2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Asynchronous sampled-data synchronization with small communications delays

Gal Barkai, Leonid Mirkin, and Daniel Zelazo

2024 IEEE 63rd Conference on Decision and Control (CDC) | 979-8-3503-1633-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CDC56724.2024.10886376

***Abstract*— This study investigates the state synchronization of linear time-invariant (LTI) agents within a networked environment characterized by intermittent and asynchronous communication, alongside heterogeneous time-varying transmission delays. These delays are not assumed to be known a-priori but only time-stamped. A hybrid controller, augmented with a special kind of predictor, is proposed to compensate for the delays and guarantee synchronization. Notably, synchronization is achieved under comparable conditions to the delay-free case, provided that transmission delays are smaller than the corresponding sampling interval. This is independent of the agents' dynamics and requires no additional knowledge of the underlying communication topology. An algorithm is presented for implementing the required predictor buffer with a size of one, offering a straightforward and scalable implementation.**

***Index Terms*— Sampled-data systems, network control systems, time delays.**

I. Introduction

The study of multi-agent systems (MAS) has been a central tenet in a multitude of scientific and engineering disciplines over the past decades, with applications including biology, optimization, and robotics. Perhaps the most widely known control problem in MAS is the consensus [1], [2] or synchronization [3], [4] problem, in which the agents must *agree* on some common steady-state trajectory. The crux of this problem is that the agents must reach agreement distributedly with limited communication, constraints which can be divided into two categories: spatial and temporal.

Spatially constrained communication reflects the requirement that each agent generate its control law based on partial information. Each agent is only privy to information from a subset of the group, its *neighbors* [5], [6]. These neighborhoods can be time-varying and induce a *graph* structure on the overall system, which has been widely exploited to design control laws [7]. While spatial constraints restrict the *identity* of interacting agents, temporal constraints restrict the communication *time*. These constraints encompass both sampled communication and various forms of time delays, both staples of networked systems.

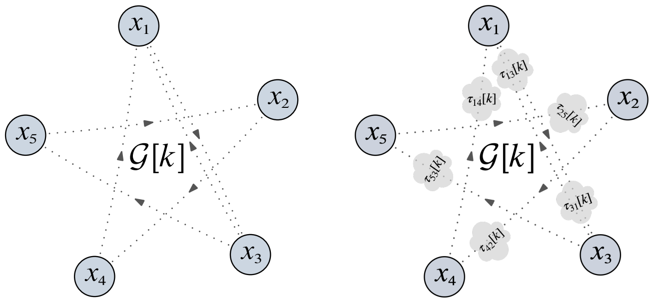
There exists substantial theory and a variety of methods to address each type of constraint individually. For example, spatial constraints have been extensively studied, with solutions proposed through graph and matrix theory [2], [8], while temporal constraints have also received attention within the context of networked control systems [9], [10]. However, in practice, systems must cope with both types of constraints. Moreover, common assumptions such as continuous communication, static graphs, or periodic synchronous sampling [11]–[13] might not be realistic. This cumulative burden makes even consensus of simple integrators challenging [14], often leading to conservative, robustness-based approaches [15], [16], even without delays. To the best of our knowledge, achieving synchronization under asynchronous communication and heterogeneous transmission delays is still an open problem.

We propose a sampled-data control protocol that guarantees asymptotic synchronization for switching graphs, asynchronous sampling, and time-varying transmission delays. The control structure is an extension of the emulation-based controller proposed in [17], with an added internal predictor. The only additional assumptions required are: i) the incoming information is time-stamped (i.e. each agent knows when incoming information was sent) and ii) that the time delays are shorter than the next sampling instance. This is made possible by leveraging the hybrid nature of the original controller, allowing for decoupled, continuous-time behavior of the agents between sampling instances.

This paper is organized as follows. In §II-A we set up the delay-free problem and its solution, and in §II-B we define the transmission delays and how they enter the system. Section III contains the main result of this note; §III-A motivates the structure of the predictor by solving the delayed problem for integrator consensus, while §III-B generalizes it to more general LTI agents. §III-C provides a simple algorithm to implement the predictor using a simple buffer of dimension 1. Section IV provides two numerical examples to illustrate the proposed controller, while Section V offers some concluding remarks.

Notation: The sets of all non-negative integers are denoted as $\mathbb{Z}_+$ and $\mathbb{N}_\nu := \{i \in \mathbb{Z} \mid 1 \le i \le \nu\}$. Sequences with indices from $\mathbb{Z}_+$ are indicated as $\{s_i\}$. The sets of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively, and $\mathbb{C}_0 := \{s \in \mathbb{C} \mid \operatorname{Re} s > 0\}$. The complex-conjugate transpose of a matrix M is denoted by M'. The image (range) and kernel (null space) of a matrix M are notated $\operatorname{Im} M$ and $\ker M$, respectively. Given two matrices (vectors) M and N, $M \otimes N$ denotes their Kronecker product, while $\operatorname{spec}(M)$ refers to the set of all eigenvalues of M. By $\mathbb{1}_\nu$, or simply $\mathbb{1}$ when the dimension is clear from the context, the all-ones vector from $\mathbb{R}^\nu$. Objects with the superscript $(\cdot)^{DF}$ refer to their delay-free counterparts.

