A Distributed Least-Squares Solver for Linear Systems of Algebraic Equations

Mohammad Jahvani and Martin Guay

Department of Chemical Engineering
Queen's University, Kingston, ON, Canada
e-mail: {mohammad.jahvani, guaym}@queensu.ca

Abstract: This paper introduces a consensus-based continuous-time distributed algorithm to find the least-squares solution to overdetermined systems of linear algebraic equations over directed multi-agent networks. It is assumed that each agent has only access to a subsystem of the algebraic equations, and the underlying communication network is strongly connected. We show that, along the flow of the proposed algorithm, the local estimate of each agent converges exponentially to the exact least-squares solution, provided that the aggregate system of linear equations has full column rank, and each agent knows an upper bound on the total number of the participating agents in the network.

Keywords: Multi-agent systems, Distributed control and estimation, Sensor networks.

In a wide range of applications in systems and control theory, such as identification, estimation, learning, signal processing, etc, it is required to solve systems of linear algebraic equations. With the penetration of new technologies and advent of internet of things, we are facing with emerging applications of large-scale nature. In addition, it is not desirable to use common shared data centres, due to increasing concerns about privacy and security issues. To address these issues, several distributed solvers are proposed to solve systems of linear algebraic equations over multi-agent networks, Mou et al. (2015); Anderson et al. (2016); Shi et al. (2016); Zeng and Cao (2017); Liu et al. (2018). For a recent survey on the subject, see Wang et al. (2019), and the references therein.

On the other hand, in many applications like distributed parameter estimation, Kar et al. (2012), filtering, Cattivelli et al. (2008), and other tasks in sensor networks, Rabbat and Nowak (2004), the given system of linear equations, may not have a solution. In such cases, it is often desirable to obtain an approximate solution to the given system of linear equations in the sense of least-squares. As a result, in recent years, we have witnessed a surge of interest in the development of least-squares solvers for systems of linear equations over multi-agent networks, Wang and Elia (2012); Wang et al. (2019); George and Yang (2019); Liu et al. (2019); Yang et al. (2020); Liu et al. (2020); Jahvani and Guay (2020).

Some of these proposed algorithms, like Liu et al. (2020), suffer from slow rate of convergence. On the other hand, the majority of the existing distributed least-squares solvers, except for Yang et al. (2020), and Jahvani and Guay (2020), can only operate on undirected or weight-balanced networks. In other words, to obtain the exact least-squares solution, these algorithms require either symmetric or weight-balanced communication links. This condition may not be satisfied, especially in applications with broadcast-based communications.

Indeed, to address such concerns, and with the assumption of *a priori* knowledge about the node out-degrees in static and strongly connected directed networks, Yang et al. (2020), and Jahvani and Guay (2020) introduced discrete-time and continuous-time distributed algorithms that can find the exact least-squares solution. These algorithms have a linear rate of convergence. The continuous-time distributed algorithm presented in our previous work, Jahvani and Guay (2020), can be implemented in discrete-time using coordinated or uncoordinated step-sizes. In addition, the proposed algorithm can be utilized as a leader-based distributed algorithm to improve the transient response and the convergence rate.

It should be noted that the knowledge of the out-degrees is not a local information in directed networks. As a result, in this work, we propose an alternative continuous-time distributed dynamics to find the least-squares solution to systems of linear algebraic equations by imposing the prior knowledge about the upper bound on the size of the network. We show that, for strongly connected directed networks, the proposed algorithm converges exponentially to the least-squares solution without any *a-priori* knowledge about the out-degrees of agents.

The remainder of the paper is organized as follows. First we introduce some mathematical notation. The statement of the problem and the underlying assumptions are presented in Section 1. The proposed dynamics is presented in Section 2. The main results are provided in Section 3, followed by some simulations in Section 4. Finally, Section 5 contains our conclusions.

