Journal of Process Control 22 (2012) 1930–1945

Contents lists available at SciVerse ScienceDirect

Journal of Process Control

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

Characterization of symmetric send-on-delta PI controllers

Manuel Beschi [a], Sebastián Dormido [b], José Sanchez [b], Antonio Visioli [a,*]

[a] Dipartimento di Ingegneria dell'Informazione, University of Brescia, Italy
[b] Departamento de Informática y Automática, UNED, Spain

ARTICLE INFO

Article history:
Received 9 May 2012
Received in revised form
11 September 2012
Accepted 12 September 2012
Available online 22 October 2012

Keywords:
Event-based PI controllers
Send-on-delta sampling
Stability

ABSTRACT

A new event-based proportional–integral controller, based on a specific send-on-delta sampling strategy, is analyzed in this paper. In particular, necessary and sufficient conditions on the controller parameters for the existence of equilibrium points without limit cycles are given for a first-order-plus-dead-time process. These conditions can be usefully exploited for the tuning of the controller, thus making the overall design easier. Practical issues related to the controller implementation are also addressed. Simulation and experimental results are provided as illustrative examples.

1. Introduction

Proportional–integral-derivative (PID) controllers are the most employed controllers in industry because of their capability to provide a satisfactory performance for many processes with a relatively easy design. Their use is also made easier by the presence of a large number of tuning rules that are available [1]. However, the advancement of the technology in industrial plants requires new developments in PID control. One of the most significant recent issues is the introduction of wireless sensors and actuators [2], which poses constraints on the communication rate in order to minimize the power consumption (and therefore to increase the battery life) and the risk of lost data and stochastic time delays [3–5]. In this context, two of the most convenient strategies that can be employed to reduce the communication load are event-based control and self-triggered control, where the reduction of communications can be obtained at the expense of a small steady-state error, which often is not of main concern in practical cases. In the former, the communication is done only when a logic condition becomes true, while in the latter the controller has to predict the next time instant when the logic condition will be verified [6]. During the last years many researchers have addressed the self-triggered approach (see, for example [7–9]) and the event-based sampling and control approach (see, for example [10–12]), in particular in the context of PID controllers (see [13] for the self-triggered strategies and [14–21] for the event-based strategies). One of the most important advantages of the self-triggered strategies is the possibility of using the master-slave protocol in the communications between control agents and in these techniques it is not necessary to continuously monitor the system outputs. However, the disturbance rejection task is more complicated and the performance of these approaches depends on the model of the system. Conversely, in an event-based control strategy, when a control agent detects an event, it sends the related information, via the communication medium, to another agent. In this way, the disturbance rejection task starts at the time instance when the logic condition is verified and the technique can be model-free.

In this context, many logical conditions have been proposed. One of the most employed is the so-called send-on-delta (SOD) sampling (also known as deadband sampling [22] or level crossing sampling [23]) where the measured value of the process variable is sent to the controller when the control error (or some function of it) crosses predefined quantization levels [24].

However, it has to be recognized that in event-based control the events occur asynchronously and therefore the tuning of the PID controller parameters is in general more challenging, as the timing of the events influences the system performance and limit cycles may arise (note that the presence of limit cycles is a typical problem in general event-based control systems [25] and see, for example [26] where sufficient conditions on the controller parameters for the existence of equilibrium points are given for a SOD-PI control system). Further, in addition to the PID gains, there are in general other parameters (threshold values) employed in the control algorithm that have to be tuned, thus making the overall control design

* Corresponding author. Tel.: +39 030 371 5460; fax: +39 030 38 0014.
E-mail addresses: manuel.beschi@ing.unibs.it (M. Beschi), sdormido@dia.uned.es (S. Dormido), jsanchez@dia.uned.es (J. Sanchez), antonio.visioli@ing.unibs.it (A. Visioli).

0959-1524/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jprocont.2012.09.005

more complex. Indeed, the tuning of a PID controller with deadband sampling has not been explicitly addressed in the literature until now, at least to the authors' knowledge.

In this paper we propose an event-based strategy which results from a modification of the SOD technique. In particular, the sampled signal is quantized by a quantity multiple of Δ so that the relationship between the input and output of the event-generator block is symmetric with respect to the origin. Thus, we call this strategy symmetric send-on-delta (SSOD) sampling. In this way, the value of the sampled signal is independent on the initial conditions. We will exploit this property to determine necessary conditions on system instability and necessary and sufficient conditions on the controller parameters for the existence of equilibrium points without limit cycles will be given for a first-order-plus-dead-time (FOPDT) process. Although this can be considered as a special case, it is significant from a practical point of view as many industrial processes can be modeled effectively in this way. Event-based P, I, and PI controllers are considered.

The case where the SSOD strategy is applied to the control variable (thus reducing the number of transmissions between the controller and the actuator) is also addressed. Both cases (denoted respectively as SSOD-PI and PI-SSOD) are treated conveniently in a unified framework. It is believed that the provided stability conditions can be usefully employed for the tuning of the overall controller. It is also shown that the choice of the parameter Δ does not influence the stability of the system (see Section 3.2) and therefore can be selected just in order to handle the trade-off between the reduction of the desired number of events and the reduction of the steady-state error. The paper is organized as follows. In Section 2 the event-based control strategies are presented. In Section 3, the two stability and limit cycles problems are investigated. Necessary and sufficient conditions for the presence of equilibrium points without the occurrence of limit cycles and necessary conditions on the system instability (and therefore sufficient conditions for the system stability) are given. In Section 4 practical issues on the controller implementation are presented. Simulation results are shown in Section 5 while experimental results related to a level control problem are presented in Section 6. Conclusions are drawn in Section 7.

2. Control architecture

In event-based control strategies, the controller can be divided into four logical blocks: the sensor unit, the control unit, the actuator unit and a governor (for short SU, CU, AU and G, respectively), as shown in Fig. 1. The units and their tasks can be described as follows:

- The sensor unit is composed of the sensor and its on-board intelligence. Its task is to measure the process output and to calculate the error between the measured signal and a constant set-point value received from the governor.

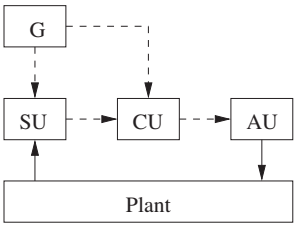

Fig. 1. Scheme of a generic event based control strategies. The dashed arrows indicate the possibility of an event-triggered data transmission.

- The control unit implements the control algorithm, which determines the control action by taking into account the last received sampled error and sends it to the AU.
- The actuator unit receives the control action signal from the CU and applies it to the actuator.
- The governor, which, in practice, can be implemented together with one of the previous two blocks, receives the desired set-point value from a user interface or from a higher hierarchical controller, and sends it to the sensor unit.

These blocks can be implemented in a unique machine or in two or more physical entities. In this last case, the data has to be sent from one to each other on a network. It is clear that the communication between two entities implies more efforts than the data exchanging into a single machine, especially when they are battery-powered. For this reason, it is recommended to use event-triggered data exchanging for all the signals which are sent between two machines and normal time-driven sampling for the data which are elaborated by an unique machine.

In this paper, we consider two special cases where two of the three control agents (namely, SU, CU and AU) are placed in a physical entity and the other one is located in another machine. We do not address the case where SU and AU are in the same machine, because in this case it is sufficient to implement also the CU task in this machine in order to obtain a standard controller. The remaining cases are: SU separated from CU and AU (which are in the same machine) and AU separated from SU and CU (which are in the same machine). In both situations, only an event-triggered data exchanging in the control loop is required. The communications between the governor and the other units (considering only the transmissions concerning the control aspects) are done only when the set-point signal changes.

2.1. Symmetric send-on-delta triggering

In this paper, the event-triggered data exchanging is done by applying a special case of the send-on-delta sampling method (see [4,15]), which can be seen also as a generalization of a relay with hysteresis.

We call this technique symmetric send-on-delta (SSOD) sampling. Denote as $v(t)$ the input signal to the sampling block and as $v^*(t)$ the sampled output signal, which can assume only values multiple of a predefined threshold Δ multiplied by a gain $\beta > 0$, namely $v^*(t) = j\Delta\beta$ with $j \in \mathbb{Z}$. The sampled signal changes its value to the upper quantization level when the input signal $v(t)$ increases more than Δ, or to the lower quantization level when $v(t)$ decreases more than Δ. This behavior can be mathematically described as:

$$v^*(t) = \mathrm{ssod}(v(t); \Delta, \beta) = \begin{cases} (i+1)\Delta\beta & \text{if } v(t) \geq (i+1)\Delta \text{ and } v^*(t^-) = i\Delta\beta \\ i\Delta\beta & \text{if } v(t) \in [(i-1)\Delta, (i+1)\Delta] \text{ and } v^*(t^-) = i\Delta\beta \\ (i-1)\Delta\beta & \text{if } v(t) \leq (i-1)\Delta \text{ and } v^*(t^-) = i\Delta\beta \end{cases}. \quad (1)$$

The relationship between $v(t)$ and $v^*(t)$ can be considered as a generalization of a relay with hysteresis, where there are an infinite number of thresholds [23], as shown in Fig. 2. The mechanism can also be analyzed by means of a state-machine representation, where j is the state number, $v(t) \geq (j+1)\Delta$ is the condition to jump to the upper state $j+1$ and $v(t) \leq (j-1)\Delta$ is the condition to jump to the lower state $j-1$, as shown in Fig. 3.

The SSOD algorithm, described in (1), can be rewritten as:

$$v^*(t) = \Delta\beta\,\mathrm{ssod}\left(\frac{v(t)}{\Delta}; 1, 1\right). \quad (2)$$

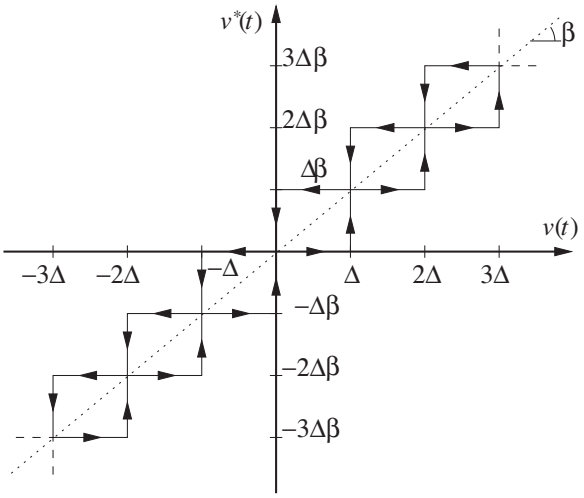

Fig. 2. Relationship between $v(t)$ and $v^*(t)$.

This representation suggests that the stability properties of a system which contains a SSOD event generator do not depend on the parameters Δ and β. In fact, it is always possible to do a variable change and to reduce the SSOD block to its normalized version $\mathrm{ssod}(v(t); 1, 1)$.

Also in the standard send-on-delta (SOD) technique, it is possible to define a state machine representation, but in this case the value assumed by v^* in the state j is equal to $(j\Delta + q)\beta$ (see [26]), where $q \in [0, \Delta)$ is an unquantizable quantity which depends on the initial value of $v(t)$. Thus, with the SSOD technique the system always has a state with $v^* = 0$ and this fact has an important influence to the controlled system behavior, as shown in Section 3.

2.2. Modeling the communications delays and the packet losses

In the SSOD (or SOD) sampling communication delays and packet losing effects can be modeled as a disturbance added to the SSOD (or SOD) output $v(t)$, as shown in Fig. 4. How this additive disturbance affects the performance depends on its position in the control loop. In fact, if the SSOD sampling is applied on the error signal (as in SSOD-PI technique) it is possible to have a deviation between the last received output and the true SSOD output. A way to avoid this situation is to force the sensor to resend the data to the controller if a state $j \neq 0$ is kept for more than a maximum time interval t_{max}.

2.3. PI controller

The controller used in this work is a (discretized version of) a continuous time PI controller, namely:

$$C(s) = K_p + \frac{K_i}{s} \tag{3}$$

where $K_p \geq 0$ is the proportional gain and $K_i \geq 0$ is the integral gain.

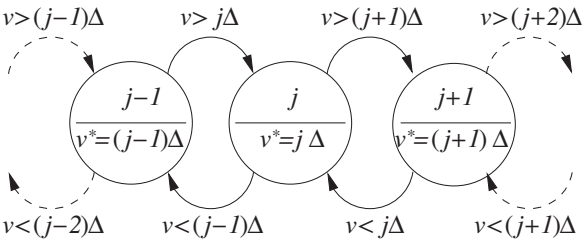

Fig. 3. State machine representation of the symmetric send-on-delta sampling technique.

