An Integrated Planning and Control Approach to Cooperative Driving with V2V Connectivity

Udit Ekansh
Purdue University
uekansh@purdue.edu

May 10, 2025

Abstract

We present an Integrated Planning and Control (IPC) framework for cooperative driving of car-like robots, extended to support multi-agent coordination via Vehicle-to-Vehicle (V2V) communication. Our work focuses on static but spatially constrained environments where collision avoidance is critical. While the environment lacks dynamic obstacles, we demonstrate that conventional IPC—where each agent independently selects reference points and computes control actions—can lead to deadlocks or inefficient behavior due to uncoordinated reference selection. To address this, we formulate a joint optimization problem that integrates local invariant set computations with inter-agent coordination, enabled through V2V sharing of geometric and positional information. Using two simulated scenarios—a shared-goal deadlock case and a distributed-goal grid navigation task—we show that multi-agent IPC enables safe and efficient convergence where independent control fails, even under purely static constraints.

1 Introduction

Safe multi-robot navigation in cluttered environments is a fundamental challenge in autonomous systems. Traditional approaches typically follow a two-stage paradigm: first, a planning module computes a reference trajectory based on perceived obstacles and map data; then, a low-level controller is used to track the planned trajectory. While effective in simple scenarios, this decoupling often introduces latency, limits reactivity, and fails to guarantee safety under evolving constraints.

Integrated Planning and Control (IPC) offers an alternative: by synthesizing control inputs directly from raw sensor data using feedback control laws, IPC eliminates the need for explicit trajectory generation. IPC frameworks compute admissible control actions by enforcing system dynamics and environmental constraints directly in the control loop, enabling reactive and safe navigation without requiring full trajectory planning.

Despite these advances, most IPC formulations focus on single-agent systems and assume isolated operation. In real-world applications, multiple autonomous agents must navigate shared environments while avoiding not only static obstacles but also each other. Independent application of single-agent IPC in multi-agent settings often leads to interference, inefficient motion, or deadlock—particularly when agents select reference points without accounting for others.

In this work, we extend the IPC framework to multi-agent systems using Vehicle-to-Vehicle (V2V) communication. By sharing local constraints, geometric information, and selected reference points, agents jointly optimize their behavior to ensure safe, coordinated motion. Our contributions are:

- We identify failure modes of decoupled single-agent IPC in static, multi-robot scenarios due to uncoordinated reference selection.
- We formulate a joint optimization problem that incorporates both individual feasibility constraints and inter-agent separation using V2V-shared information.
- We validate our approach in two simulated environments: one where multiple robots compete for a shared goal in a cluttered space, and another involving separate goals with spatial coordination requirements. In both, the multi-agent IPC approach succeeds where independent agents fail.

This study demonstrates that even in the absence of dynamic obstacles, cooperative planning is essential for robust performance in spatially constrained settings. Our findings lay the groundwork for future extensions to dynamic and learning-based environments.

2 Related Work

Conventional navigation pipelines often separate planning and control into two stages, which are executed sequentially during runtime. To improve

reactivity, several methods integrate local geometric information into reactive planning algorithms [1, 2]. Vector field histograms [3] compute safe steering directions based on instantaneous range data, while feedback motion planners such as artificial potential fields (APFs), navigation functions [4], harmonic potential fields (HPFs) [5], and non-holonomic RRT* [6] help maintain stability in the presence of disturbances by replanning from the robot's new position. However, these methods are prone to limitations like local minima and oscillatory behavior, especially under complex obstacle configurations.

To mitigate these issues, approaches like closest-gap navigation (CG) [7], nearness-diagram (ND) [2], and smooth-nearness diagram (SND) [8] extend the planning horizon using local sensory data. These methods prioritize clearance and trajectory smoothness, with CG shown to outperform ND and SND in computational efficiency and reducing oscillations [7]. A taxonomy of such algorithms in [9] highlights that although sensor-based local planners are essential for handling unknown workspaces, they carry inherent trade-offs between responsiveness and formal safety guarantees.

Model Predictive Control (MPC) has emerged as a dominant tool for local navigation, especially for robots with higher-order dynamics and onboard computation [10]. MPC frameworks encode motion, input, and space constraints directly into a finite-horizon optimization problem, making them ideal for path planning [11] and tracking [12]. Variants such as Min–max MPC [13], constraint tightening MPC [14], and Tube MPC [15] extend robustness under bounded disturbances [16, 17, 18]. However, real-time deployment often requires linearization and discretization of nonlinear dynamics, and tuning prediction horizons and weight matrices becomes critical to performance [19].

