Campus Guide Robot

Amrutha Suresh
Dept. of ECE
College of Engineering
Kallooppara
Thiruvalla, Kerala, India
amruthasuresh009@gmail.com

Bibin Binu
Dept. of ECE
College of Engineering
Kallooppara
Thiruvalla, Kerala, India
bibinbinu9510@gmail.com

Karthik Prakash
Dept. of ECE
College of Engineering
Kallooppara
Thiruvalla, Kerala, India
karthikprakash2255@gmail.com

Nandana S
Dept. of ECE
College of Engineering
Kallooppara
Thiruvalla, Kerala, India
ns930840@gmail.com

Thomas George
Dept. of ECE
College of Engineering
Kallooppara
Thiruvalla, Kerala, India
thomasgeorge2558@gmail.com

Dr.Deepa J
Professor,Dept. of ECE
College of Engineering
Kallooppara
Thiruvalla, Kerala, India
principal@cek.ac.in

***Abstract*—The deployment of autonomous mobile robots for assisting people at smart workplaces and universities has increased tremendously over the years. Most existing solutions have utilized expensive location methods such as SLAM with complex sensors and powerful computer processors. This project presents an option to reduce costs while providing a more efficient way of providing a campus indoor guide robot on a defined route. A Raspberry Pi 4 will be used for higher level reasoning and user interaction through a touch-screen while the Arduino Nano will be used for achieving lower-level motor and sensor control, achieving a highly accurate solution. A user will be able to select a destination from a touch-screen and ultrasonic sensors will provide real-time obstacle detection and allow for safe travel through the environment. Motion will be provided by using DC geared motors controlled by an L298N motor driver. The designed system will reduce compute power needs, create ease of implementation, and increase the dependability of the system. The results of testing performed in corridor systems showed consistent results in navigation, high rates of successful obstacle avoidance, and stable operation of the system. The results indicated that a defined path way to navigate could be utilized by institutions to provide guide assistance to people.**

***Index Terms*—Autonomous Robotics, Indoor Navigation, Differential Drive Robot, Predefined Path Planning, Embedded Systems, Human–Robot Interaction.**

I. INTRODUCTION

The rapid advancement of robotics and embedded systems has led to the development of intelligent service devices that assist people in performing everyday tasks [17]. Automation technologies are increasingly being adopted in many sectors such as educational institutions, hospitals, airports, and office environments to improve operational efficiency and user experience. One important application of these technologies is indoor navigation, where automated systems help visitors find their way inside large buildings while reducing the need for manual assistance [3].

Many university campuses consist of multiple interconnected buildings that include lecture halls, laboratories, administrative offices, and service facilities spread across a large area. For new students, visitors, and external participants, navigating such environments can often be confusing due to complex layouts and inconsistent building structures [2]. Traditional guidance methods such as static signboards, printed maps, or help desks may not always provide sufficient support, and they frequently require staff members to be available for providing directions. These limitations have encouraged the development of robotic guidance systems capable of offering interactive and autonomous navigation assistance.

Although advanced navigation techniques such as Simultaneous Localization and Mapping (SLAM) are widely used in mobile robotics, they often require expensive sensors, high computational power, and complex algorithms. Such requirements can make SLAM-based systems less suitable for low-cost academic deployments and small-scale institutional environments [16].

In contrast, most educational campuses have relatively structured and predictable layouts. This characteristic makes it feasible to use predefined navigation paths instead of dynamically generating navigation maps. In a predefined path system, the robot follows calibrated routes stored in its internal database to reach different destinations. This approach reduces computational complexity and improves reliability of motion. Similar strategies are commonly used in Automated Guided Vehicles (AGVs), where vehicles follow fixed paths within structured environments such as warehouses, hospitals, and industrial facilities [10].