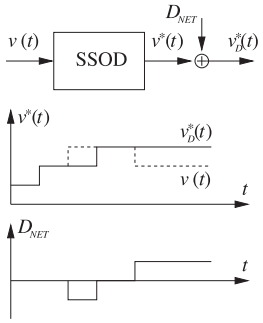

Fig. 4. Modelization of communication delays and packet losses as additive disturbance, where $v^*(t)$ is the sent SSOD output, $v^*_D(t)$ is the received signal and D_{NET} is the disturbance due to the communications delays and the packet losses.

Remark 1. Because the controller is implemented in a single machine, it is possible to implement a standard anti-windup technique without additional efforts.

2.4. SSOD-PI controller

As explained before, we consider two different architectures, which present as common characteristic the presence of only two event-triggered data exchanging. The first is the set-point value sending, which is not a complex task from a control point of view, the second is implemented with the SSOD technique.

In the first architecture, shown in Fig. 5, the SU is located in a machine while CU and AU are placed in another physical entity (for example, the communication between the two components could be wireless). We call this architecture SSOD-PI controller, because the SSOD block is placed before the PI controller in the control loop. Note that the controller computes the control action at a regular sampling rate by taking into account the last received sampled error.

Remark 2. This architecture is very interesting because, in a networked control system, the sensor unit can be powered using a battery (thus, a reduction of the communications can increase the battery life). Conversely, the AU normally requires an external power supply, therefore the power consumption reduction is not a critical issue in this control agent.

Remark 3. In many control strategies the sensor has to send the error using an event-triggered algorithm. This fact can be an important constraint when the SU cannot receive the set-point information, which is sent to the CU. When the SSOD method is used, a way to solve this problem is to apply the SSOD algorithm to the process output and to send it to the controller. With this data, the CU elaborates the error signal and rounds it to the nearest multiple of $\Delta\beta$. Note that, as mentioned in Section 1, a small steady-state error does not constitute a hard design constraints for many industrial processes.

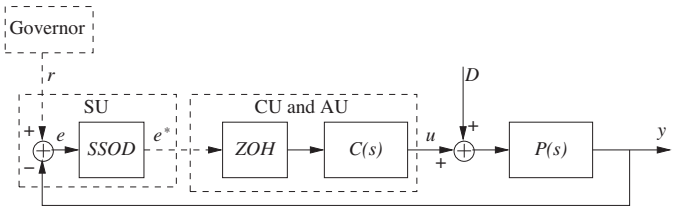

Fig. 5. Control scheme of the SSOD-PI controlled system. The dashed arrows indicate data sending via the communication medium.

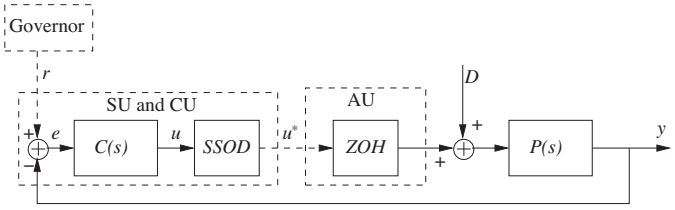

Fig. 6. Control scheme of the PI-SSOD controlled system. The dashed arrows indicate data sending via communication medium.

2.5. PI-SSOD controller

In the second architecture, shown in Fig. 6, the SU and the CU are located in the same machine, while the AU is placed in another entity. This solution is called for short PI-SSOD controller because the controller is before the SSOD block in the control loop. Note that the AU holds the last received control action value until the next data exchanging.

Remark 4. The communication delays and the packet losing effect can be modeled as disturbances (see Section 2.2). Using the PI-SSOD controller, this disturbance can be seen as a part of the load disturbance.

3. Stability and limit cycles analysis for a FOPDT process

In this section, the stability properties of the two event-based control systems are studied. In particular, the considered process is a FOPDT system, described by the following transfer function:

$$P(s) = \frac{K}{\tau s + 1} e^{-Ls} \tag{4}$$

where K is the process gain (which is assumed to be positive without loss of generality), $\tau > 0$ is the time constant and $L \geq 0$ is the apparent dead time. Then, we can write

$$Y(s) = \frac{K}{\tau s + 1} e^{-Ls} \left(U(s) + \frac{D}{s} \right) \tag{5}$$

where $Y(s)$ is the Laplace transform of the process output $y(t)$, $U(s)$ is the Laplace transform of the control action $u(t)$ and D is the amplitude of a constant load disturbance.

It is important to notice that, because of the nonlinear nature of the controllers, the system can reach an equilibrium point, or can present a limit cycle around an equilibrium point or can be unstable. In general, the behavior of each equilibrium point can be different.

3.1. Normalization of the systems

In this subsection, the SSOD-PI and PI-SSOD controller systems are mathematically described. Then, we introduce a normalization in order to show that the two systems have the same stability properties when there are not exogenous signals.

The SSOD-PI controlled system can be described in the state-space form as:

$$\begin{cases} \dot{x}_1(t) = -\dfrac{1}{\tau} x_1(t) + \dfrac{K}{\tau}(x_2(t) + K_p e^*(t) + D) \\ \dot{x}_2(t) = K_i e^*(t) \\ y(t) = x_1(t) \\ e(t) = r - y(t) \\ e^*(t) = \mathrm{ssod}(e(t-L); \Delta, \beta) \\ x_1(0) = x_{10} = y_0 \\ x_2(0) = x_{20} = IE_0 \end{cases} \tag{6}$$

where r is the constant reference signal, $x_1(t)$ is the process state, $x_2(t)$ is the integral term of the PI controller, y_0 is the initial output value, and IE_0 is the initial value of the integral term.

It is important to note that the constant load disturbance can be grouped with x_{20} without loss of generality, therefore the SSOD-PI controller can exactly compensate a constant load disturbance.

The PI-SSOD controlled system can be described in the state-space form as:

$$\begin{cases} \dot{x}_1(t) = -\dfrac{1}{\tau} x_1(t) + \dfrac{K}{\tau}(u^*(t) + D) \\ \dot{x}_2(t) = \dfrac{K_p}{\tau}(x_1(t) - K(u^*(t) + D)) + K_i(r - x_1(t)) \\ y(t) = x_1(t) \\ u(t) = x_2(t) \\ u^*(t) = \mathrm{ssod}(u(t-L); \Delta, \beta) \\ x_1(0) = y_0 \\ x_2(0) = x_{20} = K_p(r - y_0) + K_i IE_0 \end{cases} \tag{7}$$

where r is the constant reference signal, $x_1(t)$ is the process state, $x_2(t)$ is the control action, y_0 is the initial output value and x_{20} is the initial value of the PI control action.

By studying the equilibrium points of (7) and remembering that $u^*(t) = j\Delta\beta$, it is trivial to find that an equilibrium exist only if $r = K(j\Delta\beta + D)$. Thus, the PI-SSOD controller cannot exactly compensate a generic constant load disturbance. In Section 3.3, we present a method to compensate the "unquantizable" part of the load disturbance, therefore for the stability analysis we consider that the exogenous signals are equal to zero in the PI-SSOD controlled scheme.

The system (6) can be normalized in (8) by choosing: $\tilde{t} = (t/\tau)$, $\tilde{y}(\tilde{t}) = ((y(t) - r)/\Delta)$, $\tilde{v}^*(\tilde{t}) = (e^*(t)/\Delta)$, $\tilde{x}_1(\tilde{t}) = ((x_1(t) - Kx_2(t))/\Delta)$, $\tilde{x}_2(\tilde{t}) = (K(x_2(t)/\Delta))$, $K_1 = KK_p\beta$, $K_2 = KK_i\beta\tau$, $a = K_1 - K_2$, $\tilde{x}_{10} = ((y_0 - r - K(x_{20} + D))/\Delta)$, $\tilde{x}_{20} = ((K(x_{20} + D))/\Delta)$ and $l = (L/\tau)$ which is called normalized dead time.

Also the system (7) can be normalized in (8) by choosing: $\tilde{t} = (t/\tau)$, $\tilde{y}(\tilde{t}) = ((y(t) - r)/\Delta)$, $\tilde{v}^*(\tilde{t}) = (Ku^*(t)/\Delta)$, $K_1 = KK_p\beta$, $K_2 = KK_i\beta\tau$, $a = K_1 - K_2$, $\tilde{x}_1(\tilde{t}) = -(ax_1(t)/\Delta)$, $\tilde{x}_2(\tilde{t}) = -((ax_1(t) + x_2(t))/\Delta)$, $\tilde{x}_{10} = -(ay_0/\Delta)$, $\tilde{x}_{20} = -((ay_0 + x_{20})/\Delta)$ and $l = (L/\tau)$. Thus, for the two event-based controller strategies we have the following normalized system (see Appendix A):

$$\begin{cases} \dot{\tilde{x}}_1(\tilde{t}) = -\tilde{x}_1(\tilde{t}) + a\tilde{v}^*(\tilde{t}) \\ \dot{\tilde{x}}_2(\tilde{t}) = K_2\tilde{v}^*(\tilde{t}) \\ \tilde{v}(\tilde{t}) = \tilde{x}_2(\tilde{t}) + \tilde{x}_1(\tilde{t}) \\ \tilde{v}^*(\tilde{t}) = \mathrm{ssod}(-\tilde{v}(\tilde{t} - l); 1, 1) \\ \tilde{x}_1(0) = \tilde{x}_{10} \\ \tilde{x}_2(0) = \tilde{x}_{20} \end{cases} \tag{8}$$

The normalized system (8) represents both the two presented controllers, and therefore their stability properties are the same, with the exception of the disturbance exact compensation. The normalized system scheme is shown in Fig. 7. The SSOD-PI and the PI-SSOD cases and their normalization formulae are summarized in Table 1.

3.2. Stability analysis of the normalized system

In this subsection, we demonstrate that it is possible to find a region of the parameter space $K_1 - K_2$ where the system is certainly marginally or asymptotically stable for all the equilibrium points. This region can then be divided into two regions: the first where the

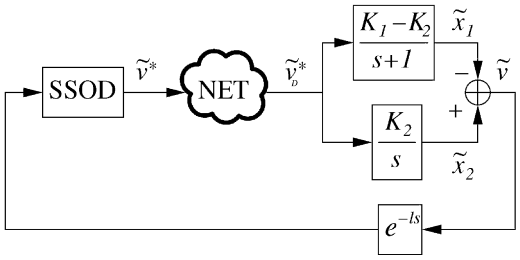

Fig. 7. Scheme of the normalized system.

system trajectory can tend to a limit cycle and the second where the system surely does not present a limit cycle.

In particular, we prove that if the system (8) is unstable then also the corresponding continuous time PI-controlled system with the same proportional and integral gains is unstable, therefore the instability region of SSOD-PI and PI-SSOD controllers is the same that the continuous-time PI controller.

To prove this necessary condition, we have first to demonstrate that if the signal $\tilde{v}(\tilde{t})$ diverges, then also its derivative diverges for all the positive proportional and integral gains (which are the only ones with a physical meaning).

Proposition 1. *In the system (8), for all the positive proportional gains, if the absolute value of $\tilde{v}(\tilde{t})$ tends to infinity then also the absolute value of its derivative tends to infinity.*

Proof. The proof consists of demonstrating the following implication:

$$\lim_{\tilde{t}\to\infty}|\tilde{v}(\tilde{t})| = \infty \Rightarrow \lim_{\tilde{t}\to\infty}|\dot{\tilde{v}}(\tilde{t})| = \infty$$

By contradiction, we suppose that:

$$\lim_{\tilde{t}\to\infty}|\tilde{v}(\tilde{t})| = \infty \quad \text{and} \quad \dot{\tilde{v}}(\tilde{t}) \in [\alpha, \beta], \quad \text{with}\, \alpha < \beta, \;\; \alpha, \beta \in \mathbb{R},$$

therefore,

$$\lim_{t\to\infty}\frac{\dot{\tilde{v}}(\tilde{t})}{\tilde{v}(\tilde{t})} = 0. \tag{9}$$

It is important to note that if the derivative $\dot{\tilde{v}}(\tilde{t})$ is limited, then the difference between $\tilde{v}^*(\tilde{t})$ and $\tilde{v}(\tilde{t})$ is a bounded signal, called $f(\tilde{t})$. In fact, the differences between $\tilde{v}(\tilde{t})$ and $\tilde{v}(\tilde{t}-l)$ and between $\tilde{v}^*(\tilde{t})$ and $-\tilde{v}(\tilde{t}-l)$ are bounded. Thus, using (8) and (9) and $\tilde{v}^*(\tilde{t}) = -\tilde{v}(\tilde{t}) + f(\tilde{t}) = -(\tilde{x}_2(\tilde{t}) + \tilde{x}_1(\tilde{t})) + f(\tilde{t})$, it is possible to write:

$$\lim_{\tilde{t}\to\infty}\frac{\dot{\tilde{x}}_2(\tilde{t}) + \dot{\tilde{x}}_1(\tilde{t})}{\tilde{x}_2(\tilde{t}) + \tilde{x}_1(\tilde{t})} = 0,$$

$$\lim_{\tilde{t}\to\infty}\frac{(K_2 + a)(f(\tilde{t}) - \tilde{x}_2(\tilde{t}) - \tilde{x}_1(\tilde{t})) - \tilde{x}_1(\tilde{t})}{\tilde{x}_2(\tilde{t}) + \tilde{x}_1(\tilde{t})} = 0$$

Table 1
Summary of the SSOD-PI and PI-SSOD cases.

