Kybernetika

Xianlin Zeng; Kai Cao

Computation of linear algebraic equations with solvability verification over multi-agent networks

Kybernetika, Vol. 53 (2017), No. 5, 803–819

Persistent URL: http://dml.cz/dmlcz/147094

KYBERNETIKA — VOLUME *53* (2017), NUMBER 5, PAGES 803–819

COMPUTATION OF LINEAR ALGEBRAIC EQUATIONS WITH SOLVABILITY VERIFICATION OVER MULTI-AGENT NETWORKS

Xianlin Zeng and Kai Cao

In this paper, we consider the problem of solving a linear algebraic equation $Ax = b$ in a distributed way by a multi-agent system with a solvability verification requirement. In the problem formulation, each agent knows a few columns of A, different from the previous results with assuming that each agent knows a few rows of A and b. Then, a distributed continuous-time algorithm is proposed for solving the linear algebraic equation from a distributed constrained optimization viewpoint. The algorithm is proved to have two properties: firstly, the algorithm converges to a least squares solution of the linear algebraic equation with any initial condition; secondly, each agent in the algorithm knows the solvability property of the linear algebraic equation, that is, each agent knows whether the obtained least squares solution is an exact solution or not.

Keywords: multi-agent network, distributed optimization, linear algebraic equation, least squares solution, solvability verification

Classification: 15A06, 93D20

1. INTRODUCTION

Nowadays, the increasing scale and exploding data of real-world problems in science and engineering fields pose new challenges for the algorithm design based on computation, communication, and control. The centralized algorithms designed for small size or modest size problems may be infeasible for large-scale problems and distributed algorithms are in demand. Due to the applications in large-scale decision or control problems, distributed optimization and control algorithms have attracted a significant amount of attention (see [8, 9, 10, 14, 15, 16, 17]). Both discrete-time and continuous-time algorithms (see [8, 9, 10, 14, 17]) have been proposed and investigated for distributed optimization problems with various types of constraints, for example, local inequality/equality constraints, set constraints, and resource allocation constraints. Distributed continuous-time solvers have gained research interests in recent years. On one hand, continuous-time methods may provide new viewpoints and tools in designing distributed algorithms, thanks to the well-development of control theory. On the other hand, physical plants or hardware devices may involve in the computation of optimization problems.

DOI: 10.14736/kyb-2017-5-0803

Linear algebraic equations are widely used in various computation tasks in science, engineering, and mathematics. As a result, there is a vast literature on solving linear algebraic equations with small or modest sizes. Recently, the distributed computation of linear algebraic equations has attracted much interest and has been extensively studied by using multi-agent networks in [4, 5, 6, 7, 12, 13]. On one hand, distributed algorithms, which use multiple processing units, might provide significant improvements in computation efficiency if the algorithms are well designed. On the other hand, various systems have distributed structure requirements, for example, one agent holds part of the information and it cannot or does not want to share its local information with other agents.

One of the most fundamental problems in linear algebraic equations is the computation of x in $Ax = b$ given matrix A and vector b. Because the computation of the (least squares) solution to linear algebraic equations is related to an optimization problem $\min_x \|Ax - b\|^2$, both discrete and continuous-time algorithms for linear equations of the form $Ax = b$ are developed from the viewpoint of distributed control and optimization (see [4, 5, 6, 7, 12, 13]). Recall that most of the distributed linear equation solvers have been proposed based on the assumption that each agent knows a few rows of A and b. Based on the solvability assumption of linear algebraic equations, various distributed algorithms have been proposed in [5, 6]. Furthermore, [13] and [4] have proposed distributed algorithms solving least squares solutions of linear algebraic equations in the approximation sense or under specific graphs.

The main purpose of this paper is to design a distributed continuous-time algorithm for solving linear algebraic equation $Ax = b$, where each agent knows a few columns of A, with solvability verification. The proposed algorithm needs to accomplish two missions: (1) it obtains a least squares solution to $Ax = b$ in a distributed way; (2) it verifies whether the problem is solvable in a distributed way, that is, whether the obtained solution is an exact solution. Compared with [5, 6], this paper considers the linear equations which may not have exact solutions. Compared with [13] and [4], the proposed algorithm in this paper is able to verify whether the problem is solvable while solving a least squares solution.

The contributions of this paper are summarized as follows.

1) This paper studies the distributed computation of a famous linear algebraic equation $Ax = b$, where each agent knows a few columns of the coefficient matrix A in our framework, with a solvability verification. This is a different formulation compared with many existing algorithms solving similar problems, which assume that each agent knows a few rows of the matrix A. This paper not only considers distributed computation of linear equations without the assumption on the existence of exact solutions, but also verifies the solvability of linear algebraic equations.

2) By using a distributed constrained optimization reformulation, we propose a distributed continuous-time algorithm for the linear algebraic equation. The proposed algorithm is able to find a least squares solution to the linear algebraic equation for any initial condition under mild assumptions. Furthermore, the agents in the proposed algorithm are able to know whether the obtained solution is an exact solution.

3) We combine techniques of optimization (saddle point dynamics of Lagrangian functions) and control (the Lyapunov approach) to prove the correctness and convergence of the proposed algorithm.

The paper is built up as following. The mathematical preliminaries are presented in Section 2, while the problem of distributed computation of a linear algebraic equation with solvability verification is formulated and transformed into an equivalent standard distributed optimization problem in Section 3. In Section 4, a distributed continuous-time algorithm for the linear equation is proposed with rigorous proofs for its correctness and convergence. Section 5 gives numerical examples to show the efficacy of the proposed algorithm and Section 6 concludes the paper.

2. MATHEMATICAL PRELIMINARIES

In this section, we review relevant notations and mathematical preliminaries.

Let $\mathbb{R}$ denote the set of real numbers; let $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the set of n-dimensional real column vectors and the set of n-by-m real matrices, respectively; $\mathfrak{B}(\mathbb{R}^q)$ denotes the collection of all subsets of $\mathbb{R}^q$; I_n denotes the $n \times n$ identity matrix and $(\cdot)^{\mathrm{T}}$ denotes the transpose. Furthermore, we write $\|\cdot\|$ as the Euclidean norm, $\mathrm{rank}(A)$ as the rank of matrix A, $\mathrm{range}(A)$ as the range of A, and $\ker(A)$ as the kernel of A. Recall that $\mathrm{range}(A)$ is the orthogonal complement of $\ker(A^{\mathrm{T}})$ (denoted by $\mathrm{range}(A) = \ker(A^{\mathrm{T}})^{\perp}$). Write 1_n for the $n \times 1$ ones vector, 0_n ($0_{n,n}$) for the $n \times 1$ zeros vector ($n \times n$ zeros matrix), and $A \otimes B$ for the Kronecker product of matrices A and B. Denote $A > 0$ (or $A \geq 0$) when matrix $A \in \mathbb{R}^{n\times n}$ is positive definite (or positive semi-definite).