Supported by the Israel Science Foundation (grants no. 3177/21 and 2285/20). GB and LM are with the Faculty of Mechanical Eng. and DŻ is with the Faculty of Aerospace Eng., all at the Technion—IIT, Haifa 3200003, Israel. E-mails: *galbarkai@campus.technion.ac.il*; {*mirkin, dzelazo*}*@technion.ac.il*.



(a) Standard. (b) With delays.

Fig. 1. Sampled-data multi-agent control architectures.

II. Problem setup and preliminaries

A. The delay-free problem and solution

This section outlines the delay-free problem and its solution as given in [17]. Consider ν homogeneous agents, each with linear dynamics given by

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \quad x_i(0) = x_{i,0}, \quad i \in \mathbb{N}_\nu \tag{1}$$

for some matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, where $x_i(t)$ and $u_i(t)$ are the ith state and control signal, respectively. The communication between the agents is restricted both spatially and temporally; however, the local state is assumed to be continuously available for each agent. The spatial constraints manifest as neighborhood sets, $\mathcal{N}_i(t) \subset \mathbb{N}_\nu \setminus \{i\}$, where each $\mathcal{N}_i(t)$ denotes the set of neighbors of agent i at time t. The temporal constraints are given by a strict monotonically increasing sequence of sampling instances $\{s_k\}$, $k \in \mathbb{Z}_+$. The agents may exchange information only at discrete instances $t = s_k$, and only with the neighboring agents. Hence, the combined communication constraints induce a switching graph structure, $\mathcal{G}[k]$, on the system. Note that $\mathcal{N}_i[k]$ can be empty for some sampling instances, and $\mathcal{G}[k]$ can be directed, which allows the setup to consider asynchronous sampling.

The objective is to design a controller which respects these communication constraints and asymptotically synchronizes the states of the agents to a common trajectory.

$\mathcal{P}_\mathbf{s}$: Given a matrix $A_0 \in \mathbb{R}^{n\times n}$ with $\operatorname{spec}(A_0) \in \mathbb{C} \setminus \mathbb{C}_0$, design control signals $u_i(t)$ satisfying the spatio-temporal constraints that ensure

$$\lim_{t\to\infty} \|x_i(t) - \mathrm{e}^{A_0 t} r_0\| = 0, \quad \forall i \in \mathbb{N}_\nu \tag{2}$$

for some constant $r_0 \in \mathbb{R}^n$ and all initial conditions of agents (1).

It should be emphasized that the matrix A_0 does not represent a leader node, but rather the shape of the required agreement trajectories. Globally, $\mathcal{P}_\mathbf{s}$ implies that not only must the aggregate state be driven to the *agreement set*, $\operatorname{Im} \mathbb{1}_\nu \otimes I_n$, but to a particular set of trajectories within it as determined by A_0.

We make the following two assumptions on the agent's dynamics, the desired trajectory, and the communication constraints.

$\mathcal{A}_\mathbf{1}$: The pair (A, B) is stabilizable, and there is $\bar{F}$ such that $A_0 = A + B\bar{F}$.

$\mathcal{A}_\mathbf{2}$: There is a strictly increasing sub-sequence of sampling indices $\{k_p\}$ such that for all $p \in \mathbb{Z}_+$ (i) the intervals $s_{k_{p+1}} - s_{k_p}$ are uniformly bounded, and (ii) $\cup_{k=k_p+1}^{k_{p+1}} \mathcal{G}[k]$ contains a directed rooted tree.

The following is a variant of the main result from [17].

Theorem 2.1: If $\mathcal{A}_\mathbf{1,2}$ hold and each agent can continuously measure its own state, then the following nth order local controllers

$$\begin{cases} \dot{z}_i(t) = (A + B\bar{F})z_i(t) + B(\bar{F}x_i(t) - u_i(t)), \quad z_i(0) = z_{i,0} \\ z_i(s_k^+) = z_i(s_k) - \dfrac{1}{\nu} \displaystyle\sum_{l\in\mathcal{N}_i[k]} (z_i(s_k) - z_l(s_k) + x_i(s_k) - x_l(s_k)) \\ u_i(t) = F_\mathrm{d} x_i(t) + \dfrac{1}{\nu}(\bar{F} - F_\mathrm{d})(z_i(t) + x_i(t)) \end{cases} \tag{3}$$

solves $\mathcal{P}_\mathbf{s}$ for all gains $F_\mathrm{d}, \bar{F}$ such that $A + B\bar{F} = A_0$ and $A_\mathrm{d} = A + BF_\mathrm{d}$ is Hurwitz. Moreover, $\lim_{t\to\infty} \|\bar{\mu}_i(t) - x_i(t)\| = 0$ for all $i \in \mathbb{N}_\nu$, where $\bar{\mu}_i := \frac{1}{\nu}(z_i + x_i)$.

