2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Network Linear Equations with Finite Data Rates

Jinlong Lei, Peng Yi, Guodong Shi, and Brian D. O. Anderson

Abstract— **In this paper, we propose a distributed quantized algorithm for solving the network linear equation $\mathrm{z} = \mathrm{H}\mathrm{y}$ subject to digital node communications, where each node only knows a single row of the partitioned matrix $[\mathrm{H}\ \mathrm{z}]$. Each node holds a dynamic state and interacts with its neighbors through an undirected connected graph. Due to the data-rate constraint, each node builds an encoder-decoder pair, with which it produces transmitted message with a zooming-in finite-level uniform quantizer and also generates estimates of its neighbors' states from the received signals. When the equation admits a unique solution, the algorithm drives all nodes' estimates to converge exponentially fast to that solution. When a unique least-squares solution exists, such a solution can be obtained with a suitably selected time-varying step size. In both cases, a minimal data rate with the three-level quantizers is shown to be enough for guaranteeing the desired convergence.**

I. INTRODUCTION

The pursuit of resilient and scalable solutions for the control and optimization in large-scale network systems has drawn much research attention in the field of systems and control in the past decade [1], [2]. In networked systems, the collective goals such as consensus, formation and estimation can be addressed through local interactions between nodes. These distributed protocols provide resilience in the sense that nodes and links can join and leave the network without significantly affecting the performance of the network; they also provide scalability compared to centralized solutions. Simultaneously, control theory has embraced to a much greater degree than graph theory, communication theory, and complexity analysis, leading to many celebrated results for both theories and applications [3].

Particularly, systems of linear algebraic equations, as one primary computation task, can be naturally defined over a network with each node holding one or a few of the linear equations [4]. Besides, network linear equations arise from resource allocation problems when node cost functions are quadratic [5]–[7]. In the context of parallel computation, the objective is to develop algorithms that eventually compute one coordinate of the solutions [8]–[11]. While in view of distributed optimization [12]–[14], distributed algorithms computing the entire solution vector at each node were proposed for both discrete-time and continuous-time node dynamics [4], [15]–[22]. In fact, when exact solutions exist for the linear equations, such first-order distributed solvers were generalized versions of the so-called alternation projection algorithms pioneered by von Neumann [12], [23], [24]. When no exact solution exists and one considers least-squares solutions, algorithms using properly selected square-summable diminishing step-sizes are needed [18], [19].

In this paper, we consider network linear equation solvers subject to digital node communications *where only a finite data rate is available* [25]–[29]. 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, and communicate with each other through an undirected connected graph, where along each link the neighboring nodes exchange information with a finite data rate. 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 its neighbors' states, based on which its own state is updated with the proposed algorithm. We have established the following results:

(i) The network linear equation admits a unique exact solution. We show that the distributed quantized algorithm drives each node's state to that solution exponentially with a proper constant step size and quantization level for any given initial bounded node states. It is also shown that a minimal data rate of m bits per step is enough for guaranteeing the convergence.

(ii) The network linear equation admits a unique least-squares solution. We show that the same encoder-decoder pair enables the algorithm to compute such a solution with a time-varying step-size. Again a data rate of m bits per step can deliver such a convergence result.

These results guarantee the practical use of the various network linear equation solvers in digital point-to-point communications. We also note that our results are closely related to the work on distributed optimization algorithms with quantized communication [30], [31]. Though the network linear equation appears to be a special case of quadratic program, new challenges lie in that gradients of the quadratic function associated with each node cannot be assumed to

J. Lei is with the Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park 16802, PA, USA (email:jxl800@psu.edu); P. Yi is with the Department of Electrical and Systems Engineering, Washington University in St. Louis, 1 Brookings Drive, St. Louis, MO 63130, USA (email: yipeng@amss.ac.cn); G. Shi is with the Research School of Engineering, The Australian National University, ACT 0200, Canberra, Australia (email: guodong.shi@anu.edu.au); B. D. O. Anderson is with the Hangzhou Dianzi University, Hangzhou, China, the Research School of Engineering, Australian National University and Data61-CSIRO, Canberra, ACT 0200, Australia (email: brian.anderson@anu.edu.au). The work of Anderson, was supported by the Australian Research Council (ARC) under grant DP-160104500, and by Data61-CSIRO.

be globally bounded a priori, a key technical assumption in distributed (sub)gradient optimization [12], [31].

The remainder of this paper is organized as follows. Section II defines the network linear equation, while Section III introduces the node encoders and decoders, and develops the distributed quantized algorithm. Section IV investigates the convergence properties for both the exact solver and the least-squares solver. Section V demonstrates the numerical examples with the concluding remarks given in Section VI.

