Deadlock-free, safe, and decentralized multi-robot navigation in social mini-games via discrete-time control barrier functions

Rohan Chandra[1] · Vrushabh Zinage[2] · Efstathios Bakolas[2] · Peter Stone[3,4] · Joydeep Biswas[3]

Abstract

We present an approach to ensure safe and deadlock-free navigation for decentralized multi-robot systems operating in constrained environments, including doorways and intersections. Although many solutions have been proposed that ensure safety and resolve deadlocks, optimally preventing deadlocks in a minimally invasive and decentralized fashion remains an open problem. We first formalize the objective as a non-cooperative, non-communicative, partially observable multi-robot navigation problem in constrained spaces with multiple conflicting agents, which we term as social mini-games. Formally, we solve a discrete-time optimal receding horizon control problem leveraging control barrier functions for safe long-horizon planning. Our approach to ensuring liveness rests on the insight that *there exists barrier certificates that allow each robot to preemptively perturb their state in a minimally-invasive fashion onto liveness sets i.e. states where robots are deadlock-free*. We evaluate our approach in simulation as well on physical robots using F1/10 robots, a Clearpath Jackal, as well as a Boston Dynamics Spot in a doorway, hallway, and corridor intersection scenario. Compared to both fully decentralized and centralized approaches with and without deadlock resolution capabilities, we demonstrate that our approach results in safer, more efficient, and smoother navigation, based on a comprehensive set of metrics including success rate, collision rate, stop time, change in velocity, path deviation, time-to-goal, and flow rate.

Keywords Social navigation · Control barrier functions · Deadlock prevention · Multi-robot systems · Multi-agent systems

✉ Rohan Chandra
  rohanchandra@virginia.edu

  Vrushabh Zinage
  vrushabh.zinage@utexas.edu

  Efstathios Bakolas
  bakolas@austin.utexas.edu

  Peter Stone
  pstone@utexas.edu

  Joydeep Biswas
  joydeepb@utexas.edu

[1] Department of Computer Science, University of Virginia, Charlottesville, USA

[2] Department of Aerospace Engineering & Engg. Mechanics, The University of Texas at Austin, Austin, USA

[3] Department of Computer Science, The University of Texas at Austin, Austin, USA

[4] Sony AI, New York, USA

1 Introduction

We consider the task of multi-robot navigation in constrained environments such as passing through narrow doors and hallways, or negotiating right of way at corridor intersections. We refer to these types of scenarios as *social mini-games*. Unlike humans, robots often collide or end up in a deadlock due to several challenges arising in social mini-games. Two key challenges, in particular, demand attention. First, without some form of cooperation, decentralized systems, even with perfect local sensing, result in deadlocks, collisions, or non-smooth trajectories. Second, humans are adept at avoiding collisions and deadlocks without having to deviate too much from their preferred walking speed or trajectory. For instance, when two individuals go through a doorway together, one person modulates their velocity by just enough to enable the other person to pass through first, while still adhering closely to their preferred speed. This type of behavior presents a significant challenge for robots, which struggle to emulate such socially adaptive maneuvers while maintaining a consistent preferred velocity. The objective of this paper is to propose

Table 1 Comparing various approaches for multi-robot navigation in social mini-games

Approach	Necessary Algorithmic Conditions			Assumptions		
	Collision-free	Liveness[†]	Dynamics	Control	Non[§] Cooperative	Observation
DRL (Long et al., 2018; Chen et al., 2017; Everett et al., 2021, 2018)	✗	✗	Differential drive	CTDE	✗	Partial
ORCA-MAPF (Dergachev & Yakovlev, 2021)	✗	✓	Single-integrator	Centralized	✗	Full
Prediction + Planning (Kamenev et al., 2022)	✗	✗	–	CTDE	✗	Full
Game-theoretic Dist. Opt. (Le Cleac'h et al., 2022)	✓	✗	Unicycle	Distributed	✓	Full
NH-TTC (Davis et al., 2019)	✓	✗	Differential drive	Decentralized	✗	Full
Auction-based (Chandra et al., 2023)	✗	✓*	Ackermann	Distributed	✓	Partial
Buffered Voronoi Cells (Chen et al., 2022)	✗	✓*	Double-integrator	Distributed	✓	Full
NH-ORCA (Alonso-Mora et al., 2013)	✓	✗	Differential drive	Decentralized	✓	Partial
CBFs (Wang et al., 2017)	✓	✗	Double-integrator	Decentralized	✓	Partial
CBFs+KKT (Grover et al., 2016)	✓	✓*	Single-integrator	Decentralized	✓	Partial
DS-MPEPC (Arul et al., 2023)	✓	✓*	Differential drive	Decentralized	✓	Partial
RLSS (Şenbaşlar et al., 2023)	✓	✗	Differentially flat	Decentralized	✗	Partial
Humans	✓	✓	–	Decentralized	✓	Partial
This paper	✓	✓	**Double-integrator**	**Decentralized**	✓	Partial

★ Under certain conditions

§ Non-cooperative robots: robots optimize their individual objective functions (Yang & Wang, 2020)

† Deadlock-free

a safe and deadlock-free navigation algorithm for multiple robots in social mini-games.

The goal is for robots to navigate in such social mini-games as humans do as much as possible. More formally, we outline a list of *necessary* conditions along with some assumptions of the multi-robot navigation problem that can shift the nature of the problem toward more or less human-like behavior:

- Necessary conditions:
  1. *Ensure collision-free controls:* Navigation algorithms must produce controls that guarantee robots traverse their environment without collisions.
  2. *Demonstrate liveness:* Navigation algorithms must detect and prevent deadlocks in a minimally invasive fashion.
  3. *Obey kinodynamic constraints:* Humans operate under physical and dynamic constraints. Navigation algorithms must both operate under complex kinody-

namic constraints of the robot such as speed and acceleration limits, as well as be deployable on real robots.

- Navigation algorithms can make several assumptions about the robot's behavior. We enumerate the possible assumptions as follows:
  1. *Control Operation:* Robots can operate in a centralized, decentralized, or distributed manner, which can affect the level of coordination required for successful navigation.
  2. *Non-cooperative agents:* Robots can be non-cooperative where each robot optimizes its own objective function or cooperative where each robot optimizes a global system objective.
  3. *Observability:* The level of observability can vary for robots, with some agents relying on partial or full observability of their surroundings, impacting their ability to make decisions and navigate successfully.

We compare different classes of methods according to these desiderata and assumptions in Table 1. Although multi-robot navigation encompasses a vast range of algorithms, we narrow our focus to algorithms that have been applied to social mini-games either on real robots or in simulation. These include methods based on deep reinforcement learning (Long et al., 2018; Chen et al., 2017; Everett et al., 2021, 2018), multi-agent path finding (Dergachev & Yakovlev, 2021), trajectory prediction (Kamenev et al., 2022), game-theoretic distributed optimization (Le Cleac'h et al., 2022), auctions (Chandra et al., 2022), geometric planning (Alonso-Mora et al., 2013; Van Den Berg et al., 2011; Dergachev & Yakovlev, 2021), and other optimization-based methods (Grover et al., 2016; Arul et al., 2023; Wang et al., 2017; Chen et al., 2022; Davis et al., 2019). From Table 1, we note that none of these methods satisfy all of the necessary conditions for optimal multi-robot navigation in social mini-games. This paper addresses the following open research question:

> **Research Question:** *how can we design an algorithm that can satisfy all of the necessary specifications for optimal multi-robot navigation in social mini-games?*

Main Contributions: We present the first approach for optimal, safe, and (preventive) deadlock-free receding-horizon navigation for robots with double-integrator dynamics in social mini-games, such as navigating through a narrow door or negotiating right of way at a corridor intersection. Our algorithm is minimally invasive, fully decentralized[1], and works in realtime both in simulation as well as on physical robots. Our main contributions include:

1. We formally define social navigation in geometrically constrained environments through the notion of social mini-games. Prior research (Grover et al., 2020; Chen et al., 2022) has only identified social mini-games as *causes* of deadlocks, which they can then only resolve once the deadlock occurs. In this work, we leverage the geometrical properties of social mini-games to detect and *prevent* deadlocks from occurring in an online fashion.
2. We present a new class of multi-agent, realtime, decentralized controllers that perform safe and deadlock-free navigation via a discrete-time, finite-horizon MPC formulation. Safety is modeled via control barrier functions and liveness is ensured by a geometrically motivated detection-and-prevention strategy. Both safety and live-

ness are guaranteed in realtime by integrating them as constraints to the optimal control problem.

3. We evaluate our proposed navigation algorithm in simulation as well in the real world using F1/10 robots, a Clearpath Jackal, and a Boston Dynamics Spot in a doorway, corridor intersection, and hallway scenario. We show that our approach results in safer and more efficient navigation compared to local planners based on geometrical constraints, optimization, multi-agent reinforcement learning, and auctions. Specifically, we show that our deadlock resolution is the smoothest in terms of smallest average change in velocity as well as path deviation. We demonstrate efficiency by measuring makespan and showing that the throughput generated by robots using our approach to navigate through these constrained spaces is identical to when humans navigate these environments naturally.

In the remainder of this paper, we discuss related work in Sect. 2 and formulate the problem in Sect. 3. We present our controller in Sect. 5 and evaluate its performance in Sect. 6. We conclude the paper in Sect. 7.

2 Related work

In this section, we review the existing approaches for multi-robot navigation in social mini-games. We categorize the approaches based on their mode of operation which could be decentralized, centralized, distributed, or a special class pertaining specifically to learning-based methods that rely on centralized training and decentralized execution (CTDE). Table 1 summarizes the comparison of these approaches based on mode of operation along with safety, liveness, real-world deployment, decentralized decision-making, self-interested agents, and observation.

2.1 Collision avoidance

Provable safety can be achieved by single-integrator systems e.g. ORCA framework from Van Den Berg et al. (2011) and its non-holonomic variant (Alonso-Mora et al., 2013), which are effective for fast and exact multi-agent navigation. ORCA conservatively imposes collision avoidance constraints on the motion of a robot in terms of half-planes in the space of velocities. The optimal collision-free velocity can then be quickly found by solving a linear program. The original framework limits itself to holonomic systems but has been extended in Alonso-Mora et al. (2013) to model non-holonomic constraints with differential drive dynamics. ORCA also generates collision-free velocities that deviate minimally from the robots' preferred velocities. The major

---

[1] while our approach is decentralized in most practical scenarios, we utilize a centralized auction protocol for tie-breaking in the rare case when robots speeds are exactly identical *up to numeric precision* Our approach then switches to a centralized mode for *only as long as the auction is active*, and returns to decentralized mode when the auction is switched off. The auction, however, is rarely used in practice.

limitation of the ORCA framework is that the structure of the half-planes so constructed often results in deadlocks (Dergachev & Yakovlev, 2021).

Proving safety is harder for systems with double-integrator dynamics, therefore safety in these systems depends on the planning frequency of the system. For example, the NH-TTC algorithm (Davis et al., 2019) uses gradient descent to minimize a cost function comprising a goal reaching term and a time-to-collision term, which rises to infinity as the agent approaches immediate collision. NH-TTC guarantees safety in the limit as the planning frequency approaches infinity. Other optimization-based approaches use model predictive control (MPC) (Chen et al., 2022); in such approaches, safety depends not only on the planning frequency but also on the length of the planning horizon.

Finally, Control Barrier Functions (CBFs) (Grover et al., 2016; Wang et al., 2017) can be used to design controllers that can guarantee safety via the notion of forward invariance of a set i.e. if an agent starts out in a safe set at the initial time step, then it remains safe for all future time steps, that is, it will never leave the safe set.

2.2 Deadlock resolution methods

Deadlocks among agents arise due to symmetry in the environment that may cause conflicts between the agents (Grover et al., 2020; Chen et al., 2022; Grover et al., 2016). To break the symmetry, and escape the deadlock, agents must be perturbed, which can be done in several ways. The most naive, and easiest, way is to randomly perturb each agent (Wang et al., 2017). Random perturbations can be implemented in decentralized controllers and can generalize to many agents, but are sub-optimal in terms of path deviation and overall cost. Next, there are several recent efforts to choreograph the perturbation according to some set rules such as the right-hand rule (Chen et al., 2022; Zhou et al., 2017) or clockwise rotation (Grover et al., 2016). These strategies improve performance over random perturbation and even give formal guarantees of optimality, but the imposed pre-determined ordering limits their generalizability; many cannot generalize to more than 3 agents. Another line of research aims towards deadlock *prevention* rather than resolution where an additional objective is to identify and mitigate potential deadlocks, even before they happen, such as in Chen et al. (2022).

Another class of deadlock resolution methods rely on priority protocols and scheduling algorithms similar to those used in the autonomous intersection management literature (Zhong et al., 2020). Some prominent protocols include first come first served (FCFS), auctions, and reservations. FCFS (Au et al., 2015) assigns priorities to agents based on their arrival order at the intersection. It is easy to implement but can lead to long wait times and high congestion if multiple vehicles arrive at the intersection simultaneously. In auctions (Carlino et al., 2013; Suriyarachchi et al., 2022), agents bid to cross the intersection based on a specific bidding strategy. Reservation-based systems (Dresner & Stone, 2008) are similar to the auction-based system in which agents reserve slots to cross the intersection based on their estimated arrival and clearance times.

As noted by recent researchers (Chen et al., 2022; Zhou et al., 2017), developing a provably optimal, decentralized, and general deadlock resolution technique is currently an open problem. In this work, we take a large step forward towards a solution.

2.3 Learning-based approaches

Coupling classical navigation with machine learning is rapidly growing field and we refer the reader to Xiao et al. (2022) for a recent survey on the state-of-the-art of learning-based motion planning. Here, we review two categories of approaches that have been reasonably succesful in multi-agent planning and navigation. These are methods based on deep reinforcement learning (DRL) and trajectory prediction. DRL has been used to train navigation policies in simulation for multiple robots in social mini-games. Long et al. (2018) presents a DRL approach for multi-robot decentralized collision avoidance, using local sensory information. They present simulations in various group scenarios including social mini-games. CADRL (Chen et al., 2017), or Collision Avoidance with Deep Reinforcement Learning, is a state-of-the-art motion planning algorithm for social robot navigation using a sparse reward signal to reach the goal and penalizes robots for venturing close to other robots. A variant of CADRL uses LSTMs to select actions based on observations of a varying number of nearby robots (Everett et al., 2021). Planning algorithms that use trajectory prediction models (Kamenev et al., 2022) estimate the future states of the robot in the presence of dynamic obstacles, and plan their actions accordingly. However, DRL-based methods are generally hard to train requiring tens of thousands of data samples, hours of training time, do not generalize to out of distribution environments, and lastly, they impose only soft constraints (collision checking) on safety, and do not provide hard guarantees.

On the other hand, Imitation Learning (IL) corresponds to a machine learning paradigm where an autonomous agent strives to learn a behavior by emulating an expert's demonstrations. IL-based approaches are much faster than traditional optimization-based approaches. Typically, the demonstrations comprise state-input pairs, generated by an expert policy during real-world execution. Addressing the imitation learning problem has led to three major approaches, namely, Behavior Cloning (BC) (Ross et al., 2011, ?; Daftry et al., 2016), which involves direct learning of an imitative

policy through supervised learning, Inverse Reinforcement Learning (IRL) (Ziebart et al., 2008; Mehr et al., 2023; Gonon & Billard, 2023), wherein a reward function is first inferred from the demonstrations, subsequently used to guide policy learning via Reinforcement Learning (RL) (Sutton & Barto, 2018), and generative models (Ho & Ermon, 2016; Kostrikov et al., 2019; Dadashi et al., 2020). We refer the reader to Zheng et al. (2022) for a more comprehensive background on IL. However, a shared drawback among the IL algorithms (Kostrikov et al., 2019; Dadashi et al., 2020; Zheng et al., 2022; Torabi et al., 2018; Karnan et al., 2022) is their inability to encode state/safety and input constraints underlying assumption of having access to the expert's action data during the demonstration phase. Additionally, these methods do not transfer to domains that are outside the distribution of the expert demonstrations.

2.4 Game-theoretic distributed optimization

Another class of methods for multi-agent planning for self-interested agents includes distributed optimization in general-sum differential games. The literature on general-sum differential games classify existing algorithms for solving Nash equilibria in robot navigation into four categories. First, there are algorithms based on decomposition (Wang et al., 2021; Britzelmeier et al., 2019), such as Jacobi or Gauss-Siedel methods, that are easy to interpret and scale well with the number of players, but have slow convergence and may require many iterations to find a Nash equilibrium. The second category consists of games (Fisac et al., 2019), such as Markovian Stackelberg strategy, that capture the game-theoretic nature of problems but suffer from the curse of dimensionality and are limited to two players. The third category consists of algorithms based on differential dynamic programming (Schwarting et al., 2021; Sun et al., 2015, 2016; Morimoto & Atkeson, 2003; Fridovich-Keil et al., 2020; Di & Lamperski, 2018) that scale polynomially with the number of players and run in real-time, but do not handle constraints well. Lastly, the fourth category contains algorithms based on direct methods in trajectory optimization (Le Cleac'h et al., 2022; Di & Lamperski, 2019, 2020), such as Newton's method, that are capable of handling general state and control input constraints, and demonstrate fast convergence. The algorithms described above give an analytical, closed-form solution that guarantees safety but not liveness. Additional limitations include the lack of deployability in the real world and the requirement of full observation.

The key distinction between our approach and traditional game-theoretic approaches is that unlike in the latter case, our approach does not explicitly enforce a Nash equilibrium which requires direct knowledge of the cost functions of other agents. Instead, our approach implicitly and indirectly achieves a solution that resembles a Nash equilibrium. So

we essentially perform a tradeoff - we deviate slightly from a precise Nash solution to something that looks like a Nash solution, but our approach works in realtime on real robots in a fully decentralized manner.

3 Problem formulation and background

In this section, we begin by formulating social mini-games followed by stating the problem objective. Notations used in this paper are summarized in Table 2.

