

ON THE EFFECT OF ASYMMETRIC COMMUNICATION ON DISTRIBUTED TIME SYNCHRONIZATION

Prabir Barooah João P. Hespanha Ananthram Swami

Abstract—Several distributed algorithms have been recently proposed to estimate clock offsets and skews in a network of processors from a set of noisy measurements of the difference between clock offsets and of the ratios of clock skews. These algorithms are designed to converge to the optimal, i.e., the best linear unbiased, estimates even in the presence of node and link failures. However, they require symmetric communication between nodes for convergence. We examine the case when communication is asymmetric, i.e., when a node can receive information from another node but not vice versa. We first show that in the presence of asymmetric communication links, these algorithms converge to an unbiased but suboptimal estimate. In fact, we show that with a distributed algorithm that is constrained to use only local information, it is generally impossible to converge to the optimal estimate when communication is asymmetric. We characterize the resulting estimate that these algorithms converge to in the presence of asymmetry, and node and link failures, and its error covariance.

I. INTRODUCTION

Time-synchronization is an important problem for sensor and actuator networks because a wide range of potential applications for these networks require tight synchronization. For example, due to the severe constraints on the energy budgets of the constituent devices in a sensor network, sleep scheduling is proposed to maximize network lifetime; see [1] and references therein. Such scheduling requires the clocks at the sensor nodes to be accurately synchronized. Collaborative processing tasks such as event detection and warning, target tracking and sensor fusion also require clock synchronization. In addition, for feedback control over communication networks to be a reality, it is important that the clocks of different nodes connected by a communication network are synchronized with one another.

In practice, clocks have two main sources of inaccuracy: *skew* and *offset*. Skew refers to the rate at which clocks measure time and offset refers to the difference between the local times of two clocks that have the same skew. At a particular “global” time t , the measured local time t_u at a clock u is given by $t_u = \alpha_u t + \beta_u$, where α_u is the clock’s skew w.r.t the global time and β_u is the offset. The relative offset between a pair of nodes u and v in a network can be measured (up to some error) by exchanging time-stamped messages between them, e.g., using the methods described in [2] and in [3, Chapter 2]. In particular, a noisy

measurement of $\beta_u - \beta_v$ can be obtained. Noisy ratios of skews α_u/α_v can also be measured by exchanging time-stamped messages [3], [4]. Measurements of skew ratios and offsets between a pair of nodes can also be obtained by the RBS (Reference Broadcast System) method, which does not require bidirectional message exchange between them but requires the involvement of a third node [5]. The problem of estimating the clock skews w.r.t. a global time can be separated from the problem of estimating offsets, and in fact, can be handled in almost exactly the same way that the offset estimation problem is handled, as described in [6]. For this reason and for simplicity of presentation, for now we restrict our attention to the offset estimation problem alone.

Many of the current synchronization protocols use relative offset measurements to synchronize node clocks to a global clock in a sequential manner [5], [7], [8]. For example, in TPSN protocol [7], nodes close to a root node, called level 1 nodes, synchronize their clocks to the root node’s clock by using such relative measurements. The nodes close to level 1 nodes, called level 2 nodes, in turn synchronize their clocks to the level 1 nodes’ clocks, and so on until all the nodes are synchronized. For a review of time synchronization protocols for sensor networks, see [4].

It is possible to construct estimates of clock offsets more accurate than what is possible by the above mentioned methods by using all the available relative offset measurements. In particular, it is possible to compute the optimal estimate of the clock offsets (w.r.t. a global time frame) by using all the relative clock offset measurements [6], [9], [10], [11]. The optimal estimate refers to the one obtained from the Best Linear Unbiased Estimator (BLUE), which has the minimum variance among all linear unbiased estimators [12].

Recently, several distributed algorithms have been proposed to iteratively compute the optimal estimates of clock offsets [6], [9], [10], [11], [2]. These algorithms are iterative; at every step nodes exchange their current estimates with their neighbors and update their own estimates based on the received estimates of their neighbors. All the references above assumed symmetric (bidirectional) communication between nodes. However, in ad-hoc wireless networks, communication between certain pairs of nodes may be asymmetric, meaning that a node may be able to receive transmissions from another node but may not be able to transmit information to that node. This could be caused by non-homogeneous interference, packet collisions, and even due to an imperfect sleep scheduling arising from inaccurate time-synchronization itself. Motivated by this possibility, we examine the behavior of these algorithms in the presence

This material is based upon work supported by the Institute for Collaborative Biotechnologies through grant DAAD19-03-D-0004 from the U.S. Army Research Office and by the National Science Foundation under Grant No. CCR-0311084. PB and JPH are with the Dept. of Electrical and Computer Engineering at Univ. of California, Santa Barbara, CA, and AS is with the U. S. Army Research Lab, Adelphi, MD.

of asymmetric communication. Since the algorithms in [6], [11], [2] are all special cases of the *Jacobi algorithm* described in [9], in this paper we focus exclusively on the Jacobi algorithm.

We first show that in the presence of asymmetric communication links, the Jacobi algorithm *does not* converge to the optimal estimates. However, the Jacobi algorithm does converge to an unbiased but sub-optimal estimate, provided certain connectivity properties of the communication graph are satisfied. In fact, we prove that it is impossible to design a distributed algorithm that converges to the optimal estimate when communication is asymmetric, if the algorithm is constrained to use only local information. The Jacobi algorithm is shown to be robust to temporary node and communication link failures.

