Automatica 93 (2018) 126–137

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Distributed time synchronization for networks with random delays and measurement noise[☆]

Miloš S. Stanković[a,c,d,*], Srdjan S. Stanković[b,c], Karl Henrik Johansson[e]

[a] Innovation Center, School of Electrical Engineering, University of Belgrade, Serbia
[b] School of Electrical Engineering, University of Belgrade, Serbia
[c] Vlatacom Institute, Belgrade, Serbia
[d] School of Technical Sciences, Singidunum University, Belgrade, Serbia
[e] ACCESS Linnaeus Centre, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 19 August 2016
Received in revised form 20 November 2017
Accepted 13 February 2018

ABSTRACT

In this paper a new distributed asynchronous algorithm is proposed for time synchronization in networks with random communication delays, measurement noise and communication dropouts. Three different types of the drift correction algorithm are introduced, based on different kinds of local time increments. Under nonrestrictive conditions concerning network properties, it is proved that all the algorithm types provide convergence in the mean square sense and with probability one (w.p.1) of the corrected drifts of all the nodes to the same value (consensus). An estimate of the convergence rate of these algorithms is derived. For offset correction, a new algorithm is proposed containing a compensation parameter coping with the influence of random delays and special terms taking care of the influence of both linearly increasing time and drift correction. It is proved that the corrected offsets of all the nodes converge in the mean square sense and w.p.1. An efficient offset correction algorithm based on consensus on local compensation parameters is also proposed. It is shown that the overall time synchronization algorithm can also be implemented as a flooding algorithm with one reference node. It is proved that it is possible to achieve bounded error between local corrected clocks in the mean square sense and w.p.1. Simulation results provide an additional practical insight into the algorithm properties and show its advantage over the existing methods.

1. 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 (Akyildiz & Vuran, 2010; Holler et al., 2014; Kim & Kumar, 2012). One of the basic requirements in networked systems is, in general, *time synchronization*, *i.e.*, all the nodes have to share a common notion of time. The problem of time synchronization in networked systems 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.*, Sundararaman, Buy, and Kshemkalyani (2005). There are numerous approaches to this problem, starting from different assumptions and using different methodologies, *e.g.*, Elson, Girod, and Estrin (2002), Sivrikaya and Yener (2004) and Sundararaman et al., (2005). An important class of time synchronization algorithms is based on full distribution of functions (Simeone, Spagnolini, Bar-Ness, & Strogatz, 2008; Solis, Borkar, & Kumar, 2006). Distributed schemes with the so-called *gradient property* have been proposed in Fan and Lynch (2006) and Sommer and Wattenhofer (2009). A class of *consensus based algorithms*, called CBTS (Consensus-Based Time Synchronization) algorithms, has attracted considerable attention, *e.g.*, He, Cheng, Chen, Shi, and Lu (2014a), He, Cheng, Shi, Chen, and Sun (2014b), Li and Rus (2006), Liao and Barooah (2013a), Schenato and Fiorentin (2011), Tian (2015) and Xiong and Kishore (2009). It has been treated in a unified way in a recent survey (Tian, Zong, & Cao, 2016), providing

☆ 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. The material in this paper was partially presented at the 24th Mediterranean Conference on Control and Automation, June 21–24, 2016, Athens, Greece, the 17th annual European Control Conference, June 12–15, 2018, Limassol, Cyprus, and the 61st annual IcETRAN, June 5–8, 2017, Kladovo, Serbia. This paper was recommended for publication in revised form by Associate Editor Antonis Papachristodoulou under the direction of Editor Christos G. Cassandras.

* Corresponding author at: Innovation Center, School of Electrical Engineering, University of Belgrade, Serbia.
E-mail addresses: milos.stankovic@ic.etf.rs (M.S. Stanković), stankovic@etf.rs (S.S. Stanković), kallej@kth.se (K.H. Johansson).

https://doi.org/10.1016/j.automatica.2018.03.054
0005-1098/

figure of merit of the principal approaches. In Carli, Chiuso, Zampieri, and Schenato (2008) and Yildirim, Carli, and Schenato (2015) a control-based approach to distributed time synchronization has been proposed. Fundamental and yet unsolved problems in all the existing approaches are connected with *communication delays and measurement noise*; see Freris, Graham, and Kumar (2011) for basic issues, and Chaudhari, Serpedin, and Qaraqe (2008), Choi, Liang, Shen, and Zhuang (2012), Garone, Gasparri, and Lamonaca (2015) and Xiong and Kishore (2009) for different aspects of delay influence.

In this paper we propose a new *asynchronous distributed* algorithm for time synchronization in lossy networks, characterized by *random communication delays*, *measurement noise* and *communication dropouts*. The algorithm is composed of two distributed recursions of *asynchronous stochastic approximation type* based on *broadcast gossip* and derived from predefined local error functions. The recursions are aimed at achieving asymptotic consensus on the *corrected drifts* and *corrected offsets* and, consequently, at obtaining *common virtual clock* for all the nodes in the network.

The proposed recursion for *drift synchronization* is based on noisy time increments, defined in three characteristic forms (a preliminary form has been presented in Stanković, Stanković, and Johansson (2016)). 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*. It is shown that the proposed recursion with the increments of unbounded length with random boundaries provides convergence rate faster than $\frac{1}{t}$, what is essential for convergence to a common global virtual clock. Compared to the analogous existing algorithms (Schenato & Fiorentin, 2011; Tian, 2015), the proposed scheme is structurally different and simpler (not involving mutual drift estimation, typical for the CBTS algorithms) and, in addition, provides the best performance. Notice that the algorithm in Schenato and Fiorentin (2011) cannot handle communication delays and measurement noise, while the papers Tian (2015, 2017), derived from a particular form of increments of unbounded length, treat random delays, but not measurement noise and communication dropouts. Moreover, the algorithm proposed therein cannot provide convergence rate achievable by the proposed methodology. The approach in Garone et al. (2015) does not ensure consensus of corrected drifts in spite of additional pairwise inter-node communications.

We also propose a novel recursion for *offset synchronization*, which starts from a special error function, obtained from the difference between local times, with two important modifications aiming at: (1) eliminating the deteriorating effect of linearly increasing absolute time, and (2) coping with the influence of delays by introducing additional *delay compensation parameters*. It is proved that the proposed algorithm provides convergence in the mean square sense and w.p.1 to a set of bounded random variables. The algorithm for the offset correction proposed in Schenato and Fiorentin (2011) cannot handle these problems, while the algorithm in Tian (2015, 2017) allows unbounded corrected offsets and assume perfect clock readings. The approach in Yildirim et al. (2015) does not provide a rigorous insight into overall network stability. Attention is also paid to an improvement of the offset correction algorithm, based on linear consensus iterations, aiming at decreasing the dispersion of the offset convergence points. Special cases related to delay and noise are discussed in order to clarify potentials of the proposed algorithms.

The resulting time synchronization algorithm composed of the proposed drift and offset correction recursions ensures finite differences between local corrected clocks in the mean square sense and w.p.1. To the authors' knowledge, this is the first method with such a performance in the case of random delays, measurement noise and communication dropouts. It is also demonstrated that the proposed algorithm can be implemented as a *flooding algorithm*, with one preselected reference node.

Finally, some illustrative simulation results are presented, giving additional insights into the theoretically discussed issues.

2. Synchronization algorithms

2.1. Time and network models

Assume a network consisting of 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. Denote by $\mathcal{N}_i^+$ the out-neighborhood and by $\mathcal{N}_i^-$ the in-neighborhood of node i, $i = 1, \dots, n$. Assume that each node has a local clock, whose output, defining the *local time*, is given for any *absolute time* $t \in \mathcal{R}$ by

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

where $\alpha_i \neq 0$ 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. Liao and Barooah (2013a, b), Schenato and Fiorentin (2011) and Stanković, Stanković, and Johansson (2012). 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 the 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*, $i = 1, \dots, n$.

Distributed time synchronization is aimed at providing a *common virtual clock*, *i.e.*, *equal corrected drifts* g_i and *equal corrected offsets* f_i, $i = 1, \dots, n$, 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.*, Aysal, Yildriz, Sarwate, and Scaglione (2009), Bolognani, Carli, Lovisari, and Zampieri (2012) and Nedić (2011), without global supervision or fusion center. Therefore, each node $j \in \mathcal{N}$ has its own *local communication clock* that ticks according to a Poisson process with rate μ_j, independently of the other nodes. At each tick of its communication clock (denoted by t_b^j, $b = 0, 1, 2, \dots$), node j broadcasts its current state 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 $\{t_l^{j,i}\}$, $l = 0, 1, 2, \dots$, 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* (for physical and technical sources of delays see Chaudhari et al. (2008), Choi et al. (2012), Freris et al. (2011), Leng and Wu (2011) and Xiong and Kishore (2009)). We assume in the sequel that

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

where $\bar{\delta}^{j,i}$ is constant, while $\eta_i(\bar{t}_l^{j,i})$ 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 any node in the network; we assume, as usually, only one tick at a given time t (Nedić, 2011).

2.2. Drift correction 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 $m \in \{0,\dots,l-1\}$,

$$\Delta\tau_j(t_l^{j,i}) = \tau_j(t_l^{j,i}) - \tau_j(t_m^{j,i}) = \alpha_j\Delta t_l^{j,i} + \Delta\xi_j(t_l^{j,i}),$$
$$\Delta\tau_i(\bar{t}_l^{j,i}) = \alpha_i\Delta\bar{t}_l^{j,i} + \Delta\xi_i(\bar{t}_l^{j,i}),$$

$\Delta t_l^{j,i} = t_l^{j,i} - t_m^{j,i}$, $\Delta\xi_j(t_l^{j,i}) = \xi_j(t_l^{j,i}) - \xi_j(t_m^{j,i})$, $\Delta\bar{t}_l^{j,i} = \bar{t}_l^{j,i} - \bar{t}_m^{j,i} = \Delta t_l^{j,i} + \Delta\delta_l^{j,i}$, 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. In this paper we shall consider the following three characteristic cases (which we denote as *AlgDrift.a*, *AlgDrift.b* and *AlgDrift.c* (see Algorithm 1)):

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

Remark 1. In *AlgDrift.a* and *AlgDrift.c* the required memory is finite; in *AlgDrift.a* the memory requirement is determined by L (in the algorithms from Schenato and Fiorentin (2011) and Stanković et al. (2012), $L = 1$). In *AlgDrift.b* and *AlgDrift.c* the increment length is unbounded. *AlgDrift.c* is based on the idea from Tian (2015), Tian et al. (2016, 2017), and uses a fixed initial time instant $m = l_0$. However, in *AlgDrift.b* we have that $\lim_{l\to\infty} m = \infty$ and $\lim_{l\to\infty}(l - m) = \infty$; it will be seen below that, as a consequence, the corresponding algorithm has the highest convergence rate.

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

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

where γ_{ij} are *a priori* chosen *nonnegative weights* (see the discussion 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 corresponding pseudocode is presented as Algorithm 1. 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 $\hat{\psi}_i^a(\bar{t}_l^{j,i}) = \alpha_i\{[\hat{g}_j(t_l^{j,i}) - \hat{g}_i(\bar{t}_l^{j,i})]\Delta t_l^{j,i} + \frac{1}{\alpha_j}\hat{g}_j(t_l^{j,i})\Delta\xi_j(t_l^{j,i}) - \frac{1}{\alpha_i}\hat{g}_i(\bar{t}_l^{j,i})\Delta\xi_i(\bar{t}_l^{j,i}) - \hat{g}_i(\bar{t}_l^{j,i})\Delta\eta_i(\bar{t}_l^{j,i})\}$.