The proposed Campus Guide Robot adopts an improved predefined navigation path model specifically designed for campus environments. The system utilizes a hybrid embedded architecture consisting of an Arduino Nano and a Raspberry Pi 4. In this architecture, the Raspberry Pi 4 manages high-level functions such as the user interface, destination selection, and path planning, while the Arduino Nano handles real-time tasks including motor control and sensor-based movement [16]. This separation of responsibilities improves timing accuracy and system efficiency. Communication between the two controllers is achieved through a TCP/IP-based network interface.

To enhance user interaction, the robot incorporates a touchscreen interface that allows visitors to easily select their desired destination without requiring technical knowledge. For safety during navigation, ultrasonic sensors are used to detect obstacles in the robot's path. When an obstacle is detected, the robot temporarily stops and resumes movement once the path is cleared. Motion control is achieved using DC geared motors driven by an L298N motor driver, ensuring stable movement and sufficient torque for indoor navigation.

The main contribution of this work is the development of a reliable and cost-effective campus navigation system that operates without complex localization algorithms. By combining calibrated motion control with a modular embedded architecture, the proposed system provides a practical solution for indoor guidance. The design emphasizes affordability, ease of implementation, and scalability, making it suitable for educational institutions with limited resources.

II. RELATED WORKS

Autonomous navigation for mobile robots has been studied extensively over the past few decades. Different approaches have been proposed based on application needs and environmental conditions. One of the most researched methods is Simultaneous Localization and Mapping (SLAM). This method allows a robot to create a map of an unknown environment while estimating its own position in real time. SLAM systems usually use sensors like LiDAR, cameras, and inertial measurement units for accurate location tracking and obstacle avoidance. While these systems offer flexibility and autonomy, they need substantial computational power and complicated sensor fusion algorithms. The hardware costs and complexity of implementation make SLAM less ideal for low-cost indoor service robots in structured environments such as educational campuses[17].

Vision-based navigation has gained attention in recent years due to improvements in image processing and embedded computing platforms. These systems use cameras to detect visual landmarks, floor patterns, or features in the environment. This allows robots to find direction and position using computer vision techniques. Researchers have created Raspberry Pi-based robots with OpenCV libraries for tasks like object detection and path tracking. Vision-based methods give detailed environmental information and support smart decision-making. However, their effectiveness greatly relies on lighting conditions and camera setup. Changes in light, crowded spaces, and obstacles often lower navigation reliability, especially in indoor corridors where visual features tend to repeat[10].

Another common approach is line-following navigation. Here, robots follow predefined tracks marked on the floor using colored lines or reflective materials. Infrared sensors identify the path and adjust the robot's wheel movements accordingly. Line-following robots are known for high accuracy and straightforward control algorithms, which makes them popular in industrial automation and educational demonstrations. However, this method needs physical changes to the environment and requires ongoing maintenance of the guiding lines. In large campuses or public areas, setting up and maintaining floor markings can be impractical, limiting scalability[16].

Indoor localization methods using RFID tags, Bluetooth beacons, or wireless signal triangulation have also been explored for service robots. In these systems, reference nodes are placed in the environment, allowing the robot to estimate its location based on signal strength or tag detection. While these techniques improve localization accuracy, they add infrastructure costs and maintenance tasks. Signal interference and environmental noise can also impact system performance, particularly in busy indoor areas.

Compared to these complex methods, predefined path navigation is widely used in automated guided vehicles in warehouses, hospitals, and delivery systems with fixed routes. Instead of creating maps on the fly, the robot follows a set sequence of programmed movements to known destinations. This approach significantly reduces computational needs while enhancing reliability in structured settings. Several studies show that pairing a high-level processor with a microcontroller boosts system efficiency by dividing decision-making tasks from real-time motor control. The high-level controller manages user interactions and navigation logic, while the microcontroller ensures precise timing and actuator control[3].

From the review of existing research, it is clear that many navigation systems focus on flexibility and autonomy, often sacrificing cost and simplicity of implementation. For campus environments where paths are often fixed, a simpler navigation strategy can deliver similar performance with less complexity. Therefore, the proposed campus guide robot uses a predefined path navigation method integrated with a hybrid Raspberry Pi and Arduino setup to create a reliable, affordable, and user-friendly indoor guidance system.

