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

An Adaptive Hybrid Control Algorithm for Sender-Receiver Clock Synchronization

Marcello Guarro * **Ricardo Sanfelice** *

* *Department of Electrical and Computer Engineering University of California, Santa Cruz, CA 95060 USA. (e-mail: mguarro@soe.ucsc.edu, ricardo@ucsc.edu)*

Abstract: This paper presents an innovative hybrid systems approach to the sender-receiver synchronization of timers. Via the hybrid systems framework, we unite the traditional sender-receiver algorithm for clock synchronization with an online, adaptive strategy to achieve synchronization of the clock rates to exponentially synchronize a pair of clocks connected over a network. Following the conventions of the algorithm, clock measurements of the nodes are given at periodic time instants, and each node uses these measurements to achieve synchronization. For this purpose, we introduce a hybrid system model of a network with continuous and impulsive dynamics that captures the sender-receiver algorithm as a state-feedback controller to synchronize the network clocks. Moreover, we provide sufficient design conditions that ensure attractivity of the synchronization set.

Keywords: Hybrid and switched systems modeling; Control over networks; Sensor networks

1. INTRODUCTION

In distributed systems, the coordination of time among discrete nodes is an inherent necessity to the implementation of many distributed systems that rely on event-ordering. However, in the case of distributed systems and algorithms acting on a dynamical process, the consensus on time among the distributed agents must also accurately capture *when* an event occurs to ensure the desired system function. In fact, the lack of accurate time consensus among the distributed agents can result in performance issues that affect the overall system, see Graham and Kumar (2004) and Wei Zhang et al. (2001). The coordination of time (or clock synchronization) is achieved through the consensus of the internal clocks at each distributed agent. However, communication delays in the distributed network, differing rates of change in the clocks of each node, and the lack of an absolute time reference, pose a unique set of challenges to the problem of clock synchronization for which has yielded a number of proposed solutions (see Wu et al. (2010) and Sundararaman et al. (2005)).

Among the number of existing algorithms for clock synchronization, sender-receiver (or two-way) based synchronization algorithm underpins several ubiquitous clock synchronization protocols including NTP (see Mills (1991)), PTP (see IEEE (2008)), and TPSN (see Ganeriwal et al. (2003)). The sender-receiver algorithm relies on the existence of a known reference that is either injected to the system or provided by an elected agent in the distributed system; synchronization is then achieved through a series of chronologically ordered two-way message exchanges between each synchronizing node and the known reference. With sufficient information from the exchanged messages, the relative differences in the clock rates and offset can be estimated and applied as a correction to the clock of the synchronizing node, see Freris et al. (2010). However, while the difference in the output can be determined and implemented online, the relative clock rate is estimated through offline filtering techniques (see Mills (1991)) or least-squares estimation (see Wu et al. (2010)).

In this paper, we present a hybrid approach to sender-receiver synchronization with an, online, adaptive method to synchronize the clock rates. Using the hybrid systems framework, we show that our algorithm exponentially synchronizes a pair of clocks connected over a network while preserving the messaging protocols and network dynamics of traditional sender-receiver algorithms. Unlike the existing algorithms of NTP, PTP, and TPSN, our proposed solution provides a Lyapunov-based convergence analysis to a set in which the clocks are synchronized with sufficient conditions ensuring their synchronization. We emphasize to the reader that previous analysis on sender-receiver synchronization has only provided results to its feasibility and that the literature lacks results to its performance in a dynamical system setting.

This paper is organized as follows: Section 2 introduces the sender-receiver algorithm as presented in the literature. Section 3 outlines the motivation for this paper. Section 5.1 presents some preliminary material on hybrid systems. Section 5 formally introduces the problem under consideration and the hybrid model that solves it. Section 5.4 details the main results, while Section 5.5 provides numerical examples. Due to space constraints, the proofs of the results along with other details have been omitted and will be published elsewhere.

⋆ This research partially supported by NSF Grants no. ECS-1710621 and CNS-1544396, by AFOSR Grants no. FA9550-16-1-0015, FA9550-19-1-0053, and Grant no. FA9550-19-1-0169, and by CITRIS and the Banatao Institute at the University of California.

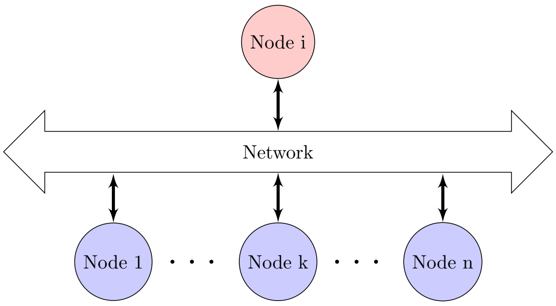


Fig. 1. General architecture of the system under consideration.

