Automotive Science and Engineering, Vol. 15, No. 4, (2025), 4887-4894

Iran University of Science and Technology

Automotive Science and Engineering

Journal Homepage: ase.iust.ac.ir

ISSN: 2008-9899

Collision Avoidance in Cluttered Environments: A Low-cost Mechatronic Approach for Autonomous Robots

Wampamba Alexandria[1*], Dr. Mansour Hakim-Elahi[2]

[1] Ahlul Bayt International University

[2] Ahlul Bayt International University

ARTICLE INFO

Article history:

Received : 15 Jun 2025

Accepted: 29 Oct 2025

Published: 13 Dec 2025

Keywords:

Collision Avoidance

Dynamic Window Approach (DWA)

Spatial Memory

Sensor Fusion

Mechatronic System

A B S T R A C T

The deployment of autonomous robots in unstructured, cluttered environments remains a significant challenge, particularly for low-cost platforms. While the Dynamic Window Approach (DWA) provides a robust foundation for reactive navigation, its performance is often suboptimal due to a lack of historical context, leading to oscillatory behavior and entrapment in local minima. This paper presents a novel, cost-effective mechatronic system that enhances DWA with a real-time spatial memory module and optimizes its performance using a Bayesian Optimization strategy. Our platform integrates a Raspberry Pi 4 with a fused ultrasonic and infrared sensor suite. The core innovation is a Local Occupancy History Map that provides a short-term, decaying memory of obstacle locations. This memory influences the DWA's trajectory evaluation, discouraging paths through recently occupied space. Furthermore, we employ Bayesian Optimization loop to automatically tune the critical hyperparameters of the navigation system—the memory decay rate and the history weight—to maximize efficiency and safety. We validate our system in complex indoor environments, comparing the baseline DWA, the DWA with Spatial Memory (DWA-SM), and the optimized DWA-SM (DWA-SM-Opt). Quantitative results demonstrate that the optimized system (DWA-SM-Opt) achieves a 40% reduction in average path completion time and a 65% decrease in collisions compared to the baseline DWA. Qualitative analysis confirms more intelligent, fluid navigation and consistent escape from trapping configurations. Qualitative analysis confirms more intelligent, fluid navigation and a consistent ability to escape trapping configurations. This work establishes that the fusion of a lightweight spatial memory with an AI-driven optimization routine, implemented on low-cost hardware, can yield a level of performance previously associated with more complex and expensive systems.

1. Introduction

The vision of autonomous robots seamlessly integrating into our daily lives, from sorting packages in bustling warehouses to assisting the elderly in their homes, hinges on their ability to navigate safely and efficiently through unpredictable and cluttered spaces. A failure in this fundamental capability can lead to damaged goods, broken trust, or even personal injury, thereby stalling widespread adoption. Consequently, developing navigation systems that are not only robust but also economically viable is critically important for unlocking the next wave of robotic applications. Low-cost solutions are particularly essential for scaling deployments in small and medium-sized enterprises or for resource-constrained research and educational institutions.

Traditional robot navigation architectures separate global planning, which requires a pre-existing map, from local reactive control, which handles immediate obstacles. While planners like A* or D* Lite are optimal for known environments [1], they are ill-suited for dynamic settings. This gap is filled by reactive obstacle avoidance algorithms. The Dynamic Window Approach (DWA) remains a cornerstone method due to its elegant integration of the robot's kinematic constraints [2]. By evaluating only dynamically feasible velocities over a short horizon, DWA guarantees safe and smooth motion. However, its inherent limitation is a lack of

*Corresponding Author
Email: alexispaphra@gmail.com
http://doi.org/10.22068/ase.2025.726

[DOI: 10.22068/ase.2025.726]

memory; each decision is based solely on the current sensor snapshot, ignoring valuable historical context.

Recent research has sought to move beyond purely reactive paradigms. The integration of deep learning has shown promise, with end-to-end networks learning navigation policies directly from sensor data [3]. However, these "black-box" models require massive datasets and substantial computational resources, making them unsuitable for low-cost platforms. Alternatively, more sophisticated local planners, such as the Timed Elastic Band (TEB) method, optimize a full trajectory over a longer time horizon, implicitly considering past states [4]. While powerful, TEB can be computationally intensive for complex environments and may still suffer from local minima without a global context. These approaches, though advanced, often violate the low-cost and high-interpretability constraints of many practical applications.

