An Introduction to Event-triggered and Self-triggered Control

W.P.M.H. Heemels K.H. Johansson P. Tabuada

***Abstract*— Recent developments in computer and communication technologies have led to a new type of large-scale resource-constrained wireless embedded control systems. It is desirable in these systems to limit the sensor and control computation and/or communication to instances when the system needs attention. However, classical sampled-data control is based on performing sensing and actuation periodically rather than when the system needs attention. This paper provides an introduction to event- and self-triggered control systems where sensing and actuation is performed when needed. Event-triggered control is reactive and generates sensor sampling and control actuation when, for instance, the plant state deviates more than a certain threshold from a desired value. Self-triggered control, on the other hand, is proactive and computes the next sampling or actuation instance ahead of time. The basics of these control strategies are introduced together with a discussion on the differences between state feedback and output feedback for event-triggered control. It is also shown how event- and self-triggered control can be implemented using existing wireless communication technology. Some applications to wireless control in process industry are discussed as well.**

I. Introduction

In today's standard control textbooks, *e.g.* [1], [2], periodic control is presented as the only choice for implementing feedback control laws on digital platforms. However, questions related to periodic vs aperiodic implementations have gone in and out of fashion since feedback control loops started being implemented on computers. Some early examples include the following references [3], [4], [5], [6], [7], [8].

This paper is concerned with the latest wave of the periodic vs aperiodic control debate or, as we prefer to call it, periodic vs event-based control. There are two fundamental reasons for the resurgence of this debate in the last 5 or 6 years. The first, is the increasing popularity of (shared) wired and wireless networked control systems that raise the importance of explicitly addressing energy, computation, and communication constraints when designing feedback control loops. Event-based control offers some clear advantages with respect to periodic control when handling these constraints but it also introduces some new theoretical and practical problems. The second reason is the appearance of two papers [9], [10] that highlighted some of the advantages of event-based control and motivated the development of the first *systematic designs* of event-based implementations of stabilizing feedback control laws, *e.g.,*[11], [12], [13], [14]. Since then, several researchers have improved and generalized these results and alternative approaches have appeared. In the meantime, also so-called self-triggered control [15] emerged. Event-triggered and self-triggered control systems consists of two elements, namely, a feedback controller that computes the control input, and a triggering mechanism that determines when the control input has to be updated again. The difference between event-triggered control and self-triggered control is that the former is reactive, while the latter is proactive. Indeed, in event-triggered control a triggering condition based on current measurements is continuously monitored and when violated, an event is triggered. In self-triggered control the next update time is precomputed at a control update time based on predictions using previously received data and knowledge on the plant dynamics.

Most of the existing event-triggered control approaches employ the assumption that the full state information is available, even though in most practical situations this assumption is violated. As the separation principle does not hold in general for event-triggered control systems [16], output-based schemes are indeed hard to design and optimize. Recently, some work on output-based event-triggered control emerged and in this paper we will discuss a few of these solutions. We distinguish these solutions based on their time sets and the adopted control laws.

The nature of the time sets will differentiate existing event-triggered control strategies based on the use of either continuous-time or discrete-time controllers/event-triggering mechanisms. This differentiation does not require much explanation, although one comment is relevant. In many works studying discrete-time event-triggered control schemes, the plant is also considered to be of a discrete-time nature. Clearly, this allows to obtain direct parallels between some of the continuous-time event-triggered control approaches and the discrete-time counterparts. However, it is more interesting to take a more "sampled-data"-like approach on discrete-time event-triggered control schemes in the sense that the behavior is studied when this controller interacts with a *continuous-time* plant. Also stability and performance properties have to be considered then in a continuous-time setting. In this setup the closed loop consists of a continuous-time plant

Maurice Heemels is with the Hybrid and Networked Systems group, Department of Mechanical Engineering, Eindhoven Univeristy of Technology, the Netherlands; Karl H. Johansson is with ACCESS Linnaeus Center, Royal Institute of Technology, Sweden; Paulo Tabuada is with Department of Electrical Engineering, University of California, Los Angeles, CA, USA. E-mails: m.heemels@tue.nl,kallej@kth.se,tabuada@ee.ucla.edu

The work of the Maurice Heemels was partially supported by the Dutch Science Foundation (STW) and the Dutch Organization for Scientific Research (NWO) under the VICI grant "Wireless controls systems: A new frontier in automation". The work of Karl Johansson was partially supported by the Knut and Alice Wallenberg Foundation and the Swedish Research Council. Maurice Heemels and Karl Johansson were also supported by the European 7th Framework Programme Network of Excellence under grant HYCON2-257462. The work of Paulo Tabuada was partially supported by NSF awards 0834771 and 0953994.

and an event-triggered control strategy, which is a discrete-time controller operating in a periodic time-triggered manner. In this context, sometimes the term *periodic event-triggered control* is used, see [17], [18].

Regarding the differentiation on the nature of the output-based control law, we distinguish the approaches based on whether there is an observer or not. In the former cases we will talk about an observer-based control law, and in the latter about a direct output-based law. Based on an observer, different strategies can be implemented. In most cases the observer reconstructs the plant state using solely event-based information in the sense that it only applies innovation steps exploiting received measurements at the event times, although some schemes also exploit the information present at synchronous (time-triggered) instants of time at which no events occur. The latter is, for instance, the case in [19], [20], [21]. In fact, [19] states literally "absence of an event is however information that can be used by the observer," see also [22]. The observer-based schemes can still be categorized further based on the fact if the corresponding event-triggering conditions use information of the observer such as the estimated state, as, e.g., in [20], [23], [24] or not [21], [25]. The former schemes typically run the observer at the sensor side using essentially all measurements available, while at the controller side a predictor-like structure also produces a state estimate, which is based on only sporadically received information from the sensor system (including the observer) based on the event-triggering mechanisms. Often the event-triggering mechanisms provide new information to the predictor when the difference between the state estimate of the predictor deviates too much from the state estimate available in the observer.

Next to providing an introductory overview on some of the works in the area, the main objective of this paper is to emphasize the key ideas in three different aspects of aperiodic control: the basics on and differences between event-triggered and self-triggered control, the use of output-based event-triggered control, and event-based control over wireless communication networks. The outline of the paper is as follows. The basic ideas of event-triggered control are introduced in Section II. Self-triggered control is discussed in Section III. Output-based event-triggered control is surveyed in Section IV followed by exemplary approaches for continuous-time direct output-based control (Section V) and discrete-time observer-based control (Section VI). Event-triggered transmission in wireless control systems is discussed in Section VII. Finally, concluding remarks are given in SectionVIII.

II. Event-triggered Control

In this section we introduce the main ideas of event-triggered control following [12]. In order to simplify the presentation we consider the linear case only even though the results in [12] were originally developed for nonlinear systems.

We start with a linear plant

$$\frac{d}{dt}x_p = A_p x_p + B_p u, \quad x_p \in \mathbb{R}^{n_p}, u \in \mathbb{R}^{n_u} \tag{1}$$

and assume that a linear feedback control law

$$u = Kx_p \tag{2}$$

has been designed rendering the ideal closed-loop system

$$\frac{d}{dt}x_p = A_p x_p + B_p K x_p \tag{3}$$

asymptotically stable, *i.e.*, rendering the real part of the eigenvalues of $A_p + B_p K$ negative. The question that now arises is how to implement the feedback control law (2) on a digital platform. One possibility is to periodically recompute (2) and keep the actuator values constant in between the periodic updates. Rather than using time (the period) to determine when (2) should be recomputed, we are interested in recomputing (2) only when performance is not satisfactory. One way to define performance is to use a Lyapunov function for the ideal closed-loop system (3). Such a Lyapunov function, that we denote by $V(x_p) = x_p^T P x_p$ for some symmetric and positive-definite matrix P, satisfies

$$\frac{d}{dt}V(x_p(t)) = \frac{\partial V}{\partial x_p}(A_p + B_p K)x_p = -x_p^T Q x_p, \tag{4}$$

where Q is guaranteed to be positive-definite. Since the time derivative of V along the solution of the closed-loop system is negative, V decreases. Moreover, the rate at which V decreases is specified by the matrix Q. If we are willing to tolerate a *slower* rate of decrease, we would require the solution of an event-triggered implementation to satisfy the *weaker* inequality

$$\frac{d}{dt}V(x_p(t)) \leq -\sigma x_p^T Q x_p \tag{5}$$

for some $\sigma \in [0, 1[$. Note that by choosing $\sigma = 1$ (5) becomes (4), while for $\sigma < 1$ (5) prescribes a *slower* rate of decrease for V.

The requirement (5) suggests that we only need to recompute (2) and update the actuator signals when (5) is about to be violated, *i.e.*, when (5) becomes an equality. In order to write such an equality in a convenient manner, we assume the inputs to be held constant in between the successive recomputations of (2). This is often referred to in the literature as sample-and-hold and can be formalized as

$$u(t) = u(t_k) \quad \forall t \in [t_k, t_{k+1}[, \ k \in \mathbb{N}, \tag{6}$$

where the sequence $\{t_k\}_{k\in\mathbb{N}}$ represents the instants at which (2) is re-computed and the actuator signals are updated. We refer to these instants as the *triggering* times or *execution* times. For simplicity, we assume that the process of collecting sensor measurements, re-computing (2) and updating the actuators can be done in zero[1] time. We now

[1]This idealized assumption describes the fact that in many implementations this time is much smaller than the time elapsed between the instants t_k and t_{k+1}. This assumption is not essential and the interested reader can consult [12] for a specific extension of the results when this assumption does not hold.

introduce the error e defined by

$$e(t) = x_p(t_k) - x_p(t) \quad \forall t \in [t_k, t_{k+1}[,\ k \in \mathbb{N}.$$

Using this error we express the evolution of the closed-loop system during the interval $[t_k, t_{k+1}[$ by

$$\begin{aligned} \frac{d}{dt}x_p(t) &= A_p x_p(t) + B_p K x_p(t_k) \\ &= A_p x_p(t) + B_p K x_p(t_k) + B_p K(x_p(t) - x_p(t)) \\ &= A_p x_p(t) + B_p K x_p(t) + B_p K e(t). \end{aligned}$$

We can now use this expression to rewrite the time derivative of $V(x_p(t))$ as

$$\begin{aligned} \frac{d}{dt}V(x_p(t)) &= \frac{\partial V}{\partial x_p}(A_p + B_p K)x_p(t) + \frac{\partial V}{\partial x_p}B_p K e(t) \\ &= -x_p^T(t)Q x_p(t) + 2x_p^T(t)P B_p K e(t) \quad (7) \end{aligned}$$

Substituting (7) in inequality (5) we arrive at

$$\begin{bmatrix} x_p^T(t) & e^T(t) \end{bmatrix} \begin{bmatrix} (\sigma-1)Q & PB_pK \\ K^T B_p^T P & 0 \end{bmatrix} \begin{bmatrix} x_p(t) \\ e(t) \end{bmatrix} \leq 0. \quad (8)$$

The triggering times t_k can now be defined as the times at which the following equality holds

$$z^T(t_k)\Psi z(t_k) = 0 \quad (9)$$

with

$$\Psi = \begin{bmatrix} (\sigma-1)Q & PB_pK \\ K^T B_p^T P & 0 \end{bmatrix}, \quad z(t_k) = \begin{bmatrix} x_p(t_k) \\ e(t_k) \end{bmatrix}.$$

The event-triggered implementation of the feedback control law (2) thus consists in keeping the actuator values constant as long as the triggering condition (9) is not satisfied and re-computing (2) and updating the actuators when the triggering condition (9) is satisfied (assuming $z(0)^T\Psi z(0) < 0$). By changing the matrix Ψ we obtain other quadratic triggering conditions. For instance, in [12] the triggering condition $\|e\|^2 \leq \sigma\|x\|^2$ is used that corresponds to the choice

$$\Psi = \begin{bmatrix} -\sigma I & 0 \\ 0 & I \end{bmatrix}$$

where I denotes the identity matrix.

