2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Linear Quadratic Leader-following Consensus of Multi-agent Systems: a Decentralized Computation and Distributed Information Fusion Strategy

Yunxiao Ren, Jiachen Qian, Zhisheng Duan*, *Senior Member, IEEE,*

Abstract— **This study delves into the leader-following consensus problem in linear multi-agent systems that are structured on weight-balanced digraphs. The primary aim is to formulate a distributed, implementable optimal controller capable of achieving both leader consensus and simultaneous optimization of the linear quadratic cost function. To attain this objective, we introduce a decentralized computation approach and advocate for a distributed information strategy. Initially, the computation of the global Riccati equation is disassembled into the computation of local Riccati equations. Following that, we propose an information fusion algorithm utilizing the dynamic average consensus approach to unravel the optimal controller, enabling its implementation in a distributed fashion. Additionally, we offer numerical simulation examples to demonstrate the efficacy of our proposed approach.**

I. Introduction

Consensus control in multi-agent systems (MASs) has garnered significant interest in the past few decades. The objective of consensus control is to attain agreement on specific parameters by relying solely on local information. Research in this field has made substantial progress, including advancements in average consensus [1], leader-following consensus [2]–[4], and others.

Expanding on this prior research, some studies aim to design controllers that can simultaneously minimize an objective function and achieve consensus, a concept referred to as optimal consensus. In the context of optimal consensus, the cost function commonly adopts a linear quadratic (LQ) metric.However, effectively managing the coupling between communication and control can pose challenges in MASs. Numerous studies have been conducted to address the optimal consensus problem. For example, Wang et al. proposed a distributed optimization-based algorithm [5] to solve the finite horizon linear quadratic (LQ) synchronization problem in MASs. This work was further enhanced by introducing an accelerated algorithm in [6]. Additionally, a distributed model predictive control algorithm was developed in [7]. It is important to note that these algorithms primarily target finite horizon MASs, despite the possibility of reformulating such problems as distributed optimization algorithms.

Y. Ren, J. Qian and Z. Duan are with the State Key Laboratory for Turbulence and Complex Systems, Department of Mechanics and Engineering Science,College of Engineering, Peking University, Beijing 100871, P. R. China (email: duanzs@pku.edu.cn).

The research conducted by Y. Ren, J. Qian, and Z. Duan received support from the National Natural Science Foundation of China under Grant T2121002 and 62173006. Additionally, the work by Y. Ren was further supported by the Peking University-BHP Carbon and Climate Wei-Ming PhD Scholars project under Grant WM202206.*(Corresponding author: Zhisheng Duan.)*

Another perspective on the optimal consensus problem, framed as a graphical game, has been explored. Fuzzy adaptive dynamic programming was used to address this problem [8], and a novel mixed iteration algorithm was designed [3].Furthermore, a data-driven controller was devised to address the optimal leader-following control problem from a game-theoretic perspective [9]. The synchronization problem of MASs was approached from a graphical game perspective using reinforcement learning in [10]. Indeed, it is essential to recognize that these problems are formulated within a graphical game framework, where the objective of the agents is to attain consensus by minimizing their individual local cost functions. This approach typically results in a Nash equilibrium rather than achieving global optimization.

The distributed optimal control problem of MASs with an infinite-horizon global objective function has been extensively studied in the literature [11]–[13]. Nevertheless, the cost functions in these literatures are often complex, which can obscure their practical significance. For instance, the optimization in [11] encompasses complex topological data in cost function, and it is essential to have a coupling coefficient of significant magnitude to guarantee the positive definiteness of the weight matrix.

The approach proposed in this paper builds upon previous work [4]. However, it introduces a more radical decoupling method, making it suitable for a broader range of scenarios. In [4], an algorithm minimizing a user-friendly objective function is proposed, but the weighting tuning process requires the local weighting matrices to be the same. As a result, if one agent changes its weighting matrices, the other agents must also change their weighting matrices to the same values, limiting the flexibility of the weighting design. Additionally, the algorithm in [4] can only be utilized for MASs on an undirected graph, rendering it unsuitable for other scenarios.