Notation: In this paper the following notation and definitions will be used. The set of natural numbers including zero, i.e., $\{0, 1, 2, \dots\}$ is denoted by $\mathbb{N}$. The set of natural numbers is denoted as $\mathbb{N}_{>0}$, i.e., $\mathbb{N}_{>0} = \{1, 2, \dots\}$. The set of real numbers is denoted as $\mathbb{R}$. The set of non-negative real numbers is denoted by $\mathbb{R}_{\geq 0}$, i.e., $\mathbb{R}_{\geq 0} = [0, \infty)$. The n-dimensional Euclidean space is denoted $\mathbb{R}^n$. Given topological spaces A and B, $F : A \rightrightarrows B$ denotes a set-valued map from A to B. For a matrix $A \in \mathbb{R}^{n\times m}$, $A^\top$ denotes the transpose of A. Given a vector $x \in \mathbb{R}^n$, $|x|$ denotes the Euclidean norm. Given two vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, $(x, y) = [x^\top \;\; y^\top\,]^\top$. For two symmetric matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{n\times m}$, $A \succ B$ means that $A - B$ is positive definite, conversely $A \prec B$ means that $A - B$ is negative definite. Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, the range of f is given by rge $f := \{y \mid \exists\, x \text{ with } y = f(x)\}$.

2. PRELIMINARIES ON THE SENDER-RECEIVER ALGORITHM

In a network of n nodes, consider nodes i and k in a sender-receiver hierarchy where Node i is a designated reference or parent agent of a synchronizing child agent Node k, see Figure 1. Each node has an attached internal clock $\tau_i, \tau_k \in \mathbb{R}$ whose dynamics are given by

$$\begin{aligned} \dot{\tau}_i &= a_i \\ \dot{\tau}_k &= a_k \end{aligned} \tag{1}$$

where a_i,$a_k \in \mathbb{R}_{>0}$ denote the respective clock drift or skew. At times t_j for $j \in \mathbb{N}$ (with $t_0 = 0$), nodes i and k exchange timing measurements with embedded timestamps

$$\begin{aligned} T^i_j &:= a_i t_j + \tau_i(0) \\ T^k_j &:= a_k t_j + \tau_k(0) \end{aligned} \tag{2}$$

The goal is to then synchronize the internal clock of Node k to that of Node i using the exchanged timing measurements.

For a sequence of time instants $\{t_j\}^\infty_{j=1}$ that is assumed to be strictly increasing and unbounded, at each t_j the standard sender-receiver synchronization algorithm as described in the literature (see Wu et al. (2010), Freris et al. (2009), and Eidson (2006)) is given as follows:

(P1) At time t_j, Node i broadcasts a synchronization message with its time T^i_j to Node k.
(P2) At time t_{j+1}, Node k receives the synchronization message and records its time of arrival T^k_{j+1}
(P3) At time t_{j+2}, Node k sends a response message with timestamp T^k_{j+2}

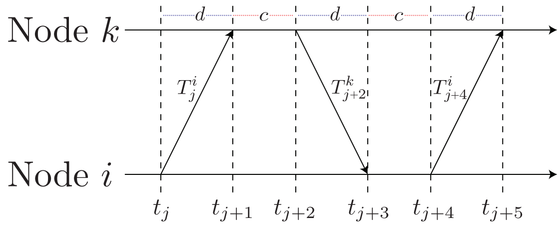


Fig. 2. Diagram illustrating the message exchange between Nodes i and k for the synchronization algorithm.

(P4) At time t_{j+3}, Node i receives the response message from Node k and records its time of arrival T^i_{j+3}
(P5) At time t_{j+4}, Node i sends a response receipt message with timestamp T^i_{j+4}
(P6) At time t_{j+5}, Node k receives the response message from Node i and records its time of arrival T^k_{j+5} and then updates its clock to synchronize with the clock of Node i using the collected timestamps T^i_j, T^k_{j+1}, T^k_{j+2}, T^i_{j+3}, and T^i_{j+4}.

Moreover, as done in the literature, it is assumed that the time elapsed between each time instant is given by

$$t_{j+1} - t_j = \begin{cases} d & \forall j \in \{2i + 1 : i \in \mathbb{N}\}, j > 0 \\ c & \forall j \in \{2i : i \in \mathbb{N}\}, j > 0 \end{cases} \tag{3}$$

where $0 < c \leq d$. The constant c defines the residence or response time delay while d defines the propagation delay of the message transmission.

Most pairwise synchronization protocols such as the Network Time Protocol (NTP), Precision Time Protocol (PTP, IEEE 1588), and the Timing-sync Protocol for Sensor Networks (TPSN) assume that the propagation delay in the message transmission from parent to child and child to parent is symmetric. If the propagation delay between the two nodes is asymmetric it introduces an error to the calculated offset correction that cannot be accounted for, see Freris et al. (2010). Thus, the propagation delay and residence time are assumed to be symmetric.

With the available timestamps, at times t_{j+5}, the relative offset $\tilde{o} := \tau_i(0) - \tau_k(0)$ is calculated via

$$u_{\tilde{o}} = \frac{1}{2}\Big((T^i_{j+3} - T^k_{j+2}) - (T^k_{j+1} - T^i_j)\Big) \tag{4}$$

by making the appropriate substitutions one has

$$\begin{aligned} u_{\tilde{o}} &= \frac{1}{2}\Big(\big((a_i t_{j+3} + \tau_i(0)) - (a_k t_{j+2} + \tau_k(0))\big) \\ &\qquad - \big((a_k t_{j+1} + \tau_k(0)) - (a_i t_j + \tau_i(0))\big)\Big) \\ &= \frac{1}{2}\Big(\big(a_i t_{j+3} - a_k t_{j+2} + \tilde{o}\big) - \big(a_k t_{j+1} - a_i t_j - \tilde{o}\big)\Big) \end{aligned}$$

Rearranging terms gives,

$$u_{\tilde{o}} = \tilde{o} + \frac{1}{2}\Big(\big(a_i t_{j+3} - a_k t_{j+2}\big) - \big(a_k t_{j+1} - a_i t_j\big)\Big) \tag{5}$$

