Decentralized Multi-Robot Social Navigation in Constrained Environments via Game-Theoretic Control Barrier Functions

Rohan Chandra[1], Vrushabh Zinage[2], Efstathios Bakolas[2], Joydeep Biswas[1], and Peter Stone[1,*]
{rchandra, vrushabh.zinage, joydeepb, pstone}@utexas.edu, bakolas@austin.utexas.edu
Dept. of Computer Science[1], Dept. of Aerospace Engineering and Engineering Mechanics[2]
The University of Texas at Austin
Project hosted at https://amrl.cs.utexas.edu/snupi.html

***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 to ensure safety, preventing deadlocks in a decentralized fashion with global consensus remains an open problem. In this work, we first formalize the above multi-robot navigation problem in constrained spaces with multiple conflicting agents, which we term as *social mini-games*. To solve social mini-games, we propose a new class of decentralized controllers that ensure both safety and deadlock resolution by attaining a game-theoretic Nash equilibrium. Our controller builds on two key insights: first, we reduce the deadlock resolution problem to solving for the mixed-Nash equilibrium of an N-player Chicken game. Second, we formulate the Nash equilibrium as a control barrier function (CBF) and integrate it with traditional CBFs to inherit their safety guarantees.**

We evaluate our proposed game-theoretic navigation algorithm in simulation as well in the real world using F$1/10$ robots, a Clearpath Jackal, as well as a Boston Dynamics Spot in a doorway, corridor intersection, roundabout, and hallway scenario. We show that (i) our approach results in safer and more efficient navigation compared to local planners based on geometrical constraints, optimization, multi-agent reinforcement learning, and auctions, (ii) our deadlock resolution strategy is the smoothest in terms of smallest average change in velocity and path deviation, and most efficient in terms of makespan (iii) our approach yields a flow rate of $2.8 - 3.3$ (ms)$^{-1}$ which is comparable to flow rate in human navigation at 4 (ms)$^{-1}$.

I. 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 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:* Successful navigation requires collision-free movement. Like humans, navigation algorithms must produce controls that guarantee robots traverse their environment without collisions.
  2) *Demonstrate liveness:* Like humans, robots must resolve deadlocks without centralized perturbation and with minimal deviation from their preferred velocities.
  3) *Obey kinodynamic constraints:* Humans operate under physical and dynamic constraints. Navigation algorithms must both operate under complex kinodynamic 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 I. Although multi-robot systems encompass a vast range of algorithms, we narrow our focus to algorithms that have been applied to social mini-

*Peter Stone is also the Executive Director of Sony AI, America.

	Necessary Algorithmic Conditions			Assumptions		
Approach	Collision-free	Liveness†	Dynamics	Control	Non§ Cooperative	Observation
DRL [1], [2], [3], [4]	✗	✗	Differential drive	CTDE	✗	Partial
ORCA-MAPF [5]	✗	✓	Single-integrator	Centralized	✗	Full
Prediction + Planning [6]	✗	✗	–	CTDE	✗	Full
Game-theoretic Dist. Opt. [7]	✗	✗	Unicycle	Distributed	✓	Full
NH-TTC [8]	✓	✗	Differential drive	Decentralized	✗	Full
Auction-based [9]	✗	✓*	Ackermann	Distributed	✓	Partial
Buffered Voronoi Cells [10]	✗	✓*	Double-integrator	Distributed	✓	Full
NH-ORCA [11]	✓	✗	Differential drive	Decentralized	✓	Partial
CBFs [12]	✓	✗	Double-integrator	Decentralized	✓	Partial
CBFs+KKT [13]	✓	✓*	Single-integrator	Decentralized	✓	Partial
DS-MPEPC [14]	✓	✗*	Differentially flat	Decentralized	✗	Partial
RLSS [15]	✗	✓*	Differential drive	Decentralized	✓	Partial
Humans	✓	✓	–	Decentralized	✓	Partial
This paper	✓	✓	**Double-integrator**	**Decentralized**	✓	Partial

* Under certain conditions.
§ Non-cooperative robots: robots optimize their individual objective functions [16].
† Deadlock-free.

TABLE I: Comparing various approaches for multi-robot navigation in social mini-games.

games either on real robots or in simulation. These include methods based on deep reinforcement learning [1], [2], [3], [4], multi-agent path finding [5], trajectory prediction [6], game-theoretic distributed optimization [7], auctions [17], geometric planning [5], [11], [18], and other optimization-based methods [8], [10], [12], [13], [14]. From Table I, we note that none of these methods satisfy all of the necessary conditions for optimal multi-robot navigation in social mini-games. In the literature on multi-robot navigation for social mini-games, typically, the problem is that algorithms are either collision-free or deadlock-free, but not both. In a similar vein, algorithms that perform well in simulation, fail when deployed on real robots [19]. With this motivation in mind, 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: This paper presents a provably safe and deadlock-free multi-robot navigation algorithm 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 fully decentralized and assumes partial observability. Our main contributions include:

1) A new class of decentralized controllers that ensure both safety and liveness by attaining a game-theoretic Nash equilibrium. We show that these controllers can be practically deployed on F1/10 robots, a Clearpath Jackal, and a Boston Dynamics Spot.
2) These controllers can generally be tacked on to any constrained optimization-based local trajectory planner such as model predictive control (MPC) or dynamic window approach (DWA), by simply adding our controller as an additional constraint.
3) Our controller builds on two key insights: first, we reduce the deadlock resolution problem to solving for the mixed-Nash equilibrium of an N-player Chicken game. Second, we formulate the Nash equilibrium as a control barrier function (CBF) and integrate it with traditional CBFs to inherit their safety guarantees, while simultaneously imbuing it with liveness guarantees.

We evaluate our proposed game-theoretic 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, roundabout, and hallway scenario. We show that (i) our approach results in safer and more efficient navigation compared to local planners based on geometrical constraints, optimization, multi-agent reinforcement learning, and auctions, (ii) our deadlock resolution strategy is the smoothest in terms of smallest average change in velocity and path deviation, and most efficient in terms of makespan (iii) our approach yields a flow rate of $2.8-3.3$ $(\text{ms})^{-1}$ which is comparable to flow rate in human navigation at 4 $(\text{ms})^{-1}$.

In the remainder of this paper, we discuss related work in Section II and formulate the problem in Section III. We present our game-theoretic controller in Section IV and evaluate its performance in Section V. We conclude the paper in Section VI.

II. 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 I 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.

A. Collision Avoidance

Provable safety can be achieved by single-integrator systems *e.g.* ORCA framework from Van Den Berg et al. [18] and its non-holonomic variant [11], which are effective for fast and exact multi-agent navigation. ORCA conservatively imposes collision avoidance constraints on the motion of a robot as half-planes in the space of velocities. The optimal collision-free velocity can then be quickly found by solving a convex optimization problem through linear programming. The original framework limits itself to holonomic systems but has been extended in [11] 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 limitation of the ORCA framework is that the structure of the half-planes so constructed often results in deadlocks.

Exact safety is harder to prove for higher-order dynamics such as double-integrator dynamics, therefore safety in these systems depend on the planning frequency of the system. For example, the NH-TTC algorithm [8] uses gradient descent to optimize 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) [10] and safety depends not only on the planning frequency but also on the length of the planning horizon.

Finally, control barrier functions (CBFs) [12], [13] guarantee safety via the notion of forward invariance *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.

B. Deadlock Resolution Methods

Deadlocks among agents arise due to symmetry in the environment that may cause conflicts between the agents [10], [13], [20]. 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 [12]. 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 [10], [21] or clockwise rotation [13]. These strategies improve optimality over random perturbation and even give formal guarantees, 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 [10].

Another class of deadlock resolution methods rely on priority protocols and scheduling algorithms similar to those used in the autonomous intersection management literature [22]. Some prominent protocols include first come first served (FCFS), auctions, and reservations. FCFS [23] 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 [24], [25], agents bid to cross the intersection based on a specific bidding strategy. Reservation-based systems [26] 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 [10], [21], 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.

