A New Drift Correction Algorithm for Distributed Time Synchronization

Miloš S. Stanković, Srdjan S. Stanković and Karl Henrik Johansson

Abstract— **This paper proposes a new distributed asynchronous algorithm for drift correction in time synchronization for networks with random communication delays, measurement noise and communication dropouts. Three variants of the algorithm are proposed, based on different current local time increments. Under nonrestrictive conditions concerning network properties, it is proved that all the algorithms provide convergence in the mean square sense and with probability one of the corrected drifts of all the nodes to the same value (consensus). Asymptotic rate of convergence of the algorithms is formulated. It is shown that it is possible to achieve convergence to the common virtual clock. Simulation results give an illustration of the properties of the algorithms.**

I. Introduction

Cyber-Physical Systems (CPS), Internet of Things (IoT) and Sensor Networks (SN) have emerged as research areas of paramount importance with many conceptual and practical challenges and numerous applications [1], [2]. One of the basic requirements in networked systems, in general, is *time synchronization*, *i.e.*, necessity for all the nodes to share a common notion of time. The problem of time synchronization has attracted a lot of attention, but still represents a challenge due to multi-hop communications, stochastic delays, communication and measurement noise, unpredictable packet losses and high probability of node failures, *e.g.*, [3]. There are numerous approaches to time synchronization starting from different assumptions and using different methodologies, *e.g.*, [3], [4]. An important class of time synchronization algorithms is based on full distribution of functions [5], [6]. A class of *consensus based algorithms*, called CBTS (Consensus-Based Time Synchronization) algorithms has attracted considerable attention, *e.g.*, [7]–[10]. It has been treated in a unified way in a recent survey [11], providing figure of merit of the principal approaches. In [12] a control-based approach to distributed time synchronization has been adopted. Fundamental and yet unsolved problems in all time synchronization approaches are connected with communication delays and measurement noise; see [13] for basic issues.

In this paper we propose a new *asynchronous distributed* algorithm for *drift correction*, used for time synchronization in lossy networks characterized by *random communication delays*, *measurement noise* and *communication dropouts*. The algorithm is composed of a distributed recursion of *asynchronous stochastic approximation type* based on *broadcast gossip* and derived from predefined local error function. The recursion is aimed at achieving asymptotic consensus on the *corrected drifts* and, together with an *offset correction algorithm*, at obtaining *common virtual clock* for all the nodes in the network.

The proposed recursion for *drift synchronization* (presented in a preliminary form in [14]) is based on noisy time increments defined in three characteristic forms. We prove *convergence to consensus* of the corrected drifts in the mean square sense and with probability one (w.p.1) under nonrestrictive conditions. Furthermore, we provide an estimate of the corresponding *asymptotic convergence rate* to consensus. Compared to the existing analogous algorithms [8], [10], the proposed scheme is structurally different and simpler and provides the best convergence rate. Notice that the algorithm proposed in [8] cannot handle communication delays or measurement noise, while the paper [10] treats random delays, but not the case of measurement noise and communication dropouts.

The proposed drift correction algorithm is a good prerequisite for achieving finite differences between local virtual clocks in the means square sense and w.p.1. To the authors knowledge, the proposed algorithm represents the first method with such a performance in the case of random delays, measurement noise and communication dropouts.

Finally, two illustrative simulation results are presented.

II. Drift Correction Algorithms

A. Time and Network Models

Assume a network consisting of n nodes, formally represented by a directed graph $\mathscr{G}=(\mathscr{N},\mathscr{E})$, where $\mathscr{N}$ is the set of nodes and $\mathscr{E}$ the set of arcs defining the structure of inter-node communications. Denote by $\mathscr{N}_i^+$ the out-neighborhood and by $\mathscr{N}_i^-$ the in-neighborhood of node i, $i=1,\dots,n$. Assume that each node has a local clock, whose output, defining *local time*, is given for any *absolute time* $t\in\mathscr{R}$ by

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

where α_i is the local *drift (gain)*, β_i is the local *offset*, while $\xi_i(t)$ is *measurement noise*, appearing due to equipment instabilities, round-off errors, thermal noise, etc. [8], [15]. Each node i applies an affine transformation to $\tau_i(t)$, producing the *corrected local time*

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

where a_i and b_i are local *correction parameters*, $g_i=a_i\alpha_i$ is the *corrected drift* and $f_i=a_i\beta_i+b_i$ the *corrected offset*,

M. S. Stanković is with the Innovation Center, School of Electrical Engineering, University of Belgrade and the Vlatacom Institute, Belgrade Serbia (e-mail: milos.stankovic@ic.etf.rs).

S. S. Stanković is with the School of Electrical Engineering, University of Belgrade, Serbia and the Vlatacom Institute, Belgrade (e-mail: stankovic@etf.rs).

K. H. Johansson is with the ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: kallej@kth.se).

This work was supported by the EU Marie Curie CIG (PCIG12-GA-2012-334098), Knut and Alice Wallenberg Foundation, the Swedish Strategic Research Foundation and the Swedish Research Council.

$i = 1, \ldots, n$.

The goal of distributed time synchronization is to provide a *common virtual clock*, *i.e.*, *equal corrected drifts* g_i and *equal corrected offsets* f_i, $i = 1, \ldots, n$ by *distributed real-time estimation* of the parameters a_i and b_i. We assume that the nodes communicate according to the *broadcast gossip scheme*, *e.g.*, [16], without global supervision or fusion center. Namely, we assume that each node $j \in \mathcal{N}$ has its own *local communication clock* that ticks according to a Poisson process with the rate μ_j, independently of other nodes. At each tick of its communication clock (denoted by t_b^j, $b = 0, 1, 2, \ldots$), node j broadcasts its current local time (together with its current estimates of correction parameters) to its out-neighbors $i \in \mathcal{N}_j^+$. Each node $i \in \mathcal{N}_j^+$ hears the broadcast with probability $p_{ij} > 0$. Let $\{t_l^{j,i}\}$, $l = 0, 1, 2, \ldots$, be the sequence of absolute time instants corresponding to the messages heard by node i. The message sent at $t_l^{j,i}$ is received at node i at the time instant

$$\bar{t}_l^{j,i} = t_l^{j,i} + \delta_l^{j,i},$$

where $\delta_l^{j,i}$ represents the corresponding *communication delay*. See [17] for presentation of physical and technical sources of the delays. We assume in the sequel that the communication delay can be decomposed as

$$\delta_l^{j,i} = \bar{\delta}^{j,i} + \eta_i(t_l^{j,i}), \tag{3}$$

where $\bar{\delta}^{j,i}$ is assumed to be constant (depending only on the chosen arc (j,i)), while $\eta_i(t_l^{j,i})$ represents a stochastically time-varying component with zero mean. After receiving a message from node j, node i reads its current local time, calculates its own current *corrected local time* and *updates the values of its correction parameters* a_i and b_i. The process is repeated after each tick of the communication clock of any node in the network; we assume, as usually, that time is dense and only one communication clock can tick at a given time [16].