Algorithm 1 *AlgDrift.a*, *AlgDrift.b* and *AlgDrift.c*

```
for All the nodes i ∈ 𝒩 do
  Initialize â_i(t̄_0^{j,i}) = 1
end for
loop
  if Tick t_b^j of a local communication clock of a node j ∈ 𝒩 then
    Read the current local time value τ_j(t_b^j)
    Broadcast τ_j(t_b^j) and â_j(t_b^j) to the out-neighbors 𝒩_j^+
  end if
end loop
loop
  if A message received by a node i ∈ 𝒩 from a node j ∈ 𝒩_j^- (at
  absolute time t̄_l^{j,i}) then
    if The first message from the node j then
      Save the received initial local time of node j τ_j(t_0^{j,i})
      Read and save the initial local time τ_i(t̄_0^{j,i})
    else
      Read the current local time value τ_i(t̄_l^{j,i})
      Calculate Δτ̂_j(t_l^{j,i}) and Δτ̂_i(t̄_l^{j,i}) according to (6) and (7),
      where m = l − L for AlgDrift.a, m = ⌊νl⌋ for AlgDrift.b,
      and m = 0 for AlgDrift.c
      Calculate a new estimate of the drift correction parameter
      according to (5)
    end if
  end if
end loop
```

Remark 2. The proposed Algorithm 1 is independent of both offset and offset correction. It does not belong to the class of CBTS algorithms (Tian et al., 2016): it is structurally different and simpler, not requiring the step of *relative drift* estimation, which introduces higher order dynamics and an additional nonlinearity.

2.3. Offset correction algorithm

The proposed recursion for updating parameter b_i in (2) is based on the following error function:

$$\bar{\varphi}_i^b(\bar{t}_l^{j,i}) = \bar{\tau}_j(t_l^{j,i}) - a_jT_j(t_l^{j,i}) - (\bar{\tau}_i(\bar{t}_l^{j,i}) - a_iT_i(\bar{t}_l^{j,i})) + c_i, \tag{9}$$

$j = 1,\dots,n$, $i \in \mathcal{N}_j^+$, where

$$T_j(t_l^{j,i}) = \Delta\tau_j(t_l^{j,i})|_{m=0},\quad T_i(\bar{t}_l^{j,i}) = \Delta\tau_i(\bar{t}_l^{j,i})|_{m=0}, \tag{10}$$

while c_i is an additional *delay compensation parameter*.

Using (9), we come up with the following updates for b_i and c_i:

$$\hat{b}_i(\bar{t}_l^{j,i+}) = \hat{b}_i(\bar{t}_l^{j,i}) + \varepsilon_i^b(\bar{t}_l^{j,i})\gamma_{ij}\hat{\varphi}_i^b(\bar{t}_l^{j,i}) \tag{11}$$
$$\hat{c}_i(\bar{t}_l^{j,i+}) = \hat{c}_i(\bar{t}_l^{j,i}) - \varepsilon_i^b(\bar{t}_l^{j,i})\gamma_{ij}\hat{\varphi}_i^b(\bar{t}_l^{j,i}) \tag{12}$$

where $\hat{\varphi}_i^b(\bar{t}_l^{j,i}) = \hat{\tau}_j(t_l^{j,i}) - \hat{a}_j(t_l^{j,i})T_j(t_l^{j,i}) - (\hat{\tau}_i(\bar{t}_l^{j,i}) - \hat{a}_i(\bar{t}_l^{j,i})T_i(\bar{t}_l^{j,i})) + \hat{c}_i(\bar{t}_l^{j,i})$, with

$$\hat{\tau}_j(t_l^{j,i}) = \hat{a}_j(t_l^{j,i})\tau_j(t_l^{j,i}) + \hat{b}_j(t_l^{j,i}), \tag{13}$$
$$\hat{\tau}_i(\bar{t}_l^{j,i}) = \hat{a}_i(\bar{t}_l^{j,i})\tau_i(\bar{t}_l^{j,i}) + \hat{b}_i(\bar{t}_l^{j,i}). \tag{14}$$

The initial estimates are supposed to be $\hat{b}_i(\bar{t}_0^{j,i}) = 0$ and $\hat{c}_i(\bar{t}_0^{j,i}) = 0$. The estimates of the drift correction parameters a_i can be generated by any appropriate algorithm; when it is generated by (5), the result is the new time synchronization algorithm.

In terms of $\hat{g}_i(\cdot) = \hat{a}_i(\cdot)\alpha_i$ and $\hat{f}_i(\cdot) = \hat{a}_i(\cdot)\beta_i + \hat{b}_i(\cdot)$, (11) and (12) become:

$$\hat{f}_i(\bar{t}_l^{j,i+}) + \Delta\hat{g}_i(\bar{t}_l^{j,i+}) = \hat{f}_i(\bar{t}_l^{j,i}) + \varepsilon_i^b(\bar{t}_l^{j,i})\gamma_{ij}\hat{\psi}_i^b(\bar{t}_l^{j,i}), \tag{15}$$

$$\hat{c}_i(\bar{t}_l^{j,i+}) = \hat{c}_i(\bar{t}_l^{j,i}) - \varepsilon_i^b(\bar{t}_l^{j,i})\gamma_{ij}\hat{\psi}_i^b(\bar{t}_l^{j,i}), \tag{16}$$

where $\Delta\hat{g}_i(\bar{t}_l^{j,i+}) = \frac{\beta_i}{\alpha_i}[\hat{g}_i(\bar{t}_l^{j,i}) - \hat{g}_i(\bar{t}_l^{j,i+})]$ and $\hat{\psi}_i^b(\bar{t}_l^{j,i}) = [\hat{g}_j(t_l^{j,i}) - \hat{g}_i(\bar{t}_l^{j,i})]t_0^{j,i} + \hat{f}_j(t_l^{j,i}) - \hat{f}_i(\bar{t}_l^{j,i}) - \hat{g}_i(\bar{t}_l^{j,i})[\bar{\delta}^{i,j} + \eta_i(\bar{t}_0^{j,i})] + \hat{c}_i(\bar{t}_l^{j,i}) + \frac{1}{\alpha_j}\hat{g}_j(t_l^{j,i})\xi_j(t_0^{j,i}) - \frac{1}{\alpha_i}\hat{g}_i(\bar{t}_l^{j,i})\xi_i(\bar{t}_0^{j,i})$.

A *consensus-based modification* of (12) and (16) is obtained by replacing $\hat{c}_i(\bar{t}_l^{j,i})$ at the right hand side of (12) and (16) by the following *convex combination*

$$\hat{c}_i^{con}(\bar{t}_l^{j,i}) = \sigma_i\hat{c}_i(\bar{t}_l^{j,i}) + (1-\sigma_i)\hat{c}_j(\bar{t}_l^{j,i}), \tag{17}$$

with tuning parameter $0 < \sigma_i \leq 1$. This modification is motivated by a realistic assumption that the communication delays do not differ very much, aiming at achieving lower dissipation of the convergence points for $\hat{f}_i(\cdot)$ (see Remark 8). We refer to this algorithm as *AlgOffset.b*. The pseudocode of *AlgOffset.a* and *AlgOffset.b* is presented as Algorithm 2.

Algorithm 2 *AlgOffset.a* and *AlgOffset.b*

```
for All the nodes i ∈ N do
    Initialize b̂_i(t̄_0^{j,i}) = 0 and ĉ_i(t̄_0^{j,i}) = 0,
end for
loop
    if Tick t_b^j of a local communication clock of a node j ∈ N then
        Read the current local time value τ_j(t_b^j)
        Broadcast τ_j(t_b^j), â_j(t_b^j), b̂_j(t_b^j) and ĉ_j(t_b^j) to the out-neighbors N_j^+
    end if
end loop
loop
    if A message received by a node i ∈ N from a node j ∈ N_j^- (at absolute time t̄_l^{j,i}) then
        Read the current local time value τ_i(t̄_l^{j,i})
        Calculate τ̂_j(t_l^{j,i}) and τ̂_i(t̄_l^{j,i}) using (13) and (14)
        Calculate T_j(t_l^{j,i}) and T_i(t̄_l^{j,i}) using (10)
        Calculate new estimates of the offset correction parameters
        according to (11) and (12) (and (17) for the algorithm AlgOffset.b)
    end if
end loop
```

Remark 3. The error function (9) is obtained from the basic error function $\varphi_i^b(\bar{t}_l^{j,i})^0 = \bar{\tau}_j(t_l^{j,i}) - \bar{\tau}_i(\bar{t}_l^{j,i})$ (typically used in the literature (Tian et al., 2016)), after *two important modifications*. The first one introduces two easily computable terms $T_j(t_l^{j,i})$ and $T_i(\bar{t}_l^{j,i})$, the role of which is to cope with the unboundedly increasing term $t_l^{j,i}$ in the expression for $\varphi_i^b(\bar{t}_l^{j,i})^0$ in such a way that it becomes replaced by the bounded term $t_0^{j,i}$ in $\varphi_i^b(\bar{t}_l^{j,i})$, while the second one introduces a *new parameter* $\hat{c}_i(\bar{t}_l^{j,i})$, coping directly with the effects of communication delays and enabling convergence of the offset correction estimates.

Remark 4. The proposed time synchronization algorithm requires small communication and computation efforts. At each tick $t_l^{j,i}$, a packet is sent by the jth node to its neighbors $i \in \mathcal{N}_j^+$, containing the current local time $\tau_j(t_l^{j,i})$ and the current local drift and offset correction parameter estimates $\hat{a}_j(t_l^{j,i})$, $\hat{b}_j(t_l^{j,i})$ and $\hat{c}_j(t_l^{j,i})$. After receiving this packet, the neighbors calculate the corresponding $\Delta\hat{\tau}_j(t_l^{j,i})$, $\Delta\hat{\tau}_i(\bar{t}_l^{j,i})$, $\hat{\tau}_j(t_l^{j,i})$, $\hat{\tau}_i(\bar{t}_l^{j,i})$, $T_j(t_l^{j,i})$ and $T_i(\bar{t}_l^{j,i})$, and update their own parameter estimates according to (5), (11), (12) and (17). The same procedure is repeated after each new tick of any of the nodes.

