Approximate optimal cooperative decentralized control for consensus in a topological network of agents with uncertain nonlinear dynamics

Rushikesh Kamalapurkar, Huyen Dinh, Patrick Walters, and Warren Dixon

***Abstract*—Efforts in this paper seek to combine graph theory with adaptive dynamic programming (ADP) as a reinforcement learning (RL) framework to determine forward-in-time, real-time, approximate optimal controllers for distributed multi-agent systems with uncertain nonlinear dynamics. A decentralized continuous time-varying control strategy is proposed, using only local communication feedback from two-hop neighbors on a communication topology that has a spanning tree. An actor-critic-identifier architecture is proposed that employs a nonlinear state derivative estimator to estimate the unknown dynamics online and uses the estimate thus obtained for value function approximation.**

I. Introduction

Combined efforts from multiple autonomous agents can yield tactical advantages including: improved munitions effects; distributed sensing, detection, and threat response; and distributed communication pipelines. While coordinating behaviors among autonomous agents is a challenging problem that has received mainstream focus, unique challenges arise when seeking autonomous collaborative behaviors in low bandwidth communication environments. For example, most collaborative control literature focuses on centralized approaches that require all nodes to continuously communicate with a central agent, yielding a heavy communication demand that is subject to failure due to delays, and missing information. Furthermore, the central agent requires to carry enough computational resources on-board to process the data and to generate command signals. These challenges motivate the need for a decentralized approach where the nodes only need to communicate with their neighbors for guidance, navigation and control tasks.

Reinforcement learning (RL) allows an agent to learn the optimal policy by interacting with its environment, and hence, is useful for control synthesis in complex dynamical systems such as a network of agents. Decentralized algorithms have been developed for cooperative control of networks of agents with finite state and action spaces in [1]–[4]. See [2] for a survey. The extension of these techniques to networks of agents with infinite state and action spaces and nonlinear dynamics is challenging due to difficulties in value function approximation, and has remained an open problem.

As the desired action by an individual agent depends on the actions and the resulting trajectories of its neighbors, the error system for each agent becomes a complex nonautonomous dynamical system. Nonautonomous systems, in general, have non-stationary value functions. As non-stationary functions are difficult to approximate using parametrized function approximation schemes such as neural networks (NNs), designing optimal policies for nonautonomous systems is not trivial. To get around this challenge, differential game theory is often employed in multi-agent optimal control, where a solution to the coupled Hamilton-Jacobi-Bellman (HJB) equation (c.f. [5]) is sought. As the coupled HJB equations are difficult to solve, some form of generalized policy iteration or value iteration [6] is often employed to get an approximate solution. It is shown in results such as [5], [7]–[11] that approximate dynamic programming (ADP) can be used to generate approximate optimal policies online for multi-agent systems. As the HJB equations to be solved are coupled, all of these results have a centralized control architecture.

Decentralized control techniques focus on finding control policies based on local data for individual agents that collectively achieve the desired goal, which, for the problem considered in this effort, is consensus to the origin. Various methods have been developed to solve the consensus problem for linear systems with exact model knowledge. An optimal control approach is used in [12] to achieve consensus while avoiding obstacles. In [13], an optimal controller is developed for agents with known dynamics to cooperatively track a desired trajectory. In [14], an optimal consensus algorithm is developed for a cooperative team of agents with linear dynamics using only partial information. A value function approximation based approach is presented in [15] for cooperative synchronization in a strongly connected network of agents with known linear dynamics. It is also shown in [15] that the obtained policies are in a cooperative Nash equilibrium.

For nonlinear systems, a model predictive control approach is presented in [16], however, no stability or convergence analysis is presented. A stable distributed model predictive controller is presented in [17] for nonlinear discrete-time systems with known nominal dynamics. Asymptotic stability is

Rushikesh Kamalapurkar, Huyen Dinh, Patrick Walters, and Warren Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA. Email: {rkamalapurkar, huyentdinh, walters8, wdixon}@ufl.edu.

This research is supported in part by NSF award numbers 0901491, 1161260, and 1217908 and ONR grant number N00014-13-1-0151. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

proved without any interaction between the nodes, however, a nonlinear optimal control problem need to be solved at every iteration to implement the controller. Decentralized optimal control synthesis for consensus in a topological network of agents with continuous-time uncertain nonlinear dynamics has remained an open problem.

In this result, an ADP-based approach is developed to solve the consensus problem for a network topology that has a spanning tree. The agents are assumed to have nonlinear control-affine dynamics with unknown drift vectors and known control effectiveness matrices. An identifier is used in conjunction with the controller enabling the algorithm to find approximate optimal decentralized policies online without the knowledge of drift dynamics. This effort thus realizes the actor-critic-identifier (ACI) architecture (c.f. [18], [19]) for networks of agents.

II. Graph Theory Preliminaries

Let $\mathcal{N} \triangleq \{\beta_1, \beta_2, \cdots, \beta_N\}$ denote a set of N agents moving in the state space $S \subset \mathbb{R}^n$. The objective is for the agents to reach a consensus state. Without loss of generality, let the consensus state be the origin of the state space, i.e. $S \ni x_0 = 0$. To aid the subsequent design, the agent β_0 (henceforth referred to as the leader) is assumed to be stationary at the origin. The agents are assumed to be on a network with a fixed communication topology modeled as a static directed graph (i.e. digraph).

