Adaptive Online Distributed Optimal Control of Very-Large-Scale Robotic Systems

Pingping Zhu, *Member, IEEE*, Chang Liu, and Silvia Ferrari, *Senior Member, IEEE*

***Abstract*—Autonomous systems comprised of many cooperative agents have the potential for enabling long-duration tasks and data collection critical to the understanding of a wide range of phenomena in spatially and temporally variable environments. The adaptive distributed optimal control approach presented in this article extends online approximate dynamic programming to very-large-scale robotics (VLSR) systems that must operate and adapt to highly uncertain and variable environments. Optimal mass transport theory is used to show that, in the Wasserstein–Gaussian mixture model space, the VLSR system's cost to go can be represented by a value functional of the robot distribution and dynamic environmental maps. The approach is demonstrated on a cooperative path planning problem in which knowledge of the obstacles in the environment changes incrementally over time based on *in situ* measurements. Numerical simulations show that the proposed approach significantly outperforms existing methods by finding an approximately optimal solution that avoids obstacles and meets a desired final robot distribution using minimum energy.**

***Index Terms*—Adaptive, cooperative, environmental adaptation, multiagent reinforcement learning (MARL), optimal control, path planning, very-large-scale robotic (VLSR).**

I. Introduction

With the advent of low-cost sensors and embedded systems, very-large-scale robotic (VLSR) systems comprised of hundreds of autonomous robots are becoming a viable solution for conducting long-duration autonomous tasks over large regions of interest [1]. To date, significant progress has been made on VLSR optimal control [2]–[5]; multiagent reinforcement learning (MARL) [6]–[13]; multiagent path planning [14], [15]; and swarm robotics [16]–[21] approaches. However, these and other VLSR methods assume that knowledge of the environment is provided *a priori*. Therefore, the resulting planning and control laws may not be adaptable to changing *in situ* conditions that autonomous robots are likely to encounter when operating over a large region for long periods of time.

Online learning and replanning are typically too computationally expensive due to the combinatorial nature of MARL [9]. In fact, even in known environments, optimal planning for N cooperative robots has been shown to be PSPACE-hard [2]. Distributed optimal control (DOC) [3]–[5] and Nash certainty equivalence (NCE) approaches [22], [23] overcome the scalability issue by defining a macroscopic state, such as the robot distribution or the robot mass, by virtue of a restriction operator and accompanying consistency relationships mapping robot kinodynamic equations onto a macroscopic evolution equation. Although the computational burden is significantly reduced [3]–[5], determining the optimal restriction operator remains too time consuming for online adaptation in response to *in situ* measurements. In VLSR long-duration applications ranging from ocean robotics to space systems, robots operate in the presence of significant uncertainties. At the same time, their performance depends on environmental conditions that cannot be accurately predicted *a priori*, thus requiring adaptation or replanning subject to online measurements [1].

A model-free MARL approach based on mean field control was recently proposed in [11] to address both scalability and uncertainty issues by trial-and-error approximations of the value function. Similarly to the DOC approach used in this article, model-free MARL employs a macroscopic state represented by the robot distribution or probability density function (PDF), and a reward (or Lagrangian) function that depends both on microscopic and macroscopic robot states. Then, the MARL approach is cast as a Markov decision process on the Wasserstein space of measures and implemented by learning a deterministic control law offline. The microscopic robot control law is given by a functional of the macroscopic and microscopic states. However, by this approach only the social average reward can be optimized and the robot decisions are myopic, namely, they are based solely on the latest robot state and robot distribution.

This article presents new adaptive DOC (ADOC) theory for VLSR systems that must operate optimally over time based on *in situ* measurements that influence immediate and future cooperative sensing and navigation performance in highly uncertain environments. The ADOC approach is assumed to be centralized for simplicity but is applicable to a distributed system of robots with a macroscopic evolution equation provided by a stochastic differential equation (SDE), also known as distributed parameter system [5]. Thus, the proposed ADOC approach can be viewed

Manuscript received January 15, 2021; revised January 17, 2021 and June 21, 2021; accepted July 1, 2021. Date of publication July 14, 2021; date of current version August 24, 2021. This work was supported in part by the Office of Naval Research Code 321, in part by the National Science Foundation under Grant ECCS-1556900 and Grant ATD-1738010, and in part by the Marshall University. Recommended by Associate Editor M. Prandini. *(Corresponding author: Pingping Zhu.)*

Pingping Zhu is with the Department of Computer Science and Electrical Engineering, Marshall University, Huntington, WV 25705 USA (e-mail: zhup@marshall.edu).

Chang Liu and Silvia Ferrari are with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: changliu@berkeley.edu; ferrari@cornell.edu).

Digital Object Identifier 10.1109/TCNS.2021.3097306

as a reinforcement learning-approximate dynamic programming (RL-ADP) approach [24], [25] applicable to distributed parameter systems for which environmental conditions must be learned online.

The new ADOC theory presented in this article is based on several key contributions. This article shows that when the environmental information is changing over time, the system's cost to go cannot be expressed as a value function but requires the use of a value *functional*. Then, the ADOC approach can be formulated in a Wasserstein–Gaussian mixture model (GMM) space using the optimal mass transport (OMT) theorem, by assuming that the optimal time-varying robot distribution can be learned and represented via GMMs. A fast ADOC online solution is afforded by proving that the ADOC problem amounts to solving a linear program in a subspace of the Wasserstein-GMM space. The ADOC computational complexity, convergence, and lower and upper bounds on the optimal value functional are also derived in this article. The effectiveness of ADOC is demonstrated on a VLSR obstacle avoidance problem, in which the obstacles are sensed online. The numerical results show that the ADOC approach significantly outperforms existing state-of-the-art methods and scales up to VLSR systems with hundreds of robots.

II. Problem Formulation

Consider the problem of adaptively planning the trajectories of a VLSR system comprised of N cooperative robots deployed in a large obstacle-populated region of interest (ROI) $\mathcal{W} \subset \mathbb{R}^2$. Although the M obstacles, $\mathcal{B}_1, \ldots, \mathcal{B}_M \subset \mathcal{W}$ are unknown *a priori*, an approximate map of the obstacle region $\hat{\mathcal{B}}(t)$ is provided over time by sensors and mapping algorithms on-board the N robots. At the initial time, t_0, an obstacle map $\hat{\mathcal{B}}_0 = \hat{\mathcal{B}}(t_0)$ is provided based on prior information, and, thus, may be subject to significant errors, potentially causing blocked passages and narrow regions to be unknown *a priori*. Let the obstacle map function (OMF) be defined as a binary time-varying function, $m(\mathbf{x}, t) : \mathcal{W} \times \mathbb{R} \to \{0, 1\}$, where 1 indicates that the position $\mathbf{x} \in \mathcal{W}$ is occupied by obstacles at time t, and 0 indicates that it is unoccupied. Let $\mathscr{M}$ denote the space of all possible OMFs in the ROI, such that $m(\cdot, t) \in \mathscr{M}$. In this article, the OMF is obtained from a Hilbert occupancy map $h(\mathbf{x}, t)$ [26], as shown in [27], such that

$$m(\mathbf{x}, t) = \begin{cases} 1, & \text{if } h(\mathbf{x}, t) > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $h(\cdot)$ provides the probability of obstacle occupancy.

The microscopic robot dynamics are modeled by a SDE

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}[\mathbf{x}_i(t), \mathbf{u}_i(t), t] \tag{2}$$

$$\mathbf{x}_i(t_0) = \mathbf{x}_{i_0}, i = 1, \ldots, N, \tag{3}$$

where $\mathbf{x}_i \in \mathcal{W}$ is the ith robot's state or configuration, $\mathbf{u}_i(t) \in \mathcal{U}$ is the ith robot's control input, and $\mathbf{x}_{i_0}$ is the ith robot's initial configuration. Also, it is assumed that the robot state is fully observable and known with negligible errors.

All robots are equipped with identical omnidirectional range sensors that allow them to update the OMF based on *in situ* measurements. The field of view (FOV) of the ith range sensor, denoted by $\mathcal{S}_i(t) \subset \mathcal{W}$, is represented by a circle of radius r centered at $\mathbf{x}_i(t)$. Then, the region covered by the sensors at time t can be represented by $\mathcal{S}(t) = \cup_{i=1}^{N} \mathcal{S}_i(t)$. Assuming connectivity and information sharing, obstacle presence at position $\mathbf{x} \in \mathcal{W}$ is observed and updated at time t if and only if $\mathbf{x} \in \mathcal{S}(t)$.

Let the VLSR macroscopic state be represented by a PDF, $\wp(\mathbf{x}, t) \in \mathscr{P}(\mathcal{W})$, where $\mathscr{P}(\mathcal{W})$ is the space of PDFs with support $\mathcal{W}$ [3]–[5]. Then, the VLSR system performance over a time interval $[t_0, t_f]$ can be expressed by an integral cost function

$$J[\wp(\mathbf{x}, t)] = \phi[\wp(t_f), \wp_f] + \int_{t_0}^{t_f} \mathscr{L}[\wp(\mathbf{x}, t), m(\mathbf{x}, t)]dt \tag{4}$$

representing the cost required for the robots to move from a given initial distribution $\wp_0$ at time t_0 to a desired distribution $\wp_f$ at t_f. The functionals $\phi[\wp(t_f), \wp_f]$ and $\mathscr{L}[\wp(\mathbf{x}, t), m(\mathbf{x}, t)]$ denote the terminal cost and the instantaneous cost or "Lagrangian," respectively. Because the VLSR system's performance depends on the changing obstacle map, $m(\cdot, \cdot)$, the Lagrangian is a functional of the OMF for all $t \in [t_0, t_f]$. The result is a new ADP problem in which the functional structure of the Lagrangian is unknown and must be learned from the range-sensor measurements over time.

III. Adaptive DOC Approach

Because sensor measurements become available at discrete sampling times and must be fused to obtain the OMF, the DOC problem is first discretized with respect to time. Let $\triangle t$ denote the time required to obtain and process the obstacle measurements, such that the time interval $[t_0, t_f]$ can be discretized into $T_f = (t_f - t_0)/\triangle t$ time steps, indexed by $t_k = t_0 + k\triangle t$, where $k = 1, \ldots, T_f$. The robot dynamics in (2) can be discretized and reformulated as

$$\mathbf{x}_{k+1,i} = \mathbf{x}_{k,i} + \dot{\mathbf{x}}_{k,i}\Delta t, \quad i = 1, \ldots, N \tag{5}$$

$$\dot{\mathbf{x}}_{k,i} = \mathbf{f}[\mathbf{x}_{k,i}, \mathbf{u}_{k,i}] \tag{6}$$

where $\mathbf{x}_{k,i} = \mathbf{x}_i(t_k)$ and $\mathbf{u}_{k,i} = \mathbf{u}_i(t_k)$. Then, the macroscopic cost function (4) can be approximated by

$$J(\boldsymbol{\wp}_{0:T_f}) \triangleq \phi(\wp_{T_f}, \wp_f) + \sum_{k=0}^{T_f - 1} \mathscr{L}(\wp_k, m_k) \tag{7}$$

where $\boldsymbol{\wp}_{0:T_f} = [\wp_0 \cdots \wp_{T_f}]$, $\wp_k = \wp(\cdot, t_k)$ and $m_k = m(\cdot, t_k)$.

