LIVEPOINT: Fully Decentralized, Safe, Deadlock-Free Multi-Robot Control in Cluttered Environments with High-Dimensional Inputs

Jeffrey Chen, Rohan Chandra
Code, Videos, Proofs at **livepoint-uva.github.io**

Abstract— **Fully decentralized, safe, and deadlock-free multi-robot navigation in dynamic, cluttered environments is a critical challenge in robotics. Current methods require exact state measurements in order to enforce safety and liveness *e.g.* via control barrier functions (CBFs), which is challenging to achieve directly from onboard sensors like lidars and cameras. This work introduces LIVEPOINT, a decentralized control framework that synthesizes universal CBFs over point clouds to enable safe, deadlock-free real-time multi-robot navigation in dynamic, cluttered environments. Further, LIVEPOINT ensures minimally invasive deadlock avoidance behavior by dynamically adjusting agents' speeds based on a novel symmetric interaction metric. We validate our approach in simulation experiments across highly constrained multi-robot scenarios like doorways and intersections. Results demonstrate that LIVEPOINT achieves zero collisions or deadlocks and a 100% success rate in challenging settings compared to optimization-based baselines such as MPC and ORCA and neural methods such as MPNet, which fail in such environments. Despite prioritizing safety and liveness, LIVEPOINT is 35% smoother than baselines in the doorway environment, and maintains agility in constrained environments while still being safe and deadlock-free.**

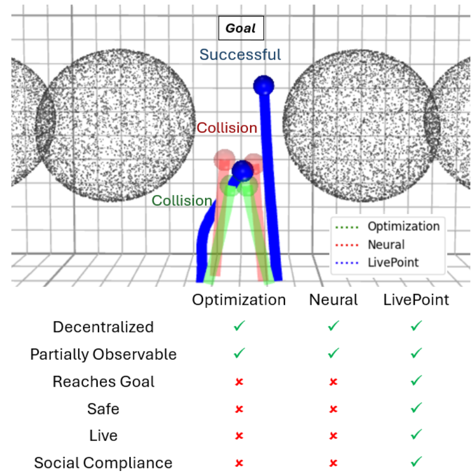


Fig. 1: Our decentralized approach enables effective, safe, live, and socially compliant robot navigation in cluttered environments using only high-dimensional point cloud input.

I. INTRODUCTION

The dream of having robots work with us in our kitchens, construction sites, and hospitals has driven interest in multi-robot navigation among autonomous vehicles, warehouse robots, and personal home robots [1]. Often, these applications feature small, cluttered environments (such as doorways or hallways filled with obstacles). Humans naturally and gracefully navigate these environments, such as by slowing down *by just enough* to let another reach the doorway first [2], [3]. For robots to seamlessly navigate through these environments, they must learn to navigate like humans– safely, gracefully, and without getting stuck (deadlock-free).

Conventional wisdom [4], [5], [6], [7] tells us that in order for robots to achieve human-like mobility in cluttered environments, their low-level controllers need exact and accurate state measurements of their surroundings, which is difficult to realize. Most roboticists, however, would ideally prefer navigation systems that produce human-like trajectories directly using input from onboard sensors, without relying on expensive mapping and perception for exact measurements [8]. For instance, Sa et al. [8] perform point cloud-based single robot navigation to handle dynamic environments, allowing robots to react to rapidly changing obstacles.

However, ensuring both safety and liveness using dense point clouds can be complex [8]. For instance, some analytical methods use control barrier functions (CBFs) [8], [9] to guarantee safety, which requires intensive computations to evaluate the barrier function and its derivatives. Additionally, in decentralized systems, we have no central authority that can coordinate agents in a manner that resolve deadlocks. Learning-based methods [10], [11], [12], [13] can struggle with generalization and lack formal safety guarantees. Moreover, cluttered environments pose additional challenges, particularly due to geometric symmetry in the environment [14]. Robots lack an inherent mechanism to break symmetry, making coordination difficult and leading to deadlocks [15]. Geometric and model-based methods [16], [17], [18] have been explored, but conflicting navigation objectives can still result in deadlock [14]. Reactive collision avoidance techniques attempt to resolve these issues by computing admissible velocity sets [19]. However, in highly constrained environments, robots can overly adjust their velocities to avoid collisions but ultimately stall [20]. Many existing approaches remain too conservative to navigate cluttered environments without deadlock [14].