We characterize the error covariance of the “limiting” estimate the Jacobi algorithm converges to when communication is asymmetric. An interesting feature of this limiting estimate is that the error variances of some of the node offset estimates may increase upon using more measurements. This is in contrast to the optimal estimate, for which more measurements (no matter how noisy) always leads to lower variances for all node offset estimates. Thus, an important lesson is that in the presence of asymmetric communication, more measurements may not lead to more accurate estimates.

II. PROBLEM STATEMENT AND PREVIOUS WORK

The local time t_u at node u is related to the global (unknown) time t by

$$t_u = \alpha_u t + \beta_u,$$

where α_u and β_u are the skew and offset of node u 's clock, respectively. By using one of the methods mentioned in Section I, it is possible to obtain noisy measurement $\zeta_{u,v}$ of the difference between clock offsets of certain node pairs (u, v) :

$$\eta_{u,v} = \beta_u - \beta_v + e_{u,v}, \quad (1)$$

where $e_{u,v}$ is a zero-mean measurement error. For the details of obtaining such measurements, see [3, Chapter 2]. Similarly, noisy ratios of their skews, α_u/α_v can also be obtained. Upon taking logarithm, we get

$$\xi_{u,v} = \log(\alpha_u) - \log(\alpha_v) + e'_{u,v}, \quad (2)$$

where $\xi_{u,v}$ is the logarithm of the measured ratio of clock skews, and $e'_{u,v}$ is a zero-mean error. The details of obtaining such measurements are explained in [3, Chapter 2]. It is clear from the similarity of (1) and (2) that both of them are special cases of

$$\zeta_{u,v} = x_u - x_v + \epsilon_{u,v}, \quad (3)$$

where $\zeta_{u,v}$ is a noisy “relative measurement” between the unknown variables x_u and x_v , and $\epsilon_{u,v}$ is a zero-mean measurement error.

In a network of n nodes, we have n such offsets (and logarithm of skews) x_1, x_2, \dots, x_n , which we will refer to

as *node variables* from now on. We associate these node variables with the nodes $\mathbf{V} = \{1, 2, \dots, n\}$ of a directed *measurement graph* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. The measurements correspond to the edges in \mathbf{E} , where each edge consists of an ordered pair (u, v) such that a noisy relative measurement of the form (3) is available. Using only relative measurements, the x_u 's can be determined only up to an additive constant. To avoid this ambiguity, we assume that a particular variable (say x_o) is known. One could imagine that the node o is a cluster-head and the times of all the nodes need to be synchronized to the cluster-head. In certain cases, more than one node's offsets and skews may be known, such as when a subset of the nodes can receive GPS signals. The node variables that are known are called the *reference variables* and the corresponding nodes are called the *reference nodes*. The set of reference nodes is denoted by \mathbf{V}_r . The problem is to estimate the unknown node variables from all the available measurements and the reference variables.

Our objective is to construct an optimal estimate \hat{x}_u^* of x_u for every node $u \in \mathbf{V} \setminus \mathbf{V}_r$. The *optimal estimate* refers to the estimate produced by the classical Best Linear Unbiased Estimator (BLUE), which achieves the minimum variance among all linear unbiased estimators [12]. To compute the optimal estimate directly one would need all the measurements and the topology of the graph (see section III). Thus, if a central processor has to compute the \hat{x}_u^* s, all this information has to be transmitted to it. In a large ad-hoc network, such centralized computation suffers from poor scalability and lack of robustness to node and communication link failures. Therefore a distributed algorithm that can compute the optimal estimate while using only local communication is advantageous.

An iterative distributed algorithm, called the Jacobi algorithm, to compute the optimal estimates was proposed in [9]. The algorithm is distributed in the sense that every sensor can compute its own node variable's estimate and all the data needed for the computation can be obtained from communication with only those neighbors with whom it has relative measurements. The algorithm is based on the Jacobi method of iteratively computing the solution to a system of linear equations. A distributed algorithm was also proposed in [2] for computing the least squares estimate of the time shifts, which was a special case of the Jacobi algorithm, where all the measurement error variances were assumed to be equal. An algorithm based on the Jacobi method was also proposed in [6], though not investigated. However, the algorithms proposed in [6], [9], [11], [2] assumed that the communication between nodes is symmetric. In other words, if node u receives a transmission from node v , then v can also receive a transmission from u . In many situations, specially in wireless ad-hoc networks, node u maybe able to receive messages from v but not vice versa.