SSOD-PI controller (Fig. 5)

System equations

$$\begin{cases} \dot{x}_1(t) = -\frac{1}{\tau}x_1(t) + \frac{K}{\tau}(x_2(t) + K_p e^*(t) + D) \\ \dot{x}_2 = K_i e^*(t) \\ y(t) = x_1(t) \\ e(t) = r - y(t) \\ e^*(t) = \text{ssod}(e(t-L); \Delta, \beta) \\ x_1(0) = y_0 \\ x_2(0) = x_{20} \end{cases}$$

Normalized variables

$\tilde{t} = \frac{t}{\tau}$ $\quad \tilde{y}(\tilde{t}) = \frac{y(t)-r}{\Delta}$

$l = \frac{L}{\tau}$ $\quad \tilde{v}^*(\tilde{t}) = \frac{e^*(t)}{\Delta}$

$K_1 = KK_p\beta$ $\quad \tilde{x}_1(\tilde{t}) = \frac{x_1(t) - Kx_2(t)}{\Delta}$

$K_2 = KK_i\beta\tau$ $\quad \tilde{x}_2(\tilde{t}) = K\frac{x_2(t)}{\Delta}$

$a = K_1 - K_2$ $\quad \tilde{x}_{10} = \frac{y_0 - r - K(x_{20} + D)}{\Delta}$

$\tilde{x}_{20} = \frac{K(x_{20} + D)}{\Delta}$

Normalized scheme (Fig. 7)

$$\begin{cases} \dot{\tilde{x}}_1(\tilde{t}) = -\tilde{x}_1(\tilde{t}) + a\tilde{v}^*(\tilde{t}) \\ \dot{\tilde{x}}_2(\tilde{t}) = K_2\tilde{v}^*(\tilde{t}) \\ \tilde{v}(\tilde{t}) = \tilde{x}_2(\tilde{t}) + \tilde{x}_1(\tilde{t}) \\ \tilde{v}^*(\tilde{t}) = \text{ssod}(-\tilde{v}(\tilde{t}-l); 1, 1) \\ \tilde{x}_1(0) = \tilde{x}_{10} \\ \tilde{x}_2(0) = \tilde{x}_{20} \end{cases}$$

PI-SSOD controller (Fig. 6)

$$\begin{cases} \dot{x}_1(t) = -\frac{1}{\tau}x_1(t) + \frac{K}{\tau}(u^*(t) + D) \\ \dot{x}_2(t) = \frac{K_p}{\tau}(x_1(t) - K(u^*(t) + D)) + K_i(r - x_1(t)) \\ y(t) = x_1(t) \\ u(t) = x_2(t) \\ u^*(t) = \text{ssod}(u(t-L); \Delta, \beta) \\ x_1(0) = y_0 \\ x_2(0) = x_{20} \end{cases}$$

$\tilde{t} = \frac{t}{\tau}$ $\quad \tilde{y}(\tilde{t}) = \frac{y(t)-r}{\Delta}$

$l = \frac{L}{\tau}$ $\quad \tilde{v}^*(\tilde{t}) = \frac{Ku^*(t)}{\Delta}$

$K_1 = KK_p\beta$ $\quad \tilde{x}_1(\tilde{t}) = -\frac{ax_1(t)}{\Delta}$

$K_2 = KK_i\beta\tau$ $\quad \tilde{x}_2(\tilde{t}) = -\frac{ax_1(t) + x_2(t)}{\Delta}$

$a = K_1 - K_2$ $\quad \tilde{x}_{10} = -\frac{ay_0}{\Delta}$

$\tilde{x}_{20} = -\frac{ay_0 + x_{20}}{\Delta}$

and finally,

$$\lim_{t\to\infty}\frac{\tilde{x}_1(\tilde{t})}{\tilde{x}_2(\tilde{t})+\tilde{x}_1(\tilde{t})} = -(a+K_2) \tag{10}$$

where $f(\tilde{t})/(\tilde{x}_2(\tilde{t})+\tilde{x}_1(\tilde{t}))$ tends to zero.

From Eq. (10) we can state that $\tilde{x}_1(\tilde{t})$ and $-\tilde{v}(\tilde{t})$ diverge in the same way. Thus, $\tilde{x}_2(\tilde{t}) = \tilde{v}(\tilde{t}) - \tilde{x}_1(\tilde{t})$ can diverge as $2\tilde{v}(\tilde{t})$ or be bounded, and therefore:

$$\lim_{t\to\infty}\frac{\dot{\tilde{x}}_2(\tilde{t})}{\tilde{v}(\tilde{t})} = K_2 = 0$$

which is an absurd if $K_2 \neq 0$, otherwise we have that $\tilde{x}_2(\tilde{t})$ is constant and equal to $\tilde{x}_{20}$, and therefore (10) becomes:

$$\lim_{t\to\infty}\frac{\tilde{x}_1(\tilde{t})}{\tilde{x}_{20}+\tilde{x}_1(\tilde{t})} = 1 = -a$$

which is an absurd because, if $K_2 = 0$, $a = K_1 \geq 0$. □

Using Proposition 1 it is possible to state the following necessary condition.

Proposition 2. *If the closed-loop system (8) is unstable then the same system controlled by a continuous-time PI controller with the same value of K_1 and K_2 is also unstable.*

Proof. The time interval $\Delta\tilde{t}$ between two events can be calculated as the ratio between 1 and the mean value of the derivative of $\tilde{v}(\tilde{t})$, called $\dot{\bar{\tilde{v}}}$, in the interval, in fact:

$$1 = \left|\int_{\tilde{\xi}}^{\tilde{\xi}+\Delta t}\dot{\tilde{v}}(\tilde{t})d\tilde{t}\right| = |\dot{\bar{\tilde{v}}}|\Delta\tilde{t}.$$

From Proposition 1 we know that if the absolute value of the output tends to infinity also the absolute value of its derivative diverges to infinity. Thus, the time interval $\Delta\tilde{t}$ decreases to zero and (8) becomes a continuous-time controller, which proves the proposition. □

The following propositions deal with the parameters region for which there are not surely limit cycles and the system converges to the equilibrium state. The idea of the proof consists in finding the parameters for which if the system is in the state $j = j_{eq} \pm n$, where j_{eq} is the equilibrium state and $n > 0$, it cannot evolve to the state $j = j_{eq} \mp n$ and therefore it has to tend to the equilibrium state. The next proposition addresses the cases of I and PI controllers. The goal of the proposition is to find the values of K_1–K_2 parameters for which there are not periodic trajectories which involve more than the equilibrium state.

Proposition 3. *In a system described by (8), with $K_1 \geq 0$ and $K_2 > 0$, then limit cycles cannot occur if the normalized gains K_1 and K_2 are inside the region of the first quadrant delimited by the following parametric curve:*

$$\begin{cases} K_1(\tilde{t}_1) = \begin{cases} \dfrac{\tilde{t}_1 - 2l + 2e^l - e^{l+\tilde{t}_1}(2l-\tilde{t}_1+2) + 4e^l\sinh(l)}{2le^l - 2le^{l+\tilde{t}_1} + 2\tilde{t}_1 e^l\sinh(l)} & \text{if } l \leq \tilde{t}_1 < 2l \\ \dfrac{2}{\tilde{t}_1} - \dfrac{2l-\tilde{t}_1}{\tilde{t}_1(e^{l-\tilde{t}_1}-1)} & \text{if } 2l \leq \tilde{t}_1 \leq \bar{t}_1 \end{cases} \\ K_2(\tilde{t}_1) = \begin{cases} \dfrac{2-2e^{2l}+2e^{l+\tilde{t}_1}-2e^l}{\tilde{t}_1 - \tilde{t}_1 e^{2l} + 2le^{l+\tilde{t}_1} - 2le^l} & \text{if } l \leq \tilde{t}_1 < 2l \\ \dfrac{2}{\tilde{t}_1} & \text{if } 2l \leq \tilde{t}_1 \leq \bar{t}_1 \end{cases} \end{cases} \tag{11}$$

where $\tilde{t}_1 \in [l, \bar{t}_1]$ with $K_1(\bar{t}_1) = 0$.

Proof. To demonstrate the proposition, we find the locus of parameters $K_1 - K_2$ which allows the system to have the smallest limit cycles that involve three states (namely $j = \{-1, 0, 1\}$). In this way, all the sets of $\{K_1, K_2\}$ inside of the part of the first quadrant delimited by that locus do not allow the system to have a limit cycle.

A three-level limit cycle with a period $\tilde{T}$ has to verify the following conditions

- $\tilde{v}(k\tilde{T} - l) = 0$ which implies $\tilde{v}^*(k\tilde{T}) = 0$;
- $\tilde{v}(k\tilde{T} - l + \tilde{t}_1) = 1$ which implies $\tilde{v}^*(k\tilde{T} + \tilde{t}_1) = 1$;
- $\tilde{v}(k\tilde{T} - l + \tilde{t}_1 + \tilde{t}_2) = 0$ which implies $\tilde{v}^*(k\tilde{T} + \tilde{t}_1 + \tilde{t}_2) = 0$;
- $\tilde{v}(k\tilde{T} - l + \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3) = -1$ which implies $\tilde{v}^*(k\tilde{T} + \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3) = 0$;
- $\tilde{v}(k\tilde{T} - l + \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3 + \tilde{t}_4) = -1$ which implies $\tilde{v}^*(k\tilde{T} + \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3 + \tilde{t}_4) = 0$;

where $T = \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3 + \tilde{t}_4$. Because of the symmetric nature of the SSOD technique, we can suppose that the limit cycle is symmetric. Thus: $\tilde{x}(\tilde{t}) = -\tilde{x}(\tilde{t} - T/2)$, $\tilde{t}_3 = \tilde{t}_1$ and $\tilde{t}_4 = \tilde{t}_2$.

During a time interval $\tilde{t}_i := [\tilde{t}_a, \tilde{t}_b]$ between two events, the signal $\tilde{v}^*(\tilde{t})$ is constant, and therefore it is easy to find the evolution of the state variables as the sum of the free and forced response to a constant input:

$$\tilde{x}(\tilde{t}_b) = R(\tilde{t}_b - \tilde{t}_a)\tilde{x}(\tilde{t}_a) + F(\tilde{t}_b - \tilde{t}_a; K_2, a)\tilde{u}(\tilde{t}) \tag{12}$$

where:

$$R(\tilde{t}) = \begin{bmatrix} e^{-\tilde{t}} & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$F(\tilde{t}; K_2, a) = \begin{bmatrix} a(1-e^{-\tilde{t}}) \\ K_2\tilde{t} \end{bmatrix},$$

Using (12) it is possible to write the following equations, which describe the evolution of the system during a time period:

$$\tilde{x}^1 = \tilde{x}(k\tilde{T} + \tilde{t}_1) = R(\tilde{t}_1)\tilde{x}^0 + F(\tilde{t}_1; K_2, a)$$
$$\tilde{x}^2 = \tilde{x}(k\tilde{T} + \tilde{t}_1 + \tilde{t}_2) = R(\tilde{t}_2)\tilde{x}^1 = -\tilde{x}^0.$$

where $\tilde{x}^0 = \tilde{x}(k\tilde{T})$. These equations can be rewritten as:

$$\begin{aligned} \tilde{x}^1 &= (I + R(\tilde{t}_2)R(\tilde{t}_1))^{-1}F(\tilde{t}_1; K_2, a) \\ \tilde{x}^0 &= -(I + R(\tilde{t}_2)R(\tilde{t}_1))^{-1}R(\tilde{t}_2)F(\tilde{t}_1; K_2, a) \end{aligned} \tag{13}$$

It is possible to write also the following switching conditions:

$$\begin{aligned} \tilde{x}^a &= \tilde{x}(k\tilde{T} + \tilde{t}_1 - l) \\ \tilde{v}^a &= \tilde{v}(k\tilde{T} + \tilde{t}_1 - l) = \tilde{x}_2^a + \tilde{x}_1^a = 0 \\ \tilde{x}^b &= \tilde{x}(k\tilde{T} + \tilde{t}_1 + \tilde{t}_2 - l) \\ \tilde{v}^b &= \tilde{v}(k\tilde{T} + \tilde{t}_1 + \tilde{t}_2 - l) = \tilde{x}_2^b + \tilde{x}_1^b = 1 \end{aligned} \tag{14}$$

Conditions (13) and (14) describe all the possible three-level limit cycles in a SSOD-PI controlled FOPDT process. To prove the proposition we want to find the smallest limit cycle and the correspondent normalized gains. Thus, we can add the following condition

$$\sup\tilde{v}(\tilde{t}) = \sup(\tilde{x}_2(\tilde{t}) + \tilde{x}_1(\tilde{t})) = 1. \tag{15}$$

To find the superior of $\tilde{v}(\tilde{t})$ we have to study the sign of its derivative in the intervals $[0, \tilde{t}_1]$ and $[\tilde{t}_1, \tilde{t}_1 + \tilde{t}_2]$ (note that, because of symmetry it is possible to study only a half of the period).

