A PRACTICAL TUNING METHODOLOGY FOR EVENT-BASED PI CONTROL

Ángel Ruiz[a*], Jorge E. Jiménez[a], José Sánchez[b], Sebastián Dormido[b]

a Department of Computer Science and Numeric Analysis, University of Córdoba, Campus de Rabanales, 14071, Córdoba, Spain
** Fax: (+34)957218729; e-mail: aruiz@uco.es*
b Department of Computer Science and Automatic Control, UNED, Juan del Rosal 16, 28040, Madrid, Spain

Abstract

This paper is focused on the tuning of an event-based PI controller for First-Order Plus Time Delay Systems (FOPTD). In this work, a novel design and combination of a controller and event generator with an easy-to-use tuning methodology is presented. The event generator combines the Smith predictor structure with the Symmetric Send-On-Delta (*SSOD*) *sampling scheme to compensate the delay and trigger the events. The controller has an adaptive structure with the purpose of improving the set-point tracking and guaranteeing stability under conditions of uncertainty. The approach is focused on FOPTD systems but can be easily extended to higher order systems. Stability and robustness analyses are conducted, and the experimental results verify the effectiveness of the approach.*

Keywords: event-based control, PI controllers, Smith predictor, tuning.

1. Introduction

In the paradigm of automatic control, event-based strategies are presented as a solution to satisfy the current requirements of distributed process control. The increasing decentralization and large scale of current industrial processes and the number of involved devices demand more effective sampling schemes for signals [1, 2]. Applications based on networks (such as Networked Control Systems, NCS) are proof of this demand, and they have been especially addressed by researchers during the last decade (see [3]). In an NCS, the challenge is to reduce the exchange of information between distributed devices without losing performance. Given that energy and bandwidth are limited resources, a small control error may be tolerated, but reduction of the traffic load is a key issue [4, 5]. In addition, the possibility of losing data and experiencing a stochastic time delay is greater when there is more traffic. These aspects contribute to the degradation of the control loop performance. In this context, the aim of event-based approaches is achieving a satisfactory trade-off between the sampling effort and the loop performance, that is, an opportune use of sources and information channels [2].

In event-based sampling and control approaches, the design of an efficient scheme is not a trivial problem. The event condition may be any mathematical function (see [6] for typical event-based conditions) included in any agent involved in the control loop, thus obtaining different architectures and responses [7]. Moreover, when wireless communications are considered, the bandwidth usage and energy consumption should be considered as a part of the control loop design [1, 4]. In this way, the control loop is governed by events and, sampling and control actions become asynchronous tasks which makes the control loop analysis more challenging than time-based approaches. However, there is a general consensus on the sampling algorithm being Send-On-Delta (SOD) the commonest (see [8-10]).

From an industrial perspective, it is known that most processes can be well represented using first-order plus time delay models and controlled by a PI strategy. Probably for this reason, event-based systems under PID control have recently been addressed by numerous researchers. In some examples, the SOD strategy is applied on the system output and the controller is considered to be a PID with variable sampling period that undertakes control actions when a new sample is available [5, 10-12]. Other authors employ the same control strategy, and the SOD sampling is focused on the control error signal

[13, 14]. In these cases, the controller actions are synchronized with event instants even though the control algorithm is a PID. In other reports, authors consider the prior structure, but the controller signal is time-based and it progresses between samples [7, 9, 15]. With this strategy, the initial conditions at the controller change in each event but the control actions are performed with regular sampling between them. The current work is focused on this situation (i.e., the control error is the event-sampled signal). In other approaches, only the controller is event-based. It presents independent event conditions for proportional, integral, and derivative actions, and the control error measures must satisfy their event conditions to trigger control actions [5, 16, 17].

Independent of the structure, sampling and control actions depend on the evolution of signals, which affects the control quality, so that the limit cycle phenomenon may arise [18]. For this reason, aspects such as the tuning of these controllers and conditions for global stability or for the absence of limit cycles are topics that are far from being fully solved [5]. In particular, the question of controller tuning has not been properly addressed. This question is addressed only in [19] and [20]. In [19], although the authors define tuning conditions for a type of event-based PI control to avoid limit cycles, the rules become complex and its range is restricted, compromising the overall system performance. In [20], the tuning of a PI controller is focused on the energy consumption of wireless sensors and the tuning rules are analyzed using a trial-and-error method.

This paper proposes a novel event-based scheme with PI control for first-order-plus-time-delay (FOPTD) processes that prevents the appearance of limit cycles as a consequence of the process delay and that is independent of parameter tuning. Although there are relevant works that propose designs in which the absence of limit cycles is guaranteed (see [19, 21-26]), the issue of controller tuning and its influence on the system performance have not been properly investigated. The proposed scheme employs a novel event generator and a controller. The event generator combines the structure of the Smith predictor and the SSOD sampling scheme (presented in [15]). The Smith predictor represents a key-element for limit cycle prevention. By contrast, the controller structure is focused on the reduction of steady-state error. Under this scenario, sufficient conditions for demonstrating the existence of an equilibrium point around the set-point without limit cycles are given and uncertainty compensation mechanisms are proposed. All theoretical studies are supported by simulations, and they are proven experimentally.

This paper is organized as follows. In Section 2, the event-based control architecture is presented. In Section 3, the behavior of the system is described. The stability and absence of limit cycles with the proposed system are demonstrated in Section 4. In Sections 5 and 6, the tuning methodology and mechanisms for uncertainty compensation are explained. The results from simulation and experiments are presented in Sections 7 and 8, respectively, and conclusions are drawn in Section 9.

2. Structure of the event-based controller

The current trend in studies of event-based control systems is to combine agents with regular sampling and event-based sampling in control loop signals. The main reason for this combination is to consider the well-known automatic discrete control theory to explain the event-based behavior [27]. The proposed scheme has been designed following this line of research. Whereas the error signal is event-based sampled, the remaining signals are time-based sampled. Fig. 1 shows the blocks that define the closed-loop system. Three main blocks are considered: the process, the event generator, and the control signal generator. The scheme in Fig. 1 represents a small variation of the generic scheme introduced in [28] where the observer has been integrated with the event generator. The control loop is time-based between the controller and the event generator and the transmissions between the event-generator and the controller are event-based. According to this structure the event generator should not trigger and send a new sample until its event condition is satisfied, which interrupts the continuity of the

feedback loop and characterizes the overall scheme as an event-based control system. These blocks are explained in the following sections.

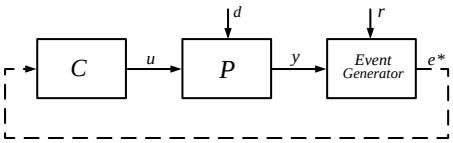

Figure 1: General scheme of the proposed event-based control. Solid lines denote continuous signal transmission, and dashed lines represent even-based signal transmission.

2.1 Process model structure

The process considered for explanations follows the structure of the FOPTD model:

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

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 process delay.

Remark 1. Although the FOPTD model can be considered as a special case, it is significant from a practical point of view because many industrial processes can be effectively represented in this way. Moreover, FOPTD models are extensively used to design tuning rules [29].

2.2 Event generator

The event generator characterizes the control loop as an event-based control system. The aim of this block is to reduce the number of error samples that are sent to the controller. This work proposes a novel solution that combines the structure of a Smith predictor with a recently presented variation of the most common event-based sampling algorithm, the typical SOD algorithm. A block diagram of the event generator is presented in Fig. 2 for clarity. This diagram is divided into two complementary parts: the prediction unit (based on the Smith predictor Structure) and the sampling unit (based on the SSOD sampling algorithm as in [15]). Both units are explained in the following paragraphs.

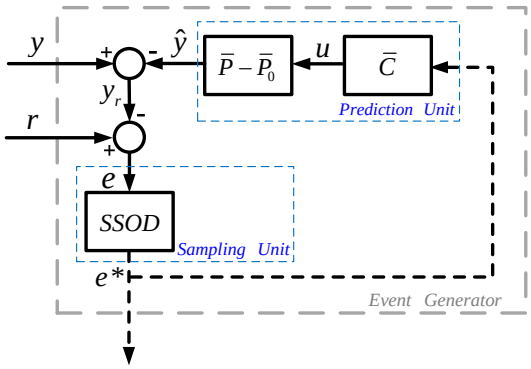

Figure 2: Units, blocks, and signals of the event generator block. Solid lines denote continuous signals, and dashed lines event-based signal transmission.

According to Fig. 2, the prediction unit involves the structure of a Smith predictor where

$$\bar{P}_0(s) = \frac{\bar{K}}{\bar{\tau} s + 1} \qquad \bar{P}(s) = \bar{P}_0(s) e^{-\bar{L}s} \qquad \bar{C}(s) = C(s) \tag{2}$$

with $\overline{K}$, $\overline{\tau}$, and $\overline{L}$ being identified parameters and $\overline{y}_0$, $\overline{y}$ being the output signals of $\overline{P}_0$ and $\overline{P}$, respectively. Additionally, a replica of the controller in Fig. 1, $\overline{C}$, is included in this unit. This is necessary to separate the sampler and control blocks so that the Smith predictor works correctly. Note that the assumption $C(s) = \overline{C}(s)$ does not impose a tight requirement because the controller is known perfectly. The action performed by the Smith predictor is fundamental for cancelling the steady-state error and preventing limit cycles, regardless of the controller used, and it also contributes to the system robustness. Because the error signal is sampled on an event basis and the process output is delayed, if the prediction unit was omitted from the event generator, the delay would not be compensated. As a result, the event-based sampling would increase the steady-state error previously originated by the delay.

