2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Distributed Optimization of Heterogeneous Agents by Adaptive Dynamic Programming

Haizhou Yang, Kedi Xie, Xiao Yu, Jinting Guan, Maobin Lu and Fang Deng

***Abstract*— In this paper, we study the distributed optimization problem of general linear multi-agent systems with heterogeneous dynamics under directed weight-unbalanced communication topologies. Compared with existing studies, we focus on the case when the dynamics of agents are unknown, which possesses higher application value. To tackle the issues brought by unknown system dynamics, the adaptive dynamic programming method is adopted to design the control law. The feedback gain in the control law and the system dynamics are derived from the input data, the state data, and the output data of the agents. Then, the remaining parameters in the control law are obtained by solving a series of matrix equations based on the identified system dynamics. Based on the certainty equivalence principle, the distributed optimization problem is solved in the sense that the outputs of all agents converge to the optimal solution of the global cost function. Finally, a simulation example concerning a group of resistor-inductor-capacitor (RLC) circuits is presented to verify the effectiveness of the proposed method.**

I. Introduction

The distributed optimization problem has been extensively investigated over the last decades [1]–[3]. The primary objective of distributed optimization is to minimize a global cost function based on communication of agents in the system, where the global cost function is defined as the summation of individual local cost functions for each agent in a multi-agent system. Distributed optimization has demonstrated its utility across a diverse range of applications, such as parameter estimation in sensor networks [4], energy consumption control of smart grids [5], and path planning of multiple robots [6].

One of the hotspots of distributed optimization problems is to address the case that the global cost function is related to all agents' states. By designing various appropriately distributed algorithms, the states of the agents are regulated to the optimal solution for the global cost function, see [7]–[12]. In particular, a distributed subgradient method is introduced in [7] to address the distributed optimization problem for agents with single integrator dynamics under jointly strongly connected communication topologies. Later, the constrained distributed optimization problem for agents with single integrator dynamics is explored in [8]. The distributed projected subgradient algorithm is further developed by introducing a projection operator with variable step size, achieving distributed optimization under jointly strongly connected communication topologies. Then, two types of distributed control laws are developed in [12] for agents with general linear dynamics. The coupling strengths are integrated into the control laws and the proposed control laws effectively solve the distributed optimization problem when communication topologies are undirected.

In some practical situations, the cost functions are associated with the agents' outputs. In particular, the optimal output consensus problem has garnered significant attention from researchers, see [13]–[17]. To be precise, an event-triggered control law based on a series of matrix equations is proposed in [14], and the approach effectively resolves the optimal output consensus problem for agents with general linear dynamics when communication topologies are undirected. Later, an integrated control law that incorporates both single and double integrals of relative output of agent i and its neighboring agents is introduced in [17]. Under the proposed control law, optimal output consensus for agents with general linear dynamics is realized when communication topologies are directed, weight-unbalanced, and strongly connected.

It is noteworthy that most of the existing results on solving the optimal output consensus problem predominantly are model-based approaches. In practical applications, it is often difficult to accurately obtain system dynamics. An effective way to address this issue is the adoption of model-free methods. Over the past decade, data-driven methods based on adaptive dynamic programming (ADP) have been widely explored, see [18]–[21] and the references therein. This type of methods utilizes real-time system data to derive control gains while simultaneously identifying system dynamics. However, the application of ADP-based data-driven methods to the optimal output consensus problem remains an open area of research.

In this paper, we further study the optimal output consensus problem for general linear systems by employing an ADP-based data-driven approach. First, we propose an ADP-based method to determine the feedback gain of the optimal output consensus control law. Moreover, we employ

This work was supported in part by National Key R&D Program of China under Grant 2021ZD0112600, in part by National Natural Science Foundation of China under Grant 62373058, 62173283, in part by the Beijing Natural Science Foundation under Grant L233003, in part by Key Program of National Natural Science Foundation of China under Grant 61933002, U20B2073, in part by National Science Fund for Distinguished Young Scholars of China under Grant 62025301, in part by the Basic Science Center Programs of NSFC under Grant 62088101, and in part by Shanghai Municipal Science and Technology Major Project 2021SHZDZX0100.

Haizhou Yang, Kedi Xie, Maobin Lu, and Fang Deng are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, and are also with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401120, China. Jinting Guan is with the Department of Automation, Xiamen University, Xiamen 361005, China. Xiao Yu is with the Institute of Artificial Intelligence, Xiamen University, Xiamen 361005, China. (e-mail:). E-mail: yanghz@bit.edu.cn, kedixie@bit.edu.cn, lumaobin@bit.edu.cn, jtguan@xmu.edu.cn, xiaoyu@xmu.edu.cn.

Corresponding author: Maobin Lu (lumaobin@bit.edu.cn).

a data-driven method to identify the parameters of the system dynamics. Then, we show the feasibility of our proposed method under some rank conditions for solving the optimal output consensus problem with unknown system dynamics. Finally, we present a comprehensive procedure for designing the optimal output consensus control law. The effectiveness of our method is validated through its application to the voltage balance optimal control of a group of RLC circuits. In comparison to [17], our method eliminates the need for prior knowledge of system dynamics. Instead, it relies solely on the data of states, inputs, and outputs of the controlled system to develop the control law, thereby broadening the potential applications of the proposed method. It is also interesting to extend the current work to some other cooperative control problem with system uncertainties and disturbances [22]–[25].