III. PROPOSED SOLUTION

The proposed system presents a campus guide robot designed to provide reliable indoor navigation using a predefined path-based navigation strategy. The primary objective of the system is to assist visitors in reaching selected campus locations through an interactive, cost-effective robotic platform while avoiding the complexity associated with fully autonomous mapping techniques such as SLAM [3]. Instead of dynamically generating maps or performing continuous localization, the robot follows a set of previously measured and stored motion sequences that correspond to specific destinations within the campus environment [16]. This approach is particularly suitable for structured indoor environments such as academic corridors and building hallways where navigation routes remain relatively constant.

The overall architecture of the system follows a hybrid embedded control model consisting of a Raspberry Pi 4 and an Arduino Nano. The Raspberry Pi acts as the high-level processing unit responsible for handling user interaction, path management, and decision-making tasks. A touchscreen display connected to the Raspberry Pi provides a graphical user interface through which users can select their desired destination. Once a destination is selected, the Raspberry Pi retrieves the corresponding predefined path stored in its memory [4]. These paths are created during a calibration phase by measuring distances between important locations on the campus and converting them into a sequence of motion commands such as forward movement, left turn, right turn, and stop operations.

The Arduino Nano functions as the low-level control unit responsible for executing the motion commands received from the Raspberry Pi [5]. This separation of control improves system efficiency because real-time motor control requires precise timing, which is more effectively handled by a microcontroller than by an operating-system-based processor. Communication between the Raspberry Pi and the Arduino Nano is established through serial UART communication, ensuring reliable transmission of control commands with minimal delay. Each command transmitted by the Raspberry Pi represents a specific movement instruction that the Arduino interprets and converts into motor control signals.

Robot locomotion is achieved using DC geared motors arranged in a differential drive configuration [8]. The motors are driven using an L298N motor driver module, which provides the required current amplification and enables bidirectional rotation of the wheels. By controlling the direction and speed of the individual motors, the robot can move forward, perform left and right turns, and stop accurately at predefined checkpoints. The use of geared motors provides sufficient torque and stable motion, which is essential for supporting the robot structure and the touchscreen interface during navigation.

To enhance operational safety, an ultrasonic sensor is integrated into the system for obstacle detection [2]. The sensor continuously measures the distance between the robot and nearby objects using the time-of-flight principle of ultrasonic waves. When an obstacle is detected within a predefined safety threshold, the Arduino Nano immediately stops the motors regardless of the current navigation command. Once the obstacle is cleared, the robot automatically resumes movement, allowing safe operation in environments where pedestrians may cross the robot's path.

The predefined navigation algorithm forms the core of the proposed solution. Each destination within the campus is represented as a sequence of calibrated motion commands derived from physical measurements of the campus pathways [10]. During operation, these commands are executed sequentially to ensure predictable and repeatable navigation behavior. Since the system does not rely on computationally intensive mapping or localization algorithms, the overall computational load is significantly reduced. This allows the robot to operate efficiently on low-power embedded hardware while maintaining stable and reliable navigation performance [17].

IV. COMPONENTS AND FUNCTIONS

The Raspberry Pi 4 acts as the primary processing unit of the robot and serves as the central controller for high-level system operations. It manages the graphical user interface displayed on the touchscreen, processes user input, and determines the predefined navigation path corresponding to the selected destination. In addition, the Raspberry Pi supervises communication with the Arduino Nano through serial communication and coordinates the execution of navigation commands. Its ability to run a Linux-based operating system enables flexible software development using programming languages such as Python and also allows future system extensions, including artificial intelligence modules, remote monitoring, or network-based control [4].

The Arduino Nano functions as the low-level embedded controller responsible for real-time hardware control. Since precise timing is essential for motor actuation and sensor monitoring, the Arduino executes motion commands received from the Raspberry Pi without the delays associated with operating systems. It generates Pulse Width Modulation (PWM) signals for controlling motor speed and uses digital output pins to regulate motor direction. Additionally, the Arduino continuously monitors the ultrasonic sensor data and performs immediate safety responses, such as halting the robot when an obstacle is detected within a predefined safety range [5].