Each agent forms a node in the digraph. If agent β_j can communicate with agent β_i then there exists a directed edge from the j^{th} to the i^{th} node of the digraph, denoted by the ordered pair $(\beta_j, \beta_i) \in \mathcal{N} \times \mathcal{N}$. Let $E \subset \mathcal{N} \times \mathcal{N}$ denote the set of all edges. Let there be a positive weight $a_{ij} \in \mathbb{R}$ associated with each edge (β_j, β_i). Note that $a_{ij} \neq 0$ if and only if $(\beta_j, \beta_i) \in E$. The digraph is assumed to have no repeated edges i.e. $(\beta_i, \beta_i) \notin E, \forall i$, which implies $a_{ii} = 0, \forall i$. Note that a_{i0} denotes the edge weight (also referred to as the pinning gain) for the edge between the leader and an agent β_i. Similar to the other edge weights, $a_{i0} \neq 0$ if and only if there exists a directed edge from the leader to the agent i. The neighborhood set of agent β_i is denoted by $\mathcal{N}_i$ defined as $\mathcal{N}_i \triangleq \{j \mid (\beta_j, \beta_i) \in E\}$. To streamline the analysis, the graph connectivity matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$ is defined as $\mathcal{A} \triangleq [a_{ij} \mid i, j = 1, \cdots, N]$, the pinning gain matrix $\mathcal{A}_0 \in \mathbb{R}^{N \times N}$ is a diagonal matrix defined as $\mathcal{A}_0 \triangleq diag\left(a_{i0}\right) \mid i = 1, \cdots, N$, the matrix $\mathcal{D} \in \mathbb{R}^{N \times N}$ is defined as $\mathcal{D} \triangleq diag\left(d_i\right)$, where $d_i \triangleq \sum_{j \in \mathcal{N}_i} a_{ij}$, and the graph Laplacian matrix $\mathcal{L} \in \mathbb{R}^{N \times N}$ is defined as $\mathcal{L} \triangleq \mathcal{D} - \mathcal{A}$. The graph is said to have a spanning tree if given any node β_i, there exists a directed path from the leader β_0 to β_i. For notational brevity, a linear operator $\Upsilon_i\left((\cdot)\right)$ is defined as

$$\Upsilon_i\left((\cdot)\right) \triangleq \left(\sum_{j \in \mathcal{N}_i} a_{ij}\left((\cdot)_i - (\cdot)_j\right) + a_{i0}\left((\cdot)_i\right)\right). \quad (1)$$

III. Problem Definition

Let the dynamics of each agent be described as

$$\dot{x}_i = f_i\left(x_i\right) + g_i u_i, \forall i = 1, 2, \cdots, N$$

where $x_i\left(\cdot\right) \in S \subset \mathbb{R}^n$ is the state, $f_i : S \to \mathbb{R}^n$ is a locally Lipschitz function, $g_i :\in \mathbb{R}^{n \times m}$ is the constant control effectiveness matrix, and $u_i\left(\cdot\right) \subset \mathbb{R}^m$ is the control policy. To achieve consensus to the leader, define the local neighborhood tracking error $e_i\left(\cdot\right) \in S \subset \mathbb{R}^n$ for each agent as [20]

$$e_i \triangleq \Upsilon_i\left(x\right) = \sum_{j \in \mathcal{N}_i} a_{ij}\left(x_i - x_j\right) + a_{i0}\left(x_i\right). \quad (2)$$

Denote the cardinality of the set $\mathcal{N}_i$ by $|\mathcal{N}_i|$. Let $\mathcal{E}_i\left(\cdot\right) \in S^{|\mathcal{N}_i|+1} \subseteq \mathbb{R}^{|\mathcal{N}_i|+1}$ be a stacked vector of local neighborhood tracking errors corresponding to the agent β_i and its neighbors, i.e., $\mathcal{E}_i \triangleq \{e_j \mid j \in \mathcal{N}_i\} \cup \{e_i\}$. To achieve consensus in an optimal cooperative way, it is desired to minimize, for each agent, the cost $J_i \triangleq \frac{1}{2}\int_0^\infty r_i\left(\mathcal{E}_i, u_i\right) dt$, where

$$r_i\left(\mathcal{E}_i, u_i\right) \triangleq e_i^T Q_{ii} e_i + u_i^T R_i u_i + \sum_{j \in \mathcal{N}_i} a_{ij} e_j^T Q_{ij} e_j. \quad (3)$$

In (3), $R_i \in \mathbb{R}^{m \times m}$ and $Q_{ii}, Q_{ij} \in \mathbb{R}^{n \times n}$ are symmetric positive definite matrices of constants. Let $\mathcal{E} \triangleq \begin{bmatrix} e_1^T & e_2^T & \cdots & e_N^T \end{bmatrix}^T \in S^{nN} \subset \mathbb{R}^{nN}$ and $\mathcal{X} \triangleq \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}^T \in S^{nN} \subset \mathbb{R}^{nN}$. Using the definition of e_i from (2) we get

$$\mathcal{E} = \begin{bmatrix} \Upsilon_1\left(x\right) \\ \Upsilon_2\left(x\right) \\ \vdots \\ \Upsilon_N\left(x\right) \end{bmatrix} = \left(\left(\mathcal{L} + \mathcal{A}_0\right) \otimes I_n\right) \mathcal{X},$$

where $\otimes$ denotes the Kronecker product and $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix.

