Hierarchical Supervisory Motion Planning for Frontal Following Mobile Robots Using Guided Model Predictive Path Integral and Dynamic Virtual Rail

Warnasooriya Mudiyanselage Shakila
Praveen Rathnayake
Faculty of Engineering Technology
The Open University of Sri Lanka
Nawala, Sri Lanka
shakilabeta@gmail.com
ORCID: 0009-0006-7672-5093

***Abstract*— Frontal following, in which a mobile robot moves ahead of a walking person while remaining a navigational follower, suffers from trajectory instability when predicted human positions are treated as navigation goals. The resulting path oscillates in straight corridors, and exit selection becomes unstable at junctions. At decision points, route intent cannot be determined with certainty from physical cues alone. A Hierarchical Supervisory Motion Planning (HSMP) framework is proposed that integrates Dynamic Virtual Rail (DVR)-based path generation, supervisory junction resolution, and Guided Model Predictive Path Integral (Guided-MPPI) trajectory optimization. The robot follows a DVR that derives path geometry from environmental structure rather than predicted human positions, thereby decoupling trajectory planning from human kinematics. Human state is estimated using an Extended Kalman Filter (EKF). Concurrently, supervisory commands dictate exit selection at junctions while the robot executes collision-avoiding paths. Guided-MPPI preserves line of sight during turns through a rear-facing field-of-view cost, and a Kinematic Predictive Controller (KPC) regulates formation distance. Execution velocity is decoupled from trajectory geometry, preventing corner-cutting during deceleration while preserving geometric validity across speed changes. Experiments demonstrate deterministic exit selection, improved trajectory stability with DVR, and reduced corner-cutting during low-speed turns.**

Keywords—frontal following, supervisory control, MPPI, ROS 2, mobile robot following, human-robot interaction, KPC, EKF, formation control, following in front

I. Introduction

Existing work on person-following robots has focused mainly on the robot-behind-human (following from behind) configuration, leaving the forward-leading paradigm (following from the front) underexplored despite its practical relevance in guidance and navigation assistance. Frontal following is useful in applications such as smart luggage carriers, tool carriers, and telepresence avatars. It inverts the control relationship: physically, the robot leads, but navigationally, it follows the human's trajectory intent without knowing the exact destination [1].

Current solutions are either reactive (safe but high user effort) or predictive (low user effort but lacking certainty, with trajectory flickering for users' micro movements) [2]. The root cause is fundamental. Treating predicted human positions as navigation goals creates flickering from side-to-side drift [2]. In corridors, if a human steps left (e.g., to avoid a smudge, to look at a sign, or to talk to someone), a predictive planner steers left; the human corrects right; the planner follows. At junctions, small heading variations cause oscillation between exits [3].

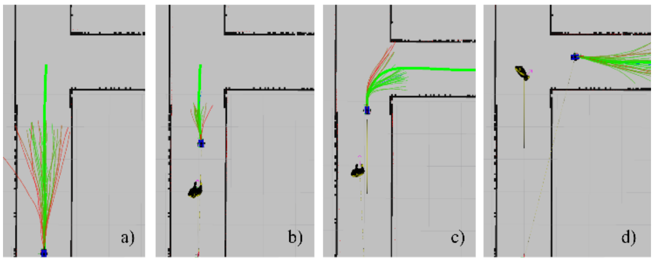

Fig. 1. Gazebo visualization of Guided-MPPI Trajectory Optimization during Frontal Following. (a) Frontal following in a straight corridor. (b) The robot decelerates into a safety wait state due to structural ambiguity at the junction. (c) Upon receiving a supervisory command, the Decision Layer stitches the entry and exit paths; the Guided-MPPI controller (green trajectories) executes the turn along this smoothed rail. (d) The robot clears the junction and realigns with the new corridor.

Frontal following therefore contains three coupled but distinct problems: stable path generation in constrained corridors, reliable intent resolution at topological ambiguities, and geometry-preserving execution during speed changes. Trajectory geometry should be anchored to environmental structure rather than noisy human motion. Meanwhile, route selection at ambiguous junctions should not rely solely on probabilistic inference, because physical cues may not align with the user's intended path and therefore cannot guarantee certainty. Practical indoor operation therefore requires explicit supervisory input for deterministic junction resolution while preserving autonomous trajectory generation and collision avoidance. Yet explicit commands alone are insufficient: once a branch is selected, the robot still requires a geometrically valid path that can be followed smoothly through corridors, turns, and multi-exit junctions.