Integrated planning and control frameworks aim to avoid online trajectory optimization by generating control inputs directly through feedback, making them attractive for resource-constrained systems. Sequential composition techniques [20] have been used to define control policies over domains in the state space such that the goal of one lies in the domain of another. While effective in known, static environments [21, 22], these pre-synthesized control structures do not generalize well to partially known or cluttered settings.

In summary, most existing approaches still depend on precomputed trajectories or heuristic-based control computations. While sensor-based methods like CG, SND, and ND are fast and reactive, they lack formal guarantees. MPC-based strategies offer strong guarantees but require significant onboard resources and careful tuning. Integrated feedback-based approaches provide

an appealing alternative by computing admissible controls without solving online optimizations. However, these methods have largely been confined to static, fully known environments, and their extension to online, sensing-driven navigation remains an open challenge. This work aims to bridge that gap by developing an integrated control structure that uses local sensing to ensure safe, online navigation for unicycle robots, while implicitly handling both motion constraints and the local geometry of the environment.

3 Problem Formulation

The Integrated Planning and Control (IPC) framework provides a feedback-driven approach to safe robot navigation without relying on explicit trajectory planning. In the single-agent setting, IPC uses local range measurements to compute a set of admissible reference points and selects one that moves the agent closer to its goal while remaining within safety constraints. A stabilizing feedback controller then drives the agent toward the selected reference point. This process is repeated iteratively, enabling safe, reactive navigation in cluttered environments.

Formally, the robot is modeled using unicycle dynamics:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega, \tag{1}$$

where (x, y) is the position, θ is the orientation, and (v, ω) are the linear and angular velocity controls, respectively. Range sensor data $\{(d_j, \phi_j)\}_{j=1}^{M}$ defines a locally sensed free space, from which a set of admissible reference points $\mathcal{A}$ is computed. At each step, the reference w^* is selected via a greedy strategy:

$$w^* = \arg\min_{w \in \mathcal{A}} \|w - g\|, \tag{2}$$

where g is the goal position. A feedback controller ensures convergence to w^* within system constraints.

3.1 Multi-Agent Extension with V2V Coordination

While effective in isolation, this formulation does not account for interactions between multiple agents. When several robots operate in proximity, independently chosen references can lead to collisions, gridlock, or inefficient trajectories—even in static environments.

To address this, we extend the IPC framework to a cooperative multi-agent setting. Each agent runs a local IPC loop but also communicates

with neighbors via Vehicle-to-Vehicle (V2V) links to share admissible regions and reference selections. This enables coordinated reference selection that enforces a minimum separation between agents.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent the communication graph, where each node is an agent and edges denote communication links. For N agents, each with local admissible set $\mathcal{A}_i$ and goal g_i, the cooperative reference selection problem is:

$$\min_{w_1,\ldots,w_N} \sum_{i=1}^{N} \|w_i - g_i\| \quad \text{s.t. } w_i \in \mathcal{A}_i, \quad \|w_i - w_j\| \geq \delta_{\text{safe}} \quad \forall (i,j) \in \mathcal{E}. \quad (3)$$

This joint formulation ensures safety through both local sensing and inter-agent coordination. By sharing only local information and constraints, the method remains distributed and scalable while resolving key failure cases of single-agent IPC in shared spaces.

4 Methodology

This section outlines the implementation of the proposed multi-agent IPC framework, from local sensing and reference point selection to coordination via V2V communication and control input synthesis.

4.1 Local Sensing and Admissible Region Computation

Each agent is equipped with a LiDAR-like range sensor that provides M angular measurements $\{(d_j, \phi_j)\}_{j=1}^{M}$ within a limited field of view. These measurements define the local obstacle-free region in which the robot can safely operate at the current time step.

The admissible set $\mathcal{A}_i$ for agent i is constructed by projecting forward-safe points based on LiDAR returns and the robot's footprint. Only those points that maintain clearance from obstacles and satisfy local geometric constraints are retained.

4.2 Reference Point Selection

Given the admissible set $\mathcal{A}_i$ and the goal g_i, the standard IPC strategy chooses a reference point that minimizes the Euclidean distance to the goal:

$$w_i^* = \arg\min_{w \in \mathcal{A}_i} \|w - g_i\|. \quad (4)$$

However, this greedy selection often results in reference conflicts when multiple agents operate in proximity, especially in narrow or congested spaces.