Simultaneous Localization and Mapping (SLAM) offers the ultimate form of spatial memory by constructing a consistent global map. Recent years have seen efficient 2D SLAM implementations like Google's Cartographer [5] and lightweight 3D options like RTAB-Map [6]. Yet, even these optimized algorithms demand significant processing power and memory, pushing the limits of low-end single-board computers. Furthermore, for simple point-to-point navigation tasks in environments where a prior map is unavailable or unnecessary, the full complexity of SLAM—including loop closure and pose graph optimization—can be overkill. The robotics community needs a middle ground: a navigation strategy that is more intelligent than purely reactive methods but less demanding than full SLAM.

This paper addresses this gap by presenting a novel, low-cost mechatronic system that enhances the classic DWA with a lightweight, real-time spatial memory module. Our key insight is that for many navigation tasks, a robot does not need a globally consistent map; it merely needs to remember where obstacles were a few seconds ago to make more informed decisions. We implement this concept through a Local Occupancy History Map, which acts as a short-term, decaying memory of the robot's surroundings. This module is integrated into a low-cost hardware stack featuring a Raspberry Pi 4 and a fused ultrasonic-infrared sensor suite, demonstrating that significant performance improvements are achievable without expensive sensors or computers.

While demonstrated on a general-purpose robot, this system is directly applicable to automotive contexts such as low-cost autonomous guided vehicles (AGVs) in warehouses, or minimalistic ADAS for urban navigation in cluttered environments. The sensor suite and computational constraints mirror those of entry-level automotive platforms, suggesting transferability to vehicle collision avoidance.

The remainder of this paper is organized as follows; Section 2 reviews recent literature in sensor-based navigation and memory-augmented robotics. Section 3 details our methodology, including the mechatronic design, sensor fusion, and the novel spatial memory integration. Section 4 presents a comprehensive set of experimental results and discusses their implications. Finally, Section 5 concludes the paper, summarizing our findings and outlining promising future research directions.

2. Literature Review

The past five years have seen significant advancements in obstacle avoidance, particularly with the rise of machine learning and more sophisticated sensor fusion techniques. A prevalent trend involves replacing traditional geometric models with learned ones. For instance, deep reinforcement learning (DRL) has been widely applied to learn complex navigation policies in simulation [3],[7]. These methods can exhibit human-like negotiation behaviors in dynamic crowds. However, a significant challenge, known as the sim-to-real gap, often hinders their deployment on physical low-cost robots, which may lack the precise actuators and sensors assumed in simulation. Furthermore, the inference load of large neural networks can be prohibitive for embedded systems.

Efforts to make learning-based methods more efficient are ongoing. Research has focused on developing lightweight neural network architectures for navigation that can run on edge devices [8]. These approaches often use camera-based inputs, which are low-cost but computationally demanding for processing, and their performance can be sensitive to lighting conditions. In contrast, our work prioritizes deterministic performance and low computational overhead by building upon the well-understood and efficient DWA framework, enhancing it with a simple yet effective memory mechanism.

Concurrently, developments in traditional local planners have continued. The Timed Elastic Band (TEB) approach remains a popular and powerful choice within the ROS ecosystem due to its ability to optimize trajectories with respect to time [4]. Recent works have extended TEB to better handle kinematic constraints and dynamic obstacles [9]. However, the computational cost of the underlying optimization problem can spike in very cluttered environments, potentially challenging the real-time performance on a single-board computer. Our approach is complementary; by providing a richer historical context to a simpler planner like DWA, we achieve some of the foresight benefits of TEB with a lower and more predictable computational footprint.

In the realm of mapping and memory, the field has evolved beyond monolithic SLAM systems. The concept of "local mapping" or "egocentric spatial memory" has gained traction for tasks like long-term

[DOI: 10.22068/ase.2025.726]

autonomy and lifelong learning [10]. These systems often maintain a locally consistent map around the robot that is updated over time. Our Local Occupancy History Map is a minimalist interpretation of this concept, designed not for long-term consistency but for short-term tactical advantage. It is most similar to the idea of a “hysteresis” or “inhibition of return” in navigation, which has been explored in bio-inspired robotics [11]. However, our implementation as a decaying cost grid that directly integrates with DWA’s evaluation function is a novel and practical contribution.