B. Algorithm

The recursion for updating the value of parameter a_i at node i, as a response to a message coming from node j, is based on the following *error function*

$$\bar{\varphi}_i^a(\bar{t}_l^{j,i}) = \Delta\bar{\tau}_j(t_l^{j,i}) - \Delta\bar{\tau}_i(\bar{t}_l^{j,i}), \tag{4}$$

where $\Delta\bar{\tau}_j(t_l^{j,i})$ and $\Delta\bar{\tau}_i(\bar{t}_l^{j,i})$ are *increments of the corrected local times*, given by

$$\Delta\bar{\tau}_j(t_l^{j,i}) = \bar{\tau}_j(t_l^{j,i}) - \bar{\tau}_j(t_m^{j,i}) = a_j \Delta\tau_j(t_l^{j,i}),$$

$$\Delta\bar{\tau}_i(\bar{t}_l^{j,i}) = \bar{\tau}_i(\bar{t}_l^{j,i}) - \bar{\tau}_i(\bar{t}_m^{j,i}) = a_i \Delta\tau_i(\bar{t}_l^{j,i}),$$

where $m \in \{0, \ldots, l-1\}$,

$$\Delta\tau_j(t_l^{j,i}) = \tau_j(t_l^{j,i}) - \tau_j(t_m^{j,i}) = \alpha_j \Delta t_l^{j,i} + \Delta\xi_j(t_l^{j,i}),$$

$$\Delta\tau_i(\bar{t}_l^{j,i}) = \alpha_i \Delta\bar{t}_l^{j,i} + \Delta\xi_i(\bar{t}_l^{j,i}),$$

$\Delta t_l^{j,i} = t_l^{j,i} - t_m^{j,i}$, $\Delta\xi_j(t_l^{j,i}) = \xi_j(t_l^{j,i}) - \xi_j(t_m^{j,i})$, $\Delta\bar{t}_l^{j,i} = \bar{t}_l^{j,i} - \bar{t}_m^{j,i} = \Delta t_l^{j,i} + \Delta\delta_l^{j,i}$, with $\Delta\delta_l^{j,i} = \delta_l^{j,i} - \delta_m^{j,i}$, and $\Delta\xi_i(\bar{t}_l^{j,i}) = \xi_i(\bar{t}_l^{j,i}) - \xi_i(\bar{t}_m^{j,i})$; by (3), we have $\Delta\delta_l^{j,i} = \Delta\eta_i(\bar{t}_l^{j,i})$, where $\Delta\eta_i(\bar{t}_l^{j,i}) = \eta_i(\bar{t}_l^{j,i}) - \eta_i(\bar{t}_m^{j,i})$.

Here m denotes the index of the past time instant with respect to which the time increment is calculated. The choice of m leads to different definitions of the time increment, and to algorithms with different properties. In this paper we shall consider the following three characteristic cases (which we denote *AlgDrift.a*, *AlgDrift.b* and *AlgDrift.c*):

1) $m = l - L$, where L is a predefined constant (*AlgDrift.a*);
2) $m = \lfloor \nu l \rfloor$ $(0 < \nu < 1)$, where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x (*AlgDrift.b*);
3) $m = l_0$, where l_0 is a fixed integer (*AlgDrift.c*).

Remark 1: In case a) we have finite memory determined by L, which can be carefully chosen in advance. The case b), when we have both $\lim_{l \to \infty} m = \infty$ and $\lim_{l \to \infty}(l - m) = \infty$, and the case c) are conceptually very important (see Theorem 2 below).

Using (4) we define the following updating procedure for parameter a_i at node i, to be executed immediately after node i receives the message from node j ($j = 1, \ldots, n$, $i \in \mathcal{N}_j^+$):

$$\hat{a}_i(\bar{t}_l^{j,i+}) = \hat{a}_i(\bar{t}_l^{j,i}) + \varepsilon_i^a(\bar{t}_l^{j,i}) \gamma_{ij} \hat{\varphi}_i^a(\bar{t}_l^{j,i}), \tag{5}$$

where γ_{ij} are *a priori* adopted nonnegative weights expressing relative importance of communication links (their role will be discussed below), $\hat{\varphi}_i^a(\bar{t}_l^{j,i}) = \Delta\hat{\tau}_j(t_l^{j,i}) - \Delta\hat{\tau}_i(\bar{t}_l^{j,i})$,

$$\Delta\hat{\tau}_j(t_l^{j,i}) = \Delta\bar{\tau}_j(t_l^{j,i})|_{a_j = \hat{a}_j(t_l^{j,i})}, \tag{6}$$

$$\Delta\hat{\tau}_i(\bar{t}_l^{j,i}) = \Delta\bar{\tau}_i(\bar{t}_l^{j,i})|_{a_i = \hat{a}_i(\bar{t}_l^{j,i})}, \tag{7}$$

$\hat{a}_j(t_l^{j,i})$ and $\hat{a}_i(\bar{t}_l^{j,i})$ are the old estimates, $\hat{a}_i(\bar{t}_l^{j,i+})$ the new estimate, while $\varepsilon_i^a(\bar{t}_l^{j,i})$ is a positive step size. The updating procedure (5) generates, in such a way, recursions of *distributed asynchronous stochastic approximation* type. It will be assumed that the initial estimates are $\hat{a}_i(\bar{t}_0^{j,i}) = 1$.