In the first interval, the state evolution is

$$\tilde{x}(\tilde{t}) = R(\tilde{t} - \tilde{t}_1)\tilde{x}^0 + F(\tilde{t} - \tilde{t}_1),$$

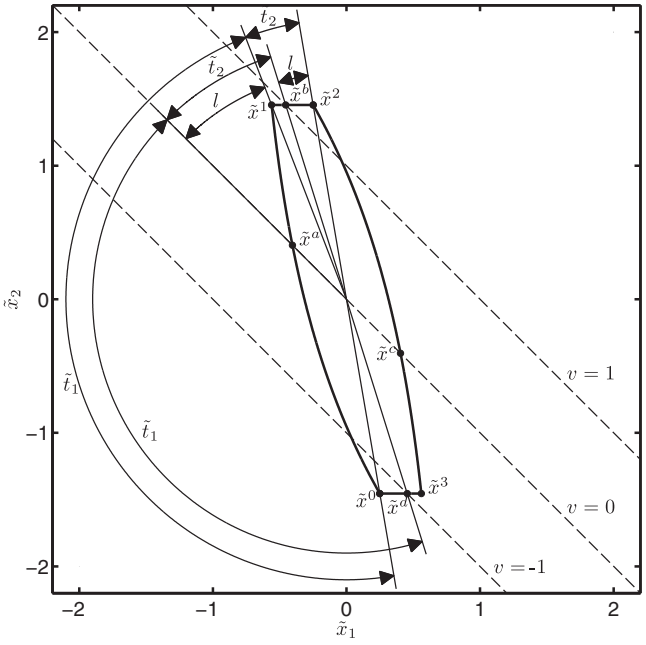

Fig. 8. Example of limit cycle with $a < 0$ where the arc arrows represent the time intervals.

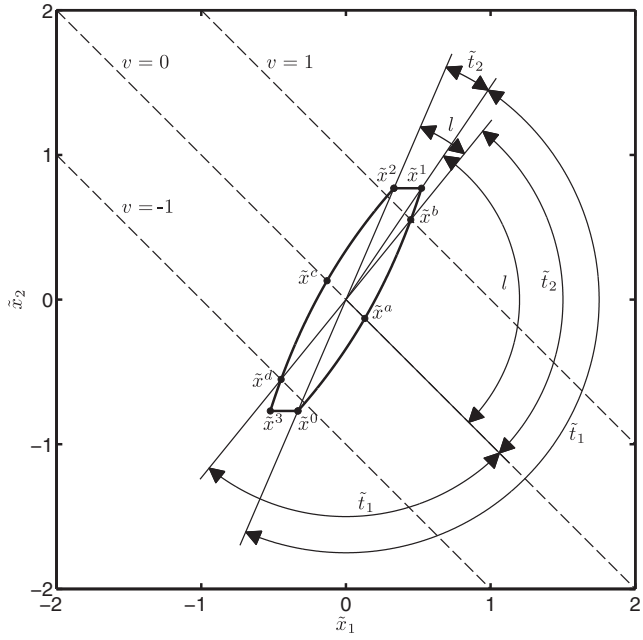

Fig. 9. Example of limit cycle with $a > 0$ where the arc arrows represent the time intervals.

where $\tilde{x}^0$ can be easily found from (13). Deriving $\tilde{v}(\tilde{t})$ with respect to the time, we obtain:

$$\dot{v}(\tilde{t}) = \frac{K_2 e^{\tilde{t}}(1 + e^{\tilde{t}_1+\tilde{t}_2}) + a e^{\tilde{t}_1}(1 + e^{\tilde{t}_2})}{e^{\tilde{t}}\,(e^{\tilde{t}_1}\,e^{\tilde{t}_2} + 1)},$$

where the numerator is positive if $a > -K_2 e^{(\tilde{t}-\tilde{t}_1)}(1 + e^{(\tilde{t}_1+\tilde{t}_2)})/(1 + e^{(\tilde{t}_2)})$ and by remembering that $K_1 = K_2 + a \geq 0$, it is possible to obtain that $\dot{v}(\tilde{t})$ which is always positive in $\tilde{t} \in [0, \tilde{t}_1]$.

In the second interval, the state evolution is:

$$\tilde{x}(\tilde{t}) = R(\tilde{t} - \tilde{t}_1)\tilde{x}^1,$$

therefore $\tilde{v}(\tilde{t}) = e^{-(\tilde{t}-\tilde{t}_1)}\tilde{x}_1^1 + \tilde{x}_2^1$ and its derivative is equal to $\dot{v}(\tilde{t}) = -e^{-(\tilde{t}-\tilde{t}_1)}\tilde{x}_1^1$. The value of $\tilde{x}_1^1$ can be easily found by solving (13) and it results:

$$\tilde{x}_1^1 = a e^{\tilde{t}_2}\,\frac{e^{\tilde{t}_1} - 1}{e^{\tilde{t}_1+\tilde{t}_2} + 1},$$

therefore $\dot{v}(\tilde{t})$ in $\tilde{t} \in [\tilde{t}_1, \tilde{t}_1 + \tilde{t}_2]$ has an inverse sign with respect to a.

Summarizing, if $a < 0$ (the corresponding limit cycle is shown in Fig. 8) then the derivative $\dot{v}(\tilde{t})$ is positive in $\tilde{t} \in [0, \tilde{t}_1 + \tilde{t}_2]$ and negative in $\tilde{t} \in [\tilde{t}_1 + \tilde{t}_2, \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3 + \tilde{t}_4]$ and therefore the maximum is $\tilde{v}(\tilde{t}_1 + \tilde{t}_2)$, else if $a > 0$ (the corresponding limit cycle is shown in Fig. 9) then the derivative $\dot{v}(\tilde{t})$ is positive in $\tilde{t} \in [0, \tilde{t}_1]$ and $\tilde{t} \in [\tilde{t}_1 + \tilde{t}_2, \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3]$, negative in $\tilde{t} \in [\tilde{t}_2, \tilde{t}_1 + \tilde{t}_2]$ and $\tilde{t} \in [\tilde{t}_1 + \tilde{t}_2, \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3 + \tilde{t}_4]$ and therefore the maximum is $\tilde{v}(\tilde{t}_1)$.

Considering the case $a < 0$, the variable $\tilde{v}(\tilde{t})$ assumes the value 1 in $t = \tilde{t}_1 + \tilde{t}_2$, where the state coincides with the point $-\tilde{x}^0$, but also in the point $\tilde{x}^b$ (namely, l instant before $-\tilde{x}^0$) the signal $\tilde{v}(\tilde{t})$ is equal to 1. This situation is possible if and only if the time interval $\tilde{t}_2$ is infinite and $-\tilde{x}^0$ is a stationary point (namely, $-\tilde{x}^0 = [0,\ 1]^T$) or l is a multiple of the period. Considering the first case (namely, $\tilde{t}_2 \to \infty$), since $\tilde{x}_2^0 = -1$ and the second variable changes linearly during the time interval $[0, \tilde{t}_1]$ and it is constant in the time interval $[\tilde{t}_1, \tilde{t}_1 + \tilde{t}_2]$, we can write:

$$\tilde{x}_2^2 = -\tilde{x}_2^0 = \tilde{x}_2^0 + K_2\tilde{t}_1$$

and obtain:

$$K_2 = \frac{2}{\tilde{t}_1}.$$

To obtain the expression of a, it is important to note that $\tilde{t}_1 > l$ because the trajectory intersects the switching surface $\tilde{x}_2 + \tilde{x}_1 = 1$ when it assumes the value $\tilde{x}^b$ and l instants after the trajectory does not have yet intersected the other switching surface $\tilde{x}_2 + \tilde{x}_1 = 0$ (because it is in a stationary value). Thus, the expression of $\tilde{x}^a$ (see (13) and (14)) becomes:

$$\tilde{x}^1 = R(l)\tilde{x}^a + F(l; K_2, a)$$

or equally:

$$\tilde{x}^a = R(l)^{-1}(\tilde{x}^1 - F(l; K_2, a)).$$

Finally, imposing $\tilde{x}_2^a + \tilde{x}_1^a = 0$ it is easy to find the relationship between a and $\tilde{t}_1$, and therefore (remember that $K_1 = K_2 + a$) the relation between K_1 and $\tilde{t}_1$, presented in (11).

Considering now the second case ($l = N\tilde{T} = 2n(\tilde{t}_1 + \tilde{t}_2)$) with $n \in \mathbb{N}^+$ (thus $\tilde{t}_1 < l$ and $\tilde{t}_2 < l$), it is possible to note that $\tilde{x}^a = \tilde{x}(\tilde{t}_1 - l) = \tilde{x}^1$ and $\tilde{x}^b = \tilde{x}(\tilde{t}_1 + \tilde{t}_2 - l) = \tilde{x}^2$. Remembering that $\tilde{v}^a = 0$ and $\tilde{v}^b = 1$, it is possible to find:

$$K_2 = \frac{2}{\tilde{t}_1}\,\frac{1}{1 - e^{-\tilde{t}_1}}$$

and this condition is less stringent than the other case ($\tilde{t}_2 \to \infty$), noting that $\tilde{t}_1$ in the second case is lower than l and in the first case is greater than l.

Considering the case $a \geq 0$, the variable $\tilde{v}(\tilde{t})$ assumes the value 1 in $t = \tilde{t}_1$, where the state vector is equal to $\tilde{x}^1$, and also in this case it corresponds to the point $\tilde{x}^b$. In this situation the delay l has to be equal to $\tilde{t}_2 + n\tilde{T}$ (with $n \in \mathbb{N}^+$), in fact when $\tilde{v}^*(\tilde{t})$ is equal to 1 (namely in the interval $[0, \tilde{t}_1]$) the signal $\tilde{v}(\tilde{t})$ monotonically increases. Similarly to the case $a < 0$, it is possible to demonstrate that the case with $n = 0$ is the most stringent case.

Thus, in the time instant $\tilde{t}_1$ the system has to switch into the state with $\tilde{v}^* = 0$ to satisfy condition (15), which means that in the

instant $\tilde{t}_1 - l$ the variable $\tilde{v}(\tilde{t})$ has to be equal to 0. Because of these conditions, we can write:

$$\tilde{x}^1 = (I + R(l)R(\tilde{t}_1))^{-1} F(\tilde{t}_1; K_2, a)$$

$$\tilde{x}^1 = \tilde{x}^b$$

$$\tilde{x}^1 = R(l)\tilde{x}^a + F(l, K_2, a)$$

$$\tilde{v}_1^b = 1$$

$$\tilde{v}_1^a = 0.$$

Solving this system of equations we can find the relationship between K_2, a (and therefore K_1) and $\tilde{t}_1$ shown in (11). □

The case of SSOD-P and P-SSOD controllers is addressed in the following proposition. Also in this case the goal is to find the values of K_1 for which there are not periodic trajectories which involve more than the equilibrium state.

