

Robustly Constrained Dynamic Games for Uncertain Nonlinear Dynamics

Shuyu Zhan^{1*}

Chih-Yuan Chiu^{2*}

Antoine P. Leeman³

Glen Chou⁴

Abstract—We propose a novel framework for robust dynamic games with nonlinear dynamics corrupted by state-dependent additive noise, and nonlinear agent-specific and shared constraints. Leveraging system-level synthesis (SLS), each agent designs a nominal trajectory and a causal affine error feedback law to minimize their own cost while ensuring that its own constraints and the shared constraints are satisfied, even under worst-case noise realizations. Building on these nonlinear safety certificates, we define the novel notion of a robustly constrained Nash equilibrium (RCNE). We then present an Iterative Best Response (IBR)-based algorithm that iteratively refines the optimal trajectory and controller for each agent until approximate convergence to the RCNE. We evaluated our method on simulations and hardware experiments involving large numbers of robots with high-dimensional nonlinear dynamics, as well as state-dependent dynamics noise. Across all experiment settings, our method generates trajectory rollouts which robustly avoid collisions, while a baseline game-theoretic algorithm for producing open-loop motion plans failed to generate trajectories that satisfy constraints.

I. INTRODUCTION

To operate reliably and efficiently, autonomous robots deployed in real-world applications must design and enact robust motion plans that account for both the future motion of surrounding agents and the impact of unpredictable disturbances on system safety. Recently, dynamic game theory has emerged as a versatile framework for jointly modeling prediction and motion planning in multi-agent interactions [1]–[5]. The core insight of the dynamic games literature is that the interaction outcome between self-interested agents can be captured by computing the *Nash equilibrium*, a strategy profile at which each agent selfishly best-responds to the actions of all other agents while satisfying a prescribed set of constraints. Across a wide range of robotics applications, the Nash equilibria of dynamic games has indeed been shown to describe nuanced interactions among self-interested agents *with deterministic, noise-free dynamics* [2], [3], [6]–[8].

During real-world deployment, however, robots inevitably encounter sources of uncertainty in their dynamics, such as wind disturbances or friction, which may compromise their safety and constraint satisfaction. Unfortunately, most existing dynamic game formulations either do not explicitly account for noisy dynamics [2], [3], [8], [9], and generate

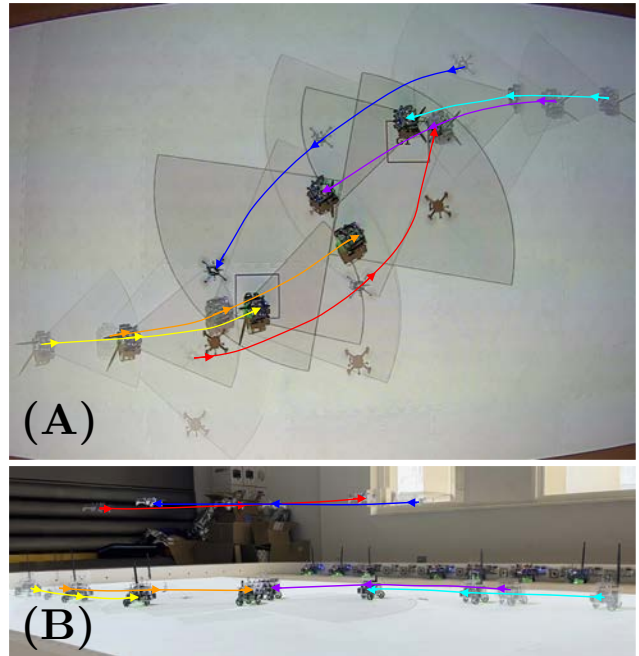


Fig. 1: (A) Top and (B) Side views of a hardware experiment in which two heterogeneous robot teams, with 1 quadcopter and 2 ground robots each, navigate within a shared environment. Within each team, one ground robot follows a second, which in turn follows the quadcopter. Our method generated interactive motion plans for all robots which obeyed prescribed proximity, line-of-sight, and collision constraints despite dynamics noise (see Sec. VI-F).

trajectories that fail to robustly satisfy safety constraints (see Sec. VI of our work), or prescribe overly conservative or defensive motion plans which ensure constraint satisfaction while greatly sacrificing computational efficiency [10].

To ensure robust constraint satisfaction in interactive motion planning despite unknown noise realizations, we construct a novel algorithm for multi-agent prediction and trajectory generation which integrates key techniques from the robust control literature within the dynamic games framework. By coupling motion planning and robust control, our method designs trajectories and error feedback policies that achieve robust constraint satisfaction, yet are not overly conservative. Concretely, we make the following contributions:

- **Robust Dynamic Games via System Level Synthesis (SLS):** We extend the *system-level synthesis* (SLS) framework for robust control to a class of multi-agent dynamic games with nonlinear dynamics corrupted with state-dependent additive noise, agent-specific con-

*Equal contribution.

^{1,2,4}Georgia Institute of Technology, Schools of ¹Interactive Computing (szhan45@gatech.edu), ²Electrical and Computer Engineering (cyc@gatech.edu), ⁴Aerospace Engineering and Cybersecurity & Privacy (chou@gatech.edu)

³Institute for Dynamic Systems and Control, ETH Zürich (aleeman@ethz.ch).

This work is supported by the European Space Agency OSIP 4000133352.

straints, and shared team constraints. Leveraging recent robust nonlinear SLS results, we jointly optimize the nominal trajectories, robust controllers, and a linearization error upper bound across all agents. We obtain tractable safety certificates through A SLS-style robust constraint satisfaction guarantees.

- **Formulation and Computation of the Robustly Constrained Nash Equilibrium (RCNE):** We introduce the novel concept of a *robustly constrained Nash equilibrium* (RCNE), which describes interactions in which each agent best responds to all other agents' actions while robustly satisfying a set of prescribed constraints. We develop a novel Iterative Best Response (IBR)-based algorithm to compute the RCNE, which greatly exceeds centralized approaches (e.g., [2, Alg. 1]) in computational efficiency when applied to the multi-agent robust games studied in this work (Remark 3).
- **Empirical validation:** On a broad range of multi-robot tasks, our approach produced trajectories and controllers that enabled multi-robot teams to interact while avoiding collisions and high-disturbance regions. Our method scales to interactive scenarios involving up to 24 robots, and can accommodate heterogeneous robot teams with multiple air and ground vehicles. Our method generated motion plans which satisfied all prescribed constraints across all simulation outcomes, while a dynamic games baseline produced open-loop trajectories which often violated constraints under additive dynamics noise.

Notation: Given integers N_1, N_2 with $0 \leq N_1 < N_2$, we define $[N_1, N_2] := \{N_1, \dots, N_2\}$ and $[N_2] := \{1, \dots, N_2\}$. Given $S \subseteq \mathbb{R}^n$ and $M \in \mathbb{R}^{m \times n}$, we define $MS := \{Mx : x \in S\}$. Given $M_1, \dots, M_K \in \mathbb{R}^{m \times n}$, we denote by $\text{diag}\{M_1, \dots, M_K\} \in \mathbb{R}^{mK \times nK}$ the block diagonal matrix with M_k as the k -th block, $\forall k \in [K]$. We denote the horizontal concatenation of M_1, \dots, M_K by $[M_1 \ \dots \ M_K] \in \mathbb{R}^{m \times Kn}$. For any $m, n \in \mathbb{N}$, we denote the $m \times m$ identity matrix by I_m and the $m \times n$ zero matrix by $O_{m \times n}$, with subscripts omitted when dimensions are clear from context. Given $m \in \mathbb{N} \cup \{+\infty\}$ and $k \in \mathbb{N}$, we denote by \mathbb{B}_m^k the unit m -norm ball in \mathbb{R}^k . ∇^2 and ∇ respectively refer to the Hessian matrix and (vertical) gradient vector of a function, taken with respect to all variables.