This paper addresses these requirements through a hierarchical frontal-following framework. The contributions are: (1) Dynamic Virtual Rail (DVR) with command latching, which derives paths from environmental structure and uses a local cubic Bézier trajectory for off-rail recovery; (2) supervisory motion planning, which latches commands issued before or at a junction and stitches entry and exit segments into a deterministic route for Guided Model Predictive Path Integral (Guided-MPPI); and (3) a velocity-geometry decoupling filter that coordinates KPC-constrained execution speed with Guided-MPPI trajectory commands to preserve planned curvature during speed reductions and prevent corner-cutting.

These components separate path generation, junction intent resolution, and motion execution. Environmental structure defines path geometry through the DVR, supervisory input resolves branch intent, and the execution layer follows the rail using Guided-MPPI, while KPC regulates formation distance and the velocity-geometry filter preserves turn curvature. Within the perception layer, a human-tracking EKF estimates human state, but predicted motion does not define the robot's path geometry.

II. Related Work

A. Prediction-Based Person Following

Rear-following, following behind the human, places cognitive load on users because the robot is not within their field of view, requiring them to check back frequently. Frontal following places the robot in the user's field of view, allowing users to interact naturally, but inverts the control problem: the robot must clear a path without knowing its destination.

Current predictive solutions use the Extended Kalman Filter (EKF) [4], human walking model [1], gait analysis [5], [6], Monte Carlo Tree Search (MCTS) with Reinforcement Learning (RL) [7], [8], transformer-based trajectory prediction [9], or neural networks for short-horizon (0.5 s) prediction [10]. However, these methods share fundamental limitations: (1) most approaches couple the robot's trajectory to the human's noisy state; observations show that this produces discontinuous motion [1], [2], [5]; (2) MCTS planners' discrete action sampling can produce inconsistent trajectories [7], [11]; (3) goal-coupled replanning forces continuous path updates as predicted positions shift [2], [3], [9]; and (4) map-conditioned predictors treating structure as probabilistic priors cause jerky motion at junctions [3].

Purely predictive approaches face a prediction ceiling as intent cannot be reasoned from observation alone. With a distance of 1.5 m maintained between human and robot (as used in [2], [7], [9]) and v = 1.0 m/s, the robot arrives approximately 1.5 s before the human. However, body posture cues typically appear only when turning begins or afterward [1], forcing a reactive response despite the expected goal. Jiang confirms "the robot reaches the turn before the human does" [3]. Despite increasing model complexity, from the EKF [1] to dual transformers [9], authors report trajectories "not as smooth" [1] or, in end-to-end learning baselines, "shaking"[2].

B. Supervisory Control and Shared Autonomy

Supervisory control separates intent specification from execution [12]. Users provide high-level commands while the system handles collision avoidance.

C. Sampling-Based Motion Planning

Sampling-based planning explores multiple junction paths, unlike Timed Elastic Band (TEB), which converges to a single optimum. MPPI [13], [14] samples control sequences and evaluates trajectory-level costs over a prediction horizon. Dynamic Window Arc-Line Planning (DWAL)-based front-following [15] is also sampling-based, but it samples arc-line path primitives rather than control sequences. DWAL is useful for local route alternatives, whereas MPPI is better suited for trajectory execution because it optimizes full rollout trajectories with multiple costs over the prediction horizon. To our knowledge, no MPPI variant has been applied to frontal following. Recent variants address guidance via Rapidly-exploring Random Trees (RRT) [16], terrain-aware A* [17], or adaptive sampling [18]. However, no existing MPPI variant decouples velocity from trajectory geometry: spatial lookahead shrinks with velocity ($H_{spatial} \propto v$), causing corner-cutting during deceleration. Additionally, no prior variant includes line-of-sight constraints for tracked humans.

The current literature offers reactive systems, which are safe but demanding, or predictive systems, which are smoother but prone to oscillation. No existing frontal-following framework jointly addresses structural trajectory decoupling, deterministic junction handling, line-of-sight preservation via field-of-view (FOV) cost during turns, and MPPI velocity-geometry decoupling to ensure trajectory validity across speed ranges in constrained indoor environments.