Taking inspiration from the aforementioned observations, this research article presents a decentralized computation and distributed information fusion methodology to tackle the global optimal leader-following consensus problem in an infinite-time horizon. The key contributions of this study can be summarized as follows:

1) The proposed objective function and optimal controller offer a user-friendly solution for achieving leader-following consensus control. The objective function is both simple and standard, facilitating the tuning of weighting matrices to attain desired performance in multi-agent systems (MASs). In comparison to previous works such as [11]–[13], the proposed objective

function is notably more straightforward.

2) This paper introduces a sophisticated decomposition computation method that breaks down the computation of the global Riccati equation into individual agents' local Riccati equations. Importantly, this is achieved without necessitating any global information. This approach represents a more radical departure from the method proposed in [4], where the weighting matrices are global. Consequently, the method presented in this paper provides greater flexibility in weighting matrix design.
3) The paper puts forward an information fusion algorithm to disentangle the optimal controller. This algorithm guarantees that the errors resulting from decoupling converge to zero within a finite timeframe. In contrast to the algorithm outlined in [4], the proposed algorithm utilizes second-order dynamics to circumvent synchronization issues between communication and the dynamic evolution of consensus states. Furthermore, it is capable of achieving consensus on strongly connected weight-balanced digraphs.

Notation: The set $\mathbb{R}$ represents real numbers.The notation $\mathbb{R}^n$ represents a collection of real vectors, while $\mathbb{R}^{n\times n}$ denotes a collection of real matrices. I_N signifies an identity matrix with dimension N. The symbol 0 can refer to either a scalar zero or a matrix with all zero elements, depending on the context and appropriate dimensions. The symbol $\otimes$ denotes the Kronecker product. For a matrix or vector denoted by M, M^T denotes its transpose, and M^{-1} signifies matrix inverse. The pseudo-inverse of M is indicated as $M^{\dagger}$. When we state $X{>}0$ ($X{<}0$), it implies that X is either a positive-definite or a negative-definite matrix. The notations $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_\infty$ are used to represent the 1-norm, 2-norm, and ∞-norm operators, respectively. For a symmetric matrix X in $\mathbb{R}^{m\times m}$ with elements x_{ij}, $vecs(X)$ is defined as $\left[x_{11}, 2x_{12}, \ldots, 2x_{1m}, x_{22}, 2x_{23}, \ldots, 2x_{m-1,m}, x_{m,m}\right]^T$ in $\mathbb{R}^{\frac{1}{2}m(m+1)}$, and $vecs^{-1}$ is its corresponding inverse function. For a vector $v=\left[v_1, v_2, \cdots, v_N\right]^T$ in $\mathbb{R}^N$, the notation $v_{m:n}$, where $1\le m<n\le N$ and $m,n\in\mathbb{N}$, represents the subvector $\left[v_m, v_{m+1}, \cdots, v_n\right]^T$ in $\mathbb{R}^{n-m+1}$.

II. Preliminaries and problem formulation

A. Basic Graph Theory

This section provides an overview of fundamental concepts in graph theory. We represent a weighted graph $\mathcal{G}$ consisting of N nodes as $\mathcal{G}=\{\mathcal{N},\mathcal{E}\}$, where $\mathcal{N}$ is defined as the vertex set $\mathcal{N}\triangleq\{i\}_1^N$, with each $i\in\mathcal{N}$ corresponding to a node. The edge set of the graph is denoted as $\mathcal{E}\subseteq\mathcal{N}\times\mathcal{N}$. We define the adjacency matrix $E=\{a_{ij}\}$, where $a_{ii}=0$ and $a_{ij}\neq 0$ if there exists an edge between nodes i and j. The neighborhood set of node i is represented as $\mathcal{N}_i=\{j\in\mathcal{N}:(j,i)\in\mathcal{E}, i\neq j\}$. To construct the graph Laplacian matrix, we define $L=D-E$, where $D=\operatorname{diag}\{d_i\}$ is known as the in-degree matrix, and d_i represents the in-degree of node i calculated as $d_i=\sum_{j\in\mathcal{N}_i} a_{ij}$.

