Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Strongly Non-Zeno Event-Triggered Wireless Clock Synchronization

James A. Berneburg * **Eloy Garcia** ** **Adam Gerlach** **
David Casbeer ** **Cameron Nowzari** *

* *Department of Electrical and Computer Engineering George Mason University, 4400 University Drive, Fairfax, VA 22030, email:* {jbernebu,cnowzari}@gmu.edu
** *Control Science Center of Excellence, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, email:* {eloy.garcia.2,adam.gerlach.1,david.casbeer}@us.af.mil

Abstract: Agreeing on a common time is essential to many coordinated tasks in wireless networks, but this is difficult to accomplish when each agent only has access to a local hardware clock. Therefore, clock synchronization is essential in order to carry out many of these coordinated tasks. While there are numerous existing algorithms for clock synchronization, many result in disruptive discontinuous virtual clocks, and most rely on regular communication, which does not scale with large systems. Instead, this paper presents a novel clock synchronization algorithm, which allows for a continuous virtual clock, with a dynamic event-triggered communication strategy which is strongly non-Zeno because it guarantees a designable positive time between communication instances.

Keywords: Event-based control, multi-agent systems, Lyapunov methods

1. INTRODUCTION

Wireless networks of individually controlled agents are being used increasingly in a broad number of applications, ranging from the coordination of unmanned air vehicles to distributed reconfigurable sensor networks, and more; see (Olfati-Saber and Murray, 2004) and (Ren et al., 2007) and references therein. Although the design of these networks may rely on the assumption of coordinated timing for actions, individual agents in these wireless networks are each equipped with their own hardware clock, based on an oscillator. This means that different agents will not agree on the notion of time because of clock drift, which is running faster or slower than absolute time, and/or an offset, which is a differing notion of time zero.

A common method to circumvent these timing discrepancies is to introduce a virtual software clock for each agent, and to then adjust that clock's drift and offset until all agents' software clocks agree on the time. This is what we refer to in this paper as *clock synchronization*. Although global synchronization among all clocks is desired, doing so would require distributed control that can scale with large systems, since for large systems communication becomes a limiting resource. Distributed means that communication must be restricted to neighboring agents, rather than the entire system. To handle this problem, many researchers employ a distributed consensus approach, where the drift and offset of the software clocks are the values to reach consensus. Consensus problems are when agents with individual dynamics are to be driven such that all the agents' states are equal, which also has applications in distributed computing, networks of sensors, flocking, and rendezvous (Olfati-Saber and Murray, 2004; Liu et al., 2011; Ren et al., 2007). A seminal work for clock synchronization is (Schenato and Fiorentin, 2011), which ensures the synchronization of the drift and offset of virtual clocks using asynchronous periodic broadcasts, with clocks updated discontinuously.

Many works use similar strategies to achieve convergence, often in the presence of communication delays and/or noise. Dengchang et al. (2013); Carli and Zampieri (2014); He et al. (2014) all offer different algorithms to solve this problem of the drift and offset synchronization of virtual clocks using a consensus-like approach. Other works are able to account for bounded noise or communication delays, such as (Zhang et al., 2019; Garone et al., 2015; He et al., 2017; Stanković et al., 2018).

Unfortunately, these algorithms are not scalable to large systems because they rely on frequent communication, such as asynchronous periodic communication or communication at random times in regular intervals, between neighboring agents. To prevent frequent communication, some consider event-triggered control, where communication occurs at an "event," when some condition on the state is satisfied. Proskurnikov and Mazo (2018) and Postoyan et al. (2015) develop general methods for achieving stability with such control strategies, and see (Nowzari et al., 2019) for a detailed survey of works applying event-triggered control to the consensus problem specifically. Two earlier works to apply event-triggered communication algorithms to the clock synchronization problem are (Kadowaki and Ishii, 2015; Li et al., 2015). They use functions of time in the trigger mechanism to prevent an infinite amount of communication in finite time (Zeno behavior). More recent approaches by Garcia et al. (2017); Xu et al. (2019) guarantee convergence for clock drift and model the virtual clock parameters with continuous dynamics, so that they remain continuous and do not disrupt concurrent processes that depend on the clock.

However, the main problem common to these event-triggered solutions is that none can account for finite hardware operating frequencies, and so they are not truly implementable event-triggered solutions. More specifically, simply ruling out Zeno behavior may still demand that communication occurs arbitrarily fast to guarantee con-

vergence. As noted by Heemels et al. (2012), Nowzari et al. (2019), and Borgers and Heemels (2014), the way to handle this is to make sure that the event-triggered communication algorithm ensures a positive minimum inter-event time (MIET) or, in other words, that it is strongly non-Zeno. It is important to note here that this property of an event-triggered protocol is different from simply *forcing* a MIET through the use of a chosen 'dwell time' or inactive time between communications, in which case the convergence guarantees of the algorithm may be lost. For example, De Persis and Frasca (2013) enforce a positive MIET for a consensus problem, but can only conclude convergence to a *neighborhood* of consensus as a result.

