



# The APC Algorithm of Solving Large-Scale Linear Systems: A Generalized Analysis

Jiyan Zhang, Yue Xue, Yuan Qi<sup>(✉)</sup>, and Jiale Wang

Beijing University of Posts and Telecommunications, Beijing, China  
{zhangjiyan,runfordream,qiyuan,wangjiale}@bupt.edu.cn

**Abstract.** A new algorithm called accelerated projection-based consensus (APC) has recently emerged as a promising approach to solve large-scale systems of linear equations in a distributed fashion. The algorithm uses a federated computational architecture, and attracts increasing research interest; however, its performance analysis is still incomplete, e.g., the error performance under noisy condition has not yet been investigated. In this paper, we focus on providing a generalized analysis by the use of the linear system theory, such that the error performance of the APC algorithm for solving linear systems in presence of additive noise can be clarified. We specifically provide a closed-form expression of the error of solution attained by the APC algorithm. Numerical results demonstrate the error performance of the APC algorithm, validating the presented analysis.

**Keywords:** Large-scale systems · linear equations · distributed algorithms · performance analysis

## 1 Introduction

Solving large-scale systems of linear equations is a fundamental problem in various signal processing, control theory, and machine learning applications [1–5]. Recently, *Azizan-Ruhi*, *Avestimehrm*, and *Hassibi* developed the APC algorithm for distributed solution of large-scale systems of linear equations [1]. APC offers significant speed-up relative to other distributed methods such as the well-known alternating direction method of multipliers (ADMM) method [1], making the algorithm appealing for many applications [1, 5, 6]. The APC algorithm was used by the probabilistic load flow calculation of power systems for the privacy-preserving purpose [6]. Inspired by the APC algorithm, our previous work designed a distributed channel estimation algorithm for mmWave massive MIMO communication systems [5].

This work is sponsored in part by the National Natural Science Foundation of China (grant no. 61971058, 61801048, 61631004, 62071063) and Beijing Natural Science Foundation (grant no. L202014, L192002).

One distinguishing attribute of the APC algorithm is that its computational architecture is a *federated* architecture (also known as the *server-worker* distributed architecture [7,8]). A federated architecture comprises of one server (master) and multiple agents (workers) [7–12]. Research attention has increasingly focused on the development of novel algorithms to distributed computation [2,7–13], including those with the federated architecture [7–12]; thus, the APC algorithm also attracts increasing research interest [3,7–13]. There are a number of references on the topics related to the APC algorithm, e.g., distributed algorithms for systems of linear equations [2], for state estimation [3], for gradient-descent method [7,9], for linear transforms [10,11], for coded matrix multiplication [12], and for phase retrieval [13].

However, the seminal work on the APC algorithm [1] only considers a model setting wherein the system is free from noise, while noise is an unavoidable factor in real systems. It is still unconfirmed whether the APC algorithm can effectively solve linear systems with the existence of noise; in other words, researchers still lack a clear understanding of the robustness of the algorithm under the noisy condition. Therefore, this study aims at deriving the analytical results by considering the additive noise. Specifically, we generalize the analysis of the APC algorithm by utilizing the linear system theory (see Theorem 1), such that the error performance of the algorithm for solving linear systems is clarified in presence of noise (see Theorem 3). Note that the study in this paper derives analytical results which can be reduced to those in [1], by letting the elements of noise vector be zeros; in this sense, the analysis conducted in this paper is the generalized one. Moreover, we provide closed-form expressions to two important parameters of the APC algorithm (see Remark 1).

*Notations* : Let  $\mathbb{C}$  be the complex field. We write  $\|\cdot\|_2$  for the  $\ell_2$  norm of a vector. For a matrix  $\mathbf{A}$ ,  $\lambda_{min}(\mathbf{A})$  and  $\lambda_{max}(\mathbf{A})$  are the smallest and largest eigenvalues, respectively, and  $\rho(\mathbf{A})$  is the spectral radius of  $\mathbf{A}$ , i.e., the largest absolute value of its eigenvalues. When  $\mathbf{A}$  has full column rank, we let  $\mathbf{A}^\dagger = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H$  be the Moore-Penrose pseudoinverse of  $\mathbf{A}$ . If  $\text{span}(\mathbf{A})$  is the span of columns in  $\mathbf{A}$ , then  $\mathbf{P}_\mathbf{A}^\perp = \mathbf{I} - \mathbf{A}\mathbf{A}^\dagger$  is the projection onto the orthogonal complement of  $\text{span}(\mathbf{A})$ . For a block diagonal matrix  $\mathbf{A} = \text{diag}(\mathbf{A}_{11}, \dots, \mathbf{A}_{kk})$ , one can write  $\mathbf{A} = \mathbf{A}_{11} \oplus \mathbf{A}_{22} \oplus \dots \oplus \mathbf{A}_{kk}$  that is the direct sum of the matrices  $\mathbf{A}_{11}, \dots, \mathbf{A}_{kk}$  [14]. Let  $\mathbf{I}_N \in \mathbb{C}^{N \times N}$  be the identity matrix and  $\mathbf{0}_N \in \mathbb{C}^N$  ( $\mathbf{O}_N \in \mathbb{C}^{N \times N}$ ) be the all-zero vector (matrix).

## 2 Model

Consider the problem of solving a large-scale system of linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{y}, \tag{1}$$

where  $\mathbf{A} \in \mathbb{C}^{M \times s}$  has full column rank,  $\mathbf{x} \in \mathbb{C}^s$ , and

$$\mathbf{y} = \mathbf{A}\mathbf{x}^* + \tilde{\mathbf{w}} \in \mathbb{C}^M. \tag{2}$$

As usual,  $\mathbf{A}$  and  $\mathbf{y}$  are known matrix and vector, respectively. While  $\tilde{\mathbf{w}} = [\tilde{w}_1, \dots, \tilde{w}_M]^T$  is an unknown noise vector (not necessarily Gaussian noise) in this paper.

We can apply the APC algorithm (presented in Algorithm 1) to find the solution to (1), which will obtain an estimate of  $\mathbf{x}^*$ . The formal description of APC [1] is provided as Algorithm 1. The implementation of the APC algorithm requires  $M$  distributed agents and one server. Every agent computes a solution to its own private equation, while all these agents can run in a parallel fashion. The server calculates the average of these  $M$  solutions and regards it as the global solution.

The APC algorithm runs for  $T$  iterations. Initially, the algorithm executes the following computations:  $\mathbf{x}_\ell(0) = \mathbf{A}_\ell^H (\mathbf{A}_\ell \mathbf{A}_\ell^H)^{-1} \mathbf{y}_\ell$  in every agent, and  $\bar{\mathbf{x}}(0) = \frac{1}{M} \sum_{\ell=1}^M \mathbf{x}_\ell(0)$  in the server. After that, the computations are performed iteratively; that is, for  $t = 0, \dots, T-1$ , the APC algorithm goes through the following steps:  $\mathbf{x}_\ell(t+1) = \mathbf{x}_\ell(t) + \gamma \mathbf{P}_\ell^\perp (\bar{\mathbf{x}}(t) - \mathbf{x}_\ell(t))$  and  $\bar{\mathbf{x}}(t+1) = \frac{\eta}{M} \sum_{\ell=1}^M \mathbf{x}_\ell(t+1) + (1-\eta)\bar{\mathbf{x}}(t)$ , with the agents and server, respectively.