The rest of this paper is organized as follows. Section II gives some preliminaries used in this paper. Section III formulates the linear data-driven optimal output consensus problem and gives some assumptions. Section IV gives the main result in this paper. Section V gives a simulation example of the main result. Section VI gives the conclusion of this paper.

Notation. For $x_i \in \mathbb{R}^n$, $i = 1, \ldots, m$, $\text{col}(x_1, \ldots, x_m) = [x_1^{\mathrm{T}}, \ldots, x_m^{\mathrm{T}}]^{\mathrm{T}}$. For $M \in \mathbb{R}^{n\times n}$, $\text{diag}(M)$ denotes the diagonal matrix obtained from M by setting all off-diagonal entries equal to zero. $\text{diag}\{d_i\}$ denotes a diagonal matrix D of appropriate dimension with its i-th diagonal element being d_i. $||\cdot||$ denotes the induced norm of a matrix by the Euclidean norm or the Euclidean norm of vectors. $\mathbf{1}_N$ denotes a vector with all its entries being 1. $\mathbf{0}_{m\times n}$ denotes a $m \times n$ matrix with all its entries being zero. $\otimes$ denotes Kronecker product. For a matrix $A \in \mathbb{R}^{n\times n}$, Tr(A) is the trace of A. $\text{vec}(B) = [b_1^{\mathrm{T}}, b_2^{\mathrm{T}}, \ldots, b_s^{\mathrm{T}}]^{\mathrm{T}}$ with $b_i \in \mathbb{R}^r$ being the columns of matrix $B \in \mathbb{R}^{r\times s}$, and $\text{rank}(B) \in \mathbb{R}$ denotes the rank of matrix B. For a symmetric matrix $C \in \mathbb{R}^{n\times n}$, $\text{vecs(C)} = [c_{11}, 2c_{12}, \ldots, 2c_{1n}, c_{22}, 2c_{23}, \ldots, 2c_{2n}, \ldots, c_{nn}]^{\mathrm{T}}$, and for a vector $v \in \mathbb{R}^n$, $\text{vecv(v)} = [v_1^2, v_1v_2, \ldots, v_1v_n, v_2^2, v_2v_3, \ldots, v_{n-1}v_n, v_n^2]^{\mathrm{T}}$. For a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, ∇f denotes its gradient.

II. PRELIMINARIES

A directed graph (in short, a digraph) is used to describe the communication topology of the agents in a multi-agent system. A weighted digraph of order N can be expressed as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, N\}$ is the node set, $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ is the edge set describing the connectivity between two different nodes. For $i, j \in \mathcal{V}$, the ordered pair $(j, i) \in \mathcal{E}$ denotes an edge from j to i if node i can receive information from node j, but not vice versa. In this case, j is called the parent node of i, and i is called the child node of j. $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ is an associated weighted adjacency matrix. The entries of $\mathcal{A}$ satisfy $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$, and $a_{ij} = 0$ if $(j, i) \notin \mathcal{E}$. a_{ii} is always assumed to be 0 since it is generally assumed that there is no self-loop in a digraph. The Laplacian matrix $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{N\times N}$ of $\mathcal{G}$ can be defined as $l_{ii} = \sum_{j=1}^{N} a_{ij}$ and $l_{ij} = -a_{ij}$ for $i \neq j$. A digraph $\mathcal{G}$ is called weight balanced if and only if $\mathbf{1}_N^{\mathrm{T}}\mathcal{L} = \mathbf{0}_{1\times N}$, and is called weight unbalanced if the equation does not hold. A strongly connected digraph is a special type of digraph, and the detailed definition can be found in [26] and the references therein.

Lemma 2.1: (see [27]) Assume that $\mathcal{G}$ is strongly connected and weight unbalanced. Let $\mathcal{L}$ be its related Laplacian matrix. Then, the following properties hold:

1) Associated with the zero eigenvalue of $\mathcal{L}$, there exists a left eigenvector $r = (r_1, r_2, \ldots, r_N)^{\mathrm{T}}$ satisfying $r^{\mathrm{T}}\mathcal{L} = \mathbf{0}_N^{\mathrm{T}}$, $r_i > 0$ and $\sum_{i=1}^{N} r_i = 1$;

2) Let $R = \text{diag}(r_1, r_2, \ldots, r_N)$. Then, $\bar{\mathcal{L}} = (R\mathcal{L} + \mathcal{L}^{\mathrm{T}}R)/2$ is positive semidefinite. And the eigenvalues of $\bar{\mathcal{L}}$ can be ordered as $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_N$.

3) $e^{-\mathcal{L}t}$ is a non-negative matrix with its diagonal entries being positive, and $\lim_{t\to\infty} e^{-\mathcal{L}t} = \mathbf{1}_N r^{\mathrm{T}}$.

III. PROBLEM FORMULATION

Consider the following linear multi-agent system

$$\begin{aligned} \dot{\varrho}_i =& \Pi_i\varrho_i + \Xi_i\vartheta_i \\ \kappa_i =& \Psi_i\varrho_i,\ i = 1, 2, \ldots, N \end{aligned} \tag{1}$$

where $\varrho_i \in \mathbb{R}^{n_i}$, $\vartheta_i \in \mathbb{R}^{p_i}$, $\kappa_i \in \mathbb{R}^q$ represent the state, input, and measurement output of the i-th agent, respectively. $\Pi_i \in \mathbb{R}^{n_i\times n_i}$, $\Xi_i \in \mathbb{R}^{n_i\times p_i}$, and $\Psi_i \in \mathbb{R}^{q\times n_i}$, $i = 1, 2, \ldots, N$ are unknown constant matrices.

