Adaptive Cost-Map-based Path Planning in Unknown Environments with Movable Obstacles

Liviu-Mihai Stan[1,3], Ranulfo Bezerra[1,2], Shotaro Kojima[1,2], Tsige Tadesse Alemayoh[1,2], Satoshi Tadokoro[1,2]
Masashi Konyo[1], Kazunori Ohno[1,2]

***Abstract*— Reliable navigation in disaster-response and other unstructured indoor settings requires robots not only to avoid obstacles but also to recognise when those obstacles can be pushed aside. We present an adaptive, LiDAR-only path-planning framework that embeds this capability into the ROS 2 *Nav2* stack without any vision pipeline or pre-training. A new *Movable Obstacles Layer* labels all LiDAR returns missing from a prior static map as tentatively movable and assigns a reduced traversal cost. A companion *Slow-Pose Progress Checker* monitors the ratio of commanded to actual velocity; when the robot slows appreciably, the local cost is raised from *light* to *heavy*, and on a stall to *lethal*, prompting the global planner to back out and re-route. Gazebo evaluations on a Scout Mini, spanning isolated objects and cluttered corridors, show higher goal-reach rates and fewer deadlocks than a no-layer baseline, with traversal times broadly comparable. Because the method relies only on planar scans and CPU-level computation, it suits resource-constrained SAR robots and integrates into heterogeneous platforms with minimal engineering. Overall, the results indicate that interaction-aware cost maps are a lightweight, ROS 2-native extension for navigating among potentially movable obstacles in unstructured settings. The full implementation will be released as open source at `https://costmap-namo.github.io`.**

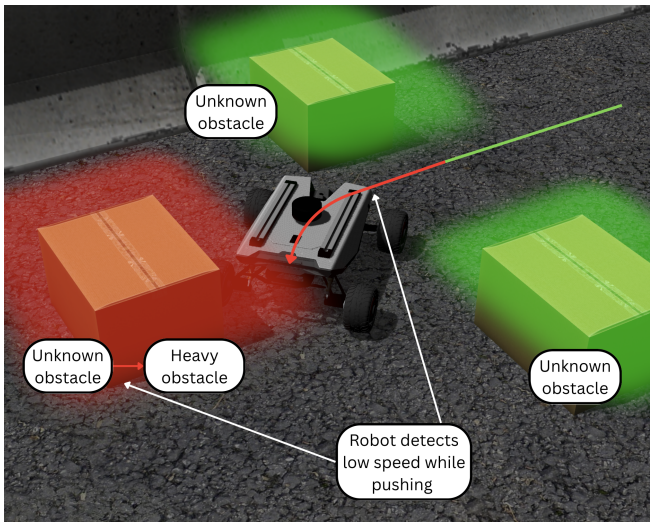

Fig. 1. Scout Mini in a corridor world. The robot pushes an obstacle, detects reduced speed, and labels it as heavy.

I. Introduction

Reliable autonomous navigation is a cornerstone capability for disaster response robotics. In collapsed buildings, smoke filled hospital corridors, or earthquake damaged industrial sites, human access may be hazardous or impossible. Search and rescue (SAR) robots must reach victims, deliver supplies, and explore confined spaces without assistance. The ability to navigate such environments reliably is essential for saving lives, minimizing rescue times, and reducing risks to human responders.

Autonomous robots are increasingly expected to operate in cluttered, *unstructured* indoor environments, such as collapsed corridors or makeshift passages in disaster zones. Researchers refer to this challenge as *Navigation Among Movable Obstacles* (NAMO), where the robot must plan paths that may involve interacting with and repositioning obstacles to reach its goal. In these SAR scenarios, the robot's survival critical task is to reach targets despite debris, overturned furniture, or equipment that *might* be movable but whose properties are unknown. Planners that assume static obstacles, or depend on camera based semantic maps easily degraded by dust and poor lighting, halt as soon as the nominal path is blocked. Approaches that attempt to *move* obstacles often require rich RGB-D perception or offline learning to decide *which* objects can be pushed [1] [2].

However, such perception heavy methods are fragile in SAR, where visibility is reduced by dust or smoke and where time and power budgets preclude GPU based inference. Vision pipelines may fail outright or yield unreliable classifications, leaving the robot unable to progress when faced with unknown obstacles.