---

**Algorithm 1:** The APC Algorithm for Finding the Solution  $\tilde{\mathbf{x}}$  to  $\mathbf{A}\mathbf{x} = \mathbf{y}$

---

**Input:**  $\mathbf{y} = [y_1, \dots, y_M]^T \in \mathbb{C}^M$ ,  $\mathbf{A} \in \mathbb{C}^{M \times s}$ , and  $T$ .

**Output:**  $\tilde{\mathbf{z}} \in \mathbb{C}^s$ .

```

1 Let  $\mathbf{A}_\ell \in \mathbb{C}^{1 \times s}$  denote the  $\ell$ -th row of  $\mathbf{A}$ ;
2  $t = 0$ ;
  /* Computations in every agent: */
3 for  $\ell = 1 : M$  do
4    $\mathbf{P}_\ell^\perp = \mathbf{I}_s - \mathbf{A}_\ell^H (\mathbf{A}_\ell \mathbf{A}_\ell^H)^{-1} \mathbf{A}_\ell$ ;
5    $\mathbf{z}_\ell(0) = \mathbf{A}_\ell^H (\mathbf{A}_\ell \mathbf{A}_\ell^H)^{-1} \mathbf{y}_\ell$ ; // Initializing each  $\mathbf{z}_\ell(0)$ 
6 end
  /* Computations in the server: */
7  $\bar{\mathbf{z}}(0) = \frac{1}{M} \sum_{\ell=1}^M \mathbf{z}_\ell(0)$ ;
8 while  $t \leq T-1$  do
  /* Computations in every agent: */
9   for  $\ell = 1 : M$  do
10     $\mathbf{z}_\ell(t+1) = \mathbf{z}_\ell(t) + \gamma \mathbf{P}_\ell^\perp (\bar{\mathbf{z}}(t) - \mathbf{z}_\ell(t))$ ;
11   end
  /* Computations in the server: */
12   $\bar{\mathbf{z}}(t+1) = \frac{\eta}{M} \sum_{\ell=1}^M \mathbf{z}_\ell(t+1) + (1-\eta)\bar{\mathbf{z}}(t)$ ;
13   $t = t + 1$ ;
14 end
15  $\tilde{\mathbf{z}} = \bar{\mathbf{z}}(t+1)$ ;

```

---

To clarify how to set the parameters  $\gamma$  and  $\eta$ , we define

$$\mathbf{X} = \frac{1}{M} \sum_{\ell=1}^M \mathbf{A}_\ell^H (\mathbf{A}_\ell \mathbf{A}_\ell^H)^{-1} \mathbf{A}_\ell, \quad (3)$$

and denote the eigenvalues of  $\mathbf{X}$  by  $\theta_i$ ,  $1 \leq i \leq s$ , where

$$\theta_s \leq \cdots \leq \theta_1 \quad (4)$$

with  $\theta_s = \theta_{\min} = \lambda_{\min}(\mathbf{X}) \geq 0$  and  $\theta_1 = \theta_{\max} = \lambda_{\max}(\mathbf{X}) \leq 1$  [1]. It is known from [14, Theorem 2.5.6] that  $\theta_1, \dots, \theta_s$  are real-valued.

*Remark 1:* The parameters  $\gamma$  and  $\eta$  in the APC algorithm are set as follows:

$$\gamma = \frac{2(\sqrt{\theta_{\max}}\sqrt{\theta_{\min}} + 1) - 2\sqrt{(1 - \theta_{\max})(1 - \theta_{\min})}}{(\sqrt{\theta_{\max}} + \sqrt{\theta_{\min}})^2}, \quad (5)$$

$$\eta = \frac{2(\sqrt{\theta_{\max}}\sqrt{\theta_{\min}} + 1) + 2\sqrt{(1 - \theta_{\max})(1 - \theta_{\min})}}{(\sqrt{\theta_{\max}} + \sqrt{\theta_{\min}})^2}, \quad (6)$$

which satisfy [1]

$$\theta_{\max}\eta\gamma = \left(1 + \sqrt{(\gamma - 1)(\eta - 1)}\right)^2, \quad (7)$$

$$\theta_{\min}\eta\gamma = \left(1 - \sqrt{(\gamma - 1)(\eta - 1)}\right)^2. \quad (8)$$

Note here that the seminal work [1] of the APC algorithm provided an indirect way of finding the optimal  $\gamma$  and  $\eta$ , which requires to solve an optimizing problem and might only achieve near-optimal values in practical use. While in this paper we present closed-form expressions of the optimal  $\gamma$  and  $\eta$ , i.e., (5) and (6), respectively, so as to simplify the parameter setting as well as the forthcoming performance analysis.

Moreover, we define

$$\alpha = \frac{\sqrt{\kappa(\mathbf{X})} - 1}{\sqrt{\kappa(\mathbf{X})} + 1}, \quad (9)$$

where  $\kappa(\mathbf{X}) = \frac{\theta_{\max}}{\theta_{\min}} \geq 1$  that is the condition number of  $\mathbf{X}$  [1].

### 3 Analysis

In this section, we conduct a performance analysis of the APC algorithm, in terms of the error of solution (Definition 1). We are now going to present the main results whose proofs are given in Appendix.

**Lemma 1.** *Consider Algorithm 1, and assume that  $\mathbf{A}_\ell$  is the  $\ell$ -th row of  $\mathbf{A}$  (see Line 1). Then, the projection matrix onto the nullspace of  $\mathbf{A}_\ell^H$  can be expressed as*

$$\mathbf{P}_\ell^\perp = \mathbf{I}_s - \mathbf{A}_\ell^H \left( \mathbf{A}_\ell \mathbf{A}_\ell^H \right)^{-1} \mathbf{A}_\ell, \quad (10)$$

which is calculated and used in Algorithm 1 (see Line 4 and Line 10, respectively).

Moreover,  $\mathbf{A}_\ell \mathbf{P}_\ell^\perp = \mathbf{0}_L^T$  and  $\left( \mathbf{P}_\ell^\perp \right)^2 = \mathbf{P}_\ell^\perp$ .

