Distributed Hierarchical Distribution Control for Very-Large-Scale Clustered Multi-Agent Systems

Augustinos D. Saravanos, Yihui Li and Evangelos A. Theodorou
Georgia Institute of Technology, GA, USA
Email: asaravanos@gatech.edu

Abstract—As the scale and complexity of multi-agent robotic systems are subject to a continuous increase, this paper considers a class of systems labeled as Very-Large-Scale Multi-Agent Systems (VLMAS) with dimensionality that can scale up to the order of millions of agents. In particular, we consider the problem of steering the state distributions of all agents of a VLMAS to prescribed target distributions while satisfying probabilistic safety guarantees. Based on the key assumption that such systems often admit a multi-level hierarchical clustered structure - where the agents are organized into cliques of different levels - we associate the control of such cliques with the control of distributions, and introduce the Distributed Hierarchical Distribution Control (DHDC) framework. The proposed approach consists of two sub-frameworks. The first one, Distributed Hierarchical Distribution Estimation (DHDE), is a bottom-up hierarchical decentralized algorithm which links the initial and target configurations of the cliques of all levels with suitable Gaussian distributions. The second part, Distributed Hierarchical Distribution Steering (DHDS), is a top-down hierarchical distributed method that steers the distributions of all cliques and agents from the initial to the targets ones assigned by DHDE. Simulation results that scale up to two million agents demonstrate the effectiveness and scalability of the proposed framework. The increased computational efficiency and safety performance of DHDC against related methods is also illustrated. The results of this work indicate the importance of hierarchical distribution control approaches towards achieving safe and scalable solutions for the control of VLMAS.

I. INTRODUCTION

Multi-agent systems in robotics are experiencing an increasing popularity with several significant applications such as multi-robot coordination [11], navigating fleets of vehicles [28], guiding teams of UAVs [36] and swarm robotics [6], to name only a few. As the scale and complexity of such systems are continuously growing, a great requirement has emerged for developing algorithmic frameworks that benefit from a distributed structure, high computational efficiency, low communication requirements, and therefore, scalability. In addition, as uncertainty is an integral component of multi-agent systems, associating such methods with safety guarantees remains of paramount importance.

Most of the existing literature in multi-robot control, has considered systems that range from a handful of units to hundreds or thousands of agents. Some notable approaches can be found in the fields of optimal control [24, 25, 32, 38], path planning [8, 13, 26], swarm robotics [6, 21, 22, 30] and multi-agent reinforcement learning [12, 14, 16, 37]. Nevertheless, empirical demonstrations show that the scalability of most methods from the previous classes is practically limited in
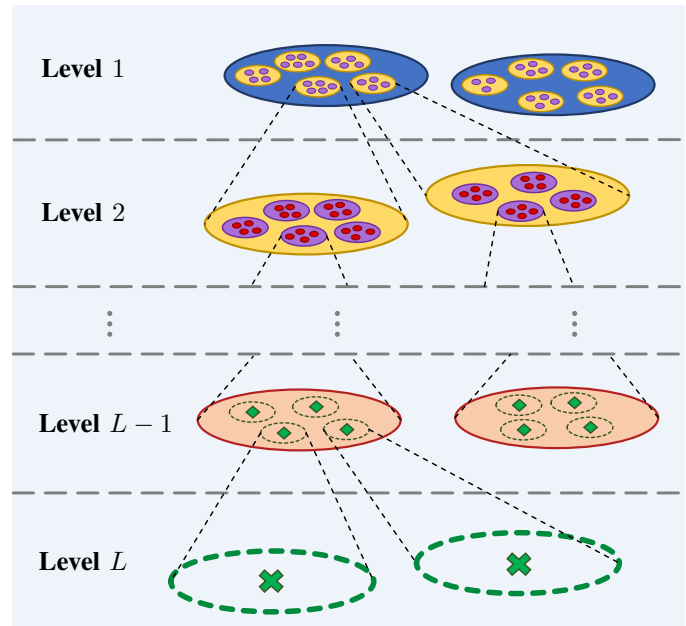

Fig. 1: A VLMAS with a hierarchical clustered structure of L levels. Level L corresponds to individual robots and their distributions (green). All robots are organized into level-$(L-1)$ cliques (orange) which are then also organized into level-$(L-2)$ cliques and so on, up to level-1 cliques (blue).

the order of a few thousands of robots.

This paper considers a class of systems, labeled as Very-Large-Scale Multi-Agent Systems (VLMAS), that can scale up to the order of millions of robots. To also account for potential uncertainties in VLMAS, all agents are modeled with stochastic dynamics. To our best knowledge, the literature for addressing the control of such systems in a safe, distributed and scalable manner while operating under uncertainty is very scarce. The key insight of this work is that VLMAS often admit a hierarchical clustered structure (Fig. 1) where robots are arranged into cliques, which are then organized into greater cliques etc. Such a structure is highly suitable for the proposed direction of hierarchical distribution control.

Multi-robot control approaches that exploit such hierarchies appear to be quite few in the literature. A multi-robot navigation method that used hierarchical clustering to identify the formation of groups of robots and guide them to their targets was proposed in [2]. Furthermore, a distributed algorithm for

the coordination of clusters of robots was recently presented in [15]. While these works have hinted towards the potential of distributed hierarchical control in robotics, their approaches were still only applicable to small-scale multi-robot teams and unrelated to the control of distributions.

As safety requirements in robotics and control are of great significance, covariance steering (CS) theory has recently emerged as a promising approach for guiding the state distribution of a system to prescribed targets while providing probabilistic safety guarantees [3, 4, 10, 20]. Successful robotics applications can be found in trajectory optimization [5, 34], path planning [23], flight control [7, 17, 27], multi-robot systems [31, 33] and robotic manipulation [18], to name a few. While the main barrier for applying CS methods for multi-agent stochastic control was due to their significant computational requirements, recent distributed optimization based approaches [31, 33] have shown that CS is a viable option for multi-agent systems. Nevertheless, the scalability of the aforementioned methods appears to be practically limited to systems with tens or hundreds of agents. In addition, to our best knowledge, combining CS with the control of the distributions of clusters of agents has not been considered yet in the literature.

In this paper, we aspire to surpass the appearing limitations of current CS approaches, by exploiting the fact that VLMAS systems can be subject to a hierarchical clustered structure. In such a multi-level hierarchical setup, the agents are organized into cliques, which are then organized into greater cliques, and so on. Our key insight lies in the fact that CS theory can also be utilized for the control of such cliques in addition to the control of individual agents. Based on this fact, we introduce a novel distributed method for the control of VLMAS, named Distributed Hierarchical Distribution Control (DHDC). The DHDC framework consists of two separate sub-frameworks. The first one, Distributed Hierarchical Distribution Estimation (DHDE), associates all cliques with suitable Gaussian distributions that satisfy the hierarchical structure, while the second one, Distributed Hierarchical Distribution Steering (DHDS), utilizes CS to drive the distributions of all cliques and agents towards their assigned targets. The specific contributions of this work can be listed as follows:

1) We illustrate how covariance steering theory can be fused with the control of clusters of agents that are linked through a hierarchical structure.
2) We propose DHDE, a bottom-up hierarchical distributed approach for estimating the optimal random distributions to be associated with the cliques of each level of the hierarchy.
3) We propose DHDS, a top-down hierarchical distributed approach for steering the distributions of all cliques and agents to their targets, by exploiting the information acquired with DHDE.
4) We demonstrate the effectiveness and scalability of the proposed approaches through simulation experiments on systems with up to two million agents.

To our best knowledge, the proposed approach is one of the few existing methods for the distributed and safe control of stochastic systems in the VLMAS scale. In addition, DHDC greatly outperforms the scalability of all current CS approaches for multi-agent control, and therefore, paves the way for the application of CS theory to robotic systems of a much larger scale. Furthermore, this work highlights the potential of hierarchical distributed optimization methods that utilize distribution characteristics for the control of large-scale multi-agent systems.

II. PROBLEM STATEMENT

A. Notation

The set of $n \times n$ symmetric positive semi-definite (definite) matrices is denoted with $\mathbb{S}_n^+$ ($\mathbb{S}_n^{++}$). Given two matrices $A, B \in \mathbb{R}^{n \times n}$, the matrix inequality $A \succeq B$ refers to $A - B \in \mathbb{S}_n^+$. Given a random variable (r.v.) $x \in \mathbb{R}^n$, its expectation and covariance are given by $\mathbb{E}[x] \in \mathbb{R}^n$ and $\mathbb{COV}[x] \in \mathbb{S}_n^+$, respectively. If a r.v. $x \in \mathbb{R}^n$ is such that $x \sim \mathcal{N}(\mu, \Sigma)$, then x is subject to a multivariate Gaussian distribution with mean $\mu = \mathbb{E}[x]$ and covariance $\Sigma = \mathbb{COV}[x]$. In addition, $x \in \mathcal{E}_\theta[\mu, \Sigma]$ implies that $x \in \mathbb{R}^n$ lies within the θ-probability confidence ellipsoid of $\mathcal{N}(\mu, \Sigma)$, i.e., $\mathcal{E}_\theta[\mu, \Sigma] : (x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu) \leq \alpha$, where $\alpha = f_{\chi^2, \nu}^{-1}(\theta)$ and $f_{\chi^2, \nu}^{-1}(\cdot)$ is the inverse cumulative distribution function of the chi-square distribution with ν degrees of freedom. The Kullback–Leibler (KL) divergence of a probability distribution $\mathcal{P}$ from another distribution $\mathcal{Q}$ is denoted with $\mathbb{D}_{\mathrm{KL}}(\mathcal{P}\|\mathcal{Q})$. Furthermore, the cardinality of a set $\mathcal{A}$ is given by $|\mathcal{A}|$. Finally, $[\![a,b]\!]$ denotes the integer set $[a, b] \cap \mathbb{Z}$ for any $a, b \in \mathbb{R}$ with $a \leq b$.

B. Problem Formulation

Let us consider a VLMAS given by the set $\mathcal{R} = \{1, \ldots, M\}$, where M is the total number of agents. Each agent $i \in \mathcal{R}$ is subject to the following homogeneous discrete-time, stochastic, linear dynamics

$$x_{i,k+1} = Ax_{i,k} + Bu_{i,k} + w_{i,k}, \tag{1}$$

where $x_{i,k} \in \mathbb{R}^{n_x}$ and $u_{i,k} \in \mathbb{R}^{n_u}$ are the state and control of the i-th agent at time k, $w_{i,k} \in \mathbb{R}^{n_x}$ is process noise such that $w_{i,k} \sim \mathcal{N}(0, W)$ with $W \in \mathbb{S}_{n_x}^+$, and $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$. If we denote the time horizon with N, i.e., $k \in [\![0, N]\!]$, then the full state, control and noise sequences of agent i are given by $x_i = [x_{i,0}; \ldots; x_{i,N}]$, $u_i = [u_{i,0}; \ldots; u_{i,N-1}]$ and $w_i = [w_{i,0}; \ldots; w_{i,N-1}]$, respectively. All initial states $x_{i,0}$ are subject to $x_{i,0} \sim \mathcal{N}(\mu_{i,0}, \Sigma_{i,0})$, where $\mu_{i,0} \in \mathbb{R}^{n_x}$ and $\Sigma_{i,0} \in \mathbb{S}_{n_x}^{++}$ are known.