4.3 Multi-Agent Coordination via V2V

To resolve potential conflicts, we use V2V communication to share local information between neighboring agents. Each agent broadcasts its current position, admissible set, and intended reference point.

A joint optimization is then performed over the network to enforce mutual separation:

$$\begin{aligned} \min_{w_1,\ldots,w_N} \quad & \sum_{i=1}^{N} \|w_i - g_i\|, \\ \text{s.t.} \quad & w_i \in \mathcal{A}_i, \\ & \|w_i - w_j\| \geq \delta_{\text{safe}} \quad \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{5}$$

where δ_{safe} is the minimum allowable separation between agents and $\mathcal{E}$ is the set of V2V communication links.

This ensures coordinated reference point selection while keeping computation decentralized and scalable.

4.4 Control Input Computation

Once the reference point w_i^* is finalized, each agent applies a local feedback control law to drive its state toward the reference. We use a proportional stabilizing controller of the form:

$$v_i = k_v \cdot \|w_i^* - x_i\|, \quad \omega_i = k_\omega \cdot \text{atan2}(w_{iy}^* - y_i, w_{ix}^* - x_i) - \theta_i, \tag{6}$$

with tuned gains k_v and k_ω. The agent proceeds toward the reference, and the cycle repeats at the next time step.

The full navigation procedure is summarized in Algorithm 1.

Algorithm 1: Multi-Agent IPC Navigation Algorithm

Input: S_i, g_i, f_s, f_w, f_c ; `// Start, goal, control/planning frequencies`
Output: $U = \{(R_i^k, \psi_i^k, w_i^k)\}$
Initialize $R_i \leftarrow S_i$, $(v_i, \omega_i) \leftarrow (0, 0)$;
$k \leftarrow 1$, $j \leftarrow 1$, $count_{way} \leftarrow 0$, $count_{control} \leftarrow 0$;
while $R_i \neq g_i$ **do**
  Read local range data $\{(d_j, \theta_j)\}$;
  if g_i *in line-of-sight* **then**
    $T_j \leftarrow g_i$;
  else
    Identify candidate region C_j for navigation;
    $T_j \leftarrow$ midpoint of C_j;
  while $R_i \neq T_j$ **do**
    $count_{way} \leftarrow 0$;
    $w_i^k \leftarrow \texttt{WaypointSelection}(R_i, T_j, \{(d_j, \theta_j)\})$;
    Broadcast w_i^k to neighbors, receive $\{w_j^k\}_{j \in \mathcal{N}_i}$;
    Apply collision resolution via joint selection (if needed);
    Transform frame: $(R_i^k, \psi_i^k) \leftarrow$ relative to w_i^k;
    while $R_i \neq w_i^k$ ***or*** $count_{way} \leq \frac{1}{f_w}$ ***and*** $R_i \in \mathcal{L}(V, V(R_i^k))$ **do**
      Compute (v_i, ω_i) from control law;
      Append (R_i^k, ψ_i^k, w_i^k) to U;
      while $count_{control} \leq \frac{1}{f_s}$ **do**
        Deploy control (v_i, ω_i);
        $count_{control} \leftarrow count_{control} + \frac{1}{f_s}$;
      $count_{way} \leftarrow count_{way} + count_{control}$;
      $count_{control} \leftarrow 0$;
    $k \leftarrow k + 1$;
    $j \leftarrow j + 1$;

5 Simulation Setup

We evaluate the proposed framework in two static environments designed to highlight the limitations of independent IPC and the benefits of coordination.

5.1 Sensing and Control Parameters

Each robot uses a front-facing LiDAR with 120 beams over a 180° field of view and a 5-meter range. Control inputs are computed using a proportional stabilizing controller with gains $k_v = 1.0$ and $k_\omega = 2.0$. The control loop runs at 10 Hz, while reference planning and communication occur at 1 Hz.

5.2 Scenarios

Scenario 1: Shared Goal Navigation. Three car-like robots—Robot 1, Robot 2, and Robot 3—start from different locations and navigate toward a common goal. Static circular obstacles are placed to force the robots to approach through narrow corridors. Figure 1 shows the initial configuration. This setup reveals a common failure mode of independent IPC: when each robot greedily selects reference points without accounting for others, they block each other's paths, leading to deadlocks or inefficient paths. The scenario tests whether multi-agent IPC with coordination can resolve these conflicts.

