IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

MargShodhak : A Maze Breakthrough Robot

[1]SOWMYA P, [2]SRINIVASA G S, [3]VUNNAM NAGA SAHITHI, [4]SHIVARAJ PRAKASH NAIK, [5]ASHWAL H U SHASTRY

[1]Assisstant Professor, [2]B.E Student, [3]B.E Student, [4]B.E Student, [5]B.E Student
[1]Electronics and Communication Engineering,
[1]Dayananda Sagar College of Engineering, Bangalore, India

Abstract : In this project, we are going to design a maze-solving robot. We will be involved in the construction and programming of a maze-solving robot. Maze solving problem is a very old problem, but still, now it is considered an important field of robotics. This field is based on decision-making algorithms. The main aim of this project is to make an Arduino-based efficient autonomous maze solver robot. Nowadays, maze solving problem is an important field of robotics. It is based on one of the most important areas of the robot, which is the "Decision-Making Algorithm". Cause, this robot will be placed in an unknown place, and it requires to have good decision-making capability. There are many types of maze-solving robots using various types of algorithms. We will be using something known as an autonomous robot. An autonomous robot is a category of robot that can perform tasks intelligently depending on themselves, without any human assistance. A maze-solving robot will come under this section.

***IndexTerms* - Maze, Arduino, Decision-Making Algorithm, Autonomous**

I. INTRODUCTION

Maze solving problem is a very old problem, but still, it is considered an important field of robotics. This field is based on decision-making algorithms. The main aim of this project is to make an Arduino-based efficient autonomous maze solver robot. In this project, Hardware development, software development, and maze construction would be done. For performance testing, the robot will implement to solve different mazes, and the capability of finding the optimized path is also verified.

Mobile robots are an increasingly growing technology and have major importance in scientific and industrial areas, as their abilities to solve complex tasks have evolved continuously. Mobile robots are used in a wide range of applications: the exploration of difficult terrain, surveillance and security, household activities, military applications, transportation, rescue operations, and even space exploration. Having said that, MargShodhak serves as a very important application for finding the destination of an unknown maze.

II. RELATED WORK

We present some relevant research work in this section. Interesting research was presented by R. Covaci, G. Harja, and I. Nascu[1].In this paper, they have discussed the sensors that are used for building an autonomous vehicle along with other actuators and drivers required to complete the robot.

Another research paper by Islam, Akib & Ahmad, Farogh & Pichandi, and Sathya[2], talk about the LSRB and RSLB algorithm. It also talks about finding the shortest path by implementing both algorithms for a single maze.

The work from S. Alamri, S. Alshehri, W. Alshehri, H. Alamri, A. Alaklabi, and T. Alhmiedat[3] deals with the study of different algorithm called "Maze routing algorithm", and also explains the advantages and disadvantages of the algorithm.

The work by Chang Yuen Chung[4] deals with the implementation of the "Flood-Fill Algorithm" which can have its advantages and the main con is having to know the maze size prior which makes it unsuitable for certain applications.

The paper from Cai, Jianping, et al[5] focuses on implementing the "Left Hand Follower Algorithm", "Center-Left Algorithm", and "Centripetal Algorithm" via simulation and does a comparative study of all three algorithms.

The paper by Gupta, Bhawna, and Smriti Sehgal[6] explains the "Wall Follower Algorithm", "Lee's Algorithm", and "Flood-Fill Algorithm" and has given a comparison between the three algorithms based on individual algorithm's time complexity, space complexity, pre assumptions, and efficiency.

III. PROPOSED METHODOLOGY

The MargShodhak works mainly in two modes:

1. The robot first traverses through the entire maze by using the LSRB algorithm with the help of ultrasonic sensors while storing all the turns taken by the robot and reaching the destination.

2. Once the destination is found, it optimizes the stored path into the simplest path based on rules. Now, the robot reaches the destination in the optimized path automatically.