IV. Control Development

A. State derivative estimation

Based on the development in [19], each agent's dynamics can be approximated using a dynamic neural network (DNN) with M_{fi} hidden layer neurons as

$$\dot{x}_i = W_{fi}^T \sigma(V_{fi}^T x_i) + \varepsilon_{fi}(x_i) + g_i(x_i) u_i,$$

where $W_{fi} \in \mathbb{R}^{M_{fi}+1 \times n}$, $V_{fi} \in \mathbb{R}^{n \times M_{fi}}$ are unknown ideal DNN weights, $\sigma_{fi} \triangleq \sigma(V_{fi}^T x_i) \in \mathbb{R}^{M_{fi}+1}$ is a bounded DNN activation function, and $\varepsilon_{fi} : \mathbb{R}^n \to \mathbb{R}^n$ is the function reconstruction error. In the following, the drift dynamics f_i are unknown and the control effectiveness functions g_i are assumed to be known. Each agent estimates the derivative of its own state using the following state-derivative estimator

$$\begin{aligned} \dot{\hat{x}}_i &= \hat{f}_i + g_i(x_i) u_i + \mu_i, \quad \hat{f}_i \triangleq \hat{W}_{fi}^T \hat{\sigma}_{fi}, \\ \dot{\hat{W}}_{fi} &= proj(\Gamma_{wfi} \hat{\sigma}'_{fi} \hat{V}_{fi}^T \dot{\hat{x}}_i \tilde{x}_i^T), \\ \dot{\hat{V}}_{fi} &= proj(\Gamma_{vfi} \dot{\hat{x}}_i \tilde{x}_i^T \hat{W}_{fi}^T \hat{\sigma}'_{fi}), \\ \mu_i &\triangleq k_{fi} \tilde{x}_i(t) - k_{fi} \tilde{x}_i(0) + v_i, \\ \dot{v}_i &= (k_{fi} \alpha_{fi} + \gamma_{fi}) \tilde{x}_i + \beta_{1fi} sgn(\tilde{x}_i), \quad v_i(0) = 0, \end{aligned} \quad (4)$$

where $\hat{W}_{fi}(\cdot) \in \mathbb{R}^{M_{fi}+1\times n}$ and $\hat{V}_{fi}(\cdot) \in \mathbb{R}^{n\times M_{fi}}$ are the estimates for the ideal DNN weights W_{fi} and V_{fi}, $\hat{x}_i(\cdot) \in \mathbb{R}^n$ is the state estimate, $\hat{\sigma}_{fi} \triangleq \sigma(\hat{V}_{fi}^T \hat{x}_i) \in \mathbb{R}^{M_{fi}+1}$, $\tilde{x}_i \triangleq x_i - \hat{x}_i \in \mathbb{R}^n$ is the state estimation error, k_{fi}, α_{fi}, γ_{fi}, $\beta_{1fi} \in \mathbb{R}$ are positive constant control gains, $proj\{\cdot\}$ is a smooth projection operator [21], and $v_i(\cdot) \in \mathbb{R}^n$ is a generalized Filippov solution to (4). For notational brevity define

$$\hat{F}_i(x_i, \hat{x}_i, u_i, t) \triangleq \hat{f}_i(\hat{x}_i) + g_i(x_i)u_i + \mu_i(t),$$
$$F_i(x_i, u_i) \triangleq f_i(x_i) + g_i(x_i)u_i, \quad \tilde{F}_i \triangleq \hat{F}_i - F_i.$$

It is shown in [19, Theorem 1] that provided the gains k_{fi} and γ_{fi} are sufficiently large and x_i and u_i are bounded, the estimation error $\tilde{x}_i$ and its derivative are bounded. Furthermore, $\lim_{t\to\infty} \|\tilde{x}_i(t)\| = 0$, $\lim_{t\to\infty} \left\|\dot{\tilde{x}}_i(t)\right\| = 0$, and $\tilde{F}_i \in \mathcal{L}_\infty$.

B. Value function approximation

The value function $V_i : S^{|\mathcal{N}_i|+1} \to \mathbb{R}^+$ is the cost-to-go for each agent given by

$$V_i(\mathcal{E}_i^o) = \frac{1}{2}\int_{t_0}^{\infty} r_i(\mathcal{E}_i(\tau), u_i(\mathcal{E}_i(\tau)))\, d\tau, \tag{5}$$

where $\mathcal{E}_i(\tau)$ denote the neighborhood tracking error trajectories associated with agent β_i and its neighbors, with the initial conditions $\mathcal{E}_i(t_0) = \mathcal{E}_i^o$. The time derivative of V_i is then given by

$$\dot{V}_i = \sum_{j\in i\cup\mathcal{N}_i} \frac{\partial V_i}{\partial e_j}\Upsilon_j(F).$$

The optimal value function $V_i^* : S^{|\mathcal{N}_i|+1} \to \mathbb{R}^+$ is defined as

$$V_i^*(\mathcal{E}_i^o) \triangleq \min_{\substack{u_i: S^{|\mathcal{N}_i|+1}\to\mathbb{R}^m \\ u_i\in U_i}} \frac{1}{2}\int_t^{\infty} r_i(\mathcal{E}_i(\tau), u_i(\mathcal{E}_i(\tau)))\, d\tau, \tag{6}$$

where U_i denotes the set of all admissible policies for the agent β_i [22]. Assuming that the minimizer in (6) exists, V_i^* is the solution to the HJB equation

$$H_i^* = r_i^*(\mathcal{E}_i, u_i^*) + \sum_{j\in i\cup\mathcal{N}_i} \frac{\partial V_i^*}{\partial e_j}\Upsilon_j(F^*) = 0, \tag{7}$$

where $F_i^*(x_i, u_i^*) \triangleq f_i(x_i) + g_i(x_i)u_i^*$, and the minimizer in (6) is the optimal policy $u_i^* : S^{|\mathcal{N}_i|+1} \to \mathbb{R}^m$, which can be obtained by solving the equation $\frac{\partial H_i^*(\mathcal{E}_i, u_i^*)}{\partial u_i^*} = 0$. Using the definition of H_i^* in (7), the optimal policy can be written in a closed form as

$$u_i^* = -\frac{1}{2}R_i^{-1}g_i^T\left((a_{i0}+d_i)\left(V_{ie_i}^*\right)^T - \sum_{j\in\mathcal{N}_i} a_{ji}\left(V_{ie_j}^*\right)^T\right), \tag{8}$$

where $V_{ie_i}^* \triangleq \frac{\partial V_i^*}{\partial e_i}$, and $V_{ie_j}^* \triangleq \frac{\partial V_i^*}{\partial e_j}$, assuming that the optimal value function V_i^* satisfies $V_i^* \in C^1$ and $V_i^*(0) = 0$.