Let us now introduce the VLMAS *distribution control problem*. The main objective is to steer the distributions of the terminal states $x_{i,N}$ of all agents to prescribed target distributions $\mathcal{N}(\mu_{i,\mathrm{f}}, \Sigma_{i,\mathrm{f}})$ with $\mu_{i,\mathrm{f}} \in \mathbb{R}^{n_x}$ and $\Sigma_{i,\mathrm{f}} \in \mathbb{S}_{n_x}^{++}$. This can be achieved by enforcing the following constraints

$$\mathbb{E}[x_{i,N}] = \mu_{i,\mathrm{f}}, \tag{2}$$

$$\mathbb{COV}[x_{i,N}] \preceq \Sigma_{i,\mathrm{f}}, \tag{3}$$

for every agent $i \in \mathcal{R}$. Furthermore, the position of each agent in space is given by $p_{i,k} = H x_{i,k} \in \mathbb{R}^{n_p}$, with $H \in \mathbb{R}^{n_p \times n_x}$ defined accordingly. To simplify exposition, in this work, we consider agents that operate on a 2D plane, i.e., $n_p = 2$. Nevertheless, all proposed ideas are readily extendable for 3D multi-robot systems. The following probabilistic inter-agent collision avoidance constraints must also be satisfied between all agents,

$$\mathbb{P}\left(\|p_{i,k} - p_{j,k}\|_2 \geq d_{\text{inter}}\right) \geq \theta, \quad \forall i, j \in \mathcal{R}, \ i \neq j, \quad (4)$$

for all $k \in [\![0, N]\!]$, where d_{inter} is the minimum allowed distance between two agents and $\theta \in (0.5, 1)$. In addition, we assume the existence of a set $\mathcal{O} = \{1, \ldots, O\}$ of circle obstacles and consider the following probabilistic safety constraints

$$\mathbb{P}\left(\|p_{i,k} - p_o\|_2 \geq d_{\text{obs}} + r_o\right) \geq \theta, \quad \forall i \in \mathcal{R}, \ \forall o \in \mathcal{O}, \quad (5)$$

for all $k \in [\![0, N]\!]$, where r_o is the radius of obstacle $o \in \mathcal{O}$ and d_{obs} is the minimum allowed distance between an agent and an obstacle. Finally, all agents also aim to minimize their control effort through the following individual costs

$$J_i(u_i) = \sum_{k=0}^{N-1} \mathbb{E}[u_{i,k}^{\mathrm{T}} R u_{i,k}], \quad i \in \mathcal{R}. \quad (6)$$

where $R \in \mathbb{S}_n^{++}$. Consequently, the VLMAS distribution control problem can be formulated as follows.

Problem 1 (VLMAS Distribution Control Problem). *Find the optimal control input sequences u_i^*, $\forall i \in \mathcal{R}$, such that*

$$\{u_i^*\}_{i \in \mathcal{R}} = \arg\min \sum_{i \in \mathcal{R}} J_i(u_i)$$

$$\text{s.t.} \quad (1), (2), (3), (4), (5).$$

C. Hierarchical Clustered Structure

This work focuses on VLMAS that are subject to a known *hierarchical clustered structure* (Fig. 1). This hierarchy consists of L levels with each level denoted with $\ell \in \{1, \ldots, L\}$. The bottom level $\ell = L$ corresponds to individual agents. All agents are organized into level-$(L-1)$ cliques $\mathcal{C}_i^{L-1} \in \mathcal{R}^{L-1}$, which are then organized into level-$(L-2)$ cliques $\mathcal{C}_i^{L-2} \in \mathcal{R}^{L-2}$, and so on, where $\mathcal{R}^\ell$ denotes the set of all cliques in each level $\ell \in [\![1, L]\!]$. If by convention, we let the level-L cliques $\mathcal{C}_i^L$ correspond to individual agents $i \in \mathcal{R}$, i.e., $\mathcal{R}^L \equiv \mathcal{R}$, then the aforementioned structure can be formally stated as follows.

Assumption 1 (Hierarchical Clustered Structure). *For all levels $\ell \in [\![2, L]\!]$, it holds that for every clique $\mathcal{C}_i^\ell \in \mathcal{R}^\ell$, there exists a clique $\mathcal{C}_a^{\ell-1} \in \mathcal{R}^{\ell-1}$ such that*

$$\mathcal{C}_i^\ell \subseteq \mathcal{C}_a^{\ell-1}. \quad (7)$$

In other words, every clique in level ℓ belongs to a "parent" clique of level $\ell - 1$. Furthermore, for all levels $\ell \in [\![1, L]\!]$, it holds that

$$\mathcal{C}_i^\ell \cap \mathcal{C}_j^\ell = \emptyset, \quad \forall \mathcal{C}_i^\ell, \mathcal{C}_j^\ell \in \mathcal{R}^\ell, \ i \neq j, \quad (8)$$

i.e., the intersection between cliques of the same level is always the empty set.

Furthermore, to lighten the notation, given a clique $\mathcal{C}_i^\ell$, we consider the statements $\mathcal{C}_i^\ell \in \mathcal{A}$ and $i \in \mathcal{A}$ to be equivalent for any arbitrary set $\mathcal{A}$. A direct consequence of Assumption 1 is the following.

Corollary 1. *For all levels $\ell \in [\![2, L]\!]$, every clique $\mathcal{C}_i^\ell$ only has one "parent" clique $\mathcal{C}_a^{\ell-1}$.*

Subsequently, we introduce the notion of *neighbor cliques*. Given a clique $\mathcal{C}_i^\ell$, the set of its neighbor cliques is defined as $\mathfrak{n}[\mathcal{C}_i^\ell]$. Of course, if $\ell \in [\![2, L]\!]$, then all cliques $\mathcal{C}_j^\ell$ with $j \in \mathfrak{n}[\mathcal{C}_i^\ell]$ have the same "parent" clique as $\mathcal{C}_i^\ell$, i.e., if $\mathcal{C}_i^\ell \subseteq \mathcal{C}_a^{\ell-1}$, then $\mathcal{C}_j^\ell \subseteq \mathcal{C}_a^{\ell-1}$, $\forall j \in \mathfrak{n}[\mathcal{C}_i^\ell]$. The set of cliques $\mathcal{C}_j^\ell$ that include $\mathcal{C}_i^\ell$ as a neighbor clique is defined with $\mathfrak{m}[\mathcal{C}_i^\ell]$, i.e., if $\mathcal{C}_j^\ell$ is such that $i \in \mathfrak{n}[\mathcal{C}_j^\ell]$, then $j \in \mathfrak{m}[\mathcal{C}_i^\ell]$. Note that it is not required that $\mathfrak{n}[\mathcal{C}_i^\ell] \equiv \mathfrak{m}[\mathcal{C}_i^\ell]$.

Next, all necessary communication assumptions are stated. First, we formulate the *basic* communication capabilities of every agent which are only limited in exchanging information with their neighboring agents within the same level-$(L-1)$ clique.

Assumption 2 (Basic Communication Capabilities). *Every agent $i \in \mathcal{R}$ (or just $\mathcal{C}_i^L \in \mathcal{R}$) is able to exchange information with all agents $j \in \mathfrak{n}[\mathcal{C}_i^L] \cup \mathfrak{m}[\mathcal{C}_i^L]$.*

Note that compared to the potential scale of a VLMAS, and assuming that neighbor sets would be relatively small, this is considered to be a minor communication requirement. In the following assumption, we establish how communication between different cliques materializes, by assigning *increased* communication capabilities to specific agents.

Assumption 3 (Increased Communication Capabilities). *For all levels $\ell \in [\![1, L-1]\!]$, in each clique $\mathcal{C}_i^\ell$, there exists one agent with "level-ℓ communication capabilities", or more briefly a "level-ℓ agent". Each level-ℓ agent is able to exchange information with:*
- *all level-$(\ell+1)$ agents of the cliques $\mathcal{C}_n^{\ell+1}$ such that $\mathcal{C}_n^{\ell+1} \subseteq \mathcal{C}_i^\ell$,*
- *all level-ℓ agents that belong in $\mathfrak{n}[\mathcal{C}_i^\ell] \cup \mathfrak{m}[\mathcal{C}_i^\ell]$.*

D. Virtual States for Clique Dynamics

Finally, towards associating the control of the cliques of levels $1, \ldots, L-1$, with covariance steering, we define their correspoding *virtual* states $x_i^\ell \in \mathbb{R}^{n_x}$ and controls $u_i^\ell \in \mathbb{R}^{n_u}$, for all $i \in \mathcal{R}^\ell$. Since the dynamics of all agents are homogeneous, the virtual states are modeled to follow the same dynamics as the agents,

$$x_{i,k+1}^\ell = A x_{i,k}^\ell + B u_{i,k}^\ell + w_{i,k}^\ell, \quad (9)$$

where $w_{i,k}^\ell \sim \mathcal{N}(0, W)$. Note that this is still far from associating the control of cliques with covariance steering, since the initial and target random distributions of the cliques of all levels $\ell \in [\![1, L-1]\!]$ are not available and require rigorous selection such that the hierarchical structure is satisfied. In
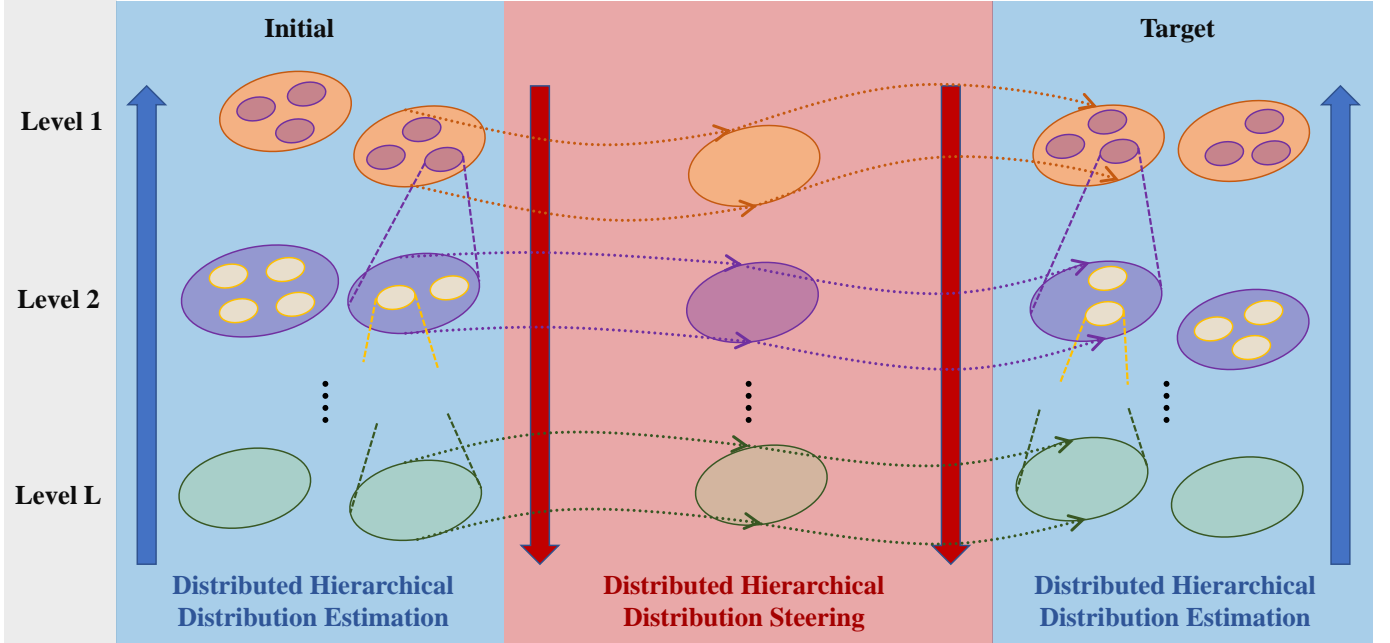
Fig. 2: The DHDC framework consists of two parts. The first part (DHDE) is a bottom-up approach that estimates the initial and target random distributions that correspond to all cliques of levels $\ell \in [\![1, L-1]\!]$, such that the assumed hierarchical clustered structure is satisfied. The second part (DHDS) is a top-down approach that steers the distributions of the cliques/agents of all levels based on the initial and target distributions provided by DHDE.

the next section, we propose an approach for estimating these initial and target random distributions as Gaussian distributions $\mathcal{N}(\mu_{i,0}^\ell, \Sigma_{i,0}^\ell)$ and $\mathcal{N}(\mu_{i,\mathrm{f}}^\ell, \Sigma_{i,\mathrm{f}}^\ell)$, $\forall i \in \mathcal{R}^\ell$, $\ell \in [\![1, L-1]\!]$.

III. DISTRIBUTED HIERARCHICAL DISTRIBUTION ESTIMATION

This section focuses on the problem of finding the optimal Gaussian distributions for representing the initial and target state configurations of the cliques $\mathcal{C}_i^\ell$, $\forall \mathcal{R}^\ell$, $\ell \in [\![1, L-1]\!]$, based on the known level-L distributions. We label this problem as the *inter-level distribution estimation* problem. Towards addressing it, we propose Distributed Hierarchical Distribution Estimation (DHDE), a hierarchical distributed approach that operates in a *bottom-up* fashion (Fig. 2) for estimating the desired random distributions.