II. RELATED WORKS

Our work draws from and contributes to the rich literature on game-theoretic multi-agent motion prediction and planning. In particular, [1], [8] designed state feedback policies under the feedback information setting, while [2], [3], [9] computed open-loop, interactive multi-agent trajectories. However, whereas [1]–[3], [8], [9] all describe agent motion via deterministic dynamics models, our work considers agent dynamics corrupted by state-dependent noise. Moreover, for each agent, we design a trajectory and controller that satisfy safety constraints despite worst-case noise realizations.

Many works in the game-theoretic motion planning literature incorporate dynamics or agent intent uncertainty within their models. For instance, [5], [11] describe interactions

with agents who have uncertain intent, while [4] presents dynamic zero-sum games-based methods to perform motion prediction and planning in the presence of an adversarial agent. [12], [13] present game-theoretic frameworks, with noise-corrupted dynamics, in which agents jointly perform multi-agent prediction, planning, and state estimation. However, unlike [4], [5], [11], which capture *uncertain or adversarial agent intent*, we introduce a robust prediction and motion planning framework which guarantees safety despite *uncertainty in agent dynamics*. [12]–[14] consider noise-corrupted agent dynamics but design open-loop trajectories without guarantees of robust constraint satisfaction. In contrast, our method designs trajectories and controllers for noise-corrupted nonlinear systems which provably satisfy prescribed constraints regardless of the noise realization.

On a technical level, our method parameterizes robust controllers via system-level synthesis (SLS) [15]. Under linear dynamics, multi-agent control via SLS has been studied in MPC formulations [16], [17], and games [14]. Meanwhile, [18], [19] extend SLS to nonlinear dynamics but only under affine constraints. Our framework extends [14], [16]–[19] to nonlinearly constrained multi-agent dynamic games.

III. PRELIMINARIES

Consider an N -agent, T -stage discrete-time dynamic game \mathcal{G} , in which $x_t^i \in \mathbb{R}^{n_i}$ and $u_t^i \in \mathbb{R}^{m_i}$ respectively represent the *state* and *control* vector of each agent $i \in [N]$ at each time $t \in [0, T]$. We use $x_t := (x_t^1, \dots, x_t^N) \in \mathbb{R}^n$ and $u_t := (u_t^1, \dots, u_t^N) \in \mathbb{R}^m$ to respectively denote the *system state* and *system control* at each time t , where $n := \sum_{i=1}^N n_i$ and $m := \sum_{i=1}^N m_i$. We also write $\mathbf{x}^i := (x_0^i, \dots, x_T^i)$ and $\mathbf{u}^i := (u_0^i, \dots, u_T^i)$ for each agent $i \in [N]$. We also write $\mathbf{x} := (x_0, \dots, x_T) \in \mathbb{R}^{n(T+1)}$ and $\mathbf{u} := (u_0, \dots, u_{T-1}) \in \mathbb{R}^{mT}$ to denote the system state trajectory and system control trajectory, respectively. Moreover, we set $u_T^i = 0 \ \forall i \in [N]$.

Each agent $i \in [N]$ is associated with an initial state $x_0^i = \bar{x}_0^i \in \mathbb{R}^{n_i}$, as well as noise-corrupted dynamics given by ¹:

$$x_{t+1}^i = f_t^i(x_t^i, u_t^i) + E_t^i(x_t^i)w_t^i, \quad (1)$$

where $f_t^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{n_i}$ denotes the dynamics map, $w_t^i \in \mathbb{R}^{n_i}$ denotes additive noise, and $E_t^i(x_t^i) \in \mathbb{R}^{n_i} \times \mathbb{R}^{n_i}$ describes state-dependent disturbance scaling.

Remark 1: Our dynamics model (1) can be straightforwardly adapted to incorporate parametric uncertainty in the agent dynamics, as considered in [19].

As specified below, each agent $i \in [N]$ aims to minimize their own overall cost, while satisfying both their individual constraints and a set of team constraints shared among the N agents. Each agent has stage-wise costs $L_t^i : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ and an overall cost $J^i : \mathbb{R}^{n(T+1)} \times \mathbb{R}^{mT} \rightarrow \mathbb{R}$, given by:

$$J^i(\mathbf{x}, \mathbf{u}) := \sum_{t=0}^T L_t^i(x_t, u_t). \quad (2)$$

We note that for each agent $i \in [N]$ and time $t \in [T]$, J^i and L_t^i depend in general on *all agents'* control decisions and

¹Our experiments also model uncertainty in x_0 , which can be incorporated into the dynamics noise. For details, see [19].

resulting trajectories. Constraints specific to each agent i at each time t are defined by a collection of n_g^i scalar constraint functions $g_{t,k}^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}$ and $g_{T,k}^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}$ across all $k \in [n_g^i]$, as shown below:

$$g_{t,k}^i(x_t^i, u_t^i) \leq 0, \quad \forall t \in [T-1], k \in [n_g^i]. \quad (3)$$

Moreover, constraints shared by the N agents at each time $t \in [T]$ are defined by a collection of n_h scalar constraint functions $h_{t,k} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ and $h_{T,k} : \mathbb{R}^n \rightarrow \mathbb{R}$ across all $k \in [n_h]$, as shown below:

$$h_{t,k}(x_t, u_t) \leq 0, \quad \forall t \in [T-1], k \in [n_h]. \quad (4)$$

In our work, we assume the dynamic game \mathcal{G} under study is a *constrained dynamic potential game* with a corresponding potential function $P : \mathbb{R}^{n(T+1)} \times \mathbb{R}^{mT} \rightarrow \mathbb{R}$, as described in [3], [20] and defined below. Dynamic potential games provide a versatile and widely used modeling framework for motion prediction and trajectory generation in multi-agent robotics and controls applications [2], [3], [12].

Definition 1: The game \mathcal{G} is called a *constrained dynamic potential game* if there exists a *potential function* $P : \mathbb{R}^{n(T+1)} \times \mathbb{R}^{mT} \rightarrow \mathbb{R}$ such that, for any feasible trajectories $(\mathbf{x}, \mathbf{u}), (\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \mathbb{R}^{n(T+1)} \times \mathbb{R}^{mT}$, and any agent $i \in [N]$:

$$\begin{aligned} & P(\hat{\mathbf{x}}^i, \mathbf{x}^{-i}, \hat{\mathbf{u}}^i, \mathbf{u}^{-i}) - P(\mathbf{x}, \mathbf{u}) \\ &= J^i(\hat{\mathbf{x}}^i, \mathbf{x}^{-i}, \hat{\mathbf{u}}^i, \mathbf{u}^{-i}) - J^i(\mathbf{x}, \mathbf{u}). \end{aligned} \quad (5)$$

In words, P tracks the change in each agent i 's cost J^i corresponding to that agent's unilateral deviation from one state-control trajectory to another.

Assumption 1: \mathcal{G} is a constrained dynamic potential game.

IV. ROBUSTLY CONSTRAINED NASH EQUILIBRIUM (RCNE) AND PROBLEM FORMULATION

Here, we introduce core concepts in *system-level synthesis* to design robust controllers for dynamic potential games (Sec. IV-A), and formulate a novel notion of Nash equilibrium that accounts for robust constraint satisfaction in addition to optimal agent actions (Sec. IV-B). We formally present our problem statement at the end of Sec. IV-B.

