Automatica 94 (2018) 81–87

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Periodic event-triggered control of nonlinear systems using overapproximation techniques[☆]

D.P. Borgers[a,*], R. Postoyan[b], A. Anta[c], P. Tabuada[d], D. Nešić[e], W.P.M.H. Heemels[a]

[a] Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
[b] Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France
[c] Austrian Institute of Technology, Vienna, Austria
[d] Cyber-Physical Systems Laboratory, Department of Electrical Engineering, University of California, Los Angeles, CA, USA
[e] Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC 3010, Australia

ARTICLE INFO

Article history:
Received 21 June 2016
Received in revised form 13 November 2017
Accepted 18 March 2018

Keywords:
Event-triggered control
Digital implementation
Sampling periods
Polytopic overapproximation
Continuous time systems
Nonlinear systems

ABSTRACT

In event-triggered control, the control task consisting of sampling the plant's output and updating the control input is executed whenever a certain event function exceeds a given threshold. The event function typically needs to be monitored *continuously*, which is difficult to realize in digital implementations. This has led to the development of *periodic* event-triggered control (PETC), in which the event function is only evaluated periodically. In this paper, we consider general nonlinear continuous event-triggered control (CETC) systems, and present a method to transform the CETC system into a PETC system. In particular, we provide an explicit sampling period at which the event function is evaluated and we present a constructive procedure to redesign the triggering condition. The latter is obtained by upper-bounding the evolution of the event function of the CETC system between two successive sampling instants by a linear time-invariant system and then by using convex overapproximation techniques. Using this approach, we are able to preserve the control performance guarantees (e.g., asymptotic stability with a certain decay rate) of the original CETC system.

1. Introduction

In digital control applications, the control task consists of sampling the outputs of the plant and computing and implementing new actuator signals. This procedure is typically executed in a time-triggered fashion, which may lead to a waste of communication and energy resources, as the execution of the control task is done irrespective of whether there actually is a need for a control update or not. To mitigate the unnecessary waste of resources, various event-triggered control (ETC) strategies have been proposed in the recent literature, see, e.g., Cassandras (2014), Dolk, Borgers, and Heemels (2017), Girard (2015), Heemels, Sandee, and Van Den Bosch (2008), Henningsson, Johannesson, and Cervin (2008), Lunze and Lehmann (2010), Miskowicz (2006), Postoyan, Tabuada, Nesić, and Anta (2015), Tabuada (2007) and Tallapragada and Chopra (2014). In ETC, the control task is executed after the occurrence of an event, generated by some well-designed triggering condition, rather than after a fixed period of time, as in conventional periodic sampled-data control. In this way, ETC is capable of significantly reducing the number of control task executions, while retaining a satisfactory closed-loop performance.

A main implementation issue of ETC (for which we will use the term *continuous* event-triggered control (CETC) from here on) is that the event function has to be monitored continuously, which is difficult to realize on digital platforms. A solution to this problem is *periodic* event-triggered control (PETC), in which the event function is only checked periodically at fixed equidistant time instances, thereby enabling (easier) implementation on a digital platform. Note that PETC differs from standard periodic sampled-data control, as in PETC the event times (which result from the triggering condition and the system's state evolution) are in general only a (specific) subset of the sampling times and can be aperiodic. Of course, event-triggered control schemes for discrete-time systems (e.g., Cogill, 2009, Eqtami, Dimarogonas, &

☆ This work is supported by the Innovational Research Incentives Scheme under the VICI grant "Wireless control systems: A new frontier in automation" (No. 11382) awarded by NWO (The Netherlands Organization for Scientific Research) and STW (Dutch Science Foundation), by the ANR under the grant COMPACS (ANR-13-BS03-0004-02), and by the Australian Research Council (DP170104102) under the Discovery Projects Scheme. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Ricardo Sanfelice under the direction of Editor Daniel Liberzon.