A. Single-Clique Distribution Estimation

To facilitate the exposition of our ideas, we first consider the simplified subproblem of *single-clique distribution estimation*, where the objective is to estimate the Gaussian distribution that best describes a clique $\mathcal{C}_i^\ell$, i.e., best captures the random state distributions of all cliques $\mathcal{C}_n^{\ell+1} \subseteq \mathcal{C}_i^\ell$ with a single Gaussian distribution, without considering potential overlaps in the 2D space between neighboring cliques. Furthermore, note that the distribution estimation problem has the same form, either we refer to the initial or target configurations, thus we make no distinction between the two and drop the corresponding notation.

In order to find the optimal distribution $\mathcal{N}_i^\ell = \mathcal{N}(\mu_i^\ell, \Sigma_i^\ell)$ for capturing all the distributions $\mathcal{N}_n^{\ell+1} = \mathcal{N}(\mu_n^{\ell+1}, \Sigma_n^{\ell+1})$ of the cliques $\mathcal{C}_n^{\ell+1}$ such that $\mathcal{C}_n^{\ell+1} \subseteq \mathcal{C}_i^\ell$, we select the KL divergence metric $\mathbb{D}_{\mathrm{KL}}(\mathcal{N}_n^{\ell+1} \| \mathcal{N}_i^\ell)$ to measure discrepancies between distributions. In addition, by defining the parts of the means and covariances that correspond to the position coordinates as $\bar{\mu}_i = H\mu_i$ and $\bar{\Sigma}_i = H\Sigma_i H^{\mathrm{T}}$, we impose the constraints

$$\mathcal{E}_\theta[\bar{\mu}_n^{\ell+1}, \bar{\Sigma}_n^{\ell+1}] \subseteq \mathcal{E}_\theta[\bar{\mu}_i^\ell, \bar{\Sigma}_i^\ell], \quad n \in \mathcal{C}_i^\ell, \qquad (10)$$

so that the assumed hierarchical clustered structure is indeed satisfied. Therefore, the single-clique distribution estimation problem can be stated for a particular clique $\mathcal{C}_i^\ell \in \mathcal{R}^\ell$ of a given level $\ell \in [\![1, L-1]\!]$, as follows.

Problem 2 (Single-Clique Distribution Estimation Problem). *Find the optimal Gaussian distribution $\mathcal{N}(\mu_i^\ell, \Sigma_i^\ell)$ such that*

$$\{\mu_i^\ell, \Sigma_i^\ell\} = \operatorname{argmin} J_i^{\mathrm{e}}(\mu_i^\ell, \Sigma_i^\ell) \qquad (11a)$$

$$\text{s.t.} \quad \mathcal{E}_\theta[\bar{\mu}_n^{\ell+1}, \bar{\Sigma}_n^{\ell+1}] \subseteq \mathcal{E}_\theta[\bar{\mu}_i^\ell, \bar{\Sigma}_i^\ell], \ n \in \mathcal{C}_i^\ell, \qquad (11b)$$

$$\Sigma_i^\ell \succ 0, \qquad (11c)$$

where

$$J_i^{\mathrm{e}} = \sum_{n \in \mathcal{C}_i^\ell} \mathbb{D}_{\mathrm{KL}}(\mathcal{N}_n^{\ell+1} \| \mathcal{N}_i^\ell). \qquad (12)$$

In the following proposition, we present a tractable optimization problem whose optimal solution provides the optimal

solution of Problem 2. From now on, the level superscripts will be omitted unless not obvious from the context.

Proposition 1. *Let us introduce the auxiliary optimization variables $Q_i = \Sigma_i^{-1}$, $q_i = \Sigma_i^{-1}\mu_i$ and $\tau_n \in \mathbb{R}$, $\forall n \in \mathcal{C}_i^\ell$. The optimal solution of Problem 2 is obtained by solving the following convex optimization problem*

$$\min \hat{J}_i^{\text{e}}(Q_i, q_i) \tag{13a}$$

$$\text{s.t} \quad S_n(\bar{Q}_i, \bar{q}_i, \tau_n) \succeq 0, \quad n \in \mathcal{C}_i^\ell, \tag{13b}$$

$$\tau_n \geq 0, \quad n \in \mathcal{C}_i^\ell, \tag{13c}$$

$$Q_i \succ 0, \tag{13d}$$

w.r.t. Q_i, q_i and $\{\tau_n\}_{n \in \mathcal{C}_i^\ell}$, where

$$\hat{J}_i^{\text{e}}(Q_i, q_i) = \sum_{n \in \mathcal{C}_i^\ell} -\log|Q_i| + \text{tr}(Q_i \Sigma_n) + q_i^{\text{T}} Q_i^{-1} q_i$$
$$- 2\mu_n^{\text{T}} q_i + \mu_n^{\text{T}} Q_i \mu_n, \tag{14a}$$

$$S_n = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12}^{\text{T}} & S_{22} & S_{23} \\ 0^{\text{T}} & S_{23}^{\text{T}} & S_{33} \end{bmatrix}, \tag{14b}$$

$$S_{11} = -\bar{Q}_i + \tau_n \bar{\Sigma}_n^{-1}, \quad S_{12} = \bar{q}_i - \tau_n \bar{\Sigma}_n^{-1} \bar{\mu}_n, \tag{14c}$$

$$S_{22} = \alpha + \tau_n \bar{\mu}_n^{\text{T}} \bar{\Sigma}_n^{-1} \bar{\mu}_n - \alpha \tau_n, \quad S_{23} = \bar{q}_i^{\text{T}}, \quad S_{33} = \bar{Q}_i,$$
$$\bar{Q}_i = HQ_i H^{\text{T}}, \quad \bar{q}_i = Hq_i, \quad \alpha = f_{\chi^2, 2}^{-1}(\theta), \tag{14d}$$

and setting $\mu_i = Q_i^{-1} q_i$ and $\Sigma_i = Q_i^{-1}$.

Proof: The proof is provided in Section I.A of the Supplementary Material (SM). ∎

B. Multi-Clique Distribution Estimation

Let us now consider an extended version of the previous problem, labeled as the *multi-clique distribution estimation* one, where the objective is to simultaneously estimate the Gaussian distributions $\mathcal{N}_i^\ell$ of all cliques $\mathcal{C}_i^\ell$ such that $\mathcal{C}_i^\ell \subseteq \mathcal{C}_a^{\ell-1}$, i.e., all cliques that belong in the same parent clique $\mathcal{C}_a^{\ell-1}$ (or just $\mathcal{R}^1$ if $\ell = 1$). In this case, it is also necessary to ensure that the θ-probability confidence ellipses $\mathcal{E}_\theta[\bar{\mu}_i^\ell, \bar{\Sigma}_i^\ell]$ of neighboring cliques will not overlap with each other, i.e., we also impose the following constraints

$$\mathcal{E}_\theta[\bar{\mu}_i^\ell, \bar{\Sigma}_i^\ell] \cap \mathcal{E}_\theta[\bar{\mu}_j^\ell, \bar{\Sigma}_j^\ell] = \emptyset, \; j \in \mathfrak{n}[\mathcal{C}_i^\ell], \tag{15}$$

between neighbor cliques of the same parent clique. Hence, the multi-clique distribution estimation problem can be formulated as follows.

Problem 3 (Multi-Clique Distribution Estimation Problem). *Given a parent clique $\mathcal{C}_a^{\ell-1}$ (or $\mathcal{R}^1$ if $\ell = 1$), find the optimal Gaussian distributions $\mathcal{N}\left(\mu_i^\ell, \Sigma_i^\ell\right)$ for all $i \in \mathcal{C}_a^{\ell-1}$, such that*

$$\left\{\mu_i^\ell, \Sigma_i^\ell\right\}_{i \in \mathcal{C}_a^{\ell-1}} = \underset{i \in \mathcal{C}_a^{\ell-1}}{\text{argmin}} \sum J_i^{\text{e}}(\mu_i^\ell, \Sigma_i^\ell) \tag{16a}$$

$$\text{s.t.} \quad \mathcal{E}_\theta[\bar{\mu}_n^{\ell+1}, \bar{\Sigma}_n^{\ell+1}] \subseteq \mathcal{E}_\theta[\bar{\mu}_i^\ell, \bar{\Sigma}_i^\ell], \; n \in \mathcal{C}_i^\ell, \tag{16b}$$

$$\mathcal{E}_\theta[\bar{\mu}_i^\ell, \bar{\Sigma}_i^\ell] \cap \mathcal{E}_\theta[\bar{\mu}_j^\ell, \bar{\Sigma}_j^\ell] = \emptyset, \; j \in \mathfrak{n}[\mathcal{C}_i^\ell], \tag{16c}$$

$$\Sigma_i^\ell \succ 0, \; i \in \mathcal{C}_a^{\ell-1}, \tag{16d}$$

where $J_i^{\text{e}}(\mu_i^\ell, \Sigma_i^\ell)$ is the same as in Problem 2.

In the following proposition, we present an optimization problem whose optimal solution provides a suboptimal solution for Problem 3.

Proposition 2. *Let us introduce the auxiliary optimization variables $Q_i = \Sigma_i^{-1}$, $q_i = \Sigma_i^{-1}\mu_i$, $\phi_i \in \mathbb{R}$ and $\tau_{i,n} \in \mathbb{R}$, $\forall n \in \mathcal{C}_i^\ell$, $\forall i \in \mathcal{C}_a^{\ell-1}$. A suboptimal solution for Problem 3 is obtained by solving the following optimization problem*

$$\min \sum_{i \in \mathcal{C}_a^{\ell-1}} \hat{J}_i^{\text{e}}(Q_i, q_i) \tag{17a}$$

$$\text{s.t} \quad S_{i,n}(\bar{Q}_i, \bar{q}_i, \tau_{i,n}) \succeq 0, \; n \in \mathcal{C}_i^\ell, \tag{17b}$$

$$\tau_{i,n} \geq 0, \; n \in \mathcal{C}_i^\ell, \tag{17c}$$

$$h_{i,j}(\bar{Q}_i, \bar{q}_i, \phi_i, \bar{Q}_j, \bar{q}_j, \phi_j) \leq 0, \; j \in \mathfrak{n}[\mathcal{C}_i^\ell], \tag{17d}$$

$$T_i(\bar{Q}_i, \phi_i) \succeq 0, \tag{17e}$$

$$Q_i \succ 0, \; i \in \mathcal{C}_a^{\ell-1}, \tag{17f}$$

w.r.t. Q_i, q_i, $\{\tau_n\}_{n \in \mathcal{C}_i^\ell}$ and ϕ_i, $\forall i \in \mathcal{C}_a^{\ell-1}$, where

$$h_{i,j} = r_i(\phi_i) + r_j(\phi_j) - \left\| \bar{Q}_i^{-1} \bar{q}_i - \bar{Q}_j^{-1} \bar{q}_j \right\|_2, \tag{18a}$$

$$r_i = \phi_i^{-1/2}, \quad T_i = \bar{Q}_i - \phi_i \alpha I, \tag{18b}$$

$\hat{J}_i^{\text{e}}$, $S_{i,n}$, $\bar{Q}_i$ and $\bar{q}_i$ are as in Proposition 1, and setting $\mu_i = Q_i^{-1} q_i$ and $\Sigma_i = Q_i^{-1}$ for every $i \in \mathcal{C}_a^{\ell-1}$.

Proof: Provided in Section I.B of the SM. ∎

Remark 1. *The optimal solution of the problem in Proposition 2 might be suboptimal - but always feasible - for Problem 3, in the sense that the constraints (17d) and (17c) formulate a "tighter" version of (16c), as shown in Section I.B of the SM.*

Note that the cost and all constraints in (17) are convex except for (17d). As explained in the next section, we address that through iterative linearization of the constraint.

C. Distributed Hierarchical Distribution Estimation

Let us now formulate the full *inter-level distribution estimation* problem, whose objective is to estimate the optimal distributions $\mathcal{N}(\mu_i^\ell, \Sigma_i^\ell)$ of all cliques $\mathcal{C}_i^\ell \in \mathcal{R}^\ell$ of all levels $\ell \in [\![1, L-1]\!]$. Of course, this problem will consist of many interconnected instances of Problem 3. In fact, the inter-level distribution estimation problem can be formulated as follows.