A touchscreen display is integrated into the system to provide a user-friendly human–machine interface. Through this graphical interface, users can easily select their desired campus destination from a list of available options. The touchscreen simplifies the interaction process and allows even first-time users to operate the robot without prior technical knowledge. Once a destination is selected, the interface signals the Raspberry Pi to retrieve and execute the corresponding predefined navigation sequence. This design eliminates the need for external control devices such as mobile applications or remote controllers, thereby improving accessibility and convenience [3].

Obstacle detection and collision prevention are achieved using an ultrasonic sensor. The sensor operates by transmitting ultrasonic waves and measuring the time taken for the reflected signal to return after striking an object. Based on the measured time-of-flight, the distance between the robot and the detected object is calculated. Continuous distance monitoring allows the robot to pause its movement when an obstacle or pedestrian appears in its path, thereby ensuring safe operation within crowded indoor environments such as university corridors [6].

The robot uses DC geared motors as the primary actuators responsible for locomotion. These motors are selected because they provide high torque at relatively low rotational speeds, which is suitable for controlled indoor navigation. The integrated gear mechanism ensures smooth motion and reduces sudden jerks during acceleration or directional changes. Two motors arranged in a differential drive configuration allow the robot to move forward, rotate left or right, and stop precisely by independently controlling the rotational speed of each wheel [7].

A motor driver module is placed between the Arduino Nano and the motors to provide sufficient current required for motor operation. Since microcontroller output pins cannot directly supply the current needed by DC motors, the motor driver acts as a power interface using an H-bridge circuit. This circuit enables bidirectional motor rotation and protects the control electronics from voltage spikes generated by motor back electromotive force (EMF) [11].

The robot chassis and wheel assembly provide the mechanical foundation of the system and support all electronic components. The chassis maintains proper alignment of motors, sensors, and control modules, ensuring structural stability during operation. The wheels enable smooth and stable movement across indoor floor surfaces, while proper weight distribution across the chassis prevents imbalance during turning and navigation maneuvers [16].

V. WORKING

The operation of the proposed campus guide robot is based on the coordinated interaction between user input, navigation control, sensor monitoring, and motor actuation. The system follows a hierarchical control architecture in which the Raspberry Pi 4 performs supervisory and decision-making functions, while the Arduino Nano manages real-time hardware control. When the robot is powered on, an initialization process is executed in which all hardware modules and communication interfaces are configured. During this stage, serial communication between the Raspberry Pi and the Arduino Nano is established, the touchscreen graphical interface is initialized, and the motor driver and sensors are set to standby mode to ensure safe system startup [3].

After initialization, the touchscreen interface displays a list of available campus destinations such as departments, laboratories, or administrative offices. The user selects the desired destination through the graphical interface. This selection is processed by the Raspberry Pi, which retrieves the corresponding predefined navigation path stored in the system memory. Each path consists of a sequence of calibrated movement commands derived from physical measurements of campus corridors and intersections obtained during the calibration phase [6].

Once the navigation path is selected, the Raspberry Pi sequentially transmits motion commands to the Arduino Nano through serial communication. These commands represent specific actions such as forward movement, left turn, right turn, and stop operations. Upon receiving each command, the Arduino Nano generates appropriate control signals for the motor driver module. The motor driver supplies the required current to the DC geared motors, allowing the robot to move according to the received instructions. The locomotion mechanism follows a differential drive configuration, where independent control of the left and right motors enables accurate straight-line motion and precise turning maneuvers [16].

