MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Decentralized, Safe, Multi-agent Motion Planning for Drones Under Uncertainty via Filtered Reinforcement Learning

Vinod, Abraham P.; Safaoui, Sleiman; Summers, Tyler; Yoshikawa, Nobuyuki; Di Cairano, Stefano

TR2024-136 October 04, 2024

Abstract

We propose a decentralized, multi-agent motion planner that guarantees probabilistic safety of a team subject to stochastic uncertainty in agent model and environment. Our scalable approach generates safe motion plans in real-time using off-the-shelf, single-agent reinforcement learning rendered safe using distributionally-robust, convex optimization and buffered Voronoi cells. We guarantee recursive feasibility of the mean trajectories and mitigate the conservativeness using a temporal discounting of safety. We show in simulation that our approach generates safe and high performant trajectories as compared to existing approaches, and further validate these observations in physical experiments using drones.

IEEE Transactions on Control Systems Technology 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Decentralized, Safe, Multi-agent Motion Planning for Drones Under Uncertainty via Filtered Reinforcement Learning

Abraham P. Vinod*, *Member, IEEE,* Sleiman Safaoui, *Member, IEEE,* Tyler H. Summers, *Member, IEEE,* Nobuyuki Yoshikawa, *IEEE Member,* Stefano Di Cairano, *Senior Member, IEEE*

***Abstract*—We propose a decentralized, multi-agent motion planner that guarantees probabilistic safety of a team subject to stochastic uncertainty in agent model and environment. Our scalable approach generates safe motion plans in real-time using off-the-shelf, single-agent reinforcement learning rendered safe using distributionally-robust, convex optimization and buffered Voronoi cells. We guarantee recursive feasibility of the mean trajectories and mitigate the conservativeness using a temporal discounting of safety. We show in simulation that our approach generates safe and high performant trajectories as compared to existing approaches, and further validate these observations in physical experiments using drones.**

***Index Terms*—Safe learning-based control, constrained control under uncertainty, decentralized model predictive control, reinforcement learning, multi-agent systems, collision avoidance.**

I. Introduction

MULTI-AGENT motion planning in a cluttered environment is a fundamental problem for robot autonomy in transportation, inventory management, and monitoring [1]–[3]. These planning problems often require steering each agent to accomplish pre-defined tasks like reaching a designated area, while avoiding collisions with other agents and elements of the environment. Existing strategies for multi-agent motion planning are often based on optimization [3], [4], sampling [5], [6], or geometric methods [7], [8]. Recently, approaches based on reinforcement learning (RL) have been proposed to tackle such planning problems, motivated by the ability of RL to handle complex tasks and leverage data [1], [2], [9]–[12]. However, pure RL-based motion plans cannot *guarantee* safety (collision avoidance between agents, and with the obstacles in the environment), since RL typically incorporates safety constraints as soft constraints and computes approximated control policies [1], [11]. Also, multi-agent RL-based approaches often require long training and large amounts of data, since they must contend with the ambiguous assignment of rewards gained by the collective team due to an individual's actions [1].

This paper builds on our recent approach of *filtered reinforcement learning* to achieve safe multi-agent motion planning [13], [14]. In [13], we proposed a two-stage approach that combines off-the-shelf, single-agent RL policies with an optimization-based safety filter. The RL policies are trained (offline) for a single agent for each reach-avoid task to enable computation of reference motion plans online. The safety filter is used (online) to compute corrections to guarantee safety. In [14], we extended the approach in [13] to Gaussian-perturbed agent dynamics as well as sensing uncertainties and provided recursive feasibility guarantees. However, [13], [14] relied on a *centralized* safety filter that prescribed corrections to all the agents simultaneously, which may cause communication and computation challenges for large teams.

In this paper, we focus on two key extensions of our prior works. First, we propose a *decentralized* safety filter that allows the computation of the safe corrections locally for each agent, without relying on centralized computations. The decentralized safety filters in *optimal reciprocal collision avoidance* [7], [8] are restricted to uncertainty-free settings and typically assume constant input corrections. Instead, we leverage a convex model predictive control (MPC) framework to accommodate a broader range of dynamics, constraints on the state and input, and uncertainty. We mitigate the conservativeness in the safety filter due to lacking access to other agents' future plans with a slack-based temporal discounting of safety.

Second, we relax the Gaussian assumption on the uncertainties in [13], [14] and require instead membership to *moment-based ambiguity sets*, i.e., the only assumption is a pre-specified mean and covariance. Thus, the uncertainty class is broader, e.g., it includes multimodal uncertainties and heavy-tailed distributions. Also, since the first two moments may often be computed to sufficient accuracy from data, these sets are often readily available in the practical applications.

Our proposed safety filter is inspired by recent works on buffered Voronoi cells [15], [16], where decentralized collision avoidance under Gaussian uncertainty is achieved by requiring agents to stay inside appropriately shrunken Voronoi cells. We derive similar chance constraint-based tightening for uncertainties characterized by moment-based ambiguity sets. Also, we guarantee recursive feasibility of the mean trajectories via terminal constraints.

Summarizing, the main contribution of this work is a decentralized multi-agent motion planner, where the agent dynamics and the real-time measurements of the agent states and the obstacles in the environment are subject to stochastic uncertainty characterized by moment-based ambiguity sets. We propose a filtered reinforcement learning-based planner that locally corrects off-the-shelf RL-based motion plans for each agent. We propose a novel temporal discounting of safety to overcome the conservativeness arising from buffered Voronoi cell-based decentralization, and include terminal constraints

* Corresponding author.

A. P. Vinod and S. Di Cairano are with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA (email: abraham.p.vinod@ieee.org, dicairano@ieee.org). S. Safaoui and T. Summers are with the Control, Optimization and Networks Lab (CONLab) at the University of Texas at Dallas, Richardson, TX 75080, USA (email: sleiman.safaoui@utdallas.edu, tyler.summers@utdallas.edu). N. Yoshikawa is with Mitsubishi Electric Corporation, Japan (email: yoshikawa.nobuyuki@ak.mitsubishielectric.co.jp).

for recursive feasibility. We validate our approach in drone-based experiments and analyze its performance in simulations.

Notation: $[a : b]$ is the inclusive set of natural numbers between $a, b \in \mathbb{N}$. 0_d, I_d are the zero-vector in $\mathbb{R}^d$, and the d-dimensional identity matrix. For a vector v, $\|v\|$ is the Euclidean norm, $\mathrm{diag}(v)$ is the matrix with v in the diagonal, $v \cdot y$ is the dot product with vector y. For a convex compact set $\mathcal{C}$, the support function is $S_{\mathcal{C}}(\nu) \triangleq \sup_{v \in \mathcal{C}} \nu \cdot v$ for any $\nu \in \mathbb{R}^d$. $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space where Ω is the sample space, $\mathcal{F} = \sigma(\Omega)$ is a σ-algebra of Ω, $\mathbb{P}$ is a probability measure on $\mathcal{F}$, and $\mathbb{E}$ is the expectation operator with respect to $\mathbb{P}$.

