Journal of Process Control 23 (2013) 826–838

Contents lists available at SciVerse ScienceDirect

Journal of Process Control

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

Characterization of limit cycles for self-regulating and integral processes with PI control and send-on-delta sampling

Jesús Chacón[a,*], José Sánchez[a], Antonio Visioli[c], Luis Yebra[b], Sebastián Dormido[a]

[a] Universidad Nacional de Educación a Distancia, Spain
[b] CIEMAT - Plataforma Solar de Almería, Spain
[c] Università degli Studi di Brescia, Italy

ARTICLE INFO

Article history:
Received 6 November 2012
Received in revised form 7 February 2013
Accepted 2 April 2013
Available online 31 May 2013

Keywords:
PID control
Limit cycles
Time delayed processes

ABSTRACT

This work is focused on the study of limit cycles that appear in a control scheme which is based on the use of a PI controller with an event-based send-on-delta sampling (SOD). The processes investigated are integrator processes plus time delay (IPTD) and first and second order processes plus time delay, which are of interest because they are frequently used to model many industrial processes. The SOD sampling is characterized as a non-linearity of n levels with hysteresis. An algorithm to calculate the limit cycles properties is proposed, and then the results obtained in simulations are compared with experiments performed on a real plant, a distributed solar collector field at the Solar Platform of Almería (PSA, Spain).

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Traditionally, the control systems used in the different engineering areas have been based on periodic sampling, for which there is a wide range of mathematical tools and well established theoretical results (for example, [1]). On the contrary, event-based sampling schemes appear in a natural way in wireless sensor networks (WSN) where the controller or the sensors send their outputs according to the violation of some condition. Also, event-based strategies have been used for long time in areas such as control of industrial processes [2], robot path planning [3] and engine control [4]. However, event-based control has been used mainly in an *ad-hoc* way, due to the lack of theoretical results, which have begun to be available only in the last years (for example [5–7]). Recently, event-based control is also being used in multi-agent [8] and distributed systems [9].

Event-based systems are frequently classified into two types, *event-triggered* systems [10], which usually do not have a model of the plant, where the changes measured in the state provoke sporadic updates of the controller, and *self-triggered* systems [11,12], where the controller calculates the time when the next event is triggered, based on an estimation of the state of the plant obtained from a model.

Some works related to event-triggered systems are [9], where authors develop a distributed event-triggered estimation algorithm for networked systems, or [13,14], where it is used a periodic event-triggered strategy, which is a variant of event-triggered where the event condition is verified with a constant sampling period. In [15], authors implement a distributed event-triggered control system based on local event generators and prove practical stability. Finally, in [16], authors analyze the relationship between sampled-data control and two event-triggered strategies, namely, deadband control and model-based event-triggered control.

With respect to self-triggered systems, a simple self-triggered sampler for nonlinear systems is presented in [17]. Another self-triggered sampler for nonlinear systems is developed in [18,19], under the assumption that the closed-loop system with continuous-time control is input-to-state stable (ISS) with respect to measurement errors. In [20], authors present a scheme that ensures finite-gain L2 stability of the resulting self-triggered feedback systems.

Proportional-Integral-Derivative (PID) controllers are widespread in industry, mainly because of their satisfactory performance for many processes and because they are relatively easy to design and tune. Thus, in recent times many authors have addressed the design and implementation of event-based PID controllers ([21,22]). A comprehensive survey of the different methods proposed in the literature for event based PID control can be found on [23].

This work is focused on *event-triggered* sampling, and in particular on the level crossing or *send-on-delta* sampling [21], which consists in taking a new sample when a change greater than a predefined threshold δ is detected in the signal. In practice, this scheme is equivalent to introduce a non-linearity that can be characterized as a quantization of n levels with hysteresis.

* Corresponding author. Tel.: +34 913987147.
E-mail address: jchacon@bec.uned.es (J. Chacón).

0959-1524/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jprocont.2013.04.001

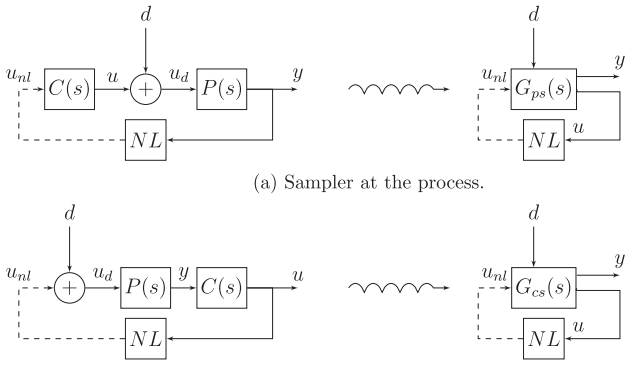


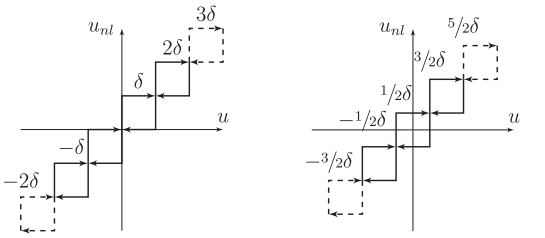
Fig. 1. Event-based control schemes. The block diagrams on the left correspond to the two proposed configurations, (a) the event-based sampler at the process output and (b) at the controller output. On the right, the continuous dynamics have been grouped in one block to simplify the analysis. Solid lines represent a continuous data flow, while dotted lines mean a discontinuous data flow.

The behaviour of a control scheme based on level crossing sampling is studied considering two possible structures, the first one is when the sampler is located after the process output (Fig. 1a), and the second one after the controller output (Fig. 1b). Each case represents a configuration of a control scheme based on wireless transmissions, and has different properties. The first case corresponds to a plant with a wireless sensor which takes measures of the process variable and sends them to the controller, physically separated from the sensor but connected to the actuator. The second case is the opposite, where the controller is directly connected to the sensor and the actuator is in other place.

The interest is to characterize with a systematic approach the behaviour of the two event-based control structures with a set of processes such as integrator processes plus time delay process (IPTD), first order processes plus time delay (FOPTD), and second-order processes plus time delay (SOPTD). The analysis is focused on the conditions for the existence of limit cycles, their period and amplitude, and the effect of external disturbances and the wind-up phenomena in the process due to the saturation of the actuators.

There are other works in the literature concerned to the study of limit cycles in event-based systems. In [24], authors analyze the existence, properties and stability of limit cycles in relay systems. Related results can also be found in [25], where an event-based control scheme with a simple threshold detector is investigated, first in a double integrator and afterwards in the general case. Another approach can be found in [26], where the proposed structure is a model-based event-triggered PI in which the events are generated by the difference between the plant and the model implemented in the controller/sensor. The effect of actuator saturation is also investigated. In [27], a new method to perform the global stability analysis in Piecewise Linear Systems (PLS) is proposed. The method is based on the use of quadratics Lyapunov functions in the switching surfaces. Finally, in [28,29], authors analize symmetric limit cycles in send-on-delta PI controllers, focusing on the stability of FOPTD processes and proposing tuning rules for this kind of controllers.