Because the OMF is time-varying and unknown for future times, the cost function (7) cannot be optimized with respect to the robot distribution using existing approaches [3]–[5]. An online optimization approach for this new class of finite horizon ADP problems is developed here by reformulating the objective function as follows:

$$J(\boldsymbol{\wp}_{0:T}) = \phi(\wp_{T_f}, \wp_f) + \sum_{k=0}^{T_f - 1} \mathscr{L}\left[\wp_k, m_k, \mathcal{C}(\wp_k, m_k)\right] \tag{8}$$

where $\mathcal{C} : \mathscr{P} \times \mathscr{M} \mapsto \mathscr{P}$ is the control law functional, and

$$\wp_{k+1} = \mathcal{C}(\wp_k, m_k). \tag{9}$$

In the rest of this article, $\mathcal{C}(\wp_k, m_k)$ is abbreviated by $\mathcal{C}_k$, and $\mathscr{L}[\wp_k, m_k, \mathcal{C}(\wp_k, m_k)]$ is abbreviated by $\mathscr{L}(\wp_k, m_k, \mathcal{C}_k)$.

Because the VLSR goal is to reach $\wp_f$ by time t_f, there exists at least one terminal macroscopic state $\wp_T \in \mathscr{P}$ that is cost-free, and, thus, can be absorbed in the state for all $k > T$, such that $\mathscr{L}(\wp_T, m_k, \mathcal{C}_k) = 0$ and $\mathcal{C}(\wp_T, m_k) = \wp_T$ when $\wp_T = \wp_f$, or when the distance between $\wp_T$ and $\wp_f$ is less than a user-defined threshold. Then, the VLSR control problem can be assumed to terminate upon reaching $\wp_T$ at time $T \leq T_f$.

Unlike traditional ADP problems [24], [25], the control law functional in (9) depends on the OMF, for which there exists no evolution or dynamic equation because it is updated based on exogenous measurements obtained by the robots over time. Then, it is assumed the latest map m_k is the best estimate of the obstacle layout available at time t_k. Without loss of generality, the $(1 \times T)$ vector of OMFs, $\mathbf{M}_k = [m_0 \cdots m_{k-1}\, m_k \cdots m_k]$, is used to represent the map history. If and when an environmental prediction model is available, the method can be easily applied by modifying the definition of $\mathbf{M}_k$.

A. ADOC Value Functional

The goal of the discrete-time ADOC approach is to learn the optimal control law functional, $\mathcal{C}_k^* : \mathscr{P} \mapsto \mathscr{P}$, online at every kth time step based on the OMF, m_k. Then, the optimal and nonmyopic policy associated with $\mathbf{M}_k$ can be expressed by the vector of functionals

$$\mathbf{\Pi}_k^* \triangleq [\mathcal{C}_0^* \cdots \mathcal{C}_{k-1}^*\, \mathcal{C}_k^* \cdots \mathcal{C}_k^*] \tag{10}$$

and must be determined over time so as to minimize (8). By optimizing the policy subject to the robot kinodynamics (6), reachability is guaranteed and the optimal robot distribution, $\wp_{k+1}^*$, can be realized by the robots using the microscopic control described in Section VI-A.

Given a map and policy, $\mathbf{M}_k$ and $\mathbf{\Pi}_k$, at time step k, the ADOC value functional is defined as

$$\mathcal{V}(\wp_l, l \,|\, \mathbf{M}_k, \mathbf{\Pi}_k, k) \triangleq \begin{cases} \phi(\wp_T, \wp_f) + \sum_{\tau=l}^{k-1} \mathscr{L}(\wp_\tau, \mathbf{M}_k(\tau), \mathcal{C}_\tau) \\ \quad + \sum_{\tau=k}^{T-1} \mathscr{L}(\wp_\tau, \mathbf{M}_k(\tau), \mathcal{C}_k), & 0 \leq l < k \\ \phi(\wp_T, \wp_f) + \sum_{\tau=l}^{T-1} \mathscr{L}(\wp_\tau, \mathbf{M}_k(\tau), \mathcal{C}_k), & k \leq l < T \\ \phi(\wp_T, \wp_f), & l = T \end{cases} \tag{11}$$

and is abbreviated by $\mathcal{V}_k(\wp_l, \mathbf{M}_k, \mathcal{C}_k)$ hereon. Then, from (8), it can be shown that $J(\boldsymbol{\wp}_{0:T}) = \mathcal{V}_{T-1}(\wp_0, \mathbf{M}_{T-1})$. The ADOC cost-to-go estimate or "Q-functional" is defined as

$$\mathcal{Q}_k(\wp_l, \mathbf{M}_k, \wp_{l+1}) \triangleq \mathscr{L}(\wp_l, \mathbf{M}_k(l), \wp_{l+1}) + \mathcal{V}_k(\wp_{l+1}, \mathbf{M}_k) \tag{12}$$

and minimized as explained in the next subsection.

B. ADOC Optimal Control Law Functional

Learning a functional operator online is computationally challenging [28], [29], and, in this article, it is approached using the critic-only Q-learning (CoQL) method developed in [30]. Applying Bellman's equation to the ADOC value- and Q-functionals introduced in the previous subsection, the optimal Q-function can be obtained as follows:

$$\mathcal{Q}_k^*(\wp_l, \mathbf{M}_k, \wp_{l+1}^*) = \min_{\wp_{l+1}} [\mathscr{L}(\wp_l, m_k, \wp_{l+1}) + \mathcal{V}_k^*(\wp_{l+1}, \mathbf{M}_k)]$$

for any $k \leq l < T$, where $\mathcal{V}_k^*(\wp_{l+1}, \mathbf{M}_k)$ is the abbreviation of the optimal value functional, $\mathcal{V}_k^*(\wp_{l+1}, \mathbf{M}_k, \mathcal{C}_k^*)$. Then, the optimal robot distribution can be obtained by solving the optimization problem

$$\begin{aligned} \wp_{k+1}^* &= \arg\min_{\wp_{k+1}} [\mathcal{Q}_k^*(\wp_k, \mathbf{M}_k, \wp_{k+1})] \\ &= \arg\min_{\wp_{k+1}} [\mathscr{L}(\wp_k, m_k, \wp_{k+1}) + \mathcal{V}_k^*(\wp_{k+1}, \mathbf{M}_k)]. \end{aligned} \tag{13}$$

IV. Background on Optimal Mass Transport

An efficient approach for the online optimization of the ADOC Q-function is developed by measuring the cost of the VLSR-distribution evolution using OMT theory [31], [32]. When compared to other approaches for measuring information divergence, such as the Kullback–Leibler (KL) divergence or the Cauchy–Schwarz (CS) divergence, OMT affords significant computational savings and also provides important metric properties [32].

Let $\wp_1, \wp_2 \in \mathcal{P}(\mathcal{W})$ denote two PDFs with support $\mathcal{W}$, and let $\Pi(\wp_1, \wp_2) \subset \mathcal{P}(\mathcal{W} \times \mathcal{W})$ denote the set of all joint PDFs characterized by marginals measures along the two coordinate directions that coincide with $\wp_1$ and $\wp_2$, respectively, and such that

$$\Pi(\wp_1, \wp_2) \triangleq \Bigg\{ \pi \in \mathcal{P}(\mathcal{W} \times \mathcal{W}) \int_{\mathbf{x}_2 \in \mathcal{W}} \pi(\cdot, \mathbf{x}_2) d\mathbf{x}_2 = \wp_1, \text{ and } \int_{\mathbf{x}_1 \in \mathcal{W}} \pi(\mathbf{x}_1, \cdot) d\mathbf{x}_1 = \wp_2 \Bigg\}. \tag{14}$$

Then, the Wasserstein metric is defined as

$$W_2(\wp_1, \wp_2) \triangleq \left[\inf_{\pi \in \Pi(\wp_1, \wp_2)} \int_{\mathcal{W} \times \mathcal{W}} \|\mathbf{x}_1 - \mathbf{x}_2\|^2 d\pi(\mathbf{x}_1, \mathbf{x}_2) \right]^{1/2} \tag{15}$$

where $\|\cdot\|$ is the Euclidean distance [31], and can be shown finite provided the second moments of $\wp_1$ and $\wp_2$ exist [33].

Furthermore, if both of the marginals $\wp_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\wp_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ are Gaussian distributions, the Wasserstein metric can be expressed as

$$W_2(\wp_1, \wp_2) = \Bigg\{ \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2 + tr \left[\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2 \left(\boldsymbol{\Sigma}_1^{1/2} \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{1/2} \right)^{1/2} \right] \Bigg\}^{1/2} \tag{16}$$

where $tr[\cdot]$ denotes the trace of a matrix [32].

Although the Wasserstein metric of two Gaussian distributions can be obtained in closed form, there is no efficient representation for general distributions [34]. Recently in [32], a new metric in the space of all GMMs, $\mathcal{G}(\mathcal{W})$, referred to as Wasserstein-GMM (WG) metric, and defined as

$$d(\wp_1,\wp_2) \triangleq \left\{ \min_{\pi\in\Pi(\boldsymbol{\omega}_1,\boldsymbol{\omega}_2)} \sum_{\imath=1}^{N_1}\sum_{\jmath=1}^{N_2} [W_2(g_1^\imath, g_2^\jmath)]^2 \pi(\imath,\jmath) \right\}^{1/2} \tag{17}$$

was proposed as an efficient approximation to the Wasserstein metric for any two distributions $\wp_1, \wp_2 \in \mathcal{G}(\mathcal{W})$, where $g_1^\imath$ and $g_2^\jmath$ are corresponding Gaussian mixture components with weights $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$, respectively. Then, $\Pi(\boldsymbol{\omega}_1,\boldsymbol{\omega}_2)$ can be used to denote the space of joint GMMs, referred to as Wasserstein-GMM space.

