Fast distributed smoothing for network clock synchronization

Nikolaos M. Freris and Anastasios Zouzias

***Abstract*— We consider the problem of estimating node variables from noisy relative measurements in a wireless network. In previous work, a distributed smoothing scheme was developed based on the Jacobi algorithm for obtaining least-squares (LS) estimates. Although the synchronous version of the algorithm was shown to converge exponentially to the optimal estimate, an analysis of the asynchronous scheme is notably missing.**

In this paper, we design and analyze a new class of distributed asynchronous smoothing algorithms based on Randomized Kaczmarz (RK) algorithm for solving linear systems. One of the proposed schemes applies RK directly to the (generally inconsistent) noisy linear system, whereas the other one operates on the normal equations for LS estimation. We analyze the expected convergence rate of the proposed algorithms depending solely on properties of the network topology. Inspired by the analytical insights, we propose a distributed algorithm for synchronization, namely *Randomized Kaczmarz Over-smoothing* (RKO) which has demonstrated significant improvement over the state-of-art protocol in numerous experiments, in terms of both convergence speedup and energy savings.

Keywords: Clock synchronization, Wireless Networks, Distributed algorithms, Smoothing, Sensor networks, Randomized algorithms, Best Linear Unbiased Estimators (BLUE).

I. Introduction

We are currently setting the stage for building very large networks of "intelligent" agents possessing sensing, communication, computation and control capabilities. In order for such systems to perform efficiently as an entity, a *decentralized* operation is indispensable. That is, each agent is responsible to make measurements and/or actuate in a predefined area, while it can communicate with others in its proximity via wireless. The typical paradigm is that of a *Wireless Sensor Network* (WSN) [1], with numerous applications ranging in the health, military, environmental and transportation industries. In the absence of centralized coordination, time synchronization among agents becomes crucial in order for any network-wide collaborative task be performed via decentralized operations.

Clocks generally do not agree. Cheap off-the-shelf motes are equipped with crystal oscillators whose frequency fluctuates due to changes in temperature, pressure or voltage [2], thus leading to time-varying clock offsets. In several applications the synchronization accuracy affects quality-of-service (QoS), performance or even safety: a) in networked control, where the stability of closed control loops can be compromised by asynchrony [3], b) in sensor networks, where applications such as tracking, surveillance, target localization [4], data fusion, and battery duty-cycling benefit from synchronization and c) in communication networks, where slotted protocols such as ALOHA, TDMA scheduling or random-access MAC [5], [6], [7] require accurate synchronization in order to avoid unnecessary resource wastage.

Centralized synchronization, typically via GPS, has high energy consumption and might even not be an option for several applications e.g., indoors, underwater, or in large downtown areas [8]. In such cases, it is important to deploy distributed clock synchronization protocols for wireless networks. In designing such protocols a multitude of factors needs to be considered among others: precision (typically in the order of microseconds), scalability, robustness to node and link failures, responsiveness to clock drifts as well as low communication and computational complexity.

We consider the problem of estimating a vector $\mathbf{x} \in \mathbb{R}^n$ based on noisy measurements of differences of its entries. In a network of n nodes, where the i-th node holds value x_i, the topology is modeled by an undirected graph $G = (V, E)$, in which two communicating nodes $i, j : (i, j) \in E$ exchange information to obtain a noisy measurement of the difference of their values $(x_j - x_i)$. The goal is to derive and analyze distributed asynchronous algorithms for the iterative smoothing [9] of the noisy relative measurements and study their convergence.

We note that the problem we study here extends far beyond the purview of clock synchronization. Smoothing algorithms can be used in any setup where two communicating agents can estimate their pairwise difference; applications are ubiquitous in networks sensing physical quantities such as temperature, pressure, as well as for distributed localization [10].

A. Main Results

The specific contributions that we outline in this paper are:

- We develop randomized schemes for asynchronous distributed smoothing in networks. The algorithms are inspired by Randomized Kaczmarz (RK) method [11], [12] for solving linear systems, and have exponential expected rate of convergence.
- We propose *Randomized Kaczmarz Smoothing* (RKS), applying RK to the linear equations corresponding to relative measurements. We study the rate of convergence in the absence of noise and deduce that it is comparable to the rate of convergence per update of the synchronous Jacobi smoothing [9], [13], whilst RKS has lower communication complexity. We also propose and analyze a variant (*Randomized Kaczmarz with Under-relaxation*

Nikolaos M. Freris is with IBM Research - Zürich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland. E-mail: nif@zurich.ibm.com

Anastasios Zouzias is with the Department of Computer Science at the University of Toronto, Canada. E-mail: zouzias@cs.toronto.edu. Part of this work was done while the author was at IBM Research - Zürich.

(RKU)) which guarantees convergence in the presence of noise.

- We propose and analyze an algorithm named *Randomized Kaczmarz Least Squares* (RKLS), which converges exponentially fast (in expectation) to the weightted least-squares estimate. The rate of convergence is significantly slower than RKS and synchronous Jacobi, so this algorithm is only applicable for the high-noise regime in which RKS will have oscillatory behavior.
- Our main proposal is a of RKS which we call *Randomized Kaczmarz Over-smoothing* (RKO). The algorithm performs a batch of RKS updates in one iteration and hence achieves substantive convergence speed-up over the state-of-art algorithm [14] in numerous experiments at the low-noise regime. This comes at the expense of increased communication complexity per iteration; however, RKO still outperforms all tested algorithms in accuracy normalized by the number of transmitted messages.

II. Related Literature

A comprehensive survey on clock synchronization protocols for sensor network applications can be found in [8]. We briefly recap here key theoretical advances in the area: A thorough analysis of the feasibility of clock synchronization was carried out in [2], where it was shown that clock synchronization is *impossible* for the case of unknown asymmetric link delays. Our work relates to the problem of obtaining global offset/skew estimates from noisy relative measurements between communicating nodes [9], [10], [15], [16], [13], [14], [17]. Karp et al. [15] derived the *Best Linear Unbiased Estimate* (BLUE), where "best" refers to minimum-variance, and made connections of the variance with *resistance* networks. This connection has been leveraged in [9], [13], [17] to establish sharp scaling laws on the variance for various classes of graphs: for connected random planar networks the mean asymptotic variance of the BLUE is $O(1)$ [13], as opposed to $O(\sqrt{n})$ for a tree with diameter $\Omega(n)$; this provides theoretical support for the scalability of *smoothing* algorithms that exploit all available relative measurements, against ad-hoc tree-based approaches. An asynchronous version of the Jacobi algorithm was developed independently in [18], [16]. Giridhar and Kumar [9], [13] obtained bounds on the convergence rate of a synchronous version of the Jacobi algorithm. Barooah et al. [14] proposed a variant featuring efficient initialization and use of information from the two-hop neighborhood which results in faster convergence in experiments. Finally, a clock model and a corresponding distributed protocol for joint estimation of skews, offsets and delays is presented in [19], [20].

This is the first approach, to the best of our knowledge, which gives rise to asynchronous algorithms for distributed smoothing with exponential convergence in expectation.