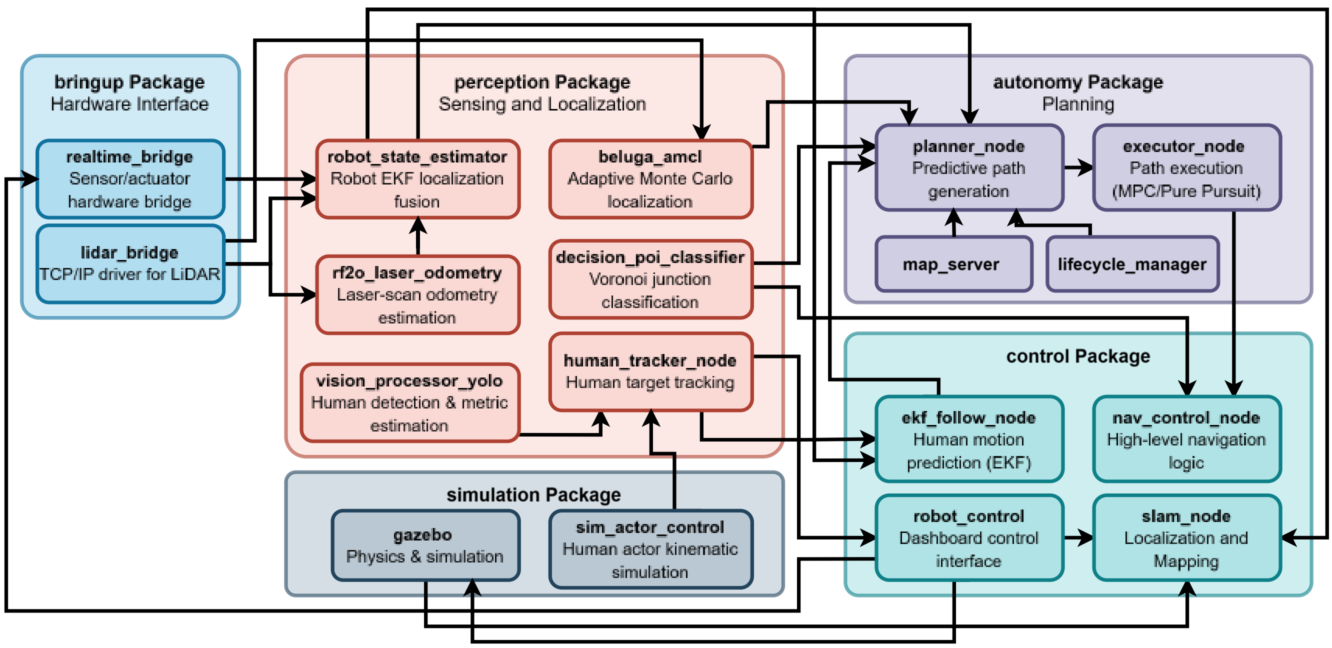

Fig. 2. ROS 2 software architecture showing the Perception, Autonomy, Control, and Simulation packages.

III. System Architecture and Methodology

The software stack runs on Robot Operating System 2 (ROS 2) as a three-layer pipeline: Perception, Decision, and Execution as illustrated in Fig. 3. The software architecture consists of multiple packages and nodes connecting as illustrated in Fig. 2.

A. System Architecture

The system implements a three-layer pipeline (Fig. 3). Perception (EKF-based human tracking, Adaptive Monte Carlo Localization (AMCL), costmap management), Decision (Dynamic Virtual Rail generation, junction resolution), and Execution (Guided-MPPI trajectory optimization, KPC distance regulation). Unlike predictive planners that treat human positions as navigation goals, the framework decouples trajectory geometry from human state by grounding path planning in environmental structure rather than prediction.

The DVR, derived from medial-axis skeletonization, provides the geometric corridor for trajectory optimization. The robot follows this structural path while KPC independently maintains a 1.5 m separation, capped at 2.5 m. MPPI enforces collision avoidance and preserves line of sight via a rear-facing FOV cost.

The system prioritizes "what navigable structure exists ahead?" for path geometry while using "where is the human?" only for speed regulation and visual tracking, eliminating trajectory flickering from prediction-coupled planning.

B. Perception Layer

1) Localization: AMCL provides map-based localization using 2D Light Detection and Ranging (LiDAR). Robot odometry fuses laser scan matching (Range Flow to Odometry (RF2O)), wheel encoders, and Inertial Measurement Unit (IMU) data via an EKF.

2) Costmap: A costmap maintains rolling-window occupancy grids with static, obstacle, and inflation layers.

3) Human State Estimation: An EKF estimates the human state $\boldsymbol{x}_k = [p_x, p_y, v, \psi, \dot{\psi}]^T$ (x-position, y-position, speed, heading, turn rate) in the robot's body-fixed frame $\mathcal{F}_R$ using Constant Turn Rate and Velocity (CTRV) dynamics [4] with ego-motion compensation:

$$p_{k|k-1} = R(-\omega_r \Delta t)\left(p_{CTRV} - \begin{bmatrix} v_r \Delta t \\ 0 \end{bmatrix}\right), \tag{1}$$

$$\psi_{k|k-1} = \psi_{CTRV} - \omega_r \Delta t, \tag{2}$$

where $\mathbf{p}_{CTRV}$ and ψ_{CTRV} are the CTRV-predicted position and heading before ego-motion compensation. $\mathbf{R}$ is the 2D rotation matrix, v_r and ω_r are the robot's linear and angular velocities, and Δt is the time step. Mahalanobis gating rejects measurement outliers; a drift reset is triggered when the error exceeds a threshold.