C. Learning-based Approaches

Coupling classical navigation with machine learning is rapidly growing field and we refer the reader to Xiao et al. [27] 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. [1] 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 [2], 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 [3].

Planning algorithms that use trajectory prediction models [6] estimate the future states of the robot in the presence of dynamic obstacles and plan their actions accordingly.

D. 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 [28], [29], 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 algorithms based on dynamic programming [30], such as Markovian Stackelberg strategy, that capture the game-theoretic nature of problems but suffer from the curse of

Symbol	Description
Problem formulation (Section III-A)	
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 ($\mathcal{O}^i : \mathcal{X} \to \Omega^i$)
$\mathcal{N}^i\left(x_t^i\right)$	set of robots detected by i
Γ^i	agent i's trajectory
$\widetilde{\Gamma}^i$	set of preferred trajectories
$\mathcal{T}$	transition dynamics (Equation 1)
$\mathcal{U}^i$	action space for agent i
$\mathcal{J}^i$	running cost for agent i ($\mathcal{J}_t^i : \mathcal{X} \times \mathcal{U}^i \to \mathbb{R}$)
$\mathcal{J}_f^i$	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 III-B)	
$h^i : \mathcal{X} \longrightarrow \mathbb{R}$	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.
Game-theoretic perturbation (Section IV)	
$[1; N]$	set of integers $\{1,\ 2, \ldots,\ N\}$
$\mathscr{C}_\ell(t)$	liveness set
v_t^i	velocity of agent i
v_t	joint velocity of all agents
$\widetilde{v}_t$	perturbed joint velocity
$h_v\left(x_t\right)$	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
$\left(r^i, p^i\right)$	auction specified by allocation and payment rule
ζ^i	private priority incentive parameter

TABLE II: Summary of notation used in this paper.

dimensionality and are limited to two players. The third category consists of algorithms based on differential dynamic programming [31], [32], [33], [34], [35], [36] 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 [7], [37], [38], 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, and does not model self-interested agents. Additional limitations include the lack of deployability in the real world and the requirement of full observation.

III. 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 II.

A. Problem Formulation

We formulate a social mini-game by augmenting a partially observable stochastic game (POSG) [39]: $\left\langle k, \mathcal{X}, \{\Omega^i\}, \{\mathcal{O}^i\}, \{\widetilde{\Gamma}^i\}, \mathcal{T}, \{\mathcal{U}^i\}, \{\mathcal{J}^i\} \right\rangle$ 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 i^{th} 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 and velocity) and hidden 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 $\mathcal{O}^i\left(x_t^i\right) = \left\{x_t^i, \overline{x}_t^j\right\}$ for all $j \in \mathcal{N}\left(x_t^i\right)$, 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 $x_T^i \in \mathcal{X}_g$ where $\mathcal{X}_i$ and $\mathcal{X}_g$ denote the sets containing the initial and final states. A discrete trajectory is specified as the vector $\Gamma^i = \left(x_0^i, x_1^i, \ldots, x_T^i\right)$ and its corresponding input sequence is denoted by $\Psi^i = \left(u_0^i, u_1^i, \ldots, u_{T-1}^i\right)$. We denote $\widetilde{\Gamma}^i$ as the set of *preferred* trajectories for robot i that solve the two-point boundary value problem. More formally, due to the local Lipschitz continuity assumption on f and g, for every state $x_t^i \in \widetilde{\Gamma}^i$, there exists $t' \leq T$ such that $x_t^i \in \mathcal{R}\left(x_t^i, \mathcal{U}^i, t'\right) \cap \widetilde{\Gamma}^i$, where $\mathcal{R}\left(x_t^i, \mathcal{U}^i, t'\right)$ is the set of reachable states from x_t^i traveling for time t'. A preferred trajectory refers to a trajectory 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 jerk, etc.

The transition function is given by $\mathcal{T} : \mathcal{X} \times \mathcal{U}^i \longrightarrow \mathcal{X}$, where $u_t^i \in \mathcal{U}^i$ is the continuous control space for robot i. Robots follow the control-affine kinodynamics,

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

where f and g are locally Lipschitz continuous functions. Each robot has a running cost $\mathcal{J}^i : \mathcal{X} \times \mathcal{U}^i \longrightarrow \mathbb{R}$ that assigns a cost to a control u_t^i at each time step based on (i) distance of the robots current position from the goal, (ii) change in the control across subsequent time steps and (iii) difference between the robots preferred and actual paths.

In addition to describing the tuple above, we also define a collision. Represent the space occupied by robot i (as a subset of the state space $\mathcal{X}$) at any time t by its convex hull as $\mathcal{C}^i\left(x_t^i\right) \in \mathcal{X}$. Then, robots i and j are said to collide at time t if $\mathcal{C}^i\left(x_t^i\right) \cap \mathcal{C}^j\left(x_t^j\right) \neq \emptyset$. Finally, we define a social mini-game as follows,

Definition 1. *A **social mini-game** occurs if for some $\delta \in \mathbb{R}^{>0}$ 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 $\Gamma^i, \Gamma^j \in \widetilde{\Gamma}^i, \widetilde{\Gamma}^j$.*

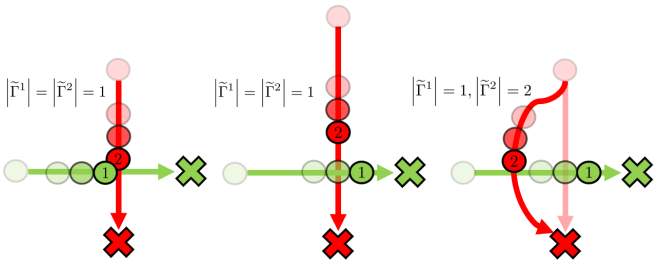


Fig. 1: **Examples/Non-examples of social mini-games:** Arrows indicate the direction of motion for two agents 1 and 2 toward their goals marked by the corresponding cross. The first scenario is a social mini-game since both the preferred trajectories of agents 1 and 2 are in conflict from some $t = a$ 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.

We depict several examples and non-examples of social mini-games in Figure 1. The first scenario can be characterized as a social mini-game due to the conflicting preferred trajectories of agents 1 and 2 within a specific time interval $[a, b] \in T$, where the duration $b - a \geq \delta$. In this case, the agents' trajectories intersect and result in conflicting paths. 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. ***Best Response for a Robot in Social Mini-Games:*** *For the i^{th} robot, given its initial state x_0^i, an optimal trajectory $\Gamma^{i,*}$ and corresponding optimal input sequence $\Psi^{i,*}$ is given by,*

$$\left(\Gamma^{i,*}, \Psi^{i,*}\right) = \arg\min_{(\Gamma^i, \Psi^i)} \sum_{t=0}^{T-1} \mathcal{J}^i\left(x_t^i, u_t^i\right) + \mathcal{J}_T^i\left(x_T^i\right) \quad (2a)$$

$$s.t\ x_{t+1}^i = f\left(x_t^i\right) + g\left(x_t^i\right) u_t^i, \quad \forall t \in [1; N-1] \quad (2b)$$

$$\mathcal{C}^i\left(x_t^i\right) \cap \mathcal{C}^j\left(x_t^j\right) = \emptyset\ \forall j \in \mathcal{N}\left(x_t^i\right) \quad (2c)$$

$$x_0^i \in \mathcal{X}_i \quad (2d)$$

$$x_T^i \in \mathcal{X}_g \quad (2e)$$

where Equation 2b is a discretized version of Equation 1 and $\mathcal{J}_T^i$ is the terminal cost. At this point, we are ready to state our problem objective.

Problem 1. ***Optimal Navigation in Social Mini-Games:*** *A solution to optimal navigation in social mini-games is a tuple $\left(\Gamma^{1,*}, \Gamma^{2,*}, \ldots, \Gamma^{k,*}\right)$*

The objective function $\mathcal{J}^i$ captures the cost of a robot deviating from its preferred trajectory between the robot i's current position and the position in the preferred trajectory at time $t \in [0, T]$. An example of such a cost function is $\mathcal{J}^i\left(x_t^i\right) = \left\|x_t^i - \widetilde{x}_t^i\right\|^2$.