The problem addressed here is how to enable a resource constrained SAR robot to determine, in real time and without vision or prior semantics, whether an obstacle can be moved, and to adjust its navigation plan accordingly. This must be done entirely within a standard ROS 2 navigation stack for full compatibility and minimal computational overhead [3].

We propose a LiDAR only, adaptive cost map approach implemented as two lightweight Nav2 plugins. Real time LiDAR scans are compared with a static map; new returns are labelled *movable* in a *Movable Obstacles Layer*. During execution, a custom *Slow Pose Progress Checker* monitors the ratio between commanded and actual velocity: if motion slows, the closest obstacle is escalated from *light* to *heavy*, and if stalled, to *lethal*, prompting replanning. Both plugins

*This research was performed by the commissioned research fund provided by F-REI (JPFR23010101).

[1]Graduate School of Information Sciences, Tohoku University, Japan.
[2]Tough Cyberphysical AI Research Center, Tohoku University, Japan.
[3]Sorbonne University, Paris, France.
bezerra.ranulfo@tr.is.tohoku.ac.jp
liviu.stan99@gmail.com

run at sensor rate, require no vision, and integrate seamlessly with existing planners. Gazebo trials from single box hallways to double row clutters show high success rates where baseline static obstacle planners fail.

Main contributions

- **ROS 2 native, SAR oriented NAMO formulation** eliminating vision and semantic prerequisites, directly integrable with Nav2.
- **Movable Obstacles Layer** for Nav2 costmaps tagging LiDAR detected unknown objects with *adaptive* traversal costs and updating them based on progress checker data.
- **Velocity based progress checker** sending "heavy" or "unmovable" messages to the movable obstacles layer in real time.

The following section reviews relevant literature to contextualize these contributions.

II. Related Work

Stilman and Kuffner's LP1 planner first showed that Navigation Among Movable Obstacles (NAMO) can be solved exactly in the plane by decomposing free space and reconnecting components for real-time solutions with many movable objects [4].

Work then targeted scalability and partial observability. Wu et al. removed the need for a priori maps by incrementally building a world model from 2-D laser scans and pruning search with cost-bounding heuristics [5]. Levihn et al. guaranteed locally optimal decisions in unknown environments using admissible bounds and dual-list pruning, scaling to tens of obstacles without losing real-time performance [6]. Clingerman et al. recast cost learning as Bayesian evidence-grid inference with gamma-distributed cell costs and a lower-confidence bound driving D*-Lite replanning [7].

Perception–manipulation loops added interaction to classification. Kakiuchi et al. used colour TOF sensing and force feedback on a humanoid to re-label obstacles after each push [8], while SwipeBot paired an FCN-ResNet RGB-D segmenter with class-dependent traversal costs updated when the motor current limit is reached [1]. Scholz et al. framed NAMO as a hierarchical MDP that schedules clearing actions while a physics-based RL module re-estimates mass, friction and locked-caster states from force–torque data [9]. Novin et al. learned Bayesian point-mass dynamics for hospital walkers from minimal interactions and coupled them with a hybrid MPC that selects both leg and force direction [10]. Yao et al. train an Advantage Actor–Critic policy for local NAMO that performs non-axial pushes and transfers to a Unitree Go1, enabling fast local planning among pushable objects under sensing noise [2].

Recent planners exploit structure and priors under sensing limits. Muguira-Iturralde et al. formalize visibility-aware NAMO and propose LAMB, which interleaves visibility and manipulation subgoals to outperform NAMO-only and VAMP-only baselines in 3-D scenes with occlusions [11]. He et al.'s Interactive-FAR tightly couples mapping, path-finding and manipulation via a two-level Directed Visibility Graph, re-routing in milliseconds while testing affordances through tactile feedback [12]. Ellis et al. extend online NAMO with multi-object pushing and designated storage zones, planning alternating-axis pushes and caching candidate plans to improve time efficiency while handling LP2, non-monotone dependencies [13].