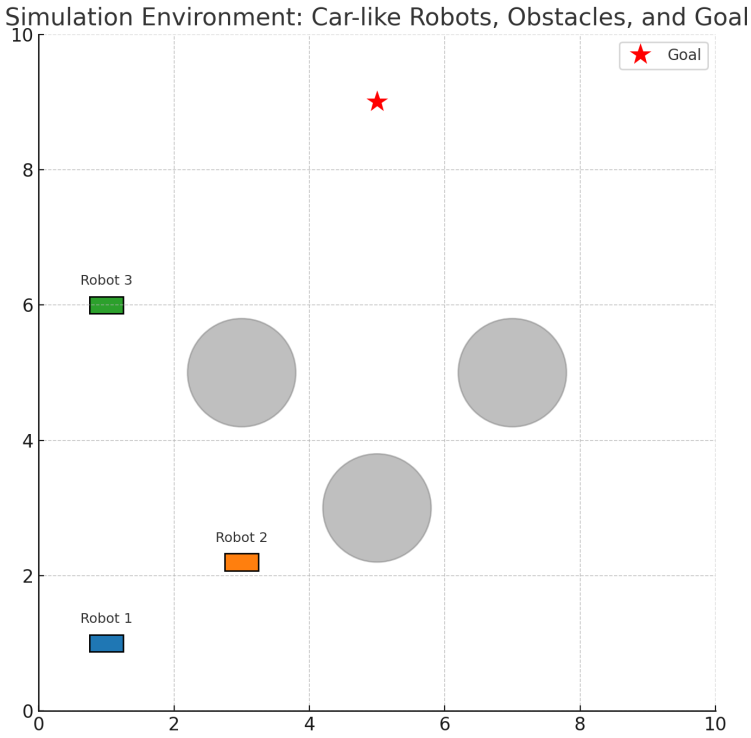


Figure 1: Scenario 1: Three robots navigating toward a shared goal in a cluttered environment. Robot 1, Robot 2, and Robot 3 are shown at their start locations, along with static circular obstacles and a common goal (red star).

Scenario 2: Distributed Parking Navigation. In this more structured environment, each robot is assigned a unique goal in a simulated parking lot. Multiple static rectangular obstacles represent parked vehicles. Figure 2 shows the environment. The layout tests how well independent versus cooperative IPC strategies manage complex spatial interactions when goals differ. Without coordination, agents may converge to overlapping regions or take suboptimal detours. With coordination, they resolve conflicts by sharing intent and adapting waypoints accordingly.

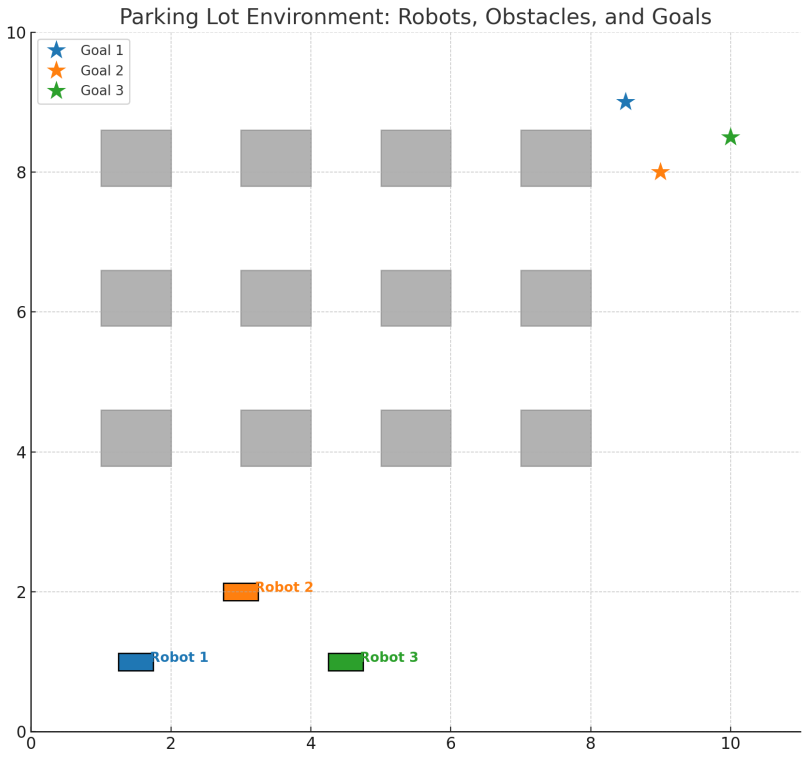


Figure 2: Scenario 2: Each robot navigates to a separate parking goal while avoiding rectangular static obstacles (gray). Robot 1, Robot 2, and Robot 3 are shown at their start locations.