In terms of the corrected drift $\hat{g}_i(\cdot) = \hat{a}_i(\cdot)\alpha_i$, (5) gives:

$$\hat{g}_i(\bar{t}_l^{j,i+}) = \hat{g}_i(\bar{t}_l^{j,i}) + \varepsilon_i^a(\bar{t}_l^{j,i}) \gamma_{ij} \hat{\psi}_i^a(\bar{t}_l^{j,i}), \tag{8}$$

where $\hat{\psi}_i^a(\bar{t}_l^{j,i}) = \alpha_i\{[\hat{g}_j(t_l^{j,i}) - \hat{g}_i(\bar{t}_l^{j,i})]\Delta t_l^{j,i} + \frac{1}{\alpha_j} \hat{g}_j(t_l^{j,i}) \Delta\xi_j(t_l^{j,i}) - \frac{1}{\alpha_i} \hat{g}_i(\bar{t}_l^{j,i}) \Delta\xi_i(\bar{t}_l^{j,i}) - \hat{g}_i(\bar{t}_l^{j,i}) \Delta\eta_i(\bar{t}_l^{j,i})\}$.

Remark 2: The algorithm does not belong to the class of the so called CBTS algorithms [11]: it is structurally different and simpler, not requiring the step of *relative drift* estimation, which introduces unnecessary dynamics and additional non-linearities.

C. Global Model

Next we derive a global model of the whole network. Parameter updating at the network level is driven by a *global virtual communication clock*, with the rate equal to $\mu = \sum_{i=1}^n \mu_i$, that ticks whenever any of the local communication clocks tick (*e.g.*, [16], [18]). Starting from this fact, a global model for the whole network has been defined in [14] in the form of a recursion in which the k-th iteration corresponds to the k-th tick of the global virtual communication clock. In this paper, we shall adopt an alternative approach, providing more direct insight into the whole updating process. Namely, we shall assume that every local update in the network produces a unique iteration number k in the global model of the parameter estimates, and, *vice versa*, that every k is connected to a local node update (for an update of i-th node, the corresponding continuous time instant is $\bar{t}_l^{j,i}$ for some j and l). In such a way, at a click of j-th communication clock we have $N(j)$ consecutive updates or iterations (assuming that we have only one update at a time), $N(j) \leq |\mathcal{N}_j^+|$. Following analogous approaches in [11], [16], we replace (with some abuse of notation) the variable $\bar{t}_l^{j,i}$ by k in all the above defined functions of time, so that we have $\tau_i(\bar{t}_l^{j,i}) = \tau_i(k)$, $\bar{\tau}_i(\bar{t}_l^{j,i}) = \bar{\tau}_i(k)$, $\xi_i(\bar{t}_l^{j,i}) = \xi_i(k)$, etc; accordingly, we also write $\tau_j(t_l^{j,i}) = \tau_j(k)$, $\bar{\tau}_j(t_l^{j,i}) = \bar{\tau}_j(k)$, $\xi_j(t_l^{j,i}) = \xi_j(k)$, etc. In the case of delays, we write $\bar{\delta}^{j,i} = \bar{\delta}_j(k)$ and $\eta_i(\bar{t}_l^{j,i}) = \eta_i(k)$.

Assume that k is connected to an update at node i, initiated by a tick of node j. Let $\hat{g}(k) = [\hat{g}_1(k)\dots\hat{g}_n(k)]^T$, $\hat{f}(k) = [\hat{f}_1(k)\dots\hat{f}_n(k)]^T$ and $\hat{c}(k) = [\hat{c}_1(k)\dots\hat{c}_n(k)]^T$, where $\hat{g}_\mu(k) = \hat{a}_\mu(k)\alpha_\mu$, $\hat{a}_\mu(k) = \hat{a}_\mu(\bar{t}_l^{j,i})$, $\hat{f}_\mu(k) = \hat{a}_\mu(k)\beta_\mu + \hat{b}_\mu(k)$, $\hat{b}_\mu(k) = \hat{b}_\mu(\bar{t}_l^{j,i})$ and $\hat{c}_\mu(k) = \hat{c}_\mu(\bar{t}_l^{j,i})$, $\mu = 1,\dots,n$, Then, (8) gives

$$\hat{g}(k+1) = \hat{g}(k) + \varepsilon^a(k)Z(k)\hat{g}(k), \tag{9}$$

where $\hat{g}(k+1) = [\hat{g}_1(\bar{t}_l^{j,1+})\dots\hat{g}_n(\bar{t}_l^{j,n+})]^T$, $\varepsilon^a(k) = \mathrm{diag}\{\varepsilon_1^a(k),\dots,\varepsilon_n^a(k)\}$, $\varepsilon_i^a(k) = \varepsilon_i^a(\bar{t}_l^{j,i})$ (see (5)),

$$Z(k) = A\Gamma(k)\Delta t(k) + N_g(k),$$

$A = \mathrm{diag}\{\alpha_1,\dots,\alpha_n\}$, $\Gamma(k) = [\Gamma(k)_{\mu\nu}]$, with $\Gamma(k)_{ii} = -\gamma_{ij}$ and $\Gamma(k)_{ij} = \gamma_{ij}$, with $\Gamma(k)_{\mu\nu} = 0$ otherwise, $\Delta t(k) = \bar{t}_l^{j,i} - \bar{t}_m^{j,i}$, while the noise term is defined as

$$N_g(k) = -A\Gamma_d(k)\Delta\eta_d(k) + A\Gamma(k)\Delta\xi_d(k)A^{-1},$$

where $\Gamma_d(k) = \mathrm{diag}\{\mathrm{diag}\{\gamma_{1j},\dots,\gamma_{nj}\}\omega(k)\}$, $\omega(k) = [\omega_1(k)\dots\omega_n(k)]^T$, $\omega_i(k) = 1$, $\omega_\mu(k) = 0$ for $\mu \neq i$, $\Delta\eta_d(k) = \mathrm{diag}\Delta\eta(k)$, $\Delta\eta(k) = [\Delta\eta_1(k)\dots\Delta\eta_n(k)]^T$, $\Delta\xi_d(k) = \mathrm{diag}\Delta\xi(k)$ and $\Delta\xi(k) = [\Delta\xi_1(k)\dots\Delta\xi_n(k)]^T$.

