VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI-590018

Mini project report
On

"SIMULATION OF A MAZE-SOLVING ROBOT"

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING
IN
ROBOTICS & AUTOMATION

Submitted by:

Sl.	USN	Name
1	1JS21RA013	R B Kailash Chandran
2	1JS21RA015	Rohith
3	1JS21RA018	Sudanva Aithal Vasudev
4	1JS21RA023	Vikas A Ram

Under the Guidance of
Mr K S Mahesh
Assistant Professor
Department of Robotics & Automation

Internal Guide **External Guide**

Department of Robotics & Automation
JSS Academy of Technical Education, Bengaluru

2023-24

DECLARATION

We hereby declare that the project report titled "**Simulation of a Maze-Solving Robot**" submitted in partial fulfilment for the award of Bachelor of Engineering in Robotics & Automation of Visvesvaraya Technological University, Belagavi during the year 2023-24 is an original work carried out by us under the supervision of Mr K S Mahesh, Assistant Professor, and Department of Robotics & Automation.

We also hereby declare that this report has not been submitted to any other institution or university for any degree or diploma. All the sources of information and references used in the preparation of this report have been duly acknowledged.

Sl.	USN	Name	Signature
1	1JS21RA013	R B KAILASH CHANDRAN	
2	1JS21RA015	ROHITH	
3	1JS21RA019	SUDANVA AITHAL VASUDEV	
4	1JS21RA023	VIKAS A RAM	

JSS Academy of Technical Education, Bengaluru

Department of Robotics & Automation

CERTIFICATE

Certified that the project work **"Simulation of a Maze-Solving Robot"** carried out by Mr. entitled **R B KAILASH CHANDRAN [1JS21RA013], ROHITH [1JS21RA015], SUDANVA AITHAL VASUDEV [1JS21RA019], VIKAS A RAM [1JS21RA023]**, bonafide students of **JSS Academy of Technical Education** in partial fulfilment for the award of Bachelor of Engineering in **Robotics & Automation** of the Visvesvaraya Technological University, Belagavi during the year 2023-24 It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report. The project report has been approved as it satisfies the academic requirements in respect of the Project work prescribed for the said Degree.

Name & Signature of the Guide Name & Signature of the HOD

External Viva

Name of the examiner Signature with date

1

2

Acknowledgement

We would like to express our sincere gratitude to everyone who has supported and guided us throughout the completion of this mini-project.

Primarily, we would like to express our gratitude to the Management of JSS Academy of Technical Education, for providing the infrastructure and resources needed for this project. The knowledge and experiences we have gained during our time here have been invaluable, and we are proud to be a part of this esteemed institution.

We are also profoundly grateful to **Dr Bhimsen Soragaon**, Principal, for their unwavering support and for fostering an environment that promotes learning and innovation. Their encouragement has been a driving force behind our efforts.

We express heartfelt thanks to **Dr Ramya M V**, Head of the Department, Robotics & Automation, for providing a conducive environment for research and development. Their leadership and support have been essential in fostering a positive and productive academic atmosphere.

We extend our deepest gratitude to my project guide, **Mr K S Mahesh**, Asst. Professor, Department of Robotics & Automation, whose expertise, encouragement, and valuable insights have been instrumental in guiding us through every phase of this project. Their constant support and constructive feedback have been crucial in refining our work. Finally, we would like to thank the Technical and non-teaching staff of the JSS Academy of Technical Education for their invaluable assistance and support. Their dedication and hard work behind the scenes ensured that all necessary resources and facilities were available to us, which greatly contributed to the successful completion of this project. Thank you all for your support and encouragement.

