Autonomous Navigation of a Mobile Robot in a Known Map with Dynamic Obstacles

Furkan Sabri Dipi[1] and Ismail Uyanik[2]

Abstract— **The expansion of autonomous mobile robots from industrial facilities into dynamic environments shared with humans has made it imperative for navigation algorithms to operate reliably under new uncertainties. While traditional navigation methods succeed in static maps, they often cause the "freezing robot" problem in chaotic scenarios involving suddenly moving obstacles. This study presents the mechanical, electronic, and software design of an integrated autonomous navigation system capable of safely and smoothly avoiding static and dynamic obstacles on a known map. Local planning is performed using the Model Predictive Path Integral (MPPI) controller on a custom-designed skid-steer mobile robot platform, featuring a slip-aware EKF configuration and an asymmetric acceleration envelope, with full parameter settings reported for reproducibility. We present this as a systems-integration and reproducibility study rather than a new algorithmic contribution. In a representative single-run real-world evaluation, the tuned stack cleared 23 unmapped static obstacles and 17 dynamic encounters (pedestrians and a quadruped robot) with zero collisions and no observed freeze events under our success definitions; we do not claim a statistical success rate, and multi-trial validation is the primary follow-up. A detailed event-level analysis demonstrates the interplay between the MPPI planner's backward yielding capabilities and Behavior-Tree (BT) recovery on human-shared corridors.**

Index Terms— **autonomous navigation, mobile robots, MPPI, skid-steer, dynamic obstacle avoidance, ROS 2, Nav2**

I. Introduction

Autonomous mobile robots operating in dynamic environments must make safe decisions in milliseconds when faced with unexpected obstacles [1]. Classical reactive methods such as Artificial Potential Fields [2] get stuck in local minima, the Dynamic Window Approach [3] reacts only to the instantaneous situation, and gradient-based Model Predictive Control [4] struggles with non-differentiable occupancy grids; modern stacks address these via hierarchical architectures [5]. The Model Predictive Path Integral (MPPI) controller [6], [7], which replaces gradients with sampling and is now featured by the ROS maintainers in Nav2 [8], has emerged as the modern alternative. This paper proposes a skid-steer platform designed from scratch for narrow indoor spaces and experimentally evaluates its performance against dynamic obstacles by combining a 3D/2D hybrid LiDAR perception system and Extended-Kalman-Filter (EKF)-based sensor fusion with an MPPI controller in the ROS 2 Nav2 ecosystem.

This work is presented as a *systems-integration and reproducibility study* rather than a new algorithmic framework: the individual building blocks (MPPI, AMCL, EKF-based wheel/IMU fusion, BT recoveries) are established, but their combined deployment on a skid-steer platform raises practical questions on parameter selection, slip handling, and safety-oriented control envelope design that are rarely addressed in application papers in reproducible detail. Our contributions are threefold: (i) a fully documented Nav2 MPPI deployment recipe for skid-steer platforms, reproducible on comparable hardware; (ii) a slip-aware integration pattern discarding EKF position channels combined with a deliberately designed asymmetric acceleration envelope justified by skid-steer-specific traction arguments; and (iii) a hybrid 3D/2D perception topology ensuring near-field safety. Together they form a reference configuration tractable on a Jetson Orin NX. Mathematical symbols are defined where first introduced (kinematics in Section II-B, planner parameters in Section III-B).

II. Skid-Steer Robot Platform

A. Mechanical and Electronic System

The mobile robot is designed in a skid-steer kinematic configuration, allowing it to rotate around its own center in indoor environments. The main chassis is constructed using aluminum sigma profiles for a high strength-to-weight ratio. The robot's dimensions are 651×263 mm, and it is driven by four 146-mm-diameter wheels.

The actuation system uses brushed DC motors with a 131.25:1 gear ratio, capable of producing high torque at low speeds. As illustrated in Fig. 1, low-level motor control and encoder readings are performed by an ESP32 microcontroller executing real-time PID control loops at 50 Hz. High-level autonomy tasks are executed on an NVIDIA Jetson Orin NX 16 GB embedded computer, which provides 100 TOPS of AI processing power via 1024 CUDA cores, critical for real-time point-cloud processing and MPPI trajectory simulation.

