2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Value Function Approximation for the Control of Multiscale Dynamical Systems

Pingping Zhu, *Member, IEEE,* Julian Morelli, *Student Member, IEEE,* and Silvia Ferrari, *Senior Member, IEEE*

***Abstract*— This paper presents a novel approximate dynamic programming (ADP) algorithm for the optimal control of multiscale dynamical systems comprised of many interacting agents. The ADP algorithm presented in this paper is obtained using a distributed optimal control approach by which the performance of the multiscale dynamical system is represented in terms of a macroscopic state, and is optimized subject to a macroscopic description provided by the continuity equation. A value function approximation scheme is proposed and tested using a data set obtained by solving the necessary conditions for optimality for the distributed optimal control problem. The results shows that the proposed approximation method can learn the value function accurately and, thus, may be applied to adapt the optimal control law.**

I. INTRODUCTION

The control and stability analysis of a formation of a large number of robots, or swarm, has been investigated by several authors and various proposed methods were successfully implemented [1]–[6]. Achieving and maintaining a desired configuration or behavior, such as translating as a group (schooling), or maintaining the center of mass of the group stationary (flocking), was illustrated by conveniently describing the group of interacting agents by probability density functions (PDFs). These methods, however, have yet to provide an approach such that the agents performance is optimized. Optimal control and approximate dynamic programming are currently the most general and effective approaches for optimizing the performance of one or few dynamical systems over time [7]–[10], but their application to multi-agent systems has been very limited due to the computation required to solve the optimal control problem for N coupled dynamical systems. For example, it was shown that optimizing the trajectories of N agents in an obstacle-populated environment is polynomial-space-hard (PSPACE-hard) in N [11].

Distributed optimal control (DOC) has recently overcome this computational complexity to effectively solve multi-agent trajectory optimization problems by seeking to optimize a *restriction operator*, such as a PDF and/or its lower moments, that capture the agents performance on large spatial and temporal scales [12], [13]. By this approach, the agents' macroscopic behavior, or *coarse dynamic equation*, can be represented by partial differential equations (PDEs), such as the continuity equation known as the advection equation [12], [13].

Considerable research efforts have focused on the optimal control and estimation of PDEs and stochastic differential equations (SDEs) driven by non-Gaussian processes, such as Brownian motion combined with Poisson processes, and various other stochastic processes [14]–[16]. The microscopic agent state is viewed as a random vector and the dynamic equation, which takes the form of an SDE that involves the evolution of the statistics of the microscopic vector function, can be integrated using stochastic integrals [14], [16]. The performance of N agents can therefore be expressed as an integral function of N vector fields to be optimized subject to N SDEs. Solutions obtained using this approach, however, are only for relatively few and highly idealized cases in which finite-dimensional, local approximations can be constructed (e.g. via moment closure [14], [15]). As a result, while optimal control of SDEs can be useful to selected applications in population biology and finance [14]–[16], it has yet to be successfully applied to multiscale dynamical systems, as their coarse dynamics do not obey these idealized conditions. Rather, they are dictated by realistic constraints, such as vehicle dynamics, and practical objectives, such as minimizing energy consumption or maximizing sensing performance.

Although DOC is ideally suited to adaptive, robust control of multiscale dynamical systems, all DOC methods to date assume that the microscopic agent dynamics, the macroscopic evolution equation, and the definition of the restriction operator are known *a priori* [13], [17].In most applications, however, these models are subject to significant sources of error, or may not be available in closed form or only in the form of a computer simulation. Approximate dynamic programming (ADP) is not affected by these limitations. Therefore, developing an ADP method associated with a DOC problem is extremely valuable. Unlike existing DOC approaches [12], [13], which assume the system dynamics are known *a priori*, the ADP approach presented can be used to learn the optimal control law and value function online, based on observations of the state obtained from the real system or its computer simulation. This method is called distributed ADP of multiscale dynamical systems or distributed ADP for short.

In this paper, an ADP approach is developed by deriving the policy-improvement routine and the value-determination

Pingping Zhu is with Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, US pingping.zhu@cornell.edu
Julian Morelli is with Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, US jam888@cornell.edu
Silvia Ferrari is with Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, US ferrari@cornell.edu

operation for DOC problems. The latter is used to learn and update the value function of the multiscale dynamical system. Because the value function of the DOC problem is a functional expressed in terms of the macroscopic state, in this case a PDF, a parametric method based on the Gaussian mixture model (GMM) [18], [19] is applied to perform the functional regression.

The main novelty of this paper resides in the value functional approximation scheme. The value functional is approximated by transforming the PDF representing the collection of agents into a Reproducing Kernal Hilbert Space (RKHS) and then performing functional regression in RKHS. This allows our value function, which is highly non-linear, to be learned by existing reinforcement learning algorithms, such as least squares temporal difference (LSTD) and recursive least squares temporal difference (RLSTD), which often fail for non-linear functions.

The rest of the paper is organized as follows. In Section II, the DOC method is reviewed briefly and an application of DOC method is demonstrated. In Section III, the distributed ADP algorithm is presented. The implementation of the value function learning is then demonstrated on the DOC problem in Section IV. Finally, conclusions and future work are discussed in Section V.

