

Distributed Model Predictive Covariance Steering

Augustinos D. Saravanos¹, Isin M. Balci², Efstathios Bakolas² and Evangelos A. Theodorou¹

Abstract—This paper proposes Distributed Model Predictive Covariance Steering (DiMPCS) for multi-agent control under stochastic uncertainty. The scope of our approach is to blend covariance steering theory, distributed optimization and model predictive control (MPC) into a single framework that is safe, scalable and decentralized. Initially, we pose a problem formulation that uses the Wasserstein distance to steer the state distributions of a multi-agent system to desired targets, and probabilistic constraints to ensure safety. We then transform this problem into a finite-dimensional optimization one by utilizing a disturbance feedback policy parametrization for covariance steering and a tractable approximation of the safety constraints. To solve the latter problem, we derive a decentralized consensus-based algorithm using the Alternating Direction Method of Multipliers. This method is then extended to a receding horizon form, which yields the proposed DiMPCS algorithm. Simulation experiments on a variety of multi-robot tasks with up to hundreds of robots demonstrate the effectiveness of DiMPCS. The superior scalability and performance of the proposed method is also highlighted through a comparison against related stochastic MPC approaches. Finally, hardware results on a multi-robot platform also verify the applicability of DiMPCS on real systems.

I. INTRODUCTION

Multi-robot control is a domain with a significant variety of applications such as swarm robotics [1], multi-UAV navigation [2], motion planning [3], underwater vehicles [4], and so forth. As the scale and complexity of such systems continuously increases, some of the most desired attributes for algorithms designed to control these systems include *safety under uncertainty*, *scalability* and *decentralization*.

Model predictive control (MPC) has found several successful multi-robot applications [5], [6], [7], thanks to its optimization-based nature and intrinsic feedback capabilities. In the case where stochastic disturbances are present, several stochastic MPC (SMPC) approaches have been proposed for handling them such as [8], [9], [10], [11]. Nevertheless, the literature in combining MPC with the steering of the state distribution of a system to exact targets for enhancing safety remains quite scarce [12], [13], [14].

Covariance steering (CS) theory considers a class of stochastic optimal control problems, where the main objective is to steer the state mean and covariance of a system to desired targets. While initial CS approaches had dealt with infinite-horizon problems for linear time-invariant systems [15], [16], finite-horizon CS methods that also address linear

time-variant dynamics, have recently gained attention such as [17], [18], [19], [20]. Several successful robotics applications of CS can be found in motion planning [21], trajectory optimization [13], [14], multi-agent control [22], [23], etc.

In SMPC based methods, it is typically the feed-forward control inputs that are treated as optimization variables, while the feedback gains are fixed to a stabilizing value for the closed-loop system [9]. However, the state covariance cannot actively be steered with such methods, while fixed static feedback gains might perform poorly for time-varying dynamics. Thus, control policies resulting from standard SMPC approaches might be suboptimal and/or overly conservative against safety criteria. On the contrary, CS methods yield the optimal feedback gains that steer the state covariance to the desired targets, thus providing more flexibility to satisfy optimality and safety guarantees at the same time.

Although CS allows for finding the optimal control policies to steer the state statistics to desired values in the unconstrained case, the latter might be unreachable in the presence of state and/or input constraints. In MPC applications, especially, such infeasibilities can occur quite frequently, since the prediction horizon is usually much smaller than the total time horizon. Therefore, it would be desirable to penalize the deviation from the desired state statistics by utilizing a distance metric between distributions such as the Wasserstein distance [24], instead of imposing hard constraints [12].

In addition, the main limitation of applying CS methods to large-scale multi-robot systems lies in the fact that computational demands increase significantly with respect to the state/control dimension and time horizon. Nevertheless, recent work [22] has shown that this computational burden can be significantly alleviated by merging CS with the Alternating Direction Method of Multipliers (ADMM), an optimization procedure that has found several recent applications in decentralized control [25], [26], [27], [28].

In this paper, we propose Distributed Model Predictive Covariance Steering (DiMPCS) for safe and scalable multi-robot navigation. First, we provide a problem formulation which utilizes the Wasserstein distance for steering the robots to prescribed target distributions and probabilistic constraints for ensuring their safe operation. Subsequently, by exploiting CS theory, a suitable disturbance feedback policy parametrization, and an efficient approximation of the safety constraints, we transform the original problem into a finite-dimensional optimization one. To solve this, we propose an ADMM-based method for establishing consensus between neighboring robots and achieving decentralization. The latter method is then extended to an MPC scheme, which yields the final DiMPCS algorithm. Simulation experiments

¹ Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, GA, USA {asaravanos, evangelos.theodorou}@gatech.edu

² Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, TX, USA {isinmertbalci, bakolas@austin}@utexas.edu,

on several multi-agent navigation tasks with up to hundreds of robots illustrate the efficacy and scalability of DiMPCS. In addition, the advantages of the proposed method in terms of scalability and safety performance are also underlined through comparing with related SMPC approaches. Finally, we provide hardware experiments on a multi-robot platform which verify the effectiveness of DiMPCS on actual systems.

II. PROBLEM DESCRIPTION

A. Notation

The space of $n \times n$ symmetric, positive semi-definite (definite) matrices is denoted with \mathbb{S}_n^+ (\mathbb{S}_n^{++}). The $n \times n$ identity matrix is denoted as I_n whereas $\mathbf{0}$ denotes the zero matrix (or vector) with appropriate dimensions. The trace operator is denoted with $\text{tr}(\cdot)$. The expectation and covariance of a random variable (r.v.) $x \in \mathbb{R}^n$ are given by $\mathbb{E}[x] \in \mathbb{R}^n$ and $\text{Cov}[x] \in \mathbb{S}_n^+$, respectively. With $x \sim \mathcal{N}(\mu, \Sigma) \in \mathbb{R}^n$, we refer to a Gaussian r.v. x with $\mathbb{E}[x] = \mu$ and $\text{Cov}[x] = \Sigma$. With $\llbracket a, b \rrbracket$, we denote the integer set $[a, b] \cap \mathbb{Z}$ for any $a, b \in \mathbb{R}$. The cardinality of a set \mathcal{X} is denoted with $|\mathcal{X}|$. Finally, given a set \mathcal{C} , we denote with $\mathcal{I}_{\mathcal{C}}(x)$ the indicator function such that $\mathcal{I}_{\mathcal{C}}(x) = 0$ if $x \in \mathcal{C}$ and $\mathcal{I}_{\mathcal{C}}(x) = +\infty$, otherwise.

B. Problem Description

Let us consider a team of N robots given by the set $\mathcal{V} = \{1, \dots, N\}$. Each robot $i \in \mathcal{V}$ is subject to the following discrete-time, stochastic, nonlinear dynamics

$$x_{i,k+1} = f_i(x_{i,k}, u_{i,k}) + w_{i,k}, \quad x_{i,0} \sim \mathcal{N}_{i,0}, \quad (1)$$

for $k \in \llbracket 0, K \rrbracket$, where K is the time horizon, $x_{i,k} \in \mathbb{R}^{n_i}$, $u_{i,k} \in \mathbb{R}^{m_i}$ and $f_i : \mathbb{R}^{n_i \times m_i} \rightarrow \mathbb{R}^{n_i}$ are the state, control input and transition dynamics of the i -th robot, and $w_{i,k} \sim \mathcal{N}(0, W_i)$ with $W_i \in \mathbb{S}_{n_i}^+$. Each robot's initial state $x_{i,0} \sim \mathcal{N}_{i,0} = \mathcal{N}(\mu_{i,0}, \Sigma_{i,0})$ with $\mu_{i,0} \in \mathbb{R}^{n_i}$ and $\Sigma_{i,0} \in \mathbb{S}_{n_i}^+$.

The position of the i -th robot in 2D (or 3D) space is denoted with $p_{i,k} \in \mathbb{R}^q$ with $q = 2$ (or $q = 3$) and can be extracted with $p_{i,k} = H_i x_{i,k}$, where $H_i \in \mathbb{R}^{q \times n_i}$ is defined accordingly. Furthermore, the environment, wherein the robots operate, includes circle (in 2D) or spherical (in 3D) obstacles given by the set $\mathcal{O} = \{1, \dots, O\}$, where each obstacle $o \in \mathcal{O}$ has position $p_o \in \mathbb{R}^q$ and radius $r_o \in \mathbb{R}$.