Therefore, similar to Garcia et al. (2017) and Xu et al. (2019), we wish to provide an algorithm where the clocks are synchronized in a continuous manner and we wish to ensure that it is fully distributed with an event-triggered communication strategy so that it is scalable. However, we also wish to make certain that a positive MIET exists for the communication strategy so that it can be implemented. We turn to existing event-triggered consensus algorithms for this. Both (Dolk et al., 2017) and (De Persis and Postoyan, 2017) are able to ensure full consensus with a positive MIET by using a dynamic virtual state in the event-triggering function, but neither is fully distributed. More recently, the work in (Berneburg and Nowzari, 2019b) has provided a guaranteed positive MIET for a fully distributed algorithm, in a similar fashion.

Statement of Contributions: We propose a dynamic event-triggered virtual clock synchronization algorithm, for only clock drift, that has significant advantages over the state-of-the-art algorithms for physical implementations. We use the same control law provided in both (Garcia et al., 2017; Xu et al., 2019) but we define a novel dynamic event-triggered communication strategy that has three main advantages. First, it is a naturally asynchronous algorithm that does not require any synchronous actions among agents, unlike the algorithm proposed in (Xu et al., 2019). Second, it does not use a time-dependent event-triggering threshold and so avoids the implementation issues of the algorithm in (Garcia et al., 2017). Finally, unlike both previous works, this algorithm guarantees a positive minimum inter-event time (MIET) for each agent, which can be chosen using a design parameter, meaning it can actually be implemented on physical platforms (Borgers and Heemels, 2014; Nowzari et al., 2019).

2. PRELIMINARIES

The Euclidean norm of a vector $v \in \mathbb{R}^n$ is denoted by $||v||$. An n-dimensional column vector with every entry equal to 1 (or 0) is denoted by $\mathbf{1}_n$ (or $\mathbf{0}_n$). Given a vector $v \in \mathbb{R}^N$, we denote by $\text{diag}(v)$ the $N \times N$ diagonal matrix with the entries of v along its diagonal. The minimum eigenvalue of a square matrix A is given by $\text{eigmin}(A)$ and its maximum eigenvalue is given by $\text{eigmax}(A)$. Young's inequality is

$$xy \leq \frac{b}{2}x^2 + \frac{1}{2b}y^2, \tag{1}$$

for $b > 0$ and $x, y \in \mathbb{R}$ (Hardy et al., 1952).

Graph Theory A graph $\mathcal{G} = (V, \mathcal{E}, A)$ has a set of vertices $V = \{1, 2, ..., N\}$, a set of edges $\mathcal{E} \subset V \times V$, and an adjacency matrix $A \in \mathbb{R}^{N \times N}$ with each entry $a_{ij} \in \mathbb{R}_{\geq 0}$, where $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. For an undirected graph, $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$. An undirected graph is connected if there exists a path (a finite sequence of edges $(i, k), \ldots, (n, j)$) between any two vertices i and j. Let $\mathcal{N}_i \subset V$ be the set of neighbors of vertex i, that is, $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}$. The Laplacian is $L = \text{diag}\left([|\mathcal{N}_1|, |\mathcal{N}_2|, \ldots, |\mathcal{N}_N|]\right) - A$.

3. PROBLEM STATEMENT

Consider a wireless network of N agents interacting via an undirected, connected graph $\mathcal{G}$ with edges $\mathcal{E}$. With $t \in \mathbb{R}_{\geq 0}$ as the absolute time, we assume that each agent $i \in \{1, \ldots, N\}$ instead only has access to a local time

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

where $a_i > 0$ and $b_i \in \mathbb{R}$ are unknown parameters representing the drift and bias of agent i's local clock, respectively. For a perfect clock we should have $a_i = 1$ and $b_i = 0$ such that the local time is always equal to the absolute time $\tau_i(t) = t$. Since the absolute time t is not available to any clock, it is impossible to estimate a_i and b_i. Therefore, the goal is *not* for each agent to properly track the absolute time t. Instead, we define a virtual clock for each agent $i \in \{1, \ldots, N\}$

$$T_i(\tau_i) = \alpha_i(\tau_i)\tau_i + \beta_i(\tau_i), \tag{3}$$

where $\alpha_i(\tau_i)$ and $\beta_i(\tau_i)$ are what we call the *synchronized* drift and bias, respectively. Although these virtual clocks must be functions of each agent's local clock τ_i, the synchronization must be done in absolute time. Thus, combining (3) and (2), we can rewrite the output of each agent's virtual clock as a function of absolute time t as

$$T_i(t) = \alpha_i(t)(a_i t + b_i) + \beta_i(t). \tag{4}$$

The goal is to synchronize all these virtual clocks so that

$$\lim_{t \to \infty} |T_i(t) - T_j(t)| = 0, \tag{5}$$

for all $i, j \in \{1, \ldots, N\}$. Additionally, the clocks are to be used to inform various other processes on the agents, so they must not synchronize to a constant virtual time, so we require that the clocks synchronize to a trajectory where $T_i(t)$ is monotonically increasing for all agents i.

In this paper we only consider local clocks without bias, that is, $b_i = 0$ for all $i \in \{1, \ldots, N\}$. This exact problem is considered in (Garcia et al., 2017) and (Xu et al., 2019), where a static time-dependent and a state-dependent event-triggered communication and control strategy are proposed, respectively. Therefore, defining $x_i(t) = \alpha_i(t)a_i$, we can rewrite the output of each agent's virtual clock as

$$T_i(t) = x_i(t)t. \tag{6}$$

Therefore, in order to achieve clock synchronization (5), we must design an algorithm such that

$$\lim_{t \to \infty} |x_i(t) - x_j(t)| = 0 \tag{7}$$

for all $i, j \in \{1, \ldots, N\}$. The value of $x_i(t)$ will never be known by any agent and is used for analysis purposes only.

