Distributed Time Synchronization in Lossy Wireless Sensor Networks ⋆

Miloš S. Stanković * **Srdjan S. Stanković** ** **Karl Henrik Johansson** *

* *ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, 100-44 Stockholm, Sweden (e-mail: milsta@kth.se, kallej@kth.se).*
** *Faculty of Electrical Engineering, University of Belgrade, Belgrade, Serbia (e-mail: stankovic@etf.rs).*

Abstract: In this paper a new distributed time synchronization algorithm is proposed for lossy wireless sensor networks with noisy local clock readings and inter-node communications. The algorithm is derived in the form of two asynchronous recursions of stochastic gradient type providing estimates of the parameters used to compensate drifts and offsets of the local clocks. A special modification of the algorithm for drift compensation based on instrumental variables is introduced in the case of internal measurement noise. It is proved that the proposed algorithm provides asymptotic synchronization in the sense that all the equivalent drifts, as well as all the equivalent offsets, converge in the mean square sense and with probability one to the same random variables.

1. INTRODUCTION

Recently, sensor networks have emerged as an important research area from the point of view of both theory and practice. Wireless sensor networks (WSNs) are networks having nodes in the form of programmable devices with local computational and sensing capabilities, communicating with their neighbors via wireless channels. One of the natural requirements for WSNs, very important in many applications, is global *time synchronization*, *i.e.*, all the nodes have to share a *common notion of time*. The problem of time synchronization in WSNs has attracted a great deal of attention; it represents a challenge, having in mind multi-hop communications, unpredictable packet losses and high probability of node failures. There are numerous approaches to time synchronization in WSNs starting from different assumptions, *e.g.*, Freris et al. (2011, 2009); Carli et al. (2011); Xia and Cao (2011); Ganeriwal et al. (1999); Elson et al. (2002). An important class of time synchronization algorithms is based on full distribution of functions, when there are no reference nodes and when all the nodes run the same algorithm, such as in, *e.g.*, Simeone et al. (2008); Solis et al. (2006). Recently, there have been attempts to apply distributed gradient optimization and consensus schemes in different forms, *e.g.*, Sommer and Wattenhofer (2009); Schenato and Fiorentin (2011); Carli et al. (2011); Li and Rus (2006).

In this paper we propose a distributed algorithm for time synchronization in lossy WSNs characterized by internal local noise and communication dropouts by using two asynchronous recursions of stochastic gradient type. In the noiseless case, the basic algorithm is obtained in the form resembling to the one presented in Schenato and Fiorentin (2011). It is proved under general conditions related to the communication protocol and the network topology that the asymptotic synchronization is achieved exponentially in the sense of having the same equivalent drifts and the same equivalent offsets. In the case when the local time readings are corrupted by internal noise, a synchronization algorithm is derived in the form of two asynchronous stochastic approximation recursions. The recursion for drift estimation is based on the introduction of specific *instrumental variables* aimed at noise decorrelation. The recursion for offset estimation is autonomous, based on the result of drift estimation. The proof of convergence of the estimates in the mean square sense and with probability one is derived using stochastic approximation arguments, taking into account asynchronous features of the recursions and time varying properties of the weighted Laplacian of the underlying directed graph. It is also shown how the proposed methodology can be extended to the case of random communication dropouts. Illustrative simulation results are also given.

⋆ This work was supported by the Knut and Alice Wallenberg Foundation, the Swedish Research Council, and the Swedish Strategic Research Foundation.

2. NOISELESS CASE

2.1 Basic Algorithm

Assume a sensor network containing n nodes, formally represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ the set of arcs. Assume that $\Gamma^a = [\gamma_{ij}^a]$, $\gamma_{ij}^a = \gamma_{ij} > 0$, $i \neq j$, $\gamma_{ii}^a = 0$, is the weighted adjacency matrix of $\mathcal{G}$ and $\Gamma = -\text{diag}\{\sum_{j=1}^n \gamma_{1j}, \dots, \sum_{j=1}^n \gamma_{nj}\} + \Gamma^a$ its Laplacian matrix; γ_{ij} represent *a priori* given positive weights. Assume also that $\mathcal{N}_i^+$ is the out-neighborhood and $\mathcal{N}_i^-$ the in-neighborhood of the node i, containing the tail nodes of the arcs leaving the node i and the head nodes of the arcs entering the node i, respectively.

Assume that the node i has its local clock providing local time readings

$$\tau_i(t) = \alpha_i t + \beta_i, \tag{1}$$

where $t \in \mathcal{R}$ is the absolute time, α_i the local clock drift and β_i the local clock offset, $i = 1, \ldots, n$. We assume that the time readings (1) are locally transformed by an affine transformation

$$\bar{\tau}_i(t) = a_i \tau_i(t) + b_i, \tag{2}$$

where a_i and b_i are parameters to be determined. The function (2) resembles the calibration function in sensor calibration problems presented in Stanković et al. (2012b); $\bar{\tau}_i(t)$ can also be treated as an estimate of a virtual reference time Schenato and Fiorentin (2011). From (2) we have

$$\Delta\bar{\tau}_i(t) = \bar{\tau}_i(t) - \bar{\tau}_i(t - \Delta t) = a_i \Delta\tau_i(t), \tag{3}$$

where $\Delta\tau_i(t) = \tau_i(t) - \tau_i(t - \Delta t) = \alpha_i \Delta t$.