We consider the problem of steering the state distributions of all robots $i \in \mathcal{V}$ to the target Gaussian ones $\mathcal{N}_{i,f} = \mathcal{N}(\mu_{i,f}, \Sigma_{i,f})$ with $\mu_{i,f} \in \mathbb{R}^{n_i}$, $\Sigma_{i,f} \in \mathbb{S}_{n_i}^{++}$. To penalize the deviation of the actual distributions from the target ones, we utilize the notion of the Wasserstein distance as a metric to describe similarity between r.v. probability distributions [24]. In particular, we define the following cost:

$$J_i := \sum_{k=1}^K \mathcal{W}_2^2(x_{i,k}, x_{i,f}) + \mathbb{E} \left[\sum_{k=0}^{K-1} u_{i,k}^T R_i u_{i,k} \right], \quad (2)$$

for each robot $i \in \mathcal{V}$, where $x_{i,f} \sim \mathcal{N}_{i,f}$, $\mathcal{W}_2^2(x_a, x_b)$ is the squared Wasserstein distance between x_a, x_b and $R_i \in \mathbb{S}_{m_i}^+$.

The following probabilistic collision avoidance constraints between the robots and the obstacles are also imposed

$$\mathbb{P}(\|p_{i,k} - p_o\|_2 \geq d_{i,o} + r_o) \geq 1 - \alpha, \quad \forall k \in \llbracket 0, K \rrbracket, i \in \mathcal{V}, o \in \mathcal{O}, \quad (3)$$

where $0 < \alpha < 0.5$ and $d_{i,o} \in \mathbb{R}$ is the minimum allowed distance between the center of robot i and obstacle o . In addition, we also wish for all robots to avoid collisions with each other, through the following constraints

$$\mathbb{P}(\|p_{i,k} - p_{j,k}\|_2 \geq d_{i,j}) \geq 1 - \alpha, \quad \forall k \in \llbracket 0, K \rrbracket, i \in \mathcal{V}, j \in \mathcal{V} \setminus \{i\}, \quad (4)$$

where $d_{i,j} \in \mathbb{R}$ is the minimum allowed distance between the centers of the robots i and j .

Let us also define the sets of admissible control policies of the robots. A control policy for robot $i \in \mathcal{V}$ is a sequence $\pi_i = \{\tau_{i,0}, \tau_{i,1}, \dots, \tau_{i,K}\}$ where each $\tau_{i,k} : \mathbb{R}^{n_i(k+1)} \rightarrow \mathbb{R}^{m_i}$ is a function of $\mathcal{X}_{i,0:k} = \{x_{i,0}, \dots, x_{i,k}\}$ that is the set of states already visited by robot i at time k . The set of admissible policies for robot i is denoted as Π_i . Finally, any additional control constraints we wish to impose are represented as $u_{i,k} \in \mathcal{U}_i$. The multi-robot distribution steering problem can now be formulated as follows.

Problem 1 (Multi-Robot Distribution Steering Problem):

Find the optimal control policies π_i^* , $\forall i \in \mathcal{V}$, such that

$$\{\pi_i^*\}_{i \in \mathcal{V}} = \underset{i \in \mathcal{V}}{\text{argmin}} \sum J_i(\pi_i)$$

s.t. (1), (3), (4), $u_{i,k} = \tau_{i,k}(\mathcal{X}_i^k) \in \Pi_i$, $u_{i,k} \in \mathcal{U}_i$, $i \in \mathcal{V}$.

III. MULTI-AGENT COVARIANCE STEERING WITH WASSERSTEIN DISTANCE

The scope of this work is to address Problem 1 through leveraging CS theory, MPC and distributed optimization. While CS methods have mainly been developed for linear dynamics, they can be extended for nonlinear ones by linearizing around the mean of some reference trajectory [29], [30], [31]. After linearization, we utilize a disturbance feedback policy parametrization which yields closed form expressions for the state means and covariances. Finally, we transform Problem 1 to an approximate finite-dimensional optimization one over the new policy parameters.

A. Dynamics Linearization

By considering the first-order Taylor expansion of $f_i(x_{i,k}, u_{i,k})$ around some nominal trajectories $x'_i = [x'_{i,0}; \dots; x'_{i,K}]$, $u'_i = [u'_{i,0}; \dots; u'_{i,K-1}]$, we obtain the discrete-time, stochastic, linear time-variant dynamics

$$x_{i,k+1} = A_{i,k} x_{i,k} + B_{i,k} u_{i,k} + r_{i,k} + w_{i,k}, \quad x_{i,0} \sim \mathcal{N}_{i,0}, \quad (5)$$

where $A_{i,k} \in \mathbb{R}^{n_i \times n_i}$, $B_{i,k} \in \mathbb{R}^{n_i \times m_i}$ and $r_{i,k} \in \mathbb{R}^{n_i}$ are given by

$$A_{i,k} = \left. \frac{\partial f}{\partial x_k} \right|_{\substack{x_k = x'_k \\ u_k = u'_k}}, \quad B_{i,k} = \left. \frac{\partial f}{\partial u_k} \right|_{\substack{x_k = x'_k \\ u_k = u'_k}}, \quad (6a)$$

$$r_{i,k} = f_i(x'_{i,k}, u'_{i,k}) - A_{i,k} x'_{i,k} - B_{i,k} u'_{i,k}. \quad (6b)$$

Therefore, each state trajectory can be expressed as

$$\mathbf{x}_i = \mathbf{G}_{i,0}\mathbf{x}_{i,0} + \mathbf{G}_{i,u}\mathbf{u}_i + \mathbf{G}_{i,w}\mathbf{w}_i + \mathbf{G}_{i,w}\mathbf{r}_i, \quad (7)$$

where $\mathbf{x}_i = [x_{i,0}; \dots; x_{i,K}] \in \mathbb{R}^{(K+1)n_i}$, $\mathbf{u}_i = [u_{i,0}; \dots; u_{i,K-1}] \in \mathbb{R}^{Km_i}$, $\mathbf{w}_i = [w_{i,0}; \dots; w_{i,K-1}] \in \mathbb{R}^{Kn_i}$, $\mathbf{r}_i = [r_{i,0}; \dots; r_{i,K-1}] \in \mathbb{R}^{Kn_i}$, and the matrices $\mathbf{G}_{i,0}$, $\mathbf{G}_{i,u}$ and $\mathbf{G}_{i,w}$ can be found in Eq. (9), (10) in [32].

B. Controller Parametrization

Let us now consider the following affine disturbance feedback control policies, introduced in [33],

$$u_{i,k} = \bar{u}_{i,k} + L_{i,k}(x_{i,0} - \mu_{i,0}) + \sum_{l=0}^{k-1} K_{i,(k-1,l)} w_{i,l}, \quad (8)$$

where $\bar{u}_{i,k} \in \mathbb{R}^{m_i}$ are the feed-forward parts of the control inputs and $L_{i,k}$, $K_{i,(k-1,l)} \in \mathbb{R}^{m_i \times n_i}$ are feedback matrices. Here, we assume perfect state measurements, such that the disturbances that have acted upon the system can be obtained. It follows that $\mathbf{u}_i = \bar{\mathbf{u}}_i + \mathbf{L}_i(x_{i,0} - \mu_{i,0}) + \mathbf{K}_i\mathbf{w}_i$, where $\bar{\mathbf{u}}_i = [\bar{u}_{i,0}; \dots; \bar{u}_{i,K-1}] \in \mathbb{R}^{Km_i}$ and $\mathbf{L}_i \in \mathbb{R}^{Km_i \times n_i}$, $\mathbf{K}_i \in \mathbb{R}^{Km_i \times Kn_i}$ are given by $\mathbf{L}_i = [L_{i,0}; \dots; L_{i,K-1}]$ and

$$\mathbf{K}_i = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ K_{i,(0,0)} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ K_{i,(1,0)} & K_{i,(1,1)} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{i,(K-2,0)} & K_{i,(K-2,1)} & \dots & K_{i,(K-2,K-2)} & \mathbf{0} \end{bmatrix}.$$

