Cooperative adaptive model-free control with model-free estimation and online gain tuning

Ali Safaei, *Senior Member, IEEE*

***Abstract*—In this paper, a distributed adaptive model-free control algorithm is proposed for consensus and formation-tracking problems in a network of agents with completely unknown nonlinear dynamic systems. The specification of the communication graph in the network is incorporated in the adaptive laws for estimation of the unknown linear and nonlinear terms, and also in the online updating of the elements in the main controller gain matrix. The decentralized control signal at each agent in the network requires the information about the states of the leader agent, as well as the desired formation variables of the agents in a local coordinate frame. These two sets of variables are provided at each agent by utilizing two recently-proposed distributed observers. It is shown that only a spanning-tree rooted at the leader agent is enough for the convergence and stability of the proposed cooperative control and observer algorithms. Two simulation studies are provided to evaluate the performance of the proposed algorithm in comparison with two state-of-the-art distributed model-free control algorithms. With lower control effort as well as fewer offline gain tuning, same level of consensus errors are achieved. Finally, the application of the proposed solution is studied in the formation-tracking control of a team of autonomous aerial mobile robots via simulation results.**

***Index Terms*—Cooperative Control, Adaptive Control, Model-free algorithm, autonomous mobile robots, online gain tuning**

I. INTRODUCTION

Comparing to single-agent systems, a multi-gent system can deliver benefits like extended functionality, extended time and area of operation and can provide new applications [1], [2], [3]. A multi-agent dynamic system is considered as a *network* or *swarm* of multiple dynamic systems having a mutual objective and operating in a *cooperative* manner [4], [5]. This cooperation requires the inter-agent communication in the network [6], [7]. In recent years, great attention has been paid to problems of cooperative control for multi-agent systems, such as consensus and flocking movements, containment control, formation control and leader-following control [4], [8]. One of the issues in this field is having agents with unknown internal dynamics and external disturbances.

Model-free controllers (MFCs) which are data-driven techniques to address the issue of unknown dynamics, have been developed for single-agent systems extensively [9], [10], [11], [12]. In a MFC algorithm, a generic structure is considered for the entire dynamics and disturbances of the system, while a portion or all parameters of the structure are estimated online based on input-output data set of the system. MFC algorithms are against the model-based controllers, in which requirement of a completely known dynamic system is essential for defining the control law. In last decade, several works are proposed to solve the cooperative consensus and formation-tracking problems in multi-agent systems with partially or completely unknown nonlinear dynamics [7], [14]. Similar to model-free concept for controllers, a parameter estimator can be either model-based or model-free. The online estimators that use a model (such as artificial neural network (ANN) and fuzzy inference system (FIS)) to represent the unknown terms of dynamic system, are named as model-based estimators. Apparently, online estimators which do not rely on a model and just estimate the unknown terms as a whole, are model-free estimators. The above concepts for design of cooperative model-free control algorithms and parameter estimators in unknown dynamic systems are reviewed in the following.

A. Safaei is a postdoctoral researcher at department of Mechanical Engineering, McGill University, Montreal, QC, Canada (e-mail: ali.safaei@mail.mcgill.ca)

A. Cooperative MFCs with model-based estimator

Lewis et al. have proposed a *distributed* adaptive control protocol in [7], [14] for *synchronization* (i.e. consensus) problem in a network of single-integrator and double-integrator unknown nonlinear dynamic systems. The protocol is a distributed or *decentralized* algorithm (not a centralized one), since it defines control laws at each agent for achieving a consensus in the network. By utilizing an ANN, the proposed control laws in [7], [14] incorporate the estimated values of the unknown terms. That work inspired other researchers for investigating the novel distributed protocols for consensus and formation-tracking problems. In [16], the design procedure of a distributed state-output feedback cooperative control algorithm is reported for a network of multiple uncertain agents with directed communication graphs. Similarly, ANNs are utilized by [17], [18], [21] to estimate unknown nonlinear terms at each agent of the network. In [19], [20], besides the use of ANNs, the controller gains matrix at each agent are defined off-line by solution of a continuous-time algebraic Riccati equation (CARE). The algorithm proposed in [22] for a formation-tracking problem, includes ANNs for estimating the unknown internal nonlinear dynamics of the agents, as well as a robust term to cancel the bounded external disturbances.

Uniformly ultimately bounded (UUB) convergence of the consensus errors and consequently that of the control inputs is proposed in [24], [28] by utilizing ANNs for online parameter estimation. In [25], ANNs are incorporated locally to reconstruct uncertain dynamics agents and unknown dynamics of a leader agent. It is assumed that the leader states are available to only a portion of the agents in the network. Later, in the works presented in [26] and [27], the Linear Matrix Inequality

(LMI) is employed to find the values of controller gains in an offline manner, while ANNs are used for online estimation of the unknown dynamics in the network. Compared to all other cooperative controllers with UUB convergence property, solution presented in [29] provides an asymptotic convergence on the consensus errors for a high-order completely unknown multi-agent dynamic system. This is delivered by assuming an extra constraint on the boundedness of the second time-derivative of unknown internal dynamics and external disturbances at agents. A solution is suggested in [30] to address the consensus problem, utilizing two sets of ANNs at each agent for estimating system states and unknown internal dynamics.

Besides the use of ANNs, FIS networks have been proposed for incorporating the fuzzy techniques for online estimation of unknown dynamics in cooperative control of multi-agent systems [32], [33], [34]. Moreover, the radial-basis function neural networks (RFBNNs), which utilize the fuzzy-type activation functions inside the neurons, are also used in several investigations for online parameter estimation [35], [36], [37]. Although the techniques are slightly different from ANNs, the basics of using them are the same.

In all above works, model-based tools are utilized for online parameter estimations in a network. The regressor parameters for each model are defined based on the corresponding consensus errors. This leads to the requirement of persistently excitation (PE) condition for input signals of the system at each agent, as well as the larger number of neurons. This increases the computation complexity and consequently the control efforts of the solution.

B. Reinforcement learning as cooperative MFC

Reinforcement learning (RL) techniques are also used to solve online optimal tracking control problem in unknown multi-agent dynamic systems [15]. These methods should be considered as model-free control algorithms with model-based parameter estimators, as they utilize ANNs for online estimating of parameters within control law and value function.

A distributed extension of *adaptive dynamic programming* algorithm is utilized in [31] to design a cooperative control policy in a network of partially unknown nonlinear dynamic systems. In [38], two separate ANNs with associated laws for online adaptation of the weights locally at each agent are used for estimating the unknown nonlinear dynamics and reinforcement control signal. In the work presented in [39], the off-policy RL algorithm is utilized for providing a solution for the consensus problem in a network of completely unknown linear dynamic systems. Moreover, online extension of *policy iteration* algorithm is proposed in [40] to deliver an optimal solution for consensus problem. In that work, the value function is approximated with a set of ANNs and weights of neural nodes are updated online utilizing a least squares technique over the input-output data gathered in a definite time window of the previous network operation. Then, a policy iteration algorithm is used for updating the optimal control inputs. In addition, a cooperative RL algorithm is proposed in [41] for optimal synchronization in a heterogeneous multi-agent system with unknown nonlinear dynamics. Also, a decentralized cooperative observer is designed for estimating leader states at agents. Then, two ANNs are utilized for estimating value function and distributed optimal control signal.

As it is observed, RL techniques used as cooperative controllers for multi-agent systems, utilize the model-based estimators for constructing the control law. Thus, the disadvantages of model-based estimation, are still counted for them.

C. Cooperative MFCs without model-based estimator

To resolve the disadvantages of having model-based estimators within model-free control solutions, some model-free estimator methods are suggested. A distributed synchronization solution is proposed in [42], using the dynamic-surface control (DSC) technique. Despite decreasing the complexity, the algorithm still relies on ANNs. The leader-follower consensus control problem is solved in [43] for a network of nonlinear dynamic agents with polynomial functions as the assumed structure for internal nonlinearities. While this solution does not use model-based estimators, it is designed only for especial case of polynomial nonlinearities.

A decentralized synchronization protocol is proposed in [44], in which no online estimation of the nonlinear dynamics is investigated. Instead, an exponential dynamics is proposed for upper bound of the absolute consensus errors in network. The proposed dynamic bound is determined according to the desired performance of the system in transient and steady state situations. In this regard, neither the prior knowledge of agents' nonlinearities nor the popular online model-based estimation tools are incorporated. However, since no adaptive laws are proposed for updating main controller gains of the algorithm, the gains need to be determined off-line via expert knowledge. This decreases the convenience of the swarm implementation. Moreover, the prescribed performance as the dynamic upper bound of consensus errors is used in [45]. The adaptive projection algorithm is considered for online estimation of unknown nonlinear terms in the agents dynamics. The proposed adaptive law for online estimation is driven by a transformed consensus error, which is a nonlinear exponential function of the ratio of consensus error to the system prescribed performance. Thus, the need for nonlinear basis functions is still evident for online parameter estimation. In [46], [47], model-free iterative learning control algorithms are presented to solve formation-tracking problem in a network of completely unknown non-affine nonlinear dynamic systems. There is an assumption that the whole unknown nonlinear dynamics at each agent is non-zero and can be modeled using the compact form dynamic linearizing approach. Hence, the PE condition is required. A robust adaptive dynamic programming algorithm is presented in [48] for cooperative output regulation problem. That algorithm incorporates an iterative CARE to design distributed control signals. Controller gains are approximated online through two consecutive iterating loops. Thus, it requires huge computations during operation, leading to inefficient energy consumption in network.

D. Current work

A cooperative adaptive model-free control (CAMFC) algorithm is presented in this work. CAMFC can be considered as a cooperative model-free controller with model-free estimations for unknown internal dynamics and external disturbances. Moreover, CAMFC algorithm is equipped with a method for online tuning of the main controller gain. In this regard, the main novelties of the CAMFC are reviewed as:

- compared to solutions based on model-based estimators (including RL techniques), the need for PE condition as well as regressor parameters is revoked. Consequently, more convenient implementation of the cooperative controller for a swarm of dynamic agents would be provided.
- compared to solutions with model-based estimators, the computation cost of cooperative controller is decreased by having a lower number of adaptive laws for estimating the unknown internal dynamics and unknown bounded external disturbances at agents.
- by incorporating a DRE which is characterized based on communication graph of the network and is adapted based on the estimated system dynamics, the main controller gains are updated online. This leads to more convenience implementation in a swarm.

In addition, in comparison with our recently published works in [51], [52], CAMFC has the following advantages:

- while previous solutions require measurements over states at each agent, CAMFC relies on relative measurements between the neighboring agents. This is important in applications like a swarm of mobile robots, where accurate position measurements may not be available at all agents.
- adaptive laws for estimating unknown linear terms in previous methods were not distributed, and was a model-based estimator for the design in [52]. But, all adaptive laws in CAMFC are distributed model-free estimators that rely on consensus errors of neighboring agents.
- while control inputs at neighboring agents are required and should be estimated at each agent in the design in [52], CAMFC does not need any exchange of control inputs or their estimated values among the agents. This decrease the computation complexity of the solution.

In the following, first CAMFC with corresponding definitions, assumptions and stability analysis are presented in Section II. A comparative analysis including two case-studies is provided in Section III. Finally, application of CAMFC on a swarm of four quadrotors is presented in Section IV.