We denote random vectors in bold $\boldsymbol{w} : \Omega \to \mathbb{R}^n$ and for an associated probability measure $\mathbb{P}$, we denote mean and covariance by $\overline{w}$ and Σ_w. An *ambiguity set* $\mathscr{P}(\mu, \Sigma)$ is the set of distributions with mean $\mu \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{R}^{n \times n}$:

$$\mathscr{P}(\mu, \Sigma) \triangleq \{\mathbb{P} \mid \overline{w} = \mu,\ \Sigma_w = \Sigma\}. \tag{1}$$

Given ambiguity sets $\mathscr{P}_1, \mathscr{P}_2$ associated with $(\Omega_1, \mathcal{F}_1), (\Omega_2, \mathcal{F}_2)$, the *ambiguity set couple* over the joint measurable space is $(\Omega_1 \times \Omega_2, \sigma(\mathcal{F}_1 \times \mathcal{F}_2))$ as,

$$\mathscr{P}_1 \times \mathscr{P}_2 = \left\{ \mathbb{P} \;\middle|\; \begin{array}{c} \mathrm{Marginal}_{\Omega_1}(\mathbb{P}) = \mathbb{P}_1 \in \mathscr{P}_1, \\ \mathrm{Marginal}_{\Omega_2}(\mathbb{P}) = \mathbb{P}_2 \in \mathscr{P}_2, \\ \mathbb{P}_1 \text{ and } \mathbb{P}_2 \text{ are independent} \end{array} \right\}, \tag{2}$$

where $\mathrm{Marginal}_{\Omega}(\mathbb{P})$ is the marginal probability over Ω.

For a random vector $\boldsymbol{x}(t)$, $\boldsymbol{x}(k|t)$ is the predicted value at $k \geq t$ based on information at time t, and $\boldsymbol{x}(t|t) \triangleq \boldsymbol{x}(t)$, and similarly for distributions of $\boldsymbol{x}(t)$.

II. Multi-agent motion planning problem

A. Model of the agents and the environment

Consider $N_A \in \mathbb{N}$ homogeneous agents with discrete-time, stochastic linear dynamics,

$$\boldsymbol{x}_i(k+1|t) = A\boldsymbol{x}_i(k|t) + Bu_i(k|t) + \boldsymbol{w}_i(k), \tag{3a}$$
$$\boldsymbol{y}_i(t) = \boldsymbol{x}_i(t) + \boldsymbol{\eta}_i(t), \tag{3b}$$
$$\boldsymbol{w}_i \sim \mathbb{P}_w \in \mathscr{P}_{w_i} = \mathscr{P}(0_n, \Sigma_{w_i}), \tag{3c}$$
$$\boldsymbol{\eta}_i \sim \mathbb{P}_\eta \in \mathscr{P}_{\eta_i} = \mathscr{P}(0_n, \Sigma_{\eta_i}). \tag{3d}$$

Equation (3a) describes the stochastic dynamics of agent $i \in [1 : N]$ based on the information up to time step $t \in \mathbb{N}$. At any $k \geq t$, $\boldsymbol{x}_i(k|t) \in \mathbb{R}^n$ is the state, $u_i(k|t) \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, where $\mathcal{U}$ is a convex and compact set, $\boldsymbol{w}_i(k) \in \mathbb{R}^n$ is an independent and identically distributed process noise with some *unknown* distribution $\mathbb{P}_{\boldsymbol{w}} \in \mathscr{P}_{w_i}$ (3c). Equation (3b) models the noisy, full-state measurements at time t, where the true state $\boldsymbol{x}_i(t)$ is corrupted by an independent and identically distributed measurement noise $\boldsymbol{\eta}_i(t) \in \mathbb{R}^n$ that follows an *unknown* distribution $\mathbb{P}_{\boldsymbol{\eta}} \in \mathscr{P}_{\eta_i}$ (3d). Thus, $\boldsymbol{w}$ accounts for actuation and/or modeling errors (3a), and $\boldsymbol{\eta}$ accounts for estimation and/or sensor errors. The state $\boldsymbol{x}_i(k|t)$ includes position $\boldsymbol{p}_i(k|t) \in \mathbb{R}^d$ and velocity $\boldsymbol{v}_i(k|t) \in \mathbb{R}^d$,

$$\boldsymbol{p}_i(k|t) = C_{\text{pos}}\boldsymbol{x}_i(k|t),\ \boldsymbol{v}_i(k|t) = C_{\text{vel}}\boldsymbol{x}_i(k|t), \tag{4}$$

for $C_{\text{pos}}, C_{\text{vel}} \in \mathbb{R}^{d \times n}$ with $d \in \{2, 3\}$, $d \leq m$.

For simplicity of the exposition, we assume that $\boldsymbol{w}$ and $\boldsymbol{\eta}$ have zero mean. Then, from the mean $\overline{y}_i(t)$ and the covariance matrix $\Sigma_{y_i}(t)$ of the measurement $\boldsymbol{y}_i(t)$ at time t for agent i, we obtain for any time $k \geq t$

$$\overline{x}_i(t) = \mathbb{E}[\boldsymbol{y}_i(t) - \boldsymbol{\eta}_i(t)] = \overline{y}_i(t), \tag{5a}$$
$$\Sigma_{x_i}(t) = \Sigma_{y_i}(t) + \Sigma_{\eta_i}(t), \tag{5b}$$
$$\overline{x}_i(k+1|t) = A\overline{x}_i(k|t) + Bu_i(k|t), \tag{5c}$$
$$\Sigma_{x_i}(k+1|t) = A\Sigma_{x_i}(k|t)A^T + \Sigma_{w_i}, \tag{5d}$$
$$\overline{p}_i(k|t) = C_{\text{pos}}\overline{x}_i(k|t), \tag{5e}$$
$$\Sigma_{p_i}(k|t) = C_{\text{pos}}\Sigma_{x_i}(k|t)C_{\text{pos}}^T. \tag{5f}$$

Equation (5) characterizes the moment-based ambiguity sets $\mathscr{P}_{x_i(k|t)}(\overline{x}_i(k|t), \Sigma_{x_i}(k|t))$ and $\mathscr{P}_{p_i(k|t)}(\overline{p}_i(k|t), \Sigma_{p_i}(k|t))$ corresponding to the state and position respectively, at any time $k \geq t$. In (5), we assume knowledge of the first two moments of $\boldsymbol{y}$, the ambiguity sets $\mathscr{P}_{\eta_i}, \mathscr{P}_{w_i}$ (i.e., Σ_w and Σ_η), and the matrices A, B, $C_{\text{pos}}, C_{\text{vel}}$.

We assume that every agent has an identical rigid-body $\mathcal{A} \subset \mathbb{R}^d$ that is a convex and compact set. Furthermore, we consider translation-only motion for the agents and ignore rotations, which is a typical assumption in motion planning for holonomic robots [17].

Remark 1. *We assumed homogeneity of all agents to simplify the exposition. The proposed approach is straightforward to extend to non-homogenous agents.*

We assume that the environment is a compact polytope $\mathcal{K} \subset \mathbb{R}^d$ with N_O obstacles, and that the obstacles are known, convex, and compact rigid bodies $\mathcal{O}_j \subset \mathbb{R}^d$, $j \in [1 : N_O]$. Thus, obstacle $j \in [1 : N_O]$ is described by the set $\{\boldsymbol{c}_j\} \oplus \mathcal{O}_j$ with a random position vector $\boldsymbol{c}_j \in \mathbb{R}^d$ to account for limitations in obstacle sensing and (possibly time-varying) perturbations in its location. Again, we only assume knowledge of mean and covariance of $\boldsymbol{c_j}$, i.e., $\boldsymbol{c}_j \sim \mathbb{P}_{c_j} \in \mathscr{P}_{c_j}(\overline{c}_j, \Sigma_{c_j})$.

The agents must eventually reach deterministic target regions $q_\ell \oplus \mathcal{Q}$ with $q_\ell \in \mathbb{R}^d$, $\ell \in [1 : N_T]$ as the target position and $\mathcal{Q} \subset \mathbb{R}^d$ are acceptable deviations from it. Each agent i has a pre-specified target ℓ.

