SIAM J. OPTIM.
Vol. 30, No. 2, pp. 1191–1222

DISTRIBUTED ALGORITHMS WITH FINITE DATA RATES THAT SOLVE LINEAR EQUATIONS*

JINLONG LEI†, PENG YI†, GUODONG SHI‡, AND BRIAN D. O. ANDERSON§

Abstract. In this paper, we study network linear equations subject to digital communications with a finite data rate, where each node is associated with one equation from a system of linear equations. Each node holds a dynamic state and interacts with its neighbors through an undirected connected graph, where along each link the pair of nodes share information. Due to the data rate constraint, each node builds an encoder/decoder pair, with which it produces transmitted messages with a zooming-in finite-level uniform quantizer and also generates estimates of its neighbors' states from the received signals. We then propose a distributed quantized algorithm and show that when the network linear equations admit a unique solution, each node's state is driven to that solution exponentially fast. We further analyze the asymptotic rate of convergence and show that a larger number of quantization levels leads to a faster convergence rate although the rate is still fundamentally bounded by the inherent network structure and the linear equations. In addition, we establish a bound on the total number of communication bits required to obtain a solution with a prescribed accuracy. When a unique least-squares solution exists, we show that the algorithm can compute such a solution with a suitably selected time-varying step-size inherited from the encoder and zooming-in quantizer dynamics. In both cases, a minimal data rate is shown to be enough for guaranteeing the desired convergence when the algorithm parameters are properly chosen. These results ensure the applicability of various network linear equation solvers when peer-to-peer communication is digital.

Key words. network linear equations, data rate, quantization, distributed algorithm

AMS subject classifications. 15A06, 90C06, 90C20, 68P30, 68Q87

DOI. 10.1137/19M1258864

1. Introduction.

The pursuit of distributed and scalable solutions for control and optimization problems over large-scale network systems has been one of the central themes in the past decades [8, 42]. For a group of interconnected computational nodes, sensing and decision making can be carried out individually based on the information flow across the interconnected links, under which collective goals such as consensus, inference, and estimation can be achieved [4, 8, 46]. Particularly, the classical distributed optimization framework, introduced in 1980s by the seminal work of Tsitsiklis, Bertsekas, and Athans [42], has been further developed into a new one where various new distributed algorithms arise from a lot of problems (cf. [19, 20, 30, 40]). Solid understandings have been reached on how the underlying network structure affects the convergence and convergence rates of such algorithms [5, 28, 49]. This might be the most important feature distinguishing the distributed optimization from

*Received by the editors April 29, 2019; accepted for publication (in revised form) February 12, 2020; published electronically April 28, 2020. A preliminary version of this work was presented in *Proceedings of the IEEE Conference on Decision and Control*, 2018.

https://doi.org/10.1137/19M1258864

Funding: The work of the fourth author was supported by the Australian Research Council (ARC) under grants DP-160104500 and DP19010088 and by Data61-CSIRO.

†Department of Control Science and Engineering, Tongji University, and Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai, China (leijinlong@tongji.edu.cn, yipeng@amss.ac.cn, yipeng@tongji.edu.cn).

‡Australian Centre for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, Sydney, NSW 2006, Australia (guodong.shi@sydney.edu.au).

§Hangzhou Dianzi University, Hangzhou, China, and the Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, ACT 0200, Australia (brian.anderson@anu.edu.au).

parallel computation [23, 37] which usually requires a common shared memory and all-to-all node communication. In big data applications, distributed schemes might even be the only choice since a single node cannot just store all the data or devote most of the CPU time to I/O operations.

Systems of linear algebraic equations, the solving of which is a primary computation task, arise from diverse engineering problems [7, 11]. Recent research has focused on the resolution of the linear equations using multiple nodes over a network, where each node holds one or a few of the linear equations [26]. The network linear equations also arise from resource allocation problems when node cost functions are quadratic; see [30, 34]. In the context of parallel computation [14, 23, 37], each node eventually computes a subset of the entries of the solutions to a linear equation. On the other hand, motivated by tasks of distributed optimization [18, 24, 30, 31], distributed algorithms were proposed to compute the entire solution vector at each node. In particular, discrete-time algorithms are studied by [12, 25, 26, 43], while continuous-time node dynamics are treated in [1, 38] to solve the network linear equation. In fact, when exact solutions exist for the linear equations, such first-order distributed solvers amount to generalized versions of the alternation projection algorithms pioneered by von Neumann [31, 39, 44]. When no exact solution exists and one considers least-squares solutions, properly selected diminishing step-sizes are required [38].

In recent years, considerable efforts have been made on the distributed protocols with quantized communications, including network control [27], consensus problems [2, 6, 10, 17, 32], and distributed optimization [21, 35, 47]. There is also emerging attention from the machine learning and AI communities on investigating communication-efficient algorithms along with communication complexity [22, 41, 45, 48] since training with big data inevitably relies on large-scale distributed computer clusters that only have digital communications over the Ethernet. For distributed network linear equations with finite data rates, the interplay between the network structure, the communication capacity, the systems of linear equations, and the convergence rate is of significant interest. A clear description for that would enhance the understanding of the distributed computation subject to finite channel capacity as a whole. On one hand, the algorithms and the resulting convergence analysis would go far beyond the distributed quantized consensus algorithms [6, 17] focusing on the task of solving linear algebraic equations. On the other hand, as a simplified problem with quadratic cost functions among the study on the distributed quantized optimization [13, 35, 36, 47], the resolution of our results can be pushed to a much higher level where the connection and distinction between low and high data rates are clearly revealed. However, such a simplification to quadratic optimization might also lead to new challenges because (i) the quadratic function associated with each node is not strongly convex, a condition required by [13, 36]; (ii) gradients of each node's objective function cannot be assumed to be globally bounded a priori, which, however, is a key technical assumption imposed on distributed quantized subgradient optimization [47].

Among various versions of the network linear equation problem, we consider the setting where each node holds one equation from a system of linear equations with m unknown variables. The nodes aim to reach consensus on the solution of the linear equations. The node interconnection is described by an undirected connected graph, where along each link the neighboring nodes exchange information constrained by a limited data rate measured in bits. We propose an encoder/decoder powered distributed quantized method to solve the network linear equation, motivated by the quantized consensus protocol proposed in [17]. Each node builds an encoder/decoder

pair with the help of a zooming-in finite-level uniform quantization function and is equipped with a dynamical internal encoder state co-evolving with the node states. At each step, each node's encoder produces a quantized message with the node state and the current internal encoder state, which will be transmitted to its neighbors through the digital communication link. After receiving the quantized information from the neighbors, each node then decodes/estimates its neighbors' states. Finally, each node updates its state by a synthesis of a consensus step (using the estimated neighboring nodes' states) and a gradient-like step (using its local linear equation).

The key results are summarized as follows:

(i) *Exponential rate as an exact solver.* When the network linear equation admits a unique exact solution, we show that the proposed encoder/decoder powered distributed quantized algorithm drives each node state to that solution asymptotically with an exponential convergence rate. We further give an explicit form of the asymptotic rate of convergence, which is related to the scale and the synchronizability of the network, the number of quantization levels, the dimension of the unknown variable, and the observation matrix. It is shown that a higher convergence rate is possible with higher data rates but is still fundamentally bounded by the inherent network structure and the linear equations. Additionally, we establish the communication complexity in terms of the total number of communication bits required to obtain an ϵ-accurate solution.

(ii) *Sublinear rate as a least-squares solver.* When the network linear equation admits a unique least-squares solution, it is shown that the proposed algorithm can drive the node state to the least-squares solution at a sublinear rate with a decreasing step-size coming from the dynamics of encoder internal states.

Particularly, it is shown that 1 bit of node-to-node communication along each dimension of the decision variable for the linear equation is enough to guarantee the convergence of our proposed algorithm to either exact or least-squares solutions with suitable parameters. Such convergence with 1-bit communication is unconditional on the network size or the network structure, which only affects the convergence speeds.

Related works. The authors of [9] studied the distributed parameter estimation in sensor networks with noisy linear observation models based on the dithered quantization, which indeed added a random dither to the sensor state before quantization to make the sequence of quantization errors to be an i.i.d. sequence of uniformly distributed random variables. The recent work [36] considered a distributed strongly convex optimization based on a random quantizer that is unbiased and has bounded variance. This makes the quantization error a "white" noise, based on which the vanishing solution error is proved in a mean-squared sense. Though the exchanged data in [9, 36] are discrete values, the finiteness of the quantization levels has not been investigated yet, let alone the minimum quantization level. Compared with [9, 36], this work proposes a distributed quantized protocol that can achieve exact convergence with a minimum data rate. The work [13] proposed a distributed algorithm with finite data rate communications to solve distributed strongly convex optimization. The exponential convergence to the optimal solution was proved for a sufficiently large quantization level, but the minimum quantization level is not shown to guarantee convergence. In comparison, we establish the low data rate theorem to show that a minimum quantization level can still guarantee exact convergence, and we further explore the influence of the scale and the synchronizability of the network, the quantization level, and the observation matrix on the exponential factor.

These obtained results serve as assurance for the practical applicability of various network linear equation solvers when digital point-to-point communications are

subject to round-up errors. Generalizations of the scenarios where the solutions of the linear equations are not unique for both exact and least-squares cases are possible along the same line of analysis but are not included in the current paper for ease of presentation. A preliminary version of the results was presented at the IEEE CDC in 2018 [16]. The current work compared to [16] makes a number of improvements and extensions: the major one we would highlight is that we specify how the rate of convergence is influenced by the quantization levels, the scale and the synchronizability of the network, the variable dimension m, as well as the problem structure. In addition, we note in particular that more simulations are carried out to discuss how data rate influences algorithm parameter selection and, thereby, influences the converge rate. Furthermore, we compare convergence rates for different types of communication graphs and give the completed proofs of all results.

The remainder of this paper is organized as follows. Section 2 defines the network linear equation, introduces the node encoders and decoders, and develops a distributed quantized algorithm. Section 3 presents the exact solver along with its convergence analysis and numerical examples. Section 4 further investigates the least-squares case. Finally, concluding remarks are given in section 5.

Notation and terminology. All vectors are column vectors and denoted by bold, lowercase letters, i.e., $\mathbf{a}, \mathbf{b}, \mathbf{c}$, etc.; matrices are denoted with bold, uppercase letters, i.e., $\mathbf{A}, \mathbf{B}, \mathbf{C}$, etc.; sets are denoted with $\mathcal{A}, \mathcal{B}, \mathcal{C}$, etc. Depending on the argument, $|\cdot|$ stands for the absolute value of a real number or the cardinality of a set. The Euclidean norm of a vector is denoted as $\|\cdot\|$. $\otimes$ denotes the Kronecker product. Let $\mathbf{1}_d$ and $\mathbf{0}_d$ denote the d-dimensional column vector with all entries equal to 1 and 0, respectively. Denote by $\mathbf{I}_d$ the $d \times d$ identity matrix. An undirected graph is denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \ldots, N\}$ is a finite set of nodes, and each element in $\mathcal{E}$ is an unordered pair of two distinct nodes in $\mathcal{V}$, called an edge. A path in $\mathcal{G}$ from v_1 to v_p is a sequence of distinct nodes, $v_1, v_2, \ldots v_p$, such that $(v_k, v_{k+1}) \in \mathcal{E}, \forall\, k = 1, \ldots, p-1$. The graph $\mathcal{G}$ is termed *connected* if for any two distinct nodes $i, j \in \mathcal{V}$, there is a path between them. The neighbor set of node i, denoted $\mathcal{N}_i$, is defined as $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. Define the degree matrix $\mathbf{D_e} = \text{diag}\{|\mathcal{N}_1|, \ldots, |\mathcal{N}_N|\}$, where $\text{diag}\{d_1, \ldots, d_N\}$ denotes the diagonal matrix with $d_1, \ldots, d_N$ in the diagonal. Denote by $\mathbf{A}$ the adjacency matrix, where $[\mathbf{A}]_{ij} = 1$ if $j \in \mathcal{N}_j$ and $[\mathbf{A}]_{ij} = 0$ otherwise. Then $\mathbf{L} \triangleq \mathbf{D_e} - \mathbf{A}$ is the Laplacian matrix of the graph $\mathcal{G}$.

2. Problem statement and algorithm design. In this section, we introduce the network linear equation problem and design a distributed quantized algorithm.

2.1. Linear equations over networks. Consider the linear equation

$$\mathbf{z} = \mathbf{H}\mathbf{y} \tag{1}$$

with respect to an unknown variable $\mathbf{y} \in \mathbb{R}^m$, where $\mathbf{H} \in \mathbb{R}^{N \times m}$ and $\mathbf{z} \in \mathbb{R}^N$. Equation (1) has a unique exact solution if $\text{rank}(\mathbf{H}) = m$ and $\mathbf{z} \in \text{span}(\mathbf{H})$; an infinite set of solutions if $\text{rank}(\mathbf{H}) < m$ and $\mathbf{z} \in \text{span}(\mathbf{H})$; and no exact solutions if $\mathbf{z} \notin \text{span}(\mathbf{H})$. When no exact solution exists, a least-squares solution of (1) can be defined via the following optimization problem:

$$\min_{\mathbf{y} \in \mathbb{R}^m} \left\|\mathbf{z} - \mathbf{H}\mathbf{y}\right\|^2, \tag{2}$$

which yields a unique solution $\mathbf{y}^\star = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z}$ if $\text{rank}(\mathbf{H}) = m$. We denote by

$$\mathbf{H} = \begin{pmatrix} \mathbf{h}_1^{\mathrm{T}} \\ \mathbf{h}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{h}_N^{\mathrm{T}} \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{pmatrix},$$

where $\mathbf{h}_i \in \mathbb{R}^m$ with $\mathbf{h}_i^{\mathrm{T}}$ being the ith row vector of $\mathbf{H}$.

Consider a network with N nodes indexed as $\mathcal{V} \triangleq \{1, \ldots, N\}$, where node i has access to the value of $\mathbf{h}_i$ and z_i while without the knowledge of $\mathbf{h}_j$ or z_j from other nodes. The node interaction is described by a connected undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with the corresponding Laplacian matrix denoted by $\mathbf{L}$. The aim of this paper is to develop a distributed method using *quantized* node communications and to explore the corresponding convergence properties with minimal data rate statements.

2.2. Distributed quantized algorithm. The discrete-time is slotted at $k = 0, 1, 2, \ldots$. Node i at time k holds an estimate $\mathbf{x}_i(k) \in \mathbb{R}^m$ for the solution to (1) and exchanges information with its neighbors. Suppose that the communication channels corresponding to each edge in the network have a limited capacity or a finite bandwidth. As such, real-valued data should be quantized before transmitting. Thus, we propose a distributed quantized algorithm, where each node is associated with an encoder while its neighbors possess a corresponding decoder. Let us begin by introducing a uniform quantization function $Q_K(\cdot)$.

DEFINITION 2.1 (quantization function). *A standard uniform quantizer is given by the function* $Q_K(\cdot) : \mathbb{R} \to \{-K, \ldots, -1, 0, 1, \ldots, K\}$, *where*

$$Q_K(z) = \begin{cases} 0 \text{ if } -1/2 \le z \le 1/2, \\ i \text{ if } \frac{2i-1}{2} < z \le \frac{2i+1}{2},\ i = 1, \ldots, K, \\ K \text{ if } z > \frac{2K+1}{2}, \\ -Q_K(-z) \quad \text{if } z < -1/2. \end{cases} \tag{3}$$

There is no need to send any information if the output of the quantizer is zero, so, for a $2K+1$-level quantizer, the communication channel is required to be capable of transmitting $\lceil \log_2(2K) \rceil$ bits. With a slight abuse of notation, we define $Q_K(\mathbf{a})$ for a vector $\mathbf{a} = (a_1, \ldots, a_m)^T \in \mathbb{R}^m$ by $Q_K(\mathbf{a}) = (Q_K(a_1), \ldots, Q_K(a_m))^T$.

Next, we propose an encoder/decoder pair for each node to quantize its state and to estimate its neighbors' states. Suppose all the nodes have a global scaling function $s(k) > 0\ \forall k$, which is a decreasing sequence used to adaptively adjust the encoder. The scaling function will be specified upon the convergence analysis.

Encoder

Node $j \in \mathcal{V}$ recursively generates m-dimensional quantized outputs $\{\mathbf{q}_j(k)\}$ and internal states $\{\mathbf{b}_j(k)\}$ from the exact state sequence $\{\mathbf{x}_i(k)\}$ as follows for any $k \ge 1$:

$$\begin{aligned} \mathbf{q}_j(k) &\triangleq Q_K\left(\frac{1}{s(k-1)}(\mathbf{x}_j(k) - \mathbf{b}_j(k-1))\right), \\ \mathbf{b}_j(k) &\triangleq s(k-1)\mathbf{q}_j(k) + \mathbf{b}_j(k-1), \end{aligned} \tag{4}$$

where $s(k) > 0$ is a global scaling function and the initial value $\mathbf{b}_j(0) = 0$.

Remark 2.2. Note that $\mathbf{b}_j(k)$ is a one-step predictor and the encoder is a difference encoder with a zooming-in scaling $s(k)$ that quantizes the prediction error $\mathbf{x}_j(k) - \mathbf{b}_j(k-1)$, the difference between $\mathbf{x}_j(k)$ and $\mathbf{b}_j(k-1)$, rather than the state $\mathbf{x}_j(k)$. Generally speaking, the amplitude of the prediction error is smaller than that of the state itself, so it can be represented by fewer bits. We use the scaling function $s(k)$ to zoom-in each node's prediction error and require that $s(k)$ decays gradually to make the quantizer persistently excited, such that the nodes gradually increase the accuracy of state recovery of their neighbors. On the other hand, $s(k)$ should be large enough such that the quantizer will not be saturated, in which case the quantization error is bounded. The detailed selection of $s(k)$ will be given subsequently in sections 3 and 4 upon convergence analysis and simulation studies.