Estimation of the parameters in (2) will be based on the *pseudo periodic broadcast* described in Schenato and Fiorentin (2011). Namely, we assume that each node j, $j = 1, \ldots, n$, transmits a packet to its neighbors at discrete time instants $t_k^j \in \mathcal{R}$, $k = 0, 1, 2, \ldots$, defined recursively by

$$t_k^j = t_{k-1}^j + T_k^j \tag{4}$$

where $T_{min} < T_k^j < T_{max}$, $T_{min}, T_{max} > 0$, starting from a given initial time t_0^j. Procedurally, at each time instant t_k^j, $j = 1, \ldots, n$, the j-th clock sends a packet to its neighbors containing a message about its status (to be specified below). After receiving the packet, each neighbor compares the received data with its own status, and updates the estimates of its own parameters in (2). According to Schenato and Fiorentin (2011), it will be adopted in the analysis that there is no delay in the packet reception. The main idea is to construct an algorithm for estimating the parameters in (2) able to achieve asymptotic consensus about the modified time readings in the sense that all the *resulting drifts* $g_i = a_i \alpha_i$ and the *resulting offsets* $f_i = a_i \beta_i + b_i$ become equal.

Starting from the general idea presented in Stanković et al. (2012b) in relation with sensor calibration, the estimation algorithm for a_i will be derived from the following set of instantaneous local criteria

$$J_i^a(t_k^j) = \sum_{j \in \mathcal{N}_i^-(t_k^j)} \gamma_{ij} [\Delta\bar{\tau}_j(t_k^j) - \Delta\bar{\tau}_i(t_k^j)]^2 \tag{5}$$

where $\Delta\bar{\tau}_j(t_k^j) = \bar{\tau}_j(t_k^j) - \bar{\tau}_j(t_{k-1}^j) = a_j \alpha_j T_k^j$ and $\Delta\bar{\tau}_i(t_k^j) = a_i \alpha_i T_k^j$, $i, j = 1, \ldots, n$, $\mathcal{N}_i^-(t_k^j)$ is the set of nodes sending their packets to the i-th node at t_k^j, while γ_{ij} are positive weights reflecting relative importance of the nodes in $\mathcal{N}_i^-$. After calculating the gradient w.r.t. a_i, the following updating is done at the i-th node for each t_k^j:

$$\hat{a}_i(t_k^{j+}) = \hat{a}_i(t_k^j) + \delta_i(t_k^j) \sum_{j \in \mathcal{N}_i^-(t_k^j)} \gamma_{ij} [\Delta\hat{\tau}_j(t_k^j) - \Delta\hat{\tau}_i(t_k^j)] \Delta\tau_i(t_k^j) \tag{6}$$

where $\delta_i(t_k^j)$ are positive weights influencing convergence of the algorithm, $\Delta\hat{\tau}_j(t_k^j) = \hat{a}_j(t_k^j)\Delta\tau_j(t_k^j) = \hat{a}_j(t_k^j)\alpha_j T_k^j$ and $\Delta\hat{\tau}_i(t_k^j) = \hat{a}_i(t_k^j)\Delta\tau_i(t_k^j) = \hat{a}_i(t_k^j)\alpha_i T_k^j$.

Following the same line of thought, estimation of b_i starts from

$$J_i^b(t_k^j) = \sum_{j \in \mathcal{N}_i^-(t_k^j)} \gamma_{ij} [\bar{\tau}_j(t_k^j) - \bar{\tau}_i(t_k^j)]^2, \tag{7}$$

resulting into the updating rule

$$\hat{b}_i(t_k^{j+}) = \hat{b}_i(t_k^j) + \delta_i(t_k^j) \sum_{j \in \mathcal{N}_i^-(t_k^j)} \gamma_{ij} (\hat{\tau}_j(t_k^j) - \hat{\tau}_i(t_k^j)), \tag{8}$$

where $\hat{\tau}_j(t_k^j) = \hat{a}_j(t_k^j)\tau_j(t_k^j) + \hat{b}_j(t_k^j)$ and $\hat{\tau}_i(t_k^j) = \hat{a}_i(t_k^j)\tau_i(t_k^j) + \hat{b}_i(t_k^j)$.

Consequently, at each time instant t_k^j, $j = 1, \ldots, n$, the packet sent by the j-th node to its neighbors contains both $\Delta\hat{\tau}_j(t_k^j)$ and $\hat{\tau}_j(t_k^j)$; after receiving the packets, the neighbors calculate the corresponding $\Delta\hat{\tau}_i(t_k^j)$ and $\hat{\tau}_i(t_k^j)$ and update their parameters according to (6) and (8). Notice that the recursion (6) is structurally different from the corresponding recursion in Schenato and Fiorentin (2011); its advantages will be clearly perceivable in the noisy case. Also, the introduced weights γ_{ij} allows achieving additional adaptivity to the desired network characteristics.

In order to represent the whole algorithm in a compact form more suitable for further analysis, introduce $n \times n$ matrix $\Gamma^{[j]} = -\text{diag}\{\gamma_{1j}, \ldots, \gamma_{nj}\} + \Gamma_c^{[j]}$, where $\Gamma_c^{[j]}$ is a matrix containing zeros everywhere except the j-th column which is equal to the j-th column of Γ^a. Defining $\hat{g}(t) = \text{col}\{\hat{g}_1(t), \ldots, \hat{g}_n(t)\}$ and $\hat{f}(t) = \text{col}\{\hat{f}_1(t), \ldots, \hat{f}_n(t)\}$, where $\hat{g}_i(t) = \hat{a}_i(t)\alpha_i$ and $\hat{f}_i(t) = \hat{a}_i(t)\beta_i + \hat{b}_i(t)$, we obtain

$$\hat{g}(t_k^{j+}) = \hat{g}(t_k^j) + \Delta(t_k^j) A \sum_{j \in \cup_i \mathcal{N}_i^-(t_k^j)} \Gamma^{[j]} \hat{g}(t_k^j), \tag{9}$$

where $\Delta(t_k^j) = \text{diag}\{\delta_1(t_k^j), \ldots, \delta_n(t_k^j)\}$, $A = (T_k^j)^2$ $\text{diag}\{\alpha_1^2, \ldots, \alpha_n^2\}$ and

$$\hat{f}(t_k^{j+}) = \hat{f}(t_k^j) + \Delta(t_k^j) \sum_{j \in \cup_i \mathcal{N}_i^-(t_k^j)} [C'(t_k^j))\Gamma^{[j]} \hat{g}(t_k^j) + \Gamma^{[j]} \hat{f}(t_k^j)], \tag{10}$$

where $C'(t_k^j) = \text{diag}\{t_k^j + \alpha_1\beta_1(T_k^j)^2, \ldots, t_k^j + \alpha_n\beta_n(T_k^j)^2\}$.