Because trajectory geometry derives from the DVR rather than from predicted human state, MPPI's structural constraints ensure valid paths regardless of estimation accuracy.

C. Decision Layer

The decision layer generates paths and resolves junctions.

1) Dynamic Virtual Rail: The Dynamic Virtual Rail (DVR) is a path regenerated dynamically from the robot's current location to decouple trajectory planning from human kinematics. By deriving the robot's trajectory from environmental geometry rather than the human's predicted state, the DVR remains stable regardless of tracking accuracy. Smooth transitions between local and global paths use Cubic Bézier segments.

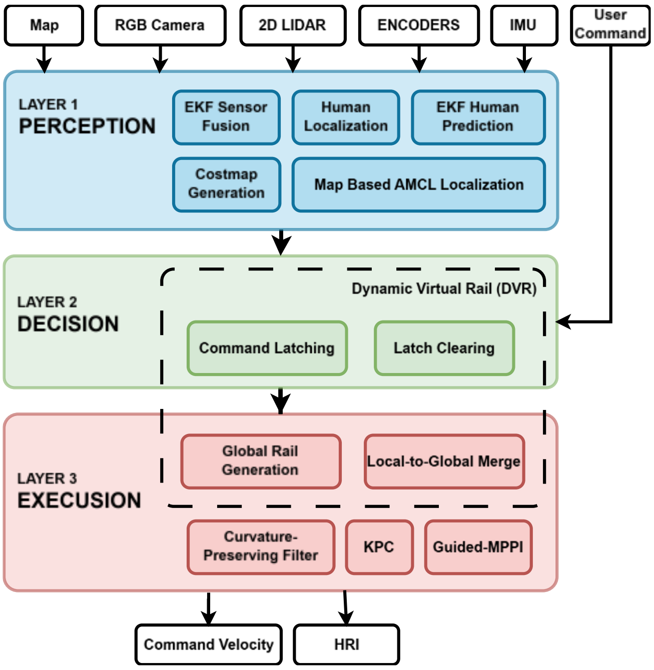

Fig. 3. Hierarchical Supervisory Motion Planning framework showing the Perception, Decision, and Execution layers.

a) Global Rail Generation: The global rail network is precomputed via Voronoi skeletonization; the DVR stitches paths in real time. At junctions, a Junction Buffer ensures C^2-continuity: entry and exit segments are trimmed by $d_{trim} = 0.5$ m and bridged with B-Spline smoothing.

b) Local-to-Global Merge: When displaced from the structural center, a Cubic Bézier ramp provides C^1-continuous transition as shown in Fig. 4(a). Given robot pose (x_r, y_r, ψ_r) and merge target (x_t, y_t) with tangent ψ_t:

$$P(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, \tag{3}$$

where control handles P_1 and P_2 are aligned with ψ_r and ψ_t, respectively, to ensure a smooth merge.

2) Junction Decision Logic: Two mechanisms handle junction ambiguity: Intent Inference selects the path; Latching stabilizes the decision.

a) Intent Inference: At junctions, directional classification (forward/left/right based on relative yaw) selects the exit; user commands latch directly in Supervisory Mode, while Predictive Mode uses velocity-branch alignment.

b) Latching & Clearing: Upon command or high-confidence inference, the decision layer latches the selected exit, ignoring later fluctuations as shown in Fig. 4(b). The latch persists until the robot traverses a clearing distance past the junction, preventing the path-splitting that produces jerky motion at T-junctions [3].

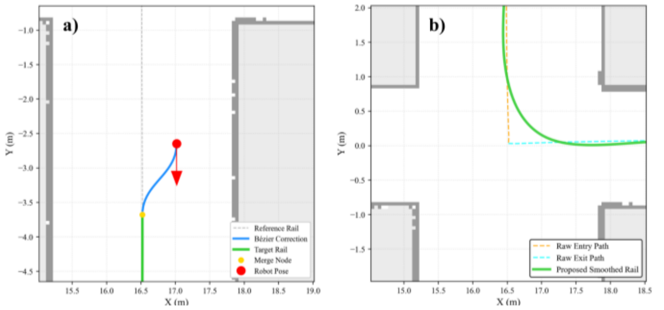

Fig. 4. Dynamic Virtual Rail generation. (a) Robot re-entry trajectory generation using Bézier correction (blue) to merge onto the Target Rail. (b) Generation of the Dynamic Virtual Rail at a T-junction, showing the transition from Raw Entry/Exit paths to a smoothed, navigable corridor.

c) Interaction Modes: Three priority modes: (1) Supervisory Flow latches commands early; (2) Supervisory Safety stops if no command received; (3) Predictive Inference falls back to velocity alignment for baseline comparison.

D. Execution Layer

The execution layer uses a dual-controller architecture with a queue depth of 1 for all high-frequency inputs.

