IEEE CSS

Received 24 November 2022; revised 28 January 2023; accepted 23 February 2023. Date of publication 27 February 2023; date of current version 16 March 2023. Recommended by Senior Editor Lacra Pavel.

Digital Object Identifier 10.1109/OJCSYS.2023.3250089

Model-Free Distributed Reinforcement Learning State Estimation of a Dynamical System Using Integral Value Functions

BABAK SALAMAT [1], GERHARD ELSBACHER [1], ANDREA M. TONELLO [2] (Senior Member, IEEE), AND LENZ BELZNER [1]

[1] AImotion Institute, Technische Hochschule Ingolstadt, 85049 Ingolstadt, Germany
[2] Institute of Embedded Systems, Alpen-Adria University Klagenfurt, 9020 Klagenfurt, Austria

CORRESPONDING AUTHOR: BABAK SALAMAT (e-mail: babak.salamat@thi.de)

ABSTRACT One of the challenging problems in sensor network systems is to estimate and track the state of a target point mass with unknown dynamics. Recent improvements in deep learning (DL) show a renewed interest in applying DL techniques to state estimation problems. However, the process noise is absent which seems to indicate that the point-mass target must be non-maneuvering, as process noise is typically as significant as the measurement noise for tracking maneuvering targets. In this paper, we propose a continuous-time (CT) model-free or model-building distributed reinforcement learning estimator (DRLE) using an integral value function in sensor networks. The DRLE algorithm is capable of learning an optimal policy from a neural value function that aims to provide the estimation of a target point mass. The proposed estimator consists of two high pass consensus filters in terms of weighted measurements and inverse-covariance matrices and a critic reinforcement learning mechanism for each node in the network. The efficiency of the proposed DRLE is shown by a simulation experiment of a network of underactuated vertical takeoff and landing aircraft with strong input coupling. The experiment highlights two advantages of DRLE: i) it does not require the dynamic model to be known, and ii) it is an order of magnitude faster than the state-dependent Riccati equation (SDRE) baseline.

INDEX TERMS Aerospace, consensus filters, deep learning, distributed filter, dynamical system model, neural dynamic programming, sensor networks.

I. INTRODUCTION

In nonlinear control theory, distributed estimation and tracking target dynamics is a fundamental task and challenging due to the unobservable estimation errors for hidden states in such loosely coupled sensor networks. If the sensing model and the target dynamics are known and linear in the states, the distributed Kalman Filtering (DKF) algorithm [1] can be realized. For instance, for a linear time-invariant (LTI) sensing model and the target object observed in additive Gaussian noise, the DKF comprises two dynamic consensus problems when the target object is estimated. Then, solving two dynamic consensus filters (i.e. a low-pass filter and a band-pass filter). The published approaches on this subject can roughly be separated into two complementary categories: (A) derivative-building filters focusing on the use of Jacobians [1], which are the truncated first-order Taylor series of the nonlinear functions, and (B) derivative-free filters focusing on avoiding the use of Jacobians [2], [3]. While both concepts are generally justified for heterogeneous multi-sensor fusion. The task becomes more intriguing for target dynamics that cannot be easily modeled or that are unknown. In such a case, a learning mechanism turns out to be an attractive solution to control and estimate dynamic systems interacting with a physical environment to do more demanding tasks like autonomous flying [4], solving decision-making problems [5], [6], etc.

Recent improvements in deep learning (DL) show a renewed interest in applying DL techniques to state estimation problems [7], [8]. The authors assume that the process noise is absent and propose a neural network trained trying to do state estimation directly via supervised learning. In practical

implementations, both process and measurement noises appear in the dynamical system models, and it is crucial to consider the effect of measurement noise.

Adaptivity and optimality are central elements combined to control unknown dynamical system models. Reinforcement learning (RL) refers to a class of such methodologies [9]. In a standard RL setting, the dynamical system model is unknown, and the optimal control mechanism is learned through the interaction with the system. The key idea is to learn and solve the value function to satisfy the Hamilton-Jacobi-Bellman (HJB) equation [10]. The celebrated policy iteration (PI) algorithm belongs to this class. The PI algorithm starts by considering the admissible average cost setting for a given control policy and uses this data to obtain an improved control policy. These two steps of policy evaluation and policy improvement are repeated until convergence to the optimal control is achieved [11]. Another closely related approach to learn continuous-time (CT) nonlinear systems is the neural network (NN) structure [12] which can be trained to yield an approximate solution of the value function at the policy evaluation step. This category of solutions is called *model-building* or *model-free* RL, and the term *model-free* is used to emphasize that no knowledge of the dynamical system is assumed.

In all the aforementioned results, assumptions were made only for availability of the state variable [13]. However, it is known they may perform weakly when the state variable is not fully observable and becomes even more challenging in distributed estimations for sensor networks setups [14]. This is particularly important in target tracking of a network, where only a noisy measurement of the output of each node is available for the tracking purpose. When a model-free methodology is engaged, it is not possible to use a filter, e.g. DKF, to estimate the state variable because the dynamical system model is unknown. The existing distributed estimation algorithms for target tracking using sensor networks are limited and highlighted by some publications [15], [16].

In this paper, we derive a CT model-free distributed reinforcement learning estimator (DRLE) using integral value function, based on PI, in sensor networks. A DRLE is a network framework of interacting intelligent microfilters. Each microfilter is implemented as an embedded component in a sensor architecture. The DRLE algorithm is capable of learning an optimal policy from a neural value function that aims to provide the estimate of a target point mass. We take one step further and replace the state of the target point mass with a noisy measurement of the state by each node of the network. It should be noted that the DRLE algorithm is in the model-free family of classical RL. The only information used for the simulation is the outputs of a highly coupled nonlinear target point mass of known order; no known dynamical system model is assumed to generate the data. There are three contributions to this paper. Our first contribution is to resolve the limitation of the existing derivative-building and derivative-free filters that needs the model dynamics by introducing a novel model-free microfilter with two identical high-pass consensus filters. Our second contribution is to increase the performance by an order of the magnitude. Lastly, our third contribution is to propose a base performance standard for distributed estimation and comparison with the state-dependent Riccati equation (SDRE) algorithm [2], [17] on target point mass tracking.

The rest of the paper is organized as follows. In Section II, we describe briefly the motivation and problem statement. Our main results including the DRLE algorithm with two high-pass consensus filters are presented in Section III. In Section IV, we give the simulation result and compare our proposed DRLE algorithm with the SDRE approach. Finally, the conclusions are drawn.

Notation: Unless stated otherwise, all vectors in this paper are column vectors. The state variables are a function of time, e.g., $x = x(t)$. The trace of an $n \times n$ square matrix A, denoted $tr(A)$, is defined to be the sum of elements on the main diagonal of A.