Notation. The set of real numbers (resp., the set of nonnegative real numbers) is denoted by $\mathbb{R}$ (resp., $\mathbb{R}_{\geq 0}$). We denote the set of $m \times n$ matrices with real entries by $\mathbb{R}^{m\times n}$, the $n \times n$ identity matrix by I_n, and the j-

⋆ This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

th column of I_n by e_j. The column vector of all-ones in $\mathbb{R}^n$ is represented by $\mathbf{1}$, and $\mathbf{0}$ denotes the zero-matrix (its dimension to be understood from the context). We write $\text{diag}(v_1, \cdots, v_n)$ for an $n \times n$ diagonal matrix whose diagonal entries starting in the upper left corner are to be $v_1, \cdots, v_n$. By a positive vector, we mean a vector whose entries are all strictly positive. For a real matrix M, we let $[M]_{ij}$ to be its (i, j)-entry and we denote its transpose by M'. For two matrices M and N, we denote their Kronecker product by $M \otimes N$. Given a symmetric matrix M, we write $M \succeq 0$ if M is positive semidefinite. For a positive integer n, we let $[n] := \{1, 2, \ldots, n\}$. Throughout this paper, we let $\|\cdot\|$ to be the standard Euclidean norm, or the induced ℓ_2 norm for matrices.

1. PROBLEM FORMULATION AND ASSUMPTIONS

Consider a network of n agents that must coordinate with another to find a solution to the following problem:

$$\underset{x \in \mathbb{R}^p}{\text{minimize}}\, f(x) = \sum_{i=1}^{n} \frac{1}{2} \|A_i x - b_i\|^2, \tag{1}$$

where $A_i \in \mathbb{R}^{q_i \times p}$ and $b_i \in \mathbb{R}^{q_i}$, $i = 1, 2, \ldots, n$, are private data that only belong to agent i.

Let

$$H := \begin{pmatrix} A_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & A_2 & \cdots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & A_n \end{pmatrix}. \tag{2}$$

It is not hard to see the connection between problem (1) and the equivalent (ordinary) least-squares problem associated with the aggregate system of linear algebraic equations

$$Ax = b, \tag{3}$$

where

$$A = \begin{pmatrix} A_1 \\ \hline A_2 \\ \hline \vdots \\ \hline A_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \hline b_2 \\ \hline \vdots \\ \hline b_n \end{pmatrix}. \tag{4}$$

Let $A \in \mathbb{R}^{q \times p}$ and $b \in \mathbb{R}^q$, where $q = \sum_i q_i$.

Assumption 1. Matrix A has full column rank.

Assumption 1 implies that (3) admits a unique least-squares solution that we denote by x^{LS}, i.e.,

$$x^{LS} = (A'A)^{-1} A'b. \tag{5}$$

The agents are labeled 1 through n. Each agent i can receive information from certain other agents called its *in-neighbours*. We denote the set of in-neighbour of agent i by $\mathcal{N}_i^{in}$. Similarly, each agent i can send information to certain other agents called its *out-neighbours*. We denote the set of out-neighbour of agent i by $\mathcal{N}_i^{out}$. We also let $d_i^{in} := |N_i^{in}|$ to be the in-degree of agent i. Notice, the communications could be asymmetric. We model the underlying communication network by a directed graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, called the *neighbour graph*, where $\mathcal{V} = \{1, 2, \cdots, n\}$ is the vertex set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represents the edge set. There is a directed edge from vertex i to vertex j in $\mathbb{G}$, if agent j can receive information from agent i.

We impose the following additional assumptions on the given problem.

Assumption 2. The directed neighbor graph $\mathbb{G}$ is static and strongly connected.

Assumption 3. Each agent knows an upper bound $\bar{n}$ on the size of the network.

Under the given assumptions, it is desired to design a distributed algorithm that can solve the problem (1) and find the least-squares solution x^{LS} to the aggregate system of linear algebraic equations (3), with an exponential rate of convergence.

Throughout this paper, we assume that agents can only acquire information from their in-neighbours.