V. ADOC Solution in Wasserstein-GMM Space

Hereon, let the Wasserstein metric $W_2(\wp_k,\wp_{k+1})$ represent the distance between two robot PDFs, $\wp_k$ and $\wp_{k+1}$. For a fixed time interval, $\triangle t$, $W_2(\wp_k,\wp_{k+1})$ is proportional to the PDF velocity

$$\nu_k \triangleq \frac{W_2(\wp_k,\wp_{k+1})}{\triangle t} \tag{18}$$

and it can be shown that the following proportionalities hold: $E_k \propto (\nu_k)^2 \propto [W_2(\wp_k,\wp_{k+1})]^2$, where E_k is the energy required to move from $\wp_k$ to $\wp_{k+1}$. Then, the WG metric in (17) can be adopted as the energy cost in the Lagrangian, provided the robot distributions obey the following assumption.

Assumption 1: Assume that within an acceptable error the optimal robot distribution can be approximated by GMMs, such that

$$\wp_k = \sum_{\imath=1}^{N_k} \omega_k^\imath g_k^\imath, \quad k=0,\ldots,T \tag{19}$$

$$\wp_f = \sum_{j=1}^{L} \omega_f^j g_f^j \tag{20}$$

where N_k and L are the numbers of Gaussian components $g_k^\imath$ and g_f^j with means $\boldsymbol{\mu}_k^\imath$ and $\boldsymbol{\mu}_f^j$, and covariance matrices $\boldsymbol{\Sigma}_k^\imath$ and $\boldsymbol{\Sigma}_f^j$, respectively.

Considering the kernel density estimation (KDE) with Gaussian kernels as a special case of GMM [35], choose $N_k \leq N$, where a relatively small number of Gaussian components is typically required in practice to represent useful robot distributions, and, thus, $N_k \ll N$. Now, let $\boldsymbol{g}_k \triangleq [g_k^1 \cdots g_k^{N_k}]$ and $\boldsymbol{g}_f \triangleq [g_f^1 \cdots g_f^L]$ denote vectors of Gaussian components that are used to approximate the robot time-varying and final PDFs, respectively, by means of corresponding GMM weight vectors $\boldsymbol{\omega}_k \triangleq [\omega_k^1 \cdots \omega_k^{N_k}]$ and $\boldsymbol{\omega}_f \triangleq [\omega_f^1 \cdots \omega_f^L]$, respectively. Then, $\wp_k$ and $\wp_f$ are fully specified by the tuples of parameters $\Theta_k = (N_k, \boldsymbol{g}_k, \boldsymbol{\omega}_k)$ and $\Theta_f = (L, \boldsymbol{g}_f, \boldsymbol{\omega}_f)$. It also follows that the control law functional can be expressed as

$$\mathcal{C}_{k+1} = \sum_{\jmath=1}^{N_{k+1}} \omega_{k+1}^\jmath g_{k+1}^\jmath = \sum_{\imath=1}^{N_k}\sum_{\jmath=1}^{N_{k+1}} \pi_k(\imath,\jmath) g_{k+1}^\jmath \tag{21}$$

where $\pi_k \in \Pi(\boldsymbol{\omega}_k, \boldsymbol{\omega}_{k+1})$ is the joint probability distribution and, given $\wp_k$ and m_k, the control law is fully specified by the tuple of parameters, $\Phi_k = (N_{k+1}, \boldsymbol{g}_{k+1}, \pi_k)$.

The VLSR performance is represented by an integral cost function of the robot PDF in the form (8), derived as follows. First, the distance between two GMM PDFs $\wp_k$ and $\wp_{k+1}$ is defined as

$$\tilde{d}(\wp_k, m_k, \mathcal{C}_k) \triangleq \left\{ \sum_{\imath=1}^{N_k}\sum_{\jmath=1}^{N_{k+1}} [W_2(g_k^\imath, g_{k+1}^\jmath)]^2 \pi_k(\imath,\jmath) \right\}^{1/2}$$

and is minimized to obtain the WG metric

$$d(\wp_k,\wp_{k+1}) = \min_{\pi_k} [\tilde{d}(\wp_k, m_k, \mathcal{C}_k)]. \tag{22}$$

Second, knowledge of the OMF can be used for obstacles avoidance by minimizing its inner product with $\wp_{k+1}$, such that the Lagrangian is defined as

$$\mathscr{L}(\wp_k, m_k, \mathcal{C}_k) = [\tilde{d}(\wp_k, m_k, \mathcal{C}_k)]^2 + \langle \mathcal{C}_k, m_k \rangle_{\mathcal{W}} \tag{23}$$

where $\langle\cdot,\cdot\rangle_{\mathcal{W}}$ is the inner product over $\mathcal{W}$, and $\mathcal{C}_k = \wp_{k+1}$.

The VLSR performance at the final time depends on the robot distance to the desired PDF, $\wp_f$, and is expressed by the terminal cost

$$\phi[\wp(t_f), \wp_f] = \phi(\wp_T, \wp_f) \triangleq [d(\wp_T, \wp_f)]^2. \tag{24}$$

Then, the VLSR integral cost function can be rewritten as

$$J \triangleq [d(\wp_T,\wp_f)]^2 + \sum_{k=0}^{T-1} \left\{ [\tilde{d}(\wp_k, m_k, \mathcal{C}_k)]^2 + \langle \wp_{k+1}, m_k \rangle_{\mathcal{W}} \right\}$$

and, from (11), the ADOC value functional is given by

$$\mathcal{V}_k(\wp_k, \mathbf{M}_k, \mathcal{C}_k) = \mathscr{L}(\wp_k, m_k, \mathcal{C}_k) + \mathcal{V}_k(\wp_{k+1}, \mathbf{M}_k, \mathcal{C}_k). \tag{25}$$

A. ADOC Value Functional Approximation

As in classical RL-ADP approaches [24], [25], the optimal cost to go, $\mathcal{V}_k^*(\wp_{k+1}, \mathbf{M}_k)$ in the recurrence relationship (13) is unknown and must be approximated over time. In particular, the approximation is obtained here by using an upper bound on the optimal value functional so as to derive convergence guarantees (Section VII). Consider an approximate control law, $\tilde{\mathcal{C}}_k(\wp_\tau, m_k) : \mathscr{P} \mapsto \mathscr{P}$, obtained by holding the number of Gaussian components (N_τ) fixed, such that

$$N_{\tau+1} = N_\tau \tag{26}$$

$$\pi_\tau(\imath,\jmath) = \begin{cases} \omega_\tau^\imath, & \text{if } \imath = \jmath \\ 0, & \text{otherwise} \end{cases}, \quad \imath,\jmath = 1,\ldots,N_\tau \tag{27}$$

$$\wp_{\tau+1} = \tilde{C}_k(\wp_\tau, m_k) \quad \forall \tau : k+1 \leq \tau \leq T-1. \tag{28}$$

By recursively applying the approximate control law, the evolution of the robot PDF can be generated as follows:

$$\wp_\tau = \sum_{\jmath=1}^{N_{k+1}} \omega_{k+1}^{\jmath} g_\tau^{\jmath}, \quad \tau = k+1, \ldots, T \tag{29}$$

according to the N_{k+1} trajectories of the Gaussian components.

Next, consider the L Gaussian components of the desired goal PDF $\wp_f$, denoted by the set $\{g_f^j\}_{j=1}^L$. There are $N_{k+1} \times L$ trajectories of Gaussian components that are characterized by the minimum cost function

$$\tilde{\mathcal{L}}_k^{\jmath,j} = \begin{cases} \min \Big\{ [W_2(g_T^{\jmath}, g_f^j)]^2 \\ + \sum_{\tau=k+1}^{T-1} [W_2(g_\tau^{\jmath}, g_{\tau+1}^{\jmath})]^2 \text{ if } k+1 < T \\ + \sum_{\tau=k+1}^{T-1} \langle g_{\tau+1}^{\jmath}, m_k \rangle \Big\} \\ [W_2(g_T^{\jmath}, g_f^j)]^2, \qquad \text{if } k+1 = T \end{cases} \tag{30}$$

and are indexed by $\jmath$ and j at the kth time step. Then, an upper bound on the optimal ADOC value functional is provided by the following theorem.

Theorem 1 (Upper bound of optimal value functional): Given $\wp_{k+1}$ and $\wp_f$ defined in (19) and (20), respectively, there exists an upper bound of the optimal value functional $\mathcal{V}_k^*(\wp_{k+1}, \mathbf{M}_k)$, which is denoted by $\tilde{\mathcal{V}}_k(\wp_{k+1}, \mathbf{M}_k) = \mathcal{V}_k(\wp_{k+1}, \mathbf{M}_k, \tilde{\mathcal{C}}_k)$, such that

$$\mathcal{V}_k^*(\wp_{k+1}, \mathbf{M}_k) \leq \tilde{\mathcal{V}}_k(\wp_{k+1}, \mathbf{M}_k) \triangleq \sum_{\jmath=1}^{N_{k+1}} \sum_{j=1}^{L} \tilde{\mathcal{L}}_k^{\jmath,j} \tilde{\pi}_k(\jmath, j) \tag{31}$$

where $\tilde{\pi}_k(\jmath, j) \in \Pi(\boldsymbol{\omega}_{k+1}, \boldsymbol{\omega}_f)$ is the robot joint PDF.

The proof of *Theorem* 1 is provided in Appendix A. Hereon, the upper bound in (31) is used to approximate the optimal value functional $\mathcal{V}_k^*(\wp_{k+1}, \mathbf{M}_k)$.

B. Optimal ADOC Control Law

From (13), (25), and (31), the optimal control law functional at time k is given by

$$\begin{aligned} \mathcal{C}_k^* &\approx \arg\min_{\mathcal{C}_k} \Big[\mathscr{L}(\wp_k, m_k, \mathcal{C}_k) + \tilde{\mathcal{V}}_k(\wp_{k+1}, \mathbf{M}_k) \Big] \\ &= \arg\min_{\mathcal{C}_k} \Big\{ [\tilde{d}(\wp_k, m_k, \mathcal{C}_k)]^2 + \langle \mathcal{C}_k, m_k \rangle_{\mathcal{W}} \\ &\quad + \sum_{\jmath=1}^{N_{k+1}} \sum_{j=1}^{L} \tilde{\mathcal{L}}_k^{\jmath,j} \tilde{\pi}_k(\jmath, j) \Big\} \end{aligned} \tag{32}$$

which amounts to a calculus of variations problem. However, because the control law functional is parameterized by the tuple Φ_k, the optimal control functional can be approximated by means of the optimal parameters

$$\begin{aligned} \Phi_k^* = \arg\min_{\Phi_k} \Big\{ & \sum_{\imath=1}^{N_k} \sum_{\jmath=1}^{N_{k+1}} [W_2(g_k^{\imath}, g_{k+1}^{\jmath})]^2 \pi_k(\imath, \jmath) \\ & + \sum_{\imath=1}^{N_k} \sum_{\jmath=1}^{N_{k+1}} \langle g_{k+1}^{\jmath}, m_k \rangle_{\mathcal{W}} \pi_k(\imath, \jmath) + \sum_{\jmath=1}^{N_{k+1}} \sum_{j=1}^{L} \tilde{\mathcal{L}}_k^{\jmath,j} \tilde{\pi}_k(\jmath, j) \Big\} \end{aligned} \tag{33}$$

where the upper bound in (31) is used to approximate the optimal value functional.