The robot will have wheels for its locomotion. We will use a solving algorithm and then make decisions for the robot to go in which particular direction, i.e., left, right, forward, and backward. The robot is kept at one position in the maze and then the robot automatically starts traversing in the maze using the LSRB algorithm, i.e., a rule which states the robot to go in which direction whenever intersections are detected. According to the algorithm, whenever there is a possibility to go left at the intersection, the robot takes a left and whenever the possibility is between straight and right, it chooses to go straight, and at the dead end, it makes a U-turn. MargShodhak travels through the entire maze until it finds a destination and then stops. Once the destination is found, the robot automatically finds the optimum distance through machine learning between the start and destination, and when kept at the start position, the robot goes to the destination in the calculated optimum path.

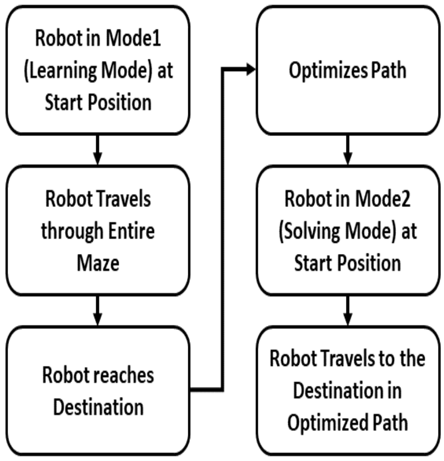

Figure 1 - Functional Flow Chart

IV. BLOCK DIAGRAM

An ultrasonic sensor is used for detecting the wall and the corresponding distances are sent to the Arduino to make decisions based on the LSRB algorithm. Based on the outcome from the Arduino after analysis, corresponding actuator signals are sent to the motor for changing direction or to keep the robot moving in the same direction.