Proof: By considering the coordinate transformation $\bar{\mu}_i = \frac{1}{\nu}(z_i + x_i)$ we obtain

$$\begin{cases} \dot{\bar{\mu}}_i(t) = (A + B\bar{F})\bar{\mu}_i(t), \quad \bar{\mu}_i(0) = \bar{\mu}_{i,0} \\ \bar{\mu}_i(s_k^+) = \bar{\mu}_i(s_k) - \dfrac{1}{\nu} \displaystyle\sum_{l\in\mathcal{N}_i[k]} (\bar{\mu}_i(s_k) - \bar{\mu}_l(s_k)) \\ u_i(t) = F_\mathrm{d} x_i(t) + (\bar{F} - F_\mathrm{d})\bar{\mu}_i(t) \end{cases} \tag{3'}$$

which is precisely the structure in [17, Corollary 4.4]. ■

For simplicity, we will use $\bar{\mu}_i$ and (3′) from here on, and the quantity the agents transmit at sampling instances is $\bar{\mu}_i(s_k)$. We further assume that the discrete component incorporating incoming information is *event driven* in the sense that updates occur whenever new information arrives, without explicit knowledge of $\{s_k\}$. Lastly, we shall use the fact that if $\bar{\mu}_i(t)$ synchronizes then so will $x_i(t)$ and we will focus the analysis and modification on the hybrid dynamics given by (3′). To this end, we require the following proposition.

Proposition 2.2: If the conditions of Theorem 2.1 hold, then

$$d\left(\bar{\mu}(s_{k_{p+1}}), \operatorname{Im} \mathbb{1}_\nu \otimes \mathrm{e}^{A_0 t}\right) < d\left(\bar{\mu}(s_{k_p}), \operatorname{Im} \mathbb{1}_\nu \otimes \mathrm{e}^{A_0 t}\right)$$

where $\bar{\mu}(s_k)$ is the aggregation of (3′) and $d(\cdot,\cdot)$ is the set (Hausdorff) distance.

Proof: The proof follows directly from applying the set-valued Lyapunov function of [18] to the aggregate solution of (3′) along the subsequence defined in $\mathcal{A}_\mathbf{2}$. ■

B. Introducing transmission delays

We now consider $\mathcal{P}_\mathbf{s}$ under the same setup and assumptions as before, but with heterogeneous time-varying delays on the *communicated* information between the agents. Denote by $\tau_{ij}[k]$ the transmission delay from agent j to agent i at time instance $t = s_k$. We assume that

$\mathcal{A}_3$: incoming information is time stamped and

$$s_k + \tau_{ij}[k] < s_{k+1}, \quad \forall i, j \in \mathbb{N}_\nu,\ k \in \mathbb{Z}_+.$$

The assumption above does not imply that the delays are known a priori, only that the receiving agent knows $\tau_{ij}[k]$ at $t = s_k + \tau_{ij}[k]$. The second part guarantees that there is no packet disorder, which is a reasonable assumption in MAS [13] and networked systems in general [10]. Note that the delays are allowed to vary between sampling instances as well as across communication channels.

In the sequel, we shall propose a modified version of the controller (3′) to solve $\mathcal{P}_\mathbf{s}$ for all transmission delays satisfying $\mathcal{A}_3$.

III. Dead time compensation

Since the agents interact only at discrete time instances and are decoupled otherwise, the delays modify only the discrete component of (3′). Essentially, $\mathcal{A}_3$ splits the delay-free update of agent i at s_k into up to $|\mathcal{N}_i[k]|$ different updates spread over the interval $[s_k, s_{k+1})$ but still verifying

$$\bigcup_{j \in \mathcal{N}_i[k]} \mathcal{N}_i[t_{ij}[k]] = \mathcal{N}_i^{DF}[k]\ \forall i \in \mathbb{N}_\nu,\ k \in \mathbb{Z}_+ \qquad (4)$$

where $t_{ij}[k] := s_k + \tau_{ij}[k]$ and $\mathcal{N}_i^{DF}[k]$ denotes the delay-free neighborhood of agent i.

Remark 3.1 (Packet loss): Note that (4) does not preclude the possibility of packet losses, but rather relegates them to the graphs induced by $\{s_k\}$. ▿

We now need to design an update rule

$$\bar{\mu}_i(t_{ij}[k]^+) = \kappa\left(\alpha_i \bar{\mu}_i(t_{ij}[k]) + \sum_{j \in \mathcal{N}_i[t_{ij}[k]]} \alpha_j \bar{\mu}_j(s_k)\right)$$

for some function $\kappa(\cdot)$. The problem at hand is qualitatively different from the standard delay problems considered in the literature. Continuous-time delays are infinite dimensional systems, and therefore so are predictors used in delay compensation. In the proposed setup, the delays affect continuous information that is sent intermittently and used to update $\bar{\mu}_i$ in a discrete fashion. In discrete time, delays are finite dimensional and occur at discrete steps synchronized with the regular increments of the system. However, $\mathcal{A}_3$ implies that the delayed information arrives and is processed *before* the next global sampling instance. Hence, the delay at hand does not fit into either of the standard descriptions. To understand how to construct a predictor for this hybrid type of delay, we shall first consider the special case of consensus of integrator agents.

