IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Autonomous Waypoint-Based Navigation System for Mobile Robots

Ayush Panigrahi[1], Samar Yadav[2], Aatif Ansari[3], Rishabh Tiwari[4] , Ravindra Shelkikar[5]

Student[1], Student[2], Student[3], Student[4], Associate Professor[5]
Department of Electronics and Telecommunication Engineering
Terna Engineering College , Navi Mumbai , India

Abstract**—**This paper presents an integrated framework for autonomous waypoint-based navigation of a four-wheeled mobile robot operating in GPS-denied indoor environments. The system integrates encoder-based odometry with visual landmark recognition to achieve reliable localization and navigation in structured indoor spaces. The robot platform employs DC geared motors with encoder feedback, an onboard RGB camera, and a Raspberry Pi–based embedded computing system running ROS 2. Sensor fusion is performed using an Extended Kalman Filter to combine odometry and visual observations for improved pose estimation. Navigation is organized through global waypoint planning using Dijkstra's algorithm and local obstacle avoidance based on the Dynamic Window Approach. The system is validated through both simulation and real-world experimental deployment. Hardware experiments conducted in indoor environments demonstrate successful multi-waypoint navigation and reliable obstacle avoidance under practical conditions such as sensor noise and varying illumination. The results confirm that the proposed framework achieves accurate localization, reduced odometric drift, and real-time performance on embedded robotic platforms.

***Index Terms* -** Autonomous Navigation, Mobile Robotics, Waypoint Planning, Visual Landmark Recognition, Odometry, Indoor Navigation, Sensor Fusion, Extended Kalman Filter, Path Planning, Dynamic Window Approach, Dijkstra Algorithm, ROS 2, Simulation and Real-World Validation

I. INTRODUCTION

Autonomous navigation in indoor environments is a fundamental problem in mobile robotics, particularly in applications where Global Positioning System (GPS) signals are unavailable or unreliable. Indoor spaces such as buildings, tunnels, and industrial facilities often experience signal attenuation and multipath interference, making GPS- based localization ineffective. As a result, autonomous robots operating in such environments must rely entirely on onboard sensing, perception, and computation to estimate their position and navigate safely. Reliable indoor navigation systems are therefore essential for applications including surveillance, infrastructure inspection, search- and-rescue operations, and defense missions. Waypoint-based navigation is a widely adopted strategy for guiding robots through structured environments. In this paradigm, the robot sequentially travels through intermediate target locations, known as waypoints, until reaching a final goal position. Early work by Wang et al. [1] demonstrated that waypoint-based navigation can effectively combine reactive and deliberative planning strategies to guide robots through complex environments. Subsequent research has further explored waypoint planning techniques and navigation frameworks capable of operating in unknown or partially known environments. Recent surveys highlight the transition from classical rule-based navigation strategies toward learning-based approaches, which improve adaptability but introduce significant computational complexity [2]. Several studies have investigated automated waypoint generation to improve navigation reliability in constrained indoor environments. Moreno et al. [3] proposed an automatic waypoint generation approach that identifies critical navigation regions such as narrow corridors and generates intermediate targets to improve navigation success rates. While such methods improve robustness, accurate localization remains a fundamental requirement for successful waypoint execution. Localization in mobile robots is commonly achieved through probabilistic sensor fusion techniques that combine multiple sensing modalities. Extended Kalman Filters (EKF) are widely used to fuse odometry measurements with external observations such as visual landmarks, LiDAR features, or GPS measurements in order to improve pose estimation accuracy [4]. In indoor environments where GPS is unavailable, visual landmarks provide an effective mechanism for correcting accumulated odometric drift and maintaining long-term localization stability. Vision-based waypoint navigation has also been demonstrated in several robotic systems using camera-based landmark detection and orientation estimation [8]. Path planning in mobile robotics is typically structured as a hierarchical process consisting of global planning and local obstacle avoidance. Global planners compute collision-free paths using map representations such as occupancy grids. Classical graph search algorithms, including Dijkstra's algorithm, provide optimal path planning for environments with non-negative edge costs and remain widely used due to their computational efficiency [6]. For real-time obstacle avoidance, the Dynamic Window Approach (DWA) introduced by Fox et al. [7] is commonly applied, as it evaluates admissible velocity commands within the robot's dynamic constraints to safely navigate dynamic environments. Recent research has explored more advanced navigation techniques including reinforcement learning and end-to-end learning-based navigation systems. These approaches can improve adaptability and decision-making in complex environments but often require extensive training data and significantly higher computational resources, limiting their practicality for embedded robotic platforms [11], [12]. Consequently, classical planning and estimation techniques remain widely used for real-time autonomous navigation systems deployed on resource-constrained hardware. Despite significant progress in individual navigation components such as sensor fusion, path planning, and obstacle avoidance, fully integrated navigation frameworks that combine these techniques into reliable systems suitable for embedded platforms remain an active research area. This work addresses this need by presenting a unified waypoint-based navigation framework that integrates odometry-based motion estimation, visual landmark localization, Extended Kalman Filter sensor fusion, and hierarchical planning using Dijkstra's algorithm and the Dynamic Window Approach.