II. MOTIVATION

We present the motivations behind the need for a novel design of a model-free distributed reinforcement learning estimator (DRLE) using an integral value function for the online generation of an optimal policy. Finally, this policy leads to the optimal observer gain without knowing the dynamical system model of the target point mass.

A. PROBLEM STATEMENT

Consider a network topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes interconnected via an undirected graph. The nodes are denoted by the set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, and the set of links $\mathcal{E} = \{(v_i, v_j), v_i \neq v_j, v_i, v_j \in \mathcal{V}\} \subset \mathcal{V} \times \mathcal{V}$. The objective is to perform distributed state estimation of a nonlinear target point mass. To be more precise, let

$$\dot{x}(t) = a(x) + b(x)u(t) + B_w w(t) \tag{1}$$

$$y_i(t) = h_i(x) + v_i(t), \quad i = 1, 2, \ldots, n \tag{2}$$

be the dynamics of the target (e.g. a moving point mass) and the observation model of node i in the considered network. Here, $x(t) \in \mathbb{R}^m$ denotes the state of the target, $y_i(t) \in \mathbb{R}^{nr}$ represents the r-dimensional measurements vector obtained by n sensors, $u(t) \in \mathbb{R}^{\hat{m}}$ is the control input, and $w(t)$ and $v_i(t)$ are white noises with covariance matrices R_{ww} and R_{vv_i}, respectively. We also assume the h_i's are different across the entire network. The statistics of the target dynamics and the measurements noise are given by

$$E[w(t_1)\, w(t_2)^\top] = R_{ww}(t_1)\delta(t_1 - t_2), \tag{3}$$

$$E[v_i(t_1)\, v_j(t_2)^\top] = R_{vv_i}(t_1)\delta(t_1 - t_2)\delta_{ij}(t_1 - t_2), \tag{4}$$

where $\delta(.)$ is the Dirac delta function.

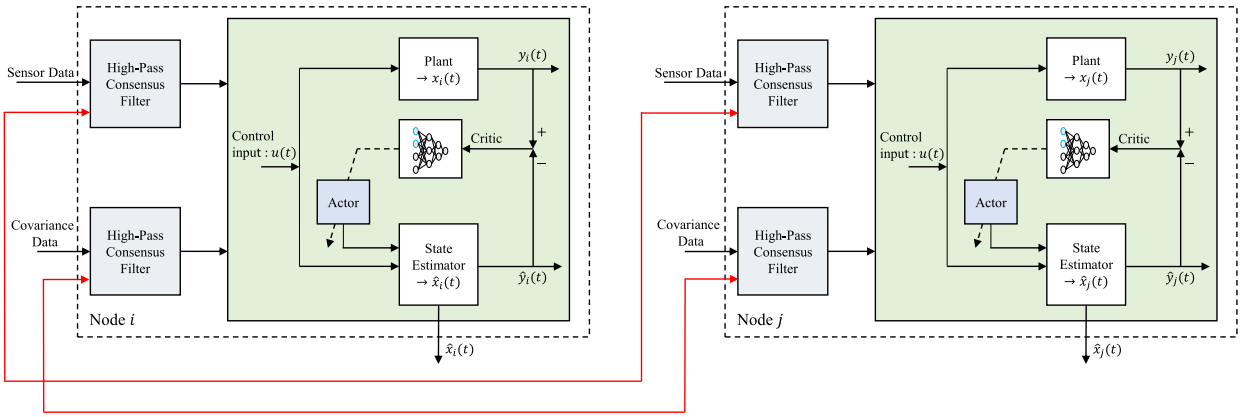

FIGURE 1. Network architecture for DRLE. The communication links between two high-pass consensus filters are shown with red arrows. The two high-pass filters shown in gray rectangles perform with the same frequency as the DRLE component in the green rectangle.

It is possible to transform (1) and (2) into a state-dependent coefficient (SDC) form as

$$\dot{x}(t) = A(x)x(t) + B(x)u(t) + B_w w(t) \tag{5}$$

$$y_i(t) = H_i(x)x(t) + v_i(t) \tag{6}$$

where $a(x) = A(x)x$, $b(x) = B(x)$ and $h_i(x) = H_i(x)x$. Due to the fact that we are trying to estimate the target point mass associated with the information data of each node $y_i(t)$ of the network, we should build the dynamics of the estimator, with the expression

$$\dot{\hat{x}}(t) = A(x)\hat{x}(t) + B(x)u(t) + L_i(t)\tilde{y}_i(t) \tag{7}$$

$$\tilde{y}_i(t) = H_i(x)\tilde{x}(t) \tag{8}$$

where $\tilde{y}_i(t) = y_i(t) - \hat{y}_i(t) = H_i(x)(x(t) - \hat{x}(t)) = H_i(x)\tilde{x}(t)$, $\hat{y}_i(t) = H_i(x)\hat{x}(t)$, $L_i(t)$ is the estimator gain to be computed online, and $\dot{\tilde{x}}(t) = (A(x) - L_i(t)H_i(x))\tilde{x}(t)$ is the closed-loop dynamics of the estimator.

The next step is to minimize the error covariance matrix for each node of the network

$$Q_i(t) = E[\tilde{x}(t)\,\tilde{x}(t)^\top]. \tag{9}$$

Taking the derivative of the error covariance matrix $Q_i(t)$ of each node, we get

$$\begin{aligned}\dot{Q}_i(t) = {} & (A(x) - L_i(t)H_i(x))\,Q_i(t) \\ & + Q_i(t)\,(A(x) - L_i(t)H_i(x))^\top \\ & + B_w R_{ww} B_w^\top + L_i(t)R_w L_i^\top(t).\end{aligned} \tag{10}$$

Therefore, the optimal estimator gain $L_i^*(t)$ is the solution of the following equation:

$$\frac{\partial tr(\dot{Q}_i(t))}{\partial L_i(t)} = -2Q_i^\top(t)H_i^\top(x) + 2L_i(t)R_{vv_i} = 0, \tag{11}$$

which gives the optimal gain:

$$L_i^*(t) = Q_i(t)H_i^\top(x)R_{vv_i}^{-1}. \tag{12}$$

It remains to determine the update rule for the error covariance matrix $Q_i(t)$ of each node. Substituting (12) in (10), we have the Riccati equation

$$\begin{aligned}\dot{Q}_i(t) = {} & A(x)Q_i(t) + Q_i(t)A^\top(x) + B_w R_{ww} B_w^\top \\ & - Q_i(t)H_i^\top(x)R_{vv_i}^{-1}H_i(x)Q_i(t).\end{aligned} \tag{13}$$

The SDRE information filter [2] can be exploited to solve the Riccati equation (13). However, the SDRE approach suffers from two key weaknesses. First, the Riccati equation in (10) needs to be solved in real-time, and the solution of the Riccati equation (if it exists) can be only represented numerically. Second, the dynamical system model of the target point mass should be known (e.g. $A(x)$). The proposed weaknesses motivate us to develop a novel DRLE for system networks with a broader range of aerospace applications. Furthermore, analytic closed-form expressions of the estimator gain $L_i(t)$ of each node approximating the numerical solutions offer several advantages: 1. Oversampling, thus, they provide a quick methodology to evaluate intermediate points. 2. In the context of optimal control, they can be used to express the observer feedback, allowing derivation and integration for state estimation. 3. They are often smooth functions (e.g. polynomials) and smoothness in observer feedback is a desired property. Furthermore, they can be easily implemented in micro-controllers.