Node $j \in \mathcal{V}$ at time k sends its quantized output $\mathbf{q}_j(k)$ to its neighboring node $i \in \mathcal{N}_j$, which then recovers node j's state using the decoder defined as follows.

Decoder

When node $i \in \mathcal{N}_j$ receives the quantized data $\mathbf{q}_j(k)$ from node j, a decoder recursively generates an estimate $\hat{\mathbf{x}}_{ij}(k)$ for $\mathbf{x}_j(k)$ by the following for any $k \geq 1$:

$$\hat{\mathbf{x}}_{ij}(k) \triangleq s(k-1)\mathbf{q}_j(k) + \hat{\mathbf{x}}_{ij}(k-1), \tag{5}$$

where $s(k) > 0$ is a global scaling function and the initial value $\hat{\mathbf{x}}_{ij}(0) = 0$.

Based on the encoder/decoder pair defined in (4) and (5), we now propose the following distributed linear equation solver with quantized communications.

Algorithm 1. Distributed quantized algorithm.

Each node $i \in \mathcal{V}$ at time $k+1$ updates its estimate $\mathbf{x}_i(k+1)$ as follows:

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + h\left[\sum_{j\in\mathcal{N}_i}\left(\hat{\mathbf{x}}_{ij}(k) - \mathbf{b}_i(k)\right) - \gamma(k)\left(\mathbf{h}_i\mathbf{h}_i^\top \mathbf{x}_i(k) - z_i\mathbf{h}_i\right)\right], \tag{6}$$

where $h > 0$ is the control gain, $\gamma(k) > 0$ is the step-size, and $\mathbf{b}_i(k)$ and $\hat{\mathbf{x}}_{ij}(k)$ are produced by the encoder (4) and the decoder (5), respectively.

It is clear that Algorithm 1 merely relies on quantized node communication since $\mathbf{q}_j(k)$ takes its value in the alphabet $\{-K, \ldots, -1, 0, 1, \ldots, K\}$ only. From the second equation of (4), using (5) and the initial conditions for $\hat{\mathbf{x}}_{ij}(0)$ and $\mathbf{b}_j(0)$, we have the following for any $k \geq 0$:

$$\hat{\mathbf{x}}_{ij}(k) = \mathbf{b}_j(k) \quad \forall j \in \mathcal{V},\ \forall i \in \mathcal{N}_j. \tag{7}$$

Note that the algorithm (6) is a discretization of the continuous-time "Consensus + Projection" flow (equation (3)) method in [38], combined with quantization. However, [38] requires continuous-time communications among the neighboring nodes, where nodes need to exchange the exact real-valued states for the execution. This requires the communication channels to have unlimited bandwidth, while in digital networks, communication channels usually have a finite channel capacity. It is worth noting that the proposed Algorithm 1 can resolve such a finite bandwidth problem since it merely requires discrete-time communications with finite bits at each round of communication.

3. Exact solutions. In this section, we consider Algorithm 1 and investigate the case that (1) has a unique solution. We establish the convergence results regarding the quantization levels along with the rate analysis and demonstrate the results with numerical simulations.

3.1. Convergence result. We impose the following assumptions:

A1. There exists a unique solution $\mathbf{y}^*$, i.e., $\text{rank}(\mathbf{H}) = m$ and $\mathbf{z} \in \text{span}(\mathbf{H})$.

A2. $\max_i \|\mathbf{x}_i(0)\|_\infty \le C_x$ for some known positive constant C_x.

A3. $\gamma(k) \equiv 1$, and $s(k) \triangleq s(0)\alpha^k$ for any $k \ge 0$ with $s(0) > 0$ and $\alpha \in (0,1)$.

A2 requires that the upper bound of initial estimates is known. Suppose we set the initial estimate of each node $i \in \mathcal{V}$ as $\mathbf{x}_i(0) = \mathbf{0}_m$. Then it immediately follows that $C_x = 0$. We will specify the selection of $s(0)$ upon convergence statements. For the ease of analysis for Algorithm 1, we assume within this section that

$$\|\mathbf{h}_i\|_2 = 1 \quad \forall i \in \mathcal{V}. \tag{8}$$

This can be easily achieved through the local normalization $\mathbf{h}_i/\|\mathbf{h}_i\|_2$ and $z_i/\|\mathbf{h}_i\|_2$ by each node i. We now introduce some notation for defining circumstances under which the quantizer will not be saturated. Define

$$\begin{aligned} &\mathbf{z}_H \triangleq col\{z_i\mathbf{h}_1, \dots, z_N\mathbf{h}_N\} = \left(z_i\mathbf{h}_1^T, \dots, z_N\mathbf{h}_N^T\right)^T \in \mathbb{R}^{mN}, \\ &\mathbf{H_d} \triangleq \text{diag}\left\{\mathbf{h}_1\mathbf{h}_1^\top, \dots, \mathbf{h}_N\mathbf{h}_N^\top\right\} \in \mathbb{R}^{mN\times mN}, \text{ and } \mathbf{F_d} \triangleq \mathbf{L}\otimes\mathbf{I}_m + \mathbf{H_d}. \end{aligned} \tag{9}$$

Note that both the Laplacian matrix $\mathbf{L}$ and the matrix $\mathbf{H_d}$ are positive semidefinite. By A1 and the condition that the undirected graph $\mathcal{G}$ is connected, the matrix $\mathbf{F_d}$ turns out to be positive definite [38, Lemma 9]; hence, all eigenvalues of $\mathbf{F_d}$ are positive. The eigenvalues of $\mathbf{L}$ can be written in an ascending order as $0 = \lambda_1(\mathbf{L}) < \lambda_2(\mathbf{L}) \le \cdots \le \lambda_N(\mathbf{L})$. Set

$$\rho_h \triangleq 1 - h\lambda_{\min}(\mathbf{F_d}),\ M_{hx} \triangleq \rho_h C_x + \lambda_{\min}^{-1}\left(\sum_{i=1}^N \mathbf{h}_i\mathbf{h}_i^T\right) N\|\mathbf{z}_H\|_\infty + h\|\mathbf{z}_H\|_\infty, \tag{10}$$

$$M(\alpha,h) \triangleq \frac{1+2hd^*}{2\alpha} + \frac{h^2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\alpha(\alpha-\rho_h)},\ \mathcal{K}(\alpha,h) \triangleq \left\lceil M(\alpha,h) - \frac{1}{2}\right\rceil,$$

where $d^* = \max_i |\mathcal{N}_i|$ denotes the maximum degree in $\mathcal{G}$, and $\lambda_{\min}(\mathbf{A})$ and $\lambda_{\max}(\mathbf{A})$ denote the smallest and the largest eigenvalue of a matrix $\mathbf{A}$, respectively.

We are ready to investigate the performance of Algorithm 1 as an exact solver for the network linear equation (1). Proposition 3.1 with the proof deferred to Appendix B.1 validates the nonsaturation of the designed quantizer.

PROPOSITION 3.1 (quantizer nonsaturation). *Let* A1, A2, *and* A3 *hold. Consider Algorithm* 1, *where* $h \in (0, \frac{2}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})})$ *and* $\alpha \in (1-h\lambda_{\min}(\mathbf{F_d}), 1)$. *Then for any* $K \ge \mathcal{K}(\alpha,h)$, *the quantizer will never be saturated provided that* $s(0)$ *satisfies*

$$s(0) > \max\left\{\frac{(1+h\|\mathbf{H_d}\|_\infty)C_x + h\|\mathbf{z}_H\|_\infty}{K+\frac{1}{2}}, \frac{2(\alpha-\rho_h)\left(M_{hx} + hC_x\lambda_N(\mathbf{L})\right)}{h\lambda_N(\mathbf{L})}\right\}. \tag{11}$$

It is noticed that the proposed method (Algorithm 1) is a synchronized algorithm, which requires some sort of networkwide coordination of algorithm parameters. In the following, we provide a few remarks on the control gain h, the initial scaling value $s(0)$, and the scaling factor α for practical implementation.

Remark 3.2. Note by (8) that for any $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)^T, \mathbf{x}_i \in \mathbb{R}^m$ with $\|\mathbf{x}\|_2^2 = 1$,

$$\sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{h}_i \mathbf{h}_i^T \mathbf{x}_i \le \sum_{i=1}^{N} \|\mathbf{x}_i\|_2^2 \|\mathbf{h}_i\|_2^2 = \sum_{i=1}^{N} \|\mathbf{x}_i\|_2^2 = 1.$$

Then the following holds:

$$\begin{aligned} \max_{\mathbf{x}\in\mathbb{R}^{mN}, \|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{F_d} \mathbf{x} &\le \max_{\mathbf{x}\in\mathbb{R}^{mN}, \|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{L} \otimes \mathbf{I}_m \mathbf{x} + \max_{\mathbf{x}\in\mathbb{R}^{mN}, \|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{H_d} \mathbf{x} \\ &= \lambda_N(\mathbf{L}) + \max_{\mathbf{x}\in\mathbb{R}^{mN}, \|\mathbf{x}\|_2=1} \sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{h}_i \mathbf{h}_i^T \mathbf{x}_i \le \lambda_N(\mathbf{L}) + 1 \\ \Rightarrow\ \lambda_{\min}(\mathbf{F_d}) &\le \lambda_{\max}(\mathbf{F_d}) \le \lambda_N(\mathbf{L}) + 1. \end{aligned} \tag{12}$$

Thus, from (12) and $\lambda_N(\mathbf{L}) \le 2d^*$ [17, Lemma A.2] it follows that

$$\frac{2}{\lambda_{\min}(\mathbf{F_d}) + \lambda_{\max}(\mathbf{F_d})} \ge \frac{1}{\lambda_N(\mathbf{L}) + 1} \ge \frac{1}{2d^* + 1}. \tag{13}$$

Then we can set the parameter h within the range $h \in (0, \frac{1}{2d^*+1})$, which merely requires knowledge of the maximum degree of the communication graph $\mathcal{G}$. Note that in the proposed scheme, all nodes use an identical control gain h; this may require additional coordination among the nodes before running the distributed algorithms. Recently, techniques utilizing uncoordinated step-sizes have been proposed in [29] for distributed strongly convex optimization. How to incorporate such a scheme into the distributed quantized method remains a future work.

Remark 3.3. The scaling function $s(k)$ is designed off-line. Suppose we set $\mathbf{x}_i(0) = \mathbf{0}_m$ for each agent $i \in \mathcal{V}$; then $C_x = 0$. Since $\mathbf{F_d}$ is positive definite, we have that

$$\begin{aligned} \lambda_{\min}(\mathbf{F_d}) &= \min_{\mathbf{x}\in\mathbb{R}^{mN}} \frac{\mathbf{x}^T \mathbf{F_d} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \le \min_{\mathbf{x}=(\mathbf{1}_N \otimes \mathbf{I}_m)\mathbf{y}, \mathbf{y}\in\mathbb{R}^m} \frac{\mathbf{x}^T \mathbf{F_d} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\ &= \min_{\mathbf{y}\in\mathbb{R}^m} \frac{\mathbf{y}^T \sum_{i=1}^{N} \mathbf{h}_i \mathbf{h}_i^T \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \lambda_{\min}\left(\sum_{i=1}^{N} \mathbf{h}_i \mathbf{h}_i^T \right). \end{aligned}$$

This combined with (10) produces $M_{hx} \le \lambda_{\min}^{-1}(\mathbf{F_d}) N \|\mathbf{z}_H\|_\infty + h\|\mathbf{z}_H\|_\infty$. By setting $\alpha \triangleq 1 - h\lambda_{\min}(\mathbf{F_d})/2$ and using $\rho_h = 1 - h\lambda_{\min}(\mathbf{F_d})$, we know that (11) holds when

$$s(0) > \|\mathbf{z}_H\|_\infty \max\left\{ \frac{h}{K + \frac{1}{2}}, \frac{1}{\lambda_N(\mathbf{L})} \left(N + h\lambda_{\min}(\mathbf{F_d})\right) \right\}. \tag{14}$$

Recall that $\lambda_N(\mathbf{L}) \ge d^*$ by [17, Lemma A.2]. Then by (12) and (14) it is seen that $s(0)$ given by the following satisfies (11):

$$s(0) > \|\mathbf{z}_H\|_\infty \left(h + \frac{N + h}{d^*} \right), \tag{15}$$

which merely requires the knowledge of $\|\mathbf{z}_H\|_\infty$, the number of nodes in the network, and the maximum degree of the communication graph $\mathcal{G}$. This is a conservative but sufficient selection of $s(0)$, while smaller $s(0)$ might be available in practice. Additionally, the identical zoom-in factor α used by all nodes may require additional coordination among the nodes. Though the theoretical analysis of Algorithm 1 with node-specific scaling function $s_i(k) = s_i(0)\alpha_i^k$ remains unknown, the simulation result shown by Figure 4 in section 3.2 still displays the geometric rate of convergence.

Based on Proposition 3.1, we are able to show that the generated sequences converge to the unique exact solution with a geometric rate. The result is shown in the following theorem, while the proof can be found in Appendix B.2.

THEOREM 3.4 (high data rate). *Suppose* A1*,* A2*, and* A3 *hold. Let* $h \in (0, \frac{2}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})})$ *and* $\alpha \in (1 - h\lambda_{\min}(\mathbf{F_d}), 1)$*. Then for any* $K \geq \mathcal{K}(\alpha, h)$*, along Algorithm* 1 *there holds*

$$\lim_{k\to\infty} x_i(k) = \mathbf{y}^* \quad \forall i \in \mathcal{V} \tag{16}$$

provided that $s(0)$ *satisfies* (11)*. The convergence is exponential with*

$$\|\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*\|_2 \leq \frac{hs(0)\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha - \rho_h)}\alpha^k \quad \forall k \geq 1, \tag{17}$$

where $\mathbf{x}(k) \triangleq col\{\mathbf{x}_1(k), \ldots, \mathbf{x}_N(k)\} = (\mathbf{x}_1^T(k), \ldots, \mathbf{x}_N^T(k))^T$.

Equation (16) in Theorem 3.4 shows that by using an exponentially decaying scaling function and a $\lceil \log_2(\mathcal{K}(\alpha, h)) \rceil$-bit uniform quantizer, Algorithm 1 can ensure asymptotic convergence to the unique solution. It is worth pointing out that for any given α, h, the quantization level $\mathcal{K}(\alpha, h)$ defined by (10) might be sufficient but conservative, while fewer bits could be necessary in practice; see the simulation results of Example 1 in section 3.2. But the value of $\mathcal{K}(\alpha, h)$ gives us some intuition on the relationship between the number of bits required and the control gain h along with the scaling factor α.

Remark 3.5. Equation (17) in Theorem 3.4 gives an estimate of the convergence rate. The convergence rate is faster with a smaller scaling factor α, implying that more bits have to be communicated by (10). Moreover, when $\alpha \to \rho_h$, the required number of bits goes to infinity. Thus, an appropriate selection of the scaling factor α amounts to a trade-off between the convergence rate and communication overhead.

With (10), the quantization level $\mathcal{K}(\alpha, h)$ will tend to infinity as $N \to \infty$ for any fixed α and h. However, the communication channel usually has a finite bandwidth in practice. In order to satisfy this requirement, we aim to use a fixed number of quantization levels at the cost of a possible slower convergence. For any given $K \geq 1$, define

$$\Xi_K \triangleq \left\{(\alpha, h) : h \in \left(0, \frac{2}{\lambda_{\min}(\mathbf{F_d}) + \lambda_{\max}(\mathbf{F_d})}\right), \alpha \in (\rho_h, 1), M(\alpha, h) < K + \frac{1}{2}\right\}. \tag{18}$$

Based on the above definition, we have the following result, for which the proof is given in Appendix B.3.

THEOREM 3.6 (low data rate). *Suppose* A1*,* A2*, and* A3 *hold. Then for any* $K \geq 1$*,* Ξ_K *is nonempty, and for any* $(\alpha, h) \in \Xi_K$*, along Algorithm* 1 *with* $s(0)$ *satisfying* (11) *there holds*

$$\|\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*\|_2 \leq \frac{hs(0)\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha - \rho_h)}\alpha^k \quad \forall k \geq 1.$$

Remark 3.7. From Theorem 3.6 it is clear that we can always design a distributed network linear equation solver to ensure an exponential convergence to the exact solution with a 3-level quantizer (namely, $K = 1$), under which each node sends merely m bits of information to its neighbors at each step.

It is seen from (18) that the set Ξ_K is defined by three nonlinear inequalities, and evidently an explicit solution of these inequalities might be difficult to obtain. Then in the following, we give an explicit method for choosing parameters (α, h) from Ξ_K for any given $K \geq 1$ by introducing a free parameter $\nu \in (0,1)$. For any $\nu \in (0,1)$, let $\alpha_\nu = 1-(1-\nu)h_\nu\lambda_{\min}(\mathbf{F_d})$ and $h_\nu \in (0, h^*_{K,\nu})$, where $h^*_{K,\nu} \triangleq \min\{\frac{2}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})}, \hat{h}_{K,\nu}\}$ with

$$\hat{h}_{K,\nu} \triangleq 2K\nu\lambda_{\min}(\mathbf{F_d})\Bigg(\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d}) + 2\nu\lambda_{\min}(\mathbf{F_d})d^* + \nu(1-\nu)(2K+1)\lambda^2_{\min}(\mathbf{F_d})\Bigg)^{-1}. \tag{19}$$

PROPOSITION 3.8. *Fix $K \geq 1$. Then for any $\nu \in (0,1)$, $(\alpha_\nu, h_\nu) \in \Xi_K$. Conversely, for any $(\alpha, h) \in \Xi_K$, there exists a $\nu > 0$ such that $\alpha = 1-(1-\nu)h\lambda_{\min}(\mathbf{F_d})$ and $h \in (0, h^*_{K,\nu})$.*

The proof of Proposition 3.8 is given in Appendix B.4. Based on Proposition 3.8, to find a pair of parameters $(\alpha, h) \in \Xi_K$, we can simply apply the following procedure: (i) pick $\nu \in (0,1)$; (ii) choose $\alpha = \alpha_\nu$; (iii) compute $h^*_{K,\nu}$ and select any $h \in (0, h^*_{K,\nu})$. Meanwhile, the parameter $\nu \in (0,1)$ can be used to tune the algorithm parameters α and h. In addition, such a selection method can find all feasible algorithm parameters.