The main contributions of this work are the proposition of two general event-based schemes, and a new method for the analysis of the limit cycles that appear in the two presented schemes when it is applied to LTI systems with time delay. The method has been applied to study a set of the most common industrial processes, obtaining results that have been confirmed in practice. In particular, a set of experiments carried out in the Distributed Collector Solar Field at the Solar Platform of Almeria (PSA) showed the expected behaviour.



(a) Non-centered sampling. (b) Centered sampling.

Fig. 2. Two cases of send-on-delta sampling with different offset.

2. The LTI and SOD sampler blocks

In the event-based control structure used in this work, three elements are present: the multilevel non-linearity with hysteresis represented by the SOD sampler, the process (IPTD, FOPTD or SOPTD), and the PI controller. In order to redefine the event-based system as a PLS, first it is necessary to group the dynamics of the two linear elements in one block, that is, process and controller. Once this is done, the redefinition of the system as a PLS is simple since feedbacking the new linear block with the non-linearity is equivalent to introduce a rule to switch between the n systems obtained by the combination of the dynamics of the linear block and the n output levels of the sampler.

Fig. 2a shows the non-linearity corresponding to the non-centered level crossing with saturation, and Fig. 2b represents a centered sampling with saturation. The dotted lines in the plots mean that the sampler has a saturation, but in general the levels can be extended in both sides.

2.1. The non-linear block: the SOD sampler

The *event-triggered* control systems analysed in this work use a level crossing sampling strategy, where, depending on the sampler location, the sensor sends information to the controller only when the sampled signal crosses certain predefined levels, or the controller sends the new values of the control action to the actuator when there is a significative change with respect to the previous value. The level crossing is considered the event that triggers the capture and the sending of a new sample.

Formally, a SOD sampler can be thought of as a block which has a continuous signal $u(t)$ as input and generates a sampled signal $u_{nl}(t)$ as output, which is a piecewise constant signal with $u_{nl}(t) = u(t_k)$, $\forall t \in [t_k, t_{k+1})$. Each t_k is denoted as *event time*, and it holds $t_{k+1} = \inf\{t | t > t_k \wedge |u(t) - u(t_k)| \geq \delta\}$, except for t_0, which is assumed to be the time instant when the block is initialized as $u_{nl}(t_0) = u(t_0)$.

In addition, this signal can be saturated due to the limitations in the sensors or in the actuators. As said before, the pattern resulting from sampling a signal with a SOD sampler can be described as a non-linearity of n levels with hysteresis. It can be characterized as

$$u_{nl}(t) = \begin{cases} (i+\alpha+1)\delta & \text{if } u(t) \geq (i+1)\delta \wedge u_{nl}(t^-) = (i+\alpha)\delta \\ (i+\alpha)\delta & \text{if } u(t) \in ((i+\alpha-1)\delta, (i+\alpha+1)\delta) \wedge u_{nl}(t^-) = (i+\alpha)\delta \\ (i+\alpha-1)\delta & \text{if } u(t) \leq (i-1)\delta \wedge u_{nl}(t^-) = (i+\alpha)\delta \end{cases} \quad (1)$$

where $\alpha \in [0, 1)$ is the offset with respect to the origin, $i \in \mathbb{Z}$ if the sampler is without saturation, and $i \in [i_{min}, i_{max}]$ when the sampler is with saturation.

Depending on the initial value, the non-linearity introduced could have an offset with respect to the origin, $\alpha = u(t_0) - i\delta$, where $i = \lfloor u(t_0)/\delta \rfloor$ is the closest level from below. The level crossing sampling with offset is formally defined in the following paragraphs.

Definition 1. Let $T = \{t_0, \ldots, t_n\}$, with $t_k \in \mathbb{R}$ and $t_{k-1} < t_k$, be a set of sampling times, and $U = \{u(t_0), \ldots, u(t_n)\}$ a set of samples. Thus, U

is a level crossing sampling of $u(t)$ if, and only if, $|u(t_k) - u(t_{k-1})| = \delta$, $k = 0, 1, \ldots, n$.

Note that every sample can be expressed as $y_s(t_k) = (i_k + \alpha)\delta$, where $i_k \in \mathbb{Z}$ is the crossed level by the input signal $y(t)$, and $\alpha \in [0, 1) \subset \mathbb{R}$ is the sampling offset which depends on the initial sample, $y(t_0)$.

Definition 2. The number of levels of a non-linearity with hysteresis is defined by subtracting 1 to the number of crossing levels.

For example, a 2-level non-linearity with hysteresis and zero offset owns three crossing levels, that is, $-\delta$, 0, and δ. It must be noticed that if the non-linearity has zero offset the number of levels is always even, and odd in the opposite case.

2.2. The linear blocks: the process and the controller

As Fig. 1 shows, the linear dynamics of the process and the PI controller can be joined in two different ways by placing the non-linear block at the process output (Fig. 1a) or at the controller output (Fig. 1b). Depending on the combination, two different linear blocks G_{ps} and G_{cs} are obtained and, once the loop is closed with the sampler, two PLS are produced with different limit cycles features. The dynamics of the G_{ps} and G_{cs} blocks will be represented by the augmented state matrices obtained by combining process and controller.

2.2.1. Sampling the process variable: the G_{ps} block

Let us start obtaining the matrices corresponding to the case presented in Fig. 1a, that is, the SOD sampler located at the process output. No delays are considered now. Assume that the process $P(s)$ is described by the time-invariant state-space system

$$P(s) \sim \begin{cases} \dot{x}_p(t) = A_p x_p(t) + B_p u_d(t) \\ y(t) = C_p x_p(t) \end{cases}, \tag{2}$$

where $x_p \in \mathbb{R}^m$ is the state, A_p is non-singular, and $u_d(t)$ is the control action resulting from adding to the output $u(t)$ of the PI controller a disturbance $d(t)$. This process is controlled by a PI with proportional gain k_p and integral gain k_i, which is modeled as

$$C(s) \sim \begin{cases} \dot{x}_c(t) = A_c x_c(t) + B_c u_{nl}(t) \\ u(t) = C_c x_c(t) + D_c u_{nl}(t) \end{cases}, \tag{3}$$

where $A_c = 0$, $B_c = 1$, $C_c = -k_i$, $D_c = -k_p$, and $u_{nl}(t)$ represents the input to the controller. Combining both dynamics and considering a constant disturbance, $d(t) = d$, the continuous part of the loop dynamics, $G_{ps}(s) = C(s)P(s)$, can be expressed by

$$G_{ps}(s) \sim \begin{cases} \dot{X}_{ps}(t) = A_{ps} X_{ps}(t) + B_{ps} U_{ps}(t) \\ Y_{ps}(t) = C_{ps} X_{ps}(t) + D_{ps} U_{ps}(t) \end{cases}, \tag{4}$$

where

$$X_{ps} = \begin{bmatrix} x_p \\ x_c \end{bmatrix} \quad Y_{ps} = \begin{bmatrix} y \\ u \end{bmatrix} \quad U_{ps} = \begin{bmatrix} u_{nl} \\ d \end{bmatrix}$$
$$A_{ps} = \begin{bmatrix} A_p & -k_i B_p \\ 0 & 0 \end{bmatrix} \quad B_{ps} = \begin{bmatrix} -k_p B_p & B_p \\ 1 & 0 \end{bmatrix}.$$
$$C_{ps} = \begin{bmatrix} C_p & 0 \\ 0 & -k_i \end{bmatrix} \quad D_{ps} = \begin{bmatrix} 0 & 0 \\ -k_p & 0 \end{bmatrix}$$