We consider a communication model in which agent i can directly communicate with any of its neighbors $j \in \mathcal{N}_i$ in the graph $\mathcal{G}$ with negligible delay. Letting $\{t_\ell^i\}_{\ell \in \mathbb{Z}_{\geq 0}} \subset \mathbb{R}_{\geq 0}$ be a sequence of times at which agent i communicates information to its neighbors $j \in \mathcal{N}_i$, each agent j maintains the last received information at any given time t. More specifically, we denote by $\widehat{\alpha}_i(t)$ the last broadcast values of agent i at any given time t. Formally,

$$\widehat{\alpha}_i(t) = \alpha_i(t_\ell^i) \qquad t \in [t_\ell^i, t_{\ell+1}^i), \tag{8}$$

is the information available to all neighbors $j \in \mathcal{N}_i$ at any given time $t \in \mathbb{R}_{\geq 0}$. Similarly, agent i has access to piece-wise constant values of $\widehat{\alpha}_j(t)$ depending on when its neighbors $j \in \mathcal{N}_i$ communicate with it.

To help ensure the existence of a positive MIET, we define an auxiliary variable χ_i (with dynamics $\dot{\chi}_i = a_i \frac{d\chi_i}{d\tau_i}$), for each agent i, which will be used to decide when that agent will communicate, as formalized later. Therefore, the extended state vector for a single agent i is then given by

$$q_i(t) = \begin{bmatrix} \alpha_i(t) \\ \widehat{\alpha}_i(t) \\ \chi_i(t) \end{bmatrix}.$$

We define the information available to agent i as

$$v_i(t) \triangleq (q_i(t), \{\widehat{\alpha}_j\}_{j\in\mathcal{N}_i}),$$

and we need to design the dynamics for each agent's state variable $\frac{d\alpha_i}{d\tau_i}(v_i)$ along with a communication strategy that determines the sequence of broadcasting times $\{t_\ell^i\}_{\ell\in\mathbb{Z}_{\geq 0}}$ for agent i. Note each agent i doesn't have access to $\dot{\alpha}_i$, which is α_i's derivative with respect to absolute time t, and instead its derivative must be written with respect to the local hardware clock. The problem is formalized below.

Problem 3.1. (Distributed Clock Synchronization). Given the communication graph $\mathcal{G}$ and unknown $a_i > 0$ for all agents $i \in \{1, \dots, N\}$, design a distributed communication and control strategy for $\alpha_i(\tau_i)$ such that $\lim_{t\to\infty} |T_i(t) - T_j(t)| = 0$ and $\lim_{t\to\infty} \dot{T}_i > 0$ for all $i, j \in \{1, \dots, N\}$.

4. DISTRIBUTED DYNAMIC EVENT-TRIGGERED CLOCK SYNCHRONIZATION

Here we present a solution to the problem described in Section 3. We will first design a dynamic event-triggered first-order consensus-based synchronization algorithm by combining ideas from (Xu et al., 2019) which presents a static event-triggered consensus-based synchronization algorithm with (Berneburg and Nowzari, 2019b,c) which presents a dynamic event-triggered consensus algorithm. Details on the different classes of event-triggered consensus algorithms are available in (Nowzari et al., 2019).

Control Strategy As proposed in (Garcia et al., 2017; Xu et al., 2019), the control strategy for α_i is given by

$$\frac{d\alpha_i}{d\tau_i} = -\sum_{j\in\mathcal{N}_i} \left(\widehat{\alpha}_i(\tau_i) - \frac{a_j}{a_i}\widehat{\alpha}_j(\tau_i) \right). \tag{9}$$

It should be noted that (9) is presented as a dynamic variable that changes with respect to agent i's local clock τ_i. Although the independent values of a_i and a_j in (9) are unknown to the agents, the relative value of $\frac{a_j}{a_i}$ can be easily computed by the following method using two independent messages separated by time as proposed in (Schenato and Fiorentin, 2011; Garcia et al., 2017). Assuming negligible delays, let t_1 and $t_2 > t_1$ be two absolute times at which agent j communicates with agent i. At each of these times, agent j sends to agent i its own local time $\tau_j(t_1)$ and $\tau_j(t_2)$. Using the times of its own local clock, agent i can compute $\frac{a_j}{a_i} = \frac{\tau_j(t_2)-\tau_j(t_1)}{\tau_i(t_2)-\tau_i(t_1)}$. Note that this is a static quantity, so in the presence of various uncertainties, the agents can take repeated measurements to refine their estimate of it. We assume this quantity can be measured exactly.