II. Distributed Optimal Control Problem

Consider a multiscale dynamical system comprised of N cooperative microscopic dynamical systems, referred to as agents, that can be described by a small system of stochastic differential equations (SDEs),

$$\begin{aligned} \dot{\mathbf{x}}_i(t) &= \mathbf{f}[\mathbf{x}_i(t),\mathbf{u}_i(t),t]+\mathbf{G}\mathbf{w}_i(t), \\ \mathbf{x}_i(T_0) &= \mathbf{x}_{i_0}, \quad i=1,...,N \end{aligned} \tag{1}$$

referred to as the detailed equation. These dynamical systems can describe mechanical, chemical, or robotic systems, or even the distribution of resources in a disaster-stricken environment. In this paper, the multiscale dynamical system describes dynamic constraints on N robotic agents. Agents are assumed to be conserved in a bounded subset of an Euclidean space, denoted by $\mathscr{X} \in \mathbb{R}^{n_x}$, for a fixed time interval $(T_0,\ T_f]$. Also, for simplicity, the approach is presented for agents with perfect measurements and communications, such that each agent state $\mathbf{x}_i \in \mathscr{X}$ and microscopic control $\mathbf{u}_i$, can be assumed known without error at any time $t \in (T_0,\ T_f]$ for all i. The agent dynamics (1) are characterized by an additive Gaussian distribution noise, denoted by $w \in \mathbb{R}^{n_x}$ and $\mathbf{G} \in \mathbb{R}^{n_x \times n_x}$ is a constant matrix.

It is assumed that, on large spatial and temporal scales, the interactions of the N agents give rise to macroscopic coherent behaviors, or coarse dynamics, that are modeled by PDEs. The macroscopic state of the multiscale system, $\mathbf{X} = \wp(\mathbf{x},t) \in \mathscr{P}$, is provided by a so-called *restriction operator*, denoted by $\Phi_t : \mathscr{X} \times \mathbb{R} \to \mathscr{P}$. A restriction operator is used to describe the collection of agents in a number of parameters much less than the number of individual agents. These parameters could be, for example, the means, covariances, and weights of the components of a Gaussian Mixture Model (GMM), where the number of the parameters are not dependent on the number of agents.

For many multiscale dynamical systems, particularly very-large-scale robotic systems (VLRSs), the restriction operator can be provided by a time-varying agent distribution. Since $\mathbf{x}_i$ is a time-varying continuous vector, the agent distribution, i.e. the restriction operator, is a time-varying PDF, $\wp(\mathbf{x},t)$, and $\mathscr{P}$ denotes the probability function space. The probability of the agent state $\mathbf{x} \in B \subset \mathscr{X}$ at any time $t \in (T_0,\ T_f]$ is,

$$P(\mathbf{x} \in B,t) = \int_B \wp(\mathbf{x},t)d\mathbf{x}, \tag{2}$$

where

$$\int_{\mathscr{X}} \wp(\mathbf{x},t)d\mathbf{x}_i = 1, \tag{3}$$

and $\wp(\mathbf{x},t)$ is a non-negative function that, hereon, will be abbreviated to $\wp$ for simplicity.

It is also assumed that the performance of the multiscale dynamical system is an integral function of the macroscopic state and microscopic control,

$$J = \phi[\wp(\cdot,T_f)] + \int_{T_0}^{T_f}\int_{\mathscr{X}} \mathscr{L}[\wp(\mathbf{x},t),\mathbf{u}(\mathbf{x},t),t]d\mathbf{x}_i dt \tag{4}$$

where $\mathscr{L}$ is the Lagrangian, ϕ is the terminal cost based on the PDF $\wp(\cdot,T_f)$ at the final time T_f, and $\mathbf{u}(\mathbf{x},t) \in \mathscr{U}$ is the control law, which is a function defined in the function space $\mathscr{U}$. The evaluations of the control law $\mathbf{u}(\mathbf{x},t)$ is the microscopic control for agent state $\mathbf{x}$ at time t.

The DOC method was developed to solve the optimization problem in (4). In the rest of this section, the DOC method is reviewed and demonstrated with a multi-agent trajectory optimization problem.

A. Review of DOC Method

From (4) it can be seen that, unlike other multi-agent or swarming control approaches [4], [20], which assume the agent distribution is given, the DOC approach seeks to determine the agent distribution such that the agents' performance is optimized. Also, it can be seen that the DOC cost function (4) is formulated as a general functional of the distribution $\wp$, not as the expectation of the Lagrangian, such as in stochastic optimal control [8], [16], or as the expectation of the PDF, as in NCE (Mean Field) methods [21], [22]. In fact, the Lagrangian in (4) denotes a functional of $\wp$ that may or may not include the expectation operator. For example, $\mathscr{L}$ can be formulated in terms of information theoretic functions, such as divergence or mutual information between different distributions, or in terms of a potential navigation function for obstacle avoidance. As a result, the DOC approach affords the optimization of a broad class of objectives, including non-convex functions and non-Gaussian distributions.

As shown in [23], when the agents are conserved in $\mathscr{X}$, the macroscopic evolution equation can be derived by considering an infinitesimal control volume in $\mathscr{X}$. Then the time-rate of change of the distribution $\wp$ is given by the advection equation, which governs the motion of a conserved

scalar quantity as it is advected by a known velocity field [24],

$$\begin{aligned}\frac{\partial \wp}{\partial t} &= -\nabla\cdot[\wp(\mathbf{x},t)\mathbf{v}] + \frac{1}{2}\nabla\cdot[(\mathbf{G}\mathbf{G}^T)\nabla\wp(\mathbf{x},t)] \\ &= -\nabla\cdot[\wp(\mathbf{x},t)\mathbf{f}(\mathbf{x},\mathbf{u},t)] + \nu\nabla^2\wp(\mathbf{x},t) \qquad (5)\end{aligned}$$

where the gradient ∇ is a row vector of partial derivatives with respect to the elements of $\mathbf{x}$, $\nu = \nabla(\mathbf{G}\mathbf{G}^T)$, and $(\cdot)$ denotes the dot product.