In practice, the optimization 2 can be solved optimally (if a solution is reached) via a Dynamic Window Approach (DWA) [40], Model predictive Controller (MPC) [41], Control Barrier Functions (CBF) [42], among others. Let $\mathscr{U}_t^i \subset \mathcal{U}^i$ be the set of controls such that Equation 2c holds and robot i is collision-free. Further, let $\pi^i \in \mathcal{K}$ refer to a chosen controller (MPC, DWA etc.) chosen by agent i, where $\mathcal{K}$ denotes the set of possible controllers.

In social mini-games, $\mathscr{U}_t^i$ often ends up being an empty set resulting in deadlocks [10], [12], [21]. As in previous work [12], [13], [20], we define a deadlock as follows,

Definition 3. ***Deadlock:*** *A robot i executing the controller given by Equation 2 enters a deadlock if, starting at time t, $u_t^i = 0$ for some large β, while $x_t^i \notin \mathcal{X}_g$.*

Here, β is typically in the order of a few seconds [15]. If agent i is assumed to be traveling along $\widetilde{\Gamma}^i$, then system 1 is small-time local controllable giving the following corollary,

Corollary 1. *If at time t, $x_t^i \in \widetilde{\Gamma}^i$ and there is no social mini-game occurring at t, then agent i is not in a deadlock.*

Proof. If there is no social mini-game occurring at time t, then by definition 1, there exist some $\widetilde{\Gamma}^i$ that is not in conflict with any other agent j. Also by definition of a preferred trajectory, for every state $x_t^i \in \widetilde{\Gamma}^i$, $\exists t' \leq T$ such that $x_t^i \in \mathcal{R}\left(x_{t-t'}^i, \mathcal{U}^i, t'\right) \cap \widetilde{\Gamma}^i$, where $\mathcal{R}\left(x_{t-t'}^i, \mathcal{U}^i, t'\right)$ is the set of reachable states from $x_{t-t'}^i$ traveling for time t'. Then $x_T^i \in \mathcal{R}\left(x_{t-t'}^i, \mathcal{U}^i, t'\right) \cap \widetilde{\Gamma}^i$ for some $x_{t-t'}^i, t'$. System 1 is therefore small-time local controllable if agent i follows $\widetilde{\Gamma}^i$, implying that a deadlock does not occur. □

Resolving these deadlocks typically involves perturbing the robots via heuristics. Commonly used strategies for λ include randomly perturbing [12], [43] the position of a robot causing it to deviate from its preferred trajectories resulting in a sub-optimal controller.

B. Control Barrier Functions

Control Barrier Functions (CBFs) are a powerful tool used in control theory for designing control inputs that guarantee safety of controlled nonlinear systems [42]. CBFs constrain the behavior of a dynamical system to enable collision-free trajectories. CBFs 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) [44] 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 $\mathscr{C}^i$ is closely related to its forward invariance which is a property that requires the system (1) solutions starting from a given set of initial conditions to remain in a desired safe region $\mathscr{C}^i$ for all $t \geq 0$. The basic

idea behind CBFs are as follows. Consider a scalar valued function, $h^i : \mathcal{X} \longrightarrow \mathbb{R}$ over the set of states $x_t^i \in \mathcal{X}$ such that the following holds true:

$$\mathscr{C}^i = \left\{x_t^i \in \mathbb{R}^n | h^i\left(x_t^i\right) \geq 0\right\} \tag{3a}$$
$$h^i\left(x_t^i\right) = 0 \; \forall \; x_t^i \in \partial\mathscr{C}^i \tag{3b}$$
$$h^i\left(x_t^i\right) < 0 \; \forall \; x_t^i \in \mathbb{R}^n \setminus \mathscr{C}^i \tag{3c}$$

where $\mathscr{C}^i$ is the safe set and $\partial\mathscr{C}^i$ denotes the boundary of $\mathscr{C}^i$. The time derivative of $h^i\left(x_t^i\right)$ along the state trajectory of agent i (1) is given as

$$\frac{d\left(h^i\left(x_t^i\right)\right)}{dt} = L_f h^i\left(x_t^i\right) + L_g h^i\left(x_t^i\right) u_t^i \tag{4}$$

where $L_f h^i\left(x_t^i\right)$ and $L_g h^i\left(x_t^i\right)$ denotes 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 $\mathcal{K}_\infty$[1] function κ such that the following holds true

$$\sup_{u_t^i \in U} L_f h^i\left(x_t^i\right) + L_g h^i\left(x_t^i\right) u_t^i + \kappa\left(h^i\left(x_t^i\right)\right) \geq 0 \tag{5}$$

We define the safe or collision-free control space over $x_t^i \in \mathcal{X}$ as the set of controls $u_t^i \in \mathscr{U}^i \subseteq \mathcal{U}^i$ such that the following holds

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

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

$$\mathscr{U}^i = \{u_t^i \in \mathcal{U}^i \, | L_f h^i\left(x_t^i\right) + L_g h^i\left(x_t^i\right) u_t^i + \kappa\left(h^i\left(x_t^i\right)\right) \geq 0\} \tag{7}$$

Equation 6 is known as the safety barrier constraint or safety barrier certificate. In summary, the set $\mathscr{C}^i \subseteq \mathcal{X}$ (defined by (3)) is guaranteed to be safe if the control input set $\mathscr{U}^i$ (defined by (7)) 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 $\mathscr{C} \subset \mathcal{X}^k$.

IV. Game-theoretic Deadlock Resolution

A state is represented as x_t^i at time t of which $p_t^i, \theta_t^i, v_t^i, \omega_t^i \in \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^2 \times \mathbb{R}$ represent the current position, heading, linear and angular rates of the i^{th} robot. We propose a perturbation strategy that can be integrated as a CBF constraint and combined with existing controllers, such as MPC or DWA, to resolve deadlocks. Our perturbation strategy is designed to ensure that:

1) the robot does not deviate from the preferred trajectory.
2) the robot performs the perturbation in a decentralized fashion.
3) the robot deviates minimally from its current velocity.

[1]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 $\lim_{r\to\infty} \alpha(r) = \infty$

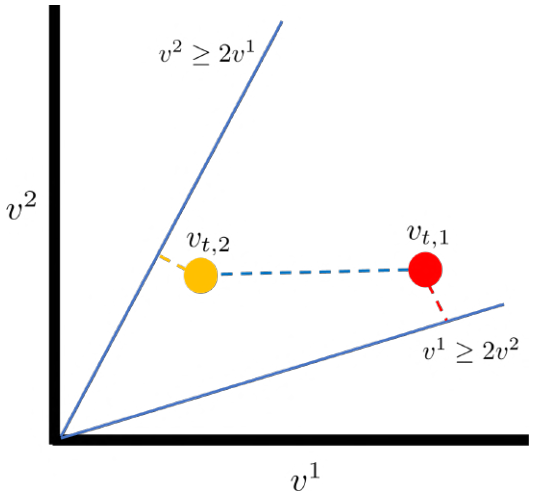


Fig. 2: **Liveness set $\mathscr{C}_\ell(t)$ for the 2 robot scenario:** Velocities for two deadlocked agents are shown by the red point $v_{t,1}$. To resolve the deadlock, Equation 8 is used to project $v_{t,1}$ onto $v_t^1 = 2v^2$. Robot 1 increases its velocity component v^1, while robot 2 decreases its velocity component v^2 to align with the barrier. The perturbation action corresponds to a pair of Nash Equilibrium strategies. If robot 1 deviates from $v_{t,1}$ and decreases its speed to the new yellow point $v_{t,2}$, 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.