A. Main Contributions

To address these challenges, we propose a new multi-robot navigation approach that takes as input point clouds and produces in real-time safe and deadlock-free trajectories for fully decentralized robots in cluttered environments. Key to our approach is a novel *universal CBF* formulation that dynamically and simultaneously computes safety and liveness certificates for the controller. The safety component of the universal CBF follows the standard CBF definition commonly employed in the literature [21]. The liveness component is driven by an interaction function that intelligently breaks deadlock-causing symmetry between agents in a minimally invasive fashion, and is cast as a CBF. In summary, our contributions are as follows:

1) **A fully decentralized multi-robot navigation algorithm with point cloud input:** We introduce a sequential control strategy that enables each agent to dynamically adjust its safety constraints based on real-time observations of other agents. Each robot processes point cloud data to construct CBFs while considering the motion of dynamic obstacles (i.e., other agents) in a real time, iterative framework.

Authors are with the Department of Computer Science, University of Virginia. {fyy2ws, aar8xx}@virginia.edu

2) **Synthesizing a universal CBF for safety and liveness:** We introduce a deadlock prevention mechanism that quantifies symmetry between interacting agents, identifying cases where robots are likely to deadlock, particularly in highly constrained spaces like doorways When symmetry is detected, we apply minimal velocity perturbations to proactively break deadlock-prone configurations while maintaining safety guarantees. By integrating the deadlock detection mechanism within the same framework as the CBF constraints, LIVEPOINT unifies CBF-based safety and deadlock avoidance into a single cohesive approach, eliminating the need for separate heuristics or rule-based strategies. Our universal CBF ensures that each robot not only avoids collisions but also maintains continuous progress.

II. Related Work

A. Vision-Based Navigation

Traditional vision-based navigation often relies on simultaneous localization and mapping-based approaches, such as occupancy grid methods [22], to construct global maps for control and planning. While effective in structured environments, these methods are computationally intensive and struggle to adapt in dynamic settings. More recent methods address some limitations by combining planning and learning [23], [24], [25], [26]. While such approaches eliminate the need to do detailed map building, they make non-universal assumptions [26], [23]. CBFs have also been explored as an alternative to enforce safety constraints without full mapping [27], with extensions incorporating learning-based methods [11] and and Neural Radiance Fields-based representations [9]. However, these approaches remain limited by by high computational costs and are generally limited to static environments. Depth-CBF [8] introduced direct point-cloud based safe navigation, offering better computational efficiency and adaptability, but does not account for multi-robot navigation or coordination.

B. Safety and Liveness in Cluttered Environments

Deadlocks commonly arise in multi-agent navigation due to symmetry in the environment or trajectory conflicts [15], [28], [5], [7], [29]. Structured, rule-based strategies such as the right-hand rule [30] or clockwise rotation [15] improve performance, but their reliance on predetermined ordering makes them less generalizable. Other priority-based frameworks for deadlock resolution are inspired by intersection management literature [31], such as first-come, first-served [32]. Such approaches can lead to inefficiencies and congestion, especially when multiple agents arrive simultaneously. To bridge the gaps in point-cloud based navigation and deadlock prevention, we introduce LIVEPOINT, which efficiently uses point cloud data for safe multi-robot navigation, while incorporating liveness-driven robot velocity perturbation mechanisms for deadlock-free multi-agent navigation. This approach retains the computational efficiency and adaptability while addressing the unique coordination challenges that arise in multi-agent settings.

