Automatica 94 (2018) 294–299

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Decentralized periodic event-triggered control with quantization and asynchronous communication ☆

Anqi Fu *, Manuel Mazo Jr.

Delft Center for Systems and Control, Delft University of Technology, The Netherlands

ARTICLE INFO

Article history:
Received 29 March 2017
Received in revised form 6 March 2018
Accepted 20 March 2018

Keywords:
Decentralized event-triggered control
Periodic sampling
Dynamic quantization
Wireless networked control systems
Cyber–physical systems

ABSTRACT

Asynchronous decentralized event-triggered control (ADETC) Mazo Jr. and Cao (2014) is an implementation of controllers characterized by decentralized event generation, asynchronous sampling updates, and dynamic quantization. Combining those elements in ADETC results in a parsimonious transmission of information which makes it suitable for wireless networked implementations. We extend the previous work on ADETC by introducing periodic sampling, denoting our proposal asynchronous decentralized periodic event-triggered control (ADPETC), and study the stability and $\mathcal{L}_2$-gain of ADPETC for implementations affected by disturbances. In ADPETC, at each sampling time, quantized measurements from those sensors that triggered a local event are transmitted to a dynamic controller that computes control actions; the quantized control actions are then transmitted to the corresponding actuators only if certain events are also triggered for the corresponding actuator. The developed theory is demonstrated and illustrated via a numerical example.

1. Introduction

In digital control applications, the control task consists of sampling and transmitting the output of the plant, and computing and implementing controller outputs. Current developments of sensor and networking technologies have enabled the emergence of wireless networked control systems (WNCS), in which communication of distributed components is established via wireless networks. WNCS can be established and updated with large flexibility and low cost, and are especially suitable to physically distributed plants. Limited energy supplies are often the case when sensors are battery powered for mobility and/or flexibility reasons. The major challenge in WNCS design is thus to achieve prescribed performance under limited bandwidth and energy supplies. Our present work is mostly inspired by Heemels, Donkers, and Teel (2013), Liberzon and Nešić (2007) and Mazo Jr. and Cao (2014). In Heemels et al. (2013), Heemels et al. present a periodic event-triggered control (PETC) mechanism. In PETC, the sensors sample the output of the plant and verify the central or local event conditions periodically. Therefore, the energy consumed by sensing is reduced compared to those continuously monitoring event-triggered mechanisms, while still a pre-designed performance can be guaranteed. In Liberzon and Nešić (2007), Liberzon and Nešić present a state dependent quantizer which zooms in and out based on the system's state, so as to provide input to state stability (ISS). In Mazo Jr. and Cao (2014), Mazo and Cao present an asynchronous decentralized event-triggered control (ADETC) mechanism combining state dependent dynamic quantization and decentralized event-triggering conditions.

We propose an asynchronous decentralized periodic event-triggered control (ADPETC) mechanism building on the aforementioned pieces of work with the goal of reducing wireless channel bandwidth occupation and energy consumption. This ADPETC incorporates: quantization in a zooming fashion, which is similar to Liberzon and Nešić (2007) and Mazo Jr. and Cao (2014); an asynchronous event-triggered mechanism, based on Mazo Jr. and Cao (2014); and periodic sampling as in Heemels et al. (2013). Moreover, compared with (Liberzon & Nešić, 2007; Mazo Jr. & Cao, 2014), in our approach the quantization error or global threshold depends on the information in the controller, instead of just on the current estimation of the system's state; compared with Heemels et al. (2013), in which the algorithm for designing decentralized event condition parameters is complex: requiring to solve a set of linear matrix inequalities (LMIs), our approach requires to solve only one LMI. This advantage is more apparent when the system output's and/or input's dimension increase, since the number of LMIs and decision variables in Heemels et al. (2013) increases with it, while they remain constant in the present approach. It is worth noting that while, in general, our approach is simpler, for some particular combinations of (small) plants and controllers, the LMIs

