

Distributed Periodic Event-Triggered Control of Nonlinear Multi-Agent Systems

Citation for published version (APA):

Scheres, K. J. A., Dolk, V. S., Chong, M. S., Postoyan, R., & Heemels, W. P. M. H. (2022). Distributed Periodic Event-Triggered Control of Nonlinear Multi-Agent Systems. *IFAC-PapersOnLine*, 55(13), 168-173.
<https://doi.org/10.1016/j.ifacol.2022.07.254>

Document license:

CC BY-NC-ND

DOI:

[10.1016/j.ifacol.2022.07.254](https://doi.org/10.1016/j.ifacol.2022.07.254)

Document status and date:

Published: 01/01/2022

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Distributed Periodic Event-Triggered Control of Nonlinear Multi-Agent Systems^{*}

Koen J.A. Scheres^{*} Victor S. Dolk^{*} Michelle S. Chong^{*} Romain Postoyan^{**}
W.P. Maurice H. Heemels^{*}

^{*} Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands. (k.j.a.scheres@tue.nl).

^{**} Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France.

Abstract: We present a general emulation-based framework to address the distributed control of multi-agent systems over packet-based networks. We consider the setup where information is only transmitted at (non-uniform) sampling times and where packets are received with unknown delays. We design local dynamic periodic event-triggering mechanisms to generate the transmissions. The triggering mechanisms can run on non-synchronized digital platforms, i.e., we ensure that the conditions must only be verified at asynchronous sampling times, which may differ for each platform. Different stability and performance characteristics can be considered as we follow a general dissipativity-based approach. Moreover, Zeno-free properties are guaranteed by design. The results are illustrated on a consensus problem.

Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Networked systems, Control over networks, Multi-agent systems, Distributed nonlinear control, Event-based control, Event-triggered and self-triggered control

1. INTRODUCTION

Distributed and multi-agent control systems, including the consensus problem, have attracted a lot of attention in recent years. When these systems communicate via a packet-based network, information is sparsely available. In these cases, event-triggered control can be used. Event-triggered control consists of generating the input signal and updating it over the packet-based network at some time instants, which are based on the available plant/agent information, to guarantee relevant stability and performance properties, see, e.g., Heemels et al. (2012). A triggering condition is thus synthesized and monitored to decide whether a new transmission is needed. Due to the fact that these conditions are often processed on a digital platform, it is essential to take the sampling behavior of the platform into account, especially when fast sampling is not possible, e.g. in case of low-power electronics, in which case we talk of periodic event-triggered control, see, e.g., Heemels et al. (2013). Moreover, in practice, the communication network suffers from imperfections such as time-varying and unknown delays, which may destroy stability of the closed-loop system. While there is abundant literature on the event-triggered control of multi-agent systems, see, e.g., Nowzari et al. (2019) and references therein, to the best of our knowledge, none of the proposed approaches in the literature consider *all* of the following aspects:

- (1) nonlinear multi-system setup,
- (2) fully distributed and asynchronous configuration,
- (3) implementability on digital platforms,
- (4) unknown and time-varying sampling and transmission delays,
- (5) general stability and performance properties for (possibly unbounded) attractors (as in consensus).

Many works treat a subset of these aspects. A prominent example is, e.g., the recent work by Yu and Chen (2021), which provides important advancements on the nonlinear case with (large) unknown transmission delays while tak-

ing sampling into account. The main focus of our paper is providing a unifying framework addressing all of these aspects.

The main contribution in this paper is the development of a unified framework for the design of Zeno-free, decentralized and asynchronous periodic event-triggering mechanisms that can be implemented on local digital platforms. The setup proposed in this paper captures a wide range of relevant multi-agent (but also centralized) control problems by adopting a general dissipativity-based framework. Using this framework, we can consider distributed stabilization of nonlinear systems, output-regulation problems (of which the consensus-seeking problem is a particular case) and vehicle-platooning problems (in which \mathcal{L}_p -contractivity, $p \in [1, \infty)$, is of interest as a string stability guarantee). A notable advantage of our setup is that clock synchronization is not required. Hence each local platform can sample and transmit independently of all other platforms, making the algorithm fully distributed.

2. PRELIMINARIES

For space reasons, some preliminaries are omitted. An extended version of this paper including all notations that are used can be found on arXiv, see Scheres et al. (2022).

We consider hybrid systems of the form $\mathcal{H}(\mathcal{C}, F, \mathcal{D}, G)$ given by $\dot{\xi} = F(\xi, v)$, when $\xi \in \mathcal{C} \subseteq \mathbb{X}$, and $\xi^+ \in G(\xi)$, when $\xi \in \mathcal{D} \subseteq \mathbb{X}$, where $v \in \mathcal{L}_{\infty}^{n_v}$ is an exogenous input (disturbance) and $\mathbb{X} \subseteq \mathbb{R}^{n_{\xi}}$ with $n_{\xi} \in \mathbb{N}_{>0}$, see Goebel et al. (2012) and Heemels et al. (2021). For these hybrid systems, we are interested in the following dissipativity property, which is close in nature to the one used in Teel (2010).

Definition 1. Let $s : \mathbb{X} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}$ be a *supply rate* and $\mathcal{S} \subseteq \mathbb{X}$ be a closed non-empty set. System \mathcal{H} is said to be *s-flow-dissipative* with respect to \mathcal{S} , or in short, *(s, S)-flow-dissipative*, if there exists a locally Lipschitz function $U : \mathbb{X} \rightarrow \mathbb{R}_{\geq 0}$, called a *storage function*, such that

^{*} This work is supported by the ANR grant HANDY 18-CE40-0010.

- (1) there exist \mathcal{K}_∞ -functions $\underline{\alpha}, \bar{\alpha}$ such that for all $\xi \in \mathbb{X}$, $\underline{\alpha}(|\xi|_{\mathcal{S}}) \leq U(\xi) \leq \bar{\alpha}(|\xi|_{\mathcal{S}})$, where $|\xi|_{\mathcal{S}}$ denotes the distance of ξ to the set \mathcal{S} ,
- (2) for all $\xi \in \mathcal{C}$, $v \in \mathbb{R}^{n_v}$ and $f \in F(\xi, v)$, $U^\circ(\xi; f) \leq s(\xi, v)$, where U° denotes the generalized directional derivative of Clarke,
- (3) for all $\xi \in \mathcal{D}$ and all $g \in G(\xi)$, $U(g) - U(\xi) \leq 0$. \square

