ISSN: 2366-1313

Autonomous Line Maze Solver Using Artificial Intelligence

POKURI PRAKASH[1], *Dr. P VENKATA NAGANJANEYULU* [2]
[1]Assistant professor, [2] Professor and Principal

ECE Department, Sri *Mittapalli College* of *Engineering*, *Guntur*, *Andhra Pradesh*-522233

Abstract—Artificial intelligence is steadily growing, especially nowadays. It starts to represent a necessity for various technological processes such as the automation of operations in various factories, certain robots that can make decisions without human help, cars that can drive alone. In this paper we present an application of artificial intelligence, namely a problem that is common in the field of robotics: navigating a robot in a labyrinth without human interaction, using the left hand rule and dead-end filling. The line maze solver using these algorithms will be able to drive through a labyrinth that has a solution and more precisely a ”disk”. The robot uses a Raspberry Pi 3 as a ”brain”, where with the Python 2 programming environment we have managed to implement these algorithms to solve our problem.

Index Terms—Artificial Intelligence, Line Maze Solver, Robot, Raspberry Pi, Left Hand Rule, Dead-end Filling

I. INTRODUCTION

This paper represents the implementation of Artificial Intelligence in a real life situation, solving a line maze. The concept of Artificial Intelligence emerged in 1956. This dream became reality in recent years, being introduced in more and more domains. Artificial intelligence by definition, resolves new situations or problems based on experience gained on the basis of continuous learning. Artificial Intelligence, which is based on acquired knowledge, began to take shape in the field of computer science, developing systems to solve technical problems through human intelligence. That means the limits are unknown because it does not have anything to compare to and the new ideas are based on the previous ones. A maze generally represents a network of paths that resembles to a puzzle in which one has to find a way from the entrance to the exit [1]. This concept is thousand years old and was invented in Egypt. Since then many mathematicians came up with various algorithms to solve this problem. Maze solving is one among the foremost common problems in the domain of mobile robotics; a solution to maze solving is using an autonomous robot. This category of robots can perform tasks without any human interaction. It is an important area of the robotics domain because its core is a ”Decision making algorithm” [2]. The robot that we present in this paper functions on two main tasks:

1) Solving the maze using the classic maze algorithm left hand rule: Thus reaching the ”disk” that is the solution of the maze.
2) Using a method similar to the dead-end filling algorithm: The robot will return to the starting point on the shortest path known, beginning from any straight line of the maze.

II. OVERVIEW MAZE SOLVING ROBOTS

The maze solving robots are divided into two categories. One is the micro mouse maze solver that is designed to get to the center of the maze in the shortest time possible [3]. The other category is the maze solver that searches for a specific target or for the exit of the maze [4].

In Table I are presented three maze solving robots that use different algorithms and the results achieved by them.

TABLE I
MAZE SOLVING ROBOTS

Entry	*Algorithm used*	*Result*
A	LSRB and RSLB	Because of its size it is not dynamic
B	Right hand rule	The robot has a 70-80 percent accuracy
C	Djikstras algorithm	Increased execution time of the algorithm

[a]Left turn =”L”, Right turn = ”R”, Turn around = ”B”, Go straight = ”S”

A. Shortest Distance Maze Solving Robot

This robot was designed by Akib Islam, Farogh Ahmad and P. Sathya from School of Electronics Engineering, VIT University, Tamilnadu, India and is represented in Fig. 1.

ISSN: 2366-1313

The robot uses the LSRB and RSLB algorithm. Using these algorithms the robot completes its purpose and finds the shortest path but due to its size it is not a dynamic and fast maze solver [5].

B. Artificially Intelligent Maze Solver Robot

The maze solver presented in Fig. 2 was made by Malkit Singh, Rajnish Kumar, Vaibhav Giradkar, Pallavi Bhole, Minu Kumari from Priyadarshini College of Engineering, Nagpur, India. The algorithm used in this project is the right hand rule and its purpose was to solve the maze by getting in the center of it. The results were a 70%-80% accuracy [6].