Note that the controller for node i only requires the tracking error and edge weight information from itself and its neighbors. The following assumptions are made to facilitate the use of NNs to approximate the optimal policy and the optimal value function.

Assumption 1. The set S is compact. Based on the subsequent stability analysis, this assumption holds as long as the initial condition $\mathcal{X}(0)$ is bounded. See Remark 1 in the subsequent stability analysis.

Assumption 2. Each optimal value function V_i^* can be represented using a NN with M_i neurons as

$$V_i^*(\mathcal{E}_i) = W_i^T\sigma_i(\mathcal{E}_i) + \epsilon_i(\mathcal{E}_i), \tag{9}$$

where $W_i \in \mathbb{R}^{M_i}$ is the ideal weight matrix bounded above by a known positive constant $\bar{W}_i \in \mathbb{R}$ in the sense that $\|W_i\|_2 \le \bar{W}_i$, $\sigma_i : S^{|\mathcal{N}_i|+1} \to \mathbb{R}^{M_i}$ is a bounded continuously differentiable nonlinear activation function, and $\epsilon_i : S^{|\mathcal{N}_i|+1} \to \mathbb{R}$ is the function reconstruction error such that $\sup_{\mathcal{E}_i}|\epsilon_i(\mathcal{E}_i)| \le \bar{\epsilon}_i$ and $\sup_{\mathcal{E}_i}\|\epsilon_i'(\mathcal{E}_i)\| \le \bar{\epsilon'}_i$, where $\epsilon_i' = \frac{\partial\epsilon_i}{\partial\mathcal{E}_i}$ and $\bar{\epsilon}_i$, $\bar{\epsilon'}_i \in \mathbb{R}$ are positive constants [23], [24].

From (8) and (9) the optimal policy can be represented as

$$u_i^* = -\frac{1}{2}R_i^{-1}g_i^T(L_{\sigma i}W_i + L_{\epsilon i}), \tag{10}$$

where $L_{\sigma i} \triangleq \left((a_{i0}+d_i)\left(\frac{\partial\sigma_i}{\partial e_i}\right)^T - \sum_{j\in\mathcal{N}_i} a_{ji}\left(\frac{\partial\sigma_i}{\partial e_j}\right)^T\right)$, and $L_{\epsilon i} \triangleq \left((a_{i0}+d_i)\left(\frac{\partial\epsilon_i}{\partial e_i}\right)^T - \sum_{j\in\mathcal{N}_i} a_{ji}\left(\frac{\partial\epsilon_i}{\partial e_j}\right)^T\right)$.

Based on (9) and (10), the NN approximations to the optimal value function and the optimal policy are given by

$$\hat{V}_i = \hat{W}_{ci}^T\sigma_i, \quad u_i = -\frac{1}{2}R_i^{-1}g_i^T L_{\sigma i}\hat{W}_{ai}, \tag{11}$$

where $\hat{W}_{ci}(\cdot) \in \mathbb{R}^{M_i}$ and $\hat{W}_{ai}(\cdot) \in \mathbb{R}^{M_i}$ are estimates of the ideal neural network weights W_i. Using (9)-(11), the approximate Hamiltonian $\hat{H}_i(\cdot)$ and the optimal Hamiltonian $H_i^*(\cdot)$ can be obtained as

$$\hat{H}_i = e_i^T Q_{ii} e_i + u_i^T R_i u_i + \sum_{j=1}^{N} a_{ij} e_j^T Q_{ij} e_j + \hat{W}_{ci}^T\omega_i,$$
$$H_i^* = e_i^T Q_{ii} e_i + u_i^{*T} R_i u_i^* + \sum_{j=1}^{N} a_{ij} e_j^T Q_{ij} e_j + W_i^T\omega_i^* + \epsilon_{iF^*}', \tag{12}$$

where $\epsilon_{iF^*}' \triangleq \sum_{j\in i\cup\mathcal{N}_i}\left(\frac{\partial\epsilon_i}{\partial e_j}\right)\Upsilon_j(F^*)$ and

$$\omega_i \triangleq \sum_{j\in i\cup\mathcal{N}_i}\left(\frac{\partial\sigma_i}{\partial e_j}\right)\Upsilon_j\left(\hat{F}\right),$$
$$\omega_i^* \triangleq \sum_{j\in i\cup\mathcal{N}_i}\left(\frac{\partial\sigma_i}{\partial e_j}\right)\Upsilon_j(F^*). \tag{13}$$

Using (7), the error between the approximate and the optimal Hamiltonian, called the Bellman error (BE) $\delta_i(\cdot) \in \mathbb{R}$, is given in a measurable form by

$$\delta_i \triangleq \hat{H}_i - H_i^* = \hat{H}_i. \tag{14}$$

Note that equations (12)-(14) imply that to compute the BE, the i^{th} agent requires the knowledge of $\Upsilon_i\left(\hat{F}\right)$ and $\Upsilon_j\left(\hat{F}\right)$ for all $j \in \mathcal{N}_i$. As each agent can compute its own $\Upsilon_i\left(\hat{F}\right)$ based on local information, the computation of δ_i for each agent can be achieved via two-hop local communication.