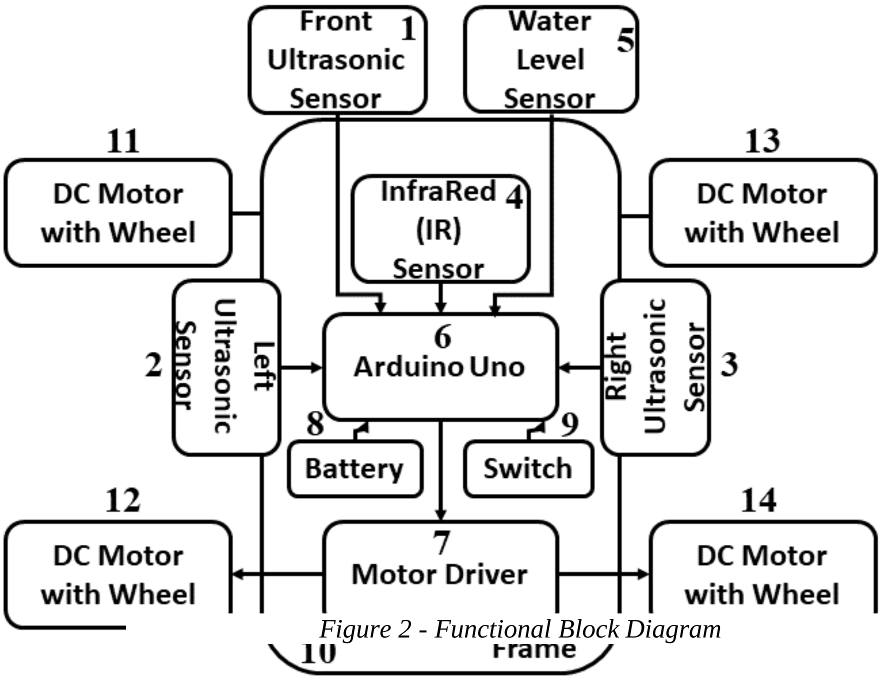

Figure 2 - Functional Block Diagram

- Block 1 gives information about an Ultrasonic sensor (HCSR04) which is located in the front of the robot for detecting the wall and the corresponding distances are sent to the Arduino to make decisions based on the LSRB algorithm. Based on the outcome from the Arduino after analysis, corresponding actuator signals are sent to the motor for changing direction or to keep the robot moving in the same direction.
- Block 2 and 3 give the information about left and right Ultrasonic sensors for detecting the walls on the left and right and corresponding decisions are made by the Arduino after analyzing, corresponding actuator signals are sent to the motor for changing direction or to keep the robot moving in the same direction.
- Block 4 gives information about the end position or location which is marked by a black path or circle, which can be easily identified by the IR sensor.
- Block 5 gives information about the presence of water bodies and their depth to decide whether the robot should avoid them or travel through them.
- Block 6 gives information about Arduino, basically a microcontroller that controls all the sensors mounted on the frame. This small computer is used as the brain of the robot. It controls the robot by analyzing the algorithm.
- Block 7 gives information about a Motor driver IC; it is an integrated circuit chip that controls motors in autonomous robots and embedded circuits. L293D and ULN2003 are the most commonly used motor Driver IC that is used in simple robots and RC cars. In our project, we used L293D.
- Block 8 gives information about the battery. This is used for powering the entire robot. We have used a 3s LiPo battery of 1000mAh capacity with a 1C discharge rate.
- Block 9 gives the information about the switch, which is used to change the robot from mode1 to mode2 that is from learning mode to solving mode.
- Block 10 gives the information about Frame. This will be used to make the body of the robot.
- Blocks 11 and 12 give the information about the out left 300 RPM DC Motor with wheels. This will be used for the translation of the robot in different directions.
- Blocks 13 and 14 give the information about the right 300 RPM DC Motor with wheels. This will be used for the translation of the robot in different directions.

V. ALGORITHM

The algorithm used here is known as LSRB Algorithm or Left Wall Following Algorithm. In this algorithm, the priority for making a turn at an intersection is assigned. The priority is Left Straight Right and Back/U-turn, where Left has the highest priority and U-turn has the least priority, meaning the robot takes left turn whenever there is a possibility, if not then moves straight and if this is also not possible then right and then last preference to the U-turn.

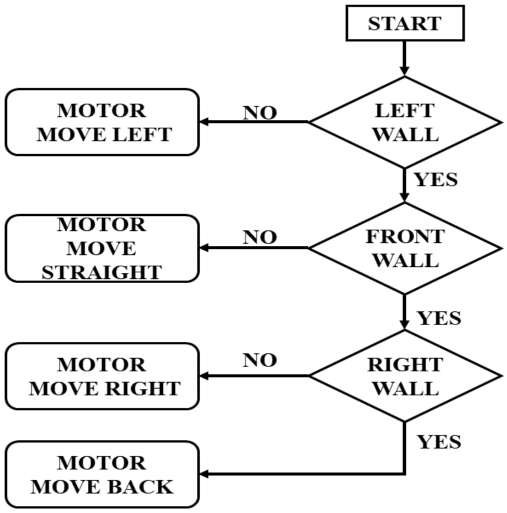

Figure 3 - Algorithm

VI. TOOLS USED

The software tools used in this project are:

- Arduino IDE – This will be used for writing codes for the microcontroller.
- Tinker CAD – This a software by AutoCAD which can be used to simulate the Arduino circuits before hardware.

The hardware tools used in this project are:

- LiPo Battery – This is used for powering the entire robot.
- Arduino UNO – The microcontroller for the robot to make all the decisions.
- Ultrasonic Sensor (HCSR04) – This is the sensor that is used to detect the wall around the robot.
- Geared Motors (300RPM) and Wheels – This will be used for the translation of the robot.
- Motor Driver (L293D) – This will be used to power the motor and control using a PWM signal.
- Switch – This will be used to switch the robot from one mode to another.
- Robot Frame (4WD) – This will be used to make the body of the robot.