Based on the exponential rate of convergence given by (17), it can be easily seen that the number of iterations required to obtain an ϵ-accurate solution $\mathbf{x}$ satisfying $\|\mathbf{x}-\mathbf{1}_N\otimes\mathbf{y}^*\|_2 \leq \epsilon$ is bounded by $\tau(\epsilon)$, where $\tau(\epsilon)$ is defined as in (20). Because the solution $\mathbf{y}^*$ is an m-dimension vector and a $2K+1$-level uniform quantizer $Q_K(\cdot)$ is used in the update, the communication bits required per step at each link is $m\lceil\log_2(2K)\rceil$. Since node i has $|\mathcal{N}_i|$ neighboring nodes, the total number of communication bits required by node i to obtain an ϵ-solution is bounded by $m|\mathcal{N}_i|\lceil\log_2(2K)\rceil\tau(\epsilon)$. Recall that there are $|\mathcal{E}|$ links in the network and the communication is bidirectional at each link; the total number of bits required to obtain an ϵ-accurate solution bounded by $2m|\mathcal{E}|\lceil\log_2(2K)\rceil\tau(\epsilon)$. We conclude these results in Corollary 3.9, which gives a bound on the number of iterations and communication bits required to obtain an ϵ-accurate solution to the linear equation (1).

COROLLARY 3.9 (communication complexity). *For any $K \geq 1$, let $(\alpha, h) \in \Xi_K$ and $s(0)$ satisfy (11). Let $\mathbf{x}(k)$ be generated by Algorithm 1. Then for any $\epsilon > 0$,*

$$\|\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*\|_2 \leq \epsilon \quad \forall k \geq \tau(\epsilon) \triangleq \left\lceil \frac{\ln\left(\epsilon^{-1}\frac{hs(0)\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)}\right)}{\ln(\alpha^{-1})} \right\rceil. \tag{20}$$

The total number of communicated bits required by node i to obtain an ϵ-accurate solution is bounded by $m|\mathcal{N}_i|\lceil log_2(2K)\rceil\tau(\epsilon)$.

The communication complexity stated in Corollary 3.9 holds for any given $K \geq 1$. Sometimes, we aim to choose the quantization level K and algorithmic parameter α, h to minimize the total communication bits, i.e., $\min_{K\geq 1,(\alpha,h)\in\Xi_K}\lceil\log_2(2K)\rceil\tau(\epsilon)$. However, this optimization problem is nonconvex, and the global optimum is generally unavailable. As such, it remains an open question to determine the best selection of K, α, h.

Though the proposed distributed protocol ensures exponential convergence with rate α^k, the parameter α is coupled with another algorithm parameter h while without

explicit dependence on the linear equations and the network. Besides, the number of nodes is relatively large in some applications. This motivates us to investigate the asymptotic property of α as $N \to \infty$ and give a very compendious expression for the asymptotic value of α. The proof of Theorem 3.10 can be found in Appendix B.5.

THEOREM 3.10 (network scalability). *Adopt the same hypotheses as Theorem 3.6. Let $K \geq 1$ and $(\alpha, h) \in \Xi_K$, where Ξ_K is defined by (18). Then*

$$\lim_{N\to\infty} \frac{\inf_{(\alpha,h)\in\Xi_K} \alpha}{\exp\left(-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}\right)} = 1. \tag{21}$$

Remark 3.11. Theorem 3.10 together with (17) suggests that for a large-scale network, the highest possible convergence rate under the proposed protocol tends to scale with $\mathcal{O}\left(\exp\left(-k\Theta_N K\right)\right)$, where $\Theta_N \triangleq \frac{\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}$ is some constant relying only on the number of nodes, the dimension of the solution, the network structure, and the observation matrix.

3.2. Numerical examples.

Example 1. Let $\mathbf{H}$ and $\mathbf{z}$ in the linear equation (1) be given by

$$\mathbf{H} = \begin{pmatrix} 0.9806 & -0.1961 \\ -0.8944 & 0.4472 \\ 0.3939 & -0.9191 \\ 0.8944 & 0.4472 \\ -0.5145 & 0.8575 \end{pmatrix}, \mathbf{z} = \begin{pmatrix} 0.3922 \\ 0.4472 \\ -2.3635 \\ 2.2361 \\ 2.0580 \end{pmatrix}, \tag{22}$$

which then yields a unique exact solution $\mathbf{y}^* = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$. The network structure among the five nodes is shown in Figure 1. Note that $\lambda_N(\mathbf{L}) = 4.1701$ and $\lambda_{\max}(\mathbf{F_d}) = 4.9687$, which satisfies the upper bound on $\lambda_{\max}(\mathbf{F_d})$ given by (12), while $\frac{2}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})} = 0.3867 > \frac{1}{2d^*+1} = 0.1429$, which implies that the estimated upper bound on h given by (13) is relatively conservative.

Validation of Theorem 3.4. Let $h = \frac{1.98}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})} = 0.3828$. Here one can compute $\rho_h = 0.922$. Set $\alpha = 0.96$ so that $\mathcal{K}(\alpha, h) = 135$. Suppose we set the initial value $\mathbf{x}_i(0) = \mathbf{0}_m$ for each node $i \in \mathcal{V}$; then the lower bound $\|\mathbf{z}_H\|_\infty \left(h + \frac{N+h}{d^*}\right)$ of $s(0)$ given by (15) equals 5.1457. As mentioned in Remark 3.3, such a selection of $s(0)$ is conservative, so we set $s(0)$ to be a smaller value $s(0) = 1$. We then implement Algorithm 1 with $h = 0.3828, s(0) = 1, \alpha = 0.96$ for $K = 100, 135, 150$, respectively. Figure 2 displays the trajectories of $w(k) = ||\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*||_2$ along with the theoretical upper bound $B(k) = \frac{hs(0)\alpha^k\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)}$ given by (17). The trajectory

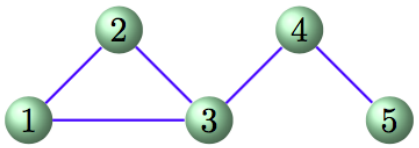

FIG. 1. *Communication graph.*

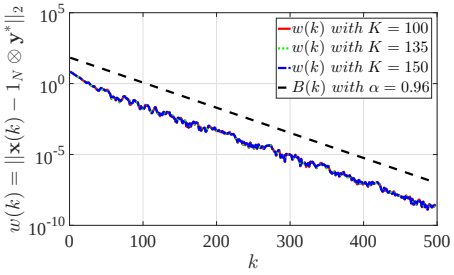

FIG. 2. *Trajectories of* $w(k) = ||\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*||_2$ *along with the upper bound* B_k.

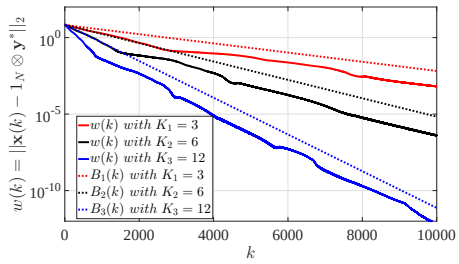

FIG. 3. *Trajectories of* $||\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*||_2$ *and* $B(k)$ *with* $K_1 = 3$, $K_2 = 6$, *and* $K_3 = 12$.

with $K = 135$ verifies that Theorem 3.4 provides a sufficient condition on the data rate to ensure convergence, while the coincidence of the trajectories for $K = 135$ and $K = 150 > \mathcal{K}(\alpha, h)$ implies that with the same algorithm parameters α, h, a quantization level higher than $\mathcal{K}(\alpha, h)$ might not lead to a faster convergence rate. It is noticed that the trajectory for $K = 100 < \mathcal{K}(\alpha, h)$ coincides with that for $K = 135$, which implies some degree of conservativeness in the sufficient conditions of Theorem 3.4.

Validation of Theorem 3.6. Let K be $K_1 = 3$, $K_2 = 6$, and $K_3 = 12$, respectively. We choose $(\alpha, h) \in \Xi_K$ according to Proposition 3.8. Let $\nu = 0.5$; we then choose $(\alpha_1, h_1) = (0.9993, 0.0068)$, $(\alpha_2, h_2) = (0.9986, 0.0136)$, and $(\alpha_3, h_3) = (0.9972, 0.0272)$. We set the initial values $\mathbf{x}(0) = \mathbf{0}_{mN}$ and $s(0) = \|\mathbf{z}_H\|_\infty \left(h + \frac{1+h}{d^*}\right)$, smaller than the bound given by (15), i.e., $s_1(0) = 0.8093, s_2(0) = 0.8307, s_3(0) = 0.8735$ for K_1, K_2, K_3, respectively. The trajectories of $||\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*||_2$ under the three sets of parameters are shown in Figure 3, which demonstrates the exponential convergence of Algorithm 1 to the exact solution. A higher data rate allows us to choose a larger h and a smaller α and therefore leads to a faster convergence rate. Figure 3 is consistent with the upper bound $B(k) = \frac{hs(0)\alpha^k\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)}$ on the rate given by (17) in all three parameter settings.

In addition, we implement Algorithm 1 with node-specific control gain h_i, scaling function $s_i(k) = s_i(0)\alpha_i^k$, and quantization level K_i. We choose a different control gain for each node (see Table 1). Then for each $i = 1, \ldots, 5$, we set $\alpha_i = 1 - h_i\lambda_{\min}(\mathbf{F_d})/2$, $s_i(0) = \|\mathbf{z}_H\|_\infty \left(h_i + \frac{1+h_i}{d^*}\right)$, and $K_i = \mathcal{K}(\alpha_i, h_i)$. The derived parameters for each node are shown in Table 1. We show the trajectory $||\mathbf{x}_i(k) - \mathbf{y}^*||_2$ of each node i in Figure 4, which still displays the exponential rate of convergence. It is seen from Figure 4 that node 5 with $K_5 = 101$ has the fastest rate, while node 1 with $K_1 = 74$ has the slowest rate. This conforms to the intuition that the node with a higher data rate might have a faster rate of convergence.

TABLE 1
Algorithm parameters for each node.

	Node 1	Node 2	Node 3	Node 4	Node 5
h_i	0.22	0.24	0.26	0.28	0.3
α_i	0 .9777	0.9756	0.9736	0.9716	0.9695
$s_i(0)$	1.4811	1.5442	1.6072	1.6702	1.7332
K_i	74	81	87	95	105

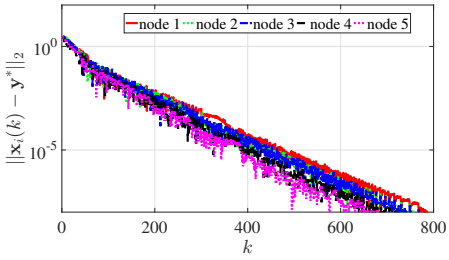

FIG. 4. *Trajectories of* $||\mathbf{x}_i(k) - \mathbf{y}^*||_2$ *for each node* i *with uncoordinated parameters.*

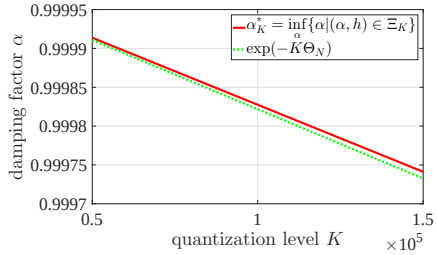

FIG. 5. *The minimal* α_K^* *and* $\exp(-K\Theta_N)$ *with respect to quantization level* K.

Example 2 (Validation of Theorem 3.10). We let $N = 100$ and $m = 5$. We randomly generate a matrix $\mathbf{H}$ and $\mathbf{z}$ such that $\mathbf{z} = \mathbf{Hy}$ has a unique solution. We set $\mathbf{L}$ as the Laplacian of a cycle graph. Then the constant $\Theta_N = \frac{\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}$ is fixed at 2.4910×10^{-9}. We let K increase from $K = 4\times10^4$ to $K = 1.5\times10^5$ in steps of 1000 and search for the minimal α such that $(\alpha, h) \in \Xi_K$ for some $h > 0$ numerically for each K, i.e., $\alpha_K^* = \inf_\alpha\{\alpha|(\alpha, h) \in \Xi_K\}$. Figure 5 shows how α_K^* varies according to the data rate K and implies that a higher data rate allows the selection of a smaller α and hence potentially leads to a faster convergence rate. Figure 5 also displays the trajectory of $\exp(-K\Theta_N)$ with respect to K and shows that $\exp(-K\Theta_N)$ is quite close to α_K^* for $N = 100$, validating Theorem 3.10.

Example 3 (Network structure). Let $N = 100$ and $m = 10$. We randomly generate a matrix $\mathbf{H}$ and $\mathbf{z}$ such that $\mathbf{z} = \mathbf{Hy}$ has a unique solution. We aim to investigate how Θ_N depends on the network structure. Note that for a complete graph, star graph, and cycle graph, Θ_N takes its value as 6.9199×10^{-9}, 1.8553×10^{-9}, and 8.2899×10^{-8}, respectively. This surprisingly indicates cycle graphs produce the fastest convergence compared to complete and star graphs. We then compute Θ_N for Erdős–Rènyi random graphs $\mathcal{G}(N, p)$, where each pair of two nodes is connected with a probability of p, independently of every other edge. We let p increase from 0.1 to

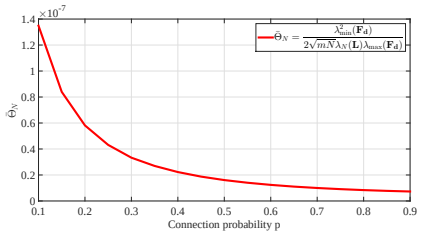

FIG. 6. *The mean $\bar{\Theta}_N$ for random graphs generated with different probability p.*

0.9 at a step of 0.05. For each probability p, we randomly generate 10^3 connected graphs with $\mathcal{G}(N,p)$ and compute the mean $\bar{\Theta}_N$. Figure 6 shows how $\bar{\Theta}_N$ varies along with probability p, which decreases as the connection probability p increases. This implies that α_K^*, the fastest possible convergence rate under a fixed data rate K, might increase with the increasing of the network connectivity.

3.3. Discussion: Improve robustness with damping. Convergence of Algorithm 1 relies on the equivalence between node i's decoder output $\hat{\mathbf{x}}_{ij}(k)$ of its neighbor j' state and node j's one-step prediction $\mathbf{b}_j(k)$, which is characterized by (7). The theoretical and numerical results have shown the effectiveness of Algorithm 1 when (7) is satisfied. In fact, (7) holds when the encoder/decoder update (4)–(5) is exact and the following initialization condition is satisfied:

$$\mathbf{b}_j(0)=0,\ \hat{\mathbf{x}}_{ij}(0)=0\ \ \forall j\in\mathcal{V},\forall i\in\mathcal{N}_j. \tag{23}$$

However, there could exist initialization errors in (23). And due to round-off noises in the storage and manipulation of real-valued vectors in digital computers, (4)–(5) may not be executed exactly. With initialization errors in (23) and the round-off noises, the update of $\mathbf{b}_j(k)$, $\hat{\mathbf{x}}_{ij}(k)$ in encoder/decoder (4)–(5) is changed to

$$\begin{aligned}&\mathbf{b}_j(0)=\mathbf{I}_j^e,\ \hat{\mathbf{x}}_{ij}(0)=\mathbf{I}_{ij}^e\ \forall j\in\mathcal{V},\ \forall i\in\mathcal{N}_j,\\ &\mathbf{b}_j(k)\triangleq s(k-1)\mathbf{q}_j(k)+\mathbf{b}_j(k-1)+\varepsilon_j^b(k),\\ &\hat{\mathbf{x}}_{ij}(k)\triangleq s(k-1)\mathbf{q}_j(k)+\hat{\mathbf{x}}_{ij}(k-1)+\varepsilon_{ij}^x(k).\end{aligned} \tag{24}$$

The initialization errors $\mathbf{I}_j^e,\mathbf{I}_{ij}^e$ and round-off noises $\varepsilon_j^b(k),\varepsilon_{ij}^x(k)$ will persist and accumulate during the execution of the algorithm.

Performance of Algorithm 1 with initialization errors and round-off noises. We continue to use the same $\mathbf{H}$ and $\mathbf{z}$ in (22). We set $h=0.0193$, $\alpha=0.998$, $K=64$, and $s(0)=1$. The initialization errors $\mathbf{I}_j^e$ and $\mathbf{I}_{ji}^e$ are independent across the nodes and are randomly drawn from a uniform distribution on $[0,0.5]$. The round-off noises $\varepsilon_j^b(k)$, $\varepsilon_{ji}^x(k)$ are mutually independent random independent and identically distributed (i.i.d.) sequences with each value drawn from a uniform distribution on $[-1,1]\times 10^{-4}$. Figure 7 shows that Algorithm 1 with (24) cannot ensure convergence when there exist initialization errors or round-off noises. In fact, the error is substantial compared with the noise magnitude and the value $||\mathbf{1}_N\otimes\mathbf{y}^*||_2$.