The proposed system is implemented within a ROS 2–based architecture and deployed on a physical mobile robot platform. The framework is validated through both simulation and real-world experiments, demonstrating reliable waypoint tracking, effective obstacle avoidance, and stable localization performance under practical indoor operating conditions.

II. SYSTEM ARCHITECTURE AND HARDWARE DESIGN

The proposed navigation system is implemented on a compact four- wheeled mobile robot designed for reliable operation in indoor environments. The platform follows a modular architecture in which sensing, localization, planning, and motion control are integrated through the Robot Operating System (ROS 2) middleware. This design enables flexible development while ensuring real-time interaction between perception and control modules.

A. *Mobile Robot Platform*

The robotic platform uses a four-wheel drive configuration mounted on an aluminium chassis to provide mechanical stability and improved traction on indoor surfaces. Each wheel is driven by an independent 12 V DC geared motor, allowing precise control of robot motion and enabling accurate trajectory tracking during waypoint navigation. The drivetrain is designed to prioritize torque consistency and smooth low- speed operation, which is essential for indoor robotic systems operating in confined environments. Integrated quadrature encoders attached to the drive motors provide high-resolution wheel rotation measurements. These encoder signals are used to estimate linear and angular motion through odometry calculations, allowing continuous estimation of robot displacement during navigation. The mechanical configuration of the drivetrain supports reliable motion control and enables accurate modelling of the robot's kinematic behaviour.

B. *Sensing and Perception Components*

Localization and perception are achieved through a combination of onboard sensing modalities. Visual perception is provided by an RGB camera mounted on the robot and interfaced directly with the onboard computing unit. The camera is used for visual landmark detection and recognition, enabling periodic pose correction during navigation in GPS-denied environments. Wheel encoders provide short-term motion estimation through differential odometry. Although odometry offers high-frequency pose updates, it is subject to cumulative drift caused by wheel slip, mechanical imperfections, and surface interactions. To reduce these effects, inertial measurements are incorporated to improve heading stability and motion estimation during dynamic manoeuvres. The complementary characteristics of visual observations, odometry measurements, and inertial sensing enable robust multi-modal localization through sensor fusion. This combination improves long-term pose accuracy while maintaining computational efficiency suitable for embedded robotic platforms.

C. *Embedded Computing and Control*

All perception and navigation algorithms are executed on an onboard single-board computer running Ubuntu 22.04 with ROS 2 Humble middleware. The embedded computing platform provides sufficient processing capability to perform image processing, localization, path planning, and control tasks in real time while maintaining low power consumption suitable for battery-powered robots. Motor actuation is handled using a TB6612FNG motor driver that supports pulse-width modulation (PWM) based speed control and bidirectional motion. The driver interface enables smooth velocity regulation and provides electrical protection features necessary for reliable long-term operation.