III. Convergence Analysis

A. Preliminaries

Within the exposed general setting, we additionally assume:

(A1) Graph $\mathcal{G}$ has a spanning tree.

(A2) $\{\xi_i(k)\}$ and $\{\eta_i(k)\}$, $i = 1,\dots n$, are mutually independent zero mean i.i.d. random sequences, bounded w.p.1.

(A3) The step sizes $\varepsilon_i^a(k)$ and $\varepsilon_i^b(k)$ are defined in the following way:
$\varepsilon_i^a(k) = \varepsilon_i(k)|_{\zeta=\zeta'}$ for *AlgDrift.a*,
$\varepsilon_i^a(k) = \varepsilon_i(k)|_{\zeta=1+\zeta'}$ for *AlgDrift.b* and *AlgDrift.c*,
where $\varepsilon_i(k) = \nu_i(k)^{-\zeta}$, $\nu_i(k) = \sum_{m=1}^k I\{\text{node } i \text{ received a message}\}$, representing the number of updates of node i up to the instant k ($I\{\cdot\}$ denotes the indicator function), while $\frac{1}{2} < \zeta', \zeta'' \leq 1$.

Remark 3: (A1) implies that graph $\mathcal{G}$ has a center node from which all the remaining nodes are reachable [19], [20]. (A2) is a standard assumption, in which boundedness, which always holds in practice, is introduced for making derivations easier. (A3) is practically very important: it eliminates the need for a centralized clock which would define the common step size for all the nodes as a function of k. The choice of the exponent in the expression for $\varepsilon_i^a(k)$ for *AlgDrift.b* and *AlgDrift.c* is motivated by the properties of the random variable $\Delta t(k)$ which diverges linearly to infinity (see Theorem 2).

Asymptotical behavior of the step size is given by the following lemma.

Lemma 1: Let (A1) and (A3) be satisfied, let p_i be the unconditional probability of node i *to update* its parameters at k-th iteration, and let $\zeta > 0$. Then, for a given $q \in (0, \frac{1}{2})$, there exists $\bar{k}$ such that w.p.1 for all $k \geq \bar{k}$

$$\varepsilon_i(k) = \frac{1}{k^\zeta}(\frac{\bar{N}}{p_i})^\zeta + \tilde{\varepsilon}_i(k), \tag{10}$$

where $\bar{N} = E_j\{E\{N(j)|j\}\}$ represents the average number of updates per one tick of the global virtual clock and $|\tilde{\varepsilon}_i(k)| \leq \tilde{\varepsilon}_i \frac{1}{k^{\zeta+\frac{1}{2}-q}}$, $0 < \tilde{\varepsilon}_i < \infty$, $i = 1,\dots,n$.

Properties of the matrix $\Gamma(k)$ defined in the previous section are essential for convergence of the algorithm; its expectation $\bar{\Gamma} = E\{\Gamma(k)\}$ has a central role, since it contains all the information about the network structure and the weights of particular links. It has the structure of a weighted Laplacian matrix for $\mathcal{G}$:

$$\bar{\Gamma} = \begin{bmatrix} -\sum\limits_{j,j\neq 1}\gamma_{1j}\pi_{1j} & \gamma_{12}\pi_{12} & \cdots & \gamma_{1n}\pi_{1n} \\ \gamma_{21}\pi_{21} & -\sum\limits_{j,j\neq 2}\gamma_{2j}\pi_{2j} & \cdots & \gamma_{2n}\pi_{2n} \\ & & \ddots & \\ \gamma_{n1}\pi_{n1} & \gamma_{n2}\pi_{n2} & \cdots & -\sum\limits_{j,j\neq n}\gamma_{nj}\pi_{nj} \end{bmatrix} \tag{11}$$

($\gamma_{ij} = 0$ when $j \notin \mathcal{N}_i^-$), where π_{ij} is unconditional probability that the node j broadcasts and node i updates its parameters as a consequence ($\pi_{ij} = \pi_j p_{ij}$, where π_j is the unconditional probability for node j to broadcast).

According to (9) and Lemma 1, we shall consider $B(k) = P^{-\zeta}A\Gamma(k)$ and $\bar{B} = E\{B(k)\} = P^{-\zeta}A\bar{\Gamma}$ ($P^{-\zeta} = \bar{N}^\zeta\mathrm{diag}\{p_1^{-\zeta},\dots,p_n^{-\zeta}\}$). The latter matrix has the following properties.

Lemma 2: [20] Matrix $\bar{B}$ has one eigenvalue at the origin, and the remaining ones in the left half plane. Let $T = \left[\mathbf{1} \,\vdots\, T_{n\times(n-1)}\right]$, where $T_{n\times(n-1)}$ is such that $\mathrm{span}\{T_{n\times(n-1)}\} = \mathrm{span}\{\bar{B}\}$, while $\mathbf{1} = [1\dots1]^T$. Then,

$$T^{-1}\bar{B}T = \left[\begin{array}{c:c} 0 & 0_{1\times(n-1)} \\ \hdashline 0_{(n-1)\times 1} & \bar{B}^* \end{array}\right], \tag{12}$$

where $\bar{B}^*$ is Hurwitz.

Consequently, there exists $R^g > 0$ satisfying

$$R^g\bar{B}^* + \bar{B}^{*T}R^g = -Q^g, \tag{13}$$

for any given $Q^g > 0$. It also follows from the derivation of (12) that $T^{-1}B(k)T = \left[\begin{array}{c:c} 0 & B_1(k) \\ \hdashline 0_{(n-1)\times 1} & B_2(k) \end{array}\right]$, with $E\{B_1(k)\} = 0$ and $E\{B_2(k)\} = \bar{B}^*$.

Lemma 3: $E\{\Delta t(k)\} = \frac{1}{\mu_j}(l-m)$, $\mathrm{var}\{\Delta t(k)\} = \frac{1}{\mu_j^2}(l-m)$, where $l-m = L$ for *AlgDrift.a*, $l-m = \lfloor (1-\nu)l \rfloor$ for *AlgDrift.b* and $l-m = l$ for *AlgDrift.c*; for large l, we have $l \sim \pi_{ij}k$.