We propose to improve algorithm robustness by adding a damping term to the encoder/decoder, where $\mathbf{b}_j(k)$ and $\hat{\mathbf{x}}_{ij}(k)$ are updated with

$$\begin{aligned}&\mathbf{b}_j(0)=\mathbf{I}_j^e,\ \hat{\mathbf{x}}_{ij}(0)=\mathbf{I}_{ij}^e\ \forall j\in\mathcal{V},\ \forall i\in\mathcal{N}_j,\\ &\mathbf{b}_j(k)\triangleq s(k-1)\mathbf{q}_j(k)+\varrho\mathbf{b}_j(k-1)+\varepsilon_j^b(k),\\ &\hat{\mathbf{x}}_{ij}(k)\triangleq s(k-1)\mathbf{q}_j(k)+\varrho\hat{\mathbf{x}}_{ij}(k-1)+\varepsilon_{ij}^x(k),\end{aligned} \tag{25}$$

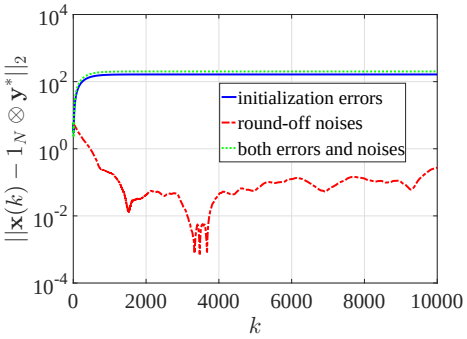

FIG. 7. *Trajectories of* $||\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*||_2$ *generated by Algorithm* 1 *for the following cases:* (i) *there exist initialization errors;* (ii) *there exist round-off noises;* (iii) *there exist both initialization errors and round-off noises.*

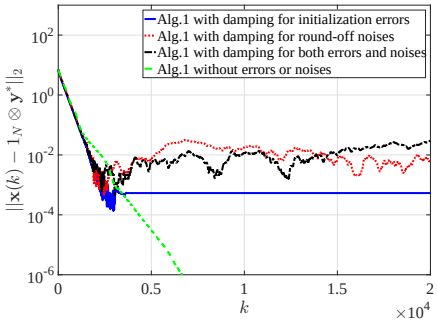

FIG. 8. *Trajectories of* $||\mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*||_2$ *of Algorithm* 1 *with a damped encoder/decoder* (25) *with* (i) *initialization errors;* (ii) *round-off noises;* (iii) *both errors and noises; and* (iv) *Algorithm* 1 *without errors or noises.*

where $\varrho \in (0,1)$ is a damping factor, $\mathbf{I}_j^e, \mathbf{I}_{ij}^e$ are initialization errors, and $\varepsilon_j^b(k), \varepsilon_{ij}^x(k)$ are round-off noises. We run Algorithm 1 with a damped encoder/decoder (25) when there are initialization errors and round-off noises, and we run Algorithm 1 with the encoder and decoder (4)–(5) where there are no initialization errors and round-off noises, both with the same algorithm parameters. The damping factor is set as $\varrho = 0.95$. Figure 8 displays the simulation results, which show that (i) the damping can significantly reduce but not fully eliminate the affect of initialization errors, and (ii) the effect of round-off noises can be tolerated in the sense that the iterates will converge to a neighborhood of the exact solution within a distance of similar magnitude to the round-off noises.

The formal convergence analysis of Algorithm 1 with a damped encoder/decoder update (25) is challenging because there will be a nonlinear coupling between the damping factor ϱ and other parameters, and the initialization errors and round-off noises as well as ϱ will enter the update equation of $\mathbf{x}(k)$ and $\mathbf{b}(k)$ given by (A.4) and (A.5). Therefore, we leave the theoretical analysis of (25) as a future research problem.

4. Least-squares solver. In this section, we investigate the case $\text{rank}(\mathbf{H}) = m$ and $\mathbf{z} \notin \text{span}(\mathbf{H})$. Then (1) does not have exact solutions, while a least-squares solution is defined as the solution to the optimization problem (2). Considering

Algorithm 1, we show its convergence results regarding the quantization level along with the data rate analysis and demonstrate the results with numerical simulations.

4.1. Convergence results. Assumptions A1 and A3 are no longer in force. Instead, we impose the following conditions on the matrix $\mathbf{H}$, the step-size $\gamma(k)$, and the scaling function $s(k)$:

A4. $\operatorname{rank}(\mathbf{H}) = m$ and $\mathbf{z} \notin \operatorname{span}(\mathbf{H})$.

A5. (i) $\gamma(0) = 1$, $\gamma(k)$ strictly monotonely decreases to zero, $\sum_{k=1}^{\infty} \gamma(k) = \infty$; (ii) $s(k) = s_r\gamma(k)$ for some $s_r > 0$; and (iii) $1 < \beta(k+1) < \beta(k)$ for any $k \geq 0$, where $\beta(k) \triangleq \frac{\gamma(k)}{\gamma(k+1)}$.

Remark 4.1. We give an example to show how to choose $\gamma(k)$ to make A5 hold. Set $\gamma(k) = \frac{k_0^\delta}{(k+k_0)^\delta}$ for some $\delta \in (\frac{1}{2}, 1]$, where $k_0 = \frac{1}{\beta(0)^{1/\delta}-1}$. Then A5(i) holds since $\gamma(0) = 1, \gamma(k) \downarrow 0$ and $\sum_{k=1}^{\infty} \gamma(k) = \infty$. By the definition, we obtain that

$$\beta(k) = \frac{\gamma(k)}{\gamma(k+1)} = \frac{(k+k_0+1)^\delta}{(k+k_0)^\delta} = \left(1 + \frac{1}{k+k_0}\right)^\delta > 1.$$

Then $\beta(k)$ monotonely decreases and $(1+1/k_0)^\delta = \beta(0)$. Thus, A5(iii) holds. We will specify the selection of s_r upon convergence statements.

Let $h \in \left(0, \frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}\right)$ and $\beta(0) \in \left(1, \frac{1}{1-h\lambda_2(\mathbf{L})}\right)$. We then define

$$\begin{aligned} &\hat{\rho}_h \triangleq 1 - h\lambda_2(\mathbf{L}),\ M'(h,\beta(0)) \triangleq (1/2 + hd^*)\beta(0) + 2hM_2(h,\beta(0)), \\ &\mathcal{K}'(h,\beta(0)) \triangleq \left\lceil M'(h,\beta(0)) - \frac{1}{2} \right\rceil, \end{aligned} \tag{26}$$

where $M_2(h,\beta(0))$ is defined as follows:

$$M_2(h,\beta(0)) \triangleq \beta(0)\sqrt{mN}\left(\frac{h\lambda_N(\mathbf{L})^2}{2(1/\beta(0)-\hat{\rho}_h)} + \frac{\lambda_N(\mathbf{L})}{\lambda_{\min}(\mathbf{F_d})}\left(\|\mathbf{H_d}\|_\infty + \frac{h\lambda_N(\mathbf{L})\|\mathbf{H_d}\|_2}{1/\beta(0)-\hat{\rho}_h}\right)\right) \tag{27}$$

with $\mathbf{H_d}$ and $\mathbf{F_d}$ defined as in (9). We further define

$$\begin{aligned} M_1(h,\beta(0)) \triangleq &\left(\sqrt{mN}C_x(1+h\lambda_N(\mathbf{L})) + \frac{2\|\mathbf{z}_H\|_2}{\lambda_{\min}(\mathbf{F_d})}\right)\left(\|\mathbf{H_d}\|_\infty + \frac{h\lambda_N(\mathbf{L})\|\mathbf{H_d}\|_2}{1/\beta(0)-\hat{\rho}_h}\right) \\ &+ \|\mathbf{z}_H\|_\infty + \lambda_N(\mathbf{L})\left(\sqrt{mN}C_x\left(1+h\beta(0)\lambda_N(\mathbf{L})\right) + \frac{h\|\mathbf{z}_H\|_2}{1/\beta(0)-\hat{\rho}_h}\right). \end{aligned}$$

We are now ready to state the main result of this section.

PROPOSITION 4.2 (quantizer nonsaturation). *Suppose* A2, A4, *and* A5 *hold. Consider Algorithm* 1, *where*

$$\beta(0) \in \left(1, \frac{1}{1-h\lambda_2(\mathbf{L})}\right), h \in \left(0, \min\left\{\frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{F_d})}\right\}\right).$$

Then for any given $K \geq \mathcal{K}'(h,\beta(0))$, *the quantizer will never be saturated provided that*

$$s_r > \max\left\{\frac{C_x + h(C_x\|\mathbf{H_d}\|_\infty + \|\mathbf{z}_H\|_\infty)}{K+\frac{1}{2}}, M_1(h,\beta(0))/M_2(h,\beta(0))\right\}. \tag{28}$$

Proposition 4.2 establishes the nonsaturation of the uniform quantizer, for which the proof is given in Appendix C.1. Although the least-squares problem (2) seems like a special case of distributed optimization, the main challenge lies in the fact that gradients of the quadratic function associated with each node cannot be assumed to be globally bounded a priori, a key technical assumption for the convergence analysis of distributed (sub)gradient optimization [31, 47]. This is because the gradient function takes a linear form of the generated sequence $\mathbf{x}(k)$, which might be unbounded with inappropriate algorithmic parameters. Thus, the main effort of the proof lies in choosing the parameters and proving the boundedness of the generated sequence.

Remark 4.3. We now discuss how to select the algorithm parameter h. For ease of understanding, suppose that $\|\mathbf{h}_i\|_2 = 1$ for any $i \in \mathcal{V}$. Note that $\frac{1}{\lambda_{\min}(\mathbf{F_d})} \geq \frac{1}{\lambda_N(\mathbf{L})+1}$ by (12), and $\frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})} \geq \frac{1}{\lambda_N(\mathbf{L})}$. Then from $\lambda_N(\mathbf{L}) \leq 2d^*$ [17, Lemma A.2] it follows that $\min\left\{\frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{F_d})}\right\} \geq \frac{1}{\lambda_N(\mathbf{L})+1} \geq \frac{1}{2d^*+1}$. Thus, the sufficient condition imposed on h holds if h satisifies $h \in (0, \frac{1}{2d^*+1})$, which merely requires the knowledge of the maximum degree of the communication graph $\mathcal{G}$.

Based on Proposition 4.2, we are able to show in the following theorem that the iterates generated by Algorithm 1 will asymptotically converge to the unique least-squares solution ${\mathbf{y}_{\mathrm{LS}}}^*$. Its proof is shown in Appendix C.2.

THEOREM 4.4 (high data rate). *Suppose* A2*,* A4*, and* A5 *hold. Let* $h \in (0, \min\left\{\frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{F_d})}\right\})$ *and* $\beta(0) \in (1, 1/\hat{\rho}_h)$ *with* $\hat{\rho}_h$ *defined as in* (26)*. Then for any given* $K \geq \mathcal{K}'(h, \beta(0))$*, along Algorithm* 1 *there hold*

$$\lim_{k\to\infty} \mathbf{x}_i(k) = {\mathbf{y}_{\mathrm{LS}}}^* \triangleq (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z} \quad \forall i \in \mathcal{V}, \tag{29}$$

$$\sup_{k\geq 1} \frac{\|\mathbf{x}_i(k) - {\mathbf{y}_{\mathrm{LS}}}^*\|_\infty}{\gamma(k)} < \infty \quad \forall i \in \mathcal{V}, \tag{30}$$

provided that s_r *satisfies* (28)*.*

Remark 4.5. Note by Theorem 4.4 that a slow rate of convergence is obtained by Algorithm 1 with decreasing step-sizes for the least-squares solver, as opposed to the exponential convergence of the exact solver shown in Theorem 3.4 for Algorithm 1 with a constant step-size. In fact, even with an unquantized communication channel, the primal domain algorithm cannot guarantee exact convergence with constant step-sizes for the distributed least-squares problem. It is, however, noted in [15, 40] that the exact convergence or even the exponential rate of convergence can be obtained using primal-dual domain algorithms. Besides, the distributed schemes in [28, 33] with gradient tracking can achieve exponential convergence for strongly convex optimization. By combining such schemes with our quantized protocol, we might be able to find the least-squares solution with finite data rates at an exponential rate. The randomized iterative least-squares solver proposed in [50], making the estimates exponentially converge in mean square to the minimum Euclidean norm solution, might be amenable to a distributed implementation. This provides another possible avenue for designing a distributed quantized least-squares solver with an exponential convergence. We leave the problem of designing a least-squares solver with nondecreasing step-size for future research.

Similar to Theorem 3.6 for the exact solver case, the following theorem shows that we can still design a distributed protocol for the least-squares solver to converge

with 3-level quantizers, which uses the minimum number of quantization levels. For the theorem statement, we define the following set Ξ'_K for any given $K \geq 1$:

$$\Xi'_K \triangleq \Bigg\{ (h, \beta(0)) : h \in \left(0, \min\left\{ \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{F_d})} \right\}\right), \\ \beta(0) \in \big(1, 1/(1 - h\lambda_2(\mathbf{L}))\big), M'(h, \beta(0)) \leq K + \frac{1}{2} \Bigg\}. \tag{31}$$

THEOREM 4.6 (low data rate). *Suppose* A2, A4, *and* A5 *hold. Then for any* $K \geq 1$, Ξ'_K *is nonempty, and for any* $(h, \beta(0)) \in \Xi'_K$, *along Algorithm* 1 *with* s_r *satisfying* (28) *there holds*

$$\sup_{k \geq 1} \frac{\|\mathbf{x}_i(k) - \mathbf{y}_{\mathrm{LS}}{}^*\|_\infty}{\gamma(k)} < \infty \quad \forall i \in \mathcal{V}.$$

The proof of Theorem 4.6 is given in Appendix C.3.

Remark 4.7. Similar to Proposition 3.8, we can also give an explicit method for choosing algorithm parameters. To find a pair of parameters $(h, \beta(0)) \in \Xi'_K$, we can apply the following procedure: (i) pick $\nu \in (0, 1)$; (ii) choose $\beta(0) = \beta_\nu(0) \triangleq \frac{1}{1-(1-\nu)h\lambda_2(\mathbf{L})}$; (iii) compute $h^*_{K,\nu} \triangleq \min\{\frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{P_d})}, \hat{h}_{K,\nu}\}$ with

$$\hat{h}_{K,\nu} \triangleq 2K\nu\lambda_{\min}(\mathbf{F_d})\Big(2d^*\nu\lambda_{\min}(\mathbf{F_d}) + (2K+1)\nu(1-\nu)\lambda_{\min}(\mathbf{F_d})\lambda_2(\mathbf{L}) \\ + 2\sqrt{mN}\lambda_N(\mathbf{L})\left(2\nu\|\mathbf{H_d}\|_\infty + \kappa_N\big(2\|\mathbf{H_d}\|_2 + \lambda_{\min}(\mathbf{F_d})\big)\right)\Big)^{-1}, \ \kappa_N \triangleq \frac{\lambda_N(\mathbf{L})}{\lambda_2(\mathbf{L})},$$

and select any $h \in (0, h^*_{K,\nu})$. Meanwhile, the parameter $\nu \in (0, 1)$ can be used to tune the parameters $\beta(0)$ and h. Additionally, such a selection method can find all feasible algorithm parameters.

4.2. Numerical examples.

Example 4. Let $\mathbf{H}, \mathbf{z}$ be given as follows:

$$\mathbf{H} = \begin{pmatrix} 1.7889 & -1.0764 \\ -1.0764 & 0.1903 \\ 0.4707 & 0.1008 \\ 0.8356 & -0.1716 \\ 0.5978 & -1.6668 \end{pmatrix}, \mathbf{z} = \begin{pmatrix} -0.2854 \\ 1.2038 \\ 1.1032 \\ 0.7088 \\ -0.9495 \end{pmatrix};$$

then the unique least-squares solution $\mathbf{y}^* \triangleq \arg\min ||\mathbf{z} - \mathbf{H}\mathbf{y}||^2 = (0.1415, 0.6391)^T$. The nodes again communicate according to the graph shown in Figure 1.

Validation of Theorem 4.4. Set $h = 0.0853$ and $\gamma(k) = (\frac{26}{k+26})^{0.85}$ such that $\beta(0) \in (1, \hat{\rho}_h^{-1})$. Hence, $\mathcal{K}(h, \beta(0)) = 870$. We set $K_1 = 900$, $K_2 = 300$, and $K_3 = 1800$, respectively. We set $s_r = 0.82$ to meet (28) in all three cases. We then run Algorithm 1 with the quantization levels K_1, K_2, and K_3, respectively, while with the same parameters $h, \gamma(k)$. The simulation results are displayed in Figure 9, which shows that the trajectories of $||\mathbf{x}(k) - 1_N \otimes \mathbf{y}||_2^2$ coincide in all three cases. It then implies that (i) once the sufficient condition of Theorem 4.4 is satisfied, increasing the data rate solely cannot speed up convergence; (ii) the condition in Theorem 4.4 is sufficient for convergence but is not necessary. Figure 9 also shows the trajectory of $\frac{||\mathbf{x}(k)-\mathbf{y}^*||_\infty}{\gamma(k)}$, which verifies the convergence rate described by (30).

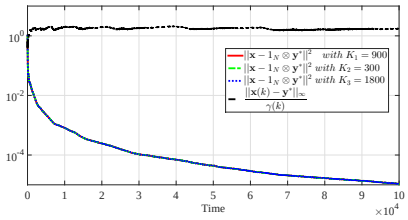

FIG. 9. *Trajectories of* $||\mathbf{x}(k)-\mathbf{1}_N\otimes\mathbf{y}^*||^2$ *under* $K=300,900,1800$.

TABLE 2
Parameter settings.

	k_0	δ	h	s_r
$K=10$	120	0.85	0.0055	0.9583
$K=30$	36	0.75	0.0164	0.6934
$K=90$	9	0.55	0.0492	0.6968

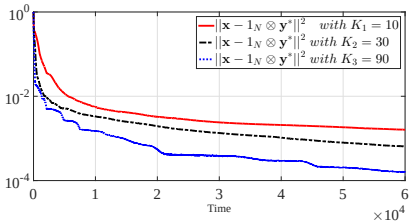

FIG. 10. *Trajectories of* $||\mathbf{x}(k)-\mathbf{1}_N\otimes\mathbf{y}^*||^2$ *for* $K=10,30,90$ *with the algorithm parameters chosen in Table* 2.