3.1 Problem formulation

We formulate a social mini-game by augmenting a partially observable stochastic game (POSG) (Hansen et al., 2004): $\langle k, \mathcal{X}, \{\Omega^i\}, \{\mathcal{O}^i\}, \{\mathcal{U}^i\}, \mathcal{T}, \{\widetilde{\Gamma}^i\}, \{\mathcal{J}^i\} \rangle$ where k denotes the number of robots. Hereafter, i will refer to the index of a robot and appear as a superscript whereas t will refer to the current time-step and appear as a subscript. The general state space $\mathcal{X}$ (e.g. SE(2), SE(3), etc.) is continuous; the ith robot at time t has a state $x_t^i \in \mathcal{X}$. A state x_t^i consists of both visible parameters (e.g. current position, linear and angular velocity and hidden (to other agents) parameters which could refer to the internal state of the robot such as preferred speed, preferred heading, etc. We denote the set of observable parameters as $\overline{x}_t^i$. On arriving at a current state x_t^i, each robot generates a local observation, $o_t^i \in \Omega^i$, via $\mathcal{O}^i : \mathcal{X} \longrightarrow \Omega^i$, where $\mathcal{O}^i \left(x_t^i \right) = \{x_t^i\} \cup \left\{ \overline{x}_t^j : j \in \mathcal{N}^i \left(x_t^i \right) \right\}$, the set of robots detected by i's sensors. Over a finite horizon T, each robot is initialized with a start state $x_0^i \in \mathcal{X}_I$, a goal state $x_N^i \in \mathcal{X}_g$ where $\mathcal{X}_I$ and $\mathcal{X}_g$ denote subsets of $\mathcal{X}$ containing the initial and final states. The transition function is given by $\mathcal{T} : \mathcal{X} \times \mathcal{U}^i \longrightarrow \mathcal{X}$, where $\mathcal{U}^i$ is the continuous control space for robot i representing the set of admissible inputs for i. A discrete trajectory is specified as $\Gamma^i = \left(x_0^i, x_1^i, \ldots, x_T^i \right)$ and its corresponding input sequence is denoted by $\Psi^i = \left(u_0^i, u_1^i, \ldots, u_{T-1}^i \right)$. Robots follow the discrete-time control-affine system,

$$x_{t+1}^i = f\left(x_t^i\right) + g\left(x_t^i\right) u_t^i \tag{1}$$

which is obtained after applying a Runge–Kutta discretization scheme to a continuous-time system describing their motion (derived from first principles) which takes the following form:

$$x^i(t) = f_c\left(x^i(t)\right) + g_c\left(x^i(t)\right) u^i(t), \tag{2}$$

where f_c and g_c are locally Lipschitz continuous functions (note that in general, the state and input vectors of

Table 2 Summary of notation used in this paper

Symbol	Description
Problem formulation (Section 3.1)	
k	Number of agents
T	planning horizon
$\mathcal{X}$	general continuous state space
$\mathcal{X}_I$	set of initial states
$\mathcal{X}_g$	set of final states
x_t^i	state of agent i at time t
$\bar{x}_t^i$	observable state of agent i to other agents
Ω^i	observation set of agent i
o_t^i	observation of agent i at time t
$\mathcal{O}^i$	observation function ($\mathcal{O}^i : \mathcal{X} \rightarrow \Omega^i$)
$\mathcal{N}^i\left(x_t^i\right)$	set of robots detected by i
Γ^i	agent i's trajectory
$\widetilde{\Gamma}^i$	set of preferred trajectories
$\mathcal{T}$	transition dynamics (Eq. 1)
$\mathcal{U}^i$	action space for agent i
$\mathcal{J}^i$	running cost for agent i ($\mathcal{J}_t^i : \mathcal{X} \times \mathcal{U}^i \rightarrow \mathbb{R}$)
$\mathcal{J}_f^i$	running cost at time T
$\mathcal{C}^i\left(x_t^i\right) \in \mathcal{X}$	convex hull of agent i
$\pi^i \in \mathcal{K}$	controller belonging to set of controllers
Control Barrier Function (Section 4)	
$h^i : \mathcal{X} \longrightarrow \mathbb{R}$	control barrier function
$\mathscr{C}^i$	safe set
$\partial \mathscr{C}^i$	boundary of $\mathscr{C}^i$
$L_f h^i\left(x_t^i\right), L_g h^i\left(x_t^i\right)$	lie derivatives of $h^i\left(x_t^i\right)$ w.r.t f and g
Deadlock Prevention (Section 5)	
$[1; N]$	set of integers $\{1, 2, \ldots, N\}$
$\mathscr{C}_\ell(t)$	liveness set
v_t^i	linear velocity of agent i
v_t	joint velocity of all agents
$\widetilde{v}_t$	perturbed joint velocity
$h_v(x_t)$	CBF of $\mathscr{C}_\ell(t)$
p_t^i	position of agent i
θ_t^i	angle of agent i
$\ell_j\left(p_t^i, v_t^i\right)$	liveness function for agent i
σ	priority ordering
σ_{OPT}	optimal ordering
α_q	time-based reward for receiving an order position q
b^i	bid made by agent i
(r^i, p^i)	auction specified by allocation and payment rule
ζ^i	private priority incentive parameter

the continuous-time and the discrete-time systems that correspond to the same time instant are not the same due to discretization induced errors). In this paper, we will mainly utilize the discrete-time model (1), especially for control design purposes (these methods will rely on optimization techniques), but we will also refer to the continuous-time model (2) for analysis purposes. We will also assume that the continuous-time system (2) is small-time controllable, which means that set of points reachable from $x^i(t)$ within the time interval $[t, t']$ will contain a neighborhood of $x^i(t)$ for any $t' > t$ (Laumond et al., 2005). Small-time controllability allows the following result to hold true:

Theorem 1 (Laumond et al. (2005)) *For symmetric small-time controllable systems the existence of an admissible collision-free path between two given configurations is equivalent to the existence of any collision-free path between these configurations.*

We denote by $\widetilde{\Gamma}^i$ as the set of *preferred* trajectories for robot i that solve the two-point boundary value problem. A preferred trajectory, as defined by existing methods (Chen et al., 2022; Van Den Berg et al., 2011; Alonso-Mora et al., 2013), refers to a collision-free path a robot would follow in the absence of dynamic or static obstacles and is generated via a default planner according to some predefined criteria such as shortest path, minimum time, etc. A collision is defined as follows. Let $\mathcal{C}^i\left(x_t^i\right) \subseteq \mathcal{X}$ represent the space occupied by robot i (as a subset of the state space $\mathcal{X}$) at any time t which can be approximated by the convex hull of a finite number of points that determine the boundary of the robot (e.g., vertices of a polytopic set). Then, robots i and j are said to collide at time t if $\mathcal{C}^i\left(x_t^i\right) \cap \mathcal{C}^j\left(x_t^j\right) \neq \emptyset$.

Each robot has a running cost $\mathcal{J}^i : \mathcal{X} \times \mathcal{U}^i \longrightarrow \mathbb{R}$ that assigns a cost to a state-input pair $\left(x_t^i, u_t^i\right)$ at each time step based on user-defined variables e.g. distance of the robots current position from the goal, change in the control across subsequent time steps, and distance between the robots preferred and actual paths. Finally, we define a social mini-game as follows,

Definition 1 A **social mini-game** 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 $\Gamma^i \in \widetilde{\Gamma}^i, \Gamma^j \in \widetilde{\Gamma}^j$, we have $\mathcal{C}^i\left(x_t^i\right) \cap \mathcal{C}^j\left(x_t^j\right) \neq \emptyset \ \forall \ t \in [a, b]$, where x_t^i, x_t^j are elements of Γ^i and Γ^j.

We depict several examples and non-examples of social mini-games in Fig. 1. The first scenario can be characterized
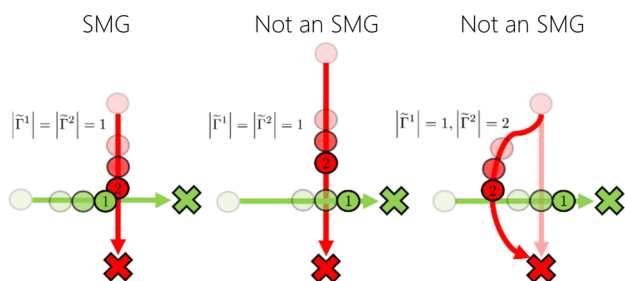
Fig. 1 Examples/Counter-examples of social mini-games: Arrows indicate the direction of motion for two agents 1 and 2 toward their goals marked by the corresponding cross. The first scenario is a social mini-game since both the preferred trajectories of agents 1 and 2 are in conflict from some $t = a$ to $t = b$ where $b - a \geq \delta$. The second and third scenarios are *not* social mini-games as there are no conflicts. In the second scenario, there is no duration where agents intersect one another. In the third scenario, agent 2 has an alternate conflict-free preferred trajectory to fall back on

as a social mini-game due to the conflicting preferred trajectories of agents 1 and 2 within a specific time interval $[a, b]$, where the duration $b - a \geq \delta$. In this case, the agents' trajectories intersect, generating a conflict. However, the second and third scenarios, in contrast, do not qualify as social mini-games since no conflicts arise between the agents. In the second scenario, there is no common time duration where agents intersect each other, and their trajectories remain independent in time. In the third scenario, agent 2 possesses an alternative preferred trajectory that avoids conflicts during any time duration, allowing for a seamless transition to a conflict-free path. A robot has the following best response in a social mini-game:

Definition 2 Optimal Solution for a Robot in Social Mini-Games: For the ith robot, given its initial state x_0^i, an optimal trajectory $\Gamma^{i,*}$ and corresponding optimal input sequence $\Psi^{i,*}$ are given by,

$$\left(\Gamma^{i,*}, \Psi^{i,*}\right) = \arg\min_{(\Gamma^i, \Psi^i)} \sum_{t=0}^{T-1} \mathcal{J}^i\left(x_t^i, u_t^i\right) + \mathcal{J}_T^i\left(x_N^i\right)$$
(3a)

$$\text{s.t } x_{t+1}^i = f\left(x_t^i\right) + g\left(x_t^i\right) u_t^i, \quad \forall t \in [1; T-1]$$
(3b)

$$\mathcal{C}^i\left(x_t^i\right) \cap \mathcal{C}^j\left(x_t^j\right) = \emptyset \ \forall j \in \mathcal{N}^i\left(x_t^i\right) \ \forall t$$
(3c)

$$x_T^i \in \mathcal{X}_g$$
(3d)

where $\mathcal{J}_T^i$ is the terminal cost. A solution to optimal navigation in social mini-games is a (finite) sequence of state-input pairs $\left(\left(\Gamma^{1,*}, \Psi^{1,*}\right), \left(\Gamma^{2,*}, \Psi^{2,*}\right), \ldots, \left(\Gamma^{k,*}, \Psi^{k,*}\right)\right)$.

3.2 Algorithm Overview for Solving Social Mini-games

As solving (3) jointly is computationally intensive (Chen et al., 2010), we solve the multi-robot navigation problem (3) for the discrete-time system (1) in a decentralized fashion. Each robot solves (3) in a receding horizon fashion using an MPC asynchronously; on the ith robot's turn, it treats all other robots as static obstacles that are incorporated into the safety constraint (3c). At each simulation iteration for the ith robot, the MPC controller solves the constrained finite-time optimal control problem (3) with horizon T. The optimal control sequence to this optimization problem for the ith robot is a sequence of inputs $\Psi^{i,*} = \left(u_t^{i,*}, u_{t+1}^{i,*} \ldots, u_{t+T-1}^{i,*}\right)$, $u_t^{i,*} \in \mathcal{U}_t^i$. Then, the first element of the solution, $u_t^i \leftarrow u_t^{i,*}$, is executed by robot i. The optimization is repeated at the next iteration based on the updated state for all robots.

Solving (3a) asynchronously, however, may lead to different agents making decisions based on different sets of information at different times. This inconsistency can lead to a lack of coordination among agents, resulting in safety and deadlock issues. We address both these issues with a combination of using a sufficiently small sampling time as well as using control barrier functions in the following manner:

- *Safety (Constraint* 3c): We model static and dynamic safety in Equation 3c in the context of set invariance using Control Barrier Functions (Ames et al., 2019) (CBFs).
- *Deadlock Prevention (Constraint* 3d): We detect deadlocks by exploiting the geometrical symmetry of social mini-games Grover et al. (2020) in an online fashion, and design a minimally invasive perturbation strategy to prevent the deadlock from occurring. The deadlock prevention algorithm can be integrated with CBFs for double-integrator (or higher order) dynamical systems.

In the remainder of this article, we describe our approaches for addressing safety via CBFs in Section 4 followed by our deadlock prevention algorithm in Section 5. We present results in Section 6 and conclude in Section 7.

4 Safety via control barrier functions (Constraint 3c)

Control Barrier Functions (CBFs) are a powerful tool used in control theory for designing controllers that guarantee safety of controlled nonlinear systems (Ames et al., 2019). CBF-based controllers constrain the behavior of a robot system to enable collision-free trajectories. CBF-based controllers also

guarantee safety and robustness in multi-agent systems and scale to a large number of robots, while easily adapting to changes in the environment or robot dynamics. Additionally, they have been combined with traditional control techniques, such as Model Predictive Control (MPC) (Zeng et al., 2021) and PID, for improved safety and performance. In this background, we will discuss the basic theory of CBFs and their mathematical formulation.

The safety of a set $\mathscr{C}^i$ for a given system is closely related to its forward invariance which is a property that requires the system starting from a given set of initial conditions inside $\mathscr{C}^i$ to remain in $\mathscr{C}^i$ for all times. We first describe CBFs for the continuous-time case and then specify the modified variant for the discrete-time case. Consider a scalar valued function, $h^i : \mathcal{X} \longrightarrow \mathbb{R}$, where $\mathcal{X}$ denotes the set of admissible states $x_t^i \in \mathcal{X}$ such that the following conditions hold:

$$\mathscr{C}^i = \left\{ x_t^i \in \mathbb{R}^n | h^i \left(x_t^i \right) \geq 0 \right\} \tag{4a}$$

$$h^i \left(x_t^i \right) = 0 \ \forall \ x_t^i \in \partial \mathscr{C}^i \tag{4b}$$

$$h^i \left(x_t^i \right) < 0 \ \forall \ x_t^i \in \mathbb{R}^n \setminus \mathscr{C}^i \tag{4c}$$

where $\mathscr{C}^i$ is the safe set and $\partial \mathscr{C}^i$ denotes its boundary. The time derivative of $h^i \left(x_t^i \right)$ along the state trajectory of agent i is given as

$$\frac{d \left(h^i \left(x_t^i \right) \right)}{dt} = L_f h^i \left(x_t^i \right) + L_g h^i \left(x_t^i \right) u_t^i \tag{5}$$

where $L_f h^i \left(x_t^i \right)$ and $L_g h^i \left(x_t^i \right)$ denote the Lie derivatives of $h^i \left(x_t^i \right)$ along f and g, respectively.

Then, $h^i \left(x_t^i \right)$ is a CBF if there exists a class $\mathcal{K}_\infty$[2] function κ such that the following holds true

$$\sup_{u_t^i \in U} L_f h^i \left(x_t^i \right) + L_g h^i \left(x_t^i \right) u_t^i + \kappa \left(h^i \left(x_t^i \right) \right) \geq 0 \tag{6}$$

We define the safe or collision-free control space $\mathscr{U}^i$ over $x_t^i \in \mathcal{X}$ to be the set of control inputs $u_t^i \in \mathcal{U}^i$ such that the following inequality holds:

$$L_f h^i \left(x_t^i \right) + L_g h^i \left(x_t^i \right) u_t^i + \kappa \left(h^i \left(x_t^i \right) \right) \geq 0 \tag{7}$$

Equation (7) is known as the safety barrier constraint or safety barrier certificate. In summary, the set $\mathscr{C}^i \subseteq \mathcal{X}$ is guaranteed to be safe if $\mathscr{U}^i$ is non-empty and $u_t^i \in \mathscr{U}^i$. The formulation of safety barrier certificates presented above is for a single robot, but it can be extended trivially to

multi-robot scenarios. The only difference is that the barrier function $h^i \left(x_t^i \right)$ will change to $h(x_t)$; that is, it will lose the indexing on i and will become a function of $\mathcal{X}^k \longrightarrow \mathbb{R}$, which denotes the aggregate states of k robots. Similarly, $\mathscr{C}^i$ will become simply $\mathscr{C} \subset \mathcal{X}^k$. As the system we consider in this paper is discrete-time, due to our MPC-based formulation, instead of (7), we will be using the discrete-time version of the CBF which is given by,

$$\Delta h_s^i \left(x_t^i, u_t^i \right) \geq -\gamma h_s^i \left(x_t^i \right) \tag{8}$$

where $h_s^i \left(x_t^i \right)$ is chosen as an obstacle avoidance barrier function (more details in Sect. 6.1), $\gamma > 0$, and $\Delta h_s^i \left(x_t^i, u_t^i \right)$ is given by,

$$\Delta h_s^i \left(x_t^i, u_t^i \right) = h_s^i \left(x_{t+1}^i \right) - h_s^i \left(x_t^i \right) \tag{9}$$

The set $\mathscr{U}^i$ that guarantees safety is thus given by,

$$\mathscr{U}^i = \{ u_t^i \in \mathcal{U}^i \ | \Delta h_s^i \left(x_t^i, u_t^i \right) + \gamma h_s^i \left(x_t^i \right) \geq 0 \} \tag{10}$$

CBFs offer a stronger guarantee of safety than directly imposing collision avoidance constraints in an MPC formulation. In Zeng et al. (2021), it was conclusively found that the problem with directly imposing collision avoidance constraints into an MPC, as opposed to using CBFs, is that the constraint will take effect *only* when the planning horizon intersects with an obstacle. In other words, the robot will not take any action to avoid obstacles until it is close to them (refer to the Figure 2 in Zeng et al. (2021)). One way to solve this problem is to use a larger horizon, but that will increase the computational complexity in the optimization. We refer the reader to Zeng et al. (2021) for more details.

5 Deadlock prevention (Constraint 3d)

In social mini-games, however, $\mathscr{U}_t^i$ often ends up being an empty set, i.e. there is no input for which (8) is satisfied, resulting in deadlocks (Wang et al., 2017; Zhou et al., 2017; Chen et al., 2022) due to the symmetry in the environment configuration (Grover et al., 2020). As in previous work (Wang et al., 2017; Grover et al., 2016, 2020), we define a deadlock as follows,

Definition 3 Deadlock: A system of k robots executing the controller given by Equation (3) enters a deadlock if, starting at time t, there exists at least one agent i such that $u_t^i = 0$ for some threshold $\beta > 0$, typically measured in the order of a few seconds, while $x_t^i \notin \mathcal{X}_g$, for each robot $i \in [1, k]$. Mathematically, if there exists at least one pair $\left(x_t^i, u_t^i \right) \in$

---

[2] A function $\alpha(\cdot) : \mathbb{R} \to \mathbb{R}$ belongs to the class of $\mathcal{K}_\infty$ functions if it is strictly increasing and in addition, $\alpha(0) = 0$ and $\lim_{r \to \infty} \alpha(r) = \infty$

$\mathcal{D}^i(t)$ where the set $\mathcal{D}^i(t)$ is defined as follows

$$\mathcal{D}^i(t) = \left\{ \left(x_t^i, u_t^i \right) : x_t^i \notin \mathcal{X}_g, \; u_t^i = 0 \text{ for some } \beta > 0 \right\} \tag{11}$$

then the system of these k robots is said to be in deadlock. Furthermore, a trajectory that contains such u_t^i that satisfy the deadlock condition (11) is said to be *inadmissible*.

Here, β is typically in the order of a few seconds (Şenbaşlar et al., 2023). Resolving or preventing deadlocks involves a two-stage procedure: (i) detecting the deadlock and (ii) resolving or preventing the deadlock. So far, the literature (Wang et al., 2017; Arul et al., 2023; Zhou et al., 2017; Grover et al., 2016; Chen et al., 2022) has focused on resolving deadlocks after they happen, which then require resolution strategies that are quite invasive. Commonly used strategies include perturbing the position of a robot according to the right-hand rule causing it to deviate from its preferred trajectories in a clockwise direction resulting in a sub-optimal controller. In the following section, we formalize an approach to detect, and the prevent, deadlocks via a minimally invasive perturbation. To aid readability, we visualize our theoretical framework in Figure 4. We begin by defining minimally invasive deadlock prevention as follows:

Definition 4 Minimally Invasive Deadlock Prevention (or Resolution). A deadlock preventive (or resolving) control strategy $u_t^i = \left[v_t^i, \omega_t^i \right]^\top$ prescribed for robot i at time t with current heading angle θ_t^i is said to be minimally invasive if:

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

In other words, $\delta_{\text{opt}}(t)$ can be computed by solving the following optimization problem

$$\delta_{\text{opt}}(t) = \underset{\delta \in \mathbb{R}}{\arg\min} \left\| v_t^i + \delta \right\|, \tag{12a}$$

$$v_{t+1}^i = v_t^i + \delta \tag{12b}$$

$$u_t^i \in \mathcal{U}^i, \; u_{t+1}^i \in \mathcal{U}^i \tag{12c}$$

$$\left(x_{t+1}^i, u_{t+1}^i \right) \notin \mathcal{D}^i(t+1) \tag{12d}$$

Note that the optimization problem (12d) is a non-trivial problem to solve as the set $\mathcal{D}^i(t)$ is not known priori. Furthermore, even if the set $\mathcal{D}^i(t)$ is known apriori, finding $\delta_{\text{opt}}(t)$ could be computationally expensive especially when the set $\mathcal{D}^i(t)$ is non-convex. A minimally invasive perturbation does

not cause a robot to deviate from its preferred trajectory (condition (1)) only allowing it to speed up or slow down (condition (2)).

Theorem 2 *Consider a symmetrical social mini-game, $\mathcal{S}$, as shown in Fig. 2a. We represent robots as single-integrator car-like objects with lengths $l^1 = l^2 = l$, speeds v_t^1, v_t^2, and the following system dynamics:*

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{bmatrix}$$

such that $0 < \left| v_t^1 - v_t^2 \right| \leq \epsilon$. Assume further that they are at distances $d^1 = d^2 = d$ from the point of collision $\mathcal{Q}$ (could be a doorway or intersection). For the robots to reach $\mathcal{Q}$ without colliding in a minimally invasive sense (Definition 4), one of the robots must slow down (or speed up) by a factor of $\zeta \geq 2$ in the limit as $(d, \epsilon) \rightarrow (l, 0)$.