B. Convergence and Convergence Rate

After coming back to (9), we first insert $\varepsilon^a(k)$ from (10). Then, we introduce $\tilde{g}(k) = T^{-1}\hat{g}(k)$ and decompose $\tilde{g}(k)$ as $\tilde{g}(k) = [\tilde{g}(k)^{[1]} \vdots \tilde{g}(k)^{[2]T}]^T$, where $\tilde{g}(k)^{[1]} = \tilde{g}_1(k)$ and $\tilde{g}(k)^{[2]} = [\tilde{g}_2(k)\ldots\tilde{g}_n(k)]^T$. After neglecting the higher order terms from (10), we obtain

$$\begin{aligned}\tilde{g}(k+1)^{[1]} =& \tilde{g}(k)^{[1]} + \frac{1}{k^\zeta}F_1(k)\Delta t(k)\tilde{g}(k)^{[2]} \\ &+ \frac{1}{k^\zeta}H_1(k)\tilde{g}(k) \qquad (14)\\ \tilde{g}(k+1)^{[2]} =& \{I + \frac{1}{k^\zeta}[\bar{B}^* + F_2(k)]\Delta t(k)\}\tilde{g}(k)^{[2]} \\ &+ \frac{1}{k^\zeta}H_2(k)\tilde{g}(k), \qquad (15)\end{aligned}$$

where matrices $F_1(k)$ and $F_2(k)$ are defined by $T^{-1}[B(k) - \bar{B}^*]T = \begin{bmatrix} 0 & F_1(k) \\ 0_{(n-1)\times 1} & F_2(k)\end{bmatrix}$, while $H_1(k)$ and $H_2(k)$ are defined by $T^{-1}P^{-c}N_g(k)T = \begin{bmatrix} H_1(k) \\ H_2(k)\end{bmatrix}$.

Theorem 1: Let assumptions (A1)–(A3) be satisfied. Then, $\tilde{g}(k)^{[1]}$ from (14) converges to a random variable χ^* with bounded second moment, and $\tilde{g}(k)^{[2]}$ from (15) to zero in the mean square sense and w.p.1; in other words, $\hat{g}(k)$ generated by (9) converges for all three choices of m to $\hat{g}_\infty = \chi^*\mathbf{1}$ in the mean square sense and w.p.1.

The rate of convergence of the drift estimation scheme is of utmost importance not only for the convergence of local clocks to a common virtual clock, but also for the convergence of the offset estimation algorithm. Asymptotic rate of convergence to consensus of the algorithm (9) will be studied through the behavior of $\tilde{g}(k)^{[2]}$ in (15), using the methodology of [21, Chapter 3].

Theorem 2: Let (A1)–(A3) hold. Then, $z(k) = k^{\zeta d}\tilde{g}(k)^{[2]}$, where $d > 0$ and $\tilde{g}(k)^{[2]}$ is defined by (15), converges to zero in the mean square sense and w.p.1, when $\zeta' < 1$ for:
$\zeta d < \zeta' - \frac{1}{2}$ (*AlgDrift.a*),
$\zeta d < \frac{1}{2} + \zeta'$ (*AlgDrift.b*) and
$\zeta d < 1$ (*AlgDrift.c*),
and when $\zeta' = 1$ for:
$d < \min(\frac{1}{2},\ 2qr)$ (*AlgDrift.a*),
$d < \min(\frac{3}{4},\ qr)$ (*AlgDrift.b*) and
$d < \min(\frac{1}{2},\ qr)$ (*AlgDrift.c*),
where $r = \frac{\lambda_{min}(Q^g)}{\lambda_{max}(R^g)}$, $q = \frac{L}{\max_{i,j}(\mu_j p_{ij})}$ for *AlgDrift.a*, $q = \frac{1-\nu}{\mu}$ for *AlgDrift.b* and $q = \frac{1}{\mu}$ for *AlgDrift.c*.

Remark 4: The results hold asymptotically, for k large enough. They indicate that the *AlgDrift.b* provides the best results: in the case when $\zeta' < 1$ the condition $\zeta d > 1$ is achieved, enabling convergence to a common virtual clock. The requirements for m ensure an effectively increasing signal-to-noise ratio, together with a sufficient number of realizations of the noise term at the left end of the interval $[m,l]$. However, in practice, it is sufficient to choose $l-m = L$ large enough and to apply *AlgDrift.a*, avoiding in such a way problems connected with the increase of memory inherent to *AlgDrift.b*. Practically the best results can be obtained by *AlgDrift.a* for L moderately high.

Notice that the CBTS algorithms discussed in [11] cannot achieve convergence rate $\zeta d > 1$, important for achieving convergence to a global virtual clock.

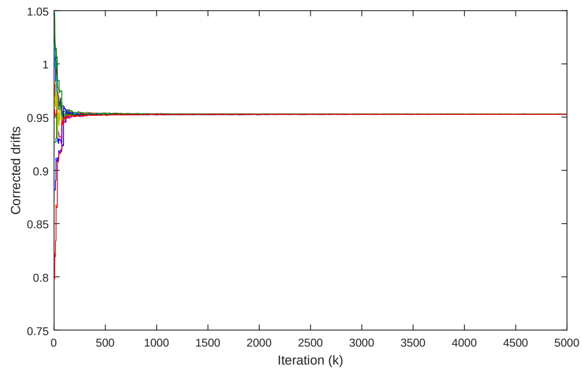

Fig. 1. *AlgDrift.a* with $L = 100$

IV. Simulations

Numerous simulation experiments have been undertaken in order to get a practical insight into the proposed distributed time synchronization algorithm. Different networks have been simulated with variable number of nodes. The assumed network topology corresponds to a modification of Geometric Random Graphs [22]. The nodes represent randomly spatially distributed agents within a square area. Initially, the nodes are assumed to be connected if their Euclidean distance is less than a predefined number: this results in an undirected graph. The obtained graph is modified in such a way as to transform a certain percentage (roughly 10 percent) of the original two-way communications into one-way communications. A program is developed for final optimization, which ensures, on the basis of additional modifications, that assumption (A1) is satisfied. Parameters α_i and β_i are randomly chosen in the intervals $(0.96, 1.04)$ and $(-0.2, 0.2)$, respectively. Average communication delays $\bar{\delta}^{j,i}$ have been chosen to be 0.1, while $\{\eta(k)\}$ and $\{\xi(k)\}$ have been simulated as zero-mean Gaussian white noise sequences with specified standard deviation σ. It has been adopted that $\zeta' = 0.99$ and that the communication dropouts occur according to the probability $p_{ij} = 0.9$.