B. Kinematic Modeling of the Skid-Steer Robot

Unlike differential-drive robots, skid-steer mobile robots (SSMR) require lateral wheel slippage to change direction. The robot's motion is defined by its generalized coordinate vector $q = [X, Y, \theta]^T$ in the inertial frame, and its local linear and angular velocities $\eta = [v_x, \omega]^T$, where v_y is the lateral velocity and $\dot{\theta}$ the yaw rate [9].

*This work was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under grant no. 125E335 awarded to Ismail Uyanık.

[1]Furkan Sabri Dipi is with the Department of Electrical and Electronics Engineering, Hacettepe University, Ankara, Türkiye furkandipi@hacettepe.edu.tr

[2]Ismail Uyanik is with the Department of Electrical and Electronics Engineering, Hacettepe University, Ankara, Türkiye uyanik@hacettepe.edu.tr

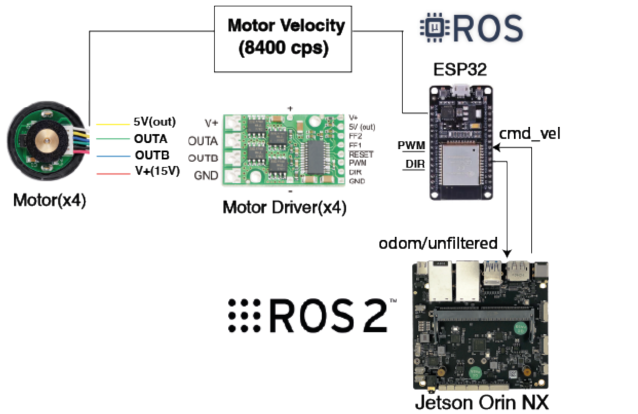

Fig. 1. Architecture of the communication between the high-level Jetson Orin NX and the low-level ESP32 motor-control group.

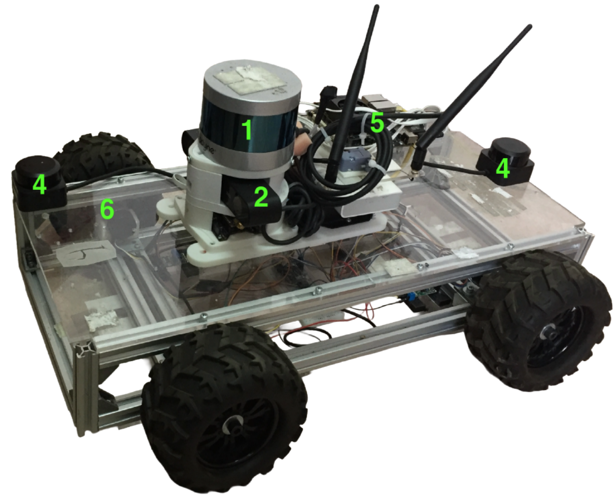

Fig. 2. Physical architecture of the developed skid-steer mobile robot, highlighting the placement of the computational units and sensory hardware.

Due to the non-holonomic nature of the skid-steer mechanism, an operational constraint is applied using a constant parameter x_0 within the wheelbase $(-a, b)$ to approximate the relationship between angular and lateral velocities [9]:

$$v_y + x_0\,\dot{\theta} = 0, \qquad x_0 \in (-a, b). \tag{1}$$

The inverse kinematics converting local velocities to wheel speeds is computed using standard skid-steer approximations [9]. This inverse kinematic relation translates the velocity commands generated by the MPPI planner into physical motor control signals.

Slip modeling assumptions. Equation (1) replaces the true instantaneous center of rotation (ICR), which migrates along the longitudinal axis with surface friction and commanded angular velocity, by a fixed design parameter x_0. The MPPI planner adopts the same approximation by setting $x_0 = 0$ (Section III-B), collapsing the kinematics to a differential-drive form and introducing a systematic model-reality mismatch during turning. This mismatch is accurate only when longitudinal slip is not dominant, a condition that holds at the sub-0.5 m/s operating speeds used throughout this work but degrades on slippery surfaces and during aggressive high-speed maneuvers. We do not identify the ICR online; instead, slip-induced corruption of wheel odometry is absorbed by the state estimator described in Section III-A, which intentionally discards the integrated position estimates of wheel odometry and retains only its velocity estimates, while the IMU dominates the orientation state and suppresses slip-induced yaw error. The smooth, centered global trajectory of Fig. 3 indicates that the residual lateral tracking error during turning stays within the corridor clearance margin at the tested speeds; a quantitative characterization across a wider speed range is left for future work.

