A Distributed Algorithm for Solving A Time-Varying Linear Equation

Xiaozhen Zhang[1], Qingkai Yang[1], Haijiao Wei[2], Wei Chen[3], Zhihong Peng[1], and Hao Fang[1]

Abstract— **This paper studies the problem of cooperatively solving a time-varying linear equation of the form $\mathbf{A}(t)\boldsymbol{x}(t) = \boldsymbol{b}(t)$, which always has a unique solution. Each agent has access to only some rows of the time-varying augmented matrix $[\mathbf{A}(t) \quad \boldsymbol{b}(t)]$. We propose a distributed algorithm for solving the time-varying linear equation. The proposed distributed algorithm enforces local solutions to track local time-varying manifolds corresponding to local linear sub-equations while simultaneously reaching a consensus. This enables all local solutions to converge to the solution of the original time-varying linear equation. Finally, the effectiveness of the proposed algorithm is demonstrated through its application in a cooperative monitoring task.**

I. INTRODUCTION

Linear equations are essential tools in biochemical science and engineering technology, particularly in mass balance of chemical reactions [1], least squares identification [2], and equilibrium in electricity market [3]. Traditionally, linear equations are generally solved in a centralized way. However, in networked problems, it is often necessary to divide the linear equation into multiple linear sub-equations using distributed algorithms, such as distributed estimation [4], and distributed solving of the Sylvester equation [5].

For the linear equation that has a unique solution, projection-consensus-based distributed algorithms can obtain the unique solution. In [6]-[7], the authors design consensus algorithms to enforce each agent's local solution to move along the tangent space of the corresponding local linear sub-equation, so that all local solutions can always satisfy their local linear sub-equations while reaching the agreement. In [8]-[10], tracking control terms are additionally added in the consensus-based algorithms to track the local manifold corresponding to the local linear sub-equation. This allows agents to have arbitrarily initial local solutions. [11] extends the result to finite-time convergence. Switching [12] and random [13] networks are also studied. [14] transforms the problem of distributed solving linear equation into the distributed optimization problem, which is solved by using the distributed fixed point method. More complex linear equation partitioning is considered in [15], where agents can only obtain the partial columns of the linear sub-equations.

[1]Xiaozhen Zhang, Qingkai Yang, Zhihong Peng, and Hao Fang are with the National Key Laboratory of Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China. {jiaozhen, qingkai.yang, peng, fangh}@bit.edu.cn

[2]Haijiao Wei is with the China North Vehicle Research Institute, Beijing 100072, China. weihaijiao1@126.com,

[3]Wei Chen is with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing, China. weichen1018@163.com

The underdetermined linear equation refers to a linear equation that has multiple solutions. The algorithms proposed in [8]-[10] are also feasible, but the obtained solution will be affected by many factors, such as the initializations and communication topologies [10]. Therefore, the minimum-norm solution is preferred because of its uniqueness. In [16], the authors propose a distributed algorithm for obtaining the specific solution with minimum l_2-norm to a prescribed point. Additionally, the distributed algorithm to obtain the minimum l_1-norm solution is studied in [17].

The overdetermined linear equation refers to a linear equation with redundant row conditions [19]. The overdetermined linear equation is usually unsolvable, but its least squares solution always exists. Discrete-time [19]-[20] and continuous-time [21]-[22] distributed algorithms for solving least squares have been proposed. In [23], a distributed form of Arrow-Hurwicz-Uzawa flow is proposed from the classical centralized form. Additionally, the least squares solution of the linear matrix equation is studied in [24].

Although the distributed solution of the time-invariant linear equation $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ has been widely studied in the past several years, there is a lack of research on the distributed solution of the time-varying linear equation $\mathbf{A}(t)\boldsymbol{x}(t) = \boldsymbol{b}(t)$. In [8], the authors discuss the case where $\boldsymbol{b}(t)$ is time-varying, and claim that the algorithm proposed in [8] will obtain an approximate consensus solution of $\mathbf{A}\boldsymbol{x}(t) = \boldsymbol{b}(t)$. However, the quality of the approximation linked to the rate of variation of $\boldsymbol{b}(t)$, and the authors do not provide a formal analysis or consider the case where $\mathbf{A}(t)$ is also time-varying.

In this paper, we propose a distributed algorithm for solving a time-varying linear equation of the form $\mathbf{A}(t)\boldsymbol{x}(t) = \boldsymbol{b}(t)$, which always has a unique solution. The proposed algorithm can obtain the precise solution of the time-varying linear equation and is insensitive to the initializations of local solutions. The proposed algorithm enforces local solutions to track the local manifolds corresponding to their local sub-equations and eventually to reach a consensus, i.e., converging to the solution of the original time-varying linear equation.

The remainder of this paper is organized as follows. First, some preliminaries are given in Section II. The problem of cooperative solving the time-varying linear equation is formally defined in Section III. Section IV describes the proposed main distributed algorithm and proves its stability. Finally, we demonstrate the effectiveness of the proposed algorithm through an application of cooperative monitoring tasks in Section V.

II. PRELIMINARIES

This section presents some notations and preliminary results that will be used in this paper.