Proof In a symmetric SMG consisting of two robots, we assume that the speeds of the robots are nearly identical. In practice, two speeds will likely differ by some small amount, which we denote as ϵ giving $0 < |v_t^1 - v_t^2| \leq \epsilon$. Suppose, without loss of generality, that robot 1 is slightly faster than robot 2 by ϵ. Thus, $v_t^1 - v_t^2 = \epsilon \implies v_t^2 = v_t^1 - \epsilon$. In order for robot 1 to clear the distance d in the time that robot 2 reaches the $\mathcal{Q}$, robot 1 must increase its speed by a factor of ζ. Therefore $v_{t+1}^1 \leftarrow \zeta v_t^1$ while $v_{t+1}^2 = v_t^1 - \epsilon$. We assume that speed changes are instantaneous as the agents enter the SMG.

Now time taken by robot 2 to reach Q is $t_2 = \frac{d}{v_{t+1}^2} = \frac{d}{v_t^1 - \epsilon}$. And the time taken by robot 1 to *clear* Q (that is, not just reach Q, but that the entire length l should clear Q) is $t_1 = \frac{d+l}{v_{t+1}^1} = \frac{d+l}{\zeta v_t^1}$. Note that $t_2 \geq t_1$. Thus,

$$\frac{d}{v_t^1 - \epsilon} \geq \frac{d+l}{\zeta v_t^1}$$

Rearranging terms gives us Equation (13),

$$\zeta \geq \left(1 + \frac{l}{d} \right) \left(1 - \frac{\epsilon}{v_t^1} \right) \tag{13}$$

In the limit, $\zeta \rightarrow 2$ as $(d, \epsilon) \rightarrow (l, 0)$. $\square$

5.1 Deadlock detection

A state is represented as x_t^i at time t of which $\left(p_t^i, \theta_t^i, v_t^i, \omega_t^i \right) \in \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^2 \times \mathbb{R}$ represent the current position, heading, linear and angular velocities of the ith robot. Previous studies on deadlock detection (Chen et al., 2022; Grover et al., 2020) show that deadlocks arise from symmetry in the environment
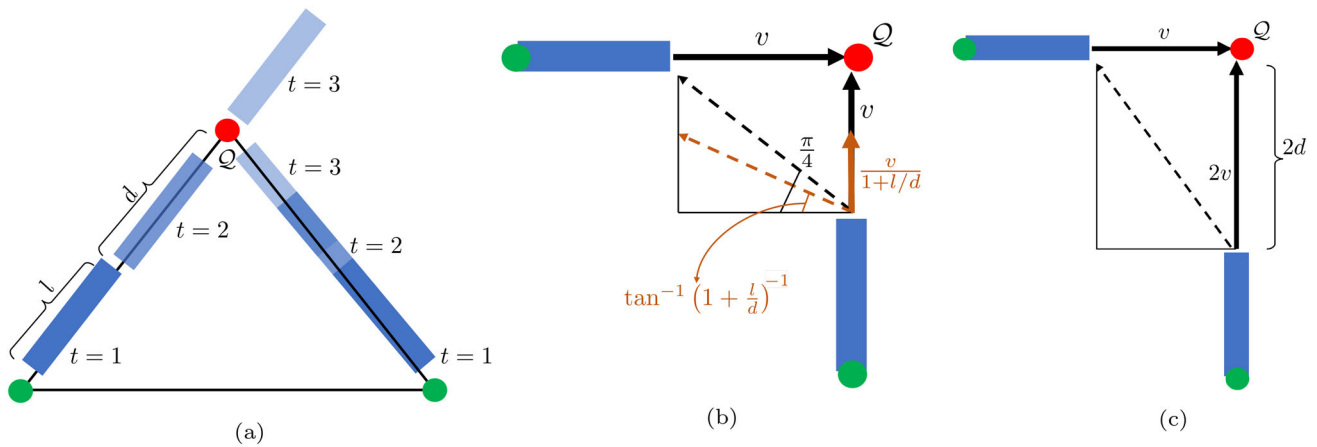
Fig. 2 Deriving the threshold, ℓ_{thresh}, in a symmetrical social minigame when $\ell_j(p_t^i, v_t^i) = 0$. Robots are blue car-like systems of length l. Green and red circles represent starting and goal positions, respectively. *(left)* Step 1: construct the worst case scenario when $\ell_j(p_t^i, v_t^i) = 0$. The second robot on the right must reach $\mathcal{Q}$ when the robot on the left passes fully through $\mathcal{Q}$. *(middle) Step* 2: Assuming the second robot slows down by $\frac{v}{1+\frac{l}{d}}$, ℓ_{thresh} is obtained via Equation (16). *(right)* Symmetric SMGs with arbitrary angles with the relative position vector configurations can be reduced to the SMG in Figure 2b (Color figure online)
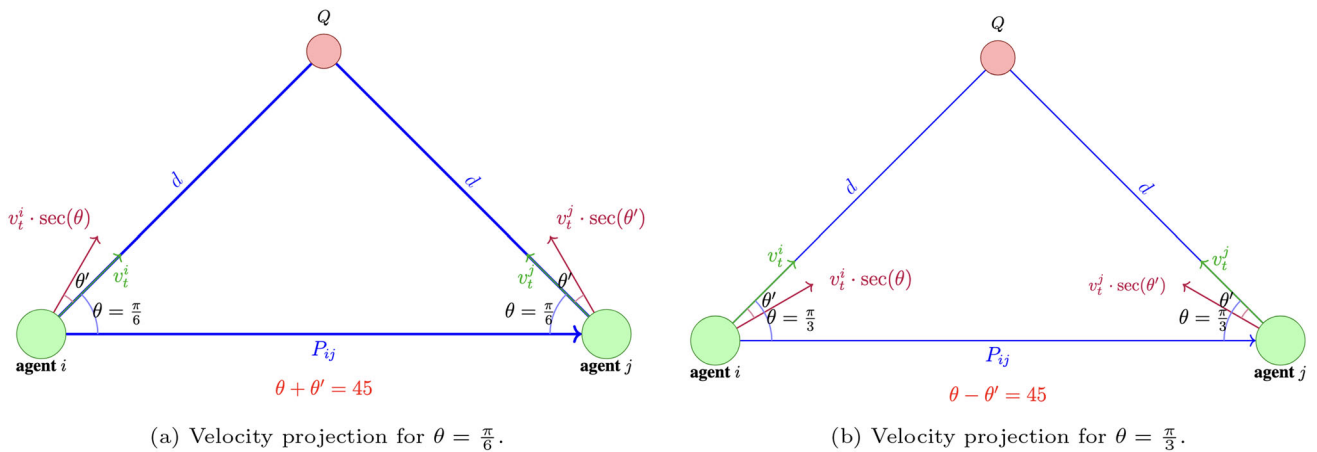
Fig. 3 Velocity projection for two symmetrical SMGs with arbitrary angles θ with respect to the relative position vector. In these two examples, we examine when $\theta = \frac{\pi}{6}$ and $\theta = \frac{\pi}{3}$ (although any arbitrary $|\theta| < \frac{\pi}{2}$ may be chosen)

configuration. Symmetry is geometrically defined in terms of the initial positions, goals, and velocities of the agents. Example of symmetrical configurations are given in Fig. 2. Figures 2a and 2b depict a symmetrical SMG with $\theta = \frac{\pi}{4}$ whereas Fig. 2c depicts a symmetrical SMG with arbitrary angles.

An important observation is that any symmetrical SMG with arbitrary angle θ can be reduced to a symmetrical SMG with angle $\frac{\pi}{4}$. In any symmetrical SMG with arbitrary angle θ, $|\theta| < \frac{\pi}{2}$ (agents diverge away from each other for $|\theta| \geq \frac{\pi}{2}$), with respect to the relative position vector, suppose two robots i and j have nearly identical velocities $v_t^i \approx v_t^j$ (otherwise, it would not be a symmetrical SMG) or $\left| v_t^i - v_t^j \right| \leq \epsilon$. Then, without loss of generality, we can project a component of each robots velocity such that it sub-

tends an θ' with the original corresponding velocity vector. The advantage of such a projection is that it eliminates the dependency on the angle, with the only dependency remaining on the scaling factor ζ. We set θ' as follows:

1. **Case 1:** When the arbitrary angle $\theta < \frac{\pi}{4}$, then we set $\theta' = \frac{\pi}{4} - \theta$.
2. **Case 2:** When the arbitrary angle $\frac{\pi}{2} > \theta \geq \frac{\pi}{4}$, then we set $\theta' = \theta - \frac{\pi}{4}$.

The new transformed velocities are given by $v_t^{i,\prime} = v_t^i \sec \theta'$ and $v_t^{j,\prime} = v_t^j \sec \theta'$. Figure 3 demonstrates the velocity projection for two symmetrical SMGs with arbitrary angles θ with respect to the relative position vector. In Figs. 3a and 3b, we examine when $\theta = \frac{\pi}{6}$ and $\theta = \frac{\pi}{3}$, respec-
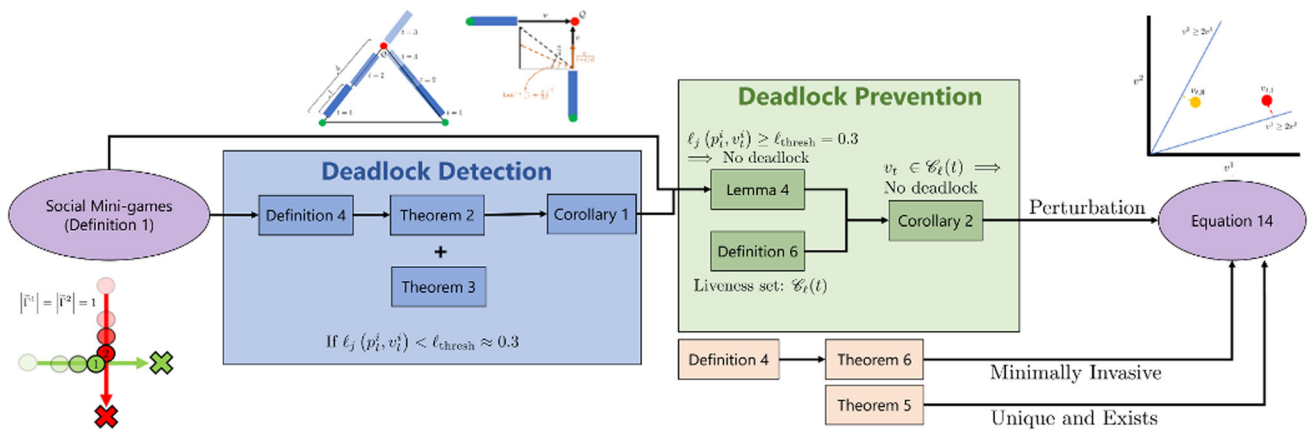
Fig. 4 Flowchart of the deadlock detect-and-prevent logic. This diagram illustrates at a high level the process for detecting and preventing deadlocks in our approach. It includes the foundational definition of social mini-games, followed by the prevention strategies enforced by liveness conditions. The interaction between these elements is visualized through connections between theoretical results in Sect. 5

tively, although any arbitrary $|\theta| < \frac{\pi}{2}$ may be chosen. We now define a liveness function as follows:

Definition 5 Liveness Function: A liveness function $\ell_j : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ between agents i and j ($i \neq j$) as follows:

$$\ell_j(p_t^i, v_t^i) = \cos^{-1}\left(\frac{\langle \overrightarrow{p_t^i - p_t^j}, \overrightarrow{v_t^{i,\prime} - v_t^{j,\prime}} \rangle}{\left\| \overrightarrow{p_t^i - p_t^j} \right\| \left\| \overrightarrow{v_t^{i,\prime} - v_t^{j,\prime}} \right\| + \epsilon} \right) \quad (14)$$

where $v_t^{i,\prime} = v_t^i \cdot \sec(\theta')$, $v_t^{j,\prime} = v_t^j \cdot \sec(\theta')$ correspond to the velocity projection component of each robot's velocity such that it subtends an angle $\theta' = \pm\left(\frac{\pi}{4} - \theta\right)$ with the original velocity vectors, v_t^i and v_t^j, as defined by the velocity transformation procedure, where $|\theta| < \frac{\pi}{2}$ is the arbitrary SMG angle. Further, $\langle a, b \rangle$ denotes dot product between vectors a and b and $\|.\|$ denotes the Euclidean norm, $\ell_j \in [0, \pi]$, and $\epsilon > 0$ ensures that the denominator is positive. Lastly, note that $\ell_j(p_t^i, v_t^i) = \ell_i(p_t^j, v_t^j)$ for $i \neq j$.

The liveness function is meant to geometrically capture the symmetry of an SMG (refer to Figure 2). Formally, the liveness function measure the angle between the relative displacement ($\overrightarrow{p_t^{12}} = \overrightarrow{p_t^1 - p_t^2}$) and the relative linear velocity ($\overrightarrow{v_t^{12,\prime}} = \overrightarrow{v_t^{1,\prime} - v_t^{2,\prime}}$) vectors. The angle between $\overrightarrow{p_t^{12}}$ and $\overrightarrow{v_t^{12,\prime}}$ determines the symmetry of the SMG. In the case of perfect symmetry ($v_t^{1,\prime} \approx v_t^{2,\prime}$), we can easily show using vector geometry that $\overrightarrow{p_t^{12}}$ and $\overrightarrow{v_t^{12,\prime}}$ will be nearly perfectly aligned (parallel) in which case the angle between them will be approximately 0 implying the dot product will be nearly 1. The dot product decreases as the symmetry decreases (as $v_t^{1,\prime}$ and $v_t^{2,\prime}$ diverge). The ϵ in the denominator is the same

ϵ used in Theorem 2 which represents the practicality of the fact that there always exists a small difference between $v_t^{1,\prime}$ and $v_t^{2,\prime}$, no matter how close they may be. We now present the following theorem

Theorem 3 *Consider the social mini-game, $\mathcal{S}$, defined in Theorem 2. Then at time t,*

$$\mathcal{S} \text{ exists} \iff \ell_j(p_t^i, v_t^i) = \ell_i(p_t^j, v_t^j) \leq \ell_{\text{thresh}} \quad (15)$$

where $\ell_{\text{thresh}} := \frac{\pi}{4} - \tan^{-1} \frac{1}{2}$.

Proof We prove the forward direction first. Using basic trigonometry, the scaling factor ζ can be used to determine the threshold ℓ_{thresh} as shown in Fig. 2b. Formally,

$$\ell_j = \frac{\pi}{4} - \tan^{-1} \frac{1}{\zeta} \quad (16)$$

Using Theorem 2, in the limit $\lim_{d \to l, \epsilon \to 0} \zeta = 2 \implies \ell_j = \ell_i = \frac{\pi}{4} - \tan^{-1} \frac{1}{2}$. Now we prove the backward direction. Assume $\ell_j = \ell_i \leq \frac{\pi}{4} - \tan^{-1} \frac{1}{2}$. We can construct a scenario S geometrically similar to $\mathcal{S}$. It will be equivalent to $\mathcal{S}$ when we can show that one robot will not be able to clear $\mathcal{Q}$ in the time when the other agent reaches $\mathcal{Q}$ (by Definition 1) which happens when $\zeta \leq 2$ (Theorem 2). As $\ell_j = \ell_i \leq 0.3 \implies \zeta \leq 2$ is true, S is equivalent to $\mathcal{S}$. □

We empirically verify in simulation that this threshold, $\ell_{\text{thresh}} \approx 0.3$, in fact, works well in practice, as shown in Fig. 11. Recall that social mini-games often yield a deadlock due to their geometrically symmetric configurations (Grover et al., 2020). Theorem 3 gives us a way to monitor for social mini-games and therefore, deadlocks.

Corollary 1 *Deadlock Detection: A system of 2 robots may result in a deadlock if $\ell_j(p_t^i, v_t^i) = \ell_i(p_t^j, v_t^j) \leq \ell_{\text{thresh}}$, where ℓ_{thresh} is defined as in Theorem 3.*

Proof The proof follows from Theorem 3.   □

Equivalence of Equations (16) ***and*** (14): It is important to observe that Theorem 3 and Definition 5 are equivalent. Specifically, we will show that Equations (16) and (14) are equivalent. First, from Definition 5, assuming W.L.O.G that $\theta < \frac{\pi}{4} \implies \theta + \theta' = \frac{\pi}{4}$, the liveness function can be equivalently written as,

$$
\begin{aligned}
\ell_j &= \cos^{-1}\left(\frac{\cos(\theta + \theta') \cdot \left(1 + \frac{1}{\zeta}\right)}{\sqrt{1 + \frac{1}{\zeta^2} + \frac{2}{\zeta}\cos\left(2\left(\theta + \theta'\right)\right)}} \right) \\
&= \cos^{-1}\left(\sqrt{\frac{1}{2} + \frac{\zeta}{\zeta^2 + 1}} \right)
\end{aligned}
\tag{17}
$$

We would substitute $\cos(\theta - \theta')$ if $\frac{\pi}{4} \leq \theta < \frac{\pi}{2}$. Let $\alpha = \cos^{-1}\left(\sqrt{\frac{1}{2} + \frac{\zeta}{\zeta^2 + 1}} \right)$, then $\cos(\alpha) = \sqrt{\frac{1}{2} + \frac{\zeta}{\zeta^2 + 1}}$. Set $\beta = \tan^{-1}\left(\frac{1}{\zeta}\right)$. It follows that:

$$
\begin{aligned}
\tan(\beta) &= \frac{1}{\zeta}, \\
\cos(\beta) &= \frac{\zeta}{\sqrt{\zeta^2 + 1}}, \\
\sin(\beta) &= \frac{1}{\sqrt{\zeta^2 + 1}}.
\end{aligned}
\tag{18}
$$

Further, using the trigonometric identity, $\cos\left(\frac{\pi}{4} - \beta\right) = \frac{\cos\beta + \sin\beta}{\sqrt{2}}$, we get:

$$
\begin{aligned}
\cos\left(\frac{\pi}{4} - \beta\right) &= \frac{\frac{\zeta}{\sqrt{\zeta^2+1}} + \frac{1}{\sqrt{\zeta^2+1}}}{\sqrt{2}} = \frac{\frac{\zeta+1}{\sqrt{\zeta^2+1}}}{\sqrt{2}} \\
&= \sqrt{\frac{1}{2} + \frac{\zeta}{\zeta^2 + 1}} = \cos(\alpha).
\end{aligned}
\tag{19}
$$

or $\frac{\pi}{4} - \beta = \alpha$. Therefore,

$$
\begin{aligned}
\cos^{-1}&\left(\frac{\cos(\theta + \theta') \cdot \left(1 + \frac{1}{\zeta}\right)}{\sqrt{1 + \frac{1}{\zeta^2} + \frac{2}{\zeta}\cos\left(2\left(\theta + \theta'\right)\right)}} \right) = \frac{\pi}{4} - \beta \\
&= \frac{\pi}{4} - \tan^{-1}\left(\frac{1}{\zeta}\right)
\end{aligned}
\tag{20}
$$

And we already have that,

$$
\begin{aligned}
\ell_j(p_t^i, v_t^i) &= \cos^{-1}\left(\frac{\left\langle \overrightarrow{p_t^i - p_t^j}, \overrightarrow{v_t^{i,'} - v_t^{j,'}} \right\rangle}{\left\| \overrightarrow{p_t^i - p_t^j} \right\| \left\| \overrightarrow{v_t^{i,'} - v_t^{j,'}} \right\| + \epsilon} \right) \\
&= \cos^{-1}\left(\frac{\cos(\theta + \theta') \cdot \left(1 + \frac{1}{\zeta}\right)}{\sqrt{1 + \frac{1}{\zeta^2} + \frac{2}{\zeta}\cos\left(2\left(\theta + \theta'\right)\right)}} \right)
\end{aligned}
\tag{21}
$$

Therefore, finally,

$$
\begin{aligned}
\ell_j(p_t^i, v_t^i) &= \cos^{-1}\left(\frac{\left\langle \overrightarrow{p_t^i - p_t^j}, \overrightarrow{v_t^{i,'} - v_t^{j,'}} \right\rangle}{\left\| \overrightarrow{p_t^i - p_t^j} \right\| \left\| \overrightarrow{v_t^{i,'} - v_t^{j,'}} \right\| + \epsilon} \right) \\
&= \frac{\pi}{4} - \tan^{-1}\left(\frac{1}{\zeta}\right)
\end{aligned}
\tag{22}
$$

In other words, we showed that Equation (14) in Definition 5 is equivalent to Equation (16) in Theorem 3.