If the clock drifts are synchronized, i.e., $a_k = a_i$, then

$$u_{\tilde{o}} = \tilde{o} + \frac{1}{2}\Big(\big(a_i(t_{j+3} - t_{j+2})\big) - \big(a_i(t_{j+1} - t_j)\big)\Big)$$

Then, by noting the bounds on the time elapsed between time instants t_j, as given in (3), one has

$$t_{j+1} - t_j = t_{j+3} - t_{j+2} = d \tag{6}$$

Making the appropriate substitutions in (5) gives

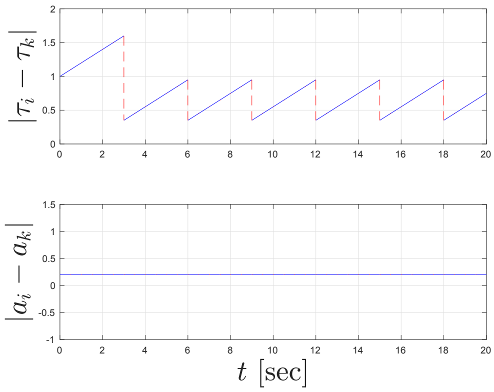


Fig. 3. The evolution of the error in the clocks and clock rates of Nodes i and k when the algorithm only applies the offset correction $u_{\tilde{o}}$.

$$u_{\tilde{o}} = \tilde{o} + \frac{1}{2}\big(a_i d - a_i d\big) = \tilde{o}$$

which is then applied to the clock state of Node k at times t_{j+5}. To demonstrate how this solves the synchronization problem, consider the error between the clocks of nodes i and k at t_{j+5},

$$\begin{aligned}\tau_i(t_{j+5}) - \tau_k(t_{j+5}) &= \tau_i(t_{j+5}) - (\tau_k(t_{j+5}) - u_{\tilde{o}}) \\ &= \big(a_i t_{j+5} + \tau_i(0)\big) \\ &\quad - \big(a_k t_{j+5} + \tau_k(0) - (\tau_i(0) - \tau_k(0))\big) \\ &= a_i t_{j+5} - a_k t_{j+5} \\ &= 0\end{aligned}$$

Here $\tau_k(t_{j+5})$ is replaced by $\tau_k(t_{j+5}) - u_{\tilde{o}}$ where $u_{\tilde{o}}$ is defined in (5). Thus, the clocks at nodes i and k synchronize for the case where the clock drifts are already assumed to be synchronized.

3. MOTIVATION FOR AN ADAPTIVE CLOCK SYNCHRONIZATION ALGORITHM

Now, consider the following system data $a_i = 1$, $a_k = 0.8$ with $c = d = 0.5$ and the given sender-receiver algorithm with only the offset correction $u_{\tilde{o}}$ being applied. After, simulating the algorithm, Figure 3 shows the plots of the behavior in the error of clocks and the clock rates. As depicted in the figure, the algorithm continually applies the offset correction but due to the mismatch in the clock rates, the error in the clocks fails to converge to zero. This is further evidenced analytically when noting that a mismatch in the clock rates in equation (5) yields an error on the offset $\tilde{o}$ in (4).

Though various strategies exist to mitigate the effects of the error from the mismatched clock rates, the choice of strategy is often left to the system designer, see Eidson (2006). Moreover, these methods are often complicated to implement and too expensive for low-cost applications such as sensor networks. In fact, protocols such as TPSN, designed specifically for low-cost sensor networks, do not provide provisions to correct for the clock rate error, see Ganeriwal et al. (2003). Finally, the authors are not aware of any proposed sender-receiver algorithm that provides convergence guarantees for both offset and clock rate correction.

4. PROPOSED ALGORITHM

Given the inability of the sender-receiver algorithm to synchronize the clocks, we propose a modification to the algorithm that incorporates an adaptive strategy to synchronize the clock rates. Consider the control law for the synchronization of the clock rate for Node k

$$u_a = \mu(T^i_{j+4} - T^i_j - T^k_{j+5} - T^k_{j+1}) \tag{7}$$

with $\mu > 0$ being a controllable parameter. Making the necessary substitutions one has

$$\begin{aligned}u_a &= \mu\Big(\big(a_i t_{j+4} + \tau_i(0)\big) - \big(a_i t_j + \tau_i(0)\big) \\ &\qquad - \big(a_k t_{j+5} + \tau_k(0)\big) - \big(a_k t_{j+1} + \tau_k(0)\big)\Big) \\ &= \mu\big(a_i(2c + 2d) - a_k(2c + 2d)\big) \\ &= \mu(2c + 2d)\big(a_i - a_k\big)\end{aligned} \tag{8}$$

The correction u_a can then be applied to the clock dynamics of Node k at times t_{j+5} as follows:

$$a_k^+ = a_k + u_a = a_k + \mu(2c + 2d)\big(a_i - a_k\big) \tag{9}$$

Observe that this strategy operates under the existing assumptions of the sender-receiver algorithm (symmetric propagation delays and residence times) and does not rely on any additional information that is not already available via the exchanged timing messages. Moreover, since it exploits the integrator dynamics of the system, the computation costs to calculate u_a are minimal. In this next example, we demonstrate the proposed strategy under the same scenario of mismatched skews between Nodes i and k.