III. Preliminaries

We consider a network of n agents each equipped with a distinct clock, and denote the display of node i's clock at reference time t by $\tau_i(t)$. We call the relative instantaneous speedup of clock i with respect to the reference clock as its "*skew*," and denote it by $a_i(t)$. The "*offset*" of clock i at time t is defined as the difference of its display from reference time, $o_i(t) := \tau_i(t) - t$. For two clocks i, j we define the "*relative offset*" $o_{ij}(t) := \tau_j(t) - \tau_i(t)$, and "*relative skew*" $a_{ij}(t) := a_j(t)/a_i(t)$.

The communication network is modeled by an undirected graph $G = (V, E)$. We let $n := |V|$ be the number of agents and $m := |E|$ be the number of communication links. For simplicity, communication is taken symmetric; asymmetric communication can be treated as in [21]. Two neighboring nodes $i, j : (i, j) \in E$, can exchange time-stamped packets to obtain noisy estimates of their relative offset $\tau_j - \tau_i$, as well as of the logarithm of their relative skew $\log a_j(t) - \log a_i(t)$ [2]. We skip time dependency henceforth assuming that the available relative measurements correspond to the same physical time; the reader is referred to [20] for a treatment of how the error in making such approximation can be related to noise variance.

The problem under study amounts to estimating node variables from noisy relative measurements. Let us label the nodes as $1, \dots, n$ and write the undirected edge $e = (i, j)$ with $i < j$. Consider the unknown vector $\boldsymbol{x} \in \mathbb{R}^n$ of node variables, where variable x_i corresponds to node i. For any edge $(i, j) \in E$, a noisy measurement of the difference of the two node variables is available:

$$y_{ij} = x_j - x_i + w_{ij}, \tag{1}$$

where w_{ij} represents the measurement noise. The *incidence* matrix of the graph $\mathsf{A} \in \mathbb{R}^{m \times n}$ has entries:

$$a_{ek} := \begin{cases} -1, & \text{if } k = i; \\ 1, & \text{if } k = j; \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

if the e-th edge corresponds to (i, j). Stacking the noise components and measurement values into vectors $\boldsymbol{y}, \boldsymbol{w} \in \mathbb{R}^m$, respectively, we get

$$\boldsymbol{y} = \mathsf{A}\mathbf{x} + \mathbf{w}. \tag{3}$$

Recall that this problem is generic and extends well beyond clock synchronization. In fact, it is straightforward to extend our formulation to account for vector-valued node variables as in [14]. For clock synchronization, x_i can represent either the logarithm of clock i's skew $\log a_i$ or clock i's offset o_i.

A. Least-squares estimation

In the absence of noise, the linear system $\boldsymbol{y} = A\mathbf{x}$ is *consistent*, i.e., has a solution. The rank of A is equal to $n - n_c$, where n_c is the number of connected components of the graph [2], in particular, the system has multiple solutions. A necessary and sufficient condition for uniqueness of solution is that the value of one node variable per connected component be set to a known value; equivalently, the network is *synchronizable* if and only if there exists at least one reference node per connected component [2], [14]. Therefore, we assume without loss of generality that the network is connected whence **rank** $(\mathsf{A}) = n-1$; a unique solution exists if *any* node is picked as reference, e.g., by setting its variable equal to 0.

Due the presence of noise, the linear system is generally inconsistent. The natural problem to address is that of least-squares (LS) estimation, where the estimator $\hat{\mathbf{x}}$ is picked as solution to the optimization problem[1]:

$$\hat{\boldsymbol{x}} \in \arg\min_{\mathbf{x}} (\boldsymbol{y} - \mathsf{A}\mathbf{x})^\top (\boldsymbol{y} - \mathsf{A}\mathbf{x}), \tag{4}$$

which is equivalent to the normal equations [14], [22]:

$$\mathsf{A}^\top \mathsf{A} \hat{\boldsymbol{x}} = \mathsf{A}^\top \boldsymbol{y}, \tag{5}$$

The minimum-norm solution is given by $\hat{\boldsymbol{x}}^* = \mathsf{A}^\dagger \boldsymbol{y}$, where $A^\dagger$ is the *pseudo-inverse* of A [22], and any solution is of the form $\hat{\boldsymbol{x}}^* + \alpha\mathbf{1}$, for some α, where $\mathbf{1}$ is the all-one vector. This indeterminacy is extinguished once we set a node r as reference, i.e., $x_r = 0$, which is equivalent to removing the r-th row of A to obtain $\bar{\mathsf{A}}$; the unique solution is given by $\bar{\mathbf{x}}^* = \bar{\mathsf{A}}^\dagger \boldsymbol{y}$, where in this case $\bar{\mathsf{A}}^\dagger = (\bar{\mathsf{A}}^\top \bar{\mathsf{A}})^{-1} \bar{\mathsf{A}}^\top$.

B. Basics from graph theory

For each $i \in V$, we define its *neighborhood*, $N_i := \{j \in V : (i,j) \in E\}$. The *degree* of the node is $d_i := |N_i|$, and we define $d_{\max} := \max_i d_i$. Let $\mathsf{L} := \mathsf{A}^\top \mathsf{A}$ be the (un-normalized) *Laplacian* of the graph with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of L. For a connected graph we have that $\lambda_1 = 0$, and $0 < \lambda_i \leq d_{\max}$, for $i = 2, \ldots, n$. The second smallest eigenvalue of L denoted by $\lambda_2(G)$, also called the *Fiedler value* or *algebraic connectivity* of G, can be lower-bounded via Cheeger's inequality [23], [24] as follows: define for each $S \subseteq V$, the *volume* to be the sum of the degrees of the vertices in S: $\text{vol}(S) := \sum_{i \in S} d_i$; furthermore, let $E(S, \bar{S})$ be the set of edges with one vertex in S and the other one in $V \setminus \bar{S}$; finally, let $h_G(S) := \frac{|E(S,\bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$. The *Cheeger constant* of G is defined as $h_G := \min_S h_G(S)$. Then:

$$\lambda_2(G) \geq \frac{h_G^2}{2d_{\max}}. \tag{6}$$

In the case when the r-th node is picked as reference, let us denote the corresponding reduced Laplacian by $\bar{\mathsf{L}} := \bar{\mathsf{A}}^\top \bar{\mathsf{A}}$, which is positive definite. Let us denote the smallest eigenvalue of $\bar{\mathsf{L}}$ by $\bar{\lambda}(G)$.

We will show that the rate of convergence of the presented smoothing algorithms depends on $\lambda_2(G)$ in the case that no node is taken as reference, and depends on $\bar{\lambda}(G)$ otherwise.

IV. Algorithms

We propose new algorithms for distributed asynchronous smoothing of relative measurements and analyze their convergence. The algorithms are inspired by a randomized version of Kaczmarz method [12], [25] applied to both the (generally) inconsistent linear system (3) and to the normal equations (5). In what follows, a node i is assumed to always have access to all relative measurements[2] $\{y_{ij}\}_{j \in N_i}$. We show that the proposed algorithms have exponential convergence in expectation, and are amenable to distributed asynchronous implementation. We compare and contrast the performance in terms of two metrics related to attaining a given error reduction: a) the number of iterations and b) the cumulative communication cost (in terms of transmitted-received messages).

