A Window-based Periodic Event-triggered Consensus Scheme for Multi-agent Systems

Marc Seidel, Michael Hertneck, Pian Yu*, Steffen Linsenmayer, Dimos V. Dimarogonas, and Frank Allgöwer

***Abstract*—In this paper we consider the periodic event-triggered consensus problem for single-integrator multi-agent systems. In existing approaches, consensus is typically achieved by trigger schemes enforcing a monotone decrease of a Lyapunov function. Such trigger schemes may however result in more transmissions than actually required to ensure consensus or to meet certain performance specifications. This is because a monotone decrease may be a rather conservative condition for a given Lyapunov function in an event-triggered setting. To overcome the conservativity, we propose a novel window-based trigger scheme, which allows to leverage existing trigger schemes from the literature to derive less conservative ones. This is achieved by taking the past system behavior into account and allowing a temporary increase of the Lyapunov function as long as a decreasing tendency is still guaranteed. For that, information from previous time steps within certain time windows is considered. We provide an explicit bound on the evolution of the Lyapunov function that is the same for the window-based scheme and the (monotone) original trigger schemes. To illustrate the benefits of the window-based scheme, we validate its efficacy by a comprehensive simulation study and demonstrate that the scheme typically reduces the average update rate of the underlying communication structure in comparison to the original trigger schemes.**

***Index Terms*—Event-triggered Consensus, Networks of Autonomous Agents, Communication Networks, Networked Control Systems**

I. Introduction

The practical need for decentralized control solutions in networked systems has led to the increasing importance of multi-agent systems (MAS) over the past few years [1]–[4]. Advantages include high flexibility and low installation and maintenance costs. Typically MAS are connected by a communication network that introduces communication constraints and bandwidth limitations. Therefore, controlling MAS typically introduces additional challenges as, e.g., to use communication resources as efficient as possible. The underlying communication structure is typically described by a graph, specifying and limiting the possible communication among the agents.

In the field of MAS, the so-called *consensus problem*, is one of the fundamental tasks [1]. It has received increasing attention in research within the past decades [5]–[7]. The objective is that all agents agree on a joint state value, which is typically achieved by designing a suitable control law, which depends on the agents' neighbors' states and leads to convergence of each agents' state(s) to the consensus value.

A particularly relevant case is the continuous-time consensus problem with single-integrator dynamics for undirected graphs [8]. However, the necessary continuous computation of the control law and continuous communication among the agents as described in these works is hard to realize in practice. To that end, the periodically sampled-time consensus problem, that overcomes this drawbacks, has been proposed [9]. This setup is also called periodically time-triggered consensus (PTTC). The idea is to communicate with neighbors and update control inputs only at predefined sampling times. Nevertheless, this approach can be successful only if the sampling period is sufficiently small, which may result in unnecessary high network load in communication and frequent actuation updates. Therefore, event-triggered consensus (ETC) has been proposed as an alternative, inspired by the developments in event-triggered control [10]. The underlying idea of ETC is to broadcast information to neighboring agents only when this is indicated as necessary according to a state-dependent trigger condition called the *trigger rule*. Control inputs are updated whenever an agent itself or one of its neighbors triggers.

Since the original paper [5], various publications dealt with the topic of ETC for MAS [7], [11], [12]. The main advantage of ETC in comparison to PTTC is that agents trigger only when it is considered necessary by the trigger rule. In many setups, ETC leads to an overall reduction of the required communication [7], reducing energy consumption and necessary network bandwidth. Moreover, the actuation frequency and update computations are typically reduced. However, the trigger rule still needs to be monitored and evaluated continuously, which is in general challenging in digital implementation. The works [13], [14] modify the standard consensus protocol, such that finite-time convergence in ETC is achieved.

F. Allgöwer thanks the German Research Foundation (DFG) for support of this work within grant AL 316/13-2 and within the German Excellence Strategy under grant EXC-2075 - 285825138; 390740016. Pian Yu and Dimos V. Dimarogonas are supported by the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation (KAW).

Marc Seidel, Michael Hertneck, Steffen Linsenmayer, and Frank Allgöwer are with the University of Stuttgart, Institute for Systems Theory and Automatic Control, 70569 Stuttgart, Germany (e-mail {seidel,hertneck,linsenmayer,allgower}@ist.uni-stuttgart.de). Pian Yu and Dimos V. Dimarogonas are with School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 10044 Stockholm, Sweden (email {piany, dimos}@kth.se). *Current address: Department of Computer Science, University of Oxford.

The concept of periodic event-triggered consensus (PETC) aims to unify ETC and PTTC, in order to combine their advantages. Instead of the continuously monitored trigger rule in ETC, the trigger rule in PETC is checked periodically with a sufficiently small sampling period. Thus, the so-called *Zeno behavior*, where infinitely many trigger instances within a finite time are required, is naturally ruled out. Furthermore, a periodically checked trigger rule is much easier to implement on a digital hardware device than a continuously monitored one. Moreover, the overall network load in comparison to PTTC can still be reduced if a meaningful trigger rule is implemented. However, PETC often requires the agents' clocks to be synchronized, which may require additional effort. Early results on multi-agent PETC are stated in [6], requiring periodic communication among the agents, i.e., periodic state updates are required for trigger rule evaluation, but not for input recomputations. PETC results without the need for periodic communication are proposed in [15] and [12] and for an ETC setup is in [16]. Furthermore, a PETC scheme for agents with nonlinear dynamics has been developed in [17]. An explicitly time-dependent trigger rule for single- and double-integrator dynamics based on a time-dependent and decreasing error bound was introduced in [7] and further generalized in [18].

Most of existing results for (P)ETC are based on standard Lyapunov theory, such that a monotone decrease of a Lyapunov function is ensured by the trigger rule. This leads however potentially to a waste of network resources. We address this problem by introducing a novel trigger scheme, which provides the possibility to leverage existing trigger rules to derive less conservative trigger rules. This is achieved by taking the past system behavior into account in the form of using a sum of past values of the trigger rule, leading to a novel *window-based trigger scheme*, that allows to exploit leftover conservativity from previous time steps. The scheme is motivated by non-monotonic Lyapunov functions, i.e., by allowing an increase of the Lyapunov function as long as a decreasing tendency can still be guaranteed [19]. A similar idea has been proposed in [20], where it is shown that this can be advantageous to reduce the amount of transferred data. However, they do not consider a multi-agent setup, but rather a general event-triggered control setup.

The contribution of this paper can be divided into three parts. The first one is the development of the window-based trigger scheme, which allows less conservative trigger behavior compared to existing trigger rules. The scheme allows to take past information into account, which happens within so-called time windows. The considered information from previous time steps can be interpreted as a conservativity excess measure. This potentially allows a less conservative trigger decision at the current time step. We provide an explicit upper bound on the evolution of the Lyapunov function. Using the presented scheme, we are able to tune the upper bound directly, independent of the choice of the time-windows. This asymptotically converging upper bound guarantees a asymptotic convergence of the MAS to the consensus value.

As a second contribution, we develop two specific standalone trigger rules and apply the novel scheme to them. One of the trigger rules uses a purely time-dependent decreasing bound and leverages a discrete-time version of the one proposed in [7]. The other one leverages a state-dependent trigger rule, using the neighbor's states to determine the trigger instances. We provide an intuitive interpretation of why the scheme is able to reduce the conservativity of trigger rules, while guaranteeing the same upper bound.

As the third contribution, we perform an extensive simulation study, considering multiple parameter combinations and communication structures and thus covering a wide range of scenarios. Our simulations provide detailed evidence on the reduction of the average update rate and can be used as guideline for how to chose the time windows for the two explicitly considered trigger rules.

This remainder of this paper is organized as follows. The considered setup and preliminaries are presented in Section II. In Section III, we introduce the proposed trigger scheme and derive a bound on the Lyapunov function. The development of the specific trigger rules and their application on the window-based scheme can be found in Subsection III-C. To demonstrate the efficacy of the proposed scheme, the results of comprehensive simulations are shown and discussed in Section IV. Section V concludes the paper.

II. Preliminaries and Setup

A. Notation

Define $\mathbb{N} := \{x \in \mathbb{Z} \mid x > 0\}$, $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and $\mathbb{R}_+ := \{x \in \mathbb{R} \mid x > 0\}$. For a real symmetric matrix $L \in \mathbb{R}^{n \times n}$ we denote the eigenvalues of L by $\lambda_i\{L\}$, where $\lambda_{\min}\{L\}$ and $\lambda_{\max}\{L\}$ are the minimum and maximum eigenvalues. Eigenvalues are always sorted in an ascending order, i.e., $\lambda_i\{L\} \leq \lambda_{i+1}\{L\}$. Further, let $\|\cdot\|$ be the 2-norm of a vector or the induced 2-norm of a matrix, depending on the argument. The identity matrix of size $n \times n$ is denoted by I and $\mathbf{1} \in \mathbb{R}^n$ is the vector containing 1 in every entry. If $y \in \mathbb{R}^n$ is a vector, y_i denotes the i-th entry of y, $i \in \{1, ..., n\}$. Finally, we denote the transposed inverse of a square matrix U as $U^{-\top}$.