3. SYSTEM SETUP

3.1 Multi-agent systems

We consider the setting where multiple agents, each with a local digital platform, communicate with each other via a packet-based network to achieve a common goal such as stabilization, consensus, \mathcal{L}_p -performance, etc., which will be captured by a dissipativity property as in Definition 1, as explained further below. To be precise, we consider a collection of $N \in \mathbb{N}_{>0}$ heterogeneous agents $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_N$, which are interconnected according to a digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V} := \{1, 2, \dots, N\}$. The digital platform of each agent is used for the implementation of algorithms and control computations. Due to the digital nature, measurements are not continuously available, but only on specific sampling times, as explained in more detail in Section 3.3. The dynamics of the i^{th} agent, $i \in \mathcal{N} := \{1, 2, \dots, N\}$, are given by

$$\mathcal{A}_i : \begin{cases} \dot{x}_i = f_i(x, \hat{y}_i^{\text{in}}, v_i), \\ y_i = h_i(x_i), \end{cases} \quad (1)$$

where $x_i \in \mathbb{R}^{n_{x,i}}$ is the local state vector, $x := (x_1, x_2, \dots, x_N) \in \mathbb{R}^{n_x}$ with $n_x := \sum_{i \in \mathcal{N}} n_{x,i}$ is the global state vector, $v_i \in \mathbb{R}^{n_{v,i}}$ is a local exogenous disturbance or input, $y_i \in \mathbb{R}^{n_{y,i}}$ is the local output, $y := (y_1, y_2, \dots, y_N) \in \mathbb{R}^{n_y}$ with $n_y := \sum_{i \in \mathcal{N}} n_{y,i}$ is the global output and $\hat{y}_i^{\text{in}} \in \mathbb{R}^{n_y}$ is agent \mathcal{A}_i 's estimate of the outputs of agents \mathcal{A}_m , $m \in \mathcal{V}_i^{\text{in}}$, where $\mathcal{V}_i^{\text{in}}$ denotes the collection of all agents that transmit information to agent \mathcal{A}_i , as will be explained in more detail in Section 3.2. We assume that the functions f_i are continuous and that the functions h_i are continuously differentiable. The maps f_i may depend on the entire vector x , implying that we can allow physical couplings between agents, for example in an interconnected physical plant, see Fig. 1 for an illustration.

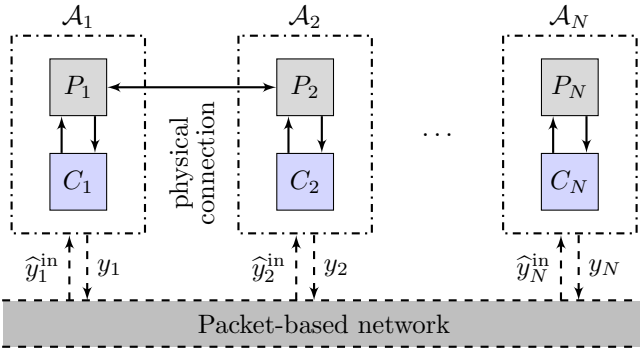


Fig. 1. Example of a networked control setup with several agents connected to a packet-based network and a physical connection between \mathcal{A}_1 and \mathcal{A}_2 .

Typical systems that can be captured by a multi-agent system are, e.g., (physically uncoupled) agents, a fleet of vehicles seeking consensus, or a distributed plant (with physical couplings) with distributed controllers. To design the controllers, we proceed with an emulation-based approach in which a (local) controller is designed such that, in the absence of a network, i.e., when $\hat{y}_i^{\text{in}} = y$ for all $i \in \mathcal{V}_i^{\text{in}}$, the system satisfies the desired stability and performance criteria. Since the controller is implemented

on a digital platform, we assume that either the controller is static and updated during the sampling times of the output (see Section 3.3), or, when the (local) controllers are dynamic, that they run on a separate platform, whose sampling times are much faster and hence they operate in (near) continuous-time.

3.2 Transmitted outputs

Due to the usage of a packet-based network, agent \mathcal{A}_i , $i \in \mathcal{N}$, does not have continuous access to the output y_m of agent \mathcal{A}_m , $m \in \mathcal{V}_i^{\text{in}}$. Instead, agent \mathcal{A}_i only has an estimate \hat{y}_m^i of y_m , $m \in \mathcal{V}_i^{\text{in}}$, collected in the estimation vector $\hat{y}_i^{\text{in}} := (\hat{y}_1^i, \hat{y}_2^i, \dots, \hat{y}_N^i) \in \mathbb{R}^{n_y}$. For all $m \in \mathcal{N} \setminus \mathcal{V}_i^{\text{in}}$, \hat{y}_m^i is not relevant and simply set to zero.

At discrete times t_k^i , $k \in \mathbb{N}$, $i \in \mathcal{N}$, that satisfy $0 = t_0^i < t_1^i < \dots$, the output y_i of agent \mathcal{A}_i is broadcasted over the network to all (connected) agents \mathcal{A}_m , where $m \in \mathcal{V}_i^{\text{out}}$ with $\mathcal{V}_i^{\text{out}}$ the collection of all agents that receive information from agent \mathcal{A}_i . Due to possible network delays, the estimate \hat{y}_i^m , $m \in \mathcal{V}_i^{\text{out}}$, is updated after a delay of $\Delta_k^{i,m} \geq 0$. Note that the delays are different for each (receiving) agent. The update of the estimate \hat{y}_i^m , $i \in \mathcal{N}$, $m \in \mathcal{V}_i^{\text{out}}$, can be expressed as

$$\hat{y}_i^m((t_k^i + \Delta_k^{i,m})^+) = y_i(t_k^i). \quad (2)$$

In between transmissions, the estimate \hat{y}_i^m is generated by a zero-order-hold (ZOH) device, i.e.,

$$\dot{\hat{y}}_i^m(t) = 0, \quad (3)$$

for all $t \in (t_k^i + \Delta_k^{i,m}, t_{k+1}^i + \Delta_{k+1}^{i,m})$, with $i \in \mathcal{N}$, $m \in \mathcal{V}_i^{\text{out}}$, $k \in \mathbb{N}$.

The transmission times t_k^i are determined by an *event generator* or *triggering mechanism*. Inspired by Girard (2015) and Dolk et al. (2017), we consider dynamic event triggering rules, where an auxiliary variable $\eta_i \in \mathbb{R}_{\geq 0}$, $i \in \mathcal{N}$, whose dynamics are designed in the sequel, is used to determine the transmission times t_k^i , $k \in \mathbb{N}$, see Section 3.4.

3.3 Digital platform

The triggering mechanism of each agent is implemented on the local digital platform, which has its own sampling times. The sequence of sampling times of agent \mathcal{A}_i is denoted $\{s_n^i\}_{n \in \mathbb{N}}$, where s_n^i denotes the n^{th} local sampling instant of agent \mathcal{A}_i . Transmissions generated by \mathcal{A}_i occur on a subset of the sampling instants, i.e.,

$$\{t_k^i\}_{k \in \mathbb{N}} \subseteq \{s_n^i\}_{n \in \mathbb{N}}. \quad (4)$$

Inspired by Wang et al. (2020), we consider the general setting where the inter-sampling times satisfy

$$0 < d_i \leq s_{n+1}^i - s_n^i \leq \tau_{\text{MASP}}^i, \quad (5)$$

where d_i is an arbitrarily small but positive constant and τ_{MASP}^i denotes the *maximum allowable sampling period* (MASP) for agent \mathcal{A}_i , $i \in \mathcal{N}$. The sampling times $\{s_n^i\}_{n \in \mathbb{N}}$ and $\{s_n^j\}_{n \in \mathbb{N}}$ of agents \mathcal{A}_i and \mathcal{A}_j , respectively, are a priori not related for $i \neq j$. In other words, all agents operate *independently* and *asynchronously*.