5.2 Deadlock prevention

From Theorem 3, we can derive the following lemma,

Lemma 1 *If at time t, $x_t^i \in \widetilde{\Gamma}^i$ and $\ell_j\left(x_t^i, v_t^i\right) \geq \ell_{\text{thresh}}$, then agent i is <u>not</u> in a deadlock with agent j, where $j \in \mathcal{N}^i\left(x_t^i\right)$.*

Proof From Theorem 3, a social mini-game implies $\ell_j\left(x_t^i, v_t^i\right) < \ell_{\text{thresh}}$. Therefore, by the contrapositive, a preventive solution can be devised by perturbing both robots such that $\ell_j\left(x_t^i, v_t^i\right) \geq \ell_{\text{thresh}}$, thereby resolving the social mini-game. If there is no social mini-game occurring at time t, then by Definition 1, there exists some $\widetilde{\Gamma}^i$ that is not in conflict with any other agent j and therefore, collision-free. By our assumption of small-time controllability of (1) and Theorem 1, there exists an admissible, and collision-free, path between configurations in $\mathcal{X}_I$ and $\mathcal{X}_G$.   □

We propose a perturbation strategy that can be integrated as a constraint (CBF constraint for double-integrator or higher order dynamics) and combined with existing controllers, such as MPC or DWA, to resolve deadlocks. Our decentralized perturbation strategy is designed to be minimally invasive.

In particular, we design a perturbation that acts on v_t^i, and not on p_t^i, through the notion of *liveness sets*, which are analogous to the safety sets in CBFs. Formally,

Definition 6 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, $\mathscr{C}_\ell(t) \subseteq \mathbb{R}^k$ 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 $i, j, \zeta \geq 2$.

As $v_t^i \geq \zeta v_t^j \equiv v_t^j \geq \zeta v_t^i$, we can permute the order of agents as $k!$, each resulting in a different convex hull. We take the union of these convex hulls to generate $\mathscr{C}_\ell(t)$.

Corollary 2 If $x_t^i \in \widetilde{\Gamma}^i$, then $v_t \in \mathscr{C}_\ell(t)$ guarantees that the system of robots are deadlock-free.

Proof From Definition 6, since $v_t^i \in \mathscr{C}_\ell(t)$, we have $v_t^i \geq \zeta v_t^j$. According to Theorem 2, we select $\zeta \geq 2$. Then, according to Theorem 3, $\zeta \geq 2 \implies \ell_j = \ell_i \geq \frac{\pi}{4} - \tan^{-1} \frac{1}{2}$. Finally, by Lemma 1, $\ell_j \geq \frac{\pi}{4} - \tan^{-1} \frac{1}{2}$ implies that robot i is not in a deadlock, *i.e.*, $\left\{ x_t^i, u_t^i \notin \mathcal{D}^i(t) \right\}$.   □

From Corollary 2, if each v_t^i is such that the joint velocity $v_t \in \mathscr{C}_\ell(t)$, 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 on to the nearest point in $\mathscr{C}_\ell(t)$ via,

$$\widetilde{v}_t = \arg \min_{\mu \in \mathscr{C}_\ell(t)} \| v_t - \mu \|_2 \tag{23}$$

Theorem 4 *A solution to the optimization problem (23) always exists and is unique if $v_t^i \neq v_t^j$ for any robots i and j with $i \neq j$.*

Proof The set $\mathscr{C}_\ell(t)$ can be described as the complement of an open convex polytopic set P whose boundary ∂P is a subset of $\mathscr{C}_\ell(t)$, that is, $\partial P \subseteq \mathscr{C}_\ell(t)$ and in particular, $\partial P \subseteq \partial \mathscr{C}_\ell(t)$. Existence of a minimum distance projection from any (interior) point of the (open) polytopic set P to its boundary $\partial P \subseteq \partial \mathscr{C}_\ell(t)$ is trivially satisfied given as the set of edges, $\mathcal{K}$, that determine ∂P is a finite set, which always yields a minimum. Formally, the Euclidean distance function $q(y) = \| x - y \|$, from a point $y \in \partial P$ to a given point $x \in \text{int}(P)$ is a continuous function, which always attains its minimum in ∂P which is a compact (closed and bounded) set comprising a finite number of edges.

Next, we will show that the minimal distance projection to $\partial P \subseteq \partial \mathscr{C}_\ell(t)$ is unique. First, we note that the projection of a point to an edge always exists and is unique (the latter projection either corresponds to the minimal distance projection to the line containing the line segment, if the latter projection belongs to the line segment, or one of the endpoints of the segment, otherwise). Therefore, there exists a unique minimizer, d_k, to the edge $k \in \mathcal{K}$. By extension, there exists a unique minimizer $d_j, j \neq k \in [\mathcal{K}]$ to each edge in $\mathcal{K}$. Define the set $\mathcal{D} = \left\{ d_j, j \neq k \in [\mathcal{K}] \right\}$. As the set $\mathcal{D}$ is finite, a minimizer will always exist. To show uniqueness, let us assume on the contrary that there exist $j, k \in [\mathcal{K}]$ with
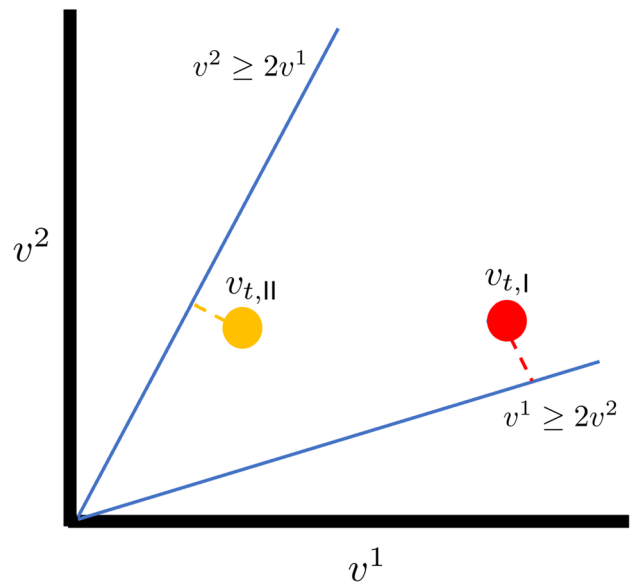

Fig. 5 Liveness set $\mathscr{C}_\ell(t)$ for the 2 robot scenario: Velocities for two deadlocked agents are shown by the red point $v_{t,I}$. To resolve the deadlock, Equation (23) is used to project $v_{t,I}$ onto $v_t^1 = 2v^2$. Robot 1 increases its velocity component v^1, while robot 2 decreases its velocity component v^2 to align with the barrier. If robot 1 deviates from $v_{t,I}$ and decreases its speed to the new yellow point $v_{t,II}$, the new optimal perturbation will be onto $v_t^2 = 2v^1$. Thus, robot 2 adjusts its strategy by increasing its speed to align with a new perturbation. Assuming no speed deviations, there will be a unique projection to one of the safety barriers (Color figure online)

$j \neq k$ such that $d_j = d_k$ (this can, for instance, happen if when P is regular and x_c is the center of P, which is where agents have identical velocities). However, $d_j = d_k$ implies that $v_t^j = v_t^k$, which contradicts our assumption that $v_t^i \neq v_t^j$ for any robots i and j with $i \neq j$.   □

Theorem 5 *The deadlock-preventing control strategy for carrying out Eq. 23 is minimally invasive.*

Proof The control strategy for carrying out the perturbation in Equation (23) to modify the speed of a robot does not affect the rotation of the robot ($\Delta \theta_t^i = 0$), therefore ensuring that the robot tracks the current trajectory. This proves condition 1 of Definition 4.

To prove that the deadlock-preventing control, say Δv_t, is indeed δ_{opt}, as defined by the second optimality criteria condition in Eq. 12 of Definition 4, we need to show that Δv_t satisfies Eqs. 12a, 12b, 12c, and 12d.

Equation 12a. We have to show that Δv_t is the smallest change in v_t. This follows directly from the geometry of liveness sets in Figure 5, where Δv_t corresponds to the perpendicular projection from v_t onto the boundary of the liveness set.

Equation 12b. This is also straightforward from basic vector addition, considering the triangle formed with v_t as the hypotenuse and Δv_t as the base in Figure 5, which gives

v_{t+1} as the perpendicular along the boundary of the liveness set.

Equation 12c. As we consider single- and double-integrator dynamical systems with unconstrained control inputs, which are fully actuated systems, there always exist valid controls to achieve a desired configuration. This follows directly from properties of fully actuated systems (c.f. Chapter 1, Tedrake (2023)).

Equation 12d. The final constraint to satisfy states that the resulting state x_{t+1}^i is deadlock-free, that is, $\left(x_{t+1}^i, u_{t+1}^i\right) \notin \mathcal{D}^i(t+1)$. Suppose after applying Δv_t, $v_{t+1} \in \mathscr{C}_\ell(t+1)$. By Corollary 2, robots are deadlock-free since $v_{t+1} \in \mathscr{C}_\ell(t+1)$, $v_t^i \geq \zeta v_t^j$ (from Definition 6). According to Theorem 2, we select $\zeta \geq 2$. Then, according to Theorem 3, $\zeta \geq 2 \implies \ell_j = \ell_i \geq \frac{\pi}{4} - \tan^{-1} \frac{1}{2}$. Finally, by Lemma 1, $\ell_j \geq \frac{\pi}{4} - \tan^{-1} \frac{1}{2}$ implies that robot i is not in a deadlock with robot j, i.e., $\left\{x_{t+1}^i, u_{t+1}^i \notin \mathcal{D}^i(t+1)\right\}$. $\qquad \square$

Example 1 (two agents): While our approach generalizes to k robots, we consider an example with 2 robots for simplicity. In the 2 robot scenario where each robot is equidistant from a doorway or intersection, the liveness set $\mathscr{C}_t^\ell$, shown in Fig. 5, is generated by scaling v_t^2 by $\frac{v_t^1}{\zeta}$, or vice-versa. We can then generate the following system of linear inequalities,

$$v_t^1 \geq \zeta v_t^2$$
$$v_t^2 \geq \zeta v_t^1$$

This can be compactly represented as

$$A_{2\times2} v_t \geq 0 \tag{24}$$

where $A_{2\times2} = \begin{bmatrix} 1 & -\zeta \\ -\zeta & 1 \end{bmatrix}$ and $v_t = \left[v_t^1, v_t^2\right]^\top$. Suppose the current value of v_t is p_1 as shown in Fig. 5. The point p_1 indicates that $A_{2\times2} p_1 < 0$ which lies outside $\mathscr{C}_\ell(t)$, implying that the two robots are in a deadlock according to Definition 6 and Definition 3. Equation (23) projects p_1 onto the nearest half-plane which is the $v^1 = \zeta v^2$ barrier. Thus, robot 1 will increase v^1 and robot 2 will decrease v^2 by projecting p_1 down on $v^1 = \zeta v^2$. This projection is the minimal deviation required on the part of both robots.

Example 2 (three agents): We can extend this to 3 robots with speeds v_t^1, v_t^2, v_t^3. Assuming W.L.O.G that $v^1 > v^2 > v^3$, We found empirically that scaling v^2 by $\frac{v^1}{2}$, and v^3 by $\frac{v^1}{3}$ generates the $\mathscr{C}_\ell(t)$ for 3 agents attempting to pass through a doorway from an equidistant. We can generate the following system of linear inequalities,

$$v_t^1 \geq 2v_t^2$$
$$v_t^1 \geq 3v^3 \tag{25}$$
$$3v_t^2 \geq 2v^3$$

This can be compactly represented as $A_{3\times3} v_t \geq 0$ where $A_{3\times3} = \begin{bmatrix} 1 & -2 & 0 \\ 1 & 0 & -3 \\ 0 & 3 & -2 \end{bmatrix}$ and $v_t = \left[v_t^1, v_t^2, v_t^3\right]^\top$. Note that there will be 6 possible permutations of A, resulting in 6 tetrahedrons within a cube. Given a point $p = \left[v^1, v^2, v^3\right]^\top$ floating in this cube that does not lie within any of the 6 tetrahedrons, then the optimal perturbation strategy is to project p onto the face of the nearest tetrahedron.

$$\left(\Gamma^{i,*}, \Psi^{i,*}\right) = \arg \min_{(\Gamma^i, \Psi^i)} \sum_{t=0}^{T-1} \mathcal{J}^i\left(x_t^i, u_t^i\right) + \mathcal{J}_T^i\left(x_T^i\right) \tag{26a}$$

$$\text{s.t } x_{t+1}^i = f\left(x_t^i\right) + g\left(x_t^i\right) u_t^i, \quad \forall t \in [1; T-1] \tag{26b}$$

$$\Delta h^i\left(x_t^i, u_t^i\right) \geq -\gamma h^i\left(x_t^i\right) \tag{26c}$$

$$x_T^i \in \mathcal{X}_g \tag{26d}$$

where constraints (26b) and (26c) must be enforced at all times. Furthermore,

$$h^i\left(x_t^i\right) = \left[h_s^i\left(x_t^{i,1}\right), \ldots, h_s^i\left(x_t^{i,k-1}\right), h_v\left(x_t\right)\right]^\top, \tag{27a}$$

$$h_s^i\left(x_t^{i,j}\right) = \left\|p_t^i - p_t^j\right\|_2^2 - r^2, \forall \quad j \in [1; k] \setminus i \tag{27b}$$

$$h_v\left(x_t\right) = \bar{A}_{k\times4k}\left(x_t\right) \tag{27c}$$

where $h_s^i(x_t^{i,j})$ represents the CBF for the agent i which ensures that agent i does not collide with agent j by maintaining a safety margin distance of at least r.

The program (26) may not be feasible for certain input-constrained systems, which we have discussed in Section 5.6. For unconstrained systems, feasibility for the program in (26) can be discussed using the definition of Control Barrier Functions (CBFs) in Equation (8). In other words, Theorem 3 in Ames et al. (2019) states that if there exists a valid CBF $h^i\left(x_t^i\right)$ that satisfies (4) and a $\mathcal{K}_\infty$ function κ, then (8) holds true. If (8) holds true, then that implies that there must exist a control input u_t^i belonging to the compact set U^i that satisfies (8).

In this approach, we can construct a valid CBF apriori (typical of the kind of problem formulation under investigation Wang et al. (2017)), and following (8), there always exists a control input u_t^i that satisfies condition (20c). In general, however, we acknowledge that computing a valid CBF remains a challenging problem. While previous research has employed neural networks (Robey et al., 2020) and Gaussian Processes (Jagtap et al., 2020) to learn CBFs, guaranteeing the validity of learned or analytically derived functions as CBFs is still an open research question.

Further, $h_v(x_t) = \bar{A}_{k \times k}(x_t)$ is the expanded form of Eqs. 24 and 25 where $x_t = \left[p_t^1, v_t^1, \theta_t^1, \omega_t^1, p_t^2, v_t^2, \theta_t^2, \omega_t^2 \right]^\top$ and controls, $u_t = \left[u_t^1, u_t^2 \right]^\top$. Following the 2 agent example, we can expand the matrix $A_{2 \times 2}$ as $\bar{A}_{2 \times 8} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & -\zeta & 0 & 0 \\ 0 & -\zeta & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ to accommodate the aggregate of both the robots' states and controls. The weights or entries in the A matrix control the sensitivity to deadlocks. Specifically, tuning the matrix towards an identity matrix ($\zeta \longrightarrow 1$) decreases the sensitivity–the robots do not perturb their velocities and the robots would either end up in a deadlock or collide (if they were moving fast enough). On the other hand, increasing the relative weights ($\zeta \gg 0$) makes the robots increasingly sensitive to deadlocks–that is, the robots would tend to react to false positive scenarios (scenarios that might be tending towards a deadlock, but are not strictly so).

For cases of single integrator dynamics where v_t^i is a input, the constraint $A v_t \geq 0$ (where $v_t = [v_t^1, \ v_t^2]^T$ for two agents) cannot be directly incorporated as a CBF, instead (27c) is invoked after the MPC selects an optimal control. However in cases of double or higher-order robotic systems (such as bipedal robots (Grizzle et al., 2014), Boston Dynamics Spot etc.), since v_t^i is also a state, guaranteeing the invariance of the set $A_{k \times k} v_t \geq 0$ becomes trivial by modeling (27c) as a CBF constraint.

Remark Despite the inherent non-convexity of the MPC-CBF program (26) due to the collision constraint (27b), our decentralized strategy converges to a local solution within each planning window. We elevate a conventional local solution by endowing it with critical safety and liveness attributes through our CBF constraint (27c), thereby rendering the solution not only viable but preferable. This strategy represents a substantial improvement over methods that necessitate convergence to a static global solution–a process often marked by inefficiency and impracticality due to the complexities associated with computing global solutions.

Remark For $v_t^i \neq v_t^j$ for all i, j, there is always a unique perturbation. Consider in Example 1 that robot 1 decides to deviate from its current speed in $v_{t,1}$ and decides to decrease its speed, shown by the new point $v_{t,2}$. In that case, robot 2's optimal strategy will no longer be to decrease its speed as before. Now, the nearest safety barrier becomes $v_t^2 = \zeta v^1$, and to project $v_{t,2}$ to this barrier, robot 2 will instead increase its speed. Therefore, assuming a robot does not deviate from its current speed, there will be a unique projection to one of the safety barriers.

5.3 Tie-breaking via priority orderings

If $v_t^i = v_t^j$ for some i, j, then there are multiple solutions and we implement the following tie breaking protocol. In social mini-games, we define a conflict zone as a region φ in the global map that overlaps goals corresponding to multiple robots. A conflict, then, is defined by the tuple $\langle \mathcal{C}_\varphi^t, \varphi, t \rangle$, which denotes a conflict between robots belonging to the set $\mathcal{C}_\varphi^t$ at time t in the conflict zone φ in $\mathcal{G}$. Naturally, robots must either find an alternate non-conflicting path or must move through φ according to a schedule informed by a priority order. A priority ordering is defined as follows,

Definition 7 Priority Orderings (σ): A priority ordering is a permutation $\sigma : \mathcal{C}_\varphi^t \rightarrow [1, k]$ over the set $\mathcal{C}_\varphi^t$. For any $i, j \in [1, k]$, $\sigma^i = j$ indicates that the ith robot will move on the jth turn with $\sigma^{-1}(j) = i$.

For a given conflict $\langle \mathcal{C}_\varphi^t, \varphi, t \rangle$, there are $\left| \mathcal{C}_\varphi^t \right|$ factorial different permutations. There exists, however, an *optimal* priority ordering, σ_{OPT}.

Definition 8 Optimal Priority Ordering (σ_{OPT}): A priority ordering, σ, over a given set $\mathcal{C}_\varphi^t$ is optimal if bidding $b^i = \zeta^i$ is a dominant strategy and maximizes $\sum_{|\mathcal{C}_\varphi^t|} \zeta^i \alpha^i$, where ζ^i is a private incentive or priority parameter known only to agent i, and α_q is a time-based reward for receiving an order position $\sigma^i = q$.

We run an auction, (r^i, p^i), with an allocation rule $r^i(b^i) = \sigma^i = q$ and payment rule p^i defined by $p^i(b^i) = \sum_{j=q}^{|\mathcal{C}_\varphi^t|} \widehat{b}^{\sigma^{-1}(j+1)} (\alpha_j - \alpha_{j+1})$. The payment rule is the "social cost" of reaching the goal ahead of the robots arriving on turns $q + 1, q + 2, \ldots, q + \mathcal{C}_\varphi^t$. The bids, $\widehat{b}^{\sigma^{-1}(q+1)}, \widehat{b}^{\sigma^{-1}(q+2)}, \ldots, \widehat{b}^{\sigma^{-1}\left(q + |\mathcal{C}_\varphi^t|\right)}$ represent proxy robot bids sampled from a uniform distribution, since robots do not have access to the bids of other robots. Using (r^i, p^i) defined as above, each robot solves

$$b^{i,*} = \arg \min_{b^i} \left(\alpha_{r^i(b^i)} - p^i(b^i) \right) \tag{28}$$

It is known Roughgarden (2016); Chandra and Manocha (2022) that the auction defined by (r^i, p^i) yields $b^{i,*} = \zeta^i$ and maximizes $\sum_{|\mathcal{C}_\varphi^t|} \zeta^i \alpha^i$. To summarize the algorithm, the robot with the highest bid, that is, the highest incentive ζ^i, is allocated the highest priority and is allowed to move first, followed by the second-highest bid, and so on.
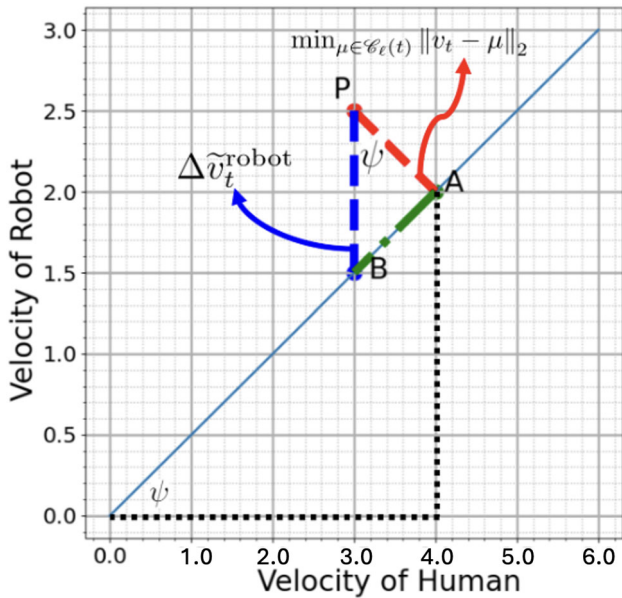
Fig. 6 Velocity perturbation for robot in human environments corresponding to the case when the robot slows down in response to the human (**blue line**). The point P corresponds to the current joint velocity of the robot and the human. The length $\overrightarrow{PB}$ corresponds to the velocity difference, $\Delta \widetilde{v}_t^{robot}$, and $\overrightarrow{PA}$ corresponds to the solution obtained via Equation (23) given by $\min_{\mu \in \mathscr{C}_\ell(t)} \|v_t - \mu\|_2$

Remark The tie breaking protocol via auctions is not decentralized as there is an assumption of an auction program that receives bids and distributes the ordering among the agents. Since control is still decentralized, we consider this a distributed optimization problem instead.