Remark 4. To overcome the well-known fundamental limitations of broadcast-based deterministic protocols over directed networks, Hendrickx and Tsitsiklis (2015); and to solve the distributed least-squares problem (1), we need to assume additional prior knowledge about the communication network. A commonly adopted assumption in the consensus literature or distributed convex optimization area is the *a-priori* knowledge about the out-degree of each agent. This assumption has already been investigated in Jahvani and Guay (2020), and Yang et al. (2020). Here, we utilize the prior knowledge about the upper bound on the size of network.

1.1 Motivating Example

Consider a sensor network of n stationary agents which is deployed to track a maneuvering target. Each agent/sensor is located at position $p_i \in \mathbb{R}^2$, known to the corresponding agent. The communication constraints between the agents is modelled by a directed graph $\mathbb{G}$. Each agent obtains bearing measurements from a maneuvering target at position $x(t) \in \mathbb{R}^2$, whose kinematics is governed by

$$\dot{x}(t) = v(t),$$

where $v(t) \in \mathbb{R}^2$ denotes the velocity of the target. The bearing measurements are represented by unit vectors as

$$u_i(t) = \frac{x(t) - p_i}{\|x(t) - p_i\|}, \quad i \in [n].$$

Let $u_i(t) = [\cos(\theta_i(t)), \sin(\theta_i(t))]'$, where $\theta_i(t) \in [0, 2\pi)$ denotes the bearing angle with respect to a fixed reference frame of the i-th agent/sensor. It is easy to show that the position of the maneuvering target can be obtained by solving the following system of linear equations associated with the entire sensor network (George and Yang (2019)):

$$A(t)x(t) = b(t),$$

where $x(t)$ is the position of the maneuvering target, $b_i(t) = a_i(t)' p_i$, and

$$A(t) = \begin{pmatrix} a_1(t)' \\ a_2(t)' \\ \vdots \\ a_n(t)' \end{pmatrix},$$

with $a_i(t) = [-\sin(\theta_i(t)), \cos(\theta_i(t))]'$, for $i \in [n]$.

Therefore, in order to track the position of the maneuvering target, it is required to design a distributed least-squares solver with exponential rate of convergence.

2. THE PROPOSED DYNAMICS

In this section, we introduce a continuous-time distributed dynamics to solve the linear least-squares problem (1) on directed networks.

To solve the linear least-squares problem (1) we define an equivalent optimization problem by assigning a local estimate $x_i \in \mathbb{R}^p$ of the global variable x to each agent and imposing a Laplacian-based consensus constraint that ensures $x_i = x_j$, for all $i, j \in [n]$.

For the neighbour graph $\mathbb{G}$, we define the associated Laplacian matrix L as follows:

$$[L]_{ij} := \begin{cases} d_i^{in}, & \text{if } i = j \\ -1, & \text{if } (j,i) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

By construction, zero is an eigenvalue of the Laplacian matrix, and $L\mathbf{1} = \mathbf{0}$. Since $\mathbb{G}$ is assumed to be strongly connected, we can show that zero is a simple eigenvalue of L. The remaining eigenvalues of L, however, have strictly positive real-part. Furthermore, there exists a unique positive vector $\omega = (\omega_1, \omega_2, \cdots, \omega_n)'$, such that $\omega' L = 0$ and $\omega' \mathbf{1} = 1$. See, e.g., Bullo et al. (2009).

To find a distributed algorithm that solves the problem (1), consider the following equivalent problem:

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{R}^{np}}{\text{minimize}}\, F(\mathbf{x}) = \sum_{i=1}^{n} \frac{1}{2}\|A_i x_i - b_i\|^2 \\ &\quad\text{subject to}:\ \mathbf{L}\mathbf{x} = \mathbf{0}, \end{aligned} \tag{7}$$

where $\mathbf{x} := (x_1', x_2', \cdots, x_n')'$, and $\mathbf{L} = L \otimes I_p$.

Remark 5. Note that $F(\mathbf{x})$ is convex and its restriction to the consensus subspace is strictly convex. However, the local cost functions $f_i(x_i) := \frac{1}{2}\|A_i x_i - b_i\|^2, i \in [n]$, are in general neither strictly convex nor have bounded gradients. Moreover, the neighbour graph $\mathbb{G}$ is not necessarily weight-balanced and the agents do not know their out-degrees.