III. PROBLEM FORMULATION

We address the problem of safe and deadlock-free navigation for multiple robots in a cluttered environment, using point clouds. We define the problem, mathematically formalize cluttered environments, and state our overall objective. We first formulate a *general* multi-agent navigation scenario using the following partially observable stochastic game (POSG) [33]: $\left\langle k, X, U^i, \mathcal{T}, \mathcal{J}^i, O^i, \Omega^i \right\rangle$. A superscript of i refers to the i^{th} agent, and a subscript of t refers to discrete time step t. At any given time step t, agent i has state $x_t^i \in X$. The dynamics of the agents are defined by the transition function $\mathcal{T} : X \times U^i \rightarrow X$, and the cost function, $J^i : X \times U^i \rightarrow \mathbb{R}$ is used to evaluate a specified control action for the agent's current state. Each agent i also has an observation $o_t^i \in \Omega^i$ which is determined via the observation function $o_t^i = O^i(x_t^i, P_t^i)$. This observation function takes in the state of the robot, as well as its perceived point cloud P_t^i, as input, and outputs the observations that the robot makes. A discrete trajectory of agent i is defined by $\Gamma^i = \left(x_0^i, x_1^i, ..., x_T^i\right)$, and has a corresponding control input sequence $\Psi^i = \left(u_0^i, ..., u_{T-1}^i\right)$. Agents follow discrete and deterministic control-affine dynamics given by $x_{t+1}^i = f(x_t^i) + g(x_t^i)u_t^i$, where f, g are locally Lipschitz continuous functions. At any time t, each agent i occupies a space given by $C^i(x_t^i) \subseteq X$. A Social Mini-Game (SMG) is a particular type of POSG and formally characterizes a cluttered environment.

Definition 1. *A SMG occurs if for some $\delta > 0$ and integers $a, b \in (0, T)$ with $b - a > \delta$, there exists at least one pair $i, j,\ i \neq j$ such that for all $\Gamma^i \in \tilde{\Gamma}^i$, $\Gamma^j \in \tilde{\Gamma}^j$, we have*

$$C^i(x_t^i) \cap C^j(x_t^j) \neq \emptyset \quad \forall t \in [a, b],$$

where $\tilde{\Gamma}^i$ is the trajectory robot i would take in the absence of any other robots.

The goal of our approach is as follows:

Objective:The optimal trajectory ($\Gamma^{i,*}$) and corresponding optimal sequence of control inputs ($\Psi^{i,*}$) for the i^{th} robot are defined as those that minimize the following cost function:

$$(\Gamma^{i,*}, \Psi^{i,*}) = \arg\min_{\Gamma^i, \Psi^i} \sum_{t=0}^{T-1} \mathcal{J}^i(x_t^i, u_t^i) + \mathcal{J}_T^i(x_T^i) \tag{1a}$$

subject to:

$$C^i(x_t^i) \cap C^j(x_t^j) = \emptyset, \quad \forall i \neq j, \quad \forall t \in [0, T] \tag{1b}$$

$$\|u_t^i\| > 0, \quad \forall i \in \{1, ..., k\}, \quad \forall t \in [0, T-1] \tag{1c}$$

$$x_{t+1}^i = f(x_t^i) + g(x_t^i)u_t^i, \quad \forall t \in [0, T-1] \tag{1d}$$

$$x_T^i \in x_g^i, \quad \forall i \in \{1, ..., k\} \tag{1e}$$

where x_g^i is the goal state region for the i^{th} robot. This optimization framework ensures that each robot computes a trajectory that adheres to safety constraints while avoiding deadlocks and reaching its goal, while minimizing deviation from its preferred trajectory and maintaining smooth control.

IV. BACKGROUND

A. Control Barrier Functions

CBFs [21] are a mathematical framework used to ensure system safety while achieving desired control objectives. A CBF is a scalar function, $h(x)$, defined over the state space of a system. The safe set, $\mathcal{C}$, is defined as the set of all states for which $h(\mathbf{x}) \geq 0$. Ensuring safety involves keeping the system state within $\mathcal{C}$ at all times. This is achieved by designing a control input, $\mathbf{u}$, such that the following condition is satisfied:

$$\dot{h}(x, u) \geq -\alpha(h(x)), \tag{2}$$

where $\dot{h}(x, u)$ is the derivative of $h(x)$, and α is an extended class $\mathcal{K}$ function that ensures the safety constraint is enforced with appropriate robustness. CBFs are integrated into a control optimization framework. Since our implementation operates in discrete time, we reformulate the CBF condition:

$$h(x_{t+1}) - h(x_t) + \alpha(h(x_t))\Delta t \geq 0. \tag{3}$$

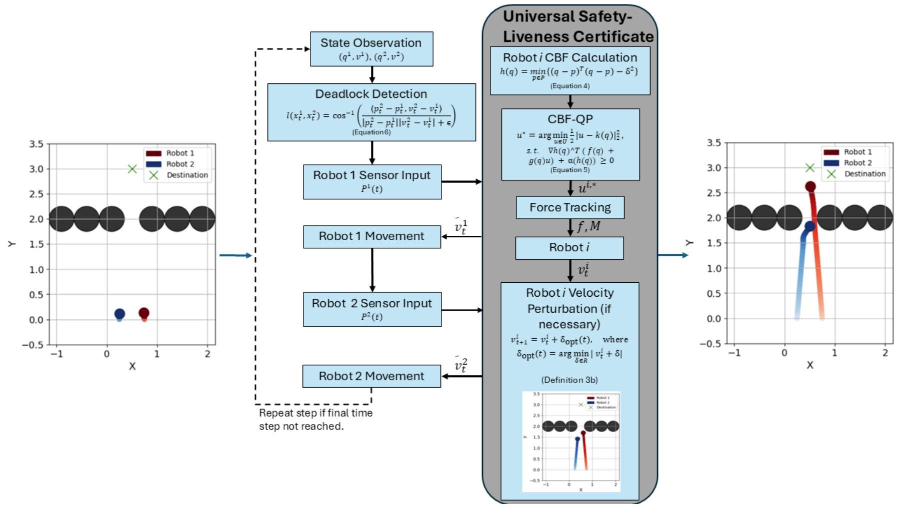


Fig. 2: Technical flowchart illustrating our multi-agent navigation approach. Our universal safety-liveness certificate processes point cloud input to generate robot velocities that ensure deadlock-free navigation. Collectively, the blue elements represent one simulation step.

B. Single Robot Navigation with Point Clouds

The Depth-Based Control Barrier Function (Depth-CBF) framework [8] ensures safety in real-time vision-based navigation by utilizing point cloud data as a direct representation of the robot's environment. The robot's position is denoted as q, and its environment is represented as a point cloud $P^i = \{p_i\}_{i=1}^N$, where $p_i \in \mathbb{R}^2$ is a point obtained from sensors. To define a safety margin, the CBF $h(q)$ is:

$$h(q) = \min_{p \in P}\{\|q - p\|^2 - \delta^2\} \tag{4}$$

where $\delta > 0$ specifies the minimum safe distance. The Depth-CBF is incorporated into a Quadratic Program (QP) to compute safe control inputs while remaining close to a nominal control input $k(q)$. The QP is formulated as:

$$u^* = \arg\min_u \frac{1}{2}\|u - k(q)\|^2 \tag{5}$$

$$\text{subject to } \nabla h(q)^T (f(q) + g(q)u) + \alpha(h(q)) \geq 0,$$

This formulation ensures that the control input u^* minimally deviates from the nominal input $k(q)$ while satisfying the safety constraints imposed by the Depth-CBF.

V. LivePoint: Universal Safety and Liveness Certificates

In this section, we introduce LivePoint for safe and deadlock-free multi-robot navigation using real-time point cloud perception in SMGs. To encourage liveness, we present a deadlock prevention strategy, which quantifies trajectory symmetry and applies minimal velocity adjustments to maintain progress toward goals. We outline the computational framework that integrates safety constraints and deadlock prevention into a single, decentralized control policy.

A. Safety

To ensure safe and collision-free navigation, we employ CBFs to define safety constraints that prevent robots from colliding with static and dynamic obstacles. These CBF constraints are derived solely from high-dimensional point cloud data. At each step, each robot perceives its environment using point cloud data P^i, which allows it to dynamically compute safety constraints at every time step. With the point cloud input, we use Equation 4 to find our CBF. Next, we compute a control input utilizing a modified CBF-QP controller, formulated using Equation 5.