II. CAMFC DESIGN PROCEDURE

Definition-1. Consider a network including N agents with completely unknown nonlinear dynamics. Referring to the discussions in [12], [50], [51], [52], the dynamics of agent $i \in [1, N]$ can be defined as follows

$$\dot{x}^i = A^i x^i + B u^i + g^i \ , \tag{1}$$

where $x^i = [x^i_1, x^i_2, ..., x^i_n]^T \in \mathbb{R}^{n\times 1}$ is the vector of states at agent i, $u^i = [u^i_1, u^i_2, ..., u^i_n]^T \in \mathbb{R}^{n\times 1}$ is the vector of control inputs for that agent, $A^i = diag(A^i_1, A^i_2, ..., A^i_n) \in \mathbb{R}^{n\times n}$ is the corresponding unknown system matrix and $g^i = [g^i_1, g^i_2, ..., g^i_n]^T \in \mathbb{R}^{n\times 1}$ is the vector of unknown nonlinear terms including the internal dynamics nonlinearities and the external disturbances at agent i. Here, n is the number of states at each agent in the network. As mentioned in [50], [52], u^i includes some virtual control inputs so as to have B equal to an identity matrix, without any loss of generality for the nonlinear system dynamics [12], [50]. Hence,

$$\dot{x}^i = A^i x^i + u^i + g^i \ . \tag{2}$$

The actual control variables at each agent can be extracted according to the specific applications. Examples for a 2D robotic manipulator, a wheeled mobile robot and a quadrotor are presented in [53], [50], respectively. Furthermore, the lumped dynamics of agents can be represented as follows

$$\dot{x} = Ax + u + g \ , \tag{3}$$

where $x = [x^1, x^2, ..., x^N]^T$, $u = [u^1, u^2, ..., u^N]^T$, $g = [g^1, g^2, ..., g^N]^T$ in $\mathbb{R}^{Nn\times 1}$; $A = diag(A^1, A^2, ..., A^N) \in \mathbb{R}^{Nn\times Nn}$.

Assumption-1. Internal dynamics and external disturbances at each agent, A^i and g^i, and their first time-derivatives are assumed bounded, while boundaries are completely unknown.

Remark-1. The concept of including all nonlinearities and disturbances of dynamic systems into single term g^i is presented in [49] for confirming robustness of the proposed control algorithm against drifting. It is shown that by satisfying a certain condition on magnitudes of nonlinearities and disturbances of the system, if the optimal cost function is chosen as the candidate Lyapunove function, then the design of a stable control algorithm would lead to a robust controller [49].

Definition-2. For the network of agents with the defined dynamics in *Definition-1*, a communication graph is defined with a *Laplacian* matrix as

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \ , \tag{4}$$

where, $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ is the corresponding *adjacency* matrix with $a_{ij} \in \{0, 1\}$ revealing the existing communication links among the agents, and $\mathcal{D} = diag(d_1, d_2, ...d_N) \in \mathbb{R}^{N\times N}$ with $d_i = \sum_{j=1}^{N} a_{ij}$. Moreover, a pinning-gain matrix is defined as $\mathcal{B} = diag(b_1, b_2, ...b_N) \in \mathbb{R}^{N\times N}$ with $b_i \in \{0, 1\}$, expressing the existence of communication between the virtual leader and agents. Then, one can define

$$\mathcal{H} = \mathcal{B} + \mathcal{L} \ . \tag{5}$$

$\mathcal{H}$ is a matrix including information on inter-agent and agent-to-leader communication links of the communication graph.

Assumption-2. The communication graph corresponding to the network defined in *Definition-1* and *Definition-2* is assumed to be strongly-connected and includes undirected communication links. In this regard, network has a spanning-tree rooted at the only agent who is connected to a virtual leader. Thus, $\mathcal{B}$ has at least one non-zero diagonal element.

Definition-3. Desired path for the whole network is defined as the states of virtual leader, i.e $x^0 \in \mathbb{R}^{n\times 1}$.

Definition-4. The desired formation topology among agents in the network is defined as

$$\Omega = \begin{bmatrix} \eta^0 & \eta^1 & \eta^2 & \dots & \eta^N \end{bmatrix}^T \in \mathbb{R}^{(N+1)\times n} \ . \tag{6}$$

In a local frame, $\eta^i \in \mathbb{R}^{n\times 1}$ is defined as vector of desired formation variables for all states at agent i with respect to virtual leader, and η^0 is formation variable for virtual leader.

Proposition-1. The objective is to achieve the desired formation topology among the agents in the network, while they are tracking the desired trajectory available to the virtual leader. This can be achieved by converging the following formation-tracking consensus errors at all agents to zero

$$e^i = \sum_{j=1}^{N} a_{ij}((\eta^i - \eta^j) - (x^i - x^j)) + b_i((\eta^i - \eta^0) - (x^i - x^0)) \ . \quad (7)$$

By defining the transformed state $z^i = \eta^i - x^i$, one reaches

$$e^i = \sum_{j=1}^{N} a_{ij}(z^i - z^j) + b_i(z^i - z^0) \ , \quad (8)$$

where $z^0 = \eta^0 - x^0$. Furthermore, lumped consensus error for the whole network is proposed as

$$e = (\mathcal{H} \otimes I_n)z - (\mathcal{B} \otimes z^0)\mathbf{1} \ , \quad (9)$$

where $\otimes$ is the Kronecker product, $e = [e^1, e^2, ..., e^N]^T \in \mathbb{R}^{Nn\times 1}$, $z = [z^1, z^2, ..., z^N]^T \in \mathbb{R}^{Nn\times 1}$, I_n is an identity matrix in $\mathbb{R}^{n\times n}$ and $\mathbf{1} = [1, 1, ..., 1]^T \in \mathbb{R}^{N\times 1}$. Moreover, one can express $z = \eta - x$, where $\eta = [\eta^1, \eta^2, ..., \eta^N]^T \in \mathbb{R}^{Nn\times 1}$.

Definition-5. We define a vectorizing function $\mathcal{V}(.)$ for generating a vector $l \in \mathbb{R}^{n_0\times 1}$ constructed by diagonal elements of a matrix $M \in \mathbb{R}^{n_0\times n_0}$ as $l = \mathcal{V}(M) = v_M$, where $l[i] = M[i, i]$ for $i \in \{1, 2, ..., n_0\}$ [50].

Definition-6. We define a function $\mathcal{M}(.)$ for generating a diagonal matrix with zero off-diagonal elements $M \in \mathbb{R}^{n_0\times n_0}$ constructed by the elements of a vector $l \in \mathbb{R}^{n_0\times 1}$ as $M = \mathcal{M}(l) = \mathcal{M}_l$, where $M[i, i] = l[i]$ for $i \in \{1, 2, ..., n_0\}$ [50].

Theorem-1. Considering the network defined in *Definition-1* and *Definition-2*, with *Assumption-1* and *Assumption-2*, if control signal at each agent is defined as follows

$$u^i = \frac{1}{2}P^i e^i + \hat{A}^i c^i - \hat{g}^i - \dot{c}^i \ , \quad (10)$$

with $c^i = \eta^0 - \eta^i - x^0$; and with two adaptive laws as

$$\dot{\hat{g}}^i = -\Gamma_1(\sum_{j=1}^{N} \mathcal{H}(i, j)P^j e^j) - \rho_1\Gamma_1\hat{g}^i \ , \quad (11a)$$

$$\dot{v}_{\hat{A}^i} = \Gamma_2\mathcal{M}_{c^i}(\sum_{j=1}^{N} \mathcal{H}(i, j)P^j e^j) - \rho_2\Gamma_2 v_{\hat{A}^i} \ , \quad (11b)$$

where Γ_1 and Γ_2 are two diagonal matrices in $\mathbb{R}^{n\times n}$ including the adaptive gains, while ρ_1 and ρ_2 are two positive scalar leakage gains; and by updating $P^i \in \mathbb{R}^{n\times n}$ using

$$\dot{P}^i = \mathcal{H}(i, i)P^i P^i - 2\hat{A}^i P^i - Q \ , \quad (12)$$

where $Q \in \mathbb{R}^{n\times n}$ is a positive definite matrix; then the objective proposed in *Proposition-1* would be satisfied.

Proof. Consider the following positive cost function

$$V = \frac{1}{2}e^T P e + \frac{1}{2}\tilde{g}^T\Gamma_{1c}^{-1}\tilde{g} + \frac{1}{2}v_{\tilde{A}}^T\Gamma_{2c}^{-1}v_{\tilde{A}} \ , \quad (13)$$

where $P = diag(P^1, P^2, ..., P^N)$, $\Gamma_{1c} = (I_N \otimes \Gamma_1)$ and $\Gamma_{2c} = (I_N \otimes \Gamma_2)$, are all defined in $\mathbb{R}^{Nn\times Nn}$. In addition, $\tilde{g} = g - \hat{g}$ and $\tilde{A} = A - \hat{A}$ are defined as the estimation errors for unknown linear and nonlinear terms, respectively. Continuing by time-derivative of the equation in (13), one can reach at

$$\dot{V} = e^T P\dot{e} + \frac{1}{2}e^T\dot{P}e + \tilde{g}^T\Gamma_{1c}^{-1}\dot{\tilde{g}} + v_{\tilde{A}}\Gamma_{2c}^{-1}v_{\dot{\tilde{A}}} \ . \quad (14)$$

Here it is assumed that the values in P are time-varying. This is to provide an equation for updating these values, later in the proof. By substituting e from (9), one leads to

$$\dot{V} = e^T P((\mathcal{H} \otimes I_n)\dot{\eta} - (\mathcal{H} \otimes I_n)Ax - (\mathcal{H} \otimes I_n)u - (\mathcal{H} \otimes I_n)g - (\mathcal{B} \otimes \dot{z}^0)\mathbf{1}) + \frac{1}{2}e^T\dot{P}e + \tilde{g}^T\Gamma_{1c}^{-1}\dot{\tilde{g}} + v_{\tilde{A}}\Gamma_{2c}^{-1}v_{\dot{\tilde{A}}} \ . \quad (15)$$

By definition, sum of elements in each row of $\mathcal{L}$ is zero (refer to *Definition 2*). Hence, one can say that

$$(\mathcal{L} \otimes \dot{z}^0)\mathbf{1} = \mathbf{0} \ , \quad (16)$$

where, $\mathbf{0} \in \mathbb{R}^{Nn\times 1}$ is a vector with all elements equal to zero. Being noted by this property, one can have

$$(\mathcal{B} \otimes \dot{z}^0)\mathbf{1} = (\mathcal{H} \otimes \dot{z}^0)\mathbf{1} \ . \quad (17)$$

Then, by recalling the mixed-product property of the Kronecker product [6], it leads to

$$(\mathcal{B} \otimes \dot{z}^0)\mathbf{1} = (\mathcal{H} \otimes I_n)(I_N \otimes \dot{z}^0)\mathbf{1} \ , \quad (18)$$

where $I_N \in \mathbb{R}^{N\times N}$ is an identity matrix. Furthermore, one can rewrite the equation in (15) as follows

$$\dot{V} = e^T P((\mathcal{H} \otimes I_n)\dot{\eta} - (\mathcal{H} \otimes I_n)Ax - (\mathcal{H} \otimes I_n)u - (\mathcal{H} \otimes I_n)g - (\mathcal{H} \otimes I_n)(I_N \otimes \dot{z}^0)\mathbf{1}) + \frac{1}{2}e^T\dot{P}e + \tilde{g}^T\Gamma_{1c}^{-1}\dot{\tilde{g}} + v_{\tilde{A}}\Gamma_{2c}^{-1}v_{\dot{\tilde{A}}} \ . \quad (19)$$