B. Reinforcement learning-based single-agent motion planner

For motion planning of a single agent in a deterministic setting with mean dynamics (5c), we can train a single-agent RL-based motion planner for reach-avoid tasks. Specifically, RL generates motion plans for agent $i \in [1 : N_A]$ to avoid collisions with obstacles centered at their nominal positions $\overline{c}_j$, $\forall j \in [1 : N_O]$, stay within the environment bounds $\mathcal{K}$, and eventually reach the corresponding target $q_i \oplus \mathcal{Q}$.

As in [14], we consider a feedforward-feedback controller $\pi : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^m$ with

$$u = \pi(x, r) = Kx + Fr, \tag{6}$$

such that (3) with (6) result in the asymptotically stable nominal dynamics,

$$\overline{x}(k+1|t) = (A + BK)\overline{x}(k|t) + BFr(k|t). \tag{7}$$

Next, we define an observation vector $o \in \mathscr{O} \subseteq \mathbb{R}^{n+(N_O+1)d}$ that augments the agent state vector with additional information regarding the displacement with respect to the target position q_ℓ as well as the obstacle positions $\bar{c}_j$. We train a neural policy $\nu : \mathscr{O} \to \mathscr{R}$ that maps an observation vector $o \in \mathscr{O}$ to a reference $r \in \mathscr{R}$. Such neural policies can be easily computed using off-the-shelf single-agent RL frameworks like `stable-baselines3` [18], see [14] for more details.

Finally, we can "rollout" the policy network ν to obtain a trajectory based on the RL motion planner for a planning horizon $T \in \mathbb{N}$. Consider an agent $i \in [1 : N_A]$ with mean measurement $\overline{y}_i(t)$ at time t. We compute the RL motion plan $\{x_i^{\mathrm{RL}}(k|t)\}_{k=t}^{t+T}$, where $x_i^{\mathrm{RL}}(t|t) = \overline{y}_i(t)$ by (5a), by alternating between finding the control $u_i^{\mathrm{RL}}(k|t)$ given the predicted RL state $x_i^{\mathrm{RL}}(k|t)$ and observation $o_i(k|t)$ by,

$$u_i^{\mathrm{RL}}(k|t) = \pi\left(x_i^{\mathrm{RL}}(k|t), q_i + \nu(o_i(k|t)\right), \tag{8}$$

and predicting the next RL state $x_i^{\mathrm{RL}}(k+1|t)$ using (7) and the corresponding observation vector $o_i(k+1|t)$.

The motion plan $\{x_i^{\mathrm{RL}}(k|t)\}_{k=t}^{t+T}$ may not result in collision-free trajectories because RL only penalizes collisions and is subject to training errors. Furthermore, to shorten and simplify the training, the generated RL motion plan ignores inter-agent collisions and the effect of the process and measurement noises. However, the single-agent RL motion plan is easy to generate online, accommodates a variety of motion planning tasks beyond reach-avoid [19], and does not suffer from the non-stationarity or scalability issues of multi-agent RL.

C. Problem Statement

Next, we formalize the required features of safety in the multi-agent motion planning problem by introducing the notion of *distributionally-robust, probabilistic collective safety*, inspired by existing literature [13], [14], [17].

Definition 1 (DISTRIBUTIONALLY ROBUST PROBABILISTIC COLLECTIVE SAFETY)**.** *The agents are* distributionally-robust, probabilistically collectively safe *(DRPC-safe) at time t if the following conditions are met:*

1) Static obstacle avoidance constraints*: The probability of collision of agent $i \in [1 : N_A]$ with obstacle $j \in [1 : N_O]$ under the worst-case distribution in $\mathscr{P}_{p_i} \times \mathscr{P}_{c_j}$ is less than a pre-specified risk bound $\alpha_{i,j,t} \in (0,1)$,*

$$\mathbb{P}((\boldsymbol{p}_i(t) \oplus \mathcal{A}) \cap (\boldsymbol{c}_j(t) \oplus \mathcal{O}_j) \neq \emptyset) \leq \alpha_{i,j,t}, \tag{9}$$

for every $\mathbb{P} \in \mathscr{P}_{p_i} \times \mathscr{P}_{c_j}$.

2) Inter-agent collision avoidance constraints*: The probability of collision between agents $i, i' \in [1 : N_A],\ i \neq i'$ under the worst-case distribution in $\mathscr{P}_{p_i} \times \mathscr{P}_{p_{i'}}$ is less than a pre-specified risk bound $\beta_{i,i',t} \in (0,1)$,*

$$\mathbb{P}((\boldsymbol{p}_i(t) \oplus \mathcal{A}) \cap (\boldsymbol{p}_{i'}(t) \oplus \mathcal{A}) \neq \emptyset) \leq \beta_{i,i',t}, \tag{10}$$

for every $\mathbb{P} \in \mathscr{P}_{p_i} \times \mathscr{P}_{p_{i'}}$.

3) Keep-in constraints*: The probability of agent $i \in [1 : N_A]$ exiting the keep-in set $\mathcal{K}$ under the worst-case distribution $\mathbb{P} \in \mathscr{P}_{p_i}$ is less than a pre-specified risk bound $\kappa_{i,t} \in (0,1)$,*

$$\mathbb{P}(\boldsymbol{p}_i(t) \oplus \mathcal{A} \not\subseteq \mathcal{K}) \leq \kappa_{i,t}, \tag{11}$$

for every $\mathbb{P} \in \mathscr{P}_{p_i}$.

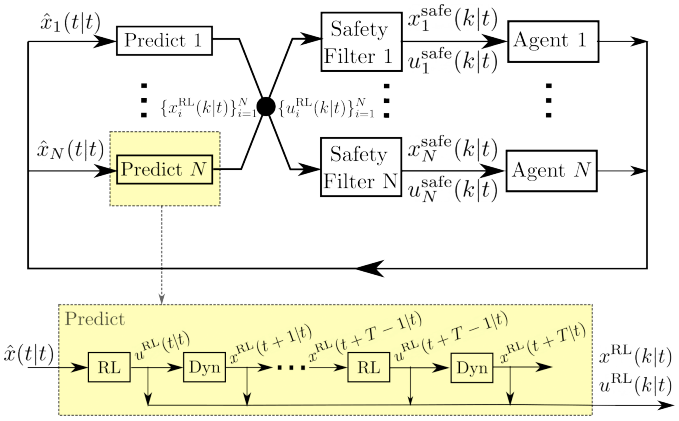

Fig. 1. Block diagram of the proposed motion planner that combines reinforcement learning (RL) with a safety filter using convex, distributionally-robust, stochastic model predictive control (MPC), and buffered Voronoi cells.

Observe that the distributionally-robust chance constraints associated with DRPC-safety are infinite-dimensional and non-convex. Consequently, we require characterization of convex sufficient conditions for DRPC-safety for tractability.

Definition 2 (DECENTRALIZED MOTION PLANNING)**.** *A multi-agent motion planner is* decentralized*, when each agent can determine only the control affecting its own motion.*

Problem 1 (DECENTRALIZED, SAFE MULTI-AGENT MOTION PLANNING)**.** *Given user-specified risk bounds $\alpha_{i,j,k},\ \beta_{i,i',k},\ \kappa_{i,k}$, for all $i, i' \in [1 : N_A],\ i \neq i',\ j \in [1 : N_O]$, design a RL-based decentralized multi-agent motion planner that navigates the agents to their respective targets according to* (3)*, while ensuring DRPC-safety at all times.*

Compared to our prior work [13], [14], Problem 1 requires the design of a *decentralized* motion planner that can accommodate *non-Gaussian uncertainties* constrained in some known moment-based ambiguity sets.