Under the stated assumptions, the macroscopic evolution equation is a hyperbolic PDE that provides the dynamic constraints for the optimization of the cost function (4). The initial and boundary conditions (I.C. and B.C.) for (5), as well as an admissability constraint (A.C.), are given by an initial agent distribution, $\wp_0(\mathbf{x}_i)$, and by the normalization condition, i.e.:

$$\text{I.C.:} \quad \wp(\mathbf{x},T_0) = \wp_0(\mathbf{x}); \qquad (6)$$

$$\text{B.C.:} \quad \wp(\mathbf{x}\in\partial\mathscr{X},t) = 0, \quad \forall t\in[T_0,T_f]; \qquad (7)$$

$$\text{A.C.:} \quad \int_{\mathscr{X}} \wp(\mathbf{x},t)d\mathbf{x}_i = 1;$$

$$\wp(\mathbf{x}\notin\partial\mathscr{X},t) = 0, \quad \forall t\in[T_0,T_f]. \qquad (8)$$

When the agents are not conserved or the disturbances $\mathbf{w}_i$ are not zero, alternate forms of the macroscopic equation (5), such as the advection-diffusion equation [17], can still be obtained in closed form from the continuity equation.

The DOC problem seeks to determine the optimal agent distribution $\wp^*$, and control law $\mathbf{u}^*$, that minimize the cost function (4) over a time interval $(T_0,T_f]$, subject to the dynamic and equality constraints (5)-(8). As shown in [12], calculus of variations can be used to derive new necessary conditions for optimality for the DOC problem in (4)-(8). Also, direct and indirect numerical methods were developed in [13], [17] for determining optimal agent distributions and control laws. As illustrated by the results in the next section, existing DOC methods and algorithms have been shown very effective at solving multi-agent trajectory optimization problems in which the agent dynamics and macroscopic evolution equation are known *a priori*.

B. Multi-agent Trajectory Optimization via Indirect DOC

The numerical results presented in this section illustrate some of the properties of indirect DOC algorithms [13], [17]. Consider the problem of determining optimal agent trajectories for a system of N agents that are each described by a single integrator model,

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \upsilon_{x_i} \\ \upsilon_{y_i} \end{bmatrix} + \sigma\mathbf{I}_2 \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} \qquad (9)$$

where $\mathbf{q} = [x,y]^T$ denotes the configuration vector of the i^{th} agent, and x and y are the xy-coordinate. The terms υ_x and υ_y are linear microscopic velocities in the x and y directions, respectively. The disturbance vector is $\mathbf{w} = [\eta_x,\eta_y]^T$, where η_x and η_y are independent random variables with values given by a standard Gaussian process, $\sigma = 0.01$, and $\mathbf{I}_2$ is a 2×2 identity matrix. The agents must travel across a workspace $W = [0,L_x]\times[0,L_y]$, with four obstacles (Fig. 1), $L_x = 20$ (Km), and $L_y = 16$ (Km), in order to move from an initial distribution g_0 to a goal distribution g, while minimizing energy consumption and avoiding collisions with the obstacles during a time interval $(t_0,t_f]$, where $t_0 = 0$ hr, and $t_f = 15$ hr.

The evolutions of the optimal agent PDF and microscopic states obtained are plotted in Fig. 2. For the details of DOC methods, the reader is referred to [13], [23], [25], and especially the multi-agent trajectory optimization problem in Fig. 1 and 2 presented in [23]. In the next section, the data obtained from that simulation is used to implement the value function learning.

III. Value Function Approximation

The results in the previous section illustrate that the DOC approach is ideally suited to adaptive, robust control of multiscale dynamical systems. However, all DOC methods to date assume that the microscopic agent dynamics, the macroscopic evolution equation, and the definition of the restriction operator are known *a priori* [13], [17]. In most applications, however, these models are subject to significant sources of errors, or may not be available in closed form but only in the form of a computer simulation. ADP algorithms seek to optimize a cost-to-go or *value* function conditioned upon knowledge of the actual system. They reduce the computational complexity of the optimal control problem by modeling the control law and the estimated future cost as parametric structures known as actor and critic, respectively.

The actor and critic parametric structures are improved by solving the ADP Recurrence Relations iteratively subject to online measurements of the system state, and knowledge of the probability distribution of uncertainties. Several authors have demonstrated that ADP can be used to solve complex, nonlinear control problems in the absence of an accurate system model, and optimize performance in the face of unforeseen changes and uncertainties [10], [26]–[29]. Due to its ability to learn the solution of an optimal control problem from repeated observations of a system state, ADP is very effective for model-free control, that is, for learning an optimal control law from a computer simulation when a system model is not available in closed form [30]–[32].