2.2.2. Sampling the control variable: the G_{cs} block

The procedure to obtain the augmented state matrices of the G_{cs} block is similar to the previous case but taking into account that now the input to the controller $u(t)$ is the process output $y(t)$ and the input to the process is the signal resulting of adding the disturbance $d(t)$ to an input named $u_s(t)$. The resulting system $G_{cs}(s)$ is

$$\begin{bmatrix} \dot{x}_p(t) \\ \dot{x}_c(t) \end{bmatrix} = \begin{bmatrix} A_p & 0 \\ C_p & 0 \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} + \begin{bmatrix} B_p & B_p \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{nl}(t) \\ d \end{bmatrix}, \quad \begin{bmatrix} y(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} C_p & 0 \\ -k_p C_p & -k_i \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} \tag{5}$$

$$G_{cs}(s) \sim \begin{cases} \dot{X}_{cs}(t) = A_{cs} X_{cs}(t) + B_{cs} U_{cs}(t) \\ Y_{cs}(t) = C_{cs} X_{cs}(t) \end{cases}, \tag{6}$$

where

$$X_{cs} = \begin{bmatrix} x_p \\ x_c \end{bmatrix} \quad Y_{cs} = \begin{bmatrix} y \\ u \end{bmatrix} \quad U_{cs} = \begin{bmatrix} u_{nl} \\ d \end{bmatrix}$$
$$A_{cs} = \begin{bmatrix} A_p & 0 \\ C_p & 0 \end{bmatrix} \quad B_{cs} = \begin{bmatrix} B_p & B_p \\ 0 & 0 \end{bmatrix} \quad C_{cs} = \begin{bmatrix} C_p & 0 \\ -k_p C_p & -k_i \end{bmatrix}.$$

The previous expressions have been obtained with the assumption that the setpoint is null ($r = 0$). Proposition 1 shows that there is no loss of generality.

Proposition 1. *The setpoint can be assumed to be null without loss of generality.*

Proof. The proof is in the appendix. □

3. Definition of the event-based systems as PLS

The results obtained in this section do not depend on where the sampler is placed. Therefore, the subindexes representing the position of the sampler, *ps* and *cs*, have been dropped to simplify the notation. For example, A is equivalent to A_{ps}, if the process variable is being sample, or to A_{cs} in the other case.

Let us consider that the input to $G(s)$ is delayed in time by τ and the loop is closed by adding the non-linearity represented by the SOD sampler. Then,

$$G(s) \sim \begin{cases} \dot{X}(t) = AX(t) + BU(t - \tau) \\ Y(t) = CX(t) + DU(t - \tau) \end{cases}. \tag{7}$$

Now, the resulting system is an infinite dimensional system because of the time delay. However, it can be simplified because closing the loop with the non-linearity makes the input signal piecewise constant. So, in the matrix U the input $u_{nl}(t)$ is redefined as

$$u_{nl_k} = (j_k + \alpha)\delta,$$

where $j_k \in \mathbb{Z}$ is the crossed level by the input signal $u_{nl}(t)$, and $\alpha \in [0, 1) \subset \mathbb{R}$ is the sampling offset which depends on the initial sample, $u_{nl}(t_0)$. Also, if limit cycles of period T with crossing times $t_i^* > \tau$ where $T = t_1^* + t_2^* + \ldots + t_{2n}^*$ are only considered, the input $\{u_{nl}(t), t_i^* - \tau < t \le t_i^*\}$ is constant and its value is given by the feedback. In this case, the state of the system at the crossing times $t_1^*, t_2^* \ldots, t_{2n}^*$ is uniquely given by $X_{ps}(t_i^*)$. Taking into account this consideration, the system can be considered as a PLS defined by

$$\begin{aligned} \dot{X}(t) &= AX(t) + B_i \\ Y(t) &= CX(t) + D_i \end{aligned}, \tag{8}$$

where $B_i = B\begin{bmatrix} u_{nl_i} & d \end{bmatrix}^T$ and $D_i = D\begin{bmatrix} u_{nl_i} & d \end{bmatrix}^T$, and $u_{nl_i} = (j_i + \alpha)\delta$ for $i = \{0, \pm 1, \ldots, \pm n\}$.

In this PLS the rule to switch between the linear systems is included in the definitions of the non-linearities. It must be noted

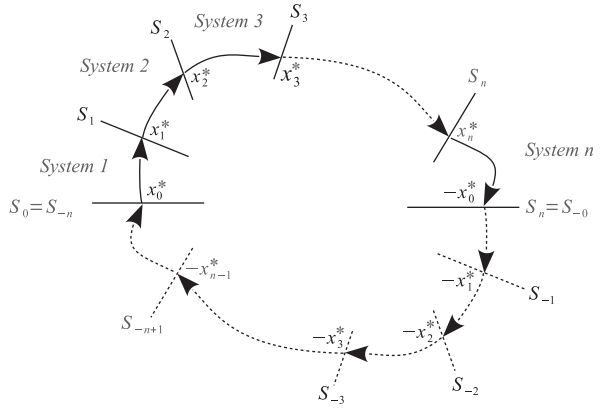


Fig. 3. Trajectory of a solution of the PLS system. In each region, the dynamics of the system is determined by an LTI system. The lines represents the switching surfaces.

that the switching rules have memory and the decision of which LTI to use may not depend only on the actual values of the state, but also on their past values. In the state-space, the points in which a rule provokes the switching from the system i to the system j define a surface which is known as *switching surface*. This surface consists of hyperplanes of dimension $m-1$, being m the order of the system, i.e. $X \in \mathbb{R}^m$,

$$S_i = \{X | CX - u_{nl_i} = 0\} \quad \text{for} \quad i = \{\pm 1, \ldots, \pm n\}$$

It is interesting to characterize the limit cycles that can appear in the system due to the effect of the non-linearities introduced with the event-based sampling scheme. The study of these limit cycles can be interesting for several reasons. There is a wide range of processes that will almost surely present limit cycles with the studied control schemes, while for others processes they can be prevented by carefully choosing the controller parameters. In any case, the limit cycles mean oscillations, which, depending on the process, may be more or less problematic. For example, a high frequency of oscillation may wear out the actuators. Also, the study of limit cycles can been used for identification purposes. For example, the relay auto tuning method is based on the properties of the limit cycle that appear in a process subject to relay feedback (which can be considered as a particular case of the studied event-based scheme). In addition, for the cases when the limit cycles cannot be prevented, it may be important to know about the stability of these limit cycles.

In order to calculate the period and amplitude, first it is assumed that the limit cycle contains $n+1$ levels, and thus it is composed of $2n$ switchings, where the first n correspond to the positive level crossings, and the rest n are the negative ones. Next, it is presented the set of equations that allows us to find the switching times and the values of the states at these switching times.