Remark 2. *We focus the multi-agent motion planning problem with drones since path planning for drones can be accomplished using linear dynamics as* (3)*, see [14], [20], [21]. However, the proposed approach may also be applied to similar problems with teams of different robots.*

III. DECENTRALIZED FILTERED REINFORCEMENT LEARNING

Fig. 1 describes the proposed approach for safe, RL-based multi-agent motion planning (see Section II-B). Similarly to [14], we use single-agent RL-based motion planning to construct a motion plan for each agent that together may not satisfy DRPC-safety. Then, we compute online corrections to each motion plan individually using decentralized safety filters. In contrast, our prior work [14] computed corrections in a centralized infrastructure that may result in lower resiliency, may incur higher communication costs, and may not scale computationally to very large team sizes.

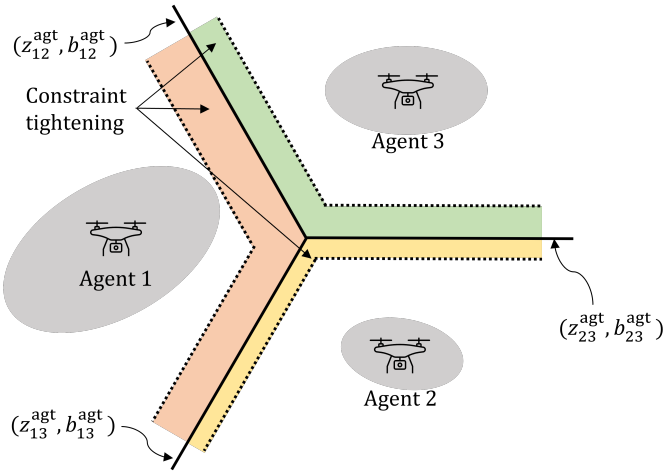

Fig. 2. Convexified, distributionally-robust, inter-agent collision avoidance constraints in Proposition 1. As in [15], we use optimal linear separation to convexify the collision avoidance constraint (10), and then use Lemma 1 to compute the distributionally-robust chance constraint tightening. The shrinkage of the Voronoi cells depends on the moment-ambiguity sets associated with each agent and can be time-varying $(z_{ii'k}^{\text{agt}}, b_{ii'k}^{\text{agt}})$ (see Proposition 1).

For each agent $i \in [1:N]$ at time $t \in \mathbb{N}$, the decentralized safety filter computes corrections to the reference motion plan $\{x_i^{\text{RL}}(k|t)\}_{k=t}^{t+T}$ and control $\{u_i^{\text{RL}}(k|t)\}_{k=t}^{t+T-1}$ by solving,

$$\underset{\{r_i^{\text{safe}}(k|t)\}_{k=t}^{t+T-1}}{\text{minimize}} \quad \sum_{k\in[t:t+T-1]} \lambda_k \|u_i^{\text{RL}}(k|t) - u_i^{\text{safe}}(k|t)\|^2 \tag{12a}$$

$$\text{subject to} \quad \text{Dynamics (7) with } r_i^{\text{safe}},\ \overline{x}_i(t) = \overline{y}_i(t), \tag{12b}$$

$$\forall k\in[t:t+T-1], \quad u_i^{\text{safe}}(k|t) = K\overline{x}_i(k|t) + Fr_i^{\text{safe}}(k|t) \tag{12c}$$

$$\forall k\in[t:t+T-1], \quad r_i^{\text{safe}}(k|t) \in \mathscr{R}, \tag{12d}$$

$$\forall k\in[t:t+T], \quad \text{Safety constraints on } \overline{x}_i(k|t) \text{ at } k, \tag{12e}$$

$$\text{Recursive feas. constraint on } \overline{x}_i(t+T|t), \tag{12f}$$

where $\lambda_k \geq 0$ are pre-specified weights.

Next, we propose convex, agent-specific, sufficient constraints to guarantee DRPC-safety (12e), and terminal constraints (12f) to guarantee recursive feasibility of the mean agent trajectories. We also discuss a temporal discounting of safety to mitigate the conservativeness.

A. Decentralized, convexified enforcement of DRPC-safety

Recall the result in robust optimization [22].

Lemma 1 (CHANCE CONSTRAINT REFORMULATION FOR MOMENT-BASED AMBIGUITY SETS)**.** *Consider a d-dimensional random vector $\boldsymbol{p} \sim \mathbb{P}_p \in \mathscr{P}(\mu, \Sigma)$ and a halfspace $\{p : a \cdot p \leq b\}, a \in \mathbb{R}^d, b \in \mathbb{R}$. For any $\delta \in (0,1)$,*

$$\sup_{\mathbb{P}\in\mathscr{P}(\mu,\Sigma)} \mathbb{P}(a\cdot\boldsymbol{p} \geq b) \leq \delta \iff a\cdot\mu \geq b + \|\Sigma^{\frac{1}{2}}a\|\sqrt{\frac{1-\delta}{\delta}}.$$

We use Lemma 1 to derive convex sufficient constraints to guarantee DRPC-safety (12e) in Proposition 1. The proof is in Appendix A.

Proposition 1. (CONVEXIFIED DRPC CONSTRAINTS) *Consider a polytopic environment with $N_{\mathcal{K}} \in \mathbb{N}$ halfspaces $\mathcal{K} = \cap_{i\in[1:N_{\mathcal{K}}]}\{p \in \mathbb{R}^d : h_i \cdot p \leq g_i\}$ for some $\{h_i, g_i\}_{i=1}^{N_{\mathcal{K}}}$ with $h_i \in \mathbb{R}^d$ and $g_i \in \mathbb{R}$. For any $i, i' \in [1:N]$, $j \in [1:N_O]$, and $k \in [t+1:t+T]$, let $\theta_{ii'k}^*$ solve*

$$z_{ii'k}^{agt}(k|t)^\top \left(\theta^2\Sigma_{p_i}(k|t) - (1-\theta)^2\Sigma_{p_j}(k|t)\right) z_{ii'k}^{agt}(k|t) = 0, \tag{13}$$

and define

$$z_{ijk}^{obs} \triangleq \frac{\overline{c}_j - p_i^{RL}(k|t)}{\|\overline{c}_j - p_i^{RL}(k|t)\|},\ z_{ii'k}^{agt} \triangleq \Theta_{ii'k}^{-1}\frac{p_j^{RL}(k|t) - p_i^{RL}(k|t)}{\|p_j^{RL}(k|t) - p_i^{RL}(k|t)\|}, \tag{14}$$

$$b_{ii'k}^{agt} \triangleq z_{ii'k}^{agt}(k|t)\cdot p_i^{RL}(k|t) + \left\|\Sigma_{p_i}^{\frac{1}{2}}(k|t) z_{ii'k}^{agt}(k|t)\right\|\theta_{ii'k}^*, \tag{15}$$

where $\Theta_{ii'k} = (\theta_{ii'k}^*\Sigma_{p_i}(k|t) + (1-\theta_{ii'k}^*)\Sigma_{p_i}(k|t)) \in \mathbb{R}^{d\times d}$. Then, for any $k \in [t+1:t+T]$, (9), (10), and (11) hold if (16) holds.

Fig. 2 illustrates the inter-agent collision avoidance constraint (16b), where the constraint tightening depends on the agents' rigid body shape $\mathcal{A}$, the associated ambiguity set, and the risk bound β in (10). We obtain a 3-way separation by solving for $\theta_{ii'k}^*$ in the nonlinear equation (13), and then computing $(z_{ii'k}^{\text{agt}}, b_{ii'k}^{\text{agt}})$ using (14) and (15) for *each pair of agents* (i, i') *and time* k. We use these optimal linear separations to convexify the non-convex collision avoidance constraints between agents and between agents and obstacles.