B. Problem Formulation

Consider a collective of N follower agents distributed across a communication network represented by the graph $\mathcal{G}$, where their dynamics are defined by the following equations [13]:

$$\dot{x}_i(t)=Ax_i(t)+Bu_i(t). \tag{1}$$

Here, $x_i(t)\in\mathbb{R}^n$ represents the state of agent i, $u_i(t)\in\mathbb{R}^{m_1}$ is the control input to be designed for agent i, where $i\in\mathcal{N}$. The system and input matrices are denoted as $A\in\mathbb{R}^{n\times n}$ and $B\in\mathbb{R}^{n\times m_1}$, respectively. The dynamics of the linear agent are described as follows:

$$\dot{x}_0(t)=Ax_0(t). \tag{2}$$

The primary objective of each follower agent is to track the state of a leader. The discrepancy between each agent and its neighboring agents, known as the neighbor error, is defined as follows (as seen in [10]):

$$\delta_i:=\sum_{j\in\mathcal{N}_i} a_{ij}\left(x_i-x_j\right)+g_i\left(x_i-x_0\right),\quad \forall i\in\mathcal{N}. \tag{3}$$

In this context, g_i denotes the pinning gain associated with agent (i). The condition $g_i\neq 0$ signifies that agent (i) has the ability to acquire state information from the leader. The neighbor error $\delta_i\in\mathbb{R}^n$. By considering equation (3), the complete neighbor error vector is expressed as follows:

$$\delta=\left(L_G\otimes I_n\right)\xi. \tag{4}$$

Here, $x=\left[x_1^T, x_2^T, \ldots x_N^T\right]^T$, $\delta=\left[\delta_1^T, \delta_2^T, \ldots, \delta_N^T\right]^T\in\mathbb{R}^{nN}$, $\underline{x}_0=\mathbf{1}_N\otimes x_0\in\mathbb{R}^{nN}$, and $\xi=(x-\underline{x}_0)$ represents the synchronization error. Additionally, $G=\operatorname{diag}\{g_1, g_2, \ldots, g_N\}$, $L_G=L+G$. The overall neighbor error dynamics is then given by:

$$\dot{\delta}(t)=(I_N\otimes A)\delta(t)+[L_G\otimes B]u(t), \tag{5}$$

where $u(t)=\left[u_1^T(t), u_2^T(t), \ldots, u_N^T(t)\right]^T$.

Assumption 1: (1) The pair (A,B) is controllable.

(2) The interaction topology of a network consisting of (n) agents is denoted by $\mathcal{G}(\mathcal{V},\mathcal{E})$, which represents a strongly connected weight-balanced digraph. It is important to note that there is at least one non-zero pinning gain g_i in this configuration.

Remark 1: Based on Assumption 1, we can conclude that L_G is nonsingular [14]. This means that $\xi=(L_G\otimes I_n)\delta$, and therefore, $\lim_{t\to\infty}\|x_i(t)-x_0(t)\|_2=0$ iff $\lim_{t\to\infty}\|\delta_i\|_2=0$, for $\forall i=1,2,\ldots,N$.

Assumption 2: (see in [4]) Each agent i is aware of the network size N, and there exists an upper bound for the network size, denoted as $\bar{N}$.

Assumption 3: (see in [4]) Agent i has the capability to retrieve the respective row from the Laplacian matrix L_G, specifically, agent i knows L_{Gi}.

Remark 2: The reasonableness of Assumption 2 is supported by the availability of numerous distributed techniques that can quickly estimate the network size (see [15] and related references). Similarly, Assumption 3 is justifiable as each agent in the network is uniquely labeled, the transmitted information to neighbors is appropriately labeled.

Assumption 4: (see in [16]) The time derivative of the neighbor error for each agent remains within certain bounds, denoted as $\|\dot{\delta}_i\|_2 \le \gamma_i$.

This paper aims to address a global linear quadratic regulator problem of the dynamics (5), which makes $\delta \to 0$. Then from the analysis of Remark 1, $\xi \to 0$, too, which solves the optimal leader-following consensus problem.

Consider the global cost function as follow:

$$J = \int_0^\infty U\left(\delta(\tau), u(\tau)\right) d\tau, \tag{6}$$

where J is the cost to minimize, and U is the one-step cost defined as

$$U(\delta(t), u(t)) = \frac{1}{2}\left(\delta^T(t) Q \delta(t) + u^T(t) R u(t)\right), \tag{7}$$

where $Q = Q^T \ge 0 \in \mathbb{R}^{nN\times nN}$, $R = R^T > 0 \in \mathbb{R}^{mN\times mN}$ to be designed, and the following assumption is hold.