Validation of Theorem 4.6. We set the quantization level K to be $K_1=10$, $K_2=30$, and $K_3=90$, respectively. Then we utilize Remark 4.7 to select algorithm parameters h and $s(k)=s_r\gamma(k)=\frac{s_r k_0^\delta}{(k+k_0)^\delta}$ such that $(h,\beta(0))\in\Xi_K'$ and s_r satisfies (28) for the three cases. By setting $\nu=0.5$, the derived parameters for the three cases are given in Table 2. Figure 10 shows the trajectories of $||\mathbf{x}(k)-\mathbf{1}_N\otimes\mathbf{y}^*||^2$ for the three cases. It demonstrates the convergence of the algorithm with the chosen parameters, verifying Theorem 4.6. It also shows that with a higher data rate, the convergence could be faster if algorithm parameters are properly chosen.

5. Conclusions. We have studied solving linear equations over a network subject to digital node communications. We propose a node encoder/decoder pair, based on which a distributed quantized algorithm is designed. For the unique exact solution case, the proposed encoder/decoder powered algorithm drives each node state to the solution asymptotically at an exponential rate. For the unique least-squares solution case, the same encoder/decoder pair enables the algorithm to compute such a solution with a properly selected time-varying step-size. A minimal data rate is shown to be enough for the desirable convergence for both cases. These results suggest the practical applicability of various network linear equation solvers in the literature.

Though we have established the communication complexity to obtain an exact solution with a prescribed accuracy for any given quantization level, it remains an open question to determine the best selection of the quantization level. Another future direction is to investigate the primal-dual type algorithm or gradient tracking

scheme for the network linear equation with a least-squares solution, under which the exponential rate of convergence might be obtainable with a constant step-size under quantized communications. It may also be of interest to investigate the asynchronous scheme under uncoordinated step-sizes and quantization levels and explore convergence conditions of the proposed algorithm in the presence of (random or deterministic) switching communication graphs.

Appendix A. Preliminary lemma.

LEMMA A.1. *Consider Algorithm* 1*, where* $\mathbf{b}_i(k)$ *and* $\hat{\mathbf{x}}_{ij}(k)$ *are defined by the encoder* (4) *and the decoder* (5)*, respectively. Define*

$$\mathbf{e}_i(k) \triangleq \mathbf{x}_i(k) - \mathbf{b}_i(k),\ \mathbf{e}(k) \triangleq col\{\mathbf{e}_1(k), \dots, \mathbf{e}_N(k)\},\ \boldsymbol{\varepsilon}(k) \triangleq \frac{\mathbf{e}(k)}{s(k)}, \tag{A.1a}$$

$$\mathbf{P}(k) \triangleq \mathbf{I}_{mN} - h\left(\mathbf{L} \otimes \mathbf{I}_m + \gamma(k)\mathbf{H_d}\right), \tag{A.1b}$$

$$\boldsymbol{\eta}(k) \triangleq (\mathbf{D} \otimes \mathbf{I}_m)\,\mathbf{x}(k)/\gamma(k) \text{ with } \mathbf{D} \triangleq \mathbf{I}_N - \frac{\mathbf{1}_N\mathbf{1}_N^T}{N}, \tag{A.1c}$$

where $\mathbf{H_d}$ *is defined as in* (9)*. Then the following hold with* $\mathbf{z}_H$ *defined by* (9)*:*

$$\mathbf{x}(k+1) = \mathbf{P}(k)\mathbf{x}(k) + h\big(s(k)\mathbf{L} \otimes \mathbf{I}_m\boldsymbol{\varepsilon}(k) + \gamma(k)\mathbf{z}_H\big), \tag{A.2a}$$

$$\begin{aligned}\boldsymbol{\eta}(k+1) = \frac{\gamma(k)}{\gamma(k+1)}\Big(&\left(\mathbf{I}_{mN} - h\mathbf{L} \otimes \mathbf{I}_m\right)\boldsymbol{\eta}(k)\\ &+ \frac{hs(k)}{\gamma(k)}\mathbf{L} \otimes \mathbf{I}_m\boldsymbol{\varepsilon}(k) + h\mathbf{D} \otimes \mathbf{I}_m\left(\mathbf{z}_H - \mathbf{H_d}\mathbf{x}(k)\right)\Big),\end{aligned} \tag{A.2b}$$

$$\boldsymbol{\varepsilon}(k+1) = \frac{s(k)}{s(k+1)}\big(\boldsymbol{\theta}(k) - Q_K\left(\boldsymbol{\theta}(k)\right)\big) \text{ with} \tag{A.2c}$$

$$\boldsymbol{\theta}(k) \triangleq \big(\mathbf{I}_{mN} + h\mathbf{L} \otimes \mathbf{I}_m\big)\boldsymbol{\varepsilon}(k) - h\frac{\gamma(k)}{s(k)}\big(\mathbf{L} \otimes \mathbf{I}_m\boldsymbol{\eta}(k) + \mathbf{H_d}\mathbf{x}(k) - \mathbf{z}_H\big). \tag{A.2d}$$

Proof. Since $j \in \mathcal{N}_i$ is equivalent to $i \in \mathcal{N}_j$, by using $\hat{\mathbf{x}}_{ij}(k) = \mathbf{b}_j(k)$ (equation (7)) and $\mathbf{e}_i(k) = \mathbf{x}_i(k) - \mathbf{b}_i(k)$, we have the following:

$$\begin{aligned}\sum_{j\in\mathcal{N}_i}\big(\hat{\mathbf{x}}_{ij}(k) - \mathbf{b}_i(k)\big) &= \sum_{j\in\mathcal{N}_i}\big(\mathbf{b}_j(k) - \mathbf{b}_i(k)\big)\\ &= \sum_{j\in\mathcal{N}_i}\big[\left(\mathbf{x}_j(k) - \mathbf{x}_i(k)\right) - \left(\mathbf{e}_j(k) - \mathbf{e}_i(k)\right)\big].\end{aligned} \tag{A.3}$$

Then we obtain a variant of (6):

$$\mathbf{x}(k+1) = \mathbf{x}(k) - h\mathbf{L} \otimes \mathbf{I}_m\mathbf{x}(k) + h\mathbf{L} \otimes \mathbf{I}_m\mathbf{e}(k) - h\gamma(k)\left(\mathbf{H_d}\mathbf{x}(k) - \mathbf{z}_H\right). \tag{A.4}$$

Hence (A.2a) holds by $\mathbf{e}(k) = s(k)\boldsymbol{\varepsilon}(k)$ and the definition of $\mathbf{P}(k)$ in (A.1b). Note that $\mathbf{DL} = \mathbf{L}$ and $\mathbf{D}(\mathbf{I}_N - h\mathbf{L}) = (\mathbf{I}_N - h\mathbf{L})\mathbf{D}$. Then by multiplying both sides of (A.4) on the left with $\frac{(\mathbf{D}\otimes\mathbf{I}_m)}{\gamma(k+1)}$, using $\mathbf{e}(k) = s(k)\boldsymbol{\varepsilon}(k)$ and $\boldsymbol{\eta}(k) = (\mathbf{D} \otimes \mathbf{I}_m)\,\mathbf{x}(k)/\gamma(k)$, we obtain (A.2b). From (4) it follows that

$$\mathbf{b}(k+1) = s(k)Q_K\left(\frac{\mathbf{x}(k+1) - \mathbf{b}(k)}{s(k)}\right) + \mathbf{b}(k). \tag{A.5}$$

By subtracting $\mathbf{b}(k)$ from both sides of (A.4), using (A.1a) and $\mathbf{L} \otimes \mathbf{I}_m\mathbf{x}(k) = \mathbf{L} \otimes \mathbf{I}_m\left(\mathbf{D} \otimes \mathbf{I}_m\right)\mathbf{x}(k) = \gamma(k)\mathbf{L} \otimes \mathbf{I}_m\boldsymbol{\eta}(k)$, we obtain that

$$\begin{aligned}\mathbf{x}(k+1)-\mathbf{b}(k) &= \big(\mathbf{I}_{mN}+h\mathbf{L}\otimes\mathbf{I}_m\big)\mathbf{e}(k)-h\mathbf{L}\otimes\mathbf{I}_m\mathbf{x}(k)-h\gamma(k)(\mathbf{H_d}\mathbf{x}(k)-h\mathbf{z}_H)\\ &= s(k)\big(\mathbf{I}_{mN}+h\mathbf{L}\otimes\mathbf{I}_m\big)\boldsymbol{\varepsilon}(k)-h\gamma(k)\big(\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\eta}(k)+\mathbf{H_d}\mathbf{x}(k)-\mathbf{z}_H\big)\overset{\text{(A.2d)}}{=}s(k)\boldsymbol{\theta}(k).\end{aligned}$$

Recalling that $\mathbf{e}(k+1)=\mathbf{x}(k+1)-\mathbf{b}(k+1)$ together with (A.5), we then have that

$$\mathbf{e}(k+1)=\mathbf{x}(k+1)-\mathbf{b}(k)-s(k)Q_K\left(\frac{\mathbf{x}(k+1)-\mathbf{b}(k)}{s(k)}\right)=s(k)\big(\boldsymbol{\theta}(k)-Q_K\left(\boldsymbol{\theta}(k)\right)\big).$$

Hence, by dividing both sides of the above equation by $s(k+1)$, we obtain (A.2c). ∎

Appendix B. Proof of results stated in section 3.

B.1. Proof of Proposition 3.1. We first give a reformulation of Algorithm 1 based on Lemma A.1. Define

$$\mathbf{w}_i(k)\triangleq\mathbf{x}_i(k)-\mathbf{y}^*,\ \mathbf{w}(k)=\mathbf{x}(k)-\mathbf{1}_N\otimes\mathbf{I}_m\mathbf{y}^*,\ \text{and}\ \boldsymbol{\omega}(k)\triangleq\frac{\mathbf{w}(k)}{s(k)}. \tag{B.1}$$

By subtracting $\mathbf{1}_N\otimes\mathbf{I}_m\mathbf{y}^*$ from both sides of (A.2a), by using $\gamma(k)\equiv1$ and (A.1b), we obtain that

$$\mathbf{w}(k+1)=\mathbf{w}(k)-h\left(\mathbf{L}\otimes\mathbf{I}_m\right)\mathbf{x}(k)-h(\mathbf{H_d}\mathbf{x}(k)-\mathbf{z}_H)+hs(k)\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\varepsilon}(k). \tag{B.2}$$

From $\mathbf{L}\mathbf{1}_N=\mathbf{0}_N$ it follows that

$$\mathbf{L}\otimes\mathbf{I}_m\mathbf{w}(k)=\mathbf{L}\otimes\mathbf{I}_m\mathbf{x}(k)-\left(\mathbf{L}\mathbf{1}_N\otimes\mathbf{I}_m\right)\mathbf{y}^*=\mathbf{L}\otimes\mathbf{I}_m\mathbf{x}(k). \tag{B.3}$$

Recall that $\mathbf{h}_i^T\mathbf{y}^*=z_i$ for each $i\in\mathcal{V}$. Then

$$\mathbf{h}_i\mathbf{h}_i^\top\mathbf{w}_i(k)=\mathbf{h}_i\mathbf{h}_i^\top\left(\mathbf{x}_i(k)-\mathbf{y}^*\right)=\mathbf{h}_i\mathbf{h}_i^\top\mathbf{x}_i(k)-\mathbf{h}_iz_i. \tag{B.4}$$

This incorporated with (B.2) and (B.3) produces

$$\mathbf{w}(k+1)=\big(\mathbf{I}_{mN}-h(\mathbf{L}\otimes\mathbf{I}_m+\mathbf{H_d})\big)\mathbf{w}(k)+hs(k)\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\varepsilon}(k), \tag{B.5}$$

where $\mathbf{H_d}$ is defined by (9). Dividing both sides of the above equation by $s(k+1)$, using $s(k)=s(0)\alpha^k$ and $\boldsymbol{\omega}(k)=\frac{\mathbf{w}(k)}{s(k)}$, we obtain that

$$\boldsymbol{\omega}(k+1)=\alpha^{-1}\mathbf{P}_h\boldsymbol{\omega}(k)+\alpha^{-1}h\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\varepsilon}(k), \tag{B.6}$$

where $\mathbf{P}_h\triangleq\mathbf{I}_{mN}-h\mathbf{F_d}$ with $\mathbf{F_d}=\mathbf{L}\otimes\mathbf{I}_m+\mathbf{H_d}$. Note by $\gamma(k)\equiv1$, (A.1c), and (B.3) that

$$\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\eta}(k)=\mathbf{L}\otimes\mathbf{I}_m\left(\mathbf{D}\otimes\mathbf{I}_m\right)\mathbf{x}(k)=\mathbf{L}\otimes\mathbf{I}_m\mathbf{x}(k)=\mathbf{L}\otimes\mathbf{I}_m\mathbf{w}(k).$$

The from (A.2c), (A.2d), (B.4), and $\boldsymbol{\omega}(k)=\frac{\mathbf{w}(k)}{s(k)}$ it follows that

$$\boldsymbol{\varepsilon}(k+1)=\alpha^{-1}\big(\boldsymbol{\theta}(k)-Q_K\left(\boldsymbol{\theta}(k)\right)\big), \tag{B.7}$$

$$\boldsymbol{\theta}(k)\triangleq\left(\mathbf{I}_{mN}+h\mathbf{L}\otimes\mathbf{I}_m\right)\boldsymbol{\varepsilon}(k)-h\mathbf{F_d}\boldsymbol{\omega}(k). \tag{B.8}$$

Therefore, Algorithm 1 can be described by (B.6), (B.7), and (B.8).

The proof of nonsaturation of the uniform quantizer is equivalent to showing that for any $k\geq0$, $\boldsymbol{\theta}(k)$ defined by (B.8) satisfies $\|\boldsymbol{\theta}(k)\|_\infty<K+\frac{1}{2}$. The proof will use induction. So we begin by showing the quantizer is not saturated at $k=0$.

Note that $\mathbf{b}_i(0) = 0$ for each $i \in \mathcal{V}$. Then by (A.1a), we obtain $\mathbf{e}(0) = \mathbf{x}(0)$ and $\boldsymbol{\varepsilon}(0) = \mathbf{x}(0)/s(0)$. Then by A2, we have

$$\|\boldsymbol{\varepsilon}(0)\|_\infty = \frac{\|\mathbf{x}(0)\|_\infty}{s(0)} \le C_x/s(0). \tag{B.9}$$

By using the definitions $\boldsymbol{\omega}(0) = \mathbf{w}(0)/s(0)$, $\boldsymbol{\varepsilon}(0) = \mathbf{x}(0)/s(0)$, and (B.3), we obtain that $\mathbf{L} \otimes \mathbf{I}_m \boldsymbol{\omega}(0) = \mathbf{L} \otimes \mathbf{I}_m \boldsymbol{\varepsilon}(0)$. Thus, by using (B.4) and (B.8), one obtains

$$\begin{aligned}\boldsymbol{\theta}(0) &= (\mathbf{I}_{mN} + h\mathbf{L} \otimes \mathbf{I}_m)\, \boldsymbol{\varepsilon}(0) - h\,(\mathbf{L} \otimes \mathbf{I}_m + \mathbf{H_d})\, \boldsymbol{\omega}(0) \\ &= \boldsymbol{\varepsilon}(0) - h\mathbf{H_d}\mathbf{w}(0)/s(0) = \boldsymbol{\varepsilon}(0) - h(\mathbf{H_d}\mathbf{x}(0) - \mathbf{z}_H)/s(0).\end{aligned}$$

As a result, by A2, (B.9), and (11), we have the following:

$$\begin{aligned}\|\boldsymbol{\theta}(0)\|_\infty &\le \|\boldsymbol{\varepsilon}(0)\|_\infty + \frac{h}{s(0)}(\|\mathbf{H_d}\|_\infty \|\mathbf{x}(0)\|_\infty + \|\mathbf{z}_H\|_\infty) \\ &\le \frac{(1 + h\|\mathbf{H_d}\|_\infty)C_x + h\|\mathbf{z}_H\|_\infty}{s(0)} < K + \frac{1}{2}.\end{aligned}$$

Hence, the quantizer is unsaturated at $k = 0$. Now for the induction, we assume that the quantizer is not saturated at $k = 0, \ldots, p$. Then by (B.7) we have that

$$\sup_{1 \le k \le p+1} \|\boldsymbol{\varepsilon}(k)\|_\infty \le \frac{1}{2\alpha}. \tag{B.10}$$

We proceed to show that the quantizer is unsaturated for $k = p + 1$.