Proposition 1 determines regions in which each agent can be without violating the DRPC-safety. These regions can be viewed as a collection of buffered Voronoi cells [15], [16], where each agent's cell boundaries are given by (14), (15), and the moment-based ambiguity set corresponding to the agents.

Remark 3. *In the presence of communication constraints where only information about "nearby" agents are available to each agent,* (16b) *only includes constraints for the nearby agents. We briefly explore this in Section V-C.*

B. Recursive feasibility constraint

Constraint (12f) aims at ensuring recursive feasibility, which is an open area of research in stochastic MPC [23]. In our prior work [14], the centralized nature of the safety filter allowed us to guarantee recursive feasibility using reachability. However, such an approach is not possible here since each agent computes r_i^{safe} in (12) independently.

For sake of tractability, we focus on ensuring nominal recursive feasibility using terminal equality constraints.

Proposition 2. (NOMINAL RECURSIVE FEASIBILITY) *Let* (12f) *be $\overline{x}_i(t+T|t) \in \mathcal{I}_i$, where $\mathcal{I}_i \subseteq \mathbb{R}^n$ for $i \in [1:N_A]$ is such that, for every $x_i \in \mathcal{I}$, there exists an input sequence $\{u_i^{recurse}(k|t)\}_{k\geq t+T}$ with $\overline{p}_i(k|t) = \overline{p}_i(T|t)$ for all $k \geq t+T$. Then, the mean trajectory of every agent obtained by solving* (12) *is guaranteed to be collision-free for $k \geq t+T$.*

The proof is in Appendix B. Using a 2D double integrator dynamics [14], for every $i \in [1:N_A]$,

$$\mathcal{I}_i \triangleq \{x : C_{\text{vel}}\overline{x}_i(t+T|t) = 0\}. \tag{17}$$

Equation (17) requires zero mean velocity at the end of the planning horizon, and each drone hovering at the terminal position $C_{\text{pos}}\overline{x}_i(t+T|t)$ with $u_i^{\text{recurse}} = 0$. Equation (17)

$$\forall j \in [1:N_O], \quad z^{\text{obs}}_{ijk} \cdot (\overline{p}_i(k|t) - \overline{c}_j) \geq S_{\mathcal{O}_j}(z^{\text{obs}}_{ijk}) + S_{(-\mathcal{A})}(z^{\text{obs}}_{ijk}) + \|(\Sigma_{p_i}(k|t) + \Sigma_{c_j})^{1/2} z^{\text{obs}}_{ijk}\| \sqrt{(T - \alpha_{i,j,t})/\alpha_{i,j,t}}, \tag{16a}$$

$$\forall j \in [1:i-1], \quad z^{\text{agt}}_{ii'k} \cdot (\overline{p}_i(k|t) - \overline{p}_j(k|t)) \geq b^{\text{agt}}_{ii'k} + S_{\mathcal{A}}(z^{\text{agt}}_{ii'k}) + \|(\Sigma_{p_i}(k|t) + \Sigma_{p_j}(k|t))^{1/2} z^{\text{agt}}_{ii'k}\| \sqrt{(T - \beta_{i,j,t})/\beta_{i,j,t}}, \tag{16b}$$

$$\forall j \in [1:N_{\mathcal{K}}], \quad h_j \cdot \overline{p}_i(k|t) \leq g_j - S_{\mathcal{A}}(h_j) - \|(\Sigma_{p_i}(k|t))^{1/2} h_j\| \sqrt{(\kappa_{i,t})/(N_{\mathcal{K}} - \kappa_{i,t})}. \tag{16c}$$

introduces conservativeness by requiring the agents to stop at the end of the planning horizon. However, due to the receding horizon nature of the framework, the observed conservativeness on the actuated command $u_i^{\text{safe}}(t|t)$ is low, especially for longer planning horizons T.

C. Reducing conservativeness via temporal safety discounting

Sections III-A and III-B provide a convex, conservative approximation of (12). Lemma 1 provides an exact reformulation of the chance constraint, but the reformulation is conservative since the distribution encountered in practice may have fewer negative effects than the worst-case one. The convexification of (12) via optimal linear separation and chance constraint reformulations (Proposition 1), and the use of buffered Voronoi cells achieve decentralization, but may add additional conservativeness in practice. Hence, we propose a temporal safety discounting to mitigate the conservativeness via slack variables.

Consider the approximation of (12),

$$\min. \quad \sum_{k\in[t:t+T-1]} \lambda_k \|u_i^{\text{RL}}(k|t) - u_i^{\text{safe}}(k|t)\|^2 + \gamma \left(\sum_{j=1}^{N_O} s_j^{\text{obs}} + \sum_{j=1}^{N_A} s_j^{\text{agt}} + \sum_{j=1}^{N_{\mathcal{K}}} s_j^{\text{env}} \right) \tag{18a}$$

$$\text{s. t.} \quad \text{(12b)} - \text{(12d)}, \text{(17)}$$

$$\forall k\in[t+1:t+T], \forall j\in[1:N_O], \quad \text{LHS of (16a)} + \|(\Sigma_{p_i}(k|t) + \Sigma_{c_j})^{1/2} z^{\text{obs}}_{ijk}\| s^{\text{obs}}_{jk} \geq \text{RHS of (16a)}, \tag{18b}$$

$$\forall k\in[t+1:t+T], \forall j\in[1:i-1], \quad \text{LHS of (16b)} + \|(\Sigma_{p_i}(k|t) + \Sigma_{p_j}(k|t))^{1/2} z^{\text{agt}}_{ii'k}\| s^{\text{agt}}_{jk} \geq \text{RHS of (16b)}, \tag{18c}$$

$$\forall k\in[t+1:t+T], \forall j\in[1:N_{\mathcal{K}}], \quad \text{LHS of (16c)} - \|(\Sigma_{p_i}(k|t))^{1/2} h_j\| s^{\text{env}}_{jk} \leq \text{RHS of (16c)}, \tag{18d}$$

$$\forall k\in[t+1:t+T-1], \forall \text{ appropriate } j, \quad 0 \leq s^{\text{obs}}_{jk} \leq s^{\text{obs}}_{j(k+1)}, \; 0 \leq s^{\text{agt}}_{jk} \leq s^{\text{agt}}_{j(k+1)}, \; 0 \leq s^{\text{env}}_{jk} \leq s^{\text{env}}_{j(k+1)}, \tag{18e}$$

Here, (18b)–(18e) together enforce (16) after relaxing (16a)–(16c) using non-negative slack variables $s^{\text{obs}}_{jk}, s^{\text{agt}}_{jk}, s^{\text{env}}_{jk}$, and (18e) requires the slack variables be non-decreasing over the planning horizon, see Fig. 3. For a sufficiently large temporal safety discounting penalty $\gamma > 0$, (18) will provide a feasible solution to (12) when possible, due to its use of ℓ_1-penalty [24, Ch. 17]. Since (18) is a convex quadratic program, it that can be efficiently solved using off-the-shelf solvers [25].