The primary contribution of this result is that the developed value function approximation scheme, together with the state derivative estimator, enables the computation of the BE δ_i with only local information, and without the knowledge of drift dynamics. Furthermore, unlike the previous results such as [15], the effect of the local tracking errors of the neighbors of an agent is explicitly considered in the HJB equation for that agent, resulting in the novel control law in (11). In the following, the update laws for the value function and the policy weight estimates based on the BE are presented. The update laws and the subsequent development leading up to the stability analysis in Section V are similar to our previous result in [19] with minor changes, and are presented here for completeness.

Note that the BE in (14) is linear in the value function weight estimates $\hat{W}_{ci}$ and nonlinear in the policy weight estimates $\hat{W}_{ai}$. The use of two different sets of weights to approximate the same ideal weights W_i is motivated by the heuristic observation that adaptive update laws based on least squares minimization perform better than those based on gradient descent. As the application of least squares technique requires linearity of the error with respect to the parameters being estimated, the use of two different sets of weights facilitates the development of a least squares minimization-based update law for the value function weights. The value function weights are updated to minimize $\int_0^t \delta_i^2(\tau)\, d\tau$ using a least squares update law with a forgetting factor as [25], [26]

$$\dot{\hat{W}}_{ci} = -\phi_{ci}\gamma_i \frac{\omega_i}{1+\nu_i \omega_i^T \gamma_i \omega_i}\delta_i, \tag{15}$$

$$\dot{\gamma}_i = -\phi_{ci}\left(-\lambda_i \gamma_i + \gamma_i \frac{\omega_i \omega_i^T}{1+\nu_i \omega_i^T \gamma_i \omega_i}\gamma_i\right), \tag{16}$$

where $\nu_i, \phi_{ci} \in \mathbb{R}$ are positive adaptation gains, $\lambda_i \in (0,1)$ is the forgetting factor for the estimation gain matrix $\gamma_i(\cdot) \in \mathbb{R}^{M_i \times M_i}$. The policy weights are updated to follow the value function weight estimates as

$$\dot{\hat{W}}_{ai} = proj\left\{-\phi_{ai}\left(\hat{W}_{ai} - \hat{W}_{ci}\right)\right\}, \tag{17}$$

where $\phi_{ai} \in \mathbb{R}$ is a positive adaptation gain, and $proj\{\cdot\}$ is a smooth projection operator [21]. The use of forgetting factor ensures that

$$\underline{\varphi_i} I_{M_i} \leq \gamma_i(t) \leq \overline{\varphi_i} I_{M_i}, \quad \forall t \in [t_0, \infty), \tag{18}$$

where $\overline{\varphi_i}, \underline{\varphi_i} \in \mathbb{R}$ are constants such that $0 < \underline{\varphi_i} < \overline{\varphi_i}$ [25], [26]. The weight estimation errors for the value function and the policy are defined as $\tilde{W}_{ci}(t) \triangleq W_i - \hat{W}_{ci}(t)$ and $\tilde{W}_{ai}(t) \triangleq W_i - \hat{W}_{ai}(t)$, respectively. Using (14), the weight estimation error dynamics for the value function can be rewritten as

$$\begin{aligned}\dot{\tilde{W}}_{ci} = &-\phi_{ci}\gamma_i \psi_i \psi_i^T \tilde{W}_{ci} + \frac{\phi_{ci}\gamma_i \omega_i}{1+\nu_i \omega_i^T \gamma_i \omega_i}\Bigg(\\ &W_i^T\left(\sum_{j \in i \cup \mathcal{N}_i} \sigma_{ie_j} \Upsilon_j\left(\tilde{F}\right)\right) - \frac{1}{4}G_{\epsilon i} - \epsilon'_{iF^*} \\ &+ \frac{1}{2}W_i^T\left(\sum_{j \in i \cup \mathcal{N}_i} \sigma_{ie_j} \Upsilon_j\left(GL_\sigma \tilde{W}_a + GL_\epsilon\right)\right) \\ &+ \frac{1}{4}\tilde{W}_{ai}^T G_{\sigma i}\tilde{W}_{ai} - \frac{1}{2}\tilde{W}_{ai}^T G_{\sigma i} W_i - \frac{1}{2}W_i^T G_{\sigma\epsilon i}\Bigg),\end{aligned} \tag{19}$$

where $\sigma_{ie_j} \triangleq \frac{\partial \sigma_i}{\partial e_j}$, $G_i \triangleq g_i R_i^{-1} g_i^T$, $G_{\sigma i} \triangleq L_{\sigma i}^T g_i R_i^{-1} g_i^T L_{\sigma i}$, $G_{\epsilon i} \triangleq L_{\epsilon i}^T g_i R_i^{-1} g_i^T L_{\epsilon i}$, $G_{\sigma\epsilon i} \triangleq L_{\sigma i}^T g_i R_i^{-1} g_i^T L_{\epsilon i}$ and $\psi_i(\cdot) \triangleq \frac{\omega_i}{\sqrt{1+\nu_i \omega_i^T \gamma_i \omega_i}} \in \mathbb{R}^{M_i}$ is the regressor vector. Based on (18), the regressor vector can be bounded as

$$\|\psi_i(t)\| \leq \frac{1}{\sqrt{\nu_i \underline{\varphi_i}}}, \quad \forall t \in [t_0, \infty). \tag{20}$$

The dynamics in (19) can be regarded as a perturbed form of the nominal system

$$\dot{\tilde{W}}_{ci} = -\phi_{ci}\gamma_i \psi_i \psi_i^T \tilde{W}_{ci}. \tag{21}$$