C. Hybrid Perception System

To ensure 360-degree continuous environmental awareness, a hybrid 3D/2D sensor architecture is employed. The physical placement of these components is detailed in Fig. 2, and their corresponding specifications are provided in Table I.

TABLE I
PHYSICAL HARDWARE COMPONENTS OF THE MOBILE ROBOT.

No.	Component	System function
1	VLP-16 3D LiDAR	Primary 3D obstacle perception
2	StereoLabs ZED2i	Video documentation only; no data to navigation stack
3	Xsens MTi-7-01-DK	IMU/GNSS for orientation
4	Slamtec RpLidar S3	2D LiDAR for blind-spot coverage
5	NVIDIA Jetson Orin NX	High-level AI computational unit
6	Pololu 37D Gearmotor	131.25:1 geared actuation

The primary sensor is a Velodyne VLP-16 3D LiDAR, which provides 300,000 points/s over 16 channels at a 10 Hz rotation rate. However, due to the robot's dimensions and the sensor's vertical field of view, a severe near-field blind spot of 0–45 cm occurs immediately around the chassis. To eliminate this critical perception gap and reliably detect close-proximity dynamic obstacles (such as human feet or moving objects), two Slamtec RpLidar S3 2D LiDAR sensors are strategically mounted on the front and rear lower sections of the chassis (component 4 in Table I). Orientation and angular velocity are tracked using an Xsens MTi-7 GNSS/INS module (6°/h drift) running at 100 Hz. An onboard StereoLabs ZED 2i camera is used strictly for video recording, providing no data to the navigation stack.

III. AUTONOMOUS NAVIGATION FRAMEWORK

A. State Estimation and Localization

Wheel slippage inherent to skid-steer vehicles introduces severe odometry drift during turning maneuvers. This is mitigated by fusing the raw wheel encoder data with the MTi-7 IMU data using an Extended Kalman Filter (EKF) running at 30 Hz. The EKF is configured to get only the velocity components of wheel odometry (i.e., v_x and ω_z) while discarding the integrated position estimates, so that slip-induced drift cannot propagate into the global pose. The IMU dominates the orientation estimation, effectively suppressing slip-induced yaw error.

Global localization on the mapped environment (mapped via SLAM Toolbox [10]) is handled by Adaptive Monte Carlo Localization (AMCL) [11], [12], dynamically adjusting the particle count between 500 and 2000 based on the convergence of the distribution.

B. Costmap Configuration and Local Planning with MPPI

Nav2 utilizes layered costmaps (Static, Obstacle, Denoise, and Inflation layers) to represent the environment [13]. The inflation radius is set to 1.75 m with a gentle cost scaling factor $\omega_{\text{scale}} = 2.58$, producing the exponential cost decay $c(d) = (c_{\text{lethal}} - 1)\, e^{-\omega_{\text{scale}}(d - r_{\text{insc}})}$ around every lethal cell.

These values were selected through a direct comparison against a sharper profile (radius 0.55 m, $\omega_{\text{scale}} = 5.0$). The chosen softer profile yields centered traversal even through doorways, consistent with the ROS navigation tuning guidance of [14] that a low-slope inflation decay biases the planner toward the corridor midline. The corridors used in the experiments are approximately 2.2 m wide; the chosen inflation radius therefore still leaves meaningful free space for the planner along the corridor midline while keeping the robot's 0.65 m footprint at a safe lateral clearance. Informal sensitivity sweeps around this operating point showed that reducing the inflation radius below $\sim$ 1.25 m produced wall-hugging during yielding maneuvers, while raising ω_{scale} beyond $\sim$4.0 produced oscillation near narrow passages.