A. Notions

$\otimes$ denotes the Kronecker product. $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. $\boldsymbol{1}_n$ and $\boldsymbol{0}_n$ denote the all-one and all-zero vector $\begin{bmatrix}1 & ... & 1\end{bmatrix} \in \mathbb{R}^n$ and $\begin{bmatrix}0 & ... & 0\end{bmatrix} \in \mathbb{R}^n$, respectively. $\|\boldsymbol{y}\|_1$ denotes the l_1-norm of vector $\boldsymbol{y}$. $\|\mathbf{F}\|_\infty$ denotes the ∞-norm of the matrix $\mathbf{F}$. Let $diag\left(c_1, ..., c_n\right)$ be the diagonal matrix with the elements in the main diagonal being c_i. $diag\left(\mathbf{D}_1, ..., \mathbf{D}_n\right)$ represents the block-diagonal matrix with each diagonal block $\mathbf{D}_i$. Let $\operatorname{sgn}(\cdot)$ denote the signum function. For a vector $\boldsymbol{z} \in \mathbb{R}^n$, $\operatorname{sgn}(\boldsymbol{z}) = \begin{bmatrix}\operatorname{sgn}(z_1) & \cdots & \operatorname{sgn}(z_n)\end{bmatrix}^{\mathrm{T}}$.

B. Graph Theory

A graph $\boldsymbol{\mathcal{G}}$ is a pair $(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$, where $\boldsymbol{\mathcal{V}} = \{v_1, ..., v_n\}$ is a non-empty finite set of nodes and $\boldsymbol{\mathcal{E}} \subseteq \boldsymbol{\mathcal{V}} \times \boldsymbol{\mathcal{V}}$ is a set of ordered pairs of nodes, called edges. An edge (v_j, v_i) represents the communication path from node v_j to node v_i.

The adjacency matrix is $\boldsymbol{\mathcal{A}} = [a_{ij}] \in \mathbb{R}^{n\times n}$, where $a_{ij} = 1$, if $(v_j, v_i) \in \boldsymbol{\mathcal{E}}$, and $a_{ij} = 0$, if $(v_j, v_i) \notin \boldsymbol{\mathcal{E}}$. In general, $a_{ii} = 0$ is defined. The set $\boldsymbol{\mathcal{N}}_i = \left\{j \in \boldsymbol{\mathcal{V}} : \ (v_j, v_i) \in \boldsymbol{\mathcal{E}}\right\}$ contains all neighbors of node v_i.

The Laplacian matrix is $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n\times n}$ is defined as

$$[\boldsymbol{\mathcal{L}}]_{ij} = \begin{cases} 0 & (v_j, v_i) \notin \boldsymbol{\mathcal{E}} \\ -a_{ij} & (v_j, v_i) \in \boldsymbol{\mathcal{E}} \\ \sum_{j\in\boldsymbol{\mathcal{N}}_i} a_{ij} & i = j \end{cases}$$

where $\boldsymbol{\mathcal{N}}_i = \{j \in \boldsymbol{\mathcal{V}} : \ (v_j, v_i) \in \boldsymbol{\mathcal{E}}\}$ is a set containing all neighbors of node v_i.

III. PROBLEM STATEMENT

The aim of this study is to investigate the problem of solving a time-varying linear equation in a distributed manner. We consider a group of N agents consisting of the node set $\boldsymbol{\mathcal{V}}$. Agents have distributed network communication under the connected undirected graph $\boldsymbol{\mathcal{G}}$, whose corresponding Laplacian matrix is denoted by $\boldsymbol{\mathcal{L}}$.

The group of N agents cooperatively solve the time-varying linear equation $[\mathbf{A}(t) \quad \boldsymbol{b}(t)]$, such that

$$\mathbf{A}(t)\boldsymbol{x}(t) = \boldsymbol{b}(t), \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$; $\mathbf{A}(t) \in \mathbb{R}^{m\times n}$ is a time-varying matrix; $\boldsymbol{b}(t) \in \mathbb{R}^m$ is a time-varying vector. It follows

$$\mathbf{A}(t) = \begin{bmatrix}\mathbf{A}_1(t) \\ \vdots \\ \mathbf{A}_N(t)\end{bmatrix} \quad \boldsymbol{b}(t) = \begin{bmatrix}\boldsymbol{b}_1(t) \\ \vdots \\ \boldsymbol{b}_N(t)\end{bmatrix} \quad \begin{matrix}\mathbf{A}_i(t) \in \mathbb{R}^{m_i\times n} \\ \boldsymbol{b}_i(t) \in \mathbb{R}^{m_i}\end{matrix}$$

where $\sum_{i=1}^N m_i = m$, $m_i \geq 1$.

The agent i only knows m_i rows of the augmented matrix $[\mathbf{A}(t) \quad \boldsymbol{b}(t)]$, holding a local linear sub-equation, such that

$$\mathbf{A}_i(t)\boldsymbol{x}(t) = \boldsymbol{b}_i(t). \tag{2}$$

Let the agent i also governs a local solution $\boldsymbol{x}_i(t)$, whose update law is to be designed in the next section.

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{u}_i(t). \tag{3}$$

The primary objective of this study is to find the solution of the time-varying linear equation (1) by the group N agents in a distributed manner, such that

$$\lim_{t\to\infty} \mathbf{A}(t)\boldsymbol{x}_i(t) = \boldsymbol{b}(t), \quad \forall i \in \boldsymbol{\mathcal{V}}.$$

We have the following assumptions.