A. Distributed Smoothing based on Jacobi algorithm [16], [18].

The *Spatial Smoothing* (SS) algorithm proposed in [16], [18] is an asynchronous implementation of Jacobi iterations applied to the reduced normal equations $\bar{\mathsf{A}}^\top \bar{\mathsf{A}} \boldsymbol{x} = \bar{\mathsf{A}}^\top \boldsymbol{y}$, obtained after a node is selected as reference. In this scheme, each node $i \neq r$ initially sets its nodal estimate to an arbitrary value $\hat{\mathbf{x}}_i^{(0)}$, while $\hat{\mathbf{x}}_r^{(k)} = x_r = 0$ for all k. In an asynchronous operation, a node $i \neq r$ is activated at arbitrary times independently from others and updates its estimate: $\hat{\mathbf{x}}_i^{\text{new}} \leftarrow \frac{1}{d_i} \sum_{j \in N_i} (\hat{\mathbf{x}}_j^{\text{old}} + y_{ji})$ [9], [16], [18]. This scheme is distributed because a node makes an update of its local estimate using information corresponding to its immediate neighbors. Alhough this asynchronous scheme is known to converge (because it is essentially an asynchronous *coordinate descent* applied to the quadratic optimization problem (4) with $\bar{\mathsf{A}}$ in the place of A [13], [26]), there is no analysis on its rate of convergence.

The *synchronous* version of spatial smoothing, in which all nodes simultaneously update their estimates using the above rule, is an instance of the Jacobi algorithm applied to the reduced normal equations. In this case, the Jacobi iterative operator has spectral radius strictly less than one, so the algorithm converges exponentially to the LS estimate [16], [18]. Let us denote the vector of node estimates after the k-th synchronous update by $\hat{\boldsymbol{x}}^{(k)}$. The analysis of the rate of convergence was carried out in [9], [13]:

Theorem 1: For all $k \geq \frac{4m^2}{\beta^2} \ln(\left\|\hat{\boldsymbol{x}}^{(0)} - \bar{\boldsymbol{x}}^*\right\|_2 / \varepsilon)$, it holds that $\left\|\hat{\boldsymbol{x}}^{(k)} - \bar{\boldsymbol{x}}^*\right\|_2 \leq \varepsilon$; β denotes the edge-connectivity of G, i.e., the minimum number of edges that, when deleted, disconnect the graph.

B. Distributed Smoothing based on Randomized Kaczmarz algorithm

We first recap the Randomized version of Kaczmarz (RK) algorithm proposed in [11], [12], and proceed to use RK to derive algorithms for distributed asynchronous smoothing. The first algorithm is an application of RK directly to the noise-free version of the linear system (3) ($\mathbf{w} = \mathbf{0}$), while the second is an application of RK to the normal equations $\mathsf{A}^\top \mathsf{A} \boldsymbol{x} = \mathsf{A}^\top \boldsymbol{y}$. We further study a variant with provable convergence in the presence of noisy measurements, and ultimately propose a hybrid scheme with superior convergence rate.

[1] The extension to the general case of weighted least-squares and Best Linear Unbiased Estimation (BLUE) is straightforward [18], [14].

[2] The variable y_{ij} has been defined for edge (i,j) with the convention that $i < j$; we extend this by defining $y_{ji} = -y_{ij}$, for each $(i,j) \in E : i < j$.

a) Randomized Kaczmarz algorithm: Consider the system of linear equations:

$$\mathsf{A}\boldsymbol{x} = \boldsymbol{y}, \tag{7}$$

where A is an $m \times n$ matrix with $m \geq n$, and $\boldsymbol{y} \in \mathbb{R}^m$ is a given vector. We denote the rows of A by $\mathbf{a}_1, \ldots, \mathbf{a}_m$ and let $\boldsymbol{y} = (y_1, \ldots, y_m)^\top$. For any matrix A we define

$$\kappa(\mathsf{A}) := \frac{\|\mathsf{A}\|_{\mathrm{F}}^2}{\sigma_{\min}(\mathsf{A})^2}, \tag{8}$$

where $\sigma_{\min}(\mathsf{A})$ is the smallest non-zero singular value of A and $\|\mathsf{A}\|_{\mathrm{F}}^2 = \sum a_{ij}^2$. Finally, let $[m] := \{1, \ldots, m\}$, and let $\langle \cdot,\ \cdot \rangle$ denote the standard inner product in L_2. Strohmer and Vershynin proposed the following randomized variant of Kaczmarz algorithm [12].

Algorithm 1 Randomized Kaczmarz (RK)

```
1: procedure (A, y)                                  ▷ $\mathsf{A} \in \mathbb{R}^{m \times n}, \boldsymbol{y} \in \mathbb{R}^m$
2:     Initialize estimates $\hat{\boldsymbol{x}}^{(0)}$
3:     for each $k \in \mathbb{Z}_+$ do
4:         Pick $i_k \in [m]$ w.p. $p_i := \frac{\|\mathbf{a}_i\|_2^2}{\|\mathsf{A}\|_{\mathrm{F}}^2}, i \in [m]$.
5:         Set $\hat{\boldsymbol{x}}^{(k)} = \hat{\boldsymbol{x}}^{(k-1)} + \frac{y_{i_k} - \langle \mathbf{a}_{i_k},\ \hat{\boldsymbol{x}}^{(k-1)} \rangle}{\|\mathbf{a}_{i_k}\|_2^2} \mathbf{a}_{i_k}$.
6:     end for
7: end procedure
```

The following is a generalization of the main result of [12] without the restrictive assumption of **rank** $(\mathsf{A}) = n$. The proof can be found in the appendix.

Theorem 2: Assume that (7) has a solution and denote $\hat{\boldsymbol{x}}^* := \mathsf{A}^\dagger \boldsymbol{y}$. If $\hat{\boldsymbol{x}}^{(0)} \in \mathcal{R}(\mathsf{A}^\top)$, where $\mathcal{R}$ denotes the range space , then Algorithm 1 converges to $\hat{\boldsymbol{x}}^*$ in the mean square, more precisely:

$$\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^*\right\|_2^2 \leq \left(1 - \frac{1}{\kappa(\mathsf{A})}\right)^k \left\|\hat{\boldsymbol{x}}^{(0)} - \hat{\boldsymbol{x}}^*\right\|_2^2. \tag{9}$$

b) RK for $\mathsf{A}\boldsymbol{x} = \boldsymbol{y}$: We propose a randomized distributed smoothing algorithm (Algorithm 2), by using RK to the system (7) pretending that it is consistent (e.g., assuming $\boldsymbol{w} = \mathbf{0}$), and analyze its convergence in the noiseless setting.