Problem 4 (Inter-Level Distribution Estimation Problem). *For all $\ell \in [\![1, L-1]\!]$ and for all $a \in \mathcal{R}^{\ell-1}$ (if $\ell \geq 2$), find the sets of optimal Gaussian distributions $\{\mathcal{N}(\mu_i^\ell, \Sigma_i^\ell)\}$, $i \in \mathcal{C}_a^{\ell-1}$, that solve each corresponding Problem 3.*

Algorithm 1 DHDE Algorithm

1: **Inputs:** HIER_STRUCTURE, $\mu_i^L, \Sigma_i^L, \forall i \in \mathcal{R}, \theta$
2: **for** $\ell = L - 1 : -1 : 1$ **do**
3:   **if** $\ell > 1$ **then**
4:     $\mathcal{A} \leftarrow \mathcal{R}^{\ell-1}$
5:   **else**
6:     $\mathcal{A} \leftarrow \{1\}$
7:   **for** $a \in \mathcal{A}$ **do** (in parallel)
8:     **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
9:       *All $n \in \mathcal{C}_i^\ell$ send $\mu_n^{\ell+1}, \Sigma_n^{\ell+1}$ to i.*
10:     **while** TERM_CRITERION $==$ False **do**
11:       **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
12:         $\{\tilde{Q}_i, \tilde{q}_i, \tilde{\phi}_i\} \leftarrow$ Solve (21).
13:         *All $j \in \mathfrak{m}[\mathcal{C}_i^\ell]$ send $\{Q_i^j, q_i^j, \phi_i^j\}$ to i.*
14:       **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
15:         $\{G_i, g_i, z_i\} \leftarrow$ Update with (23).
16:         *All $j \in \mathfrak{n}[\mathcal{C}_i^\ell]$ send $\{G_j, g_j, z_j\}$ to i.*
17:       **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
18:         $\{\Xi_i, \xi_i, y_i\} \leftarrow$ Update with (24).
19:     **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
20:       $\{\mu_i^\ell, \Sigma_i^\ell\} \leftarrow$ Compute based on Q_i, q_i.

The goal of the proposed Distributed Hierarchical Distribution Estimation (DHDE) framework is to solve Problem 4 using a bottom-up strategy, since the only a priori known distributions are the level-L ones. In this direction, we propose first solving all instances of Problem 3 in Level $L-1$, then based on the acquired information, i.e. all $\mathcal{N}_i^{L-1}$, solve all instances of Problem 3 in Level $L-2$, to obtain all $\mathcal{N}_i^{L-2}$, and so on. Of course, based on the results of Section III-B, we also replace all these instances of Problem 3 with the Proposition 2 problems. Finally, to further distribute computations, we propose solving each such problem in a distributed manner using an approach based on the Alternating Direction Method of Multipliers (ADMM) [9].

In order to solve problem (17) in a distributed fashion, we first need to address the coupling induced by the constraints (17d) between neighboring cliques. Hence, we define the copy variables Q_j^i, q_j^i and ϕ_j^i, for all $j \in \mathfrak{n}[\mathcal{C}_i^\ell]$, and subsequently the augmented variables

$$\tilde{Q}_i = [Q_i; \{Q_j^i\}_{j \in \mathfrak{n}[\mathcal{C}_i^\ell]}], \tag{19a}$$

$$\tilde{q}_i = [q_i; \{q_j^i\}_{j \in \mathfrak{n}[\mathcal{C}_i^\ell]}], \tag{19b}$$

$$\tilde{\phi}_i = [\phi_i; \{\phi_j^i\}_{j \in \mathfrak{n}[\mathcal{C}_i^\ell]}]. \tag{19c}$$

Note that the addition of the copy variables might have created multiple variables for each agent. To accommodate for that, we also define the global variables $G = [\{G_i\}_{i \in \mathcal{C}_a^{\ell-1}}]$, $g = [\{g_i\}_{i \in \mathcal{C}_a^{\ell-1}}]$, $z = [\{z_i\}_{i \in \mathcal{C}_a^{\ell-1}}]$, and impose the consensus constraints

$$\tilde{Q}_i = \tilde{G}_i, \ \tilde{q}_i = \tilde{g}_i, \ \tilde{\phi}_i = \tilde{z}_i, \tag{20}$$

where $\tilde{G}_i = [G_i; \{G_j\}_{j \in \mathfrak{n}[\mathcal{C}_i^\ell]}]$, $\tilde{g}_i = [g_i; \{g_j\}_{j \in \mathfrak{n}[\mathcal{C}_i^\ell]}]$ and $\tilde{z}_i = [z_i; \{z_j\}_{j \in \mathfrak{n}[\mathcal{C}_i^\ell]}]$.

Next, we present the algorithm updates; for a full derivation, the reader is referred to Section I.C of the SM. First, each level-ℓ agent i updates its local variables $\tilde{Q}_i, \tilde{q}_i$ and $\tilde{\phi}_i$ by solving the optimization problem

$$\{\tilde{Q}_i, \tilde{q}_i, \tilde{\phi}_i\} = \arg\min \tilde{J}_i^e(\tilde{Q}_i, \tilde{q}_i, \tilde{\phi}_i) \tag{21a}$$

$$\text{s.t} \quad S_{i,n}(Q_i, q_i, \tau_{i,n}) \succeq 0, \ n \in \mathcal{C}_i^\ell, \tag{21b}$$

$$\tau_{i,n} \geq 0, \ n \in \mathcal{C}_i^\ell, \tag{21c}$$

$$\bar{h}_{i,j}(Q_i, q_i, \phi_i, Q_j^i, q_j^i, \phi_j^i) \leq 0, \ j \in \mathfrak{n}[\mathcal{C}_i^\ell], \tag{21d}$$

$$T_i(Q_i, \phi_i) \succeq 0, \tag{21e}$$

$$Q_i \succ 0 \tag{21f}$$

where the augmented costs $\tilde{J}_i^e(\tilde{Q}_i, \tilde{q}_i, \tilde{\phi}_i)$ are given by

$$\tilde{J}_i^e = \hat{J}_i^e(Q_i, q_i) + \text{tr}(\Xi_i^T(\tilde{Q}_i - \tilde{G}_i)) + \xi_i^T(\tilde{q}_i - \tilde{g}_i)$$
$$+ y_i^T(\tilde{\phi}_i - \tilde{z}_i) + \frac{\rho_Q}{2}\|\tilde{Q}_i - \tilde{G}_i\|_F^2 + \frac{\rho_q}{2}\|\tilde{q}_i - \tilde{g}_i\|_2^2$$
$$+ \frac{\rho_\phi}{2}\|\tilde{\phi}_i - \tilde{z}_i\|_2^2, \tag{22}$$

with $\rho_Q, \rho_q, \rho_\phi > 0$ being penalty parameters and Ξ_i, ξ_i, y_i being the dual variables of the corresponding constraints. In addition, we take advantage of the iterative nature of ADMM and replace $h_{i,j}(\cdot)$ in (21d) with its linear approximation $\bar{h}_{i,j}(\cdot)$ to convexify problem (21). For more details, the reader is referred to Section I.D of the SM.

Subsequently, the components of the global variables are updated locally by each level-ℓ agent as follows,

$$G_i \leftarrow \frac{1}{|\mathfrak{m}'[\mathcal{C}_i^\ell]|} \sum_{j \in \mathfrak{m}'[\mathcal{C}_i^\ell]} Q_i^j \tag{23}$$

where $\mathfrak{m}'[\mathcal{C}_i^\ell] = \mathfrak{m}[\mathcal{C}_i^\ell] \cup \{i\}$. The updates for g_i and z_i have the same form as (23) if we replace Q_i^j with q_i^j and ϕ_i^j, respectively. Finally, the dual variables are updated with

$$\Xi_i \leftarrow \Xi_i + \rho_Q(\tilde{Q}_i - \tilde{G}_i) \tag{24a}$$

$$\xi_i \leftarrow \xi_i + \rho_q(\tilde{q}_i - \tilde{g}_i) \tag{24b}$$

$$y_i \leftarrow y_i + \rho_\phi(\tilde{\phi}_i - \tilde{z}_i). \tag{24c}$$

Note that all updates in (21), (23), (24) can be performed in parallel by each level-ℓ agent i. Therefore, all computations take place in a decentralized manner.

The DHDE algorithm with all necessary computation and communication steps is illustrated in Alg. 1. The method operates in a bottom-up fashion for $\ell = \{L-1, L-2, \ldots, 1\}$. For a particular level ℓ, the first step is that all agents n such that $\mathcal{C}_n^{\ell+1} \subseteq \mathcal{C}_i^\ell$ send $\mu_n^{\ell+1}, \Sigma_n^{\ell+1}$ to the level-ℓ agent i that corresponds to the clique $\mathcal{C}_i^\ell$ (Line 9). Then the iterative ADMM procedure starts for every different group of agents that corresponds to a clique $\mathcal{C}_a^{\ell-1}$ (Lines 10-18). Note that these procedures can of course take place in parallel. Focusing into a particular group of agents that belong in clique $\mathcal{C}_a^{\ell-1}$, the ADMM updates are performed as follows. First, each agent $i \in \mathcal{C}_a^{\ell-1}$ solves (21) to update its local variables $\{\tilde{Q}_i, \tilde{q}_i, \tilde{\phi}_i\}$ (Line 12). Subsequently, each i receives the copy variables $\{Q_i^j, q_i^j, \phi_i^j\}$ from all $j \in \mathfrak{m}[\mathcal{C}_i^\ell]$ (Line 13). As a result, each

agent i can now obtain the new iterates for $\{G_i, g_i, z_i\}$ (Line 15) through updates of the form (23). Afterwards, all agents $j \in \mathfrak{n}[\mathcal{C}_i^\ell]$ send the variables $\{G_j, g_j, z_j\}$ to each i (Line 16) so that each agent i updates its dual variables $\{\Xi_i, \xi_i, y_i\}$ (Line 18) with (24). Once a predefined termination criterion is satisfied, each agent i computes the variables μ_i^ℓ and Σ_i^ℓ (Line 20). Now that the estimates $\mu_i^\ell, \Sigma_i^\ell$ have been found for all $\ell \in \mathcal{R}^\ell$, this procedure repeats for the above level $\ell - 1$, and so on, until level 1 is reached.

IV. DISTRIBUTED HIERARCHICAL DISTRIBUTION STEERING

After associating all cliques of all levels with their initial and target Gaussian distributions, we proceed with addressing the problem of steering all state distributions from the initial to the target ones. We label this problem as the *inter-level distribution steering* one. To solve this problem, we propose a *top-down* hierarchical distributed method (Fig. 2) called Distributed Hierarchical Distribution Steering (DHDS).

A. Multi-Clique Distribution Steering

Before formulating the inter-level problem, let us again first state a subproblem that serves as the basic component of the full problem. In particular, we consider the *multi-clique distribution steering* problem whose objective is to steer the Gaussian distributions of (the virtual states of) all i such that the cliques $\mathcal{C}_i^\ell \subseteq \mathcal{C}_a^{\ell-1}$, for a specific parent clique $\mathcal{C}_a^{\ell-1}$ and level ℓ. Of course, in the case where $\ell = 1$, we consider $\mathcal{R}^1$ as the parent clique. Thus, the multi-clique distribution steering problem can be formulated as follows.

Problem 5 (Multi-Clique Distribution Steering Problem). *Given a parent clique $\mathcal{C}_a^{\ell-1}$ (or $\mathcal{R}^1$ if $\ell = 1$), find the optimal control sequences u_i^ℓ for all $i \in \mathcal{C}_a^{\ell-1}$, such that*

$$\{u_i^\ell\}_{i \in \mathcal{C}_a^{\ell-1}} = \operatorname{argmin} \sum_{i \in \mathcal{C}_a^{\ell-1}} J_i^{\mathrm{s}}(u_i^\ell) \tag{25a}$$

$$\text{s.t.} \quad x_{i,k+1}^\ell = A x_{i,k}^\ell + B u_{i,k}^\ell + w_{i,k}^\ell, \tag{25b}$$

$$\mathbb{E}[x_{i,N}^\ell] = \mu_{i,\mathrm{f}}^\ell, \ \mathbb{COV}[x_{i,N}^\ell] \preceq \Sigma_{i,\mathrm{f}}^\ell, \tag{25c}$$

$$q_{i,j,k}^\ell(p_{i,k}^\ell, p_{j,k}^\ell) \geq 0, \ j \in \mathfrak{n}[\mathcal{C}_i^\ell], \tag{25d}$$

$$s_{i,k}^\ell(p_{i,k}^\ell) \geq 0, \tag{25e}$$

$$\mathcal{E}_\theta[\bar{\mu}_{i,k}^\ell, \bar{\Sigma}_{i,k}^\ell] \subseteq \mathcal{E}_\theta[\bar{\mu}_{a,k}^{\ell-1}, \bar{\Sigma}_{a,k}^{\ell-1}], \tag{25f}$$

$$k \in [\![0, N]\!], \ i \in \mathcal{C}_a^{\ell-1},$$

where

$$J_i^{\mathrm{s}}(u_i^\ell) = \sum_{k=0}^{N-1} \mathbb{E}[u_{i,k}^{\ell\,\mathrm{T}} R u_{i,k}^\ell], \tag{26a}$$

$$q_{i,j,k}^\ell(p_{i,k}^\ell, p_{j,k}^\ell) = \mathbb{P}\left(\|p_{i,k}^\ell - p_{j,k}^\ell\|_2 \geq d_{\mathrm{inter}}^\ell\right) - \theta, \tag{26b}$$

$$s_{i,k}^\ell(p_{i,k}^\ell) = \mathbb{P}\left(\|p_{i,k}^\ell - p_o\|_2 \geq d_{\mathrm{obs}}^\ell + r_o\right) - \theta. \tag{26c}$$

and d_{inter}^ℓ, d_{obs}^ℓ are prespecified parameters.