VII. PSEUDOCODE

```
**********************************************LSRB Pseudo Code**********************************************
Getdistance(Front); //Initialize front ultrasonic sensor
Getdistance(Left); // Initialize left ultrasonic sensor
Getdistance(Right); // Initialize right ultrasonic sensor
Right_motor(activate); //Initialize right side motors
Left_motor(activate); //Initialize left side motor
DIS = 8; //Minimum distance
//Condition for turns at an intersection
Left_turn = Left > DIS; //Left
Straight = Left<DIS && Front>DIS; //Straight
Right_turn = Left<DIS && Front<DIS && Right>DIS; //Right
Back = Left<DIS && Front<DIS && Right<Threshold; //Back
//Motor signals
if(Left_turn){
Right_motor = OFF;
Left_motor = ON(forward);}
if(Straight){
Right_motor = ON(forward);
Left_motor = ON(forward);}
if(Right_turn){
Right_motor = ON(forward);
Left_motor = OFF;}
if(Back){
Right_motor = ON(backward);
Left_motor = ON(backward);}

*****************************************Path Optimization Pseudo Code*****************************************
GetDirection(Ultrasonic Sensor);
StoreinArray(Direction);
CheckArraywithConditions();
Condition1 = LBR; //□B
Condition2 = LBS; //□R
Condition3 = RBL; //□B
Condition4 = SBL; //□R
Condition5 = SBS; //□B
Condition6 = LBL; //□S
Optimise_Array();
Recheck_Array();
if( further_optimise == true){
Optimise_Array();
```

```
}
else{
Stop_Optimise();
}
StoreOptimiseArray();
```