Imposing the following constraints on the GMM weights:

$$\omega_k^{\imath} = \sum_{\jmath=1}^{N_{k+1}} \pi_k(\imath, \jmath), \quad \omega_f^j = \sum_{\jmath=1}^{N_{k+1}} \tilde{\pi}_k(\jmath, j) \tag{34}$$

$$\omega_{k+1}^{\jmath} = \sum_{\imath=1}^{N_k} \pi_k(\imath, \jmath) = \sum_{j=1}^{L} \tilde{\pi}_k(\jmath, j) \tag{35}$$

the optimization problem in (33) can be rewritten as

$$\Phi_k^* = \arg\min_{\Phi_k} \sum_{\jmath=1}^{N_{k+1}} \left[\sum_{\imath=1}^{N_k} \mathcal{L}_k^{\imath,\jmath} \pi_k(\imath, \jmath) + \sum_{j=1}^{L} \tilde{\mathcal{L}}_k^{\jmath,j} \tilde{\pi}_k(\jmath, j) \right] \tag{36}$$

where the energy cost associated with the motion of the Gaussian components with respect to the OMF is

$$\mathcal{L}_k^{\imath,\jmath} = [W_2(g_k^{\imath}, g_{k+1}^{\jmath})]^2 + \langle g_{k+1}^{\jmath}, m_k \rangle_{\mathcal{W}}. \tag{37}$$

From Φ_k^*, the optimal robot PDF, $\wp_{k+1}^*$, is approximated by marginalizing over the approximate joint distribution $\hat{\pi}_k^*$.

Substituting $\hat{\wp}_{k+1}^*$, in (23), the Lagrangian can be expressed as a function of the WG metric between $\wp_k$ and $\hat{\wp}_{k+1}^*$

$$\begin{aligned} \mathscr{L}(\wp_k, m_k, \hat{\mathcal{C}}_k^*) &= [\tilde{d}(\wp_k, m_k, \hat{\mathcal{C}}_k^*)]^2 + \langle \hat{\mathcal{C}}_k^*(\wp_k, m_k), m_k \rangle_{\mathcal{W}} \\ &= [d(\wp_k, \hat{\wp}_{k+1}^*)]^2 + \langle \hat{\wp}_{k+1}^*, m_k \rangle_{\mathcal{W}} \end{aligned} \tag{38}$$

and the velocity of the robot PDF in (18) can be approximated using the WG metric

$$\nu_k \approx \frac{d(\wp_k, \hat{\wp}_{k+1}^*)}{\Delta t} \tag{39}$$

and the Lagrangian accounts for the energy consumption E_k.

VI. ADOC Numerical Implementation

An efficient ADOC numerical solution, applicable to online adaptation by the VLSR system, is presented in this section under the following assumption.

Assumption 2: The set of Gaussian components in (19) is a subset of the union set of collocation Gaussian components $G_C = \{g_c^{\jmath}\}_{\jmath=1}^K$ and desired Gaussian components $G_f = \{g_f^j\}_{j=1}^L$

$$\{g_\tau^{\jmath}\}_{\jmath=1}^{N_\tau} \subseteq G, \quad \text{where} \quad G \triangleq G_C \cup G_f \tag{40}$$

provided $k+1 \leq \tau \leq T$.

Based on the assumption above, $\wp_\tau$ belongs to the subspace of the GMM defined as

$$\tilde{\mathcal{G}}(\mathcal{W}, G) \triangleq \left\{ \wp \;\middle|\; \wp = \sum_{\jmath=1}^{L+K} \omega^\jmath g^\jmath, \quad \sum_{\jmath=1}^{L+K} \omega^\jmath = 1, \right.$$
$$\left. 0 \le \omega^\jmath \le 1, \quad g^\jmath \in G, \quad \jmath = 1, \ldots, L+K \right\} \tag{41}$$

where K is number of collocation components, and, thus, $N_{k+1} = L + K$. Then, the metric W_2 in (37) and (30) can be calculated in advance and, since $g^\jmath_\tau \in G$ for all $\tau = k+1, \ldots, T$, the nonlinear program (NLP) in (30) can be solved using a shortest-path algorithm on a nonnegative weighted directed graph [36], [37]. At every time step k, the directed graph, $\mathcal{G} = (G, \mathcal{E})$, is formed by assigning a node to each Gaussian component in G, and by connecting all the nodes pairwise using a set of edges $\mathcal{E}$. The costs associated with the edges are defined in terms of the OMF, such that

$$c_{\imath,\jmath} = \begin{cases} [W_2(g^\imath, g^\jmath)]^2 + \langle g^\jmath, m_k \rangle_{\mathcal{W}} & \text{if } g^\jmath \notin G_f \\ [W_2(g^\imath, g^\jmath)]^2 & \text{if } g^\jmath \in G_f. \end{cases} \tag{42}$$

Then, a shortest path of length of $\tilde{\mathcal{L}}^{\jmath,j}_k$ in (30) can be found that connects node $g^\jmath_{k+1}$ to g^j_f.

From (36), the optimal control law functional can be approximated by solving the following optimization problem:

$$\hat{\pi}^*_k = \arg\min_{\pi_k} \sum_{\jmath=1}^{L+K} \left[\sum_{\imath=1}^{N_k} \mathcal{L}^{\imath,\jmath}_k \pi_k(\imath, \jmath) + \sum_{j=1}^{L} \tilde{\mathcal{L}}^{\jmath,j}_k \tilde{\pi}_k(\jmath, j) \right] \tag{43}$$

and the optimal robot PDF can be approximated by

$$\hat{\wp}^*_{k+1} = \sum_{\imath=1}^{N_k} \left[\sum_{\jmath=1}^{K} \hat{\pi}^*_k(\imath, \jmath) g^\jmath_c + \sum_{j=1}^{L} \hat{\pi}^*_k(\imath, j+K) g^j_f \right]$$
$$= \sum_{\jmath=1}^{K} (\hat{\omega}^\jmath_{k+1})^* g^\jmath_c + \sum_{j=1}^{L} (\hat{\omega}^{j+K}_{k+1})^* g^j_f \tag{44}$$

where the joint probability constraint (35) is applied. Also, for known costs of the Gaussian components motion, $\mathcal{L}^{\imath,\jmath}_k$ and $\tilde{\mathcal{L}}^{\jmath,j}_k$ (for all $\imath$, j, $\jmath$, and k), the approximation of the optimal value functional only depends on π_k and $\tilde{\pi}_k$ and the optimal control functional can be determined using LP algorithms.

From (44), $\hat{\wp}_{k+1}$, is fully specified by $L + K$ weights, many of which are equal to zero or small in magnitude. Therefore, the computational complexity can be significantly reduced by neglecting Gaussian components with weights below a user-specified threshold, and only the remaining components are normalized to approximate the optimal robot PDF.

The computational complexity of the algorithm for approximating the ADOC control law (analyzed in Section VII and shown in Table II) can be further reduced for online implementation by adopting the following assumption.

Assumption 3: The Gaussian components used to approximate the control law $\wp_{\tau+1} = \mathcal{C}_k(\wp_\tau)$, at every time step $\tau = k, \ldots, T-1$, are characterized by a transportation distance below a user-defined positive threshold, d_{th}, such that

$$\mathcal{E} = \{e_{\imath,\jmath} | g^\imath, g^\jmath \in G \text{ and } W_2(g^\imath, g^\jmath) < d_{th}\} \tag{45}$$

and

$$\pi_\tau(\imath, \jmath) = 0 \text{ if } W_2(g^\imath_\tau, g^\jmath_{\tau+1}) > d_{th}$$
$$\imath = 1, \ldots, N_\tau, \text{ and } \jmath = 1, \ldots, L+K \tag{46}$$

where the joint probability π_τ is an $N_\tau \times (L+K)$ matrix, and $g^\imath_\tau$ and $g^\jmath_{\tau+1}$ are the Gaussian components approximating the robot PDF at consecutive times τ and $(\tau+1)$, respectively.

Now, let the index set of the Gaussian components used in the control law approximation be denoted by

$$\mathcal{I}^\imath_\tau = \{\jmath | \jmath \in \mathcal{I},\ g^\jmath \in G, \text{ and } W_2(g^\imath_\tau, g^\jmath) \le d_{th}\} \subset \mathcal{I} \tag{47}$$

where $\mathcal{I} = \{1, \ldots, L+K\}$ is the index set of all components in G. Then, from *Assumption* 3, (43) can be rewritten as

$$\hat{\pi}^*_k = \arg\min_{\pi_k} \sum_{\imath=1}^{N_k} \sum_{j=1}^{L} \left\{ \sum_{\jmath \in \mathcal{I}^\imath_k} \left[\mathcal{L}^{\imath,\jmath}_k \pi_k(\imath, \jmath) + \tilde{\mathcal{L}}^{\jmath,j}_k \tilde{\pi}_k(\jmath, j) \right] \right\} \tag{48}$$

where

$$\pi_k(\imath, \jmath) = 0,\ \imath = 1, \ldots, N_k \text{ and } \jmath \notin \mathcal{I}^\imath_k \tag{49}$$

$$\tilde{\pi}_k(\jmath, j) = 0,\ j = 1, \ldots, L \text{ and } \jmath \in \mathcal{I}^C_k \tag{50}$$

and

$$\mathcal{I}^C_k = \{\jmath | \jmath \in \mathcal{I} \text{ and } \jmath \notin \mathcal{I}_k\}, \quad \mathcal{I}_k \triangleq \cup_{\imath=1}^{N_k} \mathcal{I}^\imath_k \tag{51}$$

such that (34)–(35) are satisfied. As a result, the LP in (48) can be solved using only $|\mathcal{I}_k| \times L$ shortest-paths, where "$|\cdot|$" is the cardinality operator.