Typical behavior of the corrected drifts generated by *AlgDrift.a* ($L = 100$) and *AlgDrift.b* ($\nu = \frac{1}{2}$) in the presence of stochastic delays and measurement noise with $\sigma = 0.05$ is presented in Figs. 1 and 2 for a network with ten nodes. Convergence to consensus can be clearly observed in all cases. Analogous schemes from the literature (*e.g.*, [11]) cannot achieve such a performance. It should be noticed that the best results are achieved by *AlgDrift.a* with $L = 100$; *AlgDrift.b* is practically inferior on finite intervals, in spite

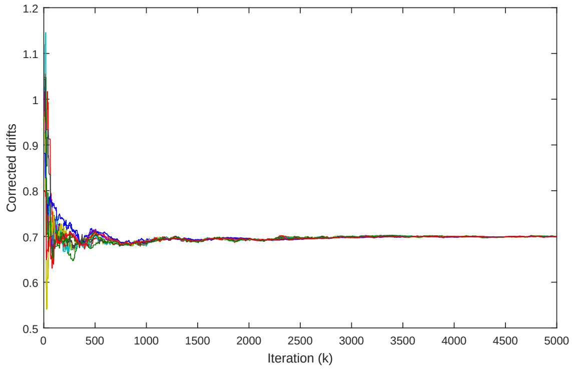

Fig. 2. *AlgDrift.b* with $\nu = 1/2$

of the asymptotic results from Theorem 2. This indicates that the best choice of drift estimation algorithm should be in practice connected to *AlgDrift.a* with a suitably selected L large enough; it represents the best compromise for practice.

V. Conclusion

In this paper a new distributed asynchronous algorithm have been proposed for drift correction within time synchronization for networks with random communication delays, measurement noise and communication dropouts. A new algorithm is proposed based on an error function derived from local time increments. It has been proved, using the stochastic approximation arguments, that this algorithm achieves asymptotic consensus of the corrected drifts in the mean square sense and w.p.1 under general conditions concerning network properties. It is important that the algorithm achieves convergence rate superior to all similar schemes, especially in view of convergence to a virtual global clock. Also, the algorithm is substantially simpler than all the existing schemes.

VI. Proof of Theorem 1

Introduce Lyapunov functions $V^g(k) = E\{(\tilde{g}(k)^{[1]})^2\}$ and $W^g(k) = E\{\tilde{g}(k)^{[2]T} R^g \tilde{g}(k)^{[2]}\}$, where $R^g > 0$ satisfies (13) for a given $Q^g > 0$.

In order to obtain an estimate of $V^g(k)$, we decompose $\tilde{g}(k+1)^{[1]}$ from (14) into the sum of zero input and zero state responses, defined by

$$\tilde{g}_1(k+1)^{[1]} = \Pi(k,1)^{[1]}\tilde{g}(1)^{[1]} \tag{16}$$

and

$$\begin{aligned}\tilde{g}_2(k+1)^{[1]} = \sum_{\sigma=1}^{k} \frac{1}{\sigma^{\zeta}} \Pi(k,\sigma+1)^{[1]}[F_1(\sigma)\Delta t(\sigma) \\ + H_1(\sigma)^{[2]}]\tilde{g}(\sigma)^{[2]},\end{aligned} \tag{17}$$

respectively, where $\Pi(k,l)^{[1]} = \prod_{\sigma=l}^{k}(1 + \frac{1}{\sigma^{\zeta}} H_1(\sigma)^{[1]})$, $\Pi(k,k+1)^{[1]} = 1$, and $H_1(k)^{[1]}$ follows from the decomposition $H_1(k) = [H_1(k)^{[1]} \vdots H_1(k)^{[2]}]$. Therefore, $V^g(k) \leq 2V_1^g(k) + 2V_2^g(k)$,where $V_1^g(k) = E\{(\tilde{g}_1(k)^{[1]})^2\}$ and $V_2^g(k) = E\{(\tilde{g}_2(k)^{[1]})^2\}$.

Introduce $\sum_{i=1}^{n} |\mathscr{N}_i^-|$ infinite subsequences $\{\kappa^{ij}(\nu)\}$ of the set of nonnegative integers $\mathscr{I}^+$, $i = 1,\ldots,n$, $j \in \mathscr{N}_i^-$, $\nu = 0,1,2,\ldots$, in which $\kappa^{ij}(\nu)$ for a given ν defines an instant k corresponding to an update at node i realized as a consequence of a tick of node j ($\kappa^{ij}(\nu_1) < \kappa^{ij}(\nu_2)$ for $\nu_1 < \nu_2$ and $\cup_{i,j}\{\kappa^{ij}(\nu)\} = \mathscr{I}^+$). Define $\Pi(k,1)_s^{[1]} = \prod_{\sigma \in \{\kappa^{ij}(\nu)\}, \sigma \leq k}(1 + \frac{1}{\sigma^{\zeta}} H_1(\sigma)^{[1]})$, $s = 1,\ldots,\sum_i |\mathscr{N}_i^-|$, $i = 1,\ldots,n$, $j \in \mathscr{N}_i^-$, so that $\prod_s \Pi(k,1)_s^{[1]} = \Pi(k,1)^{[1]}$. According to the definition of $N^g(k)$, for *AlgDrift.a* and *AlgDrift.b*, the zero mean random sequences $\{H_1(\sigma)^{[1]}\}$, $\sigma \in \{\kappa^{ij}(\nu)\}$, have the property that $\{H_1(\sigma)^{[1]}\}_{\sigma=\{\kappa^{ij}(\nu)\}}$ is correlated only with $\{H_1(\sigma)^{[1]}\}_{\sigma=\{\kappa^{ij}(\nu-1)\}}$ and $\{H_1(\sigma)^{[1]}\}_{\sigma=\{\kappa^{ij}(\nu+1)\}}$. Therefore, it follows that $E\{(\Pi(k,1)_s^{[1]})^2\} < \infty$, since $\Pi(k,1)_s^{[1]}$ are mutually independent. For *AlgDrift.c*, we have that $H_1(\sigma)^{[1]} = \tilde{H}_1(\sigma)^{[1]} - \tilde{H}_1(\sigma_0)^{[1]}$, where $\tilde{H}_1(\sigma)^{[1]}$ is zero mean i.i.d. and $\tilde{H}_1(\sigma_0)^{[1]}$ a finite w.p.1 random variable ($\sigma_0 = \kappa^{i,j}(0)$). Therefore, we have