B. Algebraic Graph Theory

The underlying communication structure of the MAS is captured by an undirected, weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n vertices from the vertex set $\mathcal{V}$ and the edges $(i, j) \in \mathcal{E}$. The set $\mathcal{E}$ is the set of all edges of the graph. Denote the *adjacency matrix* $A = (a_{ij})$ with $a_{ij} = w(i, j)$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. The function $w : \mathcal{E} \to \mathbb{R}$ assigns a weight to every edge. Two vertices i and j are called adjacent, if there is an edge directly connecting them, i.e., if $(i, j) \in \mathcal{E}$. Since the graph is the underlying structure for an MAS, each vertex represents an agent. The set of neighbors of agent i is given by $\mathcal{N}_i := \{j : (i, j) \in \mathcal{E}\}$. The agent's degree is defined as $d_i := \sum_{j \in \mathcal{N}_i} w(i, j)$ i.e., the sum of all adjacent edges' weights of the corresponding agent. Denote the *degree matrix* $\Delta \in \mathbb{R}^{n \times n}$ as the diagonal matrix of all individual agents' degrees d_i. If for each pair of distinct vertices we can find a path between them, i.e., a sequence of edges connecting them, $\mathcal{G}$ is called connected. Otherwise the graph is disconnected. The *Laplacian (matrix)* L of $\mathcal{G}$ is defined as $L := \Delta - A$. By construction, $L = L^\top$. For real symmetric $n \times n$ matrices like L, a spectral decomposition $L = U\Lambda U^\top$ exists [21], such that Λ is the diagonal matrix with the eigenvalues of L in ascending order on the diagonal and $U = U^{-\top} \in \mathbb{R}^{n\times n}$ contains the corresponding eigenvectors of L as columns. The graph $\mathcal{G}$ is connected if and only if $\lambda_2\{L\} > 0$, i.e., the eigenvalues of the Laplacian are [1, Theorem 2.8]

$$0 = \lambda_1\{L\} < \lambda_2\{L\} \leq \lambda_3\{L\} \leq \ldots \leq \lambda_n\{L\}. \tag{1}$$

Furthermore, the eigenvector corresponding to the eigenvalue $\lambda_1\{L\} = 0$ is $v_1 = \mathbf{1}$. From that, it follows that the null space of L is spanned by $\mathbf{1}$ [1].

C. Setup

We assume the underlying graph representing the network structure of the MAS to be undirected, static, and connected. Consider a distributed control setup, i.e., agents can communicate with their respective neighbors only. The dynamics of each agent i are given by a single-integrator, i.e.,

$$\dot{x}_i = u_i, \tag{2}$$

where $x_i \in \mathbb{R}$ and $u_i \in \mathbb{R}$ are the state and input of agent i, respectively.

The goal is to have all agents' states converge to the same value. This quantity is called the *consensus value*. Define the *agreement set* as

$$\mathcal{A} := \{x \in \mathbb{R}^n \mid x_i = x_j \ \forall i, j\} = \operatorname{span}\{\mathbf{1}\}. \tag{3}$$

Each agent of the MAS is controlled in a periodic event-triggered fashion. Thus, for each agent a trigger rule is checked periodically at each time $t_k := t_0 + kh$ with the common sampling period $h \in \mathbb{R}_0$. This rule decides whether an event is triggered. If an event is triggered, the current state value of the agent is sent to the agent's neighbors. To formally describe the MAS, an additional state variable is thus required. Let $\hat{x}(t_k) := [\hat{x}_1(t_k)^\top, \ldots, \hat{x}_n(t_k)^\top]^\top$ be the vector of the last transmitted states, i.e., $\hat{x}_i(t_k)$ is the state value of agent i at its latest trigger instant. Further, define

$$e = \hat{x} - x \tag{4}$$

as the error between the last transmitted states and the current ones. All agents check their trigger rules periodically and if an event is triggered, broadcast their current state $x_i(t_k)$ to the neighbors $j \in \mathcal{N}_i$. The control input of agent i is updated whenever new information is received or broadcast.

We assume that the inputs are implemented in a zero-order hold fashion. The PETC protocol is given by

$$u_i(t) = \kappa \sum_{j\in\mathcal{N}_i} w(i,j)\big(\hat{x}_j(t_k) - \hat{x}_i(t_k)\big) \quad \text{for } t \in [t_k, t_{k+1}), \tag{5}$$

with the additional controller design parameter (control gain) $\kappa \in \mathbb{R}_+$. Within consecutive time steps, the inputs (5) are thus implemented in a zero-order hold (ZOH) fashion. Control laws like (5) are of closed-loop nature and therefore typically offer advantages like robustness against disturbances. Note that it is also possible to achieve consensus by other means than (5), e.g., by using *Flooding* [22, p.368f.]. Additionally, observe that the here presented scheme requires the clocks of all agents running synchronously. In practice, there are approaches to deal with that challenge, cf. [23] and the references therein.

Since we are dealing with a linear system evaluated periodically with fixed sampling period, we consider the exactly discretized system instead of the continuous-time system. The closed-loop behavior for $t \in [t_k, t_{k+1})$ is then given by $\dot{x}(t) = -\kappa L\hat{x}(t_k)$, cf. [1], [5]. The closed-loop system in continuous-time and discrete-time can be written as

$$x(t) = x(t_k) - \kappa(t - t_k)L\hat{x}(t_k) \tag{6}$$

$$x(t_{k+1}) = x(t_k) - \kappa hL\hat{x}(t_k) \tag{7}$$

and equivalently to (7)

$$x(t_{k+1}) = (I - \kappa hL)x(t_k) - \kappa hLe(t_k) \tag{8}$$

with using the error definition (4). We assume that all agents trigger at t_0 to initialize the PETC scheme.

Remark 1: Convergence, boundedness, and stability of the exactly discretized system (7) imply the respective property for the corresponding (original) continuous-time MAS (6) [24]. Thus, results that are subsequently derived for the exactly discretized system (7) imply the same guarantees for the continuous-time MAS (6).

Remark 2: In this paper, we restrict ourselves to single-state agents $x_i \in \mathbb{R}$ for simplicity reasons. However, the presented results for single-integrator dynamics can be extended to multiple states by redefining L using the Kronecker product, cf. [1].

D. Lyapunov function

We use Lyapunov arguments for convergence analysis. To that end, we define the following quadratic form Lyapunov function $V(x) = x^\top Lx$ as the quadratic form of $x = [x_1 \ \ldots \ x_n]^\top$ and the Laplacian matrix L. By construction, $Lx = \mathbf{0} \Leftrightarrow x \in \operatorname{span}\{\mathbf{1}\} = \mathcal{A}$, cf. Subsection II-B and hence, $V(x) = 0 \Leftrightarrow x \in \mathcal{A}$, i.e., V is positive definite with respect to $\mathcal{A}$. Thus, whenever $V \to 0$ it follows that x converges to the agreement set. Moreover, observe that $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{1}^\top x(t) = -\kappa\hat{x}(t_k)^\top L\mathbf{1} \equiv 0$ for $t \in [t_k, t_{k+1})$, i.e., the average of states is constant over time. Therefore, convergence to the agreement set $\mathcal{A}$ under the input (5) is equivalent to convergence to the consensus value. As a result, $V \to 0$ is a sufficient condition for reaching consensus.

III. Window-based trigger scheme

In this section, we propose the window-based trigger scheme that leverages existing trigger rules by using information from past time steps within time windows. Under the agreement protocol (5), the trigger scheme shall rely only on local information and should guarantee that all agents' states converge to the consensus value.

Before introducing the window-based trigger scheme, we analyze the time evolution of the Lyapunov function and derive an upper bound in Subsection III-A. This bound is then used for the trigger scheme that is introduced and discussed in Subsection III-B.

A. Upper bound on the time-evolution of the Lyapunov function

Almost all multi-agent (P)ETC approaches in the literature are based on various (continuous-time) upper bounds for $\dot{V}(x(t))$. A similar bound specifically for discrete-time MAS is computed subsequently. We denote $x_k := x(t_k)$ and $e_k := e(t_k)$. Using the closed-loop description (8), we obtain

$$\begin{aligned} V(x_{k+1}) =& \big[(I-\kappa hL)x_k - \kappa h e_k\big]^\top L\big[(I-\kappa hL)x_k - \kappa h e_k\big] \\ =& x_k^\top L x_k + x_k^\top(\kappa^2h^2L^3 - 2\kappa hL^2)x_k \\ &+ e_k^\top(\kappa^2h^2L^3)e_k - 2\kappa h x_k^\top L^2(I-\kappa hL)e_k. \end{aligned}$$

We utilize *Young's Inequality* $\pm 2x^\top Lz \le a x^\top Lx + \frac{1}{a}z^\top Lz$ with some $0 < a < 2$ and vectors x, z for the last term, which leads to

$$\begin{aligned} V(x_{k+1}) \le\ & V(x_k) - x_k^\top L\big((2-a)\kappa hL - \kappa^2h^2L^2\big)x_k \\ &+ e_k^\top\left(\frac{1}{a}L + \left(1-\frac{2}{a}\right)\kappa hL^2 + \frac{1}{a}\kappa^2h^2L^3\right)\kappa hLe_k. \end{aligned}$$

Define the matrices

$$\begin{aligned} P_x &:= (2-a)\kappa hL - \kappa^2h^2L^2 \\ P_e &:= \left(\frac{1}{a}L + \left(1-\frac{2}{a}\right)\kappa hL^2 + \frac{1}{a}\kappa^2h^2L^3\right)\kappa hL \end{aligned} \tag{9}$$

then one can get

$$V(x_{k+1}) - V(x_k) \le -x_k^\top LP_xx_k + e_k^\top P_ee_k. \tag{10}$$

Since we eventually need a decrease in V between some time steps t_k and t_{k+1}, we require P_x to be positive semi-definite. By invoking standard results from linear algebra, the eigenvalues of P_x are

$$\lambda_j\{P_x\} = \big((2-a) - \kappa h\lambda_i\{L\}\big)\kappa h\lambda_i\{L\} =: g(\lambda_i\{L\}). \tag{11}$$