III. DISTRIBUTED REINFORCEMENT LEARNING STATE ESTIMATION

In this section, we present a novel DRLE that does not rely on the dynamical system model of the target point mass. Now, substituting (12) in (7), the state propagation equation can be expressed as

$$\begin{aligned}\dot{\hat{x}}(t) = {} & A(x)\hat{x}(t) + B(x)u(t) \\ & + L_i^*(t)\,(y_i(t) - H_i(x)\hat{x}(t))\end{aligned} \tag{14}$$

$$\begin{aligned}= {} & A(x)\hat{x}(t) + B(x)u(t) \\ & + Q_i(t)\left(H_i^\top(x)R_{vv_i}^{-1}y_i(t) - H_i^\top(x)R_{vv_i}^{-1}H_i(x)\hat{x}(t)\right)\end{aligned} \tag{15}$$

The terms in (15) are two network aggregate quantities and represent the average measurements and the average inverse-covariance matrix, respectively. They can be written as follows:

$$p_i(t) = H_i^\top(x) R_{vv_i}^{-1} y_i(t), \quad p(t) = \frac{1}{n}\sum_{i=1}^{n} p_i(t), \tag{16}$$

$$Z(t) = \frac{1}{n}\sum_{i=1}^{n} H_i^\top(x) R_{vv_i}^{-1} H_i(x). \tag{17}$$

Both $p(t)$ and $Z(t)$ are time-varying quantities, and there is a need to solve two dynamic consensus problems that allow computing the average $p(t)$ and $Z(t)$. If one can compute the average $p(t)$ and $Z(t)$, the estimator emerges. This means that each node in the network calculates its consensus values $\hat{p}(t)$ and $\hat{Z}(t)$, respectively. To perform distributed averaging, we use high-pass consensus filter of [18]. Let $N_i = \{j : (i, j) \in \mathcal{E}\}$ be the set of neighbors of node i on graph $\mathcal{G}$. Furthermore, let $\mathcal{L} = \mathcal{D} - \mathcal{A}$ be the Laplacian matrix of $\mathcal{G}$, $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$, $a_{ij} \neq 0$ if $(\mathcal{V}_j, \mathcal{V}_i) \in \mathcal{E}$ otherwise $a_{ij} = 0$ is the adjacency matrix, and $\mathcal{D}$ denotes the degree matrix. The high-pass filter is in the following form

$$\dot{s}_i = \gamma \sum_{j\in N_i}(s_j - s_i) + \gamma \sum_{j\in N_i}(\bar{u}_j - \bar{u}_i); \quad \gamma > 0 \tag{18}$$

$$p_i = s_i + \bar{u}_i, \tag{19}$$

$$\bar{U}_j = H_j^\top(x) R_{vv_j}^{-1} H_j(x), \quad \forall j \in N_i \cup \{i\} \tag{20}$$

$$\dot{S}_i = \gamma \sum_{j\in N_i}(S_j - S_i) + \gamma \sum_{j\in N_i}(\bar{U}_j - \bar{U}_i); \quad \gamma > 0 \tag{21}$$

$$Z_i = S_i + \bar{U}_i \tag{22}$$

where $\bar{u}_i$ is the input of node i, s_i is the state of the consensus high-pass filter, and p_i is its output. The inputs of each node are $\bar{u}_i = H_i^\top(x) R_{vv_i}^{-1} y_i(t)$ and $\bar{U}_i(t) = H_i^\top(x) R_{vv_i}^{-1} H_i(x)$ with zero initial states $s_i(0) = 0$ of two filters. The output of high-pass filters asymptotically converge to $p(t)$ and $Z(t)$ in (16) and (17), respectively. Until now, we have described a formal methodology to estimate the state of a nonlinear dynamics of a target point mass under the observation of nodes in a heterogeneous sensor network.

A. MODEL-FREE DRLE USING AN INTEGRAL VALUE FUNCTION

In this subsection, we propose a model-free DRLE to estimate the state of a target point mass in a network under two separate dynamic consensuses in terms of weighted measurements and inverse-covariance matrices. The objective is to minimize the error covariance matrix for each node $Q_i(t)$ without solving the Riccati equation (13) (i.e. $\hat{x}(t) = x(t)\ \forall t \geq 0$). To proceed, consider the system model (14) and measurement model (15), then we define for each node the following integral value function

$$V_i(\hat{x}_i(t)) = \int_t^\infty r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau = \hat{x}_i^\top(t) Q_i(t) \hat{x}_i(t). \tag{23}$$

where $r_i(\hat{x}_i, \bar{u}_i) = G_i(\hat{x}) + \bar{u}_i^\top R_i \bar{u}_i$ with $G_i(\hat{x})$ positive definite, i.e. $\forall \hat{x} \neq 0$, $G_i(\hat{x}) > 0$ and $\hat{x} = 0 \rightarrow G_i(\hat{x}) = 0$, and $R_i \in \mathbb{R}^{m\times m}$ is a positive definite matrix for each node i. If we divide the integral in (23) into two terms, then

$$V_i(\hat{x}_i(t)) = \int_t^{t+\Delta t} r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau + \int_{t+\Delta t}^\infty r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau. \tag{24}$$

Considering $\Delta t \rightarrow 0$ is sufficiently small, the first integral term is a rectangle with the length of Δt and the width of $r_i(\hat{x}_i(t), \bar{u}_i(t))$ and results in

$$V_i(\hat{x}_i(t)) \approx \Delta t\, r_i(\hat{x}_i(t), \bar{u}_i(t)) + V_i(\hat{x}_i(t + \Delta t)) \tag{25}$$