Notation and Terminology. All vectors are column vectors and denoted by bold, lower case letters, i.e., $\mathbf{a}, \mathbf{b}, \mathbf{c}$, etc.; matrices are denoted with bold, upper case 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 $\|\cdot\|$. An undirected graph is an ordered pair of two sets denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here $\mathcal{V} = \{1, \dots, N\}$ is a finite set of vertices (nodes). Each element in $\mathcal{E}$ is an unordered pair of two distinct nodes in $\mathcal{V}$, called an edge. A path in $\mathcal{G}$ with length p from v_1 to v_{p+1} is a sequence of distinct nodes, $v_1 v_2 \dots v_{p+1}$, such that $(v_m, v_{m+1}) \in \mathcal{E}$, for all $m = 1, \dots, p$. 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 $\mathbf{D} = diag\{|\mathcal{N}_1|, \cdots, |\mathcal{N}_N|\}$ and the adjacency matrix $\mathbf{A}$, where $[\mathbf{A}]_{ij} = 1$ if $j \in \mathcal{N}_j$ and $[\mathbf{A}]_{ij} = 0$, otherwise. Then $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the Laplacian matrix of the graph $\mathcal{G}$.

II. Network Linear Equations

Consider the following linear algebraic equation:

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

with respect to variable $\mathbf{y} \in \mathbb{R}^m$, where $\mathbf{H} \in \mathbb{R}^{N \times m}$ and $\mathbf{z} \in \mathbb{R}^N$. The equation (1) has a unique exact solution if $\mathrm{rank}(\mathbf{H}) = m$ and $\mathbf{z} \in \mathrm{span}(\mathbf{H})$; an infinite set of solutions if $\mathrm{rank}(\mathbf{H}) < m$ and $\mathbf{z} \in \mathrm{span}(\mathbf{H})$; and no exact solutions if $\mathbf{z} \notin \mathrm{span}(\mathbf{H})$. 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 $\mathbf{y}^\star = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z}$ if $\mathrm{rank}(\mathbf{H}) = m$.

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 i-th row vector of $\mathbf{H}$. Consider a network with N nodes indexed as $\mathcal{V} = \{1, \dots, N\}$, where node i only has access to the value of $\mathbf{h}_i$ and z_i without the knowledge of $\mathbf{h}_j$ or z_j from other nodes. The network interaction structure is described by a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the corresponding Laplacian matrix denoted by $\mathbf{L}$. Time is slotted at $k = 0, 1, 2, \dots$. Node i at time k holds an estimate $\mathbf{x}_i(k) \in \mathbb{R}^m$ for the solution to equation (1).

As the Euler approximation of the "consensus + projection" flow [17], the following algorithm is an efficient distributed linear equations solver with a discrete recursion:

$$\begin{aligned} \mathbf{x}_i(k+1) = \mathbf{x}_i(k) + h\Big(& \sum_{j \in \mathcal{N}_i} \left(\mathbf{x}_j(k) - \mathbf{x}_i(k)\right) \\ & - \gamma(k) \left(\mathbf{h}_i \mathbf{h}_i^\top \mathbf{x}_i(k) - z_i \mathbf{h}_i\right) \Big), \end{aligned} \tag{3}$$

of which the performance is investigated in [17], [19].

III. Distributed Quantized Algorithm

Consider the settings where the communication channels corresponding to the edges in the network have a finite bandwidth. As such, real-valued data should be quantized before transmitting. However, it is noticed from the algorithm (3) that nodes need to exchange their exact state values. Then in this section, we propose a distributed quantized algorithm, in which each node is associated with an encoder while its neighboring nodes possess a corresponding decoder. Let us begin by introducing a quantization function $Q_K(\cdot)$.

Definition 1 (Quantization Function): A standard uniform quantizer is given by the function $Q_K(\cdot) : \mathbb{R} \to \{-K, \dots, -1, 0, 1, \dots, K\}$ where

$$Q_K(z) = \begin{cases} 0, & \text{if } -1/2 \le z \le 1/2, \\ i, & \text{if } \frac{2i-1}{2} < z \le \frac{2i+1}{2},\ i = 1, \cdots, K, \\ K, & \text{if } z > \frac{2K+1}{2}, \\ -Q_K(-z), & \text{if } z > -1/2. \end{cases} \tag{4}$$

With slight abuse of notation, we define $Q_K(\mathbf{a})$ for a vector $\mathbf{a} = (a_1, \cdots, a_m)^T \in \mathbb{R}^m$ by

$$Q_K(\mathbf{a}) = (Q_K(a_1), \cdots, Q_K(a_m))^T.$$

Note that to transmit the quantized information from the quantizer $Q_K(\mathbf{a})$ for $\mathbf{a} \in \mathbb{R}^m$, the communication channel is required to be capable of transmitting $m\lceil \log_2(2K) \rceil$ bits.