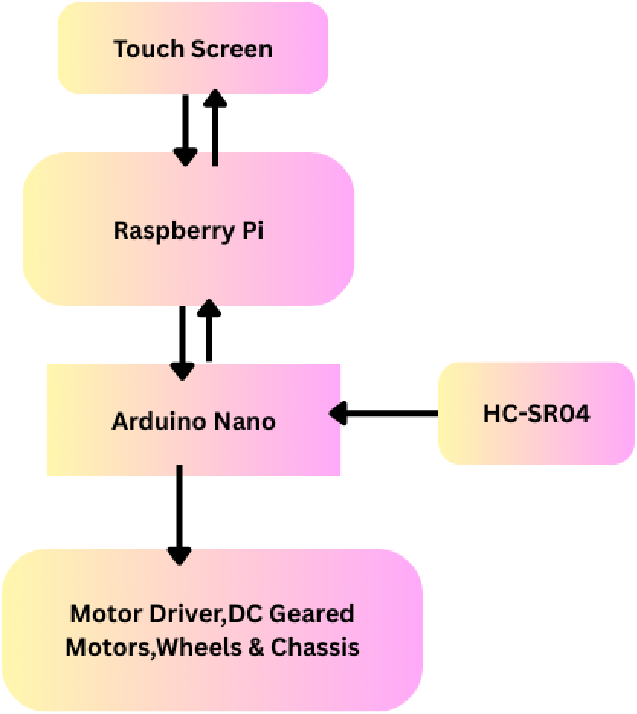

Fig. 1: Workflow diagram of the campus guide robot

During navigation, environmental monitoring is continuously performed using an ultrasonic sensor positioned at the front of the robot. The sensor periodically transmits ultrasonic pulses and measures the time taken for the reflected signals to return after striking nearby objects. Using this time-of-flight measurement, the Arduino Nano calculates the distance between the robot and potential obstacles [6]. If an obstacle is detected within a predefined safety threshold, the Arduino immediately interrupts motor operation and stops the robot to prevent collisions. This safety mechanism operates independently of the navigation commands, ensuring real-time obstacle detection and protection [5].

When the obstacle is removed or moves beyond the defined safety distance, the Arduino resumes execution of the remaining motion commands received from the Raspberry Pi. The robot continues executing the predefined navigation sequence step by step until all commands have been completed. Upon reaching the final command, the motors automatically stop, indicating that the robot has arrived at the selected destination. A completion message may then be displayed on the touchscreen interface, after which the system returns to standby mode and waits for the next user request [10].

The use of predefined navigation paths ensures predictable and repeatable robot behavior while significantly reducing computational complexity, as the system does not require real-time mapping or localization algorithms [10]. Furthermore, the division of responsibilities between the Raspberry Pi and

Arduino Nano enables efficient parallel operation, where high-level decision-making and real-time control tasks are handled independently. This architecture enhances system stability, ensures safe operation, and provides reliable navigation performance within structured indoor campus environments [17].

VI. SYSTEM FLOW

The system flow of the proposed campus guide robot describes the sequence of operations performed from system initialization to the completion of navigation. The robot follows a structured control process in which user interaction, navigation decision-making, sensing, and motion control are executed in a coordinated manner to ensure reliable and safe operation [3].

The process begins when power is supplied to the robot, activating both the Raspberry Pi 4 and the Arduino Nano. During the initialization stage, all hardware modules such as the touchscreen display, ultrasonic sensor, motor driver, and communication interfaces are configured. Serial communication between the Raspberry Pi and Arduino Nano is established to enable reliable command transmission between the two controllers. During this stage, the motors remain inactive until a valid navigation command is received [16].

After initialization, the Raspberry Pi loads the graphical user interface onto the touchscreen display. The system then enters a waiting state where it continuously monitors for user input. The interface presents a list of available campus destinations as selectable options, allowing users to interact directly with the robot. When a destination is selected, the Raspberry Pi retrieves the corresponding predefined navigation path stored in system memory. Each navigation path consists of a sequence of calibrated movement instructions representing measured distances and turning actions required to reach the selected location [13], [6].