A. Consensus of integrator agents

Consider the special case of first order integrator agents trying to achieve consensus. This corresponds to $\mathcal{P}_\mathbf{s}$ with $A = A_0 = 0$, and (3′) simplifies to

$$\begin{cases} \bar{\mu}_i(s_k^+) = \bar{\mu}_i(s_k) - \dfrac{1}{\nu} \displaystyle\sum_{l \in \mathcal{N}_i[k]} (\bar{\mu}_i(s_k) - \bar{\mu}_l(s_k)) \\ u_i(t) = F_\mathrm{d}(x_i(t) - \bar{\mu}_i(s_k^+)) \end{cases}.$$

The equation above is a generalization of the control law proposed in [19, §III.A], for which the dynamics of $\bar{\mu}_i$ are purely discrete. This significantly simplifies the analysis and, in fact, makes any predictor redundant, as demonstrated in the following proposition.

Proposition 3.1: Consider $\mathcal{P}_\mathbf{s}$ with $A = A_0 = 0$ and transmission delays. The control law

$$\begin{cases} \bar{\mu}_i(t_{ij}[k]^+) = \bar{\mu}_i(t_{ij}[k]) - \dfrac{1}{\nu} \displaystyle\sum_{l \in \mathcal{N}_i[t_{ij}[k]]} (\bar{\mu}_i(s_k) - \bar{\mu}_l(s_k)) \\ u_i(t) = F_\mathrm{d}(x_i(t) - \bar{\mu}_i(t_{ij}[k]^+)) \end{cases} \qquad (5)$$

will drive the agents asymptotically to consensus for all sampling sequences $\{s_k\}$ satisfying $\mathcal{A}_2$, all time delays $\tau_{ij}[k]$ satisfying $\mathcal{A}_3$, and all gains $F_\mathrm{d} < 0$.

Proof: Consider an ordered sequence $\{q_l[k]\}$ where each element is defined by

$$\begin{aligned} q_1[k] &= \min_{ij} t_{ij}[k] \\ q_l[k] &= \min_{ij} \{t_{ij}[k]\} \setminus \{q_j[k] : j < l\} \\ l &= 2, 3, \ldots, |\mathcal{N}_i[k]|, \end{aligned} \qquad (6)$$

i.e., the ordered time instances for the interval $[s_k, s_{k+1}]$ in which information arrives. By $\mathcal{A}_3$, $\{q_j[k]\}$ has a finite (possibly different) number of elements for each k. Assume without loss of generality that $q_p[k]$ is the last instance, since $\bar{\mu}_i$ is discrete this implies that $\bar{\mu}_i(s_{k+1}) = \bar{\mu}_i(q_p[k]^+)$. Expanding the above we have

$$\begin{aligned} \bar{\mu}_i(s_{k+1}) &= \bar{\mu}_i(q_p[k]) - \frac{1}{\nu} \sum_{l \in \mathcal{N}_i[q_p[k]]} (\bar{\mu}_i(s_k) - \bar{\mu}_l(s_k)) \\ &= \bar{\mu}_i(s_k) - \frac{1}{\nu} \sum_{r=1}^{p} \sum_{l \in \mathcal{N}_i[q_r[k]]} (\bar{\mu}_i(s_k) - \bar{\mu}_l(s_k)). \end{aligned}$$

By $\mathcal{A}_3$ and (4), we know that

$$\bigcup_{r=1}^{p} \mathcal{N}_i[q_r[k]] = \mathcal{N}_i^{DF}[k],$$

hence

$$\bar{\mu}_i(s_{k+1}) = \bar{\mu}_i^{DF}(s_{k+1}).$$

This is true for all $i \in \mathbb{N}_\nu$ and $k \in \mathbb{Z}_+$. Therefore, if (3′) will drive the delay-free system to consensus, (5) will as well. Note that for $A = A_0 = 0$, $\mathcal{A}_1$ trivially holds, and that if $\mathcal{A}_2$ holds for $\{s_k\}$ then it will also hold for the shifted sequence $\{s_{k+1}\}$. Thus, we can conclude that if the delay-free system will reach agreement for the sequence $\{s_k\}$ and its induced graphs, then the delayed system will for $\{s_{k+1}\}$. ■

Proposition 3.1 illustrates how the hybrid nature of the delay can render it redundant in certain cases. Since the updates are event-drive, i.e., an update occurs when new information arrives, and $\bar{\mu}_i$ is constant between updates, the transmission delays only amount to splitting one update into several smaller ones within the same time interval. When combined with $\mathcal{A}_2$, which considers the union of the induced graphs over some subsequence, it is evident that the delays amount to a partition of the interval $[s_k, s_{k+1}]$ for which

$\bigcup \mathcal{G}[t_{ij}[k]] = \mathcal{G}[k]$. Hence, from a consensus standpoint, there is no difference between the original problem and the delayed one. As such, there's no need to predict anything, simply to guarantee that the original sampling intervals remain disjoint, as required in $\mathcal{A}_3$.