Due to the agents operating asynchronously, the arrival times $t_k^i + \Delta_k^{i,m}$, $k \in \mathbb{N}$, of new information at agent \mathcal{A}_m from agent \mathcal{A}_i may not coincide with the sampling times $\{s_n^m\}_{n \in \mathbb{N}}$ of agent \mathcal{A}_m , hence information may be received

in between consecutive sampling times of agent \mathcal{A}_m . However, the sampling-induced delay (the time between the arrival of information from agent \mathcal{A}_i and the next sampling instant of agent \mathcal{A}_m) can be included in the total delay denoted $\bar{\Delta}_k^{i,m}$. Therefore, the total delay $\bar{\Delta}_k^{i,m}$ is equal to the combined communication delay $\Delta_k^{i,m}$ and sampling-induced delay. Through this setup, we obtain

$$\{t_k^i + \bar{\Delta}_k^{i,m}\}_{k \in \mathbb{N}} \subseteq \{s_n^m\}_{n \in \mathbb{N}} \quad (6)$$

for all $m \in \mathcal{N}$ and $i \in \mathcal{V}_m^{\text{out}}$.

We adopt the following assumption on the total delays $\bar{\Delta}_k^{i,m}$, $k \in \mathbb{N}$.

Assumption 1. For each $i \in \mathcal{N}$, there is a time-constant τ_{MAD}^i such that the delays are bounded according to $0 \leq \bar{\Delta}_k^{i,m} \leq \tau_{\text{MAD}}^i \leq t_{k+1}^i - t_k^i$ for all $m \in \mathcal{V}_i^{\text{out}}$ and all $k \in \mathbb{N}$, where τ_{MAD}^i denotes the *maximum allowable delay* (MAD) for agent \mathcal{A}_i . \square

Assumption 1 is a “small delay” condition, which also implies that packets sent from \mathcal{A}_i to \mathcal{A}_m , $m \in \mathcal{V}_i^{\text{out}}$, are received in the same order that they are transmitted.

Since the sampling-induced delays are never larger than the local MASP τ_{MASP}^m at agent m , we have that

$$\tau_{\text{MAD}}^i \geq \tau_{\text{MASP}}^m + \Delta_k^{i,m} \text{ for all } i \in \mathcal{N}, m \in \mathcal{V}_i^{\text{out}}, k \in \mathbb{N}. \quad (7)$$

3.4 Triggering rule

Our goal is to employ dynamic event triggering, which relies on locally available information, namely output measurements. Due to this information only being available at the sampling instants $\{s_n^i\}_{n \in \mathbb{N}}$, the design of, e.g., Dolk et al. (2017) cannot be directly used. Instead, we consider an event-triggering mechanism (ETM) in the form

$$t_{k+1}^i := \inf\{t \geq t_k^i + \tau_{\text{MIET}}^i \mid \eta_i(t) + \nu_i(y_i(t), \hat{y}_i^{\text{out}}(t), \tau_i(t)) \leq 0, t \in \{s_n^i\}_{n \in \mathbb{N}}\}, \quad (8)$$

for $i \in \mathcal{N}$, $k \in \mathbb{N}$, with $t_0^i = 0$ and where $\tau_{\text{MIET}}^i > 0$ denotes the (enforced lower bound on the) *minimum inter-event time* (MIET) of agent \mathcal{A}_i , $\eta_i \in \mathbb{R}_{\geq 0}$ is the auxiliary variable mentioned earlier, $\hat{y}_i^{\text{out}} := (\hat{y}_i^1, \hat{y}_i^2, \dots, \hat{y}_i^N)$ is the vector of estimates of the output y_i at the agents \mathcal{A}_m , $m \in \mathcal{V}_i^{\text{out}}$. Variable $\tau_i \in \mathbb{R}_{\geq 0}$ is a local timer that is set to zero after each transmission of the output y_i over the network, and whose dynamics are given by $\dot{\tau}_i = 1$ in between two successive transmissions of agent \mathcal{A}_i . The function $\nu_i : \mathbb{R}^{n_{y,i}} \times \mathbb{R}^{N n_{y,i}} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\leq 0}$ is to be designed.

At first glance it might seem unnatural that agent \mathcal{A}_i has to know the estimates \hat{y}_i^{out} due to the presence of the unknown and time-varying delays. However, this information is only needed when $\tau_i \geq \tau_{\text{MIET}}^i$, and since $\tau_{\text{MIET}}^i \geq \tau_{\text{MAD}}^i$ as we will see in Section 5, all agents \mathcal{A}_m , $m \in \mathcal{V}_i^{\text{out}}$, will have received the latest transmission of agent \mathcal{A}_i for $\tau_i \geq \tau_{\text{MIET}}^i$. Due to the fact that a ZOH is used to generate the estimated output, see (3), \hat{y}_i^{out} is thus the vector consisting of the last value of the output y_i sent by agent \mathcal{A}_i , which is available at agent \mathcal{A}_i for $\tau_i \geq \tau_{\text{MIET}}^i$.

The ETM (8) satisfies the constraints that arise from the usage of a digital platform, as the trigger condition in (8) only has to be evaluated at the local sampling times s_n^i , $n \in \mathbb{N}$. The triggering variable η_i generated locally by agent \mathcal{A}_i , $i \in \mathcal{N}$, evolves according to

$$\dot{\eta}_i = \Psi_i(\hat{y}_i^{\text{in}}) - \varphi_i(\eta_i), \quad (9a)$$

$$\eta_i(t^+) \in \begin{cases} \{\eta_i + \varrho_i(y_i, \hat{y}_i^{\text{out}})\}, & \text{for all } t \in \{t_k^i\}_{k \in \mathbb{N}}, \\ \{\eta_i + \nu_i(y_i, \hat{y}_i^{\text{out}}, \tau_i)\}, & \\ \text{for all } t \in \{s_n^i\}_{n \in \mathbb{N}} \setminus \{t_k^i\}_{k \in \mathbb{N}}, \end{cases} \quad (9b)$$

where the functions $\Psi_i : \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $\varrho_i : \mathbb{R}^{n_{y,i}} \times \mathbb{R}^{N n_{y,i}} \rightarrow \mathbb{R}_{\geq 0}$, $\varphi_i \in \mathcal{K}_{\infty}$ and the constant $\tau_{\text{MIET}}^i \in \mathbb{R}_{> 0}$ are designed in Section 5.3.