All of these quadratic triggering conditions are designed so as to guarantee a desired rate of decay for the Lyapunov function V through an inequality of the form (5). Hence, asymptotical stability and performance, as measured by the rate of decay of V, are guaranteed by the different choices of Ψ in the triggering condition (9). Furthermore, the triggering times implicitly defined by (9) will not be equidistant, in general, and thus event-triggered implementations result in *aperiodic* control. In fact, the set of triggering times $\{t_k\}_{k\in\mathbb{N}}$ can be formally defined by

$$t_0 = 0, \quad t_{k+1} = \inf\{t \in \mathbb{R} \mid t > t_k \ \wedge\ z^T(t)\Psi z(t) = 0\}.$$

Since these instants are only known at execution time, the scheduling of energy, computation, and communication resources for event-triggering control becomes a very challenging problem. Moreover, the implicit definition of the times raises the question of the existence of a lower bound $\tau^* > 0$ for $t_{k+1} - t_k$, $k \in \mathbb{N}$. The largest value τ^* for which $t_{k+1} - t_k \geq \tau^*$ holds for all $k \in \mathbb{N}$ along all trajectories of interest, is called the *minimal inter-event time*. If the minimal inter-event time is zero, then an event-triggered implementation will require faster and faster updates and thus cannot be implemented on a digital platform. It was shown in [12] that such minimal inter-event time is guaranteed to exist even in the nonlinear case under suitable assumptions. For linear plants and linear state-feedback controllers, the minimum inter-event time is always guaranteed to exist.

Theorem 1 ([12]): Consider the linear plant (1) and linear feedback control law (2) rendering the closed-loop system (3) asymptotically stable. For any triggering condition (9) with $\sigma \in [0, 1[$ there exists $\tau^* \in \mathbb{R}^+$ such that $t_{k+1} - t_k \geq \tau^*$ for every $k \in \mathbb{N}$.

However, in case output-feedback controllers are used in a similar setup, the minimal inter-event time might be zero and accumulations of event-times occur (Zeno behaviour). This was pointed out in [26], see also Section V below.

III. Self-triggered Control

Event-triggered implementations require the constant monitoring of a triggering condition. For some applications this is a reasonable assumption, *e.g.*, when we can use dedicated hardware for this purpose. Unfortunately, this is not always the case and the related concept of *self-triggered* control is an alternative that can be used in such cases. The term self-triggered control was coined by [15] in the context of real-time systems. A self-triggered implementation of the feedback control law (2) has for objective the computation of the actuator values *as well as* the computation of the next instant of time at which the control law should be recomputed. When dealing with linear plants and linear controllers we can leverage the closed-form expression of the trajectories to develop self-triggered implementations as we discuss next.

A. ISS self-triggered implementations

The results in this section are based on [27], [28], [29]. We start by extending the linear model (1) with disturbances

$$\frac{d}{dt}x_p = A_p x_p + B_p u + B_w w, \quad (10)$$

where $w \in \mathbb{R}^{n_w}$ is the disturbance. It is well know that if the control law (2) renders the closed-loop system (3) asymptotically stable then, in the presence of disturbances, the closed-loop system

$$\frac{d}{dt}x_p = (A_p + B_p K)x + B_w w \quad (11)$$

is so-called exponentially input-to-state stable.

Definition 1 (EISS and GES): The system (11) is said to be *exponentially input-to-state stable* (EISS) if there exist $\lambda \in \mathbb{R}^+$, $\kappa \in \mathbb{R}^+$, and $\gamma \in \mathbb{R}^+$ such that for any $w \in \mathcal{L}_\infty$ and any $x(0) = x_0 \in \mathbb{R}^{n_x}$ it holds for the corresponding trajectory that

$$\|x(t)\| \leq \kappa\|x_0\|e^{-\lambda t} + \gamma\|w\| \quad (12)$$

for all $t \in \mathbb{R}_0^+$. When this inequality holds for $w = 0$, the system (3) is said to be *globally exponentially stable* (GES).

We now describe a self-triggered implementation of (2) that results in an EISS closed-loop system. A self-triggered implementation of the linear stabilizing controller (2) for the plant (10) is given by a map $\Gamma : \mathbb{R}^{n_x} \to \mathbb{R}^+$ determining the triggering time t_{k+1} as a function of the state $x(t_k)$ at the time t_k, *i.e.*, $t_{k+1} = t_k + \Gamma(x(t_k))$. If we denote by τ_k the inter-execution time $\tau_k = t_{k+1} - t_k$, we have $\tau_k = \Gamma(x(t_k))$.

Once the map Γ is defined, the expression *self-triggered closed-loop system* refers to the system (10) and control law (2) implemented in a sample-and-hold manner (6) with triggering times t_{k+1} given by $t_0 = 0$ and $t_{k+1} = t_k + \Gamma(x(t_k))$.

In Section II we formalized the notion of performance based on the time derivative of a Lyapunov function. In this section we directly consider the time evolution of a Lyapunov function of the form $V(x) = (x^T P x)^{\frac{1}{2}}$. If, for the ideal closed-loop system (3) we have

$$V(x(t)) \leq V(x_0)e^{-\lambda_o t}, \quad \forall t \in \mathbb{R}_0^+ \ \forall x_0 \in \mathbb{R}^{n_x}, \qquad (13)$$

then we would like to enforce the weaker inequality

$$V(x(t)) \leq V(x(t_k))e^{-\lambda\tau}, \quad \forall \tau \in [0, t_{k+1} - t_k[\ \forall x_0 \in \mathbb{R}^{n_x} \qquad (14)$$

for the self-triggered implementation in the absence of disturbances ($w = 0$) where $\lambda \in [0, \lambda_o[$. If we denote by $h_c : \mathbb{R}^{n_x} \times \mathbb{R}_0^+ \to \mathbb{R}$ the map

$$h_c(x(t_k), t) = V(x(t)) - V(x(t_k))e^{-\lambda\tau},$$

then the inequality in (14) can be expressed as $h_c(x(t_k), \tau) \leq 0$. Since no digital implementation can check $h_c(x(t_k), \tau) \leq 0$ for all $\tau \in [0, t_{k+1} - t_k[$, we consider instead the following discrete-time version of h_c based on a sampling time $\Delta \in \mathbb{R}^+$

$$h_d(x(t_k), n) := h_c(x(t_k), n\Delta) \leq 0,$$

for all $n \in \left[0, \left\lceil \frac{t_{k+1} - t_k}{\Delta} \right\rceil \right[$ and for all $k \in \mathbb{N}$. This condition results in the following self-triggered implementation where we use $N_{\min} := \lfloor \tau_{\min}/\Delta \rfloor$, $N_{\max} := \lfloor \tau_{\max}/\Delta \rfloor$, and $\tau_{\min}$ and $\tau_{\max}$ are design parameters.

Definition 2: The map $\Gamma_d : \mathbb{R}^n \to \mathbb{R}^+$ is defined by

$$\begin{aligned} \Gamma_d(x) &:= \max\{\tau_{\min}, n(x)\Delta\} \text{ with} \\ n(x) &:= \max_{n \in \mathbb{N}} \{n \leq N_{\max} | h_d(x, s) \leq 0, s = 0, \ldots, n\} \end{aligned}$$

for $x \in \mathbb{R}^n$.

Using this definition of Γ_d, a self-triggered implementation of the linear stabilizing controller (2) for plant (10) is prescribed.

Note that the role of $\tau_{\min}$ and $\tau_{\max}$ is to enforce explicit lower and upper bounds, respectively, for the inter-execution times of the controller. The upper bound enforces robustness of the implementation and limits the computational complexity.

Remark 1: Linearity of (10) and (2) enables us to compute h_d^2 as a quadratic function of $x(t_k)$. Moreover, through a Veronese embedding we can implement the self-triggered policy described in Definition 2 so that its computation has space complexity $q\frac{n_x(n_x+1)}{2}$ and time complexity $q + (2q+1)\frac{n_x(n_x+1)}{2}$ where $q := N_{\max} - N_{\min}$. For reasons of space we omit these details. They can be found in [27].

The following result establishes EISS of the proposed self-triggered implementation.

Theorem 2: Let $\tau^* \in \mathbb{R}^+$ be defined by

$$\tau^* = \inf\{\tau \in \mathbb{R}^+ : \det M(\tau) = 0\}$$

where

$$\begin{aligned} M(\tau) &:= C(e^{F^T\tau}C^T P C e^{F\tau} - C^T P C e^{-\lambda\tau})C^T, \\ F &:= \begin{bmatrix} A_p + B_p K & B_p K \\ -A_p - B_p K & -B_p K \end{bmatrix}, \ C := [I \ 0]. \end{aligned}$$

If $\tau_{\min} \leq \tau^*$, the self-triggered implementation in Definition 2 renders the self-triggered closed-loop system EISS.

Remark 2: When implementing self-triggered policies on digital platforms several issues related to real-time scheduling need to be addressed. For a discussion of some of these issues we refer the readers to [30]. Here, we describe the minimal computational requirements for the proposed self-triggered implementation under the absence of other tasks. Let us assume that the computation delays dominate the measurement and actuation delays, as is the case sometimes in practice. The computation of Γ is divided in two steps: a preprocessing step performed once per execution, and a running step performed n times when computing $h_d(x, n)$. The preprocessing step computes a matrix used to evaluate h_d and has time complexity $(n_x^2 + n_x)/2$. The running step consists of testing the inequality $h_d(x, n) \leq 0$ has time complexity $n_x^2 + n_x$. If we denote by τ_c the time it takes to execute an instruction in a given digital platform, the self-triggered implementation can be executed if:

$$\frac{3}{2}(n_x^2 + n_x)\tau_c \leq \tau_{\min}, \ (n_x^2 + n_x)\tau_c \leq \Delta.$$

The first inequality imposes a minimum processing speed for the digital platform while the second equality establishes a lower bound for the choice of Δ.

Remark 3: Theorem 2 only guarantees EISS of the self-triggered implementation. In [29] the readers can find more detailed results explaining how the constants κ and γ appearing in the definition of EISS depend on the continuous dynamics (1), the control law (2), and the design parameters $\tau_{\min}$ and $\tau_{\max}$.

We refer the interested reader to [27] and [28] for numerical examples illustrating the proposed technique and the guarantees it provides. An example comparing this implementation with the implementation described in the next section appears in Section III-C.

B. Minimum attention implementations

In Section III-A we started with a linear controller and constructed a self-triggered implementation. Although the self-triggered implementation was based on the controller

and the system dynamics, the controller was designed in oblivion of the implementation details. In this section we take a step towards the *co-design* of the control laws and its implementations. We consider a different formulation of the minimum attention control problem introduced in [31]:

Given the state of the system, compute a set of inputs that guarantee a certain level of performance while maximizing the next time at which the input needs to be updated.

In this formulation of the minimum attention control problem we interpret attention as the inverse of the time elapsed between consecutive input updates.