Consider a given row of A, say $e \equiv (i, j)$. It has two non-zero entries, so $\|\mathbf{a}_e\|_2^2 = 2$, and for any $\boldsymbol{x} \in \mathbb{R}^n$ we have $\langle \mathbf{a}_e,\ \boldsymbol{x} \rangle = x_j - x_i$. Step 5 of Algorithm 1 is translated to

$$\hat{\boldsymbol{x}}^{(k)} \leftarrow \hat{\boldsymbol{x}}^{(k-1)} + \frac{y_{ij} + \hat{x}_i^{(k-1)} - \hat{x}_j^{(k-1)}}{2} \mathbf{a}_e^\top.$$

In particular, if e is selected the only estimates that are updated are those for x_i, x_j. The algorithm is clearly distributed, in the sense that at a given iteration only two neighboring nodes make updates of their local estimates based solely on exchanging their previous estimates. Note that the above algorithm does not assume a reference node is given; in the case of noise-free measurements, it will converge to the LS solution $\mathbf{x} = \bar{\mathsf{A}}^\dagger \boldsymbol{y}$, which corresponds to node offsets (skews) from a fictitious reference time. The algorithm has expected exponential convergence as it follows from Theorem 2:

Algorithm 2 Randomized Kaczmarz Smoothing (RKS)

```
procedure
    for all nodes $i \in V$ do                       ▷ Initialization step
        Set $\hat{x}_i^{(0)} = 0$ and detect neighbors $N_i$
        Obtain relative measurements $y_{ij}$ for all $j \in N_i$
    end for
    for each $k \in \mathbb{Z}_+$ do
        Pick uniformly at random an edge $e = (i, j)$
        Nodes $i, j$ exchange $\hat{x}_i^{(k-1)}, \hat{x}_j^{(k-1)}$.
        Node $i$ sets: $\hat{x}_i^{(k)} \leftarrow \frac{1}{2}(\hat{x}_i^{(k-1)} + \hat{x}_j^{(k-1)} - y_{ij})$
        Node $j$ sets: $\hat{x}_i^{(k)} \leftarrow \frac{1}{2}(\hat{x}_i^{(k-1)} + \hat{x}_j^{(k-1)} + y_{ij})$
    end for
end procedure
```

Corollary 1 (Convergence rate of Algorithm 2): For given $\varepsilon > 0$, the estimates produced by RKS 2 satisfy:

$$\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^*\right\|_2 \leq \varepsilon, \quad \forall\ k \geq \frac{4m}{\lambda_2(G)} \ln(\|\hat{\boldsymbol{x}}^*\|_2 / \varepsilon).$$

In particular, for any $\delta > 0$ and $k \geq \frac{4m}{\lambda_2(G)} \ln(\frac{\|\hat{\boldsymbol{x}}^*\|_2}{\varepsilon\delta})$, we have that $\left\|\hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^*\right\|_2 \leq \varepsilon$ with probability at least $1 - \delta$.

Proof: Since **rank** $(\mathsf{A}) = n - 1$, the solution space of $\mathsf{A}\boldsymbol{x} = \boldsymbol{y}$ is the one-dimensional affine space $\{\hat{\boldsymbol{x}}^* + \alpha\mathbf{1} \mid \alpha \in \mathbb{R}\}$ where $\hat{\boldsymbol{x}}^* = \mathsf{A}^\dagger \boldsymbol{b}$; this algorithm is a special instance of Algorithm 1, and $\hat{\boldsymbol{x}}^{(0)} \in \mathcal{R}(\mathsf{A}^\top)$. The result follows from Theorem 2 (in this case $\|\mathsf{A}\|_{\mathrm{F}}^2 = 2m$ and $\sigma_{\min}(\mathsf{A})^2 = \lambda_2(G)$) and the fact that $1 - x \leq \exp(-x)$, for any $x \in [0, 1]$. The second part follows immediately from Markov's inequality. ■

Remark 1 (Picking a node as reference): An alternative is to consider a given node r as reference, remove the corresponding column from A and apply RK to the system $\bar{\mathsf{A}}\bar{\boldsymbol{x}} = \boldsymbol{y}$. In this case the only changes are that $\hat{x}_r^{(k)} = 0$ for all k and that all edges incident to the reference node are picked with probability $p = \frac{1}{2m - d_r}$ while those non-incident to node r are picked with probability $2p$. The expected convergence is still exponential, with exponent $\frac{\bar{\lambda}(G)}{2m - d_r}$. An efficient way of implementing the randomized selection is demonstrated in Sec. IV-C.

c) RK for Normal Equations: We propose a second randomized smoothing algorithm obtained by applying RK to the normal equations $\mathsf{A}^\top\mathsf{A}\boldsymbol{x} = \mathsf{A}^\top\boldsymbol{y}$, and show that it admits a distributed implementation.

The (un-normalized) Laplacian of the graph $\mathsf{L} \in \mathbb{R}^{n \times n}$ has entries

$$l_{ij} := \begin{cases} d_i, & \text{if } i = j; \\ -1, & \text{if } (i, j) \in E; \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The i-th row of the Laplacian, L_i, has squared Euclidean norm $d_i^2 + d_i$. Moreover, for any $\boldsymbol{x} \in \mathbb{R}^n$, $\langle \mathsf{L}_i,\ \boldsymbol{x} \rangle = d_i x_i - \sum_{j \in N_i} x_j$ while the i-th entry of $\mathsf{A}^\top\boldsymbol{y}$ is equal to $-\sum_{j \in N_i} y_{ij}$. Step 5 of Algorithm 1 applied to $\mathsf{L}\boldsymbol{x} = \mathsf{A}^\top\boldsymbol{y}$ is translated to:

$$\hat{\boldsymbol{x}}^{(k)} \leftarrow \hat{\boldsymbol{x}}^{(k-1)} + \frac{-\sum_{j \in N_i} y_{ij} - (d_i \hat{x}_i^{(k-1)} - \sum_{j \in N_i} \hat{x}_j^{(k-1)})}{d_i^2 + d_i} \mathsf{L}_i^\top.$$

Re-arranging terms and simplifying the above expression we get

$$\hat{\boldsymbol{x}}^{(k)} \leftarrow \hat{\boldsymbol{x}}^{(k-1)} - \frac{\hat{x}_i^{(k-1)} - \frac{1}{d_i}\sum_{j\in N_i}(\hat{x}_j^{(k-1)} + y_{ji})}{1+d_i}\mathsf{L}_i^\top,$$

which leads to the following algorithm:

Algorithm 3 Randomized Kaczmarz Least Squares (RKLS)

```
1: **procedure**
2:     Initialize as in Algorithm 2
3:     **for** each $k \in \mathbb{Z}_+$ **do**
4:         Pick a node $i \in V$ w.p. $p_i = \frac{d_i^2+d_i}{\sum_i(d_i^2+d_i)}$
5:         Compute $\Delta_i := \frac{\hat{x}_i^{(k-1)} - \frac{1}{d_i}\sum_{j\in N_i}(\hat{x}_j^{(k-1)}+y_{ji})}{1+d_i}$ and
           set $\hat{x}_i^{(k)} \leftarrow \frac{1}{1+d_i}(\hat{x}_i^{(k-1)} + \sum_{j\in N_i}(\hat{x}_j^{(k-1)} + y_{ji}))$
6:         Every node $j \in N_i$, sets: $\hat{x}_j^{(k)} \leftarrow \hat{x}_j^{(k-1)} + \Delta_i$
7:     **end for**
8: **end procedure**
```

The following corollary studies the rate of convergence.

Corollary 2: For given $\varepsilon > 0$, the estimates produced by RKLS satisfy:

$$\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^*\right\|_2 \le \varepsilon, \quad \forall\ k \ge \frac{2\sum_i d_i^2 + 4m}{\lambda_2^2(G)}\ln(\|\hat{\boldsymbol{x}}^*\|_2/\varepsilon).$$

In particular, for any $\delta > 0$ and $k \ge \frac{2\sum_i d_i^2+4m}{\lambda_2^2(G)}\ln(\frac{\|\hat{\boldsymbol{x}}^*\|_2}{\varepsilon\delta})$, Algorithm 3 computes $\hat{\boldsymbol{x}}^{(k)}$ so that, with probability at least $1-\delta$, $\left\|\hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^*\right\|_2 \le \varepsilon$.