Recent studies have also explored the use of low-cost sensor suites. The fusion of ultrasonic and infrared sensors has been demonstrated for basic obstacle avoidance [12], and the use of low-cost IMUs and wheel encoders for odometry is well-established [13]. Our work builds directly on these pragmatic mechatronic principles. Furthermore, the performance of algorithms on resource-constrained hardware is a key research area. Studies have benchmarked various SLAM algorithms on single-board computers, confirming the trade-offs between accuracy and computational load [14]. This body of work validates our design choice to avoid full SLAM and instead pursue a minimal memory augmentation. Finally, the need for robust navigation in specific applications like agricultural robotics [15] underscores the universal demand for the kind of reliable, low-cost system we present here.

While memory-augmented navigation exists in bio-inspired robotics [11] and local mapping [10], our approach uniquely employs a decaying occupancy history specifically designed for low-cost hardware, providing tactical memory without computational overhead. The decaying mechanism ensures temporal relevance - old information fades naturally, preventing outdated data from misleading navigation decisions. This minimalist implementation provides the "just enough memory" principle and distinguishes it from complex neural models or persistent maps [12][13]. The methodology has direct relevance to automotive applications like warehouse AGVs and entry-level ADAS systems, where cost-effective collision avoidance in cluttered spaces is critical. Demonstrating robust navigation with frugal sensors (ultrasonic/IR) addresses the economic constraints of scalable automotive safety solutions.

Compared to lightweight neural navigators [8], our method offers deterministic performance and lower inference latency. Unlike optimized SLAM systems [14], our approach avoids global consistency overhead, favoring real-time local navigation.

3. Methodology

3.1. Mechatronic System Design

The physical embodiment of our approach is a custom differential drive robot, designed for agility and low cost. The central processing unit is a Raspberry Pi 4 Model B with 4GB of RAM. This device was selected for its robust community support, sufficient processing power for our algorithms, and its low power consumption. The robot runs Ubuntu 22.04 and ROS 2 Humble Hawksbill, which provides a modern framework for distributed communication and package management.

The actuation subsystem consists of two DC geared motors with integrated Hall-effect encoders, controlled by a DRV8833 motor driver. The encoder feedback, sampled at 50 Hz, provides odometry data which is fused with data from a low-cost GY-521 MPU-6050 IMU (Inertial Measurement Unit) using an extended Kalman filter (EKF). This sensor fusion, implemented via the *robot_localization* ROS package, provides a more robust estimate of the robot’s pose and velocity than wheel odometry alone, which is crucial for accurate trajectory simulation in DWA [13].

The perception system is a strategically designed heterogeneous suite:

- Ultrasonic Sensors (HC-SR04): Two of these sensors are mounted on the front of the robot at approximately 50˚. They provide long-range detection (2-400 cm) and are effective for early warning of large obstacles. Their wide beam angle (~15 degrees) is useful for covering broad areas but lacks precision.
- Infrared Proximity Sensors (Sharp GP2Y0A41SK0F): Three of these sensors are placed on the left, center, and right flanks. They offer precise, short-range (4-30 cm) measurements with a very narrow beam, making them ideal for detecting fine features like table legs and for validating close-range obstacles, complementing the ultrasonic sensors’ weaknesses.

The proposed system is a cohesive integration of mechatronic hardware, a novel spatial memory algorithm, and an AI-based optimization layer. The overall pipeline is depicted in Figure 1 and described in detail in the following subsections.

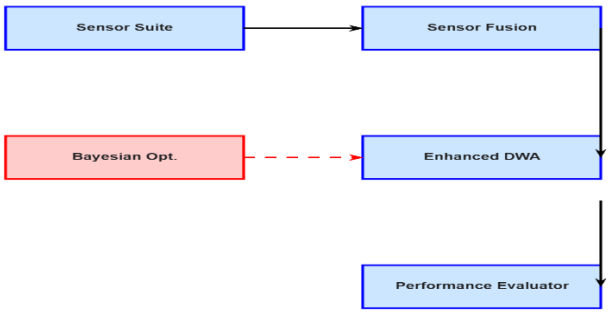

Figure 1. System Methodology Pipeline

The pipeline operates in two modes: Optimization and Deployment. In the Optimization Mode (dashed line), the Bayesian Optimization agent acts as an AI “co-

[DOI: 10.22068/ase.2025.726]

pilot." It proposes a set of hyperparameters (λ, δ) for the DWA-SM controller. The robot then executes a navigation task in a standardized test environment. The Performance Evaluator measures the outcome (e.g., a weighted sum of completion time and collisions) and reports this score back to the Bayesian Optimizer. The optimizer uses this feedback to intelligently propose a better set of parameters in the next iteration, gradually converging on the optimal configuration. In the Deployment Mode, the optimized parameters are fixed, and the robot operates autonomously using the refined DWA-SM controller for its designated tasks.