Thus, the state trajectory of the i -th robot is obtained with

$$\mathbf{x}_i = \mathbf{G}_{i,0}\mathbf{x}_{i,0} + \mathbf{G}_{i,u}\bar{\mathbf{u}}_i + \mathbf{G}_{i,u}\mathbf{L}_i(x_{i,0} - \mu_{i,0}) + (\mathbf{G}_{i,w} + \mathbf{G}_{i,u}\mathbf{K}_i)\mathbf{w}_i + \mathbf{G}_{i,w}\mathbf{r}_i. \quad (9)$$

Each state $x_{i,k}$ can be extracted with $x_{i,k} = \mathbf{T}_{i,k}\mathbf{x}_i$, where $\mathbf{T}_{i,k} := [\mathbf{0}, \dots, \mathbf{I}, \dots, \mathbf{0}] \in \mathbb{R}^{n_i \times (K+1)n_i}$ is a block matrix whose $(k+1)$ -th block is equal to the identity matrix and all the remaining blocks are equal to the zero matrix. Similarly, we also define $\mathbf{S}_{i,k} \in \mathbb{R}^{m_i \times Km_i}$ such that $u_{i,k} = \mathbf{S}_{i,k}\mathbf{u}_i$.

C. State Mean and Covariance Expressions

Given that each state trajectory \mathbf{x}_i has been approximated as an affine expression of the Gaussian vectors $x_{i,0}$ and \mathbf{w}_i , it follows that \mathbf{x}_i is also Gaussian, i.e., $\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. With similar arguments as in [33, Proposition 1], its mean $\boldsymbol{\mu}_i = \boldsymbol{\eta}_i(\bar{\mathbf{u}}_i)$ and covariance $\boldsymbol{\Sigma}_i = \boldsymbol{\theta}_i(\mathbf{L}_i, \mathbf{K}_i)$ are given by

$$\begin{aligned} \boldsymbol{\eta}_i(\bar{\mathbf{u}}_i) &:= \mathbf{G}_{i,0}\mu_{i,0} + \mathbf{G}_{i,u}\bar{\mathbf{u}}_i + \mathbf{G}_{i,w}\mathbf{r}_i, \\ \boldsymbol{\theta}_i(\mathbf{L}_i, \mathbf{K}_i) &:= (\mathbf{G}_{i,0} + \mathbf{G}_{i,u}\mathbf{L}_i)\boldsymbol{\Sigma}_{i,0}(\mathbf{G}_{i,0} + \mathbf{G}_{i,u}\mathbf{L}_i)^\top \\ &\quad + (\mathbf{G}_{i,w} + \mathbf{G}_{i,u}\mathbf{K}_i)\mathbf{W}_i(\mathbf{G}_{i,w} + \mathbf{G}_{i,u}\mathbf{K}_i)^\top, \end{aligned}$$

where $\mathbf{W}_i = \text{bdiag}(W_i, \dots, W_i) \in \mathbb{R}^{Kn_i \times Kn_i}$. It follows that for each $x_{i,k} \sim \mathcal{N}(\mu_{i,k}, \Sigma_{i,k})$, we have $\mu_{i,k} = \mathbf{T}_{i,k}\boldsymbol{\eta}_i(\bar{\mathbf{u}}_i)$ and $\Sigma_{i,k} = \mathbf{T}_{i,k}\boldsymbol{\theta}_i(\mathbf{L}_i, \mathbf{K}_i)\mathbf{T}_{i,k}^\top$. It is important to note that the mean states depend only on the feed-forward control inputs $\bar{\mathbf{u}}_i$, while the state covariances depend only on the feedback matrices $\mathbf{L}_i, \mathbf{K}_i$.

D. Problem Transformation

The fact that the distributions of the states $x_{i,k}$ can be approximated as multivariate Gaussian ones, is of paramount importance here, since the Wasserstein distance admits a closed-form expression for Gaussian distributions - which does not hold for any arbitrary probability distributions [24]. Therefore, we can rewrite each cost $J_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) = J_i^{\text{dist}}(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) + J_i^{\text{cont}}(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i)$, where J_i^{dist} corresponds to the Wasserstein distances part and J_i^{cont} to the control effort part. Detailed expressions are provided in Appendix VIII-A.

Since the control input $u_{i,k}$ is a Gaussian r.v. as well, the control constraint $u_{i,k} \in \mathcal{U}_i$ cannot be a hard constraint. For this reason, we use the following chance constraints instead,

$$\mathbb{P}(\eta_{i,n}^\top u_{i,k} \leq \gamma_{i,n}) \geq 1 - \beta, \quad n = 1, \dots, N_u, \quad (10)$$

which yields the following convex quadratic constraint through the following proposition.

Proposition 1: The constraint (10) can be equivalently expressed as

$$a_{i,n}(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) \leq 0, \quad (11)$$

with $a_{i,n} = \eta_{i,n}^\top \mathbf{S}_{i,k}\bar{\mathbf{u}}_i - \gamma_{i,n} + \bar{\beta} \|\eta_{i,n}^\top \mathbf{S}_{i,k}[\mathbf{L}_i, \mathbf{K}_i]\boldsymbol{\Psi}_i\|_2$ and $\boldsymbol{\Psi}_i = \text{bdiag}(\boldsymbol{\Sigma}_{i,0}^{1/2}, \mathbf{W}_i)$.

Proof: The proof is omitted as it follows similar steps as the one of [34, Theorem 1]. \blacksquare

These constraints can be written more compactly for all $k \in \llbracket 0, K-1 \rrbracket$ as $a_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) \leq 0$.

Finally, we also wish to express the collision avoidance constraints (3), (4) w.r.t. the new decision variables. Starting from the obstacle avoidance ones, the chance constraint (3) will always be satisfied if the following two constraints hold

$$\|\mathbb{E}[p_{i,k}] - p_o\|_2 \geq d_{i,o} + r_o, \quad i \in \mathcal{V}, \quad o \in \mathcal{O}, \quad (12)$$

$$d_{i,o} \geq \bar{\alpha} \sqrt{\lambda_{\max}(\bar{\boldsymbol{\Sigma}}_{i,k})}, \quad i \in \mathcal{V}, \quad o \in \mathcal{O}, \quad (13)$$

where $\bar{\boldsymbol{\Sigma}}_{i,k} = H_i \boldsymbol{\Sigma}_{i,k} H_i^\top$ is the position covariance, $\bar{\alpha} = \varphi^{-1}(\alpha)$ and $\varphi^{-1}(\cdot)$ is the inverse of the cumulative density function of the normal distribution with unit variance. This is equivalent with enforcing that the $(\mu \pm \bar{\alpha}\sigma)$ confidence ellipsoid of the i -th robot's position is collision free. In addition, since we are steering the covariances $\bar{\boldsymbol{\Sigma}}_{i,k}$ to be as close as possible to the target $\bar{\boldsymbol{\Sigma}}_{i,f} = H_i \boldsymbol{\Sigma}_{i,f} H_i^\top$ through minimizing $J_i^{\text{dist}}(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i)$, then assuming that the actual and target covariances will be close, we replace (13) with

$$d_{i,o} \geq \bar{\alpha} \sqrt{\lambda_{\max}(\bar{\boldsymbol{\Sigma}}_{i,f})}, \quad i \in \mathcal{V}, \quad o \in \mathcal{O}. \quad (14)$$

Therefore, depending on the values of $\bar{\boldsymbol{\Sigma}}_{i,f}$ and $\bar{\alpha}$, we must choose a value for $d_{i,o}$ such that (14) will be satisfied, and then only the constraint (12) remains part of the optimization.

