Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

Tuning Rules for Event-based SSOD-PI Controllers

Manuel Beschi, Sebastián Dormido, José Sánchez and Antonio Visioli

Abstract— **This paper deals with the tuning of the parameters of a particular type of event-based PI controller based on send-on-delta (deadband) sampling. In particular, by exploiting the stabilizing region of the parameters, a new tuning rule which aims at minimizing the step response settling time is presented. Results are compared with well-known tuning rules devised for standard (continuous-time) PI controllers. Simulation results show the effectiveness of the proposed tuning rule.**

I. INTRODUCTION

Event-based control has been the subject of much research in the last years (see, for example, [1], [2], [3]) because it allows the reduction of the information flow between the agents involved in the control system. This might be indeed the key issue in some applications (especially in the presence of wireless sensors) when there are constraints on communication rate, rather than achieving a tight performance (in particular a small steady-state error) [4], [5], [6].

Actually, many techniques have been proposed for the design of event-based Proportional-Integral-Derivative (PID) controllers [7], [8], [9], [10], [11], [12], because these kind of controllers are the most employed controllers in industry owing to their advantageous cost-benefit ratio. In this context, one of the most employed event-based sampling strategies is the so called send-on-delta sampling (also known as deadband sampling [13] or level crossing sampling [14]) where the measured value of the process variable is sent to the controller when the process variable (or some function of it) crosses predefined quantization levels [15].

However, it has to be recognized that the great success of (time-based) PID controllers is motivated by the fact that they are capable to provide a satisfactory performance for many processes with a relatively easy design, also because of the large number of tuning rules that are available [16]. Conversely, 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. It has to be noted 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 [17], [18]. Further, in addition to the PID gains, there are other parameters (threshold values) employed in the control algorithm that have to be tuned, thus making the overall control design more complex.

In this paper we present a new type of event-based PI controlled, where the event triggering is performed with a symmetric send-on-delta (for short SSOD) strategy. The derivative action is not employed (as it often happens in industrial settings), because its implementation is very critical with a variable, and possibly long, sampling period. For this controller type and for first-order-plus-dead-time (FOPDT) processes, necessary conditions of the system instability (namely, sufficient conditions of the system stability) and necessary and sufficient condition of the presence of limit cycles have been determined. Moreover, in the SSOD-PI controlled FOPDT system, the SSOD parameter Δ does not influence the stability properties but only the precision and the number of events.

The tuning problem of the SSOD-PI controller is addressed. In particular, an ad-hoc tuning rule is proposed. The rule is obtained by interpolating the controller parameters which minimize the settling time of the system step response.

II. CONTROL ARCHITECTURE AND STABILIZING PARAMETERS

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. 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 others. It is clear that the communication between two entities implies more efforts than data exchanging into a single machine. 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 work, we develope a control strategy for a system where the control and the actuator units are implemented in the same machine (see Figure 1). The first block is composed by 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 by the governor. The sensor unit samples the calculated error with the SSOD technique, which is described later, and if a new event is detected it sends the sampled error to the control unit. The second block is constituted by the control and actuator units which, at a regular sampling rate, determines the control action by taking into account the last received sampled error and applies it to the process. The governor, which, in practice, can be implemented together with one of

M. Beschi and A. Visioli are with the Dipartimento di Ingegneria dell'Informazione, University of Brescia, {manuel.beschi,antonio.visioli}@ing.unibs.it

S. Dormido and J. Sanchez are with the Departamento de Informática y Automática, Universidad Nacional de Education a Distancia, Madrid, Spain, {sdormido,jsanchez}@dia.uned.es

978-1-4673-2529-5/12/$31.00 ©2012 IEEE

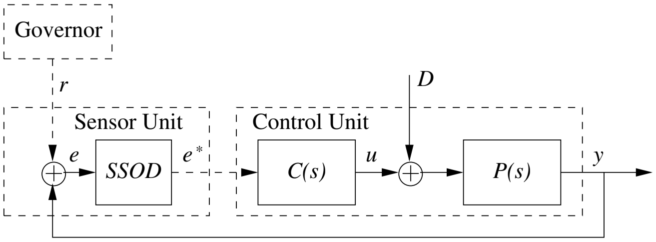

Fig. 1. Control scheme of the event-based controlled system. The dashed arrows indicate data sending on communication medium.