Sl.	USN	Name	Signature
1	1JS21RA013	R B KAILASH CHANDRAN	
2	1JS21RA015	ROHITH	
3	1JS21RA019	SUDANVA AITHAL VASUDEV	
4	1JS21RA023	VIKAS A RAM	

Table of Contents

SL.NO	CONTENTS	PAGE NO
1	Introduction, Objective & Goals	1
2	Literature Survey	2
3	Problem Statement	4
4	Design and Methodology	5
5	3D Models	6
6	Components and Their Functions	7
7	Testing and Validation	8
8	Significance	12
9	Advantages	13
9	Results and Analysis	15
10	Conclusion	16
11	Future Scope	17
12	Reference	19
13	Annexure	20

List of Tables

SL.NO	Table Name	PAGE NO
Table: 1	Literature Survey	2
Table: 2	Components List	7
Table: 3	Cost Breakdown	20

Chapter 1: Introduction

1.1 Overview of the Project:

A maze-solving robot is an autonomous machine designed to navigate through a maze and reach a specified goal. It uses sensors to perceive its environment, algorithms to make decisions, and actuators to move. Key components include sensors (proximity, line, ultrasonic), actuators (motors), a microcontroller, and algorithms (wall following, flood fill, BFS, DFS). Maze-solving robots have applications in education, search and rescue, industry, and research.

1.2 Objective & Goals:

The main objectives and goals of this project are:

Safety:

Simulations eliminate the risk of physical damage to the robot or its surroundings, particularly in complex or hazardous environments.

Efficiency:

Simulations can be run at various speeds, accelerating the testing process and reducing the time required to gather data.

Cost-effectiveness:

Simulating a robot is generally less expensive than building and maintaining a physical prototype.

Data Collection:

Simulations can generate large amounts of data that can be analysed to improve the robot's performance and identify potential issues

Chapter 2: Literature Survey

TITLE	AUTHORS	UNIVERSITY	SUMMARY	YEAR
A Simulation-Based Study of Maze-Solving-Robot Navigation for Educational Purposes	Ismu Rijal Fahmi, Dwi Joko Suroso	Dept. of Nuclear Engineering and Engineering Physics, Universitas Gadjah Mada, Yogyakarta, Indonesia	The paper proposes a simulation program for indoor navigation using an open-source code in Python to make it easier to understand and develop navigation and control algorithms. It compares five different navigation algorithms and their performance in various maze types.	2022
Intelligent Maze Solving Robot Based on Image Processing and Graph Theory Algorithms	Mohammad O.A. Aqel, Ahmed Issa, Majde Elhabbash	Department of Engineering, Faculty of Engineering and Information Technology, Al-Azhar University-Gaza	The paper proposes a new method for maze solving robots that uses image processing and artificial intelligence. The method is faster and more reliable than traditional methods. The robot can find the shortest path through a maze and avoid traps.	2017

A Systematic Literature Review of Multi-agent Pathfinding for Maze Research	Semuil Tjiharjadi, Sazalinsyah Razali, Hamzah Asyrani Sulaiman	Computer System Dept., Faculty of Engineering, Maranatha Christian University, Bandung, Indonesia	Multi-agent pathfinding (MAPF) is a challenging AI problem involving coordinating multiple agents to reach their goals without collisions. MAPF has diverse applications, from robotics to video games. This paper conducts a systematic literature review to analyze existing research on MAPF. The goal is to understand the techniques used and explore potential applications for solving maze problems.	2022

Table 1: Literature Survey

Chapter 3: Problem Statement

“The Ultimate Maze Challenge: Navigate, Calculate, and Conquer”.

Simulate a robot's autonomous navigation through a 2D maze to reach a predefined exit, optimizing its path while considering environmental constraints and computational limitations.

Chapter 4: Design and Methodology

4.1 Methodology:

4.1.1 Chassis Assembly:

- Assemble the robot chassis, attaching the motors and wheels securely.

- Mount the MSP430 development board on the chassis.

4.1.2 Motor and Motor Driver Connection:

- Connect the motors to the motor driver module.
- Connect the motor driver to the MSP430 microcontroller. For example, connect the motor driver's input pins to the MSP430's GPIO pins.

4.1.3 Sensor Placement and Connection:

- Place the sensors (e.g., IR sensors) on the front and sides of the robot for obstacle detection.
- Connect the sensor output pins to the MSP430's ADC (Analog to Digital Converter) pins or digital input pins.

4.1.4 Power Supply:

- Connect a battery pack to power the motors and the MSP430 board.
- Ensure proper voltage regulation for the MSP430 (typically 3.3V).