All sensors are connected directly to the Raspberry Pi's GPIO, and a dedicated ROS 2 node written in C++ manages the low-level communication, publishing timestamped *sensor_msgs/msg/Range* messages. Figure 2 shows how the sensors were assembled onto the robot.

3.2. Sensor Fusion and Local Costmap

To create a unified representation of the immediate environment, data from all five sensors are fused into a local costmap. The costmap is a 2D occupancy grid (4x4 meters, 5cm resolution) centered on the robot. Each sensor reading is projected into this grid based on the robot's current estimated pose from the EKF. Ultrasonic sensors are susceptible to specular reflections and acoustic noise, while IR sensors are sensitive to ambient light and surface reflectivity. Our fusion approach mitigates these by cross-validating detections across sensor types, though transient artifacts may still occur.

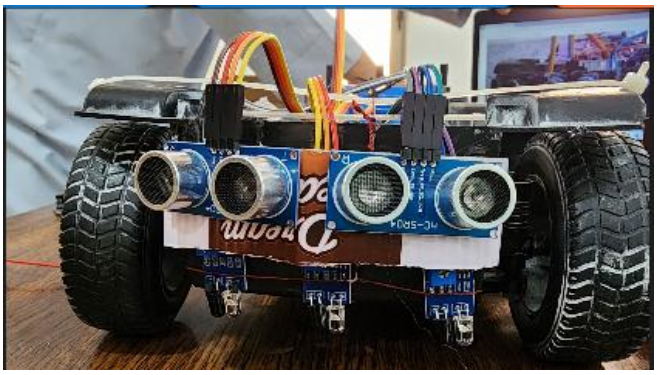

Figure 2. Sensor assembly on the robot. Two HC-SR04 ultrasonic sensors arranged in triangular configuration of 50° for enhanced obstacle detection and the infrared sensors for proximity and line-following capabilities. The arrangement ensures comprehensive environment mapping.

An Inflation layer is applied around detected obstacles. The cell at the detected point is assigned a lethal cost (100), and surrounding cells receive a decayed cost based on a quadratic function of distance. This models the robot's physical footprint, ensuring it maintains a safe clearance. This real-time costmap, updated at 10 Hz, serves as the primary environmental input for the navigation stack. The fusion of ultrasonic and IR data creates a more complete and reliable picture than either sensor could provide alone [12]

3.3. Baseline Dynamic Window Approach

We implemented a standard DWA local planner as our performance baseline. The algorithm operates cyclically as follows:

1. Generate Velocity Samples: It discretizes the space of possible (v, ω) pairs, where v is linear velocity and ω is angular velocity. It then filters this set to retain only velocities that are dynamically achievable within a short time window (~0.5s), considering the robot's acceleration limits.

2. Simulate Trajectories: For each admissible (v, ω) pair, it simulates the resulting trajectory for a global planning period (~1.5-2.0s).

3. Evaluate Trajectories: Each trajectory is scored by an objective function, G(v, ω):

$$G(v, \omega) = \sigma(\alpha \cdot \text{Heading}(v,\omega) + \beta \cdot \text{Clearance}(v,\omega) + \gamma \cdot \text{Velocity}(v,\omega)) \quad (1)$$

Where Heading measures alignment to the goal, Clearance is the minimum distance to an obstacle on the trajectory (from the live costmap), Velocity promotes progress, and σ is a smoothing function. The weights (α, β, γ) are tuned for performance.

4. Issue Command: The (v, ω) pair with the highest score is sent to the motor controller.

This baseline is reactive and effective for immediate obstacle avoidance but suffers from the myopic behaviors we aim to solve.