**Definition 1.** *Let*

$$\mathbf{e}_\ell(t) = \mathbf{x}_\ell(t) - \mathbf{x}^*, \quad (11)$$

$$\bar{\mathbf{e}}(t) = \bar{\mathbf{x}}(t) - \mathbf{x}^* = \frac{1}{M} \sum_{\ell=1}^M \mathbf{e}_\ell(t). \quad (12)$$

The key recursions in the APC algorithm, i.e., Line 10 and Line 12 of Algorithm 1, can be reformulated as

$$\mathbf{e}_\ell(t+1) = \mathbf{e}_\ell(t) + \gamma \mathbf{P}_\ell^\perp (\bar{\mathbf{e}}(t) - \mathbf{e}_\ell(t)), \quad (13)$$

$$\bar{\mathbf{e}}(t+1) = \frac{\eta}{M} \sum_{\ell=1}^M \mathbf{e}_\ell(t+1) + (1-\eta)\bar{\mathbf{e}}(t). \quad (14)$$

It follows from (2) that  $y_\ell = \mathbf{A}_\ell \mathbf{x}^* + \hat{w}_\ell$ , while every  $\mathbf{x}_\ell(t)$  computed in Line 5 or Line 10 of Algorithm 1 should be a solution of  $y_\ell = \mathbf{A}_\ell \mathbf{x}$ , i.e.,

$$y_\ell = \mathbf{A}_\ell \mathbf{x}_\ell(t). \quad (15)$$

This can be verified using the fact that, as long as  $\mathbf{A}_\ell \mathbf{x}_\ell(0) = y_\ell$ ,  $\mathbf{A}_\ell \mathbf{x}_\ell(t+1) = \mathbf{A}_\ell \mathbf{x}_\ell(t) + \gamma \mathbf{A}_\ell \mathbf{P}_\ell^\perp (\bar{\mathbf{x}}(t) - \mathbf{x}_\ell(t)) = \mathbf{A}_\ell \mathbf{x}_\ell(t) = y_\ell$  holds true for all  $t \geq 0$ , according to Lemma 1. Therefore,

$$\begin{aligned} \mathbf{P}_\ell^\perp \mathbf{e}_\ell(t) &= \mathbf{e}_\ell(t) - \mathbf{A}_\ell^H \left( \mathbf{A}_\ell \mathbf{A}_\ell^H \right)^{-1} (\mathbf{A}_\ell \mathbf{x}_\ell(t) - \mathbf{A}_\ell \mathbf{x}^*) \\ &= \mathbf{e}_\ell(t) - \mathbf{A}_\ell^H \left( \mathbf{A}_\ell \mathbf{A}_\ell^H \right)^{-1} \tilde{w}_\ell, \end{aligned} \quad (16)$$

which allows us to rewrite (13) as

$$\mathbf{e}_\ell(t+1) = (1-\gamma)\mathbf{e}_\ell(t) + \gamma \mathbf{P}_\ell^\perp \bar{\mathbf{e}}(t) + \gamma \mathbf{A}_\ell^H \left( \mathbf{A}_\ell \mathbf{A}_\ell^H \right)^{-1} \tilde{w}_\ell, \quad (17)$$

From (14) and (17), we can develop a state-space equation to describe the key recursions of the APC algorithm as follows:

$$\mathbf{d}(t+1) = \mathbf{G}\mathbf{d}(t) + \tilde{\mathbf{w}}_d, \quad (18)$$

where

$$\mathbf{d}(t) = \begin{bmatrix} \mathbf{e}_1(t) \\ \vdots \\ \mathbf{e}_M(t) \\ \bar{\mathbf{e}}(t) \end{bmatrix}, \quad \tilde{\mathbf{w}}_d = \gamma \begin{bmatrix} \mathbf{A}_1^H \left( \mathbf{A}_1 \mathbf{A}_1^H \right)^{-1} \tilde{w}_1 \\ \vdots \\ \mathbf{A}_M^H \left( \mathbf{A}_M \mathbf{A}_M^H \right)^{-1} \tilde{w}_M \\ \mathbf{0}_s \end{bmatrix}, \quad (19)$$

$$\mathbf{G} = \begin{bmatrix} (1-\gamma)\mathbf{I}_{Ms} & \gamma \begin{bmatrix} \mathbf{P}_1^\perp \\ \vdots \\ \mathbf{P}_M^\perp \end{bmatrix} \\ \frac{\eta(1-\gamma)}{M} [\mathbf{I}_s \cdots \mathbf{I}_s] & \mathbf{B} \end{bmatrix}, \quad (20)$$

with  $\mathbf{B} = \frac{\eta\gamma}{M} \sum_{\ell=1}^M \mathbf{P}_\ell^\perp + (1-\eta)\mathbf{I}_s = -\eta\gamma\mathbf{X} + (1-\eta+\eta\gamma)\mathbf{I}_s$ .

**Lemma 2.**  $\mathbf{G}$  have  $(M + 1)s$  eigenvalues, among which there are  $(M - 1)s$  eigenvalues that are equal to  $1 - \gamma$ , and  $2s$  eigenvalues  $\xi_{1,\pm}, \dots, \xi_{s,\pm}$ , where  $\xi_{i,\pm}$  ( $i = 1, \dots, s$ ) are the solutions of the quadratic equation

$$\xi^2 + (-\eta\gamma(1 - \theta_i) + \gamma + \eta - 2)\xi + (\gamma - 1)(\eta - 1) = 0, \quad (21)$$

such that

$$\xi_{i,\pm} = \frac{\theta_{max} + \theta_{min} - 2\theta_i}{(\sqrt{\theta_{max}} + \sqrt{\theta_{min}})^2} \pm \frac{2\sqrt{(\theta_i - \theta_{max})(\theta_i - \theta_{min})}}{(\sqrt{\theta_{max}} + \sqrt{\theta_{min}})^2}. \quad (22)$$

If  $\theta_{min} < \theta_i < \theta_{max}$ , then  $(\theta_i - \theta_{max})(\theta_i - \theta_{min}) < 0$  and thus  $\xi_{i,\pm}$  are complex-valued.

**Lemma 3.** For  $\mathbf{G}$ , the spectral radius  $\rho(\mathbf{G}) = \alpha < 1$ , so that  $\lim_{t \rightarrow \infty} \mathbf{G}^t = \mathbf{0}_{(M+1)s}$ , where  $\alpha$  is defined in (9), and  $t$  is a positive integer. Besides, the Neumann series  $\sum_{l=0}^{\infty} \mathbf{G}^l$  converges, i.e.,  $\sum_{l=0}^{\infty} \mathbf{G}^l = (\mathbf{I} - \mathbf{G})^{-1}$ .

Because the system (18) is in the form of a discrete-time state-space equation, its closed-form solution can be directly obtained by applying the linear system theory [15].

**Theorem 1.** The solution to the system (18) can be written as

$$\mathbf{d}(t) = \mathbf{G}^t \mathbf{d}(0) + \left( \sum_{l=0}^{t-1} \mathbf{G}^l \right) \tilde{\mathbf{w}}_d, \quad (23)$$

where  $\mathbf{G}^t \mathbf{d}(0)$  and  $\left( \sum_{l=0}^{t-1} \mathbf{G}^l \right) \tilde{\mathbf{w}}_d$  are the zero-input and zero-state responses, respectively [15, (4.20)].