5.4 Application in human environments: relaxing the perturbation

Humans are unpredictable and often make irrational decisions; it is impractical to assume that humans will precisely execute the perturbation strategy in Equation (23). By shifting the responsibility for ensuring $v_t \in \mathscr{C}_\ell(t)$ entirely to the robot, we can still guarantee deadlock prevention in human environments, albeit at the cost of trading away minimal invasiveness. More specifically, assume that the human does not perturb their velocity. Then, the robot faces two options.

If $v^{robot} < v^{human}$, then the robot will decrease its speed such that $\zeta v^{robot} \leq v^{human}$. Formally

$$\Delta \widetilde{v}_t^{robot} = \frac{\min_{\mu \in \mathscr{C}_\ell(t)} \|v_t - \mu\|_2}{\cos \psi} \tag{29}$$

where $\psi = \tan^{-1} \zeta$. We visualize this derivation geometrically in Figure 6. ψ represents the slope of the boundary lines of the liveness set, which could be either ζ or $\frac{1}{\zeta}$, which is shown in Figure 6 as the angle subtended by the black dot-

ted lines. Since the liveness set is symmetric, we can select either slope without loss of generality. Thus, $\psi = \tan^{-1}(\zeta)$. When the robot slows down in response to the human (vertical **blue** line), as shown in Figure 6, it is trivial to see that the angle between the **blue** and the **red** lines is also ψ. From trigonometry, we obtain Equation (29).

If $v^{robot} > v^{human}$, then the robot will increase its speed such that $v_{limit}^{robot} \geq v^{robot} \geq \zeta v^{human}$. Formally,

$$\Delta \widetilde{v}_t^{robot} = \min \left\{ \frac{\min_{\mu \in \mathscr{C}_\ell(t)} \|v_t - \mu\|_2}{\sin \psi}, v_{limit}^{robot} - v^{robot} \right\} \tag{30}$$

where $\psi = \tan^{-1} \zeta$. We can similarly derive Equation (30) which represents the alternative scenario when the robot speeds up in response to the human. Here the difference is that the **blue** line will now be a *horizontal* line from P, intersecting the boundary of the liveness set hyperplane. Using basic euclidean geometry, it is easy to see that the $\cos \psi$ in the formula will change to $\sin \psi$. We apply the minimum operator because we have to take into consideration the physical upper limits of the robots speed.

It is easy to observe that this is a relaxation since the perturbation is larger for the robot in this case. Let $\nu = \min_{\mu \in \mathscr{C}_\ell(t)} \|v_t - \mu\|_2$. Then,

$$\frac{\min_{\mu \in \mathscr{C}_\ell(t)} \|v_t - \mu\|_2}{\cos \psi} \geq \nu$$

as well as

$$\min \left\{ \frac{\min_{\mu \in \mathscr{C}_\ell(t)} \|v_t - \mu\|_2}{\sin \psi}, v_{limit}^{robot} - v^{robot} \right\} \geq \nu.$$

Here, the relaxation refers to the trading away of minimal invasiveness by shifting of the responsibility for ensuring $v_t \in \mathscr{C}_\ell(t)$ entirely to the robot.

5.5 Safety analysis for perturbation via equation (23)

Finally, we discuss the safety of the control strategy necessary to carry out the perturbation in Equation (23). In double-integrator (or higher order) dynamical systems, the constraints (24) and (25) can be incorporated as CBFs. Formally, we generate the liveness set $\mathscr{C}_\ell(t)$ as,

$$\mathscr{C}_\ell(t) = \{v_t \text{ s.t. } h_v(x_t) \geq 0\}$$
$$h_v(x_t) = \bar{A}_{k \times k}(x_t) \tag{31}$$

where $h_v(x_t)$ is the CBF associated with the safe set $\mathscr{C}_\ell(t)$ that can be used to ensure the forward invariance of $\mathscr{C}_\ell(t)$, $x_t = \left[p_t^1, v_t^1, \theta_t^1, \omega_t^1, p_t^2, v_t^2, \theta_t^2, \omega_t^2 \right]^\top$ and controls, $u_t =$
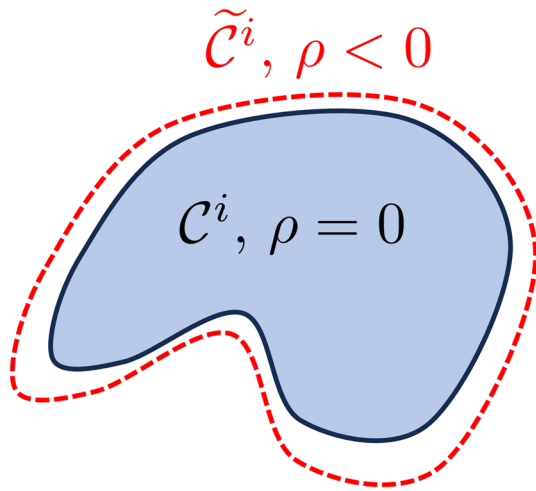
Fig. 7 Visualizing the concept of robust control barrier functions (RCBFs) used in single-integrator dynamics to ensure safety despite perturbation via Equation (23). The blue area represents the safety region $\mathcal{C}^i$ with no input uncertainty ($\rho = 0$). When input uncertainty $\rho \leq 0$ is applied, this safe region is bounded by a red dashed line to expand the safe set to $\widetilde{\mathcal{C}}^i$ (Color figure online)

$\left[u_t^1, u_t^2\right]^\top$ (Fig. 7). Following the 2 agent example, we can expand the matrix $A_{2\times 2}$ as $\bar{A}_{2\times 8} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & -\zeta & 0 & 0 \\ 0 & -\zeta & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ to accommodate the aggregate of both the robots' states and controls. The CBF $h_v(x_t)$ is then integrated as constraint in the optimization (26) as follows,

$$\Delta h^i\left(x_t^i, u_t^i\right) \geq -\gamma h^i\left(x_t^i\right) \tag{32}$$

where $h^i\left(x_t^i\right) = \left[h_s^i\left(x_t^{i,1}\right), \ldots, h_s^i\left(x_t^{i,k-1}\right), h_v\left(x_t\right)\right]^\top$. This guarantees that the control strategy necessary to ensure that $v_t \in \mathcal{C}_\ell(t)$ does not escape the safe set.

However, for single-integrator dynamics, we can use robust CBFs to guarantee safety for the perturbation control strategy. Assume that δu_t^i represents the perturbation necessary to achieve the velocity governed by Eq. 23. The system dynamics with uncertainty in the input is described by:

$$x_{t+1}^i = f\left(x_t^i\right) + g\left(x_t^i\right)\left(u_t^i + \delta u_t^i\right),$$

where δu_t^i represents the input uncertainty bounded by $\left|\delta u_t^i\right| \leq |p|$ for a known bound p. Then,

$$
\begin{aligned}
\Delta h^i\left(x_t^i\right) &= h\left(x_t^{i+1}\right) - h\left(x_t^i\right) \\
&= h\left(\left(x_t^i\right) + g\left(x_t^i\right)\left(u_t^i + \delta u_t^i\right)\right) - h\left(x_t^i\right) \\
&= h\left(\left(x_t^i\right) + g\left(x_t^i\right)\left(u_t^i + \delta u_t^i\right)\right) \\
&\quad - h\left(\left(x_t^i\right) + g\left(x_t^i\right)\left(u_t^i\right)\right) h\left(x_t^i\right)
\end{aligned}
$$

$$
\begin{aligned}
&+ h\left(\left(x_t^i\right) + g\left(x_t^i\right)\left(u_t^i\right)\right) h\left(x_t^i\right) - h\left(x_t^i\right) \\
&= \Delta_{\text{original}} h^i\left(x_t^i\right) + \Delta_{\text{modification}} h^i\left(x_t^i, u_t^i\right)
\end{aligned}
\tag{33}
$$

where

$$
\begin{aligned}
\Delta_{\text{modification}} h^i\left(x_t^i, u_t^i\right) &= \left(h\left(\left(x_t^i\right) + g\left(x_t^i\right)\left(u_t^i + \delta u_t^i\right)\right)\right. \\
&\quad \left. - h\left(\left(x_t^i\right) + g\left(x_t^i\right)\left(u_t^i\right)\right)\right), \\
\Delta_{\text{original}} h^i\left(x_t^i\right) &= \left(h\left(\left(x_t^i\right) + g\left(x_t^i\right)\left(u_t^i\right)\right) - h\left(x_t^i\right)\right).
\end{aligned}
$$

The above no longer satisfies the condition (7). In the literature, this problem is often addressed by adding a compensation term to the barrier function $h^i\left(x_t^i\right)$ such that the superlevel set $\mathcal{C}^i$ expands to accommodate the perturbation. The resulting barrier function belongs to the class of Robust CBFs or RCBFs (Alan et al., 2023). An RCBF implies the following holds for all $t \geq 0$:

$$\Delta h^i\left(x_t^i\right) - \rho\left(x_t^i\right) \geq -\gamma h^i\left(x_t^i\right). \tag{34}$$

for some $\rho : \mathcal{X} \times \mathcal{U} \longrightarrow \mathbb{R}$. We can derive an expression for ρ in terms of p to ensure the robustness of the control barrier function. Given $\left|\delta u_t^i\right| \leq |p|$, the worst-case impact occurs when δu is aligned with $g\left(x_t^i\right)\left(u_t^i\right)$ in the direction that most negatively affects $h^i\left(x_t^i\right)$. Thus, ρ must compensate for the maximum possible deviation in $\Delta_{\text{modification}} h^i\left(x_t^i, u_t^i\right)$ due to δu_t^i. Mathematically, this can be expressed as:

$$\rho\left(x_t^i\right) = \max_{\left|\delta u_t^i\right| \leq |p|}\left|\Delta_{\text{modification}} h^i\left(x_t^i, u_t^i\right)\right|. \tag{35}$$

Given that δu_t^i is bounded by $|p|$, the term $\left|\Delta_{\text{modification}} h^i\left(x_t^i, u_t^i\right) \delta u_t^i\right|$ can be maximized by taking δu_t^i at its bound. The linear approximation of this impact, assuming the worst-case alignment, leads to:

$$\rho\left(x_t^i\right) \leq \left|\Delta_{\text{modification}} h^i\left(x_t^i, u_t^i\right)\right| |p|.$$

Here, $\left|\Delta_{\text{modification}} h^i\left(x_t^i, u_t^i\right)\right|$ represents the norm (or a suitable measure of magnitude) of the Lie derivative of h along g, indicating how changes in the input directly influence the safety condition through $g(x)$. The term $|p|$ represents the maximum magnitude of the uncertainty.

5.6 Dealing with input constraints

Integral Control Barrier Functions (ICBFs) (Zinage et al., 2024) ensure safety under limited actuation for multi-agent systems by treating the control input as an auxiliary state.

With an ICBF, the concept of a safety set extends the traditional state-based safety to the following definition: $\mathscr{S} := \{(x_t, u_t) \in \mathcal{C} \times \mathcal{U}\}$, where $\mathcal{C}$ represents a state-based safety set and input constraints in $\mathcal{U}$. We can handle input constraints by using ICBFs in place of the traditional CBF.

Now, consider a continuously differentiable scalar-valued function $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ defined such that $h(x_t, u_t) > 0$ if $(x_t, u_t) \in \mathscr{S}$, $h(x_t, u_t) = 0$ if $(x_t, u_t) \in \partial\mathscr{S}$ and $h(x_t, u_t) < 0$ if $(x_t, u_t) \in (\mathbb{R}^n \times \mathbb{R}^m) \backslash \mathscr{S}$. To guarantee that $(x_t, u_t) \in \mathscr{S}$, $h(x_t, u_t)$ must satisfy the forward invariance condition i.e. $\dot{h}(x_t, u_t) + \alpha(h(x_t, u_t)) \geq 0$ where $\dot{h}(x_t, u_t)$ is computed along the system trajectories and α is a $\mathcal{K}_\infty$ function. Formally, ICBF is defined as follows:

Definition 9 (ICBF) (Ames et al., 2020) An Integral Control Barrier Function (ICBF) is defined as a function $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ that characterizes a safe set $\mathscr{S}$. Then, h is said to be an ICBF, for all $(x_t, u_t) \in \mathscr{S}$ if $p(x_t, u_t) = 0$, implies that

$$q(x_t, u_t) \leq 0, \quad \text{where}$$
$$q(x_t, u_t) := - \big(\nabla_{x_t} h(x_t, u_t) f(x_t, u_t)$$
$$+ \nabla_{u_t} h(x_t, u_t) \phi(x_t, u_t)$$
$$+ \alpha(h(x_t, u_t)) \big),$$
$$p(x_t, u_t) := \big(\nabla_{u_t} h(x_t, u_t) \big)^\top .$$

The following theorem provides a method to synthesize safe controllers via ICBFs

Theorem 6 *(Ames et al., 2020) If an integral feedback controller $\phi(x_t, u_t)$ and a safety set $\mathscr{S}$, defined by an ICBF $h(x_t, u_t)$, exist, then modifying the integral controller to $\dot{u}_t = \phi(x_t, u_t) + v^\star(x_t, u_t)$, where $v^\star(x_t, u_t)$ is obtained by solving the following quadratic program (QP):*

$$v^\star(x_t, u_t) = \underset{v \in \mathbb{R}^m}{\arg\min} \|v\|^2, \quad s.t. \quad p(x_t, u_t)^\mathrm{T} v \geq q(x_t, u_t)$$

guarantees safety, i.e., forward invariance of the set $\mathscr{S}$.

5.7 Complete algorithm

The complete algorithm is presented as pseudocode in Algorithm 1. The first step involves initializing the starting and goal positions ($\mathcal{X}_I$ and $\mathcal{X}_g$, Line 1). The algorithm then sets key simulation parameters: total simulation time (T_{sim}), sampling time (T_s), and prediction horizon ($T_{horizon} = 3$) [Line 2]. Following this setup, it defines the cost matrices and actuator limits, including the velocity limit (v_{limit}) and angular speed limit (ω_{limit}) [Line 3]. The model defines the state and control variables, system dynamics based on the cost matrices (f, g), and stage-wise cost $\mathcal{J}^i$ [Lines 4-9]. Next, the

---

Algorithm 1: Safe and deadlock-free multi-agent navigation.

Result: Run MPC for trajectory tracking or setpoint control with obstacle avoidance

1 Initialize $\leftarrow \mathcal{X}_I, \mathcal{X}_g$;
2 Define simulation parameters: $T_{\text{sim}} = 3$, $T_s = 0.1$, $T_{\text{horizon}} = 3$;
3 Define cost matrices and actuator limits:
    $f = \text{diag}([11, 11, 0.005])$, $g = [1, 0.5]$, $v_{\text{limit}} = 0.3$,
    $\omega_{\text{limit}} = 3.8$;
4 **Function** `Model()`:
5   Define state and control variables;
6   Define system dynamics using f, g in (1);
7   Define stage-wise cost $\mathcal{J}^i$;
8   **return** model;
9 **Function** `MPC()`:
10   Call `Model()`;
11   Configure parameters using T_{sim}, T_s, T_{horizon} ;
12   Set objective function via Equation (3);
13   Define state and input constraints, v_{limit}, ω_{limit};
14   Add safety constraints via Equation (26c);
15   **if** *double-integrator dynamics* **then**
16   Add liveness constraint via (27c);
17   **end**
18   **Function** `Step(`x_t`)`:
19   **return** u_t^i;
20   **return** mpc;
21 **Function** `RunSimulation()`:
22   **for** $t \leftarrow 0$ **to** T_{sim} **do**
23   **for** $i \leftarrow 0$ **to** k **do**
24   `MPC()`;
25 $u_t^i \leftarrow$`MPC.Step(`x_t`)`;
26   **if** *single-integrator dynamics* $\&$ $\ell_j \left(p_t^i, v_t^i \right) < \ell_{\text{thresh}}$ (deadlock) **then**
27   **if** $v_t^i = v_t^j$ **then**
28   break ties by computing σ_{OPT} (Section 5.3)
29   **end**
30   **end**
31   Perturb u_t^i via Equation (23)
32   **end**
33   Update state $x_{t+1}^i \leftarrow x_t^i$ via Equation (1);
34   **end**
35 Call `RunSimulation()`;

MPC function is configured, importing the model defined previously. This function sets up the parameters using the simulation settings, defines the objective function, and establishes state and input constraints, including safety constraints [Lines 10-19]. If the system has double-integrator dynamics, a liveness constraint is also added.

Finally, the algorithm executes the main simulation loop [Lines 21-31]. During each time step, the MPC controller computes the safe control input u_t^i for each agent considering the joint state x_t, using the MPC controller [Line 25]. If the system operates with single-integrator dynamics and detects a deadlock (Theorem 3), the control input (u_t^i) is perturbed to resolve the deadlock breaking ties if necessary by running the auction described in Section 5.3. The state is then updated for the next time step using the system dynam-

ics. The MPC controller in Line 25 solves an optimal control problem modified from (3). The best response, $\left(\Gamma^{i,*}, \Psi^{i,*}\right)$ can be solved by adding Equations (8) and (24) as constraints.

6 Evaluation and discussion

We deployed our approach in simulation as well as on physical robots in social mini-games occurring at doorways, hallways, and intersections, and analyze–(i) safety and efficiency compared to local planners based on reactive planning, receding horizon planning, multi-agent reinforcement learning, and auction-based scheduling and (ii) the smoothness of our deadlock resolution approach compared to alternative perturbation strategies.