Local planning and dynamic obstacle avoidance are delegated to the Model Predictive Path Integral (MPPI) controller. Because standard MPC requires differentiable cost functions, MPPI approaches the problem stochastically using an information-theoretic framework [7]. The MPPI algorithm evaluates K sampled trajectories over a prediction horizon, injecting random noise into the control sequence to explore the state space. The optimal control input is then updated via an importance-sampling-based weighted average without requiring gradient calculations, directly utilizing the standard formulation from [6] and [7].

Controller configuration. The MPPI controller runs at 20 Hz with $\Delta t = 0.05$ s over a 56-step horizon ($T = 2.8$ s). At each control cycle, $K = 1000$ noisy trajectories are sampled in parallel on the Jetson Orin NX, yielding an effective rollout throughput of 2×10^4 trajectories per second. Control noise is drawn from a zero-mean Gaussian with $\sigma_{v_x} = \sigma_{v_y} = \sigma_{\omega_z} = 0.2$. Velocity and acceleration limits are $v_x \in [-0.35, 0.5]$ m/s, $\omega_z \in [-1.9, 1.9]$ rad/s, $a_x \in [-3.0, 3.0]$ m/s^2, and $\alpha_z \in [-3.5, 3.5]$ rad/s^2. The temperature is $\lambda = 0.3$ and the importance-sampling parameter $\gamma = 0.015$. The motion model is set to *DiffDrive* (i.e., $x_0 = 0$ in (1)); see Section II-B for the resulting model-reality mismatch and its empirical impact.

Cost composition and tuning. The trajectory cost is a weighted sum $J(\tau) = \sum_{i \in \mathcal{C}} w_i\, C_i(\tau)$ over the eight active critics listed in Table II. Weights were tuned iteratively: the obstacle-related critics (CostCritic, ConstraintCritic) were fixed first to guarantee the minimum corridor clearance required by the test environment; the path-following critics (PathAlignCritic, PathFollowCritic, PathAngleCritic) were then increased until residual oscillation around the global plan vanished; the temperature λ was finally reduced until the deceleration profile reached the asymmetric braking behavior reported in Section IV-C.

TABLE II
MPPI CRITIC FUNCTIONS AND WEIGHTS USED IN THE EXPERIMENTAL STACK.

Critic	Weight	Role
ConstraintCritic	4.00	Kinematic limit enforcement
GoalCritic	5.00	Euclidean distance to goal
GoalAngleCritic	3.00	Terminal heading alignment
PreferForwardCritic	5.00	Penalty on reverse motion
CostCritic	3.81	Soft cost from inflation layer
PathAlignCritic	7.00	Lateral deviation from path
PathFollowCritic	5.00	Forward lookahead on path
PathAngleCritic	2.00	Heading–path tangent alignment

Reactive obstacle handling. In the current stack, dynamic obstacles are treated as instantaneously static within each planning cycle: the obstacle layer is rebuilt from the latest LiDAR scans at every update, and MPPI rolls out the 2.8 s horizon assuming a fixed obstacle footprint. The stack is therefore predictive on the robot's own dynamics but *reactive-with-lookahead* on surrounding agents. At the 20 Hz control update rate this proved sufficient for the lateral yielding maneuvers reported in Section IV, but it becomes conservative for head-on encounters where predictable obstacle motion could be exploited (addressed in future work).

Higher-level decision-making is modeled using BT [15]. If the route is blocked, the BT triggers multi-level recovery behaviors sequentially: clearing costmaps, spinning in place, waiting, and backing up.

IV. Experimental Results

The proposed system was evaluated at the Department of Electrical and Electronics Engineering, Hacettepe University. To systematically validate the integration choices of Sections II and III, we designed three progressively challenging scenarios that form an *implicit ablation hierarchy*: an obstacle-free baseline, a static-obstacle pre-validation, and the main dynamic-obstacle test. We briefly summarize the static tests for context before a detailed analysis of the dynamic scenario, which forms the core of our experimental results.