Assumption 5: The pair $(A, \sqrt{Q})$ is observable.

The global LQ control problem is formulated as:

Problem 1:

$$\min_u J = \min_u \int_0^\infty U\left(\delta(\tau), u(\tau)\right) d\tau,\ s.t.,\ (5). \tag{8}$$

The problem focused by this paper can be described as follow:

Problem 2: Design the control inputs u_i of each agent which can be computed and implemented **in a distributed way** and minimize the objective function (6), subject to the neighbour error dynamics (5).

The problem formulation in the paper is completed.

III. Controller Design

In this section, the global linear quadratic objective function is minimized and the corresponding distributed optimal controller is provided. The results shows that the proposed controller can minimized the objective function (6) and makes all the follower agents achieve consensus on the leader agent.

Before providing the controller design, the weighting matrices in the global objective function are defined first. The weighting matrix of the neighbour error is defined as

$$Q = diag\{Q_1, Q_2, \dots, Q_N\}, \tag{9}$$

where $Q_i \ge 0$, $(A, \sqrt{Q_i})$ is observable for $i = 1, 2, \dots, N$. Therefore, $(I_N \otimes A, Q)$ is observable. The weighting matrix of the control input is defined as

$$R = \left(L_G \otimes I_m\right)^T \bar{R}\left(L_G \otimes I_m\right), \tag{10}$$

where $\bar{R} = diag\{R_1, R_2, \dots, R_N\}$, $R_i > 0$ for $i = 1, 2, \dots N$, and $R > 0$ is guaranteed by the nonsingularity of $L + G$.

A. Decentralized Computation

Since the weighting matrices in the global objective function is defined, the optimal controller of each agent is designed in this subsection.

It is known that solving the LQ problem needs to solving the following continuous-time algebraic Riccati equation (CARE):

$$P\hat{A} + \hat{A}^T P + Q - P\hat{B}R^{-1}B^T P = 0, \tag{11}$$

where $\hat{A} = I_N \otimes A$, $\hat{B} = L_G \otimes B$.

Then, the controller is

$$u^* = -R^{-1}\hat{B}^T P \delta. \tag{12}$$

However, solving the above equation (11) and implementing the control law (12) in a distributed way is challenging. The design idea is decomposing the equation (11) and distributing the computation to each agents. Then, using the distributed information fusion method to construct the global optimal control law. The following proposition gives the decomposition method.

Proposition 1: If Assumption 1 holds, the following controller solves the global LQ control problem (Problem 1), with weighting matrices defined as (9) and (10).

$$u_i^* = \frac{\left(\sum_{j=1}^N a_{ij} u_j^* - K_i \delta_i\right)}{\left(\sum_{j=1}^N a_{ij} + g_i\right)}, \forall i \in \mathcal{N}, \tag{13}$$

where $K_i = R_i^{-1} B^T P_i$ and $P_i = P_i^T > 0$ is the solution of the following local CARE:

$$P_i A + A^T P_i + Q_i - P_i B R_i^{-1} B^T P_i = 0. \tag{14}$$

Proof: It is known that solving the controller 12 solves Problem 1 with P is the unique positive definite solution of the global ARE (11). Therefore, if (12) and (13) is proven to be equivalent, the proof can be completed.

First, it is proved that the solution of (11) has the following form:

$$P = diag\{P_1, P_2, \dots, P_N\}, \tag{15}$$

where P_i is the solution of (14), $i = 1, 2, \dots, N$. Since $(A, \sqrt{Q_i})$ is observable and Assumption 1 holds, one has that each ARE (14) has a unique positive definite solution P_i for $\forall i = 1, 2, \dots, N$.

Substitute (9) and (10) into (11), and consider

$$\begin{aligned} \hat{B}R^{-1}\hat{B} &= (L_G \otimes B)(L_G \otimes I_m)^{-1}\bar{R}^{-1} \\ &\quad \times (L_G \otimes I_m)^{-T}(L_G \otimes B)^T \\ &= (I_N \otimes B)\bar{R}^{-1}(I_N \otimes B)^T, \end{aligned} \tag{16}$$

it can be obatained that

$$P\hat{A} + \hat{A}^T P + Q - P(I_N \otimes B)\bar{R}^{-1}(I_N \otimes B)^T P = 0. \tag{17}$$

In (17), $\hat{A}$, Q, $I_N \otimes B$ and $\bar{R}^{-1}$ are all diagonal, substitute (15) into (17) and consider (14), one can obtained equation (17) also holds. Since the solution of (17) is unique, which means $diag\{P_1, P_2, \dots, P_N\}$ is the unique solution of (17), i.e., the unique solution of (11).