B. Liveness

Our two-part approach to ensuring deadlock-free navigation consists of deadlock detection and resolution.

a) Deadlock Detection: At every time step, before any movement occurs, we check for a potential deadlock using a liveness function, defined as:

$$\ell(x_t^i, x_t^j) = \cos^{-1}\left(\frac{\langle p_t^j - p_t^i, v_t^j - v_t^i\rangle}{|p_t^j - p_t^i||v_t^j - v_t^i| + \epsilon}\right), \tag{6}$$

where $\ell \in [0, \pi]$, and $\epsilon > 0$ ensures a positive denominator. The liveness function gives the degree of symmetry by measuring the angle between the relative displacement and relative velocity of two robots. In cases of near-perfect symmetry, the angle approaches zero. If $\ell_{ij}(x^i, x^j)(t) \leq \ell_{thresh} = 0.3$, a deadlock is detected.[1]

b) Deadlock Prevention: If deadlock was detected, our goal is for robots to decentrally avoid the deadlock by perturbing its speed in a minimally invasive manner, in a way that only changes the speed of the robot and not its planned trajectory or actual positions. In order to inform our approach, we introduce the concept of *liveness sets*.

Definition 2. *At any time t, given a configuration of k robots, $x_t^i \in \mathcal{X}$ for $i \in [1, k]$, a **liveness set** is defined as a union of convex sets, $\mathcal{C}_\ell(t) \subseteq \mathbb{R}^k$ of joint speed $v_t = \left[v_t^1, v_t^2, \ldots, v_t^k\right]^\top$ such that $v_t^i \geq \zeta v_t^j$ for all distinct pairs i, j, $\zeta \geq 2$.*

Liveness sets guarantee that if $v_t \in \mathcal{C}_\ell(t)$, any pair of robots in our configuration will have feasible control inputs that guarantee forward motion. But if $v_t \notin \mathcal{C}_\ell(t)$, then we must perform a minimally invasive velocity perturbation:

Definition 3. *A deadlock-resolving strategy for robot i with current heading angle θ_t^i is minimally invasive if:*

1) *$\Delta\theta_t^i = \theta_{t+1}^i - \theta_t^i = 0$ (The agent does not deviate from the preferred trajectory).*
2) *$v_{t+1}^i = v_t^i + \delta_{opt}(t)$, where $\delta_{opt}(t) = \arg\min_{\delta \in \mathbb{R}} \left\|v_t^i + \delta\right\|$, such that $v_{t+1}^i \in \mathcal{C}_\ell(t)$ (i.e., robot with speed v_{t+1}^i prevents or resolves a deadlock).*

$\delta_{\text{opt}}(t)$ found by solving the following optimization problem:

[1] We formally prove the value of the liveness threshold through two theorems, which can be found at **livepoint-uva.github.io**.

Doorway Scenario									
Method	Collisions ↓	Deadlocks ↓	Success(%) ↑	Time R1 (s) ↓	Time R2 (s) ↓	Vel R1 (u/s) ↑	Vel R2 (u/s) ↑	$\lvert\Delta V\rvert$ R1 (u/s) ↓	$\lvert\Delta V\rvert$ R2 (u/s) ↓
ORCA [34]	50	0	0	N/A	N/A	**1.4178**	**1.419**	**1.971e-02**	1.975e-02
MPC-CBF [35]	50	0	0	N/A	N/A	N/A	N/A	N/A	N/A
MPNet [36]	50	0	0	N/A	N/A	N/A	N/A	N/A	N/A
Single	50	0	0	N/A	N/A	1.248	N/A	2.294e-02	N/A
LIVEPOINT w/o Liveness	50	0	0	N/A	N/A	1.1209	1.0953	2.356e-02	2.279e-02
LIVEPOINT	**0**	**0**	**100**	**9.12**	**11.68**	1.0332	0.3143	2.446e-02	**1.424e-02**

TABLE I: Experiment Results for Doorway Scenarios, averaged over 50 runs. The best results for each category are bolded, with ↑ indicating a higher value is preferable, and ↓ indicating a lower value is preferable. N/A indicates failure to reach the goal.