Next, we propose an encoder-decoder pair for each node to quantize its state, and to estimate the neighbors' states. Suppose the nodes have a global scaling function $s(k)$. We still use $\mathbf{x}_i(k)$ to denote the un-quantized state of node i at time k, whose update will be specified at a later stage.

[Encoder] The encoder of node $j \in \mathcal{V}$ recursively generates quantized outputs $\{\mathbf{q}_j(k)\}$ and internal states $\{\mathbf{b}_j(k)\}$ from the exact state sequence $\{\mathbf{x}_j(k)\}$ as follows for any $k \ge 1$:

$$\begin{aligned} \mathbf{q}_j(k) &\triangleq Q_K\left(\frac{1}{s(k-1)}(\mathbf{x}_j(k) - \mathbf{b}_j(k-1))\right), \\ \mathbf{b}_j(k) &\triangleq s(k-1)\mathbf{q}_j(k) + \mathbf{b}_j(k-1), \end{aligned} \tag{5}$$

where the initial value $\mathbf{b}_j(0) = 0$. At time k, node j sends $\mathbf{q}_j(k)$ to its neighboring nodes $i \in \mathcal{N}_j$.

Remark 1: 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 $\mathbf{x}_j(k) - \mathbf{b}_j(k-1)$). 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 $s(k)$ decay 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.

[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 $\hat{\mathbf{x}}_j(k)$ for $\mathbf{x}_j(k)$ by the following for any $k \geq 1$:

$$\hat{\mathbf{x}}_j(k) \triangleq s(k-1)\mathbf{q}_j(k) + \hat{\mathbf{x}}_j(k-1), \tag{6}$$

where the initial value $\hat{\mathbf{x}}_j(0) \triangleq 0$.

Based on the encoder-decoder pair defined in (5) and (6), motivated by (3), we now propose the following distributed linear equation solver with quantized node communication.

Algorithm 1 Distributed quantized algorithm

$$\begin{aligned}\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + h\Big[&\sum_{j\in\mathcal{N}_i} \left(\hat{\mathbf{x}}_j(k) - \mathbf{b}_i(k)\right) \\ &- \gamma(k)\left(\mathbf{h}_i\mathbf{h}_i^\top \mathbf{x}_i(k) - z_i\mathbf{h}_i\right)\Big].\end{aligned} \tag{7}$$

It is clear that Algorithm 1 relies on quantized node communication only since $\mathbf{q}_j(k)$ takes values in the alphabet $\{-K, \ldots, -1, 0, 1, \ldots, K\}$ only. From (5), (6) and the assumed initial conditions of zero for $\hat{\mathbf{x}}_j(0)$ and $\mathbf{b}_j(0)$, we have that for any $k \geq 0$, $\hat{\mathbf{x}}_j(k) = \mathbf{b}_j(k) \quad \forall j \in \mathcal{V}$.

IV. Convergence Results

In this section, we consider Algorithm 1 and seperately address the convergence results regarding the quantization level along with the rate analysis for both the exact solver and leat-squares solver.

A. Exact Solver

We now investigate the case that equation (1) has a unique solution y^*. We need the following assumptions.

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

A2 $\max_i \|\mathbf{x}_i(0)\|_\infty \leq C_x$ and $\max_i \|\mathbf{x}_i(0) - \mathbf{y}^*\|_\infty \leq C_w$ for some positive constants C_x and C_w.

A3 $\gamma(k) \equiv 1$, and $s(k) \triangleq s(0)\alpha^k \ \forall k \geq 0$ for some $s(0) > 0$ and $\alpha \in (0, 1)$.