Then, substitute (10) and (15) into (12), it follows that

$$u^* = -\left(L_G \otimes I_m\right)^{-1} \bar{R}(I_N \otimes B)^T P\delta. \tag{18}$$

Left multiply $(L_G \otimes I_m)$ to both side, and let $u^* = \begin{bmatrix} (u_1^*)^T & (u_2^*)^T & \ldots & (u_N^*)^T \end{bmatrix}^T$, one has

$$(L_G \otimes I_m) \begin{bmatrix} u_1^* \\ u_2^* \\ \vdots \\ u_N^* \end{bmatrix} = - \begin{bmatrix} R_1^{-1} B^T P_1 \delta_1 \\ R_2^{-1} B^T P_2 \delta_2 \\ \vdots \\ R_N^{-1} B^T P_N \delta_N \end{bmatrix}. \tag{19}$$

That means

$$\left(\sum_{j=1}^{N} a_{ij} + g_i\right) u_i^* - \sum_{j=1}^{N} a_{ij} u_j^* = -K_i \delta_i, \forall i = 1, 2, \ldots, N, \tag{20}$$

which is equivalent to (13). ■

Remark 3: The controller and the associated global objective function proposed in this work offer greater adaptability compared to those in [3]. In the previous approach, it was necessary for the local weighting matrices to be synchronized to the same value, restricting each agent's ability to independently adjust its weighting matrices. In contrast, our method allows users in this paper to make local choices, such as selecting larger Q_i for quicker convergence or larger R_i to minimize control energy consumption, providing them with greater flexibility and control over the system.

Remark 4: To compute the optimal controller (13) for each agent (i), it requires access to the optimal input information from its neighboring agents, and this will make the controller can not implement directly. In the subsequent subsection, a dynamic average consensus-based information fusion algorithm is formulated to decouple the controller.

B. Information Fusion

An information fusion algorithm based on dynamic average consensus algorithms in [1], [16], [17] is proposed to make the optimal controller (13) can be implemented distributedly. The result demonstrates that the error between decoupled controller and (13) of agent i converges within finite time.

Remark 5: Algorithm 1 is implemented in a distributed way, because the neighbour indices L_{Gi}, the local optimal control gain $K_i = R_i^{-1} B^T P_i$, the neighbour error $\delta_i = \sum_{j \in \mathcal{N}_i} a_{ij}\left(x_i - x_j\right) + g_i\left(x_i - x_0\right)$, and the consensus gains α_i, β_i are information that can be get locally. $\hat{u}_i(t)$ is the decoupled $u_i^*(t)$ in (13). The forthcoming proposition will demonstrate that the errors between $\hat{u}_i(t)$ and $u_i^*(t)$ converge to 0 within a finite time.

Remark 6: Compared to the algorithm presented by [4], Algorithm 1 is second-order, which avoids any asynchronization between the communication and dynamic process in the average consensus algorithm. Specifically, in [4], the average consensus algorithm has two main steps:

$$X_i(t) = \sum_{j=1}^{N} a_{ij}\left(v_i(t) - v_j(t)\right) + \psi_i(t), \tag{24}$$

Algorithm 1 Distributed Information Fusion Algorithm

Initialization: Set $v(0) = [v_1(0), v_2(0), \cdots, v_N(0)]^T \in \mathbb{R}^{(mN+3)mN^2/2}$, $w(0) = [w_1(0), w_2(0), \cdots, w_N(0)]^T \in \mathbb{R}^{(mN+3)mN^2/2}$, running time t_{max}, $X(0) = [X_1(0), X_2(0), \cdots, X_N(0)]^T \in \mathbb{R}^{(mN+3)mN^2/2}$, and the local gain $\alpha_i \geq 1 + \gamma_i \|K_i\|_2 \frac{\bar{N}^{\frac{5}{2}}}{4}$. , $\beta_i > 2\bar{N}\alpha_i$