However, as the different indices i and j in (11) indicate, the order of $\lambda_j\{P_x\}$ is in general not the same as $\lambda_i\{L\}$, i.e., $\big((2-a) - \kappa h\lambda_i\{L\}\big)\kappa h\lambda_i\{L\}$ is not necessarily the i-th eigenvalue of P_x. The relation (11) thus induces a permutation in the mapping from $\lambda_i\{L\}$ to $\lambda_j\{P_x\}$ and their associated eigenvectors. To establish the positive semi-definiteness of P_x with respect to $\mathcal{A}$, $\lambda_j\{P_x\} \ge 0\ \forall j$ is required. Assuming $\lambda_i\{L\} \ne 0$ and with (11), we obtain

$$(2-a) - \kappa h\lambda_i\{L\} \ge 0 \quad \forall i \tag{12}$$

as a necessary condition, where h, $\kappa > 0$ by design. The case $\lambda_i\{L\} = 0$, i.e., for $i = 1$, is considered later. Above inequality (12) is the most tight for $i = n$, because of (1). Therefore, (12) is satisfied if

$$\kappa h < \frac{2-a}{\lambda_n\{L\}}. \tag{13}$$

Equation (13) is a condition on κh for our scheme to work. Since now $\lambda_j\{P_x\} \ge 0$, we have $\lambda_1\{P_x\} = 0$, because $\lambda_1\{L\} = 0$, i.e., $\lambda_1\{L\}$ is mapped onto $\lambda_1\{P_x\}$. However, all other eigenvalues may be permuted. Since $\lambda_1\{P_x\} = 0$, we are specifically interested in $\lambda_2\{P_x\}$ as this eigenvalue consequently captures the important convergence properties. It can be computed as

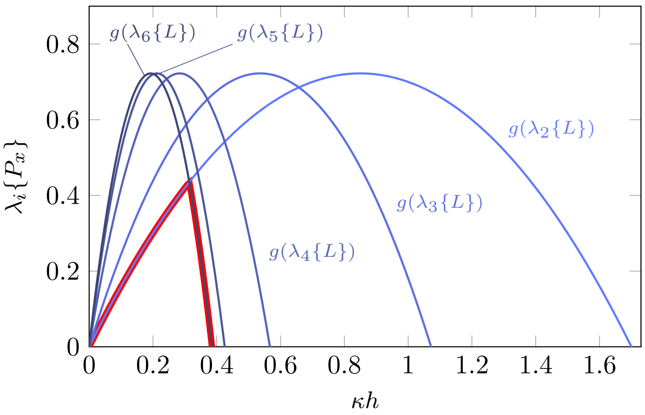


Fig. 1. Typical eigenvalues $\boldsymbol{\lambda_2}$ to $\boldsymbol{\lambda_n}$ of the matrix $\boldsymbol{P_x}$, depending on the design parameter $\boldsymbol{\kappa}$. The second smallest eigenvalue $\boldsymbol{\sigma = \lambda_2\{P_x\}}$ is highlighted in red.

$$\lambda_2\{P_x\} = \min_{i\ge 2}\Big\{\big((2-a) - \kappa h\lambda_i\{L\}\big)\kappa h\lambda_i\{L\}\Big\}. \tag{14}$$

Due to (11) being quadratic in $\lambda_i\{L\}$, the computation of $\lambda_2\{P_x\}$ can be reduced to

$$\lambda_2\{P_x\} = \min_{i\in\{2,n\}}\Big\{\big((2-a) - \kappa h\lambda_i\{L\}\big)\kappa h\lambda_i\{L\}\Big\}, \tag{15}$$

cf. Figure 1. Additionally, Figure 1 confirms graphically that (13) is the condition for $\lambda_i\{P_x\} > 0$ for $i \ge 2$ and thus for P_x being positive semi-definite. To tackle the permutation induced by (11), define the permutation map $\phi\colon j \mapsto i$ and $U = V\Phi$ for permuting the eigenvalues and eigenvectors, respectively. Hereby is Φ the corresponding permutation matrix for permuting columns with $\Phi^{-1} = \Phi^\top$. Using the spectral decomposition $L = U\Lambda U^\top = V\Phi\Lambda\Phi^\top V^\top$ and $P_x = VDV^\top$ with $y := V^\top x$, we further get that

$$\begin{aligned} -x^\top LP_xx &= -x^\top V\Phi\Lambda\Phi^\top DV^\top x \\ &= -\sum_{j=1}^{n}\lambda_{\phi(j)}\{L\}\lambda_j\{P_x\}y_j^2 \\ &= -\lambda_{\phi(1)}\{L\}\lambda_1\{P_x\}y_1^2 - \sum_{j=2}^{n}\lambda_{\phi(j)}\{L\}\lambda_j\{P_x\}y_j^2. \end{aligned} \tag{16}$$

Since $\lambda_1\{L\} = 0$ is mapped onto $\lambda_1\{P_x\} = 0$, we have $\phi(1) = 1$ and thus $\lambda_{\phi(1)}\{L\} = 0$. Moreover, $\lambda_i\{L\} > 0$ $\forall i \ge 2$ and $-\lambda_i\{P_x\} \le -\lambda_2\{P_x\}\,\forall i \ge 2$, so we can bound above equation by

$$\begin{aligned} -x^\top LP_xx &\le -\sum_{j=2}^{n}\lambda_{\phi(j)}\{L\}\lambda_2\{P_x\}y_j^2 \\ &= -\lambda_2\{P_x\}x^\top V\Phi\Lambda\Phi^\top V^\top x \\ &= -\lambda_2\{P_x\}x^\top Lx = -\lambda_2\{P_x\}V(x). \end{aligned} \tag{17}$$

Thus, with $\sigma := \lambda_2\{P_x\}$ and (10) we have

$$V(x_{k+1}) \le (1-\sigma)V(x_k) + \|P_e\|\,\|e(t_k)\|^2 \tag{18}$$

as an upper bound on the time-evolution of V. For $a \in (0,2)$ it is guaranteed that $\sigma < 1$, which can easily be verified by determining the global maximum of (15). Thus, σ can be seen as the natural and maximum convergence rate of the discrete-time system that can be guaranteed by (18). This maximum guaranteed convergence rate is achieved if $e = 0$ for all sampling times t_k, i.e., for PTTC. This means that σ is the maximum convergence rate which can be guaranteed in theory, although the actual one might be higher.

Remark 3 (Maximum allowable sampling period): Inequality (13) is the sufficient condition under which the bound (18) is valid for all initial conditions. In standard PTTC, where all agents trigger at any t_k, it is well-known that the sampling period h has to fulfill $h < \frac{2}{\lambda_n\{L\}}$ in order to reach consensus, see e.g., [9]. Thus, if we apply (5) for $\kappa = 1$, condition (13) on the sampling period is consistent with $h < \frac{2}{\lambda_n\{L\}}$, since the design parameter a has to fulfill $a \in (0,2)$. The additional parameter κ thus generalizes the PETC protocol (5) while the sampling period bound remains consistent with previous results. Given h and a bound on $\lambda_n\{L\}$, κ can always be chosen such that (13) is satisfied. The choice of κ thus has to happen either prior to, or at the same time than setting the sampling period on all agents, but not at the start of the consensus process itself. Thus, the agents do not need to agree on a common value for κ, instead the parameter is determined and set whenever the sampling period h is fixed

Remark 4 (Continuous-time convergence speed): For a fixed sampling period h we can choose the discrete-time convergence speed σ by using the controller parameter κ in order to achieve a desired continuous-time convergence speed μ. Note that the convergence speed σ of the exactly discretized system is evidently dependent on the sampling period h, cf. equation (15). However, the additional controller parameter κ allows to render the convergence speed (15) as well as the discrete-time upper bound (18) independent of the sampling period h. This is achieved by weighting the input (5) appropriately. If the weights $w(i,j)$ can be chosen freely, the same can be achieved by appropriately weighting the underlying graph of the MAS. However in a setting, where the weights $w(i,j)$ cannot be influenced, κ still allows us to render h independent of the other system parameters. This is especially useful in a setup, where h is bounded by practical limitations. Since κ can be computed based on h, no initial communication of κ to the agents is required. Whilst the maximum possible convergence rate of the exactly discretized system (7) is then independent of h, the convergence rate of the original continuous-time system (6) still depends on h. In particular, the continuous-time convergence rate is defined as the largest μ for which $V(x(t)) \leq e^{-\mu(t-t_0)}V(x(t_0))$ holds. For $t = t_0 + kh$, the discrete-time convergence rate $V(x(t)) \leq (1-\sigma)^k V(x(t_0))$ implies that the previous equation holds with $\mu = \frac{-\log(1-\sigma)}{h} > 0$.