A. Value Function for DOC Problems

Since ADP algorithms progress forward in time to improve the control law approximation in real time, the DOC problem in (4)-(8) is discretized with respect to time. Letting $t_k = k\delta t$ denote the discrete time index, where δt is a small discrete time interval, the macroscopic equation (5) can be written as

$$\begin{aligned}\wp(\mathbf{x},t_{k+1}) &= \wp(\mathbf{x},t_k) + \nu\nabla^2\wp(\mathbf{x},t_k)\delta t \\ &\quad -\nabla\cdot\{\wp(\mathbf{x},t_k)\mathbf{f}[\mathbf{x}(t_k),\mathbf{u}(\mathbf{x},t_k),t_k]\}\delta t \\ &\triangleq \mathscr{F}[\wp(\mathbf{x},t_k),\mathbf{u}(\mathbf{x},t_k),t_k] \qquad (10)\end{aligned}$$

Assume that the map $t\mapsto\wp(\mathbf{x}_i,t)$ is injective, that is to say that the distributions $\wp(\mathbf{x},t_k)$ and $\wp(\mathbf{x},t_k')$ are distinguishable for $t_k\neq t_k'$. Then, the (agent) microscopic control law $\mathscr{C}(\cdot)$

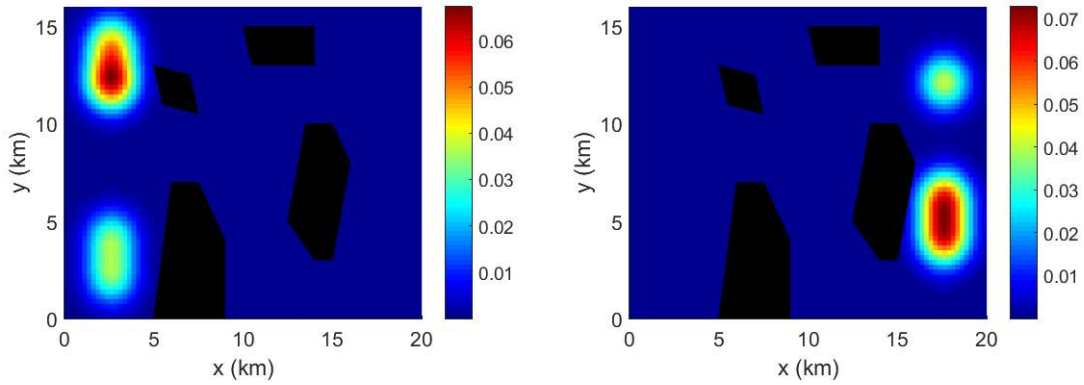

Fig. 1: Initial (left) and goal (right) agent distributions for a workspace with four obstacles shown in black.

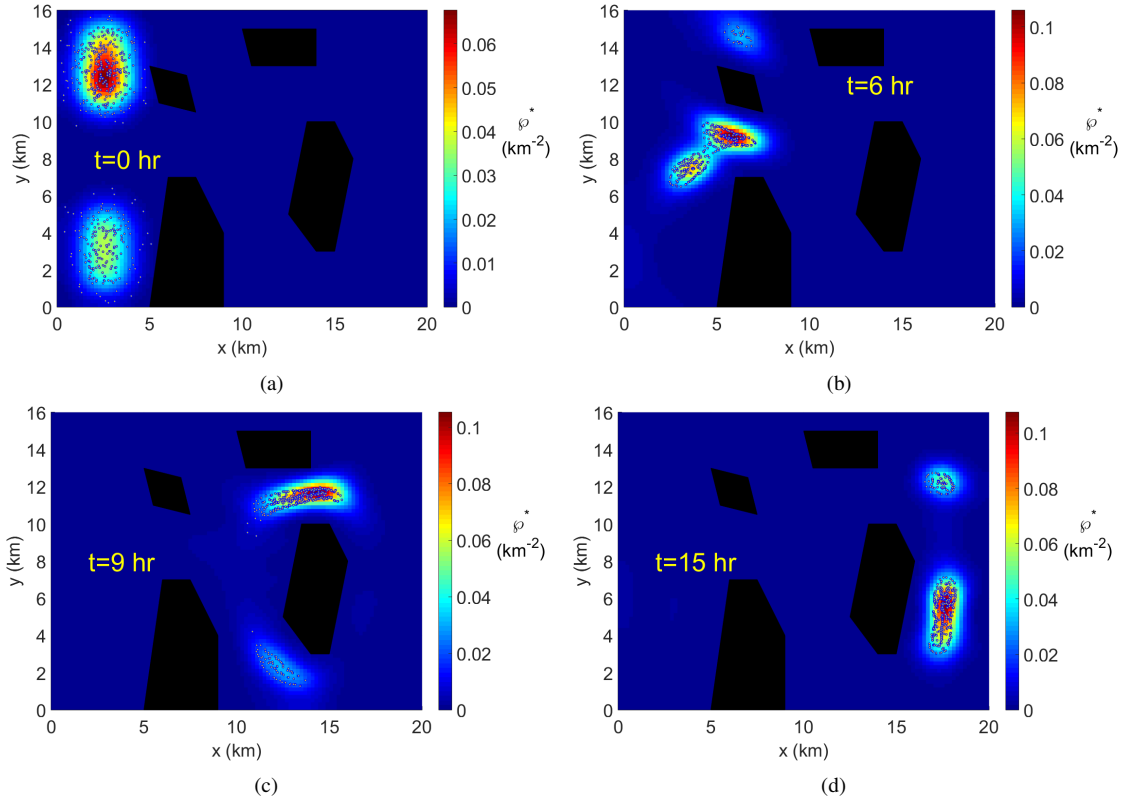

Fig. 2: Optimal agent PDF and microscopic agent states obtained by the indirect DOC algorithm at four sample moments in time.