Remark 1. In (9a), a continuous-time differential equation is used. However, since the ‘external’ variable \hat{y}_i^{in} is constant in between consecutive sampling times, exact discretization or numerical integration can be used to compute η_i a posteriori based on the elapsed time since the previous sampling time. For instance, if $\varphi_i(\eta_i) = \alpha_i \eta_i$ with a constant $\alpha_i \neq 0$, we obtain the exact discretization $\eta_i(s_{n+1}^i) = e^{-\alpha_i(s_{n+1}^i - s_n^i)} \eta_i((s_n^i)^+) + \alpha_i^{-1} [1 - e^{-\alpha_i(s_{n+1}^i - s_n^i)}] \Psi_i(\hat{y}_i^{\text{in}}((s_n^i)^+))$. Hence, exact solutions to the differential equation can be obtained on a digital platform. However, we consider the dynamics of η_i as presented in (9) to facilitate the modeling and stability analysis later on. \square

3.5 Objective

Given the descriptions above, the problem considered in this paper can now be stated informally as follows. Consider a collection of maximum allowable delays τ_{MAD}^i , $i \in \mathcal{N}$, satisfying Assumption 1. Our objective is to propose design conditions for the time constants $\tau_{\text{MIET}}^i (\geq \tau_{\text{MAD}}^i)$, the functions Ψ_i , φ_i , ϱ_i and ν_i , $i \in \mathcal{N}$, as in (8) and (9), such that the resulting system has the desired (and to be specified) closed-loop stability, performance and robustness properties formalized in terms of suitable dissipativity properties.

4. HYBRID MODELING

To facilitate the modeling of the overall networked system, some helpful notation is introduced in this section.

4.1 Network-induced errors

For all $i \in \mathcal{N}$ and $m \in \mathcal{V}_i^{\text{out}}$, we denote the *network-induced error* e_i^m as the difference between the output y_i of agent \mathcal{A}_i and the estimate \hat{y}_i^m of the output y_i at agent \mathcal{A}_m . For all $i \in \mathcal{N}$ and $m \in \mathcal{N} \setminus \mathcal{V}_i^{\text{out}}$, i.e., for all redundant variables, we set $e_i^m = 0$. Hence, we have

$$e_i^m := \begin{cases} \hat{y}_i^m - y_i, & \text{if } m \in \mathcal{V}_i^{\text{out}}, \\ 0, & \text{if } m \in \mathcal{N} \setminus \mathcal{V}_i^{\text{out}}. \end{cases} \quad (10)$$

We define two separate concatenations of the network-induced error associated to agent \mathcal{A}_i , $i \in \mathcal{N}$. The first one, denoted $e_i^{\text{out}} := (e_i^1, e_i^2, \dots, e_i^N) \in \bar{\mathbb{E}}_i$, where $\bar{\mathbb{E}}_i := \mathbb{E}_i(1) \times \mathbb{E}_i(2) \times \dots \times \mathbb{E}_i(N)$ and with

$$\mathbb{E}_i(m) := \begin{cases} \mathbb{R}^{n_{y,i}}, & \text{if } m \in \mathcal{V}_i^{\text{out}}, \\ \{\mathbf{0}_{n_{y,i}}\}, & \text{otherwise,} \end{cases}$$

is the concatenation of the network-induced errors associated to the output y_i . The second, denoted $e_i^{\text{in}} := (e_i^1, e_i^2, \dots, e_i^N) \in \mathbb{E}_i$, with $\mathbb{E}_i := \mathbb{E}_i(1) \times \mathbb{E}_2(i) \times \dots \times \mathbb{E}_N(i)$, is the concatenation of network-induced errors of the estimated outputs available at agent \mathcal{A}_i , $i \in \mathcal{N}$. Moreover, we define the concatenation of all network-induced errors e_i^{in} , for $i \in \mathcal{N}$, as $e := (e_1^{\text{in}}, e_2^{\text{in}}, \dots, e_N^{\text{in}}) \in \mathbb{E}$ with $\mathbb{E} := \mathbb{E}_1 \times \mathbb{E}_2 \times \dots \times \mathbb{E}_N$. Observe that $|e|^2 = \sum_{i \in \mathcal{N}} |e_i^{\text{out}}|^2 = \sum_{i \in \mathcal{N}} |e_i^{\text{in}}|^2$.

4.2 Clock variables

To be able to cast the overall system described in Section 3 in the form of $\mathcal{H}(\mathcal{C}, F, \mathcal{D}, G)$, we need to introduce some auxiliary variables. Each agent \mathcal{A}_i , $i \in \mathcal{N}$, has two local timers. We already saw that τ_i captures the time elapsed since the last transmission of agent \mathcal{A}_i , see (8). The second, denoted σ_i , keeps track of the time elapsed since the last sampling instant of agent \mathcal{A}_i , i.e., $\dot{\sigma}_i(t) = 1$ for all $t \in \mathbb{R} \setminus \{s_n^i\}_{n \in \mathbb{N}}$ and is reset to zero at each sampling instant, i.e., $\sigma_i(t^+) = 0$ for all $t \in \{s_n^i\}_{n \in \mathbb{N}}$. Observe that τ_i takes values in $\mathbb{R}_{\geq 0}$ and that σ_i takes values in $\mathbb{T}_i := [0, \tau_{\text{MASP}}^i]$ due to (5). Their concatenations are defined as $\tau := (\tau_1, \tau_2, \dots, \tau_N) \in \mathbb{R}_{\geq 0}^N$ and $\sigma := (\sigma_1, \sigma_2, \dots, \sigma_N) \in \mathbb{T}$ with $\mathbb{T} := \mathbb{T}_1 \times \mathbb{T}_2 \times \dots \times \mathbb{T}_N$.

4.3 Indicator variables

We also define two indicator variables, $\ell_i^m \in \{0, 1\}$ and $b_i^m \in \{0, 1\}$. The variable ℓ_i^m is used to indicate whether the most recently transmitted output value y_i of agent \mathcal{A}_i has been received by agent \mathcal{A}_m ($\ell_i^m = 0$), or that it still has to be received by agent \mathcal{A}_m ($\ell_i^m = 1$). Since information received by agent \mathcal{A}_m is *processed* at the sampling times, we assume that information is buffered if it is received between sampling instances. The variable b_i^m indicates whether agent \mathcal{A}_m will process (i.e., update \hat{y}_i^m) the most recently transmitted output value by \mathcal{A}_i ($b_i^m = 1$) or that \hat{y}_i^m will not be updated at its next sampling instance ($b_i^m = 0$). We distinguish between these two “events” to ensure that updates of \hat{y}_i^m align with the sampling times of agent \mathcal{A}_m , as described in (6). A graphical representation of ℓ_i^m and b_i^m is drawn in Fig. 2.

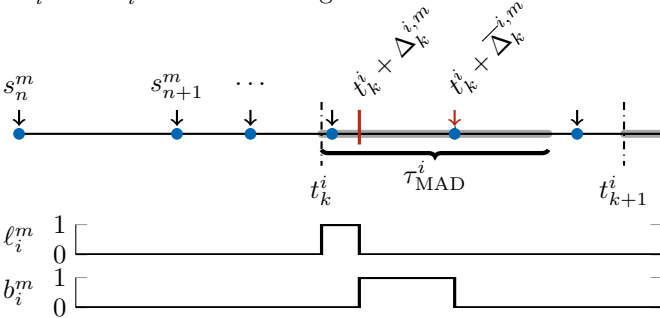


Fig. 2. Graphical representation of the indicator variables ℓ_i^m and b_i^m . Blue dots indicate s_k^m , $k \in \mathbb{N}$.