III. OPTIMAL ESTIMATION OF NODE VARIABLES

The measurement error ϵ_e on the edge $e \in \mathbf{E}$ is assumed to be of known variance, and uncorrelated with all other

measurement errors. That is, for every edge $e \in \mathbf{E}$, $\sigma_e^2 = \mathbb{E}[\epsilon_e^2]$ is known, and $\mathbb{E}[\epsilon_e \epsilon_{\bar{e}}] = 0$ if $e \neq \bar{e}$. Here and in the sequel, we use the symbol e to denote an edge as well as the index of that edge, i.e., $e = (u, v) \in \mathbf{E}$ as well as $e = 1, \dots, |\mathbf{E}|$. Consider a measurement graph \mathbf{G} with n nodes and m edges. Let $\mathbf{X} := [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ be the vector obtained by stacking together all node variables, known and unknown. Similarly define $\mathbf{z} := [\zeta_1, \zeta_2, \dots, \zeta_m]^T \in \mathbb{R}^m$ and $\boldsymbol{\epsilon} := [\epsilon_1, \epsilon_2, \dots, \epsilon_m]^T \in \mathbb{R}^m$. We can now rewrite (3) as

$$\mathbf{z} = A^T \mathbf{X} + \boldsymbol{\epsilon}, \quad (4)$$

where A is the *incidence matrix of the graph* \mathbf{G} [13]. The incidence matrix is an $n \times m$ matrix with one row per node and one column per edge defined by $A := [a_{u,e}]$, where $a_{u,e}$ is nonzero if and only if the edge $e \in \mathbf{E}$ is incident on the node $u \in \mathbf{V}$. When nonzero, $a_{ue} = -1$ if the edge e is directed towards u and $a_{ue} = 1$ otherwise.

By partitioning \mathbf{X} into a vector \mathbf{x} containing all the unknown node variables and another vector \mathbf{x}_r containing all the known reference node variables: $\mathbf{X}^T = [\mathbf{x}_r^T, \mathbf{x}^T]^T$, we can re-write (4) as $\mathbf{z} = A_r^T \mathbf{x}_r + A_b^T \mathbf{x} + \boldsymbol{\epsilon}$, where A_r contains the rows of A corresponding to the reference nodes and A_b contains the rows of A corresponding to the unknown node variables. The equation above can be further rewritten as:

$$\bar{\mathbf{z}} = A_b^T \mathbf{x} + \boldsymbol{\epsilon}, \quad (5)$$

where $\bar{\mathbf{z}} := \mathbf{z} - A_r^T \mathbf{x}_r$ is a known vector. The optimal estimate (BLUE) $\hat{\mathbf{x}}^*$ of the vector of unknown node variables \mathbf{x} for the measurement model (5) is the solution to the following system of linear equations:

$$L \hat{\mathbf{x}}^* = \mathbf{b}, \quad (6)$$

where $L := A_b P^{-1} A_b^T$, $\mathbf{b} := A_b P^{-1} \bar{\mathbf{z}}$, and $P := \mathbb{E}[\boldsymbol{\epsilon} \boldsymbol{\epsilon}^T]$ is the covariance matrix of the measurement error vector [12]. Since the measurement errors on two different edges are uncorrelated, P is a positive definite diagonal matrix with the measurement error variances along its diagonal: $P = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2) \in \mathbb{R}^{m \times m}$. The error covariance of the optimal estimate, $\boldsymbol{\Sigma} := \mathbb{E}[(\mathbf{x} - \hat{\mathbf{x}}^*)(\mathbf{x} - \hat{\mathbf{x}}^*)^T]$, is given by $\boldsymbol{\Sigma} = L^{-1}$, where L^{-1} exists if and only if every weakly connected component of the graph \mathbf{G} has at least one reference node [9]. A directed graph \mathbf{G} is said to be *weakly connected* if there is a path from every node to every other node, not necessarily respecting the direction of the edges.

The optimal estimate can be computed by a fusion center when all the nodes are able to communicate their noisy measurements to the fusion center over a noise-free channel. In the next section we describe and analyze a distributed algorithm to compute the optimal estimates that does not require a fusion center.

IV. JACOBI ALGORITHM

To describe the Jacobi algorithm with (possibly) asymmetric communication, we introduce a directed graph $\mathbf{G}^c = (\mathbf{V}, \mathbf{E}^c)$ consisting of the same nodes as the measurement graph, but with edge directions that determine which nodes

can receive information from which other nodes. We call \mathbf{G}^c the *communication graph* associated with the measurement graph \mathbf{G} . In particular, a node u can receive information from another node v if and only if there is an edge (v, u) in the communication graph \mathbf{G}^c . Communication between two nodes u and v is symmetric if and only if both (u, v) and (v, u) belong to \mathbf{E}^c . An edge e (in \mathbf{G} or \mathbf{G}^c) between two nodes u and v is said to be incident on both the nodes u and v , which is denoted by $e \sim u$ and $e \sim v$ respectively, whether the edge is directed from u to v or otherwise. For an edge e that is incident on a node u , we denote $e \setminus u$ as the *other end* of u . That is, if $e = (v, u)$, then $e \setminus u = v$ and $e \setminus v = u$.

The following conditions are assumed to hold:

- Assumption 1:*
- 1) Every weakly connected component of the measurement graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ has at least one reference node, and \mathbf{G} does not contain multiple edges between the same pair of nodes.
 - 2) The communication graph $\mathbf{G}^c = (\mathbf{V}, \mathbf{E}^c)$ is such that for every pair of nodes that have a measurement edge between them, there is at least one communication edge between them.
 - 3) If there is a communication edge between a pair of nodes, then there must be a corresponding measurement edge between them (in either direction).
 - 4) Every node that is not a reference node has at least one communication edge directed toward it. \square

The assumption of not having multiple edges between a pair of nodes is not restrictive because multiple measurements between the same pair of nodes can be combined into a single measurement (see Chapter 2 of [3]). The second condition ensures that the nodes employing the algorithm will be able to use all the available measurements. The third condition clarifies that the communication graph is used only to model the information exchange that occurs during the execution of the algorithm. The fourth condition ensures that every node (other than a reference node) is able to receive messages from at least one neighbor, since otherwise it cannot update its estimate.