can be treated as a function of $\wp(\mathbf{x},t_k)$ and, using the shorthand notations $\wp_k = \wp(\mathbf{x},t_k)$ and $\mathbf{u}_k = \mathbf{u}(\mathbf{x},t_k) = \mathscr{C}(\wp_k) \in \mathscr{U}$, the macroscopic equation in discrete time (10) can be rewritten as

$$\wp_{k+1} = \mathscr{F}(\wp_k, \mathbf{u}_k) \tag{11}$$

It can be seen that $\mathscr{F}$ and $\mathscr{C}$ are both operators, such that $\mathscr{F} : \mathscr{P} \times \mathscr{U} \mapsto \mathscr{P}$ and $\mathscr{C} : \mathscr{P} \mapsto \mathscr{U}$, and, for a given detailed equation $\mathbf{f}[\cdot]$ in (1), the operator $\mathscr{F}$ can be determined according to (10). The cost function (4) in discrete time is given by

$$J = \phi[\wp_{t_f}] + \sum_{t_k=t_0}^{t_{f-1}} \int_{\mathscr{X}} \mathscr{L}(\wp_k, \mathbf{u}_k) d\mathbf{x}. \tag{12}$$

Since $\mathbf{u}_k = \mathscr{C}(\wp_k)$, at any time $t_k \in [T_0, T_f)$, the value function representing the cost-to-go of the multiscale dynamical system is defined as

$$\mathscr{V} \triangleq \phi[\wp_{k_f}] + \sum_{t_k}^{t_{f-1}} \int_{\mathscr{X}} \mathscr{L}[\wp_k, \mathscr{C}(\wp_k)] d\mathbf{x} = \mathscr{V}[\wp_k, \mathscr{C}(\cdot)] \tag{13}$$

and, thus, depends on the present agent distribution $\wp_k$, and

on the chosen control law $\mathscr{C}$. From Bellman's principle of optimality, the optimal cost-to-go at any time t_k can be obtained by minimizing the right-hand side of (13) with respect to the present control inputs as provided by the chosen control law, such that

$$\mathscr{V}^* = \min_{\mathbf{u}_k} \left\{ \int_{\mathscr{X}} \mathscr{L}(\wp_k^*, \mathbf{u}_k) d\mathbf{x}_i + \mathscr{V}^*[\wp_{k+1}^*, \mathscr{C}^*(\cdot)] \right\} \quad (14)$$

Equation (14) can be viewed as the recurrence relationship of Dynamic Programming (DP) for the DOC problem in Section II.

Once the proposed value-function is learned, a distributed ADP algorithm can be obtained by solving the DP problem forward in time, iterating between a policy-improvement routine and a value-determination operation. At every cycle of the distributed ADP algorithm, indexed by ℓ, the control law $\mathscr{C}$ and value function $\mathscr{V}$ are approximated based on the macroscopic state $\wp_k$ of the multiscale dynamical system. For example, when the restriction operator represents the agent distribution, $\wp_k$ can be estimated using a kernel density estimation algorithm that can be decentralized as shown in [33]. Given the observed value of $\wp_k$, the distributed ADP algorithm iterates over ℓ to search for the optimal DOC solution online, by cycling between the following operations:

Policy-Improvement Routine: Given a value function $V[\cdot, \mathscr{C}_\ell]$ corresponding to a control law $\mathscr{C}_\ell$, an improved control, $\mathscr{C}_{\ell+1}$, can be obtained as follows,

$$\mathscr{C}_{\ell+1}(\wp_k) = \arg\min_{\mathbf{u}_k} \left\{ \int_{\mathscr{X}} \mathscr{L}(\wp_k, \mathbf{u}_k) d\mathbf{x} + \mathscr{V}[\wp_{k+1}, \mathscr{C}_\ell(\cdot)] \right\} \quad (15)$$

Value-Determination Operation: Given a control law $\mathscr{C}$, the value function can be updated according to the following rule

$$\mathscr{V}_{\ell+1}[\wp_k, \mathscr{C}(\cdot)] = \int_{\mathscr{X}} \mathscr{L}(\wp_k, \mathbf{u}_k) d\mathbf{x} + \mathscr{V}_\ell[\wp_{k+1}, \mathscr{C}(\cdot)] \quad (16)$$

The operator optimization required by the distributed ADP algorithm is much harder than the usual function minimization required by classical ADP algorithms. Also, while classical ADP algorithms rely on the observation of a state vector that is clearly defined, distributed ADP relies on a restriction operator that, in the simplest case, is a PDF defined over $\mathscr{X}$, but, in the most general case, is an operator that must be learned from data obtained by bursts of appropriately initialized microscopic simulation.

B. Implementation of Value Function Learning

The proposed value function in (13) is a map $\mathscr{V} : \mathscr{P} \to \mathbb{R}$, which is a functional regression problem. For such a problem, in general, there are two kinds of approaches: 1) parametric method where the distribution function $\wp_k$ is specified by some parameters $\mathbf{p}_k \in \mathbb{R}^{n_p}$ and the map $\mathbb{R}^{n_p} \to \mathbb{R}$ is learned from data sets to represent the functional map, 2) non-parametric method where the distribution function $\wp_k$ is represented by non-parametric approach and then the functional map is constructed and learned from data sets directly [34]. In this paper, a parametric method is proposed to implement the value function learning because it is simpler and less computationally demanding than the non-parametric method.