Since A and $(\mathcal{H} \otimes I_n)$ are two symmetric matrices, we have

$$(\mathcal{H} \otimes I_n)A = A(\mathcal{H} \otimes I_n) \ . \quad (20)$$

By adding and subtracting $e^T PA(\mathcal{H} \otimes I_n)\eta$, one reaches at

$$\dot{V} = e^T P((\mathcal{H} \otimes I_n)\dot{\eta} + A(\mathcal{H} \otimes I_n)z - A(\mathcal{H} \otimes I_n)\eta - (\mathcal{H} \otimes I_n)u - (\mathcal{H} \otimes I_n)g - (\mathcal{H} \otimes I_n)(I_N \otimes \dot{z}^0)\mathbf{1}) + \frac{1}{2}e^T\dot{P}e + \tilde{g}^T\Gamma_{1c}^{-1}\dot{\tilde{g}} + v_{\tilde{A}}\Gamma_{2c}^{-1}v_{\dot{\tilde{A}}} \ . \quad (21)$$

Moreover, by adding and subtracting $e^T PA(\mathcal{B} \otimes z^0)\mathbf{1}$ and recalling the equation in (9), one leads to

$$\dot{V} = e^T P((\mathcal{H} \otimes I_n)\dot{\eta} + Ae + A(\mathcal{B} \otimes z^0)\mathbf{1} - A(\mathcal{H} \otimes I_n)\eta - (\mathcal{H} \otimes I_n)u - (\mathcal{H} \otimes I_n)g - (\mathcal{H} \otimes I_n)(I_N \otimes \dot{z}^0)\mathbf{1}) + \frac{1}{2}e^T\dot{P}e + \tilde{g}^T\Gamma_{1c}^{-1}\dot{\tilde{g}} + v_{\tilde{A}}\Gamma_{2c}^{-1}v_{\dot{\tilde{A}}} \ . \quad (22)$$

Recalling the symmetric property for A and $(\mathcal{H}\otimes I_n)$ matrices, and also the discussion about having the summation of all elements in each row of $\mathcal{L}$ being equal to zero (refer to (16)), the equation in (22) would be expressed as follows

$$\dot{V} = e^T P(Ae + (\mathcal{H} \otimes I_n)\dot{\eta} + (\mathcal{H} \otimes I_n)A((I_N \otimes z^0)\mathbf{1} - \eta) - (\mathcal{H} \otimes I_n)u - (\mathcal{H} \otimes I_n)g - (\mathcal{H} \otimes I_n)(I_N \otimes \dot{z}^0)\mathbf{1}) + \frac{1}{2}e^T\dot{P}e + \tilde{g}^T\Gamma_{1c}^{-1}\dot{\tilde{g}} + v_{\tilde{A}}\Gamma_{2c}^{-1}v_{\dot{\tilde{A}}} \ . \quad (23)$$

At this point, we try to include the estimated values of $\hat{A}$ and $\hat{g}$ into the design. Thus, by adding and subtracting the terms $e^T P(\mathcal{H} \otimes I_n)\hat{A}((I_N \otimes z^0)\mathbf{1} - \eta)$ as well as $e^T P(\mathcal{H} \otimes I_n)\hat{g}$ in (23), we reach at

$$\dot{V} = e^T P(Ae - (\mathcal{H} \otimes I_n)\dot{c} + (\mathcal{H} \otimes I_n)\hat{A}c - (\mathcal{H} \otimes I_n)u - (\mathcal{H} \otimes I_n)\hat{g}) + \frac{1}{2}e^T\dot{P}e + \tilde{g}^T\Gamma_{1c}^{-1}\dot{\tilde{g}} + v_{\tilde{A}}\Gamma_{2c}^{-1}v_{\dot{\tilde{A}}} - e^T P(\mathcal{H} \otimes I_n)\tilde{g} + e^T P(\mathcal{H} \otimes I_n)\tilde{A}c \ , \quad (24)$$

where $c = [c^1, c^2, ..., c^N]^T = (I_N \otimes z^0)\mathbf{1} - \eta$. Besides, according to *Definition-5* and *Definition-6*, one can say that

$$\tilde{A}c = \mathcal{M}_c v_{\tilde{A}} \ . \quad (25)$$

Considering the online estimation of unknown nonlinear terms, i.e. $\hat{g}$, one may define

$$r_1 = \tilde{g}\Gamma_{1c}^{-1}\dot{\hat{g}} - e^T P(\mathcal{H} \otimes I_n)\tilde{g} \ . \quad (26)$$

Then, by adding and subtracting

$$\frac{1}{4\rho_1}\{\Gamma_{1c}^{-1}\dot{g} + \rho_1 g\}^T\{\Gamma_{1c}^{-1}\dot{g} + \rho_1 g\} + \rho_1\tilde{g}^T\tilde{g} + \rho_1\tilde{g}^T\hat{g} \ , \quad (27)$$

into r_1 in (26), one would have

$$r_1 = \tilde{g}^T\{-\Gamma_{1c}^{-1}\dot{\hat{g}} - \rho_1\hat{g} - (\mathcal{H} \otimes I_n)Pe\} - r_3 + r_4 \ , \quad (28)$$

where

$$\begin{aligned} r_3 = &\frac{1}{4\rho_1}\{\Gamma_{1c}^{-1}\dot{g} + \rho_1 g\}^T\{\Gamma_{1c}^{-1}\dot{g} + \rho_1 g\} + \rho_1\tilde{g}^T\tilde{g} \\ &- 2(\sqrt{\rho_1}\tilde{g}^T)\frac{1}{2\sqrt{\rho_1}}\{\rho_1\tilde{g} + \rho_1\hat{g} + \Gamma_{1c}^{-1}\dot{g}\} \ , \end{aligned} \quad (29)$$

$$r_4 = \frac{1}{4\rho_1}\{\Gamma_{1c}^{-1}\dot{g} + \rho_1 g\}^T\{\Gamma_{1c}^{-1}\dot{g} + \rho_1 g\} > 0 \ . \quad (30)$$

In this regard, it can be shown that

$$\begin{aligned} r_3 = \{&\frac{1}{2\sqrt{\rho_1}}(\Gamma_{1c}^{-1}\dot{g} + \rho_1 g) - \sqrt{\rho_1}\tilde{g}\}^T \\ &\{\frac{1}{2\sqrt{\rho_1}}(\Gamma_{1c}^{-1}\dot{g} + \rho_1 g) - \sqrt{\rho_1}\tilde{g}\} > 0 \ . \end{aligned} \quad (31)$$

Besides, one can show that

$$r_4 \le M_4 = \frac{1}{4\rho_1}\{\Gamma_{1c}^{-1}L_{\dot{g}} + \rho_1 L_g\}^T\{\Gamma_{1c}^{-1}L_{\dot{g}} + \rho_1 L_g\} > 0 \ , \quad (32)$$

where $|\dot{g}| \le L_{\dot{g}}$ and $|g| \le L_g$, for positive upper bounds $L_{\dot{g}}$ and L_g in $\mathbb{R}^{Nn\times 1}$, recalling *Assumption-1*. Hence, by defining

$$\dot{\hat{g}} = \Gamma_{1c}(\mathcal{H} \otimes I_n)Pe - \rho_1\Gamma_{1c}\hat{g} \ , \quad (33)$$

as the lumped adaptive laws for estimating unknown nonlinear terms; it infers that

$$r_1 \le -r_3 + M_4 \ . \quad (34)$$

Similarly for unknown linear terms, i.e $\hat{A}$, one can define

$$r_2 = v_{\tilde{A}}^T\Gamma_{2c}^{-1}v_{\dot{\hat{A}}} + e^T P(\mathcal{H} \otimes I_n)\mathcal{M}_c v_{\tilde{A}} \ . \quad (35)$$

Then, by adding and subtracting

$$\frac{1}{4\rho_2}\{\Gamma_{2c}^{-1}v_{\dot{A}}+\rho_2 v_A\}^T\{\Gamma_{2c}^{-1}v_{\dot{A}}+\rho_2 v_A\}+\rho_2 v_{\tilde{A}}^T v_{\tilde{A}}+\rho_2 v_{\tilde{A}}^T v_{\hat{A}} \quad (36)$$

into r_2, one leads to

$$r_2 = v_{\tilde{A}}^T\{-\Gamma_{2c}^{-1}v_{\dot{\hat{A}}} - \rho_2 v_{\hat{A}} + \mathcal{M}_c(\mathcal{H} \otimes I_n)Pe\} - r_5 + r_6 \ , \quad (37)$$

where

$$\begin{aligned} r_5 = &\frac{1}{4\rho_2}\{\Gamma_{2c}^{-1}v_{\dot{A}} + \rho_2 v_A\}^T\{\Gamma_{2c}^{-1}v_{\dot{A}} + \rho_2 v_A\} + \rho_2 v_{\tilde{A}}^T v_{\tilde{A}} - \\ &2(\sqrt{\rho_2}v_{\tilde{A}}^T)\frac{1}{2\sqrt{\rho_2}}\{\rho_2 v_{\tilde{A}} + \rho_2 v_{\hat{A}} + \Gamma_{2c}^{-1}v_{\dot{A}}\} \ , \end{aligned} \quad (38)$$

$$r_6 = \frac{1}{4\rho_2}\{\Gamma_{2c}^{-1}v_{\dot{A}} + \rho_2 v_A\}^T\{\Gamma_{2c}^{-1}v_{\dot{A}} + \rho_2 v_A\} > 0 \ . \quad (39)$$

Simply, it can be shown that

$$\begin{aligned} r_5 = \{&\frac{1}{2\sqrt{\rho_2}}(\Gamma_{2c}^{-1}v_{\dot{A}} + \rho_2 v_A) - \sqrt{\rho_2}v_{\tilde{A}}\}^T \\ &\{\frac{1}{2\sqrt{\rho_2}}(\Gamma_{2c}^{-1}v_{\dot{A}} + \rho_2 v_A) - \sqrt{\rho_2}v_{\tilde{A}}\} > 0 \ . \end{aligned} \quad (40)$$

In addition, recalling the boundedness of dynamics of agents as it is assumed in *Assumption-1*, one can achieve

$$r_6 \le M_6 = \frac{1}{4\rho_2}\{\Gamma_{2c}^{-1}L_{\dot{A}} + \rho_2 L_A\}^T\{\Gamma_{2c}^{-1}L_{\dot{A}} + \rho_2 L_A\} > 0 \ , \quad (41)$$

where $|v_{\dot{A}}| \le L_{\dot{A}}$ and $|v_A| \le L_A$ in $\mathbb{R}^{Nn\times 1}$. Thus by defining

$$v_{\dot{\hat{A}}} = \Gamma_{2c}\mathcal{M}_c(\mathcal{H} \otimes I_n)Pe - \rho_2\Gamma_{2c}v_{\hat{A}} \ , \quad (42)$$

as the lumped adaptive laws for estimating unknown linear terms; one would have

$$r_2 \le -r_5 + M_6 \ . \quad (43)$$

Integrating results from (34) and (43) into (24), one leads to

$$\dot{V} \le e^T P\{Ae+(\mathcal{H}\otimes I_n)(\hat{A}c-\dot{c}-u-\hat{g})\}+\frac{1}{2}e^T\dot{P}e-r_7+M_7 \ , \quad (44)$$

where $r_7 = r_3 + r_5$ and $M_7 = \min\{M_4, M_6\}$. If one defines

$$u = \frac{1}{2}Pe + \hat{A}c - \hat{g} - \dot{c} \ , \quad (45)$$

as the lumped control signals in the network; we reach to

$$\dot{V} \le \frac{1}{2}e^T(A^T P + PA - P(\mathcal{H} \otimes I_n)P + \dot{P})e - r_7 + M_7 \ . \quad (46)$$