The above reasoning does not hold when $\bar{\mu}_i(t)$ is no longer constant between updates, as in the general case of $\mathcal{P}_s$. However, this insight is the guiding principle in designing an appropriate predictor as will be done in the following section.

B. Synchronization of LTI agents

The key property exploited in §III-A was that the value of $\bar{\mu}(s_{k+1})$ was the same as it would have been in the delay-free case for all k. In the following lemma, we propose an update rule that will guarantee this property for arbitrary A and A_0 satisfying $\mathcal{A}_1$.

Lemma 3.2: If $\mathcal{A}_{1,3}$ hold and $A + B\bar{F} = A_0$, then under the update rule

$$\bar{\mu}_i(t_{ij}[k]^+) = \bar{\mu}_i(t_{ij}[k]) - \frac{1}{\nu} e^{A_0 \tau_{ij}[k]} \sum_{j \in \mathcal{N}_i[t_{ij}[k]]} (\bar{\mu}_i(s_k) - \bar{\mu}_j(s_k)), \tag{7}$$

we recover the same $\bar{\mu}_i(s_{k+1})$ as in the delay-free system (3′).

Proof: Consider the ordered sequence $\{q_l[k]\}$ from (6) for an arbitrary agent with index i, and assume that it receives p delayed updates in the interval $[s_k, s_{k+1})$. Define

$$\theta_i[k,l] := \sum_{j \in \mathcal{N}_i[q_l[k]]} (\bar{\mu}_i(s_k) - \bar{\mu}_j(s_k)),$$

to simplify the notation, we shall omit the argument k when it is clear from context or unimportant. Now consider $k = l = 1$, for which the update reads

$$\begin{aligned} \bar{\mu}_i(q_1^+) &= e^{A_0 q_1} \bar{\mu}_{i,0} - \frac{1}{\nu} e^{A_0 \tau_1} \theta[1,1] \\ &= e^{A_0 \tau_1} \left(\bar{\mu}(s_1) - \frac{1}{\nu} \theta[1,1] \right), \end{aligned}$$

where we used the general fact that $e^{A_0(q_{l+1} - q_l)} e^{A_0 \tau_l} = e^{A_0 \tau_{l+1}}$. From here, by induction

$$\bar{\mu}_i(q_p^+) = e^{A_0 \tau_p} \left(\bar{\mu}(s_1) - \frac{1}{\nu} \sum_{l=1}^{p} \theta[1,l] \right),$$

and once more applying $\mathcal{A}_3$ and (4) we obtain that

$$\bar{\mu}_i(q_p^+[1]) = e^{A_0 \tau_p} \bar{\mu}_i^{DF}(s_1^+) = \bar{\mu}_i^{DF}(q_p[1]).$$

Now consider an arbitrary k and $l = 1$, for which the update reads

$$\begin{aligned} \bar{\mu}_i(q_1^+[k]) &= e^{A_0(q_1[k] - q_p[k-1])} \bar{\mu}_i(q_p[k-1]^+) \\ &\quad - \frac{1}{\nu} e^{A_0 \tau_1[k]} \theta[k,1] \\ &= e^{A_0 \tau_1[k]} \left(\bar{\mu}(s_k) - \frac{1}{\nu} \theta[k,1] \right), \end{aligned}$$

where we used the fact that

$$\begin{aligned} e^{A_0(q_1[k] - q_p[k-1])} \bar{\mu}_i(q_p[k-1]^+) &= \\ e^{A_0 \tau_1[k]} e^{A_0(s_k - q_p[k-1])} \bar{\mu}_i(q_p[k-1]^+) &= \\ e^{A_0 \tau_1[k]} \bar{\mu}_i(s_k)\ . \end{aligned}$$

From here, by similar arguments, we can conclude that

$$\bar{\mu}_i(q_p^+[k]) = e^{A_0 \tau_p} \bar{\mu}_i^{DF}(s_k^+) = \bar{\mu}_i^{DF}(q_p[k]),$$

since there are no updates between $q_p[k]$ and s_{k+1} and the choice of i was arbitrary, the system evolves like its delay-free counterpart (3′). ■

One can view the Lemma 3.2 from a different angle. Consider the aggregation $\bar{\mu}(t) = \left[\ \bar{\mu}_1(t)', \ldots, \bar{\mu}_\nu(t)'\ \right]'$, then (7) in aggregate form is given by

$$\begin{aligned} \bar{\mu}(t_{ij}[k]^+) &= \bar{\mu}(t_{ij}[k]) - \frac{1}{\nu} \left(\mathcal{L}[t_{ij}[k]] \otimes e^{A_0 \tau_{ij}[k]} \right) \bar{\mu}(s_k) \\ &= \left((I_\nu - \frac{1}{\nu} \mathcal{L}[t_{ij}[k]]) \otimes I_n \right) \bar{\mu}(t_{ij}[k]), \end{aligned}$$

which is exactly the delay-free update rule for the sampling sequence $\{t_{ij}[k]\}$ instead of $\{s_k\}$. The predictor can be thought of as inducing a new sequence of graphs and sampling instances, whose union over the interval $[s_k, s_{k+1}]$ results in the same induced graph as the original sampling sequence and delay-free update mechanism. This is the key step in the proof of the main result.