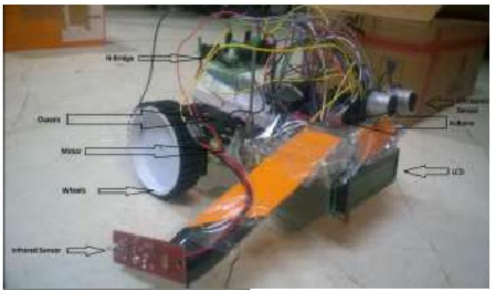


Fig. 1. Shortest Distance Maze Solving Robot

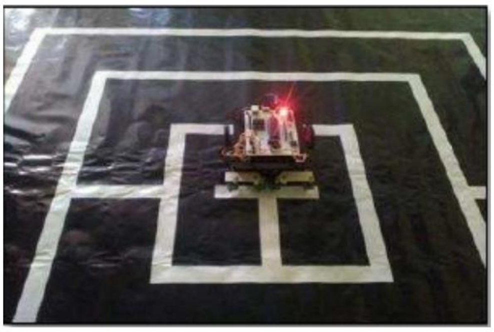


Fig. 2. Artificially Intelligent Maze Solver Robot

C. Maze Solving Algorithms for Micro Mouse

The work of Swati Mishra from Inderprastha Engineering College, Ghaziabad and Pankaj Bande from International Institute of Information Technology, Pune, presented in Fig. 3 represents a Micro Mouse project and various algorithms. The main algorithm that we take in account is Djikstra's algorithm [7].

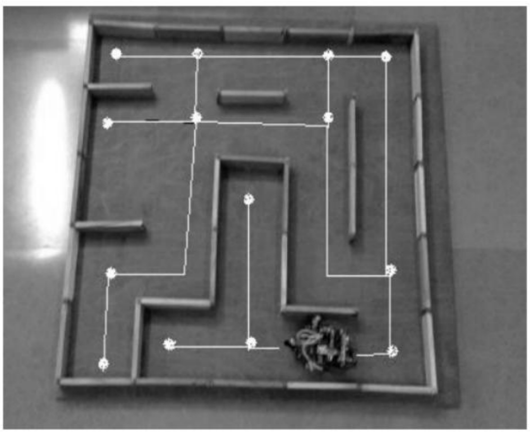


Fig. 3. Maze Solving Algorithms Djikstras algorithm for Micro Mouse

III. ALGORITHMS USED FOR MAZE SOLVER ROBOT

We use for this robot the left-hand rule and an algorithm similar to the dead-end filling algorithm.

A. The left-hand rule

The left-hand rule algorithm can be used if the mazes walls (in our case "lines") are all connected together. By always following the left "wall" the robot will get to the exit or to the designated goal (in our case "disk") [8]. Using a line array with only three sensor for the purpose of the low cost of this project means more work on the software part.

The complex maneuver to use the left hand rule using only three sensors at each node of the maze is composed of 6 moves:

1. The maze solver is going in the usual forward protocol, having the middle sensor above the line.
2. Reaching the junction the robot detects with its right sensor the line and it thinks that it is drifting to the left so a correction needs to be made to get back with the middle sensor on the line, thus going to right.
3. Using this correction protocol all the sensors are now on black. The robot detects this as a node but because of the last correction it is now too far to the right.
4. In this case the robot drives to the left to verify the existence of the line. Because the sensor does not detect the line, the robot is knowing now that there is a line at the right.
5. The direction of the robot is corrected again by turning a little to the right.
6. Driving a little further, the robot is detecting now a continuation of the line knowing that the junction was a "Tjunction to the right" presented in Fig. 4. From this position the maze solver can now decide if he moves on or turns to the right.

In the case of dead ends, meaning all sensors are on "white", the robot can start the protocol to turn almost instantly 180º.

ISSN: 2366-1313

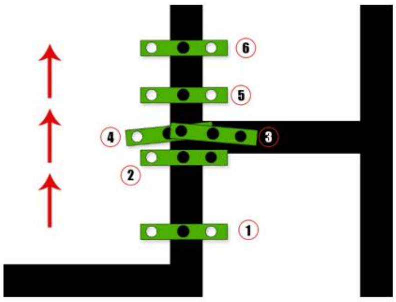


Fig. 4. T junction maneuver

B. Dead-end filling

This is an algorithm for solving mazes that fills all dead ends, leaving only the correct ways unfilled. In our case this represents the shortest way that was traveled. The method follows two main directions:

1) To find all the dead-ends in the maze: All the dead-ends are eliminated to find the shortest path.
2) To ”fill in” all the paths of each dead-ends of the maze until junctions are met: In this implementation not all deadends are found from the beginning. They are found after meeting certain junctions [9].