Then, by using the following equation

$$\dot{P} + A^T P + PA - P(\mathcal{H} \otimes I_n)P = -Q_c \ , \quad (47)$$

where $Q_c = (I_N \otimes Q) \in \mathbb{R}^{Nn\times Nn}$; we have

$$\dot{V} \le -(\frac{1}{2}e^T Qe + r_7) + M_7 \ . \quad (48)$$

According to (13) and (48) and based on LaSalle-Yoshizawa theorem [54], the cost function V is stable and the errors e, $\tilde{g}$ and $\tilde{A}$ would be uniformly ultimately bounded (UUB) in small sets around origin. Thus, one can say that the estimated values of $\hat{g}$ and $\hat{A}$ converge to actual values with UUB property. Consequently, one can rewrite the equation in (49) as follows

$$\dot{P} + \hat{A}^T P + P\hat{A} - P(\mathcal{H} \otimes I_n)P = -Q_c \ , \quad (49)$$

The stability analysis of CAMFC is now completed. But, the definition of control and adaptive laws should be extracted for each agent. In this regard, control law at ith agent (as presented in (10)) would be derived from the equation corresponding to ith row of lumped representation in (45). Similarly, the update equation P^i matrix (as suggested in (12)) would be generated by considering the equations on ith diagonal element of lumped representation in (49) (noting that $(\hat{A}^i)^T = \hat{A}^i)$. Moreover, by referring to (33) and (42), the adaptive laws for nonlinear and linear terms at ith agent, would be defined as proposed in (11a) and (11b). This completes the proof. □

Remark-2. As mentioned in the final section of the above proof, CAMFC algorithm provides an UUB convergence property to the system. It means the consensus errors and consequently the control inputs of the multi-agent system controlled by CAMFC algorithm would experience small but bounded fluctuations during the steady-state phase of response. This is a compromise to control the system with minimal information on the dynamics system at each agent.

Remark-3. No local measurements of the states at agents are incorporated neither in the cooperative control law in (10) nor in the adaptive laws in (11a) and (11b). Only the consensus errors defined in *Proposition-1* are required, which are available by measurements on relative states of neighboring agents in the network, i.e. $x^i - x^j$ (Fig. 1).

Remark-4. According to adaptive laws in (11a) and (11b), the estimators include no regressor variables. Specifically, the first terms on the right-hand side of the mentioned equations rely only on the elements of a positive definite gain matrix (refer to *Lemma-1* and *Remark-7*) and also the consensus errors in the network (refer to Fig. 1). Hence, the adaptive laws would be functional as long as the consensus errors are not converged to zero. Therefore, no PE condition is required on any input signals at the agents.

Remark-5. Time-derivative of parameter c^i required in (10) can be computed by using a sliding-mode differentiator [50].

Proposition-2. Recalling (32) and (41), we have

$$\begin{aligned} M_4 &= (\frac{1}{4\rho_1} L_{\dot{g}}^T \Gamma_{1c}^{-2} L_{\dot{g}}) + (\frac{1}{4}\rho_1 L_g^T L_g) + (\frac{1}{2} L_{\dot{g}}^T \Gamma_{1c}^{-1} L_g) \; , \\ M_6 &= (\frac{1}{4\rho_2} L_{\dot{A}}^T \Gamma_{2c}^{-2} L_{\dot{A}}) + (\frac{1}{4}\rho_1 L_A^T L_A) + (\frac{1}{2} L_{\dot{A}}^T \Gamma_{2c}^{-1} L_A) \; . \end{aligned} \tag{50}$$

In this regard, the values for adaptive gains Γ_1 and Γ_2 are chosen large enough, while the leakage gains ρ_1 and ρ_2 are tuned at enough small values, so as to have values of M_4 and M_6 and consequently M_7 to be small. According to (48), having M_7 to be small would help to observe lower amplitudes of the fluctuations on system outputs and formation-tracking errors at steady-state portion of the system responses. Recalling [13], the last terms on right-hand sides of (11a) and (11b), i.e. $\rho_1\Gamma_1\hat{g}^i$ and $\rho_2\Gamma_2 v_{\hat{A}^i}$, are the leakage terms that provide robustness for the proposed adaptive laws. Moreover, the leakage gains ρ_1 and ρ_2 are positive tuning parameters that regulate the effect of leakage terms into the corresponding adaptive laws. Tuning of these parameters would help to guarantee the robustness of the control law in CAMFC algorithm (i.e. (10)) against the unknown boundaries of the unknown internal dynamics (either linear or nonlinear) as well as the external disturbances. Note that, for whatever values of L_g, $L_{\dot{g}}$, L_A and $L_{\dot{A}}$, values of M_4 and M_6 in (50) can be adjusted by tuning Γ_1, Γ_2, ρ_1 and ρ_2.

Proposition-3. Values for states of the virtual leader (i.e. x^0) and the formation variables (i.e. η^0 and η^i for $i \in [1, N]$) are required in control law at agent i, as presented in (10). These values can be estimated using the cooperative observers suggested in [51], as follows

$$\dot{\hat{x}}_i^0 = -\lambda\tilde{x}_i^0 - \{\mathcal{M}(sgn(\sum_{j=1}^{N}\mathcal{H}(i,j)\tilde{x}_j^0))\chi^M\} \; , \tag{51}$$

$$\dot{\hat{\Omega}}^i = -\mu\epsilon^i - \{sgn(\sum_{j=1}^{N}\mathcal{H}(i,j)\epsilon^j)\mathcal{M}(\Upsilon^M)\} \; . \tag{52}$$

Here, λ and μ are two positive scalar tuning gains and

$$\tilde{x}_i^0 = \sum_{j=1}^{N} a_{ij}(\hat{x}_i^0 - \hat{x}_j^0) + b_i(\hat{x}_i^0 - x^0) \; , \tag{53}$$

$$\epsilon^i = \sum_{j=1}^{N} a_{ij}(\hat{\Omega}^i - \hat{\Omega}^j) + b_i(\hat{\Omega}^i - \Omega) \; , \tag{54}$$

where parameter Ω is defined as in *Definition-4*. Moreover, we defined $|\dot{x}^0| \le \chi^M$ and $|\dot{\eta}^0| \le \Upsilon^M$. Parameters χ^M and Υ^M (both defined in $\mathbb{R}^{n\times1}$) include upper bounds for time-derivative of the desired states at virtual leader (i.e. $\dot{x}^0$), and time derivative of the desired formation parameters at virtual leader (i.e. $\dot{\eta}^0$), respectively. The values of these parameters are defined based on the desired trajectory of entire network and the desired formation topology among agents. More information is provided in [51]. The values of $\hat{x}_i^0$, $\hat{\eta}_i^0$ and $\hat{\eta}_i^j$ (for $j \in [1, N]$) computed by the observers in (51) and (52) are used for determining the control inputs (i.e. u^i) in (10). According to *Separation principle* [55], incorporation of stable observers presented in (51) and (52) into the stable controller presented in (10), leads to a stable system (Fig. 1).

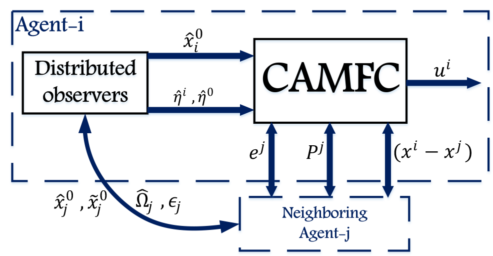

Fig. 1: Schematic for CAMFC at agent i; the data flow within the agent and between the neighboring agents are shown.

Remark-6. One concern on updating the values of P^i using differential Riccati equation in (12), is that these values should be positive during the controller operation, as it is one of the requirement expressed in *Theorem-1*. In this regard, the following lemma is presented to show that by some special considerations, there is always a positive solution for each diagonal elements of P^i in (12).

Lemma-1. A solution for rth diagonal equation in (12) is

$$P^i(r,r) = P_0^r + \frac{1}{w^r} \quad , \quad r \in [1, n] \; , \tag{55}$$

where

$$P_0^r = \frac{\hat{A}^i(r,r) - \sqrt{\Delta}}{\mathcal{H}(i,i)} \; , \tag{56}$$

$$w^r = \frac{\mathcal{H}(i,i)}{2\sqrt{\Delta}} + (w_0^r - \frac{\mathcal{H}(i,i)}{2\sqrt{\Delta}})\exp(-\int_0^t 2\sqrt{\Delta}ds) \; , \; t > 0 \; . \tag{57}$$

Here, we have

$$\Delta = (\hat{A}^i(r,r))^2 + Q(r,r)\mathcal{H}(i,i) \quad , \quad w_0^r = \frac{1}{1 - P_0^r} \; . \tag{58}$$

It can be shown that by choosing $Q(r,r)$ large enough, the value of $P^i(r,r)$ determined from (55) would be positive.

Proof. The rth diagonal equation in (12) for the ith agent in the network, can be presented as follows

$$\dot{P}^i(r,r) = -2\hat{A}^i(r,r)P^i(r,r) + \mathcal{H}(i,i)(P^i(r,r))^2 - Q(r,r) \; . \tag{59}$$

The off-diagonal equations in (12) are relaxed by choosing zero off-diagonal values in Q matrix. For simplicity, we use $p = P^i(r,r)$, $a = \hat{A}^i(r,r)$, $h = \mathcal{H}(i,i)$ and $q = Q(r,r)$ in the following. Hence, rth differential Riccati equation at the ith agent is represented as follows

$$\dot{p} = hp^2 - 2ap - q \; . \tag{60}$$

As we know, the general solution for this equation is

$$p = p_0 + \frac{1}{w} \; , \tag{61}$$

where we use $p_0 = P_0^r$ and $w = w^r$ for brevity. By definition, p_0 is a special solution as $\dot{p}_0 = 0$. Thus, we have

$$hp_0^2 - 2ap_0 - q = 0 \; . \tag{62}$$

It is evident that the value of p_0 defined by (56) and (58) is a solution of the equation in (62). Then, by replacing p from (61) into (60), we reach to

$$-\frac{\dot{w}}{w^2} = -2ap_0 - 2a\frac{1}{w} + hp_0^2 + h\frac{1}{w^2} + 2hp_0\frac{1}{w} - q \; . \tag{63}$$