The other unit of the event generator is the sampling unit. The aim of this block is to measure and to send an error sample to the controller each time the event condition is satisfied. For this purpose the SSOD sampling algorithm introduced in [15] has been considered. According to the SSOD algorithm, if $e(t)$ is the instantaneous error and $e^*(t)$ is the sampled error sent to controller, the relationship between them is defined as an integer multiple of a threshold Δ, namely, $e^*(t) = j\Delta$ with $\Delta \in \mathbb{R}^+$ and $j \in \mathbb{Z}$ representing the state of the SSOD block. This relationship can be observed in Fig. 3.

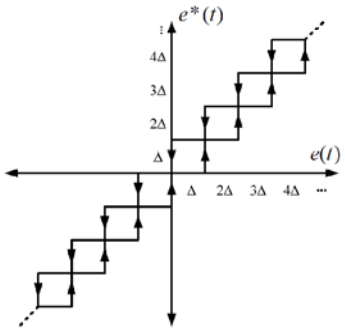

Figure 3: Quantization relationship between $e(t)$ and $e^*(t)$.

For the sake of clarity, some concepts are defined as follows:

- j : a subscript indicating the current state of the SSOD algorithm.
- e^*_j: the last sample sent to the controller at the instant t_j. Its value corresponds to $e^*_j = j\Delta$.
- t_j: the time instant in which the state j was reached and the sample e^*_j sent to the controller.
- T_j: the time interval between events or states. It is also defined as $T_j = T_{j \to j-1} = [t_j, t_{j-1})$, where j represents the last state reached and $j - 1$ the next (considering that the error is decreasing).
- s: the Laplace operator.

In addition, some premises are taken into account without loss of generality:

- The set-point $r(t)$ is considered as a positive piecewise constant signal directly proportional to Δ.
- The system output starts from a null state (with $j = 0$).
- It is assumed that $\overline{K} = K$.

Remark 2. In a typical control cycle, the state *j* will increase until its maximum value, when a set-point change occurs, and then the sequence of states *j* will decrease as the instantaneous error is reduced. Thus, the states will follow the sequence $(e^*_j, t_j), (e^*_{j-1}, t_{j-1}), \cdots, (e^*_{j-n}, t_{j-n}) \cdots, (e^*_1, t_1), (e^*_0, t_0)$. With respect to the assumption of $\overline{K} = K$, it should be noted that in processes without integrator poles the modeling errors in the steady-state gain K are infrequent.

2.3 The controller

Limit cycles are a common problem in event-based control systems. This phenomenon causes oscillations around the set-point, and it fundamentally originates because of two reasons:

- Accuracy is limited in quantized systems [18].
- The controller has integral action in its control law [5].

In accordance with the proposed design, the first drawback could be managed by the sampling algorithm. Unlike the standard SOD algorithm, the SSOD considers, by definition, the zero quantization level. This property can guarantee the existence of an equilibrium point for the system despite low quantization in the signals. On the other hand, perfect set-point tracking can be challenging when the controller presents integral action. When the system response is enclosed in a narrow band around the set-point and the controller has integral action, a small error or disturbance can easily move the system output away from the set-point (and the equilibrium point), reaching upper quantization levels. This situation can lead to unstable behaviors or limit cycles; therefore, avoiding integral action could help to prevent the limit cycles.

To overcome such drawbacks and to prevent limit cycles, a controller with adaptive topology was designed and analyzed in this work. The controller (called PI-P for short) works with periodic (time-based) sampling and is implemented as a combination of two independent well-known strategies: a PI and a P controller (Fig. 4). The PI component is designed for disturbance rejection and set-point tracking. The P component adds a second degree of freedom to improve the set-point tracking and to avoid limit cycles. The controller behavior is described as follows. When a set-point change occurs and the process output is far away from the set-point (and therefore the control error is high), the PI control performs the set-point tracking and disturbance rejection tasks. When the error is sufficiently reduced and disturbances are rejected, the controller topology is switched to the P controller such that it leads the response asymptotically to the set-point value. The controller and the logic conditions that switch topologies are defined as follows. If the transfer function of the PI controller is defined in its parallel form as

$$C_{PI}(s) = K_{P1}(1 + \frac{1}{sT_I}) \tag{3}$$

and the P controller as

$$C_P(s) = K_{P2} \tag{4}$$

according to Fig. 4, the block called Logic is designated for switching the controllers by means of the following gain scheduling strategy

$$(K_{P1}, K_{P2}) = \begin{cases} \left(\frac{\alpha T_I}{K}, 0\right), T_I = \tau & for \quad (|j| \geq 1) \wedge \left(|e^*{}_P| < |e^*{}_j|\right) \\ \left(0, \frac{1}{K}\right) & for \quad (|j| = 1) \wedge \left(|e^*{}_P| > |e^*{}_j|\right) \end{cases} \tag{5}$$

The variables e^*_P and e^*_j represent the previous and the last samples received, respectively, and the parameter T_I is set equal to τ. Thus, the criterion to switch controllers is to apply the P component when the sampled error satisfies the condition $|e^*(t)| = \Delta$ and its derivative is negative ($e^*_P(t) > e^*_j(t)$). Otherwise, the PI controller is applied. According to this behavior, the controller could be considered as a hybrid automaton with two dynamics that depend on the sequence of events e^*_P and e^*_j.

The gain scheduling proposition (5) is used to impose a specific dynamic to the response according to the previous reasoning. While the PI part is controlling the system, combining (3) and (5), the pole of the system is cancelled and the resultant open loop transfer function describes a piecewise linear trajectory. Thus, the parameter α, which represents one of the tuning parameters, is designed to regulate the

response convergence. After reaching the switching conditions, which can be arbitrary defined with the desired threshold Δ, the integrator is annulled and the gain of open loop response with the P controller could be adjusted, if it was necessary, to reach the set-point value. Such aspects are explained in detail later.

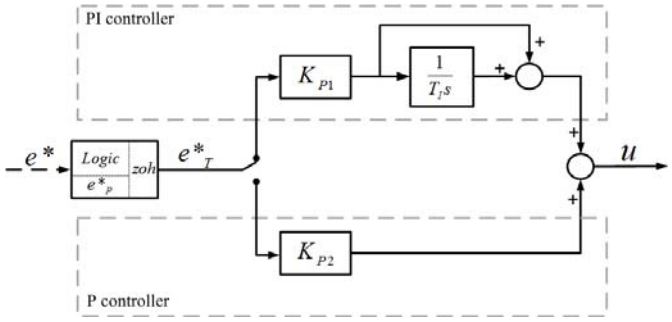

Figure 4: Block diagram of the PI-P controller. Grey dashed lines delimit each part of the controller. Black solid lines denote continuous signal transmission, and black dashed lines denote event-based signal transmission.

3. Loop response

The closed-loop response of the event-based system proposed in Fig. 1 depends mainly on three factors: the selected controller, the value of error samples, and the time instants in which events occur. The feedback loop is closed through the event generator when an event occurs. In this context, the event-based system can be considered as the open loop system represented in Fig. 5, which has two dynamics depending on (3) and(4), and whose initial conditions change each time an event is triggered.

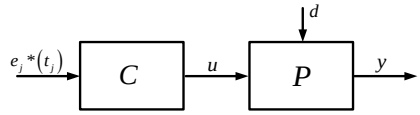

Figure 5: Block diagram of the open loop simplification of the proposed event-based system.

3.1 Mathematical model

According to the nomenclature in Fig. 2, the following signals will be used in the system response analysis: the system output $y(t)$, the instantaneous error $e(t)$, the event-based sampled error $e^*(t)$, and the emulated feedback signal called $y_r(t)$ (in advance, the feedback signal). The signal $y_r(t)$ represents the feedback signal that would be present in Fig. 1 if the scheme was time-based (i.e., if the sampling unit was omitted). This signal is considered basic for understanding the events-based behavior in the proposed system and, according to Fig. 2, it is calculated as the difference between the process output signal $y(t)$ and the prediction unit output signal $\hat{y}(t)$.

With the independence of the controller, given an error sample e^*_j, the generic expression in the frequency domain of the feedback signal is the following:

$$Y_r(s,j) = Y(s,j) - \hat{Y}(s,j) = P(s)C(s)E_j{}^*(s) + [\bar{P}_0(s) - \bar{P}(s)]\bar{C}(s)E_j{}^*(s) = \\ = [P(s) + \bar{P}_0(s) - \bar{P}(s)]C(s)E_j{}^*(s) \quad (6)$$

with s being the Laplace operator and j being the current state of the SSOD algorithm. For the sake of simplifying the subscripts in the explanations, only the case of a positive set-point change has been considered. In this way, if (6) is developed and translated into the time domain, two expressions are obtained for the signal $y(t)$. The equation, particularized when the PI component is enabled, has the following expression after simplifications:

$$y_r(t,j) = e^*{}_j \alpha t - e^*{}_j \alpha t(t - \bar{L}) + \left[e^*{}_j \alpha t + e^*{}_j \alpha(\bar{\tau} - \tau)\left(1 - e^{-\frac{t}{\bar{\tau}}}\right)\right](t - L) + y_r(t_j, j) \quad (7)$$

and when the P controller is enabled:

$$y_r(t,1) = e^*{}_1 \left(1 - e^{-\frac{t}{\bar{\tau}}}\right) - e^*{}_1 \left(1 - e^{-\frac{t}{\bar{\tau}}}\right)(t - \bar{L}) + e^*{}_1 \left(1 - e^{-\frac{t}{\bar{\tau}}}\right)(t - L) + y_r(t_1, 1) \quad (8)$$