Theorem 3.3: If assumptions $\mathcal{A}_{1,2}$ hold and $\bar{F}, F_d$ are chosen such that $A_0 = A + B\bar{F}$ and $A + BF_d$ is Hurwitz, then the controller

$$\begin{cases} \dot{\bar{\mu}}_i(t) = (A + B\bar{F}) \bar{\mu}_i(t), \quad \bar{\mu}_i(0) = \bar{\mu}_{i,0} \\ \bar{\mu}_i(t_{ij}[k]^+) = \bar{\mu}_i(t_{ij}[k]) - \frac{1}{\nu} e^{A_0 \tau_{ij}[k]} \sum_{l \in \mathcal{N}_i[t_{ij}[k]]} (\bar{\mu}_i(s_k) - \bar{\mu}_l(s_k)) \\ u_i(t) = F_d x_i(t) + (\bar{F} - F_d) \bar{\mu}_i(t) \end{cases} \tag{8}$$

solves $\mathcal{P}_s$ for all heterogeneous and time-varying transmission delays satisfying $\mathcal{A}_3$.

Proof: Consider first the aggregate delay-free state on the sequence $\{s_k\}$, denoted by $\bar{\mu}^{DF}(s_k)$. If $\mathcal{A}_2$ holds, we know by Theorem 2.1 that $\bar{\mu}^{DF}(s_k) \to \operatorname{Im} \mathbb{1}_\nu \otimes I_n$, and by Proposition 2.2 that it gets closer to that set along the sequence $\{k_p\}$ from $\mathcal{A}_2$. Applying Lemma 3.2 we know that $\bar{\mu}_i(s_{k+1}) = \bar{\mu}_i^{DF}(s_{k+1})$ for all i and all k; hence, both of them approach $\operatorname{Im} \mathbb{1}_\nu \otimes I_n$ at the same rate. Since the agreement set is an invariant set of both the continuous and discrete dynamics of (8), this implies that

$$\lim_{t \to \infty} \|\bar{\mu}_i^{DF}(t) - \bar{\mu}_i(t)\| = 0, \quad \forall i \in \mathbb{N}_\nu.$$

Applying Theorem 2.1 implies that

$$\lim_{t \to \infty} \|\bar{\mu}_i(t) - x_i(t)\| = 0,$$

hand ence the states synchronize. ■

C. Implementability

Local controllers (8) are independent of the size of the system; however, update rule (7) makes use of $\bar{\mu}_i(s_k)$. Since the sequence $\{s_k\}$ is not assumed to be known, an immediate question arises as to whether the control law can be implemented. The following proposition states that it can be using a small buffer, and details how to update this buffer accordingly.

Proposition 3.4: The update law (7) can be implemented using a buffer of size 1.

Proof: Each agent constructs its buffer as follows. Let $\left[\, b_i' \;\; t_i' \,\right]'$ denote the values of the ith buffer and corresponding timestamp, and denote by $t_i[k]$ the instance at which information is received and by s_k the time when it was sent.

1) If $t_i[k] = s_k$, assign

$$\begin{bmatrix} b_i \\ t_i \end{bmatrix} = \begin{bmatrix} \bar{\mu}_i(s_k) \\ s_k \end{bmatrix}.$$

2) If $s_k < t_i[k]$, check
   a) If $t_i = s_k$, keep the current buffer.
   b) If $t_i < s_k$ assign

$$\begin{bmatrix} b_i \\ t_i \end{bmatrix} = \begin{bmatrix} \mathrm{e}^{A_0(s_k - t_i[k])}\bar{\mu}_i(t_i[k]) \\ s_k \end{bmatrix}.$$

From $\mathcal{A}_3$ we know that if $t_i = s_k$ then we are still in the interval (s_k, s_{k+1}); hence, we need to keep the start of the interval in the buffer. Similarly, if $t_i < s_k$, this means that our buffer corresponds to the previous interval. Thus, there were no jumps in $[s_k, t_i[k]]$ and we can reconstruct $\bar{\mu}_i(s_k)$ like we would for a regular LTI system. ■

IV. Illustrative Example

To illustrate the proposed sampled-data protocol, consider two cases, both comprised of $\nu = 3$ identical agents. We assume that communication between agents is intermittent and asynchronous, meaning that each agent transmits only at a subset of sampling instances. At each sampling instance $\mathcal{G}[k]$ is a union of any nonempty combination of the three graphs in Fig. 2. The sampling instances, shown by abscissa ticks on the bottom, are a random variable such that $s_{k+1} - s_k \in 0.3\,\mathbb{N}_6$, and the induced graphs satisfy $\mathcal{A}_2$. Major ticks indicate instances where agent 1 transmits information, i.e. corresponding to $\mathcal{G}_1$ in Fig. 2. The simulations were carried out with a time step of $\Delta t = 1\times10^{-3}$ on the time interval $t \in [0, 24]$. For each sampling interval, $h_k := s_{k+1} - s_k$, a random integer m_k was drawn uniformly from the interval $[1, h_k/\Delta t]$, generating the delay $\tau_{ij}[k] = m_k\Delta t$, thus satisfying $\mathcal{A}_3$. The major ticks at the top and corresponding dashed lines correspond to the delayed updates originating from agent 1 to agent 2. Both examples are simulated for the same delays, sampling sequence, and time interval.

