construction of robots as well used in automation systems. Maze-Solving (finding a way through maze to reach the target or exit) is one of the tedious problem in Robotics. Maze-solving robot is act as tool for the variety of purpose such maz_ingiant or puzzle (entertainment & education) to real life problems in search and rescue mission, industrial Automation and exploration. In the context of maze-solving robots, one popular method is the left wall-following algorithm, where by which a robot will trace its way out of a given labyrinth by keeping itself in close proximity to these walls. This project goes out to create one such maze solving robot with the extra feature of Web Interface for remote control and surroundings monitoring.

1.2 Objectives of the Work

The primary objective of this project is to design a maze-solving robot that applies the left wall-following algorithm . The robot should be able to navigates autonomously through various maze structures . Additionally , the project aims to create a web interface that allows users to start and stop the robot remotely . another key objective is to integrate a temperature sensor with the robot , enabling the web interface to display the temperature around the robot . The ultimate goal is to demonstrate a functional and interactive robot that can solve mazes while providing remote control and environmental monitoring capabilities.

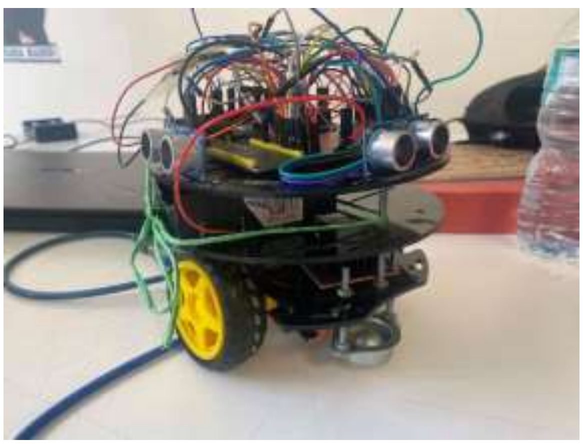

Figure 1: robot

1.3 Significance of the Work

The significance of this work lies in its combination of autonomous navigation and remote monitoring, making it useful in different areas . In industries , robots that can move through complex environments on their own and provide information about their surroundings are very valuable . The demand for smart robots is growing in fields like manufacturing, logistics, and security . By adding temperature sensing and web interface control, this project meets these needs, showing a robot is not only solve mazes but also adjust to different situations where remote control and knowing the environment are essential.

1.4 Organization of the Report

This report is organized into several chapters , each covering a different aspect of the project . Chapter 1 provides an introduction, outlining the background , objectives, significance, and organization of the work . chapter 2 covers the theoretical background, exploring existing methods and technologies used in maze-solving robots. chapter 3 will talk about methodology and shows the tools, standards and specifications , and constrains was faced in the project work . Chapter 4 will shows the results and what we built and done. Chapter 5 presents the discussion , it will discussed what if the problem solved or not and how . Finally, Chapter 6 concludes the report and suggests future work and improvements for the project.

2. Theoretical Background and Previous Work:

2.1 Introduction to Maze-Solving Robots

Maze-solving robots have been a popular topic in robotics research and development for many years . the idea of building a robot can navigates through a maze and find the exit began in the early 1970s . these robots were initially used for competitions and education, allowing students to test their programming and engineering skills . over time , maze solving robots became more famous , with different algorithms being developed to help the robots to solve the mazes more efficiently.

2.2 Common Maze-Solving Algorithms

There are several algorithms that robots use to solve mazes . each algorithm has its own approach and is suitable for different types of mazes . some of the most common maze-solving algorithms include :
Depth-First Search (DFS): This algorithm explores as far as possible along each branch before backtracking.it's like a robot trying every possible path until it finds the exit. while effective,it can be slow in complex mazes and it requires an accurate motors.

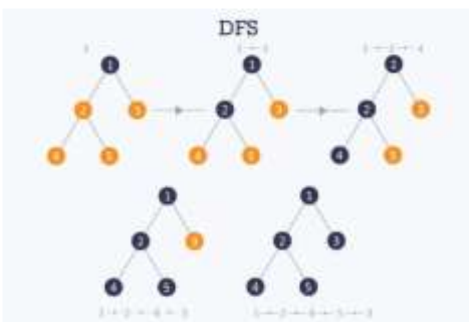

Figure 2: Depth-First Search

Breadth-First Search (BFS) : This algorithm explores all possible paths level by level. it is like a robot checking all the neighboring cells before moving further. BFS guarantees finding the shortest path to the exit , it requires more memory , it slower than BFS , and also requires an accurate motors.