The approach we will follow is based on the ideas in [32] and consists in computing all the inputs $u \in \mathbb{R}^{n_u}$ satisfying inequality (13), which we reproduce here in a version suitable for our needs:

$$V\left(e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu\,\mathrm{d}\tau\right) \leq e^{-\lambda t}V(x_0). \qquad (15)$$

We now make the important observation that by using ∞-norm based Lyapunov functions, the computation of all the inputs satisfying (15) reduces to a feasibility problem with linear constraints and thus can be efficiently done online. Specifically, we take V to be a control Lyapunov function of the form

$$V(x) = \|Px\|_\infty$$

with $P \in \mathbb{R}^{m\times n_x}$ having rank n_x and where $\|\cdot\|_\infty$ denotes the infinity norm, i.e., $\|x\|_\infty = \max_{i\in\{1,2,\ldots,n_x\}} |x_i|$. Similarly to Section III-A we define the map h_c by

$$\begin{aligned} h_c(x(t_k),u,\tau) &= \Big\|Pe^{A\tau}x(t_k) \\ &\quad + \int_{t_k}^{t_k+\tau} Pe^{A(\tau-s)}Bu\,\mathrm{d}s\Big\|_\infty \\ &\quad -e^{\lambda\tau}\|Px(t_k)\|_\infty. \end{aligned} \qquad (16)$$

We can now observe that the constraint $h_c(x(t_k),u,\tau)\leq 0$, which appears in (16), is equivalent to

$$\left|\left[Pe^{A\tau}x(t_k) + \int_{t_k}^{t_k+\tau} Pe^{A(\tau-s)}Bu\,\mathrm{d}s\right]_i\right| - e^{-\lambda\tau}\|Px(t_k)\|_\infty \leq 0,$$

for all $i \in \{1,\ldots,m\}$, which is equivalent to $\overline{h_c}(x(t_k),u,\tau)\leq 0$, where

$$\begin{aligned} \overline{h_c}(x(t_k),u,\tau) &= \begin{bmatrix} Pe^{A\tau}x(t_k) + P\int_{t_k}^{t_k+\tau} e^{A(\tau-s)}\mathrm{d}sBu \\ -Pe^{A\tau}x(t_k) - P\int_{t_k}^{t_k+\tau} e^{A(\tau-s)}\mathrm{d}\tau Bu \end{bmatrix} \\ &\quad -e^{-\lambda\tau}\|Px(t_k)\|_\infty \begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} \end{aligned} \qquad (17)$$

and the inequality is assumed to be taken element-wise, which results in $2n_x$ linear scalar constraints for u.

Since the inequality $\overline{h_c}(x(t_k),u,\tau)\leq 0$ cannot be checked for all $\tau\in\mathbb{R}_0^+$ we work, similarly as in Section III-A, with its discrete analogue

$$\overline{h_d}(x(t_k),u,n) := \overline{h_c}(x(t_k),u,n\Delta) \leq 0.$$

We note that while a self-triggered implementation of a linear control law is specified by the map $\Gamma : \mathbb{R}^{n_x} \to \mathbb{R}^+$ determining the next execution time (as the control law is already given), a minimum attention implementation addressing the co-design problem requires the map Γ as well as the map

$$\Omega : \mathbb{R}^{n_x} \to 2^{\mathbb{R}^{n_u}}$$

specifying any input $u\in\Omega(x)$ that can be used during the next $\Gamma(x)$ units of time, i.e.,

$$u(t) = u(t_k) \in \Omega(x(t_k)), \quad t\in[t_k,t_{k+1}[\qquad (18a)$$

$$t_{k+1} = t_k + \Gamma(x(t_k)) \qquad (18b)$$

with $t_0 := 0$. In a concrete implementation one uses additional criteria, *e.g.*, minimum energy, to select a single input among all the possible inputs given by the set $\Omega(x(t_k))$.

Algorithm 1 computes both Ω and Γ.

Input: $P\in\mathbb{R}^{m\times n_x}$ defining an ∞-based control Lyapunov function and $x(t_k)$
Output: $\Gamma(x(t_k))$ and $\Omega(x(t_k))$
$n := 0$;
$\Omega_0 := \mathbb{R}^{n_u}$;
while $\Omega_n \neq \varnothing$ *and* $n < N_{\max}$ **do**
  $n := n+1$;
  $\Omega_n := \Omega_{n-1} \cap \{u\in\mathbb{R}^{n_u} \mid \overline{h_d}(x(t_k),u,n)\leq 0\}$;
end
if $\Omega_n = \varnothing$ **then**
  $\Omega(x(t_k)) := \Omega_{n-1}$;
  $\Gamma(x(t_k)) := (n-1)\Delta$;
else
  $\Omega(x(t_k)) := \Omega_n$;
  $\Gamma(x(t_k)) := n\Delta$;
end

Algorithm 1: Algorithm providing Ω and Γ for a minimum attention implementation.

The correctness of Algorithm 1 is guaranteed by the following result whose proof can be found in [32].

Theorem 3 ([32]): The minimum attention implementation defined by Γ and Ω computed by Algorithm 1 renders the minimum attention closed-loop system consisting of (1) and (18) GES.

Remark 4: Since verifying that $\Omega_n \neq \emptyset$ as specified in Algorithm 1 is a feasibility test for linear constraints, the algorithm can be efficiently implemented online using existing solvers for linear programs.

Remark 5: Theorem 3 only states GES of the minimum attention implementation. In [32] the readers can also find more detailed results explaining how the constants κ and λ appearing in the definition of GES (i.e., (12) for $w=0$) depend on the continuous dynamics (1) and the choice of Δ. Reference [32] also discusses how ∞-norm based Lyapunov functions can be constructed. A study of the robustness properties of this implementation, *e.g.* EISS, has not yet appeared in the literature.

C. Illustrative example

In this section, we illustrate the self-triggered and minimum attention implementations using a well-known example from the networked control systems literature, see, e.g., [33], consisting of a linearized model of a batch reactor. The linearized batch reactor is given by (1) with

$$\left[\begin{array}{c|c} A & B \end{array}\right] = \left[\begin{array}{cccc|cc} 1.380 & -0.208 & 6.715 & -5.676 & 0 & 0 \\ -0.581 & -4.290 & 0 & 0.675 & 5.679 & 0 \\ 1.067 & 4.273 & -6.654 & 5.893 & 1.136 & -3.146 \\ 0.048 & 4.273 & 1.343 & -2.104 & 1.136 & 0 \end{array}\right].$$

We consider the linear control law (2) with

$$K = \begin{bmatrix} 0.0360 & -0.5373 & -0.3344 & -0.0147 \\ 1.6301 & 0.5716 & 0.8285 & -0.2821 \end{bmatrix}, \tag{19}$$

rendering the eigenvalues of $A + BK$ real, distinct and smaller than or equal to -2. In order to compare the self-triggered with the minimum attention aproach we use in both case the ∞-norm based Lyapunov function $V = \|Px\|_\infty$ with

$$P = \begin{bmatrix} 0.4730 & 0.7092 & 1.0979 & -0.7885 \\ -1.2568 & 1.7787 & -2.1320 & 2.1234 \\ -1.7781 & -0.1852 & -1.4692 & 0.3769 \\ -0.5042 & 1.5041 & -0.5112 & 2.4252 \end{bmatrix}.$$

Reference [32] offers more details on how P was computed.

To implement Algorithm 1 in MATLAB, we use the routine `polytope` of the MPT-toolbox [34], to handle the sets $\Omega(x(t_k))$.

When we the response of the plant is simulated with the minimum attention implementation for the initial condition $x(0) = [1\ 0\ 1\ 0]^\top$, we can observe that the closed-loop system is indeed GES, see Fig. 1(a) and Fig. 1(c). The self-triggered implementation also renders the closed-loop system GES as can be seen from Fig. 1(b) and Fig. 1(c). Note that the decay rates for both implementations are comparable as expected. However, when we compare the resulting inter-execution times as depicted in Fig. 1(d), we observe that the minimum attention implementation yields much larger inter-execution times than the self-triggered implementation. This can be explained from the fact that the former solves a co-design problem thereby optimizing current values of the control inputs with the objective to maximize the next execution time. The considered self-triggered approach does not as it has a prescribed (emulation-based) control law.

D. Other approaches to self-triggered control

Other approaches to self-triggered control have appeared in the literature. In [35], [36] the authors consider linear stabilizing control laws for linear systems enforcing a desired $\mathcal{L}_2$ gain on the closed-loop system. In a related manner to the implementations discussed in Section III preserving EISS and GES, the authors of [35], [36] propose self-triggered implementations preserving $\mathcal{L}_2$-gain stability. The interesting trade-off in this case is how much the $\mathcal{L}_2$-gain degrades as the number of inter-executions is reduced with respect to a periodic implementation. Self-triggered techniques for nonlinear control systems are reported in [37], [38] based on the notion of homogeneity and isochronous manifolds. Although the approach is based on homogeneity, it is shown how it is possible to make any smooth control system homogeneous by increasing the dimension of the state space by one. A different approach based on polynomial approximations of nonlinear systems is described in [39]. All these approaches consider implementations where the input remains constant in between re-computations of the control law. An alternative approach, based on using a model of the plant at the actuator, is reported in [40], where it is shown that non-constant inputs further reduce the number of messages that need to be sent from the controller to the actuator. Reference [41] extends the results in [29] from state feedback to output feedback. Finally [42] applies self-triggered to a coverage control problem for robotic networks thereby reducing the required communication between robots.

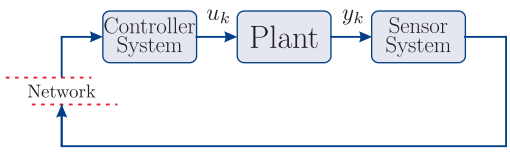


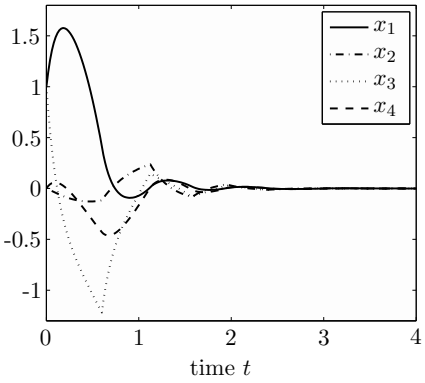
Fig. 2: Configuration with shared network only in the sensor-to-controller (s-c) channel.

IV. Output-based Event-Triggered Control

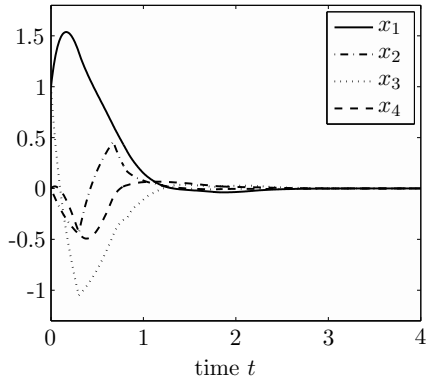
The approaches on event-triggered and self-triggered control presented previously were all based on full state feedback, although in practice the full state is often not available for feedback. In fact, in the introduction the importance of developing output-based event-triggered controllers was already indicated. Moreover, a first categorization of the existing output-based event-triggered control schemes was already provided based on their time sets (discrete-time vs continuous-time) and adopted control law (observer-based or not). In this section, we start by discussing the literature on continuous-and discrete-time output-based event-triggered control with and without observer in a bit more detail. After that two exemplary approaches will be presented.