B. Trigger scheme

Based on the bound (18), the key concept of this paper is introduced next. To keep things as general as possible, consider first some general trigger rule of the form

$$e_i(t_k)^2 < f_i(k, x_k), \tag{19}$$

where $f_i : \mathbb{N}_0 \times \mathbb{R} \to \mathbb{R}$ is called *trigger function* of agent i, for which typically $f_i > 0$ holds. In the literature, various trigger functions can be found to determine the trigger instants of agent i, see e.g., [5]–[7], [11], [12], [15], [16]. Whenever (19) is violated in a non-windowed trigger scheme, agent i triggers. Typically, the f_i's in the literature are chosen such that $V(x_{k+1}) < V(x_k)$ is ensured for all t_k. Furthermore, define

$$f(k, x_k) := \sum_{i=1}^{n} f_i(k, x_k). \tag{20}$$

The idea for our PETC scheme is: instead of merely taking the current time step in the trigger rule into account, prior information shall be exploited as well. More precisely, if at a previous sampling time the error was smaller than the bound induced by (19), the difference shall be used to increase the admissible error at the current sampling time.

To formalize the idea, define for each agent i a strictly increasing sequence $T_0^i, T_1^i, T_2^i, \ldots = \{T_m^i\}_{m\in\mathbb{N}_0} \subseteq \{t_k\}_{k\in\mathbb{N}_0}$ with $T_0^i = T_0 = t_0$ for all i. The sequences are in general distinct for different agents. For now, no further assumptions on the sequences shall be made until Section IV, where we analyze specific sequence choices. Let further k_m^i be the time index at T_m^i, i.e., $t_{k_m^i} = T_m^i$. These sequences define *time windows* of discrete sampling times for evaluating the trigger rule. From the current time step k, let T_m^i be the closest prior element of the corresponding sequence, i.e.,

$$T_m^i = \max_{\substack{\tilde{t} \in \{T_m^i\}_{m\in\mathbb{N}_0} \\ \tilde{t} \leq t_k}} \tilde{t}. \tag{21}$$

The window for which the trigger rule is evaluated starts at $T_m^i + h$, i.e., at $t_{k_m^i+1}$ and lasts up to the current time step t_k or until T_{m+1}^i is reached. At the lower part of Figure 2, a visualization is shown on how the time windows depend on the corresponding sequence $\{T_m^i\}_{m\in\mathbb{N}_0}$. Note that the time windows for agent i defined by $\{T_m^i\}_{m\in\mathbb{N}_0}$ are in general distinct and independent of the trigger instants τ^i (event times of agents), cf. Figure 2. We detail on possible window choices in Section III-D.

The precise formulation of the window-based trigger scheme as well as a resulting guaranteed bound on the evolution of the Lyapunov function is given next.

Theorem 1 (Window-based trigger scheme): Define for each agent i a strictly increasing sequence $\{T_m^i\}_{m\in\mathbb{N}_0}$ with $T_m^i \in \{t_k\}_{k\in\mathbb{N}_0}$ and $T_0^i = T_0 = t_0$ for all i. Suppose (13) holds for the chosen $a \in (0,2)$ and $\kappa \in \mathbb{R}_+$ and that the control inputs are given by (5) for all agents i. Moreover, let the trigger rule for each agent be

$$\begin{aligned} e_i(t_k)^2 &< f_i(k, x_k) \\ &\quad + \sum_{\bar{k}=k_m^i+1}^{k-1} (1-\sigma)^{k-\bar{k}} \Big(f_i(\bar{k}, x_{\bar{k}}) - e_i(t_{\bar{k}})^2 \Big) \end{aligned} \tag{22}$$

within the time window $k \in [k_m^i + 1, k_{m+1}^i]$ for the trigger function f_i. Let the sequence of trigger instants be defined by

$$\{\tau_m^i\}_{m\in\mathbb{N}_0} := \{t_k = t_0 + kh \text{ and } (22) \text{ is violated at } t_k \text{ for agent } i\}. \tag{23}$$

Then the following bound holds for all $k \in \mathbb{N}_0$:

$$V(x_k) < (1-\sigma)^k V(x_0) + \|P_e\| \sum_{\bar{k}=0}^{k-1} (1-\sigma)^{k-\bar{k}-1} f(\bar{k}, x_{\bar{k}}). \tag{24}$$

The proof of Theorem 1 can be found in Appendix A. Note here again that the sequences $\{\tau_m^i\}_{m\in\mathbb{N}_0}$ and $\{T_m^i\}_{m\in\mathbb{N}_0}$ defining the trigger instants and the time windows, respectively, are in general distinct.

Remark 5: Theorem 1 reveals an important property of the window-based trigger scheme: the bound on the Lyapunov function V is independent of the choice of the windows. Thus using (19) as trigger rule leads to the same bound as using (22) for any window choice. However, we note that the resulting performance of the two trigger rules is different, as we will show later in Section IV.

Comparing (22) to (19) reveals an important advantage of the proposed trigger scheme. At some time $t_k \neq t_{k_m^i+1} = T_m^i + h$, it is known from the trigger rule of the previous steps back to the beginning of the current time window that for agent i

$$\sum_{\bar{k}=k_m^i+1}^{k-1} (1-\sigma)^{k-\bar{k}} \left(f_i(\bar{k}, x_{\bar{k}}) - e_i(t_{\bar{k}})^2 \right) > 0. \tag{25}$$

This is because if the sum is about to get nonpositive, the agent is triggered, which in turn only adds a nonnegative f_i to the sum. Thus, (25) potentially allows additional freedom for the error at the current time step. It can be interpreted as exploiting the excess of conservativity from previous time steps. A higher error $e_i(t_k)^2$ might be allowed at the current time step if in previous time steps (up to the last $t_{k_m^i+1}$, i.e., inside the time window) the error was small compared to the allowed error bound. Therefore, it can be used as conservativity excess in the current time step. As a result, $e_i(t_k)^2$ is allowed to be higher than the associated bound $f_i(k, x_k)$ at this time instant due to this exploited excess of conservativity. Our scheme is thus motivated by the concept of non-monotonic Lyapunov functions [19], which it implicitly utilizes. While often trigger rules are designed such that a decrease in the Lyapunov function is required at every time step, the use of conservativity excess in (22) allows the Lyapunov function to be non-monotonic, as long as a decreasing tendency is guaranteed. We solely give an upper bound (24) on the time evolution of $V(t_k)$, but not in between consecutive time steps.

To emphasize the conservativity excess interpretation, define

$$S_{i,k} := \sum_{\bar{k}=k_m^i+1}^{k} (1-\sigma)^{k-\bar{k}} \left(f_i(\bar{k}, x_{\bar{k}}) - e_i(t_{\bar{k}})^2 \right). \tag{26}$$

Here, $S_{i,k}$ is exactly the conservativity excess at time t_k, including time steps prior to k up to the start of the current time window. The trigger rule (22) can then be replaced by

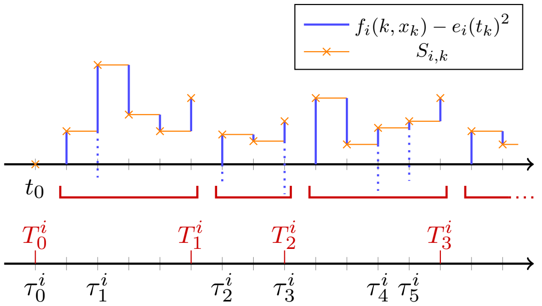


Fig. 2. Visualization of the window-based trigger scheme: time window definition based on an exemplary sequence $\{T_m^i\}_{m\in\mathbb{N}_0}$ (red) and conservativity excess ($S_{i,k}$) interpretation (blue). Dotted lines represent the conservativity excess behavior if the agent would not trigger at that specific time.

$$0 < S_{i,k} \tag{27}$$

and $S_{i,k}$ can be computed from the previous excess of conservativity as

$$S_{i,k} = (1-\sigma) S_{i,k-1} + \left(f_i(k, x_k) - e_i(t_k)^2 \right) \tag{28}$$

by using only additional information at time t_k. We note that the proposed trigger scheme can thus be implemented efficiently using the above reformulation, since not all paststate values and errors need to be stored for the current time window in contrast to when implementing it according to (22).

The main benefit of the reformulation (27) is that it allows more insight into the main idea of the scheme: At the current time step and within the current time window, the conservativity excess $S_{i,k}$ is computed. If this excess is about to get nonpositive, the agent triggers, thus setting $e_i(t_k)$ to 0. Therefore, the trigger rule prevents the conservativity excess of getting nonpositive, but at the same time accounting for conservativity excess from previous time steps. In Figure 2, a visualization of the conservativity excess interpretation is shown, illustrating the main motivation of our scheme.

For example at t_3, where $f_i(k, x_k) - e_i(t_k)^2$ is negative, the standard trigger rule (19) would trigger, whereas the windowed one (22) does not, because there is leftover conservativity from previous time steps. Due to the design of (22), $S_{i,k-1}$ is reset to 0 whenever a new window starts, cf. Figure 2. The above discussion provides thus an intuitive explanation why the presented scheme has the potential of reducing the number of trigger instances.

The scheme is conceptually similar to [16], where the authors present a dynamic trigger rule in a continuous-time ETC setup, but without the conservativity excess interpretation. In fact, (22) can be interpreted as a dynamic trigger condition, with $S_{i,k}$ being the dynamic variable for each agent i. The window-based scheme therefore resembles a specific choice of dynamic variables with an explicit motivation, namely the reduction of trigger instances for existing trigger rules by exploiting leftover conservativity from previous time steps and being motivated by the concept of non-monotonic Lyapunov functions. Using (22) therefore leads to a potentially less conservative trigger rule compared to (19) while guaranteeing the same upper bound. Note that this does not necessarily imply that there will be less trigger instances in the closed-loop. In Section IV, the resulting closed-loop behavior will be examined in detail. As a conclusion, even though (22) is a more general formulation of (19) that allows a higher bound for the error in the trigger rule and therefore leading to a potentially less conservative trigger rule, the guaranteed upper bound on the evolution of the Lyapunov function remains the same. Note that similar behavior was reported for the dynamic trigger rule from [16] in comparison to the corresponding static trigger rule.