In a similar manner, the inter-robot collision avoidance chance constraints can be substituted with

$$\|\mathbb{E}[p_{i,k}] - \mathbb{E}[p_{j,k}]\|_2 \geq d_{i,j}, \quad i \in \mathcal{V}, \quad j \in \mathcal{V} \setminus \{i\}, \quad (15)$$

$$d_{i,j} \geq \bar{\alpha} \sqrt{\lambda_{\max}(\bar{\boldsymbol{\Sigma}}_{i,f})} + \bar{\alpha} \sqrt{\lambda_{\max}(\bar{\boldsymbol{\Sigma}}_{j,f})}, \quad i \in \mathcal{V}, \quad j \in \mathcal{V} \setminus \{i\}. \quad (16)$$

The constraints (12) and (15) can be written as $b_i(\bar{\mathbf{u}}_i) \leq 0$ and $c_{i,j}(\bar{\mathbf{u}}_i, \bar{\mathbf{u}}_j) \leq 0$, respectively, with the exact expressions provided in Appendix VIII-A. Therefore, we arrive to the following transformation of Problem 1.

Problem 2 (Multi-Robot Distribution Steering Problem II): Find the optimal feed-forward control sequences $\bar{\mathbf{u}}_i^*$ and feedback matrices $\mathbf{L}_i^*, \mathbf{K}_i^*, \forall i \in \mathcal{V}$, such that

$$\{\bar{\mathbf{u}}_i^*, \mathbf{L}_i^*, \mathbf{K}_i^*\}_{i \in \mathcal{V}} = \operatorname{argmin}_{i \in \mathcal{V}} J_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) \quad (17a)$$

$$\text{s.t. } a_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) \leq 0, \quad b_i(\bar{\mathbf{u}}_i) \leq 0, \quad i \in \mathcal{V}, \quad (17b)$$

$$c_{i,j}(\bar{\mathbf{u}}_i, \bar{\mathbf{u}}_j) \leq 0, \quad i \in \mathcal{V}, \quad j \in \mathcal{V} \setminus \{i\}. \quad (17c)$$

IV. DISTRIBUTED APPROACH WITH ADMM

In this section, we present an ADMM-based methodology for solving Problem 2 in a decentralized fashion. In this direction, we first introduce the notions of copy variables and consensus between neighboring robots, so that we can reformulate the problem in an equivalent form that is suitable for ADMM. Subsequently, the derivation of the ADMM updates is illustrated, yielding a distributed soft-constrained CS algorithm in a trajectory optimization format.

A. Decentralized Consensus Approach

Problem 1 cannot be solved directly in a distributed manner due to the inter-robot constraints (17c). To address this issue, we first make the relaxation that each robot $i \in \mathcal{V}$ only considers inter-robot constraints with its closest neighbors given by the set $\mathcal{V}_i \subseteq \mathcal{V}$ - defined such that $i \in \mathcal{V}_i$ as well. Hence, the constraints (17c) can be replaced with $c_{i,j}(\bar{\mathbf{u}}_i, \bar{\mathbf{u}}_j) \leq 0, j \in \mathcal{V}_i \setminus \{i\}, i \in \mathcal{V}$. Subsequently, we introduce for each robot $i \in \mathcal{V}$, the copy variables $\bar{\mathbf{u}}_j^i$ regarding their neighbors $j \in \mathcal{V}_i$. These copy variables can be interpreted as “what is safe for robot j from the perspective of robot i ”. Thus, the augmented feed-forward control input $\bar{\mathbf{u}}_i^{\text{aug}} = [\{\bar{\mathbf{u}}_j^i\}_{j \in \mathcal{V}_i}] \in \mathbb{R}^{K\tilde{m}_i}$ can be defined with $\tilde{m}_i = \sum_{j \in \mathcal{V}_i} m_j$. As a result, the inter-robot constraints can be rewritten from the perspective of the i -th robot as $c_{i,j}(\bar{\mathbf{u}}_i, \bar{\mathbf{u}}_j^i) \leq 0, j \in \mathcal{V}_i \setminus \{i\}, i \in \mathcal{V}$, or more compactly as $c_i^{\text{aug}}(\bar{\mathbf{u}}_i^{\text{aug}}) \leq 0, i \in \mathcal{V}$.

Nevertheless, after the introduction of the copy variables, a requirement for enforcing a consensus between variables that refer to the same robot emerges. To accommodate this, let us define the global feed-forward control variable $\mathbf{g} = [\mathbf{g}_1; \dots; \mathbf{g}_N] \in \mathbb{R}^{Km}$ where $m = \sum_{i \in \mathcal{V}} m_i$. The necessary consensus constraints can be formulated as $\bar{\mathbf{u}}_j^i = \mathbf{g}_j, j \in \mathcal{V}_i \setminus \{i\}, i \in \mathcal{V}$, or written more compactly as $\bar{\mathbf{u}}_i^{\text{aug}} = \tilde{\mathbf{g}}_i, i \in \mathcal{V}$, where $\tilde{\mathbf{g}}_i = [\{\mathbf{g}_j\}_{j \in \mathcal{V}_i}] \in \mathbb{R}^{K\tilde{m}_i}$. Consequently, Problem 2 can be rewritten in the following equivalent form.

Problem 3 (Multi-Robot Distribution Steering Problem III): Find the optimal feed-forward control sequences $\bar{\mathbf{u}}_i^{\text{aug}*}$ and feedback matrices $\mathbf{L}_i^*, \mathbf{K}_i^*, \forall i \in \mathcal{V}$, such that:

$$\{\bar{\mathbf{u}}_i^{\text{aug}*}, \mathbf{L}_i^*, \mathbf{K}_i^*\}_{i \in \mathcal{V}} = \operatorname{argmin}_{i \in \mathcal{V}} J_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) \quad (18a)$$

$$\text{s.t. } a_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) \leq 0, \quad b_i(\bar{\mathbf{u}}_i) \leq 0, \quad (18b)$$

$$c_i^{\text{aug}}(\bar{\mathbf{u}}_i^{\text{aug}}) \leq 0, \quad \bar{\mathbf{u}}_i^{\text{aug}} = \tilde{\mathbf{g}}_i, \quad i \in \mathcal{V}. \quad (18c)$$

Remark 1: Since the inter-robot constraints only involve the feed-forward control inputs $\bar{\mathbf{u}}_i$, then it is sufficient to add copy variables only for the latter - and not for $\mathbf{L}_i, \mathbf{K}_i$ as well. This is an important advantage of the policy parametrization we have selected, as in previous work [22] where a state feedback parametrization was used, there was a requirement for consensus between both the feed-forward control inputs and the feedback gains, even in the case of mean inter-agent state constraints. Therefore, the affine disturbance feedback parametrization allows to significantly reduce the amount of optimization variables that each robot contains.

B. Distributed Covariance Steering with Wasserstein Metric

Subsequently, let us proceed with the derivation of a decentralized ADMM algorithm for solving Problem 3. First, let us rewrite the problem in a more convenient form as

$$\begin{aligned} \min \sum_{i \in \mathcal{V}} J_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) + \mathcal{I}_{a_i}(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) \\ + \mathcal{I}_{b_i}(\bar{\mathbf{u}}_i) + \mathcal{I}_{c_i^{\text{aug}}}(\bar{\mathbf{u}}_i^{\text{aug}}) \end{aligned} \quad (19a)$$

$$\text{s.t. } \bar{\mathbf{u}}_i^{\text{aug}} = \tilde{\mathbf{g}}_i, \quad i \in \mathcal{V}. \quad (19b)$$

The augmented Lagrangian (AL) is given by

$$\begin{aligned} \mathcal{L}_\rho = \sum_{i \in \mathcal{V}} J_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) + \mathcal{I}_{a_i}(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) + \mathcal{I}_{b_i}(\bar{\mathbf{u}}_i) \\ + \mathcal{I}_{c_i^{\text{aug}}}(\bar{\mathbf{u}}_i^{\text{aug}}) + \boldsymbol{\lambda}_i^T (\bar{\mathbf{u}}_i^{\text{aug}} - \tilde{\mathbf{g}}_i) + \frac{\rho}{2} \|\bar{\mathbf{u}}_i^{\text{aug}} - \tilde{\mathbf{g}}_i\|_2^2, \end{aligned}$$

where $\boldsymbol{\lambda}_i$ are the dual variables for the constraints $\bar{\mathbf{u}}_i^{\text{aug}} = \tilde{\mathbf{g}}_i$ and $\rho > 0$ is a penalty parameter.