2.4. Global model

All communications in the network can be considered to be driven by a *global virtual communication clock*, with the rate equal to $\mu_c = \sum_{i=1}^n \mu_i$, that ticks whenever any local communication clock ticks (*e.g.*, Aysal et al. (2009) and Nedić (2011)). We shall assume that a unique iteration number k is assigned to each *update of local parameters*, and, *vice versa*, that each k is connected to an update of ith node at the continuous time instant $\bar{t}_l^{j,i}$ for some j and l. In such a way, at a click of jth communication clock we have $N(j)$ consecutive updates or iterations, $N(j) \leq |\mathcal{N}_j^+|$. Following analogous approaches in Nedić (2011) and Tian et al. (2016), 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 at node j. Let $\hat{g}(k) = [\hat{g}_1(k)\ldots\hat{g}_n(k)]^T$, $\hat{f}(k) = [\hat{f}_1(k)\ldots\hat{f}_n(k)]^T$ and $\hat{c}(k) = [\hat{c}_1(k)\ldots\hat{c}_n(k)]^T$, 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 $\hat{c}_\mu(k) = \hat{c}_\mu(\bar{t}_l^{j,i})$, $\mu = 1, \ldots, n$, Then, (8) gives

$$\hat{g}(k+1) = \hat{g}(k) + \varepsilon^a(k)Z(k)\hat{g}(k), \tag{18}$$

where $\hat{g}(k+1) = [\hat{g}_1(\bar{t}_l^{j,1+})\ldots\hat{g}_n(\bar{t}_l^{j,n+})]^T$, $\varepsilon^a(k) = \mathrm{diag}\{\varepsilon_1^a(k), \ldots, \varepsilon_n^a(k)\}$, $\varepsilon_i^a(k) = \varepsilon_i^a(\bar{t}_l^{j,i})$ (see (5)), $Z(k) = A\Gamma(k)\Delta t(k) + N_g(k)$, $A = \mathrm{diag}\{\alpha_1, \ldots, \alpha_n\}$, $\Gamma(k) = [\Gamma(k)_{\mu\nu}]$, 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 $\Gamma_d(k) = \mathrm{diag}\{\mathrm{diag}\{\gamma_{1j}, \ldots, \gamma_{nj}\}\omega(k)\}$, $\omega(k) = [\omega_1(k)\ldots\omega_n(k)]^T$, $\omega_i(k) = 1$, $\omega_\mu(k) = 0$ for $\mu \neq i$, $\Delta\eta_d(k) = \mathrm{diag}\,\Delta\eta(k)$, $\Delta\eta(k) = [\Delta\eta_1(k)\ldots\Delta\eta_n(k)]^T$, $\Delta\xi_d(k) = \mathrm{diag}\,\Delta\xi(k)$ and $\Delta\xi(k) = [\Delta\xi_1(k)\ldots\Delta\xi_n(k)]^T$.

Similarly, from (15) and (16) we obtain

$$\hat{f}(k+1) + \Delta\hat{g}(k+1) = \hat{f}(k) + \varepsilon^b(k)Y(k) \tag{19}$$

$$\hat{c}(k+1) = \hat{c}(k) - \varepsilon^b(k)Y(k), \tag{20}$$

where $\Delta\hat{g}(k+1) = \mathrm{diag}\,\omega(k)(\hat{g}(k+1) - \hat{g}(k))$, $Y(k) = \Gamma(k)\hat{f}(k) + [t^0(k)\Gamma(k) - \Gamma_d(k)\bar{\delta}_d(k) - \Gamma_d(k)\eta_d^0(k) + \Gamma(k)\xi_d^0(k)A^{-1}]\hat{g}(k) + \Gamma_d(k)\hat{c}(k)$, $t^0(k) = t_0^{j,i}$, $\bar{\delta}_d(k) = \mathrm{diag}\,\bar{\delta}(k)$, $\bar{\delta}(k) = [\bar{\delta}^{j,1}\ldots\bar{\delta}^{j,n}]^T$, $\eta_d^0(k) = \mathrm{diag}\,\eta^0(k)$, $\eta^0(k) = [\eta_1^0(k)\ldots\eta_n^0(k)]^T$, $(\eta_i^0(k) = \eta_i(\bar{t}_0^{j,i}))$, $\xi_d^0(k) = \mathrm{diag}\,\xi^0(k)$, $\xi^0(k) = [\xi_1^0(k)\ldots\xi_n^0(k)]^T$ $(\xi_j^0(k) = \xi_j(t_0^{j,i})$, $\xi_i^0(k) = \xi_i(\bar{t}_0^{j,i})$; $\{t^0(k)\}$, $\{\eta^0(k)\}$ and $\{\xi^0(k)\}$ are random sequences with finite sets of possible realizations composed of $t_0^{j,i}$, $\eta_i(\bar{t}_0^{j,i})$ and $\xi_j(t_0^{j,i})$ (or $\xi_i(\bar{t}_0^{j,i})$), obtained at each k by choosing j and i at random.

In *AlgOffset.b*, $\hat{c}(k)$ is replaced by $\hat{c}^{con}(k) = C(k)\hat{c}(k)$, where $C(k) = [C(k)_{\mu\nu}]$, with $C(k)_{\mu\mu} = \sigma_\mu$ and $C(k)_{\mu j} = 1 - \sigma_\mu$ for all $\mu \in \mathcal{N}_j^+$, with $C(k)_{\mu\nu} = 0$ otherwise.

3. Convergence analysis

3.1. Preliminaries

Within the exposed general setting, we additionally assume:

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

(A2) $\{\xi_i(k)\}$ and $\{\eta_i(k)\}$, $i = 1, \ldots, n$, 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*, and
$\varepsilon_i^b(k) = \varepsilon_i(k)|_{\zeta=\zeta''}$ for *AlgOffset.a* and *AlgOffset.b*,
where $\varepsilon_i(k) = \nu_i(k)^{-\zeta}$, $\nu_i(k) = \sum_{m=1}^{k} I\{$ node i received a message at instant $m\}$, 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 5. (A1) implies that graph $\mathcal{G}$ has a center node from which all the remaining nodes are reachable (Olfati-Saber, Fax, & Murray, 2007; Stanković, Stanković, & Johansson, 2015). (A2) is a standard assumption; boundedness is introduced for the sake of making derivations easier. (A3) is very important for practice: 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). The choice of ζ' and ζ'' is standard for stochastic approximation algorithms.

Asymptotical behavior of the step size is defined by the following lemma. Proofs of all the lemmas and theorems are given in the Appendix.

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

$$\varepsilon_i(k) = \frac{1}{k^{\zeta}}\left(\frac{\bar{N}}{p_i}\right)^{\zeta} + \tilde{\varepsilon}_i(k), \tag{21}$$

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

Properties of the matrix $\Gamma(k)$ defined in the previous section are essential for convergence of (18)–(20); its expectation $\bar{\Gamma} = E\{\Gamma(k)\}$ has the central role in the analysis. It has the form of a weighted Laplacian matrix for $\mathcal{G}$:

$$\bar{\Gamma} = \begin{bmatrix} -\sum_{j,j\neq 1}\gamma_{1j}\pi_{1j} & \gamma_{12}\pi_{12} & \cdots & \gamma_{1n}\pi_{1n} \\ \gamma_{21}\pi_{21} & -\sum_{j,j\neq 2}\gamma_{2j}\pi_{2j} & \cdots & \gamma_{2n}\pi_{2n} \\ & & \ddots & \\ \gamma_{n1}\pi_{n1} & \gamma_{n2}\pi_{n2} & \cdots & -\sum_{j,j\neq n}\gamma_{nj}\pi_{nj} \end{bmatrix} \tag{22}$$

($\gamma_{ij} = 0$ 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 (18) and Lemma 1, we shall focus on $B(k) = P^{-\zeta}A\Gamma(k)$ and $\bar{B} = E\{B(k)\} = P^{-\zeta}A\bar{\Gamma}$ ($P^{-\zeta} = \bar{N}^{\zeta}\mathrm{diag}\{p_1^{-\zeta}, \ldots, p_n^{-\zeta}\}$).

Lemma 2 (*Stanković et al., 2015*). *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 $\mathbf{1} = [1\ldots 1]^T$. Then,*

$$T^{-1}\bar{B}T = \begin{bmatrix} 0 & 0_{1\times(n-1)} \\ 0_{(n-1)\times 1} & \bar{B}^* \end{bmatrix}, \tag{23}$$

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

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

$$R^g\bar{B}^* + \bar{B}^{*T}R^g = -Q^g, \tag{24}$$

for any given $Q^g > 0$. It also follows from the derivation of (23) that $T^{-1}B(k)T = \begin{bmatrix} 0 & B_1(k) \\ 0_{(n-1)\times 1} & B_2(k) \end{bmatrix}$, with $E\{B_1(k)\} = 0$ and $E\{B_2(k)\} = \bar{B}^*$.

Properties of the random variable $\Delta t(k)$ are important for understanding of the properties of (18).

Lemma 3. *$E\{\Delta t(k)\} = \frac{1}{\mu_j}\frac{l-m}{p_{ij}}$, $\mathrm{var}\{\Delta t(k)\} = \frac{1}{\mu_j^2}\frac{l-m}{p_{ij}}$, 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 $l \sim \pi_{ij}k$.*

3.2. Convergence of the drift correction algorithm

Coming back to (18), we first insert $\varepsilon^a(k)$ from (21). 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 (21), we obtain

$$\begin{aligned}\tilde{g}(k+1)^{[1]} =& \tilde{g}(k)^{[1]} + \frac{1}{k^{\zeta}}F_1(k)\Delta t(k)\tilde{g}(k)^{[2]} \\ &+ \frac{1}{k^{\zeta}}H_1(k)\tilde{g}(k)\end{aligned} \tag{25}$$

$$\begin{aligned}\tilde{g}(k+1)^{[2]} =& \{I + \frac{1}{k^{\zeta}}[\bar{B}^* + F_2(k)]\Delta t(k)\}\tilde{g}(k)^{[2]} \\ &+ \frac{1}{k^{\zeta}}H_2(k)\tilde{g}(k),\end{aligned} \tag{26}$$

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 $T^{-1}P^{-c}N_g(k)T = \begin{bmatrix} H_1(k) \\ H_2(k) \end{bmatrix}$.

We now have the main convergence result for the drift correction algorithm.

Theorem 1. *Let assumptions (A1)–(A3) be satisfied. Then, $\tilde{g}(k)^{[1]}$ from (25) converges to a random variable χ^* with bounded second moment, and $\tilde{g}(k)^{[2]}$ from (26) to zero in the mean square sense and w.p.1; in other words, $\hat{g}(k)$ generated by (18) 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 corrected 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 (18) will be studied through the behavior of $\tilde{g}(k)^{[2]}$ in (26), using the methodology of Chen (2002, Chap. 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 (26), 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*
– $\zeta d < \zeta'$ (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, $q = \frac{1-\nu}{\mu_c}$ *for* AlgDrift.b *and* $q = \frac{1}{\mu_c}$ *for* AlgDrift.c.

Remark 6. The above result indicates that the *AlgDrift.b* gives the best performance: in the case when $\zeta' < 1$, the important condition $\zeta d > 1$ is achieved, enabling convergence of local corrected clocks to a common virtual clock (see Corollary 1). This is a consequence of the variable left end of the intervals $[m, l]$, which introduces a white noise term in the recursion (8) (see the proof of Theorem 2); at the same time, unbounded increase of interval length $l - m$ ensures an effectively increasing signal-to-noise ratio, together with appropriate averaging. Theoretically, *AlgDrift.c* with fixed m does not allow this effect. However, in practice, it is sufficient to choose $l - m = L$ large enough and to apply *AlgDrift.a*, avoiding unbounded increase of memory inherent to *AlgDrift.b*. It will be demonstrated in Section 4 by simulation that, practically, the best results can be obtained by *AlgDrift.a* for L moderately high.

Notice that the convergence rate $\zeta d > 1$ is not achievable by structurally different CBTS algorithms discussed in Schenato and Fiorentin (2011), Tian (2015), Tian et al. (2016, 2017).

An important conclusion resulting from Theorems 1 and 2 is that

$$\hat{g}(k) = \chi(k)\mathbf{1} + \hat{g}(k)^{[2]}, \text{ w.p.1} \tag{27}$$

where $\chi(k) = \tilde{g}(k)^{[1]}$ and $\hat{g}(k)^{[2]} = T_{n\times(n-1)}\tilde{g}(k)^{[2]}$, with $\chi(k) = \chi^* + o(1)$ and $\|\hat{g}(k)^{[2]}\| = o(\frac{1}{k^{\zeta d}})$. The last relation is fundamental for the convergence analysis of the offset correction algorithm given below.