Remark 6: The presented window-based trigger scheme is related to the one given in [20] for continuous ETC, where the error difference is integrated and transmissions are only triggered when the value of the integral reaches 0. This less conservative trigger rule, which is considered in a general networked control setting, reduces the number of transmissions between plant and controller. An important difference of the proposed trigger scheme in contrast to the one presented in [20] is that our T_m^i are not restricted to represent the trigger instants. This enables a more variable trigger rule design with potentially better performance.

Remark 7: The design parameter a influences the matrices P_x and P_e and thus σ and the bound (18). With that, also the convergence speed of the corresponding continuous-time system (6) is influenced, cf. Remark 4.

C. Specific trigger rules for the proposed scheme

The general formulation (19) captures a wide range of existing non-windowed trigger functions. To illustrate the application of the general concept we consider two specific type of existing trigger functions in this subsection on which the window-based concept, i.e., (22) is applied. First we consider time-dependent trigger functions, where f_i is explicitly dependent on time only and therefore independent of the current state. Secondly, we examine a special type of state-dependent trigger rules, where the trigger function f_i is now explicitly dependent on the current state x_k.

1) Time-dependent trigger functions: When looking at (24) it is evident that $V(x_k) \to 0$ if the sum

$$\sum_{\bar{k}=0}^{k-1} (1-\sigma)^{k-\bar{k}-1} f(\bar{k}, x_{\bar{k}}) \tag{29}$$

converges to 0 as $k \to \infty$, which can be ensured for example for $f_i \to 0$. Thus we can use Theorem 1 to derive an entire class of purely time-dependent trigger rules for which convergence to the consensus value is guaranteed.

Theorem 2: Let the conditions of Theorem 1 hold, where the window-based trigger rule (22) is implemented with $f_i(k, x_k) = f_i(k)$, $0 \le f_i(k) < \infty$ for all $k \ge 0$ and for all $i \in \{1, ..., n\}$, and $\lim\limits_{k\to\infty} f_i(k) = 0$. Then the MAS asymptotically converges to the consensus value.

Proof: Recalling equation (20), if $f_i(k)$ fulfills the conditions $0 \le f_i(k) < \infty$ and $\lim\limits_{k\to\infty} f_i(k) = 0$, so does $f(k)$. Next consider (29). From Kronecker's Lemma [25, Lemma 2] we obtain convergence of (29) to 0 as $N \to \infty$. Thus, $\lim\limits_{k\to\infty} V(x_k) = 0$ and therefore asymptotic convergence of x to the consensus value follows. ■

Remark 8: Kronecker's Lemma does not require that $f_i(k) \ge 0$ for all i, k. Technically, negative $f_i(k)$ are possible as well, as long as the series $\sum_{k=1}^{N} f(k)$ is converging. However, this is in general not reasonable in view of the trigger rule (19) as a negative trigger function would always lead to triggering independent of the error.

In [7], a purely time-dependent exponentially decaying upper bound on the error was proposed as trigger rule for a continuous-time ETC setup. In view of this trigger rule, consider

$$f_i(k, x_k) = E_0(1-\theta)^k \tag{30}$$

as the corresponding discrete-time trigger function, where $E_0 > 0$ is some arbitrary constant (initial error bound) and $\theta \in (0,1)$ denotes the tunable convergence rate of the error bound. Note that the trigger function (30) fulfills the conditions of Theorem 2. Thus, the window-based version of the trigger rule $e_i(t_k)^2 < E_0(1-\theta)^k$ leads to consensus among the agents. Additionally, we can provide an explicit upper bound on the time-evolution of the Lyapunov function as captured by the following corollary.

Corollary 1: Suppose the assumptions of Theorem 1 hold for the trigger function f_i according to (30). Then the MAS asymptotically converges to the consensus value and the exponentially decreasing bound

$$\begin{aligned} V(x_k) <\, & (1-\sigma)^k V(x_0) \\ & + n \left\| P_e \right\| \frac{E_0}{\theta - \sigma} \Big((1-\sigma)^k - (1-\theta)^k \Big) \end{aligned} \tag{31}$$

holds for all $k \in \mathbb{N}_0$.

Proof: Since the trigger function (30) fulfills the assumptions of Theorem 2, convergence to the consensus value directly follows. To obtain the exponentially decreasing bound (31), we use Theorem 1 with the trigger function (30):

$$\begin{aligned} V(x_k) <\, & (1-\sigma)^k V(x_0) \\ & + \|P_e\| \sum_{\bar{k}=0}^{k-1} (1-\sigma)^{k-\bar{k}-1} \sum_{i=1}^{n} E_0(1-\theta)^{\bar{k}} \\ =\, & (1-\sigma)^k V(x_0) \\ & + \|P_e\| \sum_{\bar{k}=0}^{k-1} (1-\sigma)^{k-\bar{k}-1} n E_0 (1-\theta)^{\bar{k}}. \end{aligned}$$

This sum is finite and can be calculated explicitly. Using the relation $\sum_{\bar{k}=0}^{k-1} r^{\bar{k}} = \frac{1-r^k}{1-r}$ for $r = \frac{1-\theta}{1-\sigma}$ yields

$$\begin{aligned} V(x_k) <\, & (1-\sigma)^k V(x_0) + n \left\| P_e \right\| E_0 (1-\sigma)^{k-1} \sum_{\bar{k}=0}^{k-1} r^{\bar{k}} \\ =\, & (1-\sigma)^k V(x_0) \\ & + n \left\| P_e \right\| \frac{E_0}{\theta-\sigma} \Big((1-\sigma)^k - (1-\theta)^k \Big), \end{aligned}$$

which provides the desired result. Note that, if $\theta = \sigma$ the bound (31) is still valid when determining its limit for $\theta \to \sigma$ with l'Hôpital's rule. ■

Remark 9: Since $E_0 \in (0, \infty)$, there exists some $\gamma > 0$ such that $E_0 \leq \gamma V(x_0)$ as long as $V(x_0) \neq 0$. Then, using Corollary 1, one can derive exponential stability with respect to $\mathcal{A}$, i.e., ensure

$$V(x_k) < \Big(\big(1 + \frac{\gamma n \, \|P_e\|}{\theta - \sigma}\big)(1-\sigma)^k - (1-\theta)^k\Big) V(x_0). \tag{32}$$

Remark 10: A practical choice of θ should satisfy $\theta < \sigma$. Otherwise the error convergence rate θ is larger than the system's guaranteed convergence rate σ, which possibly leads to more frequent triggering. This is consistent with [7, Theorem 3.2], where the convergence rate of the error bound is upper bounded by the system's convergence rate as well.

2) State-dependent trigger functions: We introduce the notation of $V_i(x_k)$ as a decomposition of the Lyapunov function according to

$$V(x) = \sum_{i=1}^{n} \frac{1}{2} \sum_{j \in \mathcal{N}_i} w(i,j)(x_i - x_j) = \sum_{i=1}^{n} V_i(x), \tag{33}$$

which can be interpreted as a *local Lyapunov function*. Consider the upper bound on V that is given by (18). To decrease V over time, use

$$f_i(k, x_k) = \frac{\sigma - \zeta}{\|P_e\|} V_i(x_k) \quad \text{for some } 0 \leq \zeta < \sigma \tag{34}$$

as a trigger function, leading to $V(x_k) < (1-\zeta)^k V(x_0)$.

When using the trigger function (34) for the trigger rule (19), exponential stability of the consensus set follows from standard Lyapunov arguments. For using (34) as trigger function with the trigger scheme (22), exponential convergence to the consensus set can be proven based on Theorem 1:

Theorem 3: Suppose the assumptions of Theorem 1 hold, where the window-based trigger rule (22) is implemented with $f_i(k, x_k)$ given by (34). Then the MAS exponentially converges to the consensus value and the bound

$$V(x_k) < (1-\zeta)^k V(x_0) \tag{35}$$

holds for all $k \in \mathbb{N}_0$.

Proof: The proof uses Theorem 1 with the trigger function $f_i(k, x_k) = \frac{\sigma - \zeta}{\|P_e\|} V_i(x_k)$. We obtain

$$\begin{aligned} V(x_k) &< (1-\sigma)^k V(x_0) \\ &\quad + \|P_e\| \sum_{\bar{k}=0}^{k-1} (1-\sigma)^{k-\bar{k}-1} \sum_{i=1}^{n} \frac{\sigma - \zeta}{\|P_e\|} V_i(x_{\bar{k}}) \\ &= (1-\sigma)^k V(x_0) + \sum_{\bar{k}=0}^{k-1} (1-\sigma)^{k-\bar{k}-1} (\sigma - \zeta) V(x_{\bar{k}}). \end{aligned} \tag{36}$$

From Lemma 1 given in Appendix B for $a = 1 - \sigma$ and $b = \sigma - \zeta$ we obtain (35). ■

Remark 11: The bound (35) also implies global exponential stability of the consensus value.