☆ This work is partly funded by China Scholarship Council (CSC, no. 201306020047). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Dimos V. Dimarogonas under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: anqifu@ieee.org (A. Fu), M.Mazo@tudelft.nl (M.Mazo Jr.).

https://doi.org/10.1016/j.automatica.2018.04.045
0005-1098/

of Heemels et al. (2013) maybe smaller than the LMI of the present approach. In our preliminary version (Fu & Mazo Jr., 2016), in order to design the event condition parameters, a set of bilinear matrix inequalities (BMIs) needs to be solved. In the current version, we solve instead a single LMI which often leads to less conservative triggering conditions, i.e. less triggered events. This contributes the main differences between Fu and Mazo Jr. (2016) and the present paper.

2. Preliminaries and problem definition

We denote the positive real numbers by $\mathbb{R}^+$, by $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$, and the natural numbers including zero by $\mathbb{N}$. $|\cdot|$ denotes the Euclidean norm in the appropriate vector space, when applied to a matrix $|\cdot|$ denotes the l_2 induced matrix norm. Let us consider a linear time-invariant (LTI) plant given by

$$\begin{cases} \dot{\xi}_p(t) = A_p\xi_p(t) + B_p\hat{v}(t) + Ew(t) \\ y(t) = C_p\xi_p(t), \end{cases} \tag{1}$$

where $\xi_p(t) \in \mathbb{R}^{n_p}$ and $y(t) \in \mathbb{R}^{n_y}$ denote the state vector and output vector of the plant, respectively, and $w(t) \in \mathbb{R}^{n_w}$ denotes an unknown disturbance. The input $v(t) \in \mathbb{R}^{n_v}$ is defined as $\hat{v}(t) := \hat{v}(t_k)$, $\forall t \in [t_k, t_{k+1}[$, $\forall k \in \mathbb{N}$, where $\hat{v}(t_k)$ is a quantized version of $v(t_k)$ provided by the following discrete-time controller:

$$\begin{cases} \xi_c(t_{k+1}) = A_c\xi_c(t_k) + B_c\hat{y}(t_k) \\ v(t_k) = C_c\xi_c(t_k) + D_c\hat{y}(t_k), \end{cases} \tag{2}$$

where $\xi_c(t_k) \in \mathbb{R}^{n_c}$, $v(t_k) \in \mathbb{R}^{n_v}$, and $\hat{y}(t_k) \in \mathbb{R}^{n_y}$ denote the state vector, output vector of the controller, and input applied to the controller, respectively. Define $h > 0$ the sampling interval. A periodic sampling sequence is given by

$$\mathcal{T} := \{t_k | t_k := kh, k \in \mathbb{N}\}.$$

Define $\tau(t)$ be the elapsed time since the last sampling time, i.e. $\tau(t) := t - t_k$, $t \in [t_k, t_{k+1}[$. Define two vectors for the implementation input and output $u(t_k) := [y^{\mathrm{T}}(t_k)\, v^{\mathrm{T}}(t_k)]^{\mathrm{T}} \in \mathbb{R}^{n_u}$, $\hat{u}(t_k) := [\hat{y}^{\mathrm{T}}(t_k)\, \hat{v}^{\mathrm{T}}(t_k)]^{\mathrm{T}} \in \mathbb{R}^{n_u}$, with $n_u := n_y + n_v$. $u^i(t_k)$ $\hat{u}^i(t_k)$ are the ith elements of the vector $u(t_k)$, $\hat{u}(t_k)$, respectively. At each sampling time $t_k \in \mathcal{T}$, the input applied to the implementation $\hat{u}(t_k)$ is determined by

$$\hat{u}^i(t_k) := \begin{cases} \tilde{q}(u^i(t_k)), & \text{if a local event triggered} \\ \hat{u}^i(t_{k-1}), & \text{otherwise,} \end{cases} \tag{3}$$

where $\tilde{q}(s)$ denotes the quantized signal of s. Therefore, at each sampling time, only those inputs that triggered events are required to transmit measurements or actuation signals through the network. Between samplings, a zero-order hold mechanism is applied.