A weighted undirected graph $\mathcal{G}$ is denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of nodes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, $A = [a_{i,j}] \in \mathbb{R}^{n\times n}$ is the *adjacency matrix* such that $a_{i,j} = a_{j,i} > 0$ if $\{j, i\} \in \mathcal{E}$ and $a_{i,j} = 0$ otherwise. The *Laplacian matrix* is $L_n = D - A$, where $D \in \mathbb{R}^{n\times n}$ is diagonal with $D_{i,i} = \sum_{j=1}^{n} a_{i,j}$, $i \in \{1, \ldots, n\}$. Specifically, if the undirected graph $\mathcal{G}$ is connected, then $L_n = L_n^{\mathrm{T}} \geq 0$, $\mathrm{rank}(L_n) = n-1$ and $\ker(L_n) = \{k1_n : k \in \mathbb{R}\}$ [1].

Consider a dynamical system

$$\dot{x}(t) = \phi(x(t)), \quad x(0) = x_0, \quad t \geq 0, \tag{1}$$

where $\phi : \mathbb{R}^q \to \mathbb{R}^q$ is Lipschitz continuous. Given a trajectory $x : [0, \infty) \to \mathbb{R}^q$ of (1), y is a positive limit point of $x(\cdot)$ if there is a positive increasing divergent sequence $\{t_i\}_{i=1}^{\infty} \subset \mathbb{R}$ such that $y = \lim_{i\to\infty} x(t_i)$, and a positive limit set of $x(\cdot)$ is the set of all positive limit points of $x(\cdot)$. A set $\mathcal{M}$ is said to be *positive invariant* with respect to (1) if $x(t) \in \mathcal{M}$ for all $t \geq 0$ and every $x_0 \in \mathcal{M}$.

The following result is a special case of [3, Proposition 3.1] and is used in the convergence analysis.

Lemma 2.1. Let $\mathcal{D}$ be a compact, positive invariant set with respect to (1), and $\phi(\cdot) \in \mathbb{R}^q$ be a solution of (1) with $\phi(0) = x_0 \in \mathcal{D}$. If z is a positive limit point of $\phi(\cdot)$ and a Lyapunov stable equilibrium of (1), then $z = \lim_{t\to\infty} \phi(t)$.

3. PROBLEM FORMULATION

In this section, we first introduce the problem of solving distributed linear algebraic equations and transform the linear algebraic equation into a distributed optimization formulation.

3.1. Distributed linear equation

Consider the problem of solving a linear algebraic equation

$$Ax = b, \tag{2}$$

where $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times q}$ are known and $x \in \mathbb{R}^q$ is the variable to be solved. In the general case, A may be singular or rectangular and there may sometimes be a unique solution, no solutions, or multiple solutions.

There are sufficient conditions that (2) is solvable and has exact solutions. Specifically,

- if A is nonsingular, (2) has a unique solution given by
$$x = A^{-1}b,$$
- if A is full column rank and $b \in \text{range}(A)$, (2) has a unique solution
$$x = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b,$$
- if A has a generalized inverse $X \in \mathbb{R}^{q\times m}$ satisfying $AXA = A$ and $AXb = b$, the solution is
$$x = Xb + (I_q - XA)y,$$
where $y \in \mathbb{R}^q$ is arbitrary.

Even though (2) may has no exact solutions, it has at least one least squares solution, which is a solution to the following optimization problem

$$\min_{x\in\mathbb{R}^q} \|Ax - b\|^2. \tag{3}$$

It is clear that x^* is a least squares solution to problem (2) if and only if

$$A^{\mathrm{T}}(Ax^* - b) = 0_q. \tag{4}$$

Let $\mathcal{G}$ be an undirected and connected graph composed of n agents. Define $A = [A_1, \dots, A_n] \in \mathbb{R}^{m\times q}$, $x \triangleq [x_1^{\mathrm{T}}, \dots, x_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^q$, $q = \sum_{i=1}^n q_i$ $b = \sum_{i=1}^n b_i \in \mathbb{R}^m$, $x_i \in \mathbb{R}^{q_i}$, and $A_i \in \mathbb{R}^{m\times q_i}$ for $i \in \{1, \dots, n\}$. We rewrite (2) as

$$\sum_{i=1}^n A_i x_i = \sum_{i=1}^n b_i.$$

The aim of this paper is to solve problem (2) using a multi-agent distributed algorithm such that

- agent i computes x_i;
- agent i knows whether the obtained solution is an exact solution;
- agent i only konws $A_i \in \mathbb{R}^{m\times q_i}$, $b_i \in \mathbb{R}^m$, and information from neighboring agents in network $\mathcal{G}$.

Remark 3.1. In this paper, the algorithm is required to solve (least squares) solutions of problem (2), while the results in [5, 6] need the existence of exact solutions. Compared with the results in [13] and [4], which solve the least squares solutions to linear equations, this paper considers the problem of verifying the solvability property, that is, whether the solution obtained is an exact solution.

Remark 3.2. Unlike the problem formulation in [4, 5, 6, 7, 12, 13], where each agent knows a few rows of A and b. This paper considers the problem that each agent knows a few columns of matrix A. Hence, this framework is useful when the matrix A has a large column number.

3.2. Problem transformation

In this subsection, we transform the distributed linear equation (3) into a distributed optimization formulation.

Define $z_0 = Ax - b$. Solving the least squares solution of (3) is equivalent to solving

$$\min_{(x,z_0)\in\mathbb{R}^{q\times m}} \|z_0\|^2, \quad \text{s. t. } z_0 = Ax - b.$$

Next, we decompose the coupling constraint $z_0 = Ax - b = \sum_{i=1}^n (A_i x_i - b_i)$ by augmenting the constraint using extra variables.

Define $y_i \in \mathbb{R}^m$ for $i \in \{1,\dots,n\}$ and let $a_{i,j}$ be the (i,j)th element of the adjacency matrix of graph $\mathcal{G}$, which is undirected and connected. If

$$\frac{1}{n} z_0 = A_i x_i - \sum_{j=1}^n a_{i,j}(y_i - y_j) - b_i, \quad i \in \{1,\dots,n\}, \tag{5}$$

one can obtain $z_0 = Ax - b$ by summing both sides of (5) from $i = 1$ to $i = n$. The converse is also true for some $y_i \in \mathbb{R}^m$ for $i \in \{1,\dots,n\}$. Define $L = L_n \otimes I_m$, where L_n is the Laplacian matrix of graph $\mathcal{G}$. Because

$$\mathrm{range}(L) = \ker(L)^\perp = \Big\{ w \in \mathbb{R}^{nm} : \sum_{i=1}^n w_i = 0_m,\ w = [w_1^{\mathrm{T}}, \dots, w_n^{\mathrm{T}}]^{\mathrm{T}} \Big\},$$

and $z_0 - Ax + b = 0_m$, $\big[(\frac{1}{n}z_0 - A_i x_i + b_i)^{\mathrm{T}}, \dots, (\frac{1}{n}z_0 - A_i x_i + b_i)^{\mathrm{T}}\big]^{\mathrm{T}} \in \mathrm{range}(L)$, and hence, there exists $y_i \in \mathbb{R}^m$ such that $\frac{1}{n}z_0 = A_i x_i - \sum_{i,j} a_{i,j}(y_i - y_j) - b_i$ for $i,j \in \{1,\dots,n\}$.