Orthogonal work in semantic navigation, like IntelliMove, argues for hierarchical semantic topometric maps to guide planning at object/room levels, but requires richer perception than our setting. [14]

III. Adaptive Cost-Map-based Path Planning

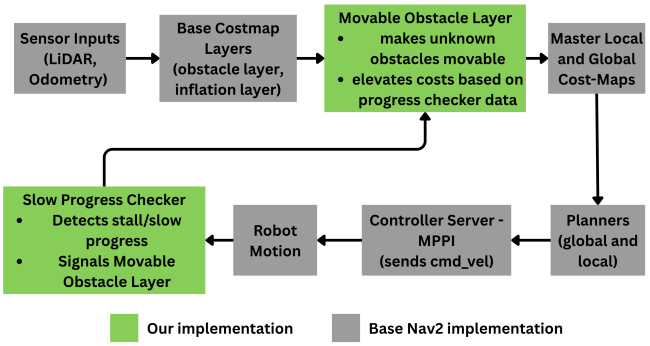

Fig. 2. System architecture of the proposed adaptive cost-map navigation framework. Sensor data are processed through cost-map layers, augmented by obstacle movability assessment and cost escalation, before global and local planning with MPPI control.

Our system equips an AgileX Robotics Scout Mini [15] robot with a LiDAR, modelled in URDF, to build a static map with SLAM Toolbox, localize via AMCL, and navigate using the ROS 2 Nav2 stack augmented by our Movable Obstacles Layer and a Slow-Pose Progress Checker (see Fig. 2 for the system architecture and Fig. 3 for the detailed processing pipeline). Each incoming laser scan is compared against the static map; returns that do not coincide with known structure are tagged as movable and written into a dedicated cost-map layer with a reduced traversal cost. While the Model Predictive Path Integral (MPPI) [16] controller executes the global path, the progress checker monitors the ratio between commanded and actual velocity: if the robot slows, the nearby obstacle's cost is escalated from light to heavy and, if a full stall is detected, to lethal, prompting the global planner to back out and re-route.

A. Costmap

The navigation capabilities of the ROS 2 Navigation (Nav2) stack are fundamentally built upon the Costmap2D representation, a multi-dimensional occupancy grid where each cell encodes a traversal cost based on information from multiple layers. We extend this architecture by introducing a custom plugin, the Movable Obstacles Layer, integrated into Nav2's layered costmap framework [17].

At each update cycle, LiDAR scans, already processed by Nav2's base Obstacle Layer, which marks all detected returns as lethal, are compared against the static SLAM map. Cells