Remark 4.1. (Relation to consensus). Because $\dot{\alpha}_i = a_i \frac{d\alpha_i}{d\tau_i}$, with $\widehat{x}_i = \widehat{\alpha}_i a_i$ one can write (9) as

$$\dot{x}_i = -a_i \sum_{j\in\mathcal{N}_i} (\widehat{x}_i - \widehat{x}_j), \tag{10}$$

which is similar to the well-established input for distributed multi-agent consensus (Olfati-Saber and Murray, 2004; Ren et al., 2007; Nowzari et al., 2019). As a result, this would can be seen as an application of such consensus algorithms. However, adapting such algorithms to work here is nontrivial, because of the gain a_i applied to each input and because absolute time is unknown, changing the convergence value and the Lyapunov function. •

It is known from (Garcia et al., 2017; Xu et al., 2019) that with sufficient sampling, the controller (9) can guarantee the clocks asymptotically achieve synchronization. However, the above solutions do not exhibit robust event separation properties as defined in (Borgers and Heemels, 2014). To address this issue, we wish to design a dynamic event-triggered communication strategy that can guarantee both that the controller (9) achieves synchronization and that a positive MIET exists for each agent.

In general, a Lyapunov-based static event-triggered communication strategy communicates when necessary to guarantee the monotonically decreasing property of the Lyapunov function, but this can become impractical when dealing with distributed systems. Dynamic event-triggered communication relaxes these requirements by introducing a virtual dynamic variable $\chi_i \geq 0$ for each agent $i \in \{1, \dots, N\}$ that ensures events are triggered among agents asynchronously and exhibit robust event separation properties (Berneburg and Nowzari, 2019c; Borgers and Heemels, 2014; Nowzari et al., 2019). Intuitively, the virtual state χ_i stores an agent's contribution to system convergence, so that events are triggered to ensure that it is constructively contributing on average.

Communication Strategy Here we derive a dynamic event-triggered strategy to determine when an agent i should communicate with its neighbors $j \in \mathcal{N}_i$. We define the broadcast times to occur when $\chi_i = 0$ but $\alpha_i - \widehat{\alpha}_i \neq 0$ and update $\widehat{\alpha}_i = \alpha_i$. More formally, the sequence of event times $\{t_\ell^i\}_{\ell\in\mathbb{Z}_{\geq 0}}$ is given by

$$t_{\ell+1}^i = \inf\{t \geq t_\ell^i | \chi_i(t) = 0 \text{ and } \alpha_i - \widehat{\alpha}_i \neq 0\}, \tag{11}$$

for all $\ell \in \mathbb{Z}_{\geq 0}$. We now define a Lyapunov function

$$V = \sum_{i=1}^N \frac{1}{a_i}(x_i - x_i^*)^2 + \sum_{i=1}^N a_i\chi_i, \tag{12}$$

where $x^* = [x_1^*, \dots, x_N^*] \triangleq [\alpha_1^* a_1, \dots, \alpha_N^* a_N]$ denotes the final convergence value of each x_i; see Lemma 4.2. Note that this Lyapunov function consists of two components; the first summation is a "physical" component because it regards the states that must reach consensus, and the second is a "cyber" component because it regards the auxiliary variables for the effects of communication.

Lemma 4.2. (Final Values). If $\widehat{\alpha} = \alpha$ (or $\widehat{\alpha}$ is updated sufficiently fast) with input (9), then the final value of each α_i is given by

$$\alpha_i^* = \frac{\sum_{j=1}^N \alpha_j(0)}{a_i \sum_{j=1}^N \frac{1}{a_j}}. \tag{13}$$

Proof. This proof is abridged for reasons of space. With the control input (10), the weighted average $\sum_{i=1}^N \frac{x_i}{Na_i}$ is constant. Therefore, one can use $\sum_{i=1}^N \frac{x_i^*}{Na_i} = \sum_{i=1}^N \frac{x_i(0)}{Na_i}$, and solve for x_i^*, because $x_i^* = x_j^*$ for i, j in $1, 2, ...N$. □

Note that each α_i^* is strictly positive, if each $\alpha_i(0) > 0$ and $a_i > 0$. The goal now is to design the dynamics of the clock χ_i for each agent such that $\dot{V} \leq 0$ and that $\chi_i \geq 0$ to ensure $V \geq 0$. Taking the time derivative, we have

$$\dot{V} = \left[\sum_{i=1}^{N} 2(x_i - x_i^*)a_i \frac{d\alpha_i}{d\tau_i} + a_i^2 \frac{d\chi_i}{d\tau_i}\right] = \sum_{i=1}^{N}\left[-2(x_i - x_i^*)\sum_{j\in\mathcal{N}_i}(\widehat{x}_i - \widehat{x}_j) + a_i^2\frac{d\chi_i}{d\tau_i}\right],$$

recalling that $\widehat{x}_i = \widehat{\alpha}_i a_i$. Note that $-2\sum_{i=1}^{N} x_i^* \sum_{j\in\mathcal{N}_i}(\widehat{x}_i - \widehat{x}_j) = x^* L\widehat{x} = 0$, so we can write

$$\dot{V} = \sum_{i=1}^{N}\left[-(2\widehat{x}_i + 2a_i e_i)\sum_{j\in\mathcal{N}_i}(\widehat{x}_i - \widehat{x}_j) + a_i^2\frac{d\chi_i}{d\tau_i}\right],$$

where $e = [e_1, \dots, e_N]^T \triangleq [\alpha_1 - \widehat{\alpha}_1, \dots, \alpha_N - \widehat{\alpha}_N]^T$ denotes the vector of errors between each agent's state α_i and the last communicated state $\widehat{\alpha}_i$. Because