A. Continuous-time observer-based event-triggered control

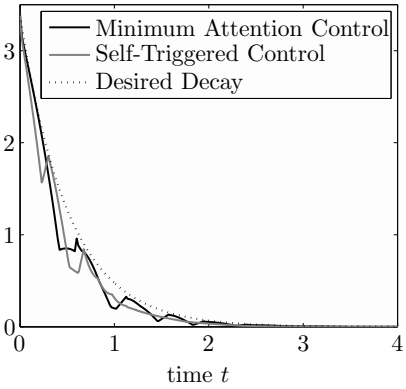
In [19] one of the first observer-based event-triggered control loops are proposed in the context of continuous-time systems, although the analysis and examples in the end focus on the situation where the full state information is available. A formal analysis can be found in the more recent work [43], which extends the work in [44] that assumed availability of the full state. The work in [43] focuses on continuous-time plants perturbed by a bounded disturbance and measured outputs affected by bounded measurement noise. A signal generator (contained in the controller system in the setup depicted in Fig. 2) produces the control input implemented at the actuators using a predictor that runs the unperturbed model equations in closed loop with a state feedback control law, in which the state variable is updated with state estimates received from the (more accurate) observer situated at the sensor system in Fig. 2. The sensor system has a copy of



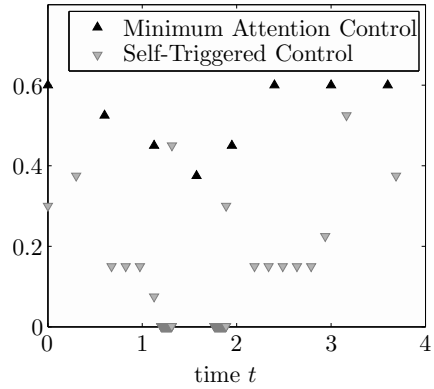
(a) Evolution of the states of the plant using the minimum attention implementation.



(b) Evolution of the states of the plant using the self-triggered implementation.



(c) The decay of the Lyapunov function using the minimum attention and the self-triggered implementations.



(d) The inter-execution times using the minimum attention and the self-triggered implementations.

Fig. 1: Comparison between the minimum attention and the self-triggered implementations.

the predictor. Only when the difference between the state estimate in the predictor and the observer exceeds in norm an absolute threshold the estimated state in the observer is sent to the controller. The analysis of this scheme shows that a stable behavior of the event-based control loop can still be guaranteed in the sense of ultimate boundedness of the plant's state. Moreover, it is shown that the maximum communication frequency within the control loop is bounded, *i.e.*, the minimal inter-event time is strictly positive. The size of the absolute threshold can be used to balance the maximum communication frequency and the size of the ultimate bound.

Event-based state estimation is considered in [21]. In that paper a state estimator adopts a hybrid update scheme in the sense that updates take place both when an event occurs that triggers the transmission of new measurements to the estimator (asynchronous times), as well as when a periodic timer expires (synchronous times). In the latter case the principle that "absence of an event is however information that can be used by the observer" [19] is used. More specifically, events are triggered only when the monitored output variable leaves a bounded set (possibly depending on latest transmitted measurement). Hence, receiving no information at a synchronous time instant indicates that the output is still in this bounded set, which is information that can be used to guarantee bounded estimation error covariances. In fact, in [21] this is formally shown based on a sum-of-Gaussians approach that is used to obtain a computationally tractable algorithm. An example of integrating this event-based state estimator with a periodically time-triggered control algorithm is provided in [25]. In [25] the triggering condition does not use the estimated state as, for instance, in [43]. For time-stamped measurements, one can also adopt a time-varying discrete-time Kalman filter approach to obtain a good estimate of the state. However, note that such a scheme does not exploit potentially valuable information contained in the absence of events.

B. Continuous-time direct event-triggered control

In contrast to the results discussed previously, next event-triggered control is considered without any intermediate processing of measurements by an observer or filter. One work belonging to this category is [45], which studies linear systems without disturbances and measurement noise and with a finite number of control actions. The method is based on hysteretic quantization. The transmission of output measurements is triggered by reaching the next quantization level. A consequence, in case of single outputs, is that only one bit has to be transmitted in order to inform the control system about the quantization level reached (assuming that

the previous value is stored at the controller). The paper proposes two systematic output feedback control design strategies. The first is an emulation-based strategy starting from an analog controller, and the second strategy is a direct design that drives the plant state to the origin in finite time after a total transmission of $2n+2$ bits, where n is the order of the plant.

In [26] it is studied how event-triggered control strategies tailored to static state-feedback control laws along the lines of Section II can be extended to output-based dynamical controllers using both centralized and decentralized event-triggering mechanisms. One of the problems identified in [26] was that using the output-based extensions of the event-triggering mechanisms adopted in [12] based on relative thresholding can result in accumulations of event times (Zeno behaviour) and thus a zero minimal inter-event time. Using a mixed event-triggering mechanism, a strictly positive minimum inter-event time could be guaranteed for output-based event-triggered control, while still guaranteeing ultimate boundedness and $\mathcal{L}_\infty$-performance. This work exploited impulsive models [46], [47] for describing the closed-loop behavior, which resulted in less conservative stability conditions compared to the original work [12]. See Section V for more details.

C. Discrete-time observer-based event-triggered control

In [11] a discrete-time control problem is considered in which the communication resources are considered to be scarce. As such, the objective is to reduce the number of communications by using more computations. The paper uses an emulation-based approach in the sense that a well-functioning output-based controller is available (assuming a standard time-triggered periodic implementation). The main idea of the proposed event-triggered control strategy is the use of a state estimator framework such that all nodes have identical estimators and thus identical estimator states. The estimated values of the remote outputs are used in the feedback control. Every sample time, the controller at the ith node compares the estimate of the ith output to its true values. If the difference is greater than a predefined threshold, the true value is communicated to the other nodes. When there is a communication from ith node, the estimators in all nodes update their states to reflect the current actual value of the system outputs or states. As a consequence, the error between the estimated data used in the control algorithm and the actual values is always bounded by a threshold, which can be chosen by the control designer to balance closeness to the original time-triggered closed-loop system responses on the one hand and the communication usage on the other hand. This bound on the error can be used to obtain BIBO stability conditions. A drawback of the scheme is that it uses a global estimator in each node, which does not scale to large systems.

In [23], [16] the problem of output-based event-triggered control in discrete-time is considered from an optimal control perspective in line with the classical Linear Quadratic Gaussian (LQG) setup. It is shown in [16] that such a set-up can lead to a stochastic control problem with a dual effect, so that the optimal event-trigger and controller are hard to find. In [23] an emulation-based approach is considered in which the observer at the sensor system and the local observer at the controller are fixed by minimizing the error covariance conditioned on the received information. Based on the appended LQG cost the problem addressed in [23] is to synthesize the ETMs in the s-c and c-a channels in a (sub)optimal manner.

In [24] an output-based event-triggered control scheme is proposed using model-based triggering schemes in both the sensor-to-controller and the controller-to-actuator communication channels. A predictive control technique is adopted in the controller-to-actuator channel. By sending control packets containing model-based predictions of future control values and only transmitting new control packets when these predictions deviate from the current control values computed in the control system (according to relative bounds), significant savings can be obtained compared to a basic zero-order hold strategy. For the sensor-to-controller channel, the triggering mechanism is based on the difference between the state estimate of a Luenberger observer running in the sensor system with the state estimate of a predictor (called "the local observer" in [23]) running in both the sensor and controller systems. If this difference gets too large, then the estimate of the Luenberger observer is transmitted to the controller system that updates the state estimate of its predictor. LMI-based tools are provided for closed-loop stability and ℓ_2-gain analysis. See Section VI below for more details on the setup.

Recently, in [48] also an output-based scheme exploiting observer-like structures for discrete-time linear systems was proposed for tracking of references signals generated by an exosystem.

D. Discrete-time direct event-triggered control

In [10], [13] output-based PID controllers are considered without the consideration of an observer or estimator. Both these approaches use a timer to avoid problems with a zero minimum inter-event time. In [10] the event detector is truly time-triggered, while in [13] a time regularization is adopted by requiring that after an event at least a fixed amount of time no new event is generated. However, [10] does not provide any analytical results, while [13] only provides them for state-based event-triggered control strategies. Recently, such results were obtained in [18]. Interestingly, these results apply to both centralised and decentralised event-triggering mechanisms, and they provide stability and $\mathcal{L}_2$-gain guarantees of the closed-loop system in continuous time, even though the event-triggered control strategy is a discrete-time controller operating in a periodic time-triggered manner. In fact, as already mentioned in the introduction, the term *periodic event-triggered control* is used in this context, cf. [17] [18].

V. Continuous-time Direct Event-triggered Control

In this section, we present an exemplary event-triggered control problem based on continuous-time output-based controllers and model the event-triggered control system as an impulsive system. This particular setup is based on [26], but connections to related methods will be mentioned using this exposition.

A. Problem Formulation

Let us consider a linear time-invariant (LTI) plant given by

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}x_p = A_p x_p + B_p \hat{u} + B_w w, \\ \quad y = C_p x_p, \end{cases} \tag{20}$$

where $x_p \in \mathbb{R}^{n_p}$ denotes the state of the plant, $\hat{u} \in \mathbb{R}^{n_u}$ the input applied to the plant, $w \in \mathbb{R}^{n_w}$ an unknown disturbance and $y \in \mathbb{R}^{n_y}$ the output of the plant. The plant is controlled using a continuous-time LTI controller given by

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}x_c = A_c x_c + B_c \hat{y}, \\ \quad u = C_c x_c, \end{cases} \tag{21}$$

where $x_c \in \mathbb{R}^{n_c}$ denotes the state of the controller, $\hat{y} \in \mathbb{R}^{n_y}$ the input of the controller, and $u \in \mathbb{R}^{n_u}$ the output of the controller. We assume that the controller is designed to render (20) and (21) with $y(t) = \hat{y}(t)$ and $u(t) = \hat{u}(t)$, for all $t \in \mathbb{R}_+$, asymptotically stable.

Here, we consider the case where the controller is implemented in a sampled-data fashion, which causes $y(t) \neq \hat{y}(t)$ and $u(t) \neq \hat{u}(t)$ for almost all $t \in \mathbb{R}_+$. In particular, we study decentralised event-triggered control which means that the outputs of the plant and controller are grouped into N nodes and the outputs of node $i \in \{1, \dots, N\}$ are only sent at the transmission instants $t^i_{k_i}$, $k_i \in \mathbb{N}$. Hence, at transmission instant $t^i_{k_i}$, node i transmits its respective entries in y and u, and the corresponding entries in $\hat{y}$ and $\hat{u}$ are updated accordingly, while the other entries in $\hat{y}$ and $\hat{u}$ remain the same. Such constrained data exchange can be expressed as

$$\hat{v}^+(t^i_{k_i}) = \Gamma_i v(t^i_{k_i}) + (I - \Gamma_i)\hat{v}(t^i_{k_i}), \tag{22}$$

in which $v = [y^\top \ u^\top]^\top$, $\hat{v} = [\hat{y}^\top \ \hat{u}^\top]^\top$, and

$$\Gamma_i = \mathrm{diag}(\gamma_i^1, \dots, \gamma_i^{n_y+n_u}), \tag{23}$$

for all $i \in \{1, \dots, N\}$. In between transmissions, we use a zero-order hold, i.e.,

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{v}(t) = 0, \quad \text{for all} \quad t \in \mathbb{R}_+ \backslash \big(\textstyle\bigcup_{i=1}^N \{t^i_{k_i} \,|\, k_i \in \mathbb{N}\} \big). \tag{24}$$

In (23), the elements γ_i^j, with $i \in \{1, \dots, N\}$ and $j \in \{1, \dots, n_y\}$, are equal to 1 if plant output y_j is in node i and are 0 elsewhere, the elements $\gamma_i^{j+n_y}$, with $i \in \{1, \dots, N\}$ and $j \in \{1, \dots, n_u\}$, are equal to 1 if controller output u_j is in node i and are 0 elsewhere. We assume that for each $j \in \{1, \dots, n_y + n_u\}$, it holds that $\sum_{i=1}^N \gamma_i^j > 0$, i.e., we assume that each sensor and actuator is at least in one node. Furthermore, we assume that at time $t = 0$, it holds that $\hat{v}(0) = v(0)$. This can be accomplished by transmitting all sensor and actuator data at the time the system is deployed.