We also introduce a performance variable $z \in \mathbb{R}^{n_z}$ given by

$$z(t) = g(\xi(t), w(t)), \tag{4}$$

where $\xi(t) := [\xi_p^{\mathrm{T}}(t)\, \xi_c^{\mathrm{T}}(t)\, \hat{y}^{\mathrm{T}}(t)\, \hat{v}^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_\xi}$, $n_\xi := n_p + n_c + n_y + n_v$, and $g(s)$ is a design function.

In this implementation, the controller, sensors, and actuators are assumed to be physically distributed, and none of the nodes are co-located. We employ the definition of uniform global pre-asymptotic stable (UGpAS), Lyapunov function candidate, and sufficient Lyapunov conditions for UGpAS from Goebel, Sanfelice, and Teel (2009).

Definition 1 (*$\mathcal{L}_2$-Gain Heemels et al., 2013*). The system (1), (2), (4) is said to have an $\mathcal{L}_2$-gain from w to z smaller than or equal to γ, if there is a $\mathcal{K}_\infty$ function $\delta : \mathbb{R}^{n_\xi} \to \mathbb{R}^+$ such that for any $w \in \mathcal{L}_2$, any initial state $\xi(0) = \xi_0 \in \mathbb{R}^{n_\xi}$ and $\tau(0) \in [0, h]$, the corresponding solution to system (1), (2), (4) satisfies $\|z\|_{\mathcal{L}_2} \le \delta(\xi_0) + \gamma\|w\|_{\mathcal{L}_2}$.

In the local event conditions in (3), an event occurs when the following inequality holds:

$$|\hat{u}^i(t_{k-1}) - u^i(t_k)| \ge \sqrt{\eta_i(t_k)},\ i \in \{1, \dots, n_u\}, \tag{5}$$

in which $\eta_i(t_k)$ is a local threshold, computed as follows:

$$\eta_i(t) := \theta_i^2\eta^2(t), \tag{6}$$

where θ_i is a designed distributed parameter satisfying $|\theta| = 1$ and $\eta : \mathbb{R}_0^+ \to \mathbb{R}^+$, determines the global threshold, which will be discussed in Section 3. When an event takes place at a sampling time t_k, $\hat{u}(t_k)$ is updated by

$$\begin{aligned} &\hat{u}^i(t_k) = \tilde{q}(u^i(t_k)) = q_\eta(u^i(t_k), \hat{u}^i(t_{k-1})) := \\ &\hat{u}^i(t_{k-1}) - \mathrm{sign}(\hat{u}^i(t_{k-1}) - u^i(t_k))m^i(t_k)\sqrt{\eta_i(t_k)}, \end{aligned} \tag{7}$$

where $m^i(t_k) := \left\lfloor \frac{|\hat{u}^i(t_{k-1}) - u^i(t_k)|}{\sqrt{\eta_i(t_k)}} \right\rfloor$. The error after this update is

$$\begin{aligned} &e_u^i(t_k) := \hat{u}^i(t_k) - u^i(t_k) = -\mathrm{sign}(\hat{u}^i(t_{k-1}) - \\ &u^i(t_k))\left(m^i(t_k) - \frac{|\hat{u}^i(t_{k-1}) - u^i(t_k)|}{\sqrt{\eta_i(t_k)}}\right)\sqrt{\eta_i(t_k)}. \end{aligned} \tag{8}$$