Figure 1 shows a measurement graph \mathbf{G} and the associated communication graph \mathbf{G}^c . Since there is only one communication edge $(4, 2)$ between 4 and 2, node 2 can receive broadcasts from 4 but not the other way around. This communication graph is therefore asymmetric. Note that lack of communication edges $(2, 1)$ and $(3, 1)$ is not a cause of asymmetry since the reference node 1 does not use any information from its neighbors (see Assumption 1). According to the terminology introduced above, if $e = (4, 2)$, then $e \setminus 2 = 4$ and $e \setminus 4 = 2$.

The Jacobi algorithm can be mathematically expressed by the following update law implemented by every node $u \in \mathbf{V} \setminus \mathbf{V}_r$, where $\hat{x}_u^{(i)}$ denotes node u 's estimate of x_u at the i^{th} iteration:

$$\left(\sum_{e \in \bar{\mathbf{E}}_u} \frac{1}{\sigma_e^2} \right) \hat{x}_u^{(i+1)} = \sum_{e \in \bar{\mathbf{E}}_u} \frac{1}{\sigma_e^2} \left(\hat{x}_{e \setminus u}^{(i)} + a_{u,e} \zeta_e \right), \quad (7)$$

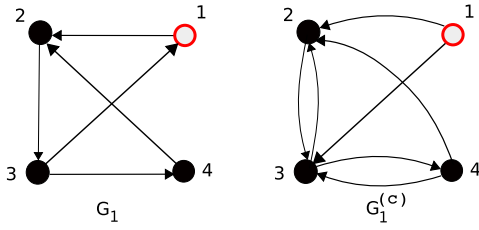


Fig. 1. A measurement graph and a communication graph associated with it. Note that the asymmetry in the communication graph comes from the lack of a communication edge from node 2 to node 4. Though there is no edge from 2 to 1, that is not a cause of asymmetry since 1 is a reference node and it does not use information from its neighbors even if available.

where $\vec{\mathbf{E}}_u$ is the set of edges in the measurement graph \mathbf{G} that are incident on the node u such that there are communication edges from their other ends toward u :

$$\vec{\mathbf{E}}_u := \{e \in \mathbf{E} \mid e \sim u, (e \setminus u, u) \in \mathbf{E}^c\}, \quad (8)$$

and $a_{u,e}$ is the $(u,e)^{\text{th}}$ entry of the incidence matrix of \mathbf{G} defined in Section III. The Jacobi algorithm has a simple interpretation. When communication is symmetric, a node obtains multiple estimates of its own variable by adding the appropriate relative measurements to its measurement-neighbors' estimates. It then computes the new estimate of its variable by taking a weighted average of those estimates. When communication is asymmetric, it carries out the same procedure, but the averaging uses information from only those measurement neighbors from which it can receive data. We have assumed above that transmitted data is digitized and coded sufficiently so that errors due to noisy communications channels can be ignored. That is, the communication channels between the nodes are error-free during the iterative updating of the estimates. Thus, when an estimate is transmitted in a data packet, if successfully delivered, the estimate is received without being corrupted by any error.

The question naturally arises as to whether this algorithm converges to an unbiased estimate of \mathbf{x} ; if so, whether it converge to the optimal estimate, and if not, what is the covariance of the resulting estimate. To answer these questions, we need to express the iteration (7) in a more compact form. To that end, we now define a few matrices.

We define the *combined incidence matrix* $A^c \in \mathbb{R}^{n \times m}$ for the pair of directed graphs $(\mathbf{G}, \mathbf{G}^c)$, where m is the number of edges in the measurement graph \mathbf{G} , in the following manner:

$$[A^c]_{u,e} = \begin{cases} a_{ue} & \text{if } e \sim u, (e \setminus u, u) \in \mathbf{E}^c \\ 0 & \text{otherwise} \end{cases}, \quad (9)$$

where $a_{u,e}$ is the $(u,e)^{\text{th}}$ entry of the incidence matrix A for the measurement graph \mathbf{G} . The incidence matrix was defined in Section III. If communication between every pair of nodes is symmetric, then $A^c(\mathbf{G}, \mathbf{G}^c) = A(\mathbf{G})$.