Using Corollary 4.3.2 in [26] and Assumption 1, (21) is globally exponentially stable if the regressor vector $\psi_i : [0,\infty) \to \mathbb{R}^{M_i}$ is persistently exciting. Given (18), (20), and (21), Theorem 4.14 in [27] can be used to show that there exists a function $V_{ci} : \mathbb{R}^{M_i} \times [0,\infty) \to \mathbb{R}$ and positive constants $\underline{v_{ci}}$, $\overline{v_{ci}}$, v_{c1i} and v_{c2i} such that for all $t \in [t_0, \infty)$,

$$\underline{v_{ci}}\left\|\tilde{W}_{ci}\right\|^2 \leq V_{ci}\left(\tilde{W}_{ci}, t\right) \leq \overline{v_{ci}}\left\|\tilde{W}_{ci}\right\|^2, \tag{22}$$

$$\frac{\partial V_{ci}}{\partial \tilde{W}_{ci}}\left(-\phi_{ci}\gamma_i \psi_i \psi_i^T \tilde{W}_{ci}\right) + \frac{\partial V_{ci}}{\partial t} \leq -v_{c1i}\left\|\tilde{W}_{ci}\right\|^2, \tag{23}$$

$$\frac{\partial V_{ci}}{\partial \tilde{W}_{ci}} \leq v_{c2i}\left\|\tilde{W}_{ci}\right\|. \tag{24}$$

Using Assumptions 1, and 2, the results of Section IV-A, and the fact the $\hat{W}_{ai}$ is bounded by projection, the following

bounds are developed to aid the subsequent stability analysis:

$$\left\| \frac{1}{4}\tilde{W}_{ai}^T G_{\sigma i}\tilde{W}_{ai} - \frac{1}{2}\tilde{W}_{ai}^T G_{\sigma i} W_i - \frac{1}{2}W_i^T G_{\sigma\epsilon i} + W_i^T \left(\sum_{j\in i\cup\mathcal{N}_i} \sigma_{ie_j} \Upsilon_j\left(\tilde{F}\right)\right) - \frac{1}{4}G_{\epsilon i} - \epsilon'_{iF^*} \right\| \leq \iota_1,$$

$$\left\| \frac{1}{2}W_i^T \left(\sum_{j\in i\cup\mathcal{N}_i} \sigma_{ie_j} \Upsilon_j\left(GL_\sigma\tilde{W}_a + GL_\epsilon\right)\right)\right\| \leq \iota_2,$$

$$\left\| \sum_{j=i\wedge j\in\mathcal{N}_i} \frac{\partial \epsilon_i}{\partial e_j}\Upsilon_j\left(\frac{1}{2}GL_\sigma\tilde{W}_a + \frac{1}{2}GL_\epsilon\right)\right\| \leq \iota_3,$$

$$\left\|\tilde{W}_{ai}\right\| \leq \iota_4, \tag{25}$$

where $\iota_1, \iota_2, \iota_3, \iota_4 \in \mathbb{R}$ are computable positive constants.

V. Stability Analysis

Theorem 1. *Provided Assumptions 1 and 2 hold, and the regressor vector $\psi_i : [0,\infty) \to \mathbb{R}^{M_i}$ is persistently exciting, the controller in (11) and the update laws in (15) - (17) guarantee that the local neighborhood tracking errors for agent β_i and its neighbors are UUB. Furthermore, the policy and the value function weight estimation errors for agent β_i are UUB, resulting in UUB convergence of the policy u_i to the optimal policy u_i^*.*

Proof: Consider the function $V_{Li} : S^{|\mathcal{N}_i|+1} \times \mathbb{R}^{2M_i} \times \mathbb{R}^+ \to \mathbb{R}$ defined as

$$V_{Li} \triangleq V_i^* + V_{ci} + \frac{1}{2}\tilde{W}_{ai}^T\tilde{W}_{ai},$$

where V_i^* is defined in (6) and V_{ci} is introduced in (22). Using the fact that V_i^* is positive definite, Lemma 4.3 from [27] and (22) yield

$$\underline{v_{li}}\left(\|Z_i\|\right) \leq V_{Li}\left(Z_i, t\right) \leq \overline{v_{li}}\left(\|Z_i\|\right), \tag{26}$$

for all $Z_i \in B_{bi}$ and for all $t \in [t_0, \infty)$, where

$$Z_i \triangleq \begin{bmatrix}\mathcal{E}_i & \tilde{W}_{ci}^T & \tilde{W}_{ai}^T\end{bmatrix}^T \in \mathcal{Z} \subseteq S^{|\mathcal{N}_i|+1} \times \mathbb{R}^{2M_i},$$

$\underline{v_{li}} : [0, b_i] \to [0,\infty)$ and $\overline{v_{li}} : [0, b_i] \to [0,\infty)$ are class $\mathcal{K}$ functions, and $B_{bi} \subset \mathcal{Z}$ denotes a ball of radius $b_i \in \mathbb{R}^+$ around the origin. The time derivative of V_{Li} is

$$\dot{V}_{Li} = \sum_{j\in i\cup\mathcal{N}_i} V_{ie_j}^* \Upsilon_j\left(F^*\right) + \sum_{j\in i\cup\mathcal{N}_i} V_{ie_j}^* \Upsilon_j\left(g\left(u-u^*\right)\right) + \left(\frac{\partial V_{ci}}{\partial \tilde{W}_{ci}}\dot{\tilde{W}}_{ci} + \frac{\partial V_{ci}}{\partial t}\right) - \left(\tilde{W}_{ai}^T\dot{\hat{W}}_{ai}\right).$$