In the following, we will be omitting level superscripts, unless not clear by the context.

B. Problem Transformation

In order to address Problem 5, we consider the affine disturbance feedback control policies proposed in [4], with

$$u_{i,k} = \bar{u}_{i,k} + L_{i,k}(x_{i,0} - \mu_{i,0}) + \sum_{\kappa=0}^{k-1} K_{i,(k-1,\kappa)} w_{i,\kappa}, \tag{27}$$

where $\bar{u}_{i,k} \in \mathbb{R}^{n_u}$ are the feed-forward control terms and $L_{i,k}, \ K_{i,(k-1,\kappa)} \in \mathbb{R}^{n_u \times n_x}$ are feedback gain matrices. After concentrating these variables for the entire time horizon, we obtain the decision variables $\bar{u}_i \in \mathbb{R}^{N n_u}$, $L_i \in \mathbb{R}^{N n_u \times n_x}$ and $K_i \in \mathbb{R}^{N n_u \times N n_x}$, with their exact forms provided in Section II.A of the SM. Thus, the control sequences u_i can be rewritten as

$$u_i = \bar{u}_i + L_i(x_{i,0} - \mu_{i,0}) + K_i w_i. \tag{28}$$

If we also write the dynamics (9) in their concatenated form,

$$x_i = \Psi_0 x_{i,0} + \Psi_u u_i + \Psi_w w_i, \tag{29}$$

where the matrices Ψ_0, Ψ_u and Ψ_w are provided in the SM, then through (28), we obtain

$$x_i = \Psi_0 x_{i,0} + \Psi_u \bar{u}_i + \Psi_u L_i(x_{i,0} - \mu_{i,0}) + (\Psi_w + \Psi_u K_i) w_i. \tag{30}$$

Therefore, given that affine transformations preserve Gaussianity, the entire state sequence x_i is a Gaussian vector, which implies that $x_{i,k} \sim \mathcal{N}(\mu_{i,k}, \Sigma_{i,k})$, with

$$\mu_{i,k} = f_{i,k}(\bar{u}_i), \quad \Sigma_{i,k} = F_{i,k}(L_i, K_i), \tag{31}$$

where the exact expressions for $f_{i,k}(\bar{u}_i)$ and $F_{i,k}(L_i, K_i)$ are provided in the SM. Note that the mean states only depend on the feed-forward controls, while the state covariances only depend on the feedback matrices. This is a useful fact that we will later exploit to further increase computational efficiency.

In the following proposition, we present an new optimization problem whose optimal solution provides a suboptimal one for Problem 5.

Proposition 3. *A suboptimal solution for Problem 5 is obtained by solving the following optimization problem. In particular, given a parent clique $\mathcal{C}_a^{\ell-1}$ (or $\mathcal{R}^1$ if $\ell = 1$), find the optimal decision variables $\bar{u}_i, L_i, K_i$ for all $i \in \mathcal{C}_a^{\ell-1}$, such that*

$$\{\bar{u}_i, L_i, K_i\}_{i \in \mathcal{C}_a^{\ell-1}} = \operatorname{argmin} \sum_{i \in \mathcal{C}_a^{\ell-1}} \hat{J}_i^{\mathrm{s}}(\bar{u}_i, L_i, K_i) \tag{32a}$$

$$\text{s.t.} \quad \mathfrak{f}_{i,N}(\bar{u}_i) = 0, \ \mathcal{F}_{i,N}(L_i, K_i) \succeq 0, \tag{32b}$$

$$\mathcal{Q}_{i,k}(L_i, K_i) \succeq 0, \ \mathfrak{s}_{i,k}(\bar{u}_i) \geq 0, \tag{32c}$$

$$\mathfrak{q}_{i,j,k}(\bar{u}_i, \bar{u}_j) \geq 0, \ j \in \mathfrak{n}[\mathcal{C}_i^\ell], \tag{32d}$$

$$\mathfrak{p}_{i,k}(\bar{u}_i) \leq 0, \ k \in [\![0, N]\!], \ i \in \mathcal{C}_a^{\ell-1}, \tag{32e}$$

with

$$\hat{J}_i^{\mathrm{s}} = \bar{u}_i^{\mathrm{T}} \bar{R} \bar{u}_i + \mathrm{tr}(\bar{R} K_i W K_i^{\mathrm{T}} + \bar{R} L_i \Sigma_{i,0} L_i^{\mathrm{T}}), \tag{33a}$$

$$\mathfrak{f}_{i,N}(\bar{u}_i) = \mu_{i,\mathrm{f}} - f_{i,N}(\bar{u}_i), \tag{33b}$$

$$\mathcal{F}_{i,N}(L_i, K_i) = \begin{bmatrix} \Sigma_{i,\mathrm{f}} & \Phi_{i,N}(L_i, K_i) \\ \Phi_{i,N}(L_i, K_i)^{\mathrm{T}} & I \end{bmatrix}, \tag{33c}$$

$$\mathcal{Q}_{i,k}(L_i, K_i) = \begin{bmatrix} \left(\frac{r^\ell}{\sqrt{\alpha}}\right)^2 I & H\Phi_{i,k}(L_i, K_i) \\ \Phi_{i,k}(L_i, K_i)^{\mathrm{T}} H^{\mathrm{T}} & I \end{bmatrix}, \tag{33d}$$

$$\mathfrak{s}_{i,k}(\bar{u}_i, \bar{u}_j) = \|\bar{f}_{i,k}(\bar{u}_i) - p_o\|_2 - r^\ell - r_o - d_{\mathrm{obs}}^\ell, \tag{33e}$$

$$\mathfrak{p}(\bar{u}_i) = \|\bar{f}_{i,k}(\bar{u}_i) - \bar{\mu}_{a,k}^{\ell-1}\|_{\hat{P}} - 1 \leq 0, \tag{33f}$$

$$\mathfrak{q}_{i,j,k}(\bar{u}_i, \bar{u}_j) = \|\bar{f}_{i,k}(\bar{u}_i) - \bar{f}_{i,k}(\bar{u}_j)\|_2 - 2r^\ell - d_{\mathrm{inter}}^\ell, \tag{33g}$$

where $\bar{R} = \mathrm{blkdiag}(\mathrm{R}, \dots, \mathrm{R}) \in \mathbb{S}_n^{++}$ $\bar{f}_{i,k} = H f_{i,k}$, r^ℓ *is a prespecified parameter and* $\Phi_{i,k}(L_i, K_i)$ *is an affine function of* L_i, K_i*, provided in Section II.B of the SM. The matrix* $\hat{P} = \frac{1}{\alpha} U \hat{\Lambda}^{-1} U^{\mathrm{T}}$*, where*

$$\hat{\Lambda} = \left(\Lambda^{1/2} - \frac{r}{\sqrt{\alpha}} I\right)^2, \tag{34}$$

with $[\Lambda, U]$ *being the eigendecomposition of the matrix* $\bar{\Sigma}_{a,k}^{\ell-1}$*.*

Proof: Provided in Section II.B of the SM. ∎

Remark 2. *The optimal solution of the problem in Proposition 3 might be suboptimal - but always feasible - for Problem 5, since the constraints* (32c)-(32d) *provide a "tighter" version of* (25d)-(25e)*, as shown in Section II.B of the SM.*

Remark 3. *A major advantage of the problem presented in Proposition 3, is that it is separable w.r.t. the feed-forward control variables* $\bar{u}_i$ *and the feedback matrices* L_i, K_i*. In addition, an inter-agent coupling only appears through the variables* $\bar{u}_i$ *- because of* (32d) *- and not through* L_i, K_i*. As shown in the next section, these two facts can be exploited to significantly increase computational efficiency.*

C. Distributed Hierarchical Distribution Steering

Let us now introduce the full *inter-level distribution steering* problem, where the objective lies in finding the optimal control sequences that will steer the state distributions of all cliques $\mathcal{C}_i^\ell \in \mathcal{R}^\ell$ of all levels $\ell \in \llbracket 1, L \rrbracket$ from the initial distributions $\mathcal{N}(\mu_{i,0}^\ell, \Sigma_{i,0}^\ell)$ to the target ones $\mathcal{N}(\mu_{i,\mathrm{f}}^\ell, \Sigma_{i,\mathrm{f}}^\ell)$. Being in symmetry with its estimation counterpart, this problem will consist of many interconnected instances of Problem 5. Indeed, the inter-level distribution steering problem is stated as follows.

Problem 6 (Inter-Level Distribution Steering Problem). *For all* $\ell \in \llbracket 1, L \rrbracket$ *and for all* $a \in \mathcal{R}^{\ell-1}$ *(where by convention* $\mathcal{R}^0 = \{1\}$*), find the optimal control policies* $\{u_i\}, i \in \mathcal{C}_a^{\ell-1}$*, that solve each corresponding Problem 5.*

Remark 4. *The combination of Problems 4 and 6 yields a relaxed form of Problem 1, that is substantially more computationally tractable as the VLMAS dimensionality increases.*

Algorithm 2 DHDS Algorithm

1: **Inputs:** HIER_STRUCTURE, $A, B, W, \mu_{i,0}^\ell, \Sigma_{i,0}^\ell, \mu_{i,\mathrm{f}}^\ell, \Sigma_{i,\mathrm{f}}^\ell$ $\forall i \in \mathcal{R}^\ell, d_{\mathrm{obs}}^\ell, d_{\mathrm{inter}}^\ell, r^\ell, \forall \ell \in \llbracket 1, L \rrbracket, \theta$
2: **for** $\ell = 1 : L$ **do**
3:     **if** $\ell > 1$ **then**
4:         $\mathcal{A} \leftarrow \mathcal{R}^{\ell-1}$
5:     **else**
6:         $\mathcal{A} \leftarrow \{1\}$
7:     **for** $a \in \mathcal{A}$ **do** (in parallel)
8:         **if** $\ell > 1$ **then**
9:             *Each* $a \in \mathcal{R}^{\ell-1}$ *sends* $\mu_{a,k}^{\ell-1}, \Sigma_{a,k}^{\ell-1}, k \in \llbracket 0, N \rrbracket$ *to all* $i \in \mathcal{C}_a^{\ell-1}$*.*
10:         **while** TERM_CRITERION $==$ False **do**
11:             **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
12:                 $\tilde{u}_i \leftarrow$ Solve (36).
13:                 *All* $j \in \mathfrak{m}[\mathcal{C}_i^\ell]$ *send* $\bar{u}_j^i$ *to each* i*.*
14:             **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
15:                 $b_i \leftarrow$ Update with (38).
16:                 *All* $j \in \mathfrak{n}[\mathcal{C}_i^\ell]$ *send* b_j *to each* i*.*
17:             **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
18:                 $v_i \leftarrow$ Update with (39).
19:         **for** $i \in \mathcal{C}_a^{\ell-1}$ **do** (in parallel)
20:             $\{L_i, K_i\} \leftarrow$ Solve (41).
21:             $\mu_{i,k}^\ell, \Sigma_{i,k}^\ell \leftarrow$ Get with (31) for all $k \in \llbracket 0, N \rrbracket$.

In contrast with DHDE, the aim of DHDS is to solve Problem 6 with a top-down procedure, since every level-ℓ subproblem depends on a level-$(\ell-1)$ subproblem. Therefore, we intend to first solve all instances of Problem 5 in Level 1, then based on the acquired distributions $\mathcal{N}(\mu_{i,k}^1, \Sigma_{i,k}^1)$, solve all instances of Problem 5 in Level 2, and so on, until Level L is reached. After replacing all instances of Problem 5 with the problem proposed in Proposition 3, we propose again an ADMM-based approach to solve each such problem in a distributed manner.