VIII. HARDWARE

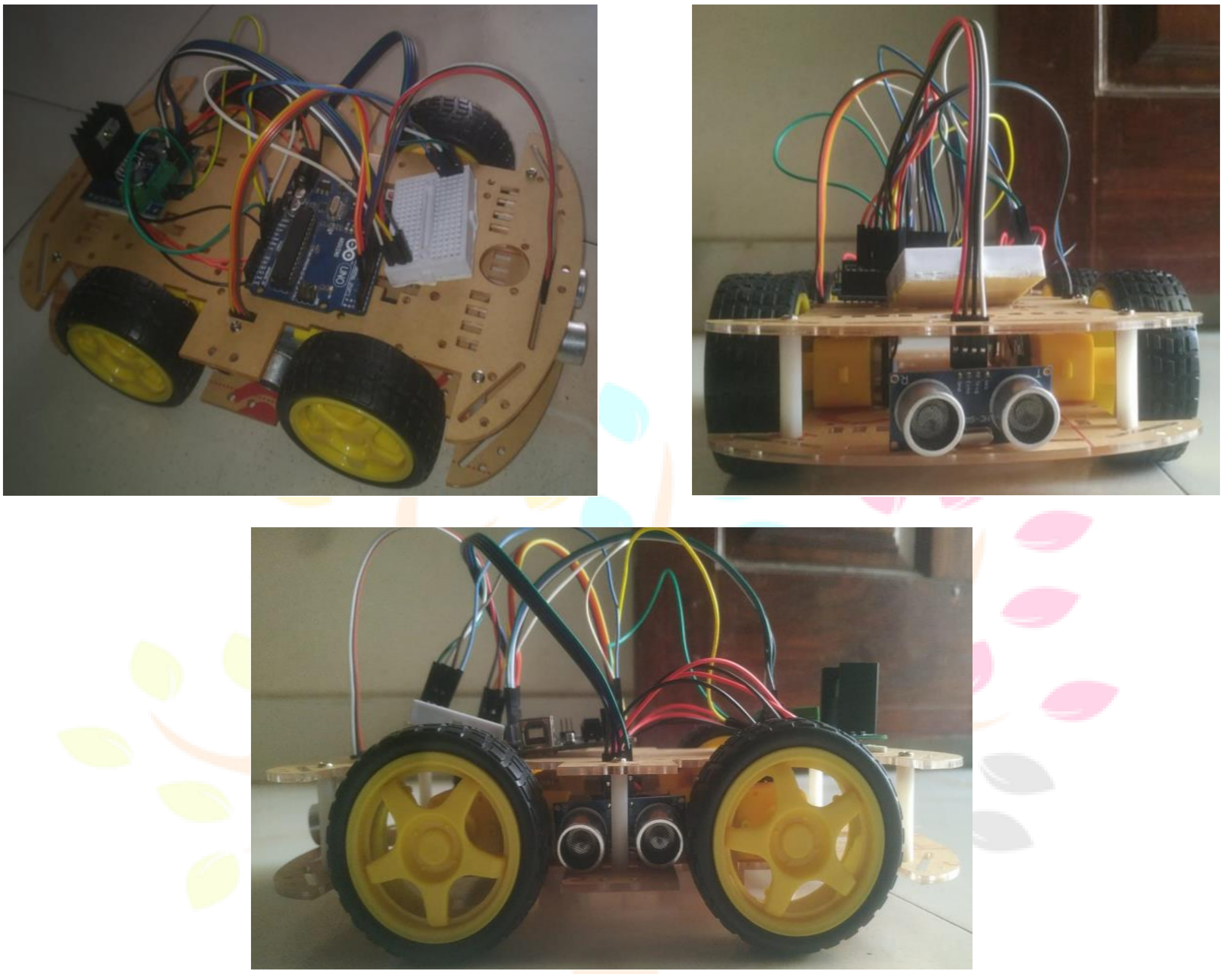

Figure 4 – MargShodhak

IX. SIMULATION OUTPUT

```
Serial Monitor        Serial Monitor
Front=69              Front=69
Right=92              Right=92
Left=162              Left=3
L                     S
Front=69              Front=69
Right=92              Right=92
Left=162              Left=3
L                     S
Front=69              Front=69
Right=92              Right=92
Left=162              Left=3
L                     S
```

Figure 5 – LSRB Decision

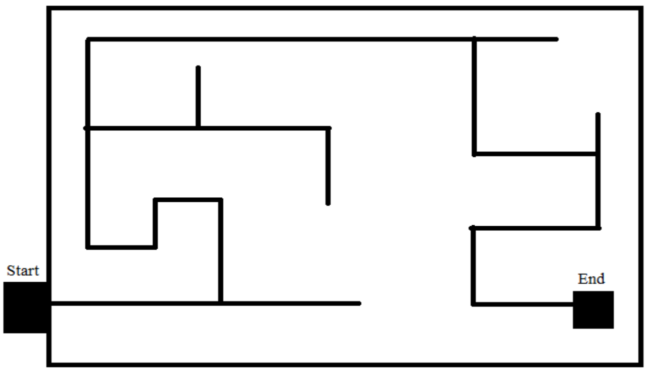

Figure 6 – Sample Maze

```
char a[]={'S','L','L','L',
          'R','R','S','R',
          'S','B','L','L',
          'L','B','S','R',
          'L','L'};
```

Figure 7 - Actual Path

```
Serial Monitor
SLLLRRSRRLRRLLNNNN
SLLLRRSRRLRRLLSLLLRRSRS
SLLLRRSRRLRRLLNNNN
SLLLRRSRRLRRLL
```

Figure 8 - Optimised Path

X. CONCLUSION

MargShodhak with an LSRB algorithm is implemented and its working is verified for different conditions in a maze.

MargShodhak consists of Ultrasonic Sensors for wall detection, an IR sensor for destination detection, a Motor driver for controlling the speed of motors, DC Geared motors for translation, and Arduino Uno for processing i.e., microcontroller.