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. More formally,

Definition 4. *At any time t, given a configuration of k robots, $x_t^i \in \mathcal{X}$ for $i \in [1, k]$, there exists a union of convex sets, $\mathscr{C}_\ell(t) \in \mathbb{R}^k$ of joint velocity $v_t = \left[v_t^1, v_t^2, \ldots, v_t^k\right]^\top$ such that if $v_t \in \mathscr{C}_\ell(t)$, then a social mini-game does not occur at time t. We call $\mathscr{C}_\ell(t)$ a **liveness set**.*

If every agent at $t = 0$ is collision-free, then perturb u_t^i for all i such that $\mathcal{R}\left(x_t^i, \mathcal{U}^i, \Delta t\right) \cap \widetilde{\Gamma}^i \neq \mathcal{R}\left(\widehat{x}_t^i, \mathcal{U}^i, \Delta t\right) \cap \widetilde{\Gamma}^i$ and $\mathcal{C}^i\left(\widehat{x}_t^i\right) \cap \mathcal{C}^j\left(\widehat{x}_t^j\right) \neq \emptyset \; \forall \; t \in [t, t + \Delta t]$, where $\widehat{x}_{t+\Delta t}^i \in \mathcal{R}\left(\widehat{x}_t^i, \mathcal{U}^i, \Delta t\right) \cap \widetilde{\Gamma}^i$. Trivially, one possible choice is $u_t^j = 0$ for all $j \neq i$ so that $\mathcal{R}\left(\widehat{x}_t^j, \mathcal{U}^j, \Delta t\right) \cap \widetilde{\Gamma}^j = \left\{x_t^j\right\}$ and $\mathcal{C}^i\left(x_t^i\right) \cap \mathcal{C}^j\left(\widehat{x}_t^j\right) \neq \emptyset \; \forall \; t \in [t, t + \Delta t]$ where $\Delta t < \beta$. Consider the resulting set of velocities as $\mathcal{Y} = \left(v_t^1, v_t^2, \ldots, v_t^k\right)$ that precludes a social mini-game. $\mathcal{Y}$ can be convexified by taking the convex hull $conv\left(\mathcal{Y}\right)$. Now, we permute the order of agents $k!$, each resulting in a different convex hull. We take the union of these convex hulls.

Definition 4 and Corollary 1 imply that 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,

$$\widetilde{v}_t = \arg\min_{\mu \in \mathscr{C}_\ell(t)} \|v_t - \mu\|_2 \tag{8}$$

Observe that a solution to 8 achieves the first and third objective, that is, agents do not deviate from their preferred

path and the new joint velocity $\widetilde{v}_t$ is such that agents deviate minimally from their current joint velocity v_t. Towards the second objective, we show that each agent selects a perturbation strategy that constitutes a Nash solution to a variant of the well-known Chicken game.

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 Figure 2, is generated by scaling v_t^2 by $\frac{v_t^1}{\zeta}$, or vice-versa. Empirically, it is observed that $\zeta \geq 2$. We can then generate the following system of linear inequalities,

$$\begin{aligned} v_t^1 &\geq \zeta v_t^2 \\ v_t^2 &\geq \zeta v_t^1 \end{aligned}$$

This can be compactly represented as

$$A_{2\times 2} v_t \geq 0 \tag{9}$$

where $A_{2\times 2} = \begin{bmatrix} 1 & -\zeta \\ -\zeta & 1 \end{bmatrix}$ 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 Figure 2. The point $v_{t,1}$ indicates that $A_{2\times 2} v_{t,1} < 0$ which lies outside $\mathscr{C}_\ell(t)$, implying that the two robots are in a deadlock according to Definition 4 and Definition 3. Equation 8 projects $v_{t,1}$ onto the nearest half-plane which is the $v_t^1 = \zeta v^2$ barrier. Thus, robot 1 will increase v^1 and robot 2 will decrease v^2 by projecting $v_{t,1}$ down on $v_t^1 = \zeta v^2$. This projection is the minimal deviation required on the part of both robots and is therefore optimal.

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,

$$\begin{aligned} v_t^1 &\geq 2v_t^2 \\ v_t^1 &\geq 3v^3 \\ 3v_t^2 &\geq 2v^3 \end{aligned} \tag{10}$$

This can be compactly represented as $A_{3\times 3} v_t \geq 0$ where $A_{3\times 3} = \begin{bmatrix} 1 & -2 & 0 \\ 1 & 0 & -3 \\ 0 & 3 & -2 \end{bmatrix}$ 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 Nash equilibrium is to project p onto the face of the nearest tetrahedron. Equations 9 and 10 can be used to generate the liveness set $\mathscr{C}_\ell(t)$ as,

$$\begin{aligned} \mathscr{C}_\ell(t) &= \{v_t \text{ s.t. } h_v\left(x_t\right) \geq 0\} \\ h_v(x_t) &= \bar{A}_{k\times k}\left(x_t\right) \end{aligned} \tag{11}$$

where $h_v\left(x_t\right)$ is the CBF of $\mathscr{C}_\ell(t)$ that ensures 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, $u_t = \left[u_t^1, u_t^2\right]^\top$. 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. Combining Equation 11 with Equation 6 using general control-affine dynamics in Equation 1, we can derive the game-theoretic safety barrier constraint. As $h_v\left(x_t\right)$ is a vector field (as opposed to a scalar-valued function),

$$L_f h_v\left(x_t\right) + u_t L_g h_v\left(x_t\right) + \kappa\left(h_v\left(x_t\right)\right) \geq 0 \tag{12}$$

where the Lie derivatives of $h_v(x_t)$ along f and g and inequality above are performed element wise. The best response, $\left(\Gamma^{i,*}, \Psi^{i,*}\right)$, in Equation 2 can be solved by adding Equation 11 in constraint 13c as follows,

$$\arg\min_{(\Gamma^i, \Psi^i)} \sum_{t=0}^{T-1} \left(x_t^i - \widetilde{x}_t^i\right)^\top Q\left(x_t^i - \widetilde{x}_t^i\right) + \left(u_t^i\right)^\top R u_t^i + \mathcal{J}_T^i\left(x_T^i\right) \tag{13a}$$

$$\text{s.t} \quad x_{t+1}^i = f\left(x_t^i\right) + g\left(x_t^i\right) u_t^i, \quad \forall t \in [1; N-1] \tag{13b}$$

$$B\left(x_t^i\right) u_t^i \geq C\left(x_t^i\right) \tag{13c}$$

$$\mathcal{C}^i\left(x_t^i\right) \cap \mathcal{C}^j\left(x_t^j\right) = \emptyset \ \forall j \in \mathcal{N}\left(x_t^i\right) \tag{13d}$$

$$x_0^i \in \mathcal{X}_i \tag{13e}$$

$$x_T^i \in \mathcal{X}_g \tag{13f}$$

where $Q \succ 0$ and $R \succ 0$ are positive definite matrices and constraints (13b)-(13d) must be enforced at all times. $B\left(x_t^i\right)$ and $C\left(x_t^i\right)$ are given by

$$B\left(x_t^i\right) = \frac{\partial h\left(x_t^i\right)}{\partial x_i^t} g\left(x_t^i\right), \tag{14a}$$

$$C\left(x_t^i\right) = -\frac{\partial h\left(x_t^i\right)}{\partial x_i^t} f\left(x_t^i\right) - \kappa h\left(x_t^i\right), \tag{14b}$$

$$h\left(x_t^i\right) = \left[h_s(x_t^{i,1}), \ldots, h_s(x_t^{i,N-1}),\ h_v\left(x_t^i\right)\right]^\top, \tag{14c}$$

$$h_s\left(x_t^{i,j}\right) = \left\|p_t^i - p_t^j\right\|_2^2 - r^2, \forall \quad j \in [1; N] \setminus i \tag{14d}$$

$$h_v\left(x_t\right) = \bar{A}_{k\times k}\left(x_t\right) \tag{14e}$$

where $h_s(x_t^{i,j})$ 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. In addition, $h_v(x_t^i)$ avoids the problem of deadlocking as $v_t \in \mathscr{C}_\ell(t)$ implies non-existence of a social mini-game. The inverse, however, may not hold true. That is, $v_t \notin \mathscr{C}_\ell(t)$ may not necessarily result in a social mini-game. For example, two agents equidistant from a doorway with equal speeds may prefer to travel parallel to each other. Therefore, perturbing agents' speeds via Equation 8 only if $v_t \notin \mathscr{C}_\ell(t)$ may be an overly conservative approach. We thus introduce the liveliness function $\ell_j\left(p_t^i, v_t^i\right)$ for agent i with respect to the agent j as follows:

$$\ell_j(p_t^i, v_t^i) = \cos^{-1}\left(\frac{\left\langle p_t^i - x_t^j, v_t^i - v_t^j\right\rangle}{\left\|p_t^i - x_t^j\right\| \left\|v_t^i - v_t^j\right\| + \epsilon}\right) \tag{15}$$

where $\langle v_t^i, v_t^i \rangle$ denotes the dot product between vectors v_t^i and v_t^j and $\epsilon > 0$ is to ensure that the denominator is non-negative. Note that $\ell_j\left(p_t^i, v_t^i\right) \in [0, \pi]$ and $\ell_j\left(x_t^i, v_t^i\right) < \ell_{\text{thresh}}$ implies that i and j are on a collision course. We claim that $\ell_j\left(x_t^i, v_t^i\right) < \ell_{\text{thresh}}$ is a sufficient condition to establish the existence of a social mini-game since according to Definition 1, a social mini-game implies that at $t = a$, there is a collision marking the beginning of the social mini-game. That is, $\exists x_t^i, x_t^j$ for $t < a$ such that $\ell_j\left(x_t^i, v_t^i\right) < \ell_{\text{thresh}}$.

In contrast to traditional CBFs where the CBF $h(x_t^i)$ is usually a function of spatial coordinates such as position only, our proposed CBF $h(x_t^i)$ in (14c) is a function of both position $h_s(x_t^{i,j})$ (defined in (14d)) and velocity $h_v(x_t)$ (defined in (14e)). Unifying these constraints in the formulation of CBF enables robots to simultaneously prevent collisions and deadlocks. Furthermore, for cases of single integrator dynamics where v_t^i is a input, the constraint Av_t (where $v_t = [v_t^1,\ v_t^2]^{\mathrm{T}}$ for two agents) can directly be incorporated in the optimization problem. However in cases of double or higher integrator robotic systems (such as bipedal robots [45], 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 non trivial.

Theorem 1. *Equation 8 is a mixed-strategy Nash Equilibrium solution.*

Proof. It is known that the classical Chicken game has a mixed-Nash solution when both agents "swerve". We consider the deadlock resolution strategy as solving a version of the Chicken game-if neither agent perturbs their state, then a social mini-game will occur, resulting in a deadlock. The only other alternatives are for either or both of the agents to perturb their current state. As the optimal perturbation solution to Equation 8 requires *both* the agents to perturb their velocities jointly, it corresponds to the mixed-Nash solution of the Chicken game. □

Remark: For $v_t^i \neq v_t^j$ for all i, j, there is always a unique mixed-Nash solution. 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. If $v_t^i = v_t^j$ for some i, j, then there are multiple mixed-Nash solutions and we implement the following tie breaking protocol.

Tie-breaking via Priority Orderings 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 5. ***Priority Orderings*** **(σ):** *A priority ordering is a permutation $\sigma : \mathcal{C}_\varphi^t \to [1, k]$ over the set $\mathcal{C}_\varphi^t$. For any $i, j \in [1, k]$, $\sigma^i = j$ indicates that the i^{th} robot will move on the j^{th} 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 6. ***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

$$b^{i,*} = \arg\min_{b^i} \left(\alpha_{r^i(b^i)} - p^i(b^i)\right) \tag{16}$$

It is known [46], [47] 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.

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.

V. Evaluation and Discussion

In this section, we deploy our approach in both simulated as well as real world social mini-games occurring at doorways, hallways, intersections, and roundabouts, and investigate the following questions–(i) is the proposed navigation algorithm better at avoiding deadlocks and collisions than the state-of-the-art? (ii) how does the proposed navigation system compare to different approaches such as multi agent reinforcement learning? (iii) is our controller robust to the choice of different kinodynamic local planners? and finally, (iv) how does our game-theoretic deadlock resolution strategy compare to alternative perturbation strategies?

A. 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 [48] for solving the MPC optimization. We consider the following differential drive robot:

$$\begin{bmatrix} \dot{p}^{i,1} \\ \dot{p}^{i,2} \\ \dot{\phi}^{i} \\ \dot{v}^{i} \\ \dot{\omega}^{i} \end{bmatrix} = \begin{bmatrix} \cos\left(\phi^{i}\right) & 0 & 0 & 0 \\ \sin\left(\phi^{i}\right) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/m & 0 \\ 0 & 0 & 0 & I^{-1} \end{bmatrix} \begin{bmatrix} v^{i} \\ \omega^{i} \\ u_1 \\ u_2 \end{bmatrix}, \quad (17)$$

where subscript i denotes the i^{th} agent, m and I are the mass and inertia respectively, $p^{i,1} \in \mathbb{R}$ and $p^{i,2} \in \mathbb{R}$ represent the horizontal and vertical positions of the robot, $\phi^i \in \mathbb{S}^1$ represents its orientation, $[v^i,\ \omega^i] \in \mathcal{U}^i$ are the linear and angular velocity of the robot respectively and $u^i = [u_1,\ u_2]^{\mathrm{T}}$ is the control input. Let the discrete dynamics of (17) be given by

$$x^i_{t+1} = f\left(x^i_t\right) + g\left(x^i_t\right) u^i_t \quad (18)$$

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

$$\begin{aligned} \min_{u_{1:T-1}} \quad & \sum_{t=0}^{T-1} {x^i_t}^{\top} Q x^i_t + {u^i_t}^{\top} R u^i_t \\ \text{s.t} \quad & x^i_{t+1} = f\left(x^i_t\right) + g\left(x^i_t\right) u^i_t, \quad \forall\, t \in 1, \ldots, T-1 \\ & x_t \in \mathcal{X},\ u_t \in \mathcal{U}^i \quad \forall\, t \in \{1, \ldots, T\} \end{aligned} \quad (19)$$

The safety for the differential drive robot is guaranteed by the CBF constraint (11) For each agent, the CBF for the obstacles is characterized by $h_s\left(x^{i,m}_t\right)$ given by

$$h_s\left(x^{i,m}_t\right) = \left(p^{i,1} - c^{1,m}\right)^2 + \left(p^{i,1} - c^{2,m}\right)^2 - r^2 \quad (20)$$

where $r > 0$, x^i_t is specified by $p^i_t, \theta^i_t, v^i_t, \omega^i_t \in \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^2 \times \mathbb{R}$ representing the current position, heading, linear and angular rates of the i^{th} robot and $(c_{1,m}, c_{2,m})$ is the center for circle of radius $r > 0$ and $m \in \{1, \ldots, M\}$ where M is sufficiently large to cover all the obstacles. Therefore, the points lying in the safe region are characterized by the set $\mathcal{X}$ given by

$$\mathcal{X} = \left\{x^i_t :\ h_s\left(x^{i,m}_t\right) > 0,\ \forall\, m \in \{1, \ldots, M\}\right\} \quad (21)$$

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 14d. The game theoretic CBF $h_v\left(x_t\right)$ for an agent i is given by

$$h_v(x^i_t) = A v_t, \quad i \neq j \quad (22)$$

where $v_t = [v^i_t,\ v^j_t]^{\top}$, $A \in \mathbb{R}^{2\times 2}$. For our doorway and intersection simulations, we choose $A = \begin{bmatrix} -1 & -2 \\ -2 & -1 \end{bmatrix}$.