3.3. Convergence of the offset correction algorithm

We start the analysis by introducing the following expressions in (19) and (20):

$$\begin{aligned} &\Gamma(k) = \bar{\Gamma} + \tilde{\Gamma}(k), \quad \Gamma_d(k) = \bar{\Gamma}_d + \tilde{\Gamma}_d(k),\\ &\xi^0(k) = \bar{\xi}^0 + \tilde{\xi}^0(k), \quad \eta^0(k) = \bar{\eta}^0 + \tilde{\eta}^0(k),\\ &\bar{\delta}(k) = \bar{\delta}^0 + \tilde{\delta}(k), \quad t^0(k) = \bar{t}^0 + \tilde{t}^0(k), \end{aligned} \tag{28}$$

where $\bar{\Gamma} = E\{\Gamma(k)\}$, $\bar{\Gamma}_d = E\{\Gamma_d(k)\}$, $\bar{\xi}^0 = E\{\xi^0(k)\} = \sum_{j=1}^n \xi(t_0^{j,i})\pi_j$, $\bar{\xi}_d^0 = \text{diag}\,\bar{\xi}^0$, $\bar{\eta}^0 = E\{\eta^0(k)\} = \sum_{j=1}^n \eta(\bar{t}_0^{j,i})\pi_j$, $\bar{\delta}^0 = E\{\bar{\delta}(k)\} = \sum_{j=1}^n [\bar{\delta}_0^{1,j} \ldots \bar{\delta}_0^{n,j}]^T\pi_j$ and $\bar{t}^0 = E\{t^0(k)\} = \sum_{j=1}^n t_0^{j,i}\pi_j$. Therefore, $\{\tilde{\Gamma}(k)\}$, $\{\tilde{\Gamma}_d(k)\}$, $\{\tilde{\xi}^0(k)\}$, $\{\tilde{\eta}^0(k)\}$, $\{\tilde{\delta}^0(k)\}$ and $\{\tilde{t}^0(k)\}$ are zero mean i.i.d. random sequences (due to randomness in determining the transmitting node for a given k).

Theorem 3. *Let assumptions (A1)–(A3) be satisfied and let $\hat{g}(k)$ be generated by AlgDrift.a with $\zeta' \in (\frac{3}{4}, 1)$, and by AlgDrift.b or AlgDrift.c with $\zeta' < 1$. Then, $\hat{f}(k)$, generated by AlgOffset.a using (19), converges to $\hat{f}^*$ and $\hat{c}(k)$ from (20) converges to $\hat{c}^*$ in the mean square sense and w.p.1 for all $\zeta'' \in (\frac{1}{2}, 1]$ in the case of AlgDrift.b and AlgDrift.c, and for all $\zeta'' \in (\frac{1}{2}, 1]$, $\zeta'' > \frac{3}{2} - \zeta'$, in the case of AlgDrift.a; $\hat{f}^*$ and $\hat{c}^*$ satisfy the equation*

$$[\bar{\Gamma} \vdots \bar{\Gamma}_d]\hat{h}^* = 0, \tag{29}$$

where $\hat{h}^* = [(\hat{f}^* + \chi^*\bar{\xi}_d^0 A^{-1}\mathbf{1})^T \vdots (\hat{c}^* - \chi^* A(\bar{\eta}^0 + \bar{\delta}))^T]^T$.

Remark 7. The rate of convergence of $\hat{g}(k)$ to consensus influences $\hat{f}(k)$ in (19) directly, through the term $\Delta\hat{g}(k+1)$, and indirectly, through the remaining terms depending on $\hat{g}(k)$. According to Theorem 3, all the above proposed algorithms for drift correction can be utilized for offset correction under appropriate assumptions. Notice that the results of Theorem 3 also hold if we use any $\hat{g}(k)$ providing sufficient convergence rate to consensus, according to (27).

Theorem 4. *Let the assumptions of Theorem 3 hold. Then $\hat{f}(k)$ and $\hat{c}(k)$, generated by the algorithm AlgOffset.b ((19), (20) with consensus iterations on $\hat{c}(k)$ using (17)), converge in the mean square sense and w.p.1 to $\hat{f}^*$ and $\hat{c}^* = \hat{c}^{con}\mathbf{1}$, respectively ($\hat{c}^{con}$ is a scalar), where $\hat{f}^*$ and $\hat{c}^{con}$ satisfy the equation $M_1^{con}\hat{h}^{con} = 0$, where*

$$M_1^{con} = \begin{bmatrix} \bar{\Gamma} & \vdots & \text{vec}\{\bar{\Gamma}_d\} \\ \cdots & & \cdots \\ -\sum_{i=1}^n \bar{\phi}_i\bar{\Gamma}^{(i)} & \vdots & -\sum_{i=1}^n \bar{\phi}_i \text{vec}\{\bar{\Gamma}_d\}_i \end{bmatrix}, \tag{30}$$

$\hat{h}^{con} = [(\hat{f}^* + \chi^*\bar{\xi}_d^0 A^{-1}\mathbf{1})^T \vdots \hat{c}^{con} - \sum_{i=1}^n \bar{\phi}_i\chi^*(A\bar{\eta}^0 + A\bar{\delta})_i]^T$, *with* $\bar{\phi} = [\bar{\phi}_1 \ldots \bar{\phi}_n]$, $\bar{\phi}\bar{C} = \bar{\phi}$ *and* $\bar{C} = E\{C(k)\}$; $\bar{\Gamma}^{(i)}$ *denotes i-th row of the matrix* $\bar{\Gamma}$, *and* $\text{vec}\{\bar{\Gamma}_d\}_i$ *i-th element of* $\text{vec}\{\bar{\Gamma}_d\}$.

Remark 8. Possible convergence points of the proposed offset correction algorithms depend not only on the network and noise properties, but also on the convergence point of the drift estimation algorithm. In general, the corrected offsets do not converge to the same point for all the nodes. However, a comparison between (29) and (30) indicates clearly that *AlgOffset.b* can achieve lower dispersion of the components of $\hat{f}^*$ within $\hat{h}^{con}$ due to lower number of degrees of freedom. Simulation results presented in Section 4 confirm this statement.

3.4. Special cases

When communication delays and measurement noise can be neglected, the algorithm *AlgOffset.b* ((18)–(20) with (17)) is able *to achieve consensus* on both corrected drifts $\hat{g}_i(k)$ and corrected offsets $\hat{f}_i(k)$ (which follows directly from (30)). However, according to Theorem 3, *AlgOffset.a* still does not guarantee convergence of $\hat{f}(k)$ to consensus, due to additional degrees of freedom in (29).

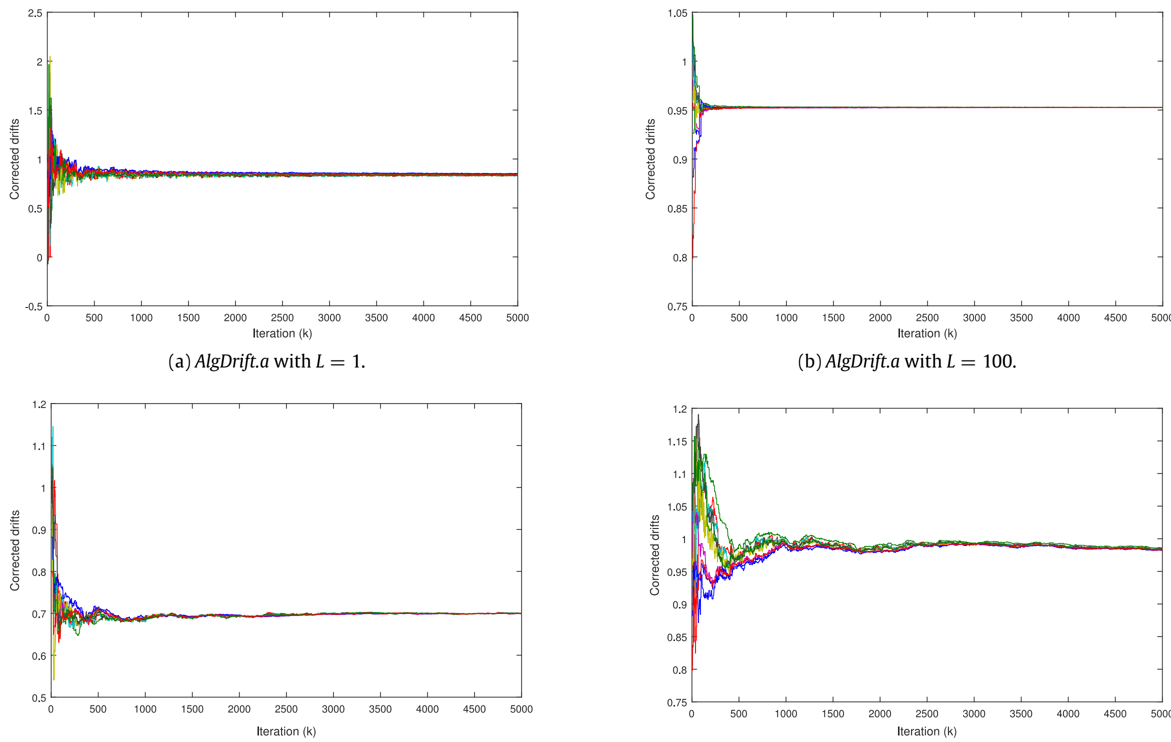
When the stochastic terms $\xi(\cdot)$ and $\eta(\cdot)$ are equal to zero, it is possible to achieve exponential convergence rate by adopting constant step size in *AlgDrift.a*, *AlgOffset.a* and *AlgOffset.b*, and $\varepsilon_i^a(k) = \varepsilon'\nu_i(k)^{-1}$ in *AlgDrift.b* and *AlgDrift.c*. However, the offset correction algorithm again does not provide consensus, in general.

When, in addition, the delay is equal to zero, it is possible to achieve *exponential convergence to consensus* for both drifts and offsets, using *AlgDrift.a* for drift estimation and (11) for offset estimation, where simply $\hat{\varphi}_i^b(\bar{t}_l^{j,i}) = \hat{\tau}_j(t_l^{j,i}) - \hat{\tau}_i(\bar{t}_l^{j,i})$. This result was obtained for the first time in Stanković et al. (2012) for pseudo periodic communication sequences.

3.5. Convergence to a common virtual clock

As pointed out, the general aim of clock synchronization is convergence of local corrected times to a common virtual time. In view of the above results, we have:

Corollary 1. *Let (A1)–(A3) be satisfied, with $\zeta' < 1$. Then, for* AlgDrift.b *and either* AlgOffset.a *or* AlgOffset.b, $\sup_{i,j} \Delta\hat{\tau}_{i,j}(k) = \hat{\tau}_i(k) - \hat{\tau}_j(k)$ *is bounded in the mean square sense and w.p.1.*



(a) *AlgDrift.a* with $L = 1$.
(b) *AlgDrift.a* with $L = 100$.
(c) *AlgDrift.b* with $\nu = 1/2$.
(d) *AlgDrift.c*.

Fig. 1. Corrected drifts: in spite of the theoretically proved asymptotic superiority of *AlgDrift.b*, *AlgDrift.a* with $L = 100$ practically achieves the best rate of convergence to consensus and noise immunity.