Assumption 1: The time-varying linear equation (1) always has a unique solution $\boldsymbol{x}^*(t)$, which implies that $m \geq n$ and the following condition always holds

$$Rank\left(\begin{bmatrix}\mathbf{A}(t) & \boldsymbol{b}(t)\end{bmatrix}\right) = Rank(\mathbf{A}(t)) = n.$$

Moreover, the derivative of $\boldsymbol{x}^*(t)$ is always bounded by

$$||\dot{\boldsymbol{x}}^*(t)||_2 < \delta,$$

where $\delta > 0$.

Assumption 2: The derivative of local linear sub-equation (2) is always bounded by

$$\left\|\mathbf{A}_i(t)^{\mathrm{T}}\left(\mathbf{A}_i(t)\mathbf{A}_i^{\mathrm{T}}(t)\right)^{-1}\dot{\mathbf{A}}_i(t)\right\|_\infty < \sigma_i, \quad \forall\ i \in \boldsymbol{\mathcal{V}},$$

where $\sigma_i > 0$.

Note that we will omit argument t when it is clear that $\mathbf{A}(t)$, $\mathbf{A}_i(t)$, $\boldsymbol{b}(t)$, $\boldsymbol{b}_i(t)$, $\boldsymbol{x}_i(t)$, and $\boldsymbol{x}^*(t)$ are time-varying.

IV. MAIN ALGORITHM

Let $\boldsymbol{\mathcal{M}}_i = \{\boldsymbol{y} \in \mathbb{R}^n : \mathbf{A}_i\boldsymbol{y} = \boldsymbol{b}_i\}$ denotes the time-varying manifold of the linear sub-equation.

To cooperatively solve the time-varying linear equation, we introduce the following sub-objectives of local solutions:

1) For each agent, the local solution $\boldsymbol{x}_i$ tracks its time-varying manifold $\boldsymbol{\mathcal{M}}_i$.
2) All local solutions $\{\boldsymbol{x}_i\}_{i=1}^N$ reach a consensus.

The first sub-objective ensures that each local solution satisfies the time-varying local linear sub-equation. The second sub-objective ensures that all agents obtain the same local solution to the original linear equation. If local solutions $\{\boldsymbol{x}_i\}_{i=1}^n$ satisfy the above two sub-objectives simultaneously, we can conclude that every local solution satisfies all of time-varying local linear sub-equations, i.e. every local solution is the solution of the time-varying original linear equation (1). To achieve this goal, we propose the following distributed algorithm for solving the original time-varying linear equation (1).

$$\boldsymbol{u}_i = \boldsymbol{g}_i + \boldsymbol{f}_i, \tag{4}$$

$$\begin{aligned}\boldsymbol{f}_i = &- k_i\mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\left(\mathbf{A}_i\boldsymbol{x}_i - \boldsymbol{b}_i\right), \\ &- \mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\left(\dot{\mathbf{A}}_i\boldsymbol{x}_i - \dot{\boldsymbol{b}}_i\right),\end{aligned} \tag{5}$$

$$\boldsymbol{g}_i = -\gamma\mathbf{P}_i\operatorname{sgn}\left[\mathbf{P}_i\sum_{j\in\boldsymbol{\mathcal{N}}_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\right], \tag{6}$$

with $k_i > \sigma_i$, and $\gamma > \delta$, where $\mathbf{P}_i \in \mathbb{R}^{n\times n}$ denotes the orthogonal projection matrix to the kernel of $\mathbf{A}_i$, such that

$$\mathbf{P}_i = \mathbf{I}_n - \mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\mathbf{A}_i.$$

If we consider a non-zero vector $\boldsymbol{z} \in \mathbb{R}^n$, its projection onto the tangent space of the manifold $\boldsymbol{\mathcal{M}}_i$ can be denoted by $\mathbf{P}_i\boldsymbol{z}$.

Now we have the following theorems.

Theorem 1: The algorithm (4) can enforce the local solution $\boldsymbol{x}_i$ with the dynamics (3) to track the time-varying manifold $\boldsymbol{\mathcal{M}}_i(t)$, such that

$$\lim_{t\to\infty}\mathbf{A}_i\boldsymbol{x}_i = \boldsymbol{b}_i.$$

Proof: The tracking error of the time-varying manifold is defined as $\boldsymbol{\varepsilon}_i = \mathbf{A}_i\boldsymbol{x}_i - \boldsymbol{b}_i$. We consider a Lyapunov function

$$V_{i\varepsilon} = \frac{1}{2}\boldsymbol{\varepsilon}_i^{\mathrm{T}}\boldsymbol{\varepsilon}_i,$$

whose derivative is

$$\begin{aligned}\dot{V}_{i\varepsilon} =& \boldsymbol{\varepsilon}_i^{\mathrm{T}}\dot{\boldsymbol{\varepsilon}}_i\\ =& \boldsymbol{\varepsilon}_i^{\mathrm{T}}\left(\dot{\mathbf{A}}_i\boldsymbol{x}_i + \mathbf{A}_i\boldsymbol{u}_i - \dot{\boldsymbol{b}}_i\right).\end{aligned}$$