Proof: Note that in this case $\|\mathsf{L}\|_{\mathrm{F}}^2 = \sum_{i\in V}(d_i + d_i^2) = \sum_{i\in V} d_i^2 + 2m$, $\sigma_{\min}(\mathsf{L}) = \lambda_2^2(G)$, and the rest of the proof follows along the same lines as in Cor. 2. ■

Remark 2: Again, one can pick a node as reference and consider RK applied to $\bar{\mathsf{L}}\bar{\boldsymbol{x}} = \bar{\mathsf{A}}^\top\boldsymbol{y}$. The distribution of selecting a node in step 4 changes to $p_i = \frac{d_i^2+d_i}{\sum_i d_i^2+2m-d_r}$ for $i \notin N_r$ and $p_i = \frac{d_i^2+d_i-1}{\sum_i d_i^2+2m-d_r}$; the convergence is exponential with rate $\frac{\bar{\lambda}^2(G)}{\sum_i d_i^2+2m-d_r}$. An efficient way of implementing the randomized selection is demonstrated in Sec. IV-C.

d) The inconsistent case: The RK algorithm was shown to converge to a solution of $\mathsf{A}\boldsymbol{x} = \boldsymbol{y}$ only for the case that such a solution exists. In the case of an inconsistent system, it is plain to check that the algorithm does not converge, and will have oscillatory behavior.

In the case of an inconsistent system we propose using an *under-relaxation* of RK, i.e., substitute the update in Step 4 of Algorithm 1 with

$$\hat{\boldsymbol{x}}^{(k)} \leftarrow \hat{\boldsymbol{x}}^{(k-1)} + \gamma_k\frac{y_{i_k} - \left\langle \mathbf{a}_{i_k},\ \hat{\boldsymbol{x}}^{(k-1)}\right\rangle}{\|\mathbf{a}_{i_k}\|_2^2}\mathbf{a}_{i_k}^\top, \tag{11}$$

where $\{\gamma_k\}$ is a sequence of receding step-sizes with $\gamma_k \in [0,1]$ and $\lim_k \gamma_k = 0$. The following theorem studies the convergence of this variant; the proof can be found in the appendix.

Theorem 3 (RK with Under-relaxation (RKU)): Let $\mathsf{A} \in \mathbb{R}^{m\times n}$ with $\textbf{rank}\,(\mathsf{A}) = n$, and assume that the system $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$ has a unique solution denoted by $\boldsymbol{x}$. Consider the Randomized Kaczmarz algorithm with under-relaxation parameters $\{\gamma_k\} \subset [0,1]$ applied to the linear system $\mathsf{A}\boldsymbol{x} = \boldsymbol{y}$ with $\boldsymbol{y} := \boldsymbol{b} + \boldsymbol{w}$. We have

$$\mathbb{E}||\hat{\boldsymbol{x}}^{(k)} - \boldsymbol{x}||_2^2 \le \big(1 - \frac{\gamma_k}{\kappa(\mathsf{A})}\big)\mathbb{E}||\hat{\boldsymbol{x}}^{(k-1)} - \boldsymbol{x}||_2^2 + \gamma_k W, \tag{12}$$

where $W := \frac{1}{\|\mathsf{A}\|_{\mathrm{F}}^2}\sum_i |w_i|^2$.

Corollary 3 (Convergence-Accuracy trade-off): For each $k \in \mathbb{Z}_+$:

$$\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k)} - \boldsymbol{x}\right\|_2^2 \le \exp\left(-\frac{\sum_{i=1}^k \gamma_i}{\kappa(\mathsf{A})}\right)\left\|\hat{\boldsymbol{x}}^{(0)} - \boldsymbol{x}\right\|_2^2 + W\sum_{i=1}^k \gamma_i \exp\left(-\frac{\sum_{j=i+1}^k \gamma_j}{\kappa(\mathsf{A})}\right). \tag{13}$$

In particular, if $\lim_{k\to+\infty}\gamma_k = 0$ then $\hat{\boldsymbol{x}}^{(k)}$ converges in the mean square.

Proof: Using (12) recursively we have that

$$\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k)} - \boldsymbol{x}\right\|_2^2 \le \prod_{i=1}^k\left(1 - \frac{\gamma_i}{\kappa(\mathsf{A})}\right)\left\|\hat{\boldsymbol{x}}^{(0)} - \boldsymbol{x}\right\|_2^2 + W\sum_{i=1}^k \gamma_i \prod_{j=i+1}^k\left(1 - \frac{\gamma_j}{\kappa(\mathsf{A})}\right). \tag{14}$$

Then (13) follows by the inequality $1 - x \le \exp(-x)$ for $x \le 1$. In addition, it follows directly from (12) that

$$\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k)} - \boldsymbol{x}\right\|_2^2 \le \max\{(1 - \frac{1}{\kappa(\mathsf{A})})\,\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k-1)} - \boldsymbol{x}\right\|_2^2, W\},$$

whence in particular $\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k)}\right\|_2^2$ is uniformly bounded, therefore it immediately follows from (11) that $\hat{\boldsymbol{x}}^{(k)}$ is Cauchy in L_2 and therefore has a mean square limit. ■

Remark 3 (Discussion): The analysis above extends the result of [27] which is recovered for $\gamma_k = 1$. Note also that our analysis easily extends to the case that $\textbf{rank}\,(\mathsf{A}) < n$ if we consider $\hat{\boldsymbol{x}}^{(0)} \in \mathcal{R}(\mathsf{A}^\top)$ (e.g., if $\hat{\boldsymbol{x}}^{(0)} = \mathbf{0}$) as in Theorem 2. Furthermore, note that if we let $\gamma_k = \gamma \in (0,1]$, then the updates in (11) do not converge with probability 1; this is because, by construction, each row will almost surely be selected infinitely often, and a necessary condition for convergence is that $\mathsf{A}\boldsymbol{x} = \boldsymbol{y}$, which is not true. From Cor. 3 it follows that $\hat{\boldsymbol{x}}^{(k)}$ is uniformly bounded in L_2 therefore the estimates converge to some compact set in the mean square sense. For each $\gamma \in (0,1]$, it follows from (14) via simple calculations that:

$$\limsup_k \left\|\hat{\boldsymbol{x}}^{(k)} - \boldsymbol{x}\right\|_2^2 \le \kappa(\mathsf{A})W.$$

In particular, the upper bound on the worst-case expected asymptotic error is independent of γ; therefore while reducing γ decreases the convergence rate, it does not help reduce the remaining uncertainty.

e) RKO: A hybrid algorithm: We propose a natural variant of RKS, which we call *Randomized Kaczmarz Over-smoothing* (RKO). In Algorithm 2, an edge is selected at random in Step 7 and an update is made concerning the estimates of the two vertices incident to that edge. The analysis in Theorem 2 shows that each such step decreases the estimation error in expectation for the case of a consistent system; therefore having multiple such updates results in a larger decrease. In RKO, we perform such *over-smoothing*, in the sense that a node is selected at random, and the updates of Steps 9 and 10 of Algorithm 2 are performed to all the edges incident to it. As we will see in Section VI, the convergence rate of Algorithm 4 is faster compared to all other methods tested.