Let $\{t_m\}$, $m = 0, 1, 2, \ldots$, be the set of all discrete time instants of packet sending, obtained by ordering $\{t_k^1\}$ $\cup \{t_k^2\} \cdots \cup \{t_k^n\}$, $k = 1, 2, \ldots$, $i = 1, \ldots, n$. Then, it is possible to represent (9) and (10) by

$$\rho(m+1) = (I + B(m))\rho(m), \tag{11}$$

where $\rho(m) = [g(m)^T \; f(m)^T]^T = [\hat{g}(t_k^j)^T \; \hat{f}(t_k^j)^T]^T$ for $t_m = t_k^j$ and $B(m) = \begin{bmatrix} D(m)A\Gamma(m) & 0 \\ D(m)C(m)\Gamma(m) & D(m)\Gamma(m) \end{bmatrix}$ in which:

- $D(m) = \text{diag}\{d_1(m), \ldots, d_n(m)\}$, where $d_i(m) = \delta_i(t_m)$ and $\Gamma(m) = \Gamma^{[j]}$, with $t_m = t_k^j$ for some k and j;

- $C(m) = \text{diag}\{\mu(m)m + \alpha_1\beta_1 T_m^2, \ldots, \mu(m)m + \alpha_n\beta_n T_m^2\}$, where $\mu(m)$ is defined in such a way that $\mu(m)m = t_m$, $0 < \mu(m) \le c < \infty$, while $T_m = T_k^j$.

2.2 Convergence

We start from two basic assumptions concerning the choice of time instants $\{t_k^i\}$ and the network properties:

(A1) There exist constants p_i, $i = 1, \ldots, n$, $0 < p_i < 1$, $\sum_{i=1}^{n} p_i = 1$, such that it is possible to find for all indices m in $\{t_m\}$ and any $\varepsilon > 0$ such an $M_0 = M_0(\varepsilon)$ that for all $M > M_0$

$$|\frac{J_i(m, m+M)}{M+1} - p_i| < \varepsilon \tag{12}$$

where $J_i(m, m+M)$ is the number of packet transmissions of the i-th node in the interval $(m, m+M)$;

(A2) Graph $\mathcal{G}$ has a spanning tree.

Assumption (A1) is a formal expression for the requirement about the persistency of sending messages by particular nodes, while (A2) represents a standard assumption for networks aimed at achieving consensus (see, *e.g.*, Ren and Beard (2005)).

Also, we assume in this section that:

(A3) $d_i(m) = d =$ const.

Lemma 1. Let (A1) hold. Let $\bar{\Gamma}$ be a matrix obtained from Γ in such a way that γ_{ij} is replaced by $\gamma_{ij} p_j$. Then, it is possible for all m and any $\varepsilon > 0$ to find such an $M_0 = M_0(\varepsilon)$ that for all $M > M_0$

$$\|\frac{1}{M+1}\sum_{m=0}^{M}\Gamma(m) - \bar{\Gamma}\| < \varepsilon. \tag{13}$$

Proof: The proof follows directly from assumption (A1). Namely, by definition, $\bar{\Gamma} = \sum_{i=1}^{n} p_j \Gamma^{[j]}$, so that

$$\|\frac{1}{M+1}\sum_{m=0}^{M}\Gamma(m) - \bar{\Gamma}\| \leq \sum_{j=1}^{n}|\frac{J_i(m, m+M)}{M+1} - p_j|\|\Gamma^{[j]}\| \tag{14}$$

and the result directly follows. ■

Theorem 1. Under (A1)-(A3) algorithm (11) achieves synchronization in the sense that there exists such a $d_0 > 0$ that for all $d \leq d_0$ $\rho(m)$ tends to $\rho^* = [g^*\mathbf{1}^T \; f^*\mathbf{1}^T]^T$, where g^* and f^* are constants and $\mathbf{1} = [1 \cdots 1]^T$.

Proof: First, we analyze the autonomous recursion in (11) generating $g(m)$:

$$g(m+1) = (I + dA\Gamma(m))g(m). \tag{15}$$

Let $\Phi_1 = [\mathbf{1} \vdots \Phi_1^*]$, where span$(\Phi_1^*)$ = span$(A\bar{\Gamma})$; then, according to (A2) and Huang and Manton (2010), $\Phi_1^{-1}A\bar{\Gamma}\Phi_1 = \begin{bmatrix} 0 & 0 \\ 0 & \Gamma_g^* \end{bmatrix}$, where $(n-1)\times(n-1)$ matrix Γ_g^* is Hurwitz; similarly, we have that $\Phi_1^{-1}A\Gamma(m)\Phi_1 = \begin{bmatrix} 0 & \Gamma_{A1}(m)^* \\ 0 & \Gamma_{A2}(m)^* \end{bmatrix}$. Introduce $\tilde{g}(m) = \Phi_1^{-1}g(m) = \begin{bmatrix} \tilde{g}(m)^{[1]} \\ \tilde{g}(m)^{[2]} \end{bmatrix}$, where dim $(\tilde{g}(m)^{[1]}) = 1$; then, accordingly,

$$\begin{aligned} \tilde{g}(m+1)^{[1]} &= \tilde{g}(m)^{[1]} + d\Gamma_{A1}(m)^*\tilde{g}(m)^{[2]} \\ \tilde{g}(m+1)^{[2]} &= \tilde{g}(m)^{[2]} + d\Gamma_{A2}(m)^*\tilde{g}(m)^{[2]} \end{aligned} \tag{16}$$