Implement:

while $0 \leq t \leq t_{max}$ **do**
  for $i = 1$ *to* N **do**
    Collect L_i, G_i and $\delta_i(t)$.
    Get $T_i = L_{Gi}^T L_{Gi} \otimes I_m \in \mathbb{R}^{mN \times mN}$.
    Get $I_i(t) = -[L_{Gi}^T \otimes I_m] K_i \delta_i(t) \in \mathbb{R}^{mN}$.
    Get $\psi_i(t) = \begin{bmatrix} vecs(T_i) \\ I_i(t) \end{bmatrix} \in \mathbb{R}^{\bar{m}}$, $\bar{m} = (mN+3)mN/2$.
    Run:

$$\dot{v}_i(t) = -\beta_i \operatorname{sgn}\left\{v_i(t) - \sum_{j=1}^{N} a_{ij}\left(w_i(t) - w_j(t)\right)\right\}. \tag{21}$$

    Run:

$$\dot{w}_i(t) = -\alpha_i \operatorname{sgn}\left\{\sum_{j=1}^{N} a_{ij}\left(X_i(t) - X_j(t)\right)\right\}. \tag{22}$$

    Run:

$$X_i(t) = v_i(t) + \psi_i(t). \tag{23}$$

    $\hat{T}_i = ves^{-1}\left([X_i(t)]_{1:(mN+1)mN/2}\right)$.
    $\hat{I}_i(t) = [X_i(t)]_{1+(mN+1)mN/2:\bar{m}}$.
    Output: $\hat{u}_i(t) = [\hat{T}_i^{-1} \hat{I}_i(t)]_{(i-1)m+1:im}$.
  end for
end while

$$\dot{v}_i(t) = -\alpha \operatorname{sgn}\left\{\sum_{j=1}^{N} a_{ij}\left(X_i(t) - X_j(t)\right)\right\}. \tag{25}$$

By substituting (24) into (25), the v_i values need to be communicated twice over the graph to complete one step update. However, the proposed algorithm in this paper does not encounter such a problem.

Theorem 1: If Assumptions 1- 4 hold, $\hat{u}_i(t)$ output by Algorithm 1 converges to u_i^* in (13) for $\forall t \geq t^* = \|\tilde{X}(0)\|_2 / \lambda_2(L)$, i.e.,

$$\hat{u}_i(t) = [\hat{T}_i^{-1} \hat{I}_i(t)]_{(i-1)m+1:im} = u_i^*, \quad i = 1, 2, \ldots, N, \tag{26}$$

for $\forall t \geq t^*$.

Proof: First, it will show that decoupling the controller (13) can be achieved by dynamic average consensus. Then, if the consensus error converges in the prescribed time, the proposition is proven.

Form (19), finding an distributedly implementable u_i^* is equivalent to solving a linear time-varying matrix equation distributedly. The local information are L_{Gi} and $R_i B^T P_i \delta_i$, and the agent i needs to using these information to solving the equation (19) and get u_i^*. The solution of (19) can be written as follow:

$$\begin{bmatrix} u_1^* \\ u_2^* \\ \vdots \\ u_N^* \end{bmatrix} = -\left((L_G \otimes I_m)^T (L_G \otimes I_m) \right)^{-1} \times (L_G \otimes I_m) \begin{bmatrix} R_1^{-1} B^T P_1 \delta_1 \\ R_2^{-1} B^T P_2 \delta_2 \\ \vdots \\ R_N^{-1} B^T P_N \delta_N \end{bmatrix}. \tag{27}$$

Also, one has the following equations

$$(L_G \otimes I_m)^T (L_G \otimes I_m) = \sum_{i=1}^{N} L_{Gi}^T L_{Gi}, \tag{28}$$

$$(L_G \otimes I_m) \begin{bmatrix} R_1^{-1} B^T P_1 \delta_1 \\ R_2^{-1} B^T P_2 \delta_2 \\ \vdots \\ R_N^{-1} B^T P_N \delta_N \end{bmatrix} = \sum_{i=1}^{N} [L_{Gi}^T \otimes I_m] K_i \delta_i. \tag{29}$$

Therefore, if each agent i can obtain the following average value $\hat{T} = \frac{1}{N} \sum_{i=1}^{N} L_{Gi}^T L_{Gi}$ and $\hat{I} = \frac{1}{N} \sum_{i=1}^{N} [L_{Gi}^T \otimes I_m] K_i \delta_i$, the u_i^* can be get from $\hat{T}^{-1} \hat{I}$.