We now introduce a few useful notations as follows:

$$\begin{aligned}&\mathbf{H_d} \triangleq \mathrm{diag}\left\{\mathbf{h}_1\mathbf{h}_1^\top, \ldots, \mathbf{h}_N\mathbf{h}_N^\top\right\} \in \mathbb{R}^{mN\times mN}, \\ &\mathbf{F_d} \triangleq \mathbf{L}\otimes\mathbf{I}_m + \mathbf{H_d}, \ \rho_h \triangleq 1 - h\lambda_{\min}(\mathbf{F_d}),\end{aligned} \tag{8}$$

where $\lambda_{\min}(\mathbf{F_d})$ denotes the smallest eigenvalue of $\mathbf{F_d}$. Note that both the Laplacian matrix $\mathbf{L}$ and the matrix $\mathbf{H_d}$ are positive semidefinite. With the assumption **A1** and the condition that the undirected graph $\mathcal{G}$ is connected, the matrix $\mathbf{F_d}$ turns out to be positive definite [17, Lemma 9], and all eigenvalues of $\mathbf{F_d}$ is positive. The eigenvalues of $\mathbf{L}$ in an ascending order are denoted by $0 = \lambda_1(\mathbf{L}) < \lambda_2(\mathbf{L}) \leq \cdots \leq \lambda_N(\mathbf{L})$. Let $h \in \left(0, \frac{2}{\lambda_{\min}(\mathbf{F_d}) + \lambda_{\max}(\mathbf{F_d})}\right)$ and $\alpha \in (1 - h\lambda_{\min}(\mathbf{F_d}), 1)$. We set

$$\begin{aligned}&M(\alpha,h) \triangleq \frac{1+2hd^*}{2\alpha} + \frac{h^2\sqrt{mN}\lambda_N(\mathbf{L})\lambda_{\max}(\mathbf{F_d})}{2\alpha(\alpha-\rho_h)}, \\ &\text{and } \mathcal{K}(\alpha,h) \triangleq \left\lceil M(\alpha,h) - \frac{1}{2}\right\rceil,\end{aligned} \tag{9}$$

where $d^* = \max_i |\mathcal{N}_i|$ denotes the degree of $\mathcal{G}$, and $\lambda_{\max}(\mathbf{F_d})$ denotes the largest eigenvalue of $\mathbf{F_d}$.

We now present our main result on the performance of Algorithm 1 as an exact solver for the linear equation (1).

Theorem 1: Suppose **A1**, **A2** and **A3** hold. Let $h \in \left(0, \frac{2}{\lambda_{\min}(\mathbf{F_d}) + \lambda_{\max}(\mathbf{F_d})}\right)$ 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

$$\lim_{k\to\infty} x_i(k) = \mathbf{y}^* \quad \forall i \in \mathcal{V} \tag{10}$$

provided $s(0)$ satisfying

$$\begin{aligned}s(0) > \max\Bigg\{&\frac{C_x + h\|\mathbf{H_d}\|_\infty C_w}{K + \frac{1}{2}}, \\ &\frac{2(\alpha-\rho_h)\left(\rho_h C_w + hC_x\lambda_N(\mathbf{L})\right)}{h\lambda_N(\mathbf{L})}\Bigg\}.\end{aligned} \tag{11}$$

The convergence is in fact exponential with

$$\limsup_{k\to\infty} \frac{\|\mathbf{x}(k) - \mathbf{1}_N\otimes\mathbf{y}^*\|_2}{\alpha^k} \leq \frac{hs(0)\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)}, \tag{12}$$

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

Proof. The proof can be found in [32, Section 3.4.3]. □

Remark 2: Theorem 1 shows that by using a scaling function decaying exponentially and a uniform quantizer, Algorithm 1 can ensure asymptotic convergence to the unique solution. It is worth pointing out that for any given α, h, the obtained quantization level $\mathcal{K}(\alpha, h)$ is conservative, while (9) gives us some intuition on the relationship between the number of bits required and the control gains and the scaling factor. In addition, Theorem 1 gives an estimate of the rate of convergence: the smaller the scaling factor α, the faster the convergence rate from (12) but more bits have to be communicated by (9), and, if $\alpha \to \rho_h$, the required number of bits goes to infinity. Thus, an appropriate selection of α amounts to a tradeoff between the rate of convergence and the communication overhead.

From (9) we know that for fixed α, the quantization level $\mathcal{K}(\alpha, h)$ will tend to infinity as $N \to \infty$. But in practical applications, the communication channel usually has a finite bandwidth. To satisfy this requirement, we could use a fixed number of quantization levels at the cost of slower convergence. The following result answers this qustion.

Theorem 2: Suppose **A1**, **A2**, and **A3** hold. Then the following hold.
(i) For any $K \geq 1$, Ξ_K is nonempty with

$$\Xi_K \triangleq \Big\{(\alpha, h) : h \in \Big(0, \frac{2}{\lambda_{\min}(\mathbf{F_d}) + \lambda_{\max}(\mathbf{F_d})}\Big), \quad \alpha \in (1 - h\lambda_{\min}(\mathbf{F_d}), 1), M(\alpha, h) < K + \frac{1}{2}\Big\}. \tag{13}$$

(ii) For any $K \geq 1$, let $(\alpha, h) \in \Xi_K$ and $s(0)$ satisfy (11). Then along Algorithm 1 there hold Eqn. (10) and Eqn. (12).
Proof. The proof can be found in [32, Section 3.4.4]. □

Remark 3: From Theorems 1 and 2, it is clear that we can always design a distributed protocol to ensure exponentially fast convergence to exact solution with $3-$level quantizers (namely, $K = 1$) and each node sending merely m bits of information (**minimum bits**) to its neighbors at each step.