Algorithm 4 Randomized Kaczmarz Over-smoothing (RKO)

```
procedure
    Initialize as in Algorithm 2
    for each $k \in \mathbb{Z}_+$ do
        Pick a node $i \in V$ w.p. $p_i = \frac{d_i}{2m}$
        For every node $j \in N_i$, nodes $i$ and $j$ perform Steps 8,9 and 10 of Algorithm 2.
    end for
end procedure
```

C. Implementation issues

All proposed algorithms are randomized, in the sense that at each iteration a node (edge) is sampled from V (E) based on some distribution, and independently from nodes (edges) sampled in the past. We will show that such mechanism can be implemented efficiently in a distributed manner in a wireless network, by exploiting the fact that nodes are equipped with clocks. That is, all proposed algorithms can be implemented using the framework described in [28]. Let us consider a probability distribution over the nodes V, $\boldsymbol{p} := \{p_1, \ldots, p_n\}$, and assume that each node i has access to θp_i for some common $\theta > 0$. Each node i is sampled at a rate[3] $\Lambda_i = \theta p_i \Lambda$, where $\Lambda > 0$ is a predetermined constant. The sampling is performed by activating a node when its local count-down timer reaches zero; waiting times are independent exponentially distributed random variables with rate Λ_i. We can implement edge sampling via node sampling, because sampling edges according to $\tilde{\boldsymbol{p}} := \{\tilde{p}_1, \ldots, \tilde{p}_m\}$, is equivalent to sampling a node i with probability $p_i = \sum_{j \in N_i} \tilde{p}_{ij}$, and then select edge (i, j) with probability $\tilde{p}_{ij}/p_i$; the implicit assumption is that both nodes i, j know $\theta\tilde{p}_{ij}$ for some $\theta > 0$. In the case of all Algorithms 2, 3 and 4, each node is aware of the numerator in the definition of the corresponding sampling probability.

V. Comparison with previous work

The convergence speed of RKS depends inverse proportionally on $\lambda_2(G)$, which is further lower-bounded via Cheeger's inequality (6). While the spatial smoothing (SS) [9], [16], [18] is provably convergent to the least-squares solution, nothing, to the best of our knowledge, is known about rate of convergence. The synchronous Jacobi algorithm was analyzed in [9], [13]; the provided bound on convergence rate depends on the edge-connectivity (a.k.a. min-cut) of the graph β and the total number of edges. Typically, in a random planar wireless network we expect β to be roughly $O(1)$ [13]; this implies that the synchronous Jacobi smoothing algorithm requires $O(n^2)$ synchronous rounds, in each of which the estimates of every node are updated. In contrast, each iteration of RKS (Alg. 2) or RKLS (Alg. 3) updates the estimates between a pair of adjacent nodes or the estimates within an 1-hop neighborhood, respectively. For a fair comparison, the rate of convergence of the first two rows of Table I has been raised to the n-th power.

TABLE I

Comparison with Prior Results in Clock Synchronization

Algorithm	Rate	Asynchronous?
RKS	$\exp(-\frac{nh_G^2}{4md_{\max}})$	Yes
RKLS	$\exp(-\frac{nh_G^4}{8d_{\max}^2 \sum_i d_i^2})$	Yes
Jacobi	$\exp(-\frac{\beta^2}{4m^2})$	No
SS [16]	Converges	Yes
OSE [14]	Converges	Yes

VI. Simulations

We compare the performance of the various algorithms for a standard network model used extensively in the literature: in a square of unit area, we deploy $n = 200$ agents independently and uniformly at random in the plane. We add an edge between two nodes if the lie within a fixed transmission range, which picked so as to guarantee asymptotic connectivity [29]. Next, we pick the clock offsets uniformly distributed in the interval $[0, 100]$. We consider relative measurements across the edges with an added Gaussian noise vector with variance $\sigma^2 = 1$, and independent entries.

We have implemented the following algorithms: synchronous Jacobi (J) and asynchronous Jacobi (spatial smoothing) (SS) as described in [9]; the overlapping subgraph estimator algorithm (OSE) [14]; the classical[4] Kaczmarz algorithm [25] applied on (3) (KS) and applied on (5) (KLS), Algorithm 2 (RKS), Algorithm 3 (RKLS) and Algorithm 4 (RKO). All asynchronous algorithms but ours are executed by selecting a node uniformly at random in each iteration. All randomized algorithms were executed independently fives times and average values are presented. We initialize by selecting node 1 as reference, and selecting all initial node estimates to be zero. As a measure of the attained accuracy, we use the normalized (per node) Root-Mean-Squared-Error (RMSE) and we count as a single

[3] To be quite precise this mechanism requires that all clocks run at the same speed, i.e., that skews be compensated. In practice, this is not an issue since skew synchronization can be very accurate [20] and the error in the sampling distribution due to considering local counters is negligible.

[4] The classical Kaczmarz algorithm at every iteration performs updates as in Step 5 of Algorithm 2 for *every* row and the updates are performed in an a priori fixed ordering of the rows.

message the exchange of one node estimate between two adjacent nodes.

We plot the results for the synchronous algorithms (J, KS, KLS) and the asynchronous ones (SS, OSE, RKS, RKLS, RKO) separately, see Fig. 1(a) and Fig. 1(b), respectively. We observe that Kaczmarz algorithm demonstrates faster convergence compared to Jacobi.

In the asynchronous setting, each algorithm transmits a different number of messages per iteration. For this reason, in addition to the plot of RMSE per iteration we also plot RMSE as function of the number of transmitted messages; this is an index for the critical energy consumption in a sensor network [14]. Our experiments showcase that: RKLS has significantly slower convergence, compared to all other schemes; this should be expected since the convergence exponent is roughly the square of RK and SS (cf. Thm. 1, Cors. 1, 2). In the left plot of Fig. 1(b) we observe that RKS and SS have roughly the same speed of convergence in terms of *number of iterations*; OSE is faster (per iteration) compared to both of them, and RKO seems to be superior compared to all. In the right sub-figure of Fig. 1(b) we see that RKS and RKO are the superior schemes compared to all others in term of number of speed of convergence per number of transmitted messages. We deduce that our proposal, RKO, can achieve synchronization faster in less time, with parallel energy savings for a target synchronization accuracy.

VII. Conclusions and Future Work

We have used the Randomized Kaczmarz (RK) algorithm to design randomized smoothing methods with exponential expected rate of convergence, and have demonstrated an efficient means for distributed asynchronous implementation. We have proposed a scheme named *Randomized Kaczmarz Smoothing* (RKS) that runs on the noisy linear system of equations; the algorithm achieves a rate of convergence that is comparable to the Jacobi algorithm in both theory and experiments, with the benefit of reduced communication complexity. For highly noisy measurements, RKS demonstrates oscillatory behavior, so we have developed a variant which is guaranteed to converge, but at a slower rate. Finally, we have performed numerous experiments with various network topologies to illustrate the benefits of using our ultimate proposal, *Randomized Kaczmarz Over-smoothing* (RKO), in terms of both error vs. time and error vs. number of sent messages plots.