(see Huang and Manton (2010)). Iterating the second recursion in (16) M times backwards, one obtains

$$\tilde{g}(m+1)^{[2]} = [I + d\bar{B}_2(m,M)^* + O(d^2, d^3, \ldots)]\tilde{g}(m-M)^{[2]} \tag{17}$$

where $\bar{B}_2(m,M)^* = \sum_{k=0}^{M}\Gamma_{A2}(m-k)^*$, while $O(d^2, d^3, \ldots)$ contains higher order terms in d. Let $\bar{B}(m,M) = \sum_{k=1}^{M} A\Gamma(m-k) = (M+1)A\bar{\Gamma} + \Delta\bar{B}(m,M)$; then,

$$\Phi_1^{-1}\bar{B}(m,M)\Phi_1 = (M+1)\begin{bmatrix} 0 & 0 \\ 0 & \Gamma_g^* \end{bmatrix} + \begin{bmatrix} 0 & \Delta\bar{B}_1(m,M)^* \\ 0 & \Delta\bar{B}_2(m,M)^* \end{bmatrix}, \tag{18}$$

wherefrom it follows that $\bar{B}_2(m,M)^* = (M+1)\Gamma_g^* + \Delta\bar{B}_2(m,M)^*$.

Define the Lyapunov function $V(m) = \tilde{g}(m)^{[2]T}P_g^*\tilde{g}(m)^{[2]}$, where $P_g^* > 0$ satisfies the Lyapunov equation $\Gamma_g^{*T}P_g^* + P_g^*\Gamma_g^* = -Q_g^*$, where $Q_g^* > 0$, having in mind that Γ_g^* is Hurwitz. Denote $\Delta Q_g(m,M)^* = \Delta\bar{B}_2(m,M)^{*T}P_g^* + P_g^*\Delta\bar{B}_2(m,M)^*$. It is possible to conclude that

$$\begin{aligned} V(m+1) \leq &[1 - d(M+1)\lambda_{min}(Q_g^*) \\ &+ \|\Delta Q_g(m,M)^*\| + O(d^2, d^3, \ldots)]V(m-M) \end{aligned} \tag{19}$$

(see Stanković et al. (2012b)). According to (A1) and Lemma 1, for M large enough $-(M+1)\lambda_{min}(Q_g^*) + \|\Delta Q_g(m,M)^*\| < 0$. Therefore, it is possible to find such $d_0 > 0$ that for all $d \leq d_0$ the whole multiplier of $V(m-M)$ at the right hand side of (19) is less than one (see Stanković et al. (2011)). Consequently, $\tilde{g}(m)^{[2]}$ tends to zero exponentially.

Coming back to the first relation in (16), one concludes easily that $\tilde{g}(m)^{[1]}$ converges to some constant g^*; by definition of $\tilde{g}(m)$, it follows that $\lim_{m\to\infty} g(m) = g^*\mathbf{1}$.

Consider now the second relation from (11)

$$f(m+1) = (I + d\Gamma(m))f(m) + dC(m)\Gamma(m))g(m). \tag{20}$$

Let $\Phi_2 = [\mathbf{1} \vdots \Phi_2^*]$, where span$(\Phi_2^*)$ = span$(\bar{\Gamma})$; then, according to (A2), $\Phi_2^{-1}\bar{\Gamma}\Phi_2 = \begin{bmatrix} 0 & 0 \\ 0 & \Gamma_f^* \end{bmatrix}$, where $(n-1)\times(n-1)$ matrix Γ_f^* is Hurwitz. Then, we also have $\Phi_2^{-1}\Gamma(m)\Phi_2 = \begin{bmatrix} 0 & \Gamma_1(m)^* \\ 0 & \Gamma_2(m)^* \end{bmatrix}$ and $\Phi_2^{-1}C(m)\Gamma(m)\Phi_1 = \begin{bmatrix} 0 & \Gamma_{C1}(m)^* \\ 0 & \Gamma_{C2}(m)^* \end{bmatrix}$. Defining $\tilde{f}(m) = \Phi_2^{-1}f(m) = \begin{bmatrix} \tilde{f}(m)^{[1]} \\ \tilde{f}(m)^{[2]} \end{bmatrix}$, we can apply the methodology presented in relation with the analysis of $g(m)$. Iterating back the recursion for $\tilde{f}^{[2]}(m)$, one can easily conclude that under the adopted assumptions $\|\tilde{f}(m)^{[2]}\|$ tends to zero as mr^m, where $|r| < 1$. Consequently, $\tilde{f}(m)^{[1]}$ tends to a constant f^*, so that $f(m)$ tends to $f^*\,\mathbf{1}$. Thus the result. ■

3. LOSSY NETWORKS

3.1 Internal Noise

Algorithm. We assume now that the clock readings figuring in the above algorithms are corrupted by internal noise. In this case, we have the "noisy versions" $\tau_j^{\xi}(t_k^j) = \alpha_j t_k^j + \beta_j + \xi_j(t_k^j)$, $\Delta\tau_j^{\xi}(t_k^j) = \alpha_j T_k^j + \Delta\xi_j(t_k^j)$ and $\Delta\tau_i^{\xi}(t_k^j) = \alpha_i T_k^j + \Delta\xi_i(t_k^j)$ instead of $\tau_j(t_k^j)$, $\Delta\tau_j(t_k^j)$ and $\Delta\tau_i(t_k^j)$, where $\{\xi_j(t_k^j)\}$ are supposed to be zero mean white noise sequences with constant variances σ_j^2, while $\Delta\xi_i(t_k^j) = \xi_i(t_k^j) - \xi_i(t_{k-1}^j)$, $j = 1, \dots, n$; physically, these sequences represent all errors connected with local clock readings, including thermal noise, signal processing noise, as well as computation and digital representation noise.