Dividing both side of (25) by Δt, we get

$$r_i(\hat{x}(t), \bar{u}_i(t)) + \left[\frac{V_i(\hat{x}_i(t + \Delta t)) - V_i(\hat{x}_i(t))}{\Delta t}\right] = 0 \tag{26}$$

$$\rightarrow r_i(\hat{x}_i(t), \bar{u}_i(t)) + \dot{V}_i(\hat{x}_i(t)) = 0 \tag{27}$$

$$\rightarrow r_i(\hat{x}_i(t), \bar{u}_i(t)) + [\nabla V_i(\hat{x}_i(t))]^\top \frac{d}{dt}\hat{x}_i(t) = 0 \tag{28}$$

$$\rightarrow r_i(\hat{x}_i(t), \bar{u}_i(t)) + [\nabla V_i(\hat{x}_i(t))]^\top \dot{\hat{x}}_i(t) = 0 \tag{29}$$

The last (29) is the continuous time (CT) Bellman equation [19]. Solving the CT Bellman equation requires the full knowledge of the dynamical system model and yields poor generalization. To avoid this, we may write

$$V_i(\hat{x}_i(t)) = V_i(\hat{x}_i(t + T)) + \int_t^{t+T} r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau$$

$$\hat{x}^\top(t) Q_i(t) \hat{x}_i(t) = \hat{x}_i^\top(t + T) Q_i(t) \hat{x}_i(t + T) + \int_t^{t+T} r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau, \quad T > 0 \tag{30}$$

Using the Kronecker product $\bigotimes$ property, we can rewrite (30) as linear in the parameter vector $\bar{Q}_i(t) = \text{vec}(Q_i(t))$ in the following form

$$\left(\bar{x}_i^\top(t) - \bar{x}_i^\top(t + T)\right)\bar{Q}_i(t) = \int_t^{t+T} r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau, \tag{31}$$

where $\bar{x}_i(t) = \hat{x}_i(t) \bigotimes \hat{x}_i(t)$ is the quadratic polynomial vector containing pairwise products of the m components of $\hat{x}_i(t)$. Since our target point mass in (1) is a nonlinear dynamical system, the value function of each node $V_i(\hat{x}(t))$ requires higher-order nonlinearities. It is possible to approximate the proposed value function by a suitable approximator network in terms of unknown parameters which can be trained to become the approximate solution of the Riccati equation in (13) at the evaluation step.

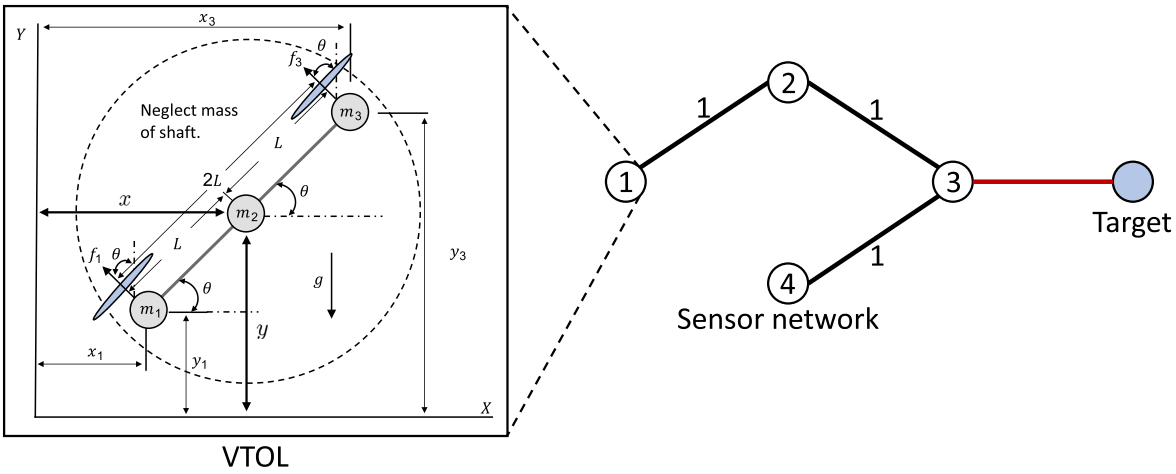

FIGURE 2. Communicating weighted network with four nodes and three links and the target point mass. Sensing appears along the x-axis, y-axis and rotation θ-axis. Edge labels indicate exemplary communication link weights.

The rationale behind the proposed integral value function is formally discussed in the next subsection.

B. CRITIC NEURAL NETWORK

By having an unknown dynamical system model of the target, a relevant question is the system identification of the plant. A critic neural network (NN) can be exploited for such a goal. Interestingly, the value function for each node can be approximated by a critic NN structure as

$$V_i(\hat{x}_i(t)) = \bar{x}_i^\top(t)\bar{Q}_i(t) = W_i^\top \Phi_i(\hat{x}), \tag{32}$$

with unknown parameters (weights) of each node $W_i = [w_{1_i}, w_{2_i}, \ldots, w_{h_i}]^\top$, the activation functions $\Phi_i(\hat{x}) = [\phi_{1_i}(\hat{x}), \phi_{2_i}(\hat{x}), \ldots, \phi_{h_i}(\hat{x})] : \mathbb{R}^m \to \mathbb{R}^h$, and h is referred to as the number of neurons in the hidden layer of the NN topology. Therefore, we can write (32) as

$$W_i^\top (\Phi_i(\hat{x}(t)) - \Phi_i(\hat{x}(t+T))) = \int_t^{t+T} r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau. \tag{33}$$

Remark 1: Equation (33) does not require knowledge of the dynamics $A(x), B(x)$. The observed data at each iteration with fixed time interval T is $(F_i(t), F_i(t+T), \int_t^{t+T} r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau)$ with $F_i(t) \equiv [\hat{x}_i^\top(t)\ \bar{u}_i^\top(t)]^\top \in \mathbb{R}^{m+\hat{m}}$. After convergence of the value function $V_i(\hat{x}(t))$ parameters, the control policy is performed. This can be accomplished by modifying the value function $V_i(\hat{x}_i(t), \bar{u}_i(t))$ containing $\bar{u}_i(t)$ as an argument. Therefore $\partial V_i(\hat{x}_i(t), \bar{u}_i(t))/\partial \bar{u}_i(t)$ can be explicitly computed.

From the relation in (33), it is clear that the optimization problem is quadratic and solvable in real time by a recursive least-squares (RLS) technique. Note that the DRLE itself is not enough to perform distributed state estimate for a target point mass. A distributed computation of $p(t)$ and $Z(t)$ is also required. The proposed approach provides the online generation of an optimal policy by measuring data by each node along the point mass system trajectories. To facilitate the implementation of the DRLE algorithm, we report the pseudo-code in Table 1.

Based on Fig. 1 and Algorithm 1, each node sends a message with size $\mathcal{O}(m(m+1))$ to its neighbors. The message consists of the state and input of its consensus filters (e.g. s_i, S_i, $\bar{u}_i$, $\bar{U}_i$).