D. *Software Architecture*

The system software follows a modular ROS 2 architecture in which individual navigation functions are implemented as separate nodes communicating through standardized message interfaces. Sensor data from encoders, camera, and inertial sensors are processed by dedicated perception and motion estimation modules. These modules publish robot state information through the ROS transform framework, enabling consistent coordinate transformations between different system components. Global pose estimation is obtained through sensor fusion using an Extended Kalman Filter that combines odometry predictions with visual landmark observations. The resulting pose estimate is continuously updated and used by the navigation stack for waypoint tracking and path planning. High-level navigation follows a hierarchical planning strategy. Global path planning is performed using Dijkstra's algorithm over an occupancy grid representation of the environment to determine feasible waypoint sequences. Local motion decisions are refined using the Dynamic Window Approach, which evaluates admissible velocity commands within the robot's dynamic constraints to avoid obstacles while maintaining progress toward the goal. A waypoint management module supervises navigation progress by monitoring waypoint completion conditions and triggering replanning when necessary. The resulting velocity commands are translated into motor control signals through a low-level motion controller that ensures smooth and stable robot motion.

III. NAVIGATION ALGORITHMS AND CONTROL METHODOLOGY

The proposed navigation framework enables autonomous waypoint-based movement of a mobile robot in GPS-denied indoor environments by combining motion estimation, visual perception, probabilistic sensor fusion, and hierarchical path planning. Instead of relying on computationally demanding SLAM pipelines or infrastructure-dependent localization methods, the system adopts a lightweight hybrid localization strategy that integrates encoder-based odometry with visual landmark observations. This combination allows the robot to maintain reliable pose estimation while remaining computationally efficient enough to operate on embedded hardware platforms. The navigation architecture further incorporates classical planning techniques, using Dijkstra's algorithm for global path generation and the Dynamic Window Approach for real-time obstacle avoidance. By integrating these components within a unified ROS 2–based framework, the system provides a practical solution for indoor autonomous navigation on resource-constrained robotic platforms.

A. Odometry-Based Motion Estimation

Robot motion is continuously estimated using wheel encoder feedback from the drive motors. Encoder pulse counts represent wheel rotation and are converted into linear and angular velocity estimates. For a differential-drive wheeled robot, the relationship between wheel velocities and body motion can be expressed as

$$v = \frac{r(v_L + v_R)}{\mathbf{2}}$$

$$\Omega = \frac{r(v_R - v_L)}{2L}$$

where v denotes the forward velocity of the robot, ω represents the angular velocity, r is the wheel radius, v_L and $v_R$correspond to the left and right wheel velocities, and L is the distance between the wheels.

Encoder measurements are processed at regular time intervals to estimate robot displacement. The robot pose in the global coordinate frame is obtained by integrating the velocity estimates over time:

$$x(k+1) = x(k) + v(k)\cos(\theta(k))\,\Delta t$$

$$y(k+1) = y(k) + v(k)\sin(\theta(k))\,\Delta t$$

$$\theta(k+1) = \theta(k) + \omega(k)\,\Delta t$$

Although odometry provides high-frequency motion updates and enables real-time pose estimation, it is subject to cumulative error. Wheel slip, surface irregularities, encoder quantization, and mechanical tolerances can introduce drift that grows over time. Consequently, additional sensing modalities are required to periodically correct the robot pose estimate.

B. Visual Landmark Detection and Recognition

To compensate for accumulated odometric drift, the navigation system incorporates visual landmark recognition using an onboard RGB camera. Visual observations allow the robot to detect known environmental features and use them as reference points for pose correction. Image frames captured by the camera are processed to extract distinctive features using oriented FAST corner detection combined with BRIEF descriptors. The extracted features are compared with previously stored landmark templates through Hamming-distance based matching. To improve robustness and eliminate incorrect matches, geometric consistency is verified using a RANSAC-based filtering procedure. When a landmark is successfully recognized, the relative pose between the robot and the observed landmark is estimated using perspective-n-point (PnP) techniques. These observations provide periodic global corrections that reduce accumulated localization error and improve long-term navigation stability.

C. Extended Kalman Filter for Sensor Fusion

Reliable pose estimation is achieved through sensor fusion using an Extended Kalman Filter (EKF). The state vector represents the robot position and orientation in the global reference frame and is defined as

$$x_k = [x_k,; y_k,; \theta_k]^T$$

The prediction step of the EKF uses the odometry-based motion model
to estimate the robot's next state:

$$x_{k+1|k} = f(x_k, u_k) + w_k$$

where u_k represents the control input derived from wheel velocities and w_k represents process noise.