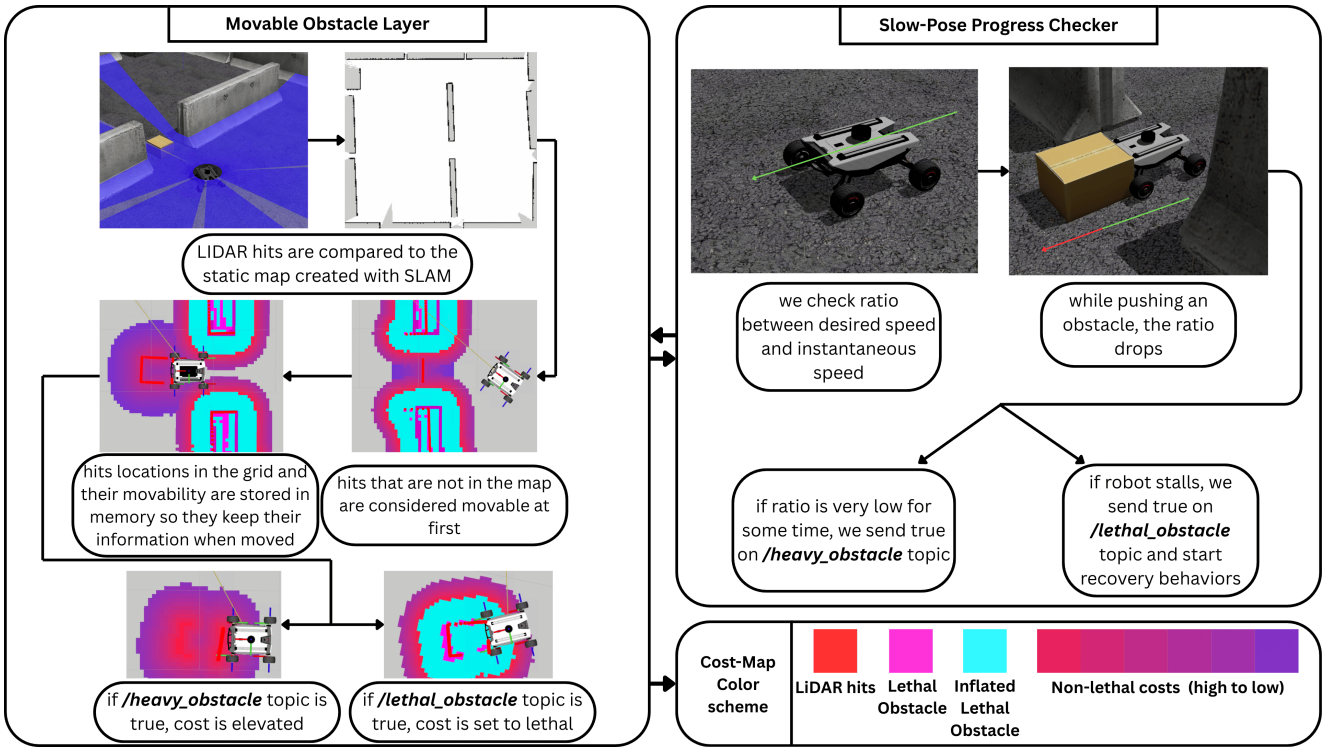

Fig. 3. Detailed flowchart of the adaptive cost-map pipeline and Rviz2 Cost-Map color scheme. The Movable Obstacles Layer compares LiDAR scans to the static SLAM map to classify movable obstacles and update cost-map values, while the Slow-Pose Progress Checker monitors velocity ratios to escalate obstacle costs and trigger recovery behaviors.

corresponding to detections not present in the static map are initially classified as movable, assigned a reduced "light" cost, and given a persistent cluster ID through a connected-component labeling process. This labeling ensures that an obstacle retains its identity and associated cost level even if temporarily occluded or partially moved. By building on the base obstacle layer rather than duplicating its functionality, our approach inherits its efficient clearing behaviour: when a detected obstacle is moved or disappears from the LiDAR view, its lethal marking is removed by the obstacle layer, and our reduced-cost cells are likewise cleared from the master costmap.

The plugin incorporates neighbourhood-based heuristics and a distance-to-wall filter to reduce false positives. Specifically, obstacles detected far enough from static walls (as determined via a precomputed distance transform) are considered movable, while those adjacent to static structures remain lethal.

Integration with the robot's interaction feedback loop enables real-time cost escalation. Two ROS topics, `/heavy_obstacle` and `/lethal_obstacle`, allow external modules such as the Slow-Pose Progress Checker to trigger cost increases. Obstacles that impede motion without complete stalling are reclassified as heavy, increasing their cost, while obstacles that cause full stalls are escalated to lethal.

Unlike the standard inflation layer, our implementation performs weighted soft inflation for all obstacle types. Each obstacle's base cost (light, heavy, lethal) is propagated outward within a configurable radius, decaying exponentially with distance. This yields more natural navigation: the robot prefers obstacle avoidance whenever a collision-free route exists, and initiates controlled pushing only when no such route is available or when doing so produces a better path.

This design enables the robot to adaptively and efficiently plan in cluttered environments, dynamically adjusting its traversability map based on both perception and interaction feedback.

B. Path Planning and Execution

For global path planning, our system utilizes the NavFn-Planner, a graph-based algorithm from the Nav2 stack. The global planner computes feasible paths considering the dynamically updated costmap, ensuring planned routes avoid high-cost or lethal obstacle regions.

The local path execution is managed by the Model Predictive Path Integral (MPPI) controller, selected for its superior reactive capabilities [16]. The MPPI controller, unlike traditional local planners, can perform complex maneuvers such as backward motion, enabling effective repositioning when encountering obstacles that cannot be immediately bypassed. This backward maneuvering significantly enhances local path adaptability, allowing the robot to reposition for optimal obstacle interaction, particularly beneficial after obstacles have been partially moved. This selection was validated through simulation scenarios involving complex obstacle interactions.

C. Progress Checker

To enable velocity-based obstacle cost escalation, we developed a custom ROS 2 Nav2 plugin, the *Slow Pose Progress Checker*, extending the standard `PoseProgressChecker` interface. This module continuously monitors the robot's translational speed relative to its commanded velocity, using both odometry-derived instantaneous speed and the latest `/cmd_vel` commands.

The plugin maintains a *baseline cruise speed* and a configurable *drop ratio* threshold. If the measured speed drops below the configured fraction of the commanded speed for longer than a *freeze window*, and the robot is not performing in-place rotation, the checker concludes that forward progress is obstructed. A short *settling period* after startup and a *cooldown period* between triggers prevent spurious activations.