To illustrate, the capabilities of the algorithm outlined above, consider the same system data as in Section 3, namely, $a_i = 1$, $a_k = 0.8$ with $c = d = 0.5$ and the given sender-receiver algorithm now with both the offset correction $u_{\tilde{o}}$ and clock rate correction u_a being applied. In Figure 4, two sets of error plots are presented for two different simulations. Figure 4(a) gives plots of the errors for the case where the μ is chosen using information on c and d following our forthcoming design conditions while Figure 4(b) provide the error plots for the case where μ is chosen arbitrarily. In the case of the ideal μ, the error in the clocks and clock rate converge to zero whereas in the case of the arbitrarily chosen μ, the error fails to converge. This suggests that a sufficient condition to appropriately design μ is necessary to ensure convergence of the error.

5. A HYBRID ALGORITHM FOR SENDER-RECEIVER CLOCK SYNCHRONIZATION

In this section, we present our hybrid model that unites the characterization of the network dynamics for the message exchange with our proposed algorithm that ensures synchronization of the clocks. In addition, we present results and simulations to validate our model.

5.1 Preliminaries on Hybrid Systems

A hybrid system $\mathcal{H}$ in $\mathbb{R}^n$ is composed by the following *data*: a set $C \subset \mathbb{R}^n$, called the flow set; a set-valued mapping $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ with $C \subset \text{dom}\, F$, called the flow map; a set $D \subset \mathbb{R}^n$, called the jump set; a set-valued mapping $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ with $D \subset \text{dom}\, G$, called the jump

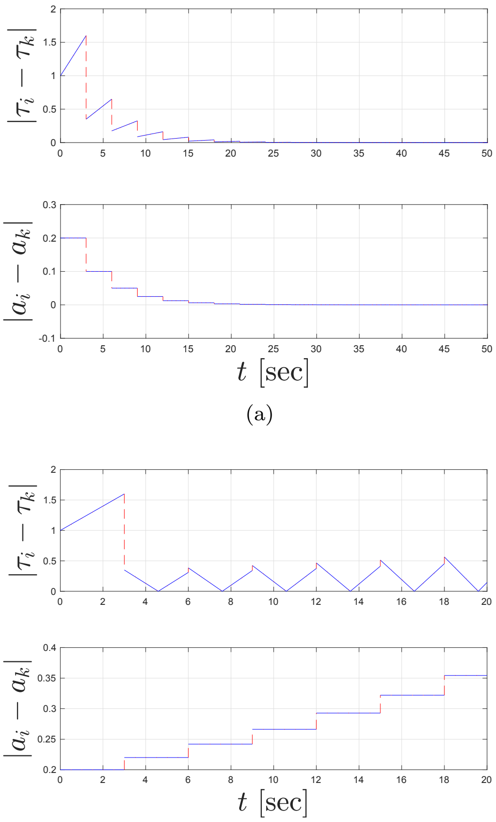


Fig. 4. The evolution of the error in the clocks and clock rates of Nodes i and k when the algorithm applies both offset correction $u_{\bar{o}}$ and clock rate correction u_a. Plot (a) depicts the scenario where μ is optimally chosen while plot (b) demonstrates the case when μ is chosen arbitrarily.

map. Then, a hybrid system $\mathcal{H} := (C, F, D, G)$ is written in the compact form

$$\mathcal{H} \begin{cases} \dot{x} \ \in F(x) & x \in C \\ x^+ \in G(x) & x \in D \end{cases} \tag{10}$$

where x is the system state. Solutions to hybrid systems are parameterized by (t, j), where $t \in \mathbb{R}_{\geq 0}$ defines ordinary time and $j \in \mathbb{N}$ is a counter that defines the number of jumps. The evolution of a solution is described by a *hybrid arc* ϕ on a *hybrid time domain* Goebel et al. (2012). A hybrid time domain is given by dom $\phi \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ if, for each $(T, J) \in$ dom ϕ, dom $\phi \cap ([0, T] \times \{0, 1, ..., J\})$ is of the form $\bigcup_{j=0}^{J}([t_j, t_{j+1}] \times \{j\})$, with $0 = t_0 \leq t_1 \leq t_2 \leq t_{J+1}$. A solution ϕ is said to be *maximal* if it cannot be extended by flow or a jump, and *complete* if its domain is unbounded. The set of all maximal solutions to a hybrid system $\mathcal{H}$ is denoted by $\mathcal{S}_{\mathcal{H}}$ and the set of all maximal solutions to $\mathcal{H}$ with initial condition belonging to a set A is denoted by $\mathcal{S}_{\mathcal{H}}(A)$. A hybrid system is *well-posed* if it satisfies the hybrid basic conditions in (Goebel et al., 2012, Assumption 6.5). Let $\mathcal{A} \subset \mathbb{R}^n$ be a closed set and $|x|_{\mathcal{A}} := \inf_{y \in \mathcal{A}} |x - y|$. A closed set $\mathcal{A} \subset \mathbb{R}^n$ is said to be: *attractive* for $\mathcal{H}$ if there exists $\mu > 0$ such that every solution ϕ to $\mathcal{H}$ with $|\phi(0, 0)|_{\mathcal{A}} \leq \mu$ is complete and satisfies $\lim_{t+j \to \infty} |\phi(t, j)|_{\mathcal{A}} = 0$.