For every agent, there exists a local cost function $f_i(\kappa) : \mathbb{R}^q \to \mathbb{R}$ that is only obtainable by the i-th agent, where $\kappa \in \mathbb{R}^q$ is a vector. Let the global cost function be

$$f(\kappa) = \sum_{i=1}^{N} f_i(\kappa). \tag{2}$$

Consider a class of distributed optimization control laws of the form

$$\begin{aligned} \vartheta_i &= f_\vartheta(\varrho_i, \varphi_i, \zeta_i) \\ \dot{\zeta}_i &= \varphi_i = f_\zeta(f_i(\kappa_i), \tau_i, \kappa_i, \kappa_j, \chi_i) \\ \dot{\chi}_i &= f_\chi(\kappa_i, \kappa_j) \\ \dot{\tau}_i &= f_\tau(\tau_i, \tau_j) \end{aligned} \tag{3}$$

where $i, j = 1, 2, \ldots, N$ and $j \neq i$. Then, the distributed optimization problem is formulated as follows.

Problem 3.1: Consider the linear controlled multi-agent system (1), design a distributed control law ϑ_i in the form (3) such that the output of every agent reaches consensus on $\kappa^\star$, where $\kappa^\star$ satisfies

$$f(\kappa^\star) = \min_{\kappa\in\mathbb{R}^q} f(\kappa).$$

Problem 3.1 has been studied in [17] where all system matrices are known. In this paper, we consider the similar problem without prior knowledge of any system matrices in (1). To do it, the following assumptions are needed [17].

Assumption 3.1: The matrix pairs (Π_i, Ξ_i), $i = 1, \ldots, N$, are stabilizable and

$$\text{rank}\begin{bmatrix} \Psi_i\Xi_i & \mathbf{0}_{q\times p_i} \\ -\Pi_i\Xi_i & \Xi_i \end{bmatrix} = n_i + q.$$

Assumption 3.2: The directed weight-unbalanced communication graph $\mathcal{G}$ is strongly connected.

Assumption 3.3: For $i = 1, 2, \ldots, N$, the local cost function f_i is continuously differentiable and strongly convex with constant d_i, i.e., $(x-y)^{\mathrm{T}}(\nabla f_i(x) - \nabla f_i(y)) \geq d_i \|x - y\|^2$ for all $x, y \in \mathbb{R}^n$. ∇f_i is globally Lipschitz on $\mathbb{R}^q$ with constant D_i, i.e., $\|\nabla f_i(x) - \nabla f_i(y)\| \leq D_i \|x - y\|$ for all $x, y \in \mathbb{R}^n$.

Remark 3.1: Assumption 3.1 guarantees the solvability of the distributed optimization problem [14], [17], that is, there exist solution triplets $(\Upsilon_i, \Phi_i, \Lambda_i)$ to the following equations

$$\begin{aligned} \Psi_i \Lambda_i - I_q &= \mathbf{0}_{q \times q} \\ \Xi_i \Phi_i - \Pi_i \Lambda_i &= \mathbf{0}_{n_i \times q} \\ \Xi_i \Upsilon_i - \Lambda_i &= \mathbf{0}_{n_i \times q}. \end{aligned} \tag{4}$$

Assumption 3.3 guarantees that there exists a unique solution $\kappa^\star$ to minimize $f(\kappa)$.

IV. Main Result

A. Model-Based Control Law Design

We now introduce a model-based distributed optimization control law proposed in [17] of the form (3) to solve the distributed optimization problem 3.1. The distributed optimization control law is designed as follows:

$$\begin{aligned} \vartheta_i &= -U_i \varrho_i + \Upsilon_i \varphi_i - (\Phi_i - U_i \Lambda_i)\zeta_i \\ \dot{\zeta}_i &= \varphi_i := -\frac{\nabla f_i(\kappa_i)}{\tau_i^i} - \gamma_1 \sum_{j=1}^N a_{ij}(\kappa_i - \kappa_j) - \gamma_2 \chi_i \\ \dot{\chi}_i &= \gamma_1 \sum_{j=1}^N a_{ij}(\kappa_i - \kappa_j) \\ \dot{\tau}_i &= -\sum_{j=1}^N a_{ij}(\tau_i - \tau_j), i = 1, 2, \ldots, N \end{aligned} \tag{5}$$

where $\tau_i = \left[\tau_i^1, \tau_i^2, \ldots, \tau_i^N\right]^T \in \mathbb{R}^N$ with its initial value $\tau_i(0)$ satisfying $\tau_i^i(0) = 1$, $\tau_i^j(0) = 0$ for all $j \neq i$. $\zeta_i \in \mathbb{R}^q$ and $\chi_i \in \mathbb{R}^q$ are auxiliary variables. The initial value of ζ_i can be set randomly while the initial value of χ_i satisfies $\chi_i(0) = 0$. $\varphi_i \in \mathbb{R}^q$ is an intermediate state while its initial value can be set randomly. γ_1, γ_2 are positive scalars. $U_i \in \mathbb{R}^{p_i \times n_i}$ is a constant feedback gain matrix such that $(\Pi_i - \Xi_i U_i)$ is Hurwitz.

For notation convenience, define $\Psi = \mathrm{diag}\{\Psi_1, \Psi_2 \ldots, \Psi_N\}$, $D = \max\{D_1, D_2, \ldots, D_N\}$, $d = \min\{d_1, d_2, \ldots, d_N\}$ and $r_{\min} = \min\{r_1, r_2, \ldots, r_N\}$. It is shown in [17] that with the distributed control law (5), Problem 3.1 is solved. For completeness of the paper, we summarize the main result of [17] in the following lemma.