Note that as emphasized in Remark 3, we can separate solving (32) into a part that would only involve the variables $\bar{u}_i, i \in \mathcal{C}_a^{\ell-1}$ and a second part including only the variables $L_i, K_i, i \in \mathcal{C}_a^{\ell-1}$. Let us start with addressing the former part, which can be formulated as

$$\{\bar{u}_i\}_{i \in \mathcal{C}_a^{\ell-1}} = \arg\min \sum_{i \in \mathcal{C}_a^{\ell-1}} \hat{J}_{i,1}^{\mathrm{s}}(\bar{u}_i) \tag{35a}$$

$$\text{s.t.} \quad \mathfrak{f}_{i,N}(\bar{u}_i) = 0, \ \mathfrak{s}_{i,k}(\bar{u}_i) \geq 0, \ \mathfrak{p}_{i,k}(\bar{u}_i) \leq 0, \tag{35b}$$

$$\mathfrak{q}_{i,j,k}(\bar{u}_i, \bar{u}_j) \geq 0, \ j \in \mathfrak{n}[\mathcal{C}_i^\ell], \tag{35c}$$

$$k \in \llbracket 0, N \rrbracket, \ i \in \mathcal{C}_a^{\ell-1}.$$

where $\hat{J}_{i,1}^{\mathrm{s}}(\bar{u}_i) = \bar{u}_i^{\mathrm{T}} \bar{R} \bar{u}_i$. To solve this problem in a distributed manner, we first need to introduce the augmented local variables $\tilde{u}_i = [\bar{u}_i; \{\bar{u}_j^i\}_{j \in \mathfrak{n}[\mathcal{C}_i^\ell]}]$, where $\bar{u}_j^i$ are copy variables for all $j \in \mathfrak{n}[\mathcal{C}_i^\ell]$. As in DHDE, we also need to introduce a global variable $b = [\{b_i\}_{i \in \mathcal{C}_a^{\ell-1}}]$ and the constraints $\tilde{u}_i = \tilde{b}_i$,
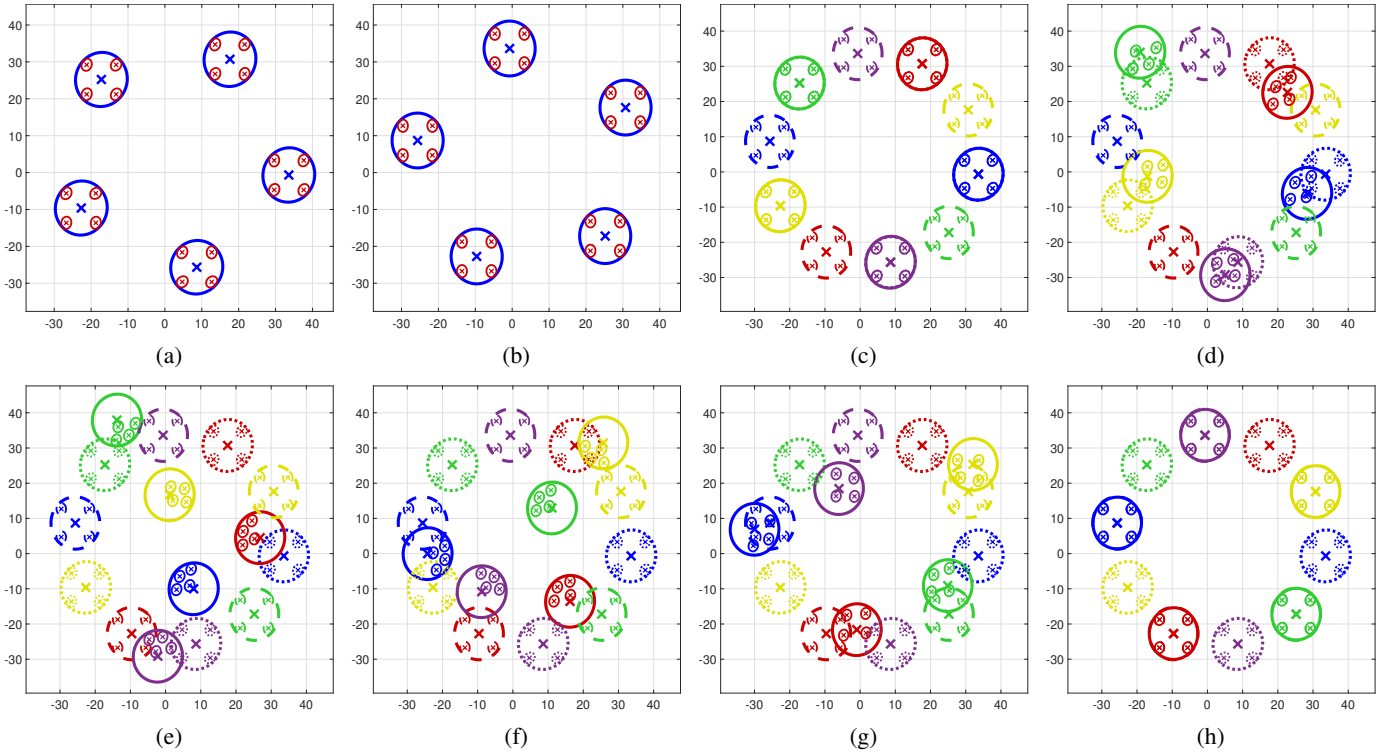
Fig. 3: **A two-level small-scale scenario.** In (a), the 99.7%-confidence ellipses of the initial distributions are shown. Each level corresponds to a specific color (1: blue, 2: red). Similarly, (b) shows the target distributions for the two levels. In (c)-(h), the steering of all distributions is demonstrated for time instants $k = 0, 20, 40, 80, 80, 100$, respectively. At each snapshot, the current, initial and target distributions are shown with solid, dotted and dashed ellipses, respectively. Here, each color corresponds to a specific level-1 clique.

where $\tilde{b}_i = [b_i; \{b_j\}_{j \in \mathfrak{n}[\mathcal{C}_i^\ell]}]$. Subsequently, we can arrive to a distributed ADMM algorithm for solving (35). For the full derivation and additional implementation details, the reader is referred to Sections II.C and II.D of the SM.

The updates of the resulting algorithm are presented below. First, each level-ℓ agent updates its local variable $\tilde{u}_i$ by solving the following local optimization problem

$$\tilde{u}_i = \arg\min \tilde{J}_{i,1}^{\text{s}}(\bar{u}_i) \tag{36a}$$

$$\text{s.t.} \quad \mathfrak{f}_{i,N}(\bar{u}_i) = 0, \ \mathfrak{s}_{i,k}(\bar{u}_i) \geq 0, \ \mathfrak{p}_{i,k}(\bar{u}_i) \leq 0, \tag{36b}$$

$$\mathfrak{q}_{i,j,k}(\bar{u}_i, \bar{u}_j^i) \geq 0, \ j \in \mathfrak{n}[\mathcal{C}_i^\ell], \ k \in [\![0, N]\!]. \tag{36c}$$

with each cost $\tilde{J}_{i,1}^{\text{s}}(\tilde{u}_i)$ given by

$$\tilde{J}_{i,1}^{\text{s}}(\tilde{u}_i) = \hat{J}_{i,1}^{\text{s}}(\bar{u}_i) + v_i^{\text{T}}(\tilde{u}_i - \tilde{b}_i) + \frac{\rho_u}{2}\|\tilde{u}_i - \tilde{b}_i\|_2^2. \tag{37}$$

where $\rho_u > 0$ and v_i are the dual variables corresponding to the constraints $\tilde{u}_i = \tilde{b}_i$. Subsequently, all global variables components are updated with

$$b_i \leftarrow \frac{1}{|\mathfrak{m}'[\mathcal{C}_i^\ell]|} \sum_{j \in \mathfrak{m}'[\mathcal{C}_i^\ell]} \bar{u}_i^j \tag{38}$$

and, finally, the dual updates are given by

$$v_i \leftarrow v_i + \rho_u(\tilde{u}_i - \tilde{b}_i). \tag{39}$$

Again, all updates in (36), (38) and (39) can be performed in parallel by each level-ℓ agent i. Therefore, all computations are operated in a decentralized fashion. This iterative algorithm repeats until a prespecified termination criterion is met.

Let us now focus on the part of problem (32) that only involves the variables L_i, K_i. This subproblem can be formulated as

$$\{L_i, K_i\}_{i \in \mathcal{C}_a^{\ell-1}} = \arg\min \sum_{i \in \mathcal{C}_a^{\ell-1}} \hat{J}_{i,2}^{\text{s}}(L_i, K_i) \tag{40a}$$

$$\text{s.t.} \quad \mathcal{F}_{i,N}(L_i, K_i) \succeq 0, \ \mathcal{Q}_{i,k}(L_i, K_i) \succeq 0, \tag{40b}$$

$$k \in [\![0, N]\!], \ i \in \mathcal{C}_a^{\ell-1},$$

where $\hat{J}_{i,2}^{\text{s}}(\bar{u}_i) = \text{tr}(\bar{R}K_i W K_i^{\text{T}} + \bar{R}L_i \Sigma_{i,0} L_i^{\text{T}})$. A key observation here is that there exists no inter-agent coupling between the optimization variables of different agents, so we can further split the problem and independently solve each agent's problem as follows

$$\{L_i, K_i\} = \arg\min \hat{J}_{i,2}^{\text{s}}(L_i, K_i) \tag{41}$$

$$\text{s.t.} \quad \mathcal{F}_{i,N}(L_i, K_i) \succeq 0, \ \mathcal{Q}_{i,k}(L_i, K_i) \succeq 0, \ k \in [\![0, N]\!].$$

for each $i \in \mathcal{C}_a^{\ell-1}$.

The DHDS algorithm with all required computation and communication steps is demonstrated in Alg. 2. The algorithm

is executed in a top-down manner for $\ell = \{1, \ldots, L\}$. For a specific level ℓ, each agent a that correspond to a parent clique $\mathcal{C}_a^{\ell-1} \in \mathcal{R}^{\ell-1}$ (if $\ell > 1$) must send the variables $\mu_{a,k}^{\ell-1}, \Sigma_{a,k}^{\ell-1}, k \in [\![0, N]\!]$, to all agents $i \in \mathcal{C}_a^{\ell-1}$ (Line 9). Afterwards, the iterative ADMM algorithm starts for every separate group of agents that corresponds to a particular clique $\mathcal{C}_a^{\ell-1}$ (Lines 10-18). First, each agent $i \in \mathcal{C}_a^{\ell-1}$ solves its local subproblem (36) to update its local variable $\tilde{u}_i$ (Line 12). Then, each agent i receives the copy variables $\bar{u}_i^j$ from all $j \in \mathfrak{m}[\mathcal{C}_i^\ell]$ (Line 13), so that the variables b_i are updated (Line 15) through (38). Subsequently, all agents $j \in \mathfrak{n}[\mathcal{C}_i^\ell]$ send b_j to each i (Line 16) and each agent i updates its dual variable v_i (Line 18) with (39). This iterative procedure terminates once a predefined termination criterion is met. Finally, each agent $i \in \mathcal{C}_a^{\ell-1}$ also computes its optimal feedback gains $\{L_i, K_i\}$ (Line 20) through solving (41). After the optimal variables $\{\bar{u}_i, L_i, K_i\}$ have been found for all $i \in \mathcal{C}_a^{\ell-1}$, then the state means and covariances $\mu_{i,k}^\ell, \Sigma_{i,k}^\ell$ can be obtained (Line 21) through (31). Subsequently, the same procedure repeats for the next level $\ell + 1$, and so on, until level L is reached.

V. Simulation Results

This section provides simulation experiments that verify the effectiveness and scalability of the proposed DHDC framework. In Section V-A, we consider a two-level small-scale system to facilitate the exposition of how DHDE and DHDS operate. A larger three-level system of agents is then used in Section V-B to further illustrate how the two proposed sub-frameworks work. Subsequently, in Section V-C, we showcase the applicability of DHDC on a six-level VLMAS of two million robots. Finally, we compare the proposed approach against other CS methods in Section V-D, and illustrate its superior computational efficiency and safety performance. The reader is encouraged to also refer to the supplementary video for a full illustration of the results.