Visual landmark observations are incorporated during the update stage according to

$$z_k = h(x_k) + v_k$$

Where $\mathrm{h}(x_k)$ represents the predicted landmark observation from the estimated pose and v_k denotes measurement noise.

The Kalman gain used to update the state estimate is computed as

$$K = P_{(k|k-1)} H^T \left(H P_{(k|k-1)} H^T + R\right)^{(-1)}$$

The corrected state estimate is then obtained as

$$x_{\{k|k\}} = x_{\{k|k-1\}} + K\,(z_k - h\,(x_{\{k|k-1\}}))$$

Through this process, the filter balances high frequency odometry predictions with intermittent but reliable visual landmark observations. As a result, the robot maintains stable and accurate pose estimation even during extended navigation tasks.

D. Global Path Planning

Global path planning is performed using Dijkstra's algorithm applied to an occupancy grid representation of the environment. The workspace is discretized into grid cells representing either free space or obstacles. These cells are treated as nodes in graph, and edges connect neighboring cells with non negative traversal costs. Dijkstra's algorithm computes the minimum cost path from the robot's current location to a specified waypoint by iteratively expanding nodes with the lowest cumulative cost. The resulting path guarantees optimality under non negative edge weights and produces a sequence of intermediate waypoints guiding the robot toward its destination. When newly detected obstacles invalidate the current path, the planner automatically recomputes a feasible route.

E. Local Obstacle Avoidance

While the global planner provides a high-level path, real-time collision avoidance is handled by the Dynamic Window Approach (DWA). This method evaluates feasible velocity commands based on the robot's dynamic constraints and local obstacle information.

For each candidate velocity pair (v,ω), the algorithm simulates short-term trajectories and evaluates them using a scoring function defined as

$$\mathrm{G}(\mathrm{v},\omega) = \alpha\,\mathrm{D}(\mathrm{v},\omega) + \beta\,\mathrm{O}(\mathrm{v},\omega) + \gamma\,\mathrm{V}(\mathrm{v},\omega)$$

where $D(v,\omega)$represents progress toward the goal, $O(v,\omega)$measures obstacle clearance, and $V(v,\omega)$ represents the velocity magnitude. The weighting parameters α , β, and γ determine the relative importance of these factors.

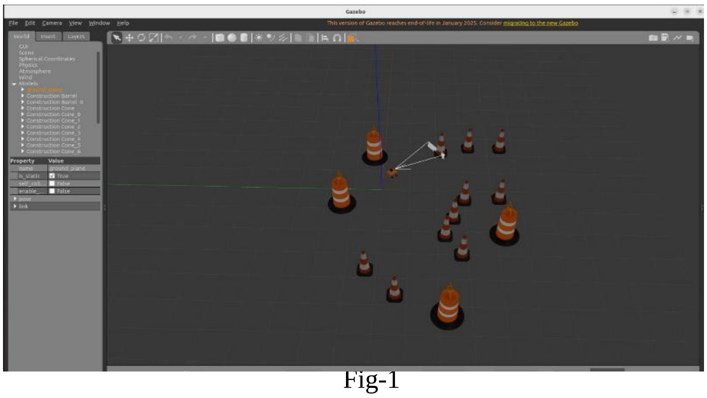
The velocity command that maximizes the evaluation function is selected and executed by the motion controller. This process is repeated continuously, enabling the robot to avoid obstacles while maintaining smooth progress toward the target waypoint.

IV. EXPERIMENTAL VALIDATION IN SIMULATION AND REAL WORLD ENVIRONMENTS

To evaluate the effectiveness of the proposed navigation framework before hardware deployment, simulation-based validation was conducted using two robotics simulation environments: Gazebo Classic and CoppeliaSim. These platforms provide complementary testing capabilities. Gazebo enables physics-based simulation with realistic modeling of robot dynamics, while CoppeliaSim allows flexible environment design and rapid testing of navigation algorithms. Using both simulators provides cross- verification of system behavior under different simulation assumptions.

A. Simulation Environment and Setup

The simulated robot platform was modeled with the same kinematic configuration as the intended hardware system. The robot operates in structured indoor environments containing obstacle-filled navigation areas. The navigation stack, implemented in ROS 2, integrates localization, planning, and motion control modules to enable autonomous waypoint tracking. In the simulation setup, the robot receives motion commands generated by the navigation framework while continuously estimating its position using the hybrid localization strategy.