A. Robust Controller Synthesis for Dynamic Games via SLS

To enable robust controller synthesis for dynamic games with noise-corrupted nonlinear dynamics, we leverage the system level synthesis (SLS)-based methods in [18]. Within the SLS framework, each agent $i \in [N]$ synthesizes a safe controller for the noise-corrupted dynamics (1) based on the decomposition of a nonlinear system into the sum of a *nominal* nonlinear system and a set of linear time-varying (LTV) error dynamics. Concretely, each agent $i \in [N]$ designs (a) a nominal state trajectory $\mathbf{z}^i := (z_0^i, \dots, z_T^i) \in \mathbb{R}^{n_i(T+1)}$, nominal control trajectory $\mathbf{v}^i := (v_0^i, \dots, v_{T-1}^i) \in \mathbb{R}^{m_i T}$, satisfying the *nominal dynamics*:

$$z_{t+1}^i = f_t^i(z_t^i, v_t^i), \quad \forall t \in [0, T-1], \quad (6)$$

and (b) a causal affine error feedback law of the form:

$$u_t^i = \pi_t^i(x_{0:t}^i) = v_t^i + \sum_{\tau=0}^{t-1} K_{t-1,\tau}^i (x_{t-\tau}^i - z_{t-\tau}^i). \quad (7)$$

Consider the linearization of (1) about Agent i 's nominal state-control trajectory $(\mathbf{z}^i, \mathbf{v}^i)$. We define the trajectory and control error, $\forall i \in [N], t \in [T]$, by $\Delta x_t^i := x_t^i - z_t^i \in \mathbb{R}^{n_i}$ and $\Delta u_t^i := u_t^i - v_t^i \in \mathbb{R}^{m_i}$ respectively. Next, we define the Jacobian matrices $A_t^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{n_i \times n_i}$ and $B_t^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{n_i \times m_i}$ for the dynamics (1) and the disturbance d_t^i , as follows. We note that d_t^i describes the impact of both the linearization error and the dynamics noise:

$$\begin{aligned} A_t^i(z_t^i, v_t^i) &:= \frac{\partial f_t^i}{\partial x_t^i}(z_t^i, v_t^i), \\ B_t^i(z_t^i, v_t^i) &:= \frac{\partial f_t^i}{\partial u_t^i}(z_t^i, v_t^i), \\ d_t^i &:= f_t^i(x_t^i, u_t^i) - A_t^i(z_t^i, v_t^i)\Delta x_t^i \\ &\quad - B_t^i(z_t^i, v_t^i)\Delta u_t^i + E_t^i(x_t^i)w_t^i. \end{aligned} \quad (8)$$

We can then express (1) in terms of the *nominal dynamics* (6) and the following LTV error dynamics:

$$\Delta x_{t+1}^i = A_t^i(z_t^i, v_t^i)\Delta x_t^i + B_t^i(z_t^i, v_t^i)\Delta u_t^i + d_t^i, \quad (9a)$$

$$\Delta u_t^i = \sum_{\tau=0}^{t-1} K_{t-1,\tau}^i \Delta x_{t-\tau}^i, \quad (9b)$$

where (9b) rewrites (7) using notation of the form Δx_t^i and Δu_t^i . We stack the state and control error as $e_t^i := (\Delta x_t^i, \Delta u_t^i) \in \mathbb{R}^{n_i+m_i}$.

A core tenet of the SLS framework is that, for each agent $i \in [N]$, all causal affine error feedback gains

$$K^i := \{K_{t,\tau}^i : t, \tau \in [0, T-1], t \geq \tau\} \quad (10)$$

can be parameterized by a set of *system response matrices*:

$$\begin{aligned} \Phi^i &:= \{\Phi_{t,\tau}^i = [(\Phi^i)_{t,\tau}^x; (\Phi^i)_{t,\tau}^u] \in \mathbb{R}^{(n_i+m_i) \times n_i} : \\ &\quad i \in [N], t, \tau \in [0, T-1], t \geq \tau\} \end{aligned} \quad (11)$$

satisfying (13) (see [18, Sec. 3] [15, Sec. 2]). We formally present this parameterization below in Prop. 1.

Proposition 1: (Adapted from [18, Sec. 3B]) Suppose each agent $i \in [N]$ is associated with dynamics (1), a nominal trajectory $(\mathbf{z}^i, \mathbf{v}^i)$, and linearized error dynamics (9a). Then for each agent $i \in [N]$, the following hold true:

- 1) Let a causal affine error feedback law (9b) be given. Then there exists a system response Φ^i of the form (11), such that $\forall t \in [0, T-1]$, the rollout error $e_t^i = (\Delta x_t^i, \Delta u_t^i)$ satisfying (9) is a linear function of the disturbance terms $\{d_t^i : i \in [N], t \in [T]\}$, given by:

$$e_t^i = \begin{bmatrix} \sum_{\tau=0}^{t-1} (\Phi^i)_{t-1,\tau}^x d_\tau^i \\ \sum_{\tau=0}^{t-1} (\Phi^i)_{t-1,\tau}^u d_\tau^i \end{bmatrix} = \sum_{\tau=0}^{t-1} \Phi_{t-1,\tau}^i d_\tau^i. \quad (12)$$

- 2) Conversely, let a system response Φ^i of the form (11), encoding disturbance propagation as in (12), be given. If Φ^i satisfies (13) below, then it induces a unique causal affine error feedback of the form (9b) corresponding to the disturbance propagation described by Φ^i :

$$\begin{aligned} (\Phi^i)_{t+1,\tau}^x &= A_t^i(z_t^i, v_t^i)(\Phi^i)_{t,\tau}^x + B_t^i(z_t^i, v_t^i)(\Phi^i)_{t,\tau}^u, \\ &\quad \forall t, \tau \in [0, T-2], t \geq \tau, \end{aligned} \quad (13a)$$

$$(\Phi^i)_{t,t}^x = I_{n_i}, \quad \forall t \in [0, T-1]. \quad (13b)$$

We now present Prop. 2, a refinement of the characterization of the robust constraint satisfaction in [18, Prop. III.3] to the setting of non-linear constraints. Prop. 2 requires the following mild assumptions on the boundedness of the states x_t , controls u_t , noise terms w_t , and state-dependent maps $E_t^i(x_t^i) \in \mathbb{R}^{n_i \times n_i}$ in our dynamic game model, as well as the regularity of the dynamics and constraint maps f_t^i, g_t^i , and h_t , across agent and time indices.

Assumption 2: Let $\Omega \subset \mathbb{R}^{n+m}$ be the set of all (x_t, u_t) satisfying (3)-(4). We assume $\Omega \subset \mathbb{R}^{n+m}$ is compact.

Assumption 3: The dynamics and constraint maps f_t^i, g_t^i , and h_t are twice continuously differentiable.

For each $i \in [N]$, $t \in [T]$, and $k \in [n_i]$, let $f_{t,k}^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}$ denote the k -th scalar output of the dynamics map f_t^i . Given Assumption 3, we define, $\forall i \in [N]$, $t \in [T]$:

$$\mu_{t,k}^i := \frac{1}{2} \max_{\substack{(\tilde{x}_t, \tilde{u}_t) \in \Omega \\ \xi \in \mathbb{B}_{\infty}^{n_i+m_i}}} \xi^\top \nabla^2 f_{t,k}^i(\tilde{x}_t^i, \tilde{u}_t^i) \xi, \quad \forall k \in [n_i], \quad (14)$$

$$\chi_{t,k}^i := \frac{1}{2} \max_{\substack{(\tilde{x}_t, \tilde{u}_t) \in \Omega \\ \xi \in \mathbb{B}_{\infty}^{n_i+m_i}}} \xi^\top \nabla^2 g_{t,k}^i(\tilde{x}_t^i, \tilde{u}_t^i) \xi, \quad \forall k \in [n_g^i], \quad (15)$$

$$\psi_{t,k} := \frac{1}{2} \max_{\substack{(\tilde{x}_t, \tilde{u}_t) \in \Omega \\ \zeta \in \mathbb{B}_{\infty}^{n+m}}} \zeta^\top \nabla^2 h_{t,k}(\tilde{x}_t, \tilde{u}_t) \zeta, \quad \forall k \in [n_h]. \quad (16)$$

Assumption 4: Each noise term w_t^i lies within the ∞ -norm ball in \mathbb{R}^{n_i} , i.e., $w_t^i \in \mathbb{B}_{\infty}^{n_i}$. Moreover, the state-dependent disturbance multiplier functions $E_t^i : \mathbb{R}^{n_i} \times \mathbb{R}^{n_i}$ are Lipschitz.