Proposition 2. *Consider the PLS in (8), with a non-linearity defined by the switching surfaces $S_i = \{X | CX - (j_i + \alpha)\delta = 0\}$, where $\alpha \in [0, 1)$, $j_i \in \mathbb{Z}$ and $0 < \delta \in \mathbb{R}$. Assume that there exists a symmetric periodic solution γ with $2n$ switching surfaces per period $T = t_1^* + t_2^* + \ldots + t_{2n}^*$, where t_1^*, t_2^*,...t_{2n}^* are the switching times when the switching surfaces S_1,..., S_n, S_{-n}.. S_{-1} are crossed, respectively (Fig. 3). Define*

$$f_k(t_1^*, \ldots, t_{2n}^*) = C(I + e^{AT})^{-1} \left[\sum_{i=1}^{2n-1} \Phi_{2n-1} \cdots \Phi_{i+1} (\Phi_i - I) \Lambda_i \right] - E_k \tag{9}$$

where $\Phi_i = \Phi(t_i) = e^{At_i}$, and $\Lambda_i = A^{-1}B_i$. Then the following conditions hold

$$\begin{cases} f_1(t_1^*, t_2^*, \ldots, t_{2n}^*) = 0 \\ f_2(t_1^*, t_2^*, \ldots, t_{2n}^*) = 0 \\ \vdots \\ f_{2n}(t_1^*, t_2^*, \ldots, t_{2n}^*) = 0 \end{cases} \tag{10}$$

and

$$\begin{cases} E_i \le CX_i(t) < E_{i+1} & \text{for} \quad 0 \le t < t_i^* \quad i = 1 \ldots n-1 \\ CX_n(t) \ge E_{n+1} & \text{for} \quad 0 \le t < t_n^* \\ E_i \ge CX_i(t) > E_{i+1} & \text{for} \quad 0 \le t < t_i^* \quad i = n+1 \ldots 2n \end{cases} \tag{11}$$

where

$$X_i(t) = e^{At} X_{i-1}^* - A^{-1}(e^{At} - I)B_i \quad \text{and} \quad E_i = (j_i + \alpha)\delta.$$

Furthermore, the limit cycle can be obtained with the initial condition

$$X_0^* = (I + e^{AT})^{-1} \left[\sum\nolimits_{i=1}^{2n-1} \Phi_{2n-1} \cdots \Phi_{i+1} (\Phi_i - I) \Lambda_i \right]. \tag{12}$$

Proof. The proof is in the appendix. □

However, if the system matrix is assumed to be singular, neither the state nor the integrals Γ_0 and Γ_1 can be computed explicitly. In this case the system of equations is obtained in the same way, but the computations are more involved. First the functions in (10) are redefined as,

$$f_i(X_i, t_1, \ldots, t_{2n}) = [I - \Phi_{2n} \ldots \Phi_1] X_i - \sum_{i=1}^{2n-1} \Phi_{2n-1} \ldots \Phi_{i+1} [\Gamma_1(t_i) U_{i-1} + \Gamma_0(t_i) U_i] \tag{13}$$

where, in contrast to the previous case, X_i appear as unknowns.

Note that the system of equations given by the functions f_i is composed of n^2 scalar equations and n^2+n unknowns. Thus, in order to solve it, the system must be completed with n additional equations. These equations are obtained from the switching conditions in the sampler, which fix the values of either the process output or the control input at the event times, depending on where the sampling is placed. Then, the complete set of equations that must be solved to obtain the features of the limit cycle is,

$$\begin{cases} f_1(X_1^*, t_1^*, \ldots, t_{2n}^*) = 0 \\ \vdots \\ f_{2n}(X_{2n}^*, t_1^*, \ldots, t_{2n}^*) = 0 \\ CX_1^* - d_1 = 0 \\ \vdots \\ CX_{2n}^* - d_{2n} = 0 \end{cases}. \tag{14}$$

How to use this result to analyze the limit cycles is demonstrated with examples in Sections 5–6.

4. Local stability

The local stability of the limit cycles described in the previous paragraph can be analized by observing the system at the switching times. The following result, which is a generalization of the approaches in [25,24,27], can be applied to study the behaviour of the trajectories of the system in the proximities of the limit cycles.

Proposition 3. *Assume that there exists a limit cycle γ with k states. The limit cycle is locally stable if and only if $W = W_k W_{k-1} \dots W_1$ has all its eigenvalues inside the unity disk, where $W_i = (I - V_i C_i / C_i V_i) e^{A t_i^*}$, $V_i = A X_i^* + B_i$, t_i^* are the switching times and X_i^* the state at the switching times.*

Proof. The proof is in the appendix. □

This result is used to prove local stability in Section 6.

5. Analysis of the limit cycles

In the following sections the expressions which take into account the effects of the event-based sampling at the output of the process and controller are presented. The method used black is based on the grouping of all the continuous dynamics into one block to obtain the state-space matrix (as shown in Fig. 1), and then to study the effect of the non-linear feedback introduced by the sampling block.

To easily distinguish between the different types of controller and sampling, the following naming convention is used: *controller*-SOD_n-*process*, when the sampler is after the controller output, and *controller-process*-SOD_n if the sampler is placed after the process output, where *process* corresponds to the type of process considered (IPTD, SOPDT,. . .) and *controller* refers to the type of controller (PI, PD, PID,. . .) The index *n* refers to the number of hysteresis bands presented in the sampler. For example, PI-SOD_1-IPTD denotes the system composed by an IPTD process controlled by a PI controller with the sampler placed after the controller output, and a limit cycle with one hysteris band (i.e. a system with relay feedback).

There are two directions in which the complexity of the analysis can be increased. The first one is to consider that the order of the process is increasing, i.e. a simple integrator, a double integrator, etc., and the second one is to consider an increase in the number of hysteresis bands of the sampler.

To simplify the analysis, it is worth noting that the solutions of (14) are linear on δ, thus the state and control signal can be normalized dividing by δ (note that $\delta > 0$).

The rest of this section presents an algorithm to compute the period and amplitude of a limit cycle in a generic process, and then shows examples of application to several common processes.

5.1. Algorithm

In this section an algorithm to obtain computationally the period of a limit cycle and the intermediate switching times is outlined. The algorithm can be implemented either in a symbolic or in a numerical computation tool.

1. Set *n* as the number of levels crossed within the limit cycle.
2. Fix the values of k_p, k_i, α, δ, d, τ, and the matrices A and B.
3. Calculate $\Phi(t) = e^{At}$, $\Gamma_0(t) = \int_0^{t-\tau} e^{As} ds$ and $\Gamma_1(t) = \int_{t-\tau}^{t} e^{As} ds$.

• To calculate the period:

1. For i from 1 to $2n$ repeat steps 2–3.
2. If $i \in (1, n)$, set $j_i = i - \lfloor \frac{n}{2} \rfloor$, else $j_i = \lfloor \frac{n}{2} \rfloor + n - i$.
3. Set $u_{nl_i} := (j_i + \alpha)\delta$, and $x_{c_i} = -\frac{u_{nl_i} + k_p x_{p_i}}{k_i}$ (controller) or $x_{p_i} = u_{nl_i}$ (process).
4. Set $eq_i := -X_{i+1} + \Phi(t_i) X_i + \Gamma_1(t_i) U_j + \Gamma_0(t_i) U_i$.
5. Solve the system of equations given by eq_i, with the unknowns t_i, and x_{p_i} or x_{c_i}.
6. $T = \sum_{i=1}^{2n} t_i$.