Proposition 4. *In a system described by (8), if* $K_1 < \frac{1}{1-e^{-l}}$ *then limit cycles cannot occur.*

Proof. Also in this case, the proof of the proposition mainly consist in finding the value of K_1 (namely, $a = K_1$ when $K_2 = 0$) which allows the system to attain the smallest limit cycle (namely, $j = -1, 0, 1$). Because of the symmetry of the map between $\tilde{v}(\tilde{t})$ and $\tilde{v}^*(\tilde{t})$, the limit cycles are symmetrical, thus $\tilde{v}(\tilde{t}) = -\tilde{v}(\tilde{t} - T/2)$. The signals $\tilde{v}^*(\tilde{t})$ and $\tilde{v}(\tilde{t})$ are defined as:

$$\tilde{v}^*(\tilde{t}) = \begin{cases} 1 & \text{if } \tilde{t} \in [0, \tilde{t}_1] \\ 0 & \text{if } \tilde{t} \in [\tilde{t}_1, \tilde{t}_1 + \tilde{t}_2] \\ -1 & \text{if } \tilde{t} \in [\tilde{t}_1 + \tilde{t}_2, \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3] \\ 0 & \text{if } \tilde{t} \in [\tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3, \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3 + \tilde{t}_4] \end{cases} \tag{16}$$

$$\tilde{v}(\tilde{t}) = \begin{cases} \tilde{v}(0)e^{-t} + K_1(1 - e^{-t}) & \text{if } \tilde{t} \in [0, \tilde{t}_1] \\ \tilde{v}(\tilde{t}_1)e^{-(t-\tilde{t}_1)} & \text{if } \tilde{t} \in [\tilde{t}_1, \tilde{t}_1 + \tilde{t}_2] \\ -\tilde{v}(0)e^{-t} - K_1(1 - e^{-(t-\tilde{t}_1-\tilde{t}_2)}) & \text{if } \tilde{t} \in [\tilde{t}_1 + \tilde{t}_2, \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3] \\ -\tilde{v}(\tilde{t}_1)e^{-(t-\tilde{t}_1-\tilde{t}_2)} & \text{if } \tilde{t} \in [\tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3, \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3 + \tilde{t}_4] \end{cases} \tag{17}$$

where $\tilde{t}_1 = \tilde{t}_3$ and $\tilde{t}_2 = \tilde{t}_4$. Imposing the condition on periodicity and symmetry $v(0) = -v(\tilde{t}_1 + \tilde{t}_2)$, it is possible to find $v(0) = -K_1((e^{-\tilde{t}_2}(1 - e^{-\tilde{t}_1}))(1 + e^{-\tilde{t}_1-\tilde{t}_2}))$.

In order to find the smallest limit cycle, we have to impose that the maximum of $\tilde{v}(\tilde{t})$ equals to 1. The maximum of $\tilde{v}(\tilde{t})$ is found by studying the sign of $\dot{\tilde{v}}(\tilde{t})$, which is:

$$\dot{\tilde{v}}(\tilde{t}) = \begin{cases} K_1 \dfrac{1 + e^{-\tilde{t}_2}}{1 + e^{-\tilde{t}_1-\tilde{t}_2}} e^{-\tilde{t}} & \text{if } \tilde{t} \in [0, \tilde{t}_1] \\ -K_1 \dfrac{1 - e^{-\tilde{t}_1}}{1 + e^{-\tilde{t}_1-\tilde{t}_2}} e^{-(\tilde{t}-\tilde{t}_1)} & \text{if } \tilde{t} \in [\tilde{t}_1, \tilde{t}_1 + \tilde{t}_2] \end{cases}$$

Thus, the maximum is in $\tilde{t} = \tilde{t}_1$ and it is equal to $\max \tilde{v}(\tilde{t}) = K_1((1 - e^{-\tilde{t}_1})(1 + e^{-\tilde{t}_1-\tilde{t}_2}))$. The maximum is equal to 1 only if:

$$K_1 = \frac{1 + e^{-\tilde{t}_1-\tilde{t}_2}}{1 - e^{-\tilde{t}_1}} \tag{18}$$

The values of $\tilde{t}_1$ and $\tilde{t}_2$ can be found using the following switching conditions:

$$\tilde{v}(\tilde{t}_1 - l + nT) = 0 \tag{19}$$

$$\tilde{v}(\tilde{t}_1 + \tilde{t}_2 - l + nT) = 1 \tag{20}$$

where $T = \tilde{t}_1 + \tilde{t}_2 + \tilde{t}_3 + \tilde{t}_4$ is the time period, and $n \in \mathbb{Z}$. Because $\max \tilde{v}(\tilde{t}) = \tilde{v}(\tilde{t}_1) = 1$, the condition (20) implies $\tilde{v}(\tilde{t}_1 + \tilde{t}_2 - l + nT) = \tilde{v}(\tilde{t}_1)$ or equally $l - nT = \tilde{t}_2$.

Considering first the case with $n = 0$, we can obtain $t_2 = l$ and the condition (19) cannot be verified in the fourth time interval ($\tilde{v}(\tilde{t})$ is always positive), therefore $\tilde{t}_1 \geq \tilde{t}_2$ and it is possible to obtain that $\tilde{t}_1 = \log(1 + e^l - e^{-l})$ and $K_1 = (1/(1 - e^{-l}))$ which proves the proposition.

Considering now the cases with $n > 0$, therefore $\tilde{t}_1 < (l/2)$ and $\tilde{t}_2 < (l/2)$. Using (18), we can state:

$$\frac{1 + e^{-\tilde{t}_1-\tilde{t}_2}}{1 - e^{-\tilde{t}_1}} \geq \frac{1}{1 - e^{-\tilde{t}_1}} \geq \frac{1}{1 - e^{-l}}.$$

Thus, limit cycles with a time period lower than l cannot occur using $K_1 \leq (1/(1 - e^{-l}))$. □

Note also that Propositions 3 and 4 are necessary and sufficient conditions to avoid the presence of limit cycles. In other words, in the region of the plane $K_1 - K_2$ delimited by these conditions there are surely no limit cycles, whereas out of this region there is surely at least a possible limit cycle. Actually, for some values of initial conditions, reference signal and load disturbance it is possible that the system state does not reach the limit cycle trajectory but tends to the equilibrium point.

Another important consideration is that the parameter Δ does not have influence on the stability analysis and therefore it can be chosen, in principle, using only considerations about the desired precision, namely, the maximum admissible steady-state error. In practical cases, measurement noise should be taken into account. Actually, it is advisable to select a value of Δ greater than the noise band, that is the peak-to-peak amplitude of the noise signal (see Section 5.1).

3.3. Disturbance estimation

As already mentioned (see Section 3.1), because of its quantized nature, the PI-SSOD control strategy (7) cannot compensate the unquantized part of the load disturbance, therefore a bimodal limit cycle surely arises. This limit cycle with period T involves two consecutive states $j_l + 1$ and j_l, where t^* is the interval time where $j = j_l + 1$. Thus, the control action assumes only the values $j_l\Delta\beta$ and $(j_l + 1)\Delta\beta$, as shown in (21).

$$u(t) = \begin{cases} (j_l + 1)\Delta\beta & \text{if } t \in [0, t^*] \\ j_l\Delta\beta & \text{if } t \in [t^*, T] \end{cases} \tag{21}$$

By applying the definition of limit cycle, the trends of the process and the controller quantities during a cycle are periodic. Considering now the integrated error $IE(t)$ and its Laplace transform $IE(s)$, which can be calculated as

$$IE(s) = \frac{R(s)}{s} - \frac{K}{s(\tau s + 1)}(U(s) + d)$$

which corresponds to the following differential equation:

$$\tau\dot{IE}(t) + IE(t) = r(t) - K(u(t) + d + \tau\dot{r}(t)).$$

By integrating over a period and remembering that, by hypothesis, r and d are constant and $IE(t)$ is periodic, we obtain:

$$\tau(\dot{IE}(T) - \dot{IE}(0)) + (IE(T) - IE(0)) = rT - K\,dT - K\int_0^T u(t)dt = 0$$

and finally:

$$\overline{u} = j_l\Delta\beta + \Delta\beta\frac{t^*}{T} = \frac{r}{K} - d \tag{22}$$

where $\overline{u}$ is the mean value of $u(t)$ during a period and $\hat{d} := \Delta\beta(t^*/T)$ is the "unquantized" part of the control action necessary to compensate the load disturbance.

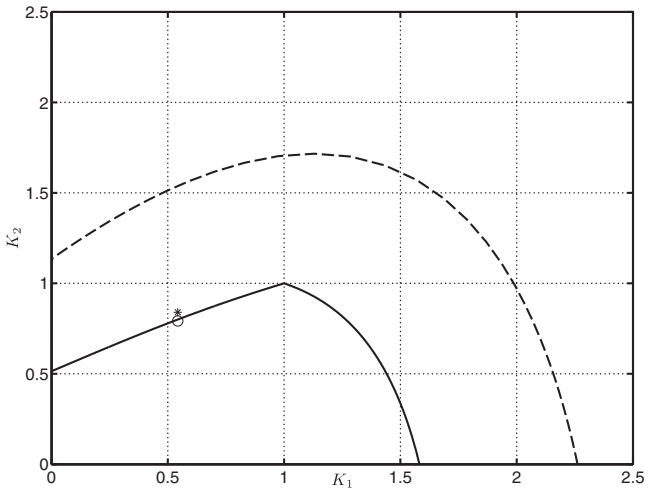

Fig. 10. Plane with the necessary condition on stability (dashed line), and condition on absence of limit cycles (solid line). The marker ∘ denotes the set of parameters used to obtain the results shown in Fig. 11, the marker ∗ denotes the set of parameters used to obtain the results shown in Fig. 12.

In real plants, the trajectory tends to the limit cycles but does not reach it. Hence, the hypothesis of periodicity is only an approximation, therefore an exact compensation of the disturbances is virtually impossible. For this reason, a small deadband is introduced, the width of this band is upper-limited by Δ and it can be set by taking into account the numerical errors in the calculus of t^* and T and the presence of noise.

4. Practical issues

The SSOD-PI and PI-SSOD control techniques can be easily implemented in very simple industrial controllers, because they do not require a large computational effort or complex routines. The algorithms for the sensor and the control units, can be outlined using the pseudocodes shown in Table 2 (for sake of brevity the initialization is not shown), where h is the control unit sampling period (note that the two algorithms do not require any synchronization and they can have different sampling periods). Note that the actuator unit task of PI-SSOD controller has to only keep the last received value until the next event. Note also that the steps 8–9 of the PI-SSOD controller's Sensor and Control Unit Task verify if a bimodal cycles is occurred (namely, the last three states were j, $j-1$ and j) and introduce the load disturbance compensation.

5. Simulation results

In this section, simulation results are given. In particular, illustrative examples on the presence of limit cycles and the exogenous signal are presented, then the two SSOD control strategies are compared with other event-based sampling techniques and a discrete time, using as a high-order plus dead time system approximated with the half-rule method and using the SIMC tuning rules [27].

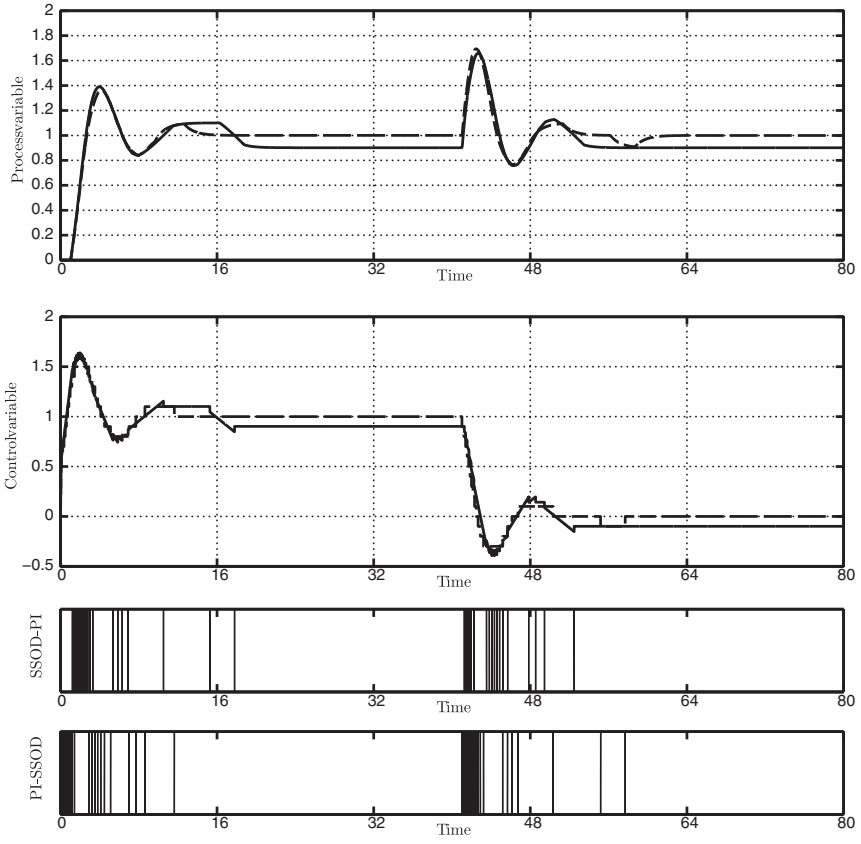

Fig. 11. Responses for process (23) using SSOD-PI and PI-SSOD controllers with the first set of parameters. First plot (from the top): process variable for the SSOD-PI controller (solid line) and PI-SSOD controller (dashed line). Second plot: control variable for the SSOD-PI controller (solid line) and PI-SSOD controller (dashed line). Third plot: events of the SSOD-PI controller system. Fourth plot: events of the PI-SSOD controller system.