MargShodhak can learn the maze by traveling through the entire maze using the LSRB algorithm and also storing the traveled path in its memory. Once the destination is reached, the robot mode is changed and it processes through the traveled path for finding the optimum path. Once this is done, the robot will now reach the destination from the source in the optimized path.

XI. REFERENCES

[1]. R. Covaci, G. Harja, and I. Nascu, “Autonomous Maze Solving Robot,” IEEE Xplore, May 01, 2020. https://ieeexplore.ieee.org/abstract/document/9129943/ (accessed Jul. 14, 2022).

[2]. Islam, Akib & Ahmad, Farogh & Pichandi, Sathya. (2016). SHORTEST DISTANCE MAZE-SOLVING ROBOT. International Journal of Research in Engineering and Technology. eISSN pISSN. 253-259.

[3]. S. Alamri, S. Alshehri, W. Alshehri, H. Alamri, A. Alaklabi, and T. Alhmiedat, “Autonomous Maze Solving Robotics: Algorithms and Systems,” International Journal of Mechanical Engineering and Robotics Research, pp. 668–675, 2021, Doi: 10.18178/ijmerr.10.12.668-675.

[4]. Chang Yuen Chung, “micromouse maze solving robot” yumpu.com, Dec. 2021. https://www.yumpu.com/en/document/view/49400832/micromouse-maze-solvingrobot-chang-yuen-chung-universiti- (accessed Nov. 05, 2022).

[5]. Cai, Jianping, et al. “A Micromouse Maze Sovling Simulator.” IEEE Xplore, 1 May 2010, ieeexplore.ieee.org/document/5497446. Accessed 3 Dec. 2022.

[6]. Gupta, Bhawna, and Smriti Sehgal. “Survey on Techniques Used in Autonomous Maze Solving Robot.” 2014 5th International Conference - Confluence the next Generation Information Technology Summit (Confluence), Sept. 2014. Accessed 18 Nov. 2021.

[7]. Sadik, Adil M.J., et al. “A Comprehensive and Comparative Study of Maze-Solving Techniques by Implementing Graph Theory.” IEEE Xplore, 1 Oct. 2010, ieeexplore.ieee.org/abstract/document/5656597.

[8]. Pinto, Márcio Tameirão, et al. “UNEXUP: Robot-Based Exploration Technology for Underground Flooded Mines.” Advances in Geosciences, vol. 54, 27 Oct. 2020, pp. 109–117, 10.5194/adgeo-54-109-2020. Accessed 25 Aug. 2022.

[9]. White, William. “Identification of Cave Minerals by Raman Spectroscopy: New Technology for Non-Destructive Analysis.” International Journal of Speleology, vol. 35, no. 2, digitalcommons.usf.edu/cgi/viewcontent.cgi?article=1197&context=ijs, July 2006, pp. 103–107, 10.5038/1827-806x.35.2.6.

[10]. R. Selvakumar, R. V. S. Abhiram, and K. P. Reddy, “Experimental analysis of maze solving robot using LSRB algorithm,” 2022 International Conference on Computer Communication and Informatics (ICCCI), Jan. 2022, DOI: 10.1109/iccci54379.2022.9740916.

[11]. Rahman, Musfiqur, “AUTONOMOUS MAZE SOLVING ROBOT,” IEEE Xplore, 2017. DOI: 10.13140/RG.2.2.34525.82403

[12]. M. Ansar, “Arduino based black Line Follower Robot,” www.youtube.com, 2020. https://www.youtube.com/watch?v=UdoXBjDOkH8 (accessed Oct. 20, 2022).

[13]. C. Digest, “How to Make a Simple Line Follower Robot using Arduino,” www.youtube.com, 2021. https://www.youtube.com/watch?v=PyKAanh_f30 (accessed Oct. 10, 2022).

[14]. MJRoBot.org, “Maze Solver Robot, Using Arduino,” Instructables. www.instructables.com/Maze-Solver-Robot-Using-Arduino (accessed Oct. 18, 2022).