Given Assumption 4, for each $i \in [N]$, $t \in [T]$, $k \in [n_i]$, let $L_{E,t,k}^i > 0$ denote the Lipschitz constant for the k -th row of $n_i \times n_i$ matrix E_t^i , denoted $E_{t,k}^i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{1 \times n_i}$ below. Concretely, for any $x_t^i, z_t^i \in \mathbb{R}^{n_i}$:

$$\|E_{t,k}^i(x_t^i) - E_{t,k}^i(z_t^i)\|_{\infty} \leq L_{E,t,k}^i \|x_t^i - z_t^i\|_{\infty}. \quad (17)$$

Remark 2: $\mu_{t,k}^i, \chi_{t,k}^i, \psi_{t,k}$, and $L_{E,t,k}^i$ can be estimated via extreme value theory [21] or interval arithmetic [22].

We now present Prop. 2, the proof of which is given in our ArXiv paper [23, App. A].

Proposition 2 (Robust Constraint Satisfaction):

Suppose nominal trajectories $\mathbf{z} = (\mathbf{z}^1, \dots, \mathbf{z}^N)$, $\mathbf{v} := (\mathbf{v}^1, \dots, \mathbf{v}^N)$ and system responses $\Phi = \{\Phi^i : i \in [N]\}$ satisfying (13) are given. For each $i \in [N]$, $t, \tau \in [T]$ such that $t \geq \tau$, we define $\Lambda_{t,\tau}^i(\mathbf{z}, \mathbf{v}) \in \mathbb{R}^{n_i \times 2n_i}$, $\Gamma_{t,\tau}(\mathbf{z}, \mathbf{v}, \Phi) \in \mathbb{R}^{n \times 2n}$, $\mu_t^i \in \mathbb{R}^{n_i \times n_i}$, and $L_{E,t}^i \in \mathbb{R}^{n_i \times n_i}$ as follows:

$$\Lambda_{t,\tau}^i(\mathbf{z}, \mathbf{v}) := [E_{t,\tau}^i(z_{t,\tau}^i) \quad \|e_{t,\tau}^i\|_{\infty}^2 \mu_t^i + \|e_{t,\tau}^i\|_{\infty} L_{E,t}^i] \quad (18)$$

$$\Gamma_{t,\tau}(\mathbf{z}, \mathbf{v}, \Phi) := \text{diag}\{\Phi_{t,\tau}^1 \Lambda_{t-\tau}^1(\mathbf{z}, \mathbf{v}), \dots, \Phi_{t,\tau}^N \Lambda_{t-\tau}^N(\mathbf{z}, \mathbf{v})\} \quad (19)$$

$$\mu_t^i := \text{diag}\{\mu_{t,1}^i, \dots, \mu_{t,n_i}^i\}, \quad (20)$$

$$L_{E,t}^i := \text{diag}\{L_{E,t,1}^i, \dots, L_{E,t,n_i}^i\}. \quad (21)$$

If there exist auxiliary *error upper bound* variables $\rho^i := (\rho_0^i, \dots, \rho_{T-1}^i) \in \mathbb{R}^T$ such that the following hold $\forall i \in [N]$:

$$g_{t,k}^i(z_t^i, v_t^i) + \sum_{\tau=0}^{t-1} \|\nabla g_{t,k}^i(z_t^i, v_t^i)^\top \Phi_{t-1,\tau}^i \Lambda_{t-1-\tau}(\mathbf{z})\|_1 + \chi_{t,k}^i (\rho_t^i)^2 \leq 0, \quad \forall t \in [T], k \in [n_g^i], \quad (22a)$$

$$h_{t,k}(z_t, v_t) + \sum_{\tau=0}^{t-1} \|\nabla h_{t,k}(z_t, v_t)^\top \Gamma_{t-1,\tau}(\mathbf{z}, \Phi)\|_1 + \psi_{t,k} \cdot \sum_{j=1}^N (\rho_t^j)^2 \leq 0, \quad \forall t \in [T], k \in [n_h], \quad (22b)$$

$$\sum_{\tau=0}^{t-1} \|\Phi_{t-1,\tau}^i [E_{\tau}^i(z_{\tau}^i) \quad (\rho_{\tau}^i)^2 \mu_{\tau}^i + \rho_{\tau}^i L_{E,\tau}^i]\|_{\infty} \leq \rho_t^i, \quad \forall t \in [T-1], \quad (22c)$$

$$\rho_0^i \geq 0. \quad (22d)$$

then, for any realization of the noise terms w_t^i , we have $g_{t,k}^i(x_t^i, u_t^i) \leq 0 \forall k \in [n_g^i]$ and $h_{t,k}(x_t, u_t) \leq 0 \forall k \in [n_h]$, for each $i \in [N]$, $t \in [T]$, across all trajectory rollouts $(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^{n(T+1)+mT}$.

B. Robustly Constrained Nash Equilibrium

We now utilize the SLS-based parameterization of the affine error feedback for the LTV system (Sec. IV-A) to define the *robustly constraint-satisfying Nash equilibrium* (RCNE) solution to our dynamic game (Def. 2). In words, the RCNE which characterize a set of steady-state agent trajectories and controls at which each agent acts optimally with respect to all other agents' actions, while ensuring constraint satisfaction even under worst-case noise realizations.

We first define the set of *robustly constraint-satisfying* agent nominal trajectories and system responses by:

$$\mathcal{S} := \{(\mathbf{z}, \mathbf{v}, \Phi, \rho) : (\mathbf{z}^i, \mathbf{v}^i, \Phi^i, \rho^i) \text{ satisfy (6), (13), (22), } \forall i \in [N]\}, \quad (23)$$

where \mathbf{z}, \mathbf{v} , and Φ are as defined in Sec. IV-A, and $\rho := \{\rho^i : i \in [N]\}$. Next, similar to [24], to control the uncertainty propagation, we introduce the following augmented cost \tilde{J}^i for each agent $i \in [N]$, which appends a convex, quadratic regularization term $H^i(\Phi^i)$ on Agent i 's system response Φ^i to Agent i 's cost:

$$\tilde{J}^i(\mathbf{z}, \mathbf{v}, \Phi) := J^i(\mathbf{z}, \mathbf{v}) + H^i(\Phi^i). \quad (24)$$

We now characterize the *best response map*² of each agent i , and the set of RCNE nominal trajectories, system responses, and error upper bounds. Below, $\mathbf{z}^{-i} \in \mathbb{R}^{(n-n_i)T}$ denotes the nominal trajectories of all agents not indexed i , and $\mathbf{v}^{-i}, \Phi^{-i}, \rho^{-i}$ are defined analogously.

Definition 2: For each $i \in [N]$, we define Agent i 's set of *best responses* given other agents' nominal trajectories, system responses, and error bounds $(\mathbf{z}^{-i}, \mathbf{v}^{-i}, \Phi^{-i}, \rho^{-i})$ by:

$$\begin{aligned} \text{BR}^i(\mathbf{z}^{-i}, \mathbf{v}^{-i}, \Phi^{-i}, \rho^{-i}, \tilde{J}^i) \\ := \arg \min_{\mathbf{z}^i, \mathbf{v}^i, \Phi^i, \rho^i} \tilde{J}^i(\mathbf{z}, \mathbf{v}, \Phi) \\ \text{s.t. } (\mathbf{z}, \mathbf{v}, \Phi, \rho) \in \mathcal{S}. \end{aligned} \quad (25)$$

We call a set of agents' nominal state-control trajectories, system responses, and error (upper) bounds $(\mathbf{z}^*, \mathbf{v}^*, \Phi^*, \rho^*)$ a *Robustly Constrained Nash Equilibrium* (RCNE) if, at

²The *system response* Φ^i , defined component-wise in Prop. 1, and the *best response map* BR, defined in (25) describe different mathematical objects.