* Corresponding author.
E-mail addresses: d.p.borgers@tue.nl (D.P. Borgers), romain.postoyan@univ-lorraine.fr (R. Postoyan), aantam@gmail.com (A. Anta), tabuada@ee.ucla.edu (P. Tabuada), dnesic@unimelb.edu.au (D. Nešić), m.heemels@tue.nl (W.P.M.H. Heemels).

https://doi.org/10.1016/j.automatica.2018.04.019
0005-1098/

Kyriakopoulos, 2010, Heemels & Donkers, 2013, Li & Lemmon, 2011, Molin & Hirche, 2013, Yook, Tilbury, & Soparkar, 2002) can also be interpreted as PETC schemes, but these do not take into account the inter-sample behavior. In the past few years, various PETC strategies have been proposed, see, e.g., Heemels, Donkers, and Teel (2013), Heemels, Dullerud, and Teel (2016), Heemels et al. (2008), Henningsson et al. (2008) and Postoyan, Anta, Heemels, Tabuada, and Nesić (2013). However, to the best of the authors' knowledge, there are hardly any design methods for PETC for *non-linear continuous-time* systems. Two exceptions are the works in Sanfelice and Teel (2006) and Wang, Postoyan, Nesić, and Heemels (2016). In Sanfelice and Teel (2006), the sample-and-hold implementation of general hybrid controllers for nonlinear systems is analyzed, which covers the PETC implementation of a nonlinear CETC system as a subcase. These results ensure that, under general conditions, if a compact set is uniformly globally asymptotically stable (UGAS) for the original CETC system, then this property is semiglobally and practically preserved for the emulated PETC system by taking the sampling period sufficiently small. In the recent work Wang et al. (2016), an approach has been proposed for the design of PETC state-feedback controllers to stabilize nonlinear systems, which ensures uniform global asymptotic properties and provides an explicit bound on the sampling period.

In this paper, we present a method to transform a general nonlinear CETC system into a PETC system which *preserves* the control performance guarantees of the given CETC system. Our method consists of two steps. First, we upper bound the evolution of the event function of the given CETC system between two successive sampling instants by a linear time-invariant (LTI) system. Based on this LTI system, we can formulate a redesigned event function for the PETC implementation which would involve checking an infinite number of conditions at every sample time. To overcome this issue, we use convex techniques to overapproximate the evolution of the LTI system over a sampling period, and end up with a redesigned event function which is implementable in practice.

In contrast to Sanfelice and Teel (2006), our method provides an explicit sampling period (in fact, the sampling period is a design parameter), it fully preserves the control performance guarantees of the given CETC system, and is not limited to stability of a compact set a priori. Compared to Wang et al. (2016), we do not focus on stabilization and we can cope with a larger class of triggering conditions. Preliminary results have been presented in Postoyan et al. (2013), in which we were only able to *approximately* preserve the control performance guarantees of the given CETC system. In addition, the new results presented here are based on less stringent conditions compared to Postoyan et al. (2013) (see Remark 3 for more details).

Nomenclature. Let $\mathbb{R} = (-\infty, \infty)$, $\mathbb{R}_{\geqslant 0} = [0, \infty)$, $\mathbb{N} = \{1, 2, \ldots\}$ and $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$. Given $N \in \mathbb{N}$, we denote the set $\{1, 2, \ldots, N\}$ by $\bar{N}$. For a vector $x \in \mathbb{R}^n$, we denote by $\|x\| := \sqrt{x^\top x}$ its 2-norm, and for a matrix $A \in \mathbb{R}^{n\times m}$, we denote by $\|A\| := \sqrt{\lambda_{\max}(A^\top A)}$ its induced 2-norm. For a signal $w : \mathbb{R}_{\geqslant 0} \to \mathbb{R}^n$, we denote the right limit at time $t \in \mathbb{R}_{\geqslant 0}$ by $w(t^+) = \lim_{s\downarrow t} w(s)$, when it exists. The solution z of a time-invariant dynamical system at time $t \in \mathbb{R}_{\geqslant 0}$ starting with the initial condition $z(0) = z_0$ will be denoted by $z(t, z_0)$ or simply by $z(t)$ when the initial state is clear from the context. The notation $\lfloor x \rfloor$ stands for the largest integer smaller than or equal to $x \in \mathbb{R}$.