Knowing an upper bound on the size of the network, we can easily assign unique labels to each agent by implementing a finite-time distributed algorithm in the initialization stage. (See, for example, Chopra et al. (2017)).

To solve the equivalent least-squares problem (7) over arbitrary strongly connected networks, we propose the following dynamics for each agent:

$$\begin{aligned} \dot{x}_i &= \alpha\,[y_i]_i \sum_{j\in\mathcal{N}_i^{in}} (x_j - x_i) - [y_i]_i\, z_i - A_i'\,(A_i x_i - b_i) \\ \dot{z}_i &= \alpha \sum_{j\in\mathcal{N}_i^{in}} (x_i - x_j) \\ \dot{y}_i &= \beta \sum_{j\in\mathcal{N}_i^{in}} (y_j - y_i)\,, \end{aligned} \tag{8}$$

where $\alpha, \beta > 0$ are design parameters, $x_i(t), z_i(t) \in \mathbb{R}^p$ and $y_i(t) \in \mathbb{R}^{\bar{n}}$ are the state vectors associated with agent i at time $t \geq 0$, and $[y_i]_i$ denotes the i-th component of y_i. The initial condition $x_i(0)$ is chosen arbitrarily, $z_i(0) = 0$, and $y_i(0) = [e_i', \mathbf{0}_{1\times(\bar{n}-n)}]'$, for $i \in [n]$. (Recall that e_i denotes the i-th standard basis of $\mathbb{R}^n$.)

The proposed dynamics (8) is distributed over the underlying directed network, in the sense that agents are able to compute the flow only using the information they receive from their in-neighbours.

Let $\mathbf{x} := (x_1', x_2', \cdots, x_n')'$, $\mathbf{z} := (z_1', z_2', \cdots, z_n')'$, and $Y(t) \in \mathbb{R}^{n\times\bar{n}}$ be a matrix whose i-th row is $y_i(t)'$, for $i \in [n]$, and all $t \geq 0$. We define the $n \times n$ diagonal matrix $Y_d(t) := \text{diag}([Y(t)]_{11}, [Y(t)]_{22}, \cdots, [Y(t)]_{nn})$, for $t \geq 0$. Accordingly, we let $\mathbf{Y_d}(t) := Y_d(t) \otimes I_p$.

Using the notation above, the proposed dynamics (8) reads as:

$$\begin{aligned} \dot{\mathbf{x}} &= -\alpha\,\mathbf{Y_d}\,\mathbf{L}\mathbf{x} - \mathbf{Y_d}\,\mathbf{z} - H'\,(H\mathbf{x} - b) \\ \dot{\mathbf{z}} &= \alpha\,\mathbf{L}\mathbf{x} \\ \dot{Y} &= -\beta\, LY, \end{aligned} \tag{9}$$

where $\mathbf{x}(0)$ is arbitrary, $\mathbf{z}(0) = 0$, and $Y(0) = [I_n, \mathbf{0}_{n\times(\bar{n}-n)}]$.

Remark 6. The proposed dynamics (8) is inspired by the distributed algorithm introduced in Kia et al. (2015). It should be noted that the proposed dynamics (8) is different from the distributed algorithm in Kia et al. (2015), from two aspects. First, the local cost functions in the distributed least-squares problem are not necessarily strongly convex. Second, the algorithm introduced in Kia et al. (2015), only works on weight-balanced directed graphs. In other words, our problem does not satisfy the conditions of the algorithm developed in Kia et al. (2015).

Remark 7. In the sequel, we will introduce an alternative distributed dynamics that can also solve the problem (1). However, due to page restrictions, we will omit the proof of convergence of these algorithms.

3. MAIN RESULT

In this section, we state the main result of this paper.