In the first ADMM block, the AL is minimized w.r.t. $\bar{\mathbf{u}}_i^{\text{aug}}, \mathbf{L}_i$ and \mathbf{K}_i , which yields the following N local subproblems

$$\begin{aligned} \bar{\mathbf{u}}_i^{\text{aug}}, \mathbf{L}_i, \mathbf{K}_i \leftarrow \operatorname{argmin}_{\bar{\mathbf{u}}_i^{\text{aug}}, \mathbf{L}_i, \mathbf{K}_i} J_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) + \boldsymbol{\lambda}_i^T (\bar{\mathbf{u}}_i^{\text{aug}} - \tilde{\mathbf{g}}_i) \\ + \frac{\rho}{2} \|\bar{\mathbf{u}}_i^{\text{aug}} - \tilde{\mathbf{g}}_i\|_2^2 \end{aligned} \quad (20)$$

$$\text{s.t. } a_i(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) \leq 0, \quad b_i(\bar{\mathbf{u}}_i) \leq 0, \quad c_i^{\text{aug}}(\bar{\mathbf{u}}_i^{\text{aug}}) \leq 0.$$

Note that each one of these subproblems can be solved in parallel by each robot i . Nevertheless, these are still non-convex problems due to the cost part J_i^{dist} and the constraints $b_i(\bar{\mathbf{u}}_i) \leq 0$ and $c_i^{\text{aug}}(\bar{\mathbf{u}}_i^{\text{aug}}) \leq 0$. In particular, as the cost J_i^{dist} is a sum of a convex and a concave term, we follow the same approach as in [33] and solve the local problems with an iterative convex-concave procedure [35]. In each such internal iteration, we also linearize the non-convex constraints around the previous mean trajectories as in [36].

Remark 2: A significant advantage of using the squared Wasserstein distance as the measure of difference between actual and target distributions, is that the convexified version of (20) is a convex quadratically constrained quadratic program (QCQP). This is in contrast with other CS approaches that yield semi-definite programs [33], [19], [18] which are more computationally demanding to solve.

In the second ADMM block, the AL is minimized w.r.t. \mathbf{g} , which gives the “per-robot” update rules

$$\mathbf{g}_i \leftarrow \frac{1}{|\mathcal{P}_i|} \sum_{j \in \mathcal{P}_i} \bar{\mathbf{u}}_i^j + \frac{1}{\rho} \boldsymbol{\lambda}_i^j, \quad (21)$$

Algorithm 1 Distributed Model Predictive Covariance Steering (DiMPCS)

```

1: Set:  $N_{\text{total}}, N_{\text{pred}}, N_{\text{comp}}, \rho, \ell_{\text{max}}, \mu_{i,\text{f}}, \Sigma_{i,\text{f}}, R_i, \gamma_i, \forall i \in \mathcal{V}$ 
2:  $\hat{x}_{i,0} \leftarrow$  Measure initial robot states,  $\forall i \in \mathcal{V}$ .
3: Initialize:  $\ell \leftarrow 0, \bar{u}_{i|0}^{\text{aug}} \leftarrow 0, \mathbf{L}_{i|0} \leftarrow 0, \mathbf{K}_{i|0} \leftarrow 0, \mathbf{g}_{i|0} \leftarrow 0, \boldsymbol{\lambda}_{i|0} \leftarrow 0, \boldsymbol{\mu}_{i|0} \leftarrow [\hat{x}_{i,0}; \dots; \hat{x}_{i,0}], \forall i \in \mathcal{V}$ 
4: for  $k = 0, \dots, N_{\text{total}}$  do
5:    $\hat{x}_{i,k} \leftarrow$  Measure current robot states,  $\forall i \in \mathcal{V}$ .
6:   if  $\text{mod}(k, N_{\text{comp}}) == 0$  then
7:      $\mathcal{V}_{i|k}, \mathcal{P}_{i|k} \leftarrow$  Adapt neighborhoods based on current positions  $\hat{p}_{i|k}$ . # In parallel  $\forall i \in \mathcal{V}$ 
8:      $\{A_{i,k'}, B_{i,k'}, r_{i,k'}\}_{k' \in [k, k+N_{\text{pred}}-1]} \leftarrow$  Linearize dynamics using (6a), (6b), around trajectories # In parallel  $\forall i \in \mathcal{V}$ 
9:      $\{\mu_{i,k'}, \bar{u}_{i,k'}\}_{k' \in [k, k+N_{\text{pred}}-1]}$ . # In parallel  $\forall i \in \mathcal{V}$ 
10:     $\mathbf{G}_{i,0|k}, \mathbf{G}_{i,u|k}, \mathbf{G}_{i,w|k} \leftarrow$  Construct using Eq. (9), (10) from [32]. # In parallel  $\forall i \in \mathcal{V}$ 
11:     $\mu_{i,k|k} \leftarrow \hat{x}_{i,k}, \Sigma_{i,k|k} \leftarrow 0, \ell \leftarrow 0$ 
12:    while  $\ell \leq \ell_{\text{max}}$  do
13:       $\bar{u}_{i|k}^{\text{aug}}, \mathbf{L}_{i|k}, \mathbf{K}_{i|k} \leftarrow$  Solve local optimization problem (20). # In parallel  $\forall i \in \mathcal{V}$ 
14:      All robots  $j \in \mathcal{P}_{i|k} \setminus \{i\}$  send  $\bar{u}_{i|k}^j$  to each robot  $i \in \mathcal{V}$ .
15:       $\mathbf{g}_{i|k} \leftarrow$  Update with (21). # In parallel  $\forall i \in \mathcal{V}$ 
16:      All robots  $j \in \mathcal{V}_{i|k} \setminus \{i\}$  send  $\mathbf{g}_{j|k}$  to each robot  $i \in \mathcal{V}$ .
17:       $\boldsymbol{\lambda}_{i|k} \leftarrow$  Update with (22). # In parallel  $\forall i \in \mathcal{V}$ 
18:       $\ell \leftarrow \ell + 1$ 
19:     $\kappa \leftarrow k$ 
20:     $u_{i,k} \leftarrow$  Compute with (23) and apply control decision. # In parallel  $\forall i \in \mathcal{V}$ 

```

where $\mathcal{P}_i = \{j \in \mathcal{V} : i \in \mathcal{V}_j\}$ defines the set that contains all robots $j \in \mathcal{V}$ that have i as a neighbor, and $\boldsymbol{\lambda}_i^j$ is the part of the dual variable $\boldsymbol{\lambda}_i$ that corresponds to the constraint $\bar{u}_{i|k}^j = \mathbf{g}_j$. Finally, the dual variables are updated as follows

$$\boldsymbol{\lambda}_i \leftarrow \boldsymbol{\lambda}_i + \rho(\bar{u}_{i|k}^{\text{aug}} - \tilde{\mathbf{g}}_i), \quad (22)$$

by all $i \in \mathcal{V}$. The updates (20), (21) and (22) are repeated in the presented order until we reach to ℓ_{max} iterations.

V. DISTRIBUTED MODEL PREDICTIVE COVARIANCE STEERING

This section presents Distributed Model Predictive Covariance Steering (DiMPCS) which uses the method proposed in Section IV at its core, by extending it in a receding horizon fashion. The full algorithm is presented in Algorithm 1.

Let us denote with N_{total} and N_{pred} , the total and prediction time horizons, respectively. With $N_{\text{comp}} (\leq N_{\text{pred}})$, we set how often a new MPC computation is performed. After setting all parameters (Line 1) and measuring the initial states $\hat{x}_{i,0}$ (Line 2), we initialize all decision variables with zeros, and the mean state trajectories with $\boldsymbol{\mu}_{i|0} \leftarrow [\hat{x}_{i,0}; \dots; \hat{x}_{i,0}]$ (Line 3). With the notation $z_{\cdot|k}$ we refer to any quantity z that is computed at time k .