Because the robot PDF velocity (39) depends on the collocation Gaussian components, the distance between two sequential robot distributions is divided evenly by a user-defined interval,

TABLE I
PERFORMANCE COMPARISON

Approach	T	Solution Time (min)	$\bar{D}(0)$ (km)	$\bar{E}(T)$ (J/kg)
ADOC	**701**	**37.0544**	**28.5623**	**545.1525**
PDF-APF	2000	60.2447	99.95	1928.0478
SAPF	2000	19.6193	100.8671	3747.0543
SPP	1414	141.2573	41.9044	808.1635

TABLE II
COMPUTATIONAL COMPLEXITY COMPARISON

Algorithm	**Microscopic Control**
ADOC	$O(N \cdot N_k + N \cdot B_k + N^2)$
PDF-APF	$O(N \cdot L + N \cdot B_k + N^2)$
SAPF	$O(N \cdot B_k + N^2)$
SPP	$O(N \cdot B_k + N^2)$
Algorithm	**Path Planning**
ADOC	$O\left[(\lvert\mathcal{E}\rvert + \lvert G\rvert \log \lvert G\rvert) N_k \cdot \lvert\mathcal{I}_k\rvert \cdot L\right]$
SPP	$O\left[(\lvert\mathcal{E}^{SPP}_i\rvert + \lvert G^{SPP}_i\rvert \log \lvert G^{SPP}_i\rvert) N\right]$

D, and other robot distributions are obtained via interpolation. In particular, let the PDF distance $d(\wp_k, \hat{\wp}^*_{k+T_k})$ be divided into $T_k = \lceil d(\wp_k, \hat{\wp}^*_{k+T_k})/D \rceil$ intervals, where "$\lceil\cdot\rceil$" is the ceiling operator. Then, $(T_k - 1)$ robot PDFs are can be obtained via interpolation between each pair of Gaussian components, g^i_k and $g^j_{k+T_k}$, $i = 1, \ldots, N_k$ and $j = 1, \ldots, N_{k+T_k}$, as shown in [32], and the robot PDF velocity can be approximated by

$$\nu_\tau \approx \frac{d(\wp_k, \hat{\wp}^*_{k+T_k})}{T_k \cdot \Delta t} \approx \frac{D}{\Delta t} \triangleq \bar{v}, \quad k \le \tau \le k + T_k. \tag{52}$$

Then, the robot PDFs travel with a relatively smooth velocity field and when $D \ll d(\wp_k, \hat{\wp}^*_{k+T_k})$ the PDF velocity can be treated as a constant.

Furthermore, considering that there are T_k time steps from $\wp_k$ to $\hat{\wp}^*_{k+T_k}$, the costs associated with the edges in (42) is modified as

$$c_{i,j} = \begin{cases} [W_2(g^i, g^j)]^2 + \langle g^j, m_k \rangle_{\mathcal{W}} & \\ + \sum_{n=1}^{T_k - 1} \langle g^n_{i,j}, m_k \rangle_{\mathcal{W}} & \text{if } g^j \notin G_f \\ [W_2(g^i, g^j)]^2 & \text{if } g^j \in G_f \end{cases}$$

where $g^n_{i,j}$, $n = 1, \ldots, T_k - 1$, are obtained via interpolation between g^i and $g^j \in G$.

A. Optimal Robot Control Law

Once the optimal evolution of the VLSR macroscopic state, represented here by the time-varying robot PDF, is obtained at a time step k, it can be used by each robot to compute a local optimal control law. For cooperative robots, the optimal PDF depends on the robot relative positions, and, thus, each robot control law also depends on the positions of other robots in the system. A centralized artificial potential field (APF) approach is adopted here that computes the microscopic robot control inputs for all N robots, represented by the vector $\mathbf{U}_k = [\mathbf{u}^T_{k,1} \cdots \mathbf{u}^T_{k,N}]^T$, so as to meet the desired PDF $\hat{\wp}^*_{k+1}$ at the next time step, based on the observed robot microscopic states $\mathbf{X}_k = [\mathbf{x}^T_{k,1} \cdots \mathbf{x}^T_{k,N}]^T$.

The attractive potential designed to "push" the robots toward the desired PDF is given by

$$U_{\text{att}} = \int_{\mathcal{W}} \left[\hat{\wp}^*_{k+1}(\mathbf{x}) - \gamma \tilde{\wp}_{k+1}(\mathbf{x}; \mathbf{X}_k, \mathbf{U}_k)\right]^2 d\mathbf{x} \tag{53}$$

where $\tilde{\wp}_{k+1}(\cdot)$ is the predicted robot PDF at time $(k+1)$ conditioned on the observed robots' positions and controls ($\mathbf{X}_k$ and $\mathbf{U}_k$). γ is a scalar parameter that determines the scattering strength of the robots chosen, such that $0 < \gamma \le 1$. The KDE method can be used to estimate the robot PDF at the present time (k) based on the VLSR observed position vector $\mathbf{X}_k$ [4], [5].

The repulsive potential is designed to "pull" the robots away from obstacles and from each other, in order to avoid collisions. Let ρ_i denote the minimum Euclidian distance between the robot position $\mathbf{x}_{k,i}$ and the obstacle region $\hat{B}(k)$ estimated from the latest OMF, m_k. Then, the repulsive obstacle potential for the ith robot is

$$U^i_{\text{obs}} = \frac{1}{2}\left(\frac{1}{\rho_i} - \frac{1}{\varrho}\right)^2 \cdot 1(\rho_i, \varrho)$$

where ϱ is a distance threshold used to create a region of influence within which obstacles repel robots, and $1(\rho_i, \varrho)$ is an indicator function that equals one if $\rho_i \le \varrho$, and equals zero otherwise. Also, let $\rho_{i,\ell} = \|\mathbf{x}_{k+1,i} - \mathbf{x}_{k+1,\ell}\|$ represent the distance between predicted robot positions $\mathbf{x}_{k+1,i}$ and $\mathbf{x}_{k+1,\ell}$ $(i \ne \ell)$. Then, a repulsive potential between robots is obtained as follows:

$$U^{i,\ell}_{\text{rob}} = \frac{1}{2}\left(\frac{1}{\rho_{i,\ell}} - \frac{1}{\varphi}\right)^2 \cdot 1(\rho_{i,\ell}, \varphi) \tag{54}$$

where φ is a distance threshold used to create a region of influence within which each robot repels other robots.

Then, the total VLSR potential field can be obtained from a weighted combination of the attractive and repulsive potentials

$$U = w_1 \cdot U_{\text{att}} + w_2 \cdot U_{\text{rep}} \tag{55}$$

where w_1 and w_2 are user-defined weights representing the desired tradeoff between attractive and repulsive objectives, and

$$U_{\text{rep}} = \sum_{i=1}^{N} U^i_{\text{obs}} + \sum_{1 \le i \ne \ell \le N} U^{i,\ell}_{\text{rob}}. \tag{56}$$

Once the potential field in (55) is determined from the desired robot PDF, the robot control inputs at time k are computed according to the (microscopic) control law

$$\mathbf{u}_{k,i} = -\frac{\partial U}{\partial \mathbf{u}_{k,i}}, \quad i = 1, \ldots, N \tag{57}$$

which is designed to minimize the potential function U by a local gradient-based approach, summarized in Algorithm 1.

VII. ADOC Algorithm Analysis

The ADOC approach is designed based on a novel value functional defined in (11), where the policy is updated at each time step, indexed by k. Because the ADOC solution is obtained online, the optimal policy $\mathbf{\Pi}^*_k$ is obtained incrementally by minimizing the value functional in (11) with respect to $\mathcal{C}_k$, where the control law history $\mathbf{\Pi}^*_{k-1}$ is given. This observation, along with the results in the previous section, is used to find a lower bound on the optimal value functional according to the following theorem.

Theorem 2 (Lower bound on optimal value functional): Given the OMF $\mathbf{M}_k$ at the kth time step, the optimal value functional $\mathcal{V}^*_k(\wp_l, \mathbf{M}_k)$ provides a lower bound for all optimal value functionals obtained during previous time steps, such that

$$\mathcal{V}^*_k(\wp_l, \mathbf{M}_k) \le \mathcal{V}^*_q(\wp_l, \mathbf{M}_k), \ 0 \le q \le k \text{ and } 0 \le l \le T$$

for all $\wp_l \in \mathscr{P}(\mathcal{W})$.

The proof of Theorem 2 is provided in Appendix B.

Now, let

$$\tilde{J}(\mathbf{\Pi}_k) \triangleq \mathcal{V}_k(\wp_0, \mathbf{M}_{T-1}), \quad 0 \le k < T \tag{58}$$

denote the ADOC cost function associated with $\mathbf{\Pi}_k$ given $\wp_0$ and $\mathbf{M}_{T-1}$. From (8), it can be shown that

$$J(\wp_{0:T}) = \tilde{J}(\mathbf{\Pi}_{T-1}) = \mathcal{V}_{T-1}(\wp_0, \mathbf{M}_{T-1}) \tag{59}$$

Algorithm 1: ADOC Numerical Implementation.

Initialization:

Initialize the set of K collocation Gaussian components, G_C

Construct the directed graph $\mathcal{G} = (G, \mathcal{E})$

Let $T_k = 0$, $l = 0$, and $k = 0$

Procedure:

while $(k \leq T)$ **and** $(\wp_k \neq \wp_f)$ **do**
  Update m_k using environmental observations
  if $l == T_k$ **then**
    Compute $\cup_{i=1}^{N_k}\{\mathcal{L}_k^{i,j} | j \in \mathcal{I}_k^i\}$ from (37)
    Update $\mathcal{G}$ according to (53)
    Compute $\{\tilde{\mathcal{L}}_k^{j,j} | j \in \mathcal{I}_k$ and $j = 1, \ldots, L\}$ by finding shortest path in $\mathcal{G}$
    Obtain $\hat{\pi}_k^*$ by solving the LP problem in (48)
    Approximate $\hat{\wp}_{k+T_k}^*$ from $\hat{\pi}_k^*$ according to (44)
    Let $T_k = \lceil d(\wp_k, \hat{\wp}_{k+T_k}^*)/D \rceil$
    $l = 0$
    Generate robot PDFs by interpolating between $\wp_k$ and $\hat{\wp}_{k+T_k}^*$
  end if
  Update the interpolation offset step, $l = l + 1$
  Update the next robot PDF, $\hat{\wp}_{k+1}^*$
  Generate $\mathbf{U}_k$ according to (57)
  Update $\mathbf{X}_{k+1}$ according to (5) and (6)
  Update time step, $k = k + 1$
end while

and that the ADOC cost function associated with $\mathbf{\Pi}_k^*$ for $0 \leq k < T$ converges to an upper bound of the minimum cost function $J(\wp_{0:T}^*)$, as stated in the following corollary.

Corollary 2.1: For any two ADOC optimal policies, $\mathbf{\Pi}_k^*$ and $\mathbf{\Pi}_q^*$, where $0 \leq q \leq k \leq (T-1)$, the following inequalities hold:

$$J(\wp_{0:T}^*) \leq \tilde{J}(\mathbf{\Pi}_{T-1}^*) \leq \tilde{J}(\mathbf{\Pi}_k^*) \leq \tilde{J}(\mathbf{\Pi}_q^*) \tag{60}$$

and, thus, it follows that the optimal ADOC cost function $\tilde{J}(\mathbf{\Pi}_k^*)$ converges monotonically to $\tilde{J}(\mathbf{\Pi}_{T-1}^*)$, which is an upper bound of the optimal cost function.