$(\mathbf{z}^*, \mathbf{v}^*, \Phi^*, \rho^*)$, each agent best responds to all other agents while robustly satisfying all constraints, i.e., for each $i \in [N]$:

$$(\mathbf{z}^{i*}, \mathbf{v}^{i*}, \Phi^{i*}, \rho^{i*}) \in \text{BR}^i(\mathbf{z}^{-i*}, \mathbf{v}^{-i*}, \Phi^{-i*}, \rho^{-i*}; \tilde{J}^i). \quad (26)$$

Problem Statement: We aim to formulate a computationally tractable algorithm for computing all agents' nominal state-control trajectories, system responses, and error upper bounds at Nash equilibrium, as defined in (26).

V. ALGORITHM

While a robustly constrained Nash equilibrium can in theory be found by solving (26) for each $i \in [N]$, in practice (26) is computationally intractable to solve with general-purpose solvers, due to the high-dimensional and coupled (across agent indices $i \in [N]$) nature of (26). To sidestep these computational difficulties, we present Alg. 1, which refines the Iterative Best Response (IBR) algorithm [20, Prop. 1] in the potential games literature to incrementally search for a Nash equilibrium solution. Concretely, Alg. 1 proceeds in iterations. Within each iteration, each agent $i \in [N]$ takes its turn applying an incremental best response step, with step size $\alpha \in [0, 1]$, with respect to all other agents' nominal state-control trajectories and system responses (Alg. 1, Lines 3-6).

It is known that, if each best-response update is performed with maximum step sizes (i.e., $\alpha = 1$ in Alg. 1) and with zero error, the IBR algorithm converges to a Nash equilibrium of a dynamic potential game [20]. However, in our robust games formulation, each computation of the best response map (25) encodes a high-dimensional optimization problem that is difficult to solve directly and efficiently. Instead, to efficiently approximate the outcome of Alg. 1, Line 3, we employ the Fast SLS algorithm introduced in [24]. Crucially, the Fast SLS algorithm exploits the structure stage-wise composition of the overall cost J^i for each agent $i \in [N]$, given by (2), to separately solve for the Agent i 's nominal trajectory $(\mathbf{z}^i, \mathbf{v}^i)$, controller \mathcal{K}^i (through the system response Φ^i), and error upper bound ρ^i . The Fast SLS algorithm alternates between updating K^i via a Riccati recursion, optimizing the nominal trajectory (z^i, v^i) as per (6), and performing an additional update of the error upper bound ρ^i . For details on the original formulation of the fast SLS algorithm, see [24, Sec. 3]. Whereas fast SLS [24, Sec. 3] solved QPs to recover solutions to LQR-style problems, our algorithm solves NLPs to recover solutions to more general control problems.

Remark 3: To motivate our use of the IBR algorithm, we note that, when applied to N -agent, T -horizon dynamic games in which the average agent state and control dimensions are \bar{n} and \bar{m} , respectively, our Fast SLS and IBR-based approach yields a per-iteration complexity $\mathcal{O}(T^2 N (\bar{n} + \bar{m})^3)$. (Here, "per-iteration complexity" is associated with the time needed to execute Alg. 1, Lines 3-6). In contrast, a centralized implementation of the form [2, Alg. 1] would yield a per-iteration complexity of $\mathcal{O}(T^2 N^3 (\bar{n} + \bar{m})^3)$ to update each agent's trajectory and control designs, even when accelerated via Fast SLS.

VI. EXPERIMENTS

We evaluated our algorithm by generating interactive, robustly constraint-satisfying motion plans across a broad

Algorithm 1: Iterative Best Response (IBR) for Computing RCNE

Input: Maximum iteration count K , Error Tolerance ϵ , Step size $\alpha \in [0, 1]$, Augmented Costs $\{\tilde{J}^i(\cdot) : i \in [N]\}$, Initial nominal trajectories and system responses $\{(\mathbf{z}^{i,(0)}, \mathbf{v}^{i,(0)}, \Phi^{i,(0)}, \rho^{i,(0)}) : i \in [N]\}$

- 1 **for** $k = 1, \dots, K$ **do**
- 2 **for** $i = 1, \dots, N$ **do**
- 3 $(\hat{\mathbf{z}}^{i,(k)}, \hat{\mathbf{v}}^{i,(k)}, \hat{\Phi}^{i,(k)}, \hat{\rho}^{i,(k)}) \in$
 $\text{BR}^i(\mathbf{z}^{-i,(k)}, \mathbf{v}^{-i,(k)}, \Phi^{-i,(k)}, \rho^{-i,(k)}; \tilde{J}^i)$
- 4 $\mathbf{v}^{i,(k)} \leftarrow \alpha \hat{\mathbf{v}}^{i,(k)} + (1 - \alpha) \mathbf{v}^{i,(k)}$
- 5 $\mathbf{z}^{i,(k)} \leftarrow \text{Unroll (6) using } \mathbf{v}^{i,(k)}$
- 6 $\Phi^{i,(k)}, \rho^{i,(k)} \leftarrow \text{Implement Riccati updates in}$
 $\text{Fast SLS using } \mathbf{z}^{i,(k)}, \mathbf{v}^{i,(k)}$.
- 7 $\delta^{(k)} \leftarrow \max_{i \in [N]} \max_{t, \tau \geq T} \max \{ \|\mathbf{z}^{i,(k)} -$
 $\mathbf{z}^{i,(k-1)}\|_2, \|\mathbf{v}^{i,(k)} - \mathbf{v}^{i,(k-1)}\|_2, \|\Phi_{t,\tau}^{i,(k)} -$
 $\Phi_{t,\tau}^{i,(k-1)}\|_2, \|\rho^{i,(k)} - \rho^{i,(k-1)}\|_2 \}$.
- 8 **if** $\delta^{(k)} \leq \epsilon$ **then**
- 9 $(\mathbf{z}^*, \mathbf{v}^*, \Phi^*, \rho^*) \leftarrow (\mathbf{z}^{(k)}, \mathbf{v}^{(k)}, \Phi^{(k)}, \rho^{(k)})$.
- 10 **Break**
- 11 **else**
- 12 $(\mathbf{z}^{(k+1)}, \mathbf{v}^{(k+1)}, \Phi^{(k+1)}, \rho^{(k+1)}) \leftarrow$
 $(\mathbf{z}^{(k)}, \mathbf{v}^{(k)}, \Phi^{(k)}, \rho^{(k)})$.

Output: $\mathbf{z}^*, \mathbf{v}^*, \Phi^*, \rho^*$

range of multi-agent interaction scenarios in simulation and on hardware. First, we present experiment settings used in our reported experiments, such as agent costs and constraint types (Sec. VI-A). We then evaluate our method on a "narrow corridor" scenario, wherein trajectories generated by our method remain robustly safe in confined spaces despite dynamics noise (Sec. VI-B). Our approach efficiently generates robustly constraint-satisfying motion plans in scenarios with up to 24 4D-unicycle agents (Sec. VI-C), with state-dependent dynamics noise (Sec. VI-D), or heterogeneous robot teams of quadcopters and 4D-unicycle ground vehicles (Sec. VI-E, VI-F). In contrast, while operating under baseline open-loop game-theoretic motion plans [9], the same multi-robot systems often violated collision-avoidance constraints when deployed in narrow spaces or in the presence of state-dependent dynamics noise (Sec. VI-B and VI-D). Our codebase can be found at: <https://github.com/nexuszhan/Robust-Dynamic-Game-SLS>.