6.1 Experiment setup

We numerically validate our approach on differential drive robots in social mini-games that occur at doorways, hallways, intersections, and roundabouts, and analyze its properties. We use the IPOPT solver (Wächter & Biegler, 2006) for solving the MPC optimization. We consider the following differential drive robot:

$$
\begin{bmatrix} \dot{p}^{i,1} \\ \dot{p}^{i,2} \\ \dot{\phi}^i \\ \dot{v}^i \\ \dot{\omega}^i \end{bmatrix} = \begin{bmatrix} \cos\phi^i & 0 & 0 & 0 \\ \sin\phi^i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/m & 0 \\ 0 & 0 & 0 & I^{-1} \end{bmatrix} \begin{bmatrix} v^i \\ \omega^i \\ u_1 \\ u_2 \end{bmatrix}, \tag{36}
$$

where subscript i denotes the ith agent, m and I are the mass and moment of inertia about the center of mass respectively, $\left[p^{i,1}, p^{i,2}\right]^\top \in \mathbb{R}^2$ represent the position vector of the robot, $\phi^i \in \mathbb{S}^1$ represents its orientation, v^i, ω^i are the linear and angular velocities of the robot, respectively, and $u^i = [u_1, u_2]^\text{T} \in \mathcal{U}^i$ is the control input. The discrete-time dynamics of (36) can be described by:

$$
x^i_{t+1} = f\left(x^i_t\right) + g\left(x^i_t\right) u^i_t \tag{37}
$$

where the sampling time period $\Delta T = 0.1s$, $x^i_t = [p^{i,1}, p^{i,2}, \phi^i, v^i, \omega^i]^\text{T}$ is the state and $u^i_t = [u_1, u_2]^\text{T}$ is the control input. The objective is to compute control inputs that solve the following non-linear optimal control problem minimize the following cost function,

$$
\min_{u_{1:T-1}} \sum_{t=0}^{T-1} x^{i\top}_t Q x^i_t + u^{i\top}_t R u^i_t
$$

$$
\text{s.t} \quad x^i_{t+1} = f\left(x^i_t\right) + g\left(x^i_t\right) u^i_t, \quad \forall\, t \in [1; T-1]
$$

$$
x_t \in \mathcal{X},\ u_t \in \mathcal{U}^i \quad \forall\, t \in \{1, \dots, T\} \tag{38}
$$

The safety for the differential drive robot i is guaranteed by the satisfaction of CBF constraint (31) as we consider control inputs that belong to $\mathcal{U}^i$. For each agent, the CBF for the obstacles is characterized by $h_s\left(x^{i,m}_t\right)$ given by

$$
h_s\left(x^{i,m}_t\right) = \left(p^{i,1} - c^{1,m}\right)^2 + \left(p^{i,1} - c^{2,m}\right)^2 - r^2
$$

where $r > 0$, x^i_t is specified by $\left[p^i_t, \theta^i_t, v^i_t, \omega^i_t\right] \in \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^2 \times \mathbb{R}$ representing the current position, heading, linear and angular rates of the ith robot and $(c_{1,m}, c_{2,m})$ is the center of a circle with radius $r > 0$ and $m \in \{1, \dots, M\}$ where M is sufficiently large to cover all the obstacles. Therefore, the safe region are characterized by the set

$$
\mathcal{X} = \left\{ x^i_t :\ h_s\left(x^{i,m}_t\right) > 0,\ \forall\, m \in \{1, \dots, M\} \right\}
$$

Further, to avoid collisions with another agent, each agent i treats the other agent j $(i \neq j)$ as an obstacle. Consequently, the CBF for agent i is given by Equation (27b). The CBF $h_v(x_t)$ for an agent i is given by

$$
h_v(x^i_t) = A v_t, \quad i \neq j \tag{39}
$$

where $v_t = [v^i_t, v^j_t]^\top$, $A \in \mathbb{R}^{2 \times 2}$ for all the experiments with two robots, but $A \in \mathbb{R}^{3 \times 3}$ matrix for the experiments with 3 robots. For our doorway and intersection simulations, we choose $A = \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}$.

6.2 Environments, robots, baselines, and metrics

Environment-Our approach has been rigorously designed and subjected to tests within two socially relevant mini-games: a doorway passage and a corridor intersection, both confined within a spatial area measuring 3 meters by 3 meters. In the doorway setting, we calibrated the gap size to an approximate width of 0.5 meters. In the corridor intersection scenario, the arms branching from the central conflict zone have been strategically set to range between 1.5 and 2 meters in width, while the conflict zone itself occupies an area between 2.5 to 4 square meters. For the doorway experiment, robots initiate their navigation from one side uniformly spaced at an approximate distance of 1.8 meters from the gap aiming to safely transit to the opposing side. Conversely, the corridor intersection tests focus on autonomous navigation through the intersection with a rotational presence of one robot on each arm. Robots have the freedom to navigate toward any of the remaining three arms, the overarching goal being secure passage through this complex spatial layout.
Robots-Our empirical evaluations involved multi-robot trials conducted on an eclectic mix of robotic platforms: UT

Automata F1/10 car platforms, Clearpath Jackal, and Boston Dynamics Spot. These platforms were also engaged in human-robot interaction tests, specifically using the Jackal. The robots were judiciously selected to capture a diverse range of shapes, sizes, and kinodynamic behaviors. The Spot distinguishes itself as a legged robot, while both the Jackal and F1/10 cars are wheeled, though the Jackal has the unique capability of executing point turns, which the cars lack. Speed capabilities also vary: both the Spot and the Jackal have a maximum speed limit of 1.5 meters per second, whereas the UT Automata F1/10 cars can accelerate up to 9 meters per second. Configurations were experimented with broadly, the sole exception being the co-deployment of Spot and Jackal with the F1/10 cars, as the latter are not detectable by the larger robots due to their smaller size.

Evaluation Baselines–In evaluating our novel local navigation algorithm designed for physical robots operating in real-world environments, we chose to compare it against the dynamic window approach (DWA), a standard local navigation algorithm provided by the `move_base` package in the Robot Operating System (ROS)[3]. This decision was motivated by the fact that `move_base` serves as a well-regarded baseline that is both widely used and accepted in the robotics community, thereby allowing us to gauge the relative merits of our approach in a context that is immediately understandable and relevant to researchers and practitioners alike.

In our simulation-based experiments, we compare our proposed algorithm against both centralized and decentralized local planning algorithms to foster a multifaceted comparison. On the centralized front, we compare with ORCA-MAPF (Dergachev & Yakovlev, 2021) that integrates decentralized collision avoidance facilitated by ORCA with centralized Multi-Agent Path Finding (MAPF) to robustly resolve deadlock. Decentralized planners, on the other hand, can be sub-categorized into reactive and receding-horizon approaches. Within the reactive domain, we examine a diverse suite of methods that encompass quadratic programming-based controllers (Wang et al., 2017), NH-TTC (Davis et al., 2019), and NH-ORCA (Alonso-Mora et al., 2013). We also compare with learning-based methods, such as CADRL (Chen et al., 2017, ?). For receding-horizon planners, our selection incorporates DWA, courtesy of the `move_base` package in ROS, MPC-CBF (Zeng et al., 2021), and IMPC (Chen et al., 2022), for its Model Predictive Control (MPC) framework coupled with deadlock resolution capabilities.

Evaluation Metrics-To comprehensively assess navigation algorithms, we employ a carefully curated set of metrics, each averaged across multiple trials, that analyze three essential dimensions: safety, smoothness, and efficiency. We incorporate metrics such as the success rate, collision rate, Avg. ΔV,

path deviation, makespan ratio, and specific flow rate. Avg. ΔV represents the average change in velocity from the starting point to the destination, measuring how fluidly a robot can adjust its velocity while still reaching its destination, without the risk of frequent stops or abrupt changes. Path deviation measures the deviation from a robot's preferred trajectory. Makespan ratio encapsulates the time-to-goal disparity between the first and last agent, and specific flow rate offers a macroscopic view, quantifying the overall 'volume of agents' successfully navigating through a constrained space, such as a doorway.

While metrics like Avg. ΔV and path deviation evaluate an individual robot's behavior, specific flow rate captures the collective efficiency of multiple agents navigating through constrained spaces, such as doorways. This distinction is significant because achieving a flow rate similar to that of human crowds implies that the navigation algorithm under examination is not merely efficient at an individual level, but also highly effective in optimizing multi-agent movement.

6.3 Liveness sets

We empirically generate the liveness sets for the 2 agent and 3 agent social mini-games occurring at doorways and intersections. We deploy different combinations of robots with varying velocities in the social mini-games and the distribution of velocities that results in safe navigation is the empirical liveness set for that social mini-game. In Fig. 8a, we report the outcomes of trials conducted in the doorway mini-game using 2 F1/10 car platforms. We conducted 5 trial runs, averaged across 3 iterations, with varying velocities of which 3 succeeded and 2 failed. We observed that the trials succeeded when the cars' velocities satisfied Equation (24), whereas the trials failed when the velocities did not satisfy Equation (24). We repeated this experiment in the intersection scenario as shown in Fig. 8b and obtained an identical liveness set. With 3 F1/10 robots we obtained a liveness set similar to Equation (25).

Finally, we show that the liveness set given by Equation (24) for the social mini-game at doorways holds for cross robot platforms. In Fig. 8c, we demonstrate the Spot and the Jackal robots in the doorway scenario using an identical setup as with the F1/10 platforms and report the results. To gather more evidence for the liveness sets, we scaled the velocities by $1/\zeta^i$ in SocialGym 2.0[4] and observed similar results. That is, using a $1/\zeta^i$ velocity scaling yields a 76% and 96% success rate in the doorway and intersection scenario, respectively. The variability and reduction in the success rates is due to the stochastic nature of the reinforcement learning control policies employed in SocialGym 2.0, a reinforcement learning simulator, that are unable to provide hard guarantees

---

[3] http://wiki.ros.org/move_base.

[4] https://github.com/ut-amrl/social_gym.

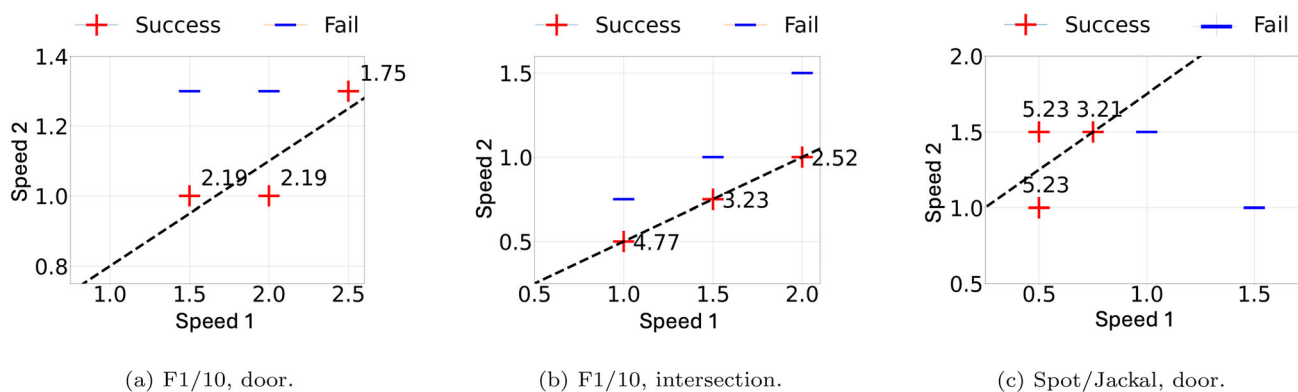(a) F1/10, door.   (b) F1/10, intersection.   (c) Spot/Jackal, door.

Fig. 8 Generating liveness sets using 2 F1/10 car platforms, Jackal, and the Spot for the doorway social mini-game: We record successful and failed trials along with the makespan in the case of successful tri-

als. Figures 8a and 8b use F1/10 cars while Fig. 8c shows results for the Jackal versus the Spot. We notice that robots succeed when the velocities ("velocity 1" and "velocity 2") satisfy Eq. 24

on safety. We clarify that the purpose of the experiments in SocialGym 2.0 is only to validate the concept of liveness sets. That is, in those instances where the agents did not collide due to stochasticity, agents respected the liveness sets.

6.4 Real world experiments

Multi-Robot Setting-In our evaluation against the `move_base` DWA algorithm, a trial is deemed a success if robots navigate through the doorway or intersection without encountering collisions or deadlocks, while any instance of these events marks a failure. As illustrated in Fig. 9, the initial three rows capture the successful performance of our algorithm across intersection and doorway scenarios, employing the F1/10 car platforms, the Jackal, and the Spot. Conversely, the last row reveals a failure case where DWA experiences a collision in the doorway scenario, thereby serving as a contrasting backdrop to the successful outcomes exhibited in the second row using our algorithm in identical configurations.

Crucially, our experimental findings reveal that our proposed navigation framework exhibits a level of efficiency strikingly similar to human navigation. Grounded in research that has studied human navigational speeds and makespan durations in bottleneck scenarios and narrow doorways (Garcimartín et al., 2016; Kretz et al., 2006), the average human specific flow rate[5] has been observed to range between $2.0 - 2.1$ $(\text{m·s})^{-1}$ in environments analogous to our test setups. In comparative terms, our navigation algorithm enabled physical robots to traverse doorways with an average specific flow rate of 2.0 $(\text{m·s})^{-1}$, essentially mirroring the navigational efficiency commonly exhibited by humans. This result amplifies the significance of the specific flow rate

metric, highlighting its utility in benchmarking the collective efficiency of robotic navigation against human standards.

Human-Robot Setting-In an extension of our work to human-robot interactions, we recreate the doorway and intersection social mini-game scenarios[6], as depicted in Fig. 10. Here, we compare our algorithm against the traditional Dynamic Window Approach (DWA), introducing a human participant into the equation for a more nuanced assessment. Figures 10a, 10b, and 10c, corresponding to rows 1 and 2 for the doorway situation, eloquently highlight how our control-theoretic, deadlock-avoidance strategy equips the robot with the finesse to decelerate strategically. This allows the robot to defer to the human participant seamlessly, thereby maintaining an uninterrupted flow of motion. This stands in stark contrast to the results achieved by DWA, where the robot ends up making physical contact with the human, as illustrated in Figs. 10d, 10e, and 10f.

Of particular significance is our algorithm's capability to adapt to varied human behaviors. As Figs. 10j and 10k demonstrate, our robot is not merely programmed to yield; it can also dynamically accelerate when the situation warrants, particularly when human participants display hesitancy or a conservative pace. This preemptive speeding up averts potential deadlocks that could otherwise ensue. Conversely, DWA lacks the agility for such assertive, situationally-responsive maneuvers, often resulting in a halted state and thereby creating a deadlock scenario.

6.5 Simulation results

We implement the MPC-CBF controller using double-integrator system dynamics in Equation (1), cost function in Equation (26), and constraints in Equation (27).

---

[5] specific flow rate is measured in $\frac{N}{zT}$, where N is the number of robots, T is the makespan in seconds, and z is the gap width in meters.

[6] We suggest the reader to watch the full video at https://www.youtube.com/watch?v=fA7BbM8iTwg.
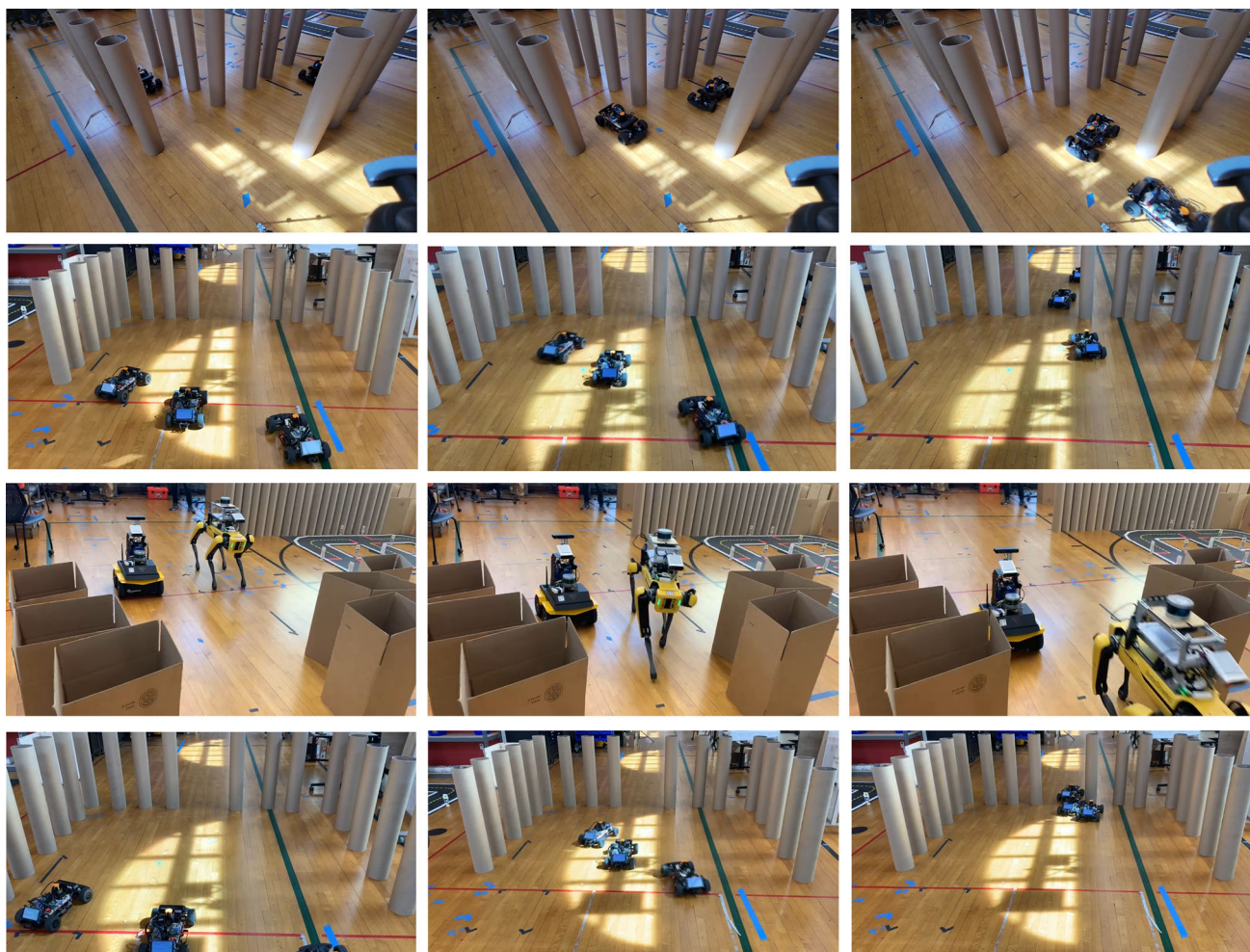
Fig. 9 Deployment in multi-robot scenarios: *(rows* 1, 2, 3) We demonstrate our approach at a corridor intersection and doorway using the F1/10 cars, Spot, and the Jackal robots. *(row* 4) DWA from the ROS `move_base` package results in a collision

All experiments have been performed on a AMD Ryzen machine@2.90GHz with 16.0GB of RAM.

Analysis–Fig. 11 presents simulation results and analysis of interaction between two agents at a narrow doorway. In the first experiment where distance d is set to $1.11m$ and the difference in goal position Δx_g is 0, shown in Fig. 11a, a deadlock is detected at $t = 5$. This is indicated by the liveness value dipping below the threshold (0.3) at this time. At this point, agent 2 already starts to slow down according to Eq. 23, as indicated by the green curve. The deadlock occurrence is around $t = 10$, which is when agent 2 slows down even further, while agent 1's speed (blue curve) shows a significant peak, in order to prevent the deadlock. Figure 11c demonstrates the same scenario in the absence of deadlock prevention. In this case, both agents slow down around $t = 10$ and eventually come to a halt. Throughout this period, the liveness value remains below the threshold.

Figures 11b and 11d follow a similar pattern with a larger distance $d = 2.55m$ and a slight difference in goal position

$\Delta x_g = 0.2m$. The goal with this second experiment is to convey that our perturbation algorithm is invariant to changes in the configuration.

Versus baselines with deadlock resolution–We compare our deadlock resolution method (MPC-CBF) with quadratic programming-based controllers (QP-CBF) (Wang et al., 2017) using random perturbation as well as a variant where we implement our approach (QP-CBF), infinite-horizon MPC and buffered voronoi cells (Chen et al., 2022), a multi-agent reinforcement learning-based algorithm, CADRL (Chen et al., 2017, ?), and finally, with an ORCA-based approach (Dergachev & Yakovlev, 2021) that uses multi-agent path finding (MAPF) to resolve deadlocks. We perform experiments in social mini-games at doorways, hallways, and intersections, and report the mean ± standard deviation for average change in velocity (ΔV), path deviation (meters), and average time steps across 2 agents in Table 3. For all three settings, deadlock resolution results in the smallest average velocity and path deviation, which

(a) *Our approach:* Initial time step. (b) *Our approach:* Robot yields to human. (c) *Our approach:* Robot follows through.

(d) *DWA:* Initial time step. (e) *DWA:* Robot does not yield to human. (f) *DWA:* Robot collides with human.

(g) *Our approach:* Initial time step. (h) *Our approach:* Human slows down. (i) *Our approach:* Robot speeds up.

(j) *DWA:* Initial time step. (k) *DWA:* Robot stops when human slows. (l) *DWA:* Robot moves once human retreats.