By Assumption 2, the underlying communication network is strongly connected. Therefore, the Laplacian matrix L associated with the neighbour graph $\mathbb{G}$ has a simple zero eigenvalue, while the rest of its eigenvalues have strictly positive real-part. Let $\mathbf{1}$ and ω, respectively, denote the unique right and left (positive) eigenvectors associated with the zero eigenvalue of L, such that $\omega' \mathbf{1} = 1$. Then, it is well-known that $\lim_{t\to\infty} \exp(-Lt) = \mathbf{1}\omega'$. In particular, there exist (strictly) positive constants ρ and κ such that $\|\exp(-Lt) - \mathbf{1}_n\omega'\| \leq \rho e^{-\kappa t}$, for all $t \geq 0$. See, for example, Bullo et al. (2009).

Note that the Laplacian matrix L is a Metzler matrix, hence, the non-negative orthant $\mathbb{R}_+^n$ is positively invariant under the Laplacian flow, Berman and Plemmons (1994). Using this fact, and the preceding discussion, we can easily deduce the following result.

Lemma 8. Consider the Y–dynamics in (9) with $Y(0) = [I_n, \mathbf{0}_{n\times(\bar{n}-n)}]$. Let Assumptions 2–3 hold. Then, there exist constants $c, C > 0$ such that $c \leq [Y]_{ii}(t) \leq C$, for all $t \geq 0$ and for all $i \in [n]$. Furthermore, there exist $\rho, \kappa > 0$, such that $|\,[Y]_{ii}(t) - \omega_i\,| \leq \rho e^{-\kappa t}$, for all $t \geq 0$ and for all $i \in [n]$.

Since, $\lim_{t\to\infty} \mathbf{Y_d}(t) = \mathbf{\Omega} := \text{diag}(\omega_1, \omega_2, \ldots, \omega_n) \otimes I_p$, the equilibria of the proposed dynamics (8) is determined by the following auxiliary dynamics:

$$\begin{aligned} \dot{\mathbf{x}} &= -\alpha\,\mathbf{\Omega}\,\mathbf{L}\mathbf{x} - \mathbf{\Omega}\,\mathbf{z} - H'\,(H\mathbf{x} - b) \\ \dot{\mathbf{z}} &= \alpha\,\mathbf{L}\mathbf{x}. \end{aligned} \tag{10}$$

Lemma 9. Consider the auxiliary dynamics (10) with $\mathbf{z}(0) = \mathbf{0}$, and $\mathbf{\Omega} = \Omega \otimes I_p$, where $\Omega := \text{diag}(\omega_1, \cdots, \omega_n)$, and $(\omega_1, \cdots, \omega_n)L = \mathbf{0}$. Let Assumptions 1–2 hold, and $(\mathbf{x_{ss}}', \mathbf{z_{ss}}')'$ be the equilibria of the auxiliary dynamics (10). Then, we have $\mathbf{x_{ss}} = \mathbf{1} \otimes x^{LS}$, where x^{LS} is the unique minimizer of the problem (1), and $\mathbf{z_{ss}} = \mathbf{\Omega}^{-1} H'(b - H\mathbf{x_{ss}})$.

Proof. Since $(\omega' \otimes I_p)\,\dot{\mathbf{z}} = 0$, we have $\sum_{i=1}^{n} \omega_i z_i(t) = \sum_{i=1}^{n} \omega_i z_i(0) = 0$, for all $t \geq 0$. To find the equilibria of the auxiliary dynamics (10), let $\dot{\mathbf{x}} = \dot{\mathbf{z}} = \mathbf{0}$. From Assumption 2, it follows that the equilibrium $(\mathbf{x_{ss}}', \mathbf{z_{ss}}')'$ must satisfy

$$\begin{aligned} \mathbf{x_{ss}} &= \mathbf{1} \otimes v, \\ \mathbf{\Omega}\, \mathbf{z_{ss}} &= H'(b - H\mathbf{x_{ss}}), \end{aligned}$$

for some vector $v \in \mathbb{R}^p$. Multiplying the latter identity on the left by $(\mathbf{1}' \otimes I_p)$, we obtain

$$\mathbf{0} = \left(\sum_{i=1}^{n} A_i'(A_i v - b_i) \right) = A'(Av - b).$$

On the other hand, Assumption 1 implies that the unique solution to the normal equation $A'Av = A'b$ is the least-squares solution x^{LS}. Hence, $\mathbf{x_{ss}} = \mathbf{1} \otimes x^{LS}$, and $\omega_i\, \mathbf{z}_{\mathbf{ss},i} = A_i'(b - A_i x^{LS})$. This completes the proof.