A. Experiment Setup

Our experiments in Sec. VI-B-VI-F use the following types of agent dynamics, constraints, and costs. Below, given the state x_t^i of an agent i at time t , we denote by p_t^i , $p_{x,t}^i$, $p_{y,t}^i$, and $p_{z,t}^i$ the component(s) of x_t^i corresponding to the position vector, x -position, y -position, and z -position respectively.

a) *Dynamics models:* We apply our methods to plan trajectories for ground robots with 4D unicycle and airborne robots with 12D quadcopter dynamics in simulation (Sec. VI-B-VI-E), as well as ground robots with 3D Dubins car

dynamics and airborne robots with 3D single integrator dynamics on hardware (Sec. VI-F). Across experiments, unless otherwise stated, we discretize the above continuous-time dynamics models at intervals of $\Delta t = 0.1$ and use a time horizon of $T = 80$. All noise terms w_t^i are drawn from the unit 2-norm ball $\mathbb{B}_2^{n_i}$. We set $E_t^i(x_t^i) = 0.002 I_4$ unless otherwise stated (e.g., in Sec. VI-D, we define $E_t^i(x_t^i)$ as state-varying.)

b) Constraint types: We consider *agent-specific* constraints that encode component-wise bounds on position, orientation, linear velocity, angular velocity, and linear acceleration. We also implement *shared* constraints $h_{t,k}$ between pairs of agents $i, j \in [N]$ (say, with radii of r^i and r^j), that encode (i) collision-avoidance, by setting $h_{t,k}(x_t) := -\|p_t^i - p_t^j\|_2 + r^i + r^j \leq 0$; (ii) proximity constraints, by setting $h_{t,k}(x_t) := \|p_t^i - p_t^j\|_2 - \hat{r}^{ij} \leq 0$, for some prescribed proximity distance $r^{ij} > r^i + r^j$; and (iii) line-of-sight constraints, of form described in [25, App. D.3]. Each line-of-sight constraint enforced on a pair of robots, say, i and j , requires the angle between (i) velocity vector of the first agent, and (ii) the relative position of the second agent to the first, to always be within a prescribed bound $[-\theta^{ij}, \theta^{ij}]$.

c) Agent Costs: Across our experiments, the cost J^i of each agent i will be a sum of some of the cost components $C_1^i, C_2^i, C_3^i, C_4^i : \mathbb{R}^{(n_i+m_i)T} \rightarrow \mathbb{R}$ defined below, which encode an LQR cost, a smoothness and goal-reaching cost, a collision avoidance cost, and a proximity cost, respectively:³

$$C_1^i(\mathbf{x}, \mathbf{u}) = (x_T^i - \hat{x}_T^i)^\top Q_f^i (x_T^i - \hat{x}_T^i) + \sum_{t=0}^{T-1} [(x_t^i - \hat{x}_t^i)^\top Q^i (x_t^i - \hat{x}_t^i) + u_t^{i\top} R^i u_t^i], \quad (27)$$

$$C_2^i(\mathbf{x}, \mathbf{u}) = (x_T^i - \hat{x}_T^i)^\top Q_f^i (x_T^i - \hat{x}_T^i) + \sum_{t=1}^T (x_t^i - x_{t-1}^i)^\top Q^i (x_t^i - x_{t-1}^i), \quad (28)$$

$$C_3^{ij}(\mathbf{x}, \mathbf{u}) = -0.001 \cdot \sum_{t=0}^T \|p_t^i - p_t^j\|_2^2, \quad (29)$$

$$C_4^{ij}(\mathbf{x}, \mathbf{u}) = 0.001 \cdot \sum_{t=0}^T \|p_t^i - p_t^j\|_2^2 \quad (30)$$

Above, for each agent i , $\hat{x}_T^i \in \mathbb{R}^{n_i}$ denote specified goal states, while $Q_f^i, Q^i \in \mathbb{R}^{n_i \times n_i}$ (resp., $R^i \in \mathbb{R}^{m_i \times m_i}$) are symmetric positive semi-definite (resp., definite) weight matrices. The above parameters in general vary from one experiment to another, and will be specified as appropriate.

Unless specified elsewhere, we set $J^i(\mathbf{x}, \mathbf{u}) = C_2^i(\mathbf{x}, \mathbf{u}) + \sum_{j \neq i} C_3^{ij}(\mathbf{x}, \mathbf{u})$ when evaluating our method, which renders the resulting game a dynamic potential game. We generate baseline results using ALGAMES with the same cost J^i .

d) Algorithm Implementation: For the parameters used in Alg. 1, we fixed a maximum iteration count of $K = 5$ and an error tolerance of $\epsilon = 0.001$. Although we varied the step size α across experiments (see Secs. VI-B-VI-F below) for the first $Q - 1$ iterations, we fixed $\alpha = 1$ for the last iteration to enforce robust constraint satisfaction. Our experimental results indicate that, even with a limited number of iterations, Alg. 1 generates solutions that encode reasonable multi-agent interactions while always satisfying constraints robustly. We run all experiments on an Intel Core i9-13950HX CPU.

³In many of our experiments, we encode collision avoidance using both a *hard* constraint and a *soft* penalty.

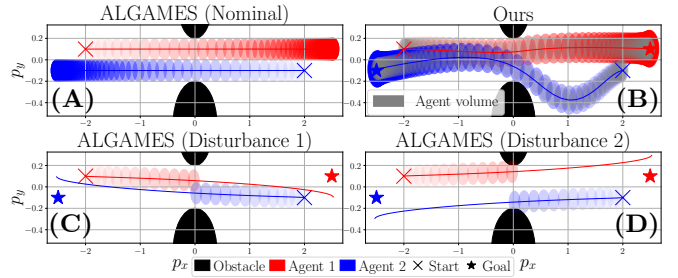


Fig. 2: **Narrow Corridor Experiment:** In a confined space, two 4D unicycle agents attempt to bypass each other safely despite dynamics noise. Across 500 rollouts, our method always generates safe trajectories (B), while the ALGAMES baseline [9] violated constraints 91.8% of the time.

e) Hessian bounds and Lipschitz constants: To reduce the conservativeness of the robust controller generated by Alg. 1, we adopt a simplified implementation by setting the Hessian bounds and Lipschitz constants $\mu_{t,k}^i, \chi_{t,k}^i, \psi_{t,k}^i$, and $L_{E,t,k}^i$ to zero in (14)-(17). Alternatively, data-driven estimates of these constants can be obtained to provide less conservative, high-probability guarantees by following [26]. Although setting these constants to zero implies that we are not guaranteed to over-approximate the error e_t^i between the nominal and realized trajectories, across our experiments, our algorithm still *empirically* satisfy all constraints robustly.

f) Figure Conventions: In each figure depicting trajectory rollouts from by our method, circles or balls indicate the Minkowski sum of the volume of each agent, as given by its radius r^i , and regions of realizable trajectories, as computed from the error upper bound values ρ^i . Figures of trajectories generated by the ALGAMES baseline [9] depict colored circles that indicate merely the volume of each agent.

B. Narrow Corridor Experiment

First, we generate trajectories in a *narrow corridor* experiment using our Alg. 1 and the baseline ALGAMES approach [9] (Fig. 2). In this experiment, 2 ground robots, each with radius 0.1m, attempt to bypass each other in a confined space over $T = 60$ time steps while robustly satisfying the set of prescribed constraints described below, despite dynamics noise. Each agent must satisfy collision-avoidance constraints with respect to all other agents, as well as two obstacles, one at $(0, 0.5)$ with radius 0.3m and one at $(0, -0.6)$ with radius 0.4m. For our Alg. 1 implementation, we set $\alpha = 0.3$, and use the cost $J^i(\mathbf{x}, \mathbf{u}) = C_1^i(\mathbf{x}, \mathbf{u}) + \sum_{j \neq i} C_3^{ij}(\mathbf{x}, \mathbf{u})$, with $Q = 2I_4$, $R = I_2$, and $Q_f = \text{diag}\{5, 5, 0, 5\}$. To implement the ALGAMES baseline, we set $Q = I_4$, $R = I_2$, and $Q_f = I_4$.

Across 500 rollouts, our method consistently generates interactive trajectories which enable the two agents to safely bypass each other while satisfying all of the constraints above (Fig. 2B), despite dynamics noise. Our Alg. 1 runtime was 32.2s. In contrast, out of 500 rollouts with dynamics noise, the baseline ALGAMES algorithm plans trajectories which resulted in collisions 91.8% of the time (Fig. 2A, C, D).