Fig. 10 Deployment in human-robot scenarios: *(Doorway)* Figs. 10a, 10b, and 10c demonstrate our deadlock avoiding strategy enables the robot to yield to the human by slowing down, rather than stopping, and smoothly follows the human through the door. Figures 10d, 10e, and 10f demonstrates a baseline DWA planner that results in the robot colliding with the human. *(Intersection)* Figs. 10j, 10k, and 10l demonstrate our game-theoretic deadlock avoiding strategy enables the robot to proactively speed up when it notices the human slow down. Figures 10d, 10e, and 10f demonstrates a baseline DWA planner that results in the robot stopping abruptly in front of the human, only moving forward once the human steps back

(a) *(deadlock prevented)* $d = 1.11m, \Delta x_g = 0$

(b) *(deadlock prevented)* $d = 2.55m, \Delta x_g = 0.2m$

(c) *(deadlocked)* $d = 1.11m, \Delta x_g = 0$

(d) *(deadlocked)* $d = 2.55m, \Delta x_g = 0.2m$
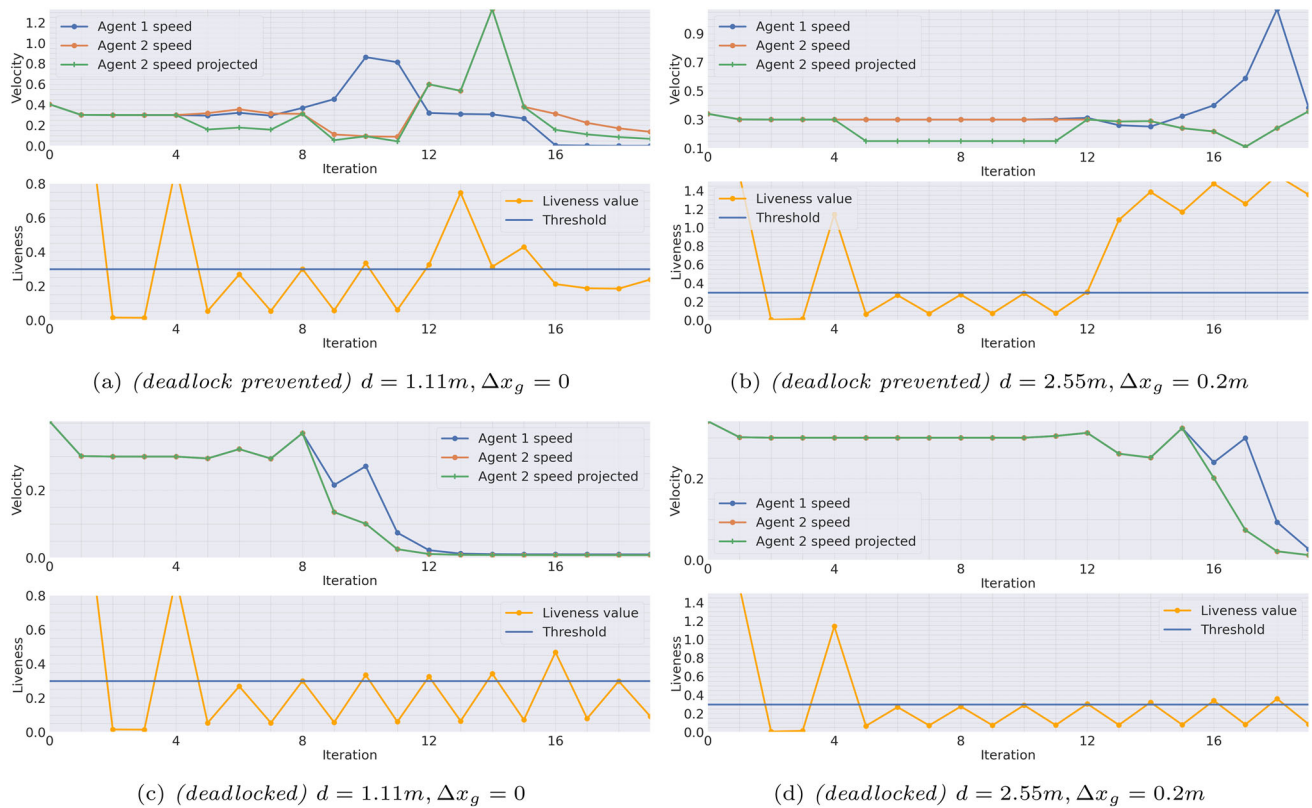
Fig. 11 Deadlock is detected around $t = 4$. Agent 2 projects its original velocity to a scaled velocity such that differs from agent 1's velocity by at least a factor of ζ. Deadlock is prevented between $t = 5$ and $t = 10$ as confirmed from $\ell_j \left(p_t^i, v_t^i \right) \geq \ell_{\text{thresh}} = 0.3$ (Color figure online)

implies that our approach allows agents to modulate their velocities more efficiently that results in a trajectory that more closely resembles the preferred trajectory. Note that path deviation for ORCA-MAPF will always be zero due to the discrete nature of the MAPF. Additionally, we observe that the average completion time depends on the social mini-game. For instance, random perturbation takes longer in the doorway setting due to the increased number of constraints in space. In the intersection and hallway settings, however, perturbation is slower or comparable since random perturbation agents move faster, albeit along inefficient trajectories. Lastly, we observe that MPC + liveness results in a very small makespan at the cost of high average ΔV and path deviation. This suggests that simply adding liveness constraints to the MPC solver encourages goal reaching but at the cost of smoothness.

Versus baselines <u>without</u> deadlock resolution-Our study compares our navigation strategy against other multi-robot navigation algorithms that lack any explicit deadlock resolution mechanisms. This includes a range of methods like MPC-CBF (Zeng et al., 2021), NH-ORCA (Alonso-Mora et al., 2013), NH-TTC (Davis et al., 2019), CADRL and its variants (Chen et al., 2017; Everett et al., 2021, 2018), and DWA. Of these methods, MPC-CBF, NH-ORCA, and NH-TTC fail

to reach their goals, either colliding or ending in a deadlock. We visually demonstrate these comparisons in Fig. 12. With our method, green agents yield to red agents, enabling smooth passage through constricted spaces like doorways or intersections. Conversely, approaches like MPC-CBF, NH-ORCA, and NH-TTC falter, leading to either collisions or unresolved deadlocks. For example, Figs. 12b and 12f display a deadlock when using MPC-CBF due to the symmetrical challenges posed by the environment. Likewise, methods that emphasize collision-free navigation, such as NH-ORCA and NH-TTC, encounter deadlocks as displayed in Figs. 12c, 12g, 12d, and 12h. The inherent symmetry of the environment leaves these agents with empty feasible action sets, leading to a deadlock unless specific perturbation strategies are deployed.

6.6 Dealing with input constraints

We performed a simulation using an empty environment and a complex maze environment which is a $10m \times 10m$ grid. For each experiments 5 trials are conducted with each trial being randomized with respect to start and goal position in both of the environments. All experiments have been conducted on an AMD Ryzen machine @2.90GHz with 8 cores and 16.0 GB of RAM. The maximum number of agents tested

Table 3 We observe that alternate perturbation strategies are less smooth due to larger average changes in velocity and path deviation compared to game-theoretic perturbation. The *Label* column describes the amount of centralization and information required–Decentralized (D), Centralized (C), Full Information (F) or Partial Information (P)

	Baseline	Label	CR(%)	DR(%)	Avg. ΔV	Path Deviation	Makespan Ratio
DOORWAY	Wang et al. (2017)	D/P	0	0	0.380 ± 0.12	1.874 ± 0.28	2.680 ± 1.00
	ORCA-MAPF (Dergachev & Yakovlev, 2021)	C/F	0	0	0.100 ± 0.00	0.000 ± 0.00	3.040 ± 0.00
	Auction-based (Wang et al., 2017)	D/P	0	0	0.220 ± 0.01	0.145 ± 0.00	3.370 ± 0.01
	IMPC-DR (Chen et al., 2022)	D/F	0	0	0.080 ± 0.00	0.160 ± 0.00	1.530 ± 0.00
	CADRL (Chen et al., 2017)	CTDE/P	50	0	0.036 ± 0.00	1.000 ± 0.00	3.500 ± 0.00
	GBPPlanner (Patwardhan et al., 2022)	D/F	0	0	0.015 ± 0.00	2.670 ± 0.00	1.250 ± 0.00
	MPC + liveness ($\gamma = 1$)	D/P	0	0	0.270 ± 0.01	0.912 ± 0.00	1.274 ± 0.01
	NH-ORCA (Alonso-Mora et al., 2013)	D/P	0	100	-	-	-
	NH-TTC (Davis et al., 2019)	D/F	0	100	-	-	-
	MPC-CBF (Zeng et al., 2021)	D/P	0	100	-	-	-
	Ours	**D/P**	0	0	$\mathbf{0.001 \pm 0.00}$	$\mathbf{0.089 \pm 0.02}$	$\mathbf{1.005 \pm 0.00}$
INTERSECTION	(Wang et al., 2017)	D/P	0	0	0.300 ± 0.10	0.400 ± 0.14	1.900 ± 0.04
	ORCA-MAPF (Dergachev & Yakovlev, 2021)	C/F	0	0	0.250 ± 0.00	0.000 ± 0.00	2.220 ± 0.00
	Auction-based (Wang et al., 2017)	D/P	0	0	0.290 ± 0.05	0.111 ± 0.04	2.240 ± 0.04
	IMPC-DR (Chen et al., 2022)	D/F	0	0	0.080 ± 0.00	0.151 ± 0.00	1.130 ± 0.00
	CADRL (Chen et al., 2017)	CTDE/P	0	0	0.031 ± 0.00	1.220 ± 0.00	2.000 ± 0.00
	GBPPlanner (Patwardhan et al., 2022)	D/F	0	0	0.017 ± 0.00	7.520 ± 0.00	0.710 ± 0.00
	MPC + liveness ($\gamma = 1$)	D/P	0	0	0.340 ± 0.01	0.922 ± 0.00	1.298 ± 0.01
	NH-ORCA (Alonso-Mora et al., 2013)	D/P	0	100	-	-	-
	NH-TTC (Davis et al., 2019)	D/F	0	100	-	-	-
	MPC-CBF (Zeng et al., 2021)	D/P	0	100	-	-	-
	Ours	**D/P**	0	0	$\mathbf{0.002 \pm 0.00}$	$\mathbf{0.066 \pm 0.01}$	$\mathbf{1.005 \pm 0.01}$
HALLWAY	Wang et al. (2017)	D/P	0	0	0.055 ± 0.01	0.327 ± 0.32	1.160 ± 0.04
	ORCA-MAPF (Dergachev & Yakovlev, 2021)	C/F	0	0	0.110 ± 0.00	1.990 ± 0.00	1.030 ± 0.00
	Auction-based (Wang et al., 2017)	D/P	0	0	0.008 ± 0.000	0.190 ± 0.00	1.440 ± 0.04
	IMPC-DR (Chen et al., 2022)	D/F	0	0	0.135 ± 0.00	0.194 ± 0.00	2.080 ± 0.00
	CADRL (Chen et al., 2017)	CTDE/P	0	0	0.047 ± 0.00	1.000 ± 0.00	1.000 ± 0.00
	GBPPlanner (Patwardhan et al., 2022)	D/F	0	0	0.018 ± 0.00	1.590 ± 0.00	1.000 ± 0.00
	MPC + liveness ($\gamma = 1$)	D/P	0	0	0.290 ± 0.01	0.824 ± 0.00	1.315 ± 0.01

Table 3 continued

Baseline	Label	CR(%)	DR(%)	Avg. ΔV	Path Deviation	Makespan Ratio
NH-ORCA (Alonso-Mora et al., 2013)	D/P	0	100	-	-	-
NH-TTC (Davis et al., 2019)	D/F	0	100	-	-	-
MPC-CBF (Zeng et al., 2021)	D/P	0	100	-	-	-
Ours	**D/P**	0	0	0.001 ± 0.00	0.047 ± 0.00	1.005 ± 0.00

(a) MPC-CBF w. liveness   (b) MPC-CBF [55]   (c) NH-ORCA [11]   (d) NH-TTC [8]

(e) MPC-CBF w. liveness   (f) MPC-CBF [55]   (g) NH-ORCA [11]   (h) NH-TTC [8]
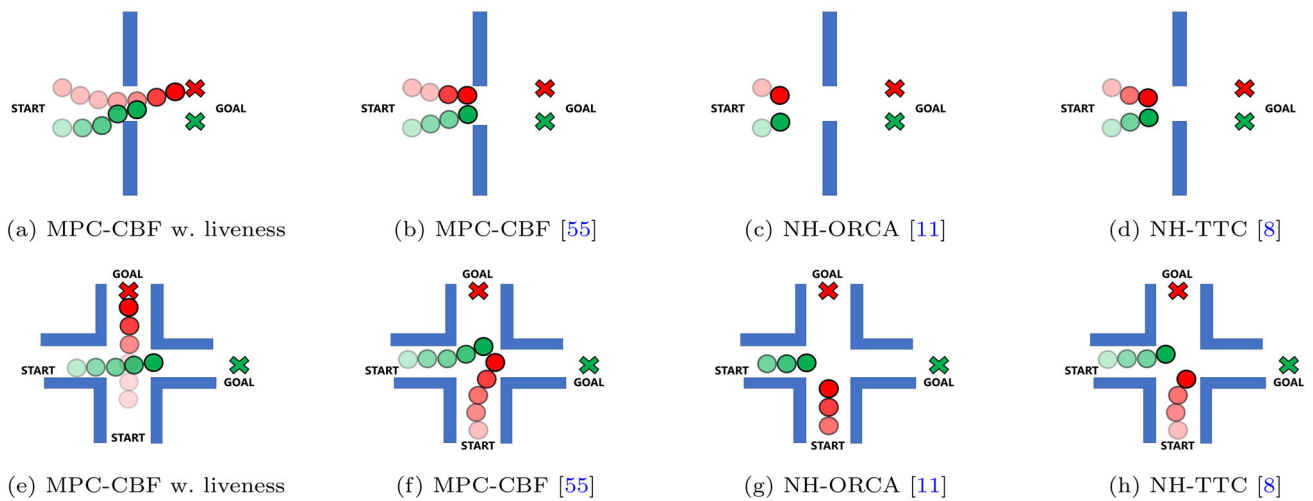
Fig. 12 Comparing with methods **without** deadlock resolution capabilities–Our approach, control-theoretic MPC-CBF enables the green agent to yield to the red agent, which is in contrast to the baselines where the agents enter a deadlock. Similar observation in the intersec- tion scenario. *Conclusion:* Our control-theoretic controller presented in Sect. 5 encourages queue formation thereby resolving a deadlock situation in a decentralized and smooth manner (Color figure online)

(a) Empty environment   (b) Maze environment
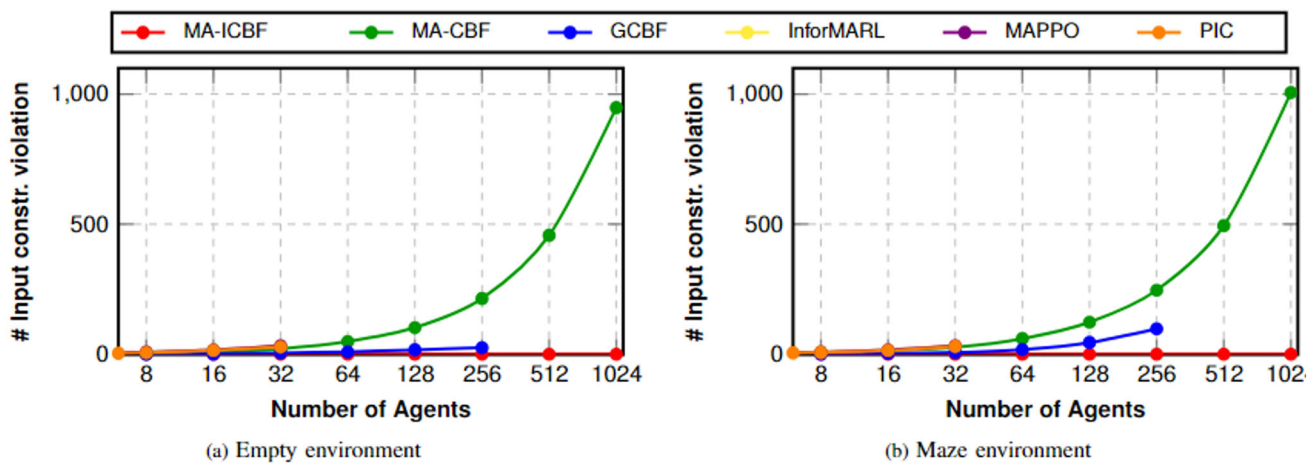
Fig. 13 Average number of input constraint violations versus the number of agents in different settings

in our experiments is 1024. We assume each agent's control policy satisfies the input constraints i.e. $\left| u_t^i \right| \leq u_{\max}$ where $u_{\max} > 0$. We set the input constraints in the range of $[-0.2, 0.2]$ meters per second for the linear velocity and $[-12, 12]$ degrees for the angular velocity. Figure 13 shows that the proposed approach successfully constrains all agents within these bounds. In contrast, baselines GCBF (Zhang et al., 2023), MACBF (Qin et al., 2021), and MARL (Yu et
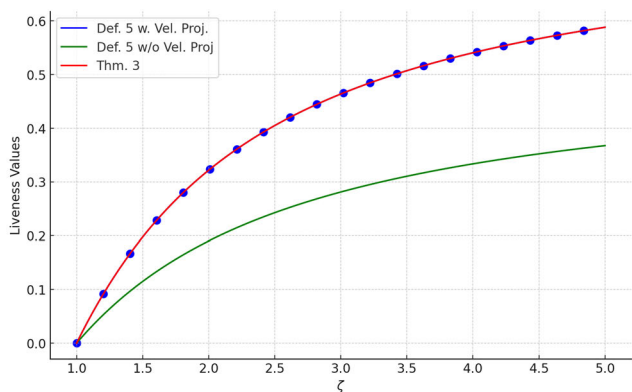
(a) For $\theta = \frac{\pi}{6}$: Showing that Definition 5 with velocity projection.
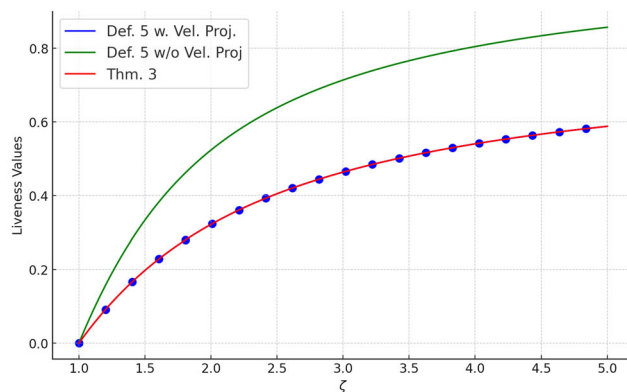
(b) For $\theta = \frac{\pi}{3}$: Showing that Definition 5 with velocity projection reconciles with Theorem 3.

Fig. 14 Comparison of Definition 5 with velocity projection with Theorem 3 for different values of θ

al., 2022; Liu et al., 2020) methods (InfoMARL, MAPPO, PIC) show an exponential increase in the number of input constraint violations. We refer the reader to (Zinage et al., 2024) for additional details.

6.7 Effect of velocity projection

In Fig. 14a, we plot the liveness values obtained via Definition 5 with velocity transformation (blue), Definition 5 without velocity transformation (green), and Theorem 3 (red) for $\theta = \frac{\pi}{6}$. Similarly, in Fig. 14b, we plot the liveness values obtained via Definition 5 with velocity transformation (blue), Definition 5 without velocity transformation (green), and Theorem 3 (red) for $\theta = \frac{\pi}{3}$.

We ran additional experiments where we tested our approach in SMGs where the agents made the following angles with the relative poisiton vector: $\left\{ \frac{\pi}{3}, \frac{\pi}{6}, \frac{\pi}{5} \right\}$. In Fig. 15, we plot trials conducted with varying speed scaling factors (each $(v1, v2)$ point characterizes a speed scaling) and angle. A green "+" trial indicates when an SMG occurrence was correctly identified (true positives and true negatives) using velocity transformation whereas a red "x" indicates when an SMG was incorrectly identified (false positives and false negatives) without velocity transformation.