Once the navigation path is determined, the Raspberry Pi begins transmitting movement commands sequentially to the Arduino Nano through serial communication. Each command corresponds to a specific movement instruction such as forward motion, left turn, right turn, or stop. The Arduino Nano interprets these commands and generates appropriate control signals for the motor driver module. The motor driver then supplies the required current to the DC geared motors, enabling the robot to move according to the received instructions. Movement is executed step by step to ensure stable and accurate navigation [10].

During robot movement, the ultrasonic sensor continuously monitors the environment to detect potential obstacles. The sensor measures the distance to nearby objects using the ultrasonic time-of-flight principle, and the measurements are processed in real time by the Arduino Nano [8]. If an obstacle is detected within the predefined safety threshold, the Arduino immediately interrupts the navigation process by stopping the motors. The robot remains in a paused state until the obstacle moves away or the path becomes clear. Once the obstacle is cleared, the Arduino resumes execution of the remaining navigation commands without restarting the entire navigation sequence [9].

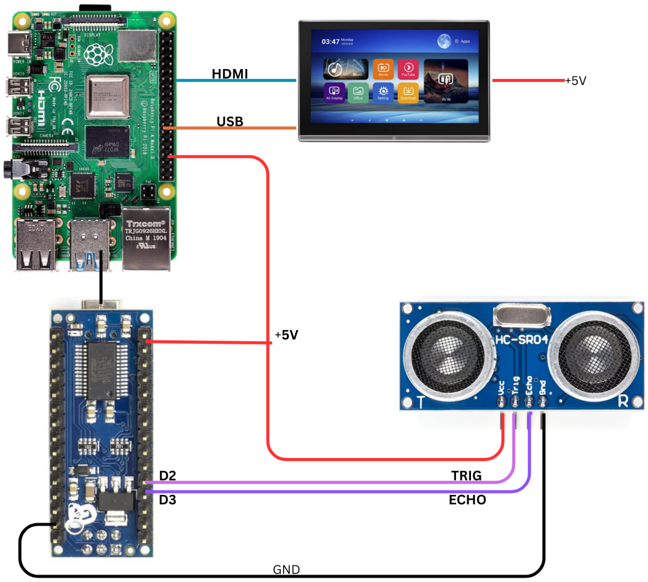

Fig. 2: Circuit diagram of the campus guide robot hardware

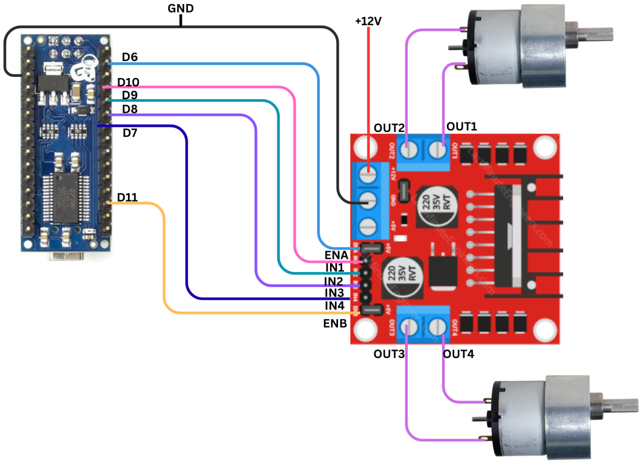

Fig. 3: Control and communication architecture of the system

The robot continues executing commands sequentially until all instructions within the predefined path are completed. When the final movement command is executed, the Arduino stops the motors and sends a completion signal back to the Raspberry Pi [16]. The Raspberry Pi then updates the system status on the touchscreen display to indicate that the robot has successfully reached the selected destination. After completion, the system returns to standby mode, allowing the robot to accept a new navigation request from the next user [9].

VII. RESULTS AND PERFORMANCE EVALUATION

The proposed campus guide robot was experimentally tested in an indoor campus environment to evaluate its navigation performance, obstacle detection capability, and overall system reliability. The robot successfully guided users from a starting location to selected campus destinations by executing predefined motion sequences stored in the system memory. Since the navigation paths were calibrated during the setup phase, the robot was able to follow measured distances and turning angles with minimal positional deviation [2], [12].