Let z_i be the estimate of z_0 by agent $i \in \{1,\dots,n\}$. Then, we define the following distributed optimization problem

$$\min_{x,y,z} \frac{1}{2}\|z\|^2, \quad \text{s. t. } Lz = 0_{nm},\ \overline{A}x - \overline{b} - Ly - \frac{1}{n} z = 0_{nm}, \tag{6}$$

where $x = [x_1^{\mathrm{T}}, \dots, x_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^q$, $y = [y_1^{\mathrm{T}}, \dots, y_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nm}$, $z = [z_1^{\mathrm{T}}, \dots, z_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nm}$, $\overline{A} = \mathrm{diag}\{A_1, \dots, A_n\}$, $\overline{b} = [b_1^{\mathrm{T}}, \dots, b_n^{\mathrm{T}}]^{\mathrm{T}}$, and $L = L_n \otimes I_m$.

The following result shows that the optimization problem (6) is equivalent to the linear equation (3) and gives a way to verify the solvability of (3).

Lemma 3.3. Suppose $\mathcal{G}$ is connected and undirected.

1) If $(x^*, y^*, z^*) \in \mathbb{R}^{q \times nm \times nm}$ is a solution to problem (6), then $x^* \in \mathbb{R}^q$ is a least squares solution to (3). In addition, $z^* = 1_n \otimes (Ax^* - b)$.

2) If $x^* \in \mathbb{R}^q$ is a least squares solution to problem (3), there exists $(y^*, z^*) \in \mathbb{R}^{nm \times nm}$ such that $(x^*, y^*, z^*) \in \mathbb{R}^{q \times nm \times nm}$ is a solution to problem (6).

Proof. 1) Suppose (x^*, y^*, z^*) is a solution to problem (6). We show that x^* is a least squares solution to (3).

By the KKT optimality condition (Theorem 3.25 of [11]), (x^*, y^*, z^*) is a solution to problem (6) if and only if

$$0_{nm} = Lz^*, \tag{7a}$$

$$0_{nm} = \overline{A}x^* - \overline{b} - Ly^* - \frac{1}{n}z^*, \tag{7b}$$

and there exist $\lambda^* \in \mathbb{R}^{nm}$ and $\mu^* \in \mathbb{R}^{nm}$ such that

$$0_q = -\overline{A}^{\mathrm{T}}\lambda^*, \tag{8a}$$

$$0_{nm} = L\lambda^*, \tag{8b}$$

$$0_{nm} = -z^* - L\mu^* + \frac{1}{n}\lambda^*. \tag{8c}$$

Due to (7a) and (8b), there exist $z_0^* \in \mathbb{R}^m$ and $\lambda_0^* \in \mathbb{R}^m$ such that $z^* = 1_n \otimes z_0^*$ and $\lambda^* = 1_n \otimes \lambda_0^*$. By left multiplying $1_n^{\mathrm{T}} \otimes I_m$ to (8c) and (7b), we have

$$z_0^* = Ax^* - b = \frac{1}{n}\lambda_0^*.$$

According to (8a), it suffices to prove $A_i^{\mathrm{T}}\lambda_0^* = 0_m$ and (4). Hence, x^* is a least squares solution to problem (3) and $z^* = 1_n \otimes (Ax^* - b)$.

2) Suppose x^* is a least squares solution to (3). We show that there exist $y^* \in \mathbb{R}^{nm}$ and $z^* \in \mathbb{R}^{nm}$ such that (x^*, y^*, z^*) is a solution to problem (6). To show this, we prove there exist $y^* \in \mathbb{R}^{nm}$, $z^* \in \mathbb{R}^{nm}$, $\lambda^* \in \mathbb{R}^{nm}$, and $\mu^* \in \mathbb{R}^{nm}$ such that the KKT condition holds, that is, (7) and (8) hold.

Define $z_0^* = Ax^* - b$ and $z^* = 1_n \otimes z_0^*$. It is clear that (7a) is true. Note that

$$\mathrm{range}(L) = \ker(L)^{\perp} = \Big\{ w \in \mathbb{R}^{nm} : \sum_{i=1}^{n} w_i = 0_m,\ w = [w_1^{\mathrm{T}}, \dots, w_n^{\mathrm{T}}]^{\mathrm{T}} \Big\},$$

and $\frac{1}{n}z^* - \overline{A}x^* + d \in \mathrm{range}(L)$. There exists $y^* \in \mathbb{R}^{nm}$ such that (7b) holds.

Let $\lambda^* = nz^* \in \mathbb{R}^{nm}$. (8b) holds and $\overline{A}^{\mathrm{T}}\lambda^* = n\overline{A}^{\mathrm{T}}z^* = n\begin{bmatrix} A_1^{\mathrm{T}}(Ax^*-b) \\ \vdots \\ A_n^{\mathrm{T}}(Ax^*-b) \end{bmatrix}$. Recall that (4) holds because x^* is a least squares solution to (3). For $i \in \{1,\ldots,n\}$, $A_i^{\mathrm{T}}(Ax^* - b) = 0_{q_i}$, and hence, (8a) holds.

Clearly, if $\mu^* = 1_n \otimes \mu_0^*$ with $\mu_0^* \in \mathbb{R}^m$, then (8c) holds.

To sum up, (7) and (8) hold, and hence, (x^*, y^*, z^*) is a solution to problem (6).

□

Remark 3.4. Problem (6) is a distributed optimization problem, whose solutions are equivalent to the least squares solutions to linear equation (2) and satisfy $z_i^* = Ax^* - b$ for all $i \in \{1,\ldots,n\}$. Note that linear equation (2) always has a least squares solution. Problem (6) always has solutions. Clearly, if a solution of problem (6) satisfies $z_i^* = Ax^* - b = 0_{nm}$, then x^* is an exact solution to linear equation (2). Hence, agent i in this formulation is able to testify the solvability of linear equation (2) by checking the variable z_i^* for all $i \in \{1,\ldots,n\}$.

4. DISTRIBUTED ALGORITHM

In this section, we propose a distributed algorithm for linear equation (2) based on the optimization problem (6) and present the theoretical proof for the correctness and convergence properties of the algorithm.

We propose a distributed algorithm for the linear algebraic equation computation as follows:

$$\dot{x}_i(t) = -A_i^{\mathrm{T}}\lambda_i(t), \tag{9a}$$

$$\dot{y}_i(t) = \sum_{j=1}^{n} a_{i,j}(\lambda_i(t) - \lambda_j(t)), \tag{9b}$$

$$\dot{z}_i(t) = -z_i(t) - \sum_{j=1}^{n} a_{i,j}(z_i(t) - z_j(t)) - \sum_{j=1}^{n} a_{i,j}(\mu_i(t) - \mu_j(t)) + \frac{1}{n}\lambda_i(t), \tag{9c}$$

$$\begin{aligned}\dot{\lambda}_i(t) = {} & A_i x_i(t) - b_i - \sum_{j=1}^{n} a_{i,j}(y_i(t) - y_j(t)) - \frac{1}{n} z_i(t) \\ & - \sum_{j=1}^{n} a_{i,j}(\lambda_i(t) - \lambda_j(t)) - A_i A_i^{\mathrm{T}}\lambda_i(t), \end{aligned}\tag{9d}$$

$$\dot{\mu}_i(t) = \sum_{j=1}^{n} a_{i,j}(z_i(t) - z_j(t)), \tag{9e}$$

where $t \geq 0$, $i \in \{1,\ldots,n\}$, $x_i(0) = x_{i,0} \in \mathbb{R}^{q_i}$, $y_i(0) = y_{i,0} \in \mathbb{R}^m$, $z_i(0) = z_{i,0} \in \mathbb{R}^m$, $\lambda_i(0) = \lambda_{i,0} \in \mathbb{R}^m$, $\mu_i(0) = \mu_{i,0} \in \mathbb{R}^m$ are the initial condition, $x_i(t)$, $y_i(t)$, and $z_i(t)$ are the estimates of the solution to problem (6) by agent i at time t, $\lambda_i(t)$, $\mu_i(t)$ are the estimates of Lagrangian multipliers.