Table 2
Pseudocodes of the SSOD-PI and PI-SSOD controllers.

SSOD-PI controller	PI-SSOD controller
Sensor unit 1. calculate e; 2. if $\|e - e^*\| \leq \Delta$ then go to 5; 3. if $e > e^*$ then $e^* = \Delta\lfloor \frac{e}{\Delta} \rfloor$ else $e^* = \Delta\lceil \frac{e}{\Delta} \rceil$; 4. send βe^* to the control unit; 5. end. *Control and actuator unit* 1. if a new βe^* is received then update the value of e^*; 2. calculate $u_{i,k} = u_{i,k-1} + K_i \beta e^* h$ and $u_k = u_{i,k} + K_p \beta e^*$; 3. send u_k to the actuator; 4. set $u_{i,k-1} = u_{i,k}$; 5. end.	*Sensor and control unit task* 1. calculate e; 2. calculate $u_{i,k} = u_{i,k-1} + K_i eh$ and $u_k = u_{i,k} + K_p e$; 3. set $u_{i,k-1} = u_{i,k}$; 4. if $\|u - u^*\| \leq \Delta$ then go to 11; 5. set $u^{*,2} = u^{*,1}$, $t^{*,2} = t^{*,1}$, $u^{*,1} = u^*$ and $t^{*,1} = t^*$; 6. save in t^* the time interval between the last and the current event. 7. if $u > u^*$ then $u^* = \Delta\lfloor \frac{u}{\Delta} \rfloor$ else $u^* = \Delta\lceil \frac{u}{\Delta} \rceil$; 8. if $u^{*,2} = u^*$ then go to 9 else go to 10; 9. if $u^* > u^{*,1}$ then $\hat{D} = \Delta\beta \frac{t^*}{t^{*,1}+t^*}$; 10. send $\beta u^* + \hat{D}$ to the actuator unit; 11. end.

5.1. Illustrative Example 1

In this example the following FOPDT system is controlled by the SSOD-PI and PI-SSOD controllers:

$$P(s) = \frac{1}{s+1} e^{-s} \tag{23}$$

Using Proposition 3 it is possible to find the region of $K_1 - K_2$ (shown in Fig. 10) where the limit cycles are avoided. In order to show the validity of Proposition 3, two sets of parameters are chosen, the first ($K_p = 0.542$ and $K_i = 0.792$) slightly inside the no limit cycle region (it is indicated with a circle in Fig. 10) and the second ($K_p = 0.542$ and $K_i = 0.840$) which does not belong to this region (it is indicated with a '*' in Fig. 10). The values of Δ and β are set equal to 0.1 and 1, respectively.

A unit set-point step signal is applied to the control system at time $t = 0$ and a unit load disturbance step is applied at time $t = 40$. Figs. 11 and 12 show the control system response using the first and the second set, respectively. It is possible to note that Proposition 3 is verified, as there is not a limit cycle in Fig. 11 while there is in Fig. 12.

Another important result of Propositions 3 and 4 is that the parameter Δ does not influence the stability and limit cycle properties, but only the system precision and the number of events. This aspect is highlighted in Fig. 13, where the controller is tuned with the second set of parameters and Δ is set equal to 0.01. It is possible to note (see the zooms inside the figure) that the system trajectories tend to limit cycles with the same period of Fig. 12 but the process variables are closer to the set-point. However, the number of events increases significantly.

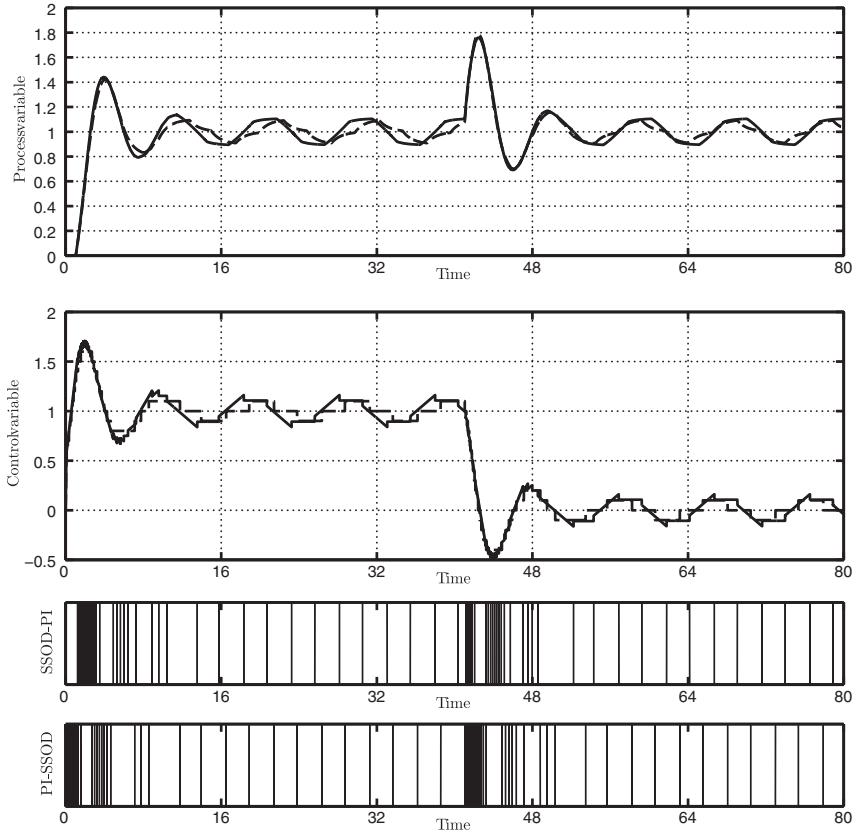

Fig. 12. Responses for Process (23) using SSOD-PI and PI-SSOD controllers with the second set of parameters. First plot (from the top): process variable for the SSOD-PI controller (solid line) and PI-SSOD controller (dashed line). Second plot: control variable for the SSOD-PI controller (solid line) and PI-SSOD controller (dashed line). Third plot: events of the SSOD-PI controller system. Fourth plot: events of the PI-SSOD controller system.

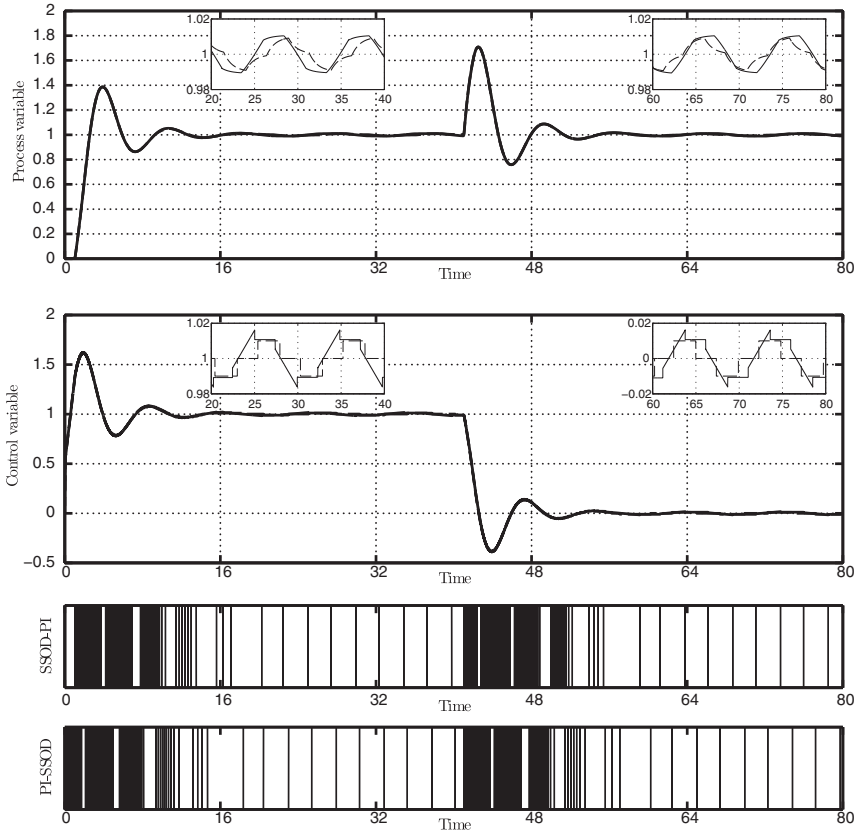

Fig. 13. Responses for process (23) using SSOD-PI and PI-SSOD controllers with the second set of parameters and a smaller value of Δ. First plot (from the top): process variable for the SSOD-PI controller (solid line) and PI-SSOD controller (dashed line). Second plot: control variable for the SSOD-PI controller (solid line) and PI-SSOD controller (dashed line). Third plot: events of the SSOD-PI controller system. Fourth plot: events of the PI-SSOD controller system.

Finally, the issue of the selection of Δ in the presence of measurement noise is analyzed by considering the same case of Fig. 11 but with a measurement noise with a noise band equal to 0.08. By choosing $\Delta = 0.1$ (that is, slightly greater than the noise band), the results shown in Fig. 14 are obtained. It appears that there is not a significant performance decrement while there is only a slight increment of the number of events. Actually, the SSOD-PI controller is less sensitive to the noise than the PI-SSOD controller because the SSOD quantization reduces the effects of the noise on the PI control unit.

5.2. Illustrative Example 2

In this example the exogenous signal compensation task of the PI-SSOD controller is shown. The considered process is the same of the Example 1 (see (23)). The controller parameters are set as $K_p = 0.542$, $K_i = 0.792$, $\Delta = 0.1$ and $\beta = 1$; the set-point value is set equal to 1.05, in this way the compensation is necessary (namely, $r \neq jK\Delta\beta$, $\forall j \in \mathbb{Z}$). In Fig. 15, it is possible to note that at the time instant $t = 23$, the control variable is set equal to 1.052 using (22). Because the compensation is exact, a small deadband (equal to $0.1K\Delta = 0.01$) is applied on the error signal calculus.

5.3. Illustrative Example 3

In this example, the high order system [27]

$$P(s) = \frac{(-0.3s+1)(.08s+1)}{(2s+1)(s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3} \tag{24}$$

is considered. The process is approximated as a FOPDT system with $K = 1$, $\tau = 2.5$ and $L = 1.47$ using the half rule.

The controller gains are tuned using the SIMC rules [27] and they result in the following controller parameters $K_p = 0.85$ and $K_i = 0.34$ which are in the no limit cycles region according to Proposition 3. The presented control strategies are compared with the SOD-PI controller presented in [26], the event-based controller (denoted as Årzén) presented in [14], the PIDplus presented in [2] and a discrete time-driven PI controller (denoted as DT-PI) with a sample period equal to $\tau/5$, as suggested in [2].

The parameters Δ and β are set equal to 0.1 and 1, respectively. The parameter t_{max}, necessary in [14,2], is set equal to τ. The SOD-PI parameters are set as follows: $\Delta_p = \Delta$, $\Delta_i = \Delta\tau$, $\epsilon = \Delta$. The deadband present in the PI-SSOD controller is set equal to $0.1K\Delta$.

Fig. 16 shows the system responses when the exogenous signals are a unit set-point step, applied at the time instant $t = 0$, and a load disturbance step, with amplitude equal to 0.95 and applied at the time instant $t = 30$. It is important to note that the performance of SSOD-PI, PI-SSOD and SOD-PI controllers are the fastest ones but present more overshoot with respect to the PIDplus controller, which is less aggressive also during the disturbance rejection task. The Arzén PI controller presents a peak on the control variable caused by the method used to elaborate the integral action (which is solved in the PIDplus controller). The performance of the controllers, shown in Table 3, confirm these considerations. Another important consideration is that the SSOD-PI and PI-SSOD control strategies present the smallest numbers of events, because of the absence of the t_{max} triggering condition. It is possible to see that the event-based control strategies (with the exception of the Arzén PI controller) have better performance than a discrete PI technique with a high sampling period. Note that, reducing the sampling period of DT-PI controller increases the performance but also the number of the events.