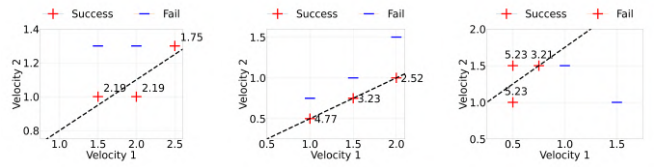
B. Environments and Hardware

Real World–We have designed and tested our approach in two social mini-games at a doorway and intersection in a 3 meters by 3 meters space. Our experiments included multi-robot tests with the UT Automata F1/10 car platforms, the Clearpath Jackal, and the Boston Dynamics Spot, as well as human-robot tests with the Jackal. We test three distinct robots: the UT Automata F1/10 car platforms, the Clearpath Jackal, and the Boston Dynamics Spot. We selected these robots for their varying shape, size, and kinodynamic parameters. The Spot is a legged robot, while the Jackal and the F1/10 cars are wheeled. Additionally, the Jackal can make point turns, while the cars cannot. The Spot and the Jackal can move at maximum speeds of 1.5 meters per second whereas the UT Automata F1/10 cars can travel up to 9 meters per second. We experiment with various configurations with the exception of using the Spot and the Jackal with the F1/10 cars; the cars are not detected by the bigger robots on account of their low height. We compare our navigation algorithm with a classical DWA planner.
Simulation–We compare our approach with NH-TTC [8] and NH-ORCA [11]. We also compare our controller with learning-based baselines in the SocialGym 2.0 multi-agent social navigation simulator [49], which is available on Github[2]. We benchmark multi-agent reinforcement learning (MARL) baselines such as CADRL, CADRL(L), Enforced Order. In addition, we also simulate a classical planner using the dynamic window approach as well as an auction-based appproach [9]. CADRL [2] and its LSTM-based variant, which we denote as CADRL(L), are state-of-the-art multi-agent social navigation methods. CADRL and CADRL(L) use a reward function where a robot is rewarded upon reaching the goal and penalized for getting too close or colliding with other robots as well as taking too long to reach the goal. Enforced Order is a MARL baseline that encourages robots to engage in social behaviors such as queue formations. We train these baselines using PettingZoo and Stable Baselines3 [50] and report results across a range of social navigation metrics. The metrics, each averaged across the number of episodes include success rate, time still, and average Δ velocity. The time still measures the number of times the robots had to stop and the average Δ velocity measures the numbers of times the robots had to deviate from their previous velocity.

C. 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 Figure 3a, 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

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



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

Fig. 3: **Results on 2 F1/10 car platforms, Jackal, and the Spot robots:** We record successful and failed trials along with the makespan in the case of successful trials. Figures 3a and 3b use F1/10 cars while Figure 3c shows results for the Jackal ("velocity 1") versus the Spot ("velocity 2"). We notice that robots succeed when the velocities satisfy Equation 9.

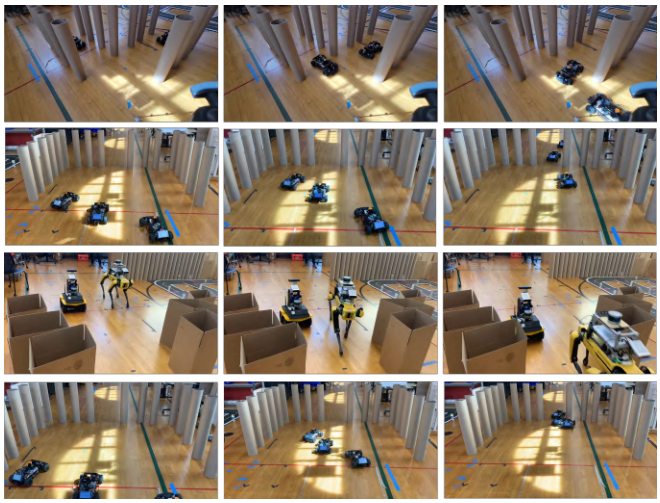
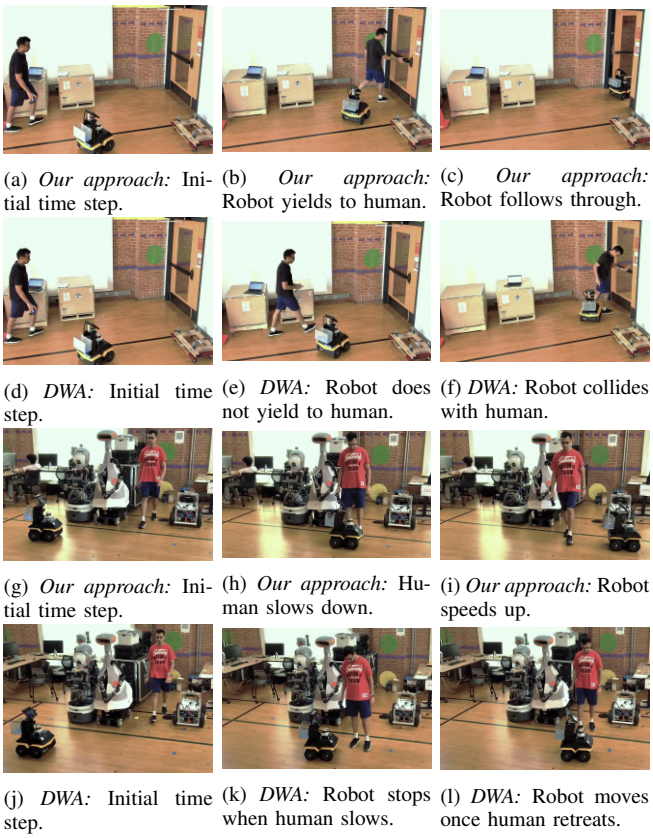


Fig. 4: **Deployment in multi-robot scenarios:** *(rows 1, 2, 3)* We demonstrate our approach at a corridor intersection and doorway using the F1/10 cars, Spot, and the Jackal robots. *(row 4)* We show decentralized local planning *without* game-theoretic deadlock resolution.

which 3 succeeded and 2 failed. We observed that the trials succeeded when the cars' velocities satisfied Equation 9, whereas the trials failed when the velocities did not satisfy Equation 9.

We repeated this experiment in the intersection scenario as shown in Figure 3b and obtained an identical liveness set. With 3 F1/10 robots we obtained a liveness set similar to Equation 10.

Finally, we show that the liveness set given by Equation 9 for the social mini-game at doorways holds for cross robot platforms. In Figure 3c, 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 the SocialGym 2.0 simulator and observed similar results. That is, our approach yields a 76% and 96% success rate with the $1/\zeta^i$ velocity scaling, but these rates drop to 48% and 44%, respectively, when the scaling used is $\left(1 - \frac{\zeta^i}{5\text{V\_MAX}}\right)^{-1}$.



(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. 5: **Deployment in human-robot scenarios:** *(doorway)* Both the human and robot start from equidistant positions from the door and the goal is to go through the doorway without colliding or deadlocking. *(Doorway)* Figures 5a, 5b, and 5c demonstrate our game-theoretic 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 5d, 5e, and 5f demonstrates a baseline DWA planner that results in the robot colliding with the human. *(Intersection)* Figures 5j, 5k, and 5l demonstrate our game-theoretic deadlock avoiding strategy enables the robot to proactively speed up when it notices the human slow down. Figures 5d, 5e, and 5f 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.

D. Real World Experiments

Multi-robot setting–In the multi-robot experiments (Figure 4), for the doorway scenario, we set the gap size to approximately 0.5 meters. In the corridor intersection scenario, we determined that the width of the four arms of the intersection should be between 1.5 and 2 meters, with the conflict zone at the center of the intersection having an area of 2.5 to 4 square meters. In the doorway scenario, all robots start from one side of the doorway, either equidistant from the gap at a distance of approximately 1.8 meters or using staggered starting positions, and their objective is to move to the other side. In the corridor intersection scenario, we test standard autonomous intersection navigation with one robot on each arm of the intersection at any time. The goal here is to navigate the intersection safely. A robot on one arm of the intersection can select any of the 3 remaining arms to go

to.

Additionally, we have found that our proposed navigation system is comparable to human navigation on average. According to several studies analyzing human walking speeds and makespan times at intersections [51] and doorways [52], humans typically move at an average speed of 1.4 meters per second at busy intersections at a flow rate of 4 $(\text{ms})^{-1}$[3]. When we compare this with our results in the doorway and intersection scenarios, we found that the robots moved at an average speed of approximately $1.25-1.5$ meters per second at a flow rate of $2.8-3.3$ $(\text{ms})^{-1}$.