Remark 4.1. In this algorithm, $x(t)$ converges to a least squares solution to linear equation (2) and $\lim_{t\to\infty} z_i(t) = \lim_{t\to\infty}(Ax(t)-b)$ for all $i \in \{1,\dots,n\}$. Hence, agent i is able to verify the exactness of solution $\lim_{t\to\infty} x(t)$ to linear equation (2) by checking the value of $\lim_{t\to\infty} z_i(t)$.

A compact form of the algorithm (9) is given as follows:

$$\dot{x} = -\overline{A}^{\mathrm{T}}\lambda, \tag{10a}$$

$$\dot{y} = L\lambda, \tag{10b}$$

$$\dot{z} = -z - Lz - L\mu + \frac{1}{n}\lambda, \tag{10c}$$

$$\dot{\lambda} = \overline{A}x - \overline{b} - Ly - \frac{1}{n}z - L\lambda - \overline{A}\overline{A}^{\mathrm{T}}\lambda, \tag{10d}$$

$$\dot{\mu} = Lz, \tag{10e}$$

where $x(0) = x_0 \in \mathbb{R}^q$, $y(0) = y_0 \in \mathbb{R}^{nm}$, $z(0) = z_0 \in \mathbb{R}^{nm}$, $\lambda(0) = \lambda_0 \in \mathbb{R}^{nm}$, $\mu(0) = \mu_0 \in \mathbb{R}^{nm}$, $L = L_n \otimes I_m$, and L_n is the Laplacian matrix of graph $\mathcal{G}$.

Remark 4.2. Algorithm (10) is inspired by the saddle point dynamics of augmented Lagrangian functions. To be specific, define the augmented Lagrangian function

$$\begin{aligned} L(x,y,z,\lambda,\mu) \;&=\; \frac{1}{2}\|z\|^2 + \lambda^{\mathrm{T}}(\overline{A}x - \overline{b} - Ly - \frac{1}{n}z) + \mu^{\mathrm{T}}Lz + \frac{1}{2}z^{\mathrm{T}}Lz \\ &\quad -\frac{1}{2}\lambda^{\mathrm{T}}L\lambda - \frac{1}{2}\|\overline{A}^{\mathrm{T}}\lambda\|^2, \end{aligned} \tag{11}$$

where $\overline{b} = [b_1^{\mathrm{T}},\dots,b_n^{\mathrm{T}}]^{\mathrm{T}}$, $b = \sum_{i=1}^n b_i \in \mathbb{R}^m$, $\lambda = [\lambda_1^{\mathrm{T}},\dots,\lambda_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nm}$ and $\mu = [\mu_1^{\mathrm{T}},\dots,\mu_n^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nm}$. Algorithm (10) is obtained by the primal-dual saddle point dynamics $\dot{x} = -\nabla_x L(x,y,z,\lambda,\mu)$, $\dot{y} = -\nabla_y L(x,y,z,\lambda,\mu)$, $\dot{z} = -\nabla_z L(x,y,z,\lambda,\mu)$, $\dot{\lambda} = \nabla_\lambda L(x,y,z,\lambda,\mu)$, and $\dot{\mu} = \nabla_\mu L(x,y,z,\lambda,\mu)$.

The following result establishes the relationship between the equilibria of algorithm (10) and solutions to problem (6).

Lemma 4.3. Suppose that $\mathcal{G}$ is connected and undirected. $(x^*,y^*,z^*) \in \mathbb{R}^{q\times nm\times nm}$ is a solution of problem (6) if and only if there exist $\lambda^* \in \mathbb{R}^{nm}$ and $\mu^* \in \mathbb{R}^{nm}$ such that $(x^*,y^*,z^*,\lambda^*,\mu^*)$ is an equilibrium of algorithm (10).

Proof. According to the KKT theorem, $(x^*,y^*,z^*) \in \mathbb{R}^{q\times nm\times nm}$ is the solution of problem (6) if and only if there exist λ^* and μ^* such that (7) and (8) hold, equivalently, $(x^*,y^*,z^*,\lambda^*,\mu^*)$ is an equilibrium of algorithm (10). □

The following theorem shows the convergence of the algorithm to a least squares solution of problem (2) with solvability verification.

Theorem 4.4. Assume $\mathcal{G}$ is connected and undirected. Let $(x(t),y(t),z(t),\lambda(t),\mu(t))$, $t \geq 0$, be the trajectory of (10).

1) Every equilibrium of (10) is Lyapunov stable and $(x(t), y(t), z(t), \lambda(t), \mu(t))$ is bounded for any initial condition;

2) $(x(t), y(t), z(t), \lambda(t), \mu(t))$ converges to an equilibrium of (10);

3) $\lim_{t\to\infty} x(t)$ is a least squares solution to problem (3) and

$$\lim_{t\to\infty} z_i(t) = \lim_{t\to\infty}(Ax(t) - b)$$

for all $i \in \{1, \ldots, n\}$;

4) if, in addition, $\lim_{t\to\infty} z_i(t) = 0_m$, then $\lim_{t\to\infty} x(t)$ is an exact solution to problem (3).

P r o o f. 1) Let $(x^*, y^*, z^*, \lambda^*, \mu^*)$ be an equilibrium of algorithm (10). Define a positive definite function as

$$V(x, y, z, \lambda, \mu) = \frac{1}{2}\|x - x^*\|^2 + \frac{1}{2}\|y - y^*\|^2 + \frac{1}{2}\|z - z^*\|^2 + \frac{1}{2}\|\lambda - \lambda^*\|^2 + \frac{1}{2}\|\mu - \mu^*\|^2.$$

By (7), (8), and (10), we have

$$\begin{aligned}
\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|x - x^*\|^2 &= -(x - x^*)^{\mathrm{T}}\overline{A}^{\mathrm{T}}(\lambda - \lambda^*), \\
\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|y - y^*\|^2 &= (y - y^*)^{\mathrm{T}}L(\lambda - \lambda^*), \\
\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|z - z^*\|^2 &= (z - z^*)^{\mathrm{T}}\big(-z - Lz - L\mu + \frac{1}{n}\lambda - (-z^* - L\mu^* + \frac{1}{n}\lambda^*)\big) \\
&= -\|z - z^*\|^2 - z^{\mathrm{T}}Lz - (z - z^*)^{\mathrm{T}}L(\mu - \mu^*) \\
&\quad + \frac{1}{n}(z - z^*)^{\mathrm{T}}(\lambda - \lambda^*), \\
\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\lambda - \lambda^*\|^2 &= (\lambda - \lambda^*)^{\mathrm{T}}\big(\overline{A}x - \overline{b} - Ly - \frac{1}{n}z - L\lambda - \overline{A}\overline{A}^{\mathrm{T}}\lambda \\
&\quad -(\overline{A}x^* - \overline{b} - Ly^* - \frac{1}{n}z^*)\big) \\
&= (\lambda - \lambda^*)^{\mathrm{T}}\overline{A}(x - x^*) - (\lambda - \lambda^*)^{\mathrm{T}}L(y - y^*) \\
&\quad - \frac{1}{n}(\lambda - \lambda^*)^{\mathrm{T}}(z - z^*) - \lambda^{\mathrm{T}}L\lambda - \|\overline{A}^{\mathrm{T}}\lambda\|^2, \\
\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\mu - \mu^*\|^2 &= (\mu - \mu^*)^{\mathrm{T}}L(z - z^*).
\end{aligned}$$