The first simulation involves integrator agents as described in §III-A with $F_\mathrm{d} = -5$. The agent's states can be seen in Fig. 3(a), while the difference $\Delta_{\mu,i}(t) := \bar{\mu}_i(t) - \bar{\mu}_i^{DF}(t)$ is shown in Fig. 3(b). It can be seen that indeed the agents asymptotically agree, and that $\Delta_{\mu,i}(t)$ repeatedly resets to zero after each agent finishes its "cycle" of delayed updates.

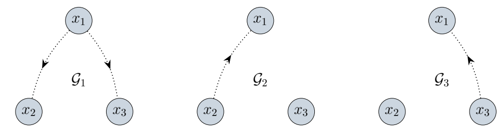
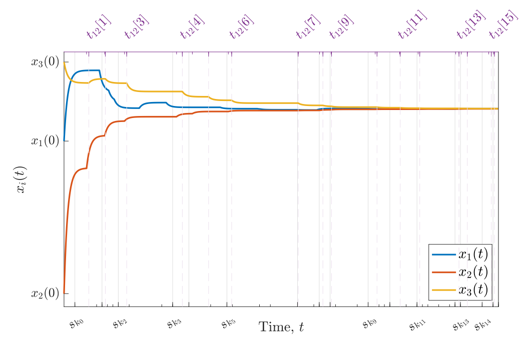
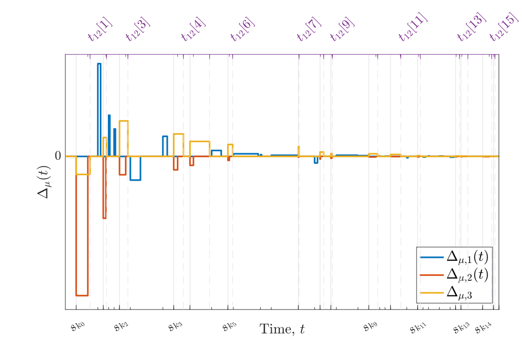


Fig. 2. The three possible graphs.

Moreover, the trajectories are piecewise constant for this case since $\bar{\mu}_i(t)$ has no continuous-time dynamics as mentioned in the proof of Proposition 3.1. The second example is



(a) Agents states.



(b) Difference between $\bar{\mu}$ and its delay-free counterpart.

Fig. 3. Simulations for the example with $A = A_0 = 0$.

comprised of identical agents with

$$\dot{x}_i(t) = \begin{bmatrix} 4 & 9 \\ 1 & 4 \end{bmatrix} x_i(t) + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u_i(t)$$

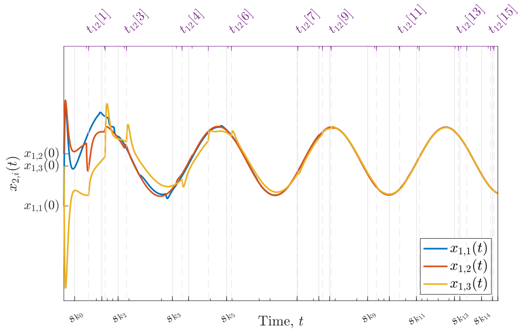
trying to synchronize to $A_0 = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$. In this case

$$\bar{F} = -\begin{bmatrix} 2 & 4 \end{bmatrix} \quad \text{and} \quad F_\mathrm{d} = \begin{bmatrix} -34.6 & 39.2 \end{bmatrix}$$

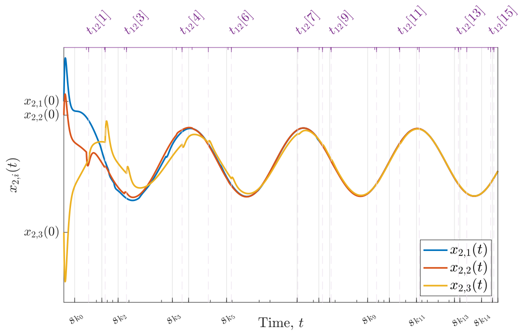
satisfy the requirements of Theorem 3.3. The components of the agents' state are shown in Fig. 4, and those of $\Delta_{\mu,i}$ are shown in Fig. 5. Once more, we can see that the agents' states synchronize to a common trajectory as in $\mathcal{P}_\mathbf{s}$ with A_0 corresponding to a sine wave with frequency 1. Furthermore, we again see that the difference between the delayed and delay-free system resets repeatedly after each "cycle" ends, and that the amplitude of the mismatch decays as the updates drive the systems closer to the agreement space. Note that this time $\bar{\mu}_i(t)$ is not piecewise constant between updates, since the synchronous trajectory is not constant.