Then, the control procedure starts for $k = 0, \dots, N_{\text{total}}$. After measuring the current state if $k > 0$ (Line 5), a new MPC computation starts if $\text{mod}(k, N_{\text{comp}}) = 0$. In this case, the neighborhood sets of all robots $i \in \mathcal{V}$ are first found, by identifying the ones that are in close distance, based on their current positions (Line 7). Subsequently, the dynamics linearization (Line 8) and the construction of the matrices $\mathbf{G}_{i,0|k}, \mathbf{G}_{i,u|k}, \mathbf{G}_{i,w|k}$ (Line 9) take place. The mean $\mu_{i,k|k}$ is always initialized being equal with $\hat{x}_{i,k}$, while the initial covariance is set to $\Sigma_{i,k|k} = 0$ (Line 10) since in this MPC

format, we have perfect information of the initial state $x_{i,k}$ before the optimization starts.

The execution of the proposed ADMM method of Section IV follows. First, the local decision variables $\bar{u}_{i|k}^{\text{aug}}, \mathbf{L}_{i|k}, \mathbf{K}_{i|k}$ of each robot are obtained (Line 12) through solving the local CS problems (20) as explained in Section IV-B. Afterwards, each robot i receives the copy variables $\bar{u}_{i|k}^j$ from all $j \in \mathcal{P}_{i|k} \setminus \{i\}$ (Line 13), so that it can compute $\bar{u}_{i|k}$ (Line 14) with (21). Subsequently, each robot i receives the variables $\mathbf{g}_{j|k}$ from all $j \in \mathcal{V}_{i|k} \setminus \{i\}$ (Line 15), so that $\tilde{\mathbf{g}}_{i|k}$ is constructed and the dual updates (22) take place (Line 16). This iterative ADMM procedure is terminated after ℓ_{max} iterations. Finally, the control input of each robot is computed (Line 19) through

$$u_{i,k|\kappa} = \bar{u}_{i,k|\kappa} + L_{i,k|\kappa}(\hat{x}_{i,\kappa} - \mu_{i,\kappa|\kappa}) + \sum_{l=0}^{k-1} K_{i,(k-1,l)|\kappa} w_{i,l} \quad (23)$$

where κ is the last time k that an MPC cycle took place. Note that in the special case where we assign $\hat{x}_{i,\kappa} = \mu_{i,\kappa|\kappa}$, the second term in the RHS of (23) becomes zero but this can change if the assumption of $\Sigma_{i,k|k} = 0$ is relaxed.

Remark 3: All computations in DiMPCS (Lines 7-9,12,14,16,19) can be performed in parallel by every robot $i \in \mathcal{V}$. In addition, all necessary communication steps (Lines 13,15) take place locally between neighboring robots. Therefore, the proposed algorithm is *fully distributed* in terms of computational and communication requirements.

Remark 4: The neighborhood adaptation, during the beginning of every MPC cycle, is an important advantage compared to the trajectory optimization approach followed in [22], as it allows for using smaller adjustable neighborhoods.

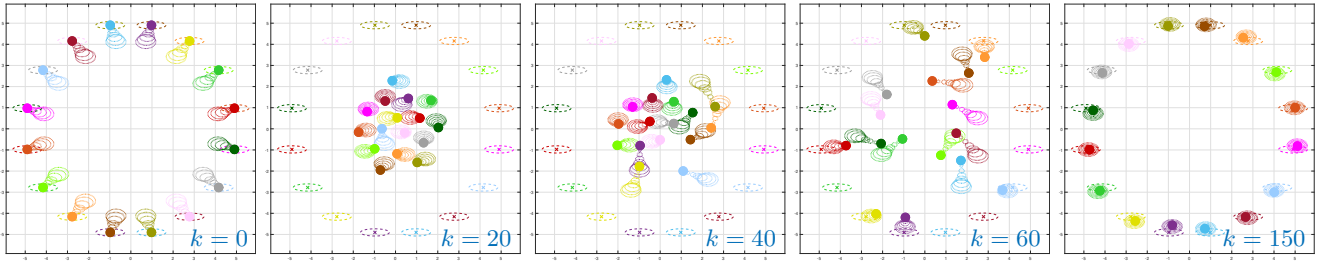


Fig. 1: A task with 16 robots that must reach a target distribution at a diametrically opposite location while avoiding collisions. Each snapshot shows their positions and the $(\mu \pm 3\sigma)$ regions of planned distribution trajectories. The target distributions are shown as dashed ellipses with “x” at the center.

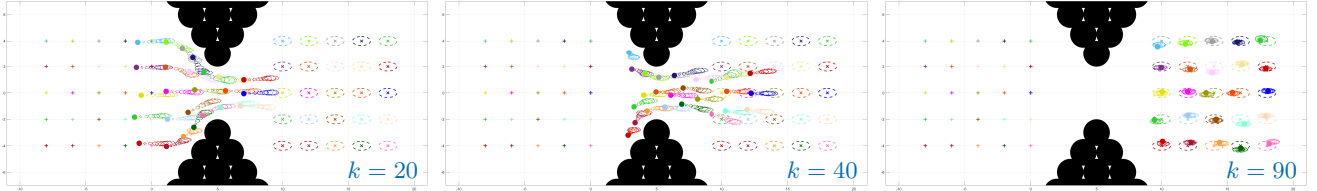


Fig. 2: A task with 25 robots required to pass through a narrow bottleneck before reaching their targets.

VI. SIMULATION EXPERIMENTS

This section presents simulation experiments that demonstrate the effectiveness and scalability of DiMPCS. In the main paper, we provide snapshots of the tasks, while we refer the reader to the supplementary video for a full demonstration. All robots have unicycle dynamics with states $x_{i,k} = [x_{i,k}; y_{i,k}; \theta_{i,k}; v_{i,k}] \in \mathbb{R}^4$ and inputs $u_{i,k} = [a_{i,k}; \omega_{i,k}] \in \mathbb{R}^2$, where $(x_{i,k}, y_{i,k})$, $\theta_{i,k}$, $v_{i,k}$, $\omega_{i,k}$, $a_{i,k}$ are their 2D position coordinates, angles, linear and angular velocities and linear accelerations, respectively. In all experiments, we use $N_{\text{pred}} = 7$ and $N_{\text{comp}} = 2$. The discretization time step is $dt = 0.05$. The process noise covariance is $W_i = \text{diag}(0.02, 0.02, \pi/180, 0.2)$. We set the control cost matrix $R_i = \text{diag}(10^{-2}, 10^{-2})$. We also enforce control limits $a_{\text{max}} = -a_{\text{min}} = 5\text{m/s}^2$ and $\omega_{\text{max}} = -\omega_{\text{min}} = 4\text{rad/s}$ through the chance constraints (10) with $\beta = 0.997$. For the collision avoidance constraints, we select $d_{i,o} = 0.75\text{m}$, $d_{i,j} = 1.5\text{m}$ and $\bar{\alpha} = \varphi^{-1}(0.997) = 3$. Finally, we set $\rho = 10^{-2}$, $\ell_{\text{max}} = 30$ and $|\mathcal{V}_i| = 6$ for all tasks.

A. Small-Scale Tasks

In the first task, 16 robots need to reach to their target distributions at the diametrically opposite locations, while avoiding collisions with each other. In Fig. 1, the performance of DiMPCS is demonstrated through five different snapshots that show the positions and planned distribution trajectories of the robots. All robots are able to successfully reach to their targets and avoid collisions throughout the task. In the next scenario (Fig. 2), 25 robots must reach to their targets while passing through a narrow “bottleneck” and avoiding collisions. Despite the difficulty of this task, all robots are again safely navigated to their targets.

B. Large-Scale Task

Subsequently, we highlight the scalability of DiMPCS to large-scale multi-robot problems. In particular, we consider a problem with 256 robots that need to move from one 16×16 square grid to another one while avoiding collisions with each other and the obstacles in between. Figure 3 shows a snapshot of the task, while the full task is available in the supplementary video. All robots are successfully driven to their targets while maintaining their safe operation.

C. Comparison with Other Stochastic MPC Approaches

Next, we illustrate the computational and performance advantages of DiMPCS against related SMPC approaches. All comparisons are on the same task as in Fig. 3. Initially, we compare against an equivalent centralized approach for solving Problem 2. We observe that as the number of robots grows (Table I), DiMPCS remains scalable, while the increasing dimensionality of the multi-robot problem makes the centralized approach computationally intractable. We should also highlight that hard-constrained CS approaches that lead to SDPs are excluded from this comparison, as their computational demands are much higher, in addition to their need for a distribution path before performing MPC.