The proposed DRLE offers a constructive learning methodology to design a distributed state estimator in a sensor network that approaches non-linearity even for unknown dynamical system models for which a unique solution for the Riccati matrix parameters in (13) does not exist. Indeed, learning involves numerical procedures that may introduce some challenges. Firstly, a critical aspect is the NN, and it is well-known that its performance depends on the architecture design, weights analysis, and hyper-parameters tuning. Secondly, the DRLE converges to the optimal weights by collecting mostly a low number of data points $(F_i(t), F_i(t+T), \int_t^{t+T} r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau)$ for each least-squares problem with the fixed time interval T.

IV. SIMULATION RESULTS

The assessment of the proposed DRLE approach without knowing the system matrix $A(x)$ is done by considering the following close-to-real simulation scenario: An agnostic target in a multi-node setup is that each node is a highly coupled nonlinear dynamical system (see Fig. 2) and in particular we show a Comparison with the SDRE algorithm [2], [17].

A. AGNOSTIC TARGET

The target point mass considered in this paper is an underactuated mechanical system with degree one which is also a complex nonlinear system [20]. The proposed novel VTOL deploys a system of three mass particles with masses m_1, m_2 and m_3, as shown in Fig. 2 and the dynamics

$$\ddot{q}_1 = -\epsilon_1 v_1 \sin(q_3),$$

Algorithm 1: Model-Building Distributed Reinforcement Learning Estimator with Two High-Pass Consensus Filters.

1: Initialization: $s_i = 0$, $Q_i = nQ_0$, $\hat{x}(0) = \bar{x}_0$, $S_i = 0_{m\times m}$
2: **while** new data exists **do**
3: Update the state of the average measurements
$\bar{u}_j = H_j^\top(x)R_{vv_j}^{-1}y_j(t),\ \forall j \in N_i \cup \{i\}$
$\dot{s}_i = \gamma \sum_{j\in N_i}(s_j - s_i) + \gamma \sum_{j\in N_i}(\bar{u}_j - \bar{u}_i),\ \gamma > 0$
$p_i(t) = s_i + \bar{u}_i$
4: Update the state of the average inverse-covariance
$\bar{U}_j = H_j^\top(x)R_{vv_j}^{-1}H_j(x),\ \forall j \in N_i \cup \{i\}$
$\dot{S}_i = \gamma \sum_{j\in N_i}(S_j - S_i) + \gamma \sum_{j\in N_i}(\bar{U}_j - \bar{U}_i),\ \gamma > 0$
$Z_i(t) = S_i + \bar{U}_i$
5: Solve RLS online ▷ Choose proper $\Phi_i(\hat{x})$
$W_i^\top(\Phi_i(\hat{x}, \bar{u}_i) - \Phi_i(\hat{x}(t+T), \bar{u}_i))$
$= \int_t^{t+T} r_i(\hat{x}_i(\tau), \bar{u}_i(\tau))d\tau$
6: Construct $Q_i(t)$ from $(\bar{x}_i^\top(t) - \bar{x}_i^\top(t+T))\bar{Q}_i(t)$
7: Reduce the estimation error of the point mass
$e_i(t) = Q_i(t)(p_i(t) - Z_i(t)\hat{x}(t))$
8: **end while**

$$\ddot{q}_2 = \alpha(g - \dot{q}_3^2 L\sin(q_3)) + \epsilon_1 v_1\cos(q_3) - \epsilon_1\epsilon_3 v_2\cos(q_3),$$
$$\ddot{q}_3 = -\epsilon_2 v_1 + \epsilon_2 v_2, \tag{34}$$

where $\epsilon_1 = \frac{-1}{m}$, $\epsilon_2 = \frac{1}{2^{\sim}L^{2^{\sim}}m}$, $\epsilon_3 = \frac{1}{L}$, $\alpha = -1.0002$, and $v = [v_1, v_2]^\top$. $q_i = [q_1,\ q_2,\ q_3]^\top = [x_1,\ y_1,\ \theta]^\top$ are the generalized coordinates.[1] An external thrust vector f_1 is applied to m_1 in the direction of $-x_1$ and y_1 respectively, and f_3 to m_3 in the direction of $-x_3$ and y_3 respectively. For simplicity, we assume that all representative particle masses are the same (e.g., $m_k = m$ for $k = 1,\ \ldots,\ 3$). The detailed dynamical system model is introduced in [20]. The objective is to track the state of the target point mass. To proceed, the VTOL dynamical model is used for nodes, $i = 1, 2, \ldots, 4$, and also the target point mass. By defining the state vector $x = [q_1,\ \dot{q}_1,\ q_2,\ \dot{q}_2,\ q_3,\ \dot{q}_3]^\top = [x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6]^\top$, and $u = [v_1,\ v_2]^\top = [u_1,\ u_2]^\top$ the system (34) for the target point mass can be transformed into the SDC form $\dot{x}(t) = A(x)x(t) + B(x)u(t) + B_w w(t)$ with $B_w = 0.01I_6$.

[1]The inertial measurement unit (IMU) is installed at the center of the shaft (e.g., x, y, θ). Therefore, the following shift is desirable $x = x_1 + L\cos(\theta)$, $y = y_1 + L\sin(\theta)$.

The network has $n = 4$ nodes with a topology shown in Fig. 2 and the Laplacian matrix denoted by

$$\mathcal{L} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{35}$$

The nodes (i.e. VTOLs) of the considered network make noisy measurement of the target VTOL along the x-axis, y-axis and rotation θ-axis, i.e.

$$y_i(t) = H_i(x)x(t) + v_i(t);\ \ H_i = I_6\ \ for\ \ i = 1, \ldots, 4.$$

Moreover, $R_{vv_1} = 3I_6, R_{vv_2} = 1.5I_6, R_{vv_3} = 1.5I_6, R_{vv_4} = 1.6I_6$. G_i and R_i in the value function are

$$G_i = diag(120,\ 120,\ 180,\ 120,\ 200,\ 5), \tag{37}$$
$$R_i = diag(15,\ 15)\ \ for\ \ i = 1, \ldots, 4. \tag{38}$$

Now, for maneuvering of the target point mass, we consider the following elaborated trajectory

$$q_1^* = 2.5\sin\left(\frac{t}{6}\right), \tag{39}$$
$$\dot{q}_1^* = \frac{5}{12}\cos\left(\frac{t}{6}\right), \tag{40}$$
$$q_2^* = 2.5\cos\left(\frac{t}{6}\right), \tag{41}$$
$$\dot{q}_2^* = -\frac{5}{12}\sin\left(\frac{t}{6}\right). \tag{42}$$