For future work, it looks intriguing to analytically establish exponential convergence of a randomized variant of the asynchronous Jacobi algorithm, as dictated by experimental evidence. Moreover, a formal treatment of the RKO algorithm as well as an analytical categorization of the performance of the introduced algorithms in large networks are both beyond the scope of the current paper due to space limitations, and will be addressed elsewhere. On the implementation front, we plan to test the impact of efficient initialization [14] in convergence speed, as well perform more realistic trace-driven simulations in which RKO has been modified to reduce communication overhead.

Appendix

Proof: (of Theorem 2)

First, note that for any $k \in \mathbb{Z}_+$, we have that $(\hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^*) \perp (\hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^{(k-1)})$. To see this, let $\alpha_i := \frac{y_i - \left\langle \mathbf{a}_i,\ \hat{\boldsymbol{x}}^{(k-1)} \right\rangle}{\|\mathbf{a}_i\|_2^2}$ for every $i \in [m]$. Then $\hat{\boldsymbol{x}}^{(k)} = \hat{\boldsymbol{x}}^{(k-1)} + \alpha_{i_k} \mathbf{a}_{i_k}$, whence:

$$\left\langle \hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^*,\ \hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^{(k-1)} \right\rangle = \alpha_{i_k} (\left\langle \hat{\boldsymbol{x}}^{(k)},\ \mathbf{a}_{i_k} \right\rangle - y_{i_k})$$

using the fact that $\langle \hat{\boldsymbol{x}}^*,\ \mathbf{a}_j \rangle = y_j$, since $\hat{\boldsymbol{x}}^*$ is a solution to $\mathsf{A}\boldsymbol{x} = \boldsymbol{y}$. The calculation $\left\langle \hat{\boldsymbol{x}}^{(k)},\ \mathbf{a}_{i_k} \right\rangle = \left\langle \hat{\boldsymbol{x}}^{(k-1)},\ \mathbf{a}_{i_k} \right\rangle + \alpha_{i_k} \|\mathbf{a}_{i_k}\|_2^2 = y_{i_k}$ proves orthogonality. In particular:

$$\left\| \hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^* \right\|_2^2 = \left\| \hat{\boldsymbol{x}}^{(k-1)} - \boldsymbol{x}^* \right\|_2^2 - \left\| \hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^{(k-1)} \right\|_2^2, \tag{15}$$

so the error does not increase at each iteration. It suffices to bound the expected error reduction at each iteration; more specifically we will show that if the index of selected row is a random variable on $[m]$ with distribution $\mathbb{P}(Z = i) = \frac{\|\mathbf{a}_i\|_2^2}{\|\mathsf{A}\|_{\mathrm{F}}^2}$ then

$$\mathbb{E}_Z \left\| \hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^{(k-1)} \right\|_2^2 \geq \frac{1}{\kappa(\mathsf{A})} \left\| \hat{\boldsymbol{x}}^{(k-1)} - \hat{\boldsymbol{x}}^* \right\|_2^2, \tag{16}$$

from which the main result follows immediately.

$$\begin{aligned} \mathbb{E}_Z \left\| \hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}}^{(k-1)} \right\|_2^2 &= \mathbb{E}_Z \frac{\left\langle \hat{\boldsymbol{x}}^* - \hat{\boldsymbol{x}}^{(k-1)},\ \mathbf{a}_Z \right\rangle^2}{\|\mathbf{a}_Z\|_2^2} \\ &= \sum_{i=1}^{m} \frac{\left\langle \hat{\boldsymbol{x}}^* - \hat{\boldsymbol{x}}^{(k-1)},\ \mathbf{a}_i \right\rangle^2}{\|\mathsf{A}\|_{\mathrm{F}}^2} \\ &= \frac{\left\| \mathsf{A}(\hat{\boldsymbol{x}}^* - \hat{\boldsymbol{x}}^{(k-1)}) \right\|_2^2}{\|\mathsf{A}\|_{\mathrm{F}}^2}. \end{aligned}$$

By induction, it is evident that $\hat{\boldsymbol{x}}^{(k)}$ is in the row span of A for any k when $\hat{\boldsymbol{x}}^{(0)}$ is; in addition, the same is true for $\hat{\boldsymbol{x}}^*$ by the definition of pseudo-inverse [22]. Therefore, $\left\| \mathsf{A}(\hat{\boldsymbol{x}}^* - \hat{\boldsymbol{x}}^{(k-1)}) \right\|_2 \geq \sigma_{\min}(\mathsf{A}) \left\| \hat{\boldsymbol{x}}^* - \hat{\boldsymbol{x}}^{(k-1)} \right\|_2$, which completes the proof. ■

Proof: (of Theorem 3)

Let us denote the k-th estimate as function of γ_k, i.e., $\hat{\boldsymbol{x}}^{(k)}(\gamma_k)$; note that $\hat{\boldsymbol{x}}^{(k)}(0) = \hat{\boldsymbol{x}}^{(k-1)}$. The expression $||\hat{\boldsymbol{x}}^{(k)}(\gamma_k) - \hat{\boldsymbol{x}}||_2^2$ is a convex function of γ_k on $[0, 1]$, therefore:

$$\left\| \hat{\boldsymbol{x}}^{(k)} - \hat{\boldsymbol{x}} \right\|_2^2 \leq \gamma_k \left\| \hat{\boldsymbol{x}}^{(k)}(1) - \boldsymbol{x} \right\|_2^2 + (1 - \gamma_k) \left\| \hat{\boldsymbol{x}}^{(k-1)} - \boldsymbol{x} \right\|_2^2. \tag{17}$$

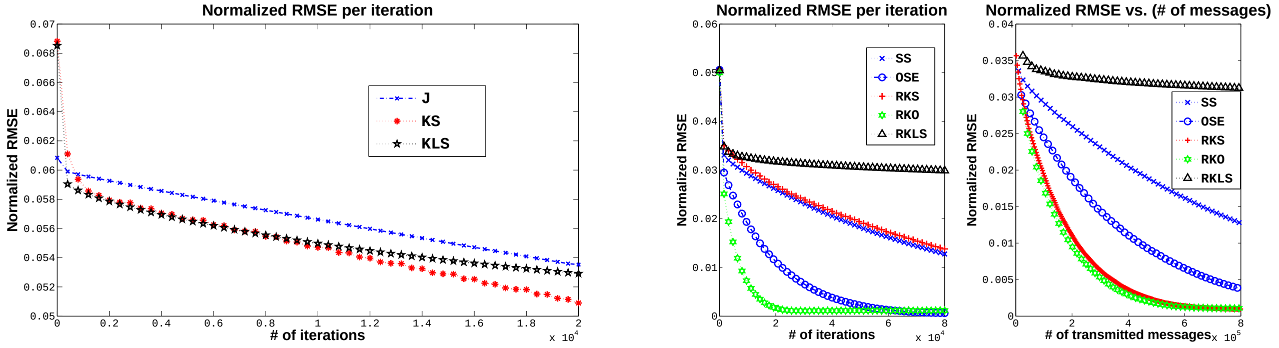
Define the linear varieties:

$$H_i \quad := \quad \{\boldsymbol{x} : \mathbf{a}_i^\top \boldsymbol{x} = b_i\} \tag{18}$$

$$H_i^w \quad := \quad \{\boldsymbol{x} : \mathbf{a}_i^\top \mathbf{x} = b_i + w_i\} \tag{19}$$

Note that $\hat{\boldsymbol{x}}^{(k)}(1)$ is the projection of $\hat{\boldsymbol{x}}^{(k-1)}$ on H_i^w. Let us denote the projection of $\hat{\boldsymbol{x}}^{(k-1)}$ on H_i as $\boldsymbol{x}^{(k)}$. The two hyper-planes are *parallel* with common normal $\mathbf{a}_i$, so $\boldsymbol{x}^{(k)}$ is the projection of $\hat{\boldsymbol{x}}^{(k)}(1)$ on H_i, whence:

$$\left\| \hat{\boldsymbol{x}}^{(k)}(1) - \boldsymbol{x} \right\|_2^2 = \left\| \boldsymbol{x}^{(k)} - \boldsymbol{x} \right\|_2^2 + \left\| \hat{\boldsymbol{x}}^{(k)}(1) - \boldsymbol{x}^{(k)} \right\|_2^2$$



(a) RMSE of synchronous algorithms

(b) RMSE of asynchronous algorithms. Left figure depicts RMSE per iteration; right figure shows RMSE versus number of transmitted messages.