Now we are ready to state the main result of the paper.

Theorem 10. Consider the least-squares problem (1), and let Assumptions 1–3 hold. Then, there exists $\alpha^\star > 0$ such that for $\alpha > \alpha^\star$, each solution $t \mapsto (\mathbf{x}(t), \mathbf{z}(t), Y(t))$ of the distributed dynamics (9) with initial conditions $(\mathbf{x_0}, \mathbf{z_0})$ in $\mathcal{S} = \{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{np} \times \mathbb{R}^{np} \,|\, (\omega' \otimes I_p)\, \mathbf{z_0} = 0\}$, and $Y(0) = [I_n, \mathbf{0}_{n \times (\bar{n}-n)}]$, converges exponentially to the point $\left(\mathbf{1} \otimes x^{LS}, \mathbf{z_{ss}}, \mathbf{1} \otimes \omega'\right)$, where $\mathbf{z_{ss}} = \mathbf{\Omega}^{-1} H' \left(b - H(\mathbf{1} \otimes x^{LS})\right)$.

Next, we present an alternative distributed dynamics that can solve the equivalent least-squares problem (7) on any strongly connected network.

Consider the distributed dynamics:

$$\begin{aligned} \dot{x}_i &= \alpha \sum_{j \in \mathcal{N}_i^{in}} (x_j - x_i) - z_i - \frac{1}{[y_i]_i} A_i' \left(A_i x_i - b_i\right) \\ \dot{z}_i &= \alpha \sum_{j \in \mathcal{N}_i^{in}} (x_i - x_j) \\ \dot{y}_i &= \beta \sum_{j \in \mathcal{N}_i^{in}} (y_j - y_i)\,. \end{aligned} \tag{11}$$

One can show that (11) enjoys similar properties to that of the distributed dynamics (8). In particular, all the trajectories that start from $\mathcal{W} := \{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{np} \times \mathbb{R}^{np} \,|\, (\omega' \otimes I_p)\, \mathbf{z_0} = 0\}$, and $Y(0) = [I_n, \mathbf{0}_{n \times (\bar{n}-n)}]$, will converge exponentially to the point $\left(\mathbf{1} \otimes x^{LS}, \bar{\mathbf{z}}_{\mathbf{ss}}, \mathbf{1} \otimes \omega'\right)$, where $\bar{\mathbf{z}}_{\mathbf{ss}} = \mathbf{\Omega}^{-1} H' \left(b - H(\mathbf{1} \otimes x^{LS})\right)$.

4. SIMULATIONS

In this section, we provide a numerical example to demonstrate the performance of the proposed distributed dynamics.

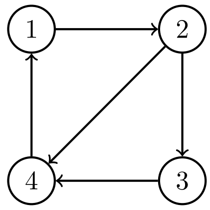

Fig. 1. The neighbour graph $\mathcal{G}$ modelling the communication topology.

We consider a network of $n = 4$ agents that can communicate with each other according to the neighbour graph $\mathcal{G}$ as shown in Fig. 1. The given communication topology is strongly connected, however, it is not weight-balanced. These agents utilize the distributed dynamics (8) to find the solution of the least-squares problem (1) without exchanging the private data (A_i, b_i), where

$$\begin{aligned} A_1 &= \begin{pmatrix} 1.4090 & 0.4889 & 0.8884 \\ 1.4172 & 1.0347 & -1.1471 \end{pmatrix} & b_1 &= \begin{pmatrix} 3.2520 \\ -7.5490 \end{pmatrix} \\ A_2 &= \begin{pmatrix} 0.6715 & 0.7269 & -1.0689 \end{pmatrix} & b_2 &= \begin{pmatrix} 13.7030 \end{pmatrix} \\ A_3 &= \begin{pmatrix} -1.2075 & -0.3034 & -0.8095 \end{pmatrix} & b_3 &= \begin{pmatrix} -17.1150 \end{pmatrix} \\ A_4 &= \begin{pmatrix} 0.7172 & 0.2939 & -2.9443 \\ 1.6302 & -0.7873 & 1.4384 \end{pmatrix} & b_4 &= \begin{pmatrix} -1.0220 \\ -2.4140 \end{pmatrix}. \end{aligned}$$