We approximate the value function in (33) as quadratic in states and inputs. Therefore, it is sufficient to choose the critic NN basis $\Phi_i(\hat{x})$ as the quadratic vector ($h = 35$) in the state and input components that is given in (36) shown at the bottom of this page. The DRLE time interval is set as $T = 1$ sec, with the total time of the simulation that is set to $T_f = 100$ sec. The nonlinear control of the target point mass $u(t)$ is also based on the proposed critic NN with the same activation function. Therefore, the closed-form control policy for the dynamics of target point mass (5) based on the considered activation functions is

$$u(t) = -\frac{1}{2}\begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \hat{x}_3 & \hat{x}_4 & \hat{x}_5 & \hat{x}_6 \\ \hat{x}_1 & \hat{x}_2 & \hat{x}_3 & \hat{x}_4 & \hat{x}_5 & \hat{x}_6 \end{bmatrix} \circ W_t, \tag{43}$$

$$\Phi_i(\hat{x}, \bar{u}) = \Big[\ \hat{x}_1^2\ \ \hat{x}_1\hat{x}_2\ \ \hat{x}_1\hat{x}_3\ \ \hat{x}_1\hat{x}_4\ \ \hat{x}_1\hat{x}_5\ \ \hat{x}_1\hat{x}_6\ \ \hat{x}_2^2\ \ \hat{x}_2\hat{x}_3\ \ \hat{x}_2\hat{x}_4\ \ \hat{x}_2\hat{x}_5\ \ \hat{x}_2\hat{x}_6\ \ \hat{x}_3^2\ \ \hat{x}_3\hat{x}_4\ \ \hat{x}_3\hat{x}_5\ \ \hat{x}_3\hat{x}_6\ \ \hat{x}_4^2\ \ \hat{x}_4\hat{x}_5\ \ \hat{x}_4\hat{x}_6\ \ \hat{x}_5^2\ \ \hat{x}_5\hat{x}_6$$
$$\hat{x}_6^2\ \ \hat{x}_1\bar{u}_1\ \ \hat{x}_2\bar{u}_1\ \ \hat{x}_3\bar{u}_1\ \ \hat{x}_4\bar{u}_1\ \ \hat{x}_5\bar{u}_1\ \ \hat{x}_6\bar{u}_1\ \ \bar{u}_1^2\ \ \hat{x}_1\bar{u}_2\ \ \hat{x}_2\bar{u}_2\ \ \hat{x}_3\bar{u}_2\ \ \hat{x}_4\bar{u}_2\ \ \hat{x}_5u_2\ \ \hat{x}_6\bar{u}_2\ \ \bar{u}_2^2\ \Big]^\top\ \ for\ \ i = 1, \ldots, 4. \tag{36}$$

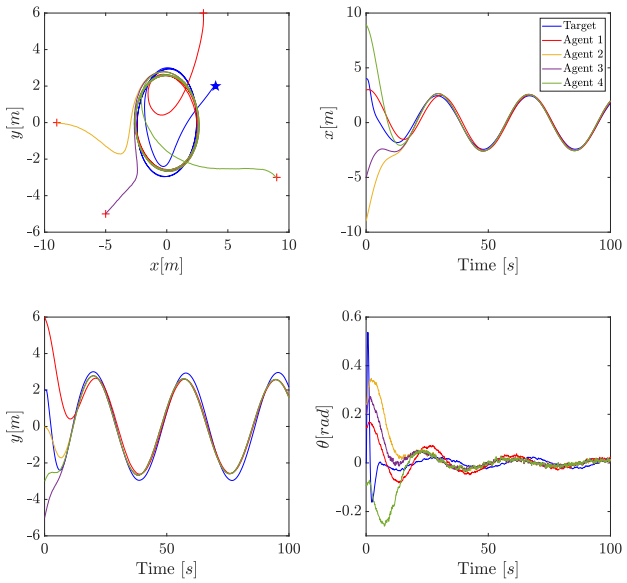

FIGURE 3. DRLE performance estimation to track the state of the target point mass by four nodes. Each node and the target is shown in red plus and blue star, respectively. Evolution of the states of the target point mass (in blue).

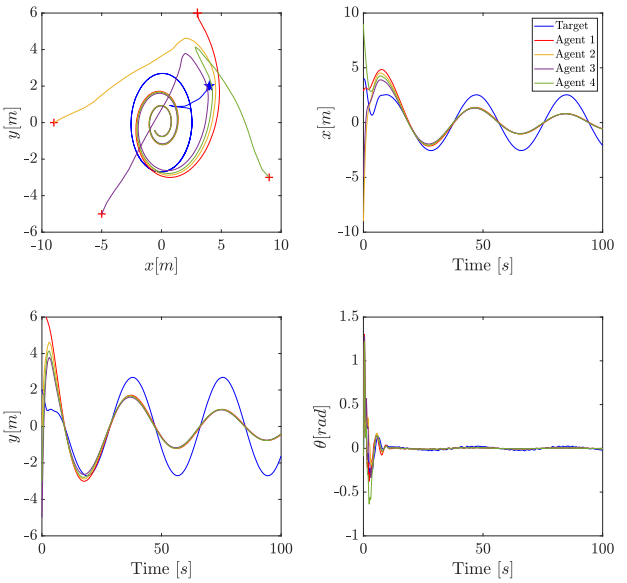

FIGURE 4. SDRE performance estimation. Each node and the target is shown in red plus and blue star, respectively. Evolution of the states of the target point mass (in blue).

where $\circ$ is the Hadamard product and

$$W_t = \begin{bmatrix} W_{22_t} & W_{23_t} & W_{24_t} & W_{25_t} & W_{26_t} & W_{27_t} \\ W_{29_t} & W_{30_t} & W_{31_t} & W_{32_t} & W_{33_t} & W_{34_t} \end{bmatrix}. \quad (44)$$

A probing noise is added to the control (43) to guarantee persistence of excitation. The initial state of the target is set as $x(0) = [4,\ 0,\ 2,\ 0,\ 0,\ 0]^\top$. The initial estimate of four nodes are: $\hat{x}_1(0) = [3,\ 0,\ 6,\ 0,\ 0.15,\ 0]^\top$, $\hat{x}_2(0) = [-9,\ 0,\ 0,\ 0,\ 0.24,\ 0]^\top$, $\hat{x}_3(0) = [-5,\ 0,\ -5,\ 0,\ 0.2,\ 0]^\top$ and $\hat{x}_4(0) = [9,\ 0,\ -3,\ 0,\ -0.09,\ 0]^\top$. Fig. 3 shows the learning performance of the DRLE in the agnostic scenario and the tracking estimation. After 366 iterations, the control policy $u(t)$ and the state estimation policies $Q_i(t)$ were acquired. The control policy of the target point mass converges to

$$W_t^* = \begin{bmatrix} 84.047 & 311.512 & 4.2042 & 20.2438 & -762.2035 & -201.7821 \\ -73.1246 & -288.9513 & -16.5996 & -70.807 & 811.8681 & 264.8218 \end{bmatrix}.$$

The estimates appear a cohesive set of VTOLs that estimate the trajectory of the target point mass. One can see that after 10 sec convergence has occurred. After that, the node's estimations remain very close to the target's trajectories, as required. A good approximation of the value function is being evolved. The results shows that DRLE can learn online the Riccati equation (13) without solving the differential equation and by using data measured along the target trajectories.