4.2 Design:

- Understand how maze-solving robots work by referring to previous research papers.
- Understand different algorithm and figure out a way to use it to solve maze.
- Develop a code for finding the shortest path to solve the maze using suitable algorithm in with the help of Code Composer Studio.
- Making a 6' * 6' maze to test the robot.

Chapter 5: 3D Models

Fig 1: In depth view

Chapter 6: Components and Their Functions

SI. No	Materials Used	Function	Quantity
1	BUCK Converter	A DC-to-DC converter that steps down the voltage from the Li-Po battery (12V) to a suitable level for the microcontroller and other components.	1
2	Li-Po Battery (12V)	Provides the primary power source for the robot.	1
3	ESP32 Wi-Fi Module	A powerful microcontroller with built-in Wi-Fi capabilities, used for communication, control, and data processing.	1
4	L298N Motor Driver	A dual H-bridge motor driver that controls the direction and speed of the N20 mini motor.	1
5	N20 Mini Motor	A small, lightweight motor used for propulsion.	2
6	MSP430(GR553)	A low-power microcontroller that can be used for specific tasks or as a secondary processor for offloading certain functions from the ESP32.	1
7	IR Sensors	Detect infrared light, often used for obstacle avoidance or line following in maze environments.	3
8	Ultrasonic Sensor	Measures distance by emitting and receiving sound waves, used for obstacle detection and navigation.	2
9	Breadboard	A temporary platform for prototyping and testing electronic circuits.	1

Table 2: Components List

Chapter 7: Testing and Validation

Linear congruential generator (LCG)

An algorithm that yields a sequence of pseudo-randomized numbers calculated with a discontinuous piecewise linear equation.

Random Mouse Program:

```
// Global variable for random number generator

unsigned int rand_seed = 0xACE1; // Seed value for random number generator

unsigned int rand(void)

{

   rand_seed = (rand_seed * 1103515245 + 12345) & 0x7FFF;

   return rand_seed;

}

void solveMaze(void)

{

   unsigned int left = readSensor(LEFT_SENSOR);

   unsigned int right = readSensor(RIGHT_SENSOR);

   unsigned int front = readSensor(FRONT_SENSOR);

   if (front > 512) // Threshold value, adjust based on your sensors

   {

      stop();
```

```
    // Generate a random decision
    unsigned int decision = rand() % 2; // 0 or 1
    if (decision == 0)
    {
      turnRight();
    }
    else
    {
      turnLeft();
    }
  }
  else if (left > 512) // Threshold value, adjust based on your sensors
  {
    stop();
    moveForward();
  }
  else if (right > 512) // Threshold value, adjust based on your sensors
  {
    stop();
    moveForward();
```

```
    }
    else
    {
        moveForward();
    }
}
```

Wall following

The "Wall Following" algorithm is a popular maze-solving strategy for robots. One common variant is the "Right-Hand Rule," where the robot maintains contact with the right wall throughout the maze. This strategy is suitable for a maze-solving robot with limited sensors and simple logic

Wall Following Program:

```
void solveMaze(void)
{
    unsigned int left = readSensor(LEFT_SENSOR);
    unsigned int right = readSensor(RIGHT_SENSOR);
    unsigned int front = readSensor(FRONT_SENSOR);
// Assuming a threshold value for detecting walls
    unsigned int wallThreshold = 512; // Adjust based on your sensor
    if (front < wallThreshold) // Wall in front
```

```
    {
        // Check if there's a wall on the right
        if (right > wallThreshold)
        {
            // Turn right if no wall on the right
            turnRight();
        }
        else
        {
            // If there is a wall on the right, turn left
            turnLeft();
        }
    }
    else
    {
        // Move forward if no wall in front
        moveForward();
    }
}
```

Chapter 8: Significance

8.1. Testing and Development: Simulations provide a safe and controlled environment to test and refine robot algorithms, hardware components, and overall performance without risking physical damage.

8.2. Algorithm Evaluation: Different pathfinding algorithms can be compared and evaluated in various maze scenarios to identify the most efficient and robust approaches.

8.3. Parameter Tuning: Simulation allows for the fine-tuning of robot parameters, such as sensor sensitivity, motor speed, and obstacle avoidance thresholds, to optimize its performance.