The estimation algorithms in the noisy case can be constructed directly starting from (6) and (8) by introducing $\Delta\hat{\tau}_j^\xi(t_k^j) = \hat{a}_j(t_k^j)\Delta\tau_j^\xi(t_k^j)$, $\Delta\hat{\tau}_i^\xi(t_k^j) = \hat{a}_i(t_k^j)\Delta\tau_i^\xi(t_k^j)$ and $\hat{\tau}_j^\xi(t_k^j) = \hat{a}_j(t_k^j)\tau_j^\xi(t_k^j) + \hat{b}_j(t_k^j)$ instead of $\Delta\hat{\tau}_j(t_k^j)$, $\Delta\hat{\tau}_j(t_k^j)$ and $\hat{\tau}_j(t_k^j)$. However, this leads in (6) to correlation between the noise terms in $\Delta\hat{\tau}_j^\xi(t_k^j)$ and $\Delta\tau_j^\xi(t_k^j)$, which prevents the achievement of the desired consensus. In order to circumvent this problem, we propose the following modified version of (6)

$$\hat{a}_i(t_k^{j+}) = \hat{a}_i(t_k^j) + \delta_i(t_k^j) \sum_{j \in \mathcal{N}_i^-(t_k^j)} \gamma_{ij}[\Delta\hat{\tau}_j^\xi(t_k^j) - \Delta\hat{\tau}_i^\xi(t_k^j)]\Delta\tau_i^\xi(t_{k-2}^j), \quad (21)$$

in which $\Delta\tau_i^\xi(t_{k-2}^j)$ is an *instrumental variable* (see, *e.g.*, Ljung and Söderström (1983)). Notice that in this case $\Delta\xi_j(t_k^j)$ and $\Delta\xi_j(t_{k-2}^j)$ become uncorrelated (a similar idea has been applied in the context of sensor calibration in Stanković et al. (2012a)). Having in mind that $\Delta\xi_j(t_k^j)$ and $\xi_j(t_{k-1}^j)$ are correlated, parameter updating in (21) is done at $\dots, t_{k-4}^j, t_k^j, t_{k+4}^j, \dots$.

The updating scheme (8) will now be supposed to use the final results of (21):

$$\hat{b}_i(t_k^{j+}) = \hat{b}_i(t_k^j) + \delta_i(t_k^j) \sum_{j \in \mathcal{N}_i^-(t_k^j)} \gamma_{ij}(\hat{\tau}_j^\xi(t_k^j)^* - \hat{\tau}_i^\xi(t_k^j)^*), \quad (22)$$

where $\hat{\tau}_j^\xi(t_k^j)^* = \hat{\tau}_j^\xi(t_k^j)|_{\hat{a}_i(t_k^j)=\hat{a}_i^*}$, $\hat{a}_i^*$ being the value of $\hat{a}_i(t_k^j)$ at convergence.

We can put (21) and (22) in a compact form similar to (11). The procedure is based on introducing the expressions for $\Delta\hat{\tau}_j^\xi(t_k^j)$, $\Delta\tau_j^\xi(t_k^j)$ and $\hat{\tau}_j^\xi(t_k^j)$. Taking $t_m = t_k^j$, one obtains (compare with (11))

$$\rho(m+1) = (I + B_1^\xi(m))\rho(m) + B_2^\xi(m)g^*, \quad (23)$$

where g^* is the value of $g(m)$ at convergence, $B_1^\xi(m) = \mathrm{diag}\{D(m)A^\xi\Gamma(m) + D(m)\Xi^{[1]}(m),\ D(m)\Gamma(m)\}$ and $B_2^\xi(m) = \begin{bmatrix} 0 \\ \hdashline D(m)[\Xi^{[1]}(m) + \Xi^{[2]}(m)] \end{bmatrix}$ with:

- $A^\xi = T_k^j T_{k-2}^j \mathrm{diag}\{\alpha_1^2, \dots, \alpha_n^2\}$,
- $\Xi^{[1]}(m) = \Xi_1(m)\Xi_2(m) + A^\xi\Xi_3(m)$, with $\Xi_1(m) = \mathrm{diag}\{\alpha_1 T_{k-2}^j + \Delta\xi_1(t_{k-2}^j), \dots, \alpha_n T_{k-2}^j + \Delta\xi_n(t_{k-2}^j)\}$, $\Xi_2(m) = \begin{bmatrix} -\Delta\xi_1(t_k^j)\gamma_{1j} & \cdots & \frac{\alpha_1}{\alpha_j}\Delta\xi_j(t_k^j)\gamma_{1j} & \cdots & 0 \\ 0 & & \vdots & & 0 \\ 0 & \cdots & \frac{\alpha_n}{\alpha_j}\Delta\xi_j(t_k^j)\gamma_{nj} & \cdots & -\Delta\xi_n(t_k^j)\gamma_{nj} \end{bmatrix}$ and $\Xi_3(m) = A^\xi \mathrm{diag}\{\Delta\xi_1(t_{k-2}^j), \dots, \Delta\xi_n(t_{k-2}^j)\}\Gamma(m)$,
- $\Xi^{[2]}(m) = \begin{bmatrix} -\xi_1(t_k^j)\gamma_{1j} & \cdots & \xi_j(t_k^j)\gamma_{1j} & \cdots & 0 \\ 0 & & \vdots & & 0 \\ 0 & & \cdots \xi_j(t_k^j)\gamma_{nj} & \cdots & -\xi_n(t_k^j)\gamma_{nj} \end{bmatrix}$.