$$-2\sum_{i=1}^{N}\widehat{x}_i\sum_{j\in\mathcal{N}_i}(\widehat{x}_i - \widehat{x}_j) = -2\widehat{x}^T L\widehat{x} = -\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}(\widehat{x}_i - \widehat{x}_j)^2,$$

we can write $\dot{V} = \sum_{i=1}^{N}\dot{V}_i$, where

$$\dot{V}_i = -\sum_{j\in\mathcal{N}_i}(\widehat{x}_i - \widehat{x}_j)^2 - 2a_i e_i\sum_{j\in\mathcal{N}_i}(\widehat{x}_i - \widehat{x}_j) + a_i^2\frac{d\chi_i}{d\tau_i}.$$

Therefore, to ensure $\dot{V} \le 0$, we choose each $\frac{d\chi_i}{d\tau_i}$ for $\dot{V}_i \le 0$:

$$\frac{d\chi_i}{d\tau_i} = \sigma_i\sum_{j\in\mathcal{N}_i}(\widehat{\alpha}_i - \frac{a_j}{a_i}\widehat{\alpha}_j)^2 + 2e_i\sum_{j\in\mathcal{N}_i}(\widehat{\alpha}_i - \frac{a_j}{a_i}\widehat{\alpha}_j), \quad (14)$$

where $\sigma_i \in (0,1)$ is a design parameter. Now, we have

$$\dot{V} = \sum_{i=1}^{N} -(1-\sigma_i)\sum_{j\in\mathcal{N}_i}(\widehat{x}_i - \widehat{x}_j)^2 \le 0. \quad (15)$$

Notice that, immediately after agent i triggers an event, $e_i = 0$ so $\frac{d\chi_i}{d\tau_i} = \sigma_i\sum_{j\in\mathcal{N}_i}(\widehat{\alpha}_i - \frac{a_j}{a_i}\widehat{\alpha}_j)^2 \ge 0$. Therefore, χ_i never becomes negative with this dynamics.

4.1 Main Results

Here we formally describe the algorithm from the perspective of a single agent i and provide the main results by discussing its properties. Each agent has a design parameter $\sigma_i \in (0,1)$, controlling the trade-off between convergence speed and amount of communication. The event-triggered clock synchronization algorithm is summarized in Table 1.

Theorem 4.3. (Positive MIET). Given the communication graph $\mathcal{G}$ and unknown $a_i > 0$ for all agents $i \in \{1, \dots, N\}$, if each agent i implements the distributed dynamic event-triggered coordination algorithm presented in Table 1 with $\sigma_i \in (0,1)$, then the inter-event times for agent i are lower-bounded by $\mathcal{T}_i \triangleq \frac{\sigma_i}{a_i|\mathcal{N}_i|}$. That is, $t_{\ell+1}^i - t_\ell^i \ge \mathcal{T}_i$ for all $i \in \{1, \dots, N\}$ and $\ell \in \mathbb{Z}_{\ge 0}$.

Proof. To show that each algorithm exhibits a positive MIET, we use the change of variables $\eta_i e_i^2 \triangleq \chi_i$. Note $\eta_i = 0$ for agent i's trigger condition to be satisfied, and so we define η_i as being reset to $\overline{\eta}_i > 0$ at each event, and lower bounding the time η_i takes to reach 0 from $\overline{\eta}_i$ gives a lower bound on the inter-event times. Taking the derivative, we have

$$\dot{\eta}_i = a_i\sigma_i\frac{\xi_i^T\xi_i}{e_i^2} + 2a_i(\eta_i + 1)\frac{\mathbf{1}_{n_i}^T\xi_i}{e_i},$$

Initialization; at time $t = 0$ each agent $i \in \{1, \dots, N\}$ performs:
1: Initialize $\widehat{\alpha}_i = \alpha_i = 1$
2: Initialize $\chi_i = 0$

At all times t each agent $i \in \{1, \dots, N\}$ performs:
1: **if** $\chi_i = 0$ and $e_i \ne 0$ **then**
2: set $\widehat{\alpha}_i = \alpha_i$ (broadcast state information to neighbors)
3: set $\frac{d\alpha_i}{d\tau_i} = -\sum_{j\in\mathcal{N}_i}(\widehat{\alpha}_i(\tau_i) - \frac{a_j}{a_i}\widehat{\alpha}_j(\tau_i))$ (update control signal)
4: propagate χ_i according to its dynamics given in (14)
5: **end if**
6: **if** new information $\widehat{\alpha}_k$ is received from some neighbor(s) $k \in \mathcal{N}_i$ **then**
7: update control signal $\frac{d\alpha_i}{d\tau_i} = -\sum_{j\in\mathcal{N}_i}(\widehat{\alpha}_i(\tau_i) - \frac{a_j}{a_i}\widehat{\alpha}_j(\tau_i))$
8: **end if**

Table 1. Distributed Dynamic Event-Triggered Clock Synchronization Algorithm.