A. Experimental Setup and Success Metric

The evaluation methodology is structured into three progressive scenarios to systematically isolate the impact of different environmental uncertainties on the navigation stack:

1) **Unobstructed Baseline (S1):** A reference run in an empty corridor to establish the nominal tracking performance and maximum cruise speed.
2) **Static Obstacles (S2):** A pre-validation run on a 139.4 m route involving 23 unmapped, stationary obstacles (e.g., bins, planters) placed post-mapping.

The robot successfully cleared all 23 encounters without contact, validating the perception-costmap-planner pipeline.

3) **Dynamic Scenario (S3):** The primary test case, consisting of a continuous 153.76 m run through indoor corridors to expose the robot to 17 independently generated dynamic encounters (9 cut-ins by a quadruped robot dog, 5 pedestrian crossings, and 3 blocked-corridor situations resolved by either MPPI reverse maneuvers or BT recovery).

For the dynamic encounters in S3, pedestrians crossed at self-selected moments, and the quadruped was commanded with randomly selected motion primitives (approach, sudden stop, lateral cut-in) at unannounced waypoints. The run reported here is the *representative instance* of the tuned stack: it was preceded by numerous development runs on the same corridor during which the critic weights, inflation profile, and EKF configuration of Section III-B were iteratively refined. The intent is to document that this operating point is reproducibly attainable, rather than to argue that it is the outcome of a randomized statistical sample; a fully randomized multi-trial validation remains an immediate line of follow-up work.

We adopt an *event-level* success metric rather than a trial-level one. Each of the 17 encounters is classified as successful if the robot cleared it without collision and resumed forward motion toward the next global-path waypoint within 30 s; an encounter not meeting this within the window would be recorded as failed (none did in the reported run). A *collision* is contact with a LiDAR distance below 0.05 m confirmed from onboard video; a *near-collision* is a minimum LiDAR distance below 0.15 m without contact; a *freeze event* is a commanded linear velocity below 0.02 m/s sustained for at least 5 s before the goal is reached. Under these definitions the run yielded $17/17$ successful encounters, 0 collisions, and 0 freeze events.

B. Trajectory and Visual Analysis

As shown in Fig. 3, the robot dynamically curved its route to yield to incoming obstacles while maintaining fluid motion. *Within the dynamic run itself*, we define an *active obstacle interaction* segment as any time interval during which at least one dynamic obstacle is present in the local costmap within the 2.8 s MPPI rollout horizon; all remaining intervals are treated as unobstructed. Under this segmentation the average speed during active interactions (0.265 m/s) dropped by only 5.9% compared to the unobstructed segments (0.282 m/s), indicating that the planner's response to each encounter is locally gentle.

As shown in Table III, average speed drops from 0.400 m/s (S1) to 0.288 m/s (S3), a 28.0% overall reduction, indicating that the MPPI stack prioritizes proactive yielding and safety over aggressive velocity tracking; no freeze events were observed under the definitions of Section IV-A. We note that, since no comparative experiment against alternative local planners was conducted, this observation is not a claim of superiority over reactive planners such as DWA or TEB.

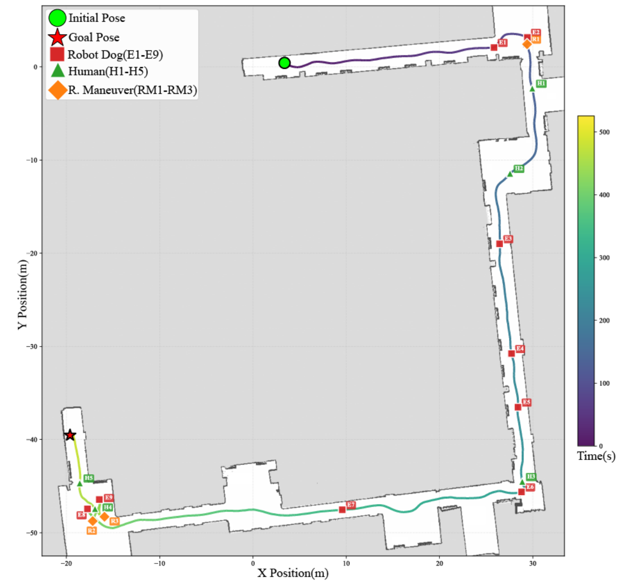

Fig. 3. Global trajectory of the robot (Y vs. X position in m, color-coded by elapsed time in s) mapping the locations of dynamic encounters and the temporal distribution of retraction events.