Remark 9. As

$$\Delta\hat{\tau}_{i,j}(k) = [\hat{g}_i(k) - \hat{g}_j(k)]t(k) + \hat{f}_i(k) - \hat{f}_j(k), \quad (31)$$

we have, according to Theorem 2, that for $\zeta' < 1$ the first term at the right-hand side of (31) tends to zero in the case of *AlgDrift.b*. For *AlgDrift.a* and *AlgDrift.c* convergence of $\Delta\hat{\tau}_{i,j}(k)$ is theoretically not achievable; however, *AlgDrift.a* with L large enough will work in practice (see Section 4). Notice that the differences $\hat{f}_i(k) - \hat{f}_j(k)$ remain bounded in the mean square sense and w.p.1., having in mind that the offsets $\hat{f}_i(k)$ are bounded by virtue of Theorems 3 and 4 (the latter statement does not hold for the algorithm proposed in Tian (2017)).

3.6. Tuning network weights; Flooding scheme

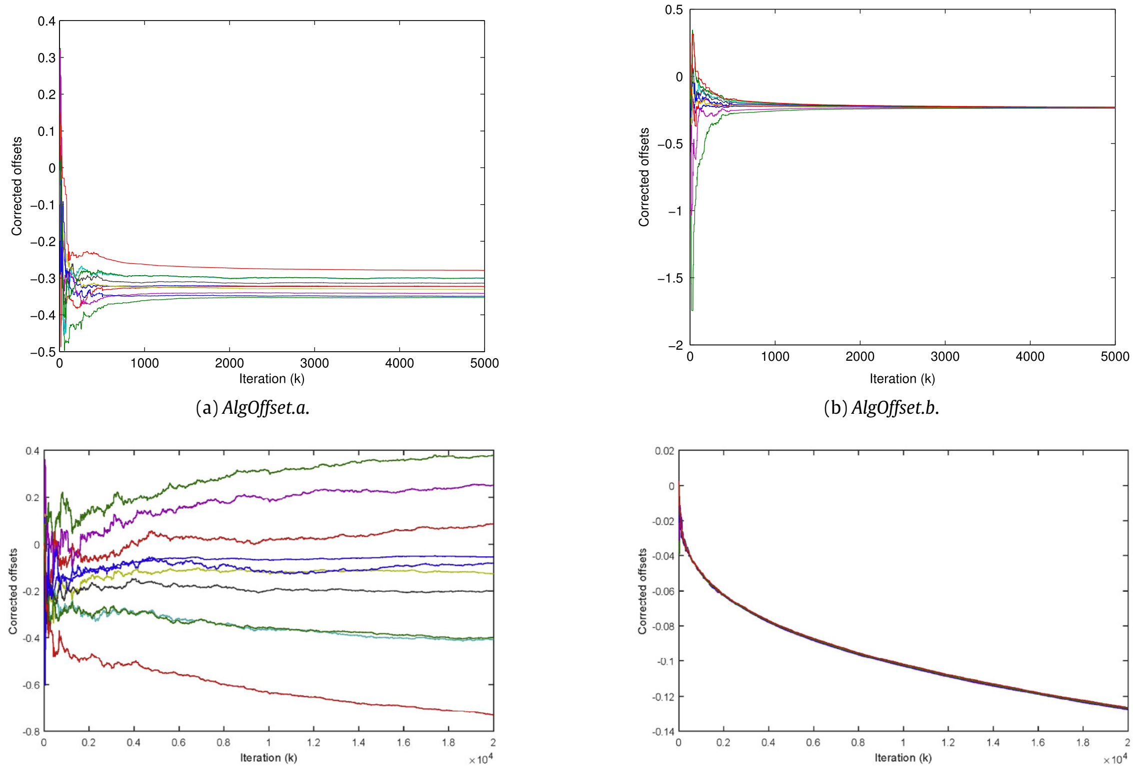
Coefficient γ_{ij} in (5) and (11) is the weight of the update at node i, occurring as a consequence of a tick at node j, $i, j = 1, \dots, n$. If one wishes to express high confidence in the precision of the clock at node i, there are three basic implementations: 1) to increase the Poisson rate μ_i, 2) to increase weights γ_{ji}, $j = 1, \dots, n$, ; (3) to decrease weights γ_{ij}, $j = 1, \dots, n$. The first way clearly gives more weight to the sender. The second way is related to node i as a receiver, while the third way implies lower increments of the local parameter changes. In the limit of the last case, node i does not update its parameters ($\gamma_{ij} = 0, j = 1, \dots, n$), and becomes a *reference node*. The whole algorithm becomes in such a way an algorithm of *flooding type* (Maroti, Kusy, Simon, & Ledeczi, 2004; Su & Akyildiz, 2005; Wu, Chaudhari, & Serpedin, 2011).

Corollary 2. *Let the assumptions of Theorem 1 be satisfied. Let node λ be a center node in $\mathcal{G}$, with the corrected drift $\hat{g}^*_\lambda$. Then, after setting $\mathcal{N}^-_\lambda = \emptyset$ (or $\gamma_{\lambda j} = 0, j = 1, \dots, n$), algorithm (5) provides convergence of all the corrected drifts $\hat{g}_i(k)$, $i = 1, \dots, n$, $i \neq \lambda$, to $\hat{g}_\lambda$ in the mean square sense and w.p.1.*

4. Simulations

In this section we present simulation results providing a practical insight into the proposed algorithm. The assumed network topology corresponds to a modification of the Geometric Random Graphs (Gupta & Kumar, 2006). 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 roughly 10 percent of the original two-way communications into one-way communications, while satisfying assumption (A1). 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 standard deviation σ specified later. It has been adopted that $\zeta' = \zeta'' = 0.99$ and that the communication dropouts occur according to the probability $p_{ij} = 0.9$.

Typical behavior of the corrected drifts generated by *AlgDrift.a* ($L = 1$ and $L = 100$), *AlgDrift.b* ($\nu = \frac{1}{2}$) and *AlgDrift.c* ($l_0 = 0$) in the presence of stochastic delays and measurement noise with $\sigma = 0.05$ is presented in Fig. 1 for a network with ten nodes. Convergence to consensus can be clearly observed. Analogous schemes from the literature (*e.g.*, Tian et al., 2016) cannot achieve such a performance. The algorithm proposed in Schenato and Fiorentin (2011) is very sensitive to noise and practically inapplicable under the given conditions, while the algorithm from Tian (2015) achieves results similar to the ones obtained by *AlgDrift.c*, but with typically lower convergence rate. 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 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*,



(a) *AlgOffset.a.*

(b) *AlgOffset.b.*

(c) *AlgOffset.a* with $T_i = T_j = 0$.

(d) *AlgOffset.a* with $c_i = 0$.

Fig. 2. Corrected offsets: *AlgOffset.b*, which includes consensus iterations on $\hat{c}(k)$, has lower dispersion of the converged estimates than *AlgOffset.a*. Corrected offsets do not converge if we set $T_i = 0$ (c) or $c_i = 0$ (d), $i = 1, \dots, n$, illustrating the importance of the introduced modifications in the error function (9).

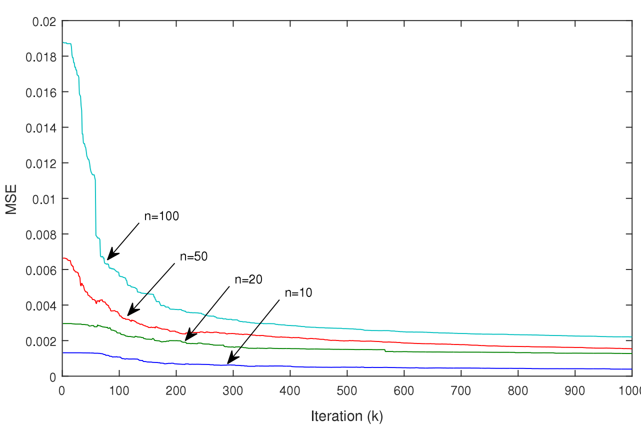


Fig. 3. Mean square disagreement for networks with 10, 20, 50 and 100 nodes.

with a suitably selected memory length L; it represents the best compromise between the signal to noise ratio and computational complexity.

Typical behavior of the proposed offset correction algorithms *AlgOffset.a* and *AlgOffset.b* is illustrated in Fig. 2(a) and (b); *AlgDrift.a* with $L = 100$ has been used for drift correction. The algorithm *AlgOffset.b* provides lower dispersion of the asymptotic values of the corrected offsets, as expected. Fig. 2(c) and (d) illustrate the importance of introducing $T_j(\cdot)$, $T_i(\cdot)$ and c_i in the definition of $\bar{\varphi}_i^b(\cdot)$ in (9). Fig. 2(c) corresponds to $T_j(\cdot) = T_i(\cdot) = 0$, and Fig. 2(d) to $c_i = 0$. It is evident that the corrected offsets diverge in both cases, indicating that the modifications introduced in (9) are essential for offset correction.

In order to provide an insight into scalability of the proposed algorithm, in Fig. 3 the mean square disagreement between the nodes is presented for networks with 10, 20, 50 and 100 nodes generated at random by the above described procedure. According to, *e.g.*, Borkar (1998), it is possible to distinguish two regions in the represented curves. In the first region, the disagreement between the nodes depends on the number of nodes approximately *linearly* and the convergence rate is fast, nearly exponential. In the second region, when k is large enough, $\nu_i(k)^{-\zeta'}$ becomes small; all curves tend to zero, according to Theorem 1. The disagreement between the nodes increases with the number of nodes, but at a rate *much slower than linear*. As stated in Theorem 2, asymptotic convergence rate is characterized by $O(k^{-\zeta d})$, where the proportionality constant depends not only on the eigenvalues of the matrix $\bar{B}$, but also on the noise level.

Fig. 4 illustrates the rate of convergence to a common virtual clock (see Section 3.5): it represents the mean square disagreement multiplied by $k^{2\rho}$, where the exponent ρ has been chosen to be 1, 1.1 and 1.2 for *AlgDrift.b*, and $\rho = 1$ for *AlgDrift.c*, as indicated in the figure (the offsets are set to zero). The curve corresponding to *AlgDrift.c* does not show convergence.

5. Conclusion

In this paper, a new distributed asynchronous algorithm has been proposed for time synchronization in networks with random communication delays, measurement noise and communication dropouts. A new algorithm has been introduced for drift correction parameter estimation, based on an error function derived from specially defined local time increments. It has been proved, using the stochastic approximation arguments, that this algorithm converges to consensus in the mean square sense and w.p.1 under very general conditions. The algorithm achieves convergence rate

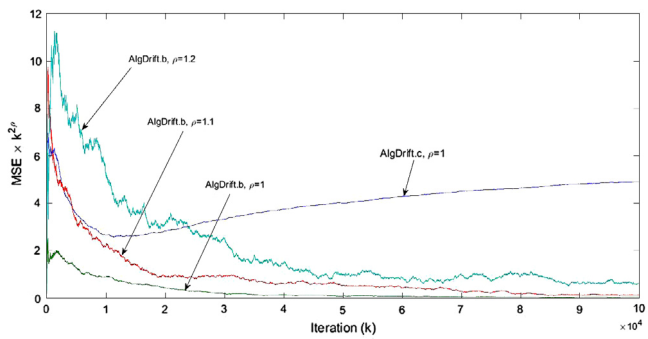


Fig. 4. Rate of convergence to a common virtual clock: the mean square disagreement multiplied by $k^{2\rho}$, for $\rho = 1, 1.1, 1.2$ (for *AlgDrift.b*) and $\rho = 1$ (for *AlgDrift.c*).

superior to all similar schemes, which is important for convergence of local corrected clocks to a common global clock. For offset estimation, a new algorithm has been proposed using a specific error function obtained by modifying local time differences. It has been proved that the corrected offsets converge in the mean square sense and w.p.1. An efficient algorithm for practical applications based on consensus on compensation parameters has also been proposed. It has been also shown that the proposed algorithm can be used as flooding algorithm with one reference node. Simulation results illustrate the presented theoretical results and confirm that the proposed algorithm can represents an efficient tool for practice, outperforming all similar algorithms.