First, the expectation-maximization (EM) algorithm [18], [19] is applied to learn the GMM representing the distribution $\wp_k$ at every time step k. Therefore, the distribution $\wp_k$ of the agents is approximated by the GMM, $\tilde{\wp}_k$, which is specified by the parameter set $\tilde{\mathbf{p}}_k = \{(\omega_{k,\tau}, \mu_{k,\tau}, \Sigma_{k,\tau})\}_{\tau=1}^n$, such that,

$$\tilde{\wp}_k(\mathbf{x}) = \sum_{\tau=1}^{n} \omega_\tau \mathscr{N}(\mathbf{x}|\mu_{k,\tau}, \Sigma_{k,\tau}), \quad (17)$$

where n is the number of the components of the GMM, $\mathscr{N}(\mathbf{x}|\mu_{k,\tau}, \Sigma_{k,\tau})$ denotes the normal distribution with mean $\mu_{k,\tau}$ and covariance matrix $\Sigma_{k,\tau}$, and $\omega_{k,\tau}$ is the corresponding mixture weight.

Because the parameter tube $(\omega_{k,\tau}, \mu_{k,\tau}, \Sigma_{k,\tau})$ for $\tau = 1, ..., n$ cannot be ordered for each distribution $\wp_k$, the map from $\tilde{\mathbf{p}}_k$ to V cannot be learned directly. To obtain the input vector for the regression, the following neural network is used to learn the map. In Fig. 3, I_j is a integral function defined by

$$I_j(\tilde{\wp}_k) = \int_{\mathbf{x} \in \mathscr{X}} \tilde{\wp}_k(\mathbf{x}) b_j(\mathbf{x}) d\mathbf{x}, \quad \text{for} \quad j = 1, ..., m \quad (18)$$

where $b_j(\mathbf{x})$ is a predefined basis function. Here, the basis functions are all normal distributions, *i.e.* $b_j(\mathbf{x}) = \mathscr{N}(\mathbf{x}|\mu_j, \Sigma_j)$ for $j = 1, ..., m$. For simplicity, the means μ_j are chosen from a grid on the workspace and the same basis covariance matrices Σ_j are applied. In addition, the parameters α_j are the output weights of the integral function in (18). Let the output of the jth integral function be denoted by $p_j = I_j(\tilde{\wp}_k)$. Then, the approximate value function $\tilde{V}(\tilde{\wp}_k)$ can be expressed by,

$$\tilde{V}(\tilde{\wp}_k) = \alpha^T \mathbf{p}_k, \quad (19)$$

where $\alpha = [\alpha_1, ..., \alpha_m]^T$ and $\mathbf{p}_k = [p_1, ..., p_m]^T$.

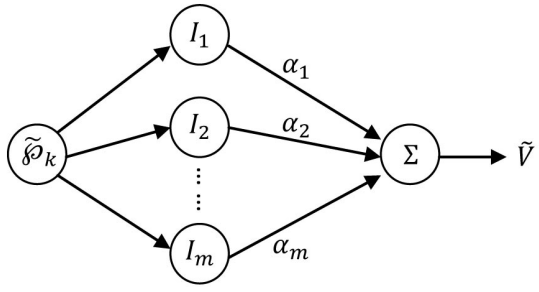

Fig. 3: Neural network for learning the map from GMM parameters to value function.

It can be shown that the structure of the proposed neural network is very similar to the radial basis function network (RBFN) with the exception of the integral function. For the RBFN, the output is the evaluation of the radial basis function, while for the proposed neural network the output is obtained from the integral in (18). According to the Gaussian

identity property [35], the integral function has the following the closed-form expression:

$$\begin{aligned} I_j(\tilde{\wp}_k) &= \int_{\mathbf{x}\in\mathscr{X}} \tilde{\wp}_k(\mathbf{x}) b_j(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathbf{x}\in\mathscr{X}} \left[\sum_{\tau=1}^{n} \mathscr{N}(\mathbf{x}|\mu_{k,\tau},\Sigma_{k,\tau})\right] \mathscr{N}(\mathbf{x}|\mu_j,\Sigma_j) d\mathbf{x} \\ &= \sum_{\tau=1}^{n} \int_{\mathbf{x}\in\mathscr{X}} \mathscr{N}(\mathbf{x}|\mu_{k,\tau},\Sigma_{k,\tau}) \mathscr{N}(\mathbf{x}|\mu_j,\Sigma_j) d\mathbf{x} \\ &= \sum_{\tau=1}^{n} \mathscr{N}(\mu_{k,\tau}|\mu_j,\Sigma_{k,\tau}+\Sigma_j). \end{aligned} \tag{20}$$

Furthermore, the integral function can be treated as the inner product between the approximate distribution $\tilde{\wp}_k$ and the basis function b_j in the function space $\mathscr{P}$, such that $I_j(\tilde{\wp}_k) = \langle \tilde{\wp}_k, b_j \rangle_{\mathscr{P}}$. Therefore, the integral function can be redefined as

$$\begin{aligned} I_j(\tilde{\wp}_k) &= \langle \tilde{\wp}_k, \frac{b_j}{\|b_j\|} \rangle_{\mathscr{P}} = \frac{\langle \tilde{\wp}_k, b_j \rangle_{\mathscr{P}}}{\|b_j\|} = \frac{\langle \tilde{\wp}_k, b_j \rangle_{\mathscr{P}}}{\langle b_j, b_j \rangle_{\mathscr{P}}} \\ &= \frac{\int_{\mathbf{x}\in\mathscr{X}} \tilde{\wp}_k(\mathbf{x}) b_j(\mathbf{x}) d\mathbf{x}}{\int_{\mathbf{x}\in\mathscr{X}} b_j(\mathbf{x}) b_j(\mathbf{x}) d\mathbf{x}} \\ &= \frac{\sum_{j=1}^{n} \mathscr{N}(\mu_{k,\tau}|\mu_j,\Sigma_{k,\tau}+\Sigma_j)}{\mathscr{N}(\mu_j|\mu_j,2\Sigma_j)}, \end{aligned} \tag{21}$$

where $b_j/\|b_j\|$ is a normalized basis.