where we define $\xi_i \triangleq [\dots, \widehat{\alpha}_i - \frac{a_j}{a_i}\widehat{\alpha}_j, \dots]^T \in \mathbb{R}^{n_i}$ and $n_i \triangleq |\mathcal{N}_i|$ to write the summations as matrix multiplications. We apply Young's inequality (1) with $b_i > 0$:

$$\dot{\eta}_i \ge a_i\sigma_i\frac{\xi_i^T\xi_i}{e_i^2} - b_i(\eta_i + 1)^2 - \frac{a_i^2}{b_i}\left(\frac{\mathbf{1}_{n_i}^T\xi_i}{e_i}\right)^2.$$

We now wish to choose b_i to achieve a lower bound on the state dependent terms, which means we must satisfy

$$b_i \ge \frac{a_i\left(\mathbf{1}_{n_i}^T\xi_i\right)^2}{\sigma_i\xi^T\xi} = \frac{a_i\xi_i^T\left(\mathbf{1}_{n_i}\mathbf{1}_{n_i}^T\right)\xi_i}{\sigma_i\xi^T I_{n_i}\xi}.$$

We can upper bound the right hand side with $\frac{a_i\text{eigmax}\left(\mathbf{1}_{n_i}\mathbf{1}_{n_i}^T\right)}{\sigma_i\text{eigmin } I_{n_i}} = \frac{a_i n_i}{\sigma_i}$, and so we choose $b_i = \frac{a_i n_i}{\sigma_i}$. Now, we have $\dot{\eta}_i \ge -\frac{a_i n_i}{\sigma_i}(\eta_i + 1)^2 \triangleq \dot{\underline{\eta}}_i$, and we take $\underline{\eta}(0) = \overline{\eta}_i$, so $\underline{\eta}_i \le \eta_i$. Solving this differential equation, finding $\mathcal{T}_i$ such that $\underline{\eta}(\mathcal{T}_i) = 0$, and taking the limit as $\overline{\eta}_i \to \infty$ (because this constant has no physical meaning), we have the formula for $\mathcal{T}_i$ given in Theorem 4.3. □

Next, we present our main convergence result.

Theorem 4.4. (Asymptotic Convergence). Given the communication graph $\mathcal{G}$ and unknown $a_i > 0$ for all agents $i \in \{1, \dots, N\}$, if each agent i implements the distributed dynamic event-triggered coordination algorithm presented in Table 1 with agent i triggering events when $\chi_i = 0$ and $e_i \ne 0$ and with $\sigma_i \in (0,1)$, then all trajectories of the system are guaranteed to asymptotically converge to the set $\mathcal{A} \triangleq \{q | a_i\widehat{\alpha}_i = a_j\widehat{\alpha}_j \forall (i,j \in \{1, \dots, N\}\}$.

Proof. For reasons of space, the full proof is omitted. Instead, we provide a brief outline of it and refer interested readers to Berneburg and Nowzari (2019a), where a very similar proof of convergence is given.

By (15), the Lyapunov function V is decreasing when the system isn't in the target set $\mathcal{A}$. Additionally, from (12) V does not depend on $\widehat{\alpha}$, so it does not change during broadcasts. Combining this with Theorem 4.3 to preclude Zeno behavior, we use an invariance principle to show that the system convergences asymptotically to the set $\mathcal{A}$.

More specifically, we write our system as a hybrid dynamical system as described in Goebel et al. (2012), and we make use of the Invariance Principle for Hybrid Sys-

tems (Goebel et al., 2012, Theorem 8.2). Please see the proof of Theorem V.2 in Berneburg and Nowzari (2019a) for more details from a very similar proof. □

Remark 4.5. (Convergence). From Theorem 4.4, it is evident that this algorithm provides only a weak convergence result and cannot guarantee that $\lim_{t\to\infty}|T_i(t)-T_j(t)|=0$ for all $i,j\in\{1,\dots,N\}$ for two reasons. The first is that agents can get stuck with $\sum_{j\in\mathcal{N}_i}(\widehat{\alpha}_i-\frac{a_j}{a_i}\widehat{\alpha}_j)^2=0$ while $e_i\neq 0$. However, if the algorithm is modified with an additional trigger that guarantees that each agent i will always broadcast again, eventually, when $e_i\neq 0$, then $\lim_{t\to\infty}|x_i(t)-x_j(t)|=0$ for all $i,j\in\{1,\dots,N\}$. A simple way to do this is to make the secondary trigger periodic with any period greater than or equal to the MIET. The second is that convergence of x is not equivalent to convergence of $T\triangleq[T_1,T_2,...T_N]^T=xt$, because the latter requires that x converges fast enough. Although *proving* x converges fast enough for T to converge is beyond the scope of this paper, simulations suggest this is true, and many existing papers conclude only that x converges. •

5. SIMULATIONS

To demonstrate our distributed event-triggered clock synchronization algorithm, we perform simulations using $N=5$ agents and a graph with adjacency matrix

$$A=\begin{bmatrix}0&1&0&1&0\\1&0&0&1&0\\0&0&0&0&1\\1&1&0&0&1\\0&0&1&1&0\end{bmatrix}. \tag{16}$$

For simplicity, all agents have the same design parameter $\sigma_i=\sigma$, for all $i\in\{1,\dots,N\}$. The drift of the clocks was $a=[5,3.2,0.6,7,1.4]$ for all simulations. Note that, in practice, $a_i\simeq 1$ for $i\in\{1,\dots,N\}$, so these clock drifts have been greatly exaggerated for demonstration purposes. Additionally, to ensure full convergence as noted in Remark 4.5, we included a secondary trigger, so that each agent will trigger an event at most 2 seconds (according to its local hardware clock) after its previous event.