Appendix A. Proof of Lemma 1

According to (A1), $p_i > 0, i = 1, \ldots, n$. Let $\tilde{k}$ correspond to the ticks of the global virtual communication clock. By Lemma 3 in Nedić (2011), for $\tilde{k}$ large enough, $\nu_i(\tilde{k}) = \tilde{k}p_i + \chi_i(\tilde{k})$, where $|\chi_i(\tilde{k})| \leq \kappa \tilde{k}^{\frac{1}{2}+q'}$ w.p.1, $\kappa > 0$. Therefore,

$$\nu_i(\tilde{k})^{-\zeta} = (\tilde{k}p_i + \xi_i(\tilde{k}))^{-\zeta} = (\tilde{k}p_i)^{-\zeta}[1 + O(\frac{\chi_i(\tilde{k})}{\tilde{k}})]$$

Consequently, there exists $\tilde{\varepsilon}_i > 0$ such that $|\frac{1}{\nu_i(\tilde{k})^{\zeta}} - \frac{1}{(\tilde{k}p_i)^{\zeta}}| \leq \tilde{\varepsilon}'_i \frac{1}{\tilde{k}^{\zeta+\frac{1}{2}-q'}}$ w.p.1. The result of Lemma 1 follows after taking into account that $\nu_i(k) = \nu_i(\tilde{k})$ for all iteration numbers k between two consecutive updates at node i, and that $k \sim \tilde{k}\bar{N}$ for $\tilde{k}$ large enough. Formally, $E\{N(j)|j\}$, the average number of updates for a broadcast from node j, can be obtained from the transmission probabilities p_{ij}, while $\bar{N} = \sum_{j=1}^{n} \pi_j E\{N(j)|j\}$. It is essential that $\nu_i(k)^{-\zeta} = O(k^{-\zeta}) + O(k^{-\zeta-\frac{1}{2}+q'})$, where $q' > 0$ is small enough.

Appendix B. 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 (24) for a given $Q^g > 0$.

Decompose $\tilde{g}(k+1)^{[1]}$ from (25) into the sum of zero input and zero state responses, defined by

$$\tilde{g}_1(k+1)^{[1]} = \Pi(k, 1)^{[1]} \tilde{g}(1)^{[1]} \quad \text{(B.1)}$$

and

$$\tilde{g}_2(k+1)^{[1]} = \sum_{\sigma=1}^{k} \frac{1}{\sigma^{\zeta}} \Pi(k, \sigma+1)^{[1]} [F_1(\sigma)\Delta t(\sigma) + H_1(\sigma)^{[2]}] \tilde{g}(\sigma)^{[2]}, \quad \text{(B.2)}$$

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 $V_2^g(k) = E\{(\tilde{g}_2(k)^{[1]})^2\}$.

Analysis of $V_1^g(k)$ starts from introducing $\sum_{i=1}^{n} |\mathcal{N}_i^-|$ infinite subsequences $\{\kappa^{ij}(\upsilon)\}$ of the set of nonnegative integers $\mathcal{I}^+$, $i = 1, \ldots, n$, $j \in \mathcal{N}_i^-$, $\upsilon = 0, 1, 2, \ldots$, where $\kappa^{ij}(\upsilon)$ for a given υ defines the instant k corresponding to an update at node i realized as a consequence of a tick of node j ($\kappa^{ij}(\upsilon_1) < \kappa^{ij}(\upsilon_2)$ for $\upsilon_1 < \upsilon_2$ and $\cup_{i,j}\{\kappa^{ij}(\upsilon)\} = \mathcal{I}^+$). Define $\Pi(k, 1)_s^{[1]} = \prod_{\sigma \in \{\kappa^{ij}(\upsilon)\}, \sigma \leq k}(1 + \frac{1}{\sigma^{\zeta}} H_1(\sigma)^{[1]})$, $s = 1, \ldots, \sum_i |\mathcal{N}_i^-|$, $i = 1, \ldots, n$, $j \in \mathcal{N}_i^-$; consequently, $\prod_s \Pi(k, 1)_s^{[1]} = \Pi(k, 1)^{[1]}$. According to the definition of $N^g(k)$ in (18), for *AlgDrift.a* and *AlgDrift.b*, the zero mean random sequences $\{H_1(\sigma)^{[1]}\}_{\sigma=\{\kappa^{ij}(\upsilon)\}}$ are correlated only with $\{H_1(\sigma)^{[1]}\}_{\sigma=\{\kappa^{ij}(\upsilon-1)\}}$ and $\{H_1(\sigma)^{[1]}\}_{\sigma=\{\kappa^{ij}(\upsilon+1)\}}$, implying that $E\{(\Pi(k, 1)_s^{[1]})^2\} < \infty$. For *AlgDrift.c*, we have $H_1(\sigma)^{[1]} = \tilde{H}_1(\sigma)^{[1]} - \tilde{H}_1(\sigma_0)^{[1]}$, where $\tilde{H}_1(\sigma)^{[1]}$ is zero mean i.i.d., and $\tilde{H}_1(\sigma_0)^{[1]}$ a bounded random variable ($\sigma_0 = \kappa^{i,j}(0)$). Therefore, we have

$$E\{(1 - \frac{1}{\sigma^{1+\zeta'}} H_1(\sigma))^2 | \mathcal{F}_{\sigma_0}\} \leq 1 - c_1 \frac{1}{\sigma^{1+\zeta'}} + c_2 \frac{1}{\sigma^{2(1+\zeta')}}, \quad \text{(B.3)}$$

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

In the analysis of $V_2^g(k)$, we decompose for *AlgDrift.a* and *AlgDrift.b* the sum at the right hand side of (B.2) into $\sum_{i=1}^{n} |\mathcal{N}_i^-|$ partial sums with indices σ belonging to $\{\kappa^{ij}(\upsilon)\}$, $\sigma \leq k$. All these sums contain weighted zero mean random variables $F_1(\sigma)\Delta t(\sigma) + H_1(\sigma)^{[2]}$ whose correlation with $\tilde{g}(\sigma)$ can be neglected for k large enough. As $\sum_{\sigma} E\{\frac{1}{\sigma^{2\zeta}} F_1(\sigma)^2 \Delta t(\sigma)^2\} \leq \infty$ for *AlgDrift.a* and *AlgDrift.b*, it follows, after some technicalities, that

$$V_2^g(k+1) \leq C_1 \sum_{\sigma=1}^{k} \frac{1}{\sigma^{1+q''}} W^g(\sigma), \quad \text{(B.4)}$$

where $C_1 > 0$ and $q'' > 0$. For *AlgDrift.c*, the sum at the right hand side of (B.2) contains the terms $H_1(\sigma)^{[2]} = \tilde{H}_1(\sigma)^{[2]} - \tilde{H}_1(\sigma_0)^{[2]}$, $\sigma \in \kappa^{ij}(\upsilon)$. Having in mind that $\{\tilde{H}_1(\sigma)^{[2]}\}$ is zero mean, and $\tilde{H}_1(\sigma_0)^{[2]}$ bounded w.p.1, for *AlgDrift.c* $\sum_{\sigma} \frac{1}{\sigma^{2\zeta}} E\{\Delta t(k)^2\} < \infty$ by Lemma 3 and $\sum_{\sigma} \frac{1}{\sigma^{\zeta}} < \infty$, so that we again obtain (B.4).

Consequently, there exists a constant $C_1 > 0$ such that

$$V^g(k+1) \leq C_2[1 + \max_{1 \leq \sigma \leq k} W^g(\sigma)]. \quad \text{(B.5)}$$

Analysis of $W^g(k)$ starts from rewriting (26) for $\{\sigma \in \kappa^{ij}(\upsilon)\}$ in the following way

$$\tilde{g}(\sigma+1)^{[2]} = \Pi(\sigma, \sigma)^{[2]} \tilde{g}(\sigma)^{[2]} + \frac{1}{\sigma^{\zeta}} H_2(\sigma)^{[1]} \tilde{g}(\sigma)^{[1]}, \quad \text{(B.6)}$$

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 $H_2(\sigma) = [H_2(\sigma)^{[1]} \vdots H_2(\sigma)^{[2]}]$ adapted to the decomposition of $\tilde{g}(\sigma)$.

We first observe that for any n-vector x and σ large enough

$$x^T E\{\Pi(\sigma, \sigma)^{[2]T} R^g \Pi(\sigma, \sigma)^{[2]}\} x \leq [1 - \frac{2}{\sigma^{\zeta'}} q \frac{\lambda_{min}(Q^g)}{\lambda_{max}(R^g)} + O(\frac{1}{\sigma^{2\zeta'}})] x^T R^g x, \quad \text{(B.7)}$$

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*, $q = \frac{1-\nu}{\mu_c}$ for *AlgDrift.b* and $q = \frac{1}{\mu_c}$ for *AlgDrift.c*. As $q > 0$ (Lemma 3), after standard technicalities based on the classical results on stochastic approximation (Chen, 2002; Kushner & Yin, 2003), it follows that $\prod_{\sigma\in\{\kappa^{ij}(v)\}}\|\Pi(\sigma,\sigma)\|\to_{\sigma\to\infty}0$, $i=1,\dots,n$, $j\in\mathcal{N}_i^-$, 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 $\{H_1(\sigma)^{[1]}\}$, it is possible to show, after technicalities similar to those utilized above, that for k large enough

$$W^g(k+1)\le[1-c_1\frac{1}{k^{\zeta'}}]W^g(k)+C_3\frac{1}{k^{\zeta^*}}V^g(k), \tag{B.8}$$

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

- $\zeta^*=2\zeta'$ for *AlgDrift.a*,
- $\zeta^*=2(1+\zeta')$ for *AlgDrift.b*, and
- $\zeta^*=1+\zeta'$ for *AlgDrift.c*.

Having in mind that $\sum_{k=1}^{\infty}k^{-\zeta^*}<\infty$ in all three cases, the methodology of Huang, Day, Nair, and Manton (2010) and Huang and Manton (2010) can be applied, leading to the conclusion that $\sup_k V^g(k)<\infty$. Therefore, $\tilde g(k)^{[1]}$ tends to a random variable χ^* ($E\{\chi^{*2}\}<\infty$) and that $\tilde g(k)^{[2]}$ tends to zero in the mean square sense and w.p.1. Consequently $\hat g_\infty = T\begin{bmatrix}\lim_{k\to\infty}\tilde g(k)^{[1]}\\ 0\end{bmatrix}=\chi^*\mathbf{1}$

Appendix C. Proof of Theorem 2

After introducing the expression for $z(k)$ into (26), we use the approximation $(1+\frac{1}{k})^{\zeta d}\approx 1+\zeta d\frac{1}{k}$ and obtain for k large enough

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

Observe that for $\zeta'<1$ the term proportional to $\frac{1}{k}$ can be neglected for k large enough. Applying the result of Theorem 1 to (C.1), we conclude from Theorem 1 that $\lim_{k\to\infty}z(k)=0$ in the mean square sense and w.p.1, provided, according to (B.8): (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 directly follows. Notice that different conditions result from different definitions of ζ and different properties of the sequence $\{H_2(k)\}$. Inequality for *AlgDrift.c* is more restrictive than the one for *AlgDrift.b* as a consequence of the fact that $\{H_2(k)\}$ contains a term depending on the initial time t_l^0, which is fixed and nonzero for almost all realizations of the sequence $\hat g(k)$.