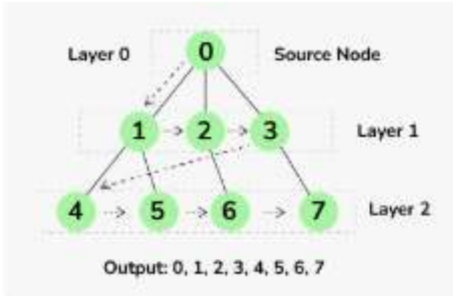

Figure 3: Breadth-First Search

Flood Fill : This algorithm is often used in maze solving competitions. The robot explores the maze and "floods" it with information, gradually marking distances from the start or goal. The robot uses this information to find the shortest path.

Figure 4: Flood Fill

A (A-Star) : it is a more advanced algorithm that uses both the distance already traveled and an estimate of the remaining distance to find the shortest path efficiently.
the previous algorithms needs complex codes to applies , expansive components , or previous knowledge about the maze so its not very popular in maze solver robots , so lets talk about most popular algorithm which is left wall follower.

2.3 The Left Wall Follower Algorithm

The left wall follower algorithm is one of the simplest and most intuitive maze solving methods . it is based on the idea that if a robot continuously follows the left wall of a maze, it will eventually find the exit , assuming the maze is simply connected (all walls are connected and the maze has no isolated sections).

How it works :

The robot starts by positioning itself at the maze entrance and then moves forward, keeping its left side close to the wall.

If the robot encounters a turn , it checks if it can continue following the left wall . If it can, it turns left; if not, it continues straight or turns right.

The robot continues this process , always prioritizing left turns , until it finds the exit.

Why it works :

The left wall follower algorithm works because it treats the maze as a connected space. by always following one wall , the robot avoids getting lost in loops and ensures that it eventually reaches the exit. however ,this method does not guarantee the shortest path.

Limitations :

left wall follower algorithm may not work in mazes that have isolated sections or if the robot starts in the middle of the maze. additionally, it may not find the shortest path to the exit.

3. Methodology:

This chapter details the materials, methods, and procedures incorporated in the design in a way that more experienced workers could add to this work to achieve comparable results. This also comprises the engineering standards applied and the design constraints, tools, and components used in the project.

3.1 Standards and Specifications (Codes):

In this project, several engineering standards were followed to ensure the robot's design and operation met industry requirements:

IEEE 802.11: This standard is used for wireless communication between the robot and the web interface. Remote control and real-time observation of the environment temperature by a robot have been implemented using a standard ESP32 microcontroller.

IEC 62133: Safety requirements for batteries have been considered in the choice and use of the power source for the robot, which helps to prevent hazards such as overheating and provides safe battery handling.

ISO 13482 (Safety Requirements for Personal Care Robots): Although this robot is not intended to have direct interaction with people, the general safety practices of this standard are conformed to in order to safely operate it in a place where the robot might be used.

3.2 Constraints

Economy:

Budget constraints: This project has been designed under a very low budget. Based on cost-effectiveness, components were selected in order to make the whole setup feasible without impairing functionalities of the robot. For example, inexpensive and versatile central controllers for this purpose are Arduino Mega and ESP32.

Environment:

-Power consumption: This robot's energy efficiency is designed with great attention to power sources. Other strategies applied in designing the robot for low power are having the DHT11 temperature sensor and ultrasonic sensors.

Project Nature and External Factors:

- Movement and External Interference: This was a problematic area with this project; the robot had to be a movable device. In that light, it can be influenced by several types of external factors that might interfere in completing the task—environmental conditions, obstacles, uneven surfaces, among other things. The dynamism the project carried along needed constant adjustments to assure all would work out smoothly.

Moreover, space was not much for a suitable place to work on and test the robot, at times having to share the working place with others, thereby cutting down the efficiency of testing.

-DC Motor Inconsistencies: Another problem was the inconsistent behavior of the DC motors. Being in nature, one could switch to a higher speed than the other, hence causing the robot to move out of course. What proved to be a task that really needed careful calibration and frequent adjustments was synchronizing and managing these inconsistencies between the two.