The value $y_r(t_j, j)$ in equations (7-8) represents the initial conditions. This value changes each time a new event occurs, and it has the following form:

$$y_r(t_j, j) = r - j\Delta \quad (9)$$

where t_j denotes the time instant in which the state j is triggered. As the signal $y_r(t)$ increases, the error signal $e(t)$ is reduced and new events are generated according to the SSOD sampling scheme. From (8) or (9), the instantaneous error in the time domain results in:

$$e(t,j) = r - y_r(t,j) \quad (10)$$

From (7-8), a particular case can be analyzed where modeling errors are not assumed, and the equality $P(s) = \bar{P}(s)$ is satisfied. This case is illustrative for explanation because the system exhibits the desired expected performance. Under this assumption, and by developing (6), the reduced expression that characterizes the feedback signal is obtained as follows:

$$Y_r(s,j) = E_j{}^*(s) C(s) \bar{P}_0(s) \quad (11)$$

If (11) is translated into the time domain and is particularized for the PI component, the result is:

$$y_r(t,j) = e^*{}_j \alpha t + y_r(t_j, j) = j\Delta\alpha t + y_r(t_j, j) \quad (12)$$

which describes a piecewise linear trajectory with a gradient $j\Delta\alpha$. Particularizing (11) for the P component, the result is:

$$y_r(t,1) = e^*{}_1 \left(1 - e^{-\frac{t}{\bar{\tau}}}\right) + y_r(t_1, 1) = \Delta\left(1 - e^{-\frac{t}{\bar{\tau}}}\right) + y_r(t_1, 1) \quad (13)$$

which describes the response of a first-order system. From (12) and (13), adding the system delay L, the system output response can be obtained for each state as (14) and (15). Note that as a consequence of the Smith predictor effect, the $y(t)$ open-loop response will be a L delayed prediction of $y_r(t)$.

$$y(t,j) = y_r(t,j)(t - L) = j\Delta\alpha t(t - L) + y(t_j, j) \quad (14)$$

$$y(t,1) = y_r(t,1)(t - L) = \Delta\left(1 - e^{-\frac{t}{\bar{\tau}}}\right)(t - L) + y(t_1, 1) \quad (15)$$

Complementary control laws (7-8) and (12-13) define the global performance of the event-based system. Without modeling errors, using the gain scheduling strategy in (5), the system response will be piecewise linear when the error is high and it will converge asymptotically to the reference when the error is reduced and disturbances are rejected. A graphical example is shown in Fig. 6.

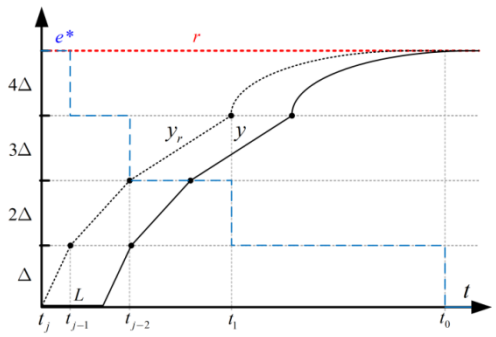

Figure 6: Characteristic response of signal $y(t)$ and $y_r(t)$ in absence of modeling errors.

Assuming that the system starts from null initial conditions, the figure describes a generic example where, a set-point change equal to 4Δ is applied at the initial instant ($t_j = 0$). As is shown, the response is linear while the PI controller is acting. The gradient of the response changes after each event because of the event samples e^* are reduced. When the switching condition is reached (at $y_r = 3\Delta$), the system responds as a first order model as a result of the P controller action. Note that while ideal conditions are assumed the behavior of the system output is similar to the feedback signal, that is, the feedback signal represents a prediction of L seconds of the output signal.

3.2 The effect of modeling errors

In contrast to the ideal case, when the model $\bar{P}(s)$ is approximated, the expression of the feedback signal is more complex. If we focus on the expression in (7), unlike its ideal form in (12), there are coupled dynamics. Specifically, the part that depends on L and $\bar{L}$ delays. As a consequence of the delay dependence, the ideal part of (7) changes each time an event is triggered. The coupled terms continue to progress during a period equal to $\max\{L, \bar{L}\}$, that is, until the instant $t_j + \max\{L, \bar{L}\}$. During this interval, the feedback signal depends on both the current triggered sample e^*_j and the previous one, e^*_{j+1}. In practice, the coupled dynamics can be considered as a shifting of the initial condition value that appears after the event in the form

$$\begin{aligned} y_r(t_j + max\{L, \bar{L}\}, j) \\ &= r - j\Delta + e^*_j\alpha(t_j + max\{L, \bar{L}\}) - e^*_j\alpha t(t_j + max\{L, \bar{L}\})(t - \bar{L}) \\ &+ \left[e^*_j\alpha(t_j + max\{L, \bar{L}\}) + e^*_j\alpha(\bar{\tau} - \tau)\left(1 - e^{-\frac{t_j + max\{L,\bar{L}\}}{\tau}}\right)\right](t - L) = \\ &= r - j\Delta + \lambda_j + \gamma_j = y_r(t_j, j) + \lambda_j + \gamma_j \end{aligned} \tag{16}$$

where $j < j_{max}$ and $\lambda_j, \gamma_j \in \mathbb{R}$ represent, respectively, the ideal and the extra dynamics in expression (7) evaluated at time instant $t_j + \max\{L, \bar{L}\}$. In accordance with (16), and depending on the modeling error and its magnitude, the coupled dynamics could exert a positive or negative effect over the feedback signal such that an upper or lower quantization level could be reached. The worst case is found when the coupled dynamics remain until the value $\pm\Delta$ is reached. In this case, the feedback signal is displaced by at least Δ with respect to the ideal response. Note that this effect arises each time an event is triggered, and as a consequence, when the P action is enabled, its response will also undergo a shifting. Obviously, an additional displacement of $\pm\Delta$ in the band around the set-point would be undesirable because new events could be triggered and limit cycles may even appear.

4. Practical stability for a FOPTD process

Typically, the stability question for an event-based control system is related to limit cycles. In accordance with this, it has been considered in this work that system stability is characterized by the absence of limit cycles and the presence of an equilibrium point around the set-point. In other words,

the instantaneous error is enclosed in an interval around the set-point value $(-\Delta, \Delta)$. In this context, the stability of the proposed scheme can be guaranteed under disturbance conditions and structured uncertainties (unstructured also, as will be observed later) by the fulfillment of two conditions: the demonstration of the existence of the equilibrium point and determination of its reach.

4.1 Equilibrium point existence

Proposition 1: Let a control system according to Fig. 1 with a controlled process as in (1). Considering step disturbances on system input and output signals and bounded structured modeling errors, it can be stated that an equilibrium point around a set-point r can be found with the PI-P controller.

Proof: The error dynamics of the event-based control system in Fig. 1, in the presence of disturbances, can be expressed as follows:

$$e(t,j) = r - y_r(t,j) + d(t) \tag{17}$$

where $\mathrm{e^*_j}$ would correspond to the last sample received in the controller block after a step change is applied to the reference signal. $\mathrm{d(t)}$ gathers possible step disturbances on the input and output signals so that $\mathrm{d(t)} \triangleq \mathrm{d_u(t) + d_y(t)}$. Considering (17) at the time instant in which the system has reached the switching condition given by (5), that is, $\mathrm{e^*(t) = e^*_k = \pm\Delta}$, the system output will remain in the equilibrium point without limit cycles if the following condition is satisfied:

$$\left|\lim_{t\to\infty} e(t)\right| < \Delta \tag{18}$$

which is equivalent to the expression in (19).

$$|r| - \Delta < \left|\lim_{t\to\infty} y_r(t_{\pm 1} + t, \pm 1)\right| < |r| + \Delta \tag{19}$$

This means that if the final value reached by the signal $\mathrm{y_r(t)}$ when P control action is performed remains enclosed in a range $(-\Delta, \Delta)$ around the set-point, new events will not occur, and the system will have reached an equilibrium point. Considering (8), the final value achieved by $\mathrm{y_r(t)}$ will depend on the displacement caused by the coupled dynamics in the interval γ_1. Note that the final value that will be obtained by expression (8) with the P control action will always be the same: the $\pm\Delta$ value. For simplification, explanations have been focused on the positive case whose switching condition is reached for $\mathrm{j = 1}$ (that is, $\mathrm{e^* = \Delta}$). If the expression for $\mathrm{y_r(t)}$ in the inequality (19) is developed as follows:

$$\begin{aligned}\left|\lim_{t\to\infty} y_r(t,1)\right| = \Big| & y_r(t_1,1) + \gamma_1 \\ & + \lim_{t\to\infty}\left[e^*{}_1\left(1 - e^{-\frac{t}{\bar{\tau}}}\right) - e^*{}_1\left(1 - e^{-\frac{t}{\bar{\tau}}}\right)(t - \bar{L}) + e^*{}_1\left(1 - e^{-\frac{t}{\bar{\tau}}}\right)(t - L)\right]\Big| = \\ = |y_r(t_1,1) + \gamma_1 + \Delta| &= |r - \Delta + \gamma_1 + \Delta| = |r| + |\gamma_1|\end{aligned} \tag{20}$$

the final condition for evaluating the existence of the equilibrium point is obtained (21).

$$|\gamma_1| < \Delta \tag{21}$$

In (20), $\mathrm{t_1}$ represents the instant in which controllers are switched in the control cycle for a given set-point r. After the instant $\mathrm{t_1 + \max\{L, \bar{L}\}}$, the P control action is performed in the coupled part of expression (8). According to (15), under ideal conditions, controllers are switched at the instant $\mathrm{t_1}$ and the P control action is performed at the same time. In contrast, when uncertainty in model (2) exists, according to (20), the $\mathrm{\lim_{t\to\infty} y_r(t_1, 1)}$ is displaced by quantity of γ_1.