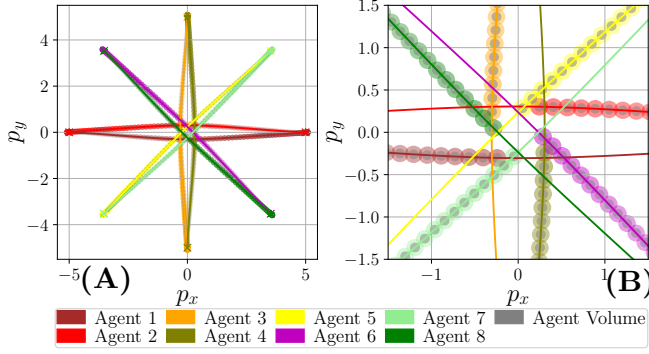


Fig. 3: **Scaling Example:** (A) Our method synthesizes robustly collision-free, interactive trajectories for 8 4D-unicycle agents. (B) presents a snapshot of (A) at $t = 40$ when the agents are in close proximity to each other.

C. Scaling Experiment

Next, we demonstrate that our method efficiently generates robustly constraint-satisfying trajectories even for multi-agent games with large numbers of agents. Concretely, for $N = 4, 8, 16$ and 24 , we computed robust motion plans for N 4D-unicycle agents, each with radius 0.05m , who aim to navigate while robustly satisfying pre-specified constraints despite dynamics noise (See Fig. 3 for the $N = 8$ setting). Each agent must also satisfy collision-avoidance constraints with respect to all other agents. We set $\alpha = 0.1$ for the $N = 4, 8$ settings and $\alpha = 0.5$ for the $N = 16, 24$ settings. We also set $Q = 2I_4$, $R = O$, and $Q_f = \text{diag}\{10, 10, 0, 10\}$. Our Alg. 1 runtimes for the $N = 4, 8, 16, 24$ settings are respectively given by 68.3s , 153.4s , 462.9s and 891.1s , with standard deviations 0.7s , 0.3s , 4.2s and 5.9s . We generated 10 trajectory rollouts per agent for each of the $N = 4, 8, 16, 24$ settings, all of which robustly satisfied all constraints.

D. State-Dependent Noise Experiment

We show that Alg. 1 produces interactive trajectories that robustly satisfy constraints in multi-agent interactions with *state-dependent* noise, i.e., when $E_t^i(\cdot)$ is *not* constant, while ALGAMES generates constraint-violating motion plans. Specifically, we consider 4 ground robots who wish to navigate a shared intersection while robustly satisfying constraints, located at the origin of a given 2D coordinate system, despite state-dependent dynamics noise (Fig. 4). In particular, we design $E_t^i(x_t^i)$ as shown below, to promote large realizations of dynamics noise near the origin, where agents would be in close proximity near the middle of the time horizon:

$$E_t^i(x_t^i) = \frac{1}{1000\pi} e^{-25(p_{x,t}^2 + p_{y,t}^2)} I_4. \quad (31)$$

Each agent must also satisfy collision-avoidance constraints with respect to all other agents. When implementing our Alg. 1, we set $\alpha = 1$, and choose $Q = 2I_4$, $R = O$, and $Q_f = \text{diag}\{10, 10, 0, 10\}$. For our ALGAMES baseline, we choose $Q = \text{diag}\{3, 3, 0, 1\}$, $R = O_{4 \times 4}$, and $Q_f = \text{diag}\{3, 3, 0, 1\}$.

Across 100 trajectory rollouts, our method consistently produced constraint-satisfying plans despite state-dependent dynamics noise (Fig. 4), while enabling each agent to reach

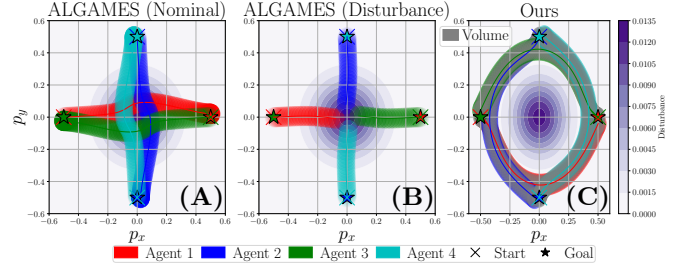


Fig. 4: **State-Dependent Noise Example:** Four 4D-unicycle agents aim to cross an intersection at the same time despite state-dependent noise with maximum noise realization at the intersection (see (31)). Across 100 rollouts, our method always generates safe trajectories (C), while the ALGAMES baseline [9] generated trajectories that always violated the constraints, in both the noise-free (A) and noisy (B) settings.

an average distance of 4.04cm away from their goal position. Our Alg. 1 runtime was 60s . In contrast, all 100 trajectory rollouts produced by the ALGAMES baseline violated the constraints, with average goal deviation of 9.82cm .

E. Heterogeneous Team Simulation

In Sec. VI-E-VI-F, we show Alg. 1 computes robustly constraint-satisfying interactive motion plans for heterogeneous robot teams in simulation and on hardware. We first simulate two teams of three robots: a 4D unicycle ground robot follows a second, which follows a 12D quadcopter. Each ground robot and quadcopter has radius $r^i = 0.06\text{m}$ and $r^i = 0.05\text{m}$, respectively. We set $E_t^i(x_t^i) = \text{diag}\{0.004 \times I_3, 1 \times 10^{-5} I_9\}$ for each quadcopter, where the first 3 coordinates represent the quadcopters' x, y, z coordinates, and $E_t^i(x_t^i) = 5 \times 10^{-4} I_4$ for each ground robot. Each pair of robots must satisfy collision-avoidance constraints. Each Dubin car must also satisfy, with respect to its leader, proximity constraints with distance $\rho^{ij} = 0.5\text{m}$ and line-of-sight constraints with maximum angle $\theta^{ij} = \pi/2$ rad. When running Alg. 1, we set $\alpha = 0.2$ and $J^i(\mathbf{x}, \mathbf{u}) = C_2^i(\mathbf{x}, \mathbf{u}) + \sum_{j \neq i} [C_3^{ij}(\mathbf{x}, \mathbf{u}) + C_4^{ij}(\mathbf{x}, \mathbf{u})]$, with $Q = \text{diag}\{2I_3, O_{3 \times 3}, 2I_6\}$, $R = O$, and $Q_f = \text{diag}\{10I_3, O_{3 \times 3}, 10I_6\}$ for each quadcopter and $Q = 2I_4$, $R = O$, and $Q_f = O$ for each ground robot. Using Alg. 1 with $\alpha = 0.6$, we computed 10 trajectory rollouts (avg. runtime: 323s), which all robustly satisfied constraints (Fig. 5).

F. Heterogeneous Team Hardware Experiment

We present a hardware analog of the heterogeneous team simulations in Sec. VI-E, over $T = 30$ time steps with $\Delta t = 0.4\text{s}$. Each ground robot and quadcopter has radius $r^i = 0.06$ and 0.1 , respectively. We set $E_t^i(x_t^i) = 0.04I_3$ for each quadcopter and $0.001I_4$ for each ground robot. Each pair of robots must satisfy collision-avoidance constraints. The Dubins car behind the quadcopter must also satisfy proximity constraints with $\rho^{ij} = 0.75\text{m}$, and line-of-sight constraints with $\theta^{ij} = \pi/2$ rad, with respect to the quadcopter. The other Dubins car must satisfy proximity constraints with $\rho^{ij} = 0.5\text{m}$, and line-of-sight constraints with $\theta^{ij} = \pi/3$ rad, both with respect to the first Dubins car. When implementing Alg. 1, we set

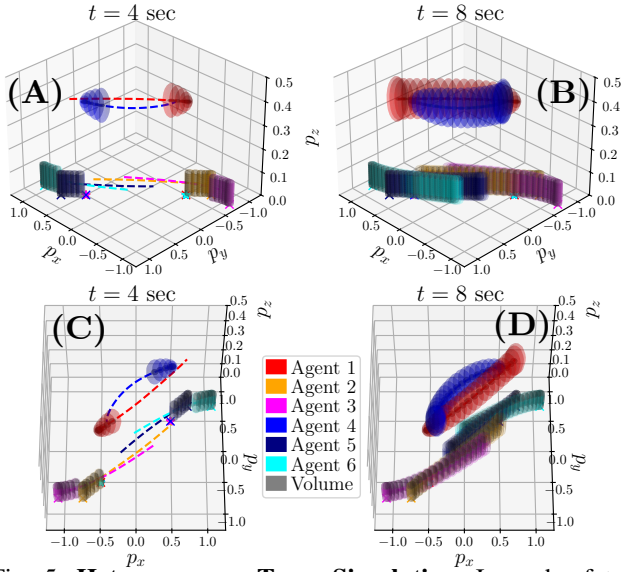


Fig. 5: **Heterogeneous Team Simulation:** In each of two teams, one ground robot follows a second, which in turn follows a quadcopter. Each team must satisfy proximity and line-of-sight constraints besides collision avoidance with other agents. Our Alg. 1 produced trajectories satisfying all pre-specified proximity, line-of-sight, and collision avoidance constraints despite dynamics noise. (A), (C) (resp., (B), (D)) show trajectories from two angles at $t = 4$ s (resp., $t = 8$ s).