5.2 Problem Statement

Our goal is to synchronize the internal clock of Node k to that of Node i. In particular, our goal is to design a hybrid algorithm incorporating the sender-receiver algorithm such that the clock τ_k and clock rate a_k of Node k is driven to synchronization with τ_i and a_i of Node i, respectively. Moreover, our objective is to provide tractable conditions that ensures attractivity. This problem is formally stated as follows:

Problem 5.1. Given two nodes in a sender-receiver hierarchy with clocks having dynamics as in (1) with timestamps T_j^i, T_j^k and parameters c, d, design a hybrid algorithm such that each trajectory $t \mapsto (\tau_i(t), \tau_k(t))$ satisfies $\lim_{t\to\infty} |\tau_i(t) - \tau_k(t)| = 0$ and $\lim_{t\to\infty} |\dot{\tau}_i(t) - \dot{\tau}_k(t)| = 0$

First, to model the hardware and communication dynamics of the system, namely, the residence and transit times elapsed between the timing messages, we consider a global timer $\tau \in [0, d]$ with dynamics

$$\dot{\tau} = -1 \qquad \tau \in [0, d] \tag{11}$$

when $\tau = 0$, the state τ^+ is reset to either c or d, respectively, corresponding to a communication or a response event while preserving the bounds given in (3). Additionally, a discrete variable $q \in \{0, 1\} =: \mathcal{Q}$ is included to indicate the "transmission" or "resident" state of the protocol. Furthermore, each reset of τ either triggers Column vectors

$$m^i = [m_1^i, m_2^i, m_3^i, m_4^i, m_5^i, m_6^i]^\top \in \mathbb{R}^6$$
$$m^k = [m_1^k, m_2^k, m_3^k, m_4^k, m_5^k, m_6^k]^\top \in \mathbb{R}^6$$

represent memory buffers to store the received and transmitted timestamps for the respective parent and child nodes. In addition, a second discrete variable $p \in \{0, 1, 2, 3, 4, 5\} =: \mathcal{P}$ is used to track the state of the protocol corresponding to the events defined in (P1)-(P6) of the sender-receiver algorithm. Then, by incorporating the clocks τ_i, τ_k and the clock rates a_i, a_k as state variables to the model, the state x of the complete hybrid system is defined as

$$x := (\tau_i, \tau_k, a_i, a_k, \tau, m^i, m^k, p, q) \in \mathcal{X}$$

where

$$\mathcal{X} := \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times [0, d] \times \mathbb{R}^6 \times \mathbb{R}^6 \times \mathcal{P} \times \mathcal{Q}$$

Then by noting the dynamics of the clocks as given in (1) and those of the timer τ above, the continuous dynamics of x is given by the following flow map

$$f(x) = (a_i, a_k, 0, 0, -1, 0, 0, 0, 0) \quad \forall x \in C := \mathcal{X}$$

To model the discrete dynamics of the communication and arrival events of the exchanged timing messages, in addition to the subsequent corrections on the clock rate and offset, we consider the jump map $G : \mathbb{R}^n \to \mathbb{R}^n$ given by

$$G(x) = \begin{cases} G_1(x) & \text{if } x \in D_1 \setminus (D_2 \cup D_3 \cup D_4 \cup D_5 \cup D_6) \\ G_2(x) & \text{if } x \in D_2 \setminus (D_1 \cup D_3 \cup D_4 \cup D_5 \cup D_6) \\ G_3(x) & \text{if } x \in D_3 \setminus (D_1 \cup D_2 \cup D_4 \cup D_5 \cup D_6) \\ G_4(x) & \text{if } x \in D_4 \setminus (D_1 \cup D_2 \cup D_3 \cup D_5 \cup D_6) \\ G_5(x) & \text{if } x \in D_5 \setminus (D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_6) \\ G_6(x) & \text{if } x \in D_6 \setminus (D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5) \end{cases}$$

where

$$G_1(x)=\begin{bmatrix}[\tau_i\ \tau_k]^\top\\ \begin{bmatrix}a_i & a_k\end{bmatrix}^\top\\ d\\ \begin{bmatrix}\begin{bmatrix}\tau_i & m_1^i & \dots & m_5^i\end{bmatrix} & m^k\end{bmatrix}^\top\\ p+1\\ 1\end{bmatrix} G_2(x)=\begin{bmatrix}[\tau_i\ \tau_k]^\top\\ \begin{bmatrix}a_i & a_k\end{bmatrix}^\top\\ c\\ \begin{bmatrix}m^i & \begin{bmatrix}\tau_k & m_1^i & \dots & m_5^i\end{bmatrix}\end{bmatrix}^\top\\ p+1\\ 0\end{bmatrix}$$

$$G_3(x)=\begin{bmatrix}[\tau_i\ \tau_k]^\top\\ \begin{bmatrix}a_i & a_k\end{bmatrix}^\top\\ d\\ \begin{bmatrix}m^i & \begin{bmatrix}\tau_k & m_1^k & \dots & m_5^k\end{bmatrix}\end{bmatrix}^\top\\ p+1\\ 1\end{bmatrix} G_4(x)=\begin{bmatrix}[\tau_i\ \tau_k]^\top\\ \begin{bmatrix}a_i & a_k\end{bmatrix}^\top\\ c\\ \begin{bmatrix}\begin{bmatrix}\tau_i & m_1^k & \dots & m_5^k\end{bmatrix} & m^k\end{bmatrix}^\top\\ p+1\\ 0\end{bmatrix}$$