$$E\{(1 - \frac{1}{\sigma^{1+\zeta'}} H_1(\sigma))^2 | \mathscr{F}_{\sigma_0}\} \leq 1 - c_1 \frac{1}{\sigma^{1+\zeta'}} + c_2 \frac{1}{\sigma^{2(1+\zeta')}}, \tag{18}$$

where $\mathscr{F}_{\sigma_0}$ is the minimal sigma algebra generated by the measurements up to σ_0. It follows that $E\{(\Pi(k,1)_s^{[1]})^2\} < \infty$. Therefore, we obtain that $\sup_k V_1^g(k) < \infty$ for all three algorithms.

Estimation of $V_2^g(k)$ for *AlgDrift.a* and *AlgDrift.b* starts from decomposing the sum at the right hand side of (17) into $\sum_{i=1}^{n} |\mathscr{N}_i^-|$ sums with indices σ belonging to $\{\kappa^{ij}(\nu)\}$, $\sigma \leq k$. All these sums contain weighted zero mean random variables $F_1(\sigma)\Delta t(\sigma) + H_1(\sigma)^{[2]}$; their correlation with $\tilde{g}(\sigma)$ is w.p.1 of the order of magnitude of $\frac{1}{\sigma^{2c}}$, so that it can be neglected for k large enough w.r.t. the corresponding terms in the expression for $V_2^g(k)$. It is important to notice that $\sum_\sigma E\{\frac{1}{\sigma^{2\zeta}} F_1(\sigma)^2 \Delta t(\sigma)^2\} \leq \infty$ for *AlgDrift.a* and *AlgDrift.b*, by virtue of Lemma 3. Noticing also that $\sup_k \Pi(k,1)^{[1]} < \infty$, it follows, after straightforward technicalities, that

$$V_2^g(k+1) \leq C_1 \sum_{\sigma=1}^{k} \frac{1}{\sigma^{1+q''}} W^g(\sigma), \tag{19}$$

where $C_1 > 0$ and $q'' > 0$. For *AlgDrift.c*, the sum at the right hand side of (17) contains the terms $H_1(\sigma)^{[2]} = \tilde{H}_1(\sigma)^{[2]} - \tilde{H}_1(\sigma_0)^{[2]}$, $\sigma \in \kappa^{ij}(\nu)$. Having in mind that $\{\tilde{H}_1(\sigma)^{[2]}\}$ is zero mean and $\tilde{H}_1(\sigma_0)^{[2]}$ is bounded w.p.1, for *AlgDrift.c* $\sum_\sigma \frac{1}{\sigma^{2\zeta}} E\{\Delta t(k)^2\} < \infty$ by Lemma 3 and that $\sum_\sigma \frac{1}{\sigma^{\zeta}} < \infty$, we obtain (19).

Consequently,

$$V^g(k+1) \leq C_2[1 + \max_{1 \leq \sigma \leq k} W^g(\sigma)], \tag{20}$$

where $C_2 > 0$, having in mind that $\sum_{k=1}^{\infty} \frac{1}{\sigma^{1+q''}} < \infty$.

Estimation of $W^g(k)$ is based on considering the recursion (9) as a set of recursions on the sets $\{\kappa^{ij}(\nu)\}$, $i = 1,\ldots,n$, $j \in \mathscr{N}_i^-$. We rewrite (15) for $\{\sigma \in \kappa^{ij}(\nu)\}$ in the following way

$$\tilde{g}(\sigma+1)^{[2]} = \Pi(\sigma,\sigma)^{[2]}\tilde{g}(\sigma)^{[2]} + \frac{1}{\sigma^{\zeta}} H_2(\sigma)^{[1]}\tilde{g}(\sigma)^{[1]}, \tag{21}$$

where $\Pi(\sigma,\sigma) = I + \frac{1}{\sigma^{\zeta}}[(\bar{B}^* + F_2(\sigma))\Delta t(\sigma) + H_2(\sigma)^{[2]}]$, while $H_2(\sigma)^{[1]}$ and $H_2(\sigma)^{[2]}$ follow from the decomposition

$H_2(\sigma) = [H_2(\sigma)^{[1]} \vdots H_2(\sigma)^{[2]}]$.

We start the analysis by observing that for any n-vector x and any σ large enough

$$x^T E\{\Pi(\sigma,\sigma)^{[2]T} R^g \Pi(\sigma,\sigma)^{[2]}\}x \\ \leq [1 - \frac{2}{\sigma^{\zeta'}} q \frac{\lambda_{min}(Q^g)}{\lambda_{max}(R^g)} + O(\frac{1}{\sigma^{2\zeta'}})]x^T R^g x, \tag{22}$$

where $0 < \lambda_{min}(Q^g), \lambda_{max}(R^g) < \infty$ and $q = \frac{L}{\max_{i,j}(\mu_j p_{ij})}$ for *AlgDrift.a*, $q = \frac{1-\nu}{\mu}$ for *AlgDrift.b* and $q = \frac{1}{\mu}$ for *AlgDrift.c*. Because $q > 0$ (Lemma 3), after standard technicalities based on the classical results on stochastic approximation [21], [23], it follows that $\prod_{\sigma \in \{\kappa^{ij}(\nu)\}} \|\Pi(\sigma,\sigma)\| \rightarrow_{\sigma\rightarrow\infty} 0$, $i = 1,\ldots,n$, $j \in \mathcal{N}_i^-$, in the mean square sense and w.p.1, for *AlgDrift.a*, *AlgDrift.b* and *AlgDrift.c*. Moreover, as $\{H_2(\sigma)^{[1]}\}$ has the properties analogous to those of $\{H_1(\sigma)^{[1]}\}$, we have for k large enough

$$W^g(\sigma+1) \leq [1 - c_1 \frac{1}{\sigma^{\zeta'}}]W^g(\sigma) + C_3 \frac{1}{\sigma^{\zeta}} V^g(\sigma), \tag{23}$$

where $0 < c_1, C_3 < \infty$.