V. Concluding Remarks

In this note, we have addressed the synchronization of LTI agents to a common time-varying trajectory while utilizing

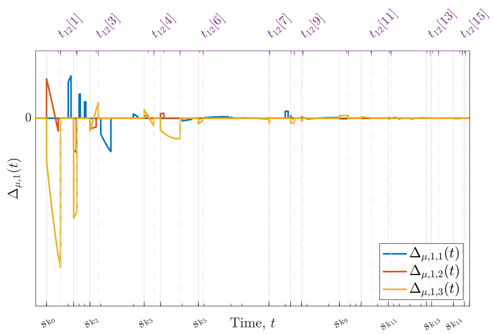


(a) Agents first state component.

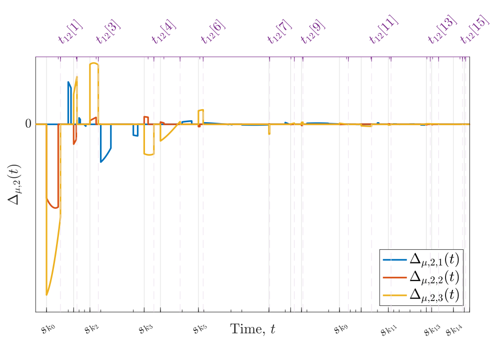


(b) Agents second state component.

Fig. 4. The state components for the second example.



(a) The firs components of $\Delta_{\mu,i}$.



(b) The second component of $\Delta_{\mu,i}$.

Fig. 5. The components of $\Delta_{\mu,i}$ for the second example.

asynchronous communication afflicted by heterogeneous and time-varying transmission delays. By augmenting a hybrid controller with a special kind of predictor, we were able to achieve synchronization under identical conditions to the delay-free case, contingent upon transmission delays remaining smaller than the corresponding sampling interval. The proposed predictor does not require a-priori knowledge of the delays or their rate of change, only that they are time-stamped. Moreover, its design is transparent, intuitive, and independent of the delay-free control law. The predictor can be implemented easily using a buffer of size one without knowledge of the sampling times, making the method implementable and scalable. The simplicity of the predictor leaves room for many possible extensions and adjustments. Current research focuses on extending the results to output measurements, noisy measurements, and delays with uncertainty.

References

[1] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[2] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. London: Springer-Verlag, 2008.

[3] L. Scardovi and R. Sepulchre, "Synchronization in networks of identical linear systems," *Automatica*, vol. 45, no. 11, pp. 2557–2562, 2009.

[4] P. Wieland, R. Sepulchre, and F. Allgöwer, "An internal model principle is necessary and sufficient for linear output synchronization," *Automatica*, vol. 47, no. 5, pp. 1068–1074, 2011.

[5] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[6] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Automat. Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[7] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.

[8] F. Bullo, *Lectures on Network Systems*, 1st ed. Kindle Direct Publishing, 2022. [Online]. Available: http://motion.me.ucsb.edu/book-lns

[9] E. Garcia, P. J. Antsaklis, and L. A. Montestruque, *Model-Based Control of Networked Systems*. Boston: Birkhäuser, 2014.

[10] A. Goldenshluger and L. Mirkin, "H_2 control under intermittent sampling and small communication delays," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 583–588, 2019.

[11] H. Chu, D. Yue, C. Dou, X. Xie, and L. Chu, "Consensus of multi-agent systems with time-varying input delay via truncated predictor feedback," *IEEE Trans. Syst. Man, and Cybernetics: Systems*, vol. 51, no. 10, pp. 6062–6073, 2021.

[12] W. Jiang, Y. Chen, and T. Charalambous, "Consensus of general linear multi-agent systems with heterogeneous input and communication delays," *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 851–856, 2021.

[13] M. Xing, F. Deng, and Z. Hu, "Sampled-data consensus for multiagent systems with time delays and packet losses," *IEEE Trans. Syst. Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 203–210, 2020.

[14] X. Ge, Q. Han, D. Ding, X. Zhang, and B. Ning, "A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems," *Neurocomputing*, vol. 275, pp. 1684–1701, 2018.

[15] Y. Wu, H. Su, P. Shi, Z. Shu, and Z.-G. Wu, "Consensus of multiagent systems using aperiodic sampled-data control," *IEEE Transactions on Cybernetics*, vol. 46, no. 9, pp. 2132–2143, 2016.

[16] J. Sun, Z.-S. Wang, and N.-N. Rong, "Sampled-data consensus of multiagent systems with switching jointly connected topologies via time-varying lyapunov function approach," *Int. J. Robust and Nonlinear Control*, vol. 30, no. 14, pp. 5369–5385, 2020.

[17] G. Barkai, L. Mirkin, and D. Zelazo, "An emulation approach to sampled-data synchronization," in *Proc. 62nd IEEE Conf. Decision and Control*, 2023, pp. 6449–6454.

[18] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Automat. Control*, vol. 50, no. 2, pp. 169–182, 2005.

[19] G. Barkai, L. Mirkin, and D. Zelazo, "On sampled-data consensus: Divide and concur," *IEEE Control Syst. Lett.*, vol. 6, pp. 343–348, 2022.