8.4. Training and Education: Simulations provide a hands-on learning experience for students and researchers, allowing them to experiment with different maze-solving strategies without the need for expensive hardware.

8.5. Risk Assessment: Simulations can help identify potential risks and challenges that the robot might encounter in real-world environments, enabling proactive measures to be taken.

8.6. Virtual Reality Integration: Simulations can be integrated with virtual reality environments to provide a more immersive and realistic experience for users.

Chapter 9: Advantages

9.1. Safety and Cost-Effectiveness:

- **Risk Mitigation:** Simulations eliminate the risk of physical damage to the robot or its surroundings, especially in complex maze environments.
- **Cost Reduction:** Hardware costs, maintenance, and potential repairs are significantly reduced.

9.2. Control and Repeatability:

- **Parameter Adjustment:** Variables like maze complexity, robot speed, and sensor sensitivity can be easily adjusted and controlled.
- **Reproducibility:** Experiments can be repeated under identical conditions, ensuring consistent results.

9.3. Data Collection and Analysis:

- **Detailed Metrics:** Gather comprehensive data on robot performance, including path length, time taken, and sensor readings.
- **Algorithm Evaluation:** Compare and optimize different algorithms for maze solving.

9.4. Rapid Iteration and Development:

- **Faster Testing:** Implement changes to the robot's design or algorithms and quickly evaluate their impact.
- **Accelerated Learning:** Learn from mistakes and refine strategies without physical constraints.

9.5. Educational Value:

- **Visualization:** Simulations provide a visual representation of the robot's behavior and decision-making process.
- **Understanding:** Students can grasp complex concepts more easily by experimenting with different parameters.

9.6. Virtual Environments:

- **Diverse Mazes:** Create a wide range of maze environments, from simple grids to intricate labyrinths.
- **Realistic Scenarios:** Simulate real-world challenges like obstacles, changing environments, and sensor noise.

Chapter 10: Results and Analysis

10.1. Wall-Following:

A robot-using wall following in a simple maze might successfully reach the goal but take a longer path than necessary.

10.2. A* Search:

In a complex maze with a well-designed heuristic, an A* search algorithm might find the shortest path efficiently.

10.3 DFS:

A robot using DFS could be trapped in a dead end and fail to reach the goal.

Chapter 11: Conclusion

The simulation of a maze-solving robot provides valuable insights into the complexities and challenges associated with autonomous navigation in constrained environments. Throughout this project, we successfully implemented algorithms that enable a robot to navigate through a maze, avoiding obstacles and making decisions at intersections to find the optimal path to the target.

Key accomplishments include the successful design and simulation of the robot's movement and decision-making processes. The robot's ability to recognize and react to different maze structures was thoroughly tested, demonstrating the effectiveness of our chosen algorithms, particularly the use of depth-first search (DFS), breadth-first search (BFS), and A* algorithms.

The project also highlighted the importance of sensor integration and real-time data processing in robotic navigation. By simulating the robot's environment and sensory inputs, we were able to refine the robot's decision-making process, ensuring that it could handle varying maze complexities. The simulation environment proved to be a critical tool for testing and validating the robot's algorithms without the need for a physical prototype, thereby saving resources and time.

One of the significant challenges encountered was the trade-off between the speed of the robot's decision-making and the accuracy of its pathfinding. While algorithms like A* provided optimal paths, they required more computational resources, which could affect real-time performance. Balancing these factors was crucial in determining the most suitable approach for different maze configurations.

In conclusion, the simulation of a maze-solving robot serves as a foundational project that illustrates the principles of autonomous navigation and algorithmic pathfinding. The knowledge gained from this project is applicable to a wide range of real-world applications, from robotic automation in warehouses to autonomous vehicles. Future work could involve implementing machine-learning techniques to enhance the robot's adaptability to dynamic environments or extending the simulation to 3D mazes for more complex scenarios. This project has successfully met its objectives and lays the groundwork for further exploration and development in the field of robotic navigation.