5.3 Implementation

Simulations were implemented in Python using a discrete-time kinematic simulator. In the cooperative case, robots exchange reference and pose information via a simulated V2V link, and conflicts are resolved by enforcing a minimum separation during reference selection. We compare both strategies using trajectory plots, convergence statistics, and qualitative analysis of path efficiency and safety.

To ensure robustness, we tested variations in start positions, goal configurations, and control gains across multiple runs. The results presented in the following sections are representative of typical behavior observed under these variations.

5.4 Evaluation Metrics

To compare the independent IPC baseline with the cooperative V2V approach we track four quantitative metrics:

- **Success Rate.** Fraction of runs in which every robot reaches its goal without any collision. A value of 1 indicates reliable, collision-free convergence for all robots.
- **Time to Goal.** Total simulated time (in seconds) taken for the last robot in the team to reach its goal. Lower values mean faster completion.
- **Path Efficiency.** For each robot we compute the ratio *actual path length* $\div$ *straight-line start–goal distance* and then average over all robots. A perfectly straight trajectory yields 1; higher numbers indicate detours.
- **Minimum Inter-Agent Distance.** The smallest Euclidean distance recorded between any pair of robots during the run. Larger values correspond to safer separation.

6 Results

This section reports the performance of the proposed cooperative IPC strategy and the independent IPC baseline across the two simulation scenarios introduced in Section 5. For each scenario we ran multiple trials that vary

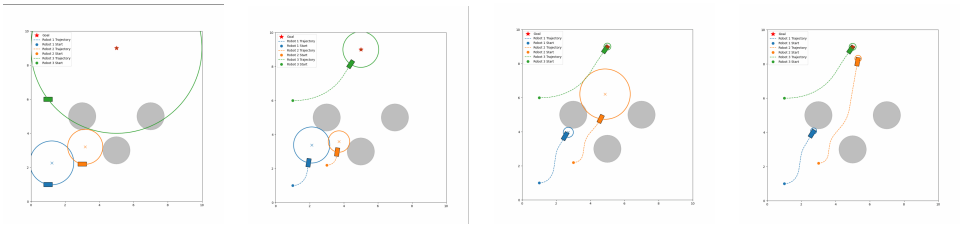
- the initial start positions of Robot 1, Robot 2, and Robot 3,
- the corresponding goal positions,
- the proportional control gains (K_1, K_2, K_3).

Each trial was executed twice: once with agents operating independently and once with V2V coordination enabled. During every run, we log the four evaluation metrics defined in Section 5.4: success rate, time to goal, path efficiency, and minimum distance between agents.

6.1 Qualitative Analysis: Independent IPC Failure

Figure 3 shows a representative failure case in the shared-goal scenario under the independent IPC controller. Robots operate without knowledge of each other's planned motion, leading to inefficient and conflicting trajectories.

In Frame a, Robot 3 (green) selects an aggressive reference heading directly toward the goal. Robot 2 (orange) and Robot 1 (blue) begin to move cautiously. By Frame b, Robot 3 has curved into the corridor, narrowing available space for the other two agents. Frame c shows Robot 3 entering the center path while Robot 2 is forced into a shallow arc and Robot 1 slows down. Finally, in Frame d, the lack of coordination results in an effective deadlock — Robot 3 dominates access to the goal, while Robots 1 and 2 are constrained behind it with no collision-free forward option. The trial is marked as a failure for robot 1 under the success metric.



(a) Greedy arc by Robot 3 (b) Corridor narrowing (c) Conflict emerging (d) Stalling and deadlock

Figure 3: Independent IPC (no V2V): A failure case where Robot 3's path blocks the shared corridor, preventing the others from reaching the goal.

6.2 Quantitative Results

Tables 1 and 2 summarize the performance of the independent IPC baseline and the cooperative V2V variant across two scenario categories: shared-goal navigation and distributed parking.

In the shared-goal setting (Table 1), the cooperative controller achieves an average success rate of 80% compared to just 53% with independent IPC. Agents also converge faster and maintain safer distances from one another. In particular, the lowest minimum distance observed improves from 0.44 m to 0.56 m with V2V, indicating effective resolution of spatial conflicts.

In the parking scenario (Table 2), V2V coordination again leads to substantial gains in both safety and convergence. Average success rate rises from 47% to 93%, and even in cases involving narrow passages and goal

clustering, agents are able to coordinate more effectively to avoid collisions and deadlocks.