Corollary 2.1 can be easily proven from (58) and *Theorem* 2, by applying the optimal control law functional in (13). $J(\wp_{0:T}^*)$ is obtained by minimizing (8) with respect to $\wp_{0:T}$, while the policy $\mathbf{\Pi}_k^*$ is improved incrementally by updating only one control law functional at every time step, up to time $(T-1)$. Hence, $\mathbf{\Pi}_{T-1}^*$ is a suboptimal solution.

Furthermore, the ADOC approach relies on approximating the upper bound of the value functional (*Theorem* 1). The optimal control law is approximated at each time step by solving (36). To reduce the computational complexity and obtain a tractable solution, the upper bound of the value functional is calculated only once per OMF, and is not updated iteratively as in conventional ADP approaches. Finally, the performance of the ADOC approach also depends on the choice of collocation Gaussian components. Although the uniform grid adopted in this article provides good performance, adaptive and multiresolution methods will be the subject of future research.

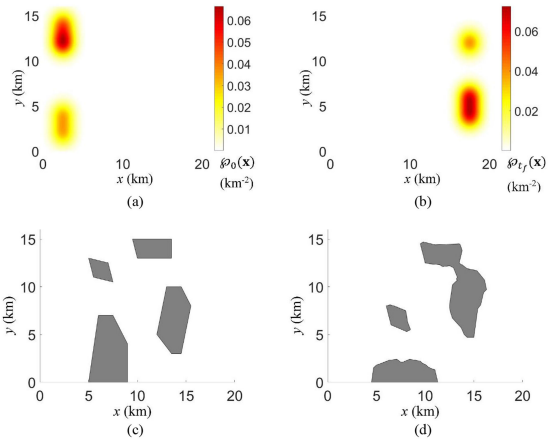

Fig. 1. VLSR system must travel from the initial robot distribution in (a) to the goal robot distribution in (b) avoiding obstacles sensed *in situ*, starting with the initial OMF (c) and learning the actual OMF (d) online.

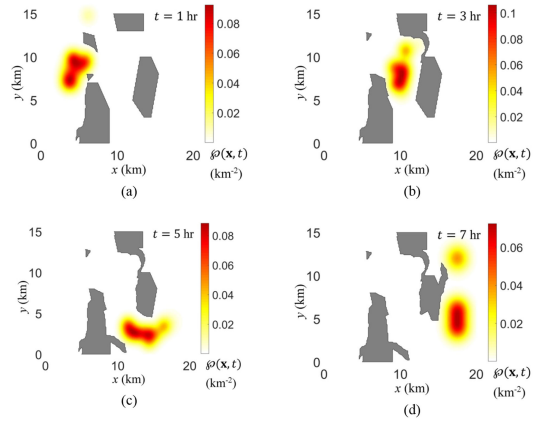

Fig. 2. Evolution of robot PDF optimized by ADOC, shown at four sample moments in time, where the gray regions represent the obstacle map updated online based on onboard sensor measurements.

VIII. Simulations and Results

The effectiveness of the ADOC approach is demonstrated on a VLSR system comprised of $N = 500$ mobile robots characterized by single-integrator dynamics

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t), \quad \mathbf{x}_i(t_0) = \mathbf{x}_{i_0}, \quad i = 1, \ldots, N \tag{61}$$

where $\mathbf{x}_i = [x_i \quad y_i]^T$ is the robot position, x_i and y_i are the robot xy-coordinates in inertial frame, and the control input $\mathbf{u}_i$ is a vector of linear velocities in the x- and y-directions. The VLSR system must travel from a given initial distribution $\wp_0 = \sum_{i=1}^{Q} \omega_0^i g_0^i$ to a desired distribution $\wp_f = \sum_{j=1}^{L} \omega_f^j g^j$ while avoiding collisions with partially unknown or uncertain obstacles in the ROI, $\mathcal{W} = [0, W] \times [0, H]$, where $Q = 4$, $L = 3$, $W = 20$ km, and $H = 16$ km.

The initial and desired robot PDFs are shown in Fig. 1(a) and (b), respectively. At the initial time t_0, 500 robots characterized

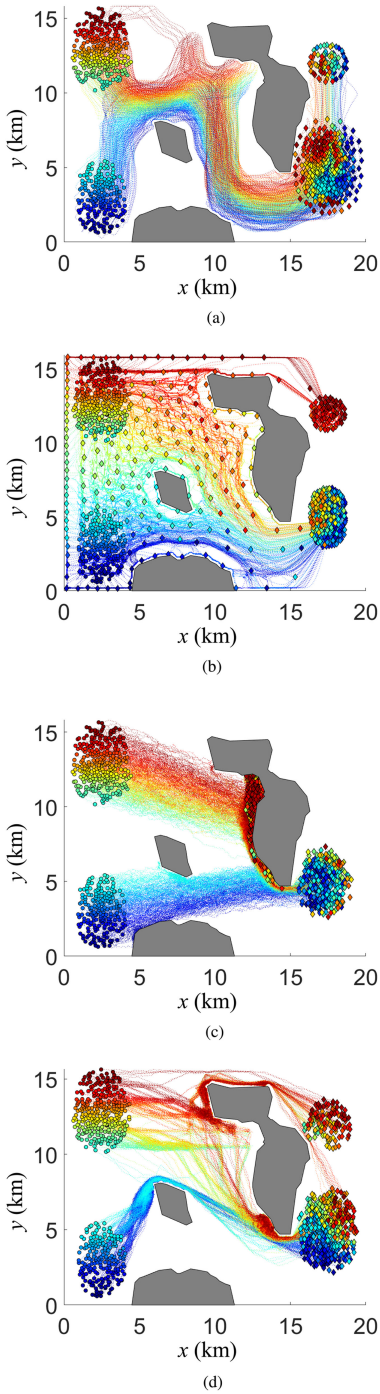

Fig. 3. Robot trajectories and final positions (diamonds) obtained by (a) ADOC, (b) PDF-APF, (c) SAPF, and (d) SPP, for the same set of initial positions (circles), and the same time-varying OMF (grey region).

by the distribution $\wp_0$ [Fig. 3(a)] begin traveling through the ROI based on the OMF shown in Fig. 1(c). The actual obstacle region, shown in Fig. 1(d) is not yet available to them and, for example, the robots are unaware of the blocked passage in the upper-right corner of the ROI. Using an onboard range sensor characterized by an FOV with $r = 1$ km, the robots are able to update their knowledge of the obstacles and fuse their measurements to update the OMF incrementally over time.

An online ADOC solution to this VLSR path planning problem is obtained by using the set of K collocation Gaussian components

$$G_c = \Big\{ g_c = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \Big| \boldsymbol{\mu} = [\xi - 0.5,\ \zeta - 0.5]\text{km} \quad \text{for } \xi = 1, 2, \ldots, W \text{ and } \zeta = 1, 2, \ldots, H \quad \text{and } \boldsymbol{\Sigma} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \text{km}^2 \Big\}. \tag{62}$$

The component distance threshold is chosen as $d_{th} = 4$ km, and time is discretized using an interval $\Delta t = 0.01$ hr. A relatively constant distribution velocity in (52) is obtained by letting $D = 0.05$ km. Finally, the individual (microscopic) control law is obtained by constructing attractive and repulsive potentials with user-defined parameters $\gamma = 0.85$, $\varrho = 0.3$ km, and $\varphi = 0.1$ km.

An example of ADOC solution is plotted in Fig. 2 along with the OMF, which is updated based on *in situ* measurements at every time step Δt and is shown by the gray obstacle region. It can be seen in Fig. 2(b) that, after approximately three hours, the blocked passage is observed by robots. Thanks to the ADOC approach, the VLSR system is able to adapt in real time and find a new optimal solution that allows the robots to reach the desired distribution efficiently [Fig. 2(c)–(d)].

A. Performance Comparison

To the best of the authors' knowledge none of the existing VLSR methods are applicable to the online planning and control problem tackled in this article (described in Section II). For comparison, three methods are developed by extending state-of-the-art techniques referred to as PDF-based APF (PDF-APF), sampling-based APF (SAPF), and sampling-based path-planning (SPP). In the PDF-APF approach, an attractive potential field is generated by replacing $\hat{\wp}^*_{k+1}(\mathbf{x})$ in (53) with the desired PDF $\wp_f$. Subsequently, APF robot control laws are obtained by means of the gradient-descent method shown in (57), based on the same online OMF used by the ADOC algorithm.

In the SAPF approach, the desired final robot positions, denoted by the set $\mathcal{X}_f = \{\mathbf{x}_{f,i}\}_{i=1}^N$, are first obtained by sampling $\wp_f$. Subsequently, since the robots are interchangeable, the desired positions are used to generate individual robot attractive potentials, and, thus, to control the robots independently of each other. In the SPP approach, the robot positions in the set $\mathcal{X}_f = \{\mathbf{x}_{f,i}\}_{i=1}^N$ are paired with the initial robot positions in $\mathcal{X}_0 = \{\mathbf{x}_i(t_0)\}_{i=1}^N$ based on the shortest relative distance. Subsequently, a shortest-path algorithm is used at every time step (k) to find the best robot trajectory based on the latest OMF, m_k. All methods avoid collisions by means of the repulsive potentials in (56). For comparison, the robot velocities are chosen to abide to $\|\mathbf{u}_i\| = 5$ km/hr for all i, although in principle they too could be optimized via ADOC in order to minimize energy consumption. Similarly, the same user-defined parameters, ϱ, φ, and d_{th}, are used in all four methods and the same maximum time $T_f = 2000$ is adopted.