When an obstruction is detected, the checker publishes a Boolean flag on the `/heavy_obstacle` topic. The *Movable Obstacles Layer* subscribes to this topic and escalates the cost of the nearest obstacle cluster from *light* to *heavy*. If further checks determine the robot is not moving at all despite attempted pushing, escalation to *lethal* occurs through the `/lethal_obstacle` topic. This direct ROS topic interface ensures low-latency coupling between velocity-based interaction feedback and the layered costmap.

By leveraging both odometry and command velocity data, the checker distinguishes between intentional slow movements (e.g., approaching a goal or turning in place) and true mobility loss. This selective escalation enables a graded response (*light* to *heavy* to *lethal*) improving replanning decisions without prematurely abandoning pushable obstacles.

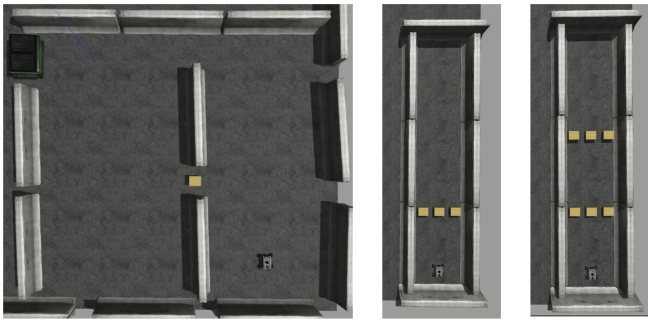

Fig. 4. The maps used as the respective 3 scenarios for the experimental evaluation.

IV. Experimental Evaluation

This section describes the simulation experiments conducted to validate the performance and effectiveness of the proposed adaptive cost-map-based robot path planning system.

The experiments evaluate the robot's navigation capabilities and obstacle movability estimation accuracy within various scenarios designed in Gazebo. The metrics selected for assessment clearly demonstrate the system's adaptability and efficiency compared to conventional static mapping methods.

A. Simulation Scenarios

The simulation tests were conducted across three primary scenarios in customized Gazebo environments (see Fig. 4), specifically created to challenge different aspects of our adaptive cost-map-based navigation system.

Scenario 1 (Individual Obstacle Interactions): This scenario involved an isolated obstacle with different movability levels: light (1-a), heavy (1-b), and immovable (1-c). This setup provided a clear baseline for evaluating the adaptive layer's ability to initially classify and adjust obstacle costs based on real-time interactions.

Scenario 2 (Corridor Mixed Configurations): The corridor setups combined multiple obstacles with varied movability levels, arranged to necessitate sequential interactions. In configurations 2-a, 2-b, and 2-c, the central obstacle directly in the robot's path had differing movability: light, heavy, and immovable respectively. These setups simulated realistic constrained environments such as hallways cluttered with mixed objects.

Scenario 3 (Complex Double-row Obstacles): Designed to test the system's robustness in more complicated situations and recovery capabilities, this scenario involved parallel rows of obstacles with identical movability properties. The complexity of navigating and interacting sequentially to clear a feasible path provided a rigorous test of our adaptive mechanism.

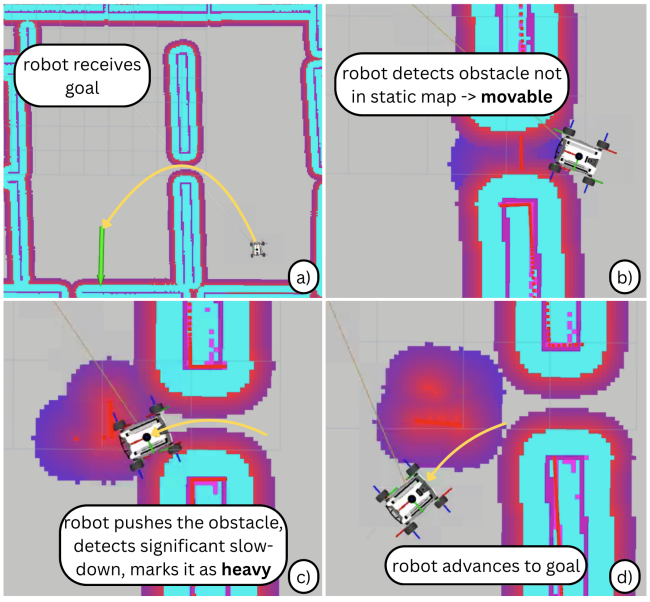

Fig. 5. Scenario 1-b (heavy obstacle): annotated costmap sequence showing the initial plan, insertion of an unmapped return as "movable," slowdown-triggered escalation to "heavy," and the resulting re-route to the goal.