A. Small-Scale Scenario

It this task, we consider a two-level small-scale hierarchy with 5 cliques of 4 agents, thus a total of 20 agents (Fig. 3). All agents are modeled with 2D double integrator dynamics. The time horizon of the task is $N = 100$. For additional details on the dynamics and algorithmic parameters, the reader is referred to Section III of the SM. The 99.7%-confidence ellipses of the initial Gaussian distributions of the robots are shown in Fig. 3a with red color. The level-1 distributions that are computed with DHDE also shown with blue color. As expected, the confidence ellipses of the assigned distributions for the level-1 cliques encompass the ones of the distributions of the robots (level 2). Similarly, the target distributions of all levels are illustrated in Fig. 3b.

The goal of all agents is to reach to a target distribution that is in diametrically opposite position to their initial one, while of course avoiding collisions with the rest of the agents. Figures 3c-3h show snapshots of the motion of all distributions for time instants $k = 0, 20, 30, 40, 50, 60, 70, 100$, respectively. As illustrated, DHDS is able to successfully steer all the

level-1 distributions to the targets provided by DHDE. In the meantime, the distributions of all agents are also steered to their targets while satisfying the constraint that forces their distributions to lie within their parent clique distributions. Thanks to the fact that the solution of the level-1 problems guarantees that the agents of one clique will not collide with the ones of other cliques, the level-2 subproblems only take into account collision avoidance constraints with agents form the same clique. In Fig. 3h, all level-1 and level-2 distributions have successfully reached to their targets.

B. Large-Scale Scenario

Next, a larger scale 3-level system of $2 \times 8 \times 9 = 144$ agents (Fig. 4) is used to further exhibit the effectiveness of the proposed method. First, we focus on illustrating the performance of the DHDE sub-framework. Figure 4a shows the initial and target distributions of the robots (level 3) and the ones corresponding to the level-2 cliques that are obtained through DHDE. In Fig. 4b, the level-1 distributions are shown, while Fig. 4c shows the results DHDE would provide if the inter-clique constraints (16c) had been omitted. This highlights the importance of including these constraints and of using the advanced Problem 3 formulation in our setup instead of the more simplistic Problem 2 one.

The performance of the DHDS algorithm is then demonstrated in Figs. 4d-4i. In particular, the motion of the distributions of all levels is shown in Figs. 4d-4f as they are being steered towards their target while avoiding the obstacles in the middle. In Figs. 4g-4i, we focus into the black dotted box of the previous plots to further emphasize on the motion of the robots (level 3). As shown, the distributions of the robots are steered while staying within the distributions of their parent cliques and not overlapping with each other.

C. VLMAS Scenario

Subsequently, we consider a VLMAS with a 6-level hierarchical clustered structure, where the first level has 2 cliques, each clique in levels $2, \ldots, 4$ contains 16 sub-cliques, and finally each level-5 clique contains 16 agents (level 6). Therefore, this VLMAS consists of $2 \times 16^5 = 2,097,152$ agents. As in the previous tasks, the initial and target distributions of all cliques of all levels are first estimated, and subsequently, the actual distributions are steered while satisfying the probabilistic safety constraints for collision and obstacle avoidance.

Figures 5a-5d show the distributions of the cliques of levels 1 and 2. In particular, the 99.7%-confidence regions of the initial (left) and target (right) distributions are shown with dotted ellipses. Note that these distributions are the result of DHDE, first computing the level-5 distributions, then the level-4 ones, and so on - for a detailed demonstration, the reader is referred to the supplementary video. The motion of all level-1 (blue) and level-2 (red) distributions is shown in Figs. 5a-5d for time instants $k = 30, 60, 80, 100$. As DHDS is a top-down framework, the level-1 distributions are first steered to their targets while successfully avoiding the obstacle. Subsequently, all level-2 clique distributions are also
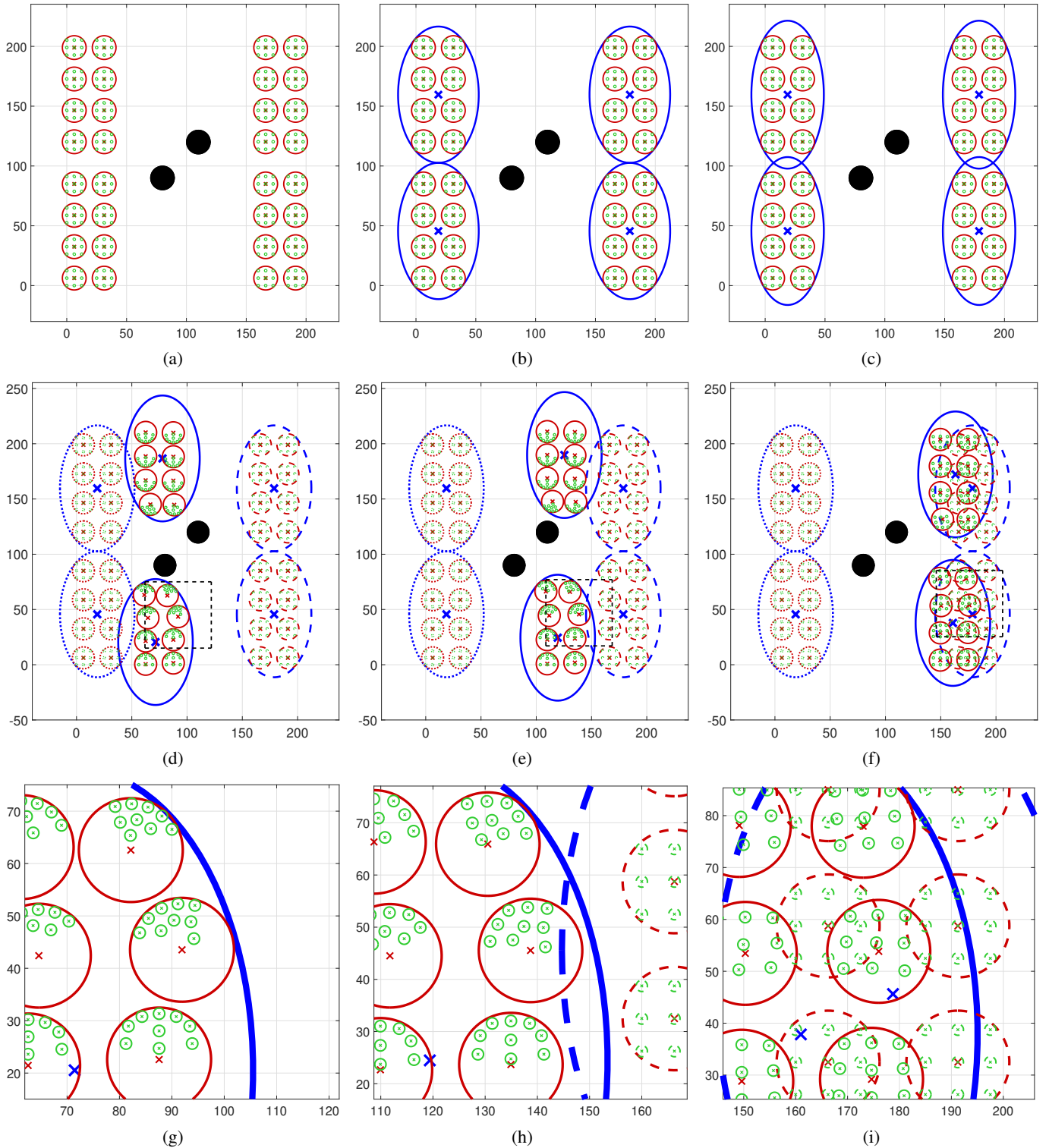
Fig. 4: **A three-level large-scale scenario.** In (a), the initial (LHS) and target (RHS) distributions of the robots (level 3, green) are shown along with the level-2 ones (red) that are computed by DHDE. In (b), the level-1 distributions (blue) that are obtained by DHDE are illustrated. In (c), we show the resulting (overlapping) level-1 distributions if the inter-clique non-overlapping constraints are omitted. In (d)-(f), the distributions of all levels are steered to their targets (time instants $k = 40, 60, 80$, respectively), while avoiding the obstacles in the middle (black circles). In (g)-(i), we focus into the black dotted boxes of the above figures and show the motion of the level-3 distributions in more detail.
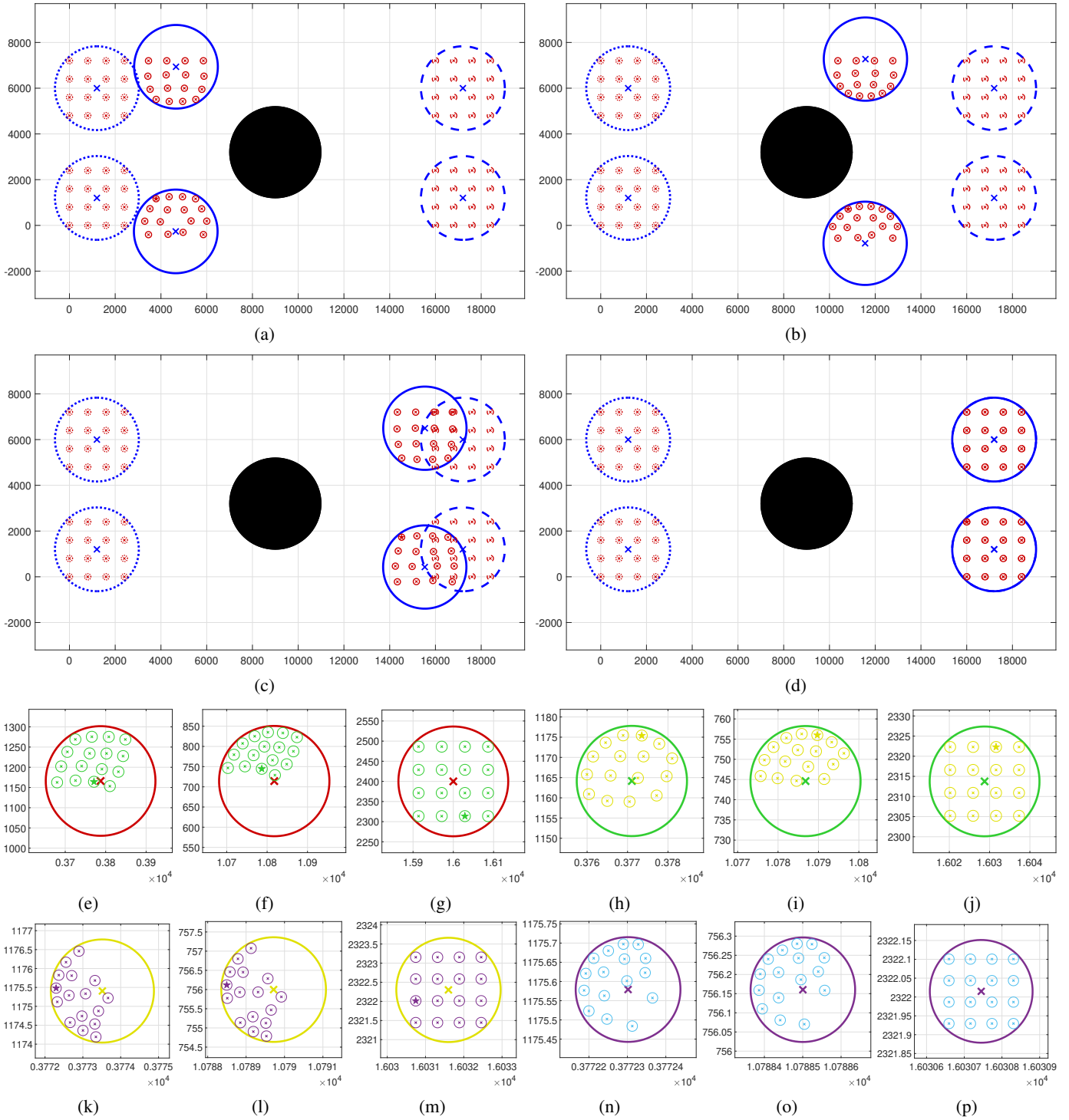
Fig. 5: **A six-level VLMAS scenario.** Each level corresponds to a specific color (1: blue, 2: red, 3: green, 4: yellow, 5: purple, 6: cyan). Fig. (a)-(d) show levels 1 and 2 at time instants $k = 30, 60, 80, 100$, respectively. The initial (left) and target (right) distributions are shown with dotted ellipses. Fig. (e)-(g) display one specific level-2 clique and its level-3 subcliques at $k = 30, 60, 100$. Similarly, Fig. (h)-(j) show one specific level-3 clique and its level-4 subcliques, Fig. (k)-(m) show one specific level-4 clique and its level-5 subcliques, and Fig. (n)-(p) show one specific level-4 clique and its level-6 subcliques, for the same time instants. Each clique we are "zooming" into, is shown with a star in the figures of the above level.

steered to their corresponding targets while staying within the limits of their parent cliques and avoiding collisions with each other. In Fig. 5d all level-$1, 2$ clique distributions have successfully reached to their targets.