Note that $\boldsymbol{g}_i$ is orthogonal to $\mathbf{A}_i$. It gives

$$\mathbf{A}_i\boldsymbol{g}_i = \boldsymbol{0}_{m_i}.$$

Thereby, we have

$$\dot{V}_{i\varepsilon} = -k_i\left\|\boldsymbol{\varepsilon}_i\right\|_2^2.$$

Then, it follows $\boldsymbol{\varepsilon}_i \to \boldsymbol{0}$, $t\to\infty$, which denotes that $\boldsymbol{x}_i$ will lie in $\boldsymbol{\mathcal{M}}_i(t)$, such that $\mathbf{A}_i\boldsymbol{x}_i \to \boldsymbol{b}_i$, $t\to\infty$. ■

Theorem 2: Under Assumption 1 and 2, the algorithm (4) can enforce $\boldsymbol{x}_i$ with the dynamics (3) to reach a consensus, such that

$$\lim_{t\to\infty}\boldsymbol{x}_i - \boldsymbol{x}_j = \boldsymbol{0}_n, \quad \forall i,j\in\boldsymbol{\mathcal{V}}.$$

Proof: The solution error is defined as $\boldsymbol{e}_i = \boldsymbol{x}_i - \boldsymbol{x}^*$. We known that $\boldsymbol{x}^*$ also satisfies each local linear sub-equation, such that $\mathbf{A}\boldsymbol{x}^* = \boldsymbol{b}_i$. It follows that

$$\dot{\boldsymbol{A}}_i\boldsymbol{x}^* + \mathbf{A}_i\dot{\boldsymbol{x}}^* = \dot{\boldsymbol{b}}_i$$

Then, it is considered that

$$\begin{aligned}\boldsymbol{f}_i =& -k_i\mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\left(\mathbf{A}_i\boldsymbol{x}_i - \boldsymbol{b}_i\right)\\ &- \mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\left(\dot{\mathbf{A}}_i\boldsymbol{x}_i - \dot{\boldsymbol{b}}_i\right)\\ =& -k_i\mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\left(\mathbf{A}_i\boldsymbol{x}_i - \mathbf{A}_i\boldsymbol{x}^*\right)\\ &- \mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\left(\dot{\mathbf{A}}_i\boldsymbol{x}_i - \dot{\mathbf{A}}_i\boldsymbol{x}^* - \mathbf{A}_i\dot{\boldsymbol{x}}^*\right)\\ =& -k_i\left(\mathbf{I}_n - \mathbf{P}_i\right)\boldsymbol{e}_i - \mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\dot{\mathbf{A}}_i\boldsymbol{e}_i\\ &+ \left(\mathbf{I}_n - \mathbf{P}_i\right)\dot{\boldsymbol{x}}^*\\ =& -k_i\left(\mathbf{I}_n - \mathbf{P}_i\right)\boldsymbol{e}_i - \left(\mathbf{I}_n - \mathbf{P}_i\right)\mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\dot{\mathbf{A}}_i\boldsymbol{e}_i\\ &+ \left(\mathbf{I}_n - \mathbf{P}_i\right)\dot{\boldsymbol{x}}^*,\end{aligned}$$

where the following condition is used:

$$\left(\mathbf{I}_n - \mathbf{P}_i\right)\mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\dot{\mathbf{A}}_i = \mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\dot{\mathbf{A}}_i.$$

It is also considered that

$$\begin{aligned}\boldsymbol{g}_i =& -\gamma\mathbf{P}_i\mathrm{sgn}\left[\mathbf{P}_i\sum_{j\in\boldsymbol{\mathcal{N}}_i}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\right]\\ =& -\gamma\mathbf{P}_i\mathrm{sgn}\left\{\mathbf{P}_i\sum_{j\in\boldsymbol{\mathcal{N}}_i}\left[\left(\boldsymbol{x}_i - \boldsymbol{x}^*\right) - \left(\boldsymbol{x}_j - \boldsymbol{x}^*\right)\right]\right\}\\ =& -\gamma\mathbf{P}_i\mathrm{sgn}\left[\mathbf{P}_i\sum_{j\in\boldsymbol{\mathcal{N}}_i}\left(\boldsymbol{e}_i - \boldsymbol{e}_j\right)\right].\end{aligned}$$

The dynamics of $\boldsymbol{e}_i$ obtained by

$$\begin{aligned}\dot{\boldsymbol{e}}_i =& \dot{\boldsymbol{x}}_i - \dot{\boldsymbol{x}}^*\\ =& \boldsymbol{u}_i - \dot{\boldsymbol{x}}^*\\ =& -k_i\left(\mathbf{I}_n - \mathbf{P}_i\right)\boldsymbol{e}_i - \left(\mathbf{I}_n - \mathbf{P}_i\right)\mathbf{A}_i^{\mathrm{T}}\left(\mathbf{A}_i\mathbf{A}_i^{\mathrm{T}}\right)^{-1}\dot{\mathbf{A}}_i\boldsymbol{e}_i\\ &- \mathbf{P}_i\dot{\boldsymbol{x}}^* - \gamma\mathbf{P}_i\mathrm{sgn}\left[\mathbf{P}_i\sum_{j\in\boldsymbol{\mathcal{N}}_i}\left(\boldsymbol{e}_i - \boldsymbol{e}_j\right)\right],\end{aligned}$$