B. Performance Metrics

To quantitatively assess the navigation capabilities, we established the following metrics:

Goal Success Rate: Defined as the percentage of successful navigation attempts to the goal out of 20 trials per scenario.

Navigation Time: Comparison of traversal times required for the robot to reach the goal, specifically contrasting performance with and without the adaptive cost-map layer enabled. Shorter navigation times indicate improved efficiency.

Movability Estimation Accuracy: Evaluated based on the correctness of the adaptive layer's real-time obstacle classification compared to ground truth set in simulation (light, heavy, immovable).

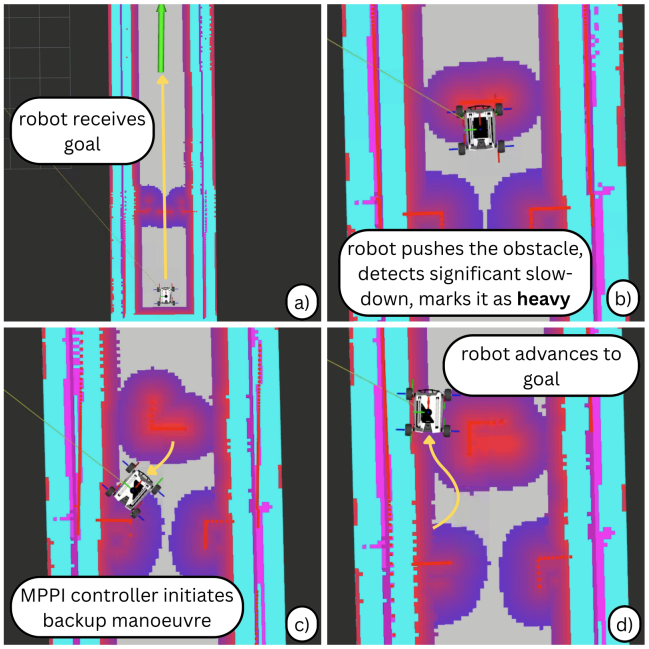

Fig. 6. Scenario 2-b (corridor, heavy center obstacle): annotated costmap sequence (a–d) showing the initial plan; there was slowdown on contact so the robot escalated the cost to heavy; MPPI triggers a backing maneuver; global plan re-routes and the robot proceeds to the goal.

C. Results and Discussion

The results obtained from the simulations (see Table 1) are summarized and discussed as follows:

Scenario 1 illustrates the adaptive layer's capacity to estimate obstacle movability through direct interaction. In 1-a, the robot successfully identified and navigated past light obstacles with minimal difficulty. Scenario 1-b (see Fig. 5 confirmed that while heavy obstacles were still occasionally pushable, their interaction results varied depending on contact angle and robot speed, leading to reduced success rates and contributing to the drop in movability estimation accuracy. Scenario 1-c introduced an immovable obstacle; the robot initially attempted to push it, detected failure via progress checking, and rerouted accordingly. This demonstrates a deliberate trade-off: longer traversal time in exchange for learning and avoiding repeated futile interactions.

Scenario 2 emphasizes the necessity of adaptive movability classification. Without the adaptive layer, baseline methods failed entirely due to treating all obstacles as static and unmovable, with no mechanism to revise that classification. The adaptive system, by contrast, succeeded across all corridor configurations. Notably, light obstacles remained

TABLE I

NAVIGATION RESULTS ACROSS SCENARIOS

Scenario	Goal Success Rate	Time (adaptive vs. no layer)	Movability Accuracy
1-a	100%	27.79s vs. 41.71s (67%)	100%
1-b	80%	34.24s vs. 41.71s (82%)	70%
1-c	85%	60.18s vs. 41.71s (144%)	95%
2-a	100%	impossible	95%
2-b	100%	impossible	60%
2-c	95%	impossible	80%
3	100%	impossible	80%

easy to push and were consistently classified correctly at the end. However, for heavy obstacles (scenario 2-b, see Fig. 6), estimation accuracy again dropped, highlighting the challenge of distinguishing pushable-but-difficult objects under varying approach conditions. Figure 7 shows how the robot would act in scenario 2-c.