The *weighted in-degree matrix* $D \in \mathbb{R}^{n \times n}$ of the directed

graph pair $(\mathbf{G}, \mathbf{G}^c)$ is defined as a diagonal matrix with

$$[D]_{u,u} = \sum_{e \in \vec{\mathbf{E}}_u} \frac{1}{\sigma_e^2}. \quad (10)$$

Similarly, the *weighted adjacency matrix* $C \in \mathbb{R}^{n \times n}$ of the directed graph pair $(\mathbf{G}, \mathbf{G}^c)$ is defined as

$$[C]_{u,v} = \begin{cases} \frac{1}{\sigma_e^2} & \text{if } (v,u) \in \mathbf{E}^c \text{ and } e \sim u, e \sim v, \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Let $M, N \in \mathbb{R}^{n_b \times n_b}$ be the sub-matrices of D and C , respectively, obtained by removing the rows and columns corresponding to the reference nodes, where n_b is the number of nodes in $\mathbf{V} \setminus \mathbf{V}_r$ that do not know their variables. Using the matrices introduced above, the Jacobi algorithm (7) with asymmetric communication links can be compactly expressed as the following discrete-time dynamical system:

$$M\mathbf{x}^{k+1} = N\mathbf{x}^k + \mathbf{b}^c, \quad (12)$$

$$\mathbf{b}^c := A_b^c P^{-1}(\mathbf{z} - A_r^T \mathbf{x}_r), \quad (13)$$

where A_b^c is a submatrix of A^c that is obtained from A^c by removing from it the rows that correspond to the reference nodes. The fixed point of (12) is given by the solution of the following system of linear equations, when it exists.

$$L_c \hat{\mathbf{x}}^\infty = \mathbf{b}^c, \quad \text{where} \quad (14)$$

$$L_c := M - N. \quad (15)$$

For $\hat{\mathbf{x}}^\infty$ to exist and be unique, L_c must be invertible. The next lemma, whose proof is in Appendix I, states when this is true.

Lemma 1: The matrix L_c is invertible if and only if there is a directed path in \mathbf{G}^c from at least one reference nodes to u for every $u \in \mathbf{V} \setminus \mathbf{V}_r$. \square

We conclude that when the path existence condition of Lemma 1 is satisfied, the Jacobi algorithm converges to the estimate $\hat{\mathbf{x}}^\infty$. In the next section, we show that this convergence is, in fact, robust to link and node failures under appropriate conditions.

It is straightforward to verify that

$$L_c = A_b^c P^{-1} A_b^T \quad (16)$$

and this matrix does not depend on the edge directions of \mathbf{G} but does depend on the edge directions in \mathbf{G}^c . Furthermore, it can be verified if \mathbf{G} is weakly connected, then $L_c = L$ if and only if communication is symmetric between all pairs of nodes. Therefore, when communication is asymmetric, the Jacobi algorithm does not converge to the optimal estimate in general.

A. Convergence and robustness to faults

The algorithm described by (12) is synchronous, since that description implicitly assumes that all the nodes update their estimates at the same time after getting updates from all of their communication neighbors. In practice, with time dependent communication failures or due to sleep scheduling,

waiting to get information from all neighbors may not be advisable. In this case, the algorithm can be implemented in an asynchronous fashion, in which nodes wait to receive estimates from their neighbors until a "time-out" occurs. If estimates from some neighbors do not arrive in this time, they use the previously received data from those nodes.

Consider time index $t \in \mathbb{N}$ that is incremented by 1 at the end of every time-out period. A communication edge (u, v) is said to fail in time t if during the time between $t - 1$ and t , all broadcasts from node u fail to reach v , which may occur due to interference etc. A node u is said to fail at time t if during the time between $t - 1$ and t , node u either does not broadcast its current estimate or does not process any information. Such node failures can occur due to sleep-scheduling [1], among other reasons. We say that a communication edge (u, v) is *active* at time t if in that time, neither of the nodes u and v fails, and the communication edge (u, v) does not fail. At every time t , the communication graph $\mathbf{G}^c(t) = (\mathbf{V}, \mathbf{E}^c(t))$ consists of all the nodes of \mathbf{G} and all the communication edges that are active in that time. We assume that every node $u \in \mathbf{V} \setminus \mathbf{V}_r$ detects its communication-neighbors, i.e., the nodes from which it can receive messages, during an initial detection phase, before the iterations begin. This detection, which may be carried out even during the process of obtaining the relative measurements, leads to an *initial communication graph* $\mathbf{G}_{\text{init}}^c = (\mathbf{V}, \mathbf{E}_{\text{init}}^c)$ consisting of those communication edges that were active over the time interval of the detection phase. A node does not update its list of communication neighbors thereafter.

The next theorem states how the algorithm behaves in the presence of asymmetric communication, and is implemented in an asynchronous manner in the presence of node and link failures. We consider the following model of random failures. At every time instant $t \in \mathbb{N}$, every communication edge may fail independently of all other edges with probability p , and every node may fail independently of all other nodes, with probability q , where $p < 1, q < 1$. This model of failure is referred to as *i.i.d. failure*.

Theorem 1: Consider the asynchronous Jacobi algorithm implemented on the measurement graph \mathbf{G} and its associated time-varying communication graph $\mathbf{G}^c(t)$, such that \mathbf{G} and $\mathbf{G}^c(t)$ satisfy Assumption 1 at every $t \in \mathbb{N}$. Let $\mathbf{G}_{\text{init}}^c$ denote the initial measurement graph that describes the neighbor relations used by the nodes to implement the algorithm. Let L_c and \mathbf{b}^c be as defined in (15) and (13) for the pair $(\mathbf{G}, \mathbf{G}_{\text{init}}^c)$. The Jacobi algorithm converges to $\hat{\mathbf{x}}^\infty$ for every initial condition, where $\hat{\mathbf{x}}^\infty$ is the unique solution of $L_c \mathbf{x} = \mathbf{b}^c$, if