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 (C.1) depend on the properties of the matrix $\bar B^*$. Hence the result.

Appendix D. Proof of Theorem 3

Let $\hat h(k)=[(\hat f(k)+\chi(k)\bar\xi_d^0A^{-1}\mathbf{1})^T\vdots(\hat c(k)-\chi(k)A(\bar\eta^0+\bar\delta))^T]^T$. From (19), (20) and (28) and Lemma 1, we obtain

$$\begin{aligned}\hat h(k+1)=&\hat h(k)+\frac{1}{k^{\zeta''}}P_d^{-\zeta''}[M_1(k)(\hat h(k)+u_1(k)\\&+u_2(k))+M_2(k)\hat G(k)]-\frac{1}{k^{\zeta}}M_3(k)\hat g(k),\end{aligned} \tag{D.1}$$

where $u_1(k)=o(\frac{1}{k^{\zeta d}})[(A^{-1}\bar\xi^0)^T\vdots(A(\bar\eta^0+\bar\delta))^T]^T$, $u_2(k)=[(\hat g_d(k)A^{-1}\tilde\xi^0(k))^T\vdots(\hat g_d(k)A(\tilde\eta^0(k)+\tilde\delta(k)))^T]^T$, $M_1(k)=\bar M_1+\tilde M_1(k)$, with $\bar M_1=\begin{bmatrix}\bar\Gamma & \bar\Gamma_d\\ -\bar\Gamma & -\bar\Gamma_d\end{bmatrix}$, $\tilde M_1(k)=\begin{bmatrix}\tilde\Gamma(k) & \tilde\Gamma_d(k)\\ -\tilde\Gamma(k) & -\tilde\Gamma_d(k)\end{bmatrix}$,

$M_2(k)=\bar M_2+\tilde M_2(k)$, $\bar M_2=\mathrm{diag}\{\bar t^0\bar\Gamma,\bar t^0\bar\Gamma\}$, $\tilde M_2(k)=\mathrm{diag}\{\tilde t^0(k)\Gamma(k)+\bar t^0\tilde\Gamma,\tilde t^0(k)\Gamma(k)+\bar t^0\tilde\Gamma\}$, $\hat g_d(k)=\mathrm{diag}\,\hat g(k)$, $\hat G(k)=[\hat g(k)^T\vdots\hat g(k)^T]^T$ and $P_d^{-\zeta''}=\mathrm{diag}\{P^{-\zeta''},P^{-\zeta''}\}$; the last term in (D.1) follows from the term $\Delta\hat g(k+1)=\epsilon^a(k)[A\Gamma(k)\Delta t(k)+N_g(k)]\hat g(k)$ in (19) and Lemma 1, so that $M_3(k)=\begin{bmatrix}P^{-\zeta}[A\Gamma(k)\Delta t(k)+N_g(k)]\\ 0\end{bmatrix}$.

From (D.1) we realize that $P_d^{-\zeta''}\bar M_1$ has n eigenvalues at the origin and n eigenvalues in the left half plane. Therefore, there exists a nonsingular transformation S such that

$$S^{-1}P_d^{-\zeta''}\bar M_1 S=\begin{bmatrix}0 & 0\\ 0 & \bar M^*\end{bmatrix}, \tag{D.2}$$

where $\bar M^*$ is Hurwitz (Stanković et al., 2015). Introduce $\tilde h(k)=S^{-1}\hat h(k)$, with $\tilde h(k)=[\tilde h(k)^{[1]T}\vdots\tilde h(k)^{[2]T}]^T$, where $\dim\tilde h(k)^{[1]}=\dim\tilde h(k)^{[2]}=n$. From (D.1) we obtain:

$$\begin{aligned}\tilde h(k+1)^{[1]}=&\tilde h(k)^{[1]}+\frac{1}{k^{\zeta''}}\{\Psi(k)^{[1]}\tilde h(k)\\&+p(k)^{[1]}+q(k)^{[1]}+r(k)^{[1]}\}\end{aligned} \tag{D.3}$$

$$\begin{aligned}\tilde h(k+1)^{[2]}=&\tilde h(k)^{[2]}+\frac{1}{k^{\zeta''}}\{\bar M^*\tilde h(k)^{[2]}+\Psi(k)^{[2]}\tilde h(k)\\&+p(k)^{[2]}+q(k)^{[2]}+r(k)^{[2]}\},\end{aligned} \tag{D.4}$$

where $S^{-1}P_d^{-\zeta''}\tilde M_1(k)S=\begin{bmatrix}\Psi(k)^{[1]}\\ \Psi(k)^{[2]}\end{bmatrix}$,

$S^{-1}P_d^{-\zeta''}[\tilde M_1(k)u_1(k)+M_1(k)u_2(k)+\tilde M_2(k)\hat g(k)]=\begin{bmatrix}p(k)^{[1]}\\ p(k)^{[2]}\end{bmatrix}$,

$S^{-1}P_d^{-\zeta''}[\bar M_1(k)u_1(k)+\bar M_2\hat g(k)]=\begin{bmatrix}q(k)^{[1]}\\ q(k)^{[2]}\end{bmatrix}$, $-S^{-1}M_3(k)\hat g(k)=\begin{bmatrix}r(k)^{[1]}\\ r(k)^{[2]}\end{bmatrix}$.

Introduce two Lyapunov functions $V^h(k)=E\{\|\tilde h(k)^{[1]}\|^2\}$ and $W^h(k)=E\{\tilde h(k)^{[2]T}R^h\,\tilde h(k)^{[2]}\}$, where $R^h>0$ satisfies the Lyapunov equation $R^h\bar M^*+\bar M^{*T}R^h=-Q^h$, for any given $Q^h>0$ (according to (D.2)).

At the first step, we set $q(k)^{[1]}=0$ and $q(k)^{[2]}=0$ and denote the corresponding solutions of (D.3) and (D.4) by $\tilde h_1(k)^{[1]}$ and $\tilde h_1(k)^{[2]}$, respectively. Then, we introduce $V_1^h(k)=E\{\|\tilde h_1(k)^{[1]}\|^2\}$ and $W_1^h(k)=E\{\tilde h_1(k)^{[2]T}R^h\tilde h_1(k)^{[2]}\}$. It is straightforward to see that the results from Huang and Manton (2010) can be directly applied to (D.3) and (D.4) (Theorem 11 and Lemma 12 therein), leading to the conclusion that $\sup_k V_1^h(k)<\infty$ and that $W_1^h(k)$ tends to zero when $k\to\infty$. It is essential for this conclusion that the sequences $\{\Psi(k)^{[1]}\}$, $\{\Psi(k)^{[2]}\}$, $\{p(k)^{[1]}\}$ and $\{p(k)^{[2]}\}$ are uncorrelated, that $\sum_{k=1}^{\infty}\frac{1}{k^{2\zeta''}}<\infty$, and that $\lambda_{min}(Q^h)>0$.

At the second step, consider the zero state responses $\tilde h_2(k)^{[1]}$ and $\tilde h_2(k)^{[2]}$ of (D.3) and (D.4) to the inputs $q(k)^{[1]}$ and $q(k)^{[2]}$, respectively. Let $V_2^h(k)=E\{\|\tilde h_2(k)^{[1]}\|^2\}$ and $W_2^h(k)=E\{\tilde h_2(k)^{[2]T}R^h\tilde h_2(k)^{[2]}\}$. By (27), we first conclude that $\bar M_2\hat g(k)=\bar M_2\hat g(k)^{[2]}$ (having in mind Theorem 2 and (27)). From (D.3), we obtain

$$\tilde h_2(k+1)^{[1]}=[I+\frac{1}{k^{\zeta''}}\Psi(k)_1^{[1]}]\tilde h_2(k)^{[1]}+\frac{1}{k^{\zeta''}}q(k)^{[1]}, \tag{D.5}$$

where $\Psi(k)_1^{[1]}$ is an $(n\times n)$ submatrix of $\Psi(k)^{[1]}$. From (D.5) we have that $E\{\tilde h_2(k+1)^{[1]}\}=E\{\tilde h_2(k)^{[1]}\}+\frac{1}{k^{\zeta''}}q(k)^{[1]}$; consequently,

$$\begin{aligned}V_2^h(k+1)\le&(1+c'\frac{1}{k^{2\zeta''}})V_2^h(k)+(\frac{1}{k^{\zeta''}}q(k)^{[1]})^2\\&+E\{\tilde h_2(k)^{[1]}\}\frac{1}{k^{\zeta''}}q(k)^{[1]},\end{aligned} \tag{D.6}$$

($c' < \infty$). Since $q(k)^{[1]} = o(\frac{1}{k^{\zeta d}})$ w.p.1, by Theorem 2 we can derive that $\sup_k E\{\tilde{h}_2(k)^{[1]}\}^2 < \infty$. Consequently, $\sup_k V_2^h(k) < \infty$ for all $\zeta'' > 1 - \zeta d$, because of the requirement that $\sum_k \frac{1}{k^{\zeta''+\zeta d}} < \infty$. When $\zeta' < 1$, this result holds, according to Theorem 2, for all $\zeta'' \in (\frac{1}{2}, 1]$ in the case of *AlgDrift.b* and *AlgDrift.c*, and for $\zeta'' > \frac{3}{2} - \zeta'$ in the case of *AlgDrift.a*. Analysis of $\tilde{h}_2(k)^{[2]}$ relies on the classical results from stochastic approximation (Chen, 2002), wherefrom we obtain that $\lim_{k\to\infty} W_2^h(k) = 0$.

At the third step, consider the zero state responses $\tilde{h}_3(k)^{[1]}$ and $\tilde{h}_3(k)^{[2]}$ of (D.3) and (D.4) to the inputs $r(k)^{[1]}$ and $r(k)^{[2]}$, respectively. Let $V_3^h(k) = E\{\|\tilde{h}_3(k)^{[1]}\|^2\}$ and $W_3^h(k) = E\{\tilde{h}_3(k)^{[2]T} R^h \tilde{h}_3(k)^{[2]}\}$. Let $r_1(k)^{[1]}$ be the part of $r(k)^{[1]}$ following from $\varepsilon^a(k) A\Gamma(k)\Delta t(k)\hat{g}(k)$, and $r_2(k)^{[1]}$ the part following from $\varepsilon^a(k) N_g(k)\hat{g}(k)$. Taking into account (27), one concludes that $r_1(k)^{[1]} \sim o(\frac{1}{k^{\zeta'+\zeta d}})$ and that $\{\}r_2(k)^{[1]}\}$ is a zero mean i.i.d. sequence on any subsequence κ^{ij} multiplied by $\frac{1}{k^{\zeta}}$. Therefore, $V_3^h(k) < \infty$, provided $\zeta' + \zeta d > 1$; this is fulfilled in the case of *AlgDrift.a* for $\zeta' > \frac{3}{4}$, and for any $\zeta' \in (\frac{1}{2}, 1]$ in the case of *AlgDrift.b* and *AlgDrift.c*. Reasoning similarly, we conclude that $\lim_{k\to\infty} W_3^h(k) = 0$.