which can be rewritten into the matrix form, such that

$$\begin{aligned}\dot{\boldsymbol{e}} =& -\mathbf{K}\left(\mathbf{I}_{nN} - \mathbf{P}\right)\boldsymbol{e} - \left(\mathbf{I}_{nN} - \mathbf{P}\right)\mathbf{G}\boldsymbol{e}\\ &- \mathbf{P}\dot{\boldsymbol{\xi}} - \gamma\mathbf{P}\mathrm{sgn}\left(\mathbf{P}\bar{\boldsymbol{\mathcal{L}}}\boldsymbol{e}\right),\end{aligned}$$

where $\bar{\boldsymbol{\mathcal{L}}} = \boldsymbol{\mathcal{L}}\otimes\mathbf{I}_n$, and

$$\begin{aligned}\boldsymbol{e} =& \begin{bmatrix}\boldsymbol{e}_1^{\mathrm{T}} & \cdots & \boldsymbol{e}_N^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\ \dot{\boldsymbol{\xi}} =& \begin{bmatrix}\left(\dot{\boldsymbol{x}}^*\right)^{\mathrm{T}} & \cdots & \left(\dot{\boldsymbol{x}}^*\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\ \mathbf{K} =& diag\left(k_1, k_2, ..., k_N\right)\otimes\mathbf{I}_n,\\ \mathbf{P} =& diag\left(\mathbf{P}_1, \cdots, \mathbf{P}_N\right),\\ \mathbf{G} =& diag\left(\mathbf{A}_1^{\mathrm{T}}\left(\mathbf{A}_1\mathbf{A}_1^{\mathrm{T}}\right)^{-1}\dot{\mathbf{A}}_1, \cdots, \mathbf{A}_N^{\mathrm{T}}\left(\mathbf{A}_N\mathbf{A}_N^{\mathrm{T}}\right)^{-1}\dot{\mathbf{A}}_N\right).\end{aligned}$$

Now, we consider a Lyapunov function

$$V_e = \frac{1}{2}\boldsymbol{e}^{\mathrm{T}}\bar{\boldsymbol{\mathcal{L}}}\boldsymbol{e},$$

which satisfies $V_e > 0$, except the equilibrium point $\boldsymbol{e}_i - \boldsymbol{e}_j = 0$, $\forall i,j\in\boldsymbol{\mathcal{V}}$. The derivative of V_e is

$$\begin{aligned}\dot{V}_e =& \boldsymbol{e}^{\mathrm{T}}\bar{\boldsymbol{\mathcal{L}}}\dot{\boldsymbol{e}}\\ =& -\mathbf{K}\boldsymbol{e}^{\mathrm{T}}\bar{\boldsymbol{\mathcal{L}}}\left(\mathbf{I}_{nN} - \mathbf{P}\right)\boldsymbol{e} - \boldsymbol{e}^{\mathrm{T}}\bar{\boldsymbol{\mathcal{L}}}\left(\mathbf{I}_{nN} - \mathbf{P}\right)\mathbf{G}\boldsymbol{e}\\ &- \boldsymbol{e}^{\mathrm{T}}\bar{\boldsymbol{\mathcal{L}}}\mathbf{P}\dot{\boldsymbol{\xi}} - \gamma\boldsymbol{e}^{\mathrm{T}}\bar{\boldsymbol{\mathcal{L}}}\mathbf{P}\mathrm{sgn}\left(\mathbf{P}\bar{\boldsymbol{\mathcal{L}}}\boldsymbol{e}\right)\\ \leq& -\boldsymbol{e}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{G}\right)\left(\mathbf{I}_{nN} - \mathbf{P}\right)\bar{\boldsymbol{\mathcal{L}}}\boldsymbol{e}\\ &- \left(\gamma - \left\|\dot{\boldsymbol{x}}^*\right\|_1\right)\sum_{i=1}^{n}\left\|\mathbf{P}_i\sum_{j\in\boldsymbol{\mathcal{N}}_i}\left(\boldsymbol{e}_i - \boldsymbol{e}_j\right)\right\|_1\\ \leq& -\boldsymbol{e}^{\mathrm{T}}\left(\mathbf{K} - \mathbf{G}\right)\left(\mathbf{I}_{nN} - \mathbf{P}\right)\bar{\boldsymbol{\mathcal{L}}}\boldsymbol{e}\\ &- \left(\gamma - \left\|\dot{\boldsymbol{x}}^*\right\|_1\right)\boldsymbol{I}_{nN}^{\mathrm{T}}\left\|\mathbf{P}\bar{\boldsymbol{\mathcal{L}}}\boldsymbol{e}\right\|_1.\end{aligned} \tag{7}$$

According to Assumption 2, we know that $\mathbf{K}-\mathbf{G}$ is positive definite, because we can estimate that all eigenvalues of $\mathbf{K}-\mathbf{G}$ are positive by using the Gerschgorin disk theorem. According to Assumption 1, we also know $(\gamma - \|\dot{\boldsymbol{x}}^*\|_1) > 0$. Further, in view of the fact that both $(\mathbf{I}_{nN} - \mathbf{P})$ and $\bar{\mathcal{L}}$ are positive semi-definite, it is concluded $\dot{V}_e \le 0$.