One can easily observe that, $|e_u^i(t_k)| < \sqrt{\eta_i(t_k)}$. That is, when there is an event locally, after the update by (7), (5) does not hold anymore. Later we show that, $\forall i \in \{1, \dots, n_u\}$, $k \in \mathbb{N}$, $m^i(t_k) \le \bar{m}_x < \infty$. Thus, in practice one only needs to send $\mathrm{sign}(\hat{u}^i(t_{k-1}) - u^i(t_k))$ and $m^i(t_k)$ for each input update. Therefore, only $\log_2(m^i(t_k)) + 1$ bits are required for each transmission from a single sensor or to a single actuator. Define $\Gamma_{\mathcal{J}} := \mathrm{diag}(\Gamma_{\mathcal{J}}^y, \Gamma_{\mathcal{J}}^v) = \mathrm{diag}(\gamma_{\mathcal{J}}^1 \cdots, \gamma_{\mathcal{J}}^{n_u})$, where $\mathcal{J}$ is an index set: $\mathcal{J} \subseteq \bar{\mathcal{J}} = \{1, \dots, n_u\}$ for $u(t)$, indicating the occurrence of events. Define $\mathcal{J}_c := \bar{\mathcal{J}} \setminus \mathcal{J}$. For $l \in \{1, \dots, n_u\}$, if $l \in \mathcal{J}$, $\gamma_{\mathcal{J}}^l = 1$; if $l \in \mathcal{J}_c$, $\gamma_{\mathcal{J}}^l = 0$. Furthermore, we use the notation $\Gamma_j = \Gamma_{\{j\}}$. Define $C := \begin{bmatrix} C_p & 0 \\ 0 & C_c \end{bmatrix}$ and $D := \begin{bmatrix} 0 & 0 \\ D_c & 0 \end{bmatrix}$. The local event-triggered condition (5) can now be reformulated as a set membership:

$$i \in \mathcal{J} \text{ iff } \xi^{\mathrm{T}}(t_k)Q_i\xi(t_k) \ge \eta_i(t_k), \tag{9}$$

where

$$Q_i = \begin{bmatrix} C^{\mathrm{T}}\Gamma_iC & C^{\mathrm{T}}\Gamma_iD - C^{\mathrm{T}}\Gamma_i \\ D^{\mathrm{T}}\Gamma_iC - \Gamma_iC & (D-I)^{\mathrm{T}}\Gamma_i(D-I) \end{bmatrix}.$$

The ADPETC implementation determined by (1), (2), (3), (4), and (9) can be re-written as an impulsive system model:

$$\begin{aligned} &\begin{bmatrix} \dot{\xi}(t) \\ \dot{\tau}(t) \end{bmatrix} = \begin{bmatrix} \bar{A}\xi(t) + \bar{B}w(t) \\ 1 \end{bmatrix}, \text{ when } \tau(t) \in [0, h[, \\ &\begin{bmatrix} \xi(t_k^+) \\ \tau(t_k^+) \end{bmatrix} = \begin{bmatrix} J_{\mathcal{J}}\xi(t_k) + \Delta_{\mathcal{J}}(t_k)\eta(t_k) \\ 0 \end{bmatrix}, \text{ when } \tau(t) = h, \\ &z(t) = g(\xi(t), w(t)), \end{aligned} \tag{10}$$

where $\bar{B} = \begin{bmatrix} E^{\mathrm{T}} & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$ and

$$\bar{A} = \begin{bmatrix} A_p & 0 & 0 & B_p \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\ \Delta_{\mathcal{J}}(t_k) = \begin{bmatrix} 0 \\ B_c\Gamma_{\mathcal{J}}^y\epsilon_y(t_k)\Theta_y \\ \Gamma_{\mathcal{J}}^y\epsilon_y(t_k)\Theta_y \\ \Gamma_{\mathcal{J}}^v\epsilon_v(t_k)\Theta_v \end{bmatrix},$$

$$J_{\mathcal{J}} = \begin{bmatrix} I & 0 & 0 & 0 \\ B_c\Gamma_{\mathcal{J}}^yC_p & A_c & B_c(I - \Gamma_{\mathcal{J}}^y) & 0 \\ \Gamma_{\mathcal{J}}^yC_p & 0 & (I - \Gamma_{\mathcal{J}}^y) & 0 \\ 0 & \Gamma_{\mathcal{J}}^vC_c & \Gamma_{\mathcal{J}}^vD_c & (I - \Gamma_{\mathcal{J}}^v) \end{bmatrix},$$