To sum up, the function derivative $\dot{V}(\cdot)$ along the trajectory of algorithm (10) is given by

$$\dot{V}(x, y, z, \lambda, \mu) = -\|z - z^*\|^2 - z^{\mathrm{T}}Lz - \lambda^{\mathrm{T}}L\lambda - \|\overline{A}^{\mathrm{T}}\lambda\|^2 \leq 0. \tag{12}$$

Hence, $(x(t), y(t), z(t), \lambda(t), \mu(t))$ is bounded for all $t \geq 0$ and $(x^*, y^*, z^*, \lambda^*, \mu^*)$ is a Lyapunov equilibrium of algorithm (10).

2) Define the set

$$\begin{aligned}\mathcal{R} &= \big\{(x,y,z,\lambda,\mu) : \dot{V}(x,y,z,\lambda,\mu)=0\big\}\\ &\subset \big\{(x,y,z,\lambda,\mu) : z=z^*,\ Lz=L\lambda=0_{nm},\ \overline{A}^{\mathrm{T}}\lambda=0_q\big\}.\end{aligned}$$

Let $\mathcal{M}$ be the largest invariant subset of $\overline{\mathcal{R}}$. It follows from the invariance principle (Theorem 2.41 of [2]) that $(x(t),y(t),z(t),\lambda(t),\mu(t))\to\mathcal{M}$ as $t\to\infty$ and $\mathcal{M}$ is positive invariant.

Because $\mathcal{M}$ is positive invariant, the trajectory $(\overline{x}(t),\overline{y}(t),\overline{z}(t),\overline{\lambda}(t),\overline{\mu}(t))$ of (10) satisfies that $(\overline{x}(t),\overline{y}(t),\overline{z}(t),\overline{\lambda}(t),\overline{\mu}(t))\in\mathcal{M}$ for all $t\geq 0$ if $(\overline{x}(0),\overline{y}(0),\overline{z}(0),\overline{\lambda}(0),\overline{\mu}(0))\in\mathcal{M}$. Hence, we have $\dot{\overline{x}}(t)\equiv 0_q$, $\dot{\overline{y}}(t)\equiv 0_{nm}$, $\dot{\overline{z}}(t)\equiv 0_{nm}$, $\dot{\overline{\mu}}(t)\equiv 0_{nm}$, and

$$\dot{\overline{\lambda}}(t)\equiv \overline{A}\overline{x}(0)-\overline{b}-L\overline{y}(0)-\frac{1}{n}\overline{z}(0).$$

Suppose $\dot{\overline{\lambda}}(t)\neq 0_{nm}$, then $\overline{\lambda}(t)\to\infty$ as $t\to\infty$, which contradicts the boundedness of the trajectory. Hence, $\dot{\overline{\lambda}}(t)\equiv 0_{nm}$ and

$$\mathcal{M}\subset\big\{(x,y,z,\lambda,\mu) : \dot{x}=0_q,\ \dot{y}=\dot{z}=\dot{\lambda}=\dot{\mu}=0_{nm}\big\}.$$

As a result, any point in $\mathcal{M}$ is an equilibrium point of algorithm (10). By part 1), any point in $\mathcal{M}$ is a Lyapunov stable equilibrium of algorithm (10). By Lemma 2.1, $(x(t),y(t),z(t),\lambda(t),\mu(t))$ converges to a point in $\mathcal{M}$ as $t\to\infty$.

3) Note that every point in $\mathcal{M}$ is a Lyapunov stable equilibrium and the trajectory $(x(t),y(t),z(t),\lambda(t),\mu(t))$ converges to a point in $\mathcal{M}$ as $t\to\infty$. It follows from Lemmas 3.3 and 4.3 that $\lim_{t\to\infty}x(t)$ is a least squares solution to problem (3).

By Lemma 3.3 1), it is straightforward that $\lim_{t\to\infty}z_i(t)=\lim_{t\to\infty}(Ax(t)-b)$ for all $i\in\{1,\dots,n\}$.

4) Clearly, if $\lim_{t\to\infty}z_i(t)=\lim_{t\to\infty}(Ax(t)-b)=0_m$ for all $i\in\{1,\dots,n\}$, then $\lim_{t\to\infty}x(t)$ is an exact solution to problem (3). ☐

Remark 4.5. In the proposed algorithm, variable $x(t)$ is the estimate of a least squares solution x^* to problem (2) and variable $z_i(t)$ is the estimate of Ax^*-b by agent $i\in\{1,\dots,n\}$. To be specific, $x(t)$ converges to a least squares solution x^* and $\lim_{t\to\infty}z_i(t)=Ax^*-b$ for $i\in\{1,\dots,n\}$. If $\lim_{t\to\infty}z_i(t)=0_m$, then $\lim_{t\to\infty}x(t)$ is an exact solution and linear equation (2) is solvable. Hence, agent i solves a least squares solution to linear equation (2) and knows the solvability of linear equation (2) by checking the value of $\lim_{t\to\infty}z_i(t)$ for $i\in\{1,\dots,n\}$.

5. NUMERICAL RESULT

In this section, we give a specific example to verify our proposed distributed algorithm. Consider the linear algebraic equation

$$Ax=\sum_{i=1}^{4}A_ix_i=b,\quad x_1\in\mathbb{R}^2,\quad x_i\in\mathbb{R},\quad i\in\{2,3,4\},$$

where $A = [A_1\ A_2\ A_3\ A_4] \in \mathbb{R}^{4\times 5}$, $x \in \mathbb{R}^5$, $b = \left[11,\ 8,\ 10,\ 12\right]^{\mathrm{T}}$, and A is defined in two cases

$$\text{(i)}\quad A_1 = \begin{bmatrix} 2 & 2 \\ 2 & 3 \\ 1 & 4 \\ 2 & 1 \end{bmatrix},\ A_2 = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 4 \end{bmatrix},\ A_3 = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 2 \end{bmatrix},\ A_4 = \begin{bmatrix} 5 \\ 2 \\ 1 \\ 3 \end{bmatrix}.$$

$$\text{(ii)}\quad A_1 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 1 & 4 \\ 2 & 1 \end{bmatrix},\ A_2 = \begin{bmatrix} 3 \\ 3 \\ 2 \\ 4 \end{bmatrix},\ A_3 = \begin{bmatrix} 3 \\ 3 \\ 1 \\ 2 \end{bmatrix},\ A_4 = \begin{bmatrix} 5 \\ 5 \\ 1 \\ 3 \end{bmatrix}.$$

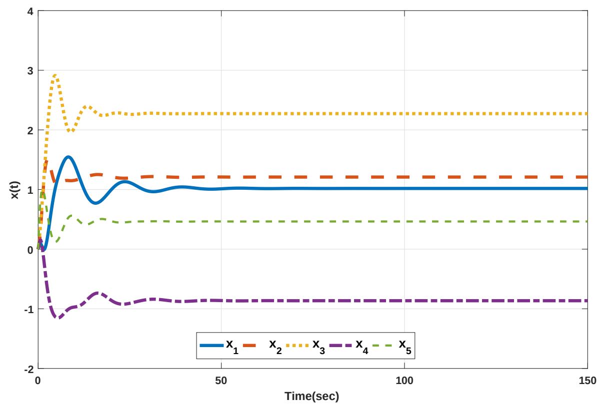

Fig. 1. The trajectory of $x(t)$ calculated.