From (B.6) it follows that

$$\begin{aligned}\boldsymbol{\omega}(p+1) = &\left(\alpha^{-1}\mathbf{P}_h\right)^{p+1} \boldsymbol{\omega}(0) + \alpha^{-1}h \left(\alpha^{-1}\mathbf{P}_h\right)^{p} \mathbf{L} \otimes \mathbf{I}_m \boldsymbol{\varepsilon}(0) \\ &+ \alpha^{-1}h \sum_{i=0}^{p-1} \left(\alpha^{-1}\mathbf{P}_h\right)^{i} \mathbf{L} \otimes \mathbf{I}_m \boldsymbol{\varepsilon}(p-i).\end{aligned} \tag{B.11}$$

We now estimate the three terms on the right-hand side of (B.11) separately. Note that for any given $h > 0$, the eigenvalues of $\mathbf{P}_h = \mathbf{I}_{mN} - h\mathbf{F_d}$ are sorted in an ascending order as $1 - h\lambda_{\max}(\mathbf{F_d}) \le \cdots \le 1 - h\lambda_{\min}(\mathbf{F_d})$, and there exists a unitary matrix $\mathbf{U}$ such that $\mathbf{U}^T\mathbf{P}_h\mathbf{U} = \operatorname{diag}\{1 - h\lambda_{\max}(\mathbf{F_d}), \ldots, 1 - h\lambda_{\min}(\mathbf{F_d})\} \triangleq \boldsymbol{\Lambda}$. Therefore, $\|\boldsymbol{\Lambda}\|_2 = \max\{|1 - h\lambda_{\min}(\mathbf{F_d})|, |1 - h\lambda_{\max}(\mathbf{F_d})|\}$ and $(\mathbf{P}_h)^k = \left(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T\right)^k = \mathbf{U}\boldsymbol{\Lambda}^k\mathbf{U}^T$. Since $h \in \left(0, \frac{2}{\lambda_{\min}(\mathbf{F_d}) + \lambda_{\max}(\mathbf{F_d})}\right)$, by using [17, Lemma 3.1], we obtain that $\|\boldsymbol{\Lambda}\|_2 = 1 - h\lambda_{\min}(\mathbf{F_d}) = \rho_h$. Thus,

$$\|\mathbf{P}_h\|_2^k \le \|\mathbf{U}\|_2 \|\boldsymbol{\Lambda}\|_2^k \|\mathbf{U}^T\|_2 \le \rho_h^k. \tag{B.12}$$

By using $\boldsymbol{\omega}(0) = (\mathbf{x}(0) - \mathbf{1}_N \otimes \mathbf{I}_m \mathbf{y}^*)/s(0)$, $\mathbf{L}\mathbf{1}_N = \mathbf{0}_N$, $\mathbf{h}_i^T\mathbf{y}^* = z_i$, and the definition of $\mathbf{z}_h$ in (9), we obtain that

$$\begin{aligned}\mathbf{P}_h\boldsymbol{\omega}(0) &= \frac{\mathbf{P}_h\mathbf{x}(0)}{s(0)} - \frac{\mathbf{I}_{mN} - h\mathbf{L} \otimes \mathbf{I}_m - h\mathbf{H_d}}{s(0)} \mathbf{1}_N \otimes \mathbf{I}_m \mathbf{y}^* \\ &= \frac{\mathbf{P}_h\mathbf{x}(0)}{s(0)} - \frac{\mathbf{1}_N \otimes \mathbf{I}_m \mathbf{y}^*}{s(0)} + h\frac{\mathbf{z}_H}{s(0)}.\end{aligned} \tag{B.13}$$

Note by A1 that $\mathbf{H}^T\mathbf{H} = \sum_{i=1}^N \mathbf{h}_i\mathbf{h}_i^T$ is positive definite. Then the following holds:

$$\mathbf{y}^* = \left(\sum_{i=1}^N \mathbf{h}_i\mathbf{h}_i^T\right)^{-1} \sum_{i=1}^N \mathbf{h}_i\mathbf{h}_i^T\mathbf{y}^* = \left(\sum_{i=1}^N \mathbf{h}_i\mathbf{h}_i^T\right)^{-1} \sum_{i=1}^N z_i\mathbf{h}_i.$$

This combined with (B.13) produces

$$\mathbf{P}_h\boldsymbol{\omega}(0) = \frac{\mathbf{P}_h\mathbf{x}(0)}{s(0)} - \frac{\mathbf{1}_N \otimes \mathbf{I}_m}{s(0)} \left(\sum_{i=1}^N \mathbf{h}_i\mathbf{h}_i^T\right)^{-1} \sum_{i=1}^N z_i\mathbf{h}_i + \frac{h\mathbf{z}_H}{s(0)}.$$

Then by using (B.12), $\|\mathbf{1}_N \otimes \mathbf{I}_m\|_\infty = 1$, **A2**, and $\|\mathbf{x}\|_\infty \le \|\mathbf{x}\|_2 \le \sqrt{m}\|\mathbf{x}\|_\infty$ for any $\mathbf{x} \in \mathbb{R}^m$, we have

(B.14)

$$\begin{aligned}
&\| \left(\alpha^{-1}\mathbf{P}_h\right)^{p+1} \boldsymbol{\omega}(0)\|_2 \le \alpha^{-(p+1)} \|\mathbf{P}_h^p\|_2 \|\mathbf{P}_h\boldsymbol{\omega}(0)\|_2 \le \alpha^{-(p+1)}\rho_h^p\sqrt{mN}\|\mathbf{P}_h\boldsymbol{\omega}(0)\|_\infty \\
&\quad \le \frac{\alpha^{-(p+1)}\rho_h^p\sqrt{mN}}{s(0)} \left(\rho_h C_x + \left\|\left(\sum_{i=1}^N \mathbf{h}_i\mathbf{h}_i^T\right)^{-1}\right\|_2 \left\|\sum_{i=1}^N z_i\mathbf{h}_i\right\|_\infty + h\|\mathbf{z}_H\|_\infty\right) \\
&\quad \le \frac{\sqrt{mN}M_{hx}}{\alpha s(0)}\alpha^{-p}\rho_h^p \text{ with } M_{hx} \text{ defined in (10)},
\end{aligned}$$

where in the last inequality we used $\|\mathbf{A}^{-1}\|_2 = 1/\lambda_{\min}(\mathbf{A})$ for an invertible matrix $\mathbf{A}$, and $\|\sum_{i=1}^N \mathbf{x}_i\|_\infty \le \sum_{i=1}^N \|\mathbf{x}_i\|_\infty \le N\max_i \|\mathbf{x}_i\|_\infty = N\|col\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}\|_\infty$. For the second term of (B.11), from (B.12), (B.9), and $\|\mathbf{L}\|_2 = \lambda_N(\mathbf{L})$ it follows that

$$\begin{aligned}
&\|\alpha^{-1}h\left(\alpha^{-1}\mathbf{P}_h\right)^p \mathbf{L} \otimes \mathbf{I}_m\boldsymbol{\varepsilon}(0)\|_2 \\
&\qquad \le \alpha^{-(p+1)}h\|\mathbf{P}_h\|_2^p\|\mathbf{L}\|_2\|\boldsymbol{\varepsilon}(0)\|_2 \le \frac{hC_x\sqrt{mN}}{\alpha s(0)}\lambda_N(\mathbf{L})\left(\frac{\rho_h}{\alpha}\right)^p.
\end{aligned} \tag{B.15}$$

Similarly, for the last term of (B.11), by using

$$\left\|\sum_{i=0}^{p-1}\left(\alpha^{-1}\mathbf{P}_h\right)^i\right\|_2 \le \sum_{i=0}^{p-1}\left\|\left(\alpha^{-1}\mathbf{P}_h\right)^i\right\|_2 \le \sum_{i=0}^{p-1}\left(\frac{\rho_h}{\alpha}\right)^i = \frac{1-(\rho_h/\alpha)^p}{1-\rho_h/\alpha},$$

and by (B.10) we have that

$$\begin{aligned}
&\left\|\alpha^{-1}h\sum_{i=0}^{p-1}\left(\alpha^{-1}\mathbf{P}_h\right)^i \mathbf{L} \otimes \mathbf{I}_m\boldsymbol{\varepsilon}(p-i)\right\|_2 \\
&\quad \le \frac{h\sqrt{mN}}{\alpha}\|\mathbf{L}\|_2 \sup_{1\le k\le p+1}\|\boldsymbol{\varepsilon}(k)\|_\infty \left\|\sum_{i=0}^{p-1}\left(\alpha^{-1}\mathbf{P}_h\right)^i\right\|_2 \\
&\quad \le \frac{h\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)}\Big(1-\Big(\frac{\rho_h}{\alpha}\Big)^p\Big).
\end{aligned}$$

Thus, by using (B.11) and adding up the bounds of (B.14)–(B.1), we have that

$$\begin{aligned}
&\|\boldsymbol{\omega}(p+1)\|_\infty \le \|\boldsymbol{\omega}(p+1)\|_2 \\
&\quad \le \frac{h\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)} + \frac{\sqrt{mN}}{\alpha}\left(\frac{M_{hx}}{s(0)} + \frac{hC_x}{s(0)}\lambda_N(\mathbf{L}) - \frac{h\lambda_N(\mathbf{L})}{2(\alpha-\rho_h)}\right)\left(\frac{\rho_h}{\alpha}\right)^p \\
&\quad \le \frac{\sqrt{mN}}{\alpha}\max\left\{\frac{M_{hx}+hC_x\lambda_N(\mathbf{L})}{s(0)}, \frac{h\lambda_N(\mathbf{L})}{2(\alpha-\rho_h)}\right\} \le \frac{h\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)},
\end{aligned} \tag{B.16}$$

where the last inequality follows by (11). This together with $\|\mathbf{x}\|_\infty \le \|\mathbf{x}\|_2$, (10), (B.8), and (B.10) leads to

$$\begin{aligned}\|\boldsymbol{\theta}(p+1)\|_\infty &\le \|\mathbf{I}_{mN} + h\mathbf{L}\otimes\mathbf{I}_m\|_\infty \|\boldsymbol{\varepsilon}(p+1)\|_\infty + h\|\mathbf{F_d}\|_2\|\boldsymbol{\omega}(p+1)\|_2 \\ &\le \frac{1+2hd^*}{2\alpha} + \frac{h^2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\alpha(\alpha-\rho_h)} = M(\alpha,h) \le \mathcal{K}(\alpha,h) + \frac{1}{2} \le K + \frac{1}{2}.\end{aligned}$$

As a result, when $k = p+1$, the quantizer is also unsaturated. Therefore, by induction, we conclude that if a $(2K+1)$-levels uniform quantizer is applied, then the quantizer will never be saturated.

B.2. Proof of Theorem 3.4. Since the conditions required by Proposition 3.1 are the same as those used in Theorem 3.4, the quantizer will never be saturated by Proposition 3.1. Then by (B.7) we conclude that $\sup_{k\ge1}\|\boldsymbol{\varepsilon}(k)\|_\infty \le 1/2\alpha$, and hence (B.16) holds for any $p \ge 0$. Thus, we obtain that $\|\boldsymbol{\omega}(k)\|_2 \le \frac{h\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)} < \infty$ for any $k \ge 1$. This combined with the definitions $\mathbf{w}(k) = \mathbf{x}(k) - \mathbf{1}_N \otimes \mathbf{y}^*$ and $\mathbf{w}(k) = s(0)\alpha^k\boldsymbol{\omega}(k)$ proves (17). Hence (16) holds by letting $k \to \infty$.

B.3. Proof of Theorem 3.6. By using $\rho_h = 1 - h\lambda_{\min}(\mathbf{F_d})$ and $M(\alpha,h)$ defined in (10), there holds

$$M(\alpha,h) = \frac{1+2hd^*}{2\alpha} + \frac{h^2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\alpha\big(\alpha-(1-h\lambda_{\min}(\mathbf{F_d}))\big)}. \tag{B.17}$$

Note that $\lim_{h\to0}\frac{1+2hd^*}{2} + \frac{h\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\lambda_{\min}(\mathbf{F_d})} = \frac{1}{2}$. Then for any given $K \ge 1$ there exists $h^* \in (0, \frac{2}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})})$ such that $\frac{1+2h^*d^*}{2} + \frac{h^*\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\lambda_{\min}(\mathbf{F_d})} \le K$. By (B.17) it follows that $\lim_{\alpha\to1} M(\alpha,h^*) = \frac{1+2h^*d^*}{2} + \frac{h^*\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\lambda_{\min}(\mathbf{F_d})} \le K$, and hence there exists $\alpha^* \in (1-h^*\lambda_{\min}(\mathbf{F_d}),1)$ such that $M(\alpha^*,h^*) < K + \frac{1}{2}$. Thus, $(\alpha^*,h^*) \in \Xi_K$, and hence Ξ_K is nonempty.

For any $(\alpha,h) \in \Xi_K$, from (18) it follows that $h \in \big(0, \frac{2}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})}\big), \alpha \in (1-h\lambda_{\min}(\mathbf{F_d}),1)$, and $M(\alpha,h) < K + \frac{1}{2}$. Then $\mathcal{K}(\alpha,h) \triangleq \lceil M(\alpha,h-\frac{1}{2}\rceil \le K$ together with Theorem 3.4 completes the proof.

B.4. Proof of Proposition 3.8 . It is noticed by $\rho_{h_\nu} = 1 - h_\nu\lambda_{\min}(\mathbf{F_d})$ that $\alpha_\nu - \rho_{h_\nu} = \nu h_\nu\lambda_{\min}(\mathbf{F_d}) > 0$. Then by the definition $M(\alpha,h)$ in (10), we have

$$\begin{aligned}M(\alpha_\nu,h_\nu) &= \frac{1+2h_\nu d^*}{2\left(1-(1-\nu)h_\nu\lambda_{\min}(\mathbf{F_d})\right)} + \frac{h_\nu\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\nu\lambda_{\min}(\mathbf{F_d})\left(1-(1-\nu)h_\nu\lambda_{\min}(\mathbf{F_d})\right)} \\ &= \frac{\nu\lambda_{\min}(\mathbf{F_d})(1+2h_\nu d^*) + h_\nu\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\nu\lambda_{\min}(\mathbf{F_d})\left(1-(1-\nu)h_\nu\lambda_{\min}(\mathbf{F_d})\right)}.\end{aligned}$$

Then for any $h_\nu < \hat{h}_{K,\nu}$, by the definition of $\hat{h}_{K,\nu}$ in (19), there holds $M(\alpha_\nu,h_\nu) < K + \frac{1}{2}$. Note by $h_\nu \in \big(0, \frac{2}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})}\big)$ and the definition of Ξ_K in (18), we conclude that $(\alpha_\nu,h_\nu) \in \Xi_K$.

For any $(\alpha,h) \in \Xi_K$, there hold $\alpha \in (\rho_h,1)$ and $h \in \big(0, \frac{2}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})}\big)$ by (18). Define $\nu \triangleq 1 - \frac{1-\alpha}{h\lambda_{\min}(\mathbf{F_d})}$. Then by $\rho_h = 1 - h\lambda_{\min}(\mathbf{F_d})$ it is seen that $\nu \in (0,1)$ and $\alpha = 1-(1-\nu)h\lambda_{\min}(\mathbf{F_d})$. It is noticed that $M(\alpha,h) < K + \frac{1}{2}$ is equivalent to $h < \hat{h}_{K,\nu_0}$ with $\hat{h}_{K,\nu}$ defined in (19), and hence $h \in (0, h^*_{K,\nu})$. This completes the proof of Proposition 3.8.

B.5. Proof of Theorem 3.10. For any given $K \geq 1$, define

$$\Gamma_K \triangleq \big\{\alpha : \alpha = 1-(1-\nu)h\lambda_{\min}(\mathbf{F_d}), \nu \in (0,1), h \in (0, h^*_{K,\nu})\big\}. \tag{B.18}$$

By $h^*_{K,\nu} \leq \hat{h}_{K,\nu}$ and (19), we know that for any $h \in (0, h^*_{K,\nu})$

$$h \leq 2K\nu\lambda_{\min}(\mathbf{F_d})\big(\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})\big)^{-1}.$$

Because $(1-\nu)\nu \leq \frac{1}{4}$ for any $\nu \in (0,1)$, then for any $\alpha = 1-(1-\nu)h\lambda_{\min}(\mathbf{F_d}), \nu \in (0,1), h \in (0, h^*_{K,\nu})$, the following holds:

$$\alpha > 1 - \frac{2K(1-\nu)\nu\lambda^2_{\min}(\mathbf{F_d})}{\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})} \geq 1 - \frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}.$$

Thus, it holds for fixed K that

$$\frac{\inf_{\alpha\in\Gamma_K}\alpha}{\exp\left(-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}\right)} \geq \frac{1-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}}{\exp\left(-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}\right)},$$

which together with $\lim_{x\downarrow 0}\frac{1-x}{\exp(-x)} = 1$ produces

$$\liminf_{N\to\infty} \frac{\inf_{\alpha\in\Gamma_K}\alpha}{\exp\left(-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}\right)} \geq 1. \tag{B.19}$$

From (19) it follows that

$$\hat{h}_{K,\nu} \geq 2K\nu\lambda_{\min}(\mathbf{F_d})\big(\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d}) + 2\lambda_{\min}(\mathbf{F_d})d^* + (2K+1)\lambda^2_{\min}(\mathbf{F_d})\big)^{-1},$$

which combined with $\inf_{h\in(0,h^*_{K,\nu})}\ \alpha \leq 1-(1-\nu)\hat{h}_{K,\nu}\lambda_{\min}(\mathbf{F_d})$ implies that

$$\begin{aligned}\inf_{\alpha\in\Gamma_K}\alpha &\leq 1 - \frac{\max_{\nu\in(0,1)}\ 2\nu(1-\nu)K\lambda^2_{\min}(\mathbf{F_d})}{2\lambda_{\min}(\mathbf{F_d})d^* + \sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d}) + (2K+1)\lambda^2_{\min}(\mathbf{F_d})} \\ &= 1 - \frac{K\lambda^2_{\min}(\mathbf{F_d})/2}{2\lambda_{\min}(\mathbf{F_d})d^* + \sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d}) + (2K+1)\lambda^2_{\min}(\mathbf{F_d})}.\end{aligned}$$

Then we have that for any given $K \geq 1$,

$$\begin{aligned}\frac{\inf_{\alpha\in\Gamma_K}\alpha}{\exp\left(-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}\right)} &\leq \frac{1-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}}{\exp\left(-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}\right)} \\ &\times \frac{1-\frac{K\lambda^2_{\min}(\mathbf{F_d})/2}{\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})+2\lambda_{\min}(\mathbf{F_d})d^*+(2K+1)\lambda^2_{\min}(\mathbf{F_d})}}{1-\frac{K\lambda^2_{\min}(\mathbf{F_d})/2}{\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}}.\end{aligned}$$

This together with $\lim_{N\to\infty}\frac{1-c_1/(\sqrt{N}+c_2)}{1-c_1/\sqrt{N}} = 1$ and $\lim_{x\downarrow 0}\frac{1-x}{\exp(-x)} = 0$ gives

$$\limsup_{N\to\infty} \frac{\inf_{\alpha\in\Gamma_K}\alpha}{\exp\left(-\frac{K\lambda^2_{\min}(\mathbf{F_d})}{2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}\right)} \leq 1. \tag{B.20}$$

Using Theorem 3.6(i) and (B.18), we have the following:

$$\inf_{(\alpha,h)\in\Xi_K}\alpha = \inf_{(\alpha,h)\in\cup_{\nu\in(0,1)}\Xi_{K,\nu}}\alpha = \inf_{\nu\in(0,1)}\ \inf_{(\alpha,h)\in\Xi_{K,\nu}}\alpha = \inf_{\alpha\in\Gamma_K}\alpha.$$

This combined with (B.19) and (B.20) produces (21).