• To calculate the amplitude:

1. Set $j_{max} = j | (X_j > X_i, \forall\, i \neq j)$ and $j_{min} = j | (X_j < X_i, \forall\, i \neq j)$.
2. Find $t_{min} = \min(\tau, t | C\dot{x}_{j_{min}}(t) = 0)$, and $t_{max} = \min(\tau, t | C\dot{x}_{j_{max}}(t) = 0)$, corresponding to the minimum and maximum values of the output.
3. Compute the amplitude of the process output, $\Delta_p = C_p[X(t_{max}) - X(t_{min})]$, and the control input, $\Delta_c = C_c[X(t_{max}) - X(t_{min})]$.

5.2. Examples of analysis

5.2.1. IPTD process

Let us consider an IPTD process $P(s) = (k/s)e^{-\tau s}$, which can be described by its state-space model in the canonical observable form as,

$$\begin{aligned} \dot{x}(t) &= -ku(t-\tau) + kd(t) \\ y(t) &= x(t) = x_p(t) \end{aligned}, \tag{15}$$

where k is the process gain, x is the state, u the process input, and y is the process output.

First it is presented the approach for the PI-lPTD-SOD_n structure, and then for the PI-SOD_n-IPTD.

5.2.1.1. Process variable sampling (PI-IPTD-SOD_1). According to (4), the state feedback matrix corresponding to the system controlled by a PI with SOD sampling at the process output is,

$$\begin{aligned} \begin{bmatrix} \dot{x}_p(t) \\ \dot{x}_c(t) \end{bmatrix} &= \begin{bmatrix} 0 & kk_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} + \begin{bmatrix} kk_p & k \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_{nl_k} \\ d \end{bmatrix} \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} \end{aligned}. \tag{16}$$

Assuming there exists a stable limit cycle with two states, then the equations that allow us to obtain the amplitude and period of the oscillations are,

$$\begin{aligned} &\Phi(t_1) X_1 + \Gamma_1(t_1) U_2 + \Gamma_0(t_1) U_1 - X_2 = 0 \\ &\Phi(t_2) X_2 + \Gamma_1(t_2) U_1 + \Gamma_0(t_2) U_2 - X_1 = 0 \\ &x_{p_1} = -u_{nl_1} = \alpha\delta \\ &x_{p_2} = -u_{nl_2} = (\alpha - 1)\delta \end{aligned}, \tag{17}$$

where $X_i = [x_{p_i} \quad x_{c_i}]^T$ and $U_i = [u_{nl_i} \quad d]^T$ correspond to the state and input of the system at the switching times.

The matrices Φ, Γ_0, and Γ_1 can be calculated as,

$$\begin{aligned} \Phi(t) &= \begin{bmatrix} 1 & kk_i t \\ 0 & 1 \end{bmatrix} \\ \Gamma_0(t) &= \begin{bmatrix} kk_p t - kk_p \tau + \frac{1}{2} kk_i t^2 - kk_i t\tau + \frac{1}{2} kk_i \tau^2 & k(t-\tau) \\ t - \tau & 0 \end{bmatrix} \\ \Gamma_1(t) &= \begin{bmatrix} kk_p \tau + kk_i t\tau - \frac{1}{2} kk_i \tau^2 & k\tau \\ \tau & 0 \end{bmatrix} \end{aligned}. \tag{18}$$

Introducing (18) in (17), and solving the resulting equations, the period of the limit cycle can be obtained (see Table 1). Looking at the expression of the period, it can be seen that the symmetry of the limit cycle, i.e. the difference between the two semiperiods t_1 and t_2 depends on the offset of the sampler α. In particular, for $\alpha = 0.5$ the two semiperiods have the same value. When $\alpha = 0$, t_2 vanishes, which can be interpreted as this limit cycle cannot exist. In this case, either the system will reach a steady-state or enter into a limit cycle with higher number of levels. With respect to the disturbance rejection, it can be seen that d does not affect the period. This is

Table 1
Summary table with the limit cycle periods and amplitudes.

$P(s)$	PI-IPTD-SOD$_1$	PI-SOD$_1$-IPTD
$\frac{k}{s}e^{-\tau s}$	$T = \frac{1}{2}\frac{kk_i\tau^2 - 2kk_p\tau - 2}{k(k_p - k_i\tau)(\alpha-1)\alpha}$ $t_1 = (1-\alpha)T,\ t_2 = \alpha T$ $\Delta_{po} = \frac{\delta}{2}\frac{kk_i\tau^2 - 2kk_p\tau - 2}{k_p - k_i\tau}$ $\Delta_{co} = \frac{k_p^2 + k_i}{k}\Delta_{po}$	$T = \frac{1}{2}\frac{kk_i\tau^2 - 2kk_p\tau - 2}{k(k_p - k_i\tau)(\alpha - 1 + \frac{d}{\delta})(\alpha + \frac{d}{\delta})}$ $t_1 = (1-\alpha-\frac{d}{\delta})T,\quad t_2 = (\alpha+\frac{d}{\delta})T$ $\Delta_{po} = \frac{\delta}{2}\frac{kk_i\tau^2 - 2kk_p\tau - 2}{k_p - k_i\tau}$ $\Delta_{co} = \delta$
$\frac{k}{s^2}e^{-\tau s}$	Limit cycle does not exist	
$\frac{ke^{-\tau s}}{\tau_1 s+1}$	T, Δ_{po}, and Δ_{co} can be obtained using numerical methods	
$\frac{ke^{-\tau s}}{(\tau_1 s+1)(\tau_2 s+1)}$		

because it is rejected by the integrator, which changes its mean value to absorb the disturbance.

With the switching times t_1 and t_2, the amplitudes of the oscillations can be computed. It is necessary to find the maximum and minimum of the output, which correspond to the times when its first derivative is null, i.e. $\dot{x}_p(t) = kk_ix_c(t) + kk_pu_{nl_k} + kd = 0$. By solving the previous expression, the value of x_c and the times when the peaks are reached can be obtained, and so for this case it can be found an analytic expression for the amplitude of the process output, A_{po}, and of the control input, A_{ci} (See Table 1).

5.2.1.2. Controller variable sampling (PI-SOD$_1$-IPTD). When the sampler is placed at the controller output, the description of the system in the state-space is given by expressions (6). Thus, for this process, it is obtained

$$\begin{bmatrix}\dot{x}_p(t)\\ \dot{x}_c(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_p(t)\\ x_c(t)\end{bmatrix} + \begin{bmatrix}0 & 0\\ k & k\end{bmatrix}\begin{bmatrix}u_{nl_k}\\ d\end{bmatrix}$$
$$y(t) = \begin{bmatrix}k_p & k_i\end{bmatrix}\begin{bmatrix}x_p(t)\\ x_c(t)\end{bmatrix} \tag{19}$$

Assuming there exists a stable limit cycle with two states, then the equations that allow to obtain the amplitude and period of the oscillations are

$$\begin{aligned}&\Phi(t_1)X_1 + \Gamma_1(t_1)U_2 + \Gamma_0(t_1)U_1 - X_2 = 0\\ &\Phi(t_2)X_2 + \Gamma_1(t_2)U_1 + \Gamma_0(t_2)U_2 - X_1 = 0\\ &k_px_{p_1} + k_ix_{c_1} = u_{nl_1} = \alpha\delta\\ &k_px_{p_2} + k_ix_{c_2} = u_{nl_2} = (\alpha-1)\delta\end{aligned} \tag{20}$$

Here, the amplitude of the limit cycle can be computed directly, since the control input is piecewise constant and the switching times and values are known.