Lemma 4.1: Under Assumptions 3.1-3.3, the distributed optimization control law (5) solves Problem 3.1 with $(\Upsilon_i, \Phi_i, \Lambda_i)$ being the solutions to (4) and

$$\left\{ \begin{array}{l} \gamma_1 > \gamma_2^2 / \lambda_2 \delta \\ \gamma_2 > \left(5\delta^2 + \|\Psi\|^2\right) / \left(4\delta r_{\min}\right) \\ \delta > \left(D^2 + \|\Psi\|^2\right) / (2d). \end{array} \right. \tag{6}$$

In this paper, we address the scenario where all the matrices Π_i, Ξ_i, $\Psi_i, i = 1, 2, \ldots, N$ are unknown. This indicates that direct solutions to the matrix equations (4) are unobtainable, rendering the gain matrices in the control law ϑ_i unavailable. To deal with the difficulty caused by unknown system dynamics, we propose a data-driven approach to design the distributed control law (5).

B. Data-Driven-Based Control Law Design

In this section, we construct the control law utilizing a data-driven approach. First, we present the ADP-based identification framework for matrices $\Xi_i, i = 1, 2, \ldots, N$. For each agent, we formulate a cost function as follows:

$$f_{ADP,i}(\varrho_i, \vartheta_i) = \int_0^\infty (\varrho_i^{\mathrm{T}} E_i \varrho_i + \vartheta_i^{\mathrm{T}} F_i \vartheta_i) d\tau \tag{7}$$

where $E_i = E_i^{\mathrm{T}} \geq 0$, $F_i = F_i^{\mathrm{T}} > 0$ and $(\sqrt{E_i}, \Pi_i)$ is observable for every agent. Then, there exists control law $\vartheta_i^\star = -U_i^\star \varrho_i, i = 1, 2, \ldots, N$ that minimizes the cost function $f_{ADP,i}$ associated with every agent, and $U_i^\star$ satisfies

$$U_i^\star = F_i^{-1} \Xi_i^{\mathrm{T}} S_i \tag{8}$$

with $S_i = S_i^{\mathrm{T}} > 0$ being a solution to the following Riccati equation

$$\Pi_i^{\mathrm{T}} S_i + S_i \Pi_i + E_i - S_i \Xi_i F_i^{-1} \Xi_i^{\mathrm{T}} S_i = 0. \tag{9}$$

To support the subsequent analysis, we present the widely-employed iteration technique for solving the Riccati equation (9) as described in [28].

Lemma 4.2: Let $U_{i,0} \in \mathbb{R}^{m_i \times n_i}$ be an arbitrary stabilizing feedback gain matrix for the pair (Π_i, Ξ_i), $S_{i,k} = S_{i,k}^{\mathrm{T}} \geq 0$ is the solution to the following equation

$$(\Pi_i - \Xi_i U_{i,k})^{\mathrm{T}} S_{i,k} + S_{i,k}(\Pi_i - \Xi_i U_{i,k}) + E_i + U_{i,k}^{\mathrm{T}} R U_{i,k} = 0 \tag{10}$$

where for every $i = 1, 2, \ldots, N$, $k = 1, 2, \ldots$

$$U_{i,k} = F_i^{-1} \Xi_i^{\mathrm{T}} S_{i,k-1}. \tag{11}$$

Then, the following properties hold:

1) The eigenvalues of $\Pi_i - \Xi_i U_{i,k}$ have negative real parts.
2) $S_i^\star \leq S_{i,k+1} \leq S_{i,k} \leq \ldots \leq S_{i,0}$.
3) $\lim_{k \to \infty} U_{i,k} = U_i^\star, \lim_{k \to \infty} S_{i,k} = S_i^\star$.

Remark 4.1: Lemma 4.2 is the basis of solving the control gain with the data-driven method, and it also plays a key role in identifying the controlled system.

By integrating the system dynamics (1) with the iterative equation (10), one can derive

$$\begin{aligned} &\varrho_i^{\mathrm{T}}(t+dt) S_{i,k} \varrho_i^{\mathrm{T}}(t+dt) - \varrho_i^{\mathrm{T}}(t) S_{i,k} \varrho_i^{\mathrm{T}}(t) \\ &= \int_t^{t+dt} \varrho_i^{\mathrm{T}}((\Pi_i - \Xi_i U_{i,k})^{\mathrm{T}} S_{i,k} + S_{i,k}(\Pi_i - \Xi_i U_{i,k})) \varrho_i d\tau \\ &\quad + 2\int_t^{t+dt} \vartheta_i^{\mathrm{T}} \Xi_i^{\mathrm{T}} S_{i,k} \varrho_i d\tau + 2\int_t^{t+dt} \varrho_i^{\mathrm{T}} U_{i,k}^{\mathrm{T}} \Xi_i^{\mathrm{T}} S_{i,k} \varrho_i d\tau \\ &= \int_t^{t+dt} -\varrho_i^{\mathrm{T}}(E_i + U_{i,k}^{\mathrm{T}} F_i U_{i,k}) \varrho_i d\tau + 2\int_t^{t+dt} \vartheta_i^{\mathrm{T}} F_i \times \\ &\quad U_{i,k+1} \varrho_i d\tau + 2\int_t^{t+dt} \varrho_i^{\mathrm{T}} U_{i,k}^{\mathrm{T}} F_i U_{i,k+1} \varrho_i d\tau \end{aligned}$$