Fig. 3(a)–(d) shows the VLSR trajectories obtained for the 500 robots using ADOC, PDF-APF, SAPF, and SPP. It can be

seen that using ADOC and SPP the robots are able to reach the goal distribution in the allotted time, while with PDF-APF and SAPF many of the robots fall behind or remain stuck nearby some of the obstacles found *in situ*. Table I shows the VLSR performance obtained by the four methods . All of the simulations are conducted on the same computer with an 18-core CPU and 16 G RAM. It can be seen that the solution time, T, required by the robots to reach the goal PDF is significantly lower for ADOC than for all other methods. The total runtime is also significantly reduced by ADOC despite its ability to optimize the VLSR system performance (not afforded by other methods). The optimized ADOC performance can be assessed by computing the average cost-to-go

$$\bar{D}(k) = \frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=k}^{T-1}\|\mathbf{x}_i\left[(\tau+1)\Delta t\right] - \mathbf{x}_i(\tau\Delta t)\| \tag{63}$$

and the average energy-cost per kg

$$\bar{E}(k) = \frac{\eta}{2\,N}\sum_{i=1}^{N}\sum_{\tau=1}^{k}\left[\frac{\|\mathbf{x}_i(\tau\Delta t) - \mathbf{x}_i\left[(\tau-1)\Delta t\right]\|}{\Delta t}\right]^2 \tag{64}$$

where η is a unit-conversion factor. As shown in Table I, despite using the same knowledge of the obstacle region, by leveraging new optimality conditions derived from Bellman's optimality principle, the ADOC approach allows the cooperative robots to minimize time, distance traveled, and energy consumption.

B. Computational Complexity Comparison

The results shown in Table I show that the time required to obtain the ADOC solution is significantly lower than the time required by other methods of solution. This section derives the computational complexity of each algorithm, summarized in Table II. As a first step, the computational complexity required by the construction of the artificial potentials at the kth time step is analyzed. Let B_k denote the number of collocation points evenly sampled on the updated OMF contours at the kth time step, as well as on the ROI boundaries, $\partial\mathcal{W}$. Then, the (microscopic) robot control law computation at the kth time step requires the times shown in Table II, based on both repulsive and attractive potential fields. Because $N_k \ll N$ and $L \ll N$, computing the robot control law requires time $O(N^2)$ for all four methods. The square power results from the artificial potential calculation and, therefore, when robots are outside the potential's region of influence the computational complexity decreases to $O(N)$.

From [36] and [37], the computational complexity of searching a directed graph $\mathcal{G} = (G, \mathcal{E})$ for the shortest path connecting two of its nodes is $O(|\mathcal{E}| + |G|\log|G|)$. Letting $\mathcal{E}_i^{\text{SPP}}$ and G_i^{SPP} denote the set of arcs and nodes generated by the SPP algorithm for every robot i, the computation required by the path-planning ADOC and SPP methods at every time step k is found to be as shown in Table II. Therefore, when $N_k \cdot |\mathcal{I}_k| \cdot L < N$, ADOC requires significantly less computation than SPP, because typically $L \ll K$. Furthermore, since N_k, $|\mathcal{I}_k|$, and L are independent of the number of robots (N), the ADOC approach developed in this article is scalable to very large systems of cooperative robots.

IX. Conclusion

This article develops a novel adaptive optimal control approach, referred to as ADOC, that is applicable to online cooperative VLSR systems applications, such as sensing and navigation. Unlike existing adaptive dynamic programming and adaptive control approaches, ADOC is developed to solve environmental adaptation problems in which the system performance, represented by the Lagrangian of the cost function, changes over time due to *in situ* measurements and observations. By optimizing the spatio-temporal evolution of the VLSR macroscopic state, such as the robot PDF, the ADOC approach provides online solutions that scale up to very large numbers of cooperative robots. The novel technical contributions in this article show that the online adaptive control of multiscale dynamical systems can be formulated as a new adaptive dynamic programming problem in the Wasserstein-GMM space, thus allowing for the application of OMT theory. The numerical simulation results presented in this article show that ADOC not only outperforms other VLSR planning methods, by optimizing the macroscopic system performance incrementally over time but it also reduces the solution time and the total amount of time required by the robots to complete the desired task.

Appendix A
Upper Bound of Optimal Value Functional

Proof: First, consider the case of $k+1 < T$. According to (25), the value functional $\mathcal{V}_k(\wp_{k+1}, m_k, \tilde{\mathcal{C}}_k)$ associated to $\tilde{\mathcal{C}}_k$, which is described in (26) and (27), can be expressed by

$$\mathcal{V}_k(\wp_{k+1}, \mathbf{M}_k, \tilde{\mathcal{C}}_k) = \left[d(\wp_T, \wp_f)\right]^2 + \sum_{\tau=k+1}^{T-1}\left[\tilde{d}(\wp_\tau, m_k, \tilde{\mathcal{C}}_k)\right]^2 + \sum_{\tau=k+1}^{T-1}\langle\wp_{\tau+1}, m_k\rangle_{\mathcal{W}}. \tag{65}$$

According to (22), the following inequality is obtained:

$$\left[d(\wp_T, \wp_f)\right]^2 \le \sum_{\jmath=1}^{N_{k+1}}\sum_{j=1}^{L}\left[W_2(g_T^\jmath, g^j)\right]^2\tilde{\pi}_k(\jmath, j). \tag{66}$$

By recursively applying (26) and (27), the term, $[\tilde{d}(\wp_\tau, m_k, \tilde{\mathcal{C}}_k)]^2$, $k+1 \le \tau \le T-1$, can be expressed as

$$\begin{aligned}\left[\tilde{d}(\wp_\tau, m_k, \tilde{\mathcal{C}}_k)\right]^2 &= \sum_{\jmath=1}^{N_{k+1}}\sum_{\imath=1}^{N_{k+1}}[W_2(g_\tau^\jmath, g_{\tau+1}^\imath)]^2\pi_k(\jmath, \imath)\\ &= \sum_{\jmath=1}^{N_{k+1}}\left[W_2(g_\tau^\jmath, g_{\tau+1}^\jmath)\right]^2\omega_{k+1}^\jmath\\ &= \sum_{\jmath=1}^{N_{k+1}}\sum_{j=1}^{L}\left[W_2(g_\tau^\jmath, g_{\tau+1}^\jmath)\right]^2\tilde{\pi}_k(\jmath, j).\end{aligned} \tag{67}$$

Substituting (66) and (67) into (65), one can have

$$\mathcal{V}_k(\wp_{k+1}, \mathbf{M}_k, \tilde{\mathcal{C}}_k) \leq \sum_{\jmath=1}^{N_{k+1}} \sum_{j=1}^{L} \Bigg\{ \left[W_2(g_T^\jmath, g^j)\right]^2 + \sum_{\tau=k+1}^{T-1} \left[W_2(g_\tau^\jmath, g_{\tau+1}^\jmath)\right]^2 + \sum_{\tau=k+1}^{T-1} \langle g_\tau^\jmath, m_k \rangle_{\mathcal{W}} \Bigg\} \tilde{\pi}_k(\jmath, j). \tag{68}$$

Because (68) holds for any trajectories of Gaussian components from $g_{k+1}^\jmath$ to κ^j, $\jmath = 1, \ldots, N_{k+1}$ and $j = 1, \ldots, L$, the following inequality can be obtained:

$$\begin{aligned}\mathcal{V}_k(\wp_{k+1}, \mathbf{M}_k, \tilde{\mathcal{C}}_k) &\leq \sum_{\jmath=1}^{N_{k+1}} \sum_{j=1}^{L} \min \Bigg\{ \left[W_2(g_T^\jmath, g^j)\right]^2 + \sum_{\tau=k+1}^{T-1} \left[W_2(g_\tau^\jmath, g_{\tau+1}^\jmath)\right]^2 + \sum_{\tau=k+1}^{T-1} \langle g_\tau^\jmath, m_k \rangle_{\mathcal{W}} \Bigg\} \tilde{\pi}_k(\jmath, j) \\ &= \sum_{\jmath=1}^{N_{k+1}} \sum_{j=1}^{L} \tilde{\mathcal{L}}_k^{\jmath,j} \tilde{\pi}_k(\jmath, j). \end{aligned} \tag{69}$$

Next, consider the case of $k+1 = T$. According to (66), the upper bound of $\mathcal{V}_k(\wp_T, \mathbf{M}_k, \tilde{\mathcal{C}}_k)$ is expressed by

$$\begin{aligned}\mathcal{V}_k(\wp_T, \mathbf{M}_k, \tilde{\mathcal{C}}_k) &\leq \sum_{\jmath=1}^{N_{k+1}} \sum_{j=1}^{L} \left[W_2(g_T^\jmath, g^j)\right]^2 \tilde{\pi}_k(\jmath, j) \\ &= \sum_{\jmath=1}^{N_{k+1}} \sum_{j=1}^{L} \tilde{\mathcal{L}}_k^{\jmath,j} \tilde{\pi}_k(\jmath, j). \end{aligned} \tag{70}$$

Finally, considering the definition of the optimal value functional, the theorem is proved.

Appendix B
Lower Bound of Optimal Value Functional

Proof: First, consider the case of $q = k-1$, $0 < k < T$, and $k \leq l < T$. From any robot PDF $\wp_l$, by using the optimal control law functional obtained at the qth time step, $\mathcal{C}_q^*$, recursively, a trajectory of robot PDFs $\{\wp_\tau\}_{\tau=l}^T$ are generated associated with the OMF m_k.

In addition, consider the optimal value functional $\mathcal{V}_k^*(\wp_l, \mathbf{M}_k)$, $k \leq l < T$. According to (11) and the Bellman equation, $\mathcal{V}_k^*(\wp_l, \mathbf{M}_k)$ can be expressed by

$$\begin{aligned}\mathcal{V}_k^*(\wp_l, \mathbf{M}_k) &= \min_{\wp_{l+1}} \left[\mathscr{L}(\wp_l, m_k, \wp_{l+1}) + \mathcal{V}_k^*(\wp_{l+1}, \mathbf{M}_k)\right] \\ &= \mathscr{L}(\wp_l, m_k, \wp_{l+1}^*) + \mathcal{V}_k^*(\wp_{l+1}^*, \mathbf{M}_k) \\ &\leq \mathscr{L}(\wp_l, m_k, \wp_{l+1}) + \mathcal{V}_k^*(\wp_{l+1}, \mathbf{M}_k) \end{aligned} \tag{71}$$

where $\wp_{l+1} = \mathcal{C}_q^*(\wp_l, m_k)$ and $\wp_{l+1}^* = \mathcal{C}_k^*(\wp_l, m_k)$.