Then, by recalling (62) we lead to

$$\dot{w} = 2bw - h \; , \tag{64}$$

where $b = b(t) = a(t) - hp_0 = a - hp_0 = \sqrt{\Delta}$. By using the integrator factor $\beta = \exp(-\int_0^t 2\sqrt{\Delta}ds)$ for the differential equation in (64), we reach to the solution proposed in (57). Here, we use $w_0 = w_0^r$, for brevity. At this point, we should show that the value of p defined as follows, is always positive;

$$p = \frac{a - \sqrt{\Delta}}{h} + \frac{1}{\alpha + \sigma} \; , \tag{65}$$

where

$$\alpha = \frac{h}{2\sqrt{\Delta}}(1-\beta) \quad , \quad \sigma = \frac{\beta}{1 - \frac{a-\sqrt{\Delta}}{h}} \; . \tag{66}$$

By recalling *Definition-2*, the diagonal elements of $\mathcal{H}$ are positive. Thus, $h > 0$ and then by noting $\sqrt{\Delta} > 0$, we reach at $0 < \beta < 1$ and consequently $\alpha > 0$. In addition, by some simple operations on (65) and (66), we lead to

$$p = \frac{h^2 - (a - \sqrt{\Delta})p_1}{h[h(\alpha+\beta) - \alpha(a - \sqrt{\Delta})]} \; , \tag{67}$$

in which

$$p_1 = [(1 - \alpha - \beta)h + \alpha(a - \sqrt{\Delta})] \; . \tag{68}$$

By recalling the definition of Δ as in (58), we know that

$$\sqrt{\Delta} = \sqrt{a^2 + qh} > a \; , \tag{69}$$

and hence we have $[-(a - \sqrt{\Delta})] > 0$. Consequently, we conclude that the denominator of the right-hand side of equation in (67) is always positive. Also on the numerator, the first term is surely positive. In addition,

$$p_1 = h[1 - \frac{h}{2\sqrt{\Delta}} + \frac{\beta h}{2\sqrt{\Delta}} - \beta] + \frac{h}{2\sqrt{\Delta}}(1-\beta)(a - \sqrt{\Delta}) \; . \tag{70}$$

Then by rearranging the terms, we reach at

$$p_1 = p_2[(\sqrt{\Delta} - \beta\sqrt{\Delta}) - (h - \beta h) + (a - \beta a)] \; , \tag{71}$$

where $p_2 = \frac{h}{2\sqrt{\Delta}} > 0$. Furthermore,

$$p_1 = p_2(1-\beta)p_3 \; , \tag{72}$$

where $p_3 = \sqrt{\Delta} + a - h$. Recalling definition of Δ in (58), p_3 would be positive by choosing q large enough so that $\sqrt{\Delta} > (a - h)$. Hence, $p > 0$. This completes the proof. $\square$

Remark-7. As declared in *Assumption-2*, the communication graph in the network should have at-least one spanning-tree. By satisfying this assumption, there would be at-least one non-zero $\mathcal{H}(i,j)$ at agent i for the neighboring agents $j \in [1, N]$. Noted by this point and also recalling (11a) and (11b), it is confirmed that the governing terms in the adaptive laws for estimating the unknown terms in CAMFC would be zero if and only if the consensus errors in the network converge to zero. Moreover, by having a spanning-tree, all $\mathcal{H}(i,i)$ for $i \in [1, N]$ would be non-zero and consequently P_0^r (as in (56)) would be a definite value. This leads to have definite controller gains in CAMFC at each agent in the network.

III. Comparative Analysis

In this section, two case studies are presented to analyze the performance of CAMFC algorithm in comparison with two state-of-the-art solutions, both in consensus and formation-tracking problems. Here, two cost functions are used for quantitative comparison. The first cost function S_1, is sum of consensus errors at all agents throughout the simulation time,

$$S_1 = \int_0^{t_f} [\sum_{i=1}^{N} \{(e^i)^T e^i\}] dt \; . \tag{73}$$

The second cost function, S_2, is sum of the consensus errors and the control efforts, i.e.

$$S_2 = \int_0^{t_f} [\sum_{i=1}^{N} \{(e^i)^T e^i + (u^i)^T u^i\}] dt \; . \tag{74}$$

Here, t_f is the final time of the simulation. In the following, the tuning parameters of the proposed distributed observers in *Proposition-3* are set at $\mu = \lambda = 10$ and $\chi^0 = \upsilon^0 = I_2$.

A. Case-1: Consensus problem

Here, the benchmark solution is a distributed model-free control algorithm (named as DMFC-1) implemented on a network of five agents with dynamics of reverse pendulum at agents [7]. DMFC-1 is formulated as [7]

$$\begin{aligned} u_i &= cr_i + \frac{\lambda_c}{d_i + \beta_i} e_i^2 - \hat{W}_i^T \phi_i \; , \\ \dot{\hat{W}}_i &= -F_c \phi_i r_i p_i (d_i + \beta_i) - k_c F_c \hat{W}_i \; , \end{aligned} \tag{75}$$

where c, λ_c, k_c and F_c (all in $\mathbb{R}^+$) are tuning parameters, $p_i > 0$ is the controller gain which is adjusted offline [7]; and

$$\begin{aligned} e_i^1 &= \sum_{j=1}^{N} a_{ij}(x_j^1 - x_i^1) + \beta_i(x_0^1 - x_i^1) \; , \\ e_i^2 &= \sum_{j=1}^{N} a_{ij}(x_j^2 - x_i^2) + \beta_i(x_0^2 - x_i^2) \quad , \quad r_i = e_i^2 + \lambda_c e_i^1 \; . \end{aligned} \tag{76}$$

Here, an ANN with N_{nr} neurons is used for online parameter estimation. $\phi_i \in \mathbb{R}^{N_{nr} \times 1}$ is vector of regressor parameters at agent i (defined based on system states) and $\hat{W}_i \in \mathbb{R}^{N_{nr} \times 1}$ is vector of weights for neurons. Dynamics for reverse pendulum system at each agent is considered as follows [7]

$$\dot{x}_1 = x_2 \ \ , \ \ \dot{x}_2 = \frac{1}{J_p}(u - b_p x_2 - m_p l_p g_e \sin x_1 + D_p), \tag{77}$$

in which, $J_p = 1\text{kg.m/}s^2$, $m_p = 0.1\text{kg}$, $l_p = 0.1\text{m}$ and $b_p = 0.01\text{kg.m/s}$ are moment of inertia, mass, length of the pendulum and damping constant, respectively. $D_p = \sin 2t$ is external disturbance and $g_e = 9.81\text{m/s}^2$ is the earth's gravity. Here, term $g^i(2)$ is responsible for unknown nonlinearities and external disturbances at ith agent (corresponding to $m_p l_p g_e \sin x_1 + D_p$). Moreover, the communication graph of network is defined as

$$\mathcal{A} = \begin{bmatrix} 0&1&0&0&1\\ 1&0&1&1&0\\ 0&1&0&1&1\\ 0&1&1&0&1\\ 1&0&1&1&0 \end{bmatrix} \ \ , \ \ \mathcal{B} = diag([0,0,1,0,0]) \ . \tag{78}$$

Only the third agent is pinned to leader. The desired consensus value for x_1 at all agents in network is $x_1^d = 3\sin 0.5t$. The agents start simulation at different initial states and the objective is to reach a consensus at x_1^d among them. The values for tuning parameters of CAMFC and DMFC-1 are presented in Table I. Performance of the algorithms are compared based on the values of S_1 and S_2. The tuning parameters for algorithms are chosen in the way to achieve equal values of S_1. Then, we can compare the algorithms based on values of S_2. CAMFC needs less control effort (lower value of S_2) to achieve same level of consensus errors in network. The simulation results for this case study are presented in Figs. 2 to 6. By utilizing DMFC-1, more fluctuations are observed in consensus errors. This comes from fluctuations in DMFC-1 control signals (Fig. 3). Recalling the UUB convergence of CAMFC (*Remark-2*), consensus errors converge within a small set around the origin. Thus, while small fluctuations are observed on system response, the stability is achieved with no information on boundaries of internal dynamics and external disturbances at agents. According to Figs. 5 and 6, all estimated parameters are bounded, while any changes in these values update the CAMFC control signal. Moreover, values for $\hat{A}^i(1,1)$ and $\hat{g}^i(1)$ are zero throughout the simulation, without any change. Because dynamic system in (77) is a double-integrator system, which has one real control variable and two system states. An idea originated from back-stepping method is used to design a virtual control variable corresponding to first state [50]. The value for $P^i(1,1)$ do not change through simulation (Fig. 4) and it is determined based on values of $\mathcal{H}(i,i)$ and Q. But the values for $A^i(2,2)$ and $g^i(2)$ are not zero and correspond to $\frac{b_p}{J_p}$ and $\frac{m_p l_p g_e}{J_p}\sin x_1 + \frac{1}{J_p}D_p$, respectively. Hence, the values for $P^i(2,2)$ change throughout the simulation based on the changes in $\hat{A}^i(2,2)$. According to Table I, values for $\Gamma_1(1,1)$ and $\Gamma_2(1,1)$ are chosen small enough to avoid any instability caused by incorporation of the virtual control signal.

B. Case-2: Formation-tracking problem

Here, the distributed model-free control algorithm in [22] (named as DMFC-2) is implemented in a network of six agents

TABLE I: Case-1: Specifications of the DMFC-1 and CAMFC algorithms implemented at agent i.

Parameter	DMFC-1	CAMFC
No. adaptive laws	4	4
Tuning variables	$F_c = 1.63$ $c = 2.2$ $k_c = 1 \times I_4$ $\lambda_c = 10$, $k_n = 1$	$\Gamma_1 = 10^3\times$ diag$([10^{-6}, 1])$ $\Gamma_2 = 1\times$ diag$([10^{-6}, 1])$ $\rho_1 = \rho_2 = 1$ $Q = 100 \times I_2$
S_1	6.628	6.621
S_2	1994	811

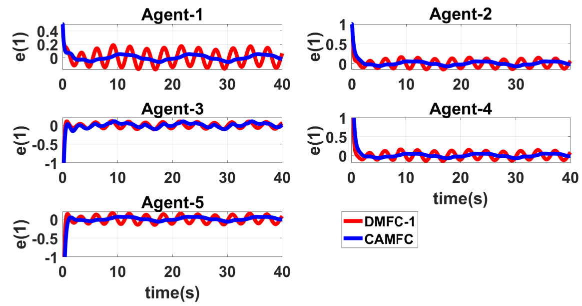

Fig. 2: Case-1: Consensus errors; values of $e^i(1)$s for CAMFC and DMFC-1 are compared.

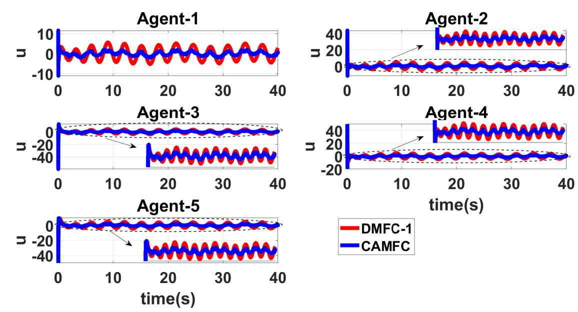

Fig. 3: Case-1: Control variables; values of u^is for CAMFC and DMFC-1 at all agents are compared.