TABLE III
PERFORMANCE COMPARISON ACROSS THREE EVALUATION SCENARIOS.

Metric	S1: Baseline	S2: Static	S3: Dynamic
Total Distance (m)	131.34	139.40	153.76
Total Time (s)	350.46	436.00	533.70
Avg. Speed (m/s)	0.400	0.320	0.288

Fig. 4 showcases the real-world perspective of these encounters. The hybrid 3D/2D perception system proved vital in these narrow corridor segments. Because the robot dog and human feet often entered the blind spot of the 3D LiDAR, the integration of the 2D LiDARs successfully prevented near-field collisions, feeding accurate instantaneous data into the local costmap.

C. Velocity Profiles and Retraction Maneuvers

To thoroughly evaluate the controller's reactivity and safety margins, we analyzed the linear velocity (V_x) profiles during specific critical events, as shown in Fig. 5. The letters A–E denote specific temporal segments corresponding to different interaction types.

Phase A (60–115 s) in Fig. 5 illustrates the initial encounter and recovery (E2, E3) where the robot aggressively decelerates to a halt upon detecting the obstacle and safely resumes motion. The system demonstrated a maximum deceleration of -1.359 m/s^2, approximately $9\times$ its maximum acceleration (0.149 m/s^2).

This 9:1 asymmetry is not an emergent property of the dynamics but a deliberate design choice, encoded at the velocity-smoother layer (`max_accel`=0.3, `max_decel`=-0.5 m/s^2) and reinforced by the MPPI acceleration limits of Section III-B. Three skid-steer-specific considerations justify

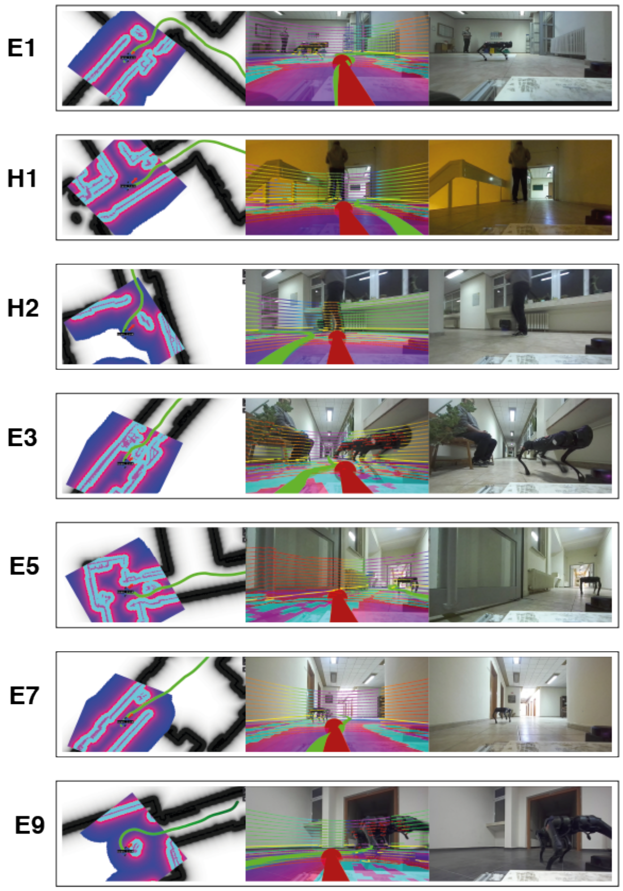

Fig. 4. Real-world snapshots demonstrating the robot executing agile avoidance maneuvers during representative encounters: five quadruped cut-ins (E1, E3, E5, E7, E9) and two pedestrian crossings (H1, H2). Each row shows the local costmap (left), the global-plan overlay (center), and the onboard camera view (right).