B. Least-Squares Solver

We proceed to investigate the case $\mathrm{rank}(\mathbf{H}) = m$ and $\mathbf{z} \notin \mathrm{span}(\mathbf{H})$. Then equation (1) does not have exact solutions, while a least-squares solution is defined as the solution to the optimization problem (2). In this case, Assumptions **A1**, **A2** and **A3** are no longer in force, instead, we impose the following conditions.
A4 $\mathrm{rank}(\mathbf{H}) = m$ and $\mathbf{z} \notin \mathrm{span}(\mathbf{H})$.
A5 $\max_i \|\mathbf{x}_i(0)\|_\infty \leq C_x$ for constant $C_x > 0$.
A6 (i) $\gamma(0) = 1$, $\gamma(k) \downarrow 0$, $\sum_{k=1}^{\infty} \gamma(k) = \infty$, (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 $\beta(k) \triangleq \frac{\gamma(k)}{\gamma(k+1)}$.

Let $h \in \big(0, \frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}\big)$ and $\beta(0) \in \big(1, \frac{1}{1-h\lambda_2(\mathbf{L})}\big)$. We introduce some useful notations:

$$\begin{aligned} &\hat{\rho}_h \triangleq 1 - h\lambda_2(\mathbf{L}), \ \mathbf{z}_H \triangleq \big(z_i\mathbf{h}_1^T, \ldots, z_N\mathbf{h}_N^T\big)^T, \\ &M'(h, \beta(0)) \triangleq (1/2 + hd^*)\beta(0) + 2hM_2(h, \beta(0)), \\ &\mathcal{K}'(h, \beta(0)) \triangleq \Big\lceil M'(h, \beta(0)) - \frac{1}{2}\Big\rceil, \end{aligned} \tag{14}$$

where $M_1(h, \beta(0))$ and $M_2(h, \beta(0))$ are defined as follows:

$$\begin{aligned} M_1(h, \beta(0)) &\triangleq \Big(\sqrt{mN}C_x(1 + h\lambda_N(\mathbf{L})) + \frac{2\|\mathbf{z}_H\|_2}{\lambda_{\min}(\mathbf{F_d})}\Big) \\ &\times \Big(\|\mathbf{H_d}\|_\infty + \frac{h\lambda_N(\mathbf{L})\|\mathbf{H_d}\|_2}{1/\beta(0) - \hat{\rho}_h}\Big) + \|\mathbf{z}_H\|_\infty \\ &+ \lambda_N(\mathbf{L})\Big(\sqrt{mN}C_x\,(1 + h\beta(0)\lambda_N(\mathbf{L})) + \frac{h\|\mathbf{z}_H\|_2}{1/\beta(0) - \hat{\rho}_h}\Big), \\ M_2(h, \beta(0)) &\triangleq \beta(0)\sqrt{mN}\lambda_N(\mathbf{L})\Big(\frac{h\lambda_N(\mathbf{L})}{2(1/\beta(0) - \hat{\rho}_h)} \\ &+ \frac{1}{\lambda_{\min}(\mathbf{F_d})}\Big(\|\mathbf{H_d}\|_\infty + \frac{h\lambda_N(\mathbf{L})\|\mathbf{H_d}\|_2}{1/\beta(0) - \hat{\rho}_h}\Big)\Big), \end{aligned}$$

where $\mathbf{H_d}$ and $\mathbf{F_d}$ are defined in (8). We now ready to state the main result of the Algorithm 1.