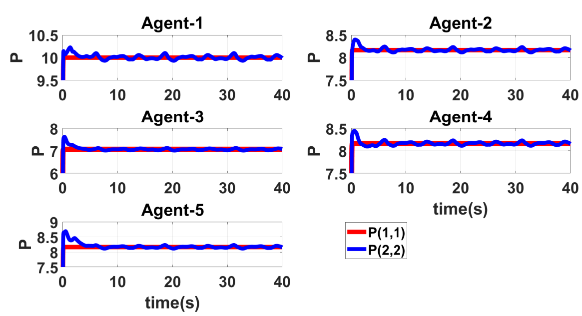

Fig. 4: Case-1: Controller gains; diagonal elements of P^is in CAMFC are presented.

with dynamics as below [22]

$$\begin{aligned} \dot{x}_1^i &= x_2^i \sin(c_1^i x_1^i) + u_1^i + D_1^i \\ \dot{x}_2^i &= x_1^i \cos(c_2^i (x_2^i)^2) + u_2^i + D_2^i \ . \end{aligned} \tag{79}$$

Here, we have $[c_1^1; c_2^1] = [0.5; 0.4]$, $[c_1^2; c_2^2] = [-0.5; 0.4]$, $[c_1^3; c_2^3] = [6; -5]$, $[c_1^4; c_2^4] = [-10; 12]$, $[c_1^5; c_2^5] = [10; 12]$ and $[c_1^6; c_2^6] = [0.01; 10]$. In addition, external disturbances

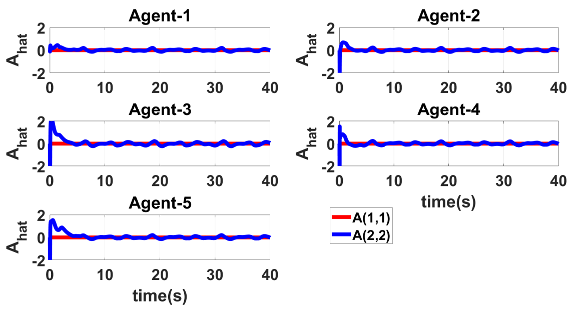

Fig. 5: Case-1: Linear terms; diagonal elements of $\hat{A}^i$s in CAMFC are presented.

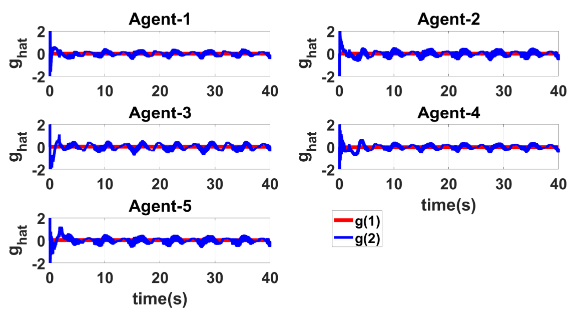

Fig. 6: Case-1: Nonlinear terms; values of $\hat{g}^i$s in CAMFC are presented.

are set as $[D_1^1; D_2^1] = [\sin(-t)\exp(-2t); \cos(t^2)]$, $[D_1^2; D_2^2] = [\exp(-3t); \sin(t^2)\cos(t)]$, $[D_1^3; D_2^3] = [\cos(t^2); \exp(-t)]$, $[D_1^4; D_2^4] = [-\sin(t^2); \cos(t)\exp(-3t)]$, $[D_1^5; D_2^5] = [\sin(t^2); \cos(t)\exp(-5t)]$ and $[D_1^6; D_2^6] = [\exp(-4t); \sin(t)\cos(t^2)]$. DMFC-2 is defined as [22]

$$u^i = -k^c e^i - (\hat{W}^i)^T \phi(x^i) - \frac{(\hat{B}^i)^2}{||e^i||\hat{B}^i + \exp(-t)} \tag{80}$$
$$\dot{\hat{W}}^i = \Gamma^c \phi(x^i) e^i \quad , \quad \dot{\hat{B}}^i = \theta^c ||e^i|| \, ,$$

where

$$e^i = \sum_{j=1}^{N_c} [a_{ij}(x^i - x^j)] \, , \tag{81}$$

for $x^i = [x_1^i; x_2^i]$. In this algorithm, $\Gamma^c \in \mathbb{R}^{N_c \times N_c}$ is a positive-definite adaptation gain matrix, $\theta^c > 0$ is the learning rate, $k^c > 0$ is the controller gain and N_c is the number of neurons chosen for the ANN. Here, the communication graph among agents is considered as follows [22]

$$\mathcal{A} = \begin{bmatrix} 0 & 0.2 & 0 & 0 & 0 & 0.4 \\ 0.2 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.3 & 0 & 0.5 \\ 0.4 & 0 & 0 & 0 & 0.5 & 0 \end{bmatrix} \, , \; \mathcal{B} = diag([1, 0, 0, 0, 0, 0]) \, . \tag{82}$$

The first agent is connected to the leader. Desired final state of the leader is $x_0^d = [0; 0]$. Desired formation variables for agents are $\eta^1 = [\frac{\sqrt{3}}{2}; \frac{1}{2}]$, $\eta^2 = [0; 1]$, $\eta^3 = [\frac{-\sqrt{3}}{2}; \frac{1}{2}]$, $\eta^4 = [\frac{-\sqrt{3}}{2}; \frac{-1}{2}]$, $\eta^5 = [0; -1]$ and $\eta^6 = [\frac{\sqrt{3}}{2}; \frac{-1}{2}]$. Agents start the simulation from different initial states as follows $x^1(0) = [6; 2]$, $x^2(0) = [3; 3\sqrt{3}]$, $x^3(0) = [-3; 3\sqrt{3}]$, $x^4(0) = [-6; -2]$, $x^5(0) = [-3; -3\sqrt{3}]$ and $x^6(0) = [3; -3\sqrt{3}]$. The values of S_1, S_2 and tuning parameters of CAMFC and DMFC-2 are presented in Table II. The tuning parameters are chosen so as to achieve similar values of S_1. Lower value of S_2 is achieved by CAMFC, meaning that lower control effort is required for reaching same consensus error at agents. There are ten adaptive laws used to implement DMFC-2 algorithm, while CAMFC needs four adaptive laws. The simulation results for this case are depicted in Figs. 7 to 11. Same rate of convergence is achieved using CAMFC and DMFC-2 algorithms. But, the results from DMFC-2 are more smooth, which is brought by high frequency changes in the control signals (Fig. 8). This leads to higher value of S_2 for DMFC-2. Instead, controller inputs in CAMFC are more smooth, agreeing lower control efforts. Recalling Figs. 9 to 11, controller gains and estimated parameters are bounded and converged to final values in 7 seconds.

TABLE II: Case-2: Specifications of the DMFC-2 and CAMFC algorithms implemented at agent i.

Parameter	DMFC-2	CAMFC
No. adaptive laws	10	4
Tuning variables	$\Gamma^c = 0.1 \times I_{10}$ $\theta_c = 10$ $k^c = 21.8 \times I_4$ $\hat{\beta}(0) = 5$, $k_n = 1$	$\Gamma_1 = 10^3 \times I_2$ $\Gamma_2 = 1 \times I_2$ $\rho_1 = \rho_2 = 1$ $Q = 10^3 \times I_2$
S_1	1.652	1.659
S_2	4358	2734

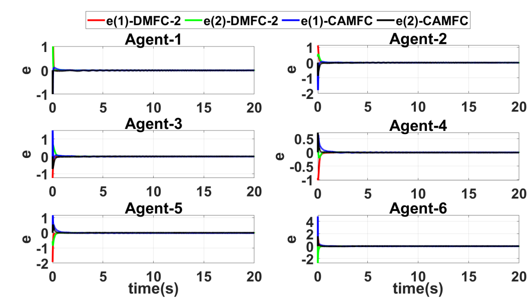

Fig. 7: Case-2: Consensus errors; values of $e^i(1)$s for CAMFC and DMFC-2 are compared.

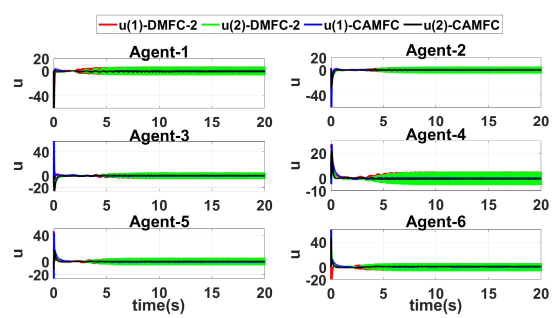

Fig. 8: Case-2: Control variables; values of u^is for CAMFC and DMFC-2 at all agents are compared.

IV. AUTONOMOUS FORMATION-TRACKING FLIGHT IN A TEAM OF QUADROTORS

In this section, CAMFC is implemented as distributed position controller in a swarm of four quadrotors to maintain

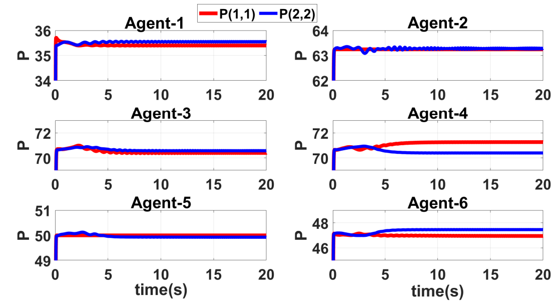

Fig. 9: Case-2: Controller gains; diagonal elements of P^is in CAMFC are presented.

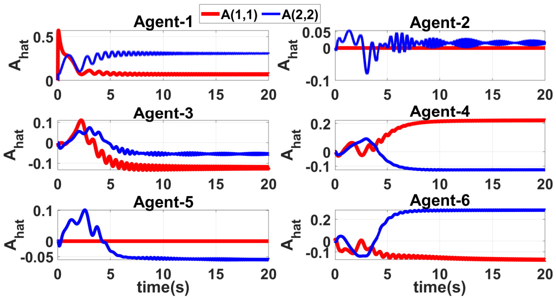

Fig. 10: Case-2: Linear terms; diagonal elements of $\hat{A}^i$s in CAMFC are presented.