Time Constraints:

-Ultrasonic Sensor Constraints: The robot's performance could be improved by adding two additional ultrasonic sensors positioned at -45 degrees (left front) and +45 degrees (right front). However, adding these sensors introduced interference issues, as the ultrasonic sensors' signals overlapped and produced inaccurate distance readings. This interference worsened the robot's performance, leading to the decision to use only three sensors, despite their lower accuracy.

-Time Constraint: Time was a major limitation in the completion of this project. Besides working on this design, my partner and I were taking other university subjects, projects, and exams that we needed to attend to. Juggling these multiple demands really made time for the robot project quite short, placing added pressure on completion of work efficiently and effectively within the given timeframe.

3.3 Tools and Components

-Ultrasonic Sensors (left , front, right): Used in the detection of obstacles and provided support for the robot to move through the maze by offering distance measurements.

Figure 5: Ultrasonic Sensors

-ESP32: A Wi-Fi-enabled microcontroller to remotely control the robot using a web interface, and it sends the temperature data to the interface from the DHT11 sensor.

Figure 6: ESP32

-Arduino Mega: the main robot controller to control the motors by reading the input from the sensors based on the left wall follower algorithm.

Figure 7: Arduino Mega

-DC Motors (Left, Right): Give the robot its movement; they are controlled by the L298N motor driver and synchronized in their movement by optical encoders for better straightness.

Figure 8:DC Motor

-L298N Motor Driver: Gives the necessary power and direction signals to the DC motors.
DHT11 Temperature Sensor: To measure the temperature in the environment surrounding the robot. The data obtained is shown on the web interface to provide control of the environmental conditions.

Figure 9:L298N Motor Driver

-TCS3200 Color Sensor: To detect the carpet of black color at the end of the maze that signifies the indication of the robot reaching its goal.

Figure 10: TCS3200 Color Sensor

-GY-521 Sensor: This is an accelerometer and gyroscope module that works to measure the yaw angle, allowing the robot to take perfect turns.

Figure 11: GY-521 Sensor

-Optical Encoders (FC-03):Mounted on each wheel to calculate the rotation of the wheel.it sends this information for making the robot go in a straight line through PID control.

Figure 12:Optical Encoders

Figure 13: Chasis

-Buzzer: Active when robot detect black carpet.

Figure 14: Buzzer

-DHT11 sensor: Measure temperature in degrees celsius.

Figure 15: DHT11 sensor

-lithium Battery: Rechargeable battery.

Figure 16: lithium Battery

3.4 Procedures

Building the Robot:
The DC motors were attached to the chassis ,and optical encoders were mounted on the wheels (FC-03). The motors connected to the L298N motor driver, which was then connected to the Arduino Mega.
The front, left, and right ultrasonic sensors were mounted to detect obstacles, giving the distance measurements of the obstacle.
ESP32 has been integrated so that it can provide wireless communication and a web interface for remote control.

Programming : Programming was set up for the Arduino Mega to left wall follower algorithm, which is implemented to solve the maze.The GY-521 was calibrated to turn more accurately, and PID control was set using the optical encoders to maintain straight movement.The ESP32 was programmed to communicate on the web interface,sending and receiving data in the form of temperature reading from the DHT11 sensor and status update for remote control.

Testing: For the controlled maze environment, the robot was tested with obstacle detection to fine-tune the left wall follower algorithm. The web interface was tested to ensure smooth operation for the control of robot and temperature measurements.

Most of the project execution time was iterating between programming and testing.

4. Results and Analysis

This chapter shows the results obtained from testing the maze-solving robot and an analysis of the data collected in the course of a number of stages of the project. The collected data were used to evaluate the robot's performance while negotiating the maze, left wall-following algorithms, and the accuracy of its sensors and control mechanisms.

Ultrasonic Sensor Distance Calculation

The ultrasonic sensors were used to measure the distances to obstacles, The distance was computed using the normal time-of-flight formula:

Distance =(Speed of Sound×Time) / 2

Speed of sound is 343 m/s. The measured time by the sensor accounts for the round trip of the ultrasonic wave; hence, the actual distance to the obstacle is half of it.

Results:

The front sensor could sense the obstacles at a range from 0 cm to 200 cm with an accuracy of ±1 cm.

Both left and right sensors behaved identically, so the robot was able to maintain an almost equal distance from both walls.

Error Analysis :

External Factors : Little noise in the background and echoes off surroundings at times caused minor deviations in the distance measurements.

Surface Reflectivity: Non-reflective surfaces, such as soft or porous materials, sometimes resulted in less accurate readings.