this choice: (i) aggressive forward acceleration amplifies lateral wheel slip under payload, degrading the very odometry the planner depends on; (ii) braking engages all four wheels symmetrically in longitudinal traction and does not induce additional slip, making rapid deceleration a "free" safety resource relative to rapid acceleration; (iii) in human-shared corridors the dominant failure cost is collision rather than journey time, so biasing the control envelope toward rapid stopping yields a favorable operating point. The interplay with PreferForwardCritic (weight 5.0 in Table II, with per-step penalty $\Psi(v_x) = |v_x|$ when $v_x < 0$ and 0 otherwise) explains why reverse motion only emerges when the robot is fully blocked: backward maneuvers are selected only once the accumulated path-following cost exceeds the forward-preference penalty on the reverse rollouts. In Phase A, a prolonged blockage eventually triggered a formal BT recovery (RM1).

Phases B and C illustrate two qualitatively different obstacle topologies. In Phase B (180–265 s) the quadruped (E5) moved in the same longitudinal direction as the robot for roughly 30 s, producing a following-vehicle geometry rather than a head-on block; the robot maintained a reduced but positive cruise speed around 0.25 m/s, confirming that the

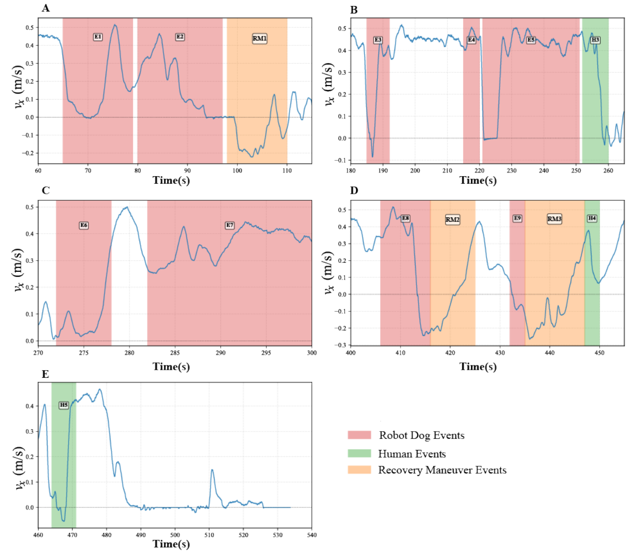

Fig. 5. Linear velocity profiles of the robot (V_x in m/s vs. time in s) during critical obstacle encounter phases (labeled A–E), highlighting aggressive deceleration and backward retraction maneuvers.

TABLE IV
STATISTICS OF BACKWARD RETRACTION MANEUVERS.

Code	Time (s)	Dur. (s)	Min v_x	Mean v_x	Source / Trigger
RM1	98–110	12	−0.223	−0.097	BT Recovery
RM2	416–425	9	−0.218	−0.010	MPPI Reverse
RM3	435–447	12	−0.266	−0.067	MPPI Reverse

critic stack distinguishes a leader-follower regime from a blocked corridor and avoids unnecessary maneuvers. Phase C (270–300 s) tests lateral maneuverability during a corner-turn encounter (E6), where the skid-steer point-turn capability combined with costmap-guided yielding produced a smooth lateral deviation without stopping. Retraction maneuvers were not invoked in these two phases.

Furthermore, Phase D highlights a "dense recovery zone" (400–455 s) where the corridor was completely blocked by the robot dog (E8, E9) and a human (H4)—a scenario that typically triggers the freezing-robot problem in purely reactive algorithms. Instead of freezing or triggering another BT recovery sequence, the MPPI planner's forward penalty was outweighed by collision costs, naturally generating backward yielding maneuvers (RM2, RM3) to create spatial clearance. MPPI's next cycle then encountered an updated costmap in which the blocked-corridor signature had dissolved, and a forward trajectory was successfully selected. Table IV consolidates all three retraction events: mean duration was 11 s, the most aggressive retraction (−0.266 m/s) occurred during RM3, and all three maneuvers were triggered by quadruped-induced blockages rather than pedestrian crossings—consistent with pedestrians passing through the corridor within seconds, whereas the quadruped tended to occupy the same stretch persistently. Finally, Phase E (460–480 s) encompasses the final human encounter (H5), resolved smoothly without triggering any retraction. These interconnected hardware and software responses yielded $17/17$ successful encounter resolutions with zero collisions in this single representative run.