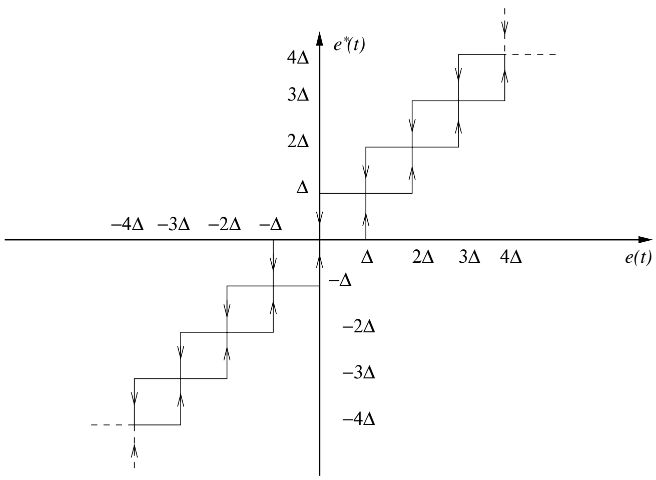

Fig. 2. Relationship between $e(t)$ and $e^*(t)$.

the previous two blocks, receives the desired set-point from a user interface or from a higher hierarchical controller, and sends it to the sensor unit.

The SSOD technique is a special case of the send-on-delta sampling method (see [5], [8]). Denote as $e(t)$ the error signal before the sampling block and as $e^*(t)$ the sampled error, which is forced to assume only values multiple of a predefined threshold Δ, namely $e^*(t) = j\Delta$ with $j \in \mathbb{Z}$. The sampled signal changes its value to the upper quantization level when the error signal $e(t)$ increases more than Δ with respect to the last event, or to the lower quantization level when $e(t)$ decreases more than Δ.

The relationship between $e(t)$ and $e^*(t)$ can be considered as a generalization of a relay with hysteresis, where there is an infinite number of thresholds [14], as shown in Figure 2. As already said, in the control and actuator unit, the control task is implemented by using a standard (discretized) continous time PI controller, namely:

$$C(s) = K_p + \frac{K_i}{s} \tag{1}$$

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

The stability properties and the tuning rules developed in this work are designed by considering a FOPDT process, which can be described by the following transfer function:

$$P(s) = \frac{K}{\tau s + 1} e^{-Ls} \tag{2}$$

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

$$Y(s) = \frac{K}{\tau s + 1} e^{-Ls} \left(U(s) + \frac{D}{s} \right) \tag{3}$$

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 d.

It is important to notice that, because the SSOD-PI algorithm is not a linear controller, the system can reach an equilibrium point, or can present a limit cycle around an equilibrium point or can be unstable. Moreover, the behavior of each equilibrium point can be different. We demonstrate (proofs are omitted for brevity) that it is possible to find a region of the parameter space $K_p - K_i$ where the system is certainly marginally or asymptotically stable for all the equilibrium points. This region can then be divided into two other regions: one where the system can present limit cycles for generic values of r and D and one where the system surely does not present a limit cycle.

The following propositions state the conditions for the system instability and for the absence of limit cycles:

Proposition 1: If the closed-loop system of Figure 1 is unstable then the same system controlled by a continuous-time PI controller with the same value of K_p and K_i is also unstable.

Proposition 2: In a SSOD-P controlled system, if $KK_p < 1$ then limit cycles cannot occur.

Proposition 3: In a SSOD-PI controlled system, with $K_p \geq 0$ and $K_i > 0$, then limit cycles cannot occur if the following conditions are satisfied:

$$\begin{cases} a > \frac{(K_2 l - 1)e^{\frac{2}{K_2}}}{e^{\frac{2}{K_2}} - e^l} & \text{if } a \leq 0 \\ a < a^*(K_2, l) & \text{if } a > 0 \end{cases} \tag{4}$$

where $K_1 = KK_p$, $K_2 = KK_i\tau$, $a = K_1 - K_2$, $l = \frac{L}{\tau}$ and $a^*(K_2, l)$ is found by solving numerically the following equation:

$$\begin{aligned} & K_2\, l e^{2\frac{-K_2 l + 3e^{-l}K_2 l + 2a}{K_2(e^{-l}-1)}} - K_2\, l e^{4\frac{ae^{-l}+1}{K_2(e^{-l}-1)}} \\ & + a e^{4\frac{ae^{-l}+1}{K_2(e^{-l}-1)}} - e^{\frac{-3K_2 l + 7e^{-l}K_2 l + 4a}{K_2(e^{-l}-1)}} \\ & + e^{\frac{4ae^{-l}+4-K_2 l + e^{-l}K_2 l}{K_2(e^{-l}-1)}} - a e^{\frac{2ae^{-l}+2-K_2 l+3e^{-l}K_2 l+2a}{K_2(e^{-l}-1)}} \\ & + a e^{\frac{2ae^{-l}+2-3K_2 l+5e^{-l}K_2 l+2a}{K_2(e^{-l}-1)}} - a e^{2\frac{-K_2 l+e^{-l}K_2 l+2ae^{-l}+2}{K_2(e^{-l}-1)}} \\ & = e^{\frac{4ae^{-l}+4-K_2 l+e^{-l}K_2 l}{K_2(e^{-l}-1)}} - e^{\frac{-3K_2 l+7e^{-l}K_2 l+4a}{K_2(e^{-l}-1)}} \\ & + e^{2\frac{-K_2 l+3e^{-l}K_2 l+2a}{K_2(e^{-l}-1)}} - e^{4\frac{ae^{-l}+1}{K_2(e^{-l}-1)}} \end{aligned}$$