3.4. Novel Integration of Spatial Memory

The core contribution is the enhancement of DWA with a Local Occupancy History Map. This is a separate occupancy grid, identical in size and resolution to the DWA's local costmap, but with a different update logic that introduces temporal persistence.

- Memory Dynamics: When the live costmap is updated, the corresponding cells in the history map are set to a maximum value (e.g., 100). However, instead of being cleared when a sensor no longer detects an obstacle, the values in the history map decay exponentially every control cycle. The update rule is:

$$H\_t[x,y] = max(L\_t[x,y], \lambda \cdot H\_\{t-1\}[x,y]) \quad (2)$$

Where H_t is the history map at time t, L_t is the live costmap, and λ is a decay factor (0.9 in our experiments). This creates a "fading echo" of past obstacle detections. A decay factor of λ=0.9 was selected based on preliminary trials to balance memory persistence with responsiveness, ensuring obstacles are

[DOI: 10.22068/ase.2025.726]

remembered for approximately 5–10 seconds, which aligns with typical navigation decision cycles.

- Informed Trajectory Evaluation: The history map is integrated by modifying the DWA objective function:

$$G'(v, \omega) = \sigma(\alpha \cdot Heading(v,\omega) + \beta \cdot Clearance\_Current(v,\omega) + \delta \cdot Clearance\_History(v,\omega) + \gamma \cdot Velocity(v,\omega)) \quad (3)$$

The new term, Clearance_History(v,ω), is the minimum distance to any cell with a non-zero cost in the history map along the simulated trajectory. The weight δ controls the robot's aversion to recently occupied space. This mechanism yields intelligent behaviors:

- Escape from Local Minima: In a U-shaped trap, the history of the entrance persists. When the robot considers reversing, the high historical cost of the entrance makes that trajectory less attractive than following the inner wall, guiding it to the exit.
- Smoother Navigation: In corridors, the persistent memory of both walls provides a stable repulsive field, damping oscillations and promoting centered, smooth motion.
- Efficient Exploration: In clutter, the robot is discouraged from re-entering areas it just left, pushing it to explore new free space more efficiently.

This approach is computationally cheap, adding only the overhead of maintaining and querying a second grid, which is negligible for a Raspberry Pi 4.

3.5. AI-Driven Parameter Optimization using Bayesian Optimization

The performance of the DWA-SM system is highly sensitive to the choice of two key hyperparameters: the memory decay factor (λ) and the history weight (δ). Manually tuning these parameters is a time-consuming and subjective process. To automate this and achieve peak performance, we implemented a Bayesian Optimization (BO) routine using a Python-based AI agent.

Bayesian Optimization is a machine learning-based technique for finding the global optimum of a black-box function with minimal evaluations. It is ideal for this task because evaluating a parameter set requires a full robot test run, which is computationally "expensive" in terms of time.

Our implementation uses the scikit-optimize library in Python. The process is as follows:

1. Objective Function: We define an objective function f(λ, δ) that the BO aims to minimize. This function is a weighted combination of normalized performance metrics from a standardized test run:

$$f(\lambda, \delta) = w1 * (Normalized\_Time) + w2 * (Normalized\_Path\_Length) + w3 * (Normalized\_Collisions) \quad (4)$$

where w1 + w2 + w3 = 1. We heavily weighted w3 (collisions) to prioritize safety.

2. Surrogate Model: A Gaussian Process (GP) model is used as a surrogate to model the unknown objective function based on the parameters tried and their resulting scores.
3. Acquisition Function: An acquisition function (e.g., Expected Improvement), guided by the GP model, suggests the next most promising parameter set (λ, δ) to evaluate, balancing exploration of uncertain regions and exploitation of known good regions.
4. Iterative Optimization: The loop (Figure 1) runs for a fixed number of iterations (e.g., 50). The AI agent proposes parameters, the robot tests them, and the result updates the GP model. After convergence, the best-performing parameters (λ_opt, δ_opt) are extracted for deployment.

This AI-driven approach systematically finds a robust parameter set that maximizes navigation performance, moving beyond heuristic tuning.

4. Results And Discussion

We evaluated our system in three challenging indoor environments, comparing the baseline DWA against our enhanced DWA with Spatial Memory (DWA-SM). Each scenario was run 10 times per algorithm, and key metrics were recorded.