When $\dot{V}_e = 0$, it represents $(\mathbf{K}-\mathbf{G})(\mathbf{I}_{nN}-\mathbf{P})\bar{\mathcal{L}}\boldsymbol{e} = 0$ and $\mathbf{P}\bar{\mathcal{L}}\boldsymbol{e} = 0$. There is no non-zero vector $\boldsymbol{z} = \bar{\mathcal{L}}\boldsymbol{e}$ that satisfies both $(\mathbf{I}_{nN}-\mathbf{P})\boldsymbol{z} = 0$ and $\mathbf{P}\boldsymbol{z} = 0$. Therefore, when $\dot{V}_e = 0$, $\bar{\mathcal{L}}\boldsymbol{e} = 0$, which implies the error $\boldsymbol{e}$ reach a consensus, such that $\boldsymbol{e}_i - \boldsymbol{e}_j \to 0, t \to \infty, \forall i, j \in \boldsymbol{\mathcal{V}}$. Further, $\boldsymbol{x}_i - \boldsymbol{x}_j \to 0$, $t \to \infty$, $\forall i, j \in \boldsymbol{\mathcal{V}}$. Local solutions also can reach a consensus. ■

In view of Theorem 1 and Theorem 2, we know that local solutions can track their corresponding time-varying manifolds and reach a consensus eventually. It implies that every local solution will satisfy all time-varying linear sub-equations (2), and these local solutions are exactly the solution of the original time-varying linear equation (1).

V. CASE STUDY

This section gives a simulation by introducing the cooperative monitoring task described in [25]. There are 6 agents cooperatively observing a linear process, such that

$$\dot{\boldsymbol{x}} = \mathbf{A}_e \boldsymbol{x}, \tag{8}$$

where $\boldsymbol{x} \in \mathbb{R}^8$, and

$$\mathbf{A}_e = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & -2 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{x}(0) = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}.$$

Each robot observes the linear process with the following linear measurement model

$$\boldsymbol{b}_i = \mathbf{A}_i \boldsymbol{x}, \tag{9}$$

where

$$\begin{aligned}
\mathbf{A}_1 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2+\cos(7t) & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \\
\mathbf{A}_2 &= \begin{bmatrix} 2+\cos(6t) & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}, \\
\mathbf{A}_3 &= \begin{bmatrix} 2+\cos(5t) & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2+\sin(4t) & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{A}_4 &= \begin{bmatrix} 2+\sin(3t) & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{A}_5 &= \begin{bmatrix} 2+\sin(2t) & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{A}_6 &= \begin{bmatrix} 2+\sin(1t) & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}$$

The objective of the cooperative monitoring task is that using the local measurement $\boldsymbol{b}_i$ and distributed communication, each robot's local estimation $\boldsymbol{x}_i$ can track the signal $\boldsymbol{x}$ governed by (8), such that

$$\lim_{t\to\infty} \boldsymbol{x}_i = \boldsymbol{x}, \quad \forall i \in \boldsymbol{\mathcal{V}}.$$

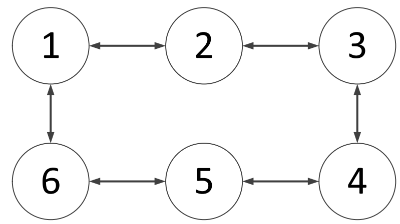

Fig. 1. The communication topology between robots.

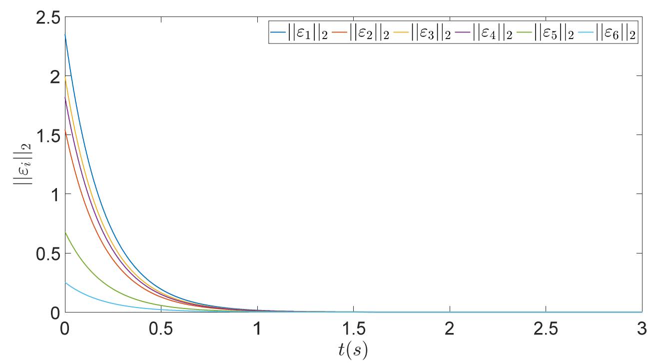

Fig. 2. Simulation results of the tracking errors of local time-varying manifolds. It is shown local solutions can converge to the time-varying manifolds.

It is noted that local linear sub-equations described in (9) constitute a linear equation, such that

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b}. \tag{10}$$

where

$$\begin{aligned} \mathbf{A} &= [\mathbf{A}_1^{\mathrm{T}} \quad \cdots \quad \mathbf{A}_6^{\mathrm{T}}]^{\mathrm{T}}, \\ \boldsymbol{b} &= [\boldsymbol{b}_1^{\mathrm{T}} \quad \cdots \quad \boldsymbol{b}_6^{\mathrm{T}}]^{\mathrm{T}}. \end{aligned}$$

Notice that it always holds $Rank(\mathbf{A}) = 8$. (10) always has a unique solution. Now, we can use the proposed distributed algorithm (4) to solve this cooperative monitoring task problem by distributed solving the time-varying linear equation (10).