In this context, the only way to guarantee the reach of the equilibrium point would be to compute the value γ_1 using recursive procedures. Because this computation can be computationally costly, a more conservative condition to guarantee the stability can be used. Given that error samples are reduced as feedback signal is increased, γ_j values are consequently reduced and it can be stated that $\gamma_{jmax-1} > \gamma_{jmax-2} > \cdots > \gamma_2 > \gamma_1$. Therefore, rather than computing γ_1 using recursive methods, it is more useful to evaluate whether any of values γ_j satisfy condition (21). In this context, the simplest condition (and the most conservative) can be evaluated without the need of recursive algorithms, as given by (22).

$$|\gamma_{j_{max-1}}| = \left| e^*_{j_{max}} \alpha \left[(L - \bar{L}) - (\tau - \bar{\tau}) \left(1 - e^{-\frac{L}{\bar{\tau}}} \right) \right] \right| < \Delta \tag{22}$$

If condition (22) is satisfied, it can be stated that the system will have an equilibrium point in the presence of structured modeling errors and step disturbances. This condition defines a sufficient condition (but not necessary) to evaluate the system stability. This solution provides a range of values for the tuning parameters that ensure the stability but the solution is biased and could be more precise if the γ_1 value was explicitly calculated by using recursive methods or by simulations. The reason for not calculating the expression of γ_1 is because of the analysis depends on the considered number of states and the development would suppose an important part of the paper without contributing substantially to the stability analysis of this kind of event-based controller. Additionally, the solution would be local to the considered states. In this context authors emphasize the importance practical of the proposed conservative condition which is easily evaluable.

4.2 Reach of the equilibrium point

In this subsection, the reach of the equilibrium point in the presence of bounded and structured uncertainty is demonstrated. Given the type of disturbances considered in this work (step disturbances), conditions (19-21) are conserved despite the presence of them because, at the instant of switching the system controlled with the PI-P design, the following condition will be satisfied:

$$|e(t_1) + d(t_1)| = \Delta \tag{23}$$

Therefore, if the equilibrium point exists, the system with the PI-P controller will be stable because the switching condition (23) will always be achieved, which is demonstrated as follows. Expanding equation (17) with (8) and expressing possible step disturbances as $\mathrm{d(t)} \triangleq \mathrm{d_y} + \mathrm{d_u K} \cdot \left(1 - \mathrm{e}^{-\mathrm{t}/\tau}\right)$:

$$\begin{aligned} e(t,j) &= r - y_r(t,j) + d(t) = \\ &= r - \left[e^*_j \alpha t - e^*_j \alpha t (t - \bar{L}) + \left[e^*_j \alpha t + e^*_j \alpha t (\bar{\tau} - \tau) \left(1 - e^{-\frac{L}{\bar{\tau}}} \right) \right] (t - L) \right] + d(t) \end{aligned} \tag{24}$$

Differentiating (24) results in:

$$\begin{aligned} \dot{e}(t,j) &= -\dot{y_r}(t,j) = \\ &= -e^*_j \alpha + e^*_j \alpha (t - \bar{L}) - e^*_j \alpha (t - L) + \left[e^*_j \alpha \left(1 - \frac{\tau}{\bar{\tau}} \right) + \frac{d_u K}{\tau} \right] e^{-\frac{1+\tau}{\tau} t} (t - L) \end{aligned} \tag{25}$$

where the term multiplied by $\mathrm{e}^{-\frac{1+\tau}{\tau}\mathrm{t}}$ tends toward zero. In this way, using the final value theorem produces:

$$\dot{e}(t,j) = -e^*_j \alpha \tag{26}$$

demonstrating that the system with the PI-P controller tends to the switching condition despite uncertainty and disturbances. In addition, note that the convergence speed depends strongly on the

parameter α. Equations (23-26) confirm that by the time the commutation state is reached, disturbances are totally cancelled or nearly cancelled.

5 Tuning methodology

The proposed tuning methodology is explicitly designed for the proposed event-based control scheme. The three basic parameters that define the control law in the PI-P design $(\mathrm{K_{P1}}, \mathrm{T_I}, \mathrm{K_{P2}})$ are replaced by two parameters (Δ, α) for the event generator and the controller, respectively. The original parameters are specified in accordance with the process model parameters as was indicated in (5). Through this change, the new tuning parameters have a more interactive effect on the system response. They provide two degrees of freedom and their influence is very intuitive for the designer. The parameter Δ controls the distance (in magnitude) between two consecutive events and therefore the resulting number of events. The parameter α regulates how fast the response converges to the set-point value, and furthermore, it is directly related to both the control effort and the capability of reducing the control error. Such features can be estimated in the tuning procedure by means of two performance indexes: the IAU (Integrated Absolute Control signal divided by the settling time (28)) and the IAE (Integrated Absolute Control Error (27)). From (10, 12, and13), the IAE index can be pre-computed for a given reference r as follows:

$$\begin{aligned} IAE(r,\Delta,\alpha) &= \int_0^{T_{ST}} |e(t)|\,dt = \int_0^{L} |e(t)|dt + \sum_{j=j_{max}}^{1} \int_0^{T_j} |e(t)|dt = \\ &= \int_0^{L} |r|dt + \sum_{j=j_{max}}^{1} \int_0^{T_j} \left| r - \left(y_r(t,j) + y_r(t_j,j) \right) \right| dt = \\ &= |Lr| + \sum_{j=j_{max}}^{2} \int_0^{T_j} \left| r - \left(j\Delta\alpha t + y_r(t_j,j) \right) \right| dt + \int_0^{T_1} \left| r - \left(\Delta\left(1 - e^{-\frac{t}{\tau}}\right) + y_r(t_1,1) \right) \right| dt \end{aligned} \tag{27}$$

where $\mathrm{y_r(t_j, j)}$ was defined in (9). Similarly, from (3-5) and (28), the IAU index can be calculated as follows:

$$\begin{aligned} IAU(r,\Delta,\alpha) &= \frac{\int_0^{T_{ST}} |u(t)|\,dt}{T_{ST}(r,\Delta,\alpha)} = \frac{\sum_{j=j_{max}}^{1} \int_0^{T_j} |u(t)|dt}{T_{ST}(r,\Delta,\alpha)} = \\ &= \frac{\sum_{j=j_{max}}^{2} \int_0^{T_j} \left| e^*{}_j K_{P1} \left(1 + \frac{T_{j+1}}{T_I}\right) + u_0(t_j,j) \right| dt + \int_0^{T_1} \left| e^*{}_1 K_{P2} + u_0(t_j,j) \right| dt}{T_{ST}(r,\Delta,\alpha)} = \\ & with\ u_0(t_j,j) = \begin{cases} \sum_{j_{max}-1}^{j} e^*{}_{j+1} K_{P1} \left(1 + \frac{T_{j+1}}{T_I}\right) - \frac{\alpha\Delta\tau}{K} & for \quad j \in [j_{max}-1, 1] \\ 0 & for \quad j = j_{max} \end{cases} \end{aligned} \tag{28}$$

where values $\mathrm{T_j}$ are:

$$T_j = T_{j\to j-1} = \begin{cases} \frac{\Delta + \mathrm{a}}{\alpha\Delta \mathrm{j}} & for \quad j = j_{max} \\ \frac{1}{\alpha \mathrm{j}} & for \quad j \in [j_{max}-1, 2] \\ T_S & for \quad j = 1 \end{cases} \tag{29}$$

With respect to the sampling effort, an additional advantage of the proposed approach is that the response has no overshoot. This aspect allows reducing the minimum number of events required to

reach the set-point (considering a SSOD sampling scheme). In this context, considering the control scheme according to Fig. 1, the expected number of events can be obtained by:

$$S = 2(r \div \Delta) \tag{30}$$

where the operation $\mathrm{r} \div \Delta$ represents the integer division. Note that (30) only depends on the tuning parameter Δ. The expected settling time can be obtained as follows:

$$T_{ST}(r, \Delta, \alpha) = L + \frac{a}{|j_{max}|\Delta\alpha} + \sum_{j=|j_{max}|}^{2} \frac{\Delta}{j\Delta\alpha} + T_S \tag{31}$$

where $\mathrm{j_{max}}$ and a represent the quotient and the remainder of the integer division $\mathrm{r} \div \Delta$, respectively. The parameter $\mathrm{T_s}$ is defined as $5\mathrm{T}$ (i.e., the time interval in which the time response of a first-order system is enclosed in a 0.7% error band around the reference). It is worth stressing that the prediction of aspects such as the settling time, the number of events, or the other presented performance indexes is a difficult task in most event-based control systems because of the asynchronous behavior of events.

As previously mentioned, the other aspect that can be considered in the tuning design is a type of actuator constraint. This work has been focused on the slew rate constraint of the actuator action. This limitation establishes an upper bound on the response slope in the form of $(\mathrm{j}\Delta\alpha)_{\mathrm{max}}$. According to the approach of this work, for a given set-point r, maximum values of j, Δ, and α can be computed such that the upper bound is satisfied. In addition, the definition of the actuator restriction during the tuning design is useful to avoid the bump effect of the control signal. Each time an event occurs, given that the error signal is quantized with a step Delta, the P part of the PI controller generates the bump effect. For a given restriction, in case of defining the tuning parameters to satisfy the slew rate restriction, the bumped effect is much reduced and consequently the performance is not significantly worsened. In practical, this effect can be annulled if the tuning is sufficiently conservative but on the contrary, this dynamic could become critical if the restriction of the actuator is severe and the design ignores such design recommendation.