2. Problem statement

We consider a nonlinear plant of the form

$$\dot{x}(t) = f(x(t), u(t)), \tag{1}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state and $u(t) \in \mathbb{R}^{n_u}$ is the control input at time $t \in \mathbb{R}_{\geqslant 0}$. We assume that we have designed a continuous event-triggered state-feedback controller for plant (1), given by

$$\hat{x}(t) = x(t_k), \qquad \text{for } t \in (t_k, t_{k+1}] \tag{2a}$$

$$u(t) = k(\hat{x}(t)) \tag{2b}$$

$$t_0 = 0$$

$$t_{k+1} = \inf\{t > t_k \mid \Gamma\left(x(t), \hat{x}(t), \chi(t)\right) > 0\}, \tag{2c}$$

where the function $k(\hat{x})$ defines the feedback law, $\hat{x}$ is the state information available to the controller, and $\chi \in \mathbb{R}^{n_\chi}$ is used to capture other relevant variables such as timers, counters, or possibly even the state of an auxiliary dynamical system (Dolk et al., 2017; Girard, 2015; Postoyan et al., 2015). The event function Γ is designed such that some desired control performance (e.g., asymptotic stability with a certain decay rate) is achieved as long as it remains non-positive along the system's trajectories.

Writing the triggering law as in (2c) allows us to consider various event-triggers considered previously in the literature, which we illustrate by the following two examples. In Tabuada (2007), the condition $\Gamma(x, \hat{x}) = \gamma(\|\hat{x} - x\|) - \sigma\alpha(\|x\|) \leqslant 0$ (for specific functions γ, α and $\sigma \in (0, 1)$) ensures that a Lyapunov function V has a guaranteed decay rate $(1 - \sigma)\alpha(\|x\|)$ along the solutions to system (2) (which guarantees global asymptotic stability of the system). In Dolk et al. (2017), we have that $\chi = (\tau, \kappa, \eta)$ (where τ is a timer, κ a counter, and η the state of an auxiliary dynamical system), and that the condition $\Gamma(x, \hat{x}, \chi) = -\eta \leqslant 0$ ensures that the system is UGAS with a guaranteed decay rate. Another example is provided in Section 4.

Let $z = (x, \hat{x}, \chi) \in \mathbb{R}^{n_z}$ with $n_z = 2n_x + n_\chi$. We model the closed-loop system (1)–(2) (and possibly auxiliary dynamics for χ) as an impulsive system like in Heemels et al. (2013), which gives

$$\dot{z} = g(z), \text{ for } t \in (t_k, t_{k+1}] \tag{3a}$$

$$z(t_k^+) = b(z(t_k)) \tag{3b}$$

$$t_0 = 0$$

$$t_{k+1} = \inf\{t > t_k \mid \Gamma\,(z(t)) > 0\}, \tag{3c}$$

for $k \in \mathbb{N}_0$, and appropriate $g : \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$ and $b : \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$. In case $n_\chi = 0$, we have that

$$g(z) = \begin{bmatrix} f(x, k(\hat{x})) \\ 0 \end{bmatrix} \text{ and } b(z) = \begin{bmatrix} x \\ x \end{bmatrix}.$$

For the definition of the functions g and b in case $n_\chi \neq 0$ we refer to Section 4 for an example.

Solutions to (3) are interpreted as follows. In between the event times t_k, $k \in \mathbb{N}$, determined by (3c), the system evolves according to the differential equation (3a), where $z(t_k^+)$ given by the update (3b) denotes the starting point for the solution to (3a) in the interval $(t_k, t_{k+1}]$, $k \in \mathbb{N}$. Hence, the solutions we consider are left-continuous signals. Note that $t_0 = 0$, and hence, we start with an update according to (3b).

Remark 1. The analysis presented in this paper is based on system (3). Therefore, our design applies to any CETC configuration that can be written in the format of (3), including the case where the control input u in (1) is generated by a dynamic controller. The states of the controller would then be incorporated in the vector x and we would obtain a model of the form (3). Similarly, the case in which the controller is not implemented using zero-order-hold functions can be considered as long as the problem can be modeled by (3). For instance, when using the model-based technique of Lunze and Lehmann (2010), $\hat{x}$ would be equal to x_s in Lunze and Lehmann (2010), which is the model-based estimate of x.

In order to transform the CETC system (3) into a PETC system, we require the following three assumptions.