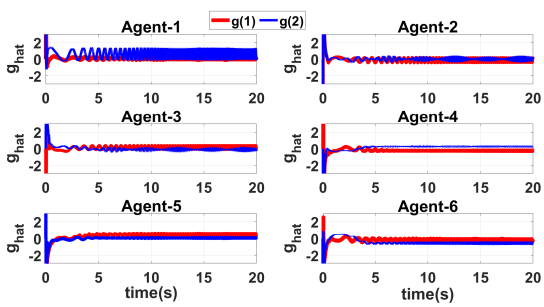

Fig. 11: Case-2: Nonlinear terms; estimated values of $\hat{g}^i$s in CAMFC are presented.

a formation-tracking objective. CAMFC is responsible for generating local velocity set-points based on desired trajectory and formation topology of the swarm. The velocity and attitude controllers at agents are provided by a recently-developed adaptive model-free control (AMFC) algorithm [50]. The dynamics of each quadrotor is considered as follows

$$\begin{aligned} &\dot{p}_q = v_q \quad , \quad \dot{\Phi}_q = R_{qt}^{-1}\omega_q \, , \\ &\dot{v}_q = \frac{1}{m_q}(R_q F_q - k_d v_q - m_q F_g + f_q) \, , \\ &\dot{\omega}_q = J_q^{-1}(\tau_q - k_a \omega_q - (\omega_q \times J_q \omega_q) + t_q) \, . \end{aligned} \tag{83}$$

Here, p_q and v_q $\mathbb{R}^{3\times 1}$ are position and velocity vectors of the quadrotor, $\Phi_q = [\phi;\theta;\psi]$ are Euler angles, $\omega_q \in \mathbb{R}^{3\times 1}$ is angular velocity vector, $F_q = [0;0;T_q]$ is thrust vector in quadrotor's body frame, $\tau_q \in \mathbb{R}^{3\times 1}$ is the vector of produced torques at quadrotor and $F_g = [0;0;-g_e]$. Moreover, m_q and $J_q \in \mathbb{R}^{3\times 3}$ are mass and inertia matrix of each quadrotor, while k_d and k_a are scalars gains for resisting forces and torques. In addition, $f_q \in \mathbb{R}^{3\times 1}$ and $t_q \in \mathbb{R}^{3\times 1}$ are external disturbances and R_q and R_{qt} in $\mathbb{R}^{3\times 3}$ are rotation matrices [50]. Here, the parameters are set as $m_q = 2$kg, $J_q = 0.001 \times diag([1;1;2])$kg.m^2, $k_d = k_a = 0.1$, $f_q = [0;0;0.1\sin(t)]$ and $t_q = 0.1\sin(t) \times [1;1;1]$. In the control scheme, distributed control inputs at CAMFC (u^i in (10)) are considered as velocity set-points, leading to local tracking error $e_v = u^i - v_q$ at each quadrotor, The required measurements for implementation of CAMFC algorithm are $3D$ relative position measurements between neighboring drones. This set of measurements can be provided by the adaptive localization algorithm proposed in [56]. Here, the communication graph is

$$\mathcal{A} = \begin{bmatrix} 0&1&0&0 \\ 1&0&1&0 \\ 0&1&0&1 \\ 0&0&1&0 \end{bmatrix} \quad , \quad \mathcal{B} = diag([1,0,0,0]) \, . \tag{84}$$

Furthermore, AMFC-1 operates on e_v and defines the control inputs $u_v \in \mathbb{R}^{3\times 1}$. These values are utilized to define the required thrust force and the set-points for Euler angles as follows [57] (assuming $\psi^d = 0$)

$$\begin{aligned} &T_q = \sqrt{u_v^T u_v} \, , \\ &\phi^d = \arcsin \frac{u_v(2)}{T_q} \quad , \quad \theta^d = \arctan \frac{u_v(1)}{u_v(3)} \, . \end{aligned} \tag{85}$$

Finally, by defining the orientation tracking error $e_\Phi = \Phi_q^d - \Phi_q$, with $\Phi^d = [\phi^d;\theta^d;\psi^d]$, AMFC-2 unit defines the required torques τ_q. Reference trajectory for virtual leader of the swarm (i.e x^0) is defined as follows (t is time in seconds)

$$\left\{ \begin{array}{lcc} x^0 = [0;0;0] & , & 0 < t < 10 \ , \ 300 < t < 360 \\ x^0 = [0;0;z_d] & , & 10 < t < 60 \ , \ 240 < t < 300 \\ x^0 = [x_d;0;z_d] & , & 60 < t < 120 \\ x^0 = [x_d;y_d;z_d] & , & 120 < t < 180 \\ x^0 = [0;y_d;z_d] & , & 180 < t < 240 \end{array} \right\} , \tag{86}$$

where $x_d = y_d = z_d = 5$m. In addition, desired formation topology in the swarm is defined as (for $i \in [1,4]$)

$$\left\{ \begin{array}{lcc} \eta^i = [0;0;0] & , & 0 < t < 10 \ , \ 300 < t < 360 \\ \eta^i = [0;0;\eta_z(i)] & , & 10 < t < 60 \ , \ 240 < t < 300 \\ \eta^i = [\eta_x(i);0;\eta_z(i)] & , & 60 < t < 120 \\ \eta^i = [\eta_x(i);\eta_y(i);\eta_z(i)] & , & 120 < t < 180 \\ \eta^i = [0;\eta_y(i);\eta_z(i)] & , & 180 < t < 240 \end{array} \right\} , \tag{87}$$

where

$$\begin{aligned} &\eta_x = [+r_x;-r_x;-r_x;+r_x] \ , \ \eta_y = [+r_y;+r_y;-r_y;-r_y] \ , \\ &\eta_z = [r_z;2r_z;3r_z;4r_z] \ , \end{aligned} \tag{88}$$

and $r_x = r_y = 1$m and $r_z = 0.5$m. Moreover, the values for tuning gains in CAMFC, AMFC-1 and AMFC-2 at each agent are $\Gamma_1 = 100 \times I_n$, $\Gamma_2 = I_n$, $\rho_1 = \rho_2 = 1$ and $Q = I_n$, with $n = 3$ for CAMFC and AMFC-1, and $n = 6$ for AMFC-2 . The tuning knobs are set trivially, since controller gains P^is are updated online. The corresponding simulation results are presented in Figs. 12 to 14. Consensus errors are bounded around zero. The desired trajectory and formation topology are followed by all agents, while the control variables are bounded.

V. Conclusion

In this paper, a distributed cooperative adaptive model-free control algorithm is proposed for consensus and formation-tracking problems in network of agents with completely unknown nonlinear dynamic systems and external disturbances.

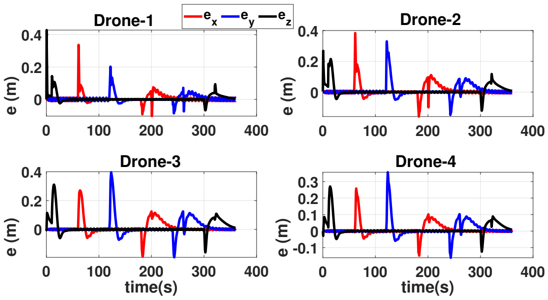

Fig. 12: Quadrotors application: Consensus errors over the 3D position at each agent.

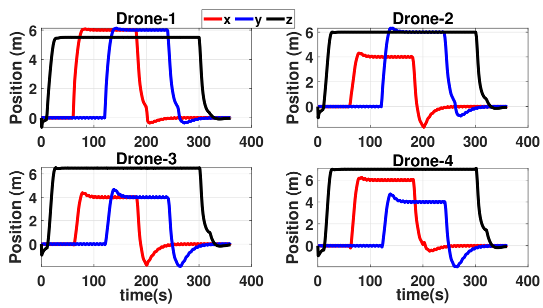

Fig. 13: Quadrotors application: 3D positions considering the reference trajectory and desired formation topology.

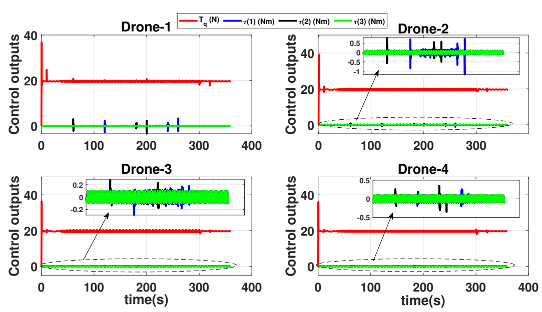

Fig. 14: Quadrotors application: Values for the main control variables (desired thrust and 3D torques) at each agent.

Two distributed model-free adaptive laws are proposed for online estimation of unknown dynamics. Specification of the network's communication graph is incorporated in adaptive laws. In addition, an online differential Riccati equation is proposed for updating the main controller gains. The decentralized control signal at each agent requires information about states of the leader agent, as well as the desired formation variables of agents in a local coordinate frame. These variables are provided at each agent by utilizing a recently-proposed distributed observer. It is shown that, only a spanning-tree is enough for stability of the proposed cooperative controller and observers. According to simulation results, by less control effort in CAMFC, one can reach the same level of consensus errors in network, compared with two other cooperative model-free control algorithms in consensus and formation-tracking problems. Moreover, CAMFC algorithm is used for formation-tracking of a network of four quadrotors. A desired formation among drones is maintained, while a desired trajectory is followed by the virtual leader of network. CAMFC algorithm can be merged with cooperative localization and path-planning algorithms, to deliver a complete package for collision-free operation of a fleet of autonomous mobile robots.

ACKNOWLEDGMENT

During this research, the author was under a Mitacs Accelerate fund at McGill University and HumanITAS start-up.

REFERENCES

[1] J. Ghommam and M. Saad, "Adaptive leader-follower formation control of underactuated surface vessels under asymmetric range and bearing constraints", *IEEE Tran. on Vehicular Technology*, vol. 67, no. 2, pp. 852-865, 2018.

[2] Y. Lu, G. Zhang, Z. Sun and W. Zhang, "Adaptive cooperative formation control of autonomous surface vessels with uncertain dynamics and external disturbances", *Ocean Engineering*, vol. 167, pp. 36-44, 2018.

[3] T. Lee, "Geometric control of quadrotor UAVs transporting a cable-suspended rigid body", *IEEE Tran. on Control Systems Technology*, vol. 26, no. 1, pp. 255-264, 2018.

[4] H. G. Tanner, A. Jadbabaie and G. J. Pappas, "Stable flocking of mobile agents, part I: fixed topology", in *42nd IEEE Int. Conference on Decision and Control*, Maui, HI, USA, 2003, pp. 2010-2015.

[5] H. G. Tanner, A. Jadbabaie and G. J. Pappas, "Stable flocking of mobile agents, part II: dynamic topology", in *42nd IEEE Int. Conference on Decision and Control*, HI, USA, 2003, pp. 2016-2021.

[6] Z. Li and Z. Duan, *Cooperative Control of Multi-agent Systems*. Boca Raton, FL, USA; CRC Press, Taylor and Francis Group: 2015.

[7] F. L. Lewis, H. Zhang, K. Hengster-Movric and A. Das, *Cooperative control of multi-agent systems*. London, UK;Springer-Verlog: 2014.

[8] R. Olfati-Saber, "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory", *IEEE Tran. Automatic Control*, vol. 51, no. 3, pp. 401-420, 2006.

[9] M. Fliess and C. Join, "Model-free control", *Int. Journal of Control*, vol. 86, no. 12, pp. 2228-2252, 2013.

[10] Z. Hou and S. Jin, *Model-free adaptive control*. Boca Raton, FL, USA; CRC Press, Taylor and Francis Group: 2014.

[11] R. Percup, M. Radac, R. Roman and E. M. Petriu, "Model-free sliding mode control of nonlinear systems: algorithms and experiments," *Information Sciences*, vol. 381, pp. 176-192, 2017.

[12] A. Safaei and M. N. Mahyuddin, "Adaptive Model-Free Control Based on an Ultra-Local Model With Model-Free Parameter Estimations for a Generic SISO System," *IEEE Access*, vol. 6, pp. 4266-4275, 2018.

[13] P. Ioannou and B. Fidan, *Adaptive control tutorial*. Philadelphia, USA; SIAM: 2006.

[14] A. Das and F.L. Lewis, "Distributed adaptive control for synchronization of unknown nonlinear networked systems", *Automatica*, vol. 46, no. 12, pp. 2014-2021, 2010.

[15] J. Zhang, Z. Wang and H. Zhang, "Data-based optimal control of multi-agent systems: A reinforcement learning design approach", *IEEE Tran. Cybernetics*, vol. 49, no. 12, pp. 4441-4449, 2019.