In practice, the aspects mentioned until now let us define a framework for performing the tuning. In this sense, the tuning can be performed by tuning regions. A tuning region has been defined in this work as a parameter space that relates a performance index to the tuning parameters (Δ, α). They have been conceived to show information about the observed performance index, the expected number of events, and the actuator constraint. Note that the tuning regions could be easily pre-computed from expressions such as (27-31). In this sense, Fig. 7 shows illustrative examples of three tuning regions based on the presented performance indexes: T_{ST}, IAU, and IAE. In this case, the system parameters have been set to $\mathrm{K} = 1, \tau = 1, \mathrm{L} = 0.6, \mathrm{r} = 1, (\mathrm{j}\Delta\alpha)_{\mathrm{max}} = 0.5, \Delta \in [0.1, 0.5]$, and $\alpha \in [0.1, 1]$. The information that shows each tuning region is explained as follows. White dashed vertical lines define points where the remainder of the integer division (according to expression (30)) is zero, and therefore, they delimit areas with the same expected number of events. This means that tunings with parameters enclosed between the same two white dashed vertical lines will perform the same sampling effort. On the other hand, red dashed lines are associated with the actuator constraint. They horizontally delimit two areas where the actuator restriction is satisfied or not. In the case of the tuning region for the settling time, considering one area with the same event rate, higher values of α correspond with lower values of $\mathrm{T_{ST}}$. In this case, the constraint defines a lower bound in the value of the performance index. In a similar way, Fig. 7-b shows the tuning region for the IAU performance index defined according to (31). Now, the area between white dashed lines represents the evolution of the control effort while the event rate is conserved. The red dashed lines delimit an upper bound. As shown in Fig. 7-b, higher values of α produce higher control effort values. The tuning region of the IAE is shown in Fig. 7-c. Here, higher

values of α generate lower IAE values for the same number of events. The red dashed lines denote a lower bound on the IAE value.

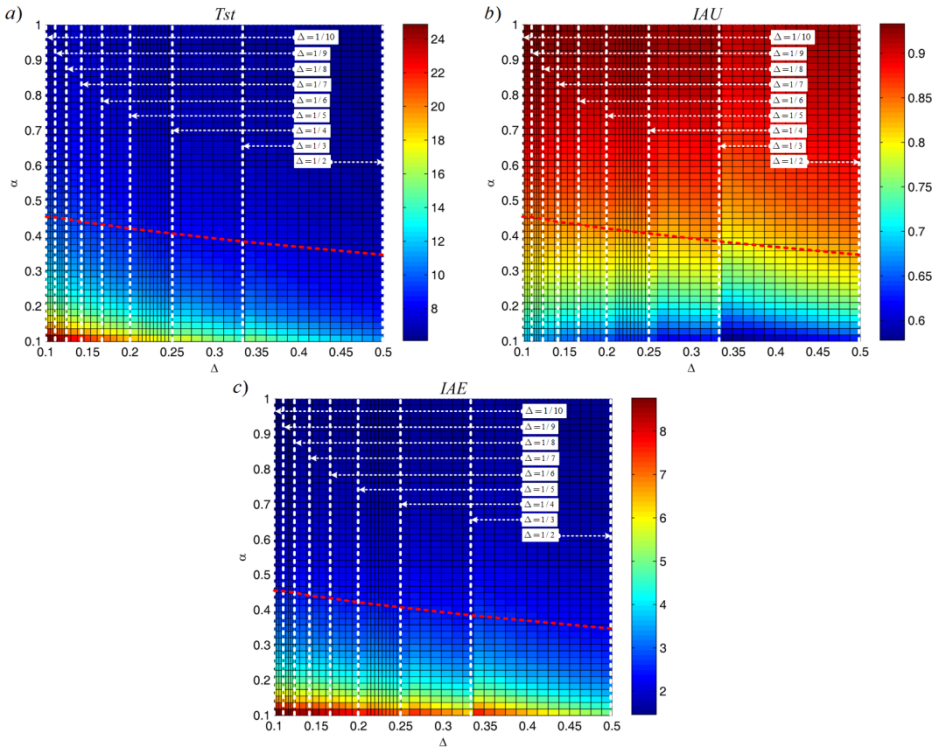

Figure 7: (a) Tuning region for the settling time performance index (31). The red dashed lines indicate a lower bound in the actuator restriction. The white dashed lines delimit areas with the same sampling effort. (b) Tuning region for the IAU performance index (28). The red dashed lines indicate an upper bound in the actuator restriction. The white dashed lines delimit areas with the same sampling effort. (c) Tuning region for the IAE performance index (27). The red dashed lines indicate a lower bound in the actuator restriction. The white dashed lines delimit areas with the same sampling effort.

With the aim of illustrating the methodology, several examples of tuning regions have been analyzed in this section. In this context, given a generic FOPTD system and a set-point r, tuning regions can be pre-computed and used to define tuning parameters. Any other index could also have been considered.

Remark 3: Tuning regions are calculated considering an accurate FOPTD model. In the non-ideal case, tuning regions will not correspond exactly with system responses, but such an assumption will not impose a great restriction on the predicted system performance while stability condition (22) is satisfied.

6. Uncertainty compensation algorithm

The main drawbacks of control schemes based on the Smith predictor are model mismatches. In the proposed event-based system, unless the control law according to (5) is adapted in the presence of model mismatches, the system response will present a steady-state offset. For this reason, one of the goals of the PI-P design is to provide an extra degree of freedom (by the means of the P part gain) to avoid this effect. Note that the retuning of the P part represents the simplest solution to fulfill such an objective.

From a practical perspective, the consequences of modeling errors are variations in triggering instants of events, that is, fluctuations in time intervals between samples with respect to the ideals (from a completely known model). As was mentioned in Section 3, a coupled dynamics appears in the feedback signal and, once the P controller is applied, the response is displaced by a quantity of γ_1. As a consequence of this displacement, the final value reached by the system output (fixed from (5) to $\Delta \cdot \mathrm{K_{P2}}$) will have steady-state error. In this situation, the system output $\mathrm{y(t)}$ would not reproduce the behavior of the feedback signal (with the consequent delay L). However, because the processes studied in this work consider non-integrator poles, the output signal and the feedback signal will have the same steady-state value while the P control action is enabled. This situation can be utilized in the algorithm

design. Because the feedback signal $y_r(t)$ is composed of a free-delay model $\overline{P}_0(s)$, which is perfectly known, the value of its output can be easily estimated to readjust the gain K_{P2}. In this sense, the proposed algorithm consists of estimating new intervals between events (as a result of modeling errors) and using them to recursively calculate how much should change the gain K_{P2} (to achieve the set-point value). Given a set-point r, the output value of $\overline{P}_0(s)$ at the switching instant t_1 can be estimated as follows:

$$\bar{y}_0(t_1) = \sum_{j=j_{max}}^{2} j\Delta\alpha T_j \tag{32}$$

where the values T_j were calculated in (29). From (32), the gain required K_{P2} to avoid the steady-state error is obtained as follows:

$$K_{P2} = \frac{r - \bar{y}_0(t_1)}{\Delta K} \tag{33}$$

In this way, by only changing the gain of the P component in the PI-P controller, the system output would reach the set-point value asymptotically. In addition, this will not affect the disturbance rejection performance of the PI component. Obviously, with the focus on the SSOD sampling scheme, once the set-point value has been reached by the response and it is enclosed in the deadband, if new disturbances would appear and they do not contribute with the necessary energy to trigger new events, the system would have state steady error but it is worth stressing that it is a common problem on event based system which considers a deadband around the set-point value independently of the controller employed.

Note that the algorithm could be executed on-line because values T_j can be measured during experiments or estimated from (32-33). As is demonstrated in Section 7, the proposed algorithm will work effectively both in the presence of structured and unstructured uncertainties.

7. Simulation results

In this section, a set of experiments are simulated to illustrate the theoretical analysis. In particular, the guidelines to tune the controller are given, the main control loop properties are evaluated, and the approach is compared with other event-based and time-based controllers.

7.1 Guidelines for the tuning of parameters

In Section 5, the tuning methodology was explained using the tuning regions. Here, the approach is qualitatively analyzed and some guidelines are presented. The explanations have been focused on the process (34) whose tuning regions were calculated in Section 5.

$$P(s) = \frac{1}{s+1} e^{-0.6s} \tag{34}$$

Figs. 8 and 9 present the time domain system response based on some tuning examples. In Fig. 8, the simulation parameters are specified as $r = 1$, $\alpha = 0.5$, and Δ changes according to the set $\{0.1, 0.25, 0.5\}$. In qualitative terms, when α is fixed and Δ changes, the convergence speed of the response is practically preserved, whereas the event rate can be reduced according to the result of the equation (30).

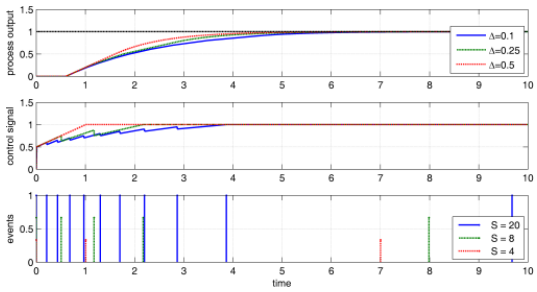

Figure 8: Top plot: Output signals of the process (34). Middle plot: Control signals. Bottom plot: Generated events.

The variations in the parameter α are analyzed in Fig. 9. Here, the parameter Δ is set equal to 0.25 and α is varied according to the set $\{0.25,0.5,1\}$. The convergence speed is accelerated when α is increased, while the number of events is kept equal. Note that the features of the time domain responses correspond with the predicted values in the tuning regions.

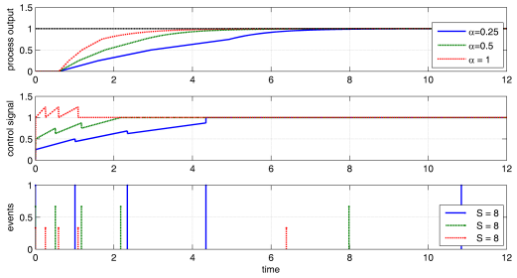

Figure 9: First plot: Output signals of the process in (34). Second plot: Control signals. Third plot: Generated events.