Theorem 3: Suppose **A4**, **A5**, and **A6** hold. Let $h \in \Big(0, \min\big\{\frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{F_d})}\big\}\Big)$ and $\beta(0) \in (1, 1/(1 - h\lambda_2(\mathbf{L})))$. Then for any given $K \geq \mathcal{K}'(h, \beta(0))$, along Algorithm 1 there hold:

$$\lim_{k\to\infty} \mathbf{x}_i(k) = \mathbf{y}_{\mathrm{LS}}{}^* \triangleq (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z} \quad \forall i \in \mathcal{V}, \tag{15}$$

$$\limsup_{k\to\infty} \frac{\|\mathbf{x}_i(k) - \mathbf{y}_{\mathrm{LS}}{}^*\|_\infty}{\gamma(k)} < \infty \tag{16}$$

provided that s_r satisfies

$$s_r > \max\Big\{\frac{C_x + h(C_x\|\mathbf{H_d}\|_\infty + \|\mathbf{z}_H\|_\infty)}{K + \frac{1}{2}}, \quad M_1(h, \beta(0))/M_2(h, \beta(0))\Big\}. \tag{17}$$

Proof. The proof can be found in [32, Section 4.3.3]. □

Similarly to Theorem 2 for the exact solver, the following result reinforces that we can also design a distributed least-squares solver to converge to a least-squares solution with $3-$level quantizers, which is the **minimal** quantizer level.

Theorem 4: Suppose **A4**, **A5** and **A6** hold. Then the following hold:
(i) For any $K \geq 1$, Ξ'_K is nonempty with

$$\begin{aligned} \Xi'_K \triangleq \Big\{&(h, \beta(0)) : \beta(0) \in \big(1, 1/(1 - h\lambda_2(\mathbf{L}))\big), \\ &h \in \Big(0, \min\big\{\frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})}, \frac{1}{\lambda_{\min}(\mathbf{F_d})}\big\}\Big), \\ &M'(h, \beta(0)) \leq K + \frac{1}{2}\Big\}. \end{aligned} \tag{18}$$

(ii) For any given $K \geq 1$, let $(h, \beta(0)) \in \Xi'_K$ and s_r satisfy (17). Then along Algorithm 1, (15) and (16) continue to hold.
Proof. The proof can be found in [32, Section 4.3.3]. □

V. Numerical Example

In this section, we conduct numerical simulations to demonstrate the established convergence properties of the developed distributed quantized algorithm (Algorithm 1).
Example 1. Let the linear equation (1) be given by

$$\mathbf{H} = \begin{pmatrix} 0.5 & -0.1 \\ -0.4 & 0.2 \\ 0.3 & -0.7 \\ 0.6 & 0.3 \\ -0.3 & 0.5 \end{pmatrix}, \mathbf{z} = \begin{pmatrix} 0.2 \\ 0.2 \\ -1.8 \\ 1.5 \\ 1.2 \end{pmatrix} \tag{19}$$

which yields a unique exact solution

$$\mathbf{y}^* = \begin{pmatrix} 1 \\ 3 \end{pmatrix}.$$

The network structure is shown in Figure 1.

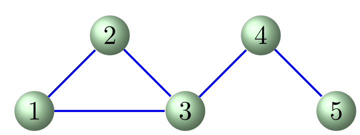

Fig. 1: Communication graph.

[**Validation of Theorem 1.**] Let $h = \frac{1.98}{\lambda_{\min}(\mathbf{F_d})+\lambda_{\max}(\mathbf{F_d})} = 0.4215$. Thereby one can compute $\rho_h = 0.9554$. Set $\alpha = 0.98$ so that $\mathcal{K}(\alpha, h) = 225$. Let K be 100, 300, 1000, respectively. We set $s(0) = 1$ and implement Algorithm 1.

Figure 2 displays the trajectories of $||\mathbf{x}(k)-\mathbf{1}_N\otimes\mathbf{y}^*||_2$ 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 (12). The trajectory with $K=300$ verifies that Theorem 1 provides a sufficient condition on the data rate to ensure convergence, while the trajectories for $K=100$ and $K=1000$ coincide with that of $K=300$. Therefore, it implies that (i) with the same algorithm parameters h,α, a higher data rate ($K=1000$) cannot guarantee a faster convergence rate; (ii) there is some degree of conservativeness in the sufficient condition of Theorem 1.

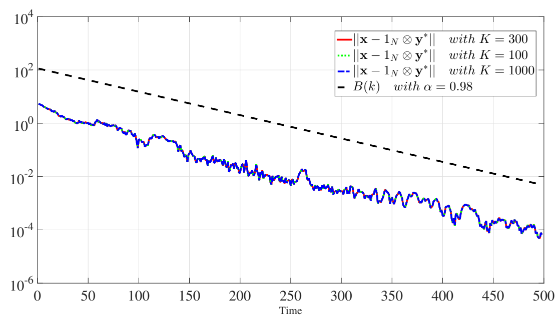

Fig. 2: Trajectories of $||\mathbf{x}-\mathbf{1}_N\otimes\mathbf{y}^*||_2$ and upper bound B_k under $K=100,300,1000$ along with the theoretical bound.