$$\delta_{\text{opt}}(t) = \arg\min_{\delta\in\mathbb{R}} \|v_t^i + \delta\|, \tag{7a}$$

$$v_{t+1}^i = v_t^i + \delta \tag{7b}$$

$$u_t^i \in \mathscr{U}^i, \;\; u_{t+1}^i \in \mathscr{U}^i \tag{7c}$$

$$\left(x_{t+1}^i, u_{t+1}^i\right) \notin \mathcal{D}^i(t+1) \tag{7d}$$

where deadlock set $\mathcal{D}^i$ is defined as follows:

$$\mathcal{D}^i(t) = \left\{ \left(x_t^i, u_t^i\right) : x_t^i \notin \mathcal{X}_g, \;\; u_t^i = 0 \text{ for threshold} > 0 \right\} \tag{8}$$

If each $v_t^i \in \mathscr{C}_\ell(t)$, then there is no deadlock. If, however, $v_t \notin \mathscr{C}_\ell(t)$, then robot i will adjust v_t^i such that v_t is projected onto the nearest point in $\mathscr{C}_\ell(t)$. We establish that our approach ensures both safety and liveness guarantees. The use of CBFs inherently enforces safety by ensuring that the system remains within a certified safe set. To guarantee liveness, we demonstrate in supplementary material that a feasible velocity perturbation-based solution always exists and is unique under reasonable assumptions.

C. Overall Algorithm

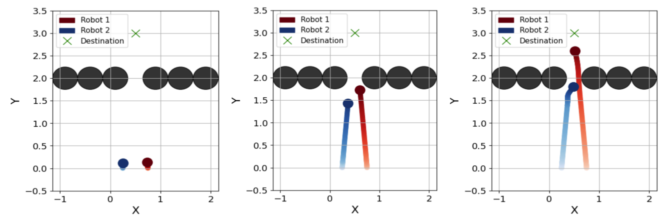
Our sequential algorithm is formulated as follows and illustrated in Figure 2. First, each robot observes its state. The liveness function (Equation 6) is then computed to determine whether the robots are at risk of a deadlock. Next, Robot 1 perceives its point cloud, computes its CBF constraints, and computes its control input and corresponding desired velocity. If a potential deadlock was detected earlier, the deadlock resolution procedure is executed as per Equation 7a. Robot 1 then updates its state. Next, Robot 2 follows the same process as Robot 1. This process repeats iteratively until both robots have reached their destinations, a collision occurs, or maximum time step T is reached.

VI. EXPERIMENTS & RESULTS

We evaluate LIVEPOINT in a Python-based simulation environment [37], simulating two robots in a constrained doorway scenario. We conduct the following comparisons:

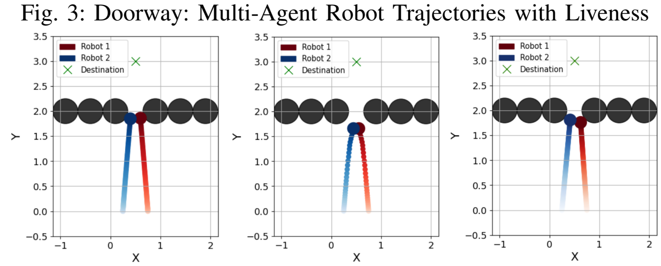
1) **Depth-CBF (Single-Agent)** [8]: We utilize Depth-CBF to control a single robot while treating the second robot as a dynamic obstacle with predefined positions.
2) **LIVEPOINT without Liveness (LoL)**: This ablation baseline eliminates our deadlock prevention algorithm.
3) **MPC-CBF** [35]: Combines Model Predictive Control with CBFs for collision avoidance, formulating an optimization problem over a finite horizon.
4) **MPNet** [36]: Learning-based motion planner which uses neural networks to generate paths directly from raw point cloud data.
5) **ORCA** [34]: Utilizes safe velocity sets to generate collision-free motion for multiple agents.

We perform 50 runs and measure key performance metrics, summarized in Table I. We track the number of collisions or deadlocks, the percentage of successful runs, the time to destination, the average velocity, and the magnitude of velocity changes ($|\Delta V|$) while navigating through the doorway.



(a) Robots moving towards the opening. (b) Robot 1 proceeds first. (c) Robot 2 follows 1 through the opening.