Furthermore, we provide a performance comparison of DiMPCS against standard SMPC methods in terms of collision percentages and control efforts (Table II). Each algorithm is tested for 5 trials. First, we compare against solving the MPC problems with LQG control instead. While the latter method also yields safe solutions, it reduces the variance of the states more aggressively which requires excessive control effort. We also compare with standard SMPC approaches which only optimize for the feed-forward controls, while selecting a fixed stabilizing gain for the initial linearized dynamics [9]. Although such approaches involve

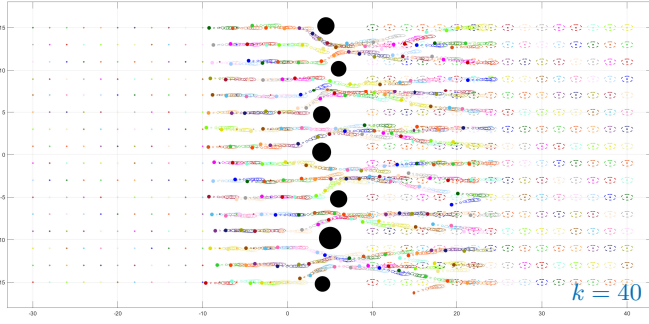


Fig. 3: A large-scale task with 256 robots.

Method	$N = 4$	$N = 16$	$N = 64$	$N = 256$
DiMPCS (Proposed)	321ms	534ms	1.02s	2.05s
Centralized MPCs	1.54s	32s	9m 49s	1h 22s

TABLE I: Computational times per MPC cycle of DiMPCS and an equivalent centralized approach.

Method	Collisions %	Control effort
DiMPCS (Proposed)	0 %	180.55
SMPC with LQG	0 %	263.83
SMPC with fixed feedback (I)	5.47 %	78.49
SMPC with fixed feedback (II)	0.33 %	244.73

TABLE II: Performance comparison between DiMPCS and other SMPC approaches.

less decision variables, the fact that the covariance is not actively steered leads to either unsafe solutions (Case I) or relatively safe solutions that require significant control effort (Case II). Therefore, the fact that DiMPCS actively steers the state distributions to match target distributions, while computing a sequence of feedback gains, provides the most advantageous combination of safety and control effort.

VII. HARDWARE EXPERIMENTS

Finally, we validate the applicability of the proposed distributed algorithm on a multi-robot system in the Robotarium platform [37] at Georgia Tech. For the dynamics of the robots, the reader is referred to [37]. In addition to collision and obstacle avoidance constraints with $d_{i,o} = 0.1\text{m}$, $d_{i,j} = 0.2\text{m}$, all robots are subject to the following control constraints, $-\mathbf{b}_{\max} \leq \mathbf{G}\mathbf{u}_i \leq \mathbf{b}_{\max}$, with $\mathbf{G} = (1/2R)[2, L; 2, -L]$ and $\mathbf{b}_{\max} = [v_{\text{wheel}}^{\max}; v_{\text{wheel}}^{\max}]$, where $R = 0.016\text{m}$ is the wheel radius, $L = 0.11\text{m}$ is the axle length and $v_{\text{wheel}}^{\max} = 12.5\text{ rad/s}$ is the maximum wheel speed. The control constraints are handled as chance constraints of the form (10) with $\beta = 0.997$. The timestep is $dt = 330\text{ms}$, while we set $N_{\text{pred}} = 7$ and $N_{\text{total}} = 100$.

We first apply the proposed algorithm on a task where three robots are required to reach to their target distributions while avoiding the obstacles in the middle of the field. As illustrated in Fig. 4, the robots are able to successfully complete the task while avoiding collisions. Next, we demonstrate in Fig. 5, a task where five robots must reach to the diametrically opposite positions while avoiding collisions with the rest of the robots. Again, all robots are safely driven to their destinations without colliding with each other.

VIII. CONCLUSION

In this work, we propose DiMPCS, a novel distributed SMPC algorithm for multi-robot control under uncertainty. Our approach combines CS theory using the Wasserstein distance and ADMM into an MPC scheme, to ensure safety while achieving scalability and parallelization. Numerical simulations verify the effectiveness of DiMPCS in various multi-robot navigation problems compared to other approaches. Finally, the applicability of the method on real robotic systems is verified through hardware experiments.

APPENDIX

A. Cost and Constraints Expressions

Following a similar derivation as in [33, Propositions 4,5], the terms J_i^{dist} and J_i^{cont} can be written equivalently as

$$\begin{aligned}
 J_i^{\text{dist}}(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) &= \sum_{k=1}^K \|\mathbf{T}_{i,k} \boldsymbol{\eta}_i(\bar{\mathbf{u}}_i) - \mu_{i,f}\|_2^2 \\
 &+ \|\boldsymbol{\zeta}_{i,k}(\mathbf{L}_i, \mathbf{K}_i)\|_F^2 + \text{tr}(\boldsymbol{\Sigma}_{i,f}) - 2\|\sqrt{\boldsymbol{\Sigma}_{i,f}} \boldsymbol{\zeta}_{i,k}(\mathbf{L}_i, \mathbf{K}_i)\|_*, \\
 J_i^{\text{cont}}(\bar{\mathbf{u}}_i, \mathbf{L}_i, \mathbf{K}_i) &= \bar{\mathbf{u}}_i^T \mathbf{R}_i \bar{\mathbf{u}}_i + \text{tr}(\mathbf{R}_i \mathbf{L}_i \boldsymbol{\Sigma}_{i,0} \mathbf{L}_i^T) \\
 &+ \text{tr}(\mathbf{R}_i \mathbf{K}_i \mathbf{W}_i \mathbf{K}_i^T),
 \end{aligned}$$

where $\mathbf{R}_i = \text{bdiag}(R_i, \dots, R_i) \in \mathbb{R}^{K m_i \times K m_i}$ and

$$\boldsymbol{\zeta}_{i,k}(\mathbf{L}_i, \mathbf{K}_i) = \mathbf{T}_{i,k} [\mathbf{G}_{i,0} + \mathbf{G}_{i,u} \mathbf{L}_i \quad \mathbf{G}_{i,w} + \mathbf{G}_{i,u} \mathbf{K}_i].$$

Futhermore, the constraints (12), (15) can be written as

$$\begin{aligned}
 b_i(\bar{\mathbf{u}}_i) &= d_{i,o} + r_o - \|H_i \mathbf{T}_{i,k} \boldsymbol{\eta}_i(\bar{\mathbf{u}}_i) - p_o\|_2 \leq 0, \\
 c_{i,j}(\bar{\mathbf{u}}_i, \bar{\mathbf{u}}_j) &= \|H_i \mathbf{T}_{i,k} \boldsymbol{\eta}_i(\bar{\mathbf{u}}_i) - H_j \mathbf{T}_{j,k} \boldsymbol{\eta}_j(\bar{\mathbf{u}}_j)\|_2 \leq 0.
 \end{aligned}$$

ACKNOWLEDGMENT

This work was supported in part by NSF under grants 1936079 and 1937957, and by the ARO Award #W911NF2010151. Augustinos Saravanos acknowledges financial support by the A. Onassis Foundation Scholarship.

REFERENCES

- [1] V. S. Chipade and D. Panagou, "Multiagent planning and control for swarm herding in 2-d obstacle environments under bounded inputs," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1956–1972, 2021.
- [2] I. Maza, K. Kondak, M. Bernard, and A. Ollero, "Multi-UAV cooperation and control for load transportation and deployment," in *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*. Springer, 2009, pp. 417–449.
- [3] Y. Kantaros, M. Malencia, V. Kumar, and G. J. Pappas, "Reactive temporal logic planning for multiple robots in unknown environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 11 479–11 485.
- [4] S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, "A predictive control approach for cooperative transportation by multiple underwater vehicle manipulator systems," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 917–930, 2022.
- [5] F. Rey, Z. Pan, A. Hauswirth, and J. Lygeros, "Fully decentralized ADMM for coordination and collision avoidance," in *2018 European Control Conference (ECC)*, 2018, pp. 825–830.
- [6] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed MPC for formation of multi-agent systems with collision avoidance and obstacle avoidance," *Journal of the Franklin Institute*, vol. 354, no. 4, pp. 2068–2085, 2017.
- [7] X. Zhang, J. Ma, Z. Cheng, S. Huang, C. W. de Silva, and T. H. Lee, "Improved hierarchical ADMM for nonconvex cooperative distributed model predictive control," *arXiv preprint arXiv:2011.00463*, 2020.
- [8] S. Yan, P. J. Goulart, and M. Cannon, "Stochastic MPC with dynamic feedback gain selection and discounted probabilistic constraints," *IEEE Transactions on Automatic Control*, 2021.