Notice that the sequences $\{\Xi_1(m)\}$ and $\{\Xi_2(m)\}$ are uncorrelated.

Convergence. In order to avoid centralized evidence of the current iteration number m distributed to all the nodes, we assume that $d_i(m) = d(I_i(m))$, where $I_i(m)$ represents the local index defined by the number of updates done by the i-th node up to the instant t_m, where, for the sake of simplicity, the function $d(.)$ is chosen to be the same for all the nodes. In general, such an algorithm belongs to the class of asynchronous stochastic approximation algorithms, *e.g.* Borkar (1998); Chen (2002). It is possible to simplify the analysis after noticing that, according to (A1) and (A2), $I_i(m)$ becomes for m large enough close to $k_i m$, where k_i is a positive number defined by $k_i = \sum_{j \in \mathcal{N}_i^-} p_j$, so that we have $d_i(m) \sim d(k_i m)$. We assume further that

(A4) $d(m) = \frac{k}{m^\lambda}$, where $\frac{1}{2} < \lambda \leq 1$, $k > 0$.

Therefore, for m large enough $d_i(m) \sim \frac{k_i'}{m^\lambda}$, $k_i' > 0$, or $D(m) \sim K' \frac{1}{m^\lambda}$, where $K' = \mathrm{diag}\{k_1', \dots, k_n'\}$.

Theorem 2. Let assumptions (A1), (A2) and (A4) be satisfied. Then the algorithm (23) achieves synchronization in the sense that $g(m)$ tends to $w_g^*\mathbf{1}$ and $f(m)$ tends to $w_f^*\mathbf{1}$ in the mean square sense and with probability one, where w_g^* and w_f^* are random variables.

Proof: We shall first focus our attention on the recursion for $g(m)$ in (23). Following the methodology of the proof of Theorem 1, we construct $\tilde{g}(m) = \Phi_1^{-1}g(m)$ and obtain for m large enough

$$\begin{aligned} \tilde{g}(m+1)^{[1]} =& \tilde{g}(m)^{[1]} + d(m)(\Gamma_{A1}^\xi(m)^*\tilde{g}(m)^{[2]} \\ &+ \Psi_1(m)^{[1]}\tilde{g}(m)^{[1]} + \Psi_1(m)^{[2]}\tilde{g}(m)^{[2]}), \\ \tilde{g}(m+1)^{[2]} =& \tilde{g}(m)^{[2]} + d(m)(\Gamma_{A2}^\xi(m)^*\tilde{g}(m)^{[2]} \\ &+ \Psi_2(m)^{[1]}\tilde{g}(m)^{[1]} + \Psi_2(m)^{[2]}\tilde{g}(m)^{[2]}), \end{aligned} \quad (24)$$

where $\Gamma_{A1}^\xi(m)^*$ and $\Gamma_{A2}^\xi(m)^*$ are obtained from $\Phi_1^{-1} K'A^\xi \Gamma(m)\Phi_1$ in the same way as $\Gamma_{A1}(m)^*$ and $\Gamma_{A2}(m)^*$ in Theorem 1 (see (16)), while $\Phi_1^{-1}K'\Xi^{[1]}(m)\Phi_1 = \begin{bmatrix} \Psi_1(m)^{[1]} \\ \hdashline \Psi_2(m)^{[1]} \end{bmatrix}$ and $\Phi_1^{-1}K'\Xi^{[2]}(m)\Phi_1 = \begin{bmatrix} \Psi_1(m)^{[2]} \\ \hdashline \Psi_2(m)^{[2]} \end{bmatrix}$ are white noise terms independent of $g(m)$.

Define $s(m) = E\{(\tilde{g}(m)^{[1]})^2\}$ and $V(m) = E\{\tilde{g}(m)^{[2]T} P_g^\xi\ \tilde{g}(m)^{[2]}\}$, where $P^\xi > 0$. After iterating back the first recursion in (24) up to the initial condition, one concludes in a straightforward way that

$$s(m+1) \leq C_0(1 + \sum_{k=1}^m d(m)^2 V(k)), \quad (25)$$

where C_0 is a generic constant, having in mind (A4) and the fact that the noise terms in (16) are mutually independent (notice that $\sum_{i=1}^{\infty} d(i)^2 < \infty$). The next step is to iterate back the second recursion in (24) M times backwards and to calculate $V(m)$ under the assumption that $P_g^* > 0$ satisfies the Lyapunov equation $\Gamma_g^{\xi T} P_g^{\xi} + P_g^{\xi}\Gamma_g^{\xi} = -Q_g^{\xi}$, where $Q_g^{\xi} > 0$, where Γ_g^{ξ} is a matrix obtained from $\bar{B}^{\xi}(m, M) = \sum_{k=1}^{M} K' A^{\xi}\Gamma(m-k)$ in the same way as Γ_g^* is obtained from $\bar{B}(m, M)$ in relation with (19) in the proof of Theorem 1; this matrix is Hurwitz in the same way as Γ_g^*. Consequently,

$$V(m+1) \leq (1 - c_0 d(m))V(m-M) \qquad (26)$$
$$+ C_1 \sum_{\sigma=m-M}^{m} d(\sigma)^2(1 + s(\sigma) + V(\sigma)),$$

where $c_0 > 0$ and C_1 are generic constants. Relations (25) and (26) can now be treated using Theorem 11 and Lemma 12 from Huang and Manton (2010). The conclusion is that $s(m)$ converges and that $V(m)$ converges to zero. Using again Huang and Manton (2010) one concludes that $\tilde{g}(m)^{[2]}$ tends to zero and $\tilde{g}(m)^{[1]}$ to a random variable w_g^* in the mean square sense and w. p. 1; consequently, $g(m)$ tends to $w_g^*\mathbf{1}$ in the mean square sense and w. p. 1. Using similar arguments, one can also derive that $f(m)$ tends to $w_f^*\mathbf{1}$ in the mean square sense and w.p. 1. ■

Remark 1. Recursions (21) and (22) are not coupled (like (6) and (8)) in order to ensure convergence of offset estimates; the rate of convergence of $g(m)$ in the stochastic case is insufficient for achieving convergence of $f(m)$.

