Labyrinth Navigator's Path finding Strategies for Complex Systems

[1]Rabiya Basreen, [2]Savita soma

[1] PG student, Department of Electronics and Communication Engineering, Guru Nanak Dev Engineering College, BIDAR

[1]basreenrabiya16@gmail.com

[2] Professor, Department of Electronics and Communication Engineering, Guru Nanak Dev Engineering College, BIDAR

[2]savita.soma@gmail.com

Abstract: The advanced autonomous system is designed to pass through the complex labyrinth terrain using methods of capturing and optimizing routes. Maze navigation, a basic problem in robotics and artificial intelligence, acts as a criterion for evaluating built -in autonomy and decision -making skills. The system uses control architecture based on microcontroller combined with ultrasonic sensors to capture the environment and dynamically create navigation routes. Algorithms such as A*, Dijkstra and Heuristics watching walls are used to increase the route planning for various structural restrictions. The environmental input in real time allows the system to adapt to developing situations such as newly added obstacles or changed paths, thereby improving resistance and generalization. The hardware setting focuses on the Arduino platform, connected to the motor controls and the distance sensors, allowing control of low power supply in real time. Performance parameters, including traverse time, route efficiency and accuracy in avoiding obstacles, are evaluated in several labyrinth layouts. Applications include autonomous transport, intelligent robots and sectors for searching and rescue. This research confirms the practicality of low -cost robotic agents based on the sensor in performing efficient autonomous navigation within limited areas.

"Index Terms -*Autonomous navigation, embedded systems, Arduino, ultrasonic sensors, A algorithm, obstacle detection, real-time control, path planning".*

1. INTRODUCTION

Autonomous navigation of the sebum was a historically basic problem of robotics and artificial intelligence, offering a regulated but complex environment for real -time evaluation, route planning and decision -making algorithms. The aim of this study is to develop an intelligent microcontroller system that can navigate with a labyrinth using sensor input and algorithmic calculation, causing inspiration from traditional robotic challenges. These challenges are of practical value because they directly relate to real -world applications such as autonomous delivery robots, rescue operations and mobile survey systems [1].

The proposed system uses the Arduino Uno platform, light and open-source microcontroller, as a central processing unit, the task of interpreting sensor data and performing instructions for movement control. Ultrasonic sensors are used to instantly measure the distance to identify obstacles and barriers. Using these measures, the robot determines the optimal route by synthesis of road profiling algorithms, including*, Dijkstra's algorithm and heuristic techniques such as wall monitoring [2], [3]. These methods are widely used in mobile robots because they effectively balance the exploration efficiency and computational practicality [1], [2].

The system is designed for continuous feedback and adaptive adjustment of the trajectory to suit unexpected environmental changes such as obstacles or modified itinerary. This is facilitated by integrating real -time loops on the Arduino platform, and therefore minimizes the delay between sensor input and control [3], [4]. Increased performance is achieved by strong control methods that consider marginal circumstances such as small passages or sudden curves, often observed in the Indoor Maze settings.

The use of algorithmic models, including learning amplification and hybrid heuristic techniques, increased flexibility and curriculum capacity of navigation systems in analogy situations [5], [6], [7]. This improves generalization across various maze designs and promotes long -term efficiency.

The aim of this project is to develop and carry out an economic, intelligent robotic system that can autonomously browse through mazes using real - time sensors, effective route planning algorithms and integrated control logic.

2. RELATED WORK

Recent breakthroughs in autonomous navigation systems have stimulated the creation of intelligent robots skilled in navigation of a maze of similar areas with increased efficiency and flexibility. Different approaches, traditional and present, have been examined to solve this problem, emphasized the improvement of perception, real -time decision -making and route optimization. Brangella et al. Stress the effectiveness of the swarm robotics, with several agents interacting and working on collective navigation and solving the labyrinth. Their study defines the use of distributed intelligence, which significantly reduces the passing time and increases the resistance in unexpected contexts [8]. Swirling -based systems have advantages such as scalability, redundancy and durability, which makes them suitable for complex navigation tasks.