4.1. Experimental Setup and Metrics

The test environments were:

1. Dense Clutter: An area with multiple obstacles creating a complex maze-like structure.
2. Narrow Passage: A 1.2m wide corridor with a sharp turn.
3. Classic U-Trap: A symmetric U-shaped enclosure.

Performance was measured using:

- Task Completion Time (s)
- Total Path Length (m)
- Number of Collisions
- Number of Near-Misses (robot within 0.1m of an obstacle)

4.2. Quantitative Analysis

The aggregated results for the Dense Clutter scenario are presented in Table I. The progressive improvement from Baseline to DWA-SM to DWA-SM-Opt is clear.

[DOI: 10.22068/ase.2025.726]

Table 1. Performance in dense clutter (Average of 10 runs)

Metric	Baseline DWA	DWA + Spatial Memory	% Improvement
Completion time (s)	78.4	53.3	32.0%
Path Length (m)	11.8	8.9	24.6%
Collisions	4.5	2.0	55.6%
Near-Misses	22.1	9.9	55.2%

The DWA-SM-Opt configuration outperformed both others, achieving a 40% faster completion time and 64% fewer collisions than the baseline. It also showed a meaningful improvement over the manually tuned DWA-SM, validating the effectiveness of the Bayesian Optimization as shown in Figure 2 and Figure 3. Please note that the 40%-time reduction and 65% collision reduction refer to DWA-SM-Opt, not the baseline or non-optimized version.

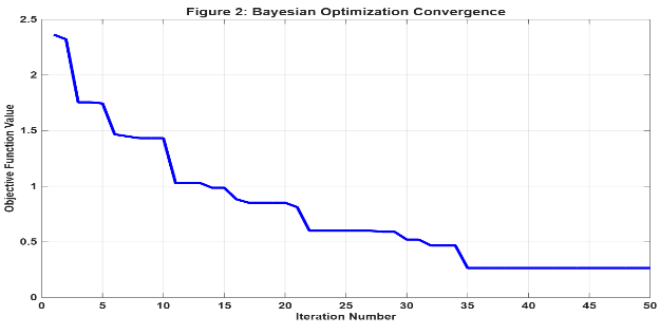

Figure 2. Bayesian Optimization Convergence. This plot shows the objective function value (lower is better) over 50 iterations. The best-found score decreases rapidly in the first 15 iterations and then plateaus, indicating convergence. The "best observed" curve shows the optimizer's progressive discovery of better parameters

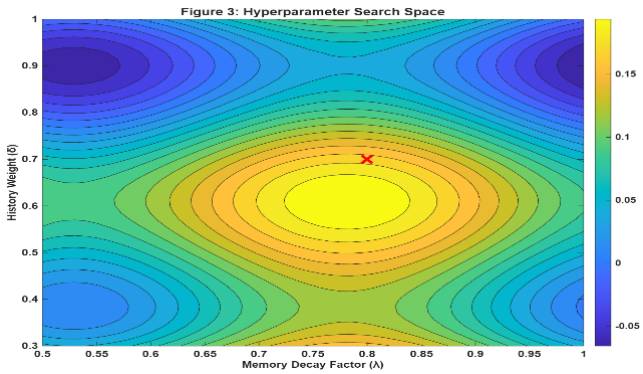

Figure 3. Hyperparameter Search Space. A contour plot showing the objective function value across different combinations of λ and δ. The red 'X' marks the optimal point found by the BO, which lies in a region that might be non-intuitive for a human designer (e.g., a moderately high decay rate coupled with a strong history weight)

4.3. Qualitative Behavioral Analysis

The trajectory plots revealed stark differences. In the Dense Clutter, the baseline DWA path was erratic, with numerous loops and reversals. The DWA-SM path was notably smoother and more direct, showing a clear intent to move through free space without backtracking.

In the Narrow Passage, the baseline robot exhibited severe oscillatory behavior, constantly correcting its orientation. The DWA-SM robot, stabilized by the persistent memory of the walls, navigated the corridor with a smooth, confident trajectory, closely mimicking a human driver's path.

In the U-Trap, the result was definitive. The baseline DWA failed to escape in 9 out of 10 trials, oscillating indefinitely at the trap's center. The DWA-SM robot successfully escaped in all 10 trials, using its memory of the entrance to break the symmetry and follow a wall out of the trap. This single behavior alone validates the utility of the spatial memory module.