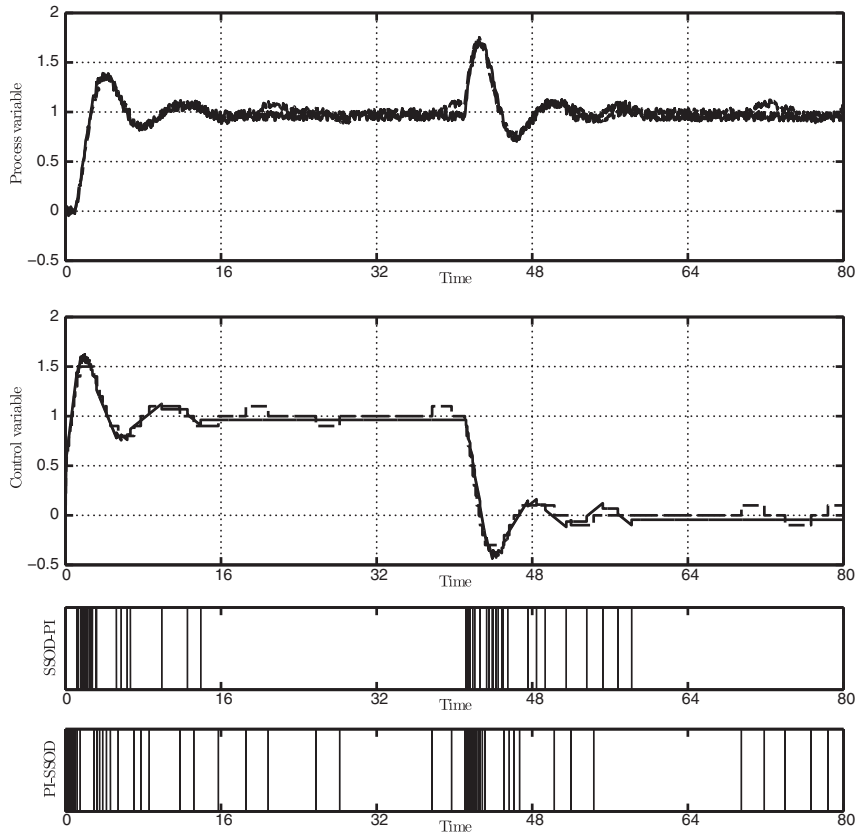

Fig. 14. Zoom of Fig. 13. First plot (from the top): process variable for the SSOD-PI controller (solid line) and PI-SSOD controller (dashed line). Second plot: control variable for the SSOD-PI controller (solid line) and PI-SSOD controller (dashed line). Third plot: events of the SSOD-PI controller system. Fourth plot: events of the PI-SSOD controller system.

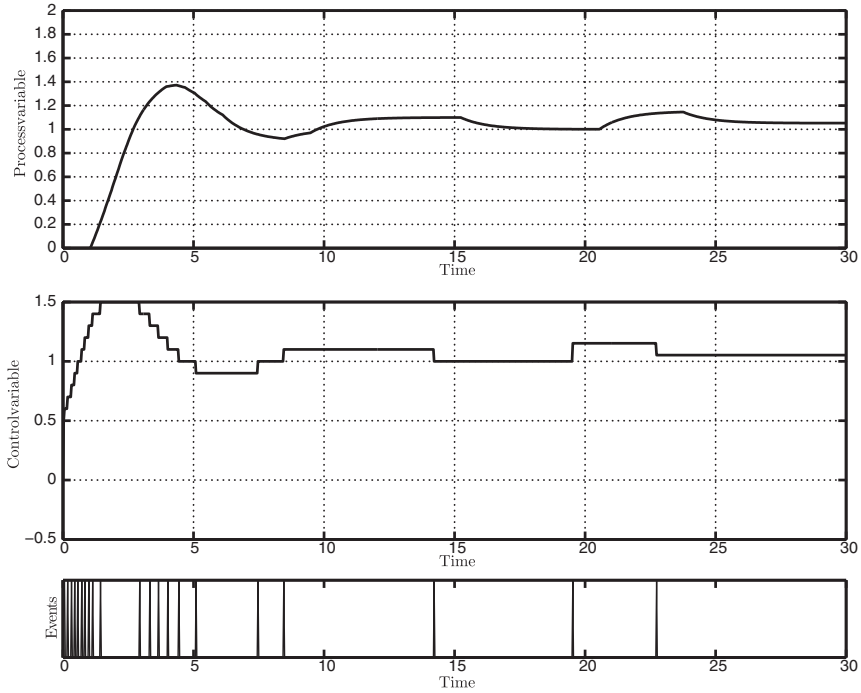

Fig. 15. Response to a step of amplitude 1.05 for Process (23) using PI-SSOD controller. First plot (from the top): process variable. Second plot: control variable. Third plot: events of the PI-SSOD controller system.

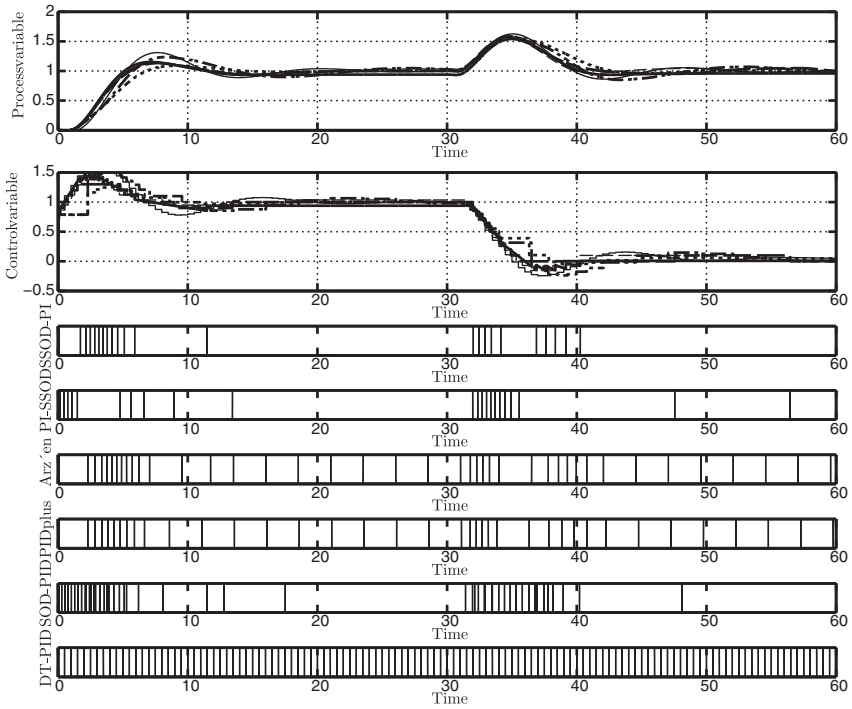

Fig. 16. Responses for Process (24). First plot (from the top): process variable for the SSOD-PI controller (thick solid line), PI-SSOD controller (thick dashed line), Årzén controller (thick dot-dashed line), PIDplus controller (thick dotted line), SOD-PI controller (thin dashed line) and DT-PI controller (thin solid line). Second plot: control variable for the SSOD-PI controller (thick solid line), PI-SSOD controller (thick dashed line), Årzén controller (thick dot-dashed line), PIDplus controller (thick dotted line), SOD-PI controller (thin dashed line) and DT-PI controller (thin solid line). From third to eighth plot: events of the SSOD-PI, PI-SSOD, Årzén, PIDplus, SOD-PI and DT-PI controller systems, respectively.

Remark 5. It is important to note that the Event Based-PI (EB-PI) strategies different from the SSOD-PI and PI-SSOD methods can be also applied when the three control agents are separated.

6. Experimental results

In order to prove the effectiveness of the presented control strategies in practical applications, a laboratory-scale setup made by Quanser has been employed (see Fig. 17). Specifically, the experimental setup is a four tanks system where just the lower left tank has been considered for level control.

Because the apparent dead-time of the system is very small with respect to its dominant time constant, and in order to provide a significant result, a time delay of 2 s has been added via software at the plant output. The FOPDT model has been obtained by applying a least-squares procedure to the response of a series of open-loop steps, resulting in:

$$P(s) = \frac{82.1}{22.2s+1}e^{-2s}. \tag{25}$$

The percentage of the output variation that is explained by the model is around 90%.

Table 3
Performance of the controllers. IAE_Δ: integrated absolute error (computed only when the error signal is out of the band $\pm\Delta$). OV: overshoot. n_e: number of events. st_Δ: setting time to a band of $\pm\Delta$ around the set-point.

	IAE	OV	n_e	st_Δ
SSOD-PI	8.510	15.185	32	9.032
PI-SSOD	7.821	13.360	30	9.135
Årzén	10.143	23.465	49	17.730
PIDplus	8.342	8.957	46	5.996
SOD-PID	7.855	12.547	58	9.062
DT-PID	9.341	30.997	120	14.897

The experiment consists in a set-point step change from 10 cm to 20 cm, then a load disturbance is applied, using an additional pump.

The PI controller are tuned using the following parameters: $K_p = 0.04$, $T_i = 16$, which are inside of the no limit cycle zone. The t_{max} (when it is necessary) is set equal to τ. The parameter Δ is set

Fig. 17. Equipment used for experiments.

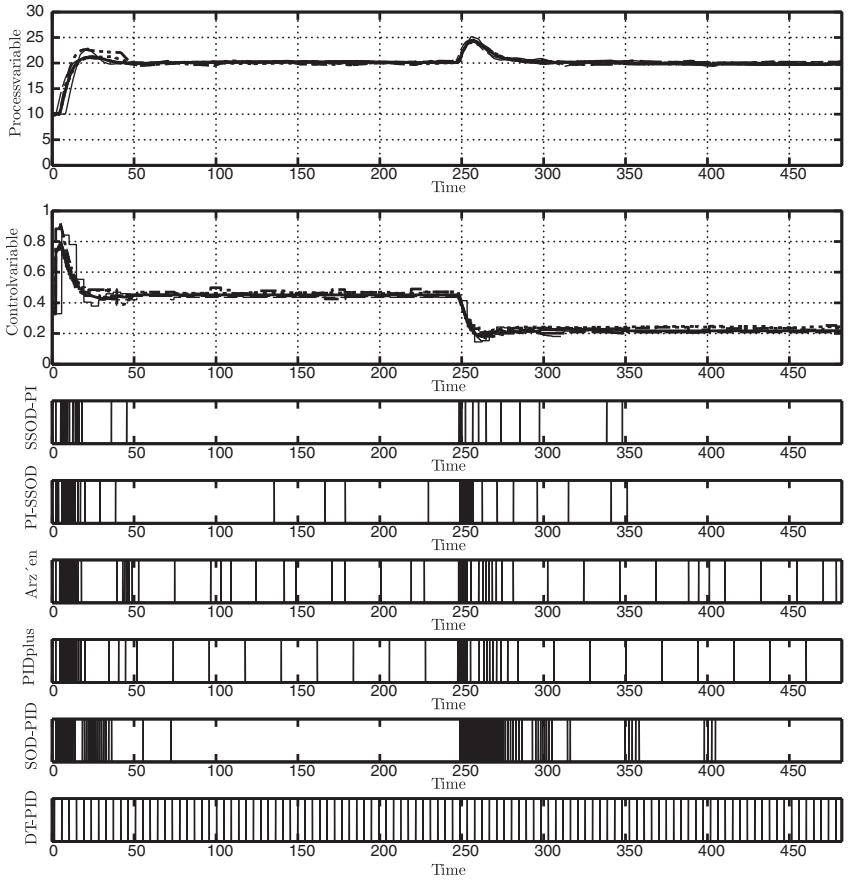

Fig. 18. Responses for process shown in Fig. 17. First plot (from the top): process variable for the SSOD-PI controller (thick solid line), PI-SSOD controller (thick dashed line), Årzén controller (thick dot-dashed line), PIDplus controller (thick dotted line), SOD-PI controller (thin dashed line) and DT-PI controller (thin solid line). Second plot: control variable for the SSOD-PI controller (thick solid line), PI-SSOD controller (thick dashed line), Årzén controller (thick dot-dashed line), PIDplus controller (thick dotted line), SOD-PI controller (thin dashed line) and DT-PI controller (thin solid line). From third to eighth plot: events of the SSOD-PI, PI-SSOD, Årzén, PIDplus, SOD-PI and DT-PI controller systems, respectively.

equal 0.4 (bigger than the noise band of the sensor) for all the controllers except the PI-SSOD strategy, where it is set equal to 0.02 (chosen in order to obtain the same number of events of the SSOD-PI controller). The deadband of the PI-SSOD controller is set equal to $0.1K\Delta = 0.16$. The parameters Δ_i and ϵ of the SOD-PI controller are set equal to 0.4 and 1, respectively. Fig. 18 and Table 4 show the experimental results and the performance index obtained using the same controllers of Example 3.

Also in this case, the best performance is achieved using the SSOD-PI, the PI-SSOD, the SOD-PI and the PIDplus controllers. However, with the two control strategies presented in this paper it is possible to reduce the number of events (and therefore the number of transmissions) without a significant loss of performance with respect to the SOD-PI and the PIDplus controllers.

Table 4
Performance of the controllers. IAE_Δ: integrated absolute error (computed only when the error signal is out of the band $\pm\Delta$). OV: overshoot. n_e: number of events. st_Δ: setting time to a band of $\pm\Delta$ around the set-point when the set-point signal is a unitary step.