Proposition 1 states that instability of the continuous-time PI controller is a necessary condition for the instability of a SSOD-PI controller with the same gains. Thus, if the continuous-time PI controlled system is stable also the SSOD-PI controlled system is (marginally or asymptotically) stable. Propositions 2-3 allow us to find the portion of the

plane of the normalized gains K_1–K_2 where the system is asymptotically stable. In other words, in this region there are surely no limit cycles, whereas out of this region there is surely at least a possible limit cycle.
Another important consideration that can be deduced from Propositions 1-3 is that the stability properties are not influenced by the parameter Δ and therefore it can be chosen using only considerations about the desired precision and the number of events (see Section III).

III. Tuning Rules

In Section II we presented conditions on the stability and on the absence of limit cycles (see Proposition 1-3). However, the choice of the correct parameters remains a complicated task because of the absence of specific tuning rules for SSOD-PI controller or/and of an analysis of the behavior of standard tuning rules applied in a SSOD-PI controlled system.
For these reasons, in this section we present a tuning rule which is specifically tailored for SSOD-PI controlled FOPDT system and we compare it with the well-know AMIGO [19] and SIMC [20] tuning rules, which have been proven to be effective for standard PI control systems.
The proposed tuning rule, called ST (namely, settling time) for the sake of brevity, has been obtained by considering many normalized FOPDT processes (namely, with $K = 1$ and $\tau = 1$) with a normalized dead time $\frac{L}{\tau} \in [0.1, 3]$. In particular, a hundred equally spaced ratios $\frac{L}{\tau} \in [0.1, 3]$ have been considered. Then, for each process, all the event-based PI controllers with normalized PI gains K_1–K_2 obtained by tightly gridding the stabilizing region determined by applying Proposition 3 have been applied (with a value of Δ equal to the 10% of the set-point amplitude). For each simulation, the 1% settling time t_{st} of the step response (that is, the time interval from the application of the step signal to the set-point to the time instant when the process variable remain inside a band $\pm 1\%$ around its final value) has been calculated. The sets K_p–K_i which minimize t_{st} for each process have been eventually interpolated to obtain the following equations:

$$K_1 = c_0 + c_1 \left(\tfrac{L}{T}\right)^{-1} + c_2 \left(\tfrac{L}{T}\right)^{-2} + c_3 \left(\tfrac{L}{T}\right)^{-3} + c_4 \left(\tfrac{L}{T}\right)^{-4},$$
$$K_2 = d_0 + d_1 \left(\tfrac{L}{T}\right)^{-1} + d_2 \left(\tfrac{L}{T}\right)^{-2} + d_3 \left(\tfrac{L}{T}\right)^{-3} + d_4 \left(\tfrac{L}{T}\right)^{-4} \quad (5)$$

which represent the ST tuning rule. The coefficients c_i and d_i are shown in Table III. The proposed tuning rules have been compared with the AMIGO and the SIMC tuning rules for (continuous-time) PI controllers. With the aim to compare the different tuning rules, we introduce eleven performance indexes which can divided into four groups related to the robustness, the set-point following task, the load disturbance rejection task and the influence of the parameter Δ. The indexes are calculated for processes with $K = 1$, $\tau = 1$ and $L \in [0.1, 3]$. If it is not specified, the value of Δ is selected equal to 0.1.
The system robustness is described by three indexes, which represent how the system parameters (K, T and L) have to

TABLE I
Coefficients of Equation (5)