Fig. 3: Doorway: Multi-Agent Robot Trajectories with Liveness



(a) ORCA [34]. (b) MPC-CBF [35]. (c) MPNet [36].

Fig. 4: Baselines: Multi-Agent Robot Trajectories for ORCA, MPC-CBF, and MPNet in doorway scenario. Collisions occur for all three.

Figure 3 illustrates the robot trajectories under LIVEPOINT. Potential deadlocks are detected early, prompting Robot 2 to reduce its speed and allow Robot 1 to pass through the doorway first. Once Robot 1 clears the doorway, Robot 2 takes its turn through the doorway. On average, LIVEPOINT achieves zero collisions and a 100% success rate across 50 runs, significantly outperforming all baselines and ablations. While LIVEPOINT is not the most agile, an expected trade-off given its prioritization of safe and deadlock-free navigation, its reduction in speed is minimal when compared to other methods. Notably, LIVEPOINT achieves the smoothest navigation, demonstrating its ability to balance safety and liveness while maintaining smooth, natural robot movement.[2]

In our doorway scenario, ORCA, MPC-CBF, and MPNet lead to collisions, as seen in Figure 4. The results highlight the limitations of single-agent Depth-CBF, LoL, and other state-of-the-art baselines in constrained multi-agent environments, while LIVEPOINT overcomes these challenges with its novel Universal Safety-Liveness Certificate. By ensuring smooth, safe, and deadlock-free navigation, LIVEPOINT demonstrates scalability and robustness for real-world multi-agent coordination in dynamic, constrained environments.

VII. CONCLUSION

In this work, we introduced a multi-agent navigation framework that synthesizes Control Barrier Functions over point cloud data for safe and deadlock-free navigation in cluttered environments. The proposed method incorporates minimal velocity perturbations to resolve potential deadlocks. The results show that our approach successfully mitigates collisions and deadlocks while maintaining smooth trajectories for multiple agents in complex environments.

[2] Additional visualizations and results for a second evaluation scenario (Intersection), can be found at **livepoint-uva.github.io**.

REFERENCES

[1] N. Sharma, J. K. Pandey, and S. Mondal, "A review of mobile robots: Applications and future prospect," *International Journal of Precision Engineering and Manufacturing*, vol. 24, no. 9, pp. 1695–1706, 2023.

[2] A. H. Raj, Z. Hu, H. Karnan, R. Chandra, A. Payandeh, L. Mao, P. Stone, J. Biswas, and X. Xiao, "Rethinking social robot navigation: Leveraging the best of two worlds," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 330–16 337.

[3] R. Chandra, H. Karnan, N. Mehr, P. Stone, and J. Biswas, "Towards imitation learning in real world unstructured social mini-games in pedestrian crowds," *arXiv preprint arXiv:2405.16439*, 2024.

[4] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal, "Learning to jump from pixels," 2021. [Online]. Available: https://arxiv.org/abs/2110.15344

[5] R. Chandra, V. Zinage, E. Bakolas, P. Stone, and J. Biswas, "Deadlock-free, safe, and decentralized multi-robot navigation in social mini-games via discrete-time control barrier functions," 2024.

[6] S. Gouru, S. Lakkoju, and R. Chandra, "Livenet: Robust, minimally invasive multi-robot control for safe and live navigation in constrained environments," 2024. [Online]. Available: https://arxiv.org/abs/2412.04659

[7] V. Zinage, A. Jha, R. Chandra, and E. Bakolas, "Decentralized safe and scalable multi-agent control under limited actuation," 2024. [Online]. Available: https://arxiv.org/abs/2409.09573

[8] M. De Sa, P. Kotaru, and K. Sreenath, "Point cloud-based control barrier function regression for safe and efficient vision-based control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, 2024, pp. 366–372.

[9] M. Tong, C. Dawson, and C. Fan, "Enforcing safety for vision-based controllers via control barrier functions and neural radiance fields," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 511–10 517.