Using the results from the dynamic average consensus algorithms presented in [1], [16], [17], if Assumption 1-4 hold, the T_i and I_i can converge to $\hat{T}$ and $\hat{I}$ in finite time, respectively. That is to say the controller is decoupled in finite time. ∎

Remark 7: It is worth noting that the algorithms presented in [1], [16] are only applicable to undirected communication graphs. However, by making a slight modification to the previous algorithm and invoking the approach presented in Section V of [17], it is possible to achieve average consensus error convergence to zero in finite time for strongly connected weight-balanced digraphs. The convergence can be proven using the techniques outlined in literature [1], [16], [17].

IV. Simulation

In this part, we provide a leader-follower consensus example. We set a multi-agent systems comprising 4 follower agents and 1 leader agent. The communication topology is depicted in Figure 1. The Laplacian matrix L and pinning gain matrix G are defined as follows:

$$L = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}, G = diag(1, 0, 0, 0). \tag{30}$$

The agent system dynamics is set as

$$A = \begin{bmatrix} 0 & 0.8 \\ -0.5 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \tag{31}$$

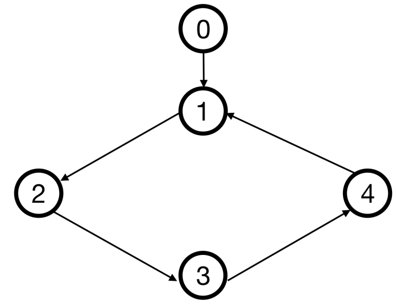

Fig. 1. Communication graph of the multi-agent systems

and the initial conditions of the leader agents and each follower are set as

$$x_0(0) = \begin{bmatrix} 9 \\ 6 \end{bmatrix}, x_1(0) = \begin{bmatrix} 7.4289 \\ 3.7812 \end{bmatrix}, x_2(0) = \begin{bmatrix} 10.2021 \\ 6.9111 \end{bmatrix},$$

$$x_3(0) = \begin{bmatrix} 12.3162 \\ 11.3451 \end{bmatrix}, x_4(0) = \begin{bmatrix} 14.6904 \\ 4.8045 \end{bmatrix}.$$

The weighting matrices of each agents are given in TABLE I

TABLE I
Weighting Matrices

Agent i	1	2	3	4
Q_i	I_2	$3I_2$	$2I_2$	$0.8I_2$
R_i	1	0.5	0.5	2

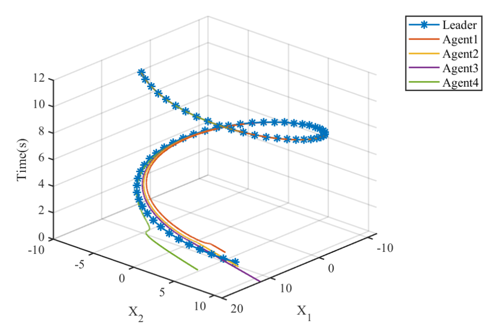

Fig. 2. Phase trajectories of the leader and followers

In Fig. 2, one can observe the 3D phase trajectories of the agents, while Fig. 3 showcases the decoupled errors of the optimal control inputs, which exhibit rapid convergence to zero. The progression of neighbor errors is depicted in Fig. 4, providing confirmation that the follower agents effectively track the leader agents.