These results highlight the value of embedding communication and joint planning within the IPC pipeline. A more detailed discussion of these findings appears in Section 7.

Table 1: Comparison of Independent (IPC) vs Cooperative (V2V) Navigation Strategies — Explicit Scenario Parameters (Shared-Goal Navigation)

Test Case	Start Positions	Goal Position	Gains	Success Rate	Time to Goal (s)	Path Efficiency	Min Distance
Default Configuration	(1.0,1.0); (3.0,2.2); (1.0,6.0)	(5.0,9.0)	1.00/1.50/0.30	0.67 | 1.00	25.30 | 22.50	0.78 | 0.85	0.48 | 0.65
Horizontal Spread	(0.5,0.5); (9.0,0.5); (5.0,0.5)	(5.0,9.0)	1.00/1.50/0.30	1.00 | 1.00	33.80 | 31.20	0.72 | 0.79	0.75 | 0.88
Tight Formation	(1.0,1.0); (3.0,1.0); (5.0,1.0)	(3.0,9.0)	1.00/1.50/0.30	0.33 | 0.67	45.20 | 41.50	0.58 | 0.62	0.31 | 0.42
Increased Gains	(1.0,1.0); (3.0,2.2); (1.0,6.0)	(5.0,9.0)	2.00/3.00/0.60	0.67 | 1.00	20.90 | 18.30	0.81 | 0.88	0.42 | 0.53
Linear Alignment	(1.0,1.0); (3.0,1.0); (5.0,1.0)	(3.0,9.0)	1.00/1.50/0.30	0.00 | 0.33	∞ | 50.20	0.48 | 0.55	0.25 | 0.31
Average	—	—	—	0.53 | 0.80	31.30 | 32.74	0.67 | 0.74	0.44 | 0.56

Table 2: Comparison of Independent (IPC) vs Cooperative (V2V) Navigation Strategies — Explicit Scenario Parameters (Distributed Parking)

Test Case	Start Positions	Goal Position	Gains	Success Rate	Time to Goal (s)	Path Efficiency	Min Distance
Default Parking	(1.5,1.0); (3.0,2.0); (4.5,1.0)	(8.5,9.0); (9.0,8.0); (10.0,8.5)	1.00/1.50/0.30	0.67 | 1.00	50.20 | 48.50	0.65 | 0.72	0.42 | 0.58
Wide Start Positions	(1.0,1.0); (5.0,1.0); (9.0,1.0)	(8.5,9.0); (9.0,8.0); (10.0,8.5)	1.00/1.50/0.30	0.67 | 1.00	52.80 | 51.30	0.62 | 0.68	0.48 | 0.64
Narrow Passages	(1.5,1.0); (3.0,2.0); (4.5,1.0)	(8.5,9.0); (9.0,8.0); (10.0,8.5)	1.00/1.50/0.30	0.33 | 0.67	∞ | 58.60	0.55 | 0.62	0.34 | 0.46
High Control Gains	(1.5,1.0); (3.0,2.0); (4.5,1.0)	(8.5,9.0); (9.0,8.0); (10.0,8.5)	2.00/3.00/0.60	0.33 | 1.00	45.20 | 42.80	0.60 | 0.68	0.32 | 0.52
Clustered Goals	(1.5,1.0); (3.0,2.0); (4.5,1.0)	(9.0,9.0); (9.5,8.5); (8.5,8.5)	1.00/1.50/0.30	0.33 | 1.00	∞ | 52.50	0.58 | 0.65	0.28 | 0.50
Average	—	—	—	0.47 | 0.93	49.40 | 50.74	0.60 | 0.67	0.37 | 0.54

7 Discussion

Figure 4 and Figure 5 present normalized comparisons of the four evaluation metrics—success rate, time to goal, path efficiency, and minimum inter-agent distance—for both independent IPC and cooperative IPC with V2V communication.

In both scenarios, the cooperative V2V approach consistently outperforms its independent counterpart across all metrics. The improvement in success rate is particularly pronounced in the parking scenario, where the independent baseline succeeds in less than half of the runs. This aligns with the nature of the task: parking scenarios often involve tighter spaces, where early coordination and intent sharing become critical to avoiding deadlocks and unsafe proximity.

Across both settings, the cooperative controller not only improves convergence but also enhances safety. The minimum inter-agent distance remains consistently higher in V2V runs, reflecting better spatial awareness and reduced likelihood of blocking or near-collision events. This is most evident in tightly packed start configurations and scenarios with intersecting goals or constrained corridors.