In Fig. 4, we consider the same elaborated trajectory as before, but now we show the performance of the SDRE. The estimates of all nodes by SDRE methodology are somewhat dispersed. Also, the results demonstrate the importance of DRLE which can be used to judge the usefulness of distributed estimation.

TABLE 1. Comparison of the standard deviation between SDRE and DRLE for 50 times random repetitions of the experiment.

Algo	node i	$\sigma_x(m)$	$\sigma_y(m)$	$\sigma_\theta(rad)$	Execution time
SDRE	VTOL 1	0.6390	0.7568	0.0472	5.2334 sec $\pm$0.4543
SDRE	VTOL 2	0.6005	0.7085	0.0447	
SDRE	VTOL 3	0.5621	0.6929	0.0422	
SDRE	VTOL 4	0.5729	0.6908	0.0409	
DRLE	VTOL 1	0.4348	0.9142	0.0536	0.3060 sec $\pm$0.0766
DRLE	VTOL 2	0.4193	0.7655	0.0498	
DRLE	VTOL 3	0.3989	0.7917	0.0483	
DRLE	VTOL 4	0.4117	0.7782	0.0495	

B. EXPERIMENTAL RESULTS

We now compare DRLE to the SDRE method. The aim is to assess the ability of DRLE to provide estimates. We randomize different components of the experiment. The randomization elements in this setup are the initial condition of the target point mass $x_0 \sim \mathcal{N}(0,\ \sigma_j^2 I_6)$ with $\sigma_j = 2$ for $j = 1, \ldots, 4$ and $\sigma_j = 0.5$ for $j = 5, 6$, the initial estimates $\hat{x}_{i_0} \sim \mathcal{N}(0,\ \sigma_j^2 I_6)$ with $\sigma_j = 3$ for $j = 1, \ldots, 4$ and $\sigma_j = 0.5$ for $j = 5, 6$, target trajectories, VTOL masses (e.g., $m_k = m$ for $k = 1, \ldots, 3$), and the lever arm L. A perturbation during the experiment is applied to VTOL masses $m_k + \epsilon_0$, and the lever arm $L + \epsilon_1$ with $\epsilon_0 = \mathcal{N}(0,\ 0.2^2)$ and $\epsilon_1 = \mathcal{N}(0,\ 0.01^2)$.

Table 1 shows, for several simulation tests, a comparison in terms of standard deviation between the actual state of the target point mass and the estimated filter (DRLE and SDRE)

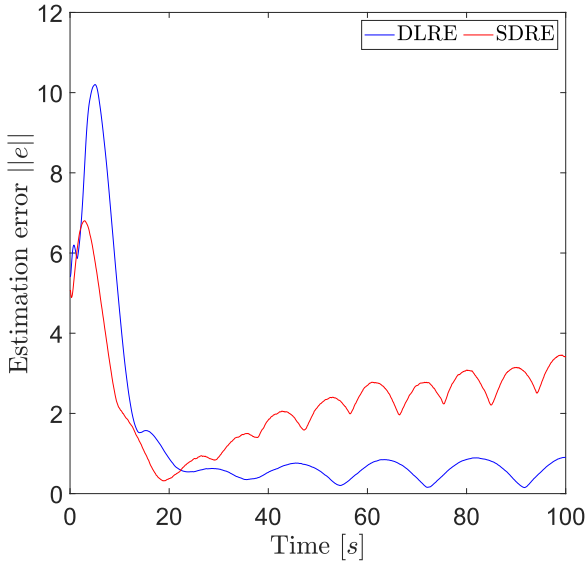

FIGURE 5. The estimation error ‖e‖ performance of DRLE vs. SDRE. Each curve is determined by averaging over 50 random iterations of each algorithm.

outputs in the network of four VTOLs. As can be seen from Table 1, both SDRE and DRLE provide good and similar estimations of the sensor network of VTOLs. However, DRLE results showed the gain provided by model-building neural based reinforcement learning estimator over the classical SDRE without knowing the system dynamics in Table 1.

It is also interesting to see the evolution of estimation error of the nodes independent of the network topology over time. To do so, we define the following measure $\|e\| = \sqrt{\sum_{i=1}^{n} e_i^2}$ with $e_i = \hat{x}_i - (\frac{1}{n}\sum_i \hat{x}_i)$. Fig. 5 demonstrates the comparison of DRLE vs. SDRE. As expected, the DRLE performs significantly better than SDRE in the asymptotic regime.

As mentioned before, the SDRE method is an online strategy for solving optimal control problems. However, it is computationally heavier because of the need for solving Riccati differential equations at each iteration. Furthermore, the dynamical system model of target $A(x)$ and $B(x)$ should be available. Another interesting fact is that the DRLE methodology is less sensitive to perturbation on the physical parameters of VTOLs in the network comparing the SDRE approach.

C. DISCUSSION

The proposed DRLE algorithm offers a constructive learning methodology to design a distributed estimation in sensor networks. It approaches collaboratively tracking, even for unknown target dynamics or for which a physical expression for the model does not exist. In summary, one may consider different factors when choosing the DRLE or SDRE approach. The DRLE and SDRE approaches are pretty much equal in terms of qualitative behavior. However, there are two advantages of DRLE: i) the DRLE method does not require the dynamic model to be known, and ii) DRLE is an order of magnitude faster.

V. CONCLUSION

In this paper, we have proposed a novel use of reinforcement learning, the distributed reinforcement learning estimator (DRLE) that solves the continuous-time model-free estimation problem in a senor network for a dynamic target. The DRLE allows the learning of an optimal policy from a neural value function that aims to provide the estimate of a target point mass. The nodes of the network agree on two central consensus sums which are high-pass filters in terms of weighted measurements and inverse-covariance matrices and a critic reinforcement learning mechanism for each node in the network. At the cost of distributed tracking, we gain the additional benefit of the model-free RL setting in identifying a complex moving target by each node. The details of the algorithm from a computational to communication architecture perspective were discussed. DRLE was applied to the novel network of underactuated VTOL aircraft with strongly coupled dynamics and simultaneous learning and collaborative estimation of the state of the target point mass, was accomplished. Our numerical evaluation shows that the DRLE algorithm produces faster policies over the classical SDRE approach. In future work, it would be interesting to extend the DRLE methodology into the direction of explainable machine learning, in which the learning process is motivated by an information-theoretic methodology.