Using the methodology of [24], [25] we can, consequently, show that $\sup_k V^g(k) < \infty$. This gives rise to the conclusion that $\tilde{g}(k)^{[1]}$ tends to a random variable χ^* ($E\{\chi^{*2}\} < \infty$) and $\tilde{g}(k)^{[2]}$ to zero in the mean square sense and w.p.1. Consequently

$$\hat{g}_\infty = T \begin{bmatrix} \lim_{k\rightarrow\infty} \tilde{g}(k)^{[1]} \\ \cdots\cdots\cdots \\ 0 \end{bmatrix} = \chi^* \mathbf{1}, \tag{24}$$

which proves the theorem.

VII. Proof of Theorem 2

After introducing the expression for $z(k)$ into (15), we use the approximation $(1+\frac{1}{k})^{\zeta d} \approx 1 + \zeta d \frac{1}{k}$ and obtain, after neglecting higher order terms, that for k large enough

$$z(k+1) = z(k) + \{\frac{1}{k^\zeta}[\bar{B}^* + F_2(k)]\Delta t(k) \\ + \zeta d \frac{1}{k} I\} z(k) + \frac{1}{k^{\zeta(1-d)}} H_2(k)\tilde{g}(k). \tag{25}$$

Applying the methodology of the proof of Theorem 1 to (25), we observe that for $\zeta' < 1$ the term proportional to $\frac{1}{k}$, introduced by the formulation of the recursion for $z(k)$, can be neglected for k large enough with respect to the term proportional to $\frac{1}{k^{\zeta'}}$. As $\bar{B}^*$ is Hurwitz, $\lim_{k\rightarrow\infty} z(k) = 0$ in the mean square sense and w.p.1 provided: a) $2\zeta'(1-d) > 1$ for *AlgDrift.a*, b) $2(1+\zeta')(1-d) > 1$ for *AlgDrift.b* and c) $(1+\zeta')(1-d) > \zeta'$ for *AlgDrift.c*, wherefrom the first part of the result follows. Notice that different conditions result from different definitions of ζ and the properties of the corresponding sequence $\{H_2(k)\}$. For $\zeta' = 1$, the terms proportional to $\frac{1}{k}$ and $\frac{1}{k^{\zeta'}}$ are of the same order of magnitude; as a result, the convergence conditions for (25) depend on the properties of the matrix $\bar{B}^*$. Hence the result follows.

References

[1] K.-D. Kim and P. R. Kumar, "Cyberphysical systems: A perspective at the centennial," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1287–1308, 2012.

[2] I. F. Akyildiz and M. C. Vuran, *Wireless sensor networks*. John Wiley & Sons, 2010.

[3] B. Sundaraman, U. Buyand, and A. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, pp. 281–323, 2005.

[4] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. Symp. Oper. Syst. Design and Implement.*, 2002, pp. 147–163.

[5] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. H. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Process. Mag.*, vol. 25, pp. 81–97, 2008.

[6] R. Solis, V. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multi hop wireless networks," in *Proc. IEEE Conf. Decision and Control*, 2006, pp. 2734–2739.

[7] J. He, P. Cheng, J. Chen, L. Shi, and R. Lu, "Time synchronization for random mobile sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, pp. 3935–3946, 2014.

[8] L. Schenato and F. Fiorentin, "Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.

[9] C. Liao and P. Barooah, "Distributed clock skew and offset estimation from relative measurements in mobile networks with Markovian switching topologies," *Automatica*, vol. 49, pp. 3015–3022, 2013.

[10] Y.-P. Tian, "LSTS: a new time synchronization protocol for networks with random communication delays," in *Proc. IEEE Conf. Decision and Control*, December 2015, pp. 7404–7409.

[11] Y.-P. Tian, S. Zong, and Q. Cao, "Structural modeling and convergence analysis of consensus-based time synchronization algorithms over networks: non-topological conditions," *Automatica*, vol. 65, pp. 64–75, 2016.

[12] R. Carli, A. Chiuso, S. Zampieri, and L. Schenato, "A PI consensus controller for networked clock synchronization," in *Proc. IFAC World Congress*, 2008.

[13] N. M. Freris, S. R. Graham, and P. R. Kumar, "Fundamental limits on synchronizing clocks over networks," *IEEE Trans. Autom. Control*, vol. 56, pp. 1352–1364, 2011.

[14] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed drift estimation for time synchronization in lossy networks," in *24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 779–784.

[15] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed time synchronization in lossy wireless sensor networks," in *3rd IFAC Workshop on Distr. Estim. Contr. Netw. Syst., NECSYS*, vol. 3, 2012, pp. 25–30.

[16] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Trans. Autom. Control*, vol. 56, pp. 1337–1351, 2011.

[17] G. Xiong and S. Kishore, "Analysis of distributed consensus time synchronization with Gaussian delay over wireless sensor networks," *EURASIP J. on Wireless Comm. and Netw.*, no. 1, pp. 1–9, 2009.

[18] T. C. Aysal, M. E. Yildriz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, pp. 2748–2761, 2009.

[19] R. Olfati-Saber, A. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, pp. 215–233, 2007.

[20] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed blind calibration in lossy sensor networks via output synchronization," *IEEE Trans. Autom. Control*, vol. 60, pp. 3257–3262, 2015.

[21] H. F. Chen, *Stochastic approximation and its applications*. Dordrecht, the Netherlands: Kluwer Academic, 2002.

[22] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, 2006.

[23] H. J. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

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

[25] M. Huang, S. Day, G. N. Nair, and J. H. Manton, "Stochastic consensus over noisy networks with Markovian and arbitrary switches," *Automatica*, vol. 46, pp. 1571–1583, 2010.