Observe in Fig. 2, for all $t \in [t_k^i, t_k^i + \bar{\Delta}_k^{i,m}]$, $\ell_i^m(t) + b_i^m(t) = 1$. The sum of b_i^m and ℓ_i^m indicates whether the most recently transmitted information of agent \mathcal{A}_i has been received and processed by agent \mathcal{A}_m ($\ell_i^m + b_i^m = 0$) or that it still has to be received or processed by agent \mathcal{A}_m ($\ell_i^m + b_i^m = 1$). Moreover, due to Assumption 1, both ℓ_i^m and b_i^m are guaranteed to be zero for all $\tau \geq \tau_{\text{MAD}}^i$. To elucidate, we have that $\ell_i^m + b_i^m \in \{0, 1\}$ for all $t \in \mathbb{R}_{\geq 0}$. These variables are concatenated as $\ell := (\ell_1^1, \ell_2^1, \dots, \ell_N^1, \ell_1^2, \ell_2^2, \dots, \ell_N^N) \in \{0, 1\}^{N^2}$ and $b := (b_1^1, b_2^1, \dots, b_N^1, b_1^2, b_2^2, \dots, b_N^N) \in \{0, 1\}^{N^2}$.

Additionally, we define for all $i \in \mathcal{N}$, the memory variable $r_i \in \mathbb{R}^{n_{y,i}}$ that stores the value of the y_i at the transmission times t_k^i , $k \in \mathbb{N}$. Hence, r_i has dynamics $\dot{r}_i = 0$ and $r_i^+ = y_i$ if \mathcal{A}_i transmits its output value. The concatenation is denoted $r := (r_1, r_2, \dots, r_N) \in \mathbb{R}^{n_y}$.

Based on the above, we can write the update of the local error e_i^{in} during the sampling times $\{s_n^i\}_{n \in \mathbb{N}}$ of agent \mathcal{A}_i

as $e_i^{\text{in}}((s_n^i)^+) = e_i^{\text{in}}(s_n^i) + \text{diag}(b_i^{\text{in}}(s_n^i))((r(s_n^i) - y(s_n^i)) - e_i^{\text{in}}(s_n^i))$, where $b_i^{\text{in}} := (b_1^i, b_2^i, \dots, b_N^i)$.

4.4 Overall system

Using these definitions, the networked system (1) can be cast into the form of a hybrid system $\mathcal{H}(\mathcal{C}, F, \mathcal{D}, G)$ with state $\xi := (x, e, \tau, \sigma, r, \ell, b, \eta) \in \mathbb{X}$, where $\eta := (\eta_1, \eta_2, \dots, \eta_N) \in \mathbb{R}_{\geq 0}^N$ and $\mathbb{X} := \{(x, e, \tau, \sigma, r, \ell, b, \eta) \in \mathbb{R}^{n_x} \times \mathbb{E} \times \mathbb{R}_{\geq 0}^N \times \mathbb{T} \times \mathbb{R}^{n_y} \times \{0, 1\}^{N^2} \times \{0, 1\}^{N^2} \times \mathbb{R}_{\geq 0}^N \mid \forall i, m \in \mathcal{N}, ((\ell_i^m + b_i^m = 0) \vee (\ell_i^m + b_i^m = 1 \wedge \tau_i \in [0, \tau_{\text{MAD}}^i]))\}$.

For space reasons, we omit the exact expressions of \mathcal{C} , F , \mathcal{D} and G , but we point out that the model can be constructed such that a jump is enforced when $\ell_i^m = 1 \vee b_i^m = 1$ and $\tau_i = \tau_{\text{MAD}}^i$, or when $\sigma_i = \tau_{\text{MASP}}^i$. As such, the hybrid model complies with Assumption 1 and with (5). Details of the exact expressions for \mathcal{C} , F , \mathcal{D} and G can be found in the full version of this paper, see Scheres et al. (2022).

4.5 Formal problem statement

We can now state the problem of Section 3.5 formally as follows.

Problem 1. Given the system $\mathcal{H}(\mathcal{C}, F, \mathcal{D}, G)$, provide design conditions for the time-constants $\tau_{\text{MAD}}^i, \tau_{\text{MIET}}^i \in \mathbb{R}_{>0}$ with $\tau_{\text{MIET}}^i \geq \tau_{\text{MAD}}^i$ and the functions Ψ_i , ς_i , ϱ_i and ν_i as in (8) and (9), for $i \in \mathcal{N}$, such that, under Assumption 1, the system \mathcal{H} is persistently flowing¹ and (\tilde{s}, \mathcal{S}) -flow-dissipative for a set $\mathcal{S} \subset \mathbb{X}$, for a given supply rate $\tilde{s} : \mathbb{X} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}$ of the form

$$\tilde{s}(\xi, v) := s(x, e, v) - \varphi(\eta), \quad (11)$$

where $\xi \in \mathbb{X}$, $v \in \mathbb{R}^{n_v}$ and $\varphi := (\varphi_1(\eta_1), \varphi_2(\eta_2), \dots, \varphi_N(\eta_N))$ with φ_i as in (9). \square

As shown in, for example, Van der Schaft (2017); Teel (2010), the use of dissipativity allows the consideration of various important system properties such as asymptotic stability, input-to-state stability, \mathcal{L}_p -stability with $p \in [1, \infty)$ and passivity, from a unified point of view. Thus, the supply rate \tilde{s} and the set \mathcal{S} capture the desired stability, performance and robustness requirements.

5. DESIGN CONDITIONS

To ensure that the hybrid system has the desired performance and stability properties, the following conditions have to be satisfied.

5.1 Growth of the network-induced error

We require that the dynamics of the network-induced error satisfy the following property.

Condition 1. For each $i \in \mathcal{N}$, there exist functions $H_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}_{\geq 0}$ and constants $L_i \geq 0$ such that for all $m \in \mathcal{V}_i^{\text{out}}$, $x \in \mathbb{R}^{n_x}$, $e \in \mathbb{R}^{n_{y,i}}$ and $v \in \mathbb{R}^{n_v}$,

$$|f_{y,i}(x, e, v)| \leq H_i(x, e_i^{\text{in}}, v) + L_i |e_i^i|, \quad (12)$$

where $f_{y,i}(x, e, v) = \frac{\partial h_i(x_i)}{\partial x_i} f_i(x, h_i^{\text{in}}(x) + e_i^{\text{in}}, v)$ with $\delta_i(m)$ given by $\delta_i(m) = 1$ when $m \in \mathcal{V}_i^{\text{out}}$ and $\delta_i(m) = 0$ otherwise, and $h_i^{\text{in}}(x) := (\delta_1(i)h_1(x_1), \delta_2(i)h_2(x_2), \dots, \delta_N(i)h_N(x_N))$ with $h_i(x_i)$ in (1). \square

¹ Persistently flowing in the sense that maximal solutions have an unbounded domain in t -direction, see Goebel et al. (2012).