$$G_5(x)=\begin{bmatrix}[\tau_i\ \tau_k]^\top\\ \begin{bmatrix}a_i & a_k\end{bmatrix}^\top\\ d\\ \begin{bmatrix}\begin{bmatrix}\tau_i & m_1^i & \dots & m_5^i\end{bmatrix} & m^k\end{bmatrix}^\top\\ p+1\\ 1\end{bmatrix} G_6(x)=\begin{bmatrix}[\tau_i\ \ \tau_k-u_{\tilde{o}}(x)]^\top\\ \begin{bmatrix}a_i & a_k+u_a(x)\end{bmatrix}^\top\\ c\\ \begin{bmatrix}m^i & \begin{bmatrix}\tau_k & m_1^i & \dots & m_5^i\end{bmatrix}\end{bmatrix}^\top\\ 0\\ 0\end{bmatrix}$$

with

$$u_{\tilde{o}}(x) = \frac{1}{2}(m_4^i - m_5^i - m_2^i + m_3^i) \tag{12}$$

and

$$u_a(x) = \mu\big((m_1^i - m_5^i) - (\tau_i - m_4^i)\big) \tag{13}$$

with $\mu > 0$. The offset correction $u_{\tilde{o}}$ in (12) is an adapted version of the offset correction algorithm given in (4) suitable for the hybrid system model. Each map in $G(x)$ corresponds to the message exchange events (P1)-(P6), i.e., G_1 is equivalent to the event (P1), G_2 is equivalent to the event (P2), etc. To trigger the jump map corresponding to the particular protocol event, we define the following set $D := D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5 \cup D_6$ where

$$D_1 := \{x \in \mathcal{X} : \tau{=}0, p{=}0\},\ \ D_2 := \{x \in \mathcal{X} : \tau{=}0, p{=}1\}$$
$$D_3 := \{x \in \mathcal{X} : \tau{=}0, p{=}2\},\ \ D_4 := \{x \in \mathcal{X} : \tau{=}0, p{=}3\}$$
$$D_5 := \{x \in \mathcal{X} : \tau{=}0, p{=}4\},\ \ D_6 := \{x \in \mathcal{X} : \tau{=}0, p{=}5\}$$

With the data defined, we let $\mathcal{H} = (C, F, D, G)$ denote the hybrid system for the pairwise broadcast synchronization algorithm between nodes i and k.

5.3 Error Model

In order to show that our proposed algorithm solves Problem 5.1, we recast the problem as a set stabilization problem. Namely, we show that solutions ϕ to $\mathcal{H}$ converge to a set of interest wherein the clock states τ_i, τ_k and clock rates a_i,a_k, respectively, coincide. To this end, we consider an augmented model of $\mathcal{H}$ in error coordinates to capture such a property. Let $\varepsilon := (\varepsilon_\tau, \varepsilon_a) \in \mathbb{R}^2$, where $\varepsilon_\tau = \tau_i - \tau_k$ and $\varepsilon_a = a_i - a_k$. Then, define

$$x_\varepsilon := (\varepsilon, x) \in \mathcal{X}_\varepsilon := \mathbb{R}^2 \times \mathcal{X}$$

For each $x_\varepsilon \in C_\varepsilon := \mathcal{X}_\varepsilon$, the flow map is given by

$$f_\varepsilon(x_\varepsilon) = (A_f\varepsilon, f(x))$$

where $A_f = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$. The discrete dynamics of the protocol are modeled through the jump map $G_\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ given by

$$G_\varepsilon(x_\varepsilon)=\begin{cases}G_{\varepsilon_1}(x_\varepsilon) \text{ if } x_\varepsilon \in D_{\varepsilon_1}\backslash(D_{\varepsilon_2}\cup D_{\varepsilon_3}\cup D_{\varepsilon_4}\cup D_{\varepsilon_5}\cup D_{\varepsilon_6})\\ G_{\varepsilon_2}(x_\varepsilon) \text{ if } x_\varepsilon \in D_{\varepsilon_2}\backslash(D_{\varepsilon_1}\cup D_{\varepsilon_3}\cup D_{\varepsilon_4}\cup D_{\varepsilon_5}\cup D_{\varepsilon_6})\\ G_{\varepsilon_3}(x_\varepsilon) \text{ if } x_\varepsilon \in D_{\varepsilon_3}\backslash(D_{\varepsilon_1}\cup D_{\varepsilon_2}\cup D_{\varepsilon_4}\cup D_{\varepsilon_5}\cup D_{\varepsilon_6})\\ G_{\varepsilon_4}(x_\varepsilon) \text{ if } x_\varepsilon \in D_{\varepsilon_4}\backslash(D_{\varepsilon_1}\cup D_{\varepsilon_2}\cup D_{\varepsilon_3}\cup D_{\varepsilon_5}\cup D_{\varepsilon_6})\\ G_{\varepsilon_5}(x_\varepsilon) \text{ if } x_\varepsilon \in D_{\varepsilon_5}\backslash(D_{\varepsilon_1}\cup D_{\varepsilon_2}\cup D_{\varepsilon_3}\cup D_{\varepsilon_4}\cup D_{\varepsilon_6})\\ G_{\varepsilon_6}(x_\varepsilon) \text{ if } x_\varepsilon \in D_{\varepsilon_6}\backslash(D_{\varepsilon_1}\cup D_{\varepsilon_2}\cup D_{\varepsilon_3}\cup D_{\varepsilon_4}\cup D_{\varepsilon_5})\end{cases}$$