Use Lemma 3 to show that

$$\sum_{l=0}^{t-1} \mathbf{G}^l = \sum_{l=0}^{\infty} \mathbf{G}^l - \mathbf{G}^t \left( \sum_{l=0}^{\infty} \mathbf{G}^l \right) = (\mathbf{I} - \mathbf{G}^t) (\mathbf{I} - \mathbf{G})^{-1}. \quad (24)$$

Substituting this result into (23) produces

$$\mathbf{d}(t) = \mathbf{G}^t \mathbf{d}(0) + (\mathbf{I} - \mathbf{G}^t) (\mathbf{I} - \mathbf{G})^{-1} \tilde{\mathbf{w}}_d, \quad (25)$$

which suggests that

$$\mathbf{d}(\infty) := \lim_{t \rightarrow \infty} \mathbf{d}(t) = (\mathbf{I} - \mathbf{G})^{-1} \tilde{\mathbf{w}}_d. \quad (26)$$

Observe that the behavior of  $\mathbf{d}(t)$  relies heavily on  $\mathbf{G}$  and  $\mathbf{G}^t$ , so we will apply the Jordan canonical form theorem to the coming analysis.

**Theorem 2.** *There exists a nonsingular matrix  $\mathbf{S} \in \mathbb{C}^{(M+1)s \times (M+1)s}$ , and there are positive integers  $q, n_1, \dots, n_q$  with  $n_1 + \dots + n_q = (M + 1)s$ , and scalars  $\xi_1, \dots, \xi_q \in \{1 - \gamma, \xi_{1,\pm}, \xi_{2,\pm}, \dots, \xi_{s,\pm}\}$  such that*

$$\mathbf{G} = \mathbf{S}^{-1} \mathbf{J} \mathbf{S}, \tag{27}$$

where  $\mathbf{J} = \mathbf{J}_{n_1}(\xi_1) \oplus \dots \oplus \mathbf{J}_{n_q}(\xi_q)$  is a Jordan matrix, and  $\mathbf{J}_{n_l}(\xi_l), l = 1, \dots, q$ , are Jordan blocks<sup>1</sup>. (Jordan Canonical Form [14, Theorem 3.1.11])

This theorem has an important consequence. Precisely, we have the following corollary.

**Corollary 1.** *Let  $\mathbf{G}$  be given. Then  $\mathbf{G}^t = \mathbf{S}^{-1} \mathbf{J}^t \mathbf{S}$ .*

Next, we investigate the properties of  $\mathbf{J}^t$  by establishing two lemmas (i.e., Lemmas 4 and 6 given below) that identify the features of Jordan blocks with different eigenvalues of  $\mathbf{G}$ .

**Lemma 4.** *The number of Jordan blocks of  $\mathbf{G}$  corresponding to eigenvalue  $1 - \gamma$  is  $(M - 1)s$ , such that every Jordan block is 1-by-1. (Jordan Blocks with Eigenvalue  $1 - \gamma$  of  $\mathbf{G}$ )*

The next result aims to provide insights into Jordan blocks with eigenvalues  $\xi_{i,\pm}, i = 1, \dots, s$ . Before stating this result, we will first give an explicit formula for the eigenvectors associated with these eigenvalues.

**Lemma 5.** *Let  $\mathbf{A} = \text{diag}(\theta_1, \dots, \theta_s)$ , where  $\theta_1, \dots, \theta_s$  are the eigenvalues of  $\mathbf{X}$ , see (3) and (4). Then*

- (a)  $\mathbf{X}$  has  $s$  orthonormal eigenvectors, denoted by  $\mathbf{v}_1, \dots, \mathbf{v}_s$ , where  $\mathbf{v}_i$  is the eigenvector associated with eigenvalue  $\theta_i$  of  $\mathbf{X}, i = 1, \dots, s$ .
- (b) Every scalar  $\xi \in \{\xi_{1,\pm}, \dots, \xi_{s,\pm}\}$  (such that  $\xi$  is a solution to (21)) and

$$\mathbf{v}_G(\xi) = \begin{bmatrix} -\frac{\gamma}{1-\gamma-\xi} \mathbf{P}_1^\perp \mathbf{v}_i \\ -\frac{\gamma}{1-\gamma-\xi} \mathbf{P}_2^\perp \mathbf{v}_i \\ \vdots \\ \mathbf{v}_i \end{bmatrix}, \tag{28}$$

form an eigenvalue-eigenvector pair for  $\mathbf{G}$ .

**Lemma 6.** *The number of Jordan blocks of  $\mathbf{G}$  corresponding to every eigenvalue  $\xi \in \{\xi_{1,\pm}, \dots, \xi_{s,\pm}\}$  is at most 2, where each Jordan block is either 1-by-1 or 2-by-2. (Jordan Blocks with Eigenvalues  $\xi_{i,\pm}, i = 1, \dots, s$ , of  $\mathbf{G}$ )*

The preceding results (including Lemmas 4 and 6) motivate the property of  $\mathbf{J}^t$ , as shown by the following lemma.

<sup>1</sup> The Jordan block  $\mathbf{J}_{n_l}(\xi_l)$  is an  $n_l$ -by- $n_l$  upper triangular matrix in which  $\xi_l$  appears  $n_l$  times on the main diagonal; if  $n_l > 1$ , there are  $n_l - 1$  elements 1 in the super-diagonal; all other elements are 0 [14].

**Lemma 7.** For the Jordan matrix  $\mathbf{J}$  in (27), we have

$$\mathbf{J}^t = \mathbf{J}_{n_1}^t(\xi_1) \bigoplus \cdots \bigoplus \mathbf{J}_{n_q}^t(\xi_q), \quad (29)$$

where

$$\mathbf{J}_n^t(\xi) = \begin{bmatrix} \xi^t & \binom{t}{1}\xi^{t-1} & \cdots & \binom{t}{n-1}\xi^{t-n+1} \\ 0 & \xi^t & \cdots & \binom{t}{n-2}\xi^{t-n+2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi^t \end{bmatrix}, \quad (30)$$

and  $n_l \in \{1, 2\}$  for all  $l = 1, \dots, q$ .

After obtaining Corollary 1 and Lemma 7, the mechanism governing the convergence of  $\mathbf{G}^t$  (in (25)) to zero is better understood. Now we are in a position to give a formulation that characterizes the behavior of  $\mathbf{d}(t)$ .