Therefore, $\sup_k V^h(k) < \infty$ and $\lim_{k\to\infty} W^h(k) = 0$. Using the arguments exposed in Huang and Manton (2010), we further obtain that $\tilde{h}(k)^{[1]}$ tends to a random n-vector $\tilde{h}^{[1]*}$, and that $\tilde{h}(k)^{[2]}$ tends to zero in the mean square sense and w.p.1, implying that $\hat{h}^* = \lim_{k\to\infty}\hat{h}(k) = S\tilde{h}^*$, where $\tilde{h}^* = [\tilde{h}^{[1]*T} \vdots 0^T_{1\times n}]^T$. The result of the theorem follows after taking into account $\chi(k) \to \chi^*$ w.p.1 and that

$$\bar{M}_1\hat{h}^* = \bar{M}_1 S\tilde{h}^* = P_d^{\zeta''} S \begin{bmatrix} 0 & 0 \\ 0 & M^* \end{bmatrix} \tilde{h}^* = 0,$$

according to (D.2) and the definition of $\hat{h}^*$.

Appendix E. Proof of Theorem 4

We shall pay attention only to the possible convergence points: the rest can be derived by following methodologically the proof of Theorem 3. Namely, according to Kushner and Yin (1987), we formulate the ODE characterizing the asymptotic behavior of the algorithm, and obtain that

$$\bar{\Gamma}\hat{f}^* - \chi^*[\bar{\Gamma}_d(\bar{\delta} + \bar{\eta}^0) - \bar{\Gamma}\bar{\xi}^0] + \bar{\Gamma}_d \mathbf{1}\hat{c}^{con} = 0$$

$$\sum_{i=1}^{n} \bar{\phi}_i\{\bar{\Gamma}^{(i)}\hat{f}^* - \chi^*[(\bar{\Gamma}_d[\bar{\delta} + \bar{\eta}^0])_i - (\bar{\Gamma}\bar{\xi}^0)_i] + (\bar{\Gamma}_d\mathbf{1})_i\hat{c}^{con}\} = 0, \tag{E.1}$$

wherefrom the result directly follows.

References

Akyildiz, I. F., & Vuran, M. C. (2010). *Wireless sensor networks*. John Wiley & Sons.

Aysal, T. C., Yildriz, M. E., Sarwate, A. D., & Scaglione, A. (2009). Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing*, *57*, 2748–2761.

Bolognani, S., Carli, R., Lovisari, E., & Zampieri, S. (2012). A randomized linear algorithm for clock synchronization in multi agent systems. In *Proc. IEEE conf. decision and control* (pp. 20–25).

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

Carli, R., Chiuso, A., Zampieri, S., & Schenato, L. (2008). A PI consensus controller for networked clock synchronization. In *Proc. IFAC world congress*.

Chaudhari, Q. M., Serpedin, E., & Qaraqe, K. (2008). On maximum likelihood estimation of clock offset and skew in networks with exponential delays. *IEEE Transactions on Signal Processing*, *56*, 1685–1697.

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

Choi, B. J., Liang, H., Shen, X., & Zhuang, W. (2012). DCS : distributed asynchronous clock synchronization in delay tolerant networks. *IEEE Transacti ons on Parallel and Distributed Systems*, *23*, 491–504.

Elson, J., Girod, L., & Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. In *Proc. symp. oper. syst. design and implement.* (pp. 147–163).

Fan, R., & Lynch, N. (2006). Gradient clock synchronization. *Distributed Computing*, *18*, 255–266.

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

Garone, E., Gasparri, A., & Lamonaca, F. (2015). Clock synchronization protocol for wireless sensor networks with bounded communication delays. *Automatica*, *59*, 60–72.

Gupta, P., & Kumar, P. R. (2006). The capacity of wireless networks. *IEEE Transactions on Information Theory*, *46*(2), 388–404.

He, J., Cheng, P., Chen, J., Shi, L., & Lu, R. (2014). Time synchronization for random mobile sensor networks. *IEEE Transactions on Vehicular Technology*, *63*, 3935–3946.

He, J., Cheng, P., Shi, L., Chen, J., & Sun, Y. (2014). Time synchronization in WSNs: a maximum value based consensus approach. *IEEE Transactions on Automatic Control*, *59*, 660–675.

Holler, J., Tsiatsis, V., Mulligan, C., Avesand, S., Karnouskos, S., & Boyle, D. (2014). *From machine-to-machine to the Internet of Things: Introduction to a new age of intelligence*. Academic Press.

Huang, M., Day, S., Nair, G. N., & Manton, J. H. (2010). Stochastic consensus over noisy networks with Markovian and arbitrary switches. *Automatica*, *46*, 1571–1583.

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

Kim, K.-D., & Kumar, P. R. (2012). Cyberphysical systems: A perspective at the centennial. *Proceedings of the IEEE*, *100*(Special Centennial Issue), 1287–1308.

Kushner, H. J., & Yin, G. (1987). Asymptotic properties of distributed and communicating stochastic approximation algorithms. *SIAM Journal on Control and Optimization*, *25*, 1266–1290.

Kushner, H. J., & Yin, G. (2003). *Stochastic approximation and recursive algorithms and applications*. Springer.

Leng, M., & Wu, Y.-C. (2011). Low complexity maximum likelihood estimator for clock synchronization of wireless sensor nodes under exponential delays. *IEEE Transactions on Signal Processing*, *59*, 4860–4870.

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

Liao, C., & Barooah, P. (2013a). Distributed clock skew and offset estimation from relative measurements in mobile networks with Markovian switching topologies. *Automatica*, *49*, 3015–3022.

Liao, C., & Barooah, P. (2013b). DiSync: accurate distributed clock synchronization in mobile ad–hoc networks from noisy difference measurements. In *Proc. American control conference* (pp. 3332–3337).

Maroti, M., Kusy, B., Simon, G., & Ledeczi, A. (2004). The flooding time synchronization protocol. In *Proc. 2nd ACM intern. conf. embedded netw. contr. syst.* (pp. 39–49).

Nedić, A. (2011). Asynchronous broadcast-based convex optimization over a network. *IEEE Transactions on Automatic Control*, *56*, 1337–1351.

Olfati-Saber, R., Fax, A., & Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, *95*, 215–233.

Schenato, L., & 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., & Strogatz, S. H. (2008). Distributed synchronization in wireless networks. *IEEE Signal Processing Magazine*, *25*, 81–97.

Sivrikaya, F., & Yener, B. (2004). Time synchronization in sensor networks: a survey. *IEEE Network*, *18*, 45–50.

Solis, R., Borkar, V., & Kumar, P. R. (2006). A new distributed time synchronization protocol for multi hop wireless networks. In *Proc. IEEE conf. decision and control* (pp. 2734–2739).

Sommer, P., & Wattenhofer, R. (2009). Gradient clock synchronization in wireless sensor networks. In *Proc. int. conf. inf. proc. in sensor networks* (pp. 37–48).

Stanković, M. S., Stanković, S. S., & Johansson, K. H. (2012). Distributed time synchronization in lossy wireless sensor networks. In *3rd IFAC workshop on distr. estim. contr. netw. syst., vol. 3* (pp. 25–30).

Stanković, M. S., Stanković, S. S., & Johansson, K. H. (2015). Distributed blind calibration in lossy sensor networks via output synchronization. *IEEE Transactions on Automatic Control*, *60*, 3257–3262.

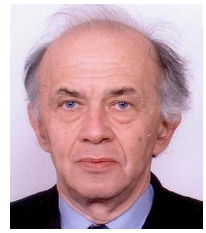
Stanković, M. S., Stanković, S. S., & Johansson, K. H. (2016). Distributed drift estimation for time synchronization in lossy networks. In *24th Mediterranean conference on control and automation* (pp. 779–784).

Su, W., & Akyildiz, I. F. (2005). Time diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Transactions on Networking*, *13*, 384–397.

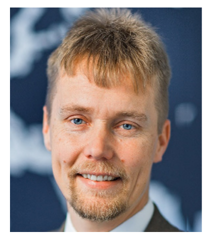
Sundararaman, B., Buy, U., & Kshemkalyani, A. (2005). Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, *3*, 281–323.

Tian, Y.-P. (2015). LSTS: a new time synchronization protocol for networks with random communication delays. In *Proc. IEEE conf. decision and control* (pp. 7404–7409).
Tian, Y.-P. (2017). Time synchronization in WSNs with random bounded communication delays. *IEEE Transactions on Automatic Control*, *62*, 5445–5450.
Tian, Y.-P., Zong, S., & Cao, Q. (2016). Structural modeling and convergence analysis of consensus-based time synchronization algorithms over networks: non-topological conditions. *Automatica*, *65*, 64–75.
Wu, Y.-C., Chaudhari, Q., & Serpedin, E. (2011). Clock synchronization in wireless sensor networks. *IEEE Signal Processing Magazine*, 124–138.
Xiong, G., & Kishore, S. (2009). Analysis of distributed consensus time synchronization with Gaussian delay over wireless sensor networks. *EURASIP Journal on Wireless Communication and Network*, *2009*(1), 528161.
Yildirim, K. S., Carli, R., & Schenato, L. (2015). Adaptive control-based clock synchronization in wireless sensor networks. In *Proc. European control conference*.

Miloš S. Stanković received his Dipl. Ing. (M.Sc.) degree from the School of Electrical Engineering at the University of Belgrade, Serbia, in 2002. He received his Ph.D. degree in Systems and Entrepreneurial Engineering from the University of Illinois at Urbana–Champaign (UIUC), USA in 2009. He was a Research and Teaching Assistant in the Control and Decision Group of the Coordinated Science Laboratory at the UIUC (2006–2009). From 2009 to 2012 he was a Postdoctoral Researcher in the Automatic Control Laboratory and the ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden. In 2012 he joined the Innovation Center, School of Electrical Engineering, University of Belgrade, Serbia, as an EU Marie Curie Research Fellow. In 2017 he joined the School of Technical Sciences, Singidunum University, Belgrade, Serbia as an Associate Professor. His research interests include decentralized decision making, networked control systems, dynamic game theory, and machine learning, with applications to mobile sensor networks, cyber–physical systems, the Internet of Things and big data analytics.



Srdjan S. Stanković got his Dipl.Ing., Mgr.Sc. and Ph.D. degrees from the Faculty of Electrical Engineering, University of Belgrade, Yugoslavia, in 1968, 1972 and 1975, respectively. He was with the Institute for Nuclear Sciences, Vinča, Belgrade, Yugoslavia, from 1968 to 1972. Since 1973 he has been with the Faculty of Electrical Engineering, University of Belgrade, where he is currently Emeritus Professor of Automatic Control. He is also President of the Research Council of the Vlatacom Institute, Belgrade. He held visiting positions at the Eindhoven University of Technology, Eindhoven, the Netherlands and at the Santa Clara University, Santa Clara, California. He was President of the National Council for Higher Education of the Republic of Serbia, as well as President of the Serbian Society for Electronics, Communications, Computers, Control and Nuclear Engineering. He has published numerous scientific papers in diverse fields, including estimation and identification, adaptive systems, large scale systems, decentralized control and neural networks. He has also been leader of numerous scientific projects for government and industry.



Karl Henrik Johansson is Director of the Stockholm Strategic Research Area ICT The Next Generation and Professor at the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology. He received M.Sc. and Ph.D. degrees from Lund University. He has held visiting positions at UC Berkeley, Caltech, NTU, HKUST Institute of Advanced Studies, and NTNU. His research interests are in networked control systems, cyber–physical systems, and applications in transportation, energy, and automation. He is a member of the IEEE Control Systems Society Board of Governors, the IFAC Executive Board, and the European Control Association Council. He has received several best paper awards and other distinctions. He is a Distinguished Professor with the Swedish Research Council and a Wallenberg Scholar and he has received the Future Research Leader Award from the Swedish Foundation for Strategic Research and the triennial Young Author Prize from IFAC. He is Fellow of the IEEE and the Royal Swedish Academy of Engineering Sciences, and he is IEEE Distinguished Lecturer.