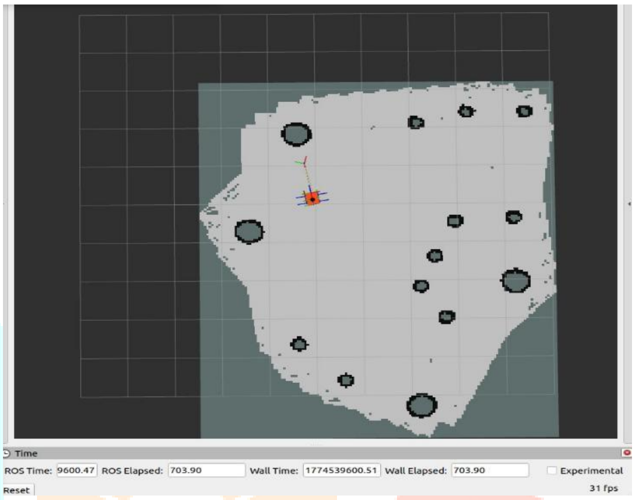
This version of Gazebo reaches end-of-life in January 2025. Consider migrating to the new Gazebo

Property	Value
name	ground_plane
is_static	☑ True
self_coll...	☐ False
enable_...	☐ False

Fig-1

B. Localization and Drift Correction Performance

One of the primary objectives of the navigation framework is to reduce localization drift during extended missions. Encoder-based odometry alone tends to accumulate error as the robot travels, particularly when wheel slip or mechanical inconsistencies are present. In the simulation experiments, this effect was evaluated by comparing pose estimation performance with and without visual landmark corrections. Results show that pure odometry leads to drift that gradually increases with traveled distance, reaching approximately nine percent of the total path length during long navigation sequences. When visual landmark observations are incorporated through the Extended Kalman Filter, accumulated drift is significantly reduced. Periodic landmark-based corrections decrease total localization error to below one percent of the traveled distance. The experiments further indicate that the hybrid localization approach achieves drift reduction between approximately sixty-five and eighty percent compared to odometry-only estimation. Visual landmark detection achieved an average success rate of around eighty-five percent within an observation range of approximately two meters.



ROS Time: 9600.47 ROS Elapsed: 703.90 Wall Time: 1774539600.51 Wall Elapsed: 703.90

Fig-2

C. *Waypoint Navigation Performance*

The effectiveness of the navigation framework was also evaluated by measuring waypoint tracking accuracy during simulated patrol missions. The robot was tasked with navigating through multiple waypoints distributed across a complex indoor environment containing static obstacles. The results demonstrate stable waypoint tracking performance, with positional errors typically ranging between 0.18 m and 0.32 m from the target waypoint location. The hierarchical navigation strategy successfully integrates global path planning and local obstacle avoidance, enabling the robot to navigate around obstacles while maintaining progress toward the desired waypoint sequence.

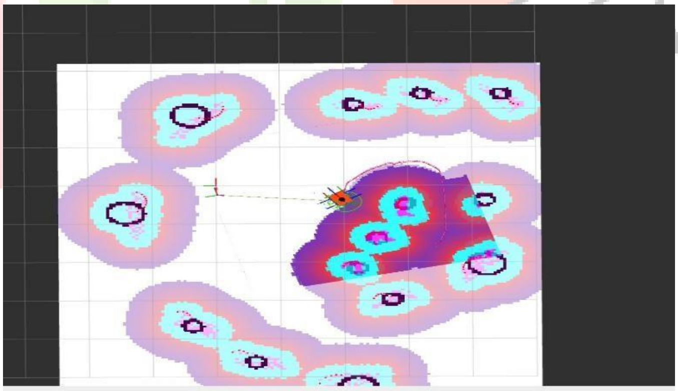


Fig-3

D. Computational Performance

The computational performance of the navigation system was analyzed to ensure suitability for embedded robotic platforms. All core modules— including localization, planning, and control—were executed on a simulated computing platform equivalent to a Raspberry Pi 4 running ROS2. During extended simulation experiments, total CPU utilization remained within the range of approximately 30–35 percent. No timing violations were observed for localization updates, path planning operations, or motion control loops. These results confirm that the proposed navigation architecture maintains sufficient computational efficiency to operate in real time on resource-constrained hardware platforms.