1) Kinematic Predictive Controller (KPC): The KPC models the robot-human link as an elastic tether and regulates the formation distance by predicting gap changes over a short horizon. KPC computes the corrective velocity from the predicted error e_{pred} over horizon. The relative velocity uses the robot's v_{robot} and the human's instantaneous velocity v_{inst} (not EKF-smoothed), providing high reactivity to sudden movements. Given separation d_t:

$$e_{pred} = d_{target} - (d_t + (v_{robot} - v_{inst})\tau), \tag{4}$$

Command velocity v_{cmd} zeros out the maximum of the current error $e_{curr} = d_{target} - d_t$ or predicted error:

$$v_{cmd} = v_{inst} + \frac{\max(e_{pred}, e_{curr})}{\tau}, \tag{5}$$

A deadband suppresses small adjustments to prevent oscillatory hunting. Final velocity is constrained by differential drive kinematic limits, reserving motor capacity for turning:

$$v_{limit} = v_{max} - \left|\omega_{robot} \cdot \frac{L}{2}\right|, \tag{6}$$

$$v_{kpc} = \min(v_{limit}, v_{cmd}), \tag{7}$$

where L is the track width, so KPC never saturates motors during high-speed turns.

Safety Wait State: A maximum separation threshold $d_{wait} = 2.5$ m enforces the lead relationship. If $d_t > d_{wait}$, the robot overrides velocity ($v_{cmd} = 0$) to wait for gap recovery, bounding the operating range to [0,2.5] m.

2) Guided Model Predictive Path Integral Controller: The controller evaluates trajectory samples over a fixed horizon, as shown in Fig. 1, using a hierarchical cost architecture with Graphics Processing Unit (GPU) acceleration [13], [14].

a) Cost Function Architecture: Total cost for each trajectory sample over the horizon N steps:

$$\begin{aligned} J_{total} = \ & J_{heading} + J_{rail} + J_{obs} + J_{fov} \\ & + J_{vel} + J_{smooth} + J_{spin} \end{aligned}, \tag{8}$$

b) FOV Constraint: The FOV cost maintains the human within a rear-facing safety cone:

$$J_{fov} = w_{fov} \cdot \left(\max\left(0, |\theta_{err}| - \delta_{fov}\right)\right)^2, \tag{9}$$

where θ_{err} is the angular error between the robot's rear-facing direction and the human bearing, w_{fov} is the cost weight, and $\delta_{fov} = \pi/4$ rad ($\pm 45°$) defines the cone half-angle.

c) Rail Constraint: The rail cost enforces corridor adherence with a hard boundary:

$$J_{rail} = w_r \cdot \min(d_{rail}, 1) + 10^5 \cdot \max(0, d_{rail} - w_c), \tag{10}$$

where w_r is the rail weight and $w_c = 0.3$ m defines the allowable lateral deviation. Future human position uses constant-velocity lookahead.

Adopting the standard MPPI cost architecture [13], the controller balances trajectory quality using weighted terms: heading (w_θ), obstacle (w_{obs}), velocity (w_{vel}), smoothness (w_{smooth}), and spin (w_{spin}). Comfort costs are discounted ($\gamma = 0.98$); obstacle costs use full weight. Rail following samples trajectories along the global path using arc-length parameterization, mixing RRT-guided [16] with history-based samples. Obstacle avoidance uses soft exponential gradients for smooth optimization.

d) Dynamics Integration: Planned velocity v_{plan} decreases as the path curvature increases. Near path end ($d_{remain} < 2$ m), velocity ramps to zero while active horizon N_{active} reduces to avoid evaluating trajectories past the edge:

$$N_{active} = \mathrm{clamp}\left(\frac{d_{remain} - d_{buffer}}{v_{plan} \cdot dt}, 2, N_{max}\right), \tag{11}$$

where N_{max} is the maximum horizon steps, dt is the MPPI rollout timestep, and d_{buffer} is a safety margin that ensures a halt before the edge. A post-optimization filter triggers emergency stops if a collision is detected.

3) Velocity-Geometry Decoupling Filter: Standard MPPI [14] couples spatial lookahead to execution velocity: rollouts give spatial coverage $H_{spatial} = v \times dt \times N$. When velocity drops (KPC slowing, obstacle proximity), spatial lookahead shrinks proportionally, reducing geometric foresight and causing corner-cutting.

Parameter tuning cannot resolve this coupling. Increasing horizon N is too costly: the controller evaluates $K \times N$ state transitions per cycle (e.g., $2048 \times 20 = 40{,}960$); increasing N requires reducing the sample count K, which reduces optimization quality. Decreasing timestep dt provides finer time-stepping but does not prevent geometry distortion.