Human-robot setting–In the human-robot experiments (Figure 5), we replicate the multi-robot test performed at the doorway with identical configurations and goals. We conducted experiments to evaluate the effectiveness of our navigation algorithm compared to a classical DWA planner. The multi-robot results are reported in Figures 3 and 4. The results for the DWA planner are not reported as the robots failed 100% of the time without any sort of coordination between them. Here, we define a failure as not completing the mini-game due to collisions or deadlocks. Each graph in Figure 3 reports a successful or a failed trial, along with the makespan times (time to goal for the last robot to complete the mini-game) for each successful trial. In Figure 5, Figures 5a, 5b, and 5c demonstrate our game-theoretic 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 5d, 5e, and 5f demonstrates a baseline DWA planner that results in the robot colliding with the human.

E. Simulation Results

We implement the game-theoretic MPC-CBF controller using double-integrator system dynamics in Equation 1, cost function in Equation 13, and constraints in Equation 14. We demonstrate game-theoretic MPC-CBF in a doorway and intersection scenario in Figures 6a and 6e. The red and and green agents are traveling from the left towards the right (and south to north) through the narrow doorway (and intersection). We observe that the green agent yields to the red agent, allowing it to pass first, and then immediately follows the red agent. In the intersection scenario, we see a similar phenomenon.

We compare our approach with conventional MPC-CBF approach in Figures 6b and 6f and show that the two agents end up in a deadlock situation in the future because of symmetry induced by the environment. Our method averts the deadlock by subtly altering the velocity of both agents in such a way that their velocities ultimately coincide with the Nash equilibrium strategy, mutually beneficial for both agents. Furthermore, our approach employs a liveness function as an early warning heuristic to anticipate potential deadlock situations. The aforementioned subtle changes in velocities are triggered only when the liveness function of each agent falls below a specified threshold. Furthermore, in contrast to standard perturbation strategies that perturb the position of each agent to avoid deadlock, we employ minimal changes to each agent to avoid deadlocks. Additionally, we show that motion planners that guarantee collision-free navigation end up in deadlocks. In Figures 6c, 6g, 6d, and 6h, we demonstrate NH-ORCA and NH-TTC in the doorway environment. The symmetry in the environment results in empty feasible action sets for the agents, resulting in a deadlock which is impossible to resolve without implicit or explicit perturbation strategies.

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

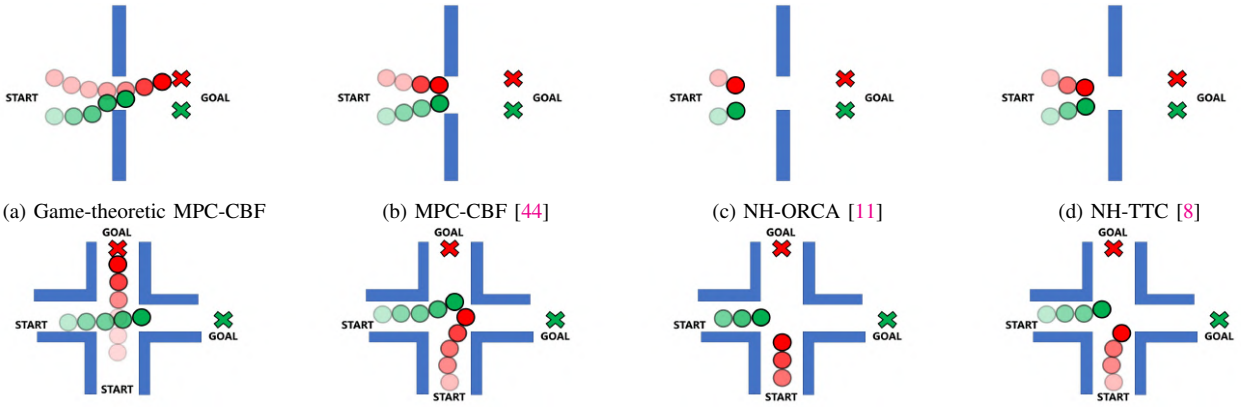
	Baseline	Success Rate	Coll. Rate	Stop Time	Avg. ΔV
DOOR	CADRL [2]	32 ± 3.125	0.00 ± 0.000	222 ± 2.749	13 ± 1.905
	CADRL(L) [3]	0 ± 0.000	0.12 ± 0.000	436 ± 0.318	36 ± 3.042
	Enforced Ordering	44 ± 2.993	0.16 ± 0.243	617 ± 1.803	117 ± 0.686
	DWA	4 ± 0.244	2.32 ± 0.589	166 ± 4.496	30 ± 2.506
	Auction-based	76 ± 2.270	0.17 ± 0.012	110 ± 4.100	6 ± 1.111
	Game-theoretic QP-CBF	**100±0.000**	**0.00±0.000**	**25±4.235**	**2±0.000**
	Game-theoretic MPC-CBF	**100±0.000**	**0.00±0.000**	**18±1.067**	**2±0.098**
INTER.	CADRL [2]	20 ± 0.453	1.20 ± 0.634	389 ± 5.153	28 ± 1.141
	CADRL(L) [3]	28 ± 2.527	0.32 ± 1.393	267 ± 6.453	127 ± 2.357
	Enforced Ordering	56 ± 0.718	0.80 ± 0.894	233 ± 1.729	104 ± 5.041
	DWA	16 ± 1.765	2.48 ± 0.57	640 ± 4.583	97 ± 2.495
	Auction-based	96 ± 3.342	0.14 ± 0.003	139 ± 5.233	6 ± 0.470
	Game-theoretic QP-CBF	**100± 0.000**	**0.00± 0.000**	**46± 2.349**	**2± 0.040**
	Game-theoretic MPC-CBF	**100± 0.000**	**0.00± 0.000**	**27± 0.140**	**0± 0.000**

TABLE III: **Comparing Game-theoretic MPC-CBF versus baselines:** We compare with MARL baselines and DWA, which is a baseline method where only the local motion planner is used without scheduling. CADRL [2] and its LSTM variant [3] are state-of-the-art RL-based social navigation algorithms and Enforced Order uses sub-goal reward policies that encourage queue formation. **Conclusion:** Navigation with game-theoretic certificates results in smaller changes to the average Δ velocity and higher success rates.

In Table III, we compare our algorithm with DWA and multi-agent reinforcement learning baselines. We make 4 observations. First, note that not only does our approach outperform all other baselines, but the second best performing baseline in the MARL category is the auction-based approach, which attempts to learn a type of priority ordering, signaling that determining an optimal coordination strategy is indeed the key to success in multi-robot social navigation. Next, we observe that the DWA approach performs the worst–nearly failing in the doorway–which agrees with our real world experiments where the local navigation without the priority ordering failed 100% of the time. Next, our approach results in the smallest average change in velocity highlighting our methods strong smoothness properties. Finally, the last two rows in Table III show that the game-theoretic deadlock resolution is not restricted to a CBF-based controller. One can add refine the system by adding different controllers without any noticeable change in performance.

F. Comparing with Alternate Perturbation Strategies

We compare our game-theoretic deadlock resolution method (Game-theoretic MPC-CBF) with [12] using random perturbation, a variant of [12] where we implement our game-theoretic approach, with an approach based on MPC and buffered voronoi cells [10], and finally, with an ORCA-based approach that uses multi-agent path finding (MAPF) to resolve deadlocks. We perform experiments in social mini-games at doorways, hallways, and intersections, and report



(a) Game-theoretic MPC-CBF (b) MPC-CBF [44] (c) NH-ORCA [11] (d) NH-TTC [8]

(e) Game-theoretic MPC-CBF (f) MPC-CBF [44] (g) NH-ORCA [11] (h) NH-TTC [8]

Fig. 6: **Liveness properties in various navigation methods** *(doorway)* Our approach, Game-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 intersection scenario. *Conclusion:* Our game-theoretic controller presented in Section IV encourages queue formation thereby resolving a deadlock situation in a decentralized and smooth manner.