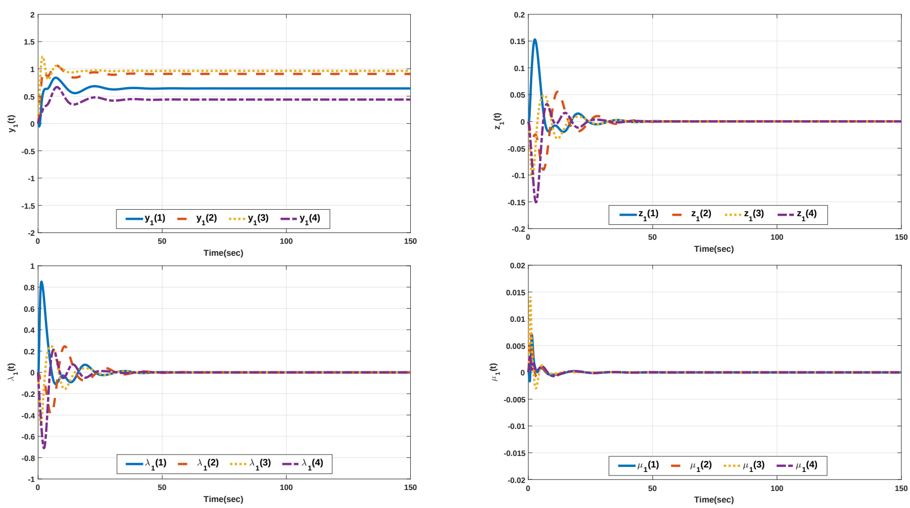

Fig. 2. The trajectories of y_1, z_1, λ_1, and μ_1 of agent 1 calculated.

In case (i), the linear algebraic equation has a unique solution

$$x = \left[1.0179,\ 1.2092,\ 2.2729,\ -0.8645,\ 0.4641\right]^{\mathrm{T}}.$$

The simulation results are shown in Figures 1−5. Figure 1 shows the trajectory of $x(t)$ converges to the solution, while Figures 2−5 show the boundedness of algorithm variables and that the obtained solution is an exact solution as the trajectory of $z_i(t)$ tend to 0_4 for $i \in \{1, 2, 3, 4\}$.

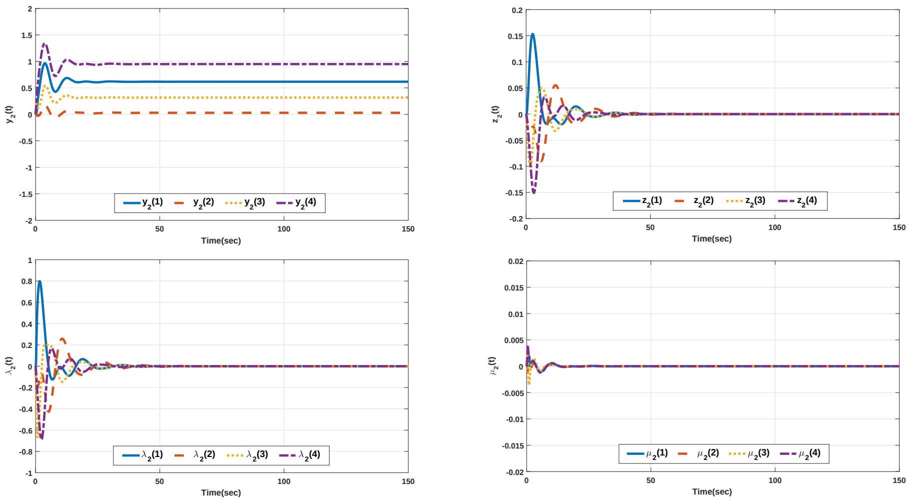

Fig. 3. The trajectories of y_2, z_2, λ_2, and μ_2 of agent 2 calculated.

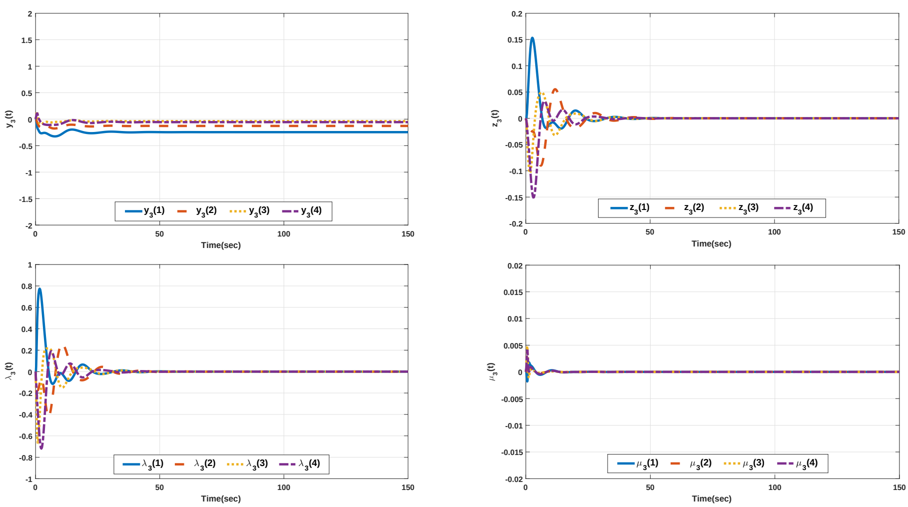

Fig. 4. The trajectories of y_3, z_3, λ_3, and μ_3 of agent 3 calculated.

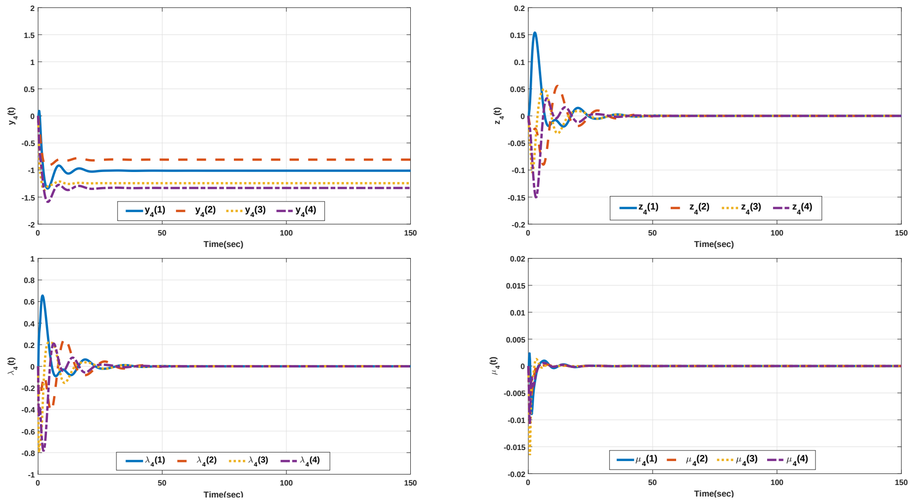

Fig. 5. The trajectories of y_4, z_4, λ_4, and μ_4 of agent 4 calculated.