To evaluate system performance, multiple experimental trials were conducted in a structured corridor environment. During each trial, the robot was instructed to navigate between predefined checkpoints representing typical campus destinations. Key performance metrics such as navigation accuracy, time required to reach the destination, obstacle avoidance success rate, and overall system reliability were recorded.

The experimental observations indicated that the robot maintained consistent navigation behavior across repeated trials. The predefined path navigation method allowed the robot to follow calibrated movement commands accurately while maintaining stable locomotion using the differential drive mechanism. The average navigation accuracy observed during testing was approximately 92–95%, with only minor deviations caused by wheel slippage or surface irregularities on the floor.

The average time required to reach a selected destination within the experimental corridor setup was approximately 18–25 seconds depending on the path length and number of turning points. These results demonstrate that the predefined navigation method provides efficient movement without requiring computationally intensive mapping or localization algorithms.

Obstacle detection performance was evaluated by intentionally placing objects in the robot's path during navigation. The ultrasonic sensor successfully detected obstacles within the predefined safety distance in 19 out of 20 test trials, resulting in an obstacle avoidance success rate of approximately 95%. When an obstacle was detected, the Arduino Nano immediately halted the motors and resumed motion once the path became clear, ensuring safe interaction with pedestrians and dynamic obstacles [8].

System reliability was also assessed by repeating navigation experiments multiple times under similar environmental conditions. Out of 20 navigation trials, the robot successfully reached the intended destination in 19 cases without requiring manual intervention, indicating a reliability of approximately 95%. These results confirm that the hybrid control architecture using the Raspberry Pi and Arduino provides stable and dependable system operation.

The touchscreen interface provided smooth and intuitive interaction, allowing users to easily select destinations without requiring technical expertise. Communication between the Raspberry Pi and Arduino Nano remained stable throughout system operation, enabling efficient coordination between navigation control and motor execution [4], [9].

TABLE I: Performance Comparison Between Proposed System and Autonomous Navigation System

Parameter	**Proposed System (Predefined Path Navigation)**	**Autonomous Navigation System (LiDAR/SLAM Based)**
Navigation Method	Predefined stored paths with calibrated movements	Real-time mapping and localization (SLAM)
System Complexity	Low	High
Hardware Requirement	Raspberry Pi, Arduino Nano, Ultrasonic Sensor, Motors	High-end processor, LiDAR, multiple sensors, cameras
Cost	Low-cost implementation	Expensive due to LiDAR and processing units
Setup Requirement	Initial path calibration required	Requires environment mapping and tuning
Computational Load	Low to moderate	Very high
Accuracy in Structured Environment	High and repeatable	High but depends on mapping accuracy

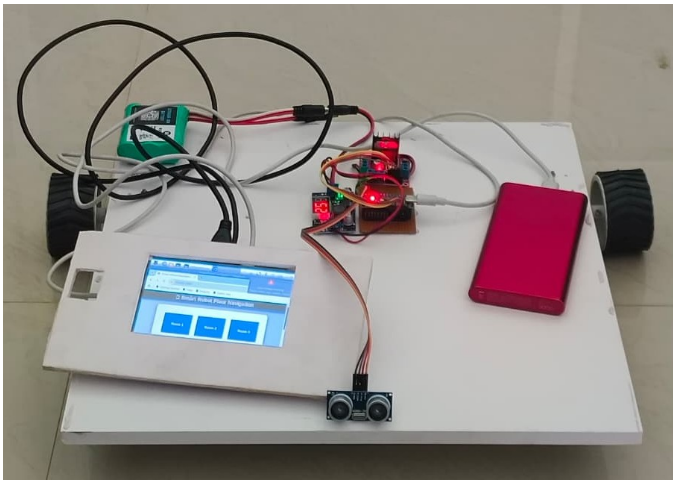

Fig. 4: Experimental result of the campus guide robot navigation

Overall experimental results indicate that the proposed system provides stable navigation, reliable obstacle detection, and consistent destination guidance within structured indoor environments. Although the system does not employ advanced localization algorithms such as SLAM, the predefined path approach offers a practical and cost-effective solution for campus guidance applications. The results demonstrate that reliable indoor navigation can be achieved using low-cost hardware and simplified navigation strategies, making the system suitable for educational institutions with limited technical resources [15].