IV. COMPONENT DESCRIPTION OF MAZE SOLVER ROBOT

In this project one of the many goals is to make a low cost robot. This means using less sensors and no encoders but an efficient line maze solver. All the used hardware components are presented in Fig. 5 and are as follows:

A. Hardware components

1) Raspberry Pi 3: This is the core of the robot that controls its every move with the help of the engine driver and the sensors.
2) Motor driver DRV8835 for Raspberry Pi 3: In the first model the driver was l298n but the new driver was made specifically for Raspberry Pi 3, because its library helps very much with the programming part. With the help of the dual motor driver DRV8835, PWM signals can be sent to the motors using the GPIO pins of the Raspberry.
3) QTR-3A Reflectance Sensor Array: Using this sensor array the robot knows where the line is situated. Implementing the left hand rule using just 3 sensors was the low cost solution on the hardware part but this means more work on the software part.
4) Step down converter from 7V-20V to USB 5V3A: This component was the best way to power up the Raspberry Pi 3 without many complications.
5) A square push button with hold: This button powers up the robot.
6) 7.2V Battery: With this battery we supplied power both to the Raspberry Pi 3 and the engines.

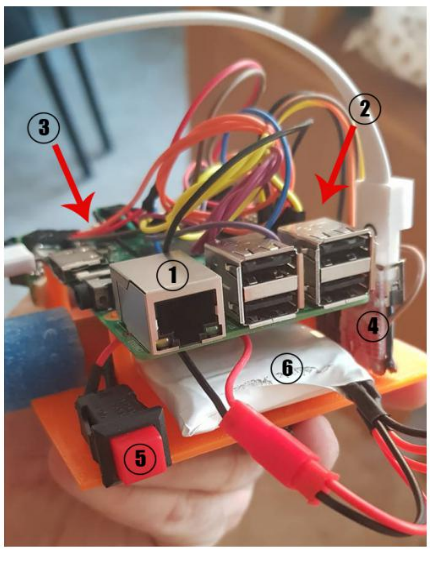
B. Development of the circuit design

In Fig. 6 is represented the circuit design that represents how we connected all the parts described in the hardware components: the ports used from the Raspberry Pi 3, the sensor ports, the motor driver and the engines and how we solved the power problem. We used the DRV8835 motor driver because it is provided with a library for Raspberry Pi 3. This is connected to the engines and the Raspberry Pi 3. Using the step down converter from 7V-20V to USB 5V3A allowed us to use only a single battery for both Raspberry Pi 3 and the engines. The battery is connected to the step down converter, the motor driver and the push button that powers up the robot. The QTR3A sensor is connected directly to the computer.

V. EXPERIMENTAL RESULTS

The maze solver must start from a straight path so that the calibration can be made. This means that the robot does two left turns: one until the line is lost and the second until the line is found. This calibration is repeated for the right side. We present three experimental cases and the difficulty growing for each experiment. In these cases the maze solver has the starting point numbered with 1, 2 or 3 and it follows the red path to find the ”disk”, the goal of the maze.

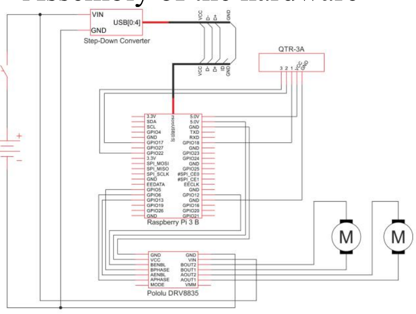
Fig. 5.



ISSN: 2366-1313

Assembly of the hardware

Fig.



6.

Circuit design

The blue path is the shortest path found from the starting point to the goal. The robot performed better then we expected so the experimental results are represented in Table II and are as follows for each entry:

TABLE II
EXPERIMENTAL RESULTS

Entry	*Number of tries*	*Successful tries*	*Accuracy*
1	10	9	90%
2	20	17	85%
3	20	14	70%

A. First experimental result entry 1

This first experimental result presented in Fig. 7 had an accuracy of 90%. The maze solver resolved in the presented mode the maze 9 times from 10 tries. The one time it did not solve it in the expected way, was when on the first turn to right, marked with X, the robot went to left, due to the battery beginning to discharge.

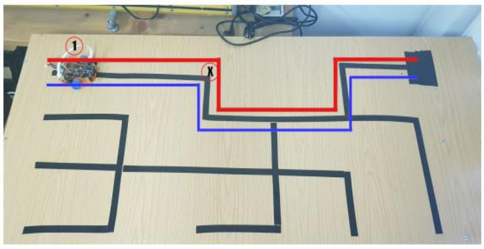


Fig. 7. First experimental result

B. Second experimental result entry 2

The second experimental result presented in Fig. 8 has an accuracy of 85%. The robot solved in the expected way 17 times from 20. One try did not succeed because of the same problem as in the first experiment. The other two were because of the two intersections, marked with X, were the robot got unexpected sequences using the algorithms described in section III.