V. Conclusion

We presented an integrated skid-steer navigation stack combining a hybrid 3D/2D LiDAR perception system, EKF-based sensor fusion with explicit handling of skid-steer slip, AMCL localization, and an MPPI local planner with BT recoveries. The contribution is the integration, the deliberate design-envelope choices, and the fully transparent reporting of established building blocks rather than a new algorithmic framework: the complete MPPI configuration, cost composition, slip-aware EKF pattern, and asymmetric acceleration envelope are reported to support reproducibility. In the representative single run reported here, the system cleared all 23 unmapped static obstacles and all 17 dynamic encounters on the main 153.76 m route without collision, and no freeze events were observed under the definitions of Section IV-A, resolving dense blockages through both MPPI-level reverse yielding and BT recoveries. We stress that this is a single-run existence demonstration, not a statistical performance estimate. Because no comparative experiment against alternative local planners was conducted, we do not claim superiority over reactive planners such as DWA or TEB; we report only that no freeze events were observed with the present MPPI stack in the tested scenarios. From a practical-deployment standpoint, the low-speed (sub-0.5 m/s) operating regime together with the fully reported configuration is intended to make the stack a directly transferable reference baseline for human-shared indoor settings, where collision cost dominates journey time. The event-level metric and the per-retraction statistics (Table IV) provide a reproducible basis for future comparisons.

Four extensions are identified as immediate future work. First, and most importantly, a fully randomized multi-trial validation with repeated independent runs is required to establish a statistical basis for the success claim; this is the main limitation of the present study. Second, a direct on-platform benchmark against reactive local planners such as DWA and TEB [16] on the same route would quantify the practical advantage of the sampling-based stack. Third, pipeline components can be improved via complementary techniques from our group: Hausdorff-distance-based LiDAR odometry [17] and adaptive Kalman fusion for shot-noise outliers [18]. Finally, augmenting the obstacle layer with short-horizon motion prediction would let MPPI exploit predictable pedestrian trajectories, paving the way for socially-aware navigation that respects human proxemics [19].

References

[1] Z. Wei, S. Wang, K. Chen, and F. Wang, "Ros-based navigation and obstacle avoidance: A study of architectures, methods, and trends," *Sensors*, vol. 25, no. 14, 2025.

[2] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[3] 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.

[4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[5] L. C. Wang, L. S. Yong, and M. H. Ang, "Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment," in *Proceedings of the IEEE International Symposium on Intelligent Control*. IEEE, 2002, pp. 821–826.

[6] 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)*. IEEE, 2016, pp. 1433–1440.

[7] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.

[8] D. V. Lu, A. Merzlyakov, M. Ferguson, S. Macenski, and T. Moore, "From the desks of ros maintainers: A survey of modern capable mobile robotics algorithms in the robot operating system 2," *Robotics and Autonomous Systems*, vol. 168, p. 104493, 2023.

[9] K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 4, pp. 477–496, 2004.

[10] S. Macenski and T. Foote, "SLAM Toolbox: The new robust and extensible laser SLAM system for the Robot Operating System," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 3, pp. 1–14, 2021.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[12] D. Fox, "KLD-sampling: Adaptive particle filters," *Advances in Neural Information Processing Systems*, vol. 14, 2001.

[13] 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*. IEEE, 2014, pp. 709–715.

[14] K. Zheng, "ROS navigation tuning guide," in *Robot Operating System (ROS): The Complete Reference, Volume 6*. Springer, 2021, pp. 197–226.

[15] M. Colledanchise and P. Ögren, *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018.

[16] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *7th German Conference on Robotics*. VDE, 2012, pp. 1–6.

[17] O. Yilmaz and I. Uyanik, "Improving LiDAR odometry with Hausdorff distance-based variance estimation," in *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*. IEEE, 2025, pp. 2477–2482.

[18] E. Erol, M. Dogru, and I. Uyanik, "An adaptive Kalman fusion technique for reference tracking under shot noise," in *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1. IEEE, 2025, pp. 2846–2851.

[19] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, 2023.