The period of the limit cycle can be obtained by solving the system of Eq. (20) (see Table 1). As opposed to the process sampling, here the disturbance appears in the expression of the period and semiperiods, thus affecting to the simmetry of the limit cycle. It is possible to have oscillations where the process is changing slowly nearly all the time and then to have an abrupt change. Since in one semiperiod the control action is more agressive, this decreases the margin of delay that can be added to the system without reaching the next sampling level.

To obtain the expression corresponding to the amplitude, and since the maximum and minimum values of the process output are reached at times $t_1 + \tau$ and $t_2 + \tau$, then integrating (19) yields the desired result (see Table 1).

5.2.2. DIPTD process

It is well known in classic control theory that the double integrator process cannot be stabilized with a continuous PI controller, due to the 90° phase lag of each integrator. It becomes then necessary to introduce the derivative action (PD controller) to compensate this lag. In the same way, either with the PI-SOD$_n$-DIPTD and PI-SOD$_n$-DIPTD the system will oscillate with unbounded growing amplitudes. Though it is not in the scope of this work, it should be possible to stabilize this kind of process by a SOD-PD controller. In practice, the implementation of the derivative action must be carefully studied, because it can be problematic specially when the sampler is placed after the process output, since the estimation of the derivative may be poor.

5.2.3. FOPTD process

The process considered in this section is a FOPTD process $P(s) = (k)/(Ts+1)e^{-\tau s}$, which is described in the state-space by the following expressions,

$$\begin{aligned}&\dot{x}(t) = -\frac{1}{T}x(t) + \frac{k}{T}u(t-\tau) + d(t)\\ &y(t) = x(t) = x_p(t)\end{aligned}, \tag{21}$$

where k is the process gain, x is the state, u the process input, y is the process output, and d is an external disturbance.

5.2.3.1. Process variable sampling (PI-FOPTD-SOD$_1$). The expressions corresponding to the PI-FOPTD-SOD$_1$ are as follows,

$$\begin{bmatrix}\dot{x}_p(t)\\ \dot{x}_c(t)\end{bmatrix} = \begin{bmatrix}\frac{1}{T} & \frac{kk_i}{T}\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_p(t)\\ x_c(t)\end{bmatrix} + \begin{bmatrix}\frac{kk_p}{T} & \frac{k}{T}\\ 1 & 0\end{bmatrix}\begin{bmatrix}u_{nl_k}\\ d\end{bmatrix}$$
$$y(t) = \begin{bmatrix}k_p & k_i\end{bmatrix}\begin{bmatrix}x_p(t)\\ x_c(t)\end{bmatrix} \tag{22}$$

From (22) the matrices Φ, Γ_0, and Γ_1 can be obtained as,

$$\begin{aligned}\Phi(t) &= \begin{bmatrix}e^{\frac{t}{T}} & kk_i(e^{\frac{t}{T}} - 1)\\ 0 & 1\end{bmatrix}\\ \Gamma_0(t) &= \begin{bmatrix}-k(k_p + k_iT)(e^{\frac{t-\tau}{T}} - 1) + kk_i(t-\tau) & T(e^{\frac{t-\tau}{T}} - 1)\\ t-\tau & 0\end{bmatrix}\\ \Gamma_1(t) &= \begin{bmatrix}k(k_p + k_iT)(e^{\frac{t-\tau}{T}} - e^{\frac{t}{T}}) - kk_i\tau & -T(e^{\frac{t-\tau}{T}} - e^{\frac{t}{T}})\\ \tau & 0\end{bmatrix}\end{aligned} \tag{23}$$

As it can be seen in the previous expressions, since the system of equations obtained for the FOPTD contains terms involving exponentials it is not possible to find analytical solutions, as opposed to the case of IPTD processes. Instead of that, the solutions have to be found numerically. However, the algorithm proposed in Section 5.1 can be applied.

5.2.3.2. Control variable sampling (PI-SOD$_1$-FOPTD). The expressions corresponding to the PI-SOD$_1$-FOPTD are the following ones,

$$\begin{bmatrix}\dot{x}_p(t)\\ \dot{x}_c(t)\end{bmatrix} = \begin{bmatrix}-\frac{1}{T} & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_p(t)\\ x_c(t)\end{bmatrix} + \begin{bmatrix}\frac{k}{T} & \frac{k}{T}\\ 1 & 0\end{bmatrix}\begin{bmatrix}u_{nl_k}\\ d\end{bmatrix}$$
$$y(t) = \begin{bmatrix}k_p & k_i\end{bmatrix}\begin{bmatrix}x_p(t)\\ x_c(t)\end{bmatrix}$$
$$y_p(t) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_p(t)\\ x_c(t)\end{bmatrix} \tag{24}$$

As in the previous case, the solutions of the equations must be found by means of numerical tools.

5.2.4. SOPTD process

The process considered in this section is a SOPTD $P(s) = (k/(\tau_1 s + 1)(\tau_2 s + 1))e^{-\tau s}$, which is described in the state-space by the following expressions,

$$\begin{aligned} \ddot{x}(t) &= -\frac{1}{\tau_1\tau_2}x - \frac{\tau_1+\tau_2}{\tau_1\tau_2}\dot{x} + \frac{k}{\tau_1\tau_2}u(t-\tau) + d(t) \\ y(t) &= x(t) = x_p(t) \end{aligned}, \tag{25}$$

where k is the process gain, τ_1 and τ_2 the time constants, x is the state, u the process input, y is the process output, and d is an external disturbance.

5.2.4.1. Process variable sampling (PI-SOPTD-SOD$_n$). The expressions corresponding to the PI-SOPTD-SOD$_n$ are the following ones,

$$\begin{bmatrix} \dot{x}_p(t) \\ \ddot{x}_p(t) \\ \dot{x}_c(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{\tau_1\tau_2} & -\frac{\tau_1+\tau_2}{\tau_1\tau_2} & \frac{kk_i}{\tau_1\tau_2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_p(t) \\ \dot{x}_p(t) \\ x_c(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{kk_p}{\tau_1\tau_2} & \frac{k}{\tau_1\tau_2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_{nl_k} \\ d \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \\ k_p & 0 & k_i \end{bmatrix} \begin{bmatrix} x_p(t) \\ \dot{x}_p(t) \\ x_c(t) \end{bmatrix} \tag{26}$$

As in the FOPTD case, for this system it is not possible to find analytical solutions due to the exponentials that appear in the equations. Therefore, numerical methods must be used to find the solutions.

5.2.4.2. Controller variable sampling (PI-SOD$_n$-SOPTD). The expressions corresponding to the PI-SOD$_n$-SOPTD are the following ones,

$$\begin{bmatrix} \dot{x}_p(t) \\ \ddot{x}_p(t) \\ \dot{x}_c(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{\tau_1\tau_2} & -\frac{\tau_1+\tau_2}{\tau_1\tau_2} & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_p(t) \\ \dot{x}_p(t) \\ x_c(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k}{\tau_1\tau_2} & \frac{k}{\tau_1\tau_2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_{nl_k} \\ d \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \\ k_p & 0 & k_i \end{bmatrix} \begin{bmatrix} x_p(t) \\ \dot{x}_p(t) \\ x_c(t) \end{bmatrix} \tag{27}$$

As in the previous case, the solutions of the equations must be found by means of numerical tools.