$\dot{\mathbf{A}}_i$ can be obtained by taking the derivative of $\mathbf{A}_i$. $\dot{\boldsymbol{b}}_i$ can be obtained by tracking the derivative of (9). The group of 6 robots form an undirected communication topology shown in Fig. 1. Their local solutions are randomly located in $[0, 1]$. We select $\mathbf{K} = 5\mathbf{I}_{nN}$, and $\gamma = 7$, which satisfy Assumption 1 and Assumption 2.

The simulation results are given in Fig. 3, 4, and 2. For each robot, the tracking error of the time-varying manifold is shown in Fig. 3. The local solution can keep moving on the manifold after converging to it, which implies that local sub-linear equations can be satisfied. The tracking performance of the local solution to the solution of the original linear equation is shown in Fig. 4, and the tracking errors are shown in Fig. 2. All local solutions reach a consensus and converge to the solution of the original linear equation. It implies that the original time-varying linear equation is cooperatively solved.

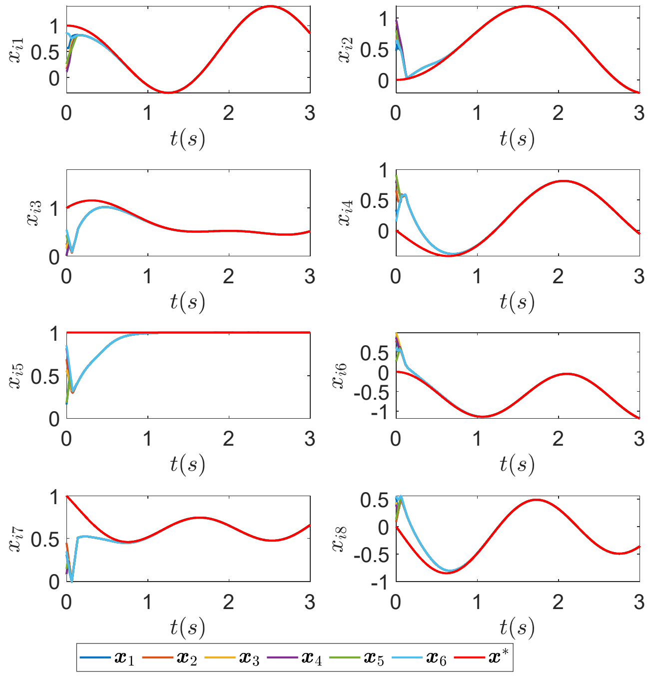

Fig. 3. Simulation results of local solutions and the solution of the original linear equation. It is shown that all local solutions can track the time-varying solution of the original linear equation.

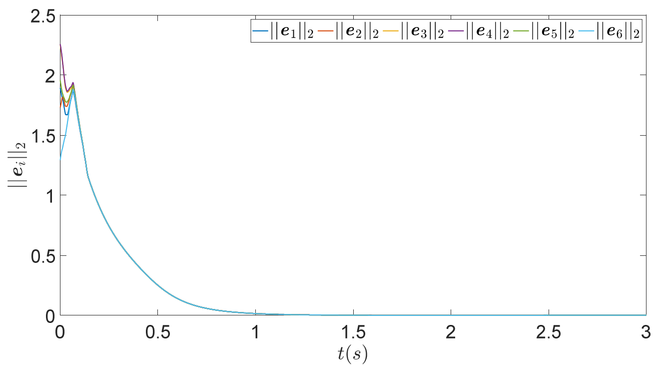

Fig. 4. Simulation results of errors of local solutions. It is shown that the errors converge to zero.

VI. CONCLUSIONS AND FUTURE WORKS

We have proposed a distributed algorithm for a group of agents cooperatively solving a time-varying linear equation, which has a unique solution. Each agent only has access to partial rows of the linear equation, i.e. each agent holds a local linear sub-equation. To cooperatively solve the time-varying linear equation, the proposed distributed algorithm enforces local solutions to track their local time-varying manifolds and to reach a consensus simultaneously, ensuring that all local solutions can converge to the solution of the original time-varying linear equation. The proposed algorithm can configure initial values of local solutions arbitrarily. Based on the simulation of the cooperative monitoring task with up to six robots, we demonstrated a successful distributed solution of the time-varying linear equation.

Future research will include exploring the distributed algorithm for least squares and minimum norm solutions of the time-varying linear equation.

VII. ACKNOWLEDGMENTS

This work was supported in part by the NSFC under grant 62373048, in part by the National Key Research and Development Program of China under No. 2022YFB4702000, No. 2022YFA1004703, in part by the Fundamental Research Funds for the Central Universities, in part by the NSFC under Grants 62133002, U1913602, 62088101, 6172010601162003277 and in part by the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100).

REFERENCES

[1] K. Balasubramanian, "Linear variational Diophantine techniques in mass balance of chemical reactions," *J. Math. Chem.*, vol. 30, no. 2, pp. 219-225, 2001.

[2] G. Xu, H. Liu, L. Tong, and T. Kailath, "A Least-Squares Approach to Blind Channel Identification," *IEEE Trans. Signal Process.*, vol. 43, no. 12, pp. 2982-2993, 1995.

[3] R. Baldick, "Electricity market equilibrium models: The effect of parametrization," *IEEE Trans. Power Syst.*, vol. 17, no. 4, pp. 1170-1176, 2002.