By construction, (18) is guaranteed to always be feasible for a sufficiently long planning horizon T. Also, when all slack variables are zero, the solution to (18) guarantees DRPC-safety for all time steps $k \in [t : t + T]$, and Proposition 2 guarantees nominal recursive feasibility. Otherwise, we define T_{safe} as the furthest time step k into the future with slack variables $s^{\text{obs}}_{jk}, s^{\text{agt}}_{jk}, s^{\text{env}}_{jk}$ as zero for all corresponding j (see Fig. 3), and the computed solution to (18) guarantees DRPC-safety for all time steps $k \in [t : t + T_{\text{safe}}]$. We recommend imposing slack variables for $k = 0$ to zero to ensure $T_{\text{safe}} \geq 1$.

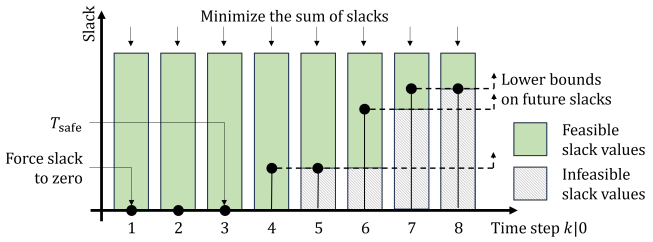

Fig. 3. Illustration of the proposed temporal safety discounting. We relax the constraints associated with safety using slack variables, where the relaxation of far-future constraints are preferred over near-future constraints. In certain instants $k \in \{4, 6, 7\}$, the slack values relax safety for the feasibility of (18), and may be higher than the minimum needed by the temporal requirement. T_{safe} is the furthest future time step with a zero slack variable, and the first slack variable is forced to be zero to ensure safety $T_{\text{safe}} \geq 1$.

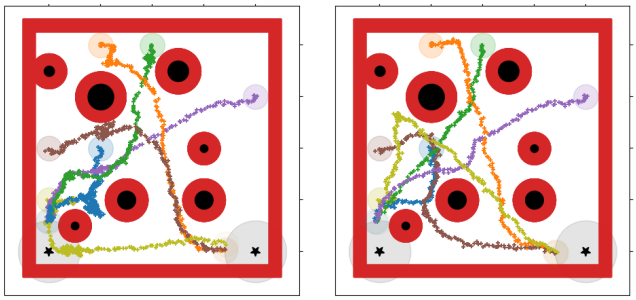

Fig. 4. Observed drone trajectories when using the motion plans generated by the proposed approach (left) and a centralized safety filter [14] (right). See https://youtu.be/Yy_ukIXwqqE for the experiment videos.

IV. Experiments

As in [14], we validate the proposed approach using six drones ($N_A = 6$) in a 3m $\times$ 3m workspace with seven circular obstacles ($N_O = 7$) and two goal regions. We depict the obstacles by black circles and the goal regions by transparent gray circles with a star at the center in Fig. 4.

We use the `Crazyswarm` platform [26] to communicate with the Crazyflie drones, and localize the drones using an OptiTrack motion capture system running at 120 Hz. We generate the motion plans and provide waypoints to the drones at 10 Hz over radio using `Crazyswarm`. The Crazyflies regulate to these waypoints using standard on-board controllers. We introduce a position estimation noise defined in (3b) to the agent dynamics as well as the nominal obstacle locations. Such measurement noises affect the safety filter, but are not visualized in the plots.

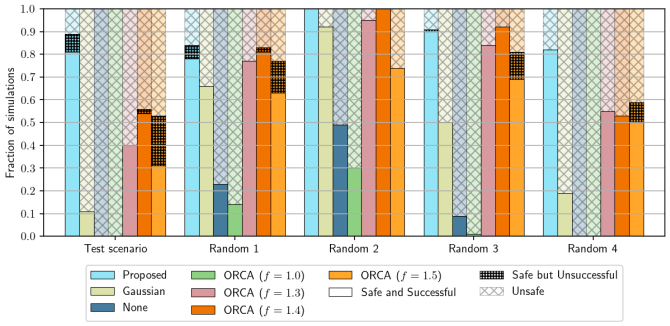

Fig. 5. Satisfaction of DRPC-safety along the entire trajectory in a Monte-Carlo simulation of 100 trials for the experiment set up used in Section IV (`Test Scenario`) and 4 other randomly chosen problem instances by various methods: Proposed approach using distributionally-robust chance constraint tightening (Proposed), Gaussian-based chance constraint tightening (Gaussian) [15], no constraint tightening (None), and different variants of optimal reciprocal collision avoidance (ORCA) [8]. Due to the non-Gaussian nature of the stochastic uncertainty in the agent model and the environment, the proposed approach exhibits safer behavior than existing approaches.

For motion plan generation, we use 2D double integrator models, similar to [14], [20], [21]. The on-board controller regulates the attitude dynamics, where yaw is not relevant for our problem, and compensates for the approximated model.

We use individual chance constraint risk thresholds $\kappa_{i,k} = \alpha_{i,j,k} = \beta_{i,i',k} = 0.1$, temporal safety discounting penalty $\gamma = 10^3$, and $\lambda_k = 1$. Since in [26], the maximum position tracking error of crazyflies was approximately 2.5 cm, we use $\Sigma_w = \Sigma_\eta = 6\text{diag}(10^{-5}, 0, 10^{-5}, 0)$ and $\Sigma_{c_j} = 6 \times 10^{-5} I_2$ for each $j \in [1 : N_O]$ to ensure that Lemma 1 provides a sufficient margin in the safety filter (≈ 2.5 cm with 0.9 probability). We use the Laplace distribution, which is a non-Gaussian, heavy-tailed distribution, to draw realizations of $\boldsymbol{w}, \boldsymbol{\eta}$, and $\boldsymbol{c}_j$.

We use an Ubuntu 20.04 LTS workstation with an AMD Ryzen 9 9590X 16-core CPU, a Nvidia GeForce GTX TITAN Black GPU, and 128GB of RAM for all training, simulation, and experiments. We use `Stable-Baselines3`'s implementation of the proximal policy optimization (PPO) algorithm [18] to train the RL agents. We run two training sessions, one for each goal, for 10^7 time steps each. Each training session took just over 11 hours, see [14] for more details. We model the quadratic program associated with (18) in Python 3.7 using `CVXPY` [27], and solve it using `ECOS` [25].

Fig. 4 shows the generated motion plans using the proposed decentralized approach and our prior centralized approach in [14]. Both approaches successfully accomplish the task while respecting the safety constraints. The centralized approach completes the task in 24.4 seconds, while the decentralized approach took 39 seconds. However, at each time step, the decentralized safety filter needs about less than 20% of compute time than the centralized safety filter (0.017 seconds for each agent vs 0.092 seconds for the team), and does not need a centralized infrastructure.

V. Simulation-based performance analysis

For a more extensive assessment of the performance of the proposed approach, we perform a simulation-based analysis. First, we compare the performance and safety provided by our proposed approach with those of some existing methods, showing that we generate safer trajectories without significant compromises in performance on several randomly chosen motion planning problem instances. Second, we study the effect of varying the temporal safety discounting penalty γ in (18) on the safety guarantees of the proposed approach. Finally, we empirically demonstrate that reasonable constraints on communication do not significantly degrade safety and performance of the proposed approach.

TABLE I

$(0.05, 0.5, 0.95)$-quantiles of the safety horizon as well as the time until success/time-out or failure, and the percentage of trials that succeeded, timed-out, or failed aggregated over the five scenarios considered. The proposed approach is safer than existing approaches without compromising on the performance of the task.