For example, of the three green points, Definition 5 with velocity transformation identifies a true positive occurrence of an SMG at v1=1.75 m/s and v2=1 m/s ($\zeta = 1.75$, $\theta = \frac{\pi}{3}$) and two true negative SMGs at v1=2.25 m/s and v2=1 m/s ($\zeta = 2.25$, $\theta = \frac{\pi}{5}$) and v1=3 m/s and v2=1 m/s ($\zeta = 3$, $\theta = \frac{\pi}{6}$), respectively. On the other hand, of the three red points, Definition 5 without velocity transformation identifies a false negative occurrence of an SMG at v2=1.75 m/s and v1=1 m/s ($\zeta = 1.75$, $\theta = \frac{\pi}{3}$) and two false positive SMGs at v2=2.25 m/s and v1=1 m/s ($\zeta = 2.25$, $\theta = \frac{\pi}{5}$) and v2=3 m/s and v1=1 m/s ($\zeta = 3$, $\theta = \frac{\pi}{6}$), respectively.
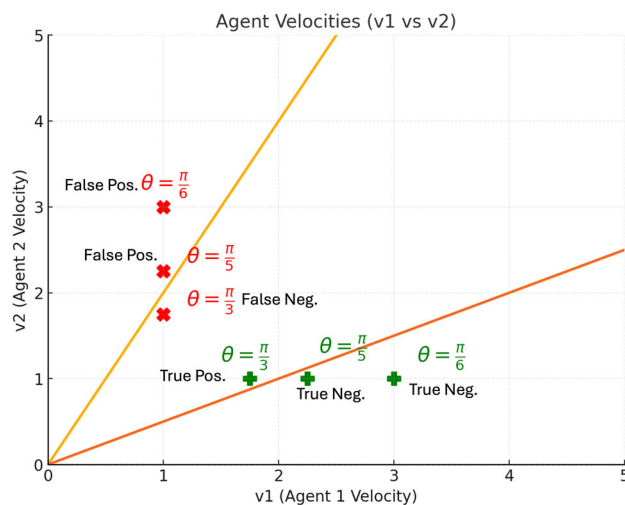
Fig. 15 Effect of velocity projection: the red trials correspond to false negatives and false positives of an SMG occurrence using Definition 5 without velocity projection, whereas the green trials correspond to true positives and true negatives of an SMG occurrence using Definition 5 with velocity projection (Color figure online)

7 Conclusion, limitations, and future work

In this work, we presented an approach to address the challenges of safe and deadlock-free navigation in decentralized, non-cooperative, and non-communicating multi-robot systems in constrained environments. We introduced the notion of *social mini-games* to formalize the multi-robot navigation problem in constrained spaces. Our navigation approach uses a combined MPC-CBF formulation for optimal and safe long-horizon planning. Our approach to ensuring liveness rests on the insight that *there exists barrier certificates that allow each robot to preemptively perturb their state in a minimally-invasive fashion onto liveness sets i.e. states where robots are deadlock-free*.

There are a few limitations of our work. First, we only guarantee safety and liveness for asynchronously performing robots at a local level. And second, the current approach is applicable only in certain geometries such as doorways, hallways, intersections. Some other geometries that our approach can handle, but have not been studied in this paper, include L-shaped corners and corridors, T-shaped junctions, blind corners, and roundabouts. For future work, we plan on investigating the following open questions: (i) investigating the integration of machine learning techniques to enhance decision-making capabilities, adaptability of the decentralized controllers, and extending scalability to multiple agents (ii) establishing theoretical connections between game-theoretic solutions (e.g. Nash equilibrium) and our control-theoretic solution to establish global optimality, (iii) extending our solution to a broader and more general class of geometries, and (iv) investigating robust control methods to establish a bound on ζ to address perception or sensor errors in observing neighbors' velocities.

Data Availability This work primarily used open source software, data, and tools that are available at **https://github.com/ut-amrl**. The data and software produced by this work are available at **https://github.com/Vrushabh27/mpc-cbf**.

Declarations

Ethical Considerations and Conflict of interest This work was conducted in full accordance with the principles highlighted in the Committee on Publication Ethics. Our experiments involving cameras, lidars and any other sensors did not capture or record identifiable information of other humans. Lastly, the authors declare that there are no financial or personal Conflict of interest regarding the publication of this paper.

References

Long, P., Fan, T., Liao, X., Liu, W., Zhang, H., & Pan, J. (2018). "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6252–6259, IEEE.

Chen, Y. F., Liu, M., Everett, M., & How, J. P. (2017). "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 285–292, IEEE.

Everett, M., Chen, Y. F., & How, J. P. (2021). Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access, 9*, 10357–10377.

Everett, M., Chen, Y. F., & How, J. P., (2018). "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3052–3059, IEEE.

Dergachev, S., & Yakovlev, K., (2021). "Distributed multi-agent navigation based on reciprocal collision avoidance and locally confined multi-agent path finding," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 1489–1494, IEEE.

Kamenev, A., Wang, L., Bohan, O. B., Kulkarni, I., Kartal, B., Molchanov, A., Birchfield, S., Nistér, D., & Smolyanskiy, N. (2022). "Predictionnet: Real-time joint probabilistic traffic prediction for planning, control, and simulation," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 8936–8942, IEEE.

Le Cleac'h, S., Schwager, M., & Manchester, Z. (2022). Algames: a fast augmented lagrangian solver for constrained dynamic games. *Autonomous Robots, 46*(1), 201–215.

Davis, B., Karamouzas, I., & Guy, S. J. (2019). "Nh-ttc: A gradient-based framework for generalized anticipatory collision avoidance," arXiv preprint arXiv:1907.05945.

Chandra, R., Menon, R., Sprague, Z., Anantula, A., & Biswas, J. (2023). "Decentralized social navigation with non-cooperative robots via bi-level optimization," arXiv preprint arXiv:2306.08815.

Chen, Y., Guo, M., & Li, Z. (2022). "Recursive feasibility and deadlock resolution in mpc-based multi-robot trajectory generation," arXiv preprint arXiv:2202.06071.

Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., & Siegwart, R. (2013). "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems: The 10th International Symposium*, pp. 203–216, Springer.

Wang, L., Ames, A. D., & Egerstedt, M. (2017). Safety barrier certificates for collisions-free multirobot systems. *IEEE Trans Robotics, 33*(3), 661–674.

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

Arul, S. H., Park, J. J., & Manocha, D. (2023). "Ds-mpepc: Safe and deadlock-avoiding robot navigation in cluttered dynamic scenes," arXiv preprint arXiv:2303.10133.

Şenbaşlar, B., Hönig, W., & Ayanian, N. (2023)."Rlss: real-time, decentralized, cooperative, networkless multi-robot trajectory planning using linear spatial separations," *Autonomous Robots*, pp. 1–26.

Yang, Y., & Wang, J. (2020). "An overview of multi-agent reinforcement learning from game theoretical perspective," arXiv preprint arXiv:2011.00583.

Chandra, R., Maligi, R., Anantula, A., & Biswas, J. (2022). "Socialmapf: Optimal and efficient multi-agent path finding with strategic agents for social navigation," arXiv preprint arXiv:2210.08390.

Van Den Berg, J., Guy, S. J., Lin, M., & Manocha, D. (2011). "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*, pp. 3–19, Springer.

Grover, J., Liu, C., & Sycara, K. (2020). Why does symmetry cause deadlocks? *IFAC-PapersOnLine, 53*(2), 9746–9753.

Zhou, D., Wang, Z., Bandyopadhyay, S., & Schwager, M. (2017). Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells. *IEEE Robotics and Automation Letters, 2*(2), 1047–1054.

Zhong, Z., Nejad, M., & Lee, E. E. (2020). Autonomous and semi-autonomous intersection management: A survey. *IEEE Intelligent Transportation Systems Magazine, 13*(2), 53–70.

Au, T.-C., Zhang, S., & Stone, P. (2015). "Autonomous intersection management for semi-autonomous vehicles," in *The Routledge handbook of transportation*, pp. 88–104, Routledge.

Carlino, D., Boyles, S. D., & Stone, P. (2013). "Auction-based autonomous intersection management," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 529–534, IEEE.

Suriyarachchi, N., Chandra, R., Baras, J. S., & Manocha, D. (2022). "Gameopt: Optimal real-time multi-agent planning and control for dynamic intersections," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2599–2606, IEEE.

Dresner, K., & Stone, P. (2008). A multiagent approach to autonomous intersection management. *J Artificial Intelligence Res, 31*, 591–656.

Xiao, X., Liu, B., Warnell, G., & Stone, P. (2022). Motion planning and control for mobile robot navigation using machine learning: a survey. *Autonomous Robots, 46*(5), 569–597.

Ross, S., Gordon, G. J., & Bagnell, J. A. (2011). "A reduction of imitation learning and structured prediction to no-regret online learning".

Ross, S., Gordon, G. J., & Bagnell, J. A. (2011). "A reduction of imitation learning and structured prediction to no-regret online learning".

Daftry, S., Bagnell, J. A., & Hebert, M. (2016). "Learning transferable policies for monocular reactive mav control".

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K. et al. (2008). "Maximum entropy inverse reinforcement learning.," in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA.

Mehr, N., Wang, M., Bhatt, M., & Schwager, M. (2023). "Maximum-entropy multi-agent dynamic games: Forward and inverse solutions," *IEEE Transactions on Robotics*.

Gonon, D., & Billard, A. (2023). "Inverse reinforcement learning of pedestrian–robot coordination," *IEEE Robotics and Automation Letters*.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

Ho, J., & Ermon, S. (2016). "Generative adversarial imitation learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4572–4580.

Kostrikov, I., Nachum, O., & Tompson, J. (2019). "Imitation learning via off-policy distribution matching".

Dadashi, R., Hussenot, L., Geist, M., & Pietquin, O. (2020). "Primal wasserstein imitation learning," *CoRR*, vol. arxiv:2006.04678.

Zheng, B., Verma, S., Zhou, J., Tsang, I. W., & Chen, F. (2022). Imitation learning: Progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems, 99*, 1–16.

Torabi, F., Warnell, G., & Stone, P. (2018). "Behavioral cloning from observation," arXiv preprint arXiv:1805.01954.

Karnan, H., Torabi, F., Warnell, G., & Stone, P. (2022). "Adversarial imitation learning from video using a state observer," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2452–2458, IEEE.

Wang, M., Wang, Z., Talbot, J., Gerdes, J. C., & Schwager, M. (2021). Game-theoretic planning for self-driving cars in multivehicle competitive scenarios. *IEEE Transactions on Robotics, 37*(4), 1313–1325.

Britzelmeier, A., Dreves, A., & Gerdts, M. (2019). "Numerical solution of potential games arising in the control of cooperative automatic vehicles," in *2019 Proceedings of the conference on control and its applications*, pp. 38–45, SIAM.

Fisac, J. F., Bronstein, E., Stefansson, E., Sadigh, D., Sastry, S. S., & Dragan, A. D. (2019). "Hierarchical game-theoretic planning for autonomous vehicles," in *2019 International conference on robotics and automation (ICRA)*, pp. 9590–9596, IEEE.

Schwarting, W., Pierson, A., Karaman, S., & Rus, D. (2021). "Stochastic dynamic games in belief space," *IEEE Transactions on Robotics*, pp. 1–16.

Sun, W., Theodorou, E. A., & Tsiotras, P. (2015). "Game theoretic continuous time differential dynamic programming," in *2015 American Control Conference (ACC)*, pp. 5593–5598, IEEE.

Sun, W., Theodorou, E. A., & Tsiotras, P. (2016). "Stochastic game theoretic trajectory optimization in continuous time," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 6167–6172, IEEE.

Morimoto, J., & Atkeson, C. G. (2003). "Minimax differential dynamic programming: An application to robust biped walking," in *Advances in neural information processing systems*, pp. 1563–1570.

Fridovich-Keil, D., Ratner, E., Peters, L., Dragan, A. D., & Tomlin, C. J. (2020). "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 1475–1481, IEEE.

Di, B., & Lamperski, A. (2018). "Differential dynamic programming for nonlinear dynamic games," arXiv preprint arXiv:1809.08302.

Di, B., & Lamperski, A. (2019). "Newton's method and differential dynamic programming for unconstrained nonlinear dynamic games," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4073–4078, IEEE.

Di, B., & Lamperski, A. (2020). "First-order algorithms for constrained nonlinear dynamic games," arXiv preprint arXiv:2001.01826.

Hansen, E. A., Bernstein, D. S., & Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. *AAAI, 4*, 709–715.

Laumond, J.-P., Sekhavat, S., & Lamiraux, F. (2005). "Guidelines in nonholonomic motion planning for mobile robots," *Robot motion planning and control*, pp. 1–53.

Chen, X., Liu, J., de la Pena, D. M., & Christofides, P. D. (2010). Sequential and iterative distributed model predictive control of nonlinear process systems subject to asynchronous measurements. *IFAC Proceedings Volumes, 43*(5), 625–630.

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

Zeng, J., Zhang, B., & Sreenath, K. (2021). "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*, pp. 3882–3889, IEEE.

Tedrake, R., (2023). *Underactuated Robotics*.

Robey, A., Hu, H., Lindemann, L., Zhang, H., Dimarogonas, D. V., Tu, S., & Matni, N. (2020). "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724, IEEE.

Jagtap, P., Pappas, G. J., & Zamani, M. (2020). "Control barrier functions for unknown nonlinear systems using gaussian processes," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3699–3704, IEEE.

Grizzle, J. W., Chevallereau, C., Sinnet, R. W., & Ames, A. D. (2014). Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica, 50*(8), 1955–1988.

Roughgarden, T. (2016). *Twenty lectures on algorithmic game theory*. Cambridge University Press.

Chandra, R., & Manocha, D. (2022). Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging. *IEEE Robotics and Automation Letters, 7*(2), 2676–2683.

Alan, A., Molnar, T. G., Ames, A. D., & Orosz, G. (2023). "Parameterized barrier functions to guarantee safety under uncertainty," *IEEE Control Systems Letters*.

Zinage, V., Chandra, R., & Bakolas, E. (2024). "Disturbance observer-based robust integral control barrier functions for nonlinear systems with high relative degree," in *2024 American Control Conference (ACC)*, pp. 2470–2475, IEEE.

Ames, A. D., Notomista, G., Wardi, Y., & Egerstedt, M. (2020). Integral control barrier functions for dynamically defined control laws. *IEEE control systems letters, 5*(3), 887–892.

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming, 106*, 25–57.

Chen, Y. F., Everett, M., Liu, M., & How, J. P. (2017). "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, IEEE.

Garcimartín, A., Parisi, D. R., Pastor, J. M., Martín-Gómez, C., & Zuriguel, I. (2016). Flow of pedestrians through narrow doors with different competitiveness. *Journal of Statistical Mechanics: Theory and Experiment, 2016*(4), 043402.

Kretz, T., Grünebohm, A., & Schreckenberg, M. (2006). Experimental study of pedestrian flow through a bottleneck. *Journal of Statistical Mechanics: Theory and Experiment, 2006*(10), P10014.

Patwardhan, A., Murai, R., & Davison, A. J. (2022). Distributing collaborative multi-robot planning with gaussian belief propagation. *IEEE Robotics and Automation Letters, 8*(2), 552–559.

Zhang, S., Garg, K., & Fan, C. (2023). "Neural graph control barrier functions guided distributed collision-avoidance multi-agent control," in *Conference on robot learning*, pp. 2373–2392, PMLR.

Qin, Z., Zhang, K., Chen, Y., Chen, J., & Fan, C. (2021). "Learning safe multi-agent control with decentralized neural barrier certificates," arXiv preprint arXiv:2101.05436.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. *Adv Neural Information Processing Systems, 35*, 24611–24624.

Liu, I.-J., Yeh, R.A., & Schwing, A.G, (2020). "Pic: permutation invariant critic for multi-agent deep reinforcement learning," in *Conference on Robot Learning*, pp. 590–602, PMLR.

Zinage, V., Jha, A., Chandra, R., & Bakolas, E., (2024). "Decentralized safe and scalable multi-agent control under limited actuation," arXiv preprint arXiv:2409.09573.

Rohan Chandra is an Assistant Professor in Computer Science at the University of Virginia and leads the CRΔL lab. From 2022 to 2024, he was a postdoctoral research fellow in Texas Robotics, advised by Dr. Joydeep Biswas and Dr. Peter Stone, at the University of Texas at Austin. His research focuses on algorithms and systems for enabling robots to navigate safely and efficiently among humans, like humans. Rohan obtained his M.S. and Ph.D. in 2018 and 2022 from the University of Maryland, College Park advised by Dr. Dinesh Manocha, and completed his B.Tech from the Delhi Technological University, New Delhi in 2016. His doctoral thesis focused on autonomous driving in dense, heterogeneous, and unstructured traffic environments. Rohan is a 2023 Microsoft Future Leader in Robotics and AI, 2023 KAUST Rising Star in AI, 2022 RSS Pioneer, and 2021 UMD Future Faculty Fellow. He is a finalist for the 2022 Charles A. Caramello Distinguished Dissertation Award and the UMD Innovation of the Year Award for 2021 and 2022. He is a recipient of the 2023 Drones Young Investigator Award, the 2023 SNU PhD Award for Autonomous Navigation, and the UMD 2020 summer research fellowship. His published work appears regularly in top computer vision, AI, and robotics conferences and he is actively involved in participating in workshops and program committees of leading conferences in robotics, computer vision, artificial intelligence, and machine learning. Rohan is serving on the editorial boards of several reputed journals including RA-L.

Vrushabh Zinage is a Ph.D. candidate in Aerospace Engineering at the University of Texas at Austin, advised by Professors Efstathios Bakolas and Takashi Tanaka. His research lies at the intersection of motion planning, trajectory optimization, control theory, data-driven control, and robotics. His current work focuses on learning-based control, operator-theoretic approaches to nonlinear systems, information-theoretic motion planning, and safety-critical control using Control Barrier Functions. He has previously interned at the GAMMA Lab at the University of Maryland under the mentorship of Professor Dinesh Manocha, where he worked on fast online motion planning and trajectory optimization algorithms. He has also held internship positions at Airbus and Honeywell Aerospace in India. Before joining UT Austin, he completed his undergraduate studies at IIT Madras, where he collaborated with Professor Satadal Ghosh on developing efficient sampling-based motion planning and trajectory optimization algorithms.

Efstathios Bakolas received his Diploma in Mechanical Engineering with highest honors from the National Technical University of Athens, Greece, in 2004 and his MS. and Ph.D. degrees in Aerospace Engineering from the Georgia Institute of Technology, Atlanta, in 2007 and 2011, respectively. He is currently an Associate Professor in the Department of Aerospace Engineering and Engineering Mechanics at The University of Texas at Austin which he joined as an Assistant Professor in fall 2012. He is an Associate Fellow of AIAA and a Senior Member of IEEE. His main research interests include optimal control theory, stochastic control, optimization-based control, differential games and game theory, computational and algorithmic geometry, learning-based and data-driven control, path planning and motion planning for robotic systems and autonomous vehicles, and distributed control and optimization for multi-agent networks.

Joydeep Biswas is an Associate Professor in the Department of Computer Science at the University of Texas at Austin and an Adjunct Assistant Professor in the College of Information and Computer Sciences at the University of Massachusetts Amherst. He received his Ph.D. in Robotics from Carnegie Mellon University in 2014 and his B.Tech. in Engineering Physics from the Indian Institute of Technology Bombay in 2008. His long-term research vision is to develop self-sufficient autonomous mobile robots capable of operating robustly and accurately in human environments. Toward this goal, his research focuses on perception, planning, and control for autonomous mobile robots.

Peter Stone holds the Truchard Foundation Chair in Computer Science at the University of Texas at Austin. He is Associate Chair of the Computer Science Department, as well as Director of Texas Robotics. In 2013 he was awarded the University of Texas System Regents' Outstanding Teaching Award and in 2014 he was inducted into the UT Austin Academy of Distinguished Teachers, earning him the title of University Distinguished Teaching Professor. Professor Stone's research interests in Artificial Intelligence include machine learning (especially reinforcement learning), multiagent systems, and robotics. Professor Stone received his Ph.D in Computer Science in 1998 from Carnegie Mellon University. From 1999 to 2002 he was a Senior Technical Staff Member in the Artificial Intelligence Principles Research Department at AT&T Labs - Research. He is an Alfred P. Sloan Research Fellow, Guggenheim Fellow, AAAI Fellow, IEEE Fellow, AAAS Fellow, ACM Fellow, Fulbright Scholar, and 2004 ONR Young Investigator. In 2007 he received the prestigious IJCAI Computers and Thought Award, given biannually to the top AI researcher under the age of 35, and in 2016 he was awarded the ACM/SIGAI Autonomous Agents Research Award. Professor Stone co-founded Cogitai, Inc., a startup company focused on continual learning, in 2015, and currently serves as Chief Scientist of Sony AI.