Using (19), (17) and the fact that from (7), $\sum_{j\in i\cup\mathcal{N}_i} \frac{\partial V_i^*}{\partial e_j}\Upsilon_j\left(F^*\right) = -r_i^*$ yields

$$\begin{aligned}\dot{V}_{Li} = {} & -e_i^T Q_{ii} e_i - u_i^{*T} R_i u_i^* - \sum_{j\in\mathcal{N}_i} a_{ij} e_j^T Q_{ij} e_j \\ & + \sum_{j\in i\cup\mathcal{N}_i} \frac{\partial V_i^*}{\partial e_j}\Upsilon_j\left(\frac{1}{2}GL_\sigma\tilde{W}_a + \frac{1}{2}GL_\epsilon\right) + \frac{\partial V_{ci}}{\partial t} \\ & - \frac{\partial V_{ci}}{\partial \tilde{W}_{ci}}\phi_{ci}\gamma_i\psi_i\psi_i^T\tilde{W}_{ci} + \tilde{W}_{ai}^T\phi_{ai}\left(\hat{W}_{ai} - \hat{W}_{ci}\right) \\ & + \frac{\partial V_{ci}}{\partial \tilde{W}_{ci}}\frac{\phi_{ci}\gamma_i\omega_i}{1+\nu_i\omega_i^T\gamma_i\omega_i}\Bigg(\frac{1}{4}\hat{W}_{ai}^T G_{\sigma i}\hat{W}_{ai} \\ & - \frac{1}{4}G_{\epsilon i} - \frac{1}{2}W_i^T G_{\sigma\epsilon i} + W_i^T\left(\sum_{j\in i\cup\mathcal{N}_i}\sigma_{ie_j}\Upsilon_j\left(\tilde{F}\right)\right) \\ & + \frac{1}{2}W_i^T\left(\sum_{j\in i\cup\mathcal{N}_i}\sigma_{ie_j}\Upsilon_j\left(GL_\sigma\tilde{W}_a + GL_\epsilon\right)\right) \\ & - \frac{1}{4}W_i^T G_{\sigma i}W_i - \epsilon'_{iF^*}\Bigg).\end{aligned} \tag{27}$$

Using the bounds in (23)-(25) the Lyapunov derivative in (27) can be upper-bounded as

$$\dot{V}_{Li} \leq -\underline{Q_{ii}}\left\|e_i\right\|^2 - \sum_{j\in\mathcal{N}_i} a_{ij}\underline{Q_{ij}}\left\|e_j\right\|^2 - v_{c1i}\left\|\tilde{W}_{ci}\right\|^2 - \phi_{ai}\left\|\tilde{W}_{ai}\right\|^2 + \iota_{\tilde{W}_{ci}}\left\|\tilde{W}_{ci}\right\| + \iota_2 + \iota_3, \tag{28}$$

where $\underline{Q_{ii}}$ and $\underline{Q_{ij}}$, are the minimum eigenvalues of the matrices Q_{ii} and Q_{ij}, respectively and

$$\iota_{\tilde{W}_{ci}} = \frac{\phi_{ci}v_{c2i}\overline{\varphi_i}}{\sqrt{\nu_i\underline{\varphi_i}}}\left(\iota_1+\iota_2\right) + \phi_{ai}\iota_4.$$

Lemma 4.3 in [27] along with completion of the squares on $\left\|\tilde{W}_{ci}\right\|$ in (28) yields

$$\dot{V}_{Li}\left(Z_i, t\right) \leq -v_{li}\left(\|Z_i\|\right), \ \forall \|Z_i\| \geq \iota_{5i} > 0, \ \forall t \in [0,\infty) \tag{29}$$

where $\iota_{5i} = v_{li}^{-1}\left(\frac{\iota_{\tilde{W}_{ci}}^2}{2v_{c1i}} + \iota_2 + \iota_3\right)$, and $v_{li} : [0, b_i] \to [0,\infty)$ is a class $\mathcal{K}$ function. Using (26), (29), and Theorem 4.18 in [27], $Z_i\left(t\right)$ is UUB. ■

The conclusion of Theorem 1 is that the local neighborhood tracking errors for agent β_i and its neighbors are UUB. Since the choice of agent β_i is arbitrary, similar analysis on each agent shows that the local neighborhood tracking errors for all the agents are UUB. Hence $\mathcal{E}\left(t\right)$ is UUB. Provided that the graph has a spanning tree and at least one of the pinning gains a_{i0} is nonzero it can be shown that [20], [28],

$$\|\mathcal{X}\| \leq \|\mathcal{E}\|/s, \tag{30}$$

where s is the minimum singular value of the matrix $\mathcal{L}+\mathcal{A}_0$. Thus, Theorem 1 along with (30) shows that the states $x_i \mid i = 1, \cdots, N$ are UUB around the origin. Based on (25), the ultimate bound can be made smaller by increasing the

state penalties Q_{ii} and Q_{ij}, and by increasing the number of neurons in the NN approximation of the value function to reduce the approximation errors ϵ_i.

Remark 1. If $\|Z_i(0)\| \geq \iota_{5i}$ then $\dot{V}_{Li}(Z_i(0),0) < 0$. Thus, $V_{Li}(Z_i(t),t)$ is decreasing at $t = 0$. Thus, $Z_i(t) \in \mathcal{L}_\infty$, and hence, $\mathcal{E}_i(t) \in \mathcal{L}_\infty$ at $t = 0^+$. Thus all the conditions of Theorem 1 are satisfied at $t = 0^+$. As a result, $V_{Li}(Z_i(t),t)$ is decreasing at $t = 0^+$. By induction, $\|Z_i(0)\| \geq \iota_{5i} \implies V_{Li}(Z_i(t),t) \leq V_{Li}(Z_i(0),0), \forall t \in \mathbb{R}^+$. Thus, from (26), $\|\mathcal{E}_i(t)\| \leq \|Z_i(t)\| \leq \underline{v_{li}}^{-1}(\overline{v_{li}}(\|Z_i(0)\|))$. If $\|Z_i(0)\| < \iota_{5i}$ then (26) and (29) can be used to determine that $\underline{v_{li}}(\|Z_i(t)\|) \leq V_{Li}(Z_i(t),t) \leq \overline{v_{li}}(\|\iota_{5i}\|), \forall t \in \mathbb{R}^+$. As a result, $\|Z_i(t)\| \leq \underline{v_{li}}^{-1}(\overline{v_{li}}(\iota_{5i}))$. Let $\overline{S} \in \mathbb{R}$ be defined as

$$\overline{S} \triangleq \frac{\sum_{i=1}^{N} \underline{v_{li}}^{-1}(\overline{v_{li}}(\max(\|Z_i(0)\|, \iota_{5i})))}{s}.$$