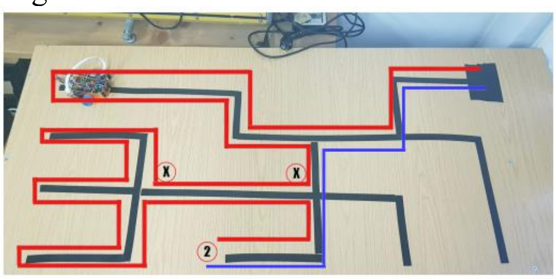


Fig. 8. Second experimental result

C. Third experimental result entry 3

In the third experiment presented in Fig. 9 the robot had a lower accuracy of 70% due to the complexity of traversing the entire maze, although the solution is a simple one. From 20 tries the maze solver got the expected result 14 times. In these 20 tries two of them were not successful because of the battery being discharged, three because of the complex intersections, marked with X and X1. In the first intersection, marked with X1, after coming back from the backward ”E” path, one of the motors did not receive the right command thus generating an unexpected sequence. The last try that was not successful happened because of hardware problems.

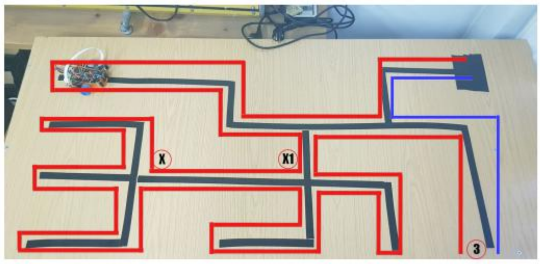


Fig. 9. Third experimental result

VI. CONCLUSIONS

The Line Maze Solver performed with an accuracy between 70% - 90% depending on the complexity of the maze. This has been a total success for us because we wanted to make this robot a low cost one, but this assumed a slightly more complex algorithm using just the three line sensors and not using encoders for the engines. Powering up both the Raspberry Pi 3 and the engines has been a problem in trying to use an eco friendly solution. As a result of experiments, the robot's artificial intelligence behavior has been reached. The labyrinth complexity factors, the batteries life and even proper labyrinth illumination have greatly influenced the behavior of the robot. Because we chose to use this computer, we were able to use the Python 2 development environment. That allowed us to implement the left hand rule and dead-end filling algorithms, along with the DRV8835 library. Changing some speed values in the setMotorsSpeed function allowed us to solve the turn problem using just three sensors. The line maze solver may have further developments. One of the most important would be using the Raspberry Pi 3 camera and implementing image recognition using a machine learning neural network so that the robot would not be restrained to line labyrinth. In this case the robot could learn using a test labyrinth and then exploring new labyrinths. In this way the maze solver would know what decisions to make. The robot using this technology could become even a self-driving car or an robotic arm that can lift heavy blocks.

REFERENCES

[1] W. Matthews, Mazes and labyrinths, 1st ed. New York: Dover Publications, 2013.

[2] Principles of Artificial Intelligence by Nils J. Nilsson, Morgan Kaufmann; Reprint edition (February 15, 1982), pp. XI-XII.

[3] Sandeep Yadav, Kamal Kumar Verma, Swetamadhab Mahanta, The Maze Problem Solved by Micro mouse, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 8958, Volume1, Issue-4, April 2012.

[4] A. M. J. Sadik, M. A. Dhali, H. M. A. B. Farid, T. U. Rashid and A. Syeed, ”A Comprehensive and Comparative Study of MazeSolving Techniques by Implementing Graph Theory,” 2010 International Conference on Artificial Intelligence and Computational Intelligence, Sanya, 2010, pp. 52-56.

[5] Akib Islam, Farogh Ahmad, P. Sathya, SHORTEST DISTANCE MAZE SOLVING ROBOT, IJRET: International Journal of Research in Engineering and Technology, vol. 5 Issue 07, pp. 253-259 July 2016.

[6] Malkit Singh, Rajnish Kumar, Vaibhav Giradkar, Pallavi Bhole, Minu Kumari, Artificially Intelligent Maze Solver Robot, International Research Journal of Engineeringand Technology(IRJET), Volume: 03 Issue: 04, April 2016.

[7] S. Mishra and P. Bande, ”Maze Solving Algorithms for Micro Mouse,” 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, Bali, 2008, pp. 86-93.

[8] Bienias ukasz, Szczepaski Krzysztof, Duch Piotr, Maze Exploration Algorithm for Small Mobile Platforms, Image Processing & Communications, September 2016.

[9] Nitin S. Choubey, MPSTME, SVKMs NMIMS, Shirpur Maharashtra, India, A-Mazer with Genetic Algorithm, International Journal of Computer Applications (0975 8887) Volume 58 No.17, November 2012.