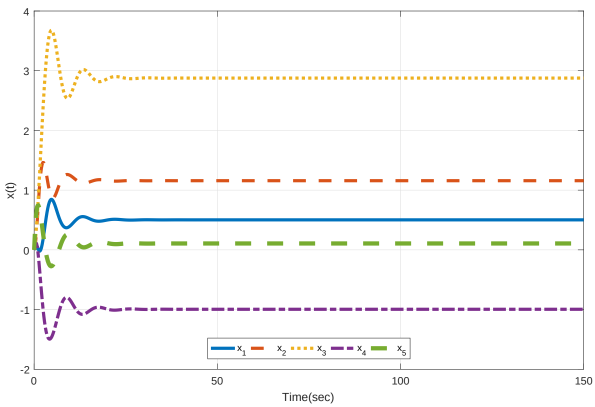

Fig. 6. The trajectory of of $x(t)$ calculated.

In case (ii), the linear algebraic equation has no exact solutions and our algorithm gives a least squares solution. The simulation results are shown in Figures 6 – 10. Figure 6 shows the trajectory of $x(t)$ converges to a least squares solution

$$x = \left[0.5024,\ 1.1581,\ 2.8767,\ -0.9953,\ 0.1070\right]^{\mathrm{T}}.$$

Figures 7 – 10 show the trajectories of $y_i(t)$, $\lambda_i(t)$, $z_i(t)$, and $Ax(t) - b$. Because $z_i(t)$ and $Ax(t) - b$ do not converge to 0_4 for $i \in \{1, 2, 3, 4\}$, the obtained solution is not an exact solution.

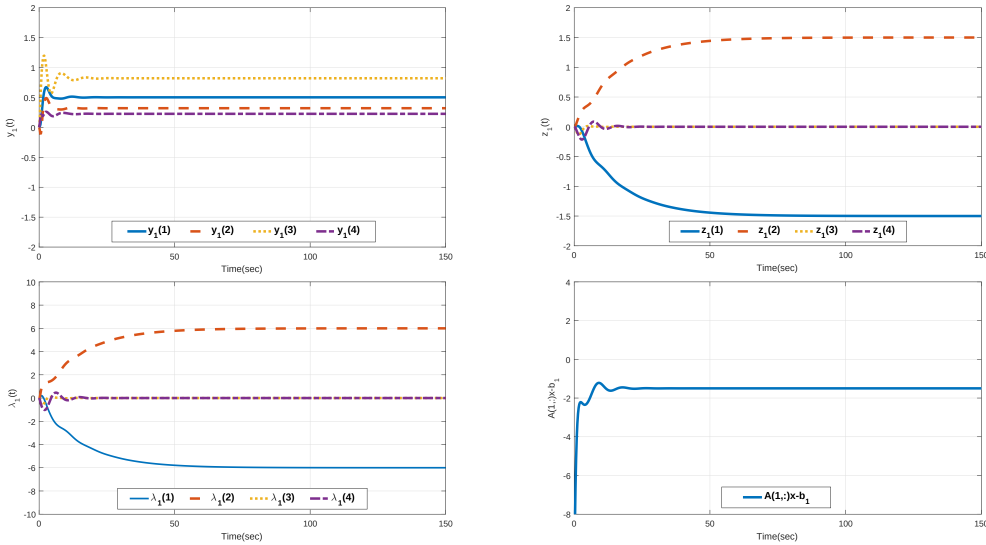

Fig. 7. The trajectories of y_1, z_1, λ_1, and $A(1,:)x - b_1$ of agent 1 calculated.

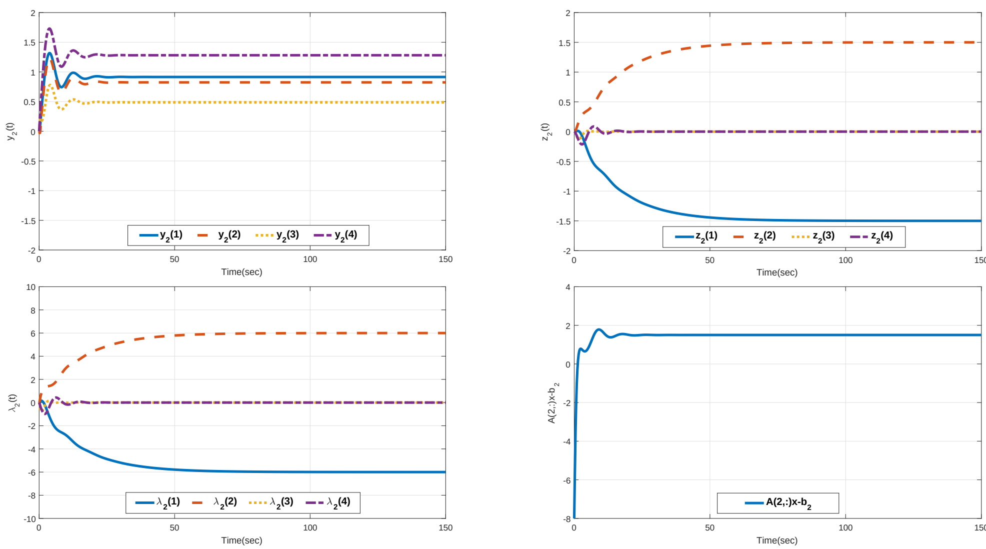

Fig. 8. The trajectories of y_2, z_2, λ_2, and $A(2,:)x - b_2$ of agent 2 calculated.

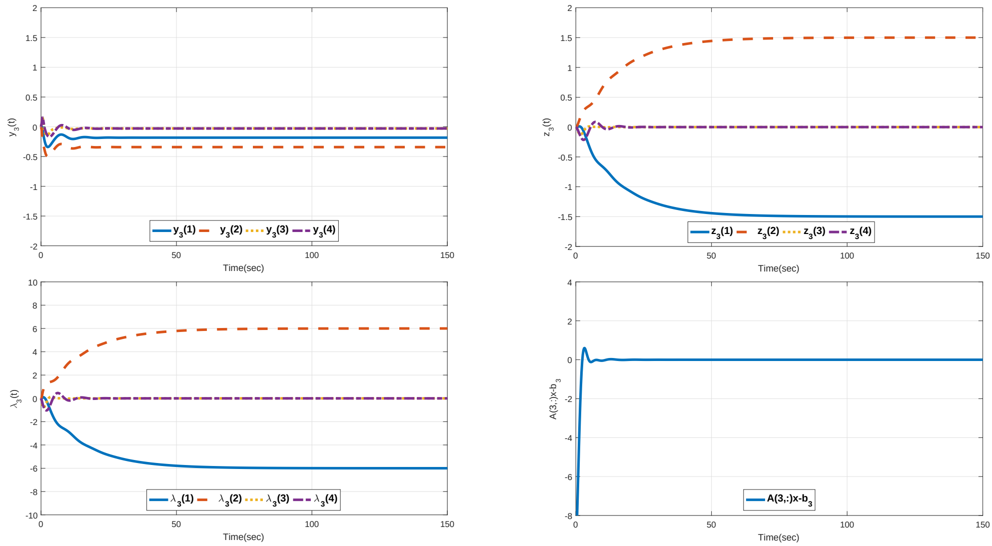

Fig. 9. The trajectories of y_3, z_3, λ_3, and $A(3,:)x - b_3$ of agent 3 calculated.