As shown in Figs. 8 and 9, the system always reaches the desired set-point value without steady-state error because of the delay compensation. If model mismatches were considered, the tuning region results will not exactly correspond with the system responses, but their validity could still be considered as satisfactory if some guidelines were followed. First and foremost, the tuning parameters should satisfy both the actuator constraints and the stability condition (22). The fulfillment of (22) would preserve the characteristic of (30). Consequently, deviations in the rest of the performance indexes would be bounded. The influence of the bound (22) is now analyzed in further detail. Maximum values of α that satisfy the stability condition have been calculated in Fig. 10 for several cases of structured modeling errors. The figure shows that for higher values of Δ, the restriction for stability is relaxed. In a broad sense, this does not necessarily entail a performance improvement because the steady-state error is also increased. In addition, small disturbances could be easily masked if the limits of the equilibrium point were considerably shifted from the set-point value.

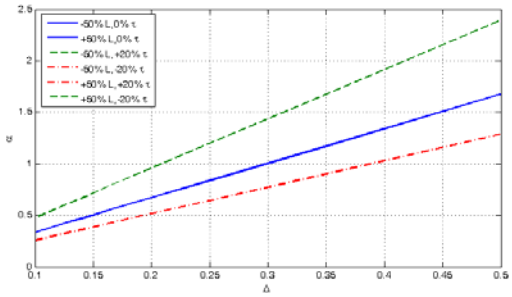

Figure 10: Evolution of the condition (22) for several cases of structured modeling errors.

Based on the above considerations, some guidelines on how to tune design parameters are suggested:

- For relatively accurate process models where actuator restrictions do not exist, the designer can compute the desired tuning region and choose any feasible tuning parameters.
- When actuator constraints are considered, the tuning parameters should satisfy the upper bound that defines $(j\Delta\alpha)_{\max}$. In addition, anti-windup schemes could be used for dealing with saturations, (see [20]).

- In the presence of noise in measurements, the event threshold Δ should be defined as greater than the noise band to avoid undesired trains of events. In addition to this consideration, a very low value of Δ could easily trigger new events as a consequence of low gain disturbances. It is also important to be aware that a large value of Δ could mask significant disturbances. In this context, a trade-off between the detection of disturbances and event triggering should be found. A recommended value could be between two times the noise band and an upper bound defined as $|r/2|$.
- If the modeling errors are significant, then the tuning regions will represent approximated values of predicted performance indexes. In this situation, it is important to satisfy the stability condition (22). Specially, the following two-step procedure is recommended: (I) the value of α should be reduced until it is admissible; and (II) the value of Δ should be increased until (22) is satisfied. If uncertainties were unknown but bounded, the worst case should be considered in the tuning.
- On architectures with energy constraints, such as wireless sensors or battery-powered systems, it is recommendable to reduce the sampling effort to the minimum required (according to (30)).

7.2 Uncertainty compensation

In the presence of structured uncertainty, the system stability can be guaranteed. However, uncertainties lead the response to the set-point with a steady-state error that should be cancelled. With this aim, one mechanism was proposed in Section 6. Considering the process in (34), two experiments with structured uncertainty were simulated. The results for uncompensated and compensated responses according to the methodology proposed (readjustment of K_{P2}) are compared with the ideal case in Fig. 11. In left-side plots, a modeling error of +20% in $\bar{L}$ and +50% in $\bar{\tau}$ has been considered, whereas in right-side plots, an error of -20% in $\bar{L}$ and -50% in $\bar{\tau}$ were considered. In both cases, tuning parameters α and Δ were set equal to 0.4 and 0.25, respectively. As is shown in Fig. 11, using the proposed algorithm, the controller achieves a steady-state error very close to zero.

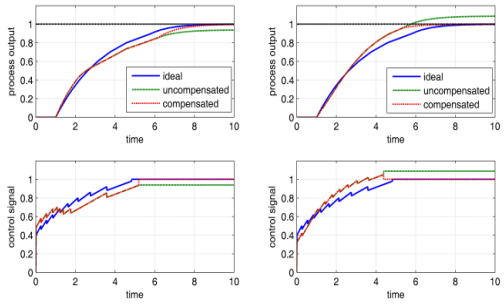

Figure 11: Top plots: Output signals of the process (34) in the presence of structured uncertainties. Bottom plots: Control signals in the presence of structured uncertainties.

The effectiveness of the approach is shown for unstructured uncertainty cases. The fourth-order process in (35) was considered.

$$P(s) = \frac{1}{(s+1)^4} \tag{35}$$

Thus, an FOPTD approximation is considered in the Smith predictor model.

$$\bar{P}(s) = \frac{1}{2.1168s + 1} e^{-1.8784s} \tag{36}$$

The results are compared in Fig. 12. In the left-side plots, the tuning parameters were specified as $\alpha = 0.5$ and $\Delta = 0.25$, whereas in right-side plots, they were $\alpha = 0.4$ and $\Delta = 0.1$. In both cases, the error is practically eliminated with the K_{P2} gain readjustment.

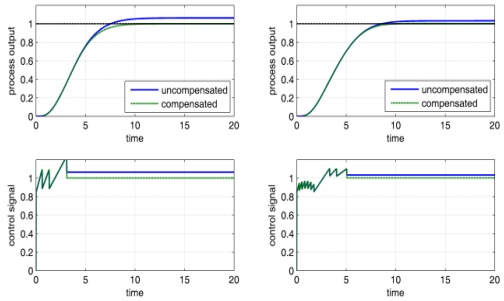

Figure 12: Top plots: Output signals of the process (34) in the presence of unstructured uncertainties. Bottom plots: Control signals in the presence of unstructured uncertainties.

7.3 Robustness evaluation

For absence of disturbance or uncertainty cases, the state-steady error can be easily cancelled with the appropriate gain scheduling strategy. If model mismatch exists and a disturbance appears, this should be estimated and taken into account to readjust $\mathrm{K_{P2}}$. Because this is not always possible in practice, it is worth verifying the system robustness if the readjustment of $\mathrm{K_{P2}}$ was not considered. With this aim, the example process in (37) has been considered as a structured uncertainty case.

$$P(s) = \frac{1}{1.5s + 1} e^{-2s} \tag{37}$$

For the controller design, it has been assumed variations in the range of $\pm 20\%$ in the set of parameters $(\bar{\mathrm{L}}, \bar{\tau})$. For the unstructured uncertainty case, the process in (35) and its FOPTD approximation were considered. In experiments, a unit load step disturbance is applied at $\mathrm{t} = 15$. Table 1 summarizes the simulation results.

Table 1

Simulation results in the assessment of system robustness for structured and unstructured uncertainty cases. IAE $(\mathrm{T_{ST}})$: integrated absolute error evaluated until settling time. IAU $(\mathrm{T_{ST}})$: integrated absolute control signal (according to (28)) evaluated until settling time. $\mathrm{E_{SS}}(\mathrm{T_{ST}})$: steady-state error as a percentage of event threshold and evaluated at the settling time instant. $\mathrm{E_{max}}$: maximum value reached by the error signal (as a consequence of the disturbance). S: the number of events. $\mathrm{T_{ST}}$: the settling time (previous to the appearance of the disturbance).

$\bar{\mathrm{K}}$	$\bar{\tau}$	$\bar{\mathrm{L}}$	α	Δ	IAE $(\mathrm{T_{ST}})$	IAU$(\mathrm{T_{ST}})$	$\mathrm{E_{SS}}(\mathrm{T_{ST}})$	$\mathrm{E_{max}}$	S	$\mathrm{T_{ST}}$
Structured uncertainties $\mathrm{P(s)} = (37)$										
1	1.5	2	0.25	0.1	3.37	0.87	3%	0.78	29	14.68
--	1.5	1.6	--	--	3.54	0.90	23.9%	0.80	30	14.63
--	1.5	2.4	--	--	3.33	0.85	24.5%	0.76	29	14.74
--	1.2	2	--	--	3.50	0.88	24.7%	0.80	30	14.73
--	1.8	2	--	--	3.38	0.87	28.8%	0.75	29	14.62
--	1.2	1.6	--	--	3.76	0.91	58.1%	0.83	32	14.74
--	1.8	1.6	--	--	3.43	0.89	9.7%	0.77	29	14.52
--	1.2	2.4	--	--	3.35	0.85	7.3%	0.78	29	14.75
--	1.8	2.4	--	--	3.32	0.84	46%	0.74	29	14.70
Unstructured uncertainties $\mathrm{P(s)} = (35)$										
1	2.1168	1.8784	0.25	0.1	3.3943	0.92	1.6%	0.66	28	14.72
--	--	--	0.4	0.1	2.3672	1.01	30%	0.63	26	12.86
--	--	--	0.25	0.125	3.3138	0.93	4.4%	0.66	22	14.66
--	--	--	0.4	0.125	2.3672	1.02	24.6%	0.63	22	13.02
--	--	--	0.25	0.2	3.1750	0.97	22.8%	0.67	13	14.29
--	--	--	0.4	0.2	2.4238	1.05	48.6%	0.64	13	13.35

For structured cases, even for variations of $\pm 20\%$ in the parameters, the system performance does not significantly worsen. Although an unavoidable offset appears in the response, the deviations in performance indexes are kept small in comparison with ideals because the stability condition is satisfied. Some performance indexes can even improve. Regarding the settling time, because it depends fundamentally on the instant in which the P control action is enabled, its lowest value is not necessarily achieved for the ideal case. In this sense, when the Smith predictor model dynamics is faster than the

process, the controller action will drive the process towards the reference value similarly to a positive disturbance effect. On the contrary, if Smith predictor model dynamics is slower, the controller will produce a sluggish action, that is, the same effect as a negative disturbance. Regarding number of events, experiments have a similar result. For stable systems with monotonic increasing responses, the number of events obtained depends on both the event thresholds and the maximum change that disturbances cause in the response before being rejected. Because of this, small variations can appear in experiments. Note that if experiments were focused on the time interval until the load disturbance appears, the characteristic (30) would be preserved for all experiments.

For the unstructured case, the model employed in the prediction unit represents a FOPTD approximation of the process. Similar to the structured case, the system performance does not worsen substantially for changes in tuning parameters or model approximations. Despite inherent modeling errors, the experiments reach the equilibrium point and the set-point is tracked with a reasonably small steady-state error.