[4] Y. Zhang, Y. Wen, F. Li, and Y. Chen, "Distributed Observer-Based Formation Tracking Control of Multi-Agent Systems with Multiple Targets of Unknown Periodic Inputs," *Unmanned Syst.*, vol. 07, no. 01, pp. 1523, Jan. 2019.

[5] W. Deng, Y. Hong, B. D. O. Anderson, and G. Shi, "Network Flows That Solve Sylvester Matrix Equations," *IEEE Trans. Automat. Contr.*, vol. 67, no. 12, pp. 6731-6738, 2022.

[6] S. Mou and A. S. Morse, "A fixed-neighbor, distributed algorithm for solving a linear algebraic equation," in *Proc. 12th Eur. Control Conf.*, 2013, pp. 2269-2273.

[7] S. Mou, J. Liu, and A. S. Morse, "A Distributed Algorithm for Solving a Linear Algebraic Equation," *IEEE Trans. Automat. Contr.*, vol. 60, no. 11, pp. 2863-2878, Nov. 2015.

[8] B. Anderson, S. Mou, A. Morse, and U. Helmke, "Decentralized gradient algorithm for solution of a linear equation," *Numer. Algebr. Control Optim.*, vol. 6, no. 3, pp. 319-328, Sep. 2016.

[9] J. Liu, S. Mou, and A. S. Morse, "Asynchronous Distributed Algorithms for Solving Linear Algebraic Equations," *IEEE Trans. Automat. Contr.*, vol. 63, no. 2, pp. 372-385, Feb. 2018.

[10] P. Wang, W. Ren, and Z. Duan, "Distributed algorithm to solve a system of linear equations with unique or multiple solutions from arbitrary initializations," *IEEE Trans. Control Netw. Syst.*, vol. 6, no. 1, pp. 82-93, 2019.

[11] X. Shi, X. Yu, J. Cao, and G. Wen, "Continuous distributed algorithms for solving linear equations in finite time," *Automatica*, vol. 113, p. 108755, 2020.

[12] J. Liu, A. Stephen Morse, A. Nedi, and T. Baar, "Exponential convergence of a distributed algorithm for solving linear algebraic equations," *Automatica*, vol. 83, pp. 37-46, 2017.

[13] S. S. Alaviani and N. Elia, "A Distributed Algorithm for Solving Linear Algebraic Equations over Random Networks," *IEEE Trans. Automat. Contr.*, vol. 66, no. 5, pp. 2399-2406, 2021.

[14] D. Jakoveti, N. Kreji, N. Krklec Jerinki, G. Malaspina, and A. Micheletti, "Distributed fixed point method for solving systems of linear algebraic equations," *Automatica*, vol. 134, p. 109924, 2021.

[15] X. Wang, S. Mou, and B. D. O. Anderson, "Scalable, Distributed Algorithms for Solving Linear Equations via Double-Layered Networks," *IEEE Trans. Automat. Contr.*, vol. 65, no. 3, pp. 1132-1143, 2020.

[16] X. Wang, S. Mou, and D. Sun, "Improvement of a Distributed Algorithm for Solving Linear Equations," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3113-3117, Apr. 2017.

[17] J. Zhou, X. Wang, S. Mou, and B. D. O. Anderson, "Finite-Time Distributed Linear Equation Solver for Solutions with Minimum l_1-Norm," *IEEE Trans. Automat. Contr.*, vol. 65, no. 4, pp. 1691-1696, 2020.

[18] P. Wang, S. Mou, J. Lian, and W. Ren, "Solving a system of linear equations: From centralized to distributed algorithms," *Annu. Rev. Control*, vol. 47, pp. 306-322, 2019.
[19] X. Wang, J. Zhou, S. Mou, and M. J. Corless, "A distributed algorithm for least squares solutions," *IEEE Trans. Automat. Contr.*, vol. 64, no. 10, pp. 4217-4222, 2019.
[20] J. Lei, P. Yi, G. Shi, and B. D. O. Anderson, "Distributed Algorithms with Finite Data Rates that Solve Linear Equations," *SIAM J. Optim.*, vol. 30, no. 2, pp. 1191-1222, Jan. 2020.
[21] G. Shi, B. D. O. Anderson, and U. Helmke, "Network Flows That Solve Linear Equations," *IEEE Trans. Automat. Contr.*, vol. 62, no. 6, pp. 2659-2674, 2017.
[22] Y. Liu, Y. Lou, B. D. O. Anderson, and G. Shi, "Network flows that solve least squares for linear equations," *Automatica*, vol. 120, p. 109108, 2020.
[23] Y. Liu, C. Lageman, B. D. O. Anderson, and G. Shi, "An Arrow-Hurwicz-Uzawa type flow as least squares solver for network linear equations," *Automatica*, vol. 100, pp. 187-193, 2019.
[24] S. Cheng, J. Lei, X. Zeng, Y. Fan, and Y. Hong, "Effective distributed algorithm for solving linear matrix equations," *Sci. China Inf. Sci.*, vol. 66, no. 8, p. 189202, Aug. 2023.
[25] R. K. Ramachandran, P. Pierpaoli, M. Egerstedt, and G. S. Ssukhatme, "Resilient Monitoring in Heterogeneous Multi-Robot Systems Through Network Reconfiguration," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 126-138, 2022.