6. Simulation results

This section shows simulations which illustrate the behaviour of the different combinations of control schemes and processes commented in the previous sections.

6.1. PI-IPTD-SOD$_n$ and PI-SOD$_n$-IPTD

Let us consider an IPTD process controlled by a PI controller with event-based sampling, and let the values of the plant parameters be $k = 1.0$, $\tau = 0.2$. The controller gains $k_p = 1.2$, $k_i = 1.0$ have been chosen to produce a limit cycle with two states, and the sampler $\alpha = 0.5$, $\delta = 0.1$. The system is described by the following expressions,

$$\begin{bmatrix} \dot{x}_p(t_k) \\ \dot{x}_c(t_k) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1.0 & 0 \end{bmatrix} \begin{bmatrix} x_p(t_k) \\ x_c(t_k) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1.2 & 1 \end{bmatrix} \begin{bmatrix} u_{nl_k} \\ d \end{bmatrix}. \tag{28}$$

There exists a symmetric limit cycle with two states (see Fig. 4a), with period $T = 5.6093$ and amplitude $A = 0.2095$, which have been computed with the Algorithm presented in Section 5.1. With the chosen gains, the system converges to the limit cycle after introducing a step change in the set-point.

When the sampler is placed at the controller output, the limit cycle that appears (see Fig. 4c) has the same period $T = 5.6093$ but different amplitude $A = 0.1402$. While in the first case an external disturbance does not vary the properties of the limit cycle, for the second case it does as it is shown in Fig. 5.

Varying the parameters of the controller it is possible to obtain limit cycles with higher number of levels. For example, increasing the integral gain of the controller to $k_i = 2.0$, the system with the sampler at the process output presents the response shown in Fig. 4b and with the sampler at the controller output it has the response of Fig. 4d. The period and amplitude of the oscillation computed are $T = 4.9745$ and $A = 0.3584$, which correspond to the results obtained in the simulation.

The simulations show that the system, with the chosen parameters, converges to a limit cycle even in presence of constant disturbances. The local stability of the limit cycles can also be proven by applying Proposition 3. As an example, for the two-level limit cycle, looking at the eigenvalues of the matrix $W = W_2W_1$, where

$$W_i = \begin{bmatrix} 1 - u_{nl_i} & 0 \\ -x_{pi} + 1.2(u_{nl_i} + d) + t_i^* & 1 \end{bmatrix}.$$

After straightforward computations, it can be shown that the eigenvalues of W are $\lambda_1 = 1$, $\lambda_2 = (1-\alpha)\alpha < 1$, and therefore the limit cycle is locally stable. If limit cycles with $2n$ switchings are considered, then the eigenvalues of W are $\lambda_1 = 1$, $\lambda_2 = \prod_{i=0}^{2n-1}(1 - \alpha - n + i)$. It is easy to verify that $|\lambda_2| \leq 1$ for every α when $n = 1, 2$, i.e. the corresponding limit cycles are locally stable. However, for $n > 2$ the local stability of the limit cycles depends on the particular value of α.

6.2. PI-SOPTD-SOD$_n$

Now, consider a SOPTD with parameters $k = 1.0$ (gain), $\tau_1 = 1.0$, $\tau_2 = 0.5$ (time constants), and $\tau = 0.2$ (time delay), which is controlled by a PI with event-based sampling placed at the process

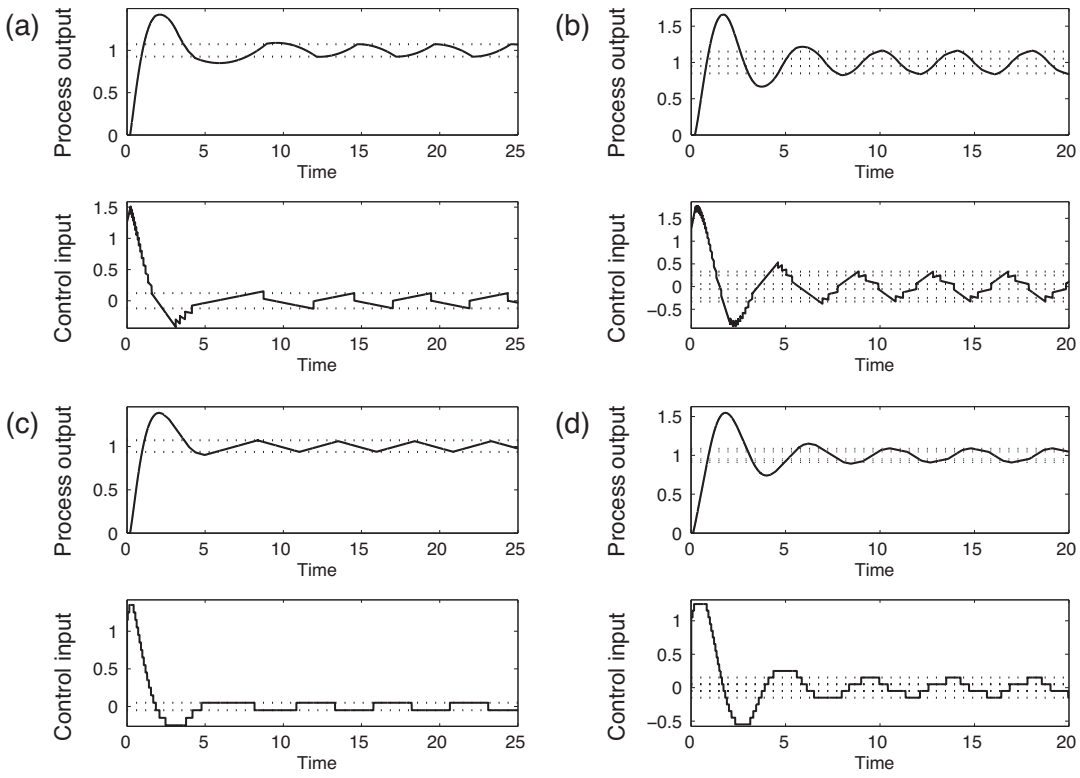


Fig. 4. Limit cycles in an IPTD process controlled by a PI with event-based sampling at the process output with one hysteresis band (a) and two (b), and with sampling at the controller output with one hysteresis band (c) and two (d). The dotted lines show the sampling levels of the process variable in (a), (c) and of the control variable in (b), (d), and the value at the switching times of the control variable in (a), (c) and of the process variable in (b), (d).

output, with $\alpha = 0.5$ and $\delta = 0.1$. Setting the controller gains to $k_p = 1.2$ and $k_i = 1.0$, the matrices A and B of the system are,

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 2 & -2 & -3 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 2.4 & 2 \end{bmatrix}. \tag{29}$$

Solving the system of equations corresponding to the limit cycle with one band of hysteresis, the values of the switching times and levels are,

$$t_1 \approx 3.0772, \qquad t_2 \approx 3.0772,$$
$$\dot{x}_{p_1} \approx -0.0517, \qquad \dot{x}_{p_2} \approx 0.0517,$$
$$\dot{x}_{c_1} \approx -0.0669, \qquad \dot{x}_{c_2} \approx 0.0669.$$

Finally, the period is $T = t_1 + t_2 \approx 6.1545$ and the amplitude is $A \approx 0.1338$.

Fig. 6.