It is not hard to verify that the associated aggregate system of linear algebraic equations (4) satisfies Assumption 1, and admits a unique least-squares solution $x^{LS} = (0.8140, 5.9143, 2.0221)'$.

We assume that the initial states $x_i(0)$ are chosen randomly and $z_i(0)$ are set to zero. The tuning parameters $\alpha = 2.5$, and $\beta = 1$ are used to demonstrate the performance of the distributed algorithm (8). The trajectories of each component of the estimated least-squares solutions $x_i(\cdot)$ are illustrated in Fig. 2 for $i \in [n]$. The corresponding components of the least-squares solution x^{LS} are also shown (in black colour) with dash-dotted lines. It can be observed that all the local estimated solutions $t \mapsto x_i(t)$, for $i \in [n]$, and $t \geq 0$, converge exponentially to the exact least-squares solution x^{LS}, as asserted by Theorem 10.

For the purpose of comparison, the simulation results associated with the proposed distributed dynamics (11) are depicted in Fig. 3. We use the same tuning parameters. In our experience, the proposed distributed dynamics (8) has a superior numerical performance to the distributed dynamics (11), potentially due to round-off errors in finite precision computations.

5. CONCLUSION

We considered the problem of obtaining the least-squares solution to systems of linear algebraic equations over multi-agent networks. In particular, we assumed that the underlying communications could be asymmetric. We proposed a consensus-based continuous-time algorithm to solve this problem in a distributed manner. The proposed algorithm converges exponentially to the exact least-squares solution when the underlying communication network is strongly connected. The approach presented in this study uses prior knowledge about an upper bound on the size of the network, rather than requiring any *a-priori* knowledge about the out-degrees of agents.

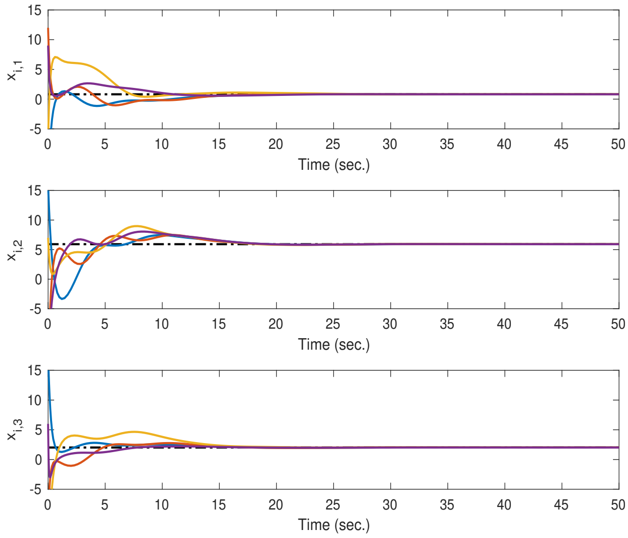

Fig. 2. The trajectories of each component of the estimated least-squares solutions $x_i(\cdot)$ in (8), and each component of the exact least-squares solution x^{LS} (dash-dotted line in black)