This relieves Assumption 1 in the sense that the compact set $S \subset \mathbb{R}^n$ that contains the system trajectories $x_i(t), \forall i = 1, \cdots, N, \forall t \in \mathbb{R}^+$ is given by $S \triangleq \left\{x \in \mathbb{R}^n \mid \|x\| \leq \overline{S}\right\}$.

VI. Conclusion

This result combines graph theory and graph theory with the ACI architecture in ADP to synthesize approximate online optimal control policies for agents on a communication network with a spanning tree. NNs are used to approximate the policy, the value function, and the system dynamics. UUB convergence of the agent states and the weight estimation errors is proved through a Lyapunov-based stability analysis. Like other ADP-based results, this result hinges on the system states being PE. Furthermore, possible obstacles and possible collisions are ignored in this work. Future efforts will focus to resolve these limitations.

References

[1] M. Lauer and M. A. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems," in *Proc. Int. Conf. Mach. Learn.*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 535–542.

[2] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 38, no. 2, pp. 156–172, 2008.

[3] G. Weiβ, "Distributed reinforcement learning," *Robot. Autom. Syst.*, vol. 15, no. 1-2, pp. 135 – 142, 1995, the Biology and Technology of Intelligent Autonomous Agents.

[4] W. Cao, G. Chen, X. Chen, and M. Wu, "Optimal tracking agent: A new framework for multi-agent reinforcement learning," in *Proc. IEEE Int. Conf. Trust Secur. Priv. Comput. Commun.*, 2011, pp. 1328–1334.

[5] K. Vamvoudakis and F. Lewis, "Multi-player non-zero-sum games: Online adaptive learning solution of coupled hamilton-jacobi equations," *Automatica*, vol. 47, pp. 1556–1569, 2011.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[7] K. Vamvoudakis, F. L. Lewis, M. Johnson, and W. E. Dixon, "Online learning algorithm for stackelberg games in problems with hierarchy," in *Proc. IEEE Conf. Decis. Control*, Maui, HI, Dec. 2012, pp. 1883–1889.

[8] K. Vamvoudakis and F. Lewis, "Online neural network solution of nonlinear two-player zero-sum games using synchronous policy iteration," in *Proc. IEEE Conf. Decis. Control*, 2010.

[9] D. Vrabie and F. Lewis, "Integral reinforcement learning for online computation of feedback nash strategies of nonzero-sum differential games," in *Proc. IEEE Conf. Decis. Control*, 2010, pp. 3066–3071.

[10] M. Johnson, T. Hiramatsu, N. Fitz-Coy, and W. E. Dixon, "Asymptotic stackelberg optimal control design for an uncertain Euler-Lagrange system," in *Proc. IEEE Conf. Decis. Control*, Atlanta, GA, 2010, pp. 6686–6691.

[11] M. Johnson, S. Bhasin, and W. E. Dixon, "Nonlinear two-player zero-sum game approximate solution using a policy iteration algorithm," in *Proc. IEEE Conf. Decis. Control*, 2011, pp. 142–147.

[12] J. Wang and M. Xin, "Multi-agent consensus algorithm with obstacle avoidance via optimal control approach," *Int. J. Control*, vol. 83, no. 12, pp. 2606–2621, 2010.

[13] ——, "Distributed optimal cooperative tracking control of multiple autonomous robots," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 572 – 583, 2012.

[14] E. Semsar-Kazerooni and K. Khorasani, "Optimal consensus algorithms for cooperative team of agents subject to partial information," *Automatica*, vol. 44, no. 11, pp. 2766 – 2777, 2008.

[15] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598 – 1611, 2012.

[16] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *Proc. IEEE Conf. Decis. Control*, vol. 4, 2003, pp. 3621–3626.

[17] L. Magni and R. Scattolini, "Stabilizing decentralized model predictive control of nonlinear systems," *Automatica*, vol. 42, no. 7, pp. 1231 – 1236, 2006.

[18] P. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand Reinhold, 1992.

[19] S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. L. Lewis, and W. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 89–92, 2013.

[20] S. Khoo and L. Xie, "Robust finite-time consensus tracking algorithm for multirobot systems," *IEEE/ASME Trans. Mechatron.*, vol. 14, no. 2, pp. 219–228, 2009.

[21] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.

[22] R. Beard, G. Saridis, and J. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, pp. 2159–2178, 1997.

[23] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Netw.*, vol. 3, no. 5, pp. 551 – 560, 1990.

[24] F. L. Lewis, R. Selmic, and J. Campos, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002.

[25] R. M. Johnstone, C. R. Johnson, R. R. Bitmead, and B. D. O. Anderson, "Exponential convergence of recursive least squares with exponential forgetting factor," in *Proc. IEEE Conf. Decis. Control*, vol. 21, 1982, pp. 994–997.

[26] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.

[27] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[28] G. Chen and F. L. Lewis, "Distributed adaptive tracking control for synchronization of unknown networked Lagrangian systems," *IEEE Trans. Syst. Man Cybern.*, vol. 41, no. 3, pp. 805–816, 2011.