where

$$G_{\varepsilon_1}(x_\varepsilon)=\begin{bmatrix}\varepsilon\\ G_1(x)\end{bmatrix},\ G_{\varepsilon_2}(x_\varepsilon)=\begin{bmatrix}\varepsilon\\ G_2(x)\end{bmatrix}$$
$$G_{\varepsilon_3}(x_\varepsilon)=\begin{bmatrix}\varepsilon\\ G_3(x)\end{bmatrix},\ G_{\varepsilon_4}(x_\varepsilon)=\begin{bmatrix}\varepsilon\\ G_4(x)\end{bmatrix}$$
$$G_{\varepsilon_5}(x_\varepsilon)=\begin{bmatrix}\varepsilon\\ G_5(x)\end{bmatrix},\ G_{\varepsilon_6}(x_\varepsilon)=\begin{bmatrix}\varepsilon+\begin{bmatrix}-u_{\tilde{o}}(x)\\ u_a(x)\end{bmatrix}\\ G_6(x)\end{bmatrix}$$

These discrete dynamics apply for x in $D_\varepsilon := D_{\varepsilon_1} \cup D_{\varepsilon_2} \cup D_{\varepsilon_3} \cup D_{\varepsilon_4} \cup D_{\varepsilon_5} \cup D_{\varepsilon_6}$ where

$$D_{\varepsilon_1}{:=}\{x_\varepsilon \in \mathcal{X}_\varepsilon : \tau{=}0, p{=}0\},\ \ D_{\varepsilon_2}{:=}\{x_\varepsilon \in \mathcal{X}_\varepsilon : \tau{=}0, p{=}1\}$$
$$D_{\varepsilon_3}{:=}\{x_\varepsilon \in \mathcal{X}_\varepsilon : \tau{=}0, p{=}2\},\ \ D_{\varepsilon_4}{:=}\{x_\varepsilon \in \mathcal{X}_\varepsilon : \tau{=}0, p{=}3\}$$
$$D_{\varepsilon_5}{:=}\{x_\varepsilon \in \mathcal{X}_\varepsilon : \tau{=}0, p{=}4\},\ \ D_{\varepsilon_6}{:=}\{x_\varepsilon \in \mathcal{X}_\varepsilon : \tau{=}0, p{=}5\}$$

This hybrid system is denoted $\mathcal{H}_\varepsilon = (C_\varepsilon, F_\varepsilon, D_\varepsilon, G_\varepsilon)$. [1]

Lemma 5.1. The hybrid system $\mathcal{H}_\varepsilon$ satisfies the hybrid basic conditions defined in (Goebel et al., 2012, Assumption 6.5).

Lemma 5.2. For every $\xi \in C_\varepsilon \cup D_\varepsilon = \mathcal{X}_\varepsilon$, there exists at least one nontrivial solution ϕ to $\mathcal{H}_\varepsilon$ such that $\phi(0, 0) = \xi$. Moreover, every maximal solution to $\mathcal{H}_\varepsilon$ is complete.

The set to render attractive so as to solve Problem 5.1 is given by

$$\mathcal{A}_\varepsilon := \{x_\varepsilon \in \mathcal{X}_\varepsilon : \varepsilon = 0\}$$

where $\varepsilon = 0$ implies synchronization of both the clock offset and clock rate.

5.4 Main Results

In this section, we present our main result showing asymptotic attractivity of the synchronization set $\mathcal{A}_\varepsilon$ for $\mathcal{H}_\varepsilon$. Consider the following Lyapunov function candidate

$$V(x_\varepsilon) = \varepsilon^\top e^{A_f^\top(\tau+d(5-p))} P e^{A_f(\tau+d(5-p))}\varepsilon$$

and positive scalars α_1 and α_2 such that

$$\alpha_1|x_\varepsilon|^2_{\mathcal{A}_\varepsilon} \le V(x_\varepsilon) \le \alpha_2|x_\varepsilon|^2_{\mathcal{A}_\varepsilon} \quad \forall x_\varepsilon \in C_\varepsilon \cup D_\varepsilon$$

Theorem 1. Let the hybrid system $\mathcal{H}_\varepsilon$ with constants $d = c > 0$ be given. If there exist a constant $\mu > 0$ and positive definite symmetric matrix P such that

$$A_g^\top e^{6cA_f^\top} P e^{6cA_f} A_g - P \prec 0 \tag{14}$$

is satisfied where $A_g = \begin{bmatrix}0 & \gamma_1\\ 0 & 1{-}\mu\gamma_2\end{bmatrix}$ with $\gamma_1 = \frac{7}{2}c$ and $\gamma_2 = 4c$, then $\mathcal{A}_\varepsilon$ is globally attractive for $\mathcal{H}_\varepsilon$.

To prove this result, we first show the existence of a forward invariant and finite time attractive set that enforces valid initialization values of the logic variables p, q and memory state vectors m^i and m^k such that the update laws $u_{\tilde{o}}$ and u_a give the input for the convergence of ε.

[1] The full state vector x to $\mathcal{H}$ is retained to facilitate the implementation of the synchronization algorithm for $\mathcal{H}_\varepsilon$.