[Validation of Theorem 2.] Let K be $K_1=3$, $K_2=6$ and $K_3=12$, respectively. We choose $(\alpha_1,h_1)=(0.9998,0.0038)$, $(\alpha_1,h_1)=(0.9996,0.0077)$, and $(\alpha_3,h_3)=(0.9992,0.0154)$. We set $s_1(0)=1500, s_2(0)=1200, s_3(0)=1000$ for K_1, K_2, K_3, respectively, to ensure (11). 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 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 also consistent with the upper bound of convergence rate $B(k)=\frac{hs(0)\alpha^k\sqrt{mN}\lambda_N(\mathbf{L})}{2\alpha(\alpha-\rho_h)}$ given by (12) in all three parameter settings.

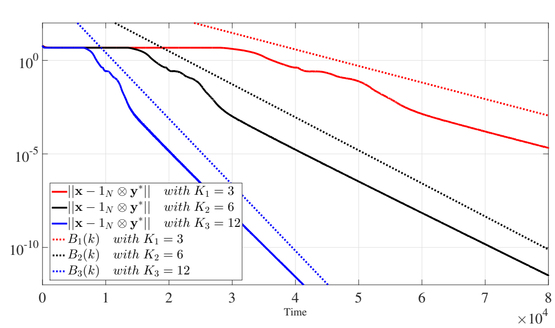

Fig. 3: Trajectories of $||\mathbf{x}-\mathbf{1}_N\otimes\mathbf{y}^*||_2$ and upper bounds $B(k)$ with $K_1=3$, $K_2=6$ and $K_3=12$, respectively.

Example 2 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 of $\mathbf{y}^*=\arg\min||\mathbf{z}-\mathbf{H}\mathbf{y}||^2$ is $\mathbf{y}^*=(0.1415,0.6391)^T$. The nodes again communicate according to the graph shown in Fig 1.

[Validation of Theorem 3.] 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 (17) 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 4, which shows that the trajectories of $||\mathbf{x}-1_N\otimes\mathbf{y}||^2$ coincide in all three cases. It then implies that i) once the sufficient condition of Theorem 3 is satisfied, increasing data rate solely cannot speed up convergence; ii) the condition in Theorem 3 is sufficient for convergence but is not necessary. Figure 4 also shows the trajectory of $\frac{||\mathbf{x}(k)-\mathbf{y}^*||_\infty}{\gamma(k)}$, which verifies the convergence rate described by (16).

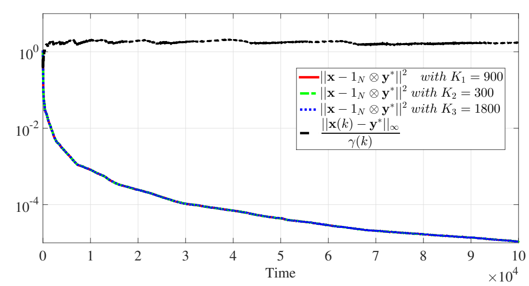

Fig. 4: The trajectories of the sum of squared distance to the least square solution under $K=300,900,1800$.

[Validation of Theorem 4.] We set the quantization level K to be $K_1=10$, $K_2=30$ and $K_3=90$, respectively. Then we select algorithm parameters h and $s(k)=s_r\gamma(k)=\frac{s_rk_0^\delta}{(k+k_0)^\delta}$ such that $(h,\beta(0))\in\Xi_K^{'}$ and s_r satisfies (17) for the three cases. The algorithm parameters for the three cases are given in Table I. The trajectories of $||\mathbf{x}-\mathbf{1}_N\otimes\mathbf{y}^*||^2$ are shown in Figure 5, which demonstrates the convergence of Algorithm 1 with the chosen parameters, verifying Theorem 4. It also shows that with a higher data rate, the convergence could be faster if algorithm parameters are properly chosen.

	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

TABLE I: Parameter settings

VI. Conclusions

We have studied solving linear equations over a network subject to digital node communications with a limited data

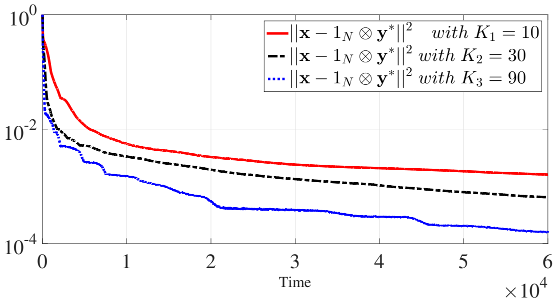

Fig. 5: Trajectories of $||\mathbf{x}-\mathbf{1}_N \otimes \mathbf{y}^*||^2$ for $K_1 = 20, 50, 100$ with the algorithm parameters chosen in Table I.

rate. 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 drove each node state to the solution asymptotically at an exponential rate; For the unique least-squares solution case, such a solution is obtained with a properly selected time-varying step size. A minimal data rate was shown to be enough for the desired convergence for both cases. These results suggest the practical applicability of various network linear equation solvers. Future work includes more explicit characterization of the trade-off between the convergence rate and the communication overhead, and convergence conditions of the proposed algorithm in the presence of switching node communication structures.