Considering each agent's difference from the average as an output, similar to (Dezfulian et al., 2018), we adopt the square of the $\mathcal{H}_2$-norm of the system, and the average communication rate, as performance metrics

$$\mathcal{C}\triangleq\int_0^\infty\sum_{i=1}^N(x_i(t)-\bar{x})^2\,dt,$$
$$r_{\text{comm}}\triangleq\frac{\text{Number of events}}{\text{Time duration of simulation}}.$$

Plots of simulation results with $\sigma=0.5$ appear in Figure 1. The top plot in (a) shows the evolution of the software clocks for all agents with respect to absolute time. Note that they do not converge to true time, but to a common clock that is faster than real time. The exaggerated clock drifts enable this convergence to be clearly visible, and also show how the software clocks may decrease but remain continuous. All these clocks start from zero because there is no bias. The bottom plot shows how the cyber component of the Lyapunov is necessary, because the physical component is allowed to increase. (b) shows the dynamic variable for one agent and the times of events for all agents. Note that events are triggered for agent 5 when the dynamic variable reaches zero, and that the first event, for that agent, instead occurs due to the secondary trigger. The lower plot shows that all agents trigger events aperiodically. Figure 2 shows the results of varying the design parameter σ, demonstrating how σ controls the trade-off between speed of convergence and communication, as increasing σ results in a lower communication rate but tends to result in a higher cost.

6. CONCLUSION

This paper proposes a novel fully-distributed event-triggered algorithm to solve the clock synchronization problem for wireless sensor networks, where only clock drift is considered, and its communication strategy ensures that a desired minimum inter-event time is observed. Future work can include modifying the control inputs to ensure that the virtual clocks do not run backwards and extending the results to cases where both clock drift and bias are present, as well as guaranteeing that the software clocks converge to consensus, instead of only guaranteeing the convergence of the software clock drifts. Another point of further research is modifying the algorithm to handle communication delays.

REFERENCES

Berneburg, J. and Nowzari, C. (2019a). Distributed dynamic event-triggered algorithms with positive minimum inter-event times on weight-balanced digraphs. In *Conference on Decision and Control*, 2598 – 2603. Nice, France.

Berneburg, J. and Nowzari, C. (2019b). Distributed dynamic event-triggered coordination with a designable minimum inter-event time. In *American Control Conference*, 1424 – 1429. Philadelphia, PA.

Berneburg, J. and Nowzari, C. (2019c). Robust dynamic event-triggered coordination ith a designable minimum inter-event time. *arXiv preprint arXiv:1903.04734*.

Borgers, D.P. and Heemels, W.P.M.H. (2014). Event-separation properties of event-triggered control systems. *IEEE Transactions on Automatic Control*, 59(10), 2644–2656.

Carli, R. and Zampieri, S. (2014). Network clock synchronization based on the second-order linear consensus algorithm. *IEEE Transactions on Automatic Control*, 59(2), 409–422.

De Persis, C. and Frasca, P. (2013). Robust self-triggered coordination with ternary controllers. *IEEE Transactions on Automatic Control*, 58(12), 3024–3038.

De Persis, C. and Postoyan, R. (2017). A Lyapunov redesign of coordination algorithms for cyber-physical systems. *IEEE Transactions on Automatic Control*, 62(2), 808–823.

Dengchang, Z., Zhulin, A., and Yongjun, X. (2013). Time synchronization in wireless sensor networks using max and average consensus protocol. *International Journal of Distributed Sensor Networks*, 9(3), 1–10.

Dezfulian, S., Ghaedsharaf, Y., and Motee, N. (2018). On performance of time-delay linear consensus networks with directed interconnection topologies. In *American Control Conference*, 4177–4182. Milwaukee, USA.

Dolk, V.S., Abdelrahim, M., and Heemels, W.P.M.H. (2017). Event-triggered consensus seeking under non-uniform time-varying delays. *IFAC-PapersOnLine*, 50(1), 10096–10101.

Garcia, E., Mou, S., Cao, Y., and Casbeer, D.W. (2017). An event-triggered consensus approach for distributed clock synchronization. In *American Control Conference*, 279–284. Seattle, WA.

Garone, E., Gasparri, A., and Lamonaca, F. (2015). Clock synchronization protocol for wireless sensor networks