	$L \leq 1.5\tau$	$L > 1.5\tau$
c_0	0.134143	-0.067030
c_1	0.312852	1.855153
c_2	0.093225	-4.625441
c_3	-0.019254	6.332283
c_4	0.001161	-3.141664
d_0	0.091140	0.225399
d_1	0.345212	-1.342304
d_2	0.081926	6.036159
d_3	-0.017906	-8.735594
d_4	0.001107	4.660420

change with respect to their nominal values ($\bar{K}$, $\bar{T}$ and $\bar{L}$) to get the normalized gains K_1 and K_2 out of the limit curve (4). It is worth to stressing that when the system parameters change (keeping K_p and K_i constant) the normalized gains and the normalized delay $l = \frac{L}{T}$ also change with respect to their nominal values $\bar{K}_1$, $\bar{K}_2$ and $\bar{l}$, in fact:

$$K_1 = \bar{K}_1 \frac{K}{\bar{K}},\ K_2 = \bar{K}_2 \frac{K}{\bar{K}} \frac{T}{\bar{T}},\ l = \bar{l} \frac{L}{\bar{L}} \frac{\bar{T}}{T} \quad (6)$$

The indexes are defined as:

- *Process gain margin* M_K, which indicates the maximum increment allowed by the process gain K by preserving the system stability. In fact, if the real process gain K increases to $M_K \bar{K}$, the normalized gains increase and reach the limit curve (4), which does not change;
- *Time constant margin* M_τ, which indicates the maximum increment allowed by the time constant τ by preserving the system stability. Note that an increment of τ results in an increment of the normalized gain K_2 but at the same time reduces the ratio l (see (6)) thus enlarging the stabilizing region;
- *Time delay margin* M_L, which indicates the maximum increment allowed by the dead time L by preserving the system stability. In case of an incorrect estimation of the L, the normalized gains do not change but the stabilizing region decreases (see (6))

The set-point step response performance is measured by using three indexes, namely:

- *The settling time* st_Δ, which is the time elapsed from the application of a unit step change of the set-point, starting by null initial conditions, to the time instant when the process variable enters and remains within a band $\pm\Delta$ around the final steady-state value;
- *The 1% settling time* t_{st}, which is the time elapsed from the application of a unit step change of the set-point, starting by null initial conditions, to the time at which the process variable derivative enters and remains within a band $\pm 1\%$ of its final value;
- *The percentage overshoot* PO which indicates how much the process variable exceeds the set-point value.

The rejection of a step load disturbance performance is measured by using three indexes, namely:

- *The settling time* st_Δ, which is the time elapsed from the application of an unit step load disturbance, starting by

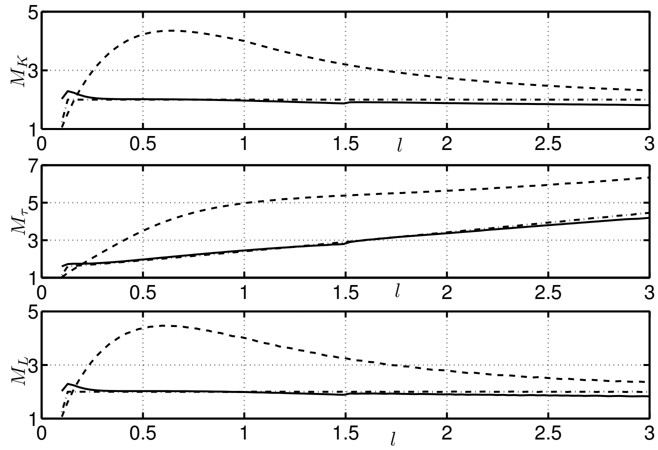

Fig. 3. Robustness indexes of the three tuning rules (solid line: ST; dashed line: AMIGO, dash-dot line: SIMC) versus $\frac{L}{\tau}$. Top: gain margin M_K. Middle: time constant margin M_τ. Bottom: time delay margin M_L.

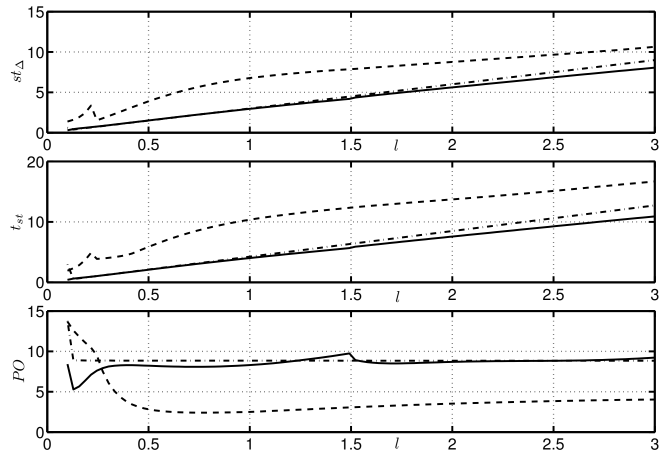

Fig. 4. Set-point step response indexes of the three tuning rules (solid line: ST; dashed line: AMIGO; dash-dot line: SIMC) versus $\frac{L}{\tau}$. Top: settling time st_Δ. Middle: steady-state time t_{st}. Bottom: percentage overshoot PO.

null initial conditions, to the time at which the process variable enters and remains within a band $\pm\Delta$ of around the set-point value;

- *The steady-state time* t_{st}, which is the time elapsed from the application of an unit step load disturbance, starting by null initial conditions, to the time at which the process variable derivative enters and remains within a band $\pm 1\%$ of its final value;
- *The maximum output error* E_{max} which indicates the maximum control error after the application of a unit step load disturbance.