The normalized path efficiency values also favor V2V coordination. Even in successful independent IPC runs, path efficiency often suffers due to unnecessary detours or hesitation caused by local deadlocks. In contrast, V2V allows agents to optimize jointly, reducing wasted motion. The increase in time is also shown to be marginal, and can be attributed to simulation numerical errors.

Overall, these trends confirm the benefit of integrating coordination at the planning level. While independent IPC can perform well in isolated or spacious environments, its limitations become clear under spatial coupling. V2V-based cooperative IPC extends the robustness of the framework to more challenging, high-interaction settings.

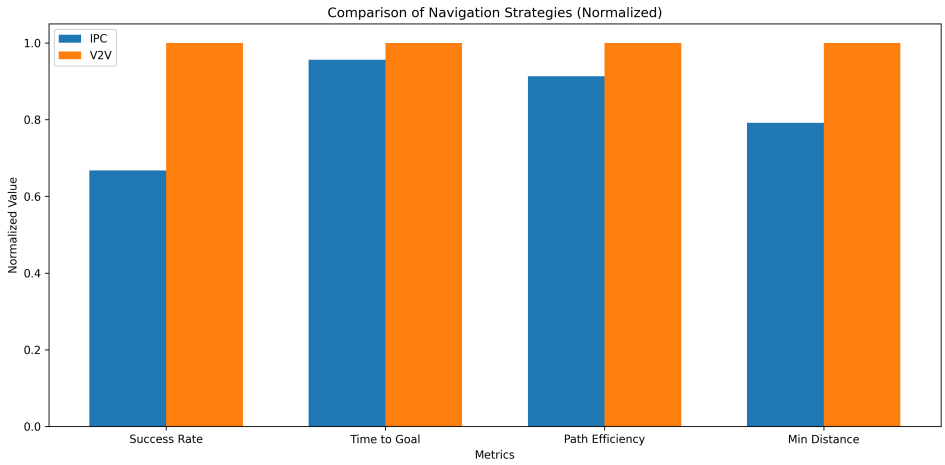


Figure 4: Normalized metric comparison for shared-goal navigation. V2V consistently outperforms IPC across all metrics.

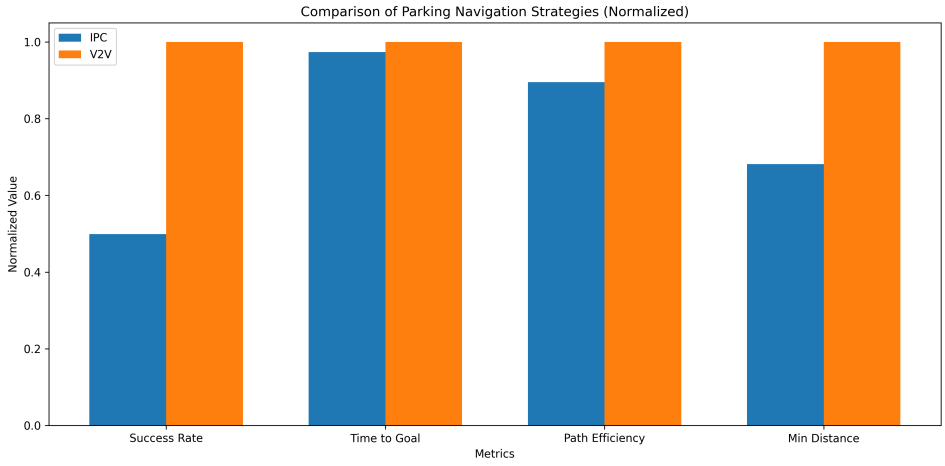


Figure 5: Normalized metric comparison for distributed parking. Largest gains are seen in success rate and minimum distance.

8 Future Work

Several extensions can further strengthen and generalize the findings of this work:

- **High-Fidelity Simulation.** Future experiments could be run in higher-fidelity simulators that more accurately model sensor noise, vehicle dynamics, and communication delay, bringing the system closer to deployment-readiness.
- **V2V Communication Protocols.** The current V2V implementation assumes perfect and instantaneous information exchange. Exploring realistic V2V protocols, including bandwidth constraints, latency, and dropout, would make the results more robust and applicable to real-world deployments.
- **Traffic-Like Constraints.** Adding structured road geometry, lane constraints, and right-of-way rules would allow the framework to better emulate cooperative driving in real traffic environments.
- **Large-Scale Scalability.** Extending the system to larger robot teams would help assess how coordination complexity and communication overhead scale with group size, and whether decentralized coordination heuristics are needed.
- **Mixed Environments.** Finally, future work could study environments with heterogeneous agents, including both coordinated and uncoordinated vehicles, or a mix of static and dynamic obstacles with unpredictable behavior.