E. Comparison with Existing Navigation Approaches

The achieved waypoint tracking accuracy of approximately 0.18–0.32 m is comparable to the performance reported by many EKF-based localization systems in the literature, where navigation errors typically fall within the range of 0.15–0.50 m. While SLAM-based methods may provide higher localization precision, they generally require significantly greater computational resources. In contrast, the proposed framework maintains competitive navigation accuracy while operating with substantially lower computational load. Compared with pure odometry-based navigation methods, which often exhibit drift rates approaching ten percent of travelled distance, the hybrid localization strategy demonstrates a significant improvement in long-term pose stability.

F. Real World Experimental Validation

To evaluate the performance of the proposed navigation framework under practical conditions, real-world experiments were conducted using the developed mobile robot platform in indoor environments including corridors and laboratory spaces. The robot was deployed with onboard sensing and computation, operating entirely in a GPS-denied setting. During experimentation, the system demonstrated stable multi-waypoint navigation while successfully avoiding static obstacles. The Extended Kalman Filter–based sensor fusion effectively reduced odometric drift by incorporating visual landmark observations, resulting in improved localization accuracy over extended trajectories. However, real-world conditions introduced several practical challenges not present in simulation. Variations in lighting conditions affected visual landmark detection reliability, while wheel slippage and surface irregularities introduced minor inconsistencies in odometry estimates. Despite these factors, the system maintained consistent navigation performance, with waypoint tracking errors remaining within an acceptable range comparable to simulation results. The experimental results validate that the proposed framework is capable of real-time operation on embedded hardware and can be effectively deployed for autonomous indoor navigation tasks under realistic environmental constraints.

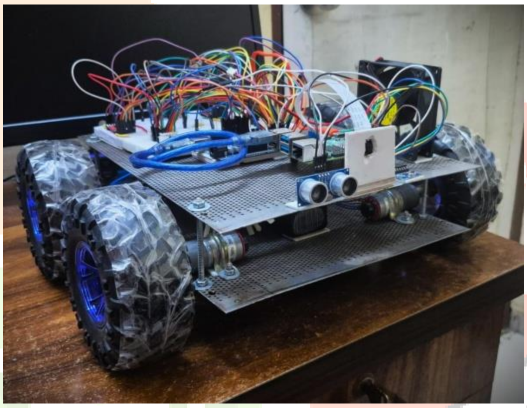


Fig-4: Hardware implementation of the autonomous mobile robot platform showing onboard computing, sensing modules, and drive system used for real-world waypoint navigation

V. *APPLICATIONS AND DEPLOYMENT CONTEXTS*

The proposed navigation framework is designed to support autonomous operation of mobile robots in environments where GPS signals are unavailable or unreliable. Such conditions commonly arise in indoor facilities, underground structures, and complex built environments where traditional satellite-based localization cannot be used. The ability to navigate using onboard sensing and computation makes the system suitable for a variety of operational scenarios requiring reliable indoor mobility. In defence and security contexts, autonomous mobile robots can be deployed for routine surveillance and patrol operations within secured installations. Waypoint-based navigation enables robots to follow predefined inspection routes through corridors, rooms, and restricted areas while maintaining consistent coverage of the environment. The system can also assist in reconnaissance missions where robots are required to explore unknown or hazardous locations before human personnel enter the area.

Emergency response and disaster management operations represent another important application domain. Following natural disasters or industrial accidents, indoor environments may become unsafe due to structural damage, toxic substances, or fire hazards. Autonomous robots equipped with reliable navigation capabilities can assist in search-and-rescue missions by exploring damaged buildings, collecting environmental data, and providing situational awareness without exposing rescue personnel to unnecessary risk. The proposed framework is also relevant for inspection and monitoring tasks in critical infrastructure facilities such as power plants, transportation hubs, and industrial plants. Autonomous robots can execute periodic inspection routes, detect anomalies, and collect operational data while navigating through indoor spaces that may not be easily accessible to humans. Reliable waypoint-based navigation ensures repeatable operation and allows robots to perform regular monitoring tasks with minimal supervision. Beyond operational deployment, the system also serves as a research and development platform for studying autonomous navigation algorithms. The modular architecture built using ROS 2 allows new perception, localization, or planning algorithms to be integrated and evaluated without major modifications to the overall system. The use of simulation environments for validation further enables safe experimentation and performance analysis before transitioning to real-world robotic platforms.