VIII. FUTURE SCOPE

Although the proposed campus guide robot successfully demonstrates reliable navigation using predefined paths, several improvements can be incorporated in future developments to enhance system intelligence, flexibility, and practical usability. One possible extension is the integration of autonomous localization techniques such as visual odometry or lightweight Simultaneous Localization and Mapping (SLAM) algorithms [17]. Incorporating such methods would enable the robot to adapt to environmental changes and navigate dynamically without requiring manual recalibration whenever the campus layout is modified [14].

Another important enhancement involves integrating a camera-based vision system. A camera module can provide additional perception capabilities, enabling functions such as human detection, facial recognition for personalized interaction, and signboard or landmark recognition for automatic destination verification [9]. Combining vision-based perception with the existing ultrasonic obstacle detection system would significantly improve environmental awareness and navigation safety in crowded indoor spaces [8].

Voice interaction can also be introduced to make the system more accessible and user-friendly. By integrating speech recognition and voice processing modules, users could provide destination commands through voice input rather than relying only on touchscreen interaction [17]. This feature would improve usability for visually impaired users and visitors who may find touch-based interfaces difficult to operate.

Future versions of the robot could also incorporate wireless communication through Wi-Fi or Internet of Things (IoT) platforms. Such connectivity would allow remote monitoring, centralized control, and real-time system updates [8]. Campus administrators could track robot location, update navigation paths remotely, and monitor system performance through a network interface. Additionally, cloud integration could enable data collection and analysis to optimize navigation routes and improve campus guidance services over time [16].

Energy management and operational autonomy can be further enhanced by implementing an automatic docking and charging mechanism. In this approach, the robot would automatically navigate to a charging station when the battery level falls below a predefined threshold, enabling continuous long-term operation without manual intervention [2]. Furthermore, incorporating feedback sensors such as wheel encoders or inertial measurement units (IMU) could improve motion accuracy by enabling closed-loop control instead of relying solely on time-based motor commands [17].

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[2] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011.

[3] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[4] Raspberry Pi Foundation, "Raspberry Pi 4 Model B Documentation," [Online]. Available: https://www.raspberrypi.org

[5] Arduino, "Arduino Nano – Technical Specifications," Arduino Official Documentation, [Online]. Available: https://www.arduino.cc

[6] HC-SR04 Ultrasonic Sensor Datasheet, "Ultrasonic Distance Measuring Module," pp. 1–5.

[7] STMicroelectronics, "L298N Dual H-Bridge Motor Driver Datasheet," Motor Driver Technical Manual.

[8] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, Sep./Oct. 1989.

[9] M. Quigley et al., "ROS: An open-source Robot Operating System," in *Proc. IEEE Int. Conf. Robotics and Automation Workshop*, 2009.

[10] D. Santos, V. Lopes, and J. Almeida, "Indoor service robot navigation using predefined paths," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, pp. 1–10, 2017.

[11] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016.

[12] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, Jun. 1991.

[13] A. Kelly, "Mobile robotics: Mathematics, models, and methods," Cambridge University Press, 2013.

[14] H. Choset et al., *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA, USA: MIT Press, 2005.

[15] M. S. Couceiro, R. P. Rocha, and N. M. Ferreira, "A novel multi-robot exploration approach based on particle swarm optimization," *IEEE International Conference on Robotics and Automation*, pp. 1920–1925, 2011.

[16] A. Pandey and D. R. Parhi, "Optimum path planning of mobile robot in unknown static and dynamic environments," *International Journal of Robotics Research*, vol. 36, no. 5–7, pp. 635–651, 2017.

[17] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, no. 3–4, pp. 143–166, 2003.