By recursively utilizing (71), the following inequality is obtained:

$$\begin{aligned}\mathcal{V}_k^*(\wp_l, \mathbf{M}_k) &\leq \mathscr{L}(\wp_l, m_k, \wp_{l+1}) + \mathcal{V}_k^*(\wp_{l+1}, \mathbf{M}_k) \\ &\leq \mathscr{L}(\wp_l, m_k, \wp_{l+1}) + \mathscr{L}(\wp_{l+1}, m_k, \wp_{l+2}) + \mathcal{V}_k^*(\wp_{l+2}, \mathbf{M}_k) \\ &\cdots \\ &\leq \sum_{\tau=l}^{T-1} \mathscr{L}(\wp_\tau, m_k, \wp_{\tau+1}) + \mathcal{V}_k^*(\wp_T, \mathbf{M}_k) \\ &= \sum_{\tau=l}^{T-1} \mathscr{L}(\wp_\tau, m_k, \wp_{\tau+1}) + \mathcal{V}_q^*(\wp_T, \mathbf{M}_k) \\ &= \mathcal{V}_q^*(\wp_l, \mathbf{M}_k), \ k \leq l < T \end{aligned} \tag{72}$$

where $\mathcal{V}_k^*(\wp_T, \mathbf{M}_k) = \mathcal{V}_q^*(\wp_T, \mathbf{M}_k) = [d(\wp_T, \wp_f)]^2$ according to (11). Because this holds for $l = T$ as well, according to (11), (72) can be rewritten by

$$\mathcal{V}_k^*(\wp_l, \mathbf{M}_k) \leq \mathcal{V}_q^*(\wp_l, \mathbf{M}_k), \ k \leq l \leq T. \tag{73}$$

Next, consider the case of $q = k-1$ and $0 \leq l \leq q$. Assume that the optimal policy $\mathbf{\Pi}_q$ exists, which includes a sequence of optimal control law functionals $\mathcal{C}_\tau^*$, $\tau = l, \ldots, q$. By using these optimal control law functionals sequentially, a trajectory of robot PDFs $\{\wp_\tau\}_{\tau=l}^q$ is generated associated with $\{m_\tau\}_{\tau=l}^q$. Moreover, similarly, by using the optimal control law functional $\mathcal{C}_q^*$ recursively, a trajectory of robot PDFs $\{\wp_\tau\}_{\tau=q}^T$ is generated from $\wp_q$ to $\wp_T$ associate with m_k. Thus, a trajectory of robot PDFs, $\{\wp_\tau\}_{\tau=l}^T$ is generated from $\wp_l$ to $\wp_T$.

Again, according to (11) and the Bellman equation, $\mathcal{V}_k^*(\wp_l, \mathbf{M}_k)$ can be expressed by

$$\begin{aligned}\mathcal{V}_k^*(\wp_l, \mathbf{M}_k) &= \sum_{\tau=l}^{q} \mathscr{L}(\wp_\tau, m_\tau, \mathcal{C}_\tau^*) + \mathcal{V}_k^*(\wp_k, \mathbf{M}_k) \\ &\leq \sum_{\tau=l}^{q} \mathscr{L}(\wp_\tau, m_\tau, \mathcal{C}_\tau^*) + \mathcal{V}_q^*(\wp_k, \mathbf{M}_k) \\ &= \mathcal{V}_q^*(\wp_l, \mathbf{M}_k), \ 0 \leq l \leq q \end{aligned} \tag{74}$$

where the inequality is obtained by applying (73). Merging (73) and (74), it is shown that for $q = k-1$

$$\mathcal{V}_k^*(\wp_l, \mathbf{M}_k) \leq \mathcal{V}_q^*(\wp_l, \mathbf{M}_k), \ 0 \leq l \leq T. \tag{75}$$

Finally, by recursively applying (75), the theorem is proved.

References

[1] M. Steinberg, J. Stack, and T. Paluszkiewicz, "Long duration autonomy for maritime systems: Challenges and opportunities," *Auton. Robots*, vol. 40, no. 7, pp. 1119–1122, 2016.

[2] L. E. Parker, "Path planning and motion coordination in multiple mobile robot teams," in *Proc. Encyclopedia Complexity Sys. Sci*., New York, NY, USA: Springer, 2009, pp. 5783–5800.

[3] G. Foderaro, S. Ferrari, and T. A. Wettergren, "Distributed optimal control for multi-agent trajectory optimization," *Automatica*, vol. 50, no. 1, pp. 149–154, 2014.

[4] S. Ferrari, G. Foderaro, P. Zhu, and T. A. Wettergren, "Distributed optimal control of multiscale dynamical systems: A tutorial," *IEEE Control Syst. Mag.*, vol. 36, no. 2, pp. 102–116, Apr. 2016.

[5] K. Rudd, G. Foderaro, P. Zhu, and S. Ferrari, "A generalized reduced gradient method for the optimal control of very-large-scale robotic systems," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1226–1232, Oct. 2017.

[6] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," 2018, *arXiv:1802.05438*.

[7] A. Khan, C. Zhang, D. D. Lee, V. Kumar, and A. Ribeiro, "Scalable centralized deep multi-agent reinforcement learning via policy gradients," 2018, *arXiv:1805.08776*.
[8] M. Hüttenrauch *et al.*, "Deep reinforcement learning for swarm systems," *J. Mach. Learn. Res.*, vol. 20, no. 54, pp. 1–31, 2019.
[9] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Auton. Agents Multi-Agent Syst.*, vol. 33, no. 6, pp. 750–797, 2019.
[10] R. Carmona, M. Laurière, and Z. Tan, "Linear-quadratic mean-field reinforcement learning: Convergence of policy gradient methods," 2019, *arXiv:1910.04295*.
[11] R. Carmona, M. Laurière, and Z. Tan, "Model-free mean-field reinforcement learning: Mean-field MDP and mean-field Q-learning," 2019, *arXiv:1910.12802*.
[12] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," 2019, *arXiv:1911.10635*.
[13] W. Ren, "Represented value function approach for large scale multi agent reinforcement learning," 2020, *arXiv:2001.01096*.
[14] C. Ferrari, E. Pagello, J. Ota, and T. Arai, "Multirobot motion coordination in space and time," *Robot. Auton. Syst.*, vol. 25, no. 3/4, pp. 219–229, 1998.
[15] M. Bennewitz, W. Burgard, and S. Thrun, "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," *Robot. Auton. Syst.*, vol. 41, no. 2/3, pp. 89–99, 2002.
[16] M. Yogeswaran and S. Ponnambalam, "Swarm robotics: An extensive research review," in *Proc. Adv. Know. App. Practice*. London, U.K.: IntechOpen, 2010.
[17] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.
[18] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using optimal transport," in *Proc. IEEE Conf. Control Appl.*, 2014, pp. 498–505.
[19] V. Krishnan and S. Martínez, "Distributed optimal transport for the deployment of swarms," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 4583–4588.
[20] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.
[21] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm Evol. Comput.*, vol. 50, 2019, Art no. 100565.
[22] M. Huang *et al.*, "Large population stochastic dynamic games: Closed-loop Mckean–Vlasov systems and the Nash certainty equivalence principle," *Commun. Inf. Syst.*, vol. 6, no. 3, pp. 221–252, 2006.
[23] M. Huang, P. E. Caines, and R. P. Malhamé, "The NCE (mean field) principle with locality dependent cost interactions," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2799–2805, Dec. 2010.
[24] S. Ferrari and R. F. Stengel, "An adaptive critic global controller," in *Proc. Amer. Control Conf.*, 2002, pp. 2665–2670.
[25] S. Ferrari and R. Stengel, "Model-based adaptive critic designs," in *Proc. Learn. Approx. Dynam. Program.*. Hoboken, NJ, USA: Wiley-IEEE Press, 2004.
[26] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1717–1730, 2016.
[27] P. Zhu, S. Ferrari, J. Morelli, R. Linares, and B. Doerr, "Scalable gas sensing, mapping, and path planning via decentralized hilbert maps," *Sensors*, vol. 19, no. 7, 2019, Art. no. 1524.
[28] P. Zhu, H. Wei, W. Lu, and S. Ferrari, "Multi-kernel probability distribution regressions," in *Proc. Int. Joint Conf. Neural Netw.*, 2015.
[29] P. Zhu, J. Morelli, and S. Ferrari, "Value function approximation for the control of multiscale dynamical systems," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 5471–5477.
[30] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only Q-learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2134–2144, Oct. 2016.
[31] C. Villani, *Topics in Optimal Transportation*. Providence, RI, USA: Amer. Math. Soc., 2003.
[32] Y. Chen, T. T. Georgiou, and A. Tannenbaum, "Optimal transport for Gaussian mixture models," *IEEE Access*, vol. 7, pp. 6269–6278, 2018.
[33] V. M. Panaretos and Y. Zemel, "Statistical aspects of Wasserstein distances," *Annu. Rev. Statist. Appl.*, vol. 6, pp. 405–431, 2019.
[34] G. Auricchio, F. Bassetti, S. Gualandi, and M. Veneroni, "Computing Kantorovich–Wasserstein distances on d-dimensional histograms using $(d+1)$-partite graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5793–5803.
[35] M. Kristan, A. Leonardis, and D. Skočaj, "Multivariate online kernel density estimation with Gaussian kernels," *Pattern Recognit.*, vol. 44, no. 10/11, pp. 2630–2642, 2011.
[36] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987.
[37] M. Thorup, "Integer priority queues with decrease key in constant time and the single source shortest paths problem," *J. Comput. Syst. Sci.*, vol. 69, no. 3, pp. 330–353, 2004.

Pingping Zhu (Member, IEEE) received the B.S. degree in electronics and information engineering and the M.S. degree from the Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2008, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2013.

He is currently an Assistant Professor with the Department of Computer Science and Electrical Engineering, Marshall University, Huntington, WV, USA. He was a Research Associate with the Department of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, USA. His research interests include approximate dynamic programming, reinforcement learning, signal processing, information theoretical learning, machine learning, artificial intelligence, and neural networks.

Chang Liu received the B.S. degrees (double major) in electrical engineering and applied mathematics from Peking University, Beijing, China, in 2011, the M.S. degrees in mechanical engineering and computer science, and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2014, 2016, and 2017, respectively.

He is currently a Postdoctoral Associate with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, USA. His research interests include planning and decision-making of robots, multiagent systems, state estimation and prediction, computer vision, and human–robot collaboration.

Silvia Ferrari (Senior Member, IEEE) received the B.S. degree from Embry–Riddle Aeronautical University, Daytona Beach, FL, USA, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, USA.

She is currently a John Brancaccio Professor of Mechanical and Aerospace Engineering with Cornell University, Ithaca, NY, USA. She was a Professor of Engineering and Computer Science with Duke University, Durham, NC, USA, and the Founder and Director of the NSF Integrative Graduate Education and Research Traineeship and the Fellowship program on Wireless Intelligent Sensor Networks. Her research interests include robust adaptive control of aircraft, learning and approximate dynamic programming, and optimal control of mobile sensor networks.

Dr. Ferrari is a Fellow of American Society of Mechanical Engineers, an Associate Fellow of the American Institute of Aeronautics and Astronautics, and a Member of Society of Photo-Optical Instrumentation Engineers and Society for Industrial and Applied Mathematics. She was the recipient of the Office of Naval Research Young Investigator Award in 2004, the National Science Foundation CAREER Award in 2005, and the Presidential Early Career Award for Scientists and Engineers Award, in 2006.