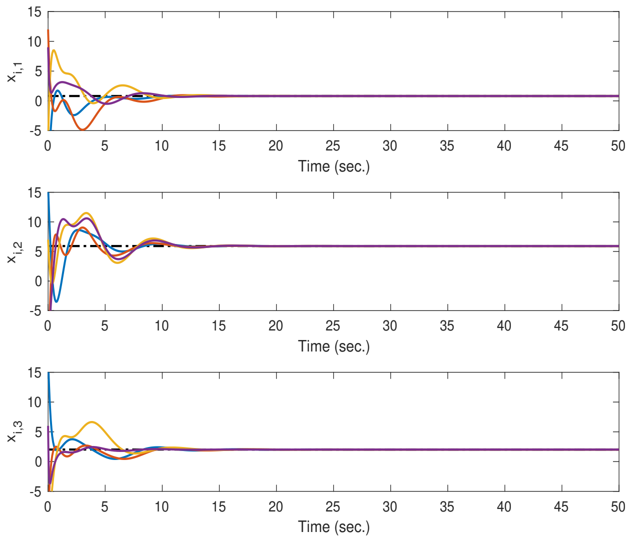

Fig. 3. The trajectories of each component of the estimated least-squares solutions $x_i(\cdot)$ in (11)

REFERENCES

Anderson, B., Mou, S., Morse, A.S., and Helmke, U. (2016). Decentralized gradient algorithm for solution of a linear equation. *Numerical Algebra, Control and Optimization*, 6(3), 319–328.

Berman, A. and Plemmons, R.J. (1994). *Nonnegative matrices in the mathematical sciences*. Society for Industrial and Applied Mathematics (SIAM).

Bullo, F., Cortes, J., and Martinez, S. (2009). *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, volume 27. Princeton University Press.

Cattivelli, F.S., Lopes, C.G., and Sayed, A.H. (2008). Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Transactions on Signal Processing*, 56(5), 1865–1877.

Chopra, S., Notarstefano, G., Rice, M., and Egerstedt, M. (2017). A distributed version of the hungarian method for multirobot assignment. *IEEE Transactions on Robotics*, 33(4), 932–947.

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

Hendrickx, J.M. and Tsitsiklis, J.N. (2015). Fundamental limitations for anonymous distributed systems with broadcast communications. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 9–16. IEEE.

Jahvani, M. and Guay, M. (2020). Solving least squares for linear equations over strongly connected directed networks. In *2020 Australian and New Zealand Control Conference (ANZCC)*, 178–183.

Kar, S., Moura, J.M.F., and Ramanan, K. (2012). Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. *IEEE Transactions on Information Theory*, 58(6), 3575–3605.

Kia, S.S., Cortés, J., and Martínez, S. (2015). Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 55, 254–264.

Liu, J., Mou, S., and Morse, A.S. (2018). Asynchronous distributed algorithms for solving linear algebraic equations. *IEEE Transactions on Automatic Control*, 63(2), 372–385.

Liu, Y., Lageman, C., Anderson, B.D., and Shi, G. (2019). An arrow–hurwicz–uzawa type flow as least squares solver for network linear equations. *Automatica*, 100, 187–193.

Liu, Y., Lou, Y., Anderson, B.D., and Shi, G. (2020). Network flows that solve least squares for linear equations. *Automatica*, 120, 109108.

Mou, S., Liu, J., and Morse, A.S. (2015). A distributed algorithm for solving a linear algebraic equation. *IEEE Transactions on Automatic Control*, 60(11), 2863–2878.

Rabbat, M. and Nowak, R. (2004). Distributed optimization in sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 20–27.

Shi, G., Anderson, B.D., and Helmke, U. (2016). Network flows that solve linear equations. *IEEE Transactions on Automatic Control*, 62(6), 2659–2674.

Wang, J. and Elia, N. (2012). Distributed least square with intermittent communications. In *2012 American Control Conference (ACC)*, 6479–6484. IEEE.

Wang, P., Mou, S., Lian, J., and Ren, W. (2019). Solving a system of linear equations: From centralized to distributed algorithms. *Annual Reviews in Control*, 47, 306–322.

Wang, X., Zhou, J., Mou, S., and Corless, M.J. (2019). A distributed algorithm for least squares solutions. *IEEE Transactions on Automatic Control*, 64(10), 4217–4222.

Yang, T., George, J., Qin, J., Yi, X., and Wu, J. (2020). Distributed least squares solver for network linear equations. *Automatica*, 113, 108798.

Zeng, X. and Cao, K. (2017). Computation of linear algebraic equations with solvability verification over multi-agent networks. *Kybernetika*, 53(5), 803–819.