	IAE_Δ	OV	n_e	st_Δ
SSOD-PI	166.826	12.802	30	36.100
PI-SSOD	174.432	10.744	45	35.700
Årzén	222.763	26.515	77	111.600
PIDplus	185.070	13.038	64	45.200
SOD-PID	166.531	9.240	148	72.300
DT-PID	209.121	29.145	107	30.500

7. Conclusions

In this paper, we have addressed the stability issue and the presence of limit cycles for two event-based PI control systems, based on the symmetric-send-on-delta event-triggering method. In particular, the main advantages of this event detection strategy have been outlined. Then, stability and limit cycles properties are investigated. In particular, necessary conditions on the system instability have been determined.

Limit cycles are surely avoided if the controller gain is inside the set of controller parameters described using two sufficient conditions. Both the stability and limit cycle properties do not depend on the event-triggering method, which can be tuned in order to handle the trade-off between the number of events and the system precision.

Simulation and experimental results have confirmed the effectiveness of the approach which represents a valuable step in the development of easy-to-use tuning rules for this kind of controllers.

Appendix A. Normalization formulae

A.1. SSOD-PI controller

The SSOD-PI controlled FOPDT process is described by Eq. (6). As mentioned in Section 3.1, the stability analysis is done without

exogenous signals (namely, $r = D = 0$). Thus, Eq. (6) can be rewritten as:

$$\begin{cases} \dot{x}_1(t) = -\dfrac{1}{\tau}x_1(t) + \dfrac{K}{\tau}(x_2(t) + K_p e^*(t)) \\ \dot{x}_2(t) = K_i e^*(t) \\ y(t) = x_1(t) \\ e(t) = -y(t) \\ e^*(t) = \text{ssod}(e(t-L); \Delta, \beta) \end{cases}$$

The signal $e^*(t)$ can be rewritten as:

$$e^*(t) = \Delta\beta\text{ssod}\left(\frac{e(t-L)}{\Delta}; 1, 1\right).$$

In order to normalize the system, we scale the time variable $\tilde{t} = (t/\tau)$ (thus, $l = (L/\tau)$). In this way, the derivatives becomes: $(dx_1/dt) = (dx_1/d\tilde{t})(1/\tau)$ and $(dx_2/dt) = (dx_2/d\tilde{t})(1/\tau)$. Thus, we can rewrite the system equation as:

$$\begin{cases} \dot{x}_1(\tilde{t}) = -x_1(\tilde{t}) + K(x_2(\tilde{t}) + K_p e^*(\tilde{t})) \\ \dot{x}_2(t) = K_i \tau e^*(\tilde{t}) \\ y(\tilde{t}) = x_1(\tilde{t}) \\ v(\tilde{t}) = x_1(\tilde{t}) \\ e^*(\tilde{t}) = \Delta\beta\text{ssod}(-\dfrac{v(\tilde{t}-l)}{\Delta}; 1, 1) \end{cases}$$

where $v(\tilde{t}) = -e(\tilde{t})$. We define $\tilde{v}^*(\tilde{t}) = \text{ssod}\left(\frac{v(\tilde{t}-l)}{\Delta}; 1, 1\right)$, therefore the system equations become:

$$\begin{cases} \dot{x}_1(\tilde{t}) = -x_1(\tilde{t}) + Kx_2(\tilde{t}) + KK_p\beta\Delta v^*(\tilde{t}) \\ \dot{x}_2(t) = K_i\beta\Delta\tau v^*(\tilde{t}) \\ y(\tilde{t}) = x_1(\tilde{t}) \\ v(\tilde{t}) = x_1(\tilde{t}) \\ \tilde{v}^*(\tilde{t}) = \text{ssod}\left(-\dfrac{v(\tilde{t}-l)}{\Delta}; 1, 1\right) \end{cases}$$

To make the stability analysis easier we diagonalize the dynamic equation using the following transformation matrix:

$$\Theta = \begin{bmatrix} 1 & -K \\ 0 & K \end{bmatrix}$$

Finally, defining $\tilde{\mathbf{x}}(\tilde{t}) = (\Theta\mathbf{x}(\tilde{t})/\Delta)$, $\tilde{y}(\tilde{t}) = (y(\tilde{t})/\Delta)$, $K_1 = KK_p\beta$, $K_2 = KK_i\tau\beta$ and $a = K_1 - K_2$, the system equations become (8).

A.2. PI-SSOD controller

The PI-SSOD controlled FOPDT process is described by Eq. (7). As mentioned, the stability analysis is done without exogenous signals. Thus, Eq. (7) can be rewritten as:

$$\begin{cases} \dot{x}_1(t) = -\dfrac{1}{\tau}x_1(t) + \dfrac{K}{\tau}u^*(t) \\ \dot{x}_2(t) = \dfrac{K_p}{\tau}(x_1(t) - Ku^*(t)) + K_i(-x_1(t)) \\ y(t) = x_1(t) \\ u(t) = x_2(t) \\ u^*(t) = \text{ssod}(u(t-L); \Delta, \beta) \end{cases}$$

The signal $u^*(t)$ can be rewritten as:

$$u^*(t) = \Delta\beta\text{ssod}\left(\frac{u(t-L)}{\Delta}; 1, 1\right).$$

In order to normalize the system, we scale the time variable $\tilde{t} = (t/\tau)$ (thus, $l = (L/\tau)$). In this way, the derivatives becomes: $(dx_1/dt) = (dx_1/d\tilde{t})(1/\tau)$ and $(dx_2/dt) = (dx_2/d\tilde{t})(1/\tau)$. Thus, we can rewrite the system equation as:

$$\begin{cases} \dot{x}_1(\tilde{t}) = -x_1(\tilde{t}) + Ku^*(\tilde{t}) \\ \dot{x}_2(\tilde{t}) = K_p(x_1(\tilde{t}) - Ku^*(\tilde{t})) - K_i\tau x_1(\tilde{t}) \\ y(\tilde{t}) = x_1(\tilde{t}) \\ v(\tilde{t}) = x_2(\tilde{t}) \\ u^*(\tilde{t}) = \text{ssod}(-v(\tilde{t}-l); \Delta, \beta) \end{cases}$$

where $v(\tilde{t}) = -u(\tilde{t}) = -x_2(\tilde{t})$. We define $\tilde{v}^*(\tilde{t}) = \text{ssod}\left(\frac{v(\tilde{t}-l)}{\Delta}; 1, 1\right)$, therefore the system equations become:

$$\begin{cases} \dot{x}_1(\tilde{t}) = -x_1(\tilde{t}) + K\Delta\beta\tilde{v}^*(\tilde{t}) \\ \dot{x}_2(\tilde{t}) = (K_p - K_i\tau)x_1(\tilde{t}) - KK_p\Delta\beta\tilde{v}^*(\tilde{t}) \\ y(\tilde{t}) = x_1(\tilde{t}) \\ v(\tilde{t}) = -x_2(\tilde{t}) \\ v^*(\tilde{t}) = \text{ssod}(-\dfrac{v(\tilde{t}-l)}{\Delta}; 1, 1) \end{cases}$$

To make the stability analysis easier we diagonalize the dynamic equation using the following transformation matrix:

$$\Theta = \begin{bmatrix} K_p - K_i\tau & 0 \\ K_i\tau - K_p & -1 \end{bmatrix}$$

Finally, defining $\tilde{\mathbf{x}}(\tilde{t}) = (\Theta\mathbf{x}(\tilde{t})/\Delta)$, $\tilde{y}(\tilde{t}) = (y(\tilde{t})/\Delta)$, $K_1 = KK_p\beta$, $K_2 = KK_i\tau\beta$ and $a = K_1 - K_2$, the system equations become (8).

References

[1] A. O'Dwyer, Handbook of PI and PID Tuning Rules, Imperial College Press, London, UK, 2006.
[2] T. Blevins, PID advances in industrial control, in: Preprints IFAC Conference on Advances in PID Control, Brescia, Italy, 2012.
[3] P. Otanez, J. Moyne, D. Tilbury, Using deadbands to reduce communication in networked control systems, in: Proceedings of American Control Conference, Anchorage, USA, 2002.
[4] M. Miskowicz, Send-on-delta: an event-based data reporting strategy, Sensors 6 (2006) 49–63.
[5] A. Pawlowskyy, J.L. Guzmán, F. Rodríguez, M. Berenguel, J. Sánchez, S. Dormido, Event-based control and wireless sensor network for greenhouse diurnal temperature control: a simulated case study, in: Proceedings 13th IEEE International Conference on Emerging Technologies and Factory Automation, 2008.
[6] U. Tiberi, C.F. Lindberg, A.J. Isaksson, A self-trigger PI controller for processes with deadtime, in: Preprints of the IFAC Conference on Advances in PID Control, Brescia, Italy, 2012.
[7] X. Wang, M.D. Lemmon, Self-triggered feedback control systems with finite-gain $\mathbb{L}_2$ stability, IEEE Transactions on Automatic Control 54 (3) (2009) 452–467.
[8] M. Mazo Jr., A. Anta Martínez, P. Tabuada, An ISS self-triggered implementation of linear controllers, Automatica 46 (8) (2010) 1310–1314.
[9] J. Araujo, A. Anta, M. Mazo, J. Faria, A. Hernández, P. Tabuada, K.H. Johansson, Self-triggered control over wireless sensor and actuator networks., in: Proceedings IEEE International Conference on Distributed Computing in Sensor Systems, 2011 June, pp. 1–9.
[10] K.J. Åström, Event based control., in: A. Astolfi, L. Marconi (Eds.), Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori, 2008.
[11] V. Vasyutynskyy, K. Kabitzsh, Event-based control Overview and generic model, in: Proceedings of 8th IEEE International Workshop on Factory Communication Systems, 2010 May.
[12] W.P.M.H. Heemels, J.H. Sandee, P.P.J. Van Den Bosch, Analysis of event-driven controllers for linear systems, International Journal of Control 81 (4) (2008) 571–590.

[13] D. Lehmann, K.H. Johansson, Event-triggered PI control subject to actuator saturation, in: Preprints IFAC Conference on Advances in PID Control, Brescia, Italy, 2012.
[14] K.E. Årzén, A simple event-based PID controller, in: Proceedings of 14th World Congress of IFAC, Beijing, China, 1999.
[15] V. Vasyutynskyy, K. Kabitzsh, Implementation of PID controller with send-on-delta sampling, in: Proceedings of International Control Conference, 2006.
[16] M. Rabi, K.H. Johansson, Event-triggered strategies for industrial control over wireless networks, in: Proceedings of 4th Annual International Conference on Wireless Internet, Maui, HI, USA, 2008.
[17] S. Durand, N. Marchand, Further results on event-based PID controller, in: Proceedings of 10th European Control Conference (ECC'09), Budapest, Hungary, 2009.
[18] V. Vasyutynskyy, K. Kabitzsh, First order observers in event-based PID controls., in: Proceedings of IEEE Conference on Emerging Technologies & Factory Automation, 2009 September.
[19] V. Vasyutynskyy, K. Kabitzsh, A comparative study of PID control algorithms adapted to send-on-delta sampling, in: Proceedings of IEEE International Symposium on Industrial Electronics (ISIE), Bari, Italy, 2010.
[20] J. Sánchez, A. Visioli, S. Dormido, A two-degree-of-freedom PI controller based on events, Journal of Process Control 21 (2011) 639–651.
[21] J. Sánchez, A. Visioli, S. Dormido, Event-based PID control, in: R. Vilanova, A. Visioli (Eds.), PID Control in the Third Millennium, Springer, 2012, pp. 495–526.
[22] V. Vasyutynskyy, K. Kabitzsh, Towards comparison of deadband sampling types., in: In Proceedings of IEEE International Symposium on Industrial Electronics, 2007 June.
[23] E. Kofman, J. Braslavsky, Level crossing sampling in feedback stabilization under data rate constraints, in: Proceedings of 45th IEEE International Conference on Decision and Control, 2006.
[24] J. Sánchez, M.A. Guarnes, S. Dormido, A. Visioli, Comparative study of event-based control strategies: an experimental approach on a simple tank, in: Proceedings of European Control Conference, 2009.
[25] A. Cervin, K.J. Åström, On limit cycles in event-based control systems, in: Proceedings of 46th IEEE International Conference on Decision and Control, New Orleans, LA, USA, 2007, pp. 3190–3195.
[26] M. Beschi, A. Visioli, S. Dormido, J. Sánchez, On the presence of equilibrium points in PI control systems with send-on-delta sampling, in: Proceedings of 50th IEEE International Conference on Decision and Control and European Control Conference, Orlando, USA, 2011.
[27] S. Skogestad, Probably the best simple PID tuning rules in the world. In AIChE Annual meeting, Reno, NV, USA, 2001.