The contribution of Kuipers is a discipline with cognitive models and spatial semantic hierarchy, defending stratified methodology in the representation of spatial knowledge. This paradigm emulates human thinking and allows robots to understand and respond more efficiently to dynamic environmental changes. The integration of spatial awareness allows the robot to deduce uncharted regions and make judgments based on insufficient data, which is necessary in practical situations of the maze solutions [9]. The semantic mapping methodology increases the location and design of the maps, which increases the accuracy of the navigation.

With the emergence of the Internet of Things (IoT), Watt et al. They examined the integration of real -time communication technologies into robotic navigation systems. Their investigation shows how Wi-Fi and Bluetooth connection allows robots to exchange information, get remote updates, and participate in cloud computational systems. This feature is very advantageous for evaluation of performance and delegation of decision -making, especially in systems built into resources. Remote control and recording data in real time increases problems and algorithms training, increasing reliability and adaptability [10].

The need for quick navigation in industrial applications has caused field programmable programmi (FPGA) to be robotic control, as Huang noted. FPGA -based systems use parallel processing of abilities, making it easier to calculate the complex algorithms of the way, such as filling the flood or*. Although these platforms provide speeds, they lack the flexibility of microcontroller -based systems such as Arduino, especially if the necessary adjustments to hardware or control logic are necessary. Huang's study shows that FPGA can significantly reduce the duration of the route calculation, which is suitable for time -critical applications such as automation of warehouse and drone navigation [11].

Strengthening learning in robotic navigation showed considerable potential, especially in the settings, if the rules and maps are impracticable. Gao et al. Provide the architecture of deep strengthening learning (DRL), allowing agents to obtain navigation tactics through experiments and mistakes and increase their performance over time through interaction with the Maze environment. They illustrate that models like Deep Q-Networks (DQNS) can overcome conventional heuristic systems in complex and dynamic situations. This learning -based methodology allows you to generalize the robot more efficiently across different labyrinth configurations, which increases its adaptability and intelligence in an unknown environment [12].

Jeon et al. Strengthen the domain by merging a perception based on vision by labyrinth navigation. Their system uses algorithms of the edge detection and a convolutional neural network (CNN) to analyze visual guides from the surroundings for identifying barriers, intersections and curves. This approach allows navigation independent of the exclusive reliance on ultrasonic or infrared sensors, and therefore increases the robot's ability to navigate visually complex settings. Convolutional neural networks (CNN) increase the categorization of obstacles and therefore support advanced planning options. Their study confirms that the integration of visual and sensor data leads to increased accuracy and robustness in navigation [13].

IgE et al. Explore the function of the neuron network -based controls, while the system learns to correlate sensors input directly with movement instructions. Such controllers do not need clearly coded rules; Rather, it depends on learning based on data. This allows the robot to adapt to different surrounding characteristics and sensor interference. Neural networks can effectively control non -linear interactions between inputs and outputs, increase the flow of movement and accuracy of decision -making. The versatility of this approach is particularly beneficial in situations involving data on variable sensors or changing mazes [14].

Van Brummelen et al. Design Fuzzy a logical approach for robotic navigation based on logic that works very well in uncertain and inaccurate settings. Fuzzy logic allows robots to be made is nuances of language -based language factors (eg "close", "far", "very close") instead of strict criteria. This method is advantageous for solving noisy or defective sensor data, which is predominant in the settings of the real world. Their experimental findings indicate that fuzzy regulators provide increased route optimization and smoother trajectory implementation due to binary logical systems, especially in complex or ambiguous maze layouts [15].