3.2 Communication Dropouts

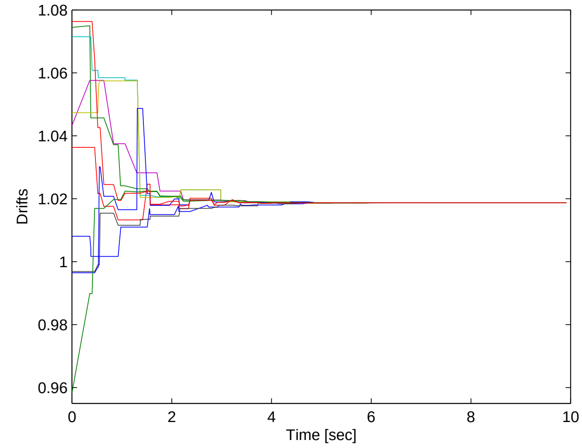
The above proposed algorithms are applicable in the case of random *communication dropouts.*

(A5) Communication gains are represented by a randomly time varying adjacency matrix $\Gamma^a(t_k^j) = [\gamma_{ij}^a(t_k^j)]$, where $\gamma_{ij}^a(t_k^j) = \zeta_{ij}(t_k^j)\gamma_{ij}$ (the gains γ_{ij}, $i, j = 1, \ldots, n$, are defined in Section II) and the sequences $\{\zeta_{ij}(t_k^j)\}$, $j = 1, \ldots, n$, are stationary independent binary random sequences with $P\{\zeta_{ij}(t_k^j) = 1\} = \pi_{ij}$, $0 \leq \pi_{ij} \leq 1$.

Theorem 3. Let assumptions (A1), (A2), (A4) and (A5) be satisfied. Then the algorithm (23) with $\Gamma(m) = \Gamma^{[j]}(t_k^j)$, where $\Gamma^{[j]}(t_k^j) = \mathrm{diag}\{\zeta_{1j}(t_k^j), \ldots, \zeta_{nj}(t_k^j)\}\Gamma^{[j]}$, achieves synchronization in the same sense as in Theorem 2.

Proof: The main idea of the proof lies in treating the recursion (23) generated under (A5) using the methodology of the proof of Theorem 2. Namely, defining $\bar{\Gamma}(m) = E\{\Gamma^{[j]}(t_k^j)\} = \mathrm{diag}\{\pi_{1j}, \ldots, \pi_{nj}\}\Gamma^{[j]}$, we can write $\Gamma(m) = \bar{\Gamma}(m) + \Delta\Gamma(m)$, where $\{\Delta\Gamma(m)\}$ is a zero mean white noise sequence. After replacing the last relation into (23), the obtained recursion can be treated in the same way as the recursion (23) in the case of the internal noise. ■

Remark 2. In general, both algorithms (11) and (23) are applicable in the case of communication dropouts which influence only the convergence rate.



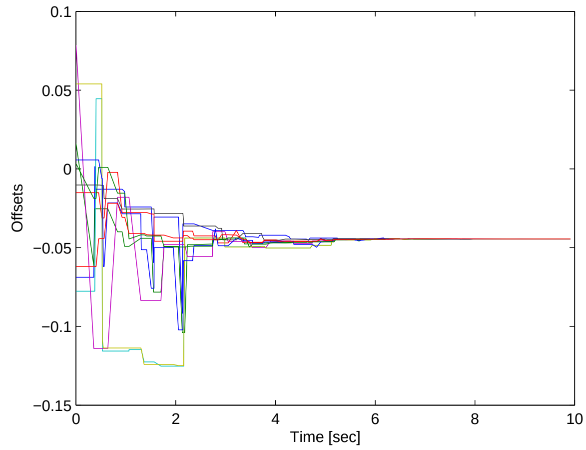


Fig. 1. Drift and offset estimates: deterministic case

4. SIMULATIONS

Two simulation experiments are presented as illustrations of the above theoretical results. In the first, a network of ten nodes forming a directed graph has been simulated, with α_i and β_i randomly chosen from the intervals $(0.95, 1.05)$ and $(-0.05, 0.05)$, respectively. Values of T_k^j in (4) have been randomly chosen within $(0, 1)$. Fig. 1 gives the drift and offset estimates obtained by using the proposed algorithm. Fig. 2 gives the results obtained at two nodes in the noisy case, with $\sigma_i = 0.1$. Fig. 2.a gives the drift estimates obtained by using (21) which incorporates instrumental variables, while Fig. 2.b corresponds to (6) with decreasing sequences $d_i(m)$ satisfying (A4). It is obvious that in the latter case consensus cannot be achieved as a result of noise correlation. Fig. 2.c gives the corresponding offset estimates obtained by (22). In comparison, we have found that the algorithm from Schenato and Fiorentin (2011) shows a similar behavior in the deterministic case, but possesses a very poor noise immunity. Notice also that coupling between (21) and (22) leads to an erratic behavior of the offset estimates, acceptable only in the case of low noise variances.