The previous indexes give us information on performance which mainly depends on K_1 and K_2. Conversely, the number of events and the precision depend mainly by the parameter Δ. To understand these dependencies, the value of Δ is varied in the interval $[10^{-3}, 0.1]$ and two indexes are calculated:

- *The number of events* n_e, which is the number of events generated by the controller when a unit step change of the set-point, starting by null initial conditions, is applied.
- *The maximum distance* Δ_y between the SSOD-controlled system response and the response of a standard discrete-time controlled system with sample time $h = 10^{-4}\tau$.

Figure 3 shows the values of the robustness indexes M_K, M_τ and M_L with respect to the normalized dead time $\frac{L}{\tau}$ for the three control strategies. It is possible to see that the AMIGO rule gives the greatest robustness indexes. Anyway, all the three strategies present a good robustness, in fact they have margins greater than 1.5 for all the values of $\frac{L}{\tau} \in [0.1, 3]$. In Figure 4, the values of the set-point step response indexes with respect to $\frac{L}{\tau}$ are shown. It can be noted that from small normalized dead times the best performance is obtained by using the ST rule. However, the SIMC rule presents a quite similar performance. It is important to note that ST and SIMC rules presents an overshoot less than 10% and

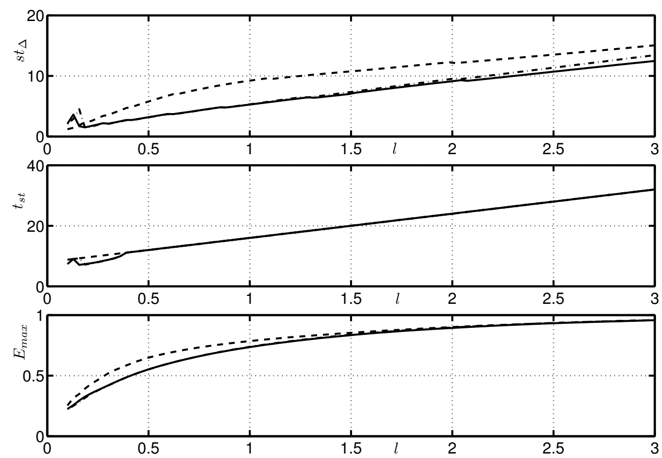

Fig. 5. Load disturbance response indexes of the three tuning rules (solid line: ST; dashed line: AMIGO; dash-dot line: SIMC) with respect to $\frac{L}{\tau}$. Top: settling time st_Δ. Middle: steady-state time t_{st}. Bottom: peak error E_{max}.

therefore they allow the system to reach the set-point value in the minimum number of events (namely, 11 with $\Delta = 0.1$ and an unitary step variation). Figure 5 shows the step load disturbance rejection indexes with respect to the ratio $\frac{L}{\tau}$. It is possible to note that also in this case the ST rule presents better indexes values than the other two control strategies. However the performance provided by the SIMC tuning rule is quite similar. It is important to note that the index E_{max} tends to 1 for high ratios $\frac{L}{\tau}$ for all the sets of K_p–K_i because the controller can start to reject the disturbance only after an interval equal to the time delay. In Figure 6 the number of events n_e and the maximum distance Δ_y with respect to Δ are plotted for the case $\frac{L}{\tau} = 0.8$ (similar conclusions can be drawn for the other values). We can observe that the maximum difference between the time-driven and the event-driven control systems increases quasi-linearly with respect to the parameter Δ and the number of events n_e depends quasi-linearly with respect to Δ^{-1}. Note that the peaks present in Figure 6 mainly depend by the steady-state