Note that for implementing the trigger function (34), state information from all neighbors is required, which can typically only be achieved with periodic communication. In practice, this is a potential drawback of the state-dependent trigger function (34). Thus, the main advantage of the choice of f_i in (34) is to reduce the frequency of actuation updates. This especially applies for settings where the process of actuating and the computation thereof is resource demanding. An extension of the here presented scheme to a state-dependent event-triggered communication setup is not straightforward. This is because the presented convergence proof relies on an explicit upper bound on V, which is challenging to provide for a window-based rule without knowing the states of all neighbors.

Remark 12: Consider the generalized version of (34)

$$f_i(k, x_k) = C_i V_i(x_k). \tag{37}$$

The trigger rule

$$e_i(t_k)^2 < \frac{1}{\lambda_n\{L\}^2} \Big(\sum_{j \in \mathcal{N}_i} w(i,j)\big(x_i(t_k) - x_j(t_k)\big) \Big)^2 \tag{38}$$

presented in [6] can be brought into the form (37), utilizing the *Cauchy-Schwarz Inequality* to obtain V_i according to (33). Furthermore, as noted in [16], the trigger rule developed in [5] is a more conservative case of (37). Thus, Theorem 3 applies to these trigger rules in the literature as well.

D. Time window choices

Any T-sequence satisfying the assumptions made in Theorem 1 can be chosen. Some convenient choices are listed subsequently.

- *Fixed window.* A simple option is to chose all time windows with a fixed length $\mu_i \in \mathbb{N} \cup \{\infty\}$. For finite μ_i, the time windows and thus the trigger rules get reset within a fixed time interval, whereas for $\mu_i = \infty$ the trigger scheme exploits any conservativity excess since the first time step. In the latter case, the length of the window goes to infinity if the current sampling instant goes to infinity. For $\mu_i = 1$ we have the shortest window length possible, which corresponds to exploiting no conservativity excess at all and reducing (22) to the non-windowed original trigger rule (19). Therefore, the presented scheme is a generalization of any trigger rule of the form (19).
- *Trigger window.* Another option is to determine the window size dynamically by starting a new window at each trigger instant. Then, T_m^i is chosen to be the m-th trigger instant τ_m^i of agent i, starting at $T_m^i = t_0$. The sequences $\{\tau_m^i\}_{m \in \mathbb{N}_0}$ and $\{T_m^i\}_{m \in \mathbb{N}_0}$ then coincide, whereas in all other cases they are in general distinct. Note that this window choice corresponds to the choice of the limits of the integral in [20] in a general event-triggered control setup. It has also similarities to the event-triggered control approach from [19], where the decrease of a Lyapunov function sampled at transmission times is ensured.
- *Hybrid window.* A third option is to combine the previous approaches. For this window type, a maximum window length is selected for each agent. The next window begins

if either the next event is triggered, or the maximum window length μ_i of that agent is reached.

Remark 13: An other option is a *moving window*, where a fixed length time window reaches back a fixed number of time steps and dynamically moves with the current time. This is however not covered by the theory presented in this paper and thus considered as an open point for future research.

IV. Numerical Evaluation

In this section, we examine the closed-loop behavior of the proposed trigger scheme. As discussed in Subsection III-B, the presented scheme is able to prolongate the time span between trigger instants from an open-loop perspective in comparison to the corresponding original non-windowed trigger rule. However, this allows no direct conclusion for the closed-loop behavior for the MAS. Note that it is in general hard to make analytic conclusions about the resulting closed-loop behavior for PETC. Thus, we examine the closed-loop behavior in an extensive simulation study to cover a wide range of scenarios. Recall that the theoretically resulting upper bound on V is the same, no matter how $\{T_m^i\}_{m\in\mathbb{N}_0}$ is chosen. However, the time window choice may have a significant influence on the average number of required trigger instances. For comparing different choices of the time windows, we define the *update rate* of the underlying communication structure as the ratio between the actual occurred trigger instances since t_0 and the maximum possible amount of trigger instances, i.e., for PTTC. Mathematically, at time t_k it is defined as

$$\text{update rate} = \frac{m+1}{k+1} \in [0,1], \tag{39}$$

where m is the number of times agent i triggered since (and including) t_0, i.e., taken from the latest τ_m^i at t_k.

Remark 14: The term update rate was chosen such that it fits for both the time- and the state-dependent trigger rule considerations. As the presented state-dependent trigger rule requires periodic communication, its main purpose is the reduction of actuation updates, as mentioned in Section III-C.2. On the other hand, the time-dependent trigger rules do not require periodic communication and thus additionally reduce the required communication among the agents. Therefore, the term update rate can in this case also be interpreted as the *network load* of the underlying communication graph.

Subsequently we investigate the influence of different quantities on the update rate, like the chosen parameters, the graph representing the network structure, the initial condition $x(t_0)$, the time window choice, and the trigger rule parameters, e.g., the factor a. Since these quantities might have an influence on the resulting update rate, it is more convenient to look at the average update rate for a large number of simulations. Further, for some initial conditions the states require more trigger instances and time steps until convergence than others. Therefore, to capture a wide range of scenarios and obtain reliable information on the average update rate, a set of various parameters and graphs is considered. The goal of our simulation is to cover a broad range of scenarios which is why we do not just on simulation run for fixed parameters. Note that it is still not possible to obtain rigorous guarantees about the efficacy of the algorithm. We evaluate the update rate for a simulation for each parameter combination and for various initial conditions. Parameters that are varied and their ranges are $\kappa \in \{0.05, 0.1, 0.15, 0.2\}$, $\theta, \zeta \in \{0.01, 0.05, ..., 0.17\}$, and $\gamma \in \{0.05, 0.15, ..., 0.45\}$. Since the trigger mechanism (22) depends on σ, which implicitly depends on a, we also iterate over $a \in \{0.1, 0.3, ..., 0.9\}$. Parameter combinations, for which $\theta \geq \sigma$ or $\zeta \geq \sigma$ hold are excluded, cf. Remark 10 and Theorem 3. For each parameter combination, the same 100 random initial conditions, drawn randomly from the interval $(-1000, 1000)$, are used. We use the PETC scheme with fixed window length $\mu = 1$ as a baseline for our comparison(s), as this corresponds to the non-windowed original trigger rule. The considered graphs are captured by Figure 3. To conveniently illustrate the update rate for all simulation runs, a histogram for the update rates for all simulation runs is plotted in Figure 4. The resulting update rates are grouped into bins, which indicate how many times the corresponding update rate occurred. The average update rate as the average of all simulation runs is depicted as a vertical red line. This allows to interpret the outcomes in a more concise way compared to looking at single simulations, as the distribution of the update rates is clearly visible. In total, 112,000 simulation runs are considered per window option. Note that from the numerical simulations in this paper one cannot obtain guarantees for any possible situation solely based on numerical simulations. However, since we are considering a broad range of possible scenarios, we can deduce a strong indication that our trigger scheme is indeed able to reduce the average update rate in many situations.

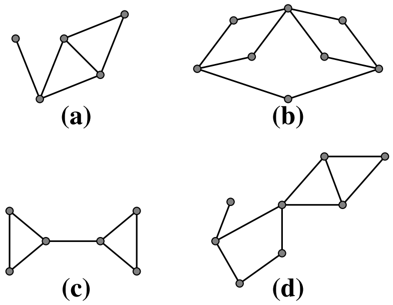


Fig. 3. The four unweighted graphs used for the update rate analysis, representing the underlying communication structure of the MAS.

For the trigger function of Corollary 1, it can be seen that the average update rate is with 12.3% the smallest among all investigated window choices for the trigger window variant. For all examined window variants with fixed window size, the resulting average update rate is higher. When comparing the trigger window to the fixed window with length $\mu_i = 1$, which is equivalent to the original trigger rule, we observe that for the considered examples our proposed scheme with the trigger window option is able to reduce the average update rate. For the conducted simulations, the relative reduction is approximately 14.3%. Moreover, also the standard deviation is decreased from 5.91 to 4.55. This indicates that a scenario in which the update rate deviates significantly from the average is rather unlikely. To illustrate the resulting convergence behav-