From (19) and (21), it can be seen that the approximate value function $\tilde{V}$ can be learned by obtaining the coefficient vector $\alpha = [\alpha_1,...,\alpha_m]^T$ using existing reinforcement learning algorithms, such as the least squares temporal difference (LSTD) and recursive least squares temporal difference (RLSTD) algorithms. The former is an off-line (batch learning) algorithm, while the latter is an on-line algorithm. Therefore, these two proposed value-function learning algorithms are named the least squares temporal difference based on GMM (LSTD-GMM) and recursive least squares temporal difference based on GMM (RLSTD-GMM), respectively.

The proposed algorithm is implemented based on the EM-GMM algorithm and the reinforcement learning. The EM-GMM algorithm is applied for each step to estimate the distributions of agents. Thus the computational complexity of the proposed algorithm is specified by EM-GMM and reinforcement learning algorithms.

IV. SIMULATIONS

To evaluate the proposed value-function learning algorithm, the data obtained from the multi-agent trajectory optimization problem in [23], presented in Section II, are used as the training data set. This data consists of the agent positions determined by the indirect DOC method and the corresponding value function evaluations obtained from (13). In this simulation, there are $n = 5$ components of the GMM for each approximated agent distribution $\tilde{\wp}_k$. The means of basis functions b_j, $j = 1,...,m$, are set on the points of the grid where the interval is 1 km and the corresponding covariance matrices are set as $\Sigma_j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. For simplicity,

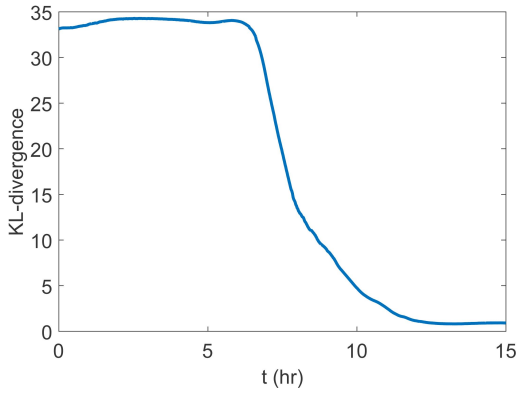

Fig. 4: The KL-divergence between the agent distribution $\wp_k$ and the goal distribution g.

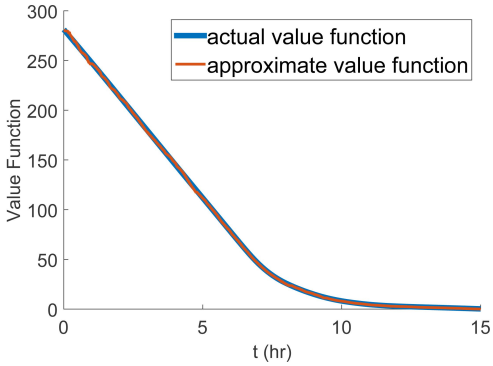

Fig. 5: The actual value function and the approximate value function.

the Lagrangian is defined by the KL-divergence between $\wp_k$ and the goal distribution g as follows,

$$\mathscr{L}(\wp_k) = \int_{\mathbf{x}\in\mathscr{X}} \wp_k(\mathbf{x}) \log \frac{\wp_k(\mathbf{x})}{g(\mathbf{x})} d\mathbf{x}, \tag{22}$$

and can be approximated from the agent positions. Furthermore, the terminal cost is also defined using the KL-divergence, plotted in Fig. 4.

Based on the Lagrangian terms (KL-divergence), the actual value function $V(\wp_k)$ can be evaluated from the agent positions. Then, $V(\wp_k)$ and the corresponding PDFs, $\wp_k$, $k = t_0,...,t_f$, are used to learn the coefficient vector α. Finally, the approximate value function, $\tilde{V}(\tilde{\wp}_k)$, is calculated from (19). The actual value function and the approximate value function are both plotted in Fig. 5 for comparison. Also, the approximation error is plotted in Fig. 6, demonstrating the high accuracy achieved by the proposed approximation scheme. The approximate value function has a normalized means squares error (NMSE) of only 3.7145×10^{-5}.

V. CONCLUSIONS AND FUTURE WORK

This paper presents new recurrence relationships for distributed ADP of multiscale dynamical systems comprised of many interacting agents. A novel functional regression algorithm is proposed to learn the value function as an

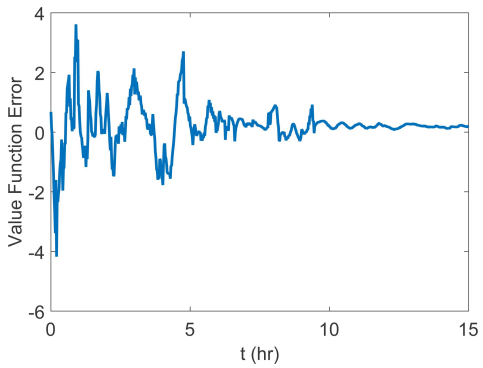

Fig. 6: The error between the actual value function and the approximate value function.

unknown functional of the agent time-varying PDF. The proposed algorithm is tested using data obtained from the DOC optimality conditions solved in [23]. The simulation results show that the approximate value function approximates the actual value function very closely, and thus the learning algorithm can be applied for distributed ADP. In future work, on-line value function learning algorithms, such as RLSTD-GMM, will be applied to implement the policy update in the distributed ADP method.