Constraint tightening	Safety horizon T_{safe}	Time until success/timed-out	Time to failure	% of 500 trials with		
				Success	Timed-out	Collisions
Proposed	10 / 10 / 10	144 / 216 / 653	1 / 66 / 178	86.2 %	3.0 %	10.8 %
Gaussian	10 / 10 / 10	124 / 144 / 332	11 / 66 / 190	49.8 %	0.0 %	50.2 %
None	10 / 10 / 10	99 / 127 / 152	6 / 33 / 84	13.6 %	0.0 %	86.4 %
ORCA ($f = 1.0$)	Not Applicable	118 / 139 / 194	7 / 34 / 115	9.0 %	0.0 %	91.0 %
ORCA ($f = 1.3$)		185 / 252 / 467	24 / 99 / 315	70.2 %	0.0 %	29.8 %
ORCA ($f = 1.4$)		175 / 279 / 603	42 / 142 / 399	76.0 %	0.8 %	23.2 %
ORCA ($f = 1.5$)		230 / 427 / 800	61 / 153 / 664	57.4 %	11.4 %	31.2 %

A. Comparison with existing approaches

We compared our proposed approach with several existing methods: 1) the buffered Voronoi cell in [15] that assumes Gaussian uncertainty, 2) a variant of [15] where no constraint tightening was applied, i.e., the Voronoi cells are not shrunk, and 3) several variants of optimal reciprocal collision avoidance (ORCA) [7], [8] where the agent radii were artificially inflated to (approximately) account for uncertainty, specifically scaling by $f \in \{1, 1.3, 1.4, 1.5\}$. These inflations of the agent geometry are heuristics, whereas our proposed approach uses a systematic constraint tightening based on dynamics and uncertainty models.

We study the performance of various approaches over four randomly generated motion planning problem instances and the `Test Scenario` used in the experiments in Section IV. We perform a Monte-Carlo simulation of 100 trials for different noise realizations on each of the scenarios.

Table I aggregates the performance of various approaches under study over the 500 trials. Our proposed approach has a significantly larger fraction of safe simulations than the other methods due to the correct treatment of non-Gaussian uncertainties, with 86.2% of the trials being safe and successful. Including timed-out cases where the proposed approach kept the team safe but was unsuccessful in steering all agents to their targets within 800 time steps, the proposed approach is safe in 89.2% of the trials.

Fig. 5 shows the safety of the various approaches considered for each of the scenarios. Our proposed approach is safer than existing approaches in all scenarios except in `Random 3`, where ORCA with $f = 1.4$ is slightly safer.

Fig. 6 shows the utilization of the proposed temporal discounting of safety by various safety filters. Since the existing buffered Voronoi cell-based approach [15] underestimates the effect of uncertainty, both the cases with Gaussian-based chance constraint tightening and without constraint tightening

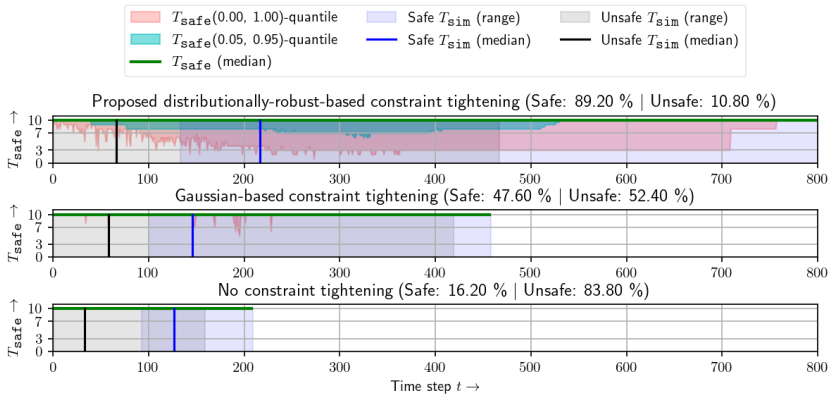

Fig. 6. Variation of the safety horizon based on the temporal safety discounting over 500 simulations. Compared to the Gaussian-based constraint tightening [15], the proposed distributionally-robust-based approach uses slack variables more effectively to recover feasibility.

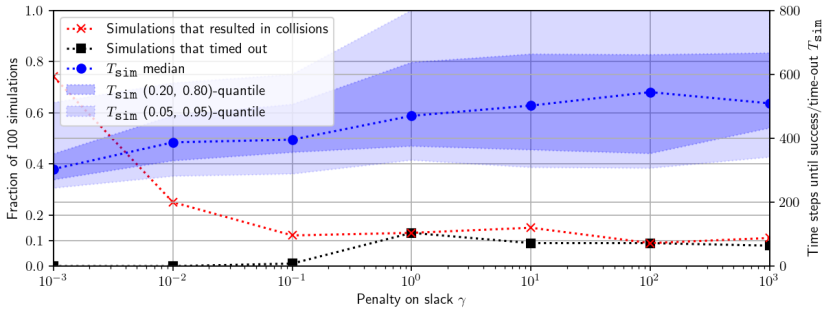

Fig. 7. Effect of varying the temporal safety discounting penalty γ in (18) on the task failure (collision or timed-out) and time taken to complete the task T_{sim}. Larger γ yields improved safety with limited performance degradation.

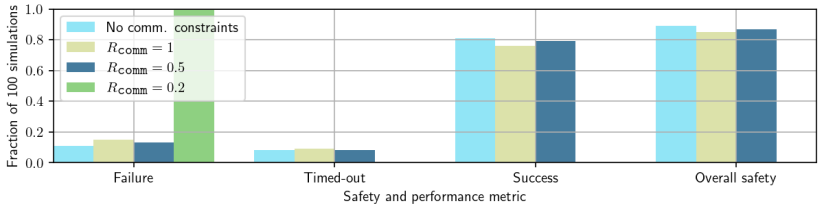

Fig. 8. Safety and performance for the proposed approach under varying communication ranges. Reasonable constraints on communication do not significantly deteriorate the performance and safety of the proposed approach.

typically do not use the slack variables. On the other hand, our approach uses the temporal safety discounting to mitigate the conservativeness effectively with $T_{\text{safe}} \geq 8$ in most of the time steps. While existing approaches complete the task faster (lower T_{sim}), they may underestimate the effect of noise and are typically unsafe, due to mismatch in the uncertainty model.

B. Trade-off between temporal discounting and correction

Fig. 7 shows the effect of varying the temporal safety discounting penalty γ in (18) on the performance and safety of the proposed approach. We perform the variation study on the proposed approach on the `Test Scenario` for 100 simulations and varied $\gamma \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\}$.

Increasing the penalty γ drastically reduces the percentage of trials in the Monte-Carlo simulation that result in collision, while moderately increasing the percentage of timed-out trials. In other words, increasing the penalty γ significantly increased the percentage of trials that are safe and successful. However, the time T_{sim} taken by the agents to safely and successfully reach the targets also increases with larger values of γ, indicating a trade-off between performance and safety.

C. Enforcement of communication constraints

Fig. 8 shows the performance and safety of the proposed approach when imposing a constraint on communication, see Remark 3. Specifically, we consider a variant of the proposed approach, where the information about other agents are available to an ego agent only when they are within a pre-defined distance characterized by a communication radius, $R_{\text{comm}} \in \{0.2, 0.5, 1, \infty\}$. Under communication constraints, we relax (18c) to only include agents that are "close" enough to exchange information.

The proposed approach exhibits only moderate degradation of safety, i.e., higher failure rates, when imposing a communication radius. The fractions of trials that were safe and successful for $R_{\text{comm}} \in \{0.5, 1, \infty\}$ are similar. This

observation follows from the intuition that the information of nearby agents are more pertinent to guarantee safety compared to far-away agents. On the other hand, as expected, reducing the communication radius further down to 0.2 resulted in a breakdown of safety with no safe trials, since agents no longer have the necessary information to prevent collisions in time. Larger teams may also face degradation of safety due to channel capacity, which was not observed in our experiments.