This module decouples geometry from execution: MPPI optimizes at v_{plan}; execution velocity is throttled independently:

$$v_{final} = \max\left(0, \min\left(v_{cmd}, v_{plan}, v_{ceil}\right)\right), \tag{12}$$

where v_{ceil} is the maximum velocity from curvature constraints.

Angular velocity scales to preserve curvature $\kappa = \omega / v$:

$$\omega_{final} = \omega_{plan} \times \left(\frac{v_{final}}{v_{plan}}\right), \tag{13}$$

Thus $\kappa_{final} = \omega_{final}/v_{final} = \omega_{plan}/v_{plan} = \kappa_{plan}$, the robot follows the same geometric path at any speed. Slew rate limiting constrains acceleration between cycles, preventing sudden velocity changes that could destabilize the platform.

Safety Governor: The Safety Governor fuses both controllers: MPPI provides trajectory shape (ω_{plan}), KPC regulates execution speed (v_{final}) based on formation distance, and the velocity-geometry decoupling filter scales ω_{plan} proportionally preserving geometry at all speeds.

IV. Experimental Setup

Validation combined Gazebo simulation with real-world experiments on a custom differential-drive platform.

A. Controller Configuration

The Guided-MPPI controller operated at 25 Hz with a 2.0 s prediction horizon ($N = 20$, $dt = 0.1$ s) and $K = 2048$ trajectory samples (70% guided, 30% exploratory).

B. Baseline Configurations

1) Trajectory Stability: An EKF trajectory baseline benchmarked the environment-coupled DVR's robustness against user-noise-induced trajectory flickering, quantified via cross-track error.

2) Junction Handling: An EKF predictive intent baseline evaluated the supervisory command-latching mechanism. Both used identical core architectures (DVR, latching, MPPI), but the baseline used human trajectory predictions rather than supervisory commands to select the path.

3) Velocity-Geometry Decoupling: A standard Guided-MPPI (elastic horizon) baseline compared against the developed system to decouple execution velocity from spatial curvature, preventing lookahead shrinkage and corner-cutting maneuvers during deceleration.

C. Simulation Environment

Experiments were conducted using Gazebo with ROS 2 Humble. The test environment modeled a structured office layout: three straight corridor segments, two 90° corners, and four T-junctions. The controller operated at 25 Hz, exceeding the 20 Hz target; end-to-end latency was approximately 80 ms, with 40 ms ROS pipelining and 23.8 ms MPPI optimization as the dominant components.

D. Real-World Platform

Real-world experiments used a custom differential-drive platform. 2D LiDAR for localization and laser-based odometry, and a Red-Green-Blue (RGB) camera for vision-based human detection. A 3-tier hardware architecture connects a real-time controller with a Real-Time Operating System (RTOS) via a Universal Asynchronous Receiver-Transmitter (UART) to a communication controller, which communicates over Wireless Fidelity (Wi-Fi) with a Personal Computer (PC) via Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Real-world validation focused on T-intersection handling, verifying command latching and turn execution in physical environments where localization drift and actuator friction effects occur.

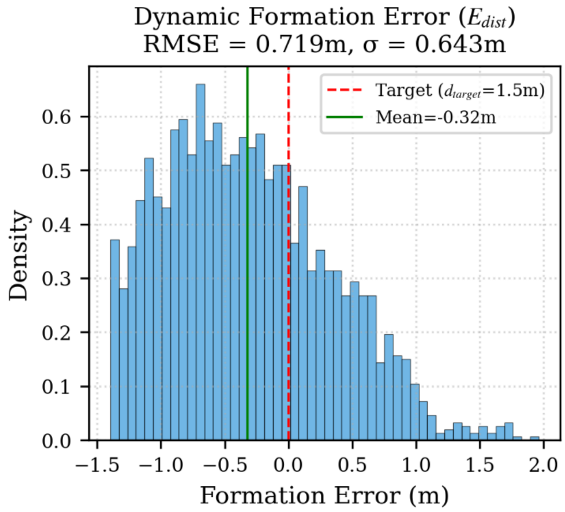

Fig. 5. Histogram of dynamic formation error relative to the 1.5 m target

V. Results and Analysis

A. Distance Regulation

Distance regulation when walking in a straight corridor gave Root Mean Square Error (RMSE) 0.719 m, and a standard deviation (σ) of 0.643 m and the histogram is shown in Fig. 5. A systematic crowding bias (-0.32 m, averaging 1.18 m against the 1.5 m target) emerged during operator acceleration, when the $v_{\max}$ = 1.5 m/s cap and acceleration limits prevent the robot from matching sudden speed increases. Unlike close-proximity walker systems [5], [6] operating at less than 0.5 m, the KPC's elastic tether-bound operation is limited to [0, 2.5] m, preventing collisions while absorbing speed variations. The system prioritizes collision-free operation over precise tracking.