$$
\begin{aligned}
&= -\int_t^{t+dt} \varrho_i^{\mathrm{T}} \otimes \varrho_i^{\mathrm{T}} d\tau \mathrm{vec}(E_i + U_{i,k}^{\mathrm{T}} F_i U_{i,k}) \\
&\quad + 2\int_t^{t+dt} (\varrho_i^{\mathrm{T}} \otimes \vartheta_i^{\mathrm{T}})(I_{ni} \otimes F_i) d\tau \mathrm{vec}(U_{i,k+1}) \\
&\quad + 2\int_t^{t+dt} (\varrho_i^{\mathrm{T}} \otimes \varrho_i^{\mathrm{T}})(I_{ni} \otimes U_{i,k}^{\mathrm{T}} F_i) d\tau \mathrm{vec}(U_{i,k+1}) \qquad (12)
\end{aligned}
$$

with $dt > 0$.

For an integer $s > 0$, define

$$
\begin{aligned}
\delta_{aa} = &[\mathrm{vecv}(a(t_1)) - \mathrm{vecv}(a(t_0)), \mathrm{vecv}(a(t_2)) - \mathrm{vecv}(a(t_1)), \\
&\ldots, \mathrm{vecv}(a(t_s)) - \mathrm{vecv}(a(t_{s-1}))]^{\mathrm{T}} \\
\Gamma_{ab} = &\left[\int_{t_0}^{t_1} a \otimes b \,\mathrm{d}\tau, \int_{t_1}^{t_2} a \otimes b \,\mathrm{d}\tau, \ldots, \int_{t_{s-1}}^{t_s} a \otimes b \,\mathrm{d}\tau\right]^{\mathrm{T}}
\end{aligned}
$$

where a and b represent certain vectors or matrices and $0 < t_0 < t_1 < \ldots < t_s$. Then, we have

$$
\Omega_{i,k} \begin{bmatrix} \mathrm{vecs}(S_{i,k}) \\ \mathrm{vec}(U_{i,k+1}) \end{bmatrix} = \Theta_{i,k} \qquad (13)
$$

where

$$
\begin{aligned}
\Omega_{i,k} &= \left[\delta_{\varrho_i \varrho_i}, -2\Gamma_{\varrho_i \varrho_i}(I_{n_i} \otimes U_{i,k}^{\mathrm{T}} F_i) - 2\Gamma_{\varrho_i \vartheta_i}(I_{n_i} \otimes F_i)\right] \\
\Theta_{i,k} &= -\Gamma_{\varrho_i \varrho_i} \mathrm{vec}(E_i + U_{i,k}^{\mathrm{T}} F_i U_{i,k}).
\end{aligned}
$$

The uniqueness of the solution for (13) is guaranteed under the following rank condition.

Lemma 4.3: If the rank condition

$$
\mathrm{rank}\left[\Gamma_{\varrho_i \varrho_i}, \Gamma_{\varrho_i \vartheta_i}\right] = \frac{n_i(n_i+1)}{2} + m_i n_i \qquad (14)
$$

is satisfied, $S_{i,k}$ and $U_{i,k+1}$ can be solved uniquely from equation (13) for every iteration $k = 0, 1, 2, \cdots$.

It is worth mentioning that the rank condition (14) can be satisfied by adding exploration noise into the control input ϑ_i during the learning process [18], [19]. The exploration noise can be chosen as random noise, sinusoidal signals, and so on. The detailed ADP procedure for solving equation (13) is outlined in Algorithm 1.

Algorithm 1 Data-Driven ADP Procedure for Multi-Agent System

1: Set $i = 1$.
2: **repeat**
3: Select a sufficiently small threshold $\epsilon > 0$ and an admissible control gain $U_{i,0}$ with $U_{i,0}$ satisfying $\Pi_i - \Xi_i U_{i,0}$ is Hurwitz. Set $k = 0$.
4: Apply $\vartheta_i = -U_{i,0}\varrho_i + \xi_i$ to collect data until (14) holds, where ξ_i represents exploration noise.
5: **repeat**
6: Solve $S_{i,k}$ and $U_{i,k+1}$ from (13).
7: $k \leftarrow k + 1$.
8: **until** $|U_{i,k+1} - U_{i,k}| < \epsilon$
9: $U_i^\star \leftarrow U_{i,k+1}$, $S_i^\star \leftarrow S_{i,k}$.
10: $i \leftarrow i + 1$.
11: **until** $i = N + 1$.

Then, the input matrices of all agents can be reconstructed as follows.

Lemma 4.4: Given $S_i^\star$ and $U_i^\star$ solved from Algorithm 1, the input matrices $\Xi_i, i = 1, 2, \ldots, N$ of all the agents can be determined from the following equation

$$
\Xi_i = (S_i^\star)^{-1}(U_i^\star)^{\mathrm{T}} F_i. \qquad (15)
$$

Lemma 4.4 suggests that the matrices $\Xi_i, i = 1, 2, \ldots, N$ can be determined once $U_i^\star$ and $S_i^\star$ are known. With the reconstructed matrices $\Xi_i, i = 1, 2, \ldots, N$, the matrices $\Pi_i, i = 1, 2, \ldots, N$ can be approximated through a data-driven approach. From (1), one can obtain

$$
\varrho_i(t) = \int_0^t \Pi_i \varrho_i d\tau + \int_0^t \Xi_i \vartheta_i d\tau + \varrho_i(0), i = 1, 2, \ldots, N. \qquad (16)
$$