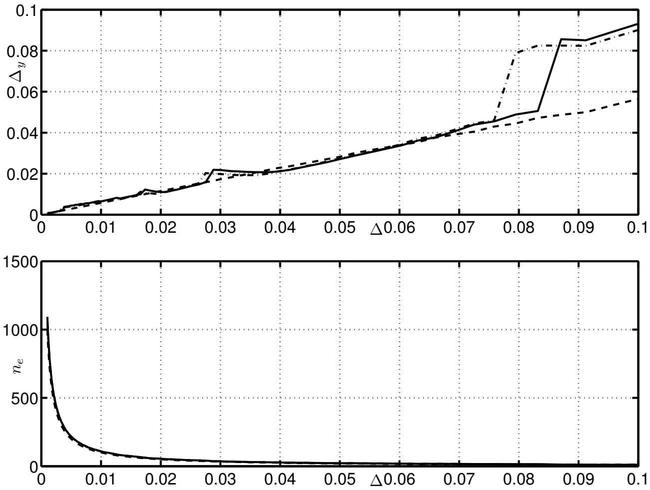

Fig. 6. Influence of Δ on the number of events and system precision of the three tuning rules (solid line: ST; dashed line: AMIGO; dash-dot line: SIMC) versus Δ with $L = 0.8T$. Top: maximum distance Δ_y. Bottom: number of events n_e.

value assumed by the process output, which can be close to the reference or close to the switches thresholds. Conversely, they are not significant affected by the value $\frac{L}{\tau}$. Thus, we can approximate the trend of n_e and E_{max} as:

$$n_e \simeq \frac{\alpha}{\Delta}, \quad E_{max} \leq \Delta \tag{7}$$

where α is a proportionality constant. It is important to note that if the tuning rule gives an overshoot smaller than Δ (which is true for the ST and SIMC tuning rules,) then $\alpha \simeq 1$. Equations (7) are very useful to choose the parameter Δ when the desired number of events or the desired precision are known. Another important consideration is that for big values of Δ (typically around a tenth of the set-point variation) the ST rule, which is an *ad-hoc* rule, gives better performance than the other two rules. Conversely, for small values of Δ the performance of the event-driven control system tends, as expected, to be equal to the correspondent time-driven control system. However, also in this case the ST rule gives a good performance in comparison with the AMIGO and SIMC rules (see Section IV).

IV. Simulation results

In this section, simulation results are provided. In particular, the three rules are compared using a FOPDT process and a high-order process approximated to an FOPDT process using the half rule (see [20]).

As a first example consider the FOPDT process

$$P(s) = \frac{K}{\tau s + 1} e^{-Ls} \tag{8}$$

with the following parameters: $K = 2$, $\tau = 6$ and $L = 4$. In Figures 7-8 the control system response to a set-point step change and to a step load disturbance are shown. In particular, in Figure 7 the parameter Δ is chosen equal to 0.2 while in Figure 8 it is set equal to 0.001. It appears that

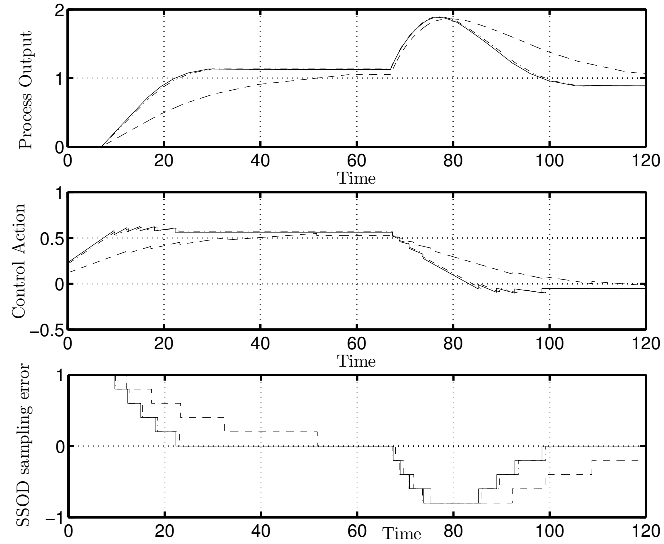

Fig. 7. Response of (8) with the three tuning rules to a set-point step change and to a step load disturbance with $\Delta = 0.2$ (solid line: ST; dashed line: AMIGO; dash-dot line: SIMC). Top: process variable. Middle: control variable. Bottom: SSOD sampled error.