B. Trajectory Stability

The environment-coupled DVR substantially reduced human-motion-induced trajectory variation. By decoupling path geometry from human micro-movements, the DVR outperformed the EKF-based baseline, reducing cross-track RMSE from 0.5329 m to 0.2024 m and the 95th-percentile error from 0.83 m to 0.26 m across 10 trials per configuration. These results indicate improved corridor stability by preventing human-state noise from perturbing trajectory geometry.

C. Junction Handling

The Supervisory Command-Latching system achieved certain path selection across both simulation and real-world trials. The baseline EKF predictive-intent method achieved 65% accuracy in simulation and 40% accuracy in real-world trials. Some predictive models have achieved higher prediction accuracy than the baseline, and comparing them is difficult because the tests are not conducted under the same conditions. And comparing predictive accuracy to supervisory latching through user commands is not a fair comparison, but it's needed to maintain the certainty of frontal following. Supervisory control may appear to increase cognitive load relative to autonomous predictive systems [2], [7]. However, it reduces cognitive load by eliminating timing constraints and reducing recalibration anxiety, hesitation at junctions, and fear of unexpected robot movements. Pure prediction is constrained by timing, distance, and angle [10]. The DVR's latching eliminates this constraint by committing before turning begins. Users move freely: frontal placement ensures constant visibility, and early commands are smoothly latched without slowing down; if no early command is given, the robot slows down to reduce cognitive load.

D. Velocity-Geometry Decoupling

Standard MPPI inherently reduces spatial lookahead during deceleration, leading to severe corner-cutting. In low-speed 90-degree cornering trials (n=10), this yielded a 100% collision rate. The proposed velocity-geometry decoupling filter resolves this by explicitly decoupling geometric curvature from execution velocity. This approach successfully preserved the trajectory shape, maintaining a minimum inner-wall clearance of 0.38 m under the same trial conditions (n=10). Velocity-geometry decoupling preserves turn shape when the robot slows, avoiding the elastic-horizon failure mode of standard MPPI.

VI. Future Work

Extending the framework to unmapped environments requires simultaneous localization and mapping with real-time DVR generation. Dynamic obstacle handling, reducing the MPPI sampling budget for GPU-less onboard deployment, extending curvature preservation to wheel slip and higher speeds, adapting cost-function weights across environments without manual retuning, and a user study to measure cognitive load are also needed. Neural network-based intention capture, advanced sensors, latency reduction, multi-person tracking, occlusion handling, and privacy-preserving perception remain open research problems.

VII. Conclusion

Frontal-following stability improves when trajectory planning places greater weight on environmental structure than on human kinematics. By grounding trajectory generation in environmental geometry rather than human-location predictions, the DVR reduces the oscillations associated with prediction-coupled approaches. The EKF estimates human state within the perception layer, while KPC regulates formation distance and DVR-derived environmental geometry defines the robot's trajectory. Such separation decouples trajectory shape from the noisy predictions that cause path oscillation in corridor following. At ambiguous junctions, prediction alone cannot determine route intent with certainty. Supervisory junction resolution addresses the limitation by allowing user input to determine the branch. In velocity-coupled MPPI operation, reducing execution speed can shorten spatial lookahead, leading to corner-cutting. The solution decouples planning velocity from execution velocity: MPPI generates the planned trajectory independently of the final execution speed to preserve geometric validity, while the velocity-geometry decoupling filter scales angular velocity during execution to preserve planned curvature during execution-speed reductions. Together, DVR-based path generation, supervisory junction resolution, and velocity-geometry decoupling provide smoother corridor motion, deterministic branch selection, and safer low-speed turning.

Acknowledgment

Dr. B. G. D. A. Madhusanka, G. M. K. B. Karunasena, and R. A. K. K. Perera of the Faculty of Engineering Technology, The Open University of Sri Lanka, are acknowledged for their supervision, guidance, and constructive feedback.

References

[1] W. Mi, X. Wang, P. Ren, and C. Hou, "A system for an anticipative front human following robot," in *Proceedings of the International Conference on Artificial Intelligence and Robotics and the International Conference on Automation, Control and Robotics Engineering*, 2016, doi: 10.1145/2952744.2952748.

[2] P. Nikdel, R. Vaughan, and M. Chen, "LBGP: Learning Based Goal Planning for Autonomous Following in Front," in *2021 ICRA*, 2021, pp. 3140–3146. doi: 10.1109/ICRA48506.2021.9560914.

[3] Q. Jiang, B. Susam, J.-J. Chao, and V. Isler, "Map-Aware Human Pose Prediction for Robot Follow-Ahead," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Abu Dhabi, United Arab Emirates: IEEE, Oct. 2024, pp. 13031–13038. doi: 10.1109/IROS58592.2024.10802110.