V. Conclusion

The optimal leader-following consensus problem of multi-agent systems with linear dynamics is considered in this paper. To solve this problem, a decentralized computation and distributed information fusion strategy is proposed. Firstly, the computation of the global CARE is decomposed into N local CAREs by designing the weighting matrices to make

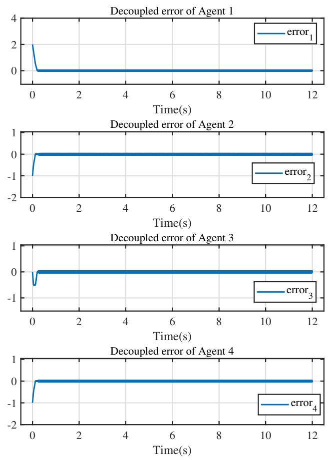

Fig. 3. The decoupling errors of each controllers

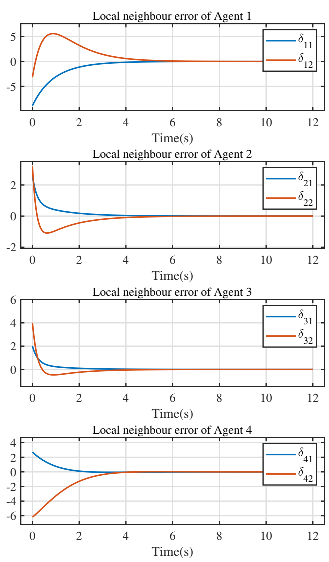

Fig. 4. The neighbour errors of the agents

the structure of the global CARE diagonal, allowing each agent to compute its control gain locally. Then, an information fusion algorithm is provided to decouple the global optimal controller, with the error between the decoupled controller and the exact optimal controller converging to zero whinin a finite time. The effectiveness of the proposed method is validated through simulation examples. Future research efforts could focus on addressing the decoupling of the LQ control problem in multi-agent systems with nonlinear dynamics. Additionally, there is potential for exploring methods to reduce both computation and communication costs associated with these systems.

References

[1] J. George and T. Yang, "Fast distributed least-squares solver for linear time-varying equations," in *2019 American Control Conference (ACC)*, pp. 4092–4097, 2019.

[2] G. Wen and W. X. Zheng, "On constructing multiple lyapunov functions for tracking control of multiple agents with switching topologies," *IEEE Transactions on Automatic Control*, vol. 64, no. 9, pp. 3796–3803, 2019.

[3] Y. Ren, Q. Wang, and Z. Duan, "Optimal leader-following consensus control of multi-agent systems: A neural network based graphical game approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3590–3601, 2022.

[4] Y. Ren, Q. Wang, and Z. Duan, "Optimal distributed leader-following consensus of linear multi-agent systems: A dynamic average consensus-based approach," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1208–1212, 2022.

[5] Q. Wang, Z. Duan, J. Wang, and G. Chen, "Lq synchronization of discrete-time multiagent systems: A distributed optimization approach," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 5183–5190, 2019.

[6] Q. Wang, Z. Duan, and J. Wang, "Distributed optimal consensus control algorithm for continuous-time multi-agent systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 102–106, 2020.

[7] Q. Wang, Z. Duan, Y. Lv, Q. Wang, and G. Chen, "Linear quadratic optimal consensus of discrete-time multi-agent systems with optimal steady state: A distributed model predictive control approach," *Automatica*, vol. 127, p. 109505, 2021.

[8] H. Zhang, J. Zhang, G. Yang, and Y. Luo, "Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 152–163, 2015.

[9] J. Zhang, Z. Wang, and H. Zhang, "Data-based optimal control of multiagent systems: A reinforcement learning design approach," *IEEE Transactions on Cybernetics*, vol. 49, no. 12, pp. 4441–4449, 2019.

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

[11] K. H. Movric and F. L. Lewis, "Cooperative optimal control for multi-agent systems on directed graph topologies," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 769–774, 2014.

[12] H. Zhang, T. Feng, G. Yang, and H. Liang, "Distributed cooperative optimal control for multiagent systems on directed graphs: An inverse optimal approach," *IEEE Transactions on Cybernetics*, vol. 45, no. 7, pp. 1315–1326, 2015.

[13] Z. Zhang, W. Yan, and H. Li, "Distributed optimal control for linear multiagent systems on general digraphs," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 322–328, 2021.

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

[15] D. Lee, S. Lee, T. Kim, and H. Shim, "Distributed algorithm for the network size estimation: Blended dynamics approach," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4577–4582, 2018.

[16] F. Chen, Y. Cao, and W. Ren, "Distributed average tracking of multiple time-varying reference signals with bounded derivatives," *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3169–3174, 2012.

[17] J. George and R. A. Freeman, "Robust dynamic average consensus algorithms," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4615–4622, 2019.