[10] G. Jagga Rao, Y. Chalapathi Rao " Robust Bit Error Rate Optimization for MASSIVE MIMOCEM System using Channel Coding Method "in Volume 8-Issue 4S2, pp. 180-184, March 2019.

[11] M Sumalatha,Dr. P V Naganjaneyulu, and K Satya Prasad ―Low-Power and Area-Efficient FIR Filter Implementation Using CSLA with BEC‖ Journal of Microelectronics, Electromagnetics and Telecommunications,https://doi.org/10.1007/978-981-10-7329-8_14 Jan-2018,Volume -8 , Issue -3 ,ISSN: 0976-4860,pages:137-142, Springer.

[12] A. Ravi,Dr. P. V. Naganjaneyulu ,Dr.M.N. Giriprasad ―SAR images denoising using a novel stochastic diffusion wavelet scheme‖,Springer, Cluster Computing Journal ,USA,July-2017,DOI 10.1007/s10586-017-1001-6,ISSN:1573-7543,Pages:01-09,Springer.

[13] P SekharBabu,Dr. P V Naganjaneyulu, and K Satya Prasad ―Adaptive Beam Forming of MIMO System using Low Complex Selection of Steering Vector‖ International Journal of Advancements in Technology,Jan-2017,Volume -8 , Issue -3 ,ISSN: 0976-4860,pages:01-06, Elsevier.

[14] D. Rajendra Prasad, Dr. P. V. Naganjaneyulu ,Dr. K. Satya Prasad ―A Hybrid Swarm Optimization for Energy Efficient Clustering in Multi-hop Wireless Sensor Network‖ Wireless PersCommun DOI 10.1007/s11277-016-3562-8 Springer Science+Business

Media New York 2016, Pages:2459-2471.

[15] P. Ravikumar, P.V. Naganjaneyulu, K. Satya Prasad ―Peak to Average Power Ratio reduction method for Orthogonal Frequency Division Multiplexing in 4G Communication‖, I J C T A, 8(5), 2015, pp. 1703-1708m.

[16] M Sumalatha, Dr.P .V .NaganjaneyuluandDr.K .Satya Prasad ―ADVANCED FAST FOURIER TRANSFORM USING RADIX-8 BOOTH MULTIPLIER‖ ISSN 2319 - 2518 Int. J. Elec&Electr.Eng& Telecoms. 2015, Special Issue, Vol. 1, No. 3,pp (27-32) November 2015.(Impact factor – 0.454).

[17] U.Sreenivasulu, Dr. P.V.Naganjaneyulu, ―Performance of bit-interleaved coded multiantenna OFDMA systems over space-frequency‖ National Conference on Advanced Communication Systems &Applications,held at Madanapalli 0n June 2013 , INDIA proceedings pp.161-168.

[18] T.Srinivas,Dr.P.V.Naganjaneyulu, P.SandeepMohan,R.Shiva Shankar ,Ch.Surender Reddy ―Face Recognition Using PCA and Bit-Plane Slicing‖ PEIE 2012, LNEE, Springer-Verlag Berlin Heidelberg 2012 , pages: 401-407, 2012.

[19] G. Jagga Rao, Y. Chalapathi Rao, Dr. Anupama Desh Pande "Detection For 6G-NOMA Based Machine Learning Optimization for Successive Adaptive Matching Pursuit Analysis", Q3, pp. 1803-1812, Jan 2020.

[20] P.V. Naganjaneyulu, Dr.K.Satya Prasad, An adaptive blind channel estimation for OFDM system multimedia by improved H∞ approach (for high data rate)‖, International conference on computer intelligence and multimedia applications held at Siva Kasi on Dec. 2008, vol.4, pages:252-256, Dec. 2008, Published also in IEEE computer society in the portal.acm.in.

[21] P.V. Naganjaneyulu, Dr.K.Satya Prasad, ―An adaptive blind channel estimation for OFDM system by improved H∞ approach‖, International Conferences on advanced communication systems held at CIT Coimbatore on Oct. 2007, preceding pages:33-36.

[22] Dr. k. Raju, A. Sampath Dakshina Murthy, Dr. B. Chinna Rao,G. Jagga Rao "A Robust and Accurate Video Watermarking System Based On SVD Hybridation For Performance Assessment" International Journal of Engineering Trends and Technology (IJETT) – Volume 68 Issue 7 - July 2020.

[23] P.V. Naganjaneyulu, Dr.K.Satya Prasad, ―A Study of blind channel estimation in OFDM system by H∞ approach‖, National conference on Communication and signal processing held at IEEE, Mumbai on Feb 2007, NCCSP-07, pages:20-24, Feb. 2007.