$\alpha = 0.2$, and use the cost $J^i(\mathbf{x}, \mathbf{u}) = C_2^i(\mathbf{x}, \mathbf{u})$, with $Q = 2I_3$, $R = O$, and $Q_f = 10I_3$ for each quadcopter and $Q = 2I_3$, $R = O$, and $Q_f = O$ for each ground robot. Using Alg. 1 with $\alpha = 0.2$, we generated robustly constraint-satisfying trajectory rollouts with an Alg. 1 runtime of 38.3s (Fig. 1). Thus, our methods enable the safe, real-world deployment of heterogeneous robot teams despite uncertain dynamics.

VII. CONCLUSION

We present a system level synthesis (SLS)-based formulation for robust dynamic games with nonlinear dynamics, nonlinear constraints, and state-dependent disturbances. By coupling nominal planning with causal affine error feedback, we derive tractable nonlinear safety certificates for individual and shared constraints. We then introduce the robust constrained Nash equilibrium (RCNE) concept and propose an iterative best-response (IBR) method that leverages Fast SLS for scalable joint trajectory-controller optimization. In experiments, our method improved safety, proactively avoided high-disturbance regions, reduced terminal error versus baseline methods, and scaled to large, heterogeneous teams. Our future work will develop convergence guarantees for Alg. 1 and compare extensively against a range of single-agent and multi-agent robust motion planning baselines.

REFERENCES

- [1] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, “The Computation of Approximate Generalized Feedback Nash Equilibria,” *SIAM Journal on Optimization*, vol. 33, no. 1, 2023.
- [2] T. Kavuncu, A. Yaraneri, and N. Mehr, “Potential iLQR: A Potential-Minimizing Controller for Planning Multi-Agent Interactive Trajectories,” *Robotics: Science and Systems*, 07 2021.

- [3] M. Bhatt, Y. Jia, and N. Mehr, “Efficient Constrained Multi-Agent Trajectory Optimization Using Dynamic Potential Games,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2023.
- [4] Z. Zhou, W. Zhang, J. Ding, H. Huang, D. M. Stipanović, and C. J. Tomlin, “Cooperative Pursuit with Voronoi Partitions,” *Automatica*, vol. 72, 2016.
- [5] L. Peters, A. Bajcsy, C.-Y. Chiu, D. Fridovich-Keil, F. Laine, L. Ferranti, and J. Alonso-Mora, “Contingency Games for Multi-Agent Interaction,” *Robotics and Automation Letters*, vol. 9, no. 3, 2024.
- [6] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. SIAM, 1998.
- [7] R. Isaacs, “Differential Games, Parts 1-4,” *The Rand Corporation, Research Memorandums RM-1391, RM-1411, RM-1486*, vol. 55, 1954.
- [8] D. Fridovich-Keil, E. Ratner, L. Peters, A. Dragan, and C. J. Tomlin, “Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games,” *ICRA*, 2020.
- [9] S. LeClerc’h, M. Schwager, and Z. Manchester, “ALGAMES: A Fast Augmented Lagrangian Solver for Constrained Dynamic Games,” *Auton. Robots*, vol. 46, no. 1, 2022.
- [10] I. Mitchell, A. Bayen, and C. Tomlin, “A time-dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games,” *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [11] W. Schwarting, A. Pierson, S. Karaman, and D. Rus, “Stochastic Dynamic Games in Belief Space,” *IEEE Transactions on Robotics*, vol. 37, no. 6, 2021.
- [12] R. Spica, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwager, “A Real-Time Game Theoretic Planner for Autonomous Two-Player Drone Racing,” *IEEE Transactions on Robotics*, vol. 36, no. 5, 2020.
- [13] C.-Y. Chiu and D. Fridovich-Keil, “GTP-SLAM: Game-Theoretic Priors for Simultaneous Localization and Mapping in Multi-Agent Scenarios,” in *Conference on Decision and Control (CDC)*, 2022.
- [14] O. B. Neto, M. Mulas, and F. Corona, “SLS-BRD: A System-level Approach to Seeking Generalised Feedback Nash Equilibria,” *IEEE Transactions on Automatic Control*, 2025.
- [15] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, “System Level Synthesis,” *Annual Reviews in Control*, vol. 47, 2019.
- [16] C. A. Alonso, J. S. Li, J. Anderson, and N. Matni, “Distributed and Localized Model-Predictive Control Part I: Synthesis and Implementation,” *IEEE Transactions on Control of Network Systems*, 2023.
- [17] S. Chen, N.-Y. Li, V. M. Preciado, and N. Matni, “Robust Model Predictive Control of Time-Delay Systems through System Level Synthesis,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022.
- [18] A. P. Leeman, J. Köhler, A. Zanelli, S. Bannani, and M. N. Zeilinger, “Robust Nonlinear Optimal Control via System Level Synthesis,” *IEEE Transactions on Automatic Control*, vol. 70, no. 7, 2025.
- [19] A. P. Leeman, J. Sieber, S. Bannani, and M. N. Zeilinger, “Robust Optimal Control for Nonlinear Systems with Parametric Uncertainties via System Level Synthesis,” in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023.
- [20] A. Zanardi, E. Mion, M. Bruschetta, S. Bolognani, A. Censi, and E. Frazzoli, “Urban Driving Games With Lexicographic Preferences and Socially Efficient Nash Equilibria,” *IEEE RA-L*, 2021.
- [21] C. Knuth, G. Chou, J. Reese, and J. Moore, “Statistical safety and robustness guarantees for feedback motion planning of unknown underactuated stochastic systems,” in *International Conference on Robotics and Automation (ICRA)*, 2023.
- [22] D. Limon, J. Bravo, T. Alamo, and E. Camacho, “Robust MPC of Constrained Nonlinear Systems Based on Interval Arithmetic,” *IEEE Proceedings on Control Theory and Applications*, vol. 152, no. 3, 2005.
- [23] S. Zhan, C.-Y. Chiu, A. P. Leeman, and G. Chou, “Robustly Constrained Dynamic Games for Uncertain Nonlinear Dynamics,” *ArXiv:2509.16826*, 2021.
- [24] A. P. Leeman, J. Köhler, F. Messerer, A. Lahr, M. Diehl, and M. N. Zeilinger, “Fast System Level Synthesis: Robust Model Predictive Control using Riccati Recursions,” *IFAC Conference on NMPC*, 2024.
- [25] Z. Zhang, C.-Y. Chiu, and G. Chou, “Constraint Learning in Multi-Agent Dynamic Games from Demonstrations of Local Nash Interactions,” *ArXiv:2508.19945*, 2025.
- [26] A. Srinivasan, A. Leeman, and G. Chou, “Safety beyond the training data: Robust out-of-distribution mpc via conformalized system level synthesis,” *arXiv preprint arXiv:2602.12047*, 2026.