In Figs. 5e-5g, we focus into a randomly selected level-2 clique and illustrate the motion of the level-3 (green) distributions that belong in this clique, for time instants $k = 30, 60, 100$, respectively. As shown, all level-3 cliques remain within the limits of their parent cliques, while also avoiding collisions. In Fig. 5g, the level-3 cliques have reached to their target configuration. Similarly, Figs. 5h-5j focus into a level-3 clique and illustrate the motion of all level-4 (yellow) sub-cliques that belong in this clique. Again, all cliques are able to safely reach to their target distributions. The motion of the distributions of specific cliques in levels 5 (purple) and 6 (cyan) are also shown in Figs. 5k-5m and 5n-5p, respectively. All distributions of all levels are successfully steered to their targets, which indicates that all agents have reached their target distributions.

D. Comparison with Related Methods

Finally, we highlight the significant computational efficiency and safety capabilities of DHDC by comparing it with the Centralized CS (CCS) [4] and Distributed CS (DCS) [31] methods. In CCS, the full level-L CS problem is solved without any splitting being considered. In DCS, the level-L CS problem is solved using an ADMM-based distributed approach, which however cannot benefit from the hierarchical structure of the VLMAS.

1) Computational Demands: To perform a computational demands comparison, we repeat the task of Section V-C, starting from small numbers of agents. All simulations were performed in Matlab R2021b using MOSEK 9.1.9 [1] as the optimization solver and a laptop computer with an 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz and a 32GB RAM memory. The computational times for tasks ranging from two agents to two million agents, are shown in Fig. 6. The results highlight the superior scalability of DHDC against DCS and CCS. This is mainly attributed to the following reasons: i) First, DHDC does not require solving problems of a significantly large scale given that there exist no cliques that contain a very large number of cliques in the level below them. ii) Second, the efficient variables splitting of the subproblems in DHDC reduces the amount of necessary copy variables, making DHDC significantly more computationally efficient than DCS. iii) Third, in CCS, the multi-agent problem is always solved in a centralized fashion, which leads to high-dimensional semidefinite programming problems that soon become computationally intractable.

2) Safety: Next, we conduct a comparison on the safety capabilities of each method. Given that CCS cannot scale for large-scale systems, we exclude it from the comparison. In Fig. 7, we demonstrate the collisions percentages for DHDC and DCS. As both methods enforce consensus through soft constraints, it follows that as tasks get more complicated, collisions might appear. Nevertheless, DHDC outperforms
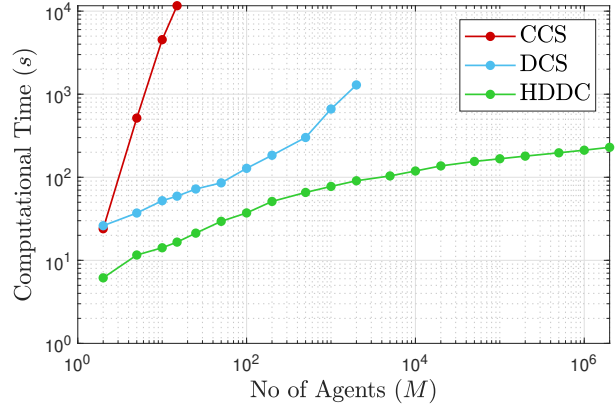

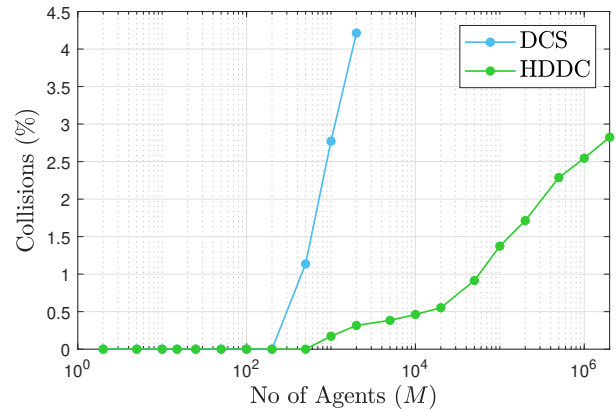Fig. 6: Comparison of computational times between DHDC (proposed), CCS and DCS.

Fig. 7: Comparison of collision percentages between DHDC (proposed) and DCS.

DCS in terms of safety as well, since the cliques/agents always need to only consider safety constraints associated with the other cliques/agents that are within the same clique, thus making the optimization problems easier to solve.

VI. CONCLUSION

This paper proposes a scalable hierarchical distributed control framework named DHDC for the control of VLMAS that admit a multi-level hierarchical clustered structure. The first part of the framework, DHDE, associates the initial and target configurations of the cliques of all levels to representative Gaussian distributions that satisfy all the requirements of the hierarchical structure. The second part, DHDS, steers the distributions of all cliques and agents to their prescribed target distributions. Simulation experiments demonstrate the scalability of DHDC to VLMAS with up to two million robots. Therefore, DHDC is shown to be able to control systems of a very-large scale by exploiting the control of distributions within hierarchical structures.

Future work will focus on further expanding DHDC for more general problem setups. A straightforward extension

would be to extend the framework for systems nonlinear dynamics by incorporating nonlinear versions of CS [29, 33]. In addition, we wish to extend DHDC for the case where the hierarchical structure is unknown by exploring the incorporation of hierarchical distribution alignment methods in unsupervised learning such as [19, 35]. Finally, future work will also focus on establishing formal convergence guarantees for the proposed method.

REFERENCES

[1] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019. URL http://docs.mosek.com/9.0/toolbox/index.html.

[2] Omur Arslan, Dan P Guralnik, and Daniel E Koditschek. Coordinated robot navigation via hierarchical clustering. *IEEE Transactions on Robotics*, 32(2):352–371, 2016.

[3] Efstathios Bakolas. Finite-horizon covariance control for discrete-time stochastic linear systems subject to input constraints. *Automatica*, 91:61–68, 2018.

[4] Isin M. Balci and Efstathios Bakolas. Covariance control of discrete-time gaussian linear systems using affine disturbance feedback control policies. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2324–2329, 2021. doi: 10.1109/CDC45484.2021.9683236.

[5] Isin M Balci, Efstathios Bakolas, Bogdan Vlahov, and Evangelos A Theodorou. Constrained covariance steering based tube-mppi. In *2022 American Control Conference (ACC)*, pages 4197–4202. IEEE, 2022.

[6] Levent Bayındır. A review of swarm robotics tasks. *Neurocomputing*, 172:292–321, 2016.

[7] Boris Benedikter, Alessandro Zavoli, Zhenbo Wang, Simone Pizzurro, and Enrico Cavallini. Convex approach to covariance control with application to stochastic low-thrust trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 45(11):2061–2075, 2022.

[8] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and autonomous systems*, 41 (2-3):89–99, 2002.

[9] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. ISSN 1935-8237. doi: 10.1561/2200000016.

[10] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal steering of a linear stochastic system to a final probability distribution, part i. *IEEE Trans. Automat. Contr.*, 61(5):1158–1169, 2015.

[11] Jorge Cortés and Magnus Egerstedt. Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, 10(6):495–503, 2017.

[12] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, pages 885–897. PMLR, 2020.

[13] Carlo Ferrari, Enrico Pagello, Jun Ota, and Tamio Arai. Multirobot motion coordination in space and time. *Robotics and autonomous systems*, 25(3-4):219–229, 1998.

[14] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797, 2019.

[15] Junyan Hu, Parijat Bhowmick, Inmo Jang, Farshad Arvin, and Alexander Lanzon. A decentralized cluster formation containment framework for multirobot systems. *IEEE Transactions on Robotics*, 37(6):1936–1955, 2021.

[16] Maximilian Hüttenrauch, Sosic Adrian, Gerhard Neumann, et al. Deep reinforcement learning for swarm systems. *Journal of Machine Learning Research*, 20(54):1–31, 2019.

[17] Georgios Kotsalis, Guanghui Lan, and Arkadi S. Nemirovski. Convex optimization for finite-horizon robust covariance control of linear stochastic systems. *SIAM Journal on Control and Optimization*, 59(1):296–319, 2021. doi: 10.1137/20M135090X.

[18] Jaemin Lee, Efstathios Bakolas, and Luis Sentis. Hierarchical task-space optimal covariance control with chance constraints. *IEEE Control Systems Letters*, 6:2359–2364, 2022.

[19] John Lee, Max Dabagia, Eva Dyer, and Christopher Rozell. Hierarchical optimal transport for multimodal distribution alignment. *Advances in neural information processing systems*, 32, 2019.

[20] Fengjiao Liu, George Rapakoulias, and Panagiotis Tsiotras. Optimal covariance steering for discrete-time linear stochastic systems. *arXiv preprint arXiv:2211.00618*, 2022.

[21] Yogeswaran Mohan and SG Ponnambalam. An extensive review of research in swarm robotics. In *2009 world congress on nature & biologically inspired computing (nabic)*, pages 140–145. IEEE, 2009.

[22] Nadia Nedjah and Luneque Silva Junior. Review of methodologies and tasks in swarm robotics towards standardization. *Swarm and Evolutionary Computation*, 50:100565, 2019.

[23] Kazuhide Okamoto and Panagiotis Tsiotras. Optimal stochastic vehicle path planning using covariance steering. *IEEE Robot. Autom. Lett.*, 4(3):2276–2281, 2019.

[24] Lynne E Parker. Path planning and motion coordination in multiple mobile robot teams. *Encyclopedia of complexity and system science*, pages 5783–5800, 2009.

[25] Marcus A Pereira, Augustinos D Saravanos, Oswin So, and Evangelos 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. doi: 10. 15607/RSS.2022.XVIII.055.

[26] Dhananjay Raju, Sudarshanan Bharadwaj, Franck Djeumou, and Ufuk Topcu. Online synthesis for runtime enforcement of safety in multiagent systems. *IEEE Transactions on Control of Network Systems*, 8(2):621–632, 2021.

[27] George Rapakoulias and Panagiotis Tsiotras. Discrete-time optimal covariance steering via semidefinite programming. *arXiv preprint arXiv:2302.14296*, 2023.

[28] Wei Ren and Randal W Beard. *Distributed consensus in multi-vehicle cooperative control*, volume 27. Springer, 2008.

[29] Jack Ridderhof, Kazuhide Okamoto, and Panagiotis Tsiotras. Nonlinear uncertainty control with iterative covariance steering. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3484–3490. IEEE, 2019.

[30] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

[31] Augustinos D Saravanos, Alexandros Tsolovikos, Efstathios Bakolas, and Evangelos Theodorou. Distributed Covariance Steering with Consensus ADMM for Stochastic Multi-Agent Systems. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. doi: 10.15607/RSS.2021.XVII.075.

[32] Augustinos D Saravanos, Yuichiro Aoyama, Hongchang Zhu, and Evangelos A Theodorou. Distributed differential dynamic programming architectures for large-scale multi-agent control. *arXiv preprint arXiv:2207.13255*, 2022.

[33] Augustinos D Saravanos, Isin M Balci, Efstathios Bakolas, and Evangelos A Theodorou. Distributed model predictive covariance steering. *arXiv preprint arXiv:2212.00398*, 2022.

[34] Ji Yin, Zhiyuan Zhang, Evangelos Theodorou, and Panagiotis Tsiotras. Trajectory distribution control for model predictive path integral control using covariance steering. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1478–1484. IEEE, 2022.

[35] Mikhail Yurochkin, Sebastian Claici, Edward Chien, Farzaneh Mirzazadeh, and Justin M Solomon. Hierarchical optimal transport for document representation. *Advances in neural information processing systems*, 32, 2019.

[36] Jialong Zhang, Jianguo Yan, and Pu Zhang. Multi-uav formation control based on a novel back-stepping approach. *IEEE Transactions on Vehicular Technology*, 69(3):2437–2448, 2020.

[37] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021.

[38] Pingping Zhu, Chang Liu, and Silvia Ferrari. Adaptive online distributed optimal control of very-large-scale robotic systems. *IEEE Transactions on Control of Network Systems*, 8(2):678–689, 2021.