	Baseline	Avg. ΔV	Path Deviation	Makespan Ratio
DOORWAY	Random QP-CBF [12]	0.38 ± 0.12	1.874 ± 0.28	2.68 ± 1.00
	ORCA-MAPF [5]	0.10 ± 0.00	0.00 ± 0.00	3.04 ± 0.00
	Game-theoretic QP-CBF [12]	0.22 ± 0.01	0.145 ± 0.00	3.37 ± 0.01
	IMPC-DR [10]	0.08 ± 0.00	0.160 ± 0.00	1.53 ± 0.00
	Game-theoretic MPC-CBF	$\mathbf{0.001 \pm 0.00}$	$\mathbf{0.089 \pm 0.02}$	$\mathbf{1.10 \pm 0.00}$
INTERSECTION	Random QP-CBF [12]	0.30 ± 0.10	0.400 ± 0.14	0.99 ± 0.04
	ORCA-MAPF [5]	0.25 ± 0.00	0.00 ± 0.00	2.22 ± 0.00
	Game-theoretic QP-CBF [12]	0.29 ± 0.05	0.111 ± 0.04	2.24 ± 0.04
	IMPC-DR [10]	0.08 ± 0.00	0.151 ± 0.0	1.13 ± 0.00
	Game-theoretic MPC-CBF	$\mathbf{0.002 \pm 0.00}$	$\mathbf{0.066 \pm 0.01}$	$\mathbf{1.05 \pm 0.01}$
HALLWAY	Random QP-CBF [12]	0.055 ± 0.01	0.327 ± 0.32	1.16 ± 0.04
	ORCA-MAPF [5]	0.11 ± 0.00	1.99 ± 0.00	1.03 ± 0.00
	Game-theoretic QP-CBF [12]	0.008 ± 0.00	0.190 ± 0.00	1.44 ± 0.04
	IMPC-DR [10]	0.135 ± 0.00	0.194 ± 0.00	2.08 ± 0.00
	Game-theoretic MPC-CBF	$\mathbf{0.001 \pm 0.00}$	$\mathbf{0.047 \pm 0.00}$	$\mathbf{1.04 \pm 0.00}$

TABLE IV: **Comparing alternate perturbation strategies [12]:** We observe that alternate perturbation strategies incur a higher cost and path deviation compared to game-theoretic perturbation.

the mean $\pm$ standard deviation for average change in velocity (ΔV), path deviation (meters), and average time steps across 2 agents in Table IV. ΔV measures how much an agent needs to modulate their velocity in order to prevent a deadlock. The path deviation measures the difference between the preferred path $\widetilde{\Gamma}^i$ and executed path, Γ^i.

For all three settings, game-theoretic deadlock resolution results in a smaller deviation across velocities and a smaller path deviation, which implies that our approach allows agents to modulate their velocities more efficiently that results in a trajectory that more closely resembles the preferred trajectory. 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 constraints in space. In the intersection and hallway settings, however, game-theoretic perturbation is slower or comparable since random perturbation agents move faster, albeit along inefficient trajectories.

VI. Conclusion, Limitations, and Future Work

In this work, we presented an approach to address the challenges of safe and deadlock-free navigation in decentralized multi-robot systems operating in constrained environments. We introduced the notion of *social mini-games* to formalize the multi-robot navigation problem in constrained spaces. We proposed a novel class of decentralized controllers capable of guaranteeing both safety and deadlock resolution by achieving a game-theoretic Nash equilibrium. Our controller relies on two critical insights: first, we reduced the deadlock resolution problem to a modified version of an N-player Chicken game, for which a Nash equilibrium solution exists. Second, we formulated the Nash equilibrium as CBF and integrated it with traditional CBFs to inherit their safety guarantees.

The main limitation of this work is that the approach assumes the liveness sets are known apriori. Liveness sets can, however, be estimated either in simulation or using learning-based methods for generating cost maps for navigation. Moving forward, further research can be conducted to explore the scalability and applicability of our approach in larger-scale multi-robot systems. Additionally, investigating the integration of machine learning techniques to enhance decision-making capabilities and adaptability of the decentralized controllers could be a fruitful direction for future work.

References

[1] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, "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, 2018.

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

[3] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10357–10377, 2021.

[4] M. Everett, Y. F. Chen, and J. P. How, "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, 2018.

[5] S. Dergachev and K. Yakovlev, "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, 2021.

[6] A. Kamenev, L. Wang, O. B. Bohan, I. Kulkarni, B. Kartal, A. Molchanov, S. Birchfield, D. Nistér, and N. Smolyanskiy, "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, 2022.

[7] S. Le Cleac'h, M. Schwager, and Z. Manchester, "Algames: a fast augmented lagrangian solver for constrained dynamic games," *Autonomous Robots*, vol. 46, no. 1, pp. 201–215, 2022.

[8] B. Davis, I. Karamouzas, and S. J. Guy, "Nh-ttc: A gradient-based framework for generalized anticipatory collision avoidance," *arXiv preprint arXiv:1907.05945*, 2019.

[9] R. Chandra, R. Menon, Z. Sprague, A. Anantula, and J. Biswas, "Decentralized social navigation with non-cooperative robots via bi-level optimization," *arXiv preprint arXiv:2306.08815*, 2023.

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

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

[12] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

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

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

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

[16] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," *arXiv preprint arXiv:2011.00583*, 2020.

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

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

[19] X. Xiao, Z. Xu, Z. Wang, Y. Song, G. Warnell, P. Stone, T. Zhang, S. Ravi, G. Wang, H. Karnan, *et al.*, "Autonomous ground navigation in highly constrained spaces: Lessons learned from the benchmark autonomous robot navigation challenge at icra 2022 [competitions]," *IEEE Robotics & Automation Magazine*, vol. 29, no. 4, pp. 148–156, 2022.

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

[21] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, online collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1047–1054, 2017.

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

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

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

[25] N. Suriyarachchi, R. Chandra, J. S. Baras, and D. Manocha, "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, 2022.

[26] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[27] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: a survey," *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, 2022.

[28] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game-theoretic planning for self-driving cars in multivehicle competitive scenarios," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1313–1325, 2021.

[29] A. Britzelmeier, A. Dreves, and M. Gerdts, "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, 2019.

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

[31] W. Schwarting, A. Pierson, S. Karaman, and D. Rus, "Stochastic dynamic games in belief space," *IEEE Transactions on Robotics*, pp. 1–16, 2021.

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

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

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

[35] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "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, 2020.

[36] B. Di and A. Lamperski, "Differential dynamic programming for nonlinear dynamic games," *arXiv preprint arXiv:1809.08302*, 2018.

[37] B. Di and A. Lamperski, "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, 2019.

[38] B. Di and A. Lamperski, "First-order algorithms for constrained nonlinear dynamic games," *arXiv preprint arXiv:2001.01826*, 2020.

[39] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *AAAI*, vol. 4, pp. 709–715, 2004.

[40] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[41] J. J. Park, C. Johnson, and B. Kuipers, "Robot navigation with model predictive equilibrium point control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4945–4952, IEEE, 2012.

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

[43] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, online collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1047–1054, 2017.

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

[45] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3d bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.

[46] T. Roughgarden, *Twenty lectures on algorithmic game theory*. Cambridge University Press, 2016.

[47] R. Chandra and D. Manocha, "Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2676–2683, 2022.

[48] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[49] J. Holtz and J. Biswas, "Socialgym: A framework for benchmarking social robot navigation," *arXiv preprint arXiv:2109.11011*, 2021.

[50] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[51] R. Akcelik, "An investigation on pedestrian movement characteristics at mid-block signalised crossings," *Akcelik & Associaties Pty Ltd, Australia*, 2001.

[52] A. Garcimartín, D. R. Parisi, J. M. Pastor, C. Martín-Gómez, and I. Zuriguel, "Flow of pedestrians through narrow doors with different competitiveness," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2016, no. 4, p. 043402, 2016.