- 1) for every node $u \in \mathbf{V} \setminus \mathbf{V}_r$, there is a directed path in $\mathbf{G}_{\text{init}}^c$ from at least one reference node to that node, and
- 2) no communication edge in $\mathbf{G}_{\text{init}}^c$ fails permanently, and no communication edge that is not in $\mathbf{G}_{\text{init}}^c$ remains

active infinitely often, i.e.,

$$\bigcap_{\ell=1}^{\infty} \bigcup_{t=\ell}^{\infty} \mathbf{G}^c(t) = \mathbf{G}_{\text{init}}^c. \quad (17)$$

When nodes and communication edges fail according to the i.i.d. failure model, if condition 1 above is satisfied, then the Jacobi algorithm converges to the unique solution of $L_c \mathbf{x} = \mathbf{b}^c$ almost surely. \square

Due to space limitations, the proof of this result is omitted. The interested reader is referred to [3, Chapter 3] for a complete proof. We briefly note that the proof of the convergence of the asynchronous Jacobi algorithm is based on the results on asynchronous iterations in [14].

Remark 1: If the communication graph $\mathbf{G}_{\text{init}}^c$ is symmetric, $L_c = L$, $A_b^c = A_b$, and therefore $\hat{\mathbf{x}}^\infty = \hat{\mathbf{x}}^*$, which means that the Jacobi algorithm will converge to the optimal estimate. When communication is asymmetric, it converges to the sub-optimal estimate $\hat{\mathbf{x}}^\infty$. Therefore, asymmetry in communication leads to poorer estimates.

B. Unbiasedness and covariance of the limiting estimate

The next result characterizes the covariance of the limiting estimate produced by the Jacobi algorithm.

Theorem 2: The limiting estimate $\hat{\mathbf{x}}^\infty$, when it exists, is unbiased and the covariance of the estimation error $\tilde{\mathbf{e}}^\infty := \mathbf{x} - \hat{\mathbf{x}}^\infty$ is given by

$$\Sigma := \mathbb{E}[\tilde{\mathbf{e}}^\infty \tilde{\mathbf{e}}^{\infty T}] = L_c^{-1} A_b^c P^{-1} A_b^{cT} L_c^{-T}. \quad \square$$

Proof. From (14), (13) and (5), we get

$$\begin{aligned} \hat{\mathbf{x}}^\infty &= L_c^{-1} A_b^c P^{-1} (\mathbf{z} - A_r^T \mathbf{x}_r) \\ &= L_c^{-1} A_b^c P^{-1} (A_b^T \mathbf{x} + \boldsymbol{\epsilon}) \\ &= \mathbf{x} + L_c^{-1} A_b^c P^{-1} \boldsymbol{\epsilon}, \end{aligned}$$

where the last equality follows from (16). It follows that $\mathbb{E}[\hat{\mathbf{x}}^\infty] = \mathbf{x}$. The expression for the covariance of the estimation error $\tilde{\mathbf{e}}^\infty$ follows immediately. \blacksquare

We simulated the Jacobi algorithm for the measurement and communication graph pair shown in Figure 1. Simulation results are shown in Figure 2. All the nodes started with 0 initial estimates for their node variables, i.e., log-skews and offsets. At every iteration of the simulation, every communication edge was allowed to fail with a probability of 0.2, independent of all other edges, i.e., $p = 0.2$ and $q = 0$. The Figure validates the predictions of Theorem 1 and 2: the estimate converges to the predicted value $\hat{\mathbf{x}}^\infty$ but not to the optimal estimate $\hat{\mathbf{x}}^*$.

C. An impossibility result

The result established in the previous section shows that the Jacobi algorithm will not lead to convergence to the optimal estimate when the communication graph is asymmetric. This raises the question if it possible to construct a distributed algorithm that will ensure convergence to the optimal estimate even when communication is asymmetric.

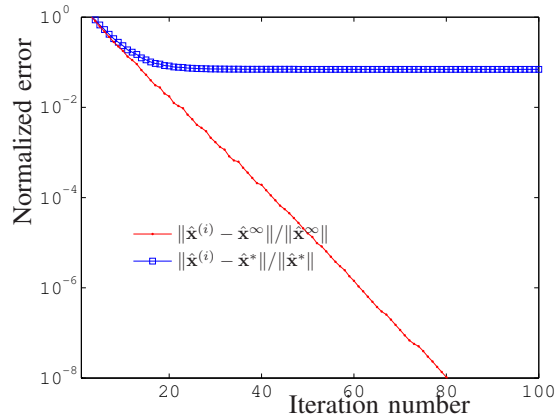


Fig. 2. Simulation results on the convergence of the Jacobi algorithm with asymmetric communication. The simulation was conducted for the measurement graph \mathbf{G}_1 and communication graph \mathbf{G}_1^c shown in Figure 1. Each communication edge was allowed fail at each iteration, with a probability of 0.1, independent of all other edges. The algorithm converges to an unbiased estimate $\hat{\mathbf{x}}^\infty$ whose variance is larger than the that of the BLU estimate.