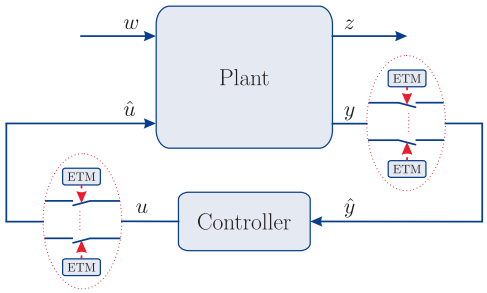


Fig. 3: Control system block diagram with indication of the event-triggering mechanism (ETM).

In a conventional sampled-data implementation, the transmission times are distributed equidistantly in time and are the same for each node, meaning that $t^i_{k_i+1} = t^i_{k_i} + h$, for all $k_i \in \mathbb{N}$ and all $i \in \{1, \dots, N\}$, and for some constant transmission interval $h > 0$, and that $t^i_k = t^j_k$, for all $k \in \mathbb{N}$ and all $i, j \in \{1, \dots, N\}$. In event-triggered control, however, these transmissions are orchestrated by an event-triggering mechanism, as is shown in Fig. 3, which in this case is decentralised. We consider a decentralised event-triggering mechanism that invokes transmissions of node data when the difference between the current values of outputs and their previously transmitted values becomes too large in an appropriate sense. In particular, the event-triggering mechanism considered in this section results in transmitting the outputs of the plant or the controller in node $i \in \{1, \dots, N\}$ at times $t^i_{k_i}$, satisfying

$$t^i_{k_i+1} = \inf\big\{ t > t^i_{k_i} \,|\, \|e_{\mathcal{J}_i}(t)\|^2 = \sigma_i \|v_{\mathcal{J}_i}(t)\|^2 + \varepsilon_i \big\}, \tag{25}$$

and $t^i_0 = 0$, for some $\sigma_i, \varepsilon_i \geqslant 0$. In these expressions, $e_{\mathcal{J}_i}$ and $v_{\mathcal{J}_i}$ denote the subvectors formed by taking the elements of the signals e and v, respectively, that are in the set $\mathcal{J}_i = \{j \in \{1, \dots, n_y + n_u\} \mid \gamma_i^j = 1\}$, and

$$e(t) = \hat{v}(t) - v(t) \tag{26}$$

denotes the error induced by the event-triggered implementation of the controller at time $t \in \mathbb{R}_+$. Note that $\mathcal{J}_i$ is the set of indices of sensors/actuators corresponding to node i. Hence, the event-triggering mechanism (25), which is based on local information available at each node, is such that when for some $i \in \{1, \dots, N\}$, it holds that $\|e_{\mathcal{J}_i}(t)\|^2 = \sigma_i \|v_{\mathcal{J}_i}(t)\|^2 + \varepsilon_i$, *i.e.*, the norm of the error induced by the event-triggered implementation of the signals in node i becomes large for the first time, node i transmits its corresponding signal $v_{\mathcal{J}_i}(t)$ in $v(t)$ and, the signal $\hat{v}(t)$ is updated according to (22). This implies that $e^+(t^i_{k_i}) = (I - \Gamma_i)e(t^i_{k_i})$ and thus $e^+_{\mathcal{J}_i}(t^i_{k_i}) = 0$. Using this update law, and the aforementioned assumption that $\hat{v}(0) = v(0)$, yielding $e(0) = 0$, we can observe that the error induced by the event-triggered control scheme satisfies

$$\|e_{\mathcal{J}_i}(t)\|^2 \leqslant \sigma_i \|v_{\mathcal{J}_i}(t)\|^2 + \varepsilon_i, \tag{27}$$

for all $t \in \mathbb{R}^+$ and all $i \in \{1, \ldots, N\}$.

The question that arises now is how to determine σ_i and ε_i for all $i \in \{1, \ldots, N\}$, such that the closed-loop event-triggered system is stable in an appropriate sense and a certain level of disturbance attenuation is guaranteed, while the number of transmissions of the outputs of the plant and the controller is small. Note that for $\varepsilon_i = 0$, $i \in \{1, \ldots, N\}$, the event-triggering conditions in (25) can be seen as an extension of the event-triggering mechanism of [12] for output-based controllers, and for $\sigma_i = 0$, $i \in \{1, \ldots, N\}$, it is equivalent to the event-triggering mechanism of [49], [45], [50]. As such, the event-triggering mechanism in (25) unifies two earlier proposals.

B. An impulsive system formulation

In this section, we reformulate the event-triggered control system as an impulsive system, *e.g.*, [46], [47], of the form

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\bar{x} = \bar{A}\bar{x} + \bar{B}w, \quad \text{when } \bar{x} \in \mathcal{C} \tag{28a}$$

$$\bar{x}^+ = \bar{G}_i\bar{x}, \qquad \text{when } \bar{x} \in \mathcal{D}_i,\ i \in \{1, \ldots, N\}, \tag{28b}$$

where $\bar{x} \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ denotes the state of the system and $w \in \mathbb{R}^{n_w}$ an external disturbance. The flow and the jump sets are denoted by $\mathcal{C} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{D}_i \subseteq \mathbb{R}^{n_x}$, $i \in \{1, \ldots, N\}$, respectively, and $\mathcal{X} = \mathcal{C} \cup (\bigcup_{i=1}^{N} \mathcal{D}_i)$. Note that the transmission times $t^i_{k_i}$, $k_i \in \mathbb{N}$, as in (25), are now related to the event times at which the jumps of $\bar{x}$, according to (28b) for $i \in \{1, \ldots, N\}$, take place.

To arrive at a system description of the event-triggered control system (20), (21), (22), (24), and (25) of the form (28), we combine (20), (21), (22), (24) and (26), and define $\bar{x} := [x^\top\ e^\top]^\top \in \mathbb{R}^{n_x}$, where $x = [x_p^\top\ x_c^\top]^\top$ and $n_x := n_p + n_c + n_y + n_u$, yielding the flow dynamics of the system

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\bar{x} = \underbrace{\begin{bmatrix} A+BC & B \\ -C(A+BC) & -CB \end{bmatrix}}_{=:\bar{A}} \bar{x} + \underbrace{\begin{bmatrix} E \\ -CE \end{bmatrix}}_{=:\bar{B}} w, \tag{29}$$

in which

$$A = \begin{bmatrix} A_p & 0 \\ 0 & A_c \end{bmatrix},\ B = \begin{bmatrix} 0 & B_p \\ B_c & 0 \end{bmatrix},\ C = \begin{bmatrix} C_p & 0 \\ 0 & C_c \end{bmatrix},\ E = \begin{bmatrix} B_w \\ 0 \end{bmatrix}. \tag{30}$$

The system continuously flows as long as the event-triggering conditions are not met, *i.e.*, as long as (27) holds for all $i \in \{1, \ldots, N\}$, which can be reformulated as $\bar{x} \in \mathcal{C}$, with

$$\mathcal{C} = \{\bar{x} \in \mathbb{R}^{n_x} \mid \bar{x}^\top Q_i \bar{x} \leqslant \varepsilon_i\ \forall\ i \in \{1, \ldots, N\}\}, \tag{31}$$

and

$$Q_i = \begin{bmatrix} -\sigma_i C^\top \Gamma_i C & 0 \\ 0 & \Gamma_i \end{bmatrix}, \tag{32}$$

because $\bar{x}^\top Q_i \bar{x} \leqslant \varepsilon_i$ is equivalent to $\|\Gamma_i e(t)\|^2 \leqslant \sigma_i \|\Gamma_i v(t)\|^2 + \varepsilon_i$, as in (27). As mentioned before, when node i transmits its data, a reset according to $e^+ = (I - \Gamma_i)e$ occurs, while x remains the same, *i.e.*, $x^+ = x$, see (22). This can be expressed as

$$\bar{x}^+ = \underbrace{\begin{bmatrix} I & 0 \\ 0 & I - \Gamma_i \end{bmatrix}}_{=:\bar{G}_i} \bar{x}, \tag{33}$$

for all $\bar{x} \in \mathcal{D}_i$, $i \in \{1, \ldots, N\}$, in which

$$\mathcal{D}_i = \{\bar{x} \in \mathbb{R}^{n_x} \mid \bar{x}^\top Q_i \bar{x} = \varepsilon_i\}, \tag{34}$$

according to (25). Combining (29), (31), (33) and (34) yields an impulsive system of the form (28).

C. Analysis methods and discussion

The available analysis techniques given in [26] build upon the impulsive system framework [47] with a focus on global asymptotic stability of sets $\mathcal{A}$ containing the origin in the interior (in absence of disturbances w) and $\mathcal{L}_\infty$-performance of the closed-loop system. As such, in case of absence of disturbances a form of practical stability, or ultimate boundedness, is obtained. The conditions guaranteeing global asymptotic stability of sets and upperbounds on the $\mathcal{L}_\infty$-gain of the system from disturbance w to performance output $z = \bar{C}\bar{x} + \bar{D}w$ are given in terms of LMIs. We refer the interested reader to [26] for the details and the precise statements of the results. To provide some insights in the consequence of the results, we note that the feasibility of the LMIs is related to the choice of the relative gains σ_i, $i \in \{1, \ldots, N\}$, in the event-triggering conditions (25), but is *not* affected by the choice of the absolute thresholds ε_i, $i \in \{1, \ldots, N\}$. Hence, once the LMIs are feasible, practical stability (for $w = 0$) and upper bounds on the $\mathcal{L}_\infty$-gain are guaranteed. The 'size' of the set $\mathcal{A}$ (ultimate bound) (when $w = 0$), is affected by both σ_i and ε_i. However, after having a feasible set of LMIs guaranteeing set stability and finite $\mathcal{L}_\infty$-gains, the parameters ε_i provide full control to adjust the size of the set $\mathcal{A}$. As we can see from (27), this will affect the number of events, enabling the designer to make trade-offs between the size of the set $\mathcal{A}$ (related to the ultimate bound of x as $t \to \infty$ for $w = 0$) and the number of transmissions over each communication channel. Indeed, larger ε_i, $i \in \{1, \ldots, N\}$, result in fewer events, and thus fewer transmissions, but in a larger set $\mathcal{A}$ (*i.e.*, a larger ultimate bound), when $w = 0$. In fact. if ε_i, $i \in \{1, \ldots, N\}$, all approach zero, we have that $\mathcal{A} \to \{0\}$. Hence, the set $\mathcal{A}$ can be made arbitrary small (at the cost of more transmissions). The naive choice to take $\varepsilon_i = 0$, for all $i \in \{1, \ldots, N\}$, seems appealing as it would yield $\mathcal{A} = \{0\}$. However, this might result in zero minimum inter-event times (Zeno behaviour) as Example 2 in [26] illustrates. In some cases, e.g., state-feedback controlled system with centralised event triggering as discussed in [12], a strictly positive minimum inter-event time can be guaranteed even for $\varepsilon_1 = 0$, and we have that $\mathcal{A} = \{0\}$ is globally asymptotically stable, see also Theorem 1. In fact, in this case also finite $\mathcal{L}_p$-gains for $p < \infty$ can be given, see Remark III.7 in [26] and Remark IV.3 in [51].