9 Conclusion

This work extended the Integrated Planning and Control (IPC) framework to support multi-agent coordination through vehicle-to-vehicle (V2V) communication. We demonstrated, through simulation, that augmenting IPC with joint reference planning and invariant region sharing significantly improves task success rate, path efficiency, and inter-agent safety.

Two representative navigation settings—shared-goal navigation and distributed parking—were used to evaluate the approach. In both cases, cooperative IPC consistently outperformed the independent baseline, especially in tightly constrained or high-interaction environments.

These results highlight the importance of integrated coordination at the planning stage. By embedding V2V into IPC, agents not only avoid collisions but also negotiate complex spatial interactions more effectively, leading to robust and scalable multi-robot behavior.

References

[1] D. J. Gonon, D. Paez-Granados, and A. Billard, "Reactive navigation in crowds for non-holonomic robots with convex bounding shape," in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 4728–4735, 2021.

[2] J. Minguez and L. Montano, "Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 45–59, 2004.

[3] Y. Chen and Y. Lou, "A vector field histogram based fuzzy planner for robot navigation in dense crowds," in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 756–761, 2021.

[4] W. Kowalczyk, "Rapid navigation function control for two-wheeled mobile robots," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 3, pp. 687–697, 2019.

[5] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Prentice Hall of India, 2005.

[6] J. J. Park and B. Kuipers, "Feedback motion planning via non-holonomic RRT* for mobile robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4035–4040, 2015.

[7] M. Mujahad, D. Fischer, B. Mertsching, and H. Jaddu, "Closest gap based (cg) reactive obstacle avoidance navigation for highly cluttered environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1805–1812, 2010.

[8] J. W. Durham and F. Bullo, "Smooth nearness-diagram navigation," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 690–695, 2008.

[9] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[10] M. Jalali, S. Khosravani, A. Khajepour, S. ken Chen, and B. Litkouhi, "Model predictive control of vehicle stability using coordinated active steering and differential brakes," *Mechatronics*, vol. 48, pp. 30–41, 2017.

[11] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 174–179, 2017.

[12] T. Ding, Y. Zhang, G. Ma, Z. Cao, X. Zhao, and B. Tao, "Trajectory tracking of redundantly actuated mobile robot by MPC velocity control under steering strategy constraint," *Mechatronics*, vol. 84, p. 102779, 2022.

[13] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. Fernández Camacho, "Min-max model predictive control of nonlinear systems: A unifying overview on stability," *European Journal of Control*, vol. 15, no. 1, pp. 5–21, 2009.

[14] A. Richards and J. How, "Robust stable model predictive control with constraint tightening," in *2006 American Control Conference*, p. 6, 2006.

[15] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[16] N. T. Nguyen and G. Schildbach, "Tightening polytopic constraint in MPC designs for mobile robot navigation," in *2021 25th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 407–412, 2021.

[17] Z. Sun, L. Dai, K. Liu, Y. Xia, and K. H. Johansson, "Robust MPC for tracking constrained unicycle robots with additive disturbances," *Automatica*, vol. 90, pp. 172–184, 2018.

[18] S. Yu, Y. Guo, L. Meng, T. Qu, and H. Chen, "MPC for path following problems of wheeled mobile robots," in *IFAC-PapersOnLine*, vol. 51, pp. 247–252, 2018. 6th IFAC Conference on Nonlinear Model Predictive Control (NMPC 2018).

[19] M. Obayashi and G. Takano, "Real-time autonomous car motion planning using nmpc with approximated problem considering traffic environment," in *IFAC-PapersOnLine*, vol. 51, pp. 279–286, 2018. 6th IFAC Conference on Nonlinear Model Predictive Control (NMPC 2018).

[20] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.

[21] D. C. Conner, H. Choset, and A. A. Rizzi, "Integrating planning and control for single-bodied wheeled mobile robots," *Autonomous Robots*, vol. 30, no. 3, pp. 243–264, 2011.

[22] U. Nagarajan, G. Kantor, and R. Hollis, "Integrated motion planning and control for graceful balancing mobile robots," *International Journal of Robotics Research*, vol. 32, no. 9–10, pp. 1005–1029, 2013.