It is worth stressing that despite uncertainties, disturbances, or the sampling technique, the controller represents a small variation of a well-evaluated solution for processes control, such as a PI controller. In practice, this feature becomes the approach suited for real applications and translatable to higher order systems (as it has been shown).

7.4 Comparison with other event-based controllers

The performance of the presented strategy is compared with other controllers based on the SOD sampling scheme. With this purpose, the response of a SOOD-PI controller [19] is simulated. A SOD-PI controller [16], the controller presented in [11], a PIDplus [2], and a time-driven controller (denoted as DT-PI) [2] are also simulated. The first simulation reproduces the experiment shown in [19] for the system described in [30]:

$$P(s) = \frac{(-0.3s + 1)(0.08s + 1)}{(2s + 1)(s + 1)(0.4s + 1)(0.2s + 1)(0.05s + 1)^3} \tag{38}$$

and its FOPTD approximation:

$$\bar{P}(s) = \frac{1}{2.5s + 1} e^{-1.47s} \tag{39}$$

As specified, the parameters of the SSOD-PI algorithm Δ and β are set to 0.1 and 1, respectively. The parameter t_{max} necessary for [2, 11] is set equal to τ (according to (39)). The parameters of the SOD-PID algorithm are set as $\Delta_P = \Delta$, $\Delta_i = \Delta\tau$, and $\in = \Delta$. All controllers are configured as PI, and they are tuned using the SIMC rules [30], resulting in a control law with $K_P = 0.85$ and $K_i = 0.34$. For a fair comparison, the PI-P controller has been tuned with the same event threshold $\Delta = 0.1$ and the parameter α has been set equal to a not much aggressive value like 0.5. Model (39) is employed in the prediction unit, and the readjustment of the gain K_{P2} is considered to guarantee the steady-state error cancellation. The monitoring of signals is simulated with fast sampling so that signals are measured with a sampling period of $T_E = 0.01$. A unit step set-point is applied at the instant $t = 0$ s, and a load disturbance step with amplitude equal to 1 is applied at the instant $t = 30$ s.

The system responses are shown in Fig. 13. The results of the considered performance indexes are summarized in Table 2. Although the proposed controller is not as fast at rejecting load disturbances because the proposed controller uses a less aggressive setting, the set-point tracking improves significantly regarding the other ones. The obtained steady-state error and settling time confirm this result. In addition, because the response has no overshoot and the t_{max} condition is not considered, the number of events can be reduced to the minimum, as the results show. Note that the PI-P controller achieves the best result in all considered performance indexes and despite the approximations in the model, feature (30) is preserved.

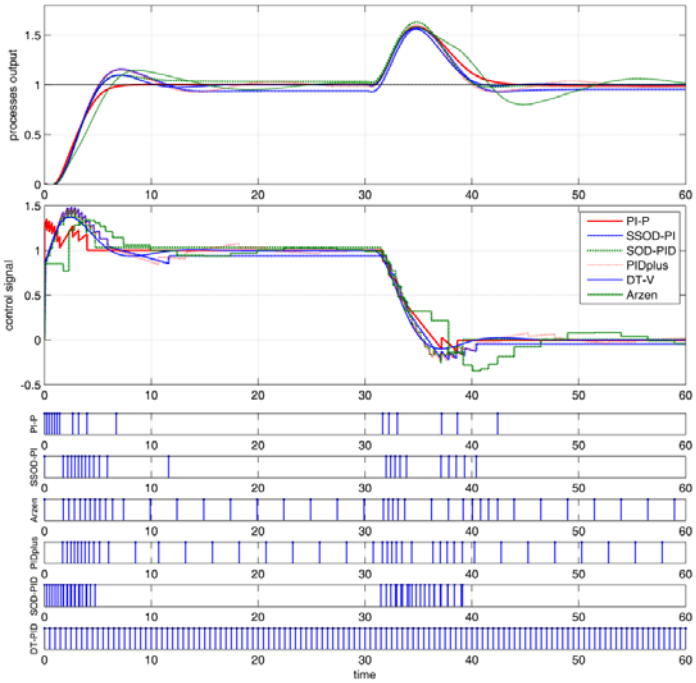

Figure 13: Responses for process (38). First plot: output signals. Second plot: control signals. From third to eighth plot: events of the PI-P, SSOD-PI, Arzen, PIDplus, SOD-PI, and DT-PI controller systems, respectively.

A second experiment for a high-order system with faster dynamics is shown. The system is included in [30] and has the transfer function

$$P(s) = \frac{1}{(s+1)(0.2s+1)(0.04s+1)(0.008s+1)} \tag{40}$$

and the FOPTD approximation:

$$\bar{P}(s) = \frac{1}{1.1s+1} e^{-0.148s} \tag{41}$$

In this case, the PI controller gains were tuned with the SIMC tuning rules again. The obtained parameters K_P and K_i were 3.72 and 3.38, respectively. The parameter t_{max} was set equal to τ. The parameter α was set equal to 1 and the remaining parameters were set as in the previous case. The responses are shown in Fig. 14, and the results are summarized in Table 3. For this process, the SIMC tuning rules generate aggressive controllers which results in control actions with considerable peaks and oscillations. In contrast, the control effort developed by the PI-P controller is moderated and the settling is not substantially higher. Newly, the number of events presented by this controller is the lowest. . In the next section, current simulations are complemented with experimental results, thus concluding the goals of this work.

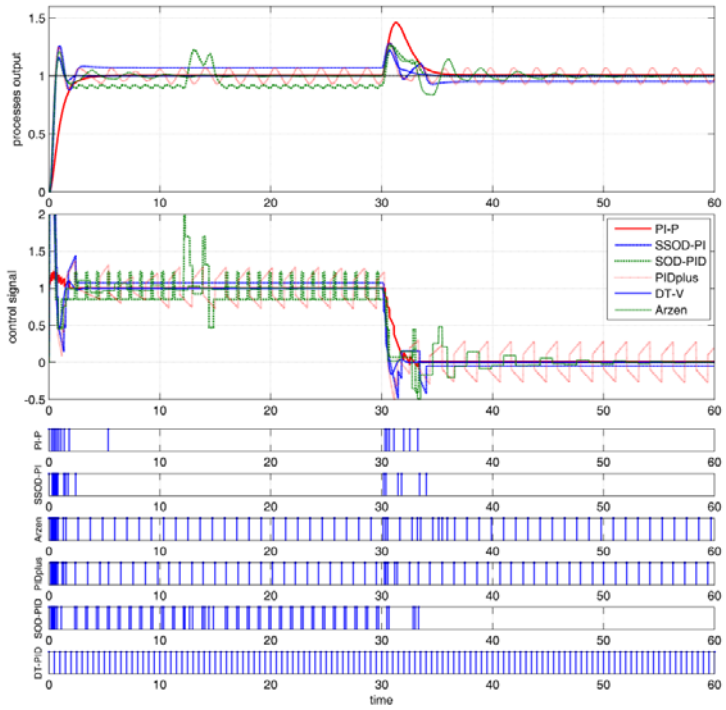

Figure 14: Responses for process (40). First plot: output signals. Second plot: control signals. From third to eighth plot: events of the PI-P, SSOD-PI, Arzen, PIDplus, SOD-PI, and DT-PI controller systems, respectively.

Table 2

Performance indexes. IAE (T_{ST}): integrated absolute error evaluated until settling time. IAU (T_{ST^1}): integrated absolute control signal (according to (28)). T_{ST^1}: for a fair comparison, it indicates that the performance index has been evaluated until the lowest reached T_{ST} for all experiments. $E_{SS}(T_{ST})$: steady-state error as a percentage of event threshold and evaluated at the settling time instant. OV: Overshoot. S: the number of events. T_{ST}: Settling time.

	IAE (T_{ST})	IAU(T_{ST^1})	$E_{SS}(T_{ST})$	OV	S	T_{ST}
PI-P	3.25	1.09	6.12	< 0.5	26	7.51
SSOD-PI	3.84	1.92	54.79	15.74	32	13.36
Arzén	4.51	2.24	47.6	14.79	49	> 20
PIDplus	4.11	2.27	33	15.5	49	> 20
SOD-PID	3.7	1.89	41.93	9.71	54	12.79
DT-PID	3.55	2.25	4.46	9.86	120	15.9

8. Experimental results

The effectiveness of the approach is demonstrated in a real application. For this purpose, an experiment was developed on a laboratory-scale setup from the Computer Science Department of the University of Córdoba. Specifically, the experimental setup consists in the speed control of a brushless DC motor. The motor is bidirectional and can rotate up to 27000 RPM in both directions. The motor is controlled by an analog servo drive that admits a bipolar voltage in the range of ± 10 V, and it uses a tachometer to measure the speed. These measures are converted to a voltage signal that is scaled in the range 0-5 V using a frequency converter. The control and sensor signals are managed using a PCI-6120 DAB from National Instruments.

Table 3

Performance indexes. IAE (T_{ST}): integrated absolute error evaluated until settling time. IAU (T_{ST^1}): integrated absolute control signal (according to (28)). T_{ST^1}: for a fair comparison, it indicates that the performance index has been evaluated until the lowest reached T_{ST} for all experiments. $E_{SS}(T_{ST})$: steady-state error as a percentage of event threshold and evaluated at the settling time instant. OV: Overshoot. S: the number of events. T_{ST}: Settling time.