References

[1] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

[2] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on automatic control*, vol. 48, no. 6, pp. 988–1001, 2003.

[3] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[4] S. Mou and A. Morse, "A fixed-neighbor, distributed algorithm for solving a linear algebraic equation," in *Control Conference (ECC), 2013 European*. IEEE, 2013, pp. 2269–2273.

[5] 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.

[6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[7] P. Yi, Y. Hong, and F. Liu, "Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems," *Automatica*, vol. 74, pp. 259–269, 2016.

[8] Y. Saad and M. Sosonkina, "Distributed schur complement techniques for general sparse linear systems," *SIAM Journal on Scientific Computing*, vol. 21, no. 4, pp. 1337–1356, 1999.

[9] C. Andersson, "Solving linear eqauations on parallel distributed memory architectures by extrapolation," *Technical Report, Royal Institute of Technology*, 1997.

[10] R. Mehmood and J. Crowcroft, "Parallel iterative solution method for large sparse linear equation systems," University of Cambridge, Computer Laboratory, Tech. Rep., 2005.

[11] J. Lei and H.-F. Chen, "Distributed randomized pagerank algorithm based on stochastic approximation," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1641–1646, 2015.

[12] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[13] J. Wang and N. Elia, "Control approach to distributed optimization," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 557–561.

[14] J. Lei, H.-F. Chen, and H.-T. Fang, "Primal–dual algorithm for distributed constrained optimization," *Systems & Control Letters*, vol. 96, pp. 110–117, 2016.

[15] J. Lu and C. Y. Tang, "Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2348–2354, 2012.

[16] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2863–2878, 2015.

[17] G. Shi, B. D. Anderson, and U. Helmke, "Network flows that solve linear equations," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2659–2674, 2017.

[18] Y. Liu, C. Lageman, B. D. Anderson, and G. Shi, "Exponential least squares solvers for linear equations over networks," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2543–2548, 2017.

[19] Y. Liu, Y. Lou, B. D. O. Anderson, and G. Shi, "Network flows as least squares solvers for linear equations," in *Conference on Decision and Control, 2017*. IEEE, 2017, pp. 1046–1051.

[20] B. Anderson, S. Mou, A. S. Morse, and U. Helmke, "Decentralized gradient algorithm for solution of a linear equation," *arXiv preprint arXiv:1509.04538*, 2015.

[21] R. Tutunov, H. B. Ammar, and A. Jadbabaie, "A fast distributed solver for symmetric diagonally dominant linear equations," *arXiv preprint arXiv:1502.03158*, 2015.

[22] C. E. Lee, A. Ozdaglar, and D. Shah, "Solving systems of linear equations: Locally and asynchronously," *preprint arXiv*, vol. 1411, 2014.

[23] J. Von Neumann, "On rings of operators. reduction theory," *Annals of Mathematics*, pp. 401–485, 1949.

[24] G. Shi, K. H. Johansson, and Y. Hong, "Reaching an optimal consensus: Dynamical systems that compute intersections of convex sets," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 610–622, 2013.

[25] R. W. Brockett and D. Liberzon, "Quantized feedback stabilization of linear systems," *IEEE transactions on Automatic Control*, vol. 45, no. 7, pp. 1279–1289, 2000.

[26] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[27] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus on networks with quantized communication," *International Journal of Robust and Nonlinear Control*, vol. 19, no. 16, pp. 1787–1816, 2009.

[28] Z. Qiu, L. Xie, and Y. Hong, "Quantized leaderless and leader-following consensus of high-order multi-agent systems with limited data rate," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2432–2447, 2016.

[29] T. Li, M. Fu, L. Xie, and J.-F. Zhang, "Distributed consensus with limited communication data rate," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 279–292, 2011.

[30] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, 2005.

[31] P. Yi and Y. Hong, "Quantized subgradient algorithm and data-rate analysis for distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 380–392, 2014.

[32] J. Lei, P. Yi, G. Shi, and B. Anderson, "Data rates for network linear equations," *arXiv:1808.03203*, 2018.