Inequality (12) is related to \dot{y}_i , which, due to the use of ZOH devices, is directly related to \dot{e}_i^m , as $\dot{e}_i^m = -\delta_i(m)\dot{y}_i$. In essence, Condition 1 is a restriction on the growth of the network-induced error between transmissions. This condition is naturally satisfied for linear systems or when the vector fields f_i are globally bounded and h_i are globally Lipschitz.

5.2 Lower-bounds on the Minimum Inter-event Times and Maximum Allowable Delays

To obtain lower-bounds on the minimum inter-event times τ_{MIET}^i and the maximum allowable delay τ_{MAD}^i for each agent \mathcal{A}_i , $i \in \mathcal{N}$, we first characterize the influence of the transmission errors e_i^{in} on the state x and the desired stability/performance property by means of the following condition.

Condition 2. There exist a locally Lipschitz function $V : \mathbb{R}^{n_x} \rightarrow \mathbb{R}_{\geq 0}$ and a non-empty closed set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, \mathcal{K}_{∞} -functions $\underline{\alpha}_V \leq \bar{\alpha}_V$, continuous functions $\varsigma_i : \mathbb{R}^{N n_y} \rightarrow \mathbb{R}_{\geq 0}$, constants $\mu_i, \gamma_i > 0$, $i \in \mathcal{N}$, such that for all $x \in \mathbb{R}^{n_x}$

$$\underline{\alpha}_V(|x|_{\mathcal{X}}) \leq V(x) \leq \bar{\alpha}_V(|x|_{\mathcal{X}}), \quad (13)$$

and for all $y \in \mathbb{R}^{n_y}$, $e \in \mathbb{R}^{N n_y}$, $v \in \mathbb{R}^{n_v}$, and almost all $x \in \mathbb{R}^{n_x}$

$$\langle \nabla V(x), f(x, e, v) \rangle \leq s(x, e, v) - \sum_{i \in \mathcal{N}} \varsigma_i(\hat{y}_i^{\text{in}}) + \sum_{i \in \mathcal{N}} (-\mu_i N_i H_i^2(x, e_i^{\text{in}}, v) + \gamma_i^2 |e_i^{\text{out}}|^2) \quad (14)$$

with N_i the cardinality of $\mathcal{V}_i^{\text{out}}$, and H_i from (12). \square

Condition 2 constitutes an \mathcal{L}_2 -gain condition from $|e_i^{\text{out}}|$ to H_i . In case of a linear system, this condition can always be verified if the system is controllable, for instance. In the absence of a network, i.e., when $e = 0$, (13)-(14) imply an (s, \mathcal{X}) -dissipativity property. However, this property is affected by the network-induced error e , and our objective is to design the triggering mechanisms such that the dissipativity property still holds for the networked system.

The constants γ_i as in Condition 2 are used to determine τ_{MIET}^i and τ_{MAD}^i , $i \in \mathcal{N}$, via the following condition.

Condition 3. Select $\tau_{\text{max}}^i > 0$ and $\tau_{\text{MAD}}^i > 0$, $i \in \mathcal{N}$, with $\tau_{\text{max}}^i \geq \tau_{\text{MAD}}^i + \tau_{\text{MASP}}^i$ such that

$$\tilde{\gamma}_i(0)\phi_{0,i}(\tau_{\text{max}}^i) \geq \lambda_i^2 \tilde{\gamma}_i(1)\phi_{1,i}(0), \quad (15)$$

$$\tilde{\gamma}_i(1)\phi_{1,i}(\tau_i) \geq \tilde{\gamma}_i(0)\phi_{0,i}(\tau_i), \text{ for all } \tau_i \in [0, \tau_{\text{MAD}}^i], \quad (16)$$

where $\phi_{l,i}$, $l \in \{0, 1\}$, evolves according to

$$\frac{d}{d\tau_i} \phi_{l,i} = - \left(2\tilde{L}_i(l)\phi_{l,i} + \tilde{\gamma}_i(l) \left(\frac{1}{\mu_i \epsilon_i} \phi_{l,i}^2 + 1 \right) \right), \quad (17)$$

for some fixed initial conditions $\phi_{l,i}(0)$, $l \in \{0, 1\}$, that satisfy $\tilde{\gamma}_i(1)\phi_{1,i}(0) \geq \tilde{\gamma}_i(0)\phi_{0,i}(0) > \lambda_i^2 \tilde{\gamma}_i(1)\phi_{1,i}(0) > 0$, where, for each $i \in \mathcal{N}$ and $l \in \{0, 1\}$, the functions $\tilde{L}_i : \{0, 1\} \rightarrow \mathbb{R}_{\geq 0}$ and $\tilde{\gamma}_i : \{0, 1\} \rightarrow \mathbb{R}_{\geq 0}$ are given by

$$\tilde{L}_i(l) := \lambda_i^{-l} \sqrt{N_i} L_i, \quad \tilde{\gamma}_i(l) := \lambda_i^{-l} \gamma_i, \quad (18)$$

with N_i the cardinality of $\mathcal{V}_i^{\text{out}}$ and where μ_i and γ_i satisfy Condition 2. The constants $\lambda_i \in (0, 1)$ and $\epsilon_i \in (0, 1]$, $i \in \mathcal{N}$, are tuning parameters. If the above conditions are satisfied, τ_{MIET}^i is defined as $\tau_{\text{MIET}}^i := \tau_{\text{max}}^i - \tau_{\text{MASP}}^i$. \square

Condition 3 can always be ensured, as long as sufficiently fast sampling is available. In practice, based on the constants γ_i , $(\tau_{\text{max}}^i, \tau_{\text{MAD}}^i)$ curves can be generated to intuitively select appropriate values for λ_i , $\phi_{0,i}(0)$ and $\phi_{1,i}(0)$.

These conditions are similar to the conditions in Dolk et al. (2017), even though PETC or the effect of sampling is not

considered. Indeed, in the continuous-time case, i.e., when τ_{MASP} approaches zero, $\tau_{\text{MIET}}^i = \tau_{\text{max}}^i$. This statement underlines that, if faster sampling is used, the continuous-time ETC behavior is recovered in the proposed setup.

5.3 Event-triggering Mechanism Design

To facilitate the design of the ETM, consider the following condition.