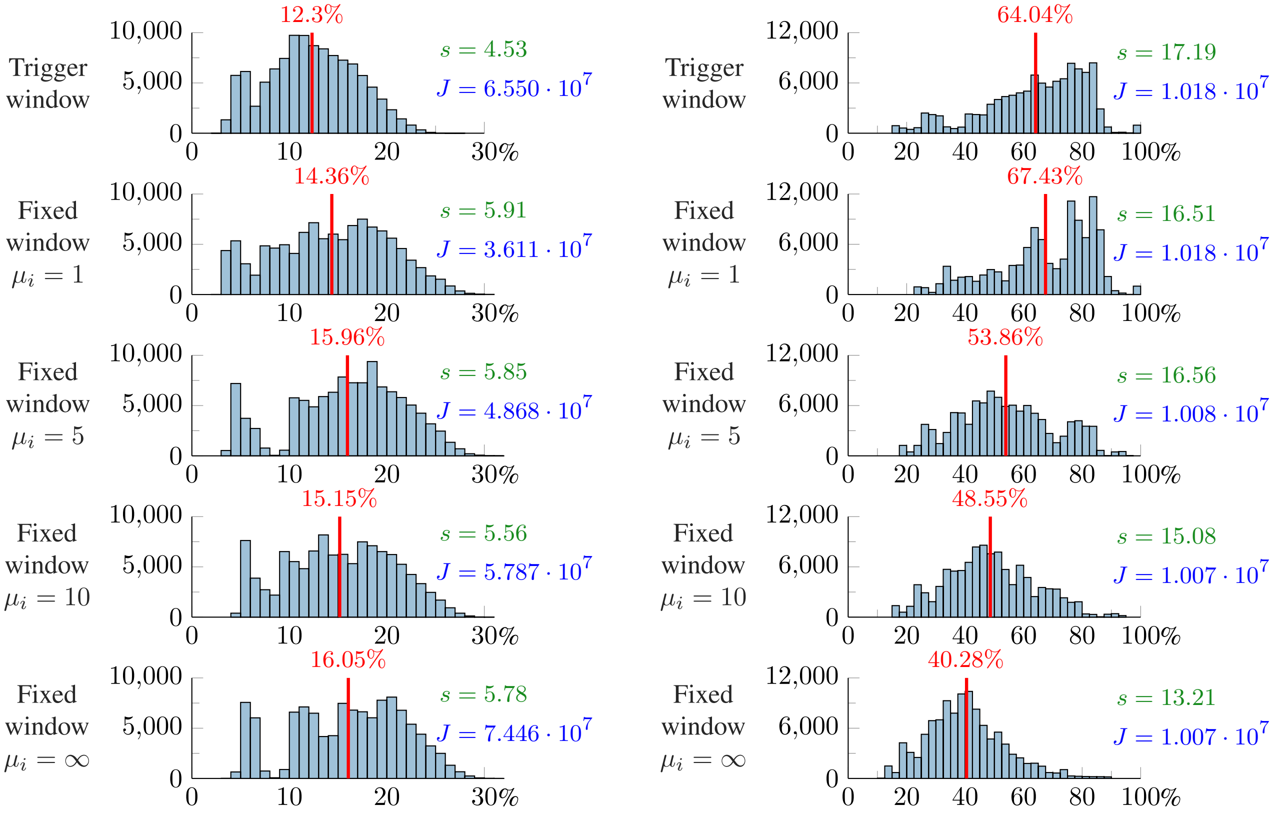


Fig. 4. Update rate histograms for the simulation of different window options for the time-dependent trigger rule of Corollary 1 (left) and for the state-dependent trigger rule of Theorem 3 (right). For comparison, the average value for each window option is indicated by a red vertical line. The corresponding standard deviation and the average quadratic cost $\boldsymbol{J}$ are denoted $\boldsymbol{s}$, respectively.

ior, one can consider the quadratic cost $J = \sum_k V(t_k)$. For the trigger rule of Corollary 1, J is larger as for the non-windowed trigger rule. This is not surprising, as the proposed approach aims at reducing update rate whilst exploiting conservativity in the trigger rule, leading potentially to a convergence behavior closer to the chosen bound.

We now consider the state-dependent trigger function (34), for which the simulation results are illustrated on the right side of Figure 4. For our simulations, the trigger window also leads to a smaller average update rate than the fixed window with $\mu_i = 1$. However, we observe that for the considered simulations a larger fixed window size leads to even less average update rate. So in contrast to the previously studied time-dependent trigger function, a larger window size is preferable. If we again take the fixed window with length $\mu_i = 1$ as a baseline for our comparison, the relative difference compared to the largest possible window size $\mu_i = \infty$ is around 40%. The standard deviation decreases from 16.51 to 13.21. When looking at the resulting quadratic cost J for the different window options for the state-dependent trigger function, there is essentially no difference between the considered window options, i.e., all window options perform similar regarding quadratic cost, even though they differ significantly in terms of update rate. This emphasizes that for the state-dependent trigger rule (34), the proposed scheme can reduce the update rate even without performance loss.

Based on the average costs, the state converges faster for the state-dependent variant (34) than for the time-dependent one (30), at the cost of a higher triggering frequency.

To summarize, our simulation study provide strong evidence that the trigger window option typically performs best for the time-dependent trigger function (30) according to Corollary 1 and leads for our investigated scenarios to a update rate reduction of around 14.1%. For the state-dependent trigger function (34) according to Theorem 3, our simulations study indicates that a large fixed window size should be used, due to the resulting update rate reduction of around 40%.

V. Conclusions

The proposed window-based trigger scheme offers options for rendering existing trigger rules less conservative. This is achieved by relaxing the requirement of a decrease of the Lyapunov function at each time step to a decrease in tendency. It is shown that the choice of time windows has no effect on the upper bound of the chosen Lyapunov function, which is in particular, the same as for the non-windowed basic trigger rules in the literature. As our simulations show, the scheme is not only able to reduce the update rate from an open-loop point of view, but can also reduce the number of required transmissions in closed-loop if the time windows are chosen conveniently. Additionally the standard deviation is reduced.

To involve a larger class of multi-agent systems, future work includes the generalization of the MAS to directed graphs, to double-integrator, general linear, or nonlinear dynamics.

Further possible extensions of the scheme might, e.g., include the consideration of network unreliabilities such as package dropouts and communication delays. Finally, it is of interest to apply the window-based scheme to modified consensus protocols, such as the one presented in [13], [14].

Appendix

A. Proof of Theorem 1

We denote in this proof $f_i(k) = f_i(k, x_k)$. To prove (24), we split up the sequence of sampling times t_0, ..., t_k into time windows defined by the sequence $\{T^i_m\}_{m\in\mathbb{N}_0}$ and evaluate the trigger rule for each window separately. First rewrite (22) in a compact form to obtain $\sum_{\bar{k}=k^i_m+1}^{k}(1-\sigma)^{k-\bar{k}}e_i(t_{\bar{k}})^2 < \sum_{\bar{k}=k^i_m+1}^{k}(1-\sigma)^{k-\bar{k}}f_i(\bar{k})$. For each element T^i_j+h, ..., T^i_{j+1} within the windows $j = 0$, ..., $m-1$ the trigger rule ensures

$$\sum_{\bar{k}=k^i_j+1}^{k^i_{j+1}}(1-\sigma)^{k^i_{j+1}-\bar{k}}e_i(t_{\bar{k}})^2 < \sum_{\bar{k}=k^i_j+1}^{k^i_{j+1}}(1-\sigma)^{k^i_{j+1}-\bar{k}}f_i(\bar{k}). \quad (40)$$

Recursively applying (18) results in

$$\begin{aligned} V(x_k) - (1-\sigma)^k V(x_0) &\leq \|P_e\| \sum_{\bar{k}=0}^{k-1}(1-\sigma)^{k-\bar{k}-1}\|e(t_{\bar{k}})\|^2 \\ &= \|P_e\| \sum_{\bar{k}=0}^{k-1}(1-\sigma)^{k-\bar{k}-1}\sum_{i=1}^{n} e_i(t_{\bar{k}})^2 \\ &= \|P_e\| \sum_{i=1}^{n}\sum_{\bar{k}=0}^{k-1}(1-\sigma)^{k-\bar{k}-1}e_i(t_{\bar{k}})^2. \quad (41) \end{aligned}$$

Using the sequence $\{T^i_m\}_{m\in\mathbb{N}_0}$, the sum on the left-hand side of (40) can be reformulated by splitting it up into the time windows, on which the trigger rule is operating:

$$\sum_{\bar{k}=0}^{k-1}(1-\sigma)^{k-\bar{k}-1}e_i(t_{\bar{k}})^2 = (1-\sigma)^{k-1}e_i(t_0)^2 \quad (42a)$$

$$+ \sum_{j=0}^{m-1}\sum_{\bar{k}=k^i_j+1}^{k^i_{j+1}} (1-\sigma)^{k-\bar{k}-1}e_i(t_{\bar{k}})^2 \quad (42b)$$

$$+ \sum_{\bar{k}=k^i_m+1}^{k-1} (1-\sigma)^{k-\bar{k}-1}e_i(t_{\bar{k}})^2. \quad (42c)$$

The summation (42b) captures all past time windows up to T^i_m whereas (42c) accounts for the time from $k = k^i_m + 1$ up to the current time step k. Due to the design of the trigger rule, each element in (42b) can be bounded using (40) as

$$\begin{aligned} &\sum_{\bar{k}=k^i_j+1}^{k^i_{j+1}} (1-\sigma)^{k-\bar{k}-1}e_i(t_{\bar{k}})^2 \\ &= (1-\sigma)^{k-k^i_{j+1}-1}\sum_{\bar{k}=k^i_j+1}^{k^i_{j+1}} (1-\sigma)^{k^i_{j+1}-\bar{k}}e_i(t_{\bar{k}})^2 \\ &\overset{(40)}{<} (1-\sigma)^{k-k^i_{j+1}-1}\sum_{\bar{k}=k^i_j+1}^{k^i_{j+1}} (1-\sigma)^{k^i_{j+1}-\bar{k}}f_i(\bar{k}) \\ &= \sum_{\bar{k}=k^i_j+1}^{k^i_{j+1}} (1-\sigma)^{k-\bar{k}-1}f_i(\bar{k}). \quad (43) \end{aligned}$$

We do the same with the sum in (42c), yielding

$$\sum_{\bar{k}=k^i_m+1}^{k-1}(1-\sigma)^{k-\bar{k}-1}e_i(t_{\bar{k}})^2 < \sum_{\bar{k}=k^i_m+1}^{k-1}(1-\sigma)^{k-\bar{k}-1}f_i(\bar{k}). \quad (44)$$

Recall that at t_0 an event is triggered for all agents. Thus, $e_i(t_0) = 0$ and $(1-\sigma)^{k-1}e_i(t_0)^2 \leq (1-\sigma)^{k-1}f_i(0)$. Inserting this together with (43) and (44) in (42) and pulling the sums together, we end up with $\sum_{\bar{k}=0}^{k-1}(1-\sigma)^{k-\bar{k}-1}e_i(t_{\bar{k}})^2 < \sum_{\bar{k}=0}^{k-1}(1-\sigma)^{k-\bar{k}-1}f_i(\bar{k})$. Inserting this in (41) yields $V(x_k) < (1-\sigma)^k V(x_0) + \|P_e\| \sum_{i=1}^{n}\sum_{\bar{k}=0}^{k-1}(1-\sigma)^{k-\bar{k}-1}f_i(\bar{k})$ and with the definition (20) we obtain the desired result (24).