Fig. 1. Convergence analysis of synchronous and asynchronous algorithms

From the proof of Theorem 2 it follows that

$$\mathbb{E}\left\|\boldsymbol{x}^{(k)}-\boldsymbol{x}\right\|_2^2 \le \left(1-\frac{1}{\kappa(\mathsf{A})}\right)\left\|\hat{\boldsymbol{x}}^{(k-1)}-\boldsymbol{x}\right\|_2^2 .$$

If the selected row is i we have $\left\|\hat{\boldsymbol{x}}^{(k)}(1)-\boldsymbol{x}^{(k)}\right\|_2^2 = \frac{w_i^2}{\|\mathbf{a}_i\|_2^2}$, therefore $\mathbb{E}\left\|\hat{\boldsymbol{x}}^{(k)}(1)-\boldsymbol{x}^{(k)}\right\|_2^2 = W$, and the result immediately follows. ■

References

[1] N. Freris, H. Kowshik, and P. R. Kumar, “Fundamentals of Large Sensor Networks: Connectivity, Capacity, Clocks and Computation,” *Proceedings of the IEEE*, vol. 98, no. 1, pp. 1828–1846, Nov. 2010.

[2] N. Freris, S. Graham, and P. R. Kumar, “Fundamental Limits on Synchronizing Clocks over Networks,” *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 1352–1364, Jun. 2011.

[3] S. Graham and P. R. Kumar, “The Convergence of Control, Communication, and Computation,” in *Proceedings of Personal Wireless Communication*. Heidelberg: Springer-Verlag, Oct. 2003, pp. 458–475.

[4] K. Plarre and P. R. Kumar, “Tracking objects with networked scattered directional sensors,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008, article ID: 360912, 10 pages.

[5] R. McCabe, N. Freris, and P. R. Kumar, “Controlled Random Access MAC for Network Utility Maximization in Wireless Networks,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Dec. 2008, pp. 2350–2355.

[6] J. Ni, B. Tan, and R. Srikant, “Q-CSMA: Queue-Length Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks,” in *Proceedings of the 29th IEEE Conference on Computer Communications*, March 2010, pp. 1–5.

[7] N. Freris, “Performance bounds for csma-based medium access control,” in *Proceedings of the 50th IEEE Conference on Decision and Control*, Orlando, Florida, Dec. 2011, pp. 5945–5950.

[8] B. Sundararaman, U. Buy, and A. Kshemkalyani, “Clock Synchronization for Wireless Sensor Networks: a survey,” *Ad Hoc Networks*, vol. 3, pp. 281–323, May 2005.

[9] A. Giridhar and P. R. Kumar, “Distributed Clock Synchronization over Wireless Networks: Algorithms and Analysis,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, Dec. 2006, pp. 4915–4920.

[10] P. Barooah and J. Hespanha, “Estimation on Graphs from Relative Measurements,” *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 57–74, Aug. 2007.

[11] T. Strohmer and R. Vershynin, “A randomized solver for linear systems with exponential convergence,” in *Approximation, Randomization and Combinatorial Optimization Algorithms and Techniques, Lecture Notes in Computer Science*. Springer, 2006, pp. 499–507.

[12] ——, “A Randomized Kaczmarz Algorithm with Exponential Convergence,” *Journal of Fourier Analysis and Applications*, vol. 15, no. 1, pp. 262–278, 2009.

[13] A. Giridhar and P. R. Kumar, “The Spatial Smoothing Method of Clock Synchronization in Wireless Networks,” *Theoretical Aspects of Distributed Computing in Sensor Networks*, pp. 227–256, 2011.

[14] P. Barooah, N. da Silva, and J. Hespanha, “Distributed Optimal Estimation from Relative Measurements for Localization and Time Synchronization,” *Distributed Computing in Sensor Systems, Lecture Notes in Computer Science*, vol. 4026, pp. 266–281, Jun. 2006.

[15] J. Elson, R. Karp, C. H. Papadimitriou, and S. Shenker, “Global Synchronization in Sensornets,” in *Proceedings of LATIN*. Buenos Aires: Springer, 2004, pp. 609–624.

[16] R. Solis, V. Borkar, and P. R. Kumar, “A New Distributed Time Synchronization Protocol for Multihop Wireless Networks,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, Dec. 2006, pp. 2734–2739.

[17] P. Barooah and J. Hespanha, “Error Scaling Laws for Optimal Estimation from Relative Measurements,” *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5661–5673, Dec. 2009.

[18] ——, “Distributed Estimation from Relative Measurements in Sensor Networks,” in *Proceedings of the 2nd International Conference on Intelligent Sensing and Information Processing*, Dec. 2005, pp. 226–231.

[19] N. Freris, V. Borkar, and P. R. Kumar, “A model-based approach to clock synchronization,” in *Proceedings of the 48th IEEE Conference on Decision and Control*, Sanghai, Dec. 2009, pp. 5744–5749.

[20] ——, “A model-based approach to network clock synchronization,” *In preparation*, 2012.

[21] P. Barooah, J. Hespanha, and A. Swami, “On the effect of Asymmetric Communication on Distributed Time Synchronization,” in *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, Dec. 2007, pp. 5465–5471.

[22] D. Luenberger, *Optimization by Vector Space Methods*. John Wiley and Sons, 1969.

[23] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[24] M. Mihail, “Conductance and Convergence of Markov Chains-A Combinatorial Treatment of Expanders,” in *FOCS*, 1989, pp. 526–531.

[25] S. Kaczmarz, “Angenäherte Auflsung von Systemen Linearer Gleichungen,” *Bulletin International de l’Acadmie Polonaise des Sciences et des Lettres*, vol. 35, pp. 355–357, 1937.

[26] D. Bertsekas, *Nonlinear Programming: 2nd Edition*. Athena Scientific, 1999.

[27] D. Needell, “Randomized Kaczmarz Solver for Noisy Linear Systems,” *Bit Numerical Mathematics*, vol. 50, no. 2, pp. 395–403, 2010.

[28] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, “Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms,” *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986. [Online]. Available: http://dx.doi.org/10.1109/TAC.1986.1104412

[29] P. Gupta and P. R. Kumar, “Critical Power for Asymptotic Connectivity,” in *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, FL, Dec. 1998, pp. 1106–1110.