[16] W. Wang, D. Wang and Z.H. Peng, "Cooperative fuzzy adaptive output feedback control for synchronisation of nonlinear multi-agent systems under directed graphs", *Int. Journal of Systems Science*, vol. 46, no. 16, pp. 2982-2995, 2015.

[17] Z. Peng, D. Wang, H. Zhang, G. Sun and H. Wang, "Distributed model reference adaptive control for cooperative tracking of uncertain dynamical multi-agent systems", *IET Control Theory and Applications*, vol. 7, no. 8, pp. 1079-1087, 2013.

[18] Z. Peng, D. Wang, H. Zhang and Y. Lin, "Cooperative output feedback adaptive control of uncertain nonlinear multi-agent systems with a dynamic leader", *Neurocomputing*, vol. 149, pp. 132-141, 2015.

[19] W.C. Meng, Q.M. Yang, S. Jagannathan, Y.X. Sun "Adaptive neural control of high-order uncertain non-affine systems: A transformation to affine systems approach", *Automatica*,vol.50,no.5, pp.1473-1480, 2014.

[20] G. Cui, S. Xu, F.L. Lewis,B. Zhang and Q. Ma, "Distributed consensus tracking for non-linear multi-agent systems with input saturation: a command filtered backstepping approach", *IET Control Theory and Applications*, vol. 10, no.5, pp. 509-516, 2016.

[21] Q. Shen and P. Shi, "Output consensus control of multiagent systems with unknown nonlinear dead zone", *IEEE Tran. Systems, Man and Cybernetics: Sys.*, vol. 46, no. 10, pp. 1329-1337, 2016.

[22] W. Meng, Q. Yang, J. Sarangapani and Y. Sun, "Distributed control of nonlinear multiagent systems with asymptotic consensus", *IEEE Tran. Systems, Man and Cybernetics: Sys.*, vol. 47, no. 5, pp. 749-757, 2017.

[23] W. Wang, D. Wang and Z. Peng, "Predictor-based adaptive dynamic surface control for consensus of uncertain nonlinear systems in strict-feedback form", *Int. Journal of Adaptive Control and Signal Processing*, vol. 31, no. 1, pp. 68-82, 2017.

[24] W. Wang, C. Wen and J. Huang, "Distributed adaptive asymptotically consensus tracking control of nonlinear multi-agent systems with unknown parameters and uncertain disturbances", *Automatica*, vol. 77, pp. 133-142, 2017.

[25] J. Wang, "Distributed coordinated tracking control for a class of uncertain multiagent systems", *IEEE Tran. on Automatic Control*, vol. 62, no. 7, pp. 3423-3429, 2017.

[26] P. Shi and Q. K. Shen, "Observer-based leader-following consensus of uncertain nonlinear multi-agent systems", *Int. Journal of Robust and Nonlinear Control*, vol. 27, no. 17, pp. 3794-3811, 2017.

[27] G. Wen, W. Yu, Z. Li, X. Yu and J. Cao, "Neuro-adaptive consensus tracking of multiagent systems with a high-dimensional leader", *IEEE Tran. Cybernetics*, vol. 47, no. 7, pp. 1730-1742, 2017.

[28] X. Li, P. Shi and Y. Wang, "Distributed cooperative adaptive tracking control for heterogeneous systems with hybrid nonlinear dynamics", *Nonlinear Dynamics*, vol. 95, pp. 2131-2141, 2019.

[29] L. Cui, Y. Li, R. Yang and X. Zhang, "Asymptotic cooperative tracking control for unknown high-order multi-agent systems via distributed adaptive critic design", *IEEE Access*, vol. 6, pp. 24650-24659, 2018.

[30] W. Meng, Q. Yang, S. Jagannathan and Y. Sun, "Distributed control of high-order nonlinear input constrained multiagent systems using a backstepping-free method", *IEEE Tran. Cybernetics*, vol. 49, no. 11, pp. 3923-3933, 2019.

[31] N. T. Luy, "Adaptive dynamic programming-based design of integrated neural network structure for cooperative control of multiple MIMO nonlinear systems", *Neurocomputing*, vol. 137, pp. 12-24, 2017.

[32] V. Sakhre, P. Singh and S. Jain, "FCPN approach for uncertain nonlinear dynamical system with unknown disturbance," *Int. Journal of Fuzzy Systems*, vol. 19, no. 2, pp. 452-469, 2017.

[33] C.-E. Ren, L. Chen and C. L. P. Chen, "Adaptive fuzzy leader-following consensus control for stochastic multiagent systems with heterogeneous nonlinear dynamics", *IEEE Tran. Fuzzy Systems*, Vol. 25, no. 1, pp. 181-190, 2017.

[34] G. Cui, S. Xu, Q. Ma, Y. Li, Z. Zhang, "Prescribed performance distributed consensus control for nonlinear multi-agent systems with unknown dead-zone input", *Int. Journal of Control*, vol. 95, no. 5, pp. 1053-1065, 2018.

[35] S. Yang, Y. Cao, Z. Peng, G. Wen and K. Guo, "Distributed formation control of nonholonomic autonomous vehicle via RBF neural network," *Mechanical Systems and Signal Processing*, vol. 87, pp. 81-95, 2017.

[36] H. Wang, P. Shi, H. Li and Q. Zhou, "Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties," *IEEE Tran. on Cybernetics*, vol. 47, no. 10, pp. 3075-3087, 2016.

[37] Kaixin. Lu, Zhi. Liu, Guanyu. Lai, C. L. Philip Chen and Yun. Zhang, "Adaptive consensus tracking control of uncertain nonlinear multiagent systems with predefined accuracy," *IEEE Tran. on Cybernetics*, in Press, 10.1109/TCYB.2019.2933436, 2019.

[38] L. Cui, X. Wang and Y. Zhang, "Reinforcement learning-based asymptotic cooperative tracking of a class multi-agent dynamic systems using neural networks", *Neurocomputing*, vol. 171, pp. 220-229, 2016.

[39] S. Zuo, Y. Song, F. L. Lewis and A. Davoudi, "Optimal robust output containment of unknown heterogeneous multiagent system using off-policy reinforcement learning", *IEEE Tran. on Cybernetics*, vol. 48, no. 11, pp. 3197-3207, 2018.

[40] J. Zhang, H. Zhang, T. Feng "Distributed optimal consensus control for nonlinear multiagent system with unknown dynamics", *IEEE Tran. Neural Networks and Learning Systems*, vol.29, no.8, pp. 3339-3348, 2018.

[41] H. Modares, F.L. Lewis, W. Kang and A. Davoudi, "Optimal Synchronization of Heterogeneous Nonlinear Systems With Unknown Dynamics", *IEEE Tran. Automatic Control*, vol. 63, no. 1, pp. 117-131, 2018.

[42] G. Cui, G. Zhuang and J. Lu, "Neural-network-based distributed adaptive synchronization for nonlinear multi-agent systems in pure-feedback form", *Neurocomputing*, vol. 218, pp. 234-241, 2016.

[43] Z. Ding and Z. Li, "Distributed adaptive consensus control of nonlinear output-feedback systems on directed graphs", *Automatica*, vol. 72, pp. 46-52, 2016.

[44] C. P. Bechlioulis and G. A. Rovithakis, "Decentralized robust synchronization of unknown high order nonlinear multi-agent systems with prescribed transient and steady state performance", *IEEE Tran. on Automatic Control*, vol. 62, no. 1, pp. 123-134, 2017.

[45] H. A. Hashim, S. El-Ferik and F. L. Lewis, "Adaptive synchronization of unknown nonlinear networked systems with prescribed performance", *Int. Journal of Systems Science*, vol. 48, no. 4, pp. 885-898, 2018.

[46] X. Bu, L. Cui, Z. Hou and W. Qian, "Formation control for a class of nonlinear multiagent systems using model-free adaptive iterative learning", *Int. Journal of Robust and Nonlinear Control*, vol. 28, no. 4, pp. 1402-1412, 2018.

[47] X. Bu, Q. Yu, Z. Hou and W. Qian, "Model-free adaptive iterative learning consensus tracking control for a class of nonlinear multiagent systems", *IEEE Tran. on Systems, Man and Cybernetics: Systems*, vol. 49, no. 4, pp. 677-686, 2019.

[48] W. Gao, Y. Jiang and M. Davari, "Data-driven cooperative output regulation of multi-agent systems via robust adaptive dynamic programming", *IEEE Tran. on Circuits and Systems II: Express Briefs*, vol. 66, no. 3, pp. 447-451, 2019.

[49] C. Mu and Y. Zhang, "Learning-based robust tracking control of quadrotor with time-varying and coupling uncertainties," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 31, no. 1, pp. 259-273, 2019.

[50] A. Safaei and M. N. Mahyuddin, "Optimal model-free control for a generic MIMO nonlinear system with application to autonomous mobile robots," *Int. Journal of Adaptive Control and Signal Processing*, vol. 32, no. 6, pp. 792-815, 2018.

[51] A. Safaei and M. N. Mahyuddin, "A solution for the cooperative formation-tracking problem in a network of completely unknown nonlinear dynamic systems without relative position information," *Int. Journal of Systems Science*, vol. 49, no. 16, pp. 3459-3475, 2018.

[52] A. Safaei and M. N. Mahyuddin, "Distributed adaptive model-free cooperative control for a network of generic unknown nonlinear systems," *Int. Journal of Advanced Robotic Systems*, vol. 15, no. 5, pp. 1729881418801481, 2019.

[53] A. Safaei, Y. C. Koo and M. N. Mahyuddin, "Adaptive model-free control for robotic manipulators", in *Proceedings of 2017 IEEE Int. Symposium on Robotics and Intelligent Sensors*, Ottawa, ON, Canada, 2017, pp. 7-12.

[54] N. Fischer, R. Kamalpurkar and W. E. Dixon, "Lasalle-yoshizawa corollaries for nonsmooth systems," *IEEE Tran. on Automatic Control*, vol. 58, no. 9, pp. 2333-2338, 2013.

[55] A. N. Atassi and H. K. Khalil, "A separation principle for the stabilization of a class of nonlinear systems," *IEEE Tran. on Automatic Control*, vol. 44, no. 9, pp. 1672-1687, 1999.

[56] A. Safaei, M.N. Mahyuddin, "Adaptive cooperative localization using relative position estimation for networked systems with minimum number of communication links",*IEEE Access*,vol.7,pp. 32368-32382, 2019.

[57] A. Safaei and M.N. Mahyuddin, "Lyapunov-based nonlinear controller for quadrotor position and attitude tracking with GA optimization", in *IEEE Industrial Electronics and Applications Conference*, Kota Kinabalu, Malaysia, 2016, pp. 342-347.

Ali Safaei (SM'19) is a postdoctoral researcher at Mechanical Engineering Department, McGill University, Montreal, QC. He got BSc. and MSc. in Mechanical Engineering from Isfahan University of Technology and University of Tehran, Iran in 2009 and 2012, respectively. After gaining experiences for five years in automotive, oil-gas and power plant industries, he won a TWAS-USM postgraduate fellowship and started his PhD in 2017 at School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Penang, Malaysia. In 2019, he finished the PhD studies and received a MITACS Accelerate fund for a postdoctoral period at McGill University and HumanITAS start-up. His research interests include cooperative adaptive model-free control, cooperative localization and cooperative observers for multi-agent dynamic systems,