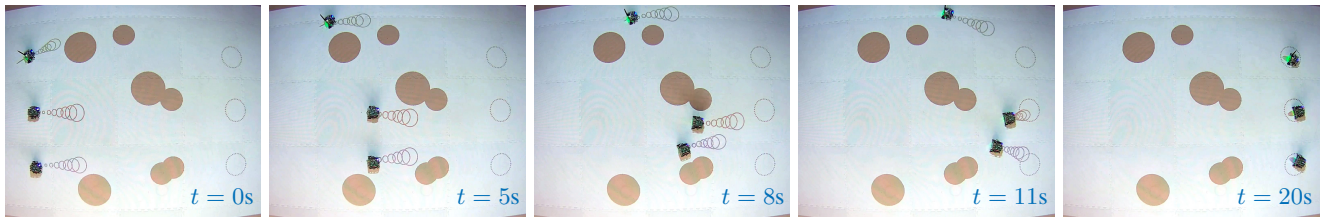


Fig. 4: Hardware experiment with three robots that are required to reach their targets while avoiding collisions.

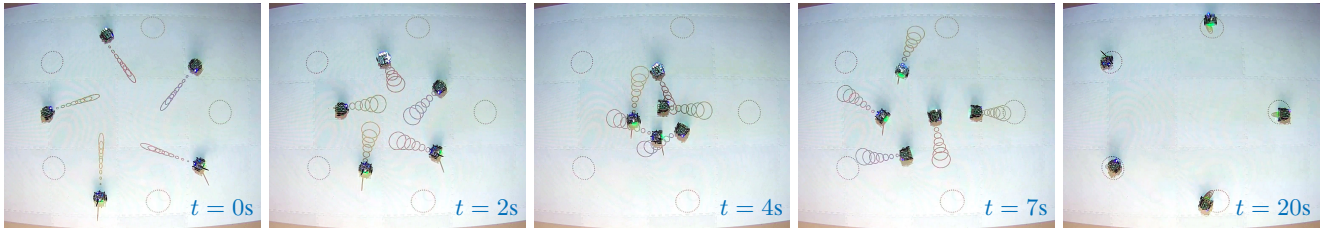


Fig. 5: Hardware experiment with five robots that are required to reach the diametrically opposite positions without collisions.

- [9] E. Arcari, A. Iannelli, A. Carron, and M. N. Zeilinger, “Stochastic MPC with robustness to bounded parametric uncertainty,” *arXiv preprint arXiv:2205.10275*, 2022.
- [10] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, “The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations,” *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.
- [11] F. Oldewurtel, C. N. Jones, and M. Morari, “A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback,” in *2008 47th IEEE conference on decision and control*. IEEE, 2008, pp. 4731–4736.
- [12] K. Okamoto and P. Tsiotras, “Stochastic model predictive control for constrained linear systems using optimal covariance steering,” *arXiv preprint arXiv:1905.13296*, 2019.
- [13] I. M. Balci, E. Bakolas, B. Vlahov, and E. A. Theodorou, “Constrained covariance steering based tube-MPPI,” in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 4197–4202.
- [14] J. Yin, Z. Zhang, E. Theodorou, and P. Tsiotras, “Trajectory distribution control for model predictive path integral control using covariance steering,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1478–1484.
- [15] A. Hotz and R. E. Skelton, “Covariance control theory,” *Int. J. Control.*, vol. 46, no. 1, pp. 13–32, 1987.
- [16] J.-H. Xu and R. E. Skelton, “An improved covariance assignment theory for discrete systems,” *IEEE Trans. Automat. Contr.*, vol. 37, no. 10, pp. 1588–1591, 1992.
- [17] Y. Chen, T. T. Georgiou, and M. Pavon, “Optimal steering of a linear stochastic system to a final probability distribution, part i,” *IEEE Trans. Automat. Contr.*, vol. 61, no. 5, pp. 1158–1169, 2015.
- [18] G. Kotsalis, G. Lan, and A. S. Nemirovski, “Convex optimization for finite-horizon robust covariance control of linear stochastic systems,” *SIAM Journal on Control and Optimization*, vol. 59, no. 1, pp. 296–319, 2021. [Online]. Available: <https://doi.org/10.1137/20M135090X>
- [19] I. M. Balci and E. Bakolas, “Exact SDP formulation for discrete-time covariance steering with Wasserstein terminal cost,” *arXiv preprint arXiv:2205.10740*, 2022.
- [20] F. Liu, G. Rapakoulias, and P. Tsiotras, “Optimal covariance steering for discrete-time linear stochastic systems,” *arXiv preprint arXiv:2211.00618*, 2022.
- [21] K. Okamoto and P. Tsiotras, “Optimal stochastic vehicle path planning using covariance steering,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2276–2281, 2019.
- [22] A. D. Saravanos, A. Tsolovikos, E. Bakolas, and E. Theodorou, “Distributed Covariance Steering with Consensus ADMM for Stochastic Multi-Agent Systems,” in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.
- [23] A. D. Saravanos, Y. Li, and E. Theodorou, “Distributed Hierarchical Distribution Control for Very-Large-Scale Clustered Multi-Agent Systems,” in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.
- [24] C. R. Givens and R. M. Shortt, “A class of Wasserstein metrics for probability distributions,” *Michigan Mathematical Journal*, vol. 31, no. 2, pp. 231–240, 1984.
- [25] T. Halsted, O. Shorinwa, J. Yu, and M. Schwager, “A survey of distributed optimization methods for multi-robot systems,” *arXiv preprint arXiv:2103.12840*, 2021.
- [26] Z. Cheng, J. Ma, X. Zhang, C. W. de Silva, and T. H. Lee, “ADMM-based parallel optimization for multi-agent collision-free model predictive control,” *arXiv preprint arXiv:2101.09894*, 2021.
- [27] A. D. Saravanos, Y. Aoyama, H. Zhu, and E. A. Theodorou, “Distributed differential dynamic programming architectures for large-scale multi-agent control,” *arXiv preprint arXiv:2207.13255*, 2022.
- [28] M. A. Pereira, A. D. Saravanos, O. So, and E. A. Theodorou, “Decentralized Safe Multi-agent Stochastic Optimal Control using Deep FBSDEs and ADMM,” in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.
- [29] J. Ridderhof, K. Okamoto, and P. Tsiotras, “Nonlinear uncertainty control with iterative covariance steering,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 3484–3490.
- [30] E. Bakolas and A. Tsolovikos, “Greedy finite-horizon covariance steering for discrete-time stochastic nonlinear systems based on the unscented transform,” in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3595–3600.
- [31] Z. Yi, Z. Cao, E. Theodorou, and Y. Chen, “Nonlinear covariance control via differential dynamic programming,” in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3571–3576.
- [32] I. M. Balci and E. Bakolas, “Covariance steering of discrete-time stochastic linear systems based on wasserstein distance terminal cost,” *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2000–2005, 2021.
- [33] —, “Covariance control of discrete-time gaussian linear systems using affine disturbance feedback control policies,” in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 2324–2329.
- [34] K. Okamoto, M. Goldshtein, and P. Tsiotras, “Optimal covariance control for stochastic systems under chance constraints,” *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 266–271, 2018.
- [35] A. L. Yuille and A. Rangarajan, “The concave-convex procedure,” *Neural computation*, vol. 15, no. 4, pp. 915–936, 2003.
- [36] F. Augugliaro, A. P. Schoellig, and R. D’Andrea, “Generation of collision-free trajectories for a quadcopter fleet: A sequential convex programming approach,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1917–1922.
- [37] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, “The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems,” *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.