VI. Conclusion

This paper proposes a decentralized, multi-agent motion planner that guarantees probabilistic safety of multi-agent teams under uncertainty. The proposed planner uses a combination of off-the-shelf, single-agent reinforcement learning, distributionally-robust and convex optimization, and buffered Voronoi cells to generate the motion plans in real-time. We analyzed the performance of our approach in simulation and demonstrated it in physical experiments using drones.

References

[1] S. Semnani, H. Liu, M. Everett, A. de Ruiter, and J. How, "Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning," *IEEE Rob. Auto. Lett.*, pp. 3221–3226, 2020.

[2] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook Reinf. Learn. Ctrl.*, pp. 321–384, 2021.

[3] D. Malyuta, T. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Açıkmeşe, "Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently," *IEEE Ctrl. Syst. Mag.*, vol. 42, no. 5, pp. 40–113, 2022.

[4] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. Ctrl. Syst. Tech.*, vol. 29, pp. 972–983, 2020.

[5] B. Luders, S. Karaman, and J. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *AIAA Guid. Nav. Ctrl. Conf.*, p. 5097, 2013.

[6] K. Ekenberg, V. Renganathan, and B. Olofsson, "Distributionally robust RRT with risk allocation," in *IEEE Int'l Conf. Rob. Autom.*, 2023.

[7] J. van den Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.

[8] D. Bareiss and J. Van Den Berg, "Generalized reciprocal collision avoidance," *Int'l J. Rob. Res.*, vol. 34, no. 12, pp. 1501–1514, 2015.

[9] L. Lv, S. Zhang, D. Ding, and Y. Wang, "Path planning via an improved DQN-based learning policy," *IEEE Access*, pp. 67319–67330, 2019.

[10] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Int'l. Conf. Neural Info. Process. Syst.*, pp. 6382–6393, 2017.

[11] M. Everett, Y. Chen, and J. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *IEEE Int'l Conf. Intel. Rob. Syst.*, pp. 3052–3059, 2018.

[12] L. Hewing, J. Kabzan, and M. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Trans. Ctrl. Syst. Tech.*, vol. 28, no. 6, pp. 2736–2743, 2019.

[13] A. Vinod, S. Safaoui, A. Chakrabarty, R. Quirynen, N. Yoshikawa, and S. Di Cairano, "Safe multi-agent motion planning via filtered reinforcement learning," in *IEEE Int'l Conf. Rob. Autom.*, pp. 7270–7276, 2022.

[14] S. Safaoui, A. Vinod, A. Chakrabarty, R. Quirynen, N. Yoshikawa, and S. Di Cairano, "Safe multi-agent motion planning under uncertainty using filtered reinforcement learning," *IEEE Trans. Rob.*, vol. 40, pp. 2529–2542, 2024.

[15] H. Zhu, B. Brito, and J. Alonso-Mora, "Decentralized probabilistic multi-robot collision avoidance using buffered uncertainty-aware Voronoi cells," *Autonomous Robots*, pp. 1–20, 2022.

[16] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, on-line collision avoidance for dynamic vehicles using buffered Voronoi cells," *IEEE Rob. Auto. Lett.*, vol. 2, no. 2, pp. 1047–1054, 2017.

[17] S. LaValle, *Planning algorithms*. Cambridge Univ. Press, 2006.

[18] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *J. Mach. Learn. Res.*, pp. 1–8, 2021.

[19] X. Li, C. Vasile, and C. Belta, "Reinforcement learning with temporal logic rewards," in *IEEE Int'l Conf. Intel. Rob. Syst.*, 2017.

[20] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *IEEE Int'l Conf. Intel. Rob. Syst.*, pp. 1917–1922, 2012.

[21] Y. Chen, M. Cutler, and J. P. How, "Decoupled multiagent path planning via incremental sequential convex programming," in *IEEE Int'l Conf. Rob. Autom.*, pp. 5954–5961, 2015.

[22] J. Paulson, E. Buehler, R. Braatz, and A. Mesbah, "Stochastic model predictive control with joint chance constraints," *Intn'l J. Ctrl.*, vol. 93, no. 1, pp. 126–139, 2020.

[23] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Ctrl. Syst. Mag.*, pp. 30–44, 2016.

[24] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.

[25] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *Proc. Euro. Ctrl. Conf.*, pp. 3071–3076, 2013.

[26] J. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *IEEE Int'l Conf. Rob. Autom.*, pp. 3299–3304, 2017.

[27] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2909–2913, 2016.

[28] J. Rawlings, D. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*, vol. 2. Nob Hill Publishing, 2017.

Appendix

A. Proof of Proposition 1

(16a): First, we use the supporting hyperplane theorem to convexify the collision constraint as follows,

$$z_{ijk}^{\text{obs}} \cdot (\boldsymbol{p}_i(k|t) - \boldsymbol{c}_j) \geq S_{\mathcal{O}_j}(z_{ijk}^{\text{obs}}) + S_{(-\mathcal{A})}(z_{ijk}^{\text{obs}}), \tag{19}$$

similarly to our prior work [14, Prop. 1]. Specifically, in a non-stochastic setting, (19) is sufficient to ensure that $(\overline{p}_i(k|t) \oplus \mathcal{A}) \cap (\overline{c}_j(t) \oplus \mathcal{O}_j) \neq \emptyset$. Then, we obtain (16a) using Lemma 1 to characterize a deterministic reformulation of (9).

(16b): For inter-agent collision avoidance, we first compute an optimal linear separator by solving

$$\min_{a,b,\delta} \ \delta \tag{20a}$$

$$\text{s. t. } \sup_{\mathbb{P}\in\mathscr{P}_i\times\mathscr{P}_j} \mathbb{P}(a \cdot \boldsymbol{p}_i(k|t) > b) \leq \delta, \tag{20b}$$

$$\sup_{\mathbb{P}\in\mathscr{P}_i\times\mathscr{P}_j} \mathbb{P}(a \cdot \boldsymbol{p}_j(k|t) \leq b) \leq \delta, \tag{20c}$$

where $\mathscr{P}_i \times \mathscr{P}_j = \mathscr{P}_i(p_i^{\text{RL}}(k|t), \Sigma_{p_i}(k|t)) \times \mathscr{P}_j(p_j^{\text{RL}}(k|t), \Sigma_{p_j}(k|t))$. Since $1/(1+x^2)$ is monotonic in x, (20) is equivalent to

$$\min_{a,b} \max_{i,j} \left\{ \frac{b - a \cdot p_i^{\text{RL}}(k|t)}{\sqrt{a \cdot (\Sigma_{p_i}(k|t)a)}}, \frac{a \cdot p_j^{\text{RL}}(k|t) - b}{\sqrt{a \cdot (\Sigma_{p_j}(k|t)a)}} \right\}. \tag{21}$$

by the epigraph reformulation and Chebyshev-Cantelli inequality. Equation (21) is identical to the optimal linear separation problem for Gaussian distributions with the same specified mean and covariance matrices [15]. Using the discussion in [15], (a, b) may be obtained as in (13), (14), (15). We obtain a convexification similar to (19) using a^*, b^*, and (16b) from Lemma 1.

(16c): Follows from Boole's inequality and Lemma 1.

B. Proof of Proposition 2

By definition, when $\overline{x}_i(t+T|t) \in \mathcal{I}_i$, by applying u_i^{recurse}, the mean positions of the agents remain constant beyond the planning horizon. Thus, this is a particular case of guaranteeing recursive feasibility in MPC using terminal equality constraints [28], and the recursive feasibility of (12) follows.