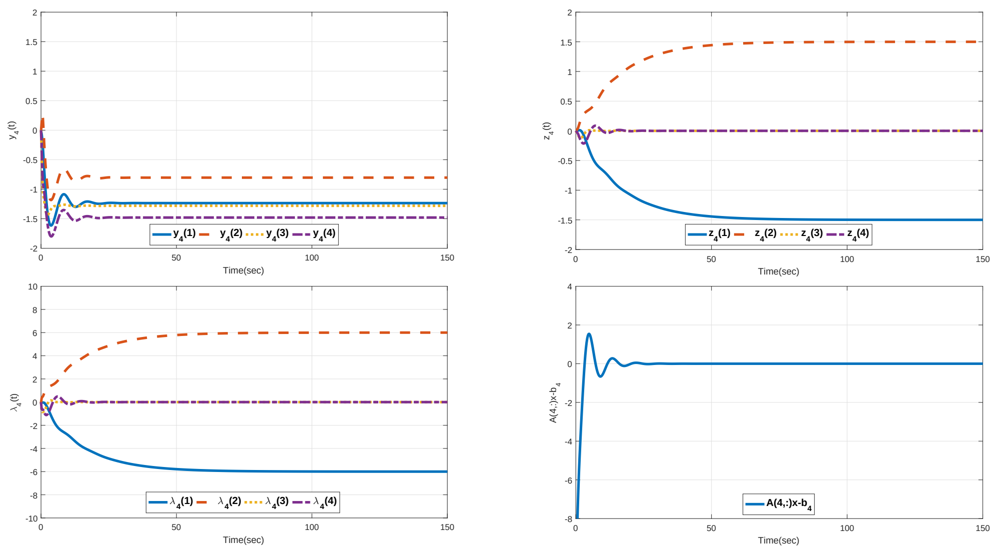

Fig. 10. The trajectories of y_4, z_4, λ_4, and $A(4,:)x - b_4$ of agent 4 calculated.

6. CONCLUSIONS

This paper investigated the distributed computation of a linear algebraic equation $Ax = b$ over a multi-agent network, where each agent knows a few columns of matrix A, with solvability verification. Based on the optimization reformulation of the problem, this paper proposed a novel distributed algorithm solving least squares solutions to linear equations with solvability verification properties. Under mild graph conditions, it was shown that the agents find a least squares solution for the linear algebraic equation and the agents are able to verify whether the obtained solution is exact under any initial condition. Finally, a numerical simulation illustrated the performance of the proposed algorithm.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (61603378, 61573344, and 61333001) and China Postdoctoral Science Foundation (2015M581190).

(Received July 17, 2017)

REFERENCES

[1] C. Godsil and G. F. Royle: Algebraic Graph Theory. Springer-Verlag, New York 2001. DOI:10.1007/978-1-4613-0163-9

[2] W. M. Haddad and V. Chellaboina: Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach. Princeton University Press, New Jersey 2008.

[3] Q. Hui, W. M. Haddad, and S. P. Bhat: Semistability, finite-time stability, differential inclusions, and discontinuous dynamical systems having a continuum of equilibria. IEEE Trans. Automat. Control *54* (2009), 2465–2470. DOI:10.1109/tac.2009.2029397

[4] Y. Liu, C. Lageman, B. Anderson, and G. Shi: An Arrow-Hurwicz-Uzawa type flow as least squares solver for network linear equations. arXiv:1701.03908v1.

[5] J. Liu, A. S. Morse, A. Nedic, and T. Basar: Exponential convergence of a distributed algorithm for solving linear algebraic equations. Automatica *83* (2017), 37–46. DOI:10.1016/j.automatica.2017.05.004

[6] J. Liu, S. Mou, and A. S. Morse: Asynchronous distributed algorithms for solving linear algebraic equations. IEEE Trans. Automat Control *PP* (2017), 99, 1–1. DOI:10.1109/TAC.2017.2714645

[7] S. Mou, J. Liu, and A. S. Morse: A distributed algorithm for solving a linear algebraic equation. IEEE Trans. Automat. Control *60* (2015), 2863–2878. DOI:10.1109/tac.2015.2414771

[8] A. Nedic, A. Ozdaglar, and P. A. Parrilo: Constrained consensus and optimization in multi-agent networks. IEEE Trans. Automat. Control *55* (2010), 922–938. DOI:10.1109/tac.2010.2041686

[9] W. Ni and X. Wang: Averaging approach to distributed convex optimization for continuous-time multi-agent systems. Kybernetika *52* (2016), 898–913. DOI:10.14736/kyb-2016-6-0898

[10] Z. Qiu, S. Liu, and L. Xie: Distributed constrained optimal consensus of multi-agent systems. Automatica *68* (2016), 209–215. DOI:10.1016/j.automatica.2016.01.055

[11] A. Ruszczynski: Nonlinear Optimization. Princeton University Press, New Jersey 2006.

[12] G. Shi and B. D. O. Anderson: Distributed network flows solving linear algebraic equations. In: American Control Conference, Boston 2016, pp. 2864–2869. DOI:10.1109/acc.2016.7525353

[13] G. Shi, B. D. O. Anderson, and U. Helmke: Network flows that solve linear equations. IEEE Trans. Automat. Control *62* (2017), 2659–2764.DOI:10.1109/tac.2016.2612819

[14] G. Shi and K. H. Johansson: Randomized optimal consensus of multi-agent systems. Automatica *48* (2012), 3018–3030. DOI:10.1016/j.automatica.2012.08.018

[15] P. Yi, Y. Hong, and F. Liu: Distributed gradient algorithm for constrained optimization with application to load sharing in power systems. Systems Control Lett. *83* (2015), 45–52. DOI:10.1016/j.sysconle.2015.06.006

[16] X. Zeng and Q. Hui: Energy-event-triggered hybrid supervisory control for cyber-physical network systems. IEEE Trans. Automat. Control *60* (2015), 3083–3088. DOI:10.1109/tac.2015.2409900

[17] X. Zeng, P. Yi, and Y. Hong: Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach. IEEE Trans. Automat. Control *62* (2017), 5227–5233.DOI:10.1109/tac.2016.2628807

Xianlin Zeng, Beijing Institute of Technology, Beijing. P. R. China.
e-mail: xianlin.zeng@bit.edu.cn

Kai Cao, University of Chinese Academy of Sciences, Beijing. P. R. China.
e-mail: ck420@mail.ustc.edu.cn