Condition 4. For $i \in \mathcal{N}$, consider the function H_i satisfying Condition 2. There exist locally Lipschitz functions $\underline{H}_i : \mathbb{R}^{n_y} \rightarrow \mathbb{R}_{\geq 0}$ that for all $e \in \mathbb{R}^{N n_y}$, $v \in \mathbb{R}^{n_v}$ and $x \in \mathbb{R}^{n_x}$, satisfy $\underline{H}_i(\hat{y}_i^{\text{in}}) \leq H_i(x, e_i^{\text{in}}, v)$. \square

The function Ψ_i in (9a) is given by, for any \hat{y}_i^{in} ,

$$\Psi_i(\hat{y}_i^{\text{in}}) := \varsigma_i(\hat{y}_i^{\text{in}}) + (1 - \epsilon_i)\mu_i N_i \underline{H}_i^2(\hat{y}_i^{\text{in}}), \quad (19)$$

where ς_i and \underline{H}_i come from Conditions 2 and 4, respectively. The function ϱ_i is given by, for any $y_i \in \mathbb{R}^{n_y}$ and $\hat{y}_i^{\text{out}} \in \mathbb{R}^{N n_y}$,

$$\varrho_i(y_i, \hat{y}_i^{\text{out}}) := \varepsilon_{\varrho} |e_i^{\text{out}}|^2 \quad (20)$$

with $\varepsilon_{\varrho} := (\tilde{\gamma}_i(0)\phi_{0,i}(\tau_{\text{MIET}}^i + \sigma_i) - \tilde{\gamma}_i(1)\phi_{1,i}(0)\lambda_i^2)$ where $\phi_{l,i}$, $l \in \{0, 1\}$ as in (17) and $\tilde{\gamma}_i : \{0, 1\} \rightarrow \mathbb{R}$ is as in (18). Finally, the function $\nu_i : \mathbb{R}^{n_y} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\leq 0}$ is defined as

$$\nu_i(y_i, \hat{y}_i^{\text{out}}, \tau_i) := (1 - \omega_i(\tau_i))\tilde{\gamma}_i(0)\varepsilon_{\nu} |e_i^{\text{out}}|^2, \quad (21)$$

where $\varepsilon_{\nu} := -(\phi_{0,i}(\tau_{\text{MIET}}^i) - \phi_{0,i}(\tau_{\text{MIET}}^i + \sigma_i))$ and

$$\omega_i(\tau_i) = \begin{cases} \{1\}, & \text{for } \tau_i \in [0, \tau_{\text{MIET}}^i) \\ [0, 1], & \text{for } \tau_i = \tau_{\text{MIET}}^i, \\ \{0\}, & \text{for } \tau_i > \tau_{\text{MIET}}^i. \end{cases} \quad (22)$$

Note that ν_i is single-valued for all $\tau_i \neq \tau_{\text{MIET}}^i$, and set-valued for $\tau_i = \tau_{\text{MIET}}^i$. Since the proof holds for all points in the set-valued map, in essence we can use the discontinuous version ($\omega_i(\tau_i) = 1$ if $\tau_i \leq \tau_{\text{MIET}}^i$ and 0 otherwise) to verify the condition in (8). Hence, the fact that ν_i is set-valued is not an issue with respect to (8).

In the proposed setup, each agent needs to know (and compute) constants ε_{ϱ} and ε_{ν} on-line due to the dependence on σ_i . If, from a computational standpoint, this is infeasible, a conservative upper-bound can be used by taking $\varepsilon_{\varrho} := (\tilde{\gamma}_i(0)\phi_{0,i}(\tau_{\text{max}}^i) - \tilde{\gamma}_i(1)\phi_{1,i}(0)\lambda_i^2)$ and $\varepsilon_{\nu} := (\phi_{0,i}(\tau_{\text{max}}^i) - \phi_{0,i}(\tau_{\text{MIET}}^i))$, which can be computed a priori.

We emphasize that the local ETMs as described by (8), (9), (19), (20) and (21), can operate fully asynchronously in the sense that clock synchronization or acknowledgment signals are not required.

5.4 Main result

Given the ETM design and the corresponding hybrid model presented above, we can now state the following result. Its proof is provided in the full version of this paper available on arXiv, see Scheres et al. (2022).

Theorem 1. Consider the system $\mathcal{H}(\mathcal{C}, F, \mathcal{D}, G)$ where Ψ_i , ϱ_i and ν_i are given by (19), (20) and (21), respectively. Moreover, suppose that Conditions 1-4 hold. Then the MAS described by \mathcal{H} is (\tilde{s}, \mathcal{S}) -flow-dissipative with the supply rate $\tilde{s} : \mathbb{X} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}$ as given in (11) and $\mathcal{S} = \{\xi \in \mathbb{X} \mid x \in \mathcal{X}, e = 0, \eta = 0\}$. In addition, if there are no finite escape times during the flow², then the system \mathcal{H} is persistently flowing. \square

² The absence of finite escape times during flow is meant here in the sense that case (b) in Prop. 2.10 in Goebel et al. (2012) cannot occur.

Theorem 1 implies that the desired stability and/or performance properties, guaranteed by the local controllers in absence of the network, are preserved by the original dissipativity property when the network is taken into account.

6. CASE STUDY

We apply the results to the single-integrator consensus problem, where we have a multi-agent system with $N \in \mathbb{N}_{>0}$ agents. All agents have state $x_i \in \mathbb{R}$, $i \in \mathcal{N}$, whose dynamics evolve according to $\dot{x}_i = u_i$ with $u_i \in \mathbb{R}$ the control input. The output of the system is the state x_i , i.e., $y_i = x_i$. We assume that the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with Laplacian matrix L is connected and undirected, i.e., $L^\top = L$. The control objective is for the states of all agents to asymptotically converge, i.e., $\lim_{t \rightarrow \infty} |x_i(t) - x_m(t)| = 0$ for all $i, m \in \mathcal{N}$. To achieve consensus, we implement the control law

$$u_i = -\sum_{m \in \mathcal{V}_i^{\text{in}}} (x_i + e_i^i - x_m - e_m^i). \quad (23)$$

We consider the Lyapunov candidate $V(x) = x^\top Lx$ where $x := (x_1, x_2, \dots, x_N)$. According to Dolk et al. (2019), the derivative of this Lyapunov function can be upper-bounded as $\langle \nabla V(x), -Lx - Le \rangle \leq \sum_{i \in \mathcal{N}} (-d_i z_i^2 - c_i u_i^2 + (\gamma_i^2 - \alpha_i) |e_i^i|^2)$ with $d_i := \delta(1 - aN_i)$, $c_i := (1 - \delta)(1 - aN_i)$ and $\gamma_i = \sqrt{a^{-1}N_i + \alpha_i}$, and where $\delta \in (0, 1)$, $a \in (0, \frac{1}{N_i})$ and $\alpha_i > 0$ are tuning parameters. The theorem below shows the exact expressions for all the required conditions. Its proof is omitted for space reasons.

Theorem 2. The system with local dynamics $\dot{x}_i = u_i$ and local controller (23) satisfies Conditions 1, 2 and 4 with $H_i = |u_i|$, $L_i = 0$, $s(x, e) = \sum_{i \in \mathcal{N}} (-d_i z_i^2 - \mu_i e_i^2)$, $\mathcal{X} = \{x \in \mathbb{R}^N \mid x_1 = x_2 = \dots = x_N\}$, $\varsigma_i = 0$, $\mu_i = c_i \frac{1}{N_i}$, $\gamma_i = \sqrt{a^{-1}N_i + \alpha_i}$, and $\underline{H}_i = |u_i|$. \square

Constants τ_{\max}^i and τ_{MAD}^i can be generated via an intuitive procedure, as described in Dolk et al. (2019). Theorem 2 implies that asymptotic consensus is achieved with the proposed control configurations in this paper.