Scenario 3 was designed to test the system in a more complex and cluttered environment. The robot successfully navigated through the double-row setup where all obstacles were light. However, the tight spaces and increased number of obstacles led to clustering and localization difficulties, particularly affecting consistent obstacle distinction and map alignment.

These issues in Scenario 3 reflect a broader pattern observed across all scenarios. First, clustered obstacles, particularly those near walls, occasionally led to ambiguous or erroneous classification due to occlusion in LiDAR data. Second, localization drift caused misalignments in costmap updates, resulting in inconsistent obstacle treatment. Finally, while the velocity-based progress checker generally performed well, it

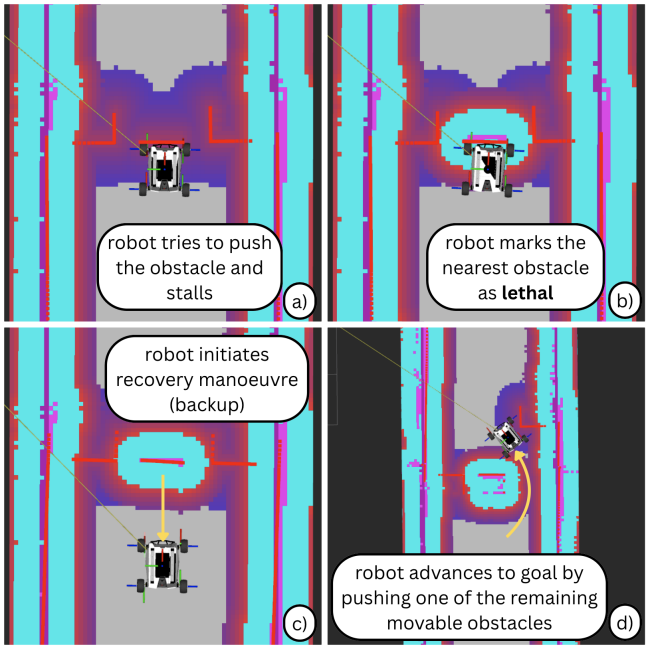

Fig. 7. Scenario 2-c (corridor, immovable center obstacle): annotated costmap sequence (a–d) showing attempted push and stall; escalation of the nearest obstacle to lethal; Recovery behavior (back-up); re-route to the goal by pushing a remaining movable side obstacle.

sometimes struggled to distinguish between legitimate low-speed maneuvering and actual pushing failure, especially during tight turns or cautious movements.

V. Conclusions and Future Work

This research introduced an adaptive cost-map-based path-planning system designed for autonomous robot navigation in unknown environments cluttered with movable obstacles. The system assigns a provisional 'movable' cost to any LiDAR returns not in the static map and updates that cost online when interaction cues (slowdown/stall) indicate resistance. Core contributions include the development of the specialized "Movable Obstacles Layer" plugin for ROS2's Nav2 costmap stack, an adaptive costmap methodology that dynamically adjusts obstacle traversal costs, and a velocity-based progress checker that monitors interaction feedback to reassess obstacle movability during navigation. Simulations across seven Gazebo scenarios indicate improved goal-reach rates and fewer deadlocks relative to a static obstacles cost-map baseline that cannot push; traversal times were mostly improved, and movability cues were reliable for light/immovable objects but less so for heavy ones.

The primary strengths of our approach include its generalizability across diverse obstacle setups, the simplicity and computational efficiency of integration into standard ROS2 navigation frameworks, and the elimination of the need for extensive pre-trained semantic or vision-based models.

Despite these strengths, several challenges remain. Using robot velocity alone to infer pushing effectiveness revealed limitations, particularly in instances of near-stalling or during turns. Additionally, occasional localization drift led to temporary costmap inaccuracies, affecting obstacle status updates. Obstacle clustering also posed challenges, occasionally resulting in ambiguous classifications due to partial sensor occlusion.

Addressing these limitations offers clear avenues for future research. We plan to enhance the reliability of movability estimation by incorporating motor effort and load measurements, providing a complementary metric alongside velocity-based methods. Advanced clustering and obstacle disambiguation algorithms will also be explored to better distinguish obstacles in close proximity to static structures. Further improvements in localization and map consistency techniques will be integrated to reduce transient costmap errors. Finally, transitioning from simulation-based evaluations to real-world robotic deployments in indoor environments will be pursued to comprehensively assess the practical applicability and robustness of our adaptive navigation approach.