REFERENCES

[1] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. IEEE 46th Conf. Decis. Control*, 2007, pp. 5492–5498.
[2] K. Pakki, B. Chandra, M. Kothari, D.-W. Gu, and I. Postlethwaite, "Multi-sensor state estimation using SDRE information filters," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 9141–9146, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016430570
[3] A. Sharma, S. C. Srivastava, and S. Chakrabarti, "A multi-agent-based power system hybrid dynamic state estimator," *IEEE Intell. Syst.*, vol. 30, no. 3, pp. 52–59, May/Jun. 2015.
[4] N. A. Letizia, B. Salamat, and A. M. Tonello, "A novel recursive smooth trajectory generation method for unmanned vehicles," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1792–1805, Oct. 2021.
[5] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
[6] T. Phan, L. Belzner, T. Gabor, and K. Schmid, "Leveraging statistical multi-agent online planning with emergent value function approximation," in *Proc. 17th Int. Conf. Auton. Agents MultiAgent Syst.*, 2018, pp. 730–738.
[7] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep learning," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4910–4920, Nov. 2019.
[8] N. J. Nair and A. Goza, "Leveraging reduced-order models for state estimation using deep learning," *J. Fluid Mechanics*, vol. 897, Jun. 2020, Art. no. R1, doi: 10.10172Fjfm.2020.409.
[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (Adaptive Computation and Machine Learning Series), 2nd ed., Cambridge, MA, USA: MIT Press, 2018.
[10] B. Salamat, N. A. Letizia, and A. M. Tonello, "Control based motion planning exploiting calculus of variations and rational functions: A formal approach," *IEEE Access*, vol. 9, pp. 121716–121727, 2021.
[11] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[12] K. G. Vamvoudakis and F. L. Lewis, "Online actor critic algorithm to solve the continuous-time infinite horizon optimal control problem," in *Proc. Int. Joint Conf. Neural Netw.*, 2009, pp. 3180–3187.
[13] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 3051–3056, Nov. 2014.
[14] R. Olfati-Saber and P. Jalalkamali, "Coupled distributed estimation and control for mobile sensor networks," *IEEE Trans. Autom. Control*, vol. 57, no. 10, pp. 2609–2614, Oct. 2012.
[15] P. Barooah, W. J. Russell, and J. P. Hespanha, "Approximate distributed Kalman filtering for cooperative multi-agent localization," in *Distributed Computing in Sensor Systems*, R. Rajaraman, T. Moscibroda, A. Dunkels, and A. Scaglione, Eds., Berlin, Germany: Springer, 2010, pp. 102–115.
[16] C. Soto, B. Song, and A. K. Roy-Chowdhury, "Distributed multi-target tracking in a self-configuring camera network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1486–1493.
[17] A. Nemra and N. Aouf, "Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering," *IEEE Sensors J.*, vol. 10, no. 4, pp. 789–798, Apr. 2010.
[18] D. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus for mobile networks," in *Proc. IFAC World Congr.*, 2005, pp. 1–8.
[19] L. C. Baird, "Reinforcement learning in continuous time: Advantage updating," in *Proc. IEEE Int. Conf. Neural Netw.*, 1994, pp. 2448–2453.
[20] B. Salamat and G. Elsbacher, "Centralized control in networks of underactuated nonidentical Euler–Lagrange systems using a generalised multicoordinates transformation," *IEEE Access*, vol. 10, pp. 58311–58319, 2022.

BABAK SALAMAT received the B.S. degree in mechanical engineering and the M.S. degrees in aerospace engineering from the Air-force University of Shahid Sattari, Tehran, Iran, in 2012 and 2014, respectively, and the Ph.D degree from the University of Klagenfurt, Klagenfurt, Austria, in 2021. He is currently a Postdoctoral Researcher of aerospace engineering with Technische Hochschule Ingolstadt, Ingolstadt, Germany. His research interests include navigation systems, path planning, nonlinear control of multi-agent systems, and reinforcement learning mechanisms. He was the co-recipient with A. Tonello of the 2018 Best Paper Award in the *Aerospace* journal.

GERHARD ELSBACHER received the Diploma in electrical engineering with a focus on automation technology from the Technical University of Munich, Munich, Germany, in 1993, and the Doctoral degree in mechanical engineering, with a dissertation on expert systems from the Vienna University of Technology, Vienna, Austria, in 2000. He joined LFK Lenkflugkörpersysteme GmbH as a development Engineer in 1997, and became after six years working on several projects in the field of G&C the Head of navigation, guidance and control, systems and real time simulation in 2003. After two years, in 2005, when LFK-Lenkflugkörpersysteme joined the international MBDA Group, he became the Vice President Subsystem Development Missile and Weapon Systems of MBDA Germany GmbH. After nine years in this role, in 2014, he became the Operations Director Germany and a Member of the Management Board of MBDA-Germany GmbH, responsible for Development, Production and Quality Management. In addition, from November 2019 to 2021, he was in engineering the international Director Capability and Governance of the MBDA Group, transversal responsible for Digitalisation, Skills and Improvement. Beside his industrial role, he has been a Curator with the Fraunhofer Institute IOSB since 2014 and Fraunhofer FHR since 2014. Since January 2022, he has been a Professor for AI aided Aeronautical Engineering and Product Development, Technische Hochschule Ingolstadt.

ANDREA M. TONELLO (Senior Member, IEEE) received the D.Eng. degree (Hons.) in electronics and the D.Res. degree in electronics and telecommunications from the University of Padova, Padua, Italy, in 1996 and 2002, respectively. From 1997 to 2002, he was with Bell Labs-Lucent Technologies, Whippany, NJ, USA, as a Member of the Technical Staff. Then, he was promoted to a Technical Manager and appointed to the Managing Director of the Bell Labs Italy Division. In 2003, he joined the University of Udine, Udine, Italy, where he became an Aggregate Professor in 2005 and an Associate Professor in 2014. He is currently the Chair of the Embedded Communication Systems Group, University of Klagenfurt, Klagenfurt, Austria. He is also the Founder of the spinoff company, WiTiKee. He was the recipient of several awards, including the Distinguished Visiting Fellowship from the Royal Academy of Engineering, U.K., in 2010, IEEE VTS and COMSOC Distinguished Lecturer Awards in 2011, 2015, and 2018, UC3M Chair of Excellence in 2019, and ten best paper awards. He was an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE ACCESS, *IET Smart Grid*, and Elsevier Journal of *Energy and AI*. During 2020–2021, he was the Director of industry outreach in the IEEE ComSoc Board of Governors

LENZ BELZNER is currently a Research Professor of software engineering for autonomous mobility systems with Technische Hochschule Ingolstadt, Ingolstadt, Germany. He is the co-author of more than 40 peer-reviewed publications at internationally renowned conferences. His research group works on autonomous mobility systems, statistical learning and planning, and multinode systems.