B. Auxiliary results

Lemma 1: Assume $V_k < a^k V_0 + \sum_{\bar{k}=0}^{k-1} a^{k-\bar{k}-1}bV_{\bar{k}}$ for all $k\in\mathbb{N}$, $a\in\mathbb{R}_+$ and $b\in\mathbb{R}$. Then

$$V_k < (a+b)^k V_0. \quad (45)$$

Proof: First note that for $b = 0$, the proof of Lemma 1 is trivial. Thus, we subsequently assume that $b \neq 0$. The proof is performed by induction. For $k = 1$ the statement (45) directly follows from the assumption. Now assume that (45) holds for all $k \leq \hat{k}$, $\hat{k} \geq 1$. In order to show $V_{\hat{k}+1} < (a+b)^{\hat{k}+1}V_0$ use the assumption and insert (45) for $\bar{k} \leq \hat{k}$. $V_{\hat{k}+1} < a^{\hat{k}+1}V_0 + \sum_{\bar{k}=0}^{\hat{k}} a^{\hat{k}-\bar{k}}bV_{\bar{k}} \overset{(45)}{<} a^{\hat{k}+1}V_0 + \sum_{\bar{k}=0}^{\hat{k}} a^{\hat{k}-\bar{k}}b(a+b)^{\bar{k}}V_0 = a^{\hat{k}+1}V_0 + a^{\hat{k}}bV_0\sum_{\bar{k}=0}^{\hat{k}}\left(\frac{a+b}{a}\right)^{\bar{k}}$. Similar as in the proof of Corollary 1, we use $\sum_{\bar{k}=0}^{\hat{k}} r^{\bar{k}} = \frac{1-r^{\hat{k}+1}}{1-r}$ to obtain

$$\begin{aligned} V_{\hat{k}+1} &< a^{\hat{k}+1}V_0 + b\frac{a^{\hat{k}+1}}{a}\frac{1-\left(\frac{a+b}{a}\right)^{\hat{k}+1}}{1-\frac{a+b}{a}}V_0 \quad (46) \\ &= (a+b)^{\hat{k}+1}V_0. \end{aligned}$$

We observe by induction that (45) holds for all k. ■

References

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

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

[3] W. Ren and Y. Cao, *Distributed Coordination of Multi-agent Networks*. Springer, 2011.

[4] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10-11, pp. 1002–1033, 2007.

[5] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[6] X. Meng and T. Chen, "Event based agreement protocols for multi-agent networks," *Automatica*, vol. 49, no. 7, pp. 2125–2132, 2013.

[7] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.

[8] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[9] G. Xie, H. Liu, L. Wang, and Y. Jia, "Consensus in networked multi-agent systems via sampled control: Fixed topology case," in *2009 American Control Conference*, 2009.

[10] W. Heemels, K. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *51st IEEE Conference on Decision and Control (CDC)*, 2012.

[11] E. Garcia, Y. Cao, H. Yu, P. Antsaklis, and D. Casbeer, "Decentralised event-triggered cooperative control with limited communication," *International Journal of Control*, vol. 86, no. 9, pp. 1479–1488, 2013.

[12] C. Nowzari and J. Cortés, "Distributed event-triggered coordination for average consensus on weight-balanced digraphs," *Automatica*, vol. 68, pp. 237–244, 2016.

[13] H. Zhang, D. Yue, X. Yin, S. Hu, and C. xia Dou, "Finite-time distributed event-triggered consensus control for multi-agent systems," *Information Sciences*, vol. 339, pp. 132–142, 2016.

[14] Y. Dong and J. guo Xian, "Finite-time event-triggered consensus for non-linear multi-agent networks under directed network topology," *IET Control Theory & Applications*, vol. 11, no. 15, pp. 2458–2464, 2017.

[15] X. Meng, L. Xie, Y. C. Soh, C. Nowzari, and G. J. Pappas, "Periodic event-triggered average consensus over directed graphs," in *54th IEEE Conference on Decision and Control (CDC)*, 2015.

[16] X. Yi, K. Liu, D. V. Dimarogonas, and K. H. Johansson, "Distributed dynamic event-triggered control for multi-agent systems," in *56th Conference on Decision and Control (CDC)*, 2017.

[17] P. Yu and D. V. Dimarogonas, "Explicit computation of sampling period in periodic event-triggered multiagent control under limited data rate," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 4, pp. 1366–1378, 2019.

[18] D. Yang, W. Ren, X. Liu, and W. Chen, "Decentralized event-triggered consensus for linear multi-agent systems under general directed graphs," *Automatica*, vol. 69, pp. 242–249, 2016.

[19] S. Linsenmayer, D. V. Dimarogonas, and F. Allgöwer, "Periodic event-triggered control for networked control systems based on non-monotonic lyapunov functions," *Automatica*, vol. 106, pp. 35–46, 2019.

[20] S. H. Mousavi, M. Ghodrat, and H. J. Marquez, "Integral-based event-triggered control scheme for a general class of non-linear systems," *IET Control Theory & Applications*, vol. 9, no. 13, pp. 1982–1988, 2015.

[21] M. T. Nair and A. Singh, *Linear Algebra*. Springer Singapore, 2018.

[22] A. Tanenbaum and D. Wetherall, *Computer networks*. Boston: Pearson Prentice Hall, 2011.

[23] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[24] D. Nešić, A. R. Teel, and E. D. Sontag, "Formulas relating kl stability estimates of discrete-timeand sampled-data nonlinear systems," in *Systems & Control Letters*, vol. 38, pp. 49–60, 1999.

[25] Y. S. Chow and H. Teicher, *Probability Theory*. Springer US, 1978.

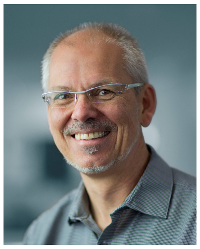
Marc Seidel received his Master's degree in Engineering Cybernetics from the University of Stuttgart, Germany, in 2021. Since then, he has been a research and teaching assistant at the Institute for Systems Theory and Automatic Control and a member of the Graduate School Simulation Technology at the University of Stuttgart. His research interests include Networked Control Systems.

Michael Hertneck received the Master's degree in Mechatronics from the University of Stuttgart, Stuttgart, Germany, in 2019. He has since been a research and teaching assistant at the Institute for Systems Theory and Automatic Control and a member of the Graduate School Simulation Technology at the University of Stuttgart. His research interests include Networked Control Systems with a focus on time- and event-triggered sampling strategies.

Pian Yu received the M.Sc. in Control Science and Control Engineering in 2016 from Wuhan University, China and the Ph.D. in Electrical Engineering in 2021 from KTH Royal Institute of Technology, Sweden. From April 2021, she is a Postdoctoral Researcher at the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology. Her main research interest includes multi-agent systems, motion planning and coordination, event-triggered control, and formal methods.

Steffen Linsenmayer received the Master's degree in Engineering Cybernetics in 2014 and the Ph.D. degree in mechanical engineering in 2021, both from the University of Stuttgart, Stuttgart, Germany. From 2015 to 2021, he was a Research and Teaching Assistant in the Institute for Systems Theory and Automatic Control and a member of the Graduate School Simulation Technology at the University of Stuttgart. Since 2021 he works as a simulation engineer at Robert Bosch GmbH. His research interests include Networked Control Systems.

Dimos V. Dimarogonas was born in Athens, Greece, in 1978. He received the Diploma degree in electrical and computer engineering and the Ph.D. degree in mechanical engineering from the National Technical University of Athens, Athens, Greece, in 2001 and 2007, respectively. Between 2007 and 2010, he held Postdoctoral positions with the Department of Automatic Control, KTH Royal Institute of Technology and the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology. His current research interests include multiagent systems, hybrid systems and control, robot navigation and manipulation, human–robot interaction, and networked control. Prof. Dimarogonas serves in the Editorial Board of the Automatica and IEEE Transactions on Control of Network Systems. He was a recipient of the ERC starting Grant in 2014, the ERC Consolidator Grant in 2019, and the Knut och Alice Wallenberg Academy Fellowship in 2015.



Frank Allgöwer is professor of mechanical engineering at the University of Stuttgart, Germany, and Director of the Institute for Systems Theory and Automatic Control (IST) there.

He is active in serving the community in several roles: Among others he has been President of the International Federation of Automatic Control (IFAC) for the years 2017-2020, Vicepresident for Technical Activities of the IEEE Control Systems Society for 2013/14, and Editor of the journal Automatica from 2001 until 2015. From 2012 until 2020 he served in addition as Vice-president for the German Research Foundation (DFG), which is Germany's most important research funding organization.

His research interests include predictive control, data-based control, networked control, cooperative control, and nonlinear control with application to a wide range of fields including systems biology.