	IAE (T_{ST})	IAU(T_{ST^1})	$E_{SS}(T_{ST})$	OV	S	T_{ST}
PI-P	1.04	1.18	4.5	< 0.5	26	4.54
SSOD-PI	0.73	1.28	77.69	26.13	31	4.01
Arzén	0.67	2.73	3.2	20.93	85	>20

PIDplus	0.81	2.76	70	24.5	82	L.C.
SOD-PID	1.19	2.51	>100	15.3	86	L.C.
DT-PID	0.44	1.26	2.45	16.1	120	3.41

Because the apparent dead-time of the system is very small with respect to its dominant time constant (and to provide a significant result), a time delay of 2 s was added via software at the plant output. The resulting FOPDT model is given by (42). The model has been obtained by applying the area method to the open-loop response with an input step to move the speed from 1 to 1.5 V.

$$\bar{P}(s) = \frac{0.526}{0.5402s + 1} e^{-2s} \tag{42}$$

The percentage of the output variation that is explained by the model is approximately 91%. With this information, the values obtained for parameters K_P and K_i were 0.26 and 0.48, respectively. The parameter t_{max} was set equal to 0.54. The event thresholds were specified as $\Delta = \Delta_P = \epsilon = 0.05$, $\Delta_i = 0.54\Delta$, and α as 0.4. The sampling time of the time-driven system was set equal to 0.1. The experimental results are shown in Fig. 15. In general, the performance of the controllers is quite similar, but only the PI-P and the SOD-PID controllers achieve a stationary response before the disturbance appears. As in aforementioned experiments, because of the consideration of the Smith predictor in the event-based sampling scheme, the set-point following and the sampling effort are further improved regarding the others control schemes. However, in presented experiments the disturbance rejection is not as fast as the other ones. It is in part because of the dependence of the disturbance rejection task with the system time constant caused by the P controller in the deadband. Even so, during the major part of the control cycle, the PI controller is acting and, consequently, the tuning of the controller (specifically the parameter alpha) plays also a key role to this aim. In this context, thanks to the event generator designed, the tuning of parameters α and Δ that improve such a result (or any desired performance index) can be easily found.

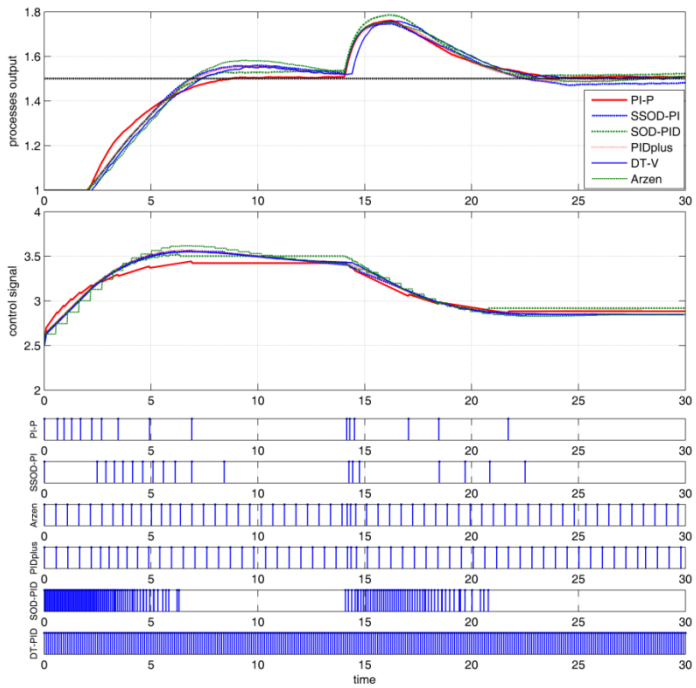

Figure 15: Responses for the DC motor. First plot: output signals. Second plot: control signals of processes. From third to eighth plot: events of the PI-P, SSOD-PI, Arzen, PIDplus, SOD-PI, and DT-PI controller systems, respectively.

9. Conclusions and future work

A new event-based control architecture with a simple and effective tuning methodology has been presented.

Regarding the event-based system structure, a novel event generator and an adaptive controller were used to design the resulting event-based system. The event generator integrates the Smith predictor scheme with an SSOD sampling technique. Such a structure has been proven to reduce the steady-state error, improve the sampling effort and avoid limit cycles. In addition, the designed controller guarantees stability in both disturbance and uncertainty conditions. The stability conditions were characterized, and a robustness analysis was developed. The theoretical analysis is supported by simulations.

Regarding the tuning approach, a framework has been presented and three performance indexes were explained in demonstrations. The approach results both very effective and intuitive because the influence that tuning parameters have on system performance is very clear. Thanks to this predictable behavior of the approach, the considered performance indexes can be estimated in accordance with designer requirements and, moreover, the tuning framework can be easily focused on other required indexes. Simulations and experimental results confirmed the significance of the work in the development of simple tuning rules for this type of event-based controllers.

An algorithm to achieve the set-point tracking with a satisfactory performance despite uncertainties was proposed. In practice, this algorithm would require a measurement of the time interval between events, which could be accomplished using time-stamp methods.

Additionally, system performance was compared with prior works, demonstrating that this event-based controller achieves similar or better performance. An experimental lab process of a brushless DC motor was used to verify the effectiveness of the methodology in real applications.

With the aim of proving the approach in an industrial environment, future work will be focused on analyzing the effect of demanding networked conditions. In addition, feedforward and anti-windup strategies will be explored, the analyses of the stability condition by recursive methods will be revisited and more performance indexes characterized into heuristic or analytic tuning rules.

Acknowledgements

The work of the third and fourth authors has been funded by the National Plan Project DPI2011-27818-C02-02 of the Spanish Ministry of Science and Innovation and FEDER funds.

References

[1] A. Willig, Recent and emerging topics in wireless industrial communication. IEEE Transactions on Industrial Informatics, 4 (2) (2008) 102–124.
[2] T. Blevins, PID advances in industrial control, in: Preprints IFAC Conference on Advances in PID Control, Brescia, Italy, 2012.
[3] R.A. Gupta, M. Y. Chow, Networked control system: Overview and research trends, IEEE Transactions on Industrial Electronics 57 (7) (2010) 2527-2535.
[4] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: a survey, Ad Hoc Networks 7 (3) (2009) 537–568.
[5] J. Sánchez, A. Visioli, S. Dormido, A two-degree-of-freedom PI controller based on events, Journal of Process Control 21 (4) (2011) 639–651.
[6] J. Sánchez, M. Guarnes, S. Dormido, On the Application of Different Event-Based Sampling Strategies to the Control of a Simple Industrial Process, Sensors, 9 (9) (2009), 6795-6818.
[7] V. Vasyutynskyy, K. Kabitzsh, Event-based control Overview and generic model, in: Proceedings of 8th IEEE International Workshop on Factory Communication Systems, 2010 May.

[8] M. Miskowicz, Send-on-delta concept: an event-based data reporting strategy, Sensors 6 (1) (2006) 49–63.
[9] 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.
[10] V. Vasyutynskyy, K. Kabitzsh, Towards comparison of deadband sampling types., in: In Proceedings of IEEE International Symposium on Industrial Electronics, 2007 June.
[11] K.E. Årzén, A simple event-based PID controller, in: Proc. 14th World Congress of IFAC, Beijing, China, 1999.
[12] 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.
[13] S. Durand, N. Marchand, Further results on event-based PID controller, in: Proc. 10th European Control Conference (ECC'09), Budapest, Hungary, 2009.
[14] A. Pawlowski, 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: Proc. 13th IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, 2008.
[15] M. Beschi, S. Dormido, J. Sánchez, A. Visioli, Tuning Rules for Event-based SSOD-PI Controllers, In Proceedings of 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, 2012.
[16] M. Beschi, A. Visioli, S. Dormido, J. Sanchez, 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.
[17] 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.
[18] A. Cervin, K.J. Åström, On limit cycles in event-based control systems, in: Proc. 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 2007.
[19] M. Beschi, S. Dormido, J. Sanchez, A. Visioli , Characterization of symmetric send-on-delta PI controllers. Journal of Process Control 22 (10) (2012) 1930–1945.
[20] B. Hensel, J. Poennigs, V. Vasyutynskyy, K. Kabitzsch, A Simple PI Controller Tuning Rule for Sensor Energy Efficiency with Level-Crossing Sampling, in: Procedings of 9th International Multi-Conference on Systems, Signals and Devices, (SSD 2012), Chemnitz, Germany.
[21] U. Tiberi, J Araujo, K.H. Johansson, On event-based PI control of first-order processes, in: Proceedings IFAC Conference on Advances in PID Control (PID'12), Brescia, Italy, 2012.
[22] P. Tabuada, Event-triggered real-time scheduling of stabilizing control tasks, IEEE Transactions on Automatic Control, 52 (9) 2007 1680-1685.
[23] D. Lehmann. Event-based state-feedback control. PhD thesis, Ruhr-Universitat Bochum, 2011.
[24] X. Wang, M.D. Lemmon, Event-triggering in distributed networked control systems. Automatic Control, IEEE Transactions on, 56 (3) (2011) 586 –601.
[25] S. Durand, N. Marchand, J. Fermi, G. Castellanos, Simple Lyapunov Sampling for Event-Driven Control, 18th IFAC World Congress, Milano, Italy, 2011.
[26] M. C. F. Donkers, W. P. M. H. Heemels, Output-Based Event-Triggered Control With Guaranteed -Gain and Improved and Decentralized Event-Triggering, IEEE Transactions on Automatic Control 57 (6) (2012).
[27] Dormido, S., J. Sánchez, and E. Kofman. Muestreo, control y comunicación basados eneventos. Revista Iberoamericana De Automática e Informática Industrial, 5 (1) (2008) 1697-7912.
[28] K.J. Åström, Event based control, in: A. Astolfi, L. Marconi (Eds.), Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori, Springer Verlag, 2008.
[29] A. O'Dwyer, Handbook of PI and PID Tuning Rules, Imperial College Press, London, UK, 2006.
[30] S. Skogestad, Probably the best simple PID tuning rules in the world. In AIChE Annual meeting, Reno, NV, USA, 2001.