We then show that for $x_\varepsilon \in C_\varepsilon$, V has the infinitesimal property of being constant during flows, namely

$$\langle \nabla V(x_\varepsilon), f_\varepsilon(x_\varepsilon) \rangle = 0$$

By continuity of the condition in (14), there exists $\sigma > 0$ such that, within the initialization, set V is constant or strictly decreasing during jumps. Namely for each $x_\varepsilon \in D_\varepsilon$,

$$V\big(G_\ell(x_\varepsilon)\big) - V\big(x_\varepsilon\big) \leq 0$$

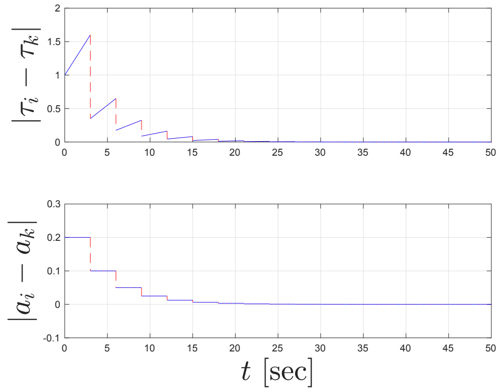
for each $\ell \in \{1, 2, 3, 4, 5\}$, and

$$V\big(G_6(x_\varepsilon)\big) - V\big(x_\varepsilon\big) \leq -\sigma \varepsilon^\top \varepsilon$$

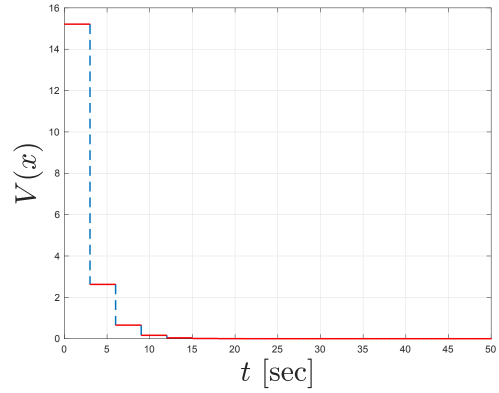
Then by picking a solution ϕ from the set of solutions to $\mathcal{H}_\varepsilon$ and evaluating V along the solution ϕ. We show that, following attractivity to the invariant initialization set, the infinitesimal properties of V gives that ϕ converges to $\mathcal{A}_\varepsilon$ thus, Problem 5.1 is solved. Due to space constraints, the complete proof has been omitted and will be published elsewhere.

5.5 Numerical Results

Consider Nodes i and k with dynamics as in (1) with data $a_i = 1$, $a_k = 0.8$ and $c = d = 0.5$. Setting $\mu = 0.25$, condition (14) is satisfied with $P = \begin{bmatrix} 5.429 & -0.134 \\ -0.134 & 35.010 \end{bmatrix}$. Simulating the system, Figure 5 shows the trajectories of the error in the clocks and error in the clock rates of Nodes i and k along with a plot of V evaluated along the solution. Notice, that V converges to zero asymptotically following several periodic executions of the algorithm. [2]



(a)



(b)

Fig. 5. Figure 5(a) gives the evolution of the error in the clocks and clock rates of Nodes i and k. Figure 5(b) gives $V(x_\varepsilon)$ evaluated along the solution.

6. CONCLUSION

In this paper, we introduced a sender-receiver clock synchronization algorithm with sufficient design conditions ensuring synchronization. Results were given to show asymptotic attractivity of a set of interest reflecting the desired synchronized setting. Numerical results validating the attractivity of the system to the set of interest were also given. In future work we will relax the condition $c = d$ and extend the framework for the analysis of other message-based clock synchronization schemes.

REFERENCES

Eidson, J.C. (2006). *Measurement, control, and communication using IEEE 1588*. Springer Science & Business Media.

Freris, N.M., Borkar, V.S., and Kumar, P. (2009). A model-based approach to clock synchronization. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 5744–5749. IEEE.

Freris, N.M., Graham, S.R., and Kumar, P. (2010). Fundamental limits on synchronizing clocks over networks. *IEEE Transactions on Automatic Control*, 56(6), 1352–1364.

Ganeriwal, S., Kumar, R., and Srivastava, M.B. (2003). Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, 138–149. ACM.

Goebel, R., Sanfelice, R.G., and Teel, A.R. (2012). *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press.

Graham, S. and Kumar, P.R. (2004). Time in general-purpose control systems: the control time protocol and an experimental evaluation. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, volume 4, 4004–4009 Vol.4. doi:10.1109/CDC.2004.1429378.

Guarro, M. and Sanfelice, R.G. (2020). HyNTP: An Adaptive Hybrid Network Time Protocol for Clock Synchronization in Heterogeneous Distributed Systems. In *American Control Conference*. Denver, United States.

IEEE (2008). IEEE standard for a precision clock synchronization protocol for networked measurement and control systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, 1–300.

Mills, D.L. (1991). Internet time synchronization: the network time protocol. *IEEE Transactions on communications*, 39(10), 1482–1493.

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

Wei Zhang, Branicky, M.S., and Phillips, S.M. (2001). Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1), 84–99. doi:10.1109/37.898794.

Wu, Y.C., Chaudhari, Q., and Serpedin, E. (2010). Clock synchronization of wireless sensor networks. *IEEE Signal Processing Magazine*, 28(1), 124–138.

[2] Code at github.com/HybridSystemsLab/HybridSenRecClockSync