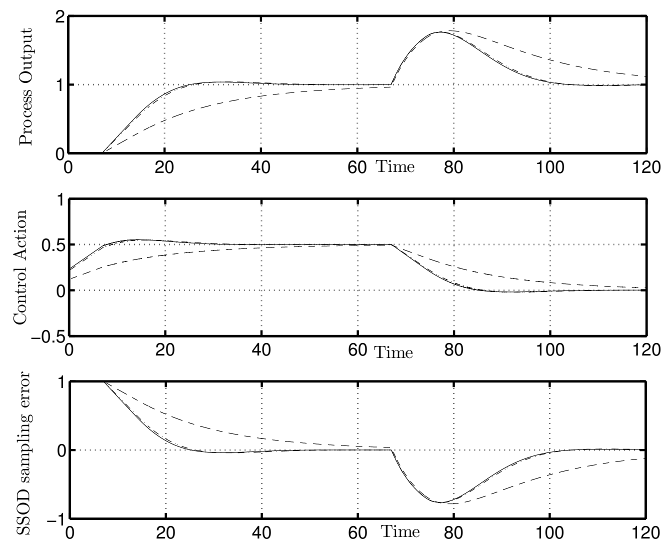

Fig. 8. Response of (8) with the three tuning rules to a set-point step change and to a step load disturbance with $\Delta = 0.001$ (solid line: ST; dashed line: AMIGO; dash-dot line: SIMC). Top: process variable. Middle: control variable. Bottom: SSOD sampled error.

the ST tuning rules give the best performance with a great value of Δ and more or less the same performance of the SIMC tuning rules with small value of the SSOD parameter. Actually, in this case the SSOD-PI control system behaves quite similarly to the time-driven control system.

As a second example, consider the process (see [21])

$$G(s) = \frac{1}{(s+1)(0.2s+1)(0.04s+1)(0.008s+1)} \tag{9}$$

which is approximated as a FODPT process with $K = 1$, $\tau = 1.1$ and $L = 0.148$ using the half rule method. Figures 9-10 show the step responses obtained with the three tuning rules with a large value of Δ ($\Delta = 0.2$) and a small value ($\Delta = 0.001$), respectively. Also in this case, we can do the same considerations of the first example, confirming the

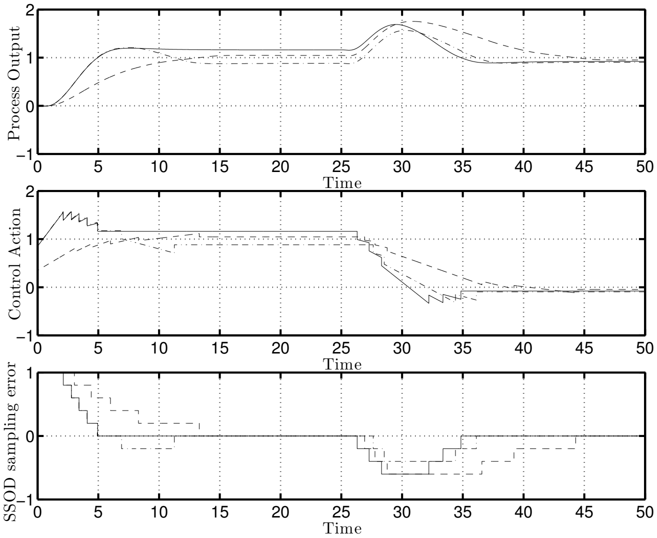

Fig. 9. Response of (9) with the three tuning rules to a set-point step change and to a step load disturbance with $\Delta = 0.2$ (solid line: ST; dashed line: AMIGO; dash-dot line: SIMC). Top: process variable. Middle: control variable. Bottom: SSOD sampled error.

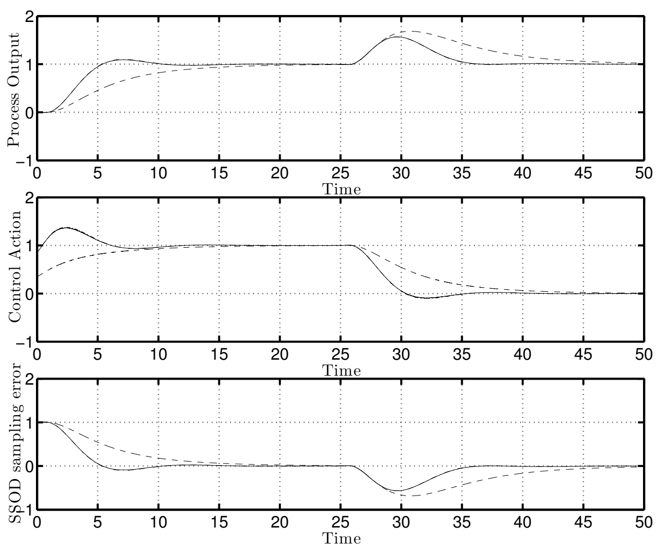

Fig. 10. Response of (9) with the three tuning rules to a set-point step change and to a step load disturbance with $\Delta = 0.001$ (solid line: ST; dashed line: AMIGO; dash-dot line: SIMC). Top: process variable. Middle: control variable. Bottom: SSOD sampled error.

effectiveness of the ST rule.