Then, to avoid utilizing the unknown information of $\varrho_i(0)$, it can be derived from (16) that

$$
\begin{aligned}
\varrho_i(t_1) - \varrho_i(t_0) &= \int_{t_0}^{t_1} \Pi_i \varrho_i d\tau + \int_{t_0}^{t_1} \Xi_i \vartheta_i d\tau \\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad (17) \\
\varrho_i(t_s) - \varrho_i(t_{s-1}) &= \int_{t_{s-1}}^{t_s} \Pi_i \varrho_i d\tau + \int_{t_{s-1}}^{t_s} \Xi_i \vartheta_i d\tau.
\end{aligned}
$$

By applying the Kronecker product, one can obtain

$$
\begin{aligned}
\int_{t_{s-1}}^{t_s} \Pi_i \varrho_i d\tau &= \int_{t_{s-1}}^{t_s} \varrho_i^{\mathrm{T}} \otimes I_{n_i} d\tau \mathrm{vec}(\Pi_i) \\
\int_{t_{s-1}}^{t_s} \Xi_i \vartheta_i d\tau &= \int_{t_{s-1}}^{t_s} \vartheta_i^{\mathrm{T}} \otimes I_{n_i} d\tau \mathrm{vec}(\Xi_i).
\end{aligned} \qquad (18)
$$

Define

$$
\Delta_{aa} = [a(t_1) - a(t_0), a(t_2) - a(t_1), \ldots, a(t_s) - a(t_{s-1})]^{\mathrm{T}} \qquad (19)
$$

and (17) can be expressed as

$$
\Delta_{\varrho_i \varrho_i} - \Gamma_{\vartheta_i I_{n_i}} \mathrm{vec}(\Xi_i) = \Gamma_{\varrho_i I_{n_i}} \mathrm{vec}(\Pi_i). \qquad (20)
$$

To identify matrices $\Pi_i, i = 1, 2, \ldots, N$, similar to (14), the following rank condition is given.

Lemma 4.5: The equation (20) has a unique solution Π_i if the following rank condition is satisfied

$$
\mathrm{rank}(\Gamma_{\varrho_i I_{n_i}}) = (n_i)^2. \qquad (21)
$$

During the identification process, by adding the exploration noise ξ_i into ϑ_i as step 4 in Algorithm 1, the rank condition (21) can also be easily satisfied.

Similar to Π_i, we now identify the matrices $\Psi_i, i = 1, 2, \ldots, N$. From (1), κ_i can be written as

$$
\kappa_i = (\varrho_i^{\mathrm{T}} \otimes I_q) \mathrm{vec}(\Psi_i). \qquad (22)
$$

Define $K_{i,0}^s = [\kappa_i^{\mathrm{T}}(t_0), \kappa_i^{\mathrm{T}}(t_1), \ldots, \kappa_i^{\mathrm{T}}(t_s)]^{\mathrm{T}}$ and $P_{i,0}^s = \left[(\varrho_i^{\mathrm{T}}(t_0) \otimes I_q)^{\mathrm{T}}, (\varrho_i^{\mathrm{T}}(t_1) \otimes I_q)^{\mathrm{T}}, \ldots, (\varrho_i^{\mathrm{T}}(t_s) \otimes I_q)^{\mathrm{T}}\right]^{\mathrm{T}}$. Then, we have

$$
K_{i,0}^s = P_{i,0}^s \mathrm{vec}(\Psi_i). \qquad (23)
$$

To identify Ψ_i, the following rank condition similar to (21) is needed.

Lemma 4.6: The equation (23) has a unique solution Ψ_i if the following condition is satisfied

$$
\mathrm{rank}(P_{i,0}^s) = n_i q. \qquad (24)
$$

Algorithm 2 Optimal Output Consensus Control Law Design

Set $i = 0$.
repeat
    $i \leftarrow i + 1$.
    Collect data until (14), (21) and (24) are satisfied.
    Obtain $U_i^\star$ and $S_i^\star$ from Algorithm 1.
    Solve Ξ_i from (15).
    Solve Π_i from (20).
    Solve Ψ_i from (23).
    Solve matrix equations (4).
until $i = N$.
Select γ_1, γ_2 under (6).

Finally, we give the complete design mechanism for the control law, which is outlined in Algorithm 2.

Theorem 4.1: Under Assumptions 3.1-3.3, the optimal output consensus control law can be designed by Algorithm 2, and the designed control law solves Problem 3.1.

Proof: From Lemma 4.4, Ξ_i can be obtained from (15) under rank condition (14). Besides, it follows from [19] that Algorithm 1 is equivalent to Lemma 4.2, and according to Lemma 4.2, $U_i^\star, i = 1, 2, \ldots, N$ obtained from Algorithm 1 are admissible, i.e., $\Pi_i - \Xi_i U_i^\star, i = 1, 2, \ldots, N$ are Hurwitz. Thus, $U_i^\star$ obtained from Algorithm 2 is admissible. From Lemmas 4.5 and 4.6, under rank conditions (21) and (24), Π_i and Ψ_i can be uniquely solved from (20) and (23) respectively. Under Lemma 4.1, there exist solution triples $(\Upsilon_i, \Phi_i, \Lambda_i)$ of (4). Therefore, the distributed control law obtained from Algorithm 2 can solve Problem 3.1 by invoking Lemma 4.1. This completes the proof. □