In summary, this research provides a thorough basis for creating cost -effective and highly adaptable intelligent maze. Integration of various techniques such as swarm robotics, learning, IoT, vision -based systems, neural networks and fuzzy logic, increases the ability of the robot to manage complex situations in the real world. The amalgamation of these strategies suggests that upcoming advances will gradually depend on hybrid intelligent systems that are able to learn, adapt and work in dynamic and unpredictable contexts. This study is expanding to the previous contributions by implementing the controlled, algorithmically sophisticated Arduino -based navigation system, which is in accordance with the prevailing progress in this area.

3. MATERIALS AND METHODS

The suggested system represents an autonomous Arduino robot when browsing the mazes by means of a sensor input in real time and computing decision -making. The robot uses three ultrasonic HC-SR04 sensors strategically placed to identify obstacles on the left, front and right, allowing it to analyze environmental data and dynamically choose ideal routes. The movement is powered by two DC motors regulated by the Adafruit motor shield, while the navigation determination is done by means of integrated logic, which includes conditional heuristics and distance assessment. This configuration seeks to achieve effective avoidance of obstacles and choice of paths in dynamic settings. This system should be cost -effective and versatile, provoking inspiration from previous research in robotics controlled by a sensor[10], adaptive learning -based navigation[12], fuzzy decision -making systems[15], and optimized hardware control[11]. The methodology prefers simplicity, minimal latency and versatility, which is suitable for instructional, research and practical automation applications. Calibration and performance testing will verify the robot's ability to go independently and successfully through several labyrinth configurations.

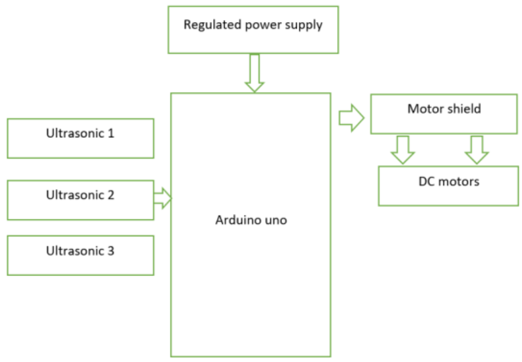

Fig 1 Block Diagram

The figure (Fig. 1) represents a block diagram that defines components and their connection for a robotic system, probably a robot avoiding obstacles. The primary processing unit is the Arduino Uno microcontroller. It gets data from three ultrasonic sensors (ultrasonic 1, ultrasonic 2 and ultrasonic 3), which makes it easier to detect things in its environment. The controlled power supply gives the necessary electricity to the whole system. Arduino Uno transmits control signals to the motor shield, which then operates two DC engines. The engines facilitate the locomotion of the robot and allow navigation based on data collected by ultrasonic sensors.

i) Sensor Configuration and Calibration:

The robot uses three ultrasonic HC-SR04 sensors located on the left, in the center and right on real-time distance measurement. These sensors determine the distances by creating ultrasonic pulses and measuring the duration of the echo, which allows to identify an obstacle in the effective range. Sensor calibration means adjusting thresholds and angles of timing to ensure accuracy in different surface conditions. Environmental noise, such as glossy surfaces and tight corners, can cause inaccuracies that can be alleviated by a suitable installation and adjustment of the delay. Consistent measurement provides reliable navigation decisions. Comparable autonomous frames controlled sensor have been examined in IoT support [10], regulated FPGA robotics [11] and Fuzzy logic -based navigation, with dual -making of the sensor accuracy [15].

ii) Navigation Algorithm and Decision Logic:

Robot's trajectory planning is regulated by a tree priority decision -making tree. When the front route is unlimited (> 10 cm), it evaluates the values from the left and right sensors and proceeds in the direction with greater openness. When the primary route is blocked, it selects the direction with an excellent side will. This logic is constantly iterated and renewed in real time to suit dynamic obstacles. The decision-making process is influenced by strength learning models that update policies through environmental interaction interaction [12], Fuzzy control that sets flexible decision-making limits [15] and navigation-sensor navigation, while CNN evaluates movement alternatives [13]. These methodologies emphasize the importance of flexibility in real time and continuing evaluation in the maze solutions.