**Theorem 3.** Consider the solution  $\mathbf{d}(t)$  of the state-space equation (18). We have

$$\mathbf{d}(t) = \mathbf{d}(\infty) + \mathbf{G}^t \left( \mathbf{d}(0) - (\mathbf{I}_{(M+1)s} - \mathbf{G})^{-1} \tilde{\mathbf{w}}_d \right), \quad (31)$$

since  $\mathbf{d}(\infty) = (\mathbf{I}_{(M+1)s} - \mathbf{G})^{-1} \tilde{\mathbf{w}}_d$ , such that

$$\mathbf{d}(t) = (\mathbf{I}_{(M+1)s} - \mathbf{G})^{-1} \tilde{\mathbf{w}}_d + (\mathbf{I}_{(M+1)s} - \mathbf{G})^{-1} \boldsymbol{\epsilon}, \quad (32)$$

where

$$\boldsymbol{\epsilon} = \mathbf{G}^t \left( (\mathbf{I}_{(M+1)s} - \mathbf{G}) \mathbf{d}(0) - \tilde{\mathbf{w}}_d \right). \quad (33)$$

Moreover,  $\|\boldsymbol{\epsilon}\|_2 = O(\alpha^t)$ .

**Theorem 4.** Consider the problem of solving a large-scale system (1) of linear equations, we have

$$\bar{\mathbf{x}}(t) - \mathbf{x}^* = \frac{1}{(1+\eta)M} \mathbf{X}^{-1} \mathbf{A}^H \boldsymbol{\Xi} \tilde{\mathbf{w}} + \frac{1}{\gamma(1+\eta)} \mathbf{X}^{-1} \bar{\boldsymbol{\epsilon}}, \quad (34)$$

where

$$\bar{\boldsymbol{\epsilon}} = \left[ \frac{1}{M} \mathbf{I}_s \quad \frac{1}{M} \mathbf{I}_s \quad \cdots \quad \mathbf{I}_s \right] \boldsymbol{\epsilon}. \quad (35)$$

Moreover,  $\|\bar{\boldsymbol{\epsilon}}\|_2 = O(\alpha^t)$ .

Theorem 3 characterizes the error of solution attained by the APC algorithm when the system (1) is faced with unknown noise  $\tilde{\mathbf{w}}$  as in (2). It is easy to see that Theorem 3 is a generalization of Theorem 1 derived in [1], because by letting  $\tilde{\mathbf{w}}$  be the all-zero vector in Theorem 3, our obtained result can be reduced to Theorem 1 of [1].

## 4 Numerical Results

In this section, we carry out a series of simulations to evaluate the performance of the APC algorithm in the complex white gaussian noise scenarios with the aim to verify the main result (Theorem 3). The noise vector  $\tilde{\mathbf{w}}$  has independent identically distributed elements satisfying  $\tilde{w}_i \sim CN(0, P_n)$ . By letting the iteration number  $T$  vary, we focus on the mean square error (MSE) performance which defined as:

$$\text{MSE} = \mathbb{E} \left\{ \|\mathbf{x}^* - \tilde{\mathbf{x}}\|_2^2 \right\}. \quad (36)$$

We mainly consider the influence of three factors: the algorithm parameter  $M$ , the condition number of matrix  $\mathbf{X}$  (i.e.,  $\kappa(\mathbf{X})$ ), and the noise power  $P_n$ .

Figure 1 shows the MSE performance of the APC algorithm with different size of matrix  $\mathbf{A}$ . We set  $\kappa(\mathbf{X}) = 1.6$  (i.e.,  $\alpha = 0.23$ ), and the noise power  $P_n = 1.0e-4$ . When  $M = \{8, 32, 128\}$ , the simulation results are shown in Fig. 1. We can find that lines are similar to the exponential decay curve. No matter what  $M$  is equal to, the algorithm can quickly converge after  $T = 4$ . When  $T \geq 5$ ,  $\|\tilde{\boldsymbol{\epsilon}}\|_2 = O(0.23^T)$  in (34) is small enough ( $\|\tilde{\boldsymbol{\epsilon}}\|_2 \leq 10^{-3}$ ) to be neglected.

Figure 2 is plotted with  $M = 32$  and  $P_n = 1.0e-4$ . It shows the influence of  $\kappa(\mathbf{X})$  on the MSE performance. By comparing the curves, we can observe that if we keep  $M$  and  $P_n$  the same, the converged value of the MSE varies only slightly. But there are differences in convergence speed. When the condition number is larger, the convergence speed is lower. Specifically, when  $\kappa(\mathbf{X}) = 1.56$ , the algorithm converges after 3 iterations; but when  $\kappa(\mathbf{X})$  increases to 6.0, while the algorithm needs 9 iterations to achieve the convergence. Combined with (9), we can analyze that the closer  $\kappa(\mathbf{X})$  is to 1, the closer  $\alpha$  is to 0, and the faster  $\|\tilde{\boldsymbol{\epsilon}}\|_2$  decreases which influences the MSE performance according to (32) and (36). The theoretical analysis is in accordance with the simulation results.

Figure 3 illustrates the effect of the noise power  $P_n$  on the MSE performance for the APC algorithm. Let  $P_n$  gradually increase from  $1.0e-5$  to  $1.0e-3$ . It can be seen that the increase of  $P_n$  does not have evident influence on the convergence speed of the algorithm, but the final MSE performance (after the algorithm converges) will get worse. We also plot ADMM method in Fig. 3 as the comparison algorithm. The figure clearly presents that, ADMM needs 30 times iterations to decrease the MSE to 0.74 with  $P_n = 1.0e-5$ , but APC gets convergence by 5 times iterations under higher noise level ( $P_n = 1.0e-3$ ). Comparing with ADMM, APC can achieve better MSE performance with faster convergence in the scene with higher noise level.

In summary, as the value of  $T$  increases,  $\frac{1}{\gamma(1+\eta)} \mathbf{X}^{-1} \tilde{\boldsymbol{\epsilon}}$  in (34) tends to 0, the error of APC algorithm will only be effected by  $\frac{1}{(1+\eta)^M} \mathbf{X}^{-1} \mathbf{A}^H \boldsymbol{\Xi} \tilde{\mathbf{w}}$  in (34). In other words, the error is determined by the noise level after enough iterations. This fits with Theorem 3.

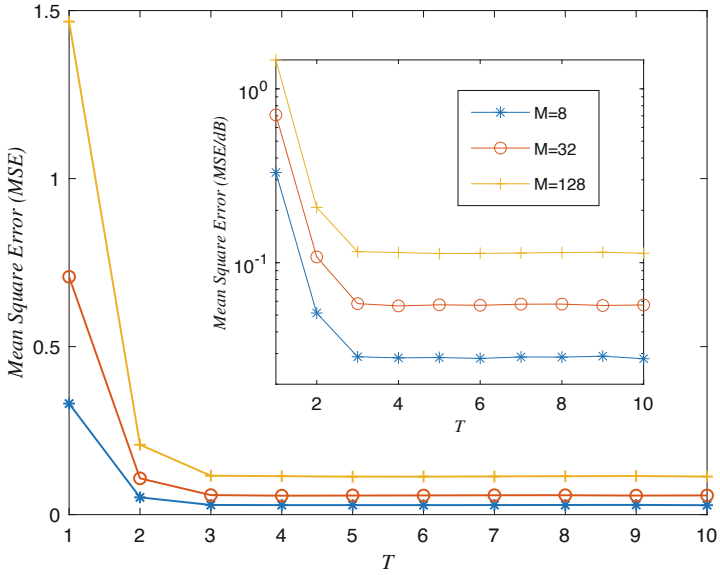


Fig. 1. MSE performance of the APC algorithm as a function of  $T$  with different  $M$ .