Chapter 12: Future Scope

12.1. Real-World Implementation:

- **Hardware Integration:** Transition the simulation to a physical robot platform, incorporating sensors like LiDAR, ultrasonic sensors, or cameras for real-time environment perception.
- **Obstacle Avoidance:** Enhance the robot's ability to navigate around dynamic obstacles, such as moving objects or people.
- **Autonomous Exploration:** Develop algorithms that enable the robot to autonomously explore and map unknown environments.

12.2. Algorithm Refinement:

- **Advanced Pathfinding:** Explore more sophisticated pathfinding algorithms, such as A* with heuristics or Dijkstra's algorithm for large-scale environments.
- **Learning-Based Approaches:** Implement machine-learning techniques, like reinforcement learning, to enable the robot to learn optimal paths through experience.
- **Adaptive Behaviour:** Develop algorithms that allow the robot to adapt its behaviour based on environmental changes or unforeseen challenges.

12.3. Multi-Robot Coordination:

- **Cooperative Problem-Solving:** Explore how multiple robots can collaborate to solve complex mazes or tasks more efficiently.
- **Conflict Resolution:** Develop strategies for preventing collisions and ensuring smooth coordination among multiple robots.
- **Distributed Decision-Making:** Investigate decentralized approaches where robots make decisions based on local information.

12.4. Simulation Enhancements:

- **Realistic Environments:** Create more realistic simulation environments with varying terrains, obstacles, and lighting conditions.

- **Dynamic Challenges:** Introduce dynamic elements like moving walls, disappearing paths, or time constraints to increase the simulation's complexity.
- **Benchmarking and Evaluation:** Develop standardized benchmarks to compare the performance of different maze-solving algorithms.

12.5. Applications and Extensions:

- **Search and Rescue:** Explore how maze-solving robots can be adapted for search and rescue operations in disaster-stricken areas.
- **Industrial Automation:** Investigate applications in industrial settings, such as inspecting complex machinery or navigating warehouses.
- **Entertainment and Games:** Create interactive maze-solving games or simulations for educational or entertainment purposes.

Chapter 13: Reference

- An Autonomous Maze-Solving Robotic System Based on an Enhanced Wall-Follower Approach - https://www.mdpi.com/2075-1702/9/7/81
- Design and Implementation of Autonomous Maze-Solving Robot based on an Optimized Flood-Fill Algorithm – https://www.ijert.org/design-and-implementation-of-autonomous-maze-solving-robot-based-on-an-optimized-flood-fill-algorithm
- Maze Solving Robot with Automated Obstacle Avoidance - https://www.ijrte.org/wp-content/uploads/papers/v7i6/F4721047619.pdf
- An Efficient Autonomous Maze Solving Robot based on Microcontroller - https://www.ijeat.org/wp-content/uploads/papers/v8i6/F8735088619.pdf
- An Efficient Algorithm for Robot Maze-Solving - https://www.researchgate.net/publication/330361795_An_Efficient_Algorithm_for_Robot_Maze-Solving
- Autonomous Maze Solving Robotics: Algorithms and Systems - https://www.sciencedirect.com/science/article/pii/S1877050916315338
- Design and Implementation of a Robot for Maze-Solving using Flood-Fill Algorithm - https://www.sciencedirect.com/science/article/abs/pii/S221201731300400X
- A Comprehensive and Comparative Study Of Maze-Solving Techniques - https://ieeexplore.ieee.org/document/8424331

Chapter 14: Annexure A: Detailed Cost Breakdown

SI. No	Material Used	Quantity	Cost Per Unit	Total Cost
1	MSP430(GR553)	1	4100/-	4100/-
2	L298N Motor Driver	1	150/-	150/-
3	N20 Mini Motor	2	100/-	200/-
4	Ultrasonic Sensor	2	100/-	200/-
5	IR Sensors	3	50	150/-
6	Li-Po Battery (12V)	1	1000/-	1000/-
7	Jumper Cables	1 set	100/-	100/-
8	Chassis	1	300/-	300/-
9	ESP32 WIFI Module	2	500/-	500/-
10	Breadboard	1	100/-	100/-
11	BUCK Convertor	1	100/-	100/-
		Total:		6900/-

Table 3: Cost Breakdown