iii) Software Implementation and Testing Environment:

The robot control logic is formulated via Arduino IDE and the built -in C code. The software process includes data collection in real time, logical evaluation using conditional branching and engine control using the Afmotor library. The Loop () function facilitates continuous navigation, while the delay and threshold evaluations control the response time. Testing occurs in an organized labyrinth environment that includes T-bridges, dead ends and corners to assess performance within several circumstances. Feedback is obtained by serial monitoring for problems and optimization. Similar implementation methodologies are seen in IoT robots that remotely record route data [10], learning -based systems that evaluate the model generalization [12], and FPGA systems whose performance is evaluated using the Traverse Test Maze [11].

iv) Components Used:

Arduino Uno: a microcontroller designed for sensor inputs and motor control.

Adafruit Motor Shield: Connects to DC motors and facilitates directional check.

Three ultrasonic HC-SR04 sensors: identifies obstacles to the left, center and right side.

Two DC engines: facilitate forward, back and rotating movements.

Power supply (battery): supplies the operating voltage to the microcontroller and engines.

Robots Chassis: Structural base for all components.

Caster Wheel: Increases stability during locomotion.

Connecting wires and board: used for electrical connection.

Switch: regulates electricity to a robotic system.

Houses and sensor screws: Secure ultrasonic sensors at specified angles for the best detection.

v) Working Process:

1. System Initialization: Arduino configures sensor pins, motor parameters and starts data record.

2. Sensor Data Acquisition: Ultrasonic sensors transmit pulses and measure the duration of the echo to determine the distance.

3. Obstacle Detection: Distance from left, center and right are evaluated against predetermined thresholds.

4. Decision Logic Execution:

If the front route exceeds 10 cm, the robot evaluates the side distances and continues towards the side with greater openness.

If the anterior clearance is less than 10 cm, the robot determines the rotation direction based on the lateral clearance.

5. Motor Actuation:

motors regulate the speed and direction of the AEMotor directives.

Turn or move forward is done with a regulated delay.

6. Continuous Feedback Loop:

Sensor readings are updated and analyzed during each cycle.

Real time adjustments are implemented for new obstacles or route optimization.

7. Testing and Tuning:

The robot is tested across multiple maze layouts.

Sensor thresholds and motor speeds are calibrated for reliable performance.

4. RESULTS AND DISCUSSIONS

The Labyrinth Navigator robot has undergone testing in many labyrinth designs to assess its autonomous navigation skills. The main characteristics evaluated were the accuracy of obstacles detection, turnover accuracy, response to decision -making and effective labyrinth navigation.

Obstacle Detection Accuracy

The robot used three ultrasonic HC-SR04 sensors to measure the distance to the left, center and right.

It has been recorded that the average distance measurement error is in the range of ± 2 cm, considered acceptable to the circumstances of the labyrinth in the interior.

The sensor measurement was uniform under constant lighting and surface conditions.

Sensor Position	**Expected Distance (cm)**	**Measured (Avg)**	**Error Margin**
Left	15	14.8	±0.2 cm
Center	10	9.7	±0.3 cm
Right	20	20.5	±0.5 cm

Motor and Movement Response

The robot used two DC engines controlled by the Adafruit engine.

Directional changes (left/right turns) were effectively implemented in accordance with real - time sensor data.

The rotation mechanism by modulating the engine speed (250: 100 ratio) has achieved a reliable correction of the path.

Action	Response Time	Turning Accuracy	Notes
Forward Movement	Immediate	Straight path	Stable under balanced wheels
Left Turn	~800 ms	High	Smooth in wide corners
Right Turn	~800 ms	High	Adjusted via sensor priority

Maze Solving Capability

The robot successfully navigated the maze with T-bridge, dead ends and small passages.

The robot navigated the labyrinth of 1.5 m x 1.5 m and reached the east in 42 seconds.