V. Simulation

Consider the voltage balance optimal control of the RLC network, and the circuit is shown in Fig. 1 [17]. The state-space model of the RLC network is described as

$$\begin{aligned} \dot{\varrho}_i &= \Pi_i \varrho_i + \Xi_i \vartheta_i \\ \kappa_i &= \Psi_i \varrho_i, i = 1, 2, 3, 4, 5 \end{aligned} \tag{25}$$

where ϱ_{i1} denotes the voltage across C_{i1}, ϱ_{i2} denotes the voltage across C_{i2}, ϱ_{i3} denotes the current through L_i. Π_i, Ξ_i, Ψ_i are constant matrices and satisfy

$$\Pi_i = \begin{bmatrix} -\frac{1}{C_{i1}R_{i1}} & 0 & -\frac{1}{C_{i1}} \\ 0 & 0 & \frac{1}{C_{i2}} \\ \frac{1}{L_i} & -\frac{1}{L_i} & -\frac{R_{i2}}{L_i} \end{bmatrix} \Xi_i = \begin{bmatrix} \frac{1}{C_{i1}R_{i1}} & \frac{1}{C_{i1}} \\ 0 & -\frac{1}{C_{i2}} \\ 0 & \frac{1}{L_i} \end{bmatrix},$$
$$\Psi_i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

The specific parameters of the state-space model are given as

$$\begin{aligned} &R_{11} = 2, R_{12} = 2, L_1 = 1, C_{11} = 2, C_{12} = 2 \\ &R_{21} = 0.5, R_{22} = 2, L_2 = 2, C_{21} = 2, C_{22} = 0.5 \\ &R_{31} = 1, R_{32} = 3, L_3 = 0.5, C_{31} = 1, C_{32} = 1 \\ &R_{41} = 2, R_{42} = 1, L_4 = 1, C_{41} = 1, C_{42} = 1 \\ &R_{51} = 2, R_{52} = 1, L_5 = 2, C_{51} = 1, C_{52} = 1. \end{aligned}$$

The communication topology is depicted in Fig. 2.

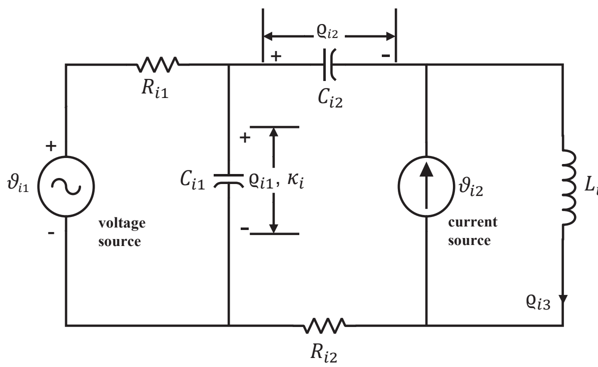

Fig. 1. The circuit of the RLC network [17]

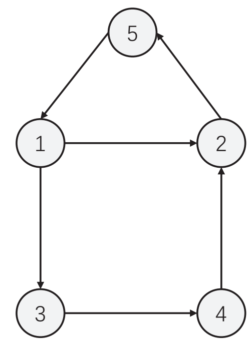

Fig. 2. The communication topology of the circuits

The cost functions of the agents are chosen as

$$f_i(\kappa_i) = (\kappa_i - \kappa_i(t_l))^2, i = 1, 2, 3, 4, 5, \tag{26}$$

where t_l denotes the instant when the data-driven learning period ends, and $\kappa_i(t_l)$ denotes the output of the i-th agent at t_l.

For Algorithm 1, the initial parameters of control laws $\vartheta_i, i = 1, 2, 3, 4, 5$ are chosen such that $\Pi_i - \Xi_i U_{i,0}$ are Hurwitz, and select the exploration noise such that rank conditions (14) (21), and (24) are satisfied.

Under Algorithm 1, the learned parameters of the control law are

$$U_1^\star = \begin{bmatrix} 0.36 & 0.32 \\ 0.36 & -0.53 \\ -0.01 & 0.16 \end{bmatrix}^{\mathrm{T}}, U_2^\star = \begin{bmatrix} 0.48 & 0.12 \\ 0.11 & -0.79 \\ 0.21 & -0.07 \end{bmatrix}^{\mathrm{T}}$$
$$U_3^\star = \begin{bmatrix} 0.41 & 0.17 \\ 0.25 & -0.66 \\ 0 & 0.08 \end{bmatrix}^{\mathrm{T}}, U_4^\star = \begin{bmatrix} 0.44 & 0.4 \\ 0.29 & -0.47 \\ 0.06 & 0.41 \end{bmatrix}^{\mathrm{T}}$$
$$U_5^\star = \begin{bmatrix} 0.44 & 0.42 \\ 0.29 & -0.57 \\ 0.12 & 0.41 \end{bmatrix}^{\mathrm{T}}.$$

For the design of the control law, we choose $\gamma_1 = 1200$ and $\gamma_2 = 55$. The optimal solution for the cost function $f(\kappa) = \sum_{i=1}^{5} f_i(\kappa_i)$ is $\kappa^\star = 1.3004$. The output of the multi-agent system under Algorithm 2 is shown in Fig. 3.