The answer to this question will depend on the precise characterization of what is meant by 'distributed'. If we restrict our attention to algorithms in which the information exchange is purely local, i.e., a node is allowed to have information about only those measurements that are incident on itself and is allowed to broadcast its current estimate only to its communication-neighbors, then the answer is no. This can be seen by the example shown in Figure 3, where the reference variable is $x_1 = 0$ and the measurement error variances are all equal to one another. Consider the case when the communication graph (shown in Figure 3) is time-invariant. It satisfies the conditions of Theorem 1, and therefore the Jacobi algorithm will converge. However, due to the asymmetry in the communication graph, the limiting estimate will be different from the optimal estimate. It is clear from the figure that due to the information flow structure imposed by the communication graph, node 2 will only have information of the reference variable, which is 0, and the measurement ζ_{12} . The optimal estimate of x_2 is, however, a combination of all three measurements: $\hat{x}_2^* = -\frac{2}{3}\zeta_{12} - \frac{1}{3}(\zeta_{13} - \zeta_{32})$. Clearly no distributed algorithm can converge to the optimal estimate, since information on ζ_{13} will never reach node 2. Even if nodes are allowed to transmit their neighbors' information in addition to their own, similar examples can be constructed that shows the impossibility of optimal estimation in the presence of communication asymmetry.

D. More measurements need not reduce error further

Another important effect of asymmetric communication is that using more measurements need not lead to more accurate estimates of all node variables, in the sense that the variance of some of the node variables' estimation error can in fact increase.

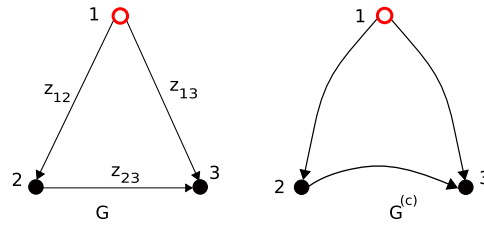


Fig. 3. A measurement graph and its associated communication graph.

When communication is symmetric, the Jacobi algorithm converges to the optimal estimate, which has the property that its estimation error variance can only decrease upon using more measurements. This follows from the so-called Rayleigh's monotonicity law of effective resistances; see [15] for details. We thus conclude that with symmetric communication, having more measurement edges, regardless of the associated error, produces more accurate (less variance) estimates when the Jacobi algorithm is used. However, the presence of asymmetry in the communication graph destroys this monotonicity. We illustrate this effect of asymmetry with an example, where the addition of a measurement edge makes the error variances of *all* the node estimates increase. Figure 4 shows two measurements graphs \mathbf{G}_1 and \mathbf{G}_2 and their associated communication graphs \mathbf{G}_1^c and \mathbf{G}_2^c . The measurement graph \mathbf{G}_1 contains all the nodes and edges of the measurement graph \mathbf{G}_2 . Similarly, \mathbf{G}_1^c contains all the nodes and edges in \mathbf{G}_1^c . Every measurement error variance in both the measurements graphs is unity. The estimation error variances of the limiting estimates $\hat{\mathbf{x}}^\infty$ computed from Theorem 2 are shown alongside the graphs. It is clear from the variances that the estimates in \mathbf{G}_1 are poorer than those in \mathbf{G}_2 , even though \mathbf{G}_1 contains more measurements than \mathbf{G}_2 .

V. SUMMARY

We examined the behavior of recently proposed distributed algorithms for network-wide clock synchronization under asymmetric communication between neighboring nodes. Clock synchronization consists of estimating the skews and offsets of the clocks w.r.t. a global reference. We showed that in the presence of asymmetry, these algorithms converge to a unbiased but sub-optimal estimate, provided certain conditions on the communication graph are satisfied. An example was provided to show that it is impossible to converge to the optimal estimate with a distributed algorithm. We showed that when communication is asymmetric, the error variance of the resulting estimate may actually increase upon considering more measurements. This is in contrast to the symmetric communication case where more measurements lead to more accurate estimates. This raises several questions that point to new research directions, such as, when should one not use certain measurements, and how can the nodes collaboratively decide which measurements should be excluded.

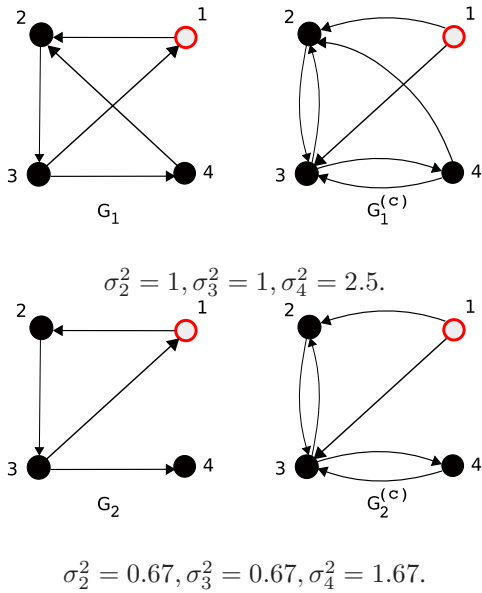


Fig. 4. Two measurements graphs \mathbf{G}_1 and \mathbf{G}_2 , and their associated communication graphs \mathbf{G}_1^c and \mathbf{G}_2^c . The variance of every measurement error is 1 for both the measurement graphs. The estimation error variances of the limiting estimate (that the Jacobi algorithm converges to) computed from Theorem 2 are shown alongside the graphs. Even though $\mathbf{G}_2 \subset \mathbf{G}_1$ and $\mathbf{G}_2^c \subset \mathbf{G}_1^c$, the resulting estimation error variances are still higher in $(\mathbf{G}_1, \mathbf{G}_1^c)$ than in $(\mathbf{G}_2, \mathbf{G}_2^c)$.