We simulate the same system as Dolk et al. (2019) with non-uniform and time-varying transmission delays. However, in our case we implement our periodic event-triggered control framework instead of continuous-event triggered control as in Dolk et al. (2019). The system has $N = 8$ agents which are connected as described by a graph \mathcal{G} with undirected edges (1, 2), (1, 8), (2, 3), (2, 7), (3, 4), (3, 6), (4, 5), (5, 6), (5, 8) and (7, 8). We use the parameters $\delta = \alpha_i = 0.05$, $a = 0.1$ and $\epsilon_i = 0.5$ for all $i \in \mathcal{N}$. Given these tuning parameters, we obtain $\gamma_i = 4.478$ and $c_i = 0.76$ for agents $i \in \mathcal{N}$ with two neighbors (i.e., $N_i = 2$, thus agents P_1, P_4, P_6 and P_7) and $\gamma_i = 5.482$ and $c_i = 0.665$ for agents $i \in \mathcal{N}$ with three neighbors (i.e., $N_i = 3$, thus agents P_2, P_3, P_5 and P_8). The function $\varphi_i(\eta_i)$ is designed as $\varphi_i(\eta_i) = -\epsilon_\eta(\eta_i)$ with $\epsilon_\eta = 0.05$. We select $\lambda_i = 0.2$ for all agents, and pick $\phi_{0,i}(0) = 5$ and $\phi_{1,i} = 2$. For these values, we obtain $(\tau_{\max}^i, \tau_{\text{MAD}}^i) = (0.12, 0.016)$ for agents $i \in \mathcal{N}$ for which $N_i = 2$ and $(\tau_{\max}^i, \tau_{\text{MAD}}^i) = (0.09, 0.012)$ for agents $i \in \mathcal{N}$ for which $N_i = 3$. We select $\tau_{\text{MIET}}^i = 0.07$ ($\tau_{\text{MIET}}^i = 0.05$) for all agents for which $N_i = 2$ ($N_i = 3$), respectively, $\tau_{\text{MASP}}^i = 10^{-2}$ and $d_i = 10^{-3}$ for all $i \in \mathcal{N}$. At each sampling moment s_n^i , the next sampling moment is scheduled randomly such that $s_{n+1}^i \in [s_n^i + d_i, s_n^i + \tau_{\text{MASP}}^i]$ for each $i \in \mathcal{N}$, hence the sampling of each agent is aperiodic, asynchronous and independent of the other agents. The state evolution and inter-event times are depicted in Fig. 3, confirming our main theorem.

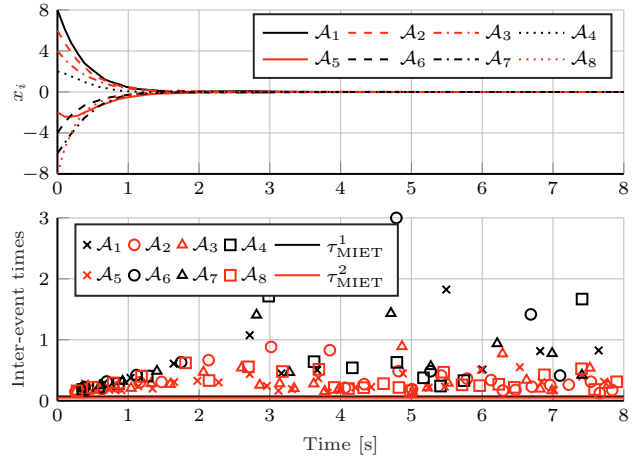


Fig. 3. States and inter-event times for the example in Section 6.

7. CONCLUSION

We presented a framework for the design of Zeno-free dynamic periodic triggering mechanisms for the control of nonlinear multi-agent systems via a packet-based network. The method can cope with non-uniform and time-varying delays. By ensuring that the conditions of the local trigger mechanisms only have to be verified at the local (asynchronous) sampling times, the proposed framework is suitable for implementation on digital platforms. With a dissipativity property, the framework can handle several relevant stability and performance properties such as asymptotic (set) stability, input-to-state stability, \mathcal{L}_p -stability with $p \in [1, \infty)$ and consensus, from a unified point of view. Thereby, capturing a wide range of systems to which these techniques can be applied.

REFERENCES

- Dolk, V.S., Borgers, D.P., and Heemels, W.P.M.H. (2017). Output-based and decentralized dynamic event-triggered control with guaranteed \mathcal{L}_p -gain performance and zeno-freeness. *IEEE Trans. Autom. Control*, 62(1), 34–49.
- Dolk, V.S., Postoyan, R., and Heemels, W.P.M.H. (2019). Event-triggered consensus for multi-agent systems with guaranteed robust positive minimum inter-event times. In *IEEE Conf. Decis. Control*, 2604–2609.
- Girard, A. (2015). Dynamic triggering mechanisms for event-triggered control. *IEEE Trans. Autom. Control*, 60(7), 1992–1997.
- Goebel, R., Sanfelice, R.G., and Teel, A.R. (2012). *Hybrid dynamical systems: modeling, stability, and robustness*. Princeton UP.
- Heemels, W.P.M.H., Donkers, M.C.F., and Teel, A.R. (2013). Periodic event-triggered control for linear systems. *IEEE Trans. Autom. Control*, 58(4), 847–861.
- Heemels, W., Bernard, P., Scheres, K., Postoyan, R., and Sanfelice, R.G. (2021). Hybrid systems with continuous-time inputs: Subtleties in solution concepts and existence results. In *IEEE Conf. Decis. Control*, 5368–5373. doi:10.1109/CDC45484.2021.9683389.
- Heemels, W., Johansson, K., and Tabuada, P. (2012). An introduction to event-triggered and self-triggered control. In *IEEE Conf. Decis. Control*, 3270–3285. IEEE.
- Nowzari, C., Garcia, E., and Cortés, J. (2019). Event-triggered communication and control of networked systems for multi-agent consensus. *Automatica*, 105, 1–27.
- Scheres, K.J.A., Dolk, V.S., Chong, M.S., Postoyan, R., and Heemels, W.P.M.H. (2022). Distributed periodic event-triggered control of nonlinear multi-agent systems. <https://arxiv.org/abs/2202.06282>.
- Teel, A.R. (2010). Asymptotic stability for hybrid systems via decomposition, dissipativity, and detectability. In *IEEE Conf. Decis. Control*, 7419–7424.
- Van der Schaft, A.J. (2017). *\mathcal{L}_2 -Gain and Passivity Techniques in Nonlinear Control, 3rd Edition*. Lect. Notes Control Inf. Sci. Springer International Publishing.
- Wang, W., Postoyan, R., Nešić, D., and Heemels, W.P.M.H. (2020). Periodic event-triggered control for nonlinear networked control systems. *IEEE Trans. Autom. Control*, 65(2), 620–635.
- Yu, H. and Chen, T. (2021). Periodic event-triggered networked control systems subject to large transmission delays. <https://arxiv.org/abs/2106.15806>.