Here, we discussed an impulsive system formulation (28) with subsequent LMI-based stability and performance analysis. This leads to less conservative values for ε_i, σ_i, $i \in \{1, \ldots, N\}$ guaranteeing stability than the "perturbed system" approach given in [12], as is formally proven in [26]. The benefit of adopting the impulsive system formulation can be explained by the fact that the impulsive system truly describes the behaviour of the event-triggered control

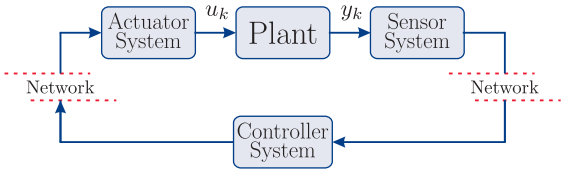


Fig. 4: Networked control configuration.

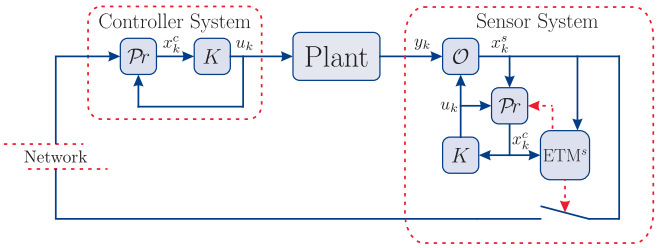


Fig. 5: Observer-based PETC strategy with only s-c ETM.

system as it includes the dynamics for the error e induced by the event-triggered implementation. Besides the exact modelling of the error dynamics as above, also the fact that LMI-based formulations are used is beneficial as this allows to use an optimisation-based procedure to find better values for σ_i and ε_i guaranteeing stability and specific levels of $\mathcal{L}_\infty$-performance. Larger values of σ_i and ε_i result in larger minimum inter-event time, see (25). More recently, the impulsive system framework was also used for other event-triggered and self-triggered controller setups, see, *e.g.*, [52].

VI. Discrete-time Observer-based Event-triggered Control

Just as in the previous section, we present an exemplary event-triggered control problem in this section but now for discrete-time observer-based controllers. This particular setup follows [24].

A. Problem Formulation

In [24], the networked control configuration shown in Fig. 4 is studied, in which the plant is given by a discrete-time linear time-invariant model of the form

$$\mathcal{P}: \quad \begin{cases} x_{k+1} = Ax_k + Bu_k + Ew_k \\ y_k = Cx_k, \end{cases} \tag{35}$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, $w_k \in \mathbb{R}^{n_w}$ and $y_k \in \mathbb{R}^{n_y}$ denote the state, control input, disturbance and measured output, respectively, at discrete time instant $k \in \mathbb{N}$. The sensors of the plant transmit their measurements to the controller, and the controller transmits the control data to the actuators over a shared, possibly wireless, network, for which communication and energy resources are limited. For this reason, it is desirable to reduce the transmissions over the sensor-to-controller and controller-to-actuator channels as much as possible, while still guaranteeing desirable closed-loop behavior. Hence, the problem is now to design smart sensor, controller and actuator systems for the setup in Fig. 4 such that this objective is realized.

B. An observer-based strategy

In this section, we present a possible solution as given in [24] for the problem formulated in Section VI-A in the context of Fig. 2 in which the controller-to-actuator channel is removed.

The smart sensor system in Fig. 2 consists of a Luenberger observer $\mathcal{O}$, a predictor $\mathcal{P}r$ and an event-triggering mechanism ETM^s that determines when information should be transmitted to the controller system, see Fig. 5. The Luenberger observer is given by

$$\mathcal{O}: \quad x_{k+1}^s = Ax_k^s + Bu_k + L(y_k - Cx_k^s) \tag{36}$$

in which x_k^s denotes the estimated state at the sensor system at time $k \in \mathbb{N}$, and the matrix L is a suitable observer gain. The predictor $\mathcal{P}r$ is given by

$$\mathcal{P}r: \quad x_{k+1}^c = \begin{cases} Ax_k^c + Bu_k, \text{ when } x_k^s \text{ is not sent} \\ Ax_k^s + Bu_k, \text{ when } x_k^s \text{ is sent.} \end{cases} \tag{37}$$

Finally, the event-triggering mechanism is given at time $k \in \mathbb{N}$ by the condition

$$ETM^s: \quad x_k^s \text{ is sent} \quad \Leftrightarrow \quad \|x_k^s - x_k^c\| > \sigma_s \|x_k^s\|, \tag{38}$$

where $\sigma_s \geq 0$ is a design parameter. Before explaining the functioning of $\mathcal{P}r$ and ETM^s in more detail, it is convenient to introduce also the controller system. The controller system consists of a copy of the predictor $\mathcal{P}r$, and a controller gain K, see Fig. 5. In fact, the control signal is given by

$$u_k = \begin{cases} Kx_k^c, \text{ when } x_k^s \text{ is not sent} \\ Kx_k^s, \text{ when } x_k^s \text{ is sent.} \end{cases} \tag{39}$$

As the sensor system also runs a copy of the predictor $\mathcal{P}r$ (both initialized at the same initial estimate), the sensor system is aware of the estimate x_k^c the controller system has, and, consequently, can determine u_k to compute the next state estimate x_{k+1}^s according to (36). Clearly, the estimate x_k^s of the observer is typically better than the estimate x_k^c of the predictor, as the observer has access to all measurements, while the predictor only receives sporadic updates.

The rationale now is that if the sensors detects at $k \in \mathbb{N}$ that the estimate x_k^s of the Luenberger observer (36) deviates significantly from the estimate x_k^c, *i.e.*, $\|x_k^s - x_k^c\| > \sigma_s \|x_k^s\|$ as in (38), the estimate x_k^s is transmitted to the controller, and corresponding updates of the estimate x_{k+1}^c (cf. the second case in (37)) and the control signal u_k as in (39) are made. Hence, as long as $\|x_k^s - x_k^c\|$ is sufficiently small, no transmissions between the sensor and controller systems are needed.

This observer-based strategy can provide similar stability and ℓ_2-gain properties, while requiring significantly less transmissions compared to both a standard periodic time-triggered implementation and a baseline event-triggered implementation as in [53], [12], [26], [17], [18]. See the example presented below.

In [24] extensions are provided for the network configuration in Fig. 4 with communication savings both for the sensor-to-controller and the controller-to-actuator communications. In particular, predictive control techniques are adopted computing model-based predictions of future control values, which are sent in one (or more) control packets to the actuator system. Only when these predicted future control values (known in the controller system) deviate from the current control values computed in the controller system, new control packets with future values are transmitted to the actuator system. In this manner, significant savings can be obtained compared to a basic zero-order hold strategy. In [24] also decentralised observer-based controllers and event-triggering mechanisms are presented for large-scale weakly-coupled plants.

Remark 6: Extensions of the observer (36) including disturbance estimators (assuming a suitable linear disturbance model) are possible following the same rationale as in [24]. This extension can enhance further communication savings in the sensor-to-controller channel.

C. Analysis methods and discussion

The analysis of the above mentioned model-based strategies are presented in [24] based on perturbed linear and piecewise linear systems. Based on these modeling paradigms, LMI-based conditions for global exponential stability and guaranteed ℓ_2-gains can be derived. The usage of model-based predictions are quite powerful for the reduction of network resource utilization, as will also be illustrated in the numerical example below. These observations are in line with the results in the networked control literature in which model-based approaches indeed often perform better [54]. It is also worthwhile to mention the connection of the usage of model-based predictions to the work in [19], where the relevance of generalized holds was mentioned, and the work in [44], [43] in which the term signal generator was used based on model-based predictions (although in absence of a resource-constrained controller-to-actuator channel).

D. Illustrative example

In this section, the model-based event-triggered control strategies discussed previously will be illustrated using a time-discretization of the batch reactor example discussed in Section III-C. Proper values for the observer and state feedback gains K and L as in (36) and (39), respectively, are chosen corresponding to the sampling period $h = 0.15$. See [24] for the exact setup.

We compare the model-based event-triggered control scheme with a corresponding periodic time-triggered controller and with the following baseline event-triggered scheme: The baseline implementation uses ideas presented in [53], [12], [26], [17], [18] and leads to a strategy given by the dynamic controller

$$x_{k+1}^c = Ax_k^c + Bu_k + L(\hat{y}_k - Cx_k^c), \tag{40a}$$

a certainty-equivalence control law

$$u_k = Kx_k^c, \tag{40b}$$

and a sensor-to-controller event-triggering mechanism

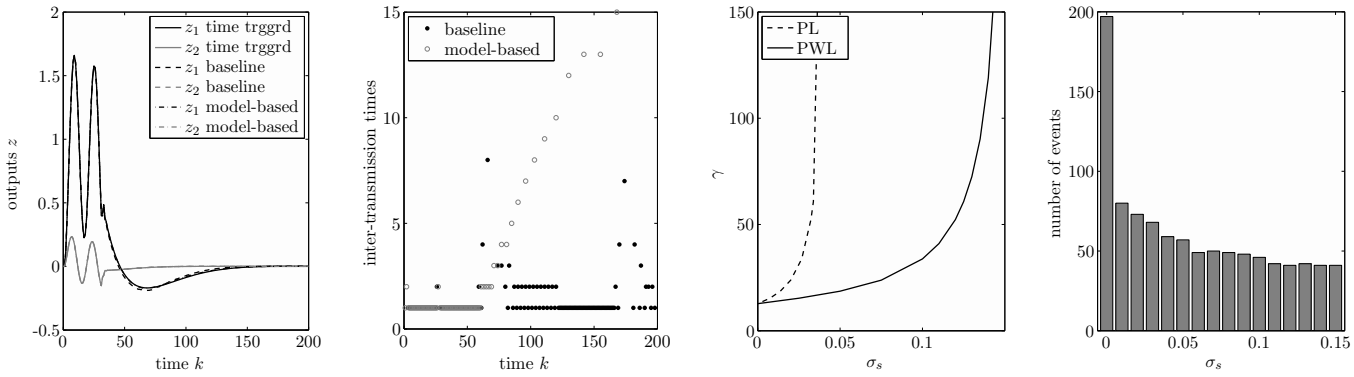
$$\hat{y}_k = \begin{cases} y_k, \text{ when } \|\hat{y}_{k-1} - y_k\| > \sigma_s \|y_k\| \\ \hat{y}_{k-1}, \text{ when } \|\hat{y}_{k-1} - y_k\| \leq \sigma_s \|y_k\|. \end{cases} \tag{41}$$

Hence, in this baseline setup a sensor reading is transmitted to the controller only when the difference between the latest transmitted value and the current sensor reading is large compared to the value of the reading. In addition, the hold strategy $\hat{y}_k = \hat{y}_{k-1}$ is used when no new output measurement is transmitted.

To make a fair comparison between the model-based and the baseline strategies, we select σ_s for both cases such that the guaranteed upper bound γ on the ℓ_2-gain of the resulting closed-loop system satisfies $\gamma = 100$ and use the piecewise linear system approach of [24] to construct the corresponding values for σ_s. This results in $\sigma_s = 0.135$ for the model-based strategy. Using similar techniques for the baseline strategy gives $\sigma_s = 0.0343$. The corresponding periodic time-triggered control strategy results in an (exact) ℓ_2-gain of $\gamma^* = 12.75$.