Appendix C. Proof of results stated in section 4.

C.1. Proof of Proposition 4.2. Based on Lemma A.1, by using $s(k) = s_r\gamma(k)$ and $\beta(k) = \frac{\gamma(k)}{\gamma(k+1)}$, we have that

$$\mathbf{x}(k+1) = \mathbf{P}(k)\mathbf{x}(k) + h\gamma(k)\big(s_r\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\varepsilon}(k) + \mathbf{z}_H\big), \tag{C.1}$$

$$\boldsymbol{\eta}(k+1) = \beta(k)\big(\left(\mathbf{I}_{mN} - h\mathbf{L}\otimes\mathbf{I}_m\right)\boldsymbol{\eta}(k)$$

$$+ hs_r\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\varepsilon}(k) + h\mathbf{D}\otimes\mathbf{I}_m\left(\mathbf{z}_H - \mathbf{H_d}\mathbf{x}(k)\right)\big), \tag{C.2}$$

$$\boldsymbol{\varepsilon}(k+1) = \beta(k)\big(\boldsymbol{\theta}(k) - Q_K\left(\boldsymbol{\theta}(k)\right)\big), \tag{C.3}$$

$$\boldsymbol{\theta}(k) \triangleq \big(\mathbf{I}_{mN} + h\mathbf{L}\otimes\mathbf{I}_m\big)\boldsymbol{\varepsilon}(k) - hs_r^{-1}\big(\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\eta}(k) + \mathbf{H_d}\mathbf{x}(k) - \mathbf{z}_H\big), \tag{C.4}$$

where $\boldsymbol{\varepsilon}(k)$, $\mathbf{P}(k)$, and $\boldsymbol{\eta}(k)$ are defined in (A.1a), (A.1b), and (A.1c), respectively

The proof of nonsaturation of the uniform quantizer is equivalent to showing that for any $k \geq 0$, $\boldsymbol{\theta}(k)$ defined by (C.4) satisfies $\|\boldsymbol{\theta}(k)\|_\infty < K + \frac{1}{2}$. Again, we use an induction proof. So we begin by showing the quantizer is not saturated at $k = 0$.

Recalling that $\gamma(0) = 1, \mathbf{b}_i(0) = 0$ for each $i \in \mathcal{V}$, we obtain $\mathbf{e}(0) = \mathbf{x}(0)$ and $\boldsymbol{\varepsilon}(0) = \mathbf{x}(0)/s_r$. Then by using $\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\eta}(0) = \mathbf{L}\otimes\mathbf{I}_m\mathbf{x}(0)$ and (C.4), we obtain that $\boldsymbol{\theta}(0) = \mathbf{x}(0)/s_r - hs_r^{-1}(\mathbf{H_d}\mathbf{x}(0) - \mathbf{z}_H)\big)$. Then from A2, and (28) it follows that

$$\|\boldsymbol{\theta}(0)\|_\infty \leq \frac{C_x + h(C_x\|\mathbf{H_d}\|_\infty + \|\mathbf{z}_H\|_\infty)}{s_r} < K + \frac{1}{2}.$$

Hence, the quantizer is unsaturated at $k = 0$. Next, for the induction, we assume that the quantizer is unsaturated for $k = 0, \ldots, p$. Then by (C.3) and A5, there holds

$$\|\boldsymbol{\varepsilon}(k)\|_\infty \leq \frac{\beta(k)}{2} \leq \frac{\beta(0)}{2} \quad \forall k : 1 \leq k \leq p+1. \tag{C.5}$$

We aim to show that the quantizer is unsaturated for $k = p+1$. Define

$$\boldsymbol{\Gamma}(k, k+1) \triangleq \mathbf{I}_{mN} \text{ and } \boldsymbol{\Gamma}(k_1, k_2) \triangleq \mathbf{P}(k_1)\mathbf{P}(k_1-1)\ldots\mathbf{P}(k_2)\ \forall k_1 \geq k_2 \geq 0. \tag{C.6}$$

Then from (C.1) and $\gamma(0) = 1$ it follows that

$$\begin{aligned}\mathbf{x}(p+1) &= \boldsymbol{\Gamma}(p,0)\mathbf{x}(0) + hs_r\boldsymbol{\Gamma}(p,1)\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\varepsilon}(0)\\ &\quad + hs_r\sum_{i=1}^{p}\gamma(i)\boldsymbol{\Gamma}(p,i+1)\mathbf{L}\otimes\mathbf{I}_m\boldsymbol{\varepsilon}(i) + h\sum_{i=0}^{p}\gamma(i)\boldsymbol{\Gamma}(p,i+1)\mathbf{z}_H.\end{aligned} \tag{C.7}$$

We now estimate the bound of $\mathbf{x}(p+1)$. By $\mathbf{F_d} = \mathbf{L}\otimes\mathbf{I}_m + \mathbf{H_d}$, one obtains

$$\min\{1,\gamma(k)\}x^T\mathbf{F_d}x \leq x^T\left(\mathbf{L}\otimes\mathbf{I}_m + \gamma(k)\mathbf{H_d}\right)x \leq \max\{1,\gamma(k)\}x^T\mathbf{F_d}x \quad \forall x \in \mathbb{R}^{mN}.$$

Notice that $\min\{1,\gamma(k)\} = \gamma(k)$ and $\max\{1,\gamma(k)\} = 1$ by $0 < \gamma(k) \leq 1$. Thus, for the matrix $\mathbf{L}\otimes\mathbf{I}_m + \gamma(k)\mathbf{H_d}$, the smallest eigenvalue is greater than or equal to $\gamma(k)\lambda_{\min}(\mathbf{F_d})$ while the largest eigenvalue is smaller than or equal to $\lambda_{\max}(\mathbf{F_d})$. Then by $\mathbf{P}(k)$ defined in (A.1b), the eigenvalues of $\mathbf{P}(k)$ can be sorted in an ascending order as $1 - h\lambda_{\max}(\mathbf{F_d}) \leq \lambda_1(\mathbf{P}(k)) \leq \cdots \leq \lambda_{mN}(\mathbf{P}(k)) \leq 1 - h\gamma(k)\lambda_{\min}(\mathbf{F_d})$. Thus, for any $k \geq 0$, $\|\mathbf{P}(k)\|_2 \leq \max\{\left|1 - h\gamma(k)\lambda_{\min}(\mathbf{F_d})\right|, \left|1 - h\lambda_{\max}(\mathbf{F_d})\right|\}$. Then by recalling that $0 < h < \frac{2}{\lambda_{\min}(\mathbf{F_d}) + \lambda_{\max}(\mathbf{F_d})}$ and $\gamma(k) \leq 1$, the following holds:

$$\|\mathbf{P}(k)\|_2 \leq 1 - h\gamma(k)\lambda_{\min}(\mathbf{F_d}) \leq \exp\left(-h\gamma(k)\lambda_{\min}(\mathbf{F_d})\right),$$

where the last inequality holds since for any $x \geq 0$, $1-x \leq \exp(-x)$. Then from (C.6) it follows that for any $k_1 \geq k_2 \geq 0$,

$$\|\mathbf{\Gamma}(k_1,k_2)\|_2 < \exp\left(-h\lambda_{\min}(\mathbf{F_d})\sum_{k=k_2}^{k_1}\gamma(k)\right).$$

Also, using (C.5), (C.7), A2, $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{m}\|\mathbf{x}\|_\infty$ for any $\mathbf{x} \in \mathbb{R}^m$, and $\mathbf{x}(0) = s_r\boldsymbol{\varepsilon}(0)$, we obtain that

$$\begin{aligned}
\|\mathbf{x}(p+1)\|_2 &\leq hs_r\|\mathbf{L}\|_2\sum_{i=1}^{p}\gamma(i)\|\mathbf{\Gamma}(p,i+1)\|_2\|\boldsymbol{\varepsilon}(i)\|_2 + \|\mathbf{\Gamma}(p,0)\|_2\|\mathbf{x}(0)\|_2 \\
&\quad + h\|\mathbf{\Gamma}(p,1)\|_2\|\mathbf{L}\|_2\|\mathbf{x}(0)\|_2 + h\|\mathbf{z}_H\|_2\sum_{i=1}^{p}\gamma(i)\|\mathbf{\Gamma}(p,i+1)\|_2 \\
&\leq \sqrt{mN}(1+h\lambda_N(\mathbf{L}))C_x + h\|\mathbf{z}_H\|_2\sum_{i=0}^{p}\gamma(i)\exp\left(-h\lambda_{\min}(\mathbf{F_d})\sum_{k=i+1}^{p}\gamma(k)\right) \\
&\quad + \frac{hs_r\beta(0)\sqrt{mN}\lambda_N(\mathbf{L})}{2}\sum_{i=1}^{p}\gamma(i)\exp\left(-h\lambda_{\min}(\mathbf{F_d})\sum_{k=i+1}^{p}\gamma(k)\right).
\end{aligned} \tag{C.8}$$

Since $h\gamma(k) \leq 1/\lambda_{\min}(\mathbf{F_d})$ for any $k \geq 0$, the following holds for any $k \geq 0$:

$$h\gamma(k) \leq 2\left(h\gamma(k) - \frac{\lambda_{\min}(\mathbf{F_d})}{2}(h\gamma(k))^2\right).$$

Now observe that for any $x \in (0,1)$, $x - x^2/2 < 1 - \exp(-x)$. Then we have the following sequence of inequalities:

$$\begin{aligned}
&\sum_{i=k_1}^{p} h\lambda_{\min}(\mathbf{F_d})\gamma(i)\exp\left(-h\lambda_{\min}(\mathbf{F_d})\sum_{k=i+1}^{p}\gamma(k)\right) \\
&\quad\leq 2\sum_{i=k_1}^{p}\left(h\lambda_{\min}(\mathbf{F_d})\gamma(i) - \left(h\lambda_{\min}(\mathbf{F_d})\gamma(i)\right)^2/2\right)\exp\left(-h\lambda_{\min}(\mathbf{F_d})\sum_{k=i+1}^{p}\gamma(i)\right) \\
&\quad\leq 2\sum_{i=k_1}^{p}\left(1-\exp\left(-h\lambda_{\min}(\mathbf{F_d})\gamma(i)\right)\right)\exp\left(-h\lambda_{\min}(\mathbf{F_d})\sum_{k=i+1}^{p}\gamma(k)\right) \\
&\quad= 2\sum_{i=k_1}^{p}\left[\exp\Big(-h\lambda_{\min}(\mathbf{F_d})\sum_{k=i+1}^{p}\gamma(k)\Big) - \exp\left(-h\lambda_{\min}(\mathbf{F_d})\sum_{k=i}^{p}\gamma(k)\right)\right] \leq 2.
\end{aligned}$$

Using this in (C.8) yields

$$\|\mathbf{x}(p+1)\|_2 \leq \sqrt{mN}C_x(1+h\lambda_N(\mathbf{L})) + \frac{s_r\beta(0)\sqrt{mN}\lambda_N(\mathbf{L})}{\lambda_{\min}(\mathbf{F_d})} + \frac{2\|\mathbf{z}_H\|_2}{\lambda_{\min}(\mathbf{F_d})} \triangleq M_x. \tag{C.9}$$

Since $\mathbf{L}$ is symmetric, we can define an orthogonal matrix $\mathbf{T} \triangleq (\frac{\mathbf{1}_N}{\sqrt{N}}, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N)$, where $\mathbf{L}\boldsymbol{\phi}_i = \lambda_i(\mathbf{L})\boldsymbol{\phi}_i$, $i = 2, \ldots, N$. Let $\tilde{\boldsymbol{\eta}}(k) = \left(\mathbf{T}^{-1}\otimes\mathbf{I}_m\right)\boldsymbol{\eta}(k) = \left(\mathbf{T}^T\otimes\mathbf{I}_m\right)\boldsymbol{\eta}(k)$ and decompose it as $\tilde{\boldsymbol{\eta}}(k) = (\tilde{\boldsymbol{\eta}}_1(k)^T, \tilde{\boldsymbol{\eta}}_2(k)^T)^T$ with $\tilde{\boldsymbol{\eta}}_1(k) = \frac{1}{\sqrt{N}}\left(\mathbf{1}_N^T\otimes\mathbf{I}_m\right)\boldsymbol{\eta}(k)$

and $\tilde{\boldsymbol{\eta}}_2(k) = \left(\mathbf{T}_2^T \otimes \mathbf{I}_m\right)\boldsymbol{\eta}(k)$, where $\mathbf{T}_2 = (\boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N)$. Then $\tilde{\boldsymbol{\eta}}_1(k) = \mathbf{0}_m$ by $\boldsymbol{\eta}(k) = (\mathbf{D} \otimes \mathbf{I}_m)\frac{\mathbf{x}(k)}{\gamma(k)}$ and $\mathbf{1}_N^T\mathbf{D} = \mathbf{0}_m^T$. Then by multiplying both sides of (C.2) with $\mathbf{T}_2^T \otimes \mathbf{I}_m$ from the left, and noting $\mathbf{T}_2^T\mathbf{L} = \operatorname{diag}\{\lambda_2(\mathbf{L}), \ldots, \lambda_N(\mathbf{L})\}\mathbf{T}_2^T$, we have that

$$\begin{aligned}\tilde{\boldsymbol{\eta}}_2(k+1) = {} & \beta(k)hs_r\left(\mathbf{T}_2^T\mathbf{L} \otimes \mathbf{I}_m\right)\boldsymbol{\varepsilon}(k) \\ & + \beta(k)\Big(\underbrace{\operatorname{diag}\{1 - h\lambda_2(\mathbf{L}), \ldots, 1 - h\lambda_N(\mathbf{L})\}}_{\mathbf{D}_h} \otimes \mathbf{I}_m\Big)\tilde{\boldsymbol{\eta}}_2(k) \\ & + \beta(k)h\left(\mathbf{T}_2^T\mathbf{D} \otimes \mathbf{I}_m\right)(\mathbf{z}_H - \mathbf{H}_{\mathbf{d}}\mathbf{x}(k)). \end{aligned} \tag{C.10}$$

Thus, there holds

$$\begin{aligned}\tilde{\boldsymbol{\eta}}_2(p+1) = {} & \left(\mathbf{D}_h^{p+1} \otimes \mathbf{I}_m\right)\tilde{\boldsymbol{\eta}}_2(0)\prod_{k=0}^{p}\beta(k) \\ & + h\sum_{k=0}^{p}\left(\mathbf{D}_h^k\mathbf{T}_2^T\mathbf{D} \otimes \mathbf{I}_m\right)\prod_{i=p-k}^{p}\beta(i)(\mathbf{z}_H - \mathbf{H}_{\mathbf{d}}\mathbf{x}(p-k)) \\ & + hs_r\sum_{k=0}^{p}\left(\mathbf{D}_h^k\mathbf{T}_2^T\mathbf{L} \otimes \mathbf{I}_m\right)\prod_{i=p-k}^{p}\beta(i)\boldsymbol{\varepsilon}(p-k). \end{aligned} \tag{C.11}$$

Note that $(\mathbf{T}_2 \otimes \mathbf{I}_m)\tilde{\boldsymbol{\eta}}_2(k) = \left(\mathbf{T}_2\mathbf{T}_2^T \otimes \mathbf{I}_m\right)\boldsymbol{\eta}(k) = (\mathbf{I}_N - \frac{\mathbf{1}_N\mathbf{1}_N^T}{N} \otimes \mathbf{I}_m)\boldsymbol{\eta}(k) = \boldsymbol{\eta}(k)$. Then by multiplying both sides of (C.11) on the left with $(\mathbf{T}_2 \otimes \mathbf{I}_m)$, there holds

$$\begin{aligned}\boldsymbol{\eta}(p+1) = {} & \left(\mathbf{T}_2\mathbf{D}_h^{p+1}\mathbf{T}_2^T \otimes \mathbf{I}_m\right)\boldsymbol{\eta}(0)\prod_{k=0}^{p}\beta(k) \\ & + h\sum_{k=0}^{p}\left(\mathbf{T}_2\mathbf{D}_h^k\mathbf{T}_2^T\mathbf{D} \otimes \mathbf{I}_m\right)(\mathbf{z}_H - \mathbf{H}_{\mathbf{d}}\mathbf{x}(p-k))\prod_{i=p-k}^{p}\beta(i) \\ & + hs_r\sum_{k=0}^{p}\left(\mathbf{T}_2\mathbf{D}_h^k\mathbf{T}_2^T\mathbf{L} \otimes \mathbf{I}_m\right)\boldsymbol{\varepsilon}(p-k)\prod_{i=p-k}^{p}\beta(i). \end{aligned} \tag{C.12}$$