VI. ACKNOWLEDGMENTS

This research was funded by the National Science Foundation grant ECCS-1556900.

REFERENCES

[1] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, no. 3, pp. 171–194, 1999.

[2] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, 1998.

[3] J. P. Desai, J. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 905–908, 2001.

[4] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 1, pp. 539–557, 2004.

[5] L. P. F. Giulietti and M. Innocenti, "Autonomous formation flight," *IEEE Control Systems Magazine*, vol. 20, pp. 3444–3457, 2000.

[6] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proceedings of the Conference on Decision and Control*, 2001, pp. 2968–2973.

[7] M. Athans and P. Falb, *Optimal Control*. New York: McGraw-Hill, 1966.

[8] R. F. Stengel, *Optimal Control and Estimation*. Dover Publications, Inc., 1986.

[9] J. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[10] J. Si, A. Barto, and W. Powell, *Learning and Approximate Dynamic Programming*. John Wiley and Sons, 2004.

[11] J. E. Hopcroft, J. T. Schwartz, and M. Sharir, "On the complexity of motion planning for multiple independent objects; pspace- hardness of the "warehouseman's problem"," *The International Journal of Robotics Research*, vol. 3, no. 4, pp. 76–88, 1984.

[12] G. Foderaro and S. Ferrari, "Necessary conditions for optimality for a distributed optimal control problem,," in *Proceedings of the IEEE Conference on Decision and Control*, 2010, pp. 4831–4838.

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

[14] A. Singh and J. P. Hespanha, "Moment closure techniques for stochastic models in population biology," *IEEE Proceedings of the American Control Conference*, pp. 4730–4735, 2006.

[15] ——, "A derivative matching approach to moment closure for the stochastic logistic model," *Bulletin of Mathematical Biology*, vol. 69, no. 6, pp. 1909–1925, 2007.

[16] S. Peng and Z. Wu, "Fully coupled forward-backward stochastic differential equations and applications to optimal control," *SIAM Journal on Control and Optimization*, vol. 37, no. 3, pp. 825–843, 1999.

[17] K. Rudd, G. Foderaro, and S. Ferrari, "A generalized reduced gradient method for the op- timal control of multiscale dynamical systems," in *Proceedings of the IEEE Conference on Decision and Control*, Florence, Italy, 2013, pp. 3857–3863.

[18] J. A. Bilmes *et al.*, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[19] C. Tomasi, "Estimating gaussian mixture densities with em–a tutorial," *Duke University*, 2004.

[20] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shaped control for a swarm of robots," *Automatica*, vol. 45, pp. 2406–2411, 2009.

[21] M. Huang, P. E. Caines, and R. P. Malhame, "The nce (mean field) principle with locality dependent cost interactions," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2799–2805, 2010.

[22] M. Huang, R. P. Malhame, and P. E. Caines, "Large population stochastic dynamic games: Closed-loop mckean-vlasov systems and the nash certainty equivalence principle," *Communications in Information Systems*, vol. 6, no. 3, pp. 221–252, 2006.

[23] K. Rudd, G. Foderaro, P. Zhu, and S. Ferrari, "A generalized reduced gradient method for the optimal control of mobile robotic networks," *IEEE Transactions on Robotics*, p. submitted, 2016.

[24] J. P. Boyd, *Chebyshev and Fourier Spectral Methods, II Ed.* New York, NY: Dover, 2001.

[25] S. Ferrari, G. Foderaro, and P. Zhu, "Distributed optimal control of multiscale dynamical systems: A tutorial," *IEEE Contorl System Magazine*, p. in press, 2016.

[26] S. Ferrari and R. Stengel, "On-line adaptive critic flight control," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, pp. 777–786, 2004.

[27] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "Optimal control synthesis of a class of nonlinear systems using single network adaptive critics," *Neural Networks*, vol. 19, no. 1, pp. 1648–1660, 2006.

[28] R. E. Parr, *Hierarchical Control and Learning for Markov Decision Processes*. Berkeley, CA: Ph.D. Thesis, University of California, 1998.

[29] F. lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. John Wiley and Sons, 2012.

[30] A. Al-Tamimi, F. Lewis, and M. Abu-Khalaf, "Model-free q-learning designs for linear discrete-time zero-sum games with application to h-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, 2007.

[31] S. N. Balakrishnan, J. Ding, and F. Lewis, "Issues on stability of adp feedback controllers for dynamical systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 4, pp. 913–917, 2008.

[32] F. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, 2009.

[33] G. Foderaro, S. Ferrari, and M. Zavlanos, "A decentralized kernel density estimation ap- proach to distributed robot path planning,," in *Proceedings of the Neural Information Processing Systems Conference*, Lake Tahoe, NV.

[34] P. Zhu, H. Wei, W. Lu, and S. Ferrari, "Multi-kernel probability distribution regressions," in *Proceedings of the International Joint Conference on Neural Networks*, Killarney, Ireland, 2015, pp. 1–7.

[35] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.