The response of the performance output z to the initial condition $x_0 = 0$ and the disturbance satisfying $w_k = \sin \frac{3\pi \cdot k}{25} \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$ for $0 \leqslant k \leqslant 30$ and $w_k = 0$ for $k > 30$, for the three strategies is shown in Fig. 6a. We can conclude that all three control strategies show almost indistinguishable responses. However, the number of transmissions that are needed is 200 for periodic time-triggered control, 148 for the baseline strategy and only 41 for the model-based strategy. This demonstrates that the newly proposed model-based event-triggered control strategy needs significantly fewer transmissions than the other two approaches to realize similar responses, at the price of more computations. This is also further illustrated in Fig. 6b showing the inter-transmission times.

We will study now more closely the influence of the parameter σ_s in (38) on the upper bound γ on the ℓ_2-gain of the model-based event-triggered control strategy and the number of transmissions that are generated for the aforementioned initial condition and disturbance, see Fig. 6c and Fig. 6d, respectively. Fig. 6c shows that the upper bound on the ℓ_2-gain increases as σ_s increases, indicating that closed-loop performances degrades as σ_s increases. This figure also shows that the guaranteed upperbounds on the ℓ_2-gain provided by the piecewise linear (PWL) approach are less conservative than the perturbed linear (PL) approach. From Fig. 6d, it can be seen that the increase of the guaranteed ℓ_2-gain, through an increased σ_s, leads to fewer transmissions, which demonstrates the tradeoff that can be made between the closed-loop performance and the number of transmissions. Note that for σ_s approaching zero, the upper bound of the ℓ_2 gain for the model-based PETC strategy approaches $\gamma^* = 12.75$, which is the ℓ_2-gain of the corresponding periodic time-triggered control strategy. This demonstrates, as formally proven in [24], that the ℓ_2-gain of the model-based event-triggered control strategy can approach the ℓ_2-gain of the periodic time-triggered implementation arbitrarily



(a) The evolution of the outputs as function of time k for the time-triggered (trggrd), the baseline and the observer-based strategies.

(b) The inter-transmission times as function of time k for the baseline and the observer-based strategies.

(c) The upper bound on the ℓ_2-gain as function of σ_s for the model-based strategy.

(d) The number of transmissions as function of σ_s for the model-based strategy.

Fig. 6: Comparison of discrete-time observer-based event-triggered control strategies.

close. Interestingly, even for a small σ_s, which only leads to a minor degradation of the closed-loop performance in terms of the ℓ_2-gain, the amount of data transmitted over the network, is already significantly reduced. For instance, starting from a periodic time-triggered observer-based controller, we can set $\sigma_s = 0.01$, which leads to an upper bound on the ℓ_2-gain of the corresponding model-based event-triggered control strategy of 13.77 as guaranteed by the PWL approach, see Fig. 6c, indicating an 8% performance degradation, while the number of transmissions reduce from 200 to 80, see Fig. 6d, which is a reduction of 60%. Of course, it should be noted that the actual savings depend heavily on the considered disturbance (classes). Using disturbance estimators as pointed out in Remark 6 might be beneficial for further reduction of the number of transmissions.

VII. Wireless Event-Triggered Control Systems

In a networked control system, the communication medium is often shared between multiple control loops as indicated in Fig. 7. In Fig. 7 a wireless network connects the sensors with the controllers. For such as system, as already mentioned, it is often desirable to limit the amount of communication, due to that either the transmission is battery powered or the network might get congested. An important approach to efficiently utilize the communication network is to let sensors transmit only when their measurements exceed a certain value, *i.e.*, to apply an event-triggered sampling rule. Other alternatives include having a network manager deciding when each sensor can communicate. That decision can be based on information available in the scheduler. A challenge in general is to limit not only the communication of sensor data, but also limit the need of communication between nodes in order to take communication decisions. Hence, an architecture in which the decision-making is distributed can be desirable, but it is then hard to provide guarantees of no collision or congestion in the network. In practice, it is often reasonable to have a more 'hybrid' approach where some decisions are made centrally by the network manager and some by the individual sensors and controllers.

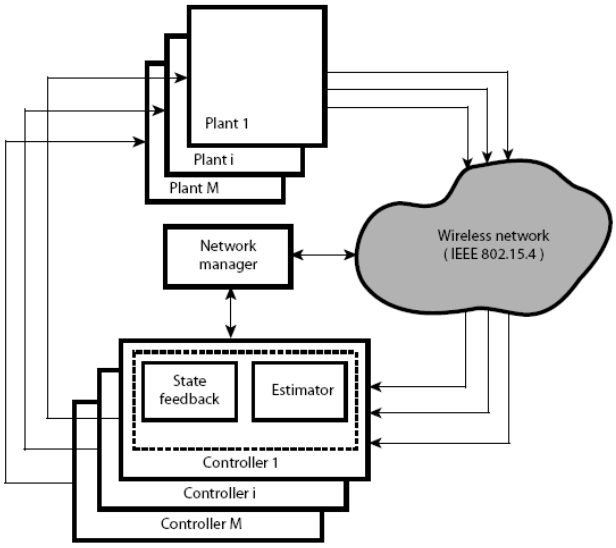


Fig. 7: Wireless control systems.

A. Optimal event-triggered control

It is natural to pose the question if it is possible to design an optimal event-triggering condition for a networked control system as the one depicted in Fig. 7. Unfortunately, the separation principle does not hold for the optimal controller and the optimal scheduler, as the closed-loop system yields a dual effect in general [16]. However, by a suitable filter on the sensor side, it is possible to obtain a control architecture for which certainty equivalence holds. Such an architecture suggests an observer-based controller, but with sacrificed optimality.

In certain situations, it is possible to find the optimal event-based controller. In the work [9] on Lebesgue sampling, a first-order system with an optimal event-based sampling for an impulse controller was considered. For zero-order hold actuation, it was shown in [55] that the optimal threshold in the event-triggering mechanism is time varying. The influence of limited control actions or sensing for optimal event-based control was considered in [56].

An important issue not touched upon previously, but important in a wireless large-scale control system, is the possible occurrence of data drops. For event-triggered sensor communication such data loss might seem to be critical, as fewer transmissions are generated in event-based control systems. In [57], the influence of independent and identically distributed packet drops was considered. It was shown how the control performance deteriorated as the probability of packet drops tends to one. It was also shown that if the (sensor) transmitter receives a (negative) acknowledgement for each packet the (controller) receiver does not receive, then the event-triggering condition can be improved. In particular, the threshold should be lowered each time a packet has been lost, so that the chance of a new transmission is increased. The influence of such acknowledgements on the closed-loop performance can in some cases be explicitly computed.

B. Event-triggered control over wireless networks

It is important to have accurate and efficient communication models of the wireless networks for the design of event-triggered wireless control systems. Here we will briefly discuss how self- and event-triggered control can be adapted to a common wireless network protocol. There is obviously a vast literature on wireless communication, but fewer studies have focused on models suitable for control purposes. Some exceptions include the Markov model developed by Bianchi to study the performance of the communication protocol IEEE 802.11 [58]. Similar Markov models have been developed also for IEEE 802.15.4, which is one of the dominating protocol standard for wireless sensor networks, see, *e.g.*, [59].

The superframe time organization of the slotted IEEE 802.15.4 is shown in Fig. 8. Each superframe Γ_i starts with a beacon. The rest of the superframe is divided into an active and an inactive period. During the inactive period, no device is supposed to transmit so they can save power by being in a so-called sleep mode. The active period is split into a contention access period (CAP) and a collision free period (CFP). During the CAP, the medium access control (MAC) scheme is carrier sense multiple access/collision avoidance (CSMA/CA), where the nodes in the network sense if the channel is busy before transmitting a message. The CAP is used by nodes to send best effort messages, as packet drops can happen due to collision or channel congestion. The CFP is intended to provide real-time guaranteed service, by allocating guaranteed time slots to the nodes in a time division multiple access (TDMA) scheme. Since during the CFP there are no packet losses due to collisions or channel congestion, this mechanism is an attractive period for control tasks.

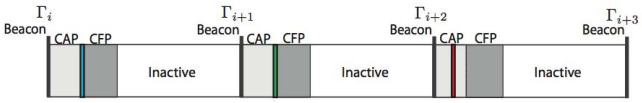


Fig. 8: Superframe time organization of the slotted IEEE 802.15.4 protocol.

It was recently shown that event-triggered and self-triggered control can be implemented over IEEE 802.15.4, see [60]. By allocating a guaranteed time slot within the CFP of a future superframe, it is possible to approximately sample the system according to the time computed by the self-triggered algorithm. In this way, the sensor does not have to transmit until it is suggested by the controller. As disturbances might act on the plant, an event-triggered sampler, which reacts if the sensor measurement starts deviating from its predicted value, needs to be added as well.

For large-scale systems with many sensors within the same wireless range, the guaranteed time slots of the CFP are not enough, and, as a consequence, also the CAP needs to be used. Even if there is contention, event-triggered and self-triggered control can be utilized. Analyzing these schemes under a CSMA/CA MAC is however challenging, as the state of the protocol in general will be correlated with plant state. Various ways to tackle this problem have only recently been considered in the literature, for example, [61].

VIII. Concluding remarks

In this paper the aim was to provide an introductory overview of the fields of event-triggered and self-triggered control. The literature on these classes of aperiodic control is rapidly expanding. In the paper we did not try to cover all of the most recent results in order to be comprehensive, but instead focused on some of the main developments in the latest wave of the periodic versus aperiodic debate. Next to introducing the basics on event-triggered and self-triggered control, the emphasis was on the use of output-based control and implementation issues of event-based control over wireless communication networks. This paper can form a good starting to become acquainted with the research areas of event-triggered and self-triggered control, and in fact several references are provided as suggestions for further reading.

After the enormous growth of the literature on this topic in the past 5 to 6 years, it seems time to take the next steps. Even though many results are currently available, it is fair to say that the system theory for event-triggered and self-triggered control is far from being mature, certainly compared to the vast literature on time-triggered periodic sampled-data control. One possible next step, that is certainly needed, is to develop the necessary system theoretic results underlying complete and efficient (co-)design methodologies for event-triggered and self-triggered control. This should enhance the usage of these control strategies in practical applications. In fact, their validation in practice is an important next step (which will undoubtedly raise new theoretical

questions). Indeed, even though many simulation and experimental results show that event-triggered and self-triggered control strategies are capable of reducing the number of control task executions, while retaining a satisfactory closed-loop performance, see, e.g., [62], [63], [64], [9], [10], [65], [66], [67], [68], [69], the actual deployment of these novel control paradigms in relevant applications is still rather marginal. A possible stimulus for changing this situation, being a third important step, is to demonstrate quantitatively how and when event-triggered and self-triggered control outperform the classical periodic sampled-data control approach. The quantitative evaluation of all these strategies should reflect both *control costs* such as quadratic costs as in LQR control or relevant $\mathcal{L}_p$-gains, and *communication costs* such as average sampling rates, minimal inter-event times, or transmission power. Fair assessments and comparisons are needed helping the practitioners to identify the situations in which these aperiodic control strategies offer benefits that can not be guaranteed by the conventional periodic paradigm.

IX. Acknowledgements

The work described in this paper was the fruit of several students and other collaborators that in many ways lead the way in event-triggered and self-triggered control. We would like to acknowledge the hard work of Adolfo Anta, Jose Araujo, Tijs Donkers, Manuel Mazo Jr., and Chithrupa Ramesh.

References

[1] K. J. Aström and B. Wittenmark, *Computer Controlled Systems*. Prentice Hall, 1997.

[2] G. Franklin, J. Powel, and A. Emami-Naeini, *Feedback Control of Dynamical Systems*. Prentice Hall, 2010.