VI. REAL WORLD DEPLOYMENT AND SYSTEM VALIDATION

The proposed navigation framework was successfully deployed on a physical mobile robot platform to evaluate its performance under real-world operating conditions. The deployment process involved system integration, sensor calibration, controller tuning, and experimental validation to ensure reliable autonomous navigation. Accurate calibration of sensors was performed prior to experimentation. Wheel encoders were calibrated to establish the relationship between encoder counts and linear displacement, minimizing systematic odometry errors. Camera calibration was carried out to determine intrinsic and extrinsic parameters, ensuring accurate visual landmark detection and pose estimation. Additionally, inertial measurements were incorporated to improve heading stability during motion. Controller parameters initially developed in simulation were experimentally tuned to account for real-world effects such as friction, motor nonlinearities, actuator delays, and surface irregularities. Velocity control was adjusted to achieve smooth motion, stable trajectory tracking, and reliable response under varying operating conditions. Extended Kalman Filter parameters were also refined to balance prediction uncertainty from odometry with measurement uncertainty from visual observations. A structured validation process was followed to verify system performance. Initial tests included teleoperated motion to confirm drivetrain functionality and sensor reliability. This was followed by controlled single-waypoint navigation experiments to validate localization and control performance. Subsequently, multi-waypoint autonomous navigation tasks were executed, incorporating real-time obstacle avoidance using the Dynamic Window Approach. Experimental results demonstrate that the system is capable of stable and reliable autonomous navigation in indoor environments. The robot successfully completed waypoint tracking tasks while maintaining localization accuracy and avoiding obstacles under practical constraints. The deployment confirms that the proposed framework can be effectively implemented on embedded robotic platforms and operate in real-time conditions outside simulation environments.

VII. *CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS*

This work presented an integrated framework for autonomous waypoint-based navigation of mobile robots operating in GPS-denied indoor environments. The implemented system combines encoder-based odometry with visual landmark observations through Extended Kalman Filter–based sensor fusion to achieve reliable pose estimation during navigation. A hierarchical planning structure integrating Dijkstra's algorithm for global path planning and the Dynamic Window Approach for local obstacle avoidance enables efficient and safe navigation in structured indoor spaces. The complete system is implemented within a ROS 2–based software framework and deployed on an embedded robotic platform. The proposed framework was validated through both simulation and real-world experimental deployment. Hardware experiments conducted in indoor environments demonstrate that the robot successfully performs multi-waypoint navigation while avoiding obstacles under practical conditions such as sensor noise, lighting variations, and mechanical uncertainties. The system achieves waypoint tracking errors in the range of approximately 0.20 m to 0.35 m, while the hybrid localization approach significantly reduces accumulated odometric drift compared to odometry-only estimation. Real-time performance is maintained within the computational constraints of a Raspberry Pi–class platform. Despite successful deployment, certain limitations were observed during real-world operation. Localization accuracy is influenced by sensor noise, wheel slippage, and calibration imperfections. The reliance on visual landmarks introduces sensitivity to environmental conditions such as illumination changes and partial occlusions. Additionally, the limited computational resources of embedded hardware constrain the scalability of perception and planning algorithms in more complex environments.

Future research will focus on improving system robustness and autonomy in dynamic and unstructured environments. Integration of additional sensing modalities such as LiDAR and inertial measurement units (IMU) can enhance localization reliability. The incorporation of advanced SLAM techniques may enable operation in previously unknown environments without reliance on predefined landmarks. Further optimization of computational efficiency and exploration of learning-based perception methods will support more scalable and adaptive autonomous navigation systems.