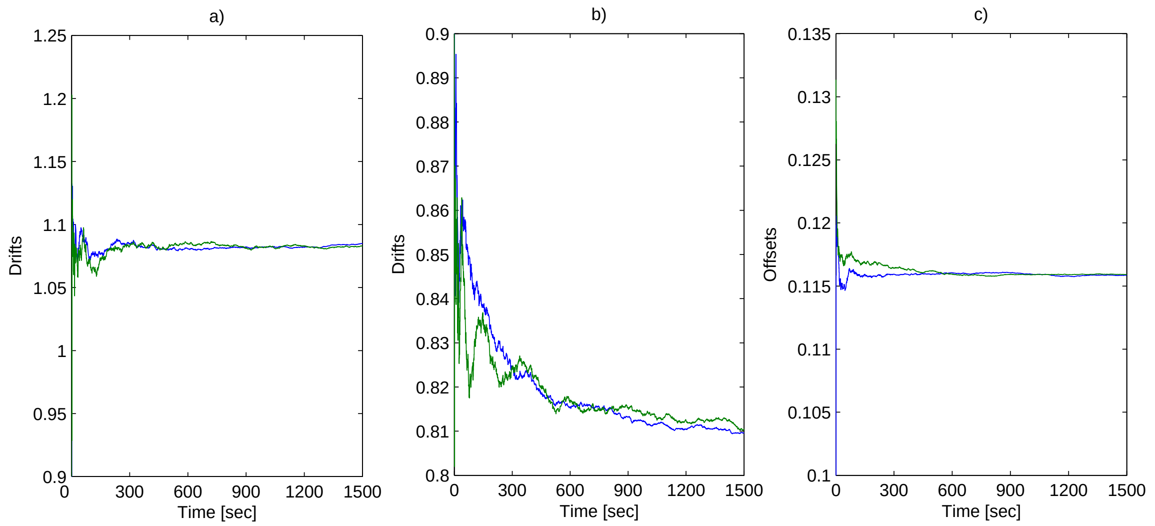


Fig. 2. Drift and offset estimates: stochastic case

5. CONCLUSION

In this paper an algorithm of stochastic gradient type is proposed for distributed time synchronization in lossy WSNs, using a pseudo periodic broadcast protocol. Starting from general assumptions related to the properties of the network and the communication protocol, it has been proved that the proposed algorithm ensures asymptotic consensus with respect to the equivalent drifts and offsets of the local clocks. In the case of measurement noise a specific algorithm of instrumental variable type is proposed for drift estimation.

One of the immediate challenges is to investigate optimization of the algorithm with respect to the communication gains γ_{ij} and tracking time-varying drifts and offsets. It would be also interesting to construct similar gradient schemes which incorporate matrix gain sequences in order to achieve higher convergence rate.

REFERENCES

Borkar, V. (1998). Asynchronous stochastic approximation. *SIAM Journal Control Optim.*, 36, 840–851.

Carli, R., D'Elia, E., and Zampieri, S. (2011). A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks. In *Proc. IEEE Conf. Decision and Control*.

Chen, H.F. (2002). *Stochastic approximation and its applications*. Kluwer Academic, Dordrecht, the Netherlands.

Elson, J., Girod, L., and Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. In *Proc. Symp. Oper. Syst. Design and Implement.*, 147–163.

Freris, N.M., Borkar, V.S., and Kumar, P.R. (2009). A model-based approach to clock synchronization. In *Proc. IEEE Conf. Decision and Control*.

Freris, N.M., Graham, S.R., and Kumar, P.R. (2011). Fundamental limits on synchronizing clocks over networks. *IEEE Trans. Autom. Control*, 56, 1352–1364.

Ganeriwal, S., Kumar, R., and Srivastava, M. (1999). Next century challenges: scalable coordination in sensor networks. In *Proc. ACM/IEEE Intern. Conference on Mobile Computing and Networking*, 263–270.

Huang, M. and Manton, J.H. (2010). Stochastic consensus seeking with noisy and directed inter-agent communications: fixed and randomly varying topologies. *IEEE Trans. Autom. Control*, 55, 235–241.

Li, Q. and Rus, D. (2006). Global clock synchronization in sensor networks. *IEEE Trans. on Computers*, 55, 214–226.

Ljung, L. and Söderström, T. (1983). *Theory and pracice of recursive identification*. MIT Press, Cambridge, Mass.

Ren, W. and Beard, R. (2005). Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control*, 50, 655–661.

Schenato, L. and Fiorentin, F. (2011). Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks. *Automatica*, 47(9), 1878–1886.

Simeone, O., Spagnolini, U., Bar-Ness, Y., and Strogatz, S.H. (2008). Distributed synchronization in wireless networks. *IEEE Sign. Proc. Magazine*, 25, 81–97.

Solis, R., Borkar, V., and Kumar, P.R. (2006). A new distributed time synchronization protocol for multi hop wireless networks. In *Proc. IEEE Conf. Decision and Control*, 2734–2739.

Sommer, P. and Wattenhofer, R. (2009). Gradient clock synchronization in wireless sensor networks. In *Proc. Int. Conf. Inf. Proc. in Sensor Networks*, 37–48.

Stanković, M.S., Stanković, S.S., and Johansson, K.H. (2012a). Distributed calibration for sensor networks under communication errors and measurement noise. Submitted to *IEEE Conf. Decision and Control*.

Stanković, M.S., Stanković, S.S., and Johansson, K.H. (2012b). Distributed macro calibration in sensor networks. In *Proc. 20th Mediterranean Conference on Control and Automation*.

Stanković, S.S., Stanković, M.S., and Stipanović, D.M. (2011). Decentralized parameter estimation by consensus based stochastic approximation. *IEEE Trans. Autom. Control*, 47, 531–543.

Xia, W. and Cao, M. (2011). Synchronizing clocks in distributed networks. In *Proc. IEEE Conf. Decision and Control*.