The parameters in the control law are learned during $t \in [0, 3]s$, and the control law is applied to the controlled system

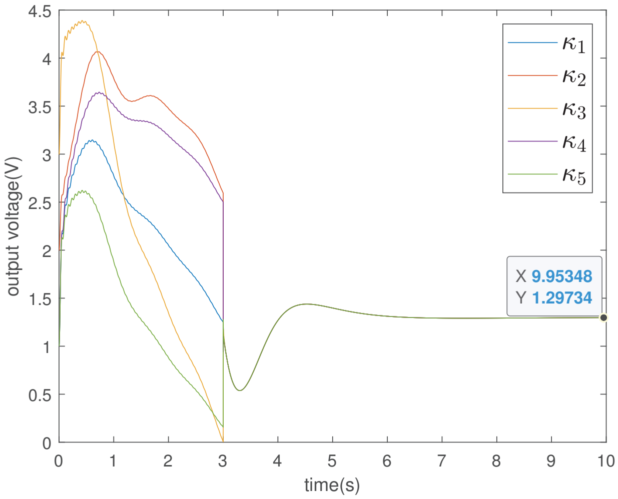

Fig. 3. Output of the multi-agent system

after $t = 3s$. It can be seen in Fig. 3 that the distributed algorithm solves the optimal output consensus problem when system dynamics are unknown.

VI. Conclusion

In this paper, we have investigated the distributed optimization problem of general linear multi-agent systems with unknown heterogeneous agent dynamics. To solve the problem, we have proposed a data-driven method based on ADP to identify the system dynamics and approximate the feedback gain in the control law by accessing the available data of input, state, and output. The remaining parameters in the control law are obtained by solving a series of matrix equations based on the identified system dynamics. Moreover, we have shown that our proposed control law is capable of solving the distributed optimization problem by regulating the outputs of all agents to the optimal solution of the global cost function.

References

[1] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[2] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2011.

[3] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3141–3155, 2018.

[4] B. Johansson, M. Rabi, and M. Johansson, "A simple peer-to-peer algorithm for distributed optimization in sensor networks," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 4705–4710.

[5] M. Ye and G. Hu, "Distributed extremum seeking for constrained networked optimization and its application to energy consumption control in smart grid," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2048–2058, 2016.

[6] H. Jaleel and J. S. Shamma, "Distributed optimization for robot networks: From real-time convex optimization to game-theoretic self-organization," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1953–1967, 2020.

[7] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[8] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[9] C. Liu, H. Li, and Y. Shi, "A unitary distributed subgradient method for multi-agent optimization with different coupling sources," *Automatica*, vol. 114, p. 108834, 2020.

[10] P. Wang, P. Lin, W. Ren, and Y. Song, "Distributed subgradient-based multiagent optimization with more general step sizes," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2295–2302, 2017.

[11] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[12] Y. Zhao, Y. Liu, G. Wen, and G. Chen, "Distributed optimization for linear multiagent systems: Edge-and node-based adaptive designs," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3602–3609, 2017.

[13] Y. Tang, Z. Deng, and Y. Hong, "Optimal output consensus of high-order multiagent systems with embedded technique," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1768–1779, 2018.

[14] Z. Li, Z. Wu, Z. Li, and Z. Ding, "Distributed optimal coordination for heterogeneous linear multiagent systems with event-triggered mechanisms," *IEEE Transactions on Automatic Control*, vol. 65, no. 4, pp. 1763–1770, 2019.

[15] J. Zhang, L. Liu, and H. Ji, "Optimal output consensus of second-order uncertain nonlinear systems on weight-unbalanced directed networks," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 8, pp. 4878–4898, 2022.

[16] Y. Tang and K. Zhu, "Optimal consensus for uncertain high-order multi-agent systems by output feedback," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 4, pp. 2084–2099, 2022.

[17] J. Zhang, L. Liu, H. Ji, and X. Wang, "Optimal output consensus of heterogeneous linear multiagent systems over weight-unbalanced directed networks," *IEEE Transactions on Cybernetics*, vol. 54, no. 2, pp. 1167–1177, 2022.

[18] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[19] W. Gao and Z.-P. Jiang, "Adaptive dynamic programming and adaptive optimal output regulation of linear systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4164–4169, 2016.

[20] C. Chen, L. Xie, K. Xie, F. L. Lewis, and S. Xie, "Adaptive optimal output tracking of continuous-time systems via output-feedback-based reinforcement learning," *Automatica*, vol. 146, p. 110581, 2022.

[21] W. Gao, C. Deng, Y. Jiang, and Z.-P. Jiang, "Resilient reinforcement learning and robust output regulation under denial-of-service attacks," *Automatica*, vol. 142, p. 110366, 2022.

[22] M. Lu and L. Liu, "Leader-following consensus of multiple uncertain euler–lagrange systems subject to communication delays and switching networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2604–2611, 2017.

[23] H. Cai, F. L. Lewis, G. Hu, and J. Huang, "The adaptive distributed observer approach to the cooperative output regulation of linear multi-agent systems," *Automatica*, vol. 75, pp. 299–305, 2017.

[24] M. Lu and L. Liu, "Cooperative output regulation of linear multi-agent systems by a novel distributed dynamic compensator," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6481–6488, 2017.

[25] ——, "Leader–following consensus of multiple uncertain euler–lagrange systems with unknown dynamic leader," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4167–4173, 2019.

[26] C. Godsil and G. F. Royle, *Algebraic graph theory*. New York: Springer Science & Business Media, 2001.

[27] Y. Zhu, W. Yu, G. Wen, and W. Ren, "Continuous-time coordination algorithm for distributed convex optimization over weight-unbalanced directed networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 7, pp. 1202–1206, 2018.

[28] D. Kleinman, "On an iterative technique for riccati equation computations," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 114–115, 1968.