REFERENCES

- [1] R. Naik, S. Biswas, and S. Datta, "Distributed sleep-scheduling protocols for energy conservation in wireless networks," in *38th Hawaii International Conference on System Sciences*, 2005.
- [2] R. Solis, V. S. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proc. of the 45th IEEE Conference on Decision and Control*, December 2006.
- [3] P. Barooah, "Estimation and control with relative measurements: Algorithms and scaling laws." Ph.D. dissertation, University of California, Santa Barbara, July 2007.
- [4] B. M. Sadler and A. Swami, "synchronization in sensor networks: an overview," in *IEEE MILCOM*, October 2006.
- [5] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.
- [6] R. Karp, J. Elson, D. Estrin, and S. Shenker, "Optimal and global time synchronization in sensor networks," Center for Embedded Networked Sensing, Univ. of California, Los Angeles, Tech. Rep., 2003.
- [7] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.
- [8] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.
- [9] P. Barooah and J. P. Hespanha, "Distributed optimal estimation from relative measurements," in *Proceedings of the 3rd International Conference on Intelligent Sensing and Information Processing (ICISIP)*, December 2005, pp. 226–231.
- [10] P. Barooah, N. M. da Silva, and J. P. Hespanha, "Distributed optimal estimation from relative measurements for localization and time synchronization," in *Distributed Computing in Sensor Systems DCOSS*, ser. LNCS, P. B. Gibbons, T. Abdelzaher, J. Aspnes, and R. Rao, Eds. Springer, 2006, vol. 4026, pp. 266 – 281.
- [11] A. Giridhar and P. R. . Kumar, "Distributed time synchronization in wireless networks: Algorithms and analysis (I)," in *45th IEEE Conference on Decision and Control*, December 2006.
- [12] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications and Control*, A. V. Oppenheim, Ed. Prentice Hall, 1995.

- [13] C. Godsil and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2001.
- [14] A. Frommer and D. Szyld, "On asynchronous iterations," *Journal of Comp. Appl. Math.*, 123, pp. 201–216, 2000.
- [15] P. Barooah and J. P. Hespanha, "Estimation from relative measurements : Algorithms and scaling laws," *IEEE Control Systems Magazine*, 2007, in press.
- [16] M. Fiedler, *Special Matrices and their Applications in Numerical Mathematics*. Martinus Nijhoff Publishers, 1986.
- [17] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, ser. Computer Science and Applied Mathematics. Academic Press, 1979.
- [18] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, February 2005.

APPENDIX I TECHNICAL PROOFS

In the proofs, we will use the following result due to Fiedler, where \succeq (\succ) is used to denote entry-wise ordering. In addition, $\rho(X)$ denotes the spectral radius of a matrix X .

Theorem 3 (Theorems 5.1 and 5.12 of [16]): Let X be a real square $n \times n$ matrix whose off-diagonal terms are non-positive. Then, the following statements are equivalent:

- 1) There exists a splitting $X = Y - Z$ of the matrix X such that $Y^{-1} \succeq 0$, $Z \succeq 0$ and $\rho(Y^{-1}Z) < 1$.
- 2) X is a non-singular M-matrix. \square

The reader is referred to [17] for properties of M-matrices.

Proof of Lemma 1. Consider the case when the communication graph is time-invariant and the Jacobi algorithm operates in a synchronous fashion. It can be deduced from (12) that the error at the i -th iteration $\mathbf{e}^{(i)} := \hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^*$ evolves according to :

$$\mathbf{e}^{(i+1)} = M^{-1}N\mathbf{e}^{(i)} \quad (18)$$

Due to the structure of the matrices M^{-1} and N , (18) implies that in every iteration, each node $u \in \mathbf{V}$ computes its new state as the weighted average of the states of those nodes that have an edge with u directed toward u in the graph \mathbf{G}^c . In other words, it is a distributed average-consensus algorithm where the reference nodes keep their values at 0, and the remaining nodes try to reach consensus by averaging with their neighbors. The system (18) satisfies the strict convexity assumption of [18]. Thus, from Theorem 2 of [18] we know that the system (18) is uniformly globally attractive with respect to the equilibrium (which in this case is 0) if and only if for every node $u \in \mathbf{V} \setminus \mathbf{V}_r$, there is at least one reference node such that there is a directed path in \mathbf{G}^c from the reference node to u . Note that here we have used a slight specialization of the results in [18] to the case when one or more agents do not participate in the consensus algorithm but keep their values fixed.

It is also easy to see that the iteration (18) converges to 0 if and only if $\rho(M^{-1}N) < 1$. Since $M^{-1} \succeq 0$ (follows from Assumption 1) and $N \succeq 0$, it follows from Theorem 3 that non-singularity of L_c is equivalent to $\rho(M^{-1}N) < 1$, which was shown to be equivalent to the path existence condition. This proves the lemma. \blacksquare