References

[1] N. Zherdev, M. Kurenkov, K. Belikova, and D. Tsetserukou, "SwipeBot: DNN-based Autonomous Robot Navigation among Movable Obstacles in Cluttered Environments," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*. Florence, Italy: IEEE, Jun. 2023, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/10200601/

[2] L. Yao, V. Modugno, A. M. Delfaki, Y. Liu, D. Stoyanov, and D. Kanoulas, "Local path planning among pushable objects based on reinforcement learning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 3062–3068.

[3] S. Macenski, F. Martin, R. White, and J. G. Clavero, "The Marathon 2: A Navigation System," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 2718–2725. [Online]. Available: https://ieeexplore.ieee.org/document/9341207/

[4] M. Stilman and J. Kuffner, "Navigation among movable obstacles: real-time reasoning in complex environments," vol. 1, pp. 322–341 Vol. 1, 2004.

[5] Hai-Ning Wu, M. Levihn, and M. Stilman, "Navigation Among Movable Obstacles in unknown environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: IEEE, Oct. 2010, pp. 1433–1438. [Online]. Available: http://ieeexplore.ieee.org/document/5649744/

[6] M. Levihn, M. Stilman, and H. Christensen, "Locally optimal navigation among movable obstacles in unknown environments," in *2014 IEEE-RAS International Conference on Humanoid Robots*. Madrid, Spain: IEEE, Nov. 2014, pp. 86–91. [Online]. Available: http://ieeexplore.ieee.org/document/7041342/

[7] C. Clingerman, P. J. Wei, and D. D. Lee, "Dynamic and probabilistic estimation of manipulable obstacles for indoor navigation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany: IEEE, Sep. 2015, pp. 6121–6128. [Online]. Available: http://ieeexplore.ieee.org/document/7354249/

[8] Y. Kakiuchi, R. Ueda, K. Kobayashi, K. Okada, and M. Inaba, "Working with movable obstacles using on-line environment perception reconstruction using active sensing and color range sensor," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: IEEE, Oct. 2010, pp. 1696–1701. [Online]. Available: http://ieeexplore.ieee.org/document/5650206/

[9] J. Scholz, N. Jindal, M. Levihn, C. L. Isbell, and H. I. Christensen, "Navigation Among Movable Obstacles with learned dynamic constraints," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea: IEEE, Oct. 2016, pp. 3706–3713. [Online]. Available: http://ieeexplore.ieee.org/document/7759546/

[10] R. S. Novin, A. Yazdani, T. Hermans, and A. Merryweather, "Dynamic Model Learning and Manipulation Planning for Objects in Hospitals Using a Patient Assistant Mobile (PAM)Robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/8593989/

[11] J. Muguira-Iturralde, A. Curtis, Y. Du, L. P. Kaelbling, and T. Lozano-Pérez, "Visibility-aware navigation among movable obstacles," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 083–10 089.

[12] B. He, G. Chen, W. Wang, J. Zhang, C. Fermuller, and Y. Aloimonos, "Interactive-FAR:Interactive, Fast and Adaptable Routing for Navigation Among Movable Obstacles in Complex Unknown Environments," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Abu Dhabi, United Arab Emirates: IEEE, Oct. 2024, pp. 5402–5409. [Online]. Available: https://ieeexplore.ieee.org/document/10802169/

[13] K. Ellis, D. Hadjivelichkov, V. Modugno, D. Stoyanov, and D. Kanoulas, "Navigation among movable obstacles via multi-object pushing into storage zones," *IEEE Access*, vol. 11, pp. 3174–3183, 2023.

[14] F. Ngom, H. Zhang, L. Zhang, K. Godary-Dejean, and M. Huchard, "Intellimove: Enhancing robotic planning with semantic mapping," 2024. [Online]. Available: https://arxiv.org/abs/2410.14851

[15] AgileX Robotics, "Scout mini," https://global.agilex.ai/products/scout-mini, 2025, [Online; accessed 8-Aug-2025].

[16] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm: IEEE, May 2016, pp. 1433–1440. [Online]. Available: https://ieeexplore.ieee.org/document/7487277/

[17] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Chicago, IL, USA: IEEE, Sep. 2014, pp. 709–715. [Online]. Available: http://ieeexplore.ieee.org/document/6942636/