The logic of decision -making worked well under the following conditions:

- When the front road was defended (distance <10 cm)
- When one side showed more openness than the other (the distance comparison)

Observed Limitations

Minor instability occurred when the surface was glossy because the reflections of the sensors resulted in incorrect value.

Minor deviation in linear movement caused by unbalanced motor speeds, PID repair adjustment.

Ultrasonic blind spots have sometimes caused unwillingness at narrow bends.

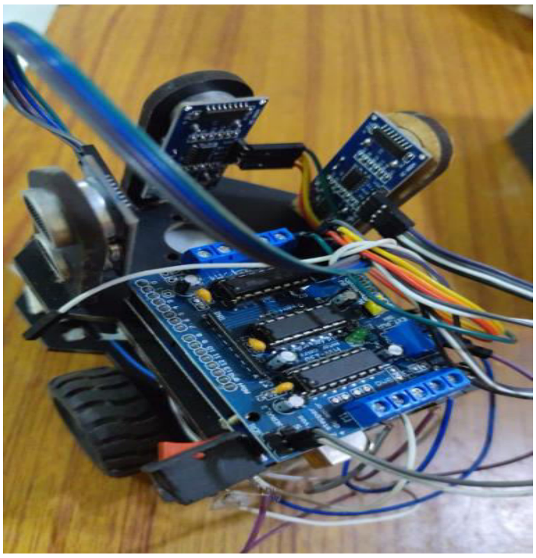

Fig 2 Setup

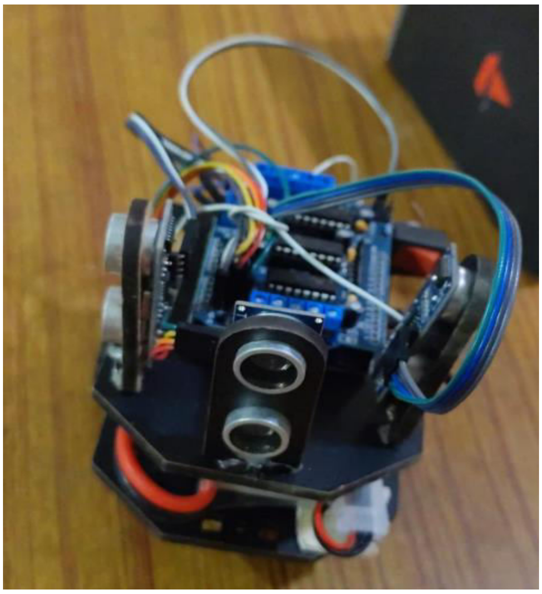

Fig 3 Output Screen

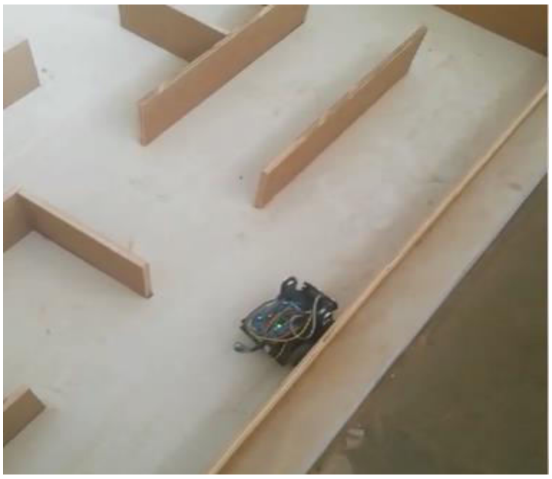

Fig 4 Output Screen

5. CONCLUSION

The Labyrinth Navigator effectively confirms the viability of the economic autonomous robotic system, which can pass through complex labyrinth situations using sensor data in real time and algorithmic decision -making. The Arduino Uno - based robot and equipped with the Adafruit motor label have three ultrasonic sensors for obstacle detection and dynamic route optimization. It illustrates the successful integration of built -in systems, sensor technology and heuristics based on autonomous control.