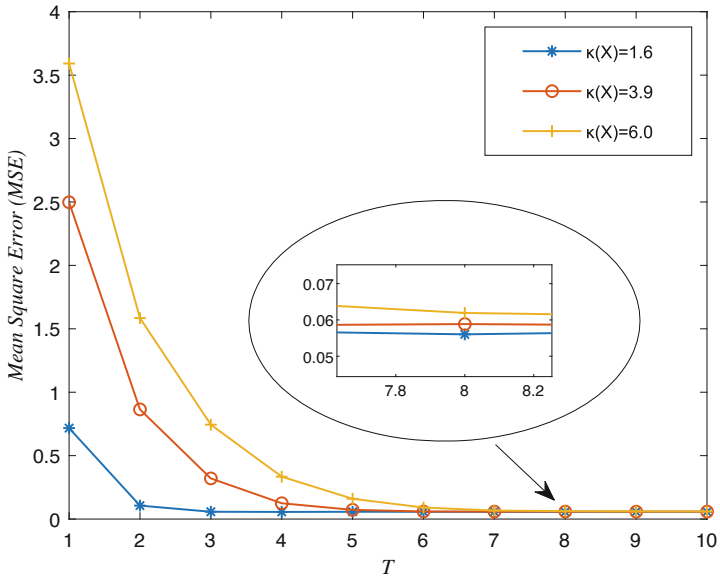


Fig. 2. MSE performance of the APC algorithm as a function of  $T$  with different  $\kappa(\mathbf{X})$ .

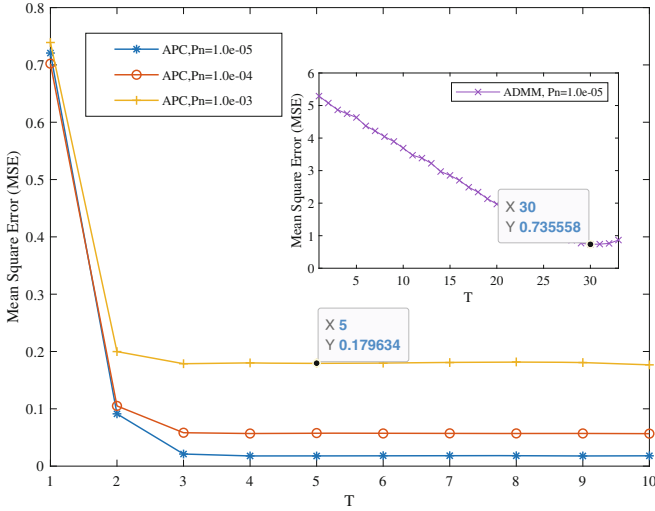


Fig. 3. MSE performance of the APC algorithm as a function of  $T$  with different  $P_n$ .

## 5 Conclusion

In this paper, we generalize the analysis of the APC algorithm, and clarify the error performance of the algorithm in presence of noise for solving linear systems. The generalization should be necessary to refine the theoretical framework of the APC algorithm. Also, we provide closed-form expressions to two important parameters of APC. In view of the growing interest in distributed signal processing, to some degree, this study also has significance in regard to the analysis of distributed signal processing algorithms using the federated architecture.

**Acknowledgment.** This work is sponsored in part by the National Natural Science Foundation of China (grant no. 61801048, 61971058, 62071063) and Beijing Natural Science Foundation (grant no. L202014, L192002).

## Appendix: Proofs of Main Results

*Proof of Lemma 3:* A computation reveals that

$$|\xi_{i,\pm}| = \sqrt{(\gamma - 1)(\eta - 1)} = \alpha, \quad (37)$$

for all  $i = 1, \dots, s$ . Then, it's a simple consequence of Lemma A.2 that  $\rho(\mathbf{G}) = \alpha < 1$  [1], i.e., the largest magnitude eigenvalue of  $\mathbf{G}$  is less than 1. This implies that  $\lim_{t \rightarrow \infty} \mathbf{G}^t = \mathbf{O}_{(M+1)s}$  and  $\sum_{l=0}^{\infty} \mathbf{G}^l = (\mathbf{I} - \mathbf{G})^{-1}$  [16]. ■

*Proof of Lemma 4:* All we need to do is to show that the geometric multiplicity of eigenvalue  $1 - \gamma$  is  $(M - 1)s$  [14,16]. We begin by noting that

$$\mathbf{G} - (1 - \gamma)\mathbf{I} = \begin{bmatrix} & \mathbf{O}_{Ms} & \gamma \begin{bmatrix} \mathbf{P}_1^\perp \\ \vdots \\ \mathbf{P}_M^\perp \end{bmatrix} \\ \frac{\eta(1-\gamma)}{M} [\mathbf{I}_s \cdots \mathbf{I}_s] & \mathbf{B} - (1 - \gamma)\mathbf{I}_s & \end{bmatrix}.$$

Applying elementary row and column operations [14], we can transform  $\mathbf{G} - (1 - \gamma)\mathbf{I}$  into a simple form

$$\mathbf{G}_T = \begin{bmatrix} & \mathbf{O}_{Ms} & \gamma \begin{bmatrix} \mathbf{P}_1^\perp \\ \vdots \\ \mathbf{P}_M^\perp \end{bmatrix} \\ \frac{\eta(1-\gamma)}{M} [\mathbf{O}_s \cdots \mathbf{O}_s \mathbf{I}_s] & \mathbf{O}_s & \end{bmatrix}.$$

Since elementary operations do not change the rank of a matrix [14], it follows that

$$\text{rank}(\mathbf{G} - (1 - \gamma)\mathbf{I}) = \text{rank}(\mathbf{G}_T) \leq 2s. \quad (38)$$

The geometric multiplicity of eigenvalue  $1 - \gamma$  of  $\mathbf{G}$  is equal to  $(M + 1)s - \text{rank}(\mathbf{G} - (1 - \gamma)\mathbf{I})$  that is not less than  $(M - 1)s$  according to (38). Furthermore, the geometric multiplicity should not be larger than algebraic multiplicity, i.e.,  $(M - 1)s$ , for  $1 - \gamma$ , and thus can only be  $(M - 1)s$  [14, Theorem 1.4.10]. This implies that the number of Jordan blocks of  $\mathbf{G}$  corresponding to  $1 - \gamma$ , is  $(M - 1)s$  [14]. Finally, because the geometric and algebraic multiplicities of  $1 - \gamma$  are equal, every Jordan block corresponding to  $1 - \gamma$  is 1-by-1 [14]. ■

*Proof of Lemma 5:* First note that  $\mathbf{X}$  is Hermitian and  $\mathbf{X}$  is unitarily diagonalizable [14, Theorem 2.5.6]. Then, applying [14, Theorem 2.5.3] yields the assertion (a).