[3] S. Gupta, "Increasing the sampling efficiency for a control system," *Automatic Control, IEEE Transactions on*, vol. 8, no. 3, pp. 263 – 264, 1963.

[4] A. Liff and J. Wolf, "On the optimum sampling rate for discrete-time modeling of continuous-time systems," *Automatic Control, IEEE Transactions on*, vol. 11, no. 2, pp. 288 – 290, 1966.

[5] G. Bekey and R. Tomovic, "Sensitivity of discrete systems to variation of sampling interval," *Automatic Control, IEEE Transactions on*, vol. 11, no. 2, pp. 284 – 287, 1966.

[6] R. Tomovic and G. Bekey, "Adaptive sampling based on amplitude sensitivity," *Automatic Control, IEEE Transactions on*, vol. 11, no. 2, pp. 282 – 284, 1966.

[7] D. Ciscato and L. Martiani, "On increasing sampling efficiency by adaptive sampling," *Automatic Control, IEEE Transactions on*, vol. 12, no. 3, p. 318, 1967.

[8] J. Mitchell and J. McDaniel, W., "Adaptive sampling technique," *Automatic Control, IEEE Transactions on*, vol. 14, no. 2, pp. 200 – 201, 1969.

[9] K. Aström and B. Bernhardsson, "Comparison of periodic and event based sampling for first order stochastic systems," in *Proc. IFAC World Conf.*, 1999, pp. 301–306.

[10] K.-E. Arzén, "A simple event-based PID controller," in *Preprints IFAC World Conf.*, vol. 18, 1999, pp. 423–428.

[11] J. Yook, D. Tilbury, and N. Soparkar, "Trading computation for bandwidth: Reducing communication in distributed control systems using state estimators," *IEEE Trans. Control Systems Technology*, vol. 10, no. 4, pp. 503–518, 2002.

[12] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *Automatic Control, IEEE Transactions on*, vol. 52, no. 9, pp. 1680 –1685, sept. 2007.

[13] W. Heemels, J. Sandee, and P. van den Bosch, "Analysis of event-driven controllers for linear systems," *Int. J. Control*, vol. 81, pp. 571–590, 2008.

[14] T. Henningsson, E. Johannesson, and A. Cervin, "Sporadic event-based control of first-order linear stochastic systems," *Automatica*, vol. 44, pp. 2890–2895, 2008.

[15] M. Velasco, P. Marti, and J. M. Fuertes, "The self triggered task model for real-time control systems," in *Proceedings of 24th IEEE Real-Time Systems Symposium, Work-in-Progress Session*, 2003.

[16] C. Ramesh, H. Sandberg, L. Bao, and K. H. Johansson, "Dual effect in state-based scheduling of networked control systems," in *American Control Conference*, San Francisco, CA, USA, 2011.

[17] W. Heemels, M. Donkers, and A. Teel, "Periodic event-triggered control: The state-feedback case," in *Prc. IEEE Joint Conf. Decision & Control and European Control Conf.*, 2011.

[18] ——, "Periodic event-triggered control for linear systems," *IEEE Trans. Autom. Control*, to appear.

[19] K. Aström, "Event based control," in *Analysis and Design of Nonlinear Control Systems*. Springer, 2008, pp. 127–148.

[20] A. Molin and S. Hirche, "Structural characterization of optimal event-based controllers for linear stochastic systems," in *Proc. IEEE Conf. Decision and Control*, 2010, pp. 3227–3233.

[21] J. Sijs and M. Lazar, "On event based state estimation," in *Proc. Conf. Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2009, pp. 336–350.

[22] T. Henningsson and K. Aström, "Log-concave observers," in *Proc. of Mathematical Theory of Networks and Systems*, 2006.

[23] L. Li and M. Lemmon, "Weakly coupled event triggered output feedback system in wireless networked control systems," *Procedia Computer Science*, to appear.

[24] W. Heemels and M. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, to appear.

[25] J. Sijs, M. Lazar, and W. Heemels, "On integration of event-based estimation and robust MPC in a feedback loop," in *Proc. Conf. Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2010, pp. 31–41.

[26] M. Donkers and W. Heemels, "Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$-gain and improved and decentralised event-triggering," *IEEE Trans. Autom. Control*, 2012,to appear.

[27] M. Mazo Jr., A. Anta, and P. Tabuada, "On self-triggered control for linear systems: Guarantees and complexity," *European Control Conference*, 2009.

[28] M. Mazo Jr. and P. Tabuada, "Input-to-state stability of self-triggered control systems," *48th IEEE Conference on Decision and Control.*, 2009.

[29] M. Mazo Jr., A. Anta, and P. Tabuada, "An ISS self-triggered implementation of linear controllers," *Automatica*, vol. 46, no. 8, pp. 1310 – 1314, 2010.

[30] A. Anta and P. Tabuada, "On the benefits of relaxing the periodicity assumption for networked control systems over can," *The 30th IEEE Real-Time Systems Symposium.*, 2009.

[31] R. W. Brockett, "Minimum attention control," in *IEEE Conference on Decision and Control*, San Diego, CA, 1997.

[32] M. Donkers, P. Tabuada, and W. Heemels, "On the minimum attention control problem for linear systems: A linear programming approach," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011, pp. 4717 –4722.

[33] G. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Control Syst. Mag.*, vol. 21, no. 1, pp. 57–65, 2001.

[34] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: http://control.ee.ethz.ch/~mpt/

[35] X. Wang and M. Lemmon, "Self-triggered feedback control systems with finite-gain stability," *Automatic Control, IEEE Transactions on*, vol. 54, no. 3, pp. 452 –467, 2009.

[36] ——, "Self-triggering under state-independent disturbances," *Automatic Control, IEEE Transactions on*, vol. 55, no. 6, pp. 1494 –1500, 2010.

[37] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 55, no. 9, pp. 2030–2042, 2010.

[38] ——, "Exploiting isochrony in self-triggered control," *Automatic Control, IEEE Transactions on*, vol. 57, no. 4, pp. 950–962, 2010.

[39] M. Di Benedetto, S. Di Gennaro, and A. D'Innocenzo, "Digital self triggered robust control of nonlinear systems," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011, pp. 1674 –1679.

[40] J. Almeida, C. Silvestre, and A. Pascoal, "Self-triggered state feedback control of linear plants under bounded disturbances," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 7588 –7593.

[41] ——, "Self-triggered output feedback control of linear plants," in *American Control Conference (ACC), 2011*, 2011, pp. 2831 –2836.

[42] C. Nowzari and J. Cortes, "Self-triggered coordination of robotic networks for optimal deployment," in *American Control Conference (ACC), 2011*, 2011, pp. 1039 –1044.

[43] D. Lehmann and J. Lunze, "Event-based output-feedback control," in *19th Mediterranean Conference on Control and Automation*, Corfu, Greece, 2011.

[44] J. Lunze and D. Lehmann, "A state-feedback approach to event-based control," *Automatica*, vol. 46, pp. 211–215, 2010.

[45] E. Kofman and J. Braslavsky, "Level crossing sampling in feedback stabilization under data-rate constraints," in *Proc. IEEE Conf. Decision & Control*, 2006, pp. 4423–4428.

[46] W. Haddad, V. Chellaboina, and S. Nersesov, *Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity, and Control*. Princeton University Press, 2006.

[47] R. Goebel, R. Sanfelice, and A. Teel, "Hybrid dynamical systems," *IEEE Control Syst. Mag.*, vol. 29, pp. 28–93, 2009.

[48] L. Jetto and V. Orsini, "Event-triggered internally stabilizing sporadic control for MIMO plants with non measurable state," in *IFAC World congres*, Milano, Italy, 2011, pp. 10 225–10 230.

[49] P. Otanez, J. Moyne, and D. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proc. American Control Conf.*, 2002, pp. 3015 – 3020.

[50] M. Miskowicz, "Send-on-delta concept: An event-based data-reporting strategy," *Sensors*, vol. 6, pp. 49–63, 2006.

[51] X. Wang and M. Lemmon, "Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability," *IEEE Trans. Autom. Control*, vol. 45, pp. 452–467, 2009.

[52] R. Postoyan, A. Anta, D. Nešić, and P. Tabuada, "A unifying Lyapunov-based framework for the event-triggered control of nonlinear systems," in *Proc. Joint IEEE Conf. Decision and Control and European Control Confernce*, 2011, pp. 2559–2564.

[53] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked systems with data dropouts and delays," *IEEE Trans. Autom. Control*, pp. 586–601, 2011.

[54] L. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Trans. Autom. Control*, vol. 49, pp. 1562–1572, 2004.

[55] M. Rabi, K. H. Johansson, and M. Johansson, "Optimal stopping for event-triggered sensing and actuation," in *IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

[56] T. Henningsson, E. Johannesson, and A. Cervin, "Sporadic event-based control of first-order linear stochastic systems," *Automatica*, vol. 44, no. 11, pp. 2890–2895, 2008.

[57] M. Rabi and K. H. Johansson, "Scheduling packets for event-triggered control," in *European Control Conference*, Budapest, Hungary, 2009.

[58] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal of Selected Areas in Telecommunications*, vol. 18, no. 3, pp. 535–547, 2000.

[59] P. Park, C. Fischione, and K. H. Johansson, "Modelling and stability analysis of hybrid multiple access in IEEE 802.15.4 protocol," *ACM Transactions on Sensor Networks*, vol. 9, no. 2, 2013.

[60] J. Araujo, A. Anta, M. Mazo Jr., J. Faria, A. Hernandez, P. Tabuada, and K. H. Johansson, "Self-triggered control over wireless sensor and actuator networks," in *IEEE International Conference on Distributed Computing in Sensor Systems*, Barcelona, Spain, 2011.

[61] C. Ramesh, H. Sandberg, and K. H. Johansson, "Steady state performance analysis of multiple state-based schedulers with CSMA," in *IEEE Conference on Decision and Control*, Orlando, FL, 2011.

[62] T. Henningsson and A. Cervin, "Comparison of LTI and event-based control for a moving cart with quantized position measurements," in *Proc. European Control Conf.*, 2009.

[63] W. Heemels, R. Gorter, A. van Zijl, P. v.d. Bosch, S. Weiland, W. Hendrix, and M. Vonder, "Asynchronous measurement and control: a case study on motor synchronisation," *Control Eng. Prac.*, vol. 7, pp. 1467–1482, 1999.

[64] E. Hendricks, M. Jensen, A. Chevalier, and T. Vesterholm, "Problems in event based engine control," in *Proc. American Control Conf.*, vol. 2, 1994, pp. 1585–1587.

[65] W. Kwon, Y. Kim, S. Lee, and K.-N. Paek, "Event-based modeling and control for the burnthrough point in sintering processes," *IEEE Trans. Control Syst. Technol.*, vol. 7, pp. 31–41, 1999.

[66] T.-J. Tarn, N. Xi, and A. Bejczy, "Path-based approach to integrated planning and control for robotic systems," *Automatica*, vol. 32, pp. 1675–1687, 1996.

[67] J. Sandee, W. Heemels, S. Hulsenboom, and P. van den Bosch, "Analysis and experimental validation of a sensor-based event-driven controller," in *Proc. American Control Conf.*, 2007, pp. 2867–2874.

[68] D. Lehmann and J. Lunze, "Extension and experimental evaluation of an event-based state-feedback approach," *Control Engineering Practice*, vol. 19, pp. 101 – 112, 2011.

[69] R. Cogill, "Event-based control using quadratic approximate value functions," in *Joint IEEE Conference on Decision and Control and Chinese Control Conference*, Shanghai, China, 2009, pp. 5883–5888.