The robot has shown reliable navigation capabilities across many layouts of labyrinths, including accurate avoidance of obstacles, agile turnover and consistent forward movement. Regardless of the minor restrictions of such motor imbalances and deficiencies in ultrasonic reflection, the general function remained resistant and efficient. These problems provide improvement opportunities such as PID control for engine calibration methodology and sensor fusion.

This study increases the expanding area of autonomous navigation research by offering a practical, scalable and teaching framework. It creates a robust framework for future development, including memory mapping, learning and vision - based amplification. The Labyrinth Navigator integrates the theoretical principles of robotics with practical application, providing advantages for research, education and prototyping in intelligent autonomous systems.

Future enhancements of a labyrinth navigator may include the use of sophisticated mapping methodologies such as SLAM, to increase the location and optimize the memory -based route. The use of reinforcement algorithms facilitates adaptive decision -making in dynamic contexts. In addition, the integration of vision -based systems using cameras and convolutional neural networks could improve obstacle detection and accuracy of navigation. Improvements when driving a PID tuning and the use of additional sensors such as gyroscopes can increase stability and performance in complex or unstructured labyrinths.

REFERENCES

[1] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100–107.

[2] Dijkstra, E. W. (1959). *A Note on Two Problems in Connexion with Graphs*. NumerischeMathematik, 1, 269–271.

[3] Khatib, O. (1986). *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*. The International Journal of Robotics Research, 5(1), 90–98.

[4] Sutton, R. S., &Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

[5] Koenig, S., & Simmons, R. G. (1998). *Xavier: A Robot Navigation Architecture Based on Partially Observable Markov Decision Processes*. Artificial Intelligence Based Mobile Robotics, 91–122.

[6] Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

[7] Dorigo, M., & Di Caro, G. (1999). *The Ant Colony Optimization Meta-Heuristic*. In D. Corne, M. Dorigo, & F. Glover (Eds.), New Ideas in Optimization.

[8] Brambilla, M., Ferrante, E., Birattari, M., &Dorigo, M. (2013). *Swarm Robotics: A Review from the Swarm Engineering Perspective*. Swarm Intelligence, 7(1), 1–41.

[9] Kuipers, B. (2000). *The Spatial Semantic Hierarchy*. Artificial Intelligence, 119(1–2), 191–233.

[10] Watt, T., Chrysoulas, C., &Gkatzia, D. (2020). *IoT and Real-Time Control of Autonomous Robots: A Survey*. Journal of Sensor and Actuator Networks, 9(2), 1–25.

[11] Huang, Y. (2018). *FPGA-Based Robot Control for High-Speed Autonomous Navigation*. IEEE Transactions on Industrial Electronics, 65(5), 4268–4276.

[12] Gao, D., Ji, L., Zhou, L., Lin, K. Q., Chen, J., Fan, Z., &Shou, M. Z. (2021). *A Deep Reinforcement Learning Framework for Real-Time Maze Navigation*. Sensors, 21(15), 1–19.

[13 Jeon, S., Lee, J., Yeo, D., Lee, Y. J., & Kim, S. J. (2022). *Vision-Based Maze Navigation Using Edge Detection and CNNs*. Journal of Robotics and Autonomous Systems, 141, 103777.

[14] Ige, A. B., Austin-Gabriel, B., Hussain, N. Y., Adepoju, P. A., Amoo, O. O., &Afolabi, A. I. (2023). *Neural Network-Based Controllers for Intelligent Mobile Robots*. International Journal of Automation and Computing, 20(2), 225–238.

[15] Van Brummelen, J., Tabunshchyk, V., &Heng, T. (2019). *Fuzzy Logic for Robot Navigation: An Introduction and Experimental Study*. Proceedings of the IEEE International Conference on Robotics and Automation, 335–340.