Yaw Angle Calculation (GY-521 Sensor)

The Yaw angle was obtained from the GY-521 sensor while preserving the robot's orientation. The equation is shown below:

Yaw Angle =arctan*(GyroY/GyroX)

Sensor data was integrated over time to find the orientation of the robot, and with that information, PID control algorithm corrections were done to maintain the robot on course.

Results :

The GY-521 sensor recorded the orientation of the robot with an error of ±2 - ±15 degrees.

Error Analysis :

Drift: It experienced some drift over time and needed periodic recalibration.

Vibration: somtimes , the mechanical vibrations caused by the motors would introduce noise to the sensors and slightly shift the readings for yaw.

5. Discussion

In this chapter we discuss the significance of the results based on the goals of the project. The built robot was designed to navigate through the maze using left-wall-following algorithm autonomously using sensors and control systems to ensure accuracy and efficiency .

Achievements and Contributions :

The experimental results tells that integrating ultrasonic sensors with the GY-521 sensor for orientation along with PID control , successfully guided the robot through the maze and its variations. This suggests that systems like this one can be applied to various tasks, from industrial automation to exploration. The robot was able to adjust its direction through the maze as the task conditions changed, acheaving the primary goal of the project.

However, we faced some practical challenges During testing the robot , sometimes deviated from its course and collided with obstacles , particularly when one of the DC motors ran faster than the other . To fix this, we added a corrective maneuver where the robot would reverse slightly and then turn to realign itself . While this solution improved the robot's performance , motor speed variability still affected its overall efficiency.

Additionally, the ultrasonic sensors sometimes provided false readings, causing the robot to repeat the same path or fail to detect obstacles in time . This impacted the robot performance which make it slower sometimes . but even with these issues , the robot's ability to recover and continue navigating the maze demonstrated its resilience and adaptability .

Overall , the project was successful in creating a functional maze-solving robot , and it also highlighted key areas for improvement .The lessons learned from this experience will guide us for the future work .

Future Work :

The following enhancements should be made to further improve the robot’s capabilitis and address the limitations encountered :

Improved Motors : Upgrading to stepper motors would increase the accurecy and control over the robot's movement , reducing the impact of speed variability and improving navigation accuracy .

Sensor Upgrades : Replacing ultrasonic sensors with infrared sensors could enhance obstacle detection accuracy , reducing false readings and enabling the robot to navigate more complex environments .

Camera Integration : Adding a camera would allow manual control through live video from the camera displayed on a web interface . This would expand the robot's applications , such as maze scanning , remote surveillance , or tasks requiring real-time visual feedback .These improvements would further upgrade the robot , making it more accurate and adaptable for a wide range of real-world applications . The success of this project lays a strong foundation for future developments, with many opportunities for refinement and expansion .

6. Conclusions and Recommendations

Through this project , we learned that integrating multiple sensors and control systems can create an effective autonomous robot capable of navigating complex environments . However , the challenges faced such as motor speed variability and sensor inaccuracies highlight the need for further refinement to improve performance . These issues also underscore the importance of precise control and accurate sensing in robotics .

The experience gained from this project has provided valuable insights into both the potential and limitations of the current design . The robot's ability to solve the maze while adapting to its environment demonstrates the success of the project , but it also reveals opportunities for future enhancements .

Recommendations for Improvement:

as mentioned in future work, upgrading the DC motors to stepper motors and ultrasonic sensor to infrared sensors will improve the performance and accurecy of the project, adding a camera will also unloack many features for these types of projects.

In conclusion, the presented project established a solid ground for explicit advancements in autonomous robotics but still has well defined scope that if addressed and new proven modalities explored would result into technologies that enhance exponentially many application scenarios with more robust performance.

7.References:

-Arduino Mega Documentation: documentation of the Arduino Mega board

: https://www.arduino.cc/en/Main/arduinoBoardMega

-ESP32 Official Documentation: provide a detailed informations for programming the esp32 including also WI-FI, WebServer, Wire, MPU6050, and DHT libraries

: https://docs.espressif.com/projects/esp-idf/en/latest/

https://www.arduino.cc/reference/en/libraries/dht-sensor-library-for-espx/

https://www.arduino.cc/reference/en/language/functions/communication/wire/

-Wikipedia: give us a general overview of maze-solving algorithms and PID control

: https://en.wikipedia.org/wiki/Maze-solving_algorithm https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative_controller