REFERENCES

[1] Wang, Y., Mulvaney, D., Sillitoe, I., Swere, E. (2008). Robot navigation by waypoints. *Journal of Intelligent Robotic Systems*, 52, 175–207. https://doi.org/10.1007/s10846-008-9209-6

[2] Xiao, X., Liu, B., Warnell, G., Stone, P. (2021). Motion planning and control for mobile robot navigation using machine learning: A survey. *arXiv preprint arXiv:2011.13112*.

[3] Moreno, F. A., et al. (2019). Automatic waypoint generation to improve robot navigation through narrow spaces. *Sensors*, 20(1), 240. https://doi.org/10.3390/s20010240

[4] Welch, G., Bishop, G. (2006). *An introduction to the Kalman filter*. Technical Report, University of North Carolina at Chapel Hill.

[5] Sharma, S., Taylor, M. E. (2012). Autonomous waypoint generation strategy for on-line navigation. *IROS Workshop*.

[6] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269–271.

[7] Fox, D., Burgard, W., Thrun, S. (1997). The dynamic window approach to obstacle avoidance. *IEEE Robotics & Automation Magazine*, 4(1), 23– 33.

[8] Muslimin, S., Prihatini, E., Husni, N. L., Pebrian, S. (2025). Development of a waypoint navigation system for RVM robots using path planning algorithms. *Proceedings of the 8th FIRST 2024 International Conference*, Atlantis Press.

[9] Hayet, J. B., Piater, J., Vetter, T., Fua, P. (2007). A visual landmark framework for mobile robot navigation. *International Journal of Computer Vision*, 71(2), 189–203.

[10] Ghafourian, A., et al. (2024). Hierarchical end-to-end autonomous navigation through few-shot waypoint detection. *IEEE Robotics and Automation Letters*, 9(4), 3211–3218.

[11] Raj, A., et al. (2024). Deep Q-Network approach for intelligent mobile robot navigation. *Nature Scientific Reports*, 41598–024– 72857–3.

[12] Prusakiewicz, L., Tnnes, S. (2020). *Comparison of autonomous waypoint navigation methods for an indoor blimp robot*. M.Sc. Thesis, KTH Royal Institute of Technology.

[13] Ghorab, M., Lorenzen, M. (2025). Multi-waypoint path planning and motion control for non-holonomic mobile robots in agricultural applications. *arXiv preprint arXiv:2507.23350v1*.

[14] Ichihara, K., et al. (2022). Waypoint-based human-tracking navigation for museum guide robot. *Journal of Robotics and Mechatronics*, 34(6), 1412–1421.

[15] Thamrin, N. M., et al. (2024). Assessing NEO-M8N GPS-based waypoint navigation performance. *Journal of Engineering Science and Technology*, 19(6), 19605–19620.

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

[17] Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. *Proceedings of IEEE International Conference on Robotics and Automation*, 500–505.

[18] Liu, L., Chen, X., Zhu, S., Tan, P. (2023). Path planning techniques for mobile robots: Review and challenges in autonomous navigation. *Robotics and Computer-Integrated Manufacturing*, 79, 102–126.

[19] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y. (2009). ROS: An open-source robot operating system. *Proceedings of the ICRA Workshop on Open Source Software*, 3(3.2), 5.

[20] Fu, X., et al. (2025). Research on path planning of mobile robots based on improved A* algorithm. *PMC Articles*.

[21] Tariq, M. T., Wang, C., Hussain, Y. (2025). Robust mobile robot path planning via LLM-based dynamic waypoint generation. *arXiv preprint arXiv:2501.15901*.

[22] Thamrin, N. M., et al. (2024). NEO-M8N GPS receiver specifications and integration. *Technical Documentation*, u-blox.

[23] Raspberry Pi Foundation. (2024). *Raspberry Pi 5 Technical Specifications*. Retrieved from https://www.raspberrypi.com

[24] Arduino. (2024). *Arduino Uno Datasheet and Reference*.
Retrieved from https://www.arduino.cc

[25] Open Robotics. (2024). *Gazebo Classic Simulator Documentation*. Retrieved from https://gazebosim.org Coppelia Robotics. (2024). *CoppeliaSim Documentation*. Retrieved from https://www.coppeliarobotics.com

[26] ROS 2 Community. (2024). *ROS 2 Navigation Framework*.
Retrieved from https://docs.ros.org