[10] K. Lobos-Tsunekawa and T. Harada, "Point cloud based reinforcement learning for sim-to-real and partial observability in visual navigation," 2020. [Online]. Available: https://arxiv.org/abs/2007.13715

[11] R. K. Cosner, I. D. J. Rodriguez, T. G. Molnar, W. Ubellacker, Y. Yue, A. D. Ames, and K. L. Bouman, "Self-supervised online learning for safety-critical control using stereo vision," in *International Conference on Robotics and Automation (ICRA)*, 2022, pp. 11 487–11 493.

[12] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan, "Learning safe, generalizable perception-based hybrid control with certificates," 2022. [Online]. Available: https://arxiv.org/abs/2201.00932

[13] W. Xiao, T.-H. Wang, M. Chahine, A. Amini, R. Hasani, and D. Rus, "Differentiable control barrier functions for vision-based end-to-end autonomous driving," 2022. [Online]. Available: https://arxiv.org/abs/2203.02401

[14] S. H. Arul, J. J. Park, and D. Manocha, "Ds-mpepc: Safe and deadlock-avoiding robot navigation in cluttered dynamic scenes," 2023. [Online]. Available: https://arxiv.org/abs/2303.10133

[15] J. Grover, C. Liu, and K. Sycara, "The before, during, and after of multi-robot deadlock," *The International Journal of Robotics Research*, p. 02783649221074718, 2016.

[16] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 01 2017.

[17] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, pp. 760 – 772, 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:9073894

[18] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 1928–1935.

[19] J. van den Berg, S. Guy, M. Lin, and D. Manocha, *Reciprocal n-Body Collision Avoidance*, 04 2011, vol. 70, pp. 3–19.

[20] S. Dergachev and K. Yakovlev, "Distributed multi-agent navigation based on reciprocal collision avoidance and locally confined multi-agent path finding," 2021. [Online]. Available: https://arxiv.org/abs/2107.00246

[21] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[22] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, June 1989.

[23] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," 2020. [Online]. Available: https://arxiv.org/abs/2004.05155

[24] A. Faust, O. Ramirez, M. Fiser, K. Oslund, A. Francis, J. Davidson, and L. Tapia, "Prm-rl: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," 2018. [Online]. Available: https://arxiv.org/abs/1710.03937

[25] H.-T. L. Chiang, A. Faust, M. Fiser, and A. Francis, "Learning navigation behaviors end-to-end with autorl," 2019. [Online]. Available: https://arxiv.org/abs/1809.10124

[26] X. Meng, N. Ratliff, Y. Xiang, and D. Fox, "Scaling local control to large-scale topological navigation," 2020. [Online]. Available: https://arxiv.org/abs/1909.12329

[27] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[28] J. Grover, C. Liu, and K. Sycara, "Why does symmetry cause deadlocks?" *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9746–9753, 2020.

[29] S. Gouru, S. Lakkoju, and R. Chandra, "Livenet: Robust, minimally invasive multi-robot control for safe and live navigation in constrained environments," *arXiv preprint arXiv:2412.04659*, 2024.

[30] Y. Chen, M. Guo, and Z. Li, "Recursive feasibility and deadlock resolution in mpc-based multi-robot trajectory generation," *arXiv preprint*, vol. arXiv:2202.06071, 2022.

[31] Z. Zhong, M. Nejad, and E. E. Lee, "Autonomous and semiautonomous intersection management: A survey," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 2, pp. 53–70, 2020.

[32] T.-C. Au, S. Zhang, and P. Stone, "Autonomous intersection management for semi-autonomous vehicles," in *The Routledge Handbook of Transportation*. Routledge, 2015, pp. 88–104.

[33] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 4, 2004, pp. 709–715.

[34] S. Dergachev, K. Yakovlev, and R. Prakapovich, "A combination of theta*, orca and push and rotate for multi-agent navigation," 2020. [Online]. Available: https://arxiv.org/abs/2008.01227

[35] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," 2021. [Online]. Available: https://arxiv.org/abs/2007.11718

[36] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," 2020. [Online]. Available: https://arxiv.org/abs/1907.06013

[37] "GitHub - HybridRobotics/depth_cbf: Point Cloud-Based Control Barrier Function Regression for Safe and Efficient Vision-Based Control — github.com," https://github.com/HybridRobotics/depth_cbf, [Accessed 26-02-2025].