[4] X. Yuan, F. Lian, and C. Han, "Models and Algorithms for Tracking Target with Coordinated Turn Motion," *Math. Probl. Eng.*, vol. 2014, no. 1, p. 649276, Jan. 2014, doi: 10.1155/2014/649276.

[5] Z. Chongyu, G. Wenzhi, W. Rongwei, Z. Wang, and C. Wu, "Deep Learning-driven Front-Following within Close Proximity: a Hands-Free Control Model on a Smart Walker," in *2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA: IEEE, May 2022, pp. 812–818. doi: 10.1109/ICRA46639.2022.9811910.

[6] C. Zhao, G. Lingyu, R. Wen, Y. Wang, and C. Wu, "Depth-Temporal Attention with Dual Modality Data for Walking Intention Prediction in Close-Proximity Front-Following," in *2025 IEEE ICRA*, Atlanta, GA, USA: IEEE, May 2025, pp. 13920–13926. doi: 10.1109/ICRA55743.2025.11128562.

[7] S. Leisiazar, S. R. R. Rohani, E. J. Park, A. Lim, and M. Chen, "Adapting to Frequent Human Direction Changes in Autonomous Frontal Following Robots," *IEEE Robot. Autom. Lett.*, vol. 10, no. 3, pp. 2934–2941, Mar. 2025, doi: 10.1109/LRA.2025.3537860.

[8] P. Petrov, V. Georgieva, I. Kralov, and S. Nikolov, "An Adaptive Mobile Robot Control for Autonomous Following in Front of a Person," in *2021 12th National Conference with International Participation (ELECTRONICA)*, Sofia, Bulgaria: IEEE, May 2021, pp. 1–4. doi: 10.1109/ELECTRONICA52725.2021.9513709.

[9] M. Mahdavian, P. Nikdel, M. TaherAhmadi, and M. Chen, "STPOTR: Simultaneous Human Trajectory and Pose Prediction Using a Non-Autoregressive Transformer for Robot Following Ahead," 2022, *arXiv*. doi: 10.48550/ARXIV.2209.07600.

[10] A. Wang, Y. Makino, and H. Shinoda, "Real-Time Person-Following Robot: Front-Following Using Human Motion Prediction," in *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Kuching, Malaysia: IEEE, Oct. 2024, pp. 3851–3856. doi: 10.1109/SMC54092.2024.10831013.

[11] S. Leisiazar, E. J. Park, A. Lim, and M. Chen, "An MCTS-DRL Based Obstacle and Occlusion Avoidance Methodology in Robotic Follow-Ahead Applications," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, MI, USA: IEEE, Oct. 2023, pp. 221–228. doi: 10.1109/IROS55552.2023.10342150.

[12] J. R. Peters, V. Srivastava, G. S. Taylor, A. Surana, M. P. Eckstein, and F. Bullo, "Human Supervisory Control of Robotic Teams: Integrating Cognitive Modeling with Engineering Design," *IEEE Control Syst.*, vol. 35, no. 6, pp. 57–80, Dec. 2015, doi: 10.1109/MCS.2015.2471056.

[13] G. Williams, A. Aldrich, and E. A. Theodorou, "Model Predictive Path Integral Control: From Theory to Parallel Computation," *J. Guid. Control Dyn.*, vol. 40, no. 2, pp. 344–357, Feb. 2017, doi: 10.2514/1.G001921.

[14] G. Williams, A. Aldrich, and E. Theodorou, "Model Predictive Path Integral Control using Covariance Variable Importance Sampling," 2015, *arXiv*. doi: 10.48550/ARXIV.1509.01149.

[15] G. Moustris and C. Tzafestas, "A Shared-Control Framework for A Human-Robot Front-Following Behaviour in Unknown Dynamic Environments," *Int. J. Soc. Robot.*, Feb. 2026, doi: 10.1007/s12369-026-01376-0.

[16] C. Tao, H. Kim, and N. Hovakimyan, "RRT Guided Model Predictive Path Integral Method," in *2023 American Control Conference (ACC)*, San Diego, CA, USA: IEEE, May 2023, pp. 776–781. doi: 10.23919/ACC55779.2023.10155837.

[17] J. Yang, M. Kang, S. Lee, and S. Kim, "Hybrid A*-Guided Model Predictive Path Integral Control for Robust Navigation in Rough Terrains," *Mathematics*, vol. 13, no. 5, p. 810, Feb. 2025, doi: 10.3390/math13050810.

[18] D. M. Asmar, R. Senanayake, S. Manuel, and M. J. Kochenderfer, "Model Predictive Optimized Path Integral Strategies," in *2023 IEEE International Conference on Robotics and Automation (ICRA), London*, May 2023, doi: 10.1109/ICRA48891.2023.10160929.