V. Conclusions

This paper deals with the design of a new *ad-hoc* tuning rule for event-based SSOD-PI controlled systems. This tuning rule has been compared with AMIGO and SIMC rules confirming its effectiveness.

Another aspect addressed in this paper has been to define indexes to compare tuning rules in SSOD control and to analyze how the new and the standard tuning rules are influenced by the SSOD parameter Δ. The SSOD-PI controller appears to be a valuable event-based strategy, in fact it provides the same properties of the tuning rules developed for time-driven systems. Moreover, it requires only a parameter (namely, Δ) to describe the SSOD triggering, which has a clear relation with the number of events and the system precision.

VI. Acknowledge

Part of his work 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] K. Åström, "Event based control," *In Astolfi, A. and Marconi, L. (eds.), Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori*, 2008.

[2] V. Vasyutynskyy and K. Kabitzsh, "Event-based control: Overview and generic model," in *Proccedings of 8th IEEE International Workshop on Factory Communication Systems (WFCS)*, May 2010.

[3] W. Heemels, J. Sandee, and P. Van Den Bosch, "Analysis of event-driven controllers for linear systems," *International Journal of Control*, vol. 81, no. 4, pp. 571–590, 2008.

[4] P. Otanez, J. Moyne, and D. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proceedings of American Control Conference*, Anchorage, USA, 2002.

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

[6] V. Pawlowskyy, J. L. Guzmán, F. Rodríguez, M. Berenguel, J. Sanchez, and S. Dormido, "Event-based control and wireless sensor network for greenhouse diurnal temperature control: A simulated case study," in *Proc. of 13th IEEE International Conference on Emerging Technologies and Factory Automation (EFTA-2008)*, Hamburg, Germany, 2008.

[7] K. Årzèn, "A simple event-based PID controller," in *Proceedings of 14th World Congress of IFAC*, Bejining, China, 1999.

[8] V. Vasyutynskyy and K. Kabitzsh, "Implementation of PID controller with send-on-delta sampling," in *International Conference Control*, 2006.

[9] M. Rabi and K. H. Johansson, "Event-triggered strategies for industrial control over wireless networks," in *Proceedings of 4th Annual International Conference on Wireless Internet*, Maui, Hawaii, USA, 2008.

[10] S. Durand and N. Marchand, "Further results on event-based pid controller," in *Proceedings of 10th European Control Conference (ECC'09)*, Budapest, Hungary, 2009.

[11] J. Sanchez, A. Visioli, and S. Dormido, *PID Control in the Third Millennium (R. Vilanova and A. Visioli (eds.))*. Springer, 2012, ch. Event-based PID control, pp. 495–526.

[12] ——, "A two-degree-of-freedom PI controller based on events," *Journal of Process Control*, vol. 21, pp. 639–651, 2011.

[13] V. Vasyutynskyy and K. Kabitzsh, "Towards comparison of deadband sampling types," in *IEEE International Symposium on Industrial Electronics (ISIE 2007)*, June 2007.

[14] E. Kofman and J. Braslavsky, "Level crossing sampling in feedback stabilization under data rate constraints," in *Proceedings of 45th IEEE International Conference on Decision and Control*, 2006.

[15] J. Sánchez, M. Guarnes, S. Dormido, and A. Visioli, "Comparative study of event-based control strategies: An experimental approach on a simple tank," in *Proceedings European Control Conference*, Budapest, Hungary, 2009.

[16] A. O'Dwyer, *Handbook of PI and PID Tuning Rules*. London, UK: Imperial College Press, 2006.

[17] V. Vasyutynskyy, "Limit cycles in PI control loops with absolute deadband sampling," in *Proceedings of 18th IEEE International Conference on Microwave & Telecommunication Technology (CRIMICO'2008)*, Sevastopol, Ukraine, 2008.

[18] A. Cervin and K. J. Astrom, "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.

[19] K.J. Åström and T. Hägglund, "Revisiting the Ziegler-Nichols tuning rules for PI control," *Asian Journal of Control*, vol. 4, no. 4, p. 364380, Dec. 2002.

[20] S. Skogestad, "Probably the best simple PID tuning rules in the world," in *AIChE Annual meeting*, Reno, NV, USA, 2001.

[21] K. Åström, H. Panagopoulos, and T. Hägglund, "Design of PI controllers based on non-convex optimization," *Automatica*, vol. 34, no. 5, pp. 585–601, 1998.