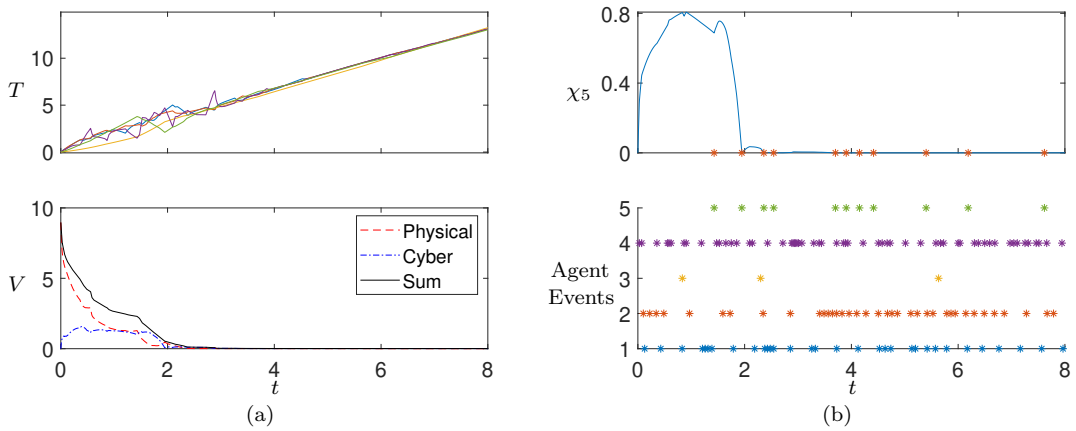

Fig. 1. Plots of the simulation results of the clock synchronization algorithm with $\sigma = 0.5$. (a) shows the trajectories of the virtual clocks $T_i = x_i t = \alpha_i a_i t$ (top), and the evolution of the Lyapunov function and its components (bottom). (b) shows the dynamic variable χ_5 for agent 5 with the times of its events marked with stars (top), and rows of stars indicating the event times for each agent (bottom).

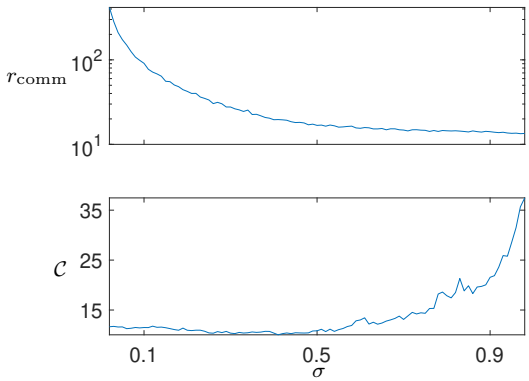

Fig. 2. Statistical results of varying σ. The top plot shows the average number of broadcasts per second, and the bottom plot shows the cost.

with bounded communication delays. *Automatica*, 59, 60–72.

Goebel, R., Sanfelice, R.G., and Teel, A.R. (2012). *Hybrid Dynamical Systems*. Princeton University Press, 41 William Street, Princeton, New Jersey 08540.

Hardy, G.H., Littlewood, J.E., and Polya, G. (1952). *Inequalities*. Cambridge University Press, Cambridge, UK.

He, J., Cheng, P., Shi, L., Chen, J., and Sun, Y. (2014). Time synchronization in WSNs: A maximum-value based consensus approach. *IEEE Transactions on Automatic Control*, 59(3), 660–675.

He, J., Duan, X., Cheng, P., Shi, L., and Cai, L. (2017). Accurate clock synchronization in wireless sensor networks with bounded noise. *Automatica*, 81, 350–358.

Heemels, W.P.M.H., Johansson, K.H., and Tabuada, P. (2012). An introduction to event-triggered and self-triggered control. In *IEEE Conference on Decision and Control*, 3270–3285. Maui, Hawaii, USA.

Kadowaki, Y. and Ishii, H. (2015). Event-based distributed clock synchronization for wireless sensor networks. *IEEE Transactions on Automatic Control*, 60(8), 2266–2271.

Li, Z.C.D., Huang, Y., and Tang, C. (2015). Event-triggered communication for distributed time synchronization in wsns. In *Control Conference (CCC), 2015 34th Chinese*, 7789–7794. Hangzhou, China.

Liu, B., Lu, W., and Chen, T. (2011). Consensus in networks of multiagents with switching topologies modeled as adapted stochastic processes. *SIAM Journal on Control and Optimization*, 49(1), 227–253.

Nowzari, C., Garcia, E., and Cortés, J. (2019). Event-triggered communication and control of networked systems for multi-agent consensus. *Automatica*, 105, 1 – 27.

Olfati-Saber, R. and Murray, R.M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), 1520–1533.

Postoyan, R., Tabuada, P., Nesic, D., and Anta, A. (2015). A framework for the event-triggered stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 60(4), 982–996.

Proskurnikov, A.V. and Mazo, Jr., M. (2018). Lyapunov design for event-triggered exponential stabilization. In *HSCC '18: 21st International Conference on Hybrid Systems: Computation and Control*, 111–119. Porto, Portugal.

Ren, W., Beard, R.W., and Atkins, E.M. (2007). Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2), 71–82.

Schenato, L. and Fiorentin, F. (2011). Average TimeSynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica*, 49(9), 1878–1886.

Stanković, M.S., Stanković, S.S., and Johansson, K.H. (2018). Distributed offset correction for time synchronization in networks with random delays. In *European Control Conference*, 2212–2217. Limassol, Cyprus.

Xu, P., Nowzari, C., and Tian, Z. (2019). A class of distributed event-triggered average consensus algorithms for multi-agent systems. *arXiv preprint arXiv:1906.02649*.

Zhang, X., Chen, H., Lin, K., Wang, Z., Yu, J., and Shi, L. (2019). RMTS: A robust clock synchronization scheme for wireless sensor networks. *Journal of Network and Computer Applications*, 135(1), 1–10.