By the definition of $\mathbf{D}_h$ in (C.10), $\|\mathbf{D}_h\|_2 = \max\{|1 - h\lambda_2(\mathbf{L})|, |1 - h\lambda_N(\mathbf{L})|\}$. Thus, by using $h \in (0, \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})})$ and [17, Lemma 3.1], we obtain that $\|\mathbf{D}_h\|_2 = 1 - h\lambda_2(\mathbf{L}) = \hat{\rho}_h$. Taking two-norms of (C.12), by recalling that $\beta(k) \le \beta(0)\ \forall k \ge 0$, $\|\mathbf{D}\|_2 = \|\mathbf{T_2}\|_2 = 1$, we have the following:

$$\begin{aligned}\|\boldsymbol{\eta}(p+1)\|_2 \le {} & (\beta(0)\hat{\rho}_h)^{p+1}\|\boldsymbol{\eta}(0)\|_2 + hs_r\beta(0)\|\mathbf{L}\|_2(\beta(0)\hat{\rho}_h)^p\|\boldsymbol{\varepsilon}(0)\|_2 \\ & + hs_r\beta(0)\|\mathbf{L}\|_2\sum_{k=0}^{p-1}(\beta(0)\hat{\rho}_h)^k\|\boldsymbol{\varepsilon}(p-k)\|_2 \\ & + h\beta(0)\sum_{k=0}^{p}(\beta(0)\hat{\rho}_h)^k\|\mathbf{z}_H - \mathbf{H}_{\mathbf{d}}\mathbf{x}(p-k)\|_2. \end{aligned}$$

Note by $\gamma(0) = 1, s(0) = s_r$, $\|\mathbf{D}\|_2 = 1$, and A2 that

$$\|\boldsymbol{\eta}(0)\|_2 = \|\mathbf{D} \otimes \mathbf{I}_m\mathbf{x}(0)\|_2 \le \|\mathbf{D}\|_2\|\mathbf{x}(0)\|_2 \le \sqrt{mN}C_x,$$
$$\|\boldsymbol{\varepsilon}(0)\|_2 \le \|\mathbf{x}(0)\|_2/s(0) \le \sqrt{mN}C_x/s_r.$$

Similar to (C.9) we can easily show that $\|\mathbf{x}(k)\|_2 \le M_x \ \forall k = 0, \dots, p$. Then by using (C.5), $\beta(0)\hat{\rho}_h < 1$, and $\sum_{k=0}^{p} (\beta(0)\hat{\rho}_h)^k \le \frac{1}{1-\beta(0)\hat{\rho}_h}$, we obtain the following:

$$\|\boldsymbol{\eta}(p+1)\|_2 \le \sqrt{mN} C_x \left(1 + h\beta(0)\lambda_N(\mathbf{L})\right) + \frac{\sqrt{mN} h s_r \beta(0)^2 \lambda_N(\mathbf{L})}{2(1-\beta(0)\hat{\rho}_h)}$$
$$+ \left(\|\mathbf{z}_H\|_2 + \|\mathbf{H_d}\|_2 M_x\right) \frac{h\beta(0)}{1-\beta(0)\hat{\rho}_h}. \tag{C.13}$$

This together with (26)–(28), (C.4), (C.5), and (C.9) leads to

$$\begin{aligned}
\|\boldsymbol{\theta}(p+1)\|_\infty &\le \| \left(\mathbf{I}_{mN} + h\mathbf{L} \otimes \mathbf{I}_m\right) \boldsymbol{\varepsilon}(p+1)\|_\infty \\
&\quad + h s_r^{-1} \big(\lambda_N(\mathbf{L}) \|\boldsymbol{\eta}(p+1)\|_2 + \|\mathbf{z}_H\|_\infty + \|\mathbf{H_d}\|_\infty \|\mathbf{x}(p+1)\|_\infty\big) \\
&\le \frac{\beta(0)(1+2hd^*)}{2} + h\left(s_r^{-1} M_1(h, \beta(0)) + M_2(h, \beta(0))\right) \\
&\le \beta(0)(1/2 + hd^*) + 2hM_2(h, \beta(0)) = M'(h, \beta(0)) \\
&\le \left\lceil M'(h, \beta(0)) - \frac{1}{2} \right\rceil + \frac{1}{2} = \mathcal{K}'(h, \beta(0)) + \frac{1}{2} \le K + \frac{1}{2}.
\end{aligned}$$

As a result, when $k = p+1$, the quantizer is also unsaturated. Therefore, by induction, we conclude that if a $(2K+1)$-levels uniform quantizer is applied, then the quantizer will never be saturated.

C.2. Proof of Theorem 4.4. From Proposition 4.2 it follows that (C.13) holds for any $p \ge 0$. This implies that $\sup_{k\ge 1} \|\boldsymbol{\eta}(k)\|_\infty < \infty$. Then using the definition of $\boldsymbol{\eta}(k)$ in (A.1c), we obtain that $\|\mathbf{x}_i(k) - \mathbf{y}(k)\|_\infty = \mathcal{O}(\gamma(k))$, where $\mathbf{y}_k = \sum_{i=1}^{N} \mathbf{x}_{i,k}/N$. Then by multiplying both sides of (A.4) from the left by $\frac{1}{N}\left(\mathbf{1}_N^T \otimes \mathbf{I}_m\right)$ and using $\mathbf{1}_N^T \mathbf{L} = \mathbf{0}_N^T$, we obtain that

$$\mathbf{y}_{k+1} = \mathbf{y}_k - h\gamma(k)(\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{y}_k - \mathbf{H}^T\mathbf{z})/N - \frac{h\gamma(k)}{N} \sum_{i=1}^{N} \mathbf{h}_i \mathbf{h}_i^\top \left(\mathbf{x}_{i,k} - \mathbf{y}_k\right).$$

Then by recalling that $\mathbf{y}^\star = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z}$, we obtain that

$$\mathbf{y}_{k+1} - \mathbf{y}_{\mathrm{LS}}{}^* = \mathbf{y}_k - \mathbf{y}_{\mathrm{LS}}{}^* - \frac{h\gamma(k)}{N}\mathbf{H}^{\mathrm{T}}\mathbf{H}\left(\mathbf{y}_k - \mathbf{y}_{\mathrm{LS}}{}^*\right) - \frac{h\gamma(k)}{N} \sum_{i=1}^{N} \mathbf{h}_i \mathbf{h}_i^\top \left(\mathbf{x}_{i,k} - \mathbf{y}_k\right).$$

Because $\gamma(k)$ monotone decreases to zero by A6, by $\|\mathbf{x}_i(k) - \mathbf{y}(k)\|_\infty = \mathcal{O}(\gamma(k))$ we conclude that $\lim_{k\to\infty}(\mathbf{x}_{i,k} - \mathbf{y}_k) = \mathbf{0}$. Since $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ is positive definite by $\mathrm{rank}(\mathbf{H}) = m$, from [3, Lemma 3.1.1] it follows that $\lim_{k\to\infty} \mathbf{y}_k = \mathbf{y}_{\mathrm{LS}}{}^*$, which together with $\mathbf{x}_i(k) - \mathbf{y}(k) \to \mathbf{0}$ implies (29). Then by $\|\mathbf{x}_i(k) - \mathbf{y}(k)\|_\infty = \mathcal{O}(\gamma(k))$ and applying [3, Theorem 3.1.1], we obtain that for any $k \ge 1$, $\|\mathbf{y}(k) - \mathbf{y}_{\mathrm{LS}}{}^*\|_\infty = \mathcal{O}(\gamma(k))$. Thus, (30) holds.

C.3. Proof of Theorem 4.6. The proof is similar to that of Theorem 3.6. So, we just show the key steps and skip the detailed calculations. Using $\hat{\rho}_h = 1 - h\lambda_2(\mathbf{L})$, $\kappa_N = \frac{\lambda_N(\mathbf{L})}{\lambda_2(\mathbf{L})}$, (26), and (27), we obtain that $\lim_{h\to 0} M'(h, 1) = \frac{1}{2}$. Then there exists $h^* \in \left(0, \min\left\{\frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{P_d})}\right\}\right)$ such that $M'(h^*, 1) \le K$ for any $K \ge 1$. By recalling the definition of $M'(h, \beta(0))$ in (26), $\lim_{\beta(0)\to 1} M'(h^*, \beta(0)) = M'(h^*, 1) \le K$. Then there exists $\beta^*(0) \in (1, \frac{1}{1-h\lambda_2(\mathbf{L})})$ such that $M'(h^*, \beta^*(0)) \le K + \frac{1}{2}$. Therefore, $(h^*, \beta^*(0)) \in \Xi'_K$. Hence Ξ'_K is nonempty.

For any $(h, \beta(0)) \in \Xi'_K$, by (31) we have $h \in \big(0, \min\big\{\frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{P_d})}\big\}\big)$, $\beta(0) \in \big(1, \frac{1}{1-h\lambda_2(\mathbf{L}}\big)$, and $M'(h, \beta(0)) \leq K + \frac{1}{2}$. Then by definition (26), $\mathcal{K}'(h, \beta(0)) = \big\lceil M'(h, \beta(0)) - \frac{1}{2}\big\rceil \leq K$, which together with Theorem 4.4 completes the proof.

REFERENCES

[1] B. D. O. Anderson, S. Mou, A. S. Morse, and U. Helmke, *Decentralized gradient algorithm for solution of a linear equation*, Numer. Algebra Control Optim., 6 (2016), pp. 319–328.

[2] R. W. Brockett and D. Liberzon, *Quantized feedback stabilization of linear systems*, IEEE Trans. Automat. Control, 45 (2000), pp. 1279–1289.

[3] H.-F. Chen, *Stochastic Approximation and Its Applications*, Nonconvex Optim. Appl. 64, Springer, New York, 2006.

[4] A. G. Dimakis, S. Kar, J. M. Moura, M. G. Rabbat, and A. Scaglione, *Gossip algorithms for distributed signal processing*, Proc. IEEE, 98 (2010), pp. 1847–1864.

[5] J. C. Duchi, A. Agarwal, and M. J. Wainwright, *Dual averaging for distributed optimization: Convergence analysis and network scaling*, IEEE Trans. Automat. Control, 57 (2012), pp. 592–606.

[6] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, *Average consensus on networks with quantized communication*, Internat. J. Robust Nonlinear Control, 19 (2009), pp. 1787–1816.

[7] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, and V. Volkov, *Parallel computing experiences with CUDA*, IEEE Micro, 28 (2008), pp. 13–27.

[8] A. Jadbabaie, J. Lin, and A. S. Morse, *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, IEEE Trans. Automat. Control, 48 (2003), pp. 988–1001.

[9] S. Kar, J. M. Moura, and K. Ramanan, *Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication*, IEEE Trans. Inform. Theory, 58 (2012), pp. 3575–3605.

[10] A. Kashyap, T. Başar, and R. Srikant, *Quantized consensus*, Automatica, 43 (2007), pp. 1192–1203.

[11] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, and D. Glasco, *GPUs and the future of parallel computing*, IEEE Micro, 31 (2011), pp. 7–17.

[12] C. E. Lee, A. Ozdaglar, and D. Shah, *Solving Systems of Linear Equations: Locally and Asynchronously*, Computing Research Repository, 2014.

[13] C.-S. Lee, N. Michelusi, and G. Scutari, *Finite rate quantized distributed optimization with geometric convergence*, in Proceedings of the 52nd Asilomar Conference on Signals, Systems, and Computers, IEEE, 2018, pp. 1876–1880.

[14] J. Lei and H.-F. Chen, *Distributed randomized pagerank algorithm based on stochastic approximation*, IEEE Trans. Automat. Control, 60 (2015), pp. 1641–1646.

[15] J. Lei, H.-F. Chen, and H.-T. Fang, *Primal–dual algorithm for distributed constrained optimization*, Systems Control Lett., 96 (2016), pp. 110–117.

[16] J. Lei, P. Yi, G. Shi, and B. D. O. Anderson, *Network linear equations with finite data rates*, in Proceedings of the IEEE Conference on Decision and Control, IEEE, 2018, pp. 3373–3378.

[17] T. Li, M. Fu, L. Xie, and J.-F. Zhang, *Distributed consensus with limited communication data rate*, IEEE Trans. Automat. Control, 56 (2011), pp. 279–292.

[18] I. Lobel, A. Ozdaglar, and D. Feijer, *Distributed multi-agent optimization with state-dependent communication*, Math. Program., 129 (2011), pp. 255–284.

[19] J. Lu and C. Y. Tang, *Distributed asynchronous algorithms for solving positive definite linear equations over networks—Part* I*: Agent networks*, IFAC Proceedings Volumes, 42 (2009), pp. 252–257.

[20] J. Lu and C. Y. Tang, *Distributed asynchronous algorithms for solving positive definite linear equations over networks—part* II*: Wireless networks*, IFAC Proceedings Volumes, 42 (2009), pp. 258–263.

[21] S. Magnússon, C. Enyioha, N. Li, C. Fischione, and V. Tarokh, *Communication complexity of dual decomposition methods for distributed resource allocation optimization*, IEEE J. Selected Topics Signal Process., 12 (2018), pp. 717–732.

[22] S. Magnússon, H. Shokri-Ghadikolaei, and N. Li, *On Maintaining Linear Convergence of Distributed Learning and Optimization Under Limited Communication*, https://arxiv.org/abs/1902.11163, 2019.

[23] R. Mehmood and J. Crowcroft, *Parallel Iterative Solution Method for Large Sparse Linear Equation Systems*, Tech. report, Computer Laboratory, University of Cambridge, 2005.
[24] D. Mosk-Aoyama, T. Roughgarden, and D. Shah, *Fully distributed algorithms for convex optimization problems*, SIAM J. Optim., 20 (2010), pp. 3260–3279.
[25] S. Mou, J. Liu, and A. S. Morse, *A distributed algorithm for solving a linear algebraic equation*, IEEE Trans. Automat. Control, 60 (2015), pp. 2863–2878.
[26] S. Mou and A. Morse, *A fixed-neighbor, distributed algorithm for solving a linear algebraic equation*, in Proceedings of ECC, IEEE, 2013, pp. 2269–2273.
[27] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, *Feedback control under data rate constraints: An overview*, Proc. IEEE, 95 (2007), pp. 108–137.
[28] A. Nedic, A. Olshevsky, and W. Shi, *Achieving geometric convergence for distributed optimization over time-varying graphs*, SIAM J. Optim., 27 (2017), pp. 2597–2633.
[29] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, *Geometrically convergent distributed optimization with uncoordinated step-sizes*, in Proceedings of the American Control Conference, IEEE, 2017, pp. 3950–3955.
[30] A. Nedić and A. Ozdaglar, *Distributed subgradient methods for multi-agent optimization*, IEEE Trans. Automat. Control, 54 (2009), pp. 48–61.
[31] A. Nedić, A. Ozdaglar, and P. A. Parrilo, *Constrained consensus and optimization in multi-agent networks*, IEEE Trans. Automat. Control, 55 (2010), pp. 922–938.
[32] Z. Qiu, L. Xie, and Y. Hong, *Quantized leaderless and leader-following consensus of high-order multi-agent systems with limited data rate*, IEEE Trans. Automat. Control, 61 (2016), pp. 2432–2447.
[33] G. Qu and N. Li, *Harnessing smoothness to accelerate distributed optimization*, IEEE Trans. Control Netw. Syst., 5 (2017), pp. 1245–1260.
[34] M. Rabbat and R. Nowak, *Distributed optimization in sensor networks*, in Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, ACM, 2004, pp. 20–27.
[35] M. G. Rabbat and R. D. Nowak, *Quantized incremental algorithms for distributed optimization*, IEEE J. Selected Areas Commun., 23 (2005), pp. 798–808.
[36] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, *An exact quantized decentralized gradient descent algorithm*, IEEE Trans. Signal Process., 67 (2019), pp. 4934–4947.
[37] Y. Saad and M. Sosonkina, *Distributed Schur complement techniques for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 1337–1356.
[38] G. Shi, B. D. O. Anderson, and U. Helmke, *Network flows that solve linear equations*, IEEE Trans. Automat. Control, 62 (2017), pp. 2659–2674.
[39] G. Shi, K. H. Johansson, and Y. Hong, *Reaching an optimal consensus: Dynamical systems that compute intersections of convex sets*, IEEE Trans. Automat. Control, 58 (2013), pp. 610–622.
[40] W. Shi, Q. Ling, G. Wu, and W. Yin, *Extra: An exact first-order algorithm for decentralized consensus optimization*, SIAM J. Optim., 25 (2015), pp. 944–966.
[41] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, *Communication compression for decentralized training*, in Advances in Neural Information Processing Systems, 2018, pp. 7663–7673.
[42] J. Tsitsiklis, D. Bertsekas, and M. Athans, *Distributed asynchronous deterministic and stochastic gradient optimization algorithms*, IEEE Trans. Automat. Control, 31 (1986), pp. 803–812.
[43] R. Tutunov, H. B. Ammar, and A. Jadbabaie, *A Fast Distributed Solver for Symmetric Diagonally Dominant Linear Equations*, https://arxiv.org/abs/1502.03158, 2015.
[44] J. Von Neumann, *On rings of operators. reduction theory*, Ann. of Math., 50 (1949), pp. 401–485.
[45] J. Wu, W. Huang, J. Huang, and T. Zhang, *Error compensated quantized SGD and its applications to large-scale distributed optimization*, in Proceedings of the International Conference on Machine Learning, 2018, pp. 5321–5329.
[46] L. Xiao and S. Boyd, *Fast linear iterations for distributed averaging*, Systems Control Lett., 53 (2004), pp. 65–78.
[47] P. Yi and Y. Hong, *Quantized subgradient algorithm and data-rate analysis for distributed optimization*, IEEE Trans. Control Netw. Syst., 1 (2014), pp. 380–392.
[48] M. Yu, Z. Lin, K. Narra, S. Li, Y. Li, N. S. Kim, A. Schwing, M. Annavaram, and S. Avestimehr, *GradiVeQ: Vector quantization for bandwidth-efficient gradient aggregation in distributed CNN training*, in Advances in Neural Information Processing Systems, 2018, pp. 5129–5139.

[49] K. Yuan, Q. Ling, and W. Yin, *On the convergence of decentralized gradient descent*, SIAM J. Optim., 26 (2016), pp. 1835–1854.
[50] A. Zouzias and N. M. Freris, *Randomized extended kaczmarz for solving least squares*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 773–793.