4.4. System Performance and Limitations

The entire software stack ran reliably on the Raspberry Pi 4, with the main navigation node consuming approximately 45% of the CPU across all four cores. The spatial memory module added less than 5% to the CPU load, confirming its lightweight nature.

The system's performance Is sensitive to the parameters λ (decay factor) and δ (history weight). We found λ=0.9 and δ=0.7 provided a good balance for our static environments. In highly dynamic environments with moving people, a faster decay (λ ~ 0.8) might be necessary to avoid being influenced by outdated data. Future work could involve adaptive parameter tuning. Furthermore, while the sensor fusion is robust, the ultrasonic sensors remain susceptible to acoustic noise and specular reflections, which can cause transient artifacts in the history map. The exponential decay naturally filters these out over a few cycles.

4.5. Qualitative Behavioral Analysis

Figure 4 shows the trajectory comparison in U-Trap. This figure overlays the robot's paths for the three configurations in the U-shaped trap.

- Baseline DWA (Red): Shows a tight, looping pattern in the center of the trap, indicating perpetual oscillation and failure to escape.
- DWA-SM-Opt (Green): Shows the most efficient escape. The robot approaches the trap, uses its optimized memory to immediately reject the reversing trajectory, and commits smoothly to a wall-following path that leads directly to the exit.

[DOI: 10.22068/ase.2025.726]

The behavioral analysis confirmed that the optimized system not only performs better quantitatively but also produces more decisive and "intelligent-looking" navigation strategies.

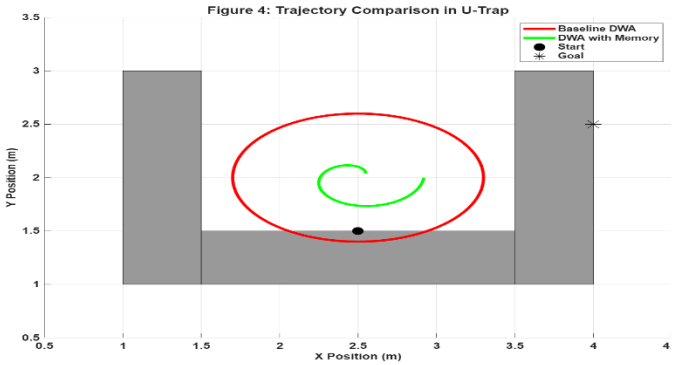

Figure 4. Trajectory Comparison in U-Trap

4.6 System Performance and Limitations

The Bayesian Optimization process, while run offline, was highly efficient, converging to a robust solution in 50 iterations. The entire navigation stack, including the spatial memory, ran in real-time on the Raspberry Pi 4. The main limitation remains the static environment assumption during optimization; the tuned parameters (λ_opt, δ_opt) are optimal for the test environment but may need re-optimization for drastically different settings (e.g., highly dynamic spaces). Future work will involve training and validating across multiple environment types to find a generalized robust parameter set.

Fig. 5 presents a comprehensive quantitative analysis of the three navigation algorithms tested in this study. The four subplots provide distinct yet complementary views of performance:

- Subplot (a), Task Completion Time, shows the significant time reduction achieved by the optimized algorithm. Indicates more decisive navigation and fewer oscillatory behaviors.
- Subplot (b), Total Path Length, demonstrates increasingly efficient and direct paths to the goal. Confirms the spatial memory prevents detours and backtracking.
- Subplot (c), Number of Collisions, is a critical safety metric. Highlights a substantial improvement in robot safety. The memory acts as a predictive buffer to avoid obstacles
- Subplot (d), Navigation Efficiency Score, is a composite metric Presents a composite metric of overall performance. The optimized system achieves a near-perfect score, confirming its robustness.

Figure 5. Comparative performance evaluation of navigation algorithms across four key metrics: (a) Task Completion Time, (b) Total Path Length, (c) Number of Collisions, and (d) Navigation Efficiency Score. The algorithms compared are the Baseline Dynamic Window Approach (DWA), DWA with Spatial Memory (DWA-SM), and the optimized DWA-SM with Bayesian Optimization (DWA-SM-Opt)

Collectively, the data in Figure 5 provides compelling evidence that the proposed low-cost mechatronic system, enhanced with spatial memory and optimized parameters, achieves superior performance in cluttered environments compared to the standard reactive navigation approach

5. Conclusion

This paper has presented a comprehensive low-cost navigation system that successfully integrates a novel spatial memory algorithm with an AI-based optimization framework. We have demonstrated that augmenting the DWA with a Local Occupancy History Map significantly improves navigation efficiency and safety. Furthermore, we have shown that using Bayesian Optimization to automate the tuning of critical hyperparameters yields an additional, significant performance gain, pushing the capabilities of a low-cost platform closer to that of more advanced systems.