To prove the assertion (b), let us verify whether  $\mathbf{v}_G(\xi)$  satisfies the eigenvalue-eigenvector equation  $(\mathbf{G} - \xi\mathbf{I})\mathbf{v}_G(\xi) = \mathbf{0}_{(M+1)s}$ , where

$$\mathbf{G} - \xi\mathbf{I}_{(M+1)s} = \begin{bmatrix} & (1 - \gamma - \xi)\mathbf{I}_{Ms} & \gamma \begin{bmatrix} \mathbf{P}_1^\perp \\ \vdots \\ \mathbf{P}_M^\perp \end{bmatrix} \\ \frac{\eta(1-\gamma)}{M} [\mathbf{I}_s \cdots \mathbf{I}_s] & \mathbf{B} - \xi\mathbf{I}_s & \end{bmatrix}.$$

On the one hand, it is easy to check that

$$\begin{bmatrix} (1 - \gamma - \xi)\mathbf{I}_{ML} & \gamma \begin{bmatrix} \mathbf{P}_1^\perp \\ \vdots \\ \mathbf{P}_M^\perp \end{bmatrix} \end{bmatrix} \mathbf{v}_G(\xi) = \mathbf{0}_{Ms}. \quad (39)$$

On the other hand, since  $\mathbf{X}\mathbf{v}_i = \theta_i\mathbf{v}_i$  and  $\frac{\sum_{\ell=1}^M \mathbf{P}_\ell^\perp}{M} = \mathbf{I}_s - \mathbf{X}$ , we have  $\frac{\sum_{\ell=1}^M \mathbf{P}_\ell^\perp}{M} \mathbf{v}_i = (1 - \theta_i)\mathbf{v}_i$  and  $(\mathbf{B} - \xi\mathbf{I}_s)\mathbf{v}_i = (-\eta\gamma\theta_i + 1 - \eta + \eta\gamma - \xi)\mathbf{v}_i$ . As (21) ensures that  $-\frac{\eta\gamma(1-\gamma)}{1-\gamma-\xi}(1-\theta) - \eta\gamma\theta + 1 - \eta + \eta\gamma - \xi = 0$  if  $\xi \neq 1 - \gamma$ , it yields

$$\begin{aligned} & \left[ \frac{\eta(1-\gamma)}{M} [\mathbf{I}_s \cdots \mathbf{I}_s] \mathbf{B} - \xi\mathbf{I}_s \right] \mathbf{v}_G(\xi) \\ &= \left[ -\frac{\eta\gamma(1-\gamma)}{(1-\gamma-\xi)} \frac{\sum_{\ell=1}^M \mathbf{P}_\ell^\perp}{M} + (\mathbf{B} - \xi\mathbf{I}_s) \right] \mathbf{v}_i = \mathbf{0}_s. \end{aligned} \quad (40)$$

Finally, (39) and (40) together imply that  $(\mathbf{G} - \xi\mathbf{I})\mathbf{v}_G(\xi) = \mathbf{0}_{(M+1)s}$  which completes the proof of the assertion (b).  $\blacksquare$

*Proof of Lemma 6:* It follows from (22) that  $\xi_{1,+} = \xi_{1,-}$  and  $\xi_{s,+} = \xi_{s,-}$  since  $\theta_s = \theta_{min}$  and  $\theta_1 = \theta_{max}$ , while  $\xi_{i,+} \neq \xi_{i,-}$  if  $\theta_{min} < \theta_i < \theta_{max}$ . One of the consequences of Lemma 5 is that the number of linearly dependent eigenvectors associated with every eigenvalue  $\xi \in \{\xi_{1,\pm}, \dots, \xi_{s,\pm}\}$  is not larger than 2. Therefore, the geometric multiplicity of  $\xi$  is 1, and the Jordan block of  $\mathbf{G}$  with eigenvalue  $\xi$  is 2-by-2, when  $\xi = \xi_{1,+} = \xi_{1,-}$  or  $\xi = \xi_{s,+} = \xi_{s,-}$ . If  $\xi = \xi_{i,+}$  or  $\xi = \xi_{i,-}$  with  $\xi_{i,+} \neq \xi_{i,-}$ , then the geometric multiplicity of  $\xi$  is 2 such that the Jordan block with eigenvalue  $\xi$  is 1-by-1.  $\blacksquare$

*Proof of Lemma 7:* It can easily be verified by using the property of direct sum together with [16, (7.10.7)].  $\blacksquare$

*Proof of Theorem 2:* Combining (25) and (26) yields (31), and then multiplying on both sides of (31) by  $\mathbf{I}_{(M+1)s} - \mathbf{G}$  produces (33). By the Rayleigh quotient theorem [16] and Corollary 1, we obtain

$$\|\boldsymbol{\epsilon}\|_2 \leq \|\mathbf{J}^t\|_2 \left\| (\mathbf{I}_{(M+1)s} - \mathbf{G}) \mathbf{d}(0) - \tilde{\mathbf{w}}_d \right\|_2, \quad (41)$$

where  $\|\mathbf{J}^t\|_2 = O(\alpha^t)$  [15].

The remaining part of the proof is to derive an upper bound on  $\left\| (\mathbf{I}_{(M+1)s} - \mathbf{G}) \mathbf{d}(0) - \tilde{\mathbf{w}}_d \right\|_2$ . First, it is clear that

$$\mathbf{d}(0) = \begin{bmatrix} -\mathbf{P}_1^\perp \mathbf{x}^* + \mathbf{A}_1^H \left( \mathbf{A}_1 \mathbf{A}_1^H \right)^{-1} \tilde{w}_1 \\ \vdots \\ -\mathbf{P}_M^\perp \mathbf{x}^* + \mathbf{A}_M^H \left( \mathbf{A}_M \mathbf{A}_M^H \right)^{-1} \tilde{w}_M \\ -\frac{1}{M} \sum_{\ell=1}^M \mathbf{P}_\ell^\perp \mathbf{x}^* + \frac{1}{M} \sum_{\ell=1}^M \mathbf{A}_\ell^H \left( \mathbf{A}_\ell \mathbf{A}_\ell^H \right)^{-1} \tilde{w}_\ell \end{bmatrix}, \quad (42)$$

according to the initialization of the APC algorithm as in Line 5 of Algorithm 1 together with (19), and thus

$$(\mathbf{I}_{(M+1)s} - \mathbf{G}) \mathbf{d}(0) - \tilde{\mathbf{w}}_d = \begin{bmatrix} -\gamma \mathbf{P}_1^\perp (\mathbf{x}^* + \bar{\mathbf{e}}(0)) \\ \vdots \\ -\gamma \mathbf{P}_M^\perp (\mathbf{x}^* + \bar{\mathbf{e}}(0)) \\ -\eta\gamma \mathbf{X} \bar{\mathbf{e}}(0) \end{bmatrix}.$$

This yields

$$\begin{aligned} \left\| \left( \mathbf{I}_{(M+1)s} - \mathbf{G} \right) \mathbf{d}(0) - \tilde{\mathbf{w}}_d \right\|_2^2 &\leq 2\gamma^2 M (1 - \theta_{min}) \|\mathbf{x}^*\|_2^2 \\ &\quad + \left( 2\gamma^2 M (1 - \theta_{min}) + \eta^2 \gamma^2 \theta_{max}^2 \right) \|\bar{\mathbf{e}}(0)\|_2^2. \end{aligned}$$

Then, since  $\|\bar{\mathbf{e}}(0)\|_2^2 \leq \frac{\sum_{\ell=1}^M \|\mathbf{e}_\ell(0)\|_2^2}{M}$ , it follows from Lemma 1 and (42) that

$$\begin{aligned} \|\bar{\mathbf{e}}(0)\|_2^2 &\leq \frac{M (\mathbf{x}^*)^H (\mathbf{I} - \mathbf{X}) \mathbf{x}^* + \sum_{\ell=1}^M \left( \mathbf{A}_\ell \mathbf{A}_\ell^H \right)^{-2} \tilde{w}_\ell}{M} \\ &\leq (1 - \theta_{min}) \|\mathbf{x}^*\|_2^2 + \frac{\sum_{\ell=1}^M \left( \mathbf{A}_\ell \mathbf{A}_\ell^H \right)^{-2} \tilde{w}_\ell}{M}, \end{aligned}$$

and

$$\begin{aligned} \left\| \left( \mathbf{I}_{(M+1)s} - \mathbf{G} \right) \mathbf{d}(0) - \tilde{\mathbf{w}}_d \right\|_2 &\leq \left( 2\gamma^2 M (1 - \theta_{min}) + \eta^2 \gamma^2 \theta_{max}^2 \right) \\ &\quad \times \left( (2 - \theta_{min}) \|\mathbf{x}^*\|_2^2 + \frac{\sum_{\ell=1}^M \left( \mathbf{A}_\ell \mathbf{A}_\ell^H \right)^{-2} \tilde{w}_\ell}{M} \right). \end{aligned}$$

Substituting this inequality into (41) gives  $\|\epsilon\|_2 = O(\alpha^t)$ .  $\blacksquare$

*Proof of Theorem 3:* Observe that

$$\begin{aligned} \left[ \frac{1}{M} \mathbf{I}_s \quad \frac{1}{M} \mathbf{I}_s \quad \cdots \quad \mathbf{I}_s \right] \mathbf{d}(t) &= 2\bar{\mathbf{e}}(t), \\ \left[ \frac{1}{M} \mathbf{I}_s \quad \frac{1}{M} \mathbf{I}_s \quad \cdots \quad \mathbf{I}_s \right] \mathbf{G} \mathbf{d}(t) &= (2\mathbf{I}_s - \gamma(1 + \eta)\mathbf{X}) \bar{\mathbf{e}}(t). \end{aligned}$$

A calculation also shows that  $\left[ \frac{1}{M} \mathbf{I}_s \quad \frac{1}{M} \mathbf{I}_s \quad \cdots \quad \mathbf{I}_s \right] \tilde{\mathbf{w}}_d = \frac{\gamma}{M} \mathbf{A}^H \Xi \tilde{\mathbf{w}}$ . Combining these results, we get

$$\gamma(1 + \eta)\mathbf{X}\bar{\mathbf{e}}(t) = \frac{\gamma}{M} \mathbf{A}^H \Xi \tilde{\mathbf{w}} + \left[ \frac{1}{M} \mathbf{I}_s \quad \frac{1}{M} \mathbf{I}_s \quad \cdots \quad \mathbf{I}_s \right] \epsilon,$$

which can be rewritten as (34). From Theorem 1, it follows that  $\|\bar{\mathbf{e}}\|_2 = O(\alpha^t)$ . Finally, based on the definition in (12), Theorem 3 can be verified.  $\blacksquare$

## References

1. Azizan-Ruhi, N., Lahouti, F., Avestimehrm, A., Hassibi, B.: Distributed solution of large-scale linear systems via accelerated projection based consensus. *IEEE Trans. Sig. Process.* **67**(14), 3806–3817 (2019)
2. Alaviani, S., Elia, N.: A distributed algorithm for solving linear algebraic equations over random networks. *IEEE Trans. Autom. Control* **66**(5), 2399–2406 (2021)
3. Zivojevic, D., et al.: Distributed weighted least-squares and Gaussian belief propagation: an integrated approach. In: *Proceedings of 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 432–437 (2021)

4. Wang, X., Mou, S., Sun, D.: Improvement of a distributed algorithm for solving linear equations. *IEEE Trans. Ind. Electron.* **64**(4), 3113–3117 (2017)
5. Zuo, C., Deng, H., Zhang, J., Qi, Y.: Distributed channel estimation algorithm for mmWave massive MIMO communication systems. In: *Proceedings of 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pp. 1–6 (2021)
6. Jia, M., Wang, Y., Shen, C., Hug, G.: Privacy-preserving distributed probabilistic load flow. *IEEE Trans. Power Syst.* **36**(2), 1616–1627 (2021)
7. Chakrabarti, K., Gupta, N., Chopra, N.: Iterative pre-conditioning to expedite the gradient-descent method. In: *Proceedings of 2020 American Control Conference (ACC)*, pp. 3977–3982 (2020)
8. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: challenges, methods, and future directions. *IEEE Sig. Process. Mag.* **37**(3), 50–60 (2020)
9. Chakrabarti, K., Gupta, N., Chopra, N.: Robustness of iteratively pre-conditioned gradient-descent method: the case of distributed linear regression problem. In: *Proceedings of 2021 American Control Conference (ACC)*, pp. 2248–2253 (2021)
10. Dutta, S., Cadambe, V., Grover, P.: Short-Dot: computing large linear transforms distributedly using coded short dot product. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS)*, pp. 2092–2100 (2016)
11. Dutta, S., Cadambe, V., Grover, P.: Short-Dot: computing large linear transforms distributedly using coded short dot products. *IEEE Trans. Inf. Theory* **65**(10), 6171–6193 (2019)
12. Dutta, S., et al.: On the optimal recovery threshold of coded matrix multiplication. *IEEE Trans. Inf. Theory* **66**(1), 278–301 (2020)
13. Zhao, Z., Lu, S., Hong, M., Palomar, D.: Distributed optimization for generalized phase retrieval over networks. In: *Proceedings of 2018 52nd Asilomar Conference on Signals, Systems and Computers*, pp. 48–52 (2018)
14. Horn, R., Johnson, C.: *Matrix Analysis*. Cambridge University Press, 2nd ed., Cambridge, USA (2013)
15. Chen, C.: *Linear System Theory and Design*. Oxford University Press, 3rd ed., Oxford, USA (1999)
16. Meyer, C.: *Matrix Analysis and Applied Linear Algebra*. Cambridge University Press, Cambridge (2000)