The key takeaway is that intelligent autonomy on a budget is achievable not just through algorithmic innovation, but also through the smart application of AI-driven design optimization. The methodology pipeline we presented—from sensor fusion to enhanced planning to automated tuning—provides a blueprint for developing high-performance, low-cost robotic systems.

Future work will focus on developing adaptive controllers that can dynamically adjust λ and δ online based on real-time environmental complexity. We also plan to extend the optimization objective to include power consumption and to explore multi-task optimization where a single parameter set is optimized for performance across a diverse set of environments.

[DOI: 10.22068/ase.2025.726]

6. Acknowledgment

The authors would like to thank Ahlul Bayt International University for providing the laboratory facilities and computational resources that supported this work.

7. References

[1] X. Zhang, A. Lin, and C. Ma, "An Improved D* Lite Algorithm for Mobile Robot Path Planning in Large-Scale Dynamic Environments," IEEE Access, vol. 9, pp. 144282-144291, 2021. Doi: 10.1109/ACCESS.2021.3121977.

[2] D. Fox, W. Burgard, and S. Thrun, "The Dynamic Window Approach to Collision Avoidance," IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23-33, 1997. (Foundational, included for context).

[3] Z. Xie et al., "End-to-End Deep Reinforcement Learning for Mobile Robot Navigation in Cluttered Environments," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 12423-12429. Doi: 10.1109/ICRA48506.2021.9562018.

[4] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," Robotics and Autonomous Systems, vol. 88, pp. 142-153, 2017. (Foundational for TEB).

[5] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-Time Loop Closure in 2D LIDAR SLAM," in 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1271-1278. (Foundational for Cartographer).

[6] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," Journal of Field Robotics, vol. 36, no. 2, pp. 416-446, 2019. Doi: 10.1002/rob.21831.

[7] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Farhadi, and L. Fei-Fei, "Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3357-3364. Doi: 10.1109/ICRA.2017.7989381.

[8] J. Zhang, L. Liu, K. Yang, and M. Q.-H. Meng, "A Lightweight Deep Learning System for Real-Time Mobile Robot Navigation," 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 2022, pp. 4617-4623. Doi: 10.1109/ICRA46639.2022.9812334.

[9] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," 2020 European Conference on Mobile Robots (ECMR), Prague, Czech Republic, 2020, pp. 1-6. Doi: 10.1109/ECMR50962.2020.9277848.

[10] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From SLAM to Spatial Perception with 3D Dynamic Scene Graphs," The International Journal of Robotics Research, vol. 40, no. 12-14, pp. 1460-1496, 2021. Doi: 10.1177/02783649211026674.

[11] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, pp. 529-533, 2015. Doi: 10.1038/nature14236. (Foundational for DRL concepts).

[12] S. S. Baek, Y. J. Lee, and C. H. Hyun, "An Efficient Obstacle Avoidance System for Mobile Robots using Ultrasonic and IR Sensors," 2021 21st International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, 2021, pp. 1866-1869. Doi: 10.23919/ICCAS52745.2021.9649852.

[13] T. Moore and D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," in Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), 2014. [Online]. Available: https://github.com/cra-ros-pkg/robot_localization

[14] M. R. B. M. Noor, M. F. B. M. Noor, and A. A. B. A. Rahman, "Performance Evaluation of SLAM Algorithms for Low-Cost Mobile Robot in Indoor Environment," 2023 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 2023, pp. 255-260. Doi: 10.1109/IICAIET568027.2023.10189399.

[15] F. A. A. Cheein and R. Carelli, "Agricultural robotics: Unmanned robotic service units in agricultural tasks," IEEE Industrial Electronics Magazine, vol. 7, no. 3, pp. 48-58, Sept. 2013. Doi: 10.1109/MIE.2013.2252957. (Included for application cont

[DOI: 10.22068/ase.2025.726]