It is interesting to note that, when the sampler is placed at the process output, the limit cycle does not vary when a constant disturbance affects the system, because it is rejected by the controller.

The local stability of the limit cycle can be studied by looking at the eigenvalues of the matrix W,

$$W = W_2 W_1 = \begin{bmatrix} 0 & 0 & 0 \\ -0.0787 - 2.5953d & 0.0042 & 0.0021 \\ 0.0699 + 2.4253d & -0.0042 & -0.0021 \end{bmatrix}. \tag{30}$$

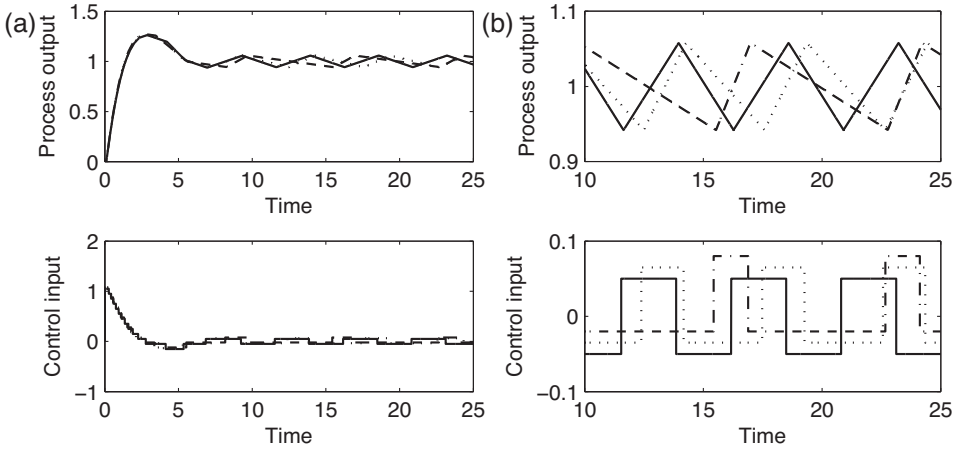


Fig. 5. Limit cycles in an IPTD process controlled by a PI with event-based sampling at the controller output (a) with an external disturbance $d = 0.0$ (solid line), $d = 0.015$ (dotted line) and $d = .03$ (dashed-dotted line). (b) Detail of the limit cycles.

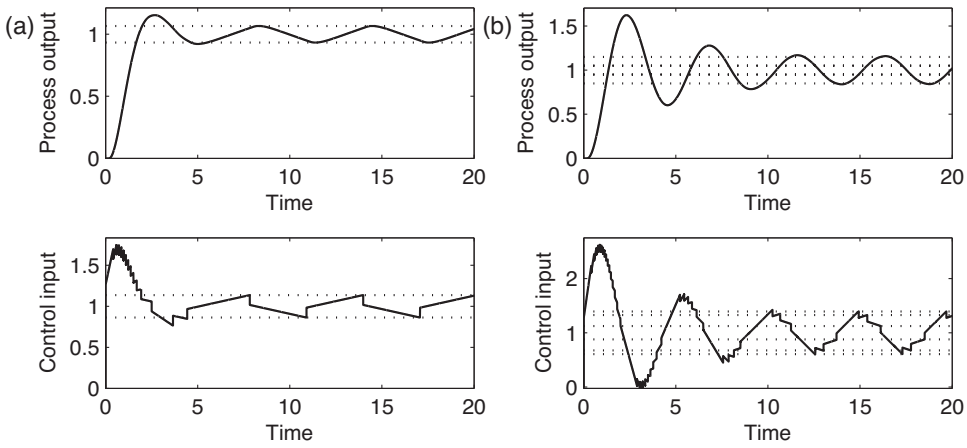


Fig. 6. Limit cycles in an SOPTD process controlled by a PI with event-based sampling at the controller output, involving (a) one hysteresis band and (b) two.

Since the eigenvalues of the W are inside the unit disk ($\lambda_1 \approx 0.0021, \lambda_1 \approx 0.0000, \lambda_1 \approx 0.0000$), the limit cycle is locally stable.

7. Experimental results

This section shows experimental results which clearly evidence the existence of the results derived in the previous sections in a real system. Therefore the experiences carried out were focused on finding limit cycles in the Acurex system to compare them with that predicted by the theory and simulations. As shown in the following paragraphs, it is worth noting that even when the model used to represent the system is a simplification of the process which ignores many of the complex dynamics existent in the system, the results are very close to those predicted in theory. Below, the Acurex system and the model identified from experimental data are presented, commenting some implementation issues of the controller and, finally, the obtained results are shown and interpreted.

7.1. The Acurex system

The experiments have been done with an equipment, known as Acurex, built in 1981 at the Almeria Solar Platform (PSA, Spain [30]). In this plant (Fig. 7), two types of collecting systems were considered, a central receiver system (CRS) and a distributed collector system (DCS) using parabolic troughs. Parabolic trough systems concentrate sunlight onto a receiver pipe which contains a heat transfer fluid (HTF) that is heated as it flows along the receiver pipe. Then, the HTF is used to produce steam that may be used for example to feed an industrial process. A survey of basic and advanced control approaches for distributed solar collector fields can be found in [31] and [32]. For more information on control of solar plants see [33].

Fig. 7. Acurex distributed collector system at the Solar Platform of Almeria (PSA), Spain.

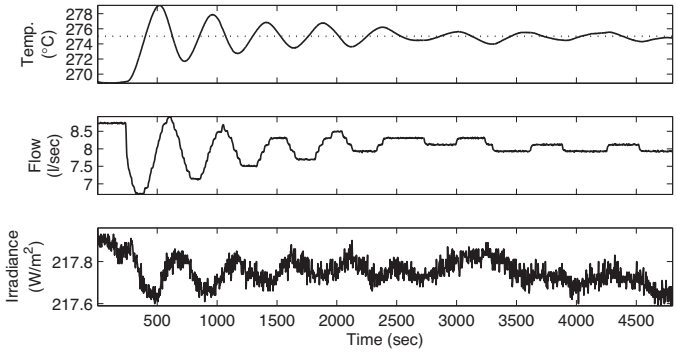


Fig. 8. Response of the Acurex system (solid line) to a step change in the setpoint, with the event-based sampler placed at the controller output.

7.2. The model

The plant was identified as a FOTPD, where the process input is the oil flow (l/s) in the pipes and the process output is the temperature of the oil at the collector field outlet (°C). There are unmodeled dynamics that are considered as external disturbances, such as the oil temperature at the input, or the solar irradiance. However, because of the time scale of the tests performed, which is smaller than the rate of variation of these variables (under clear day conditions without clouds), the model obtained seems to be a valid representation of the process for our purposes.

The transfer function was identified from experimental data obtained from the plant in a set of step tests. The procedure followed in each test is to drive the temperature manually to the working point, and when the process has reached it to introduce a step change in the input, registering the data measured from the sensors until the process stabilizes again. The FOPTD model, obtained by applying a least-squares procedure, is,

$$P(s) = -\frac{6.0715}{103.2723s + 1} e^{-67.5021s}. \quad (31)$$

where the time constant (τ_1) and time delay (τ) are given in seconds, and the gain k in $^\circ Cs/l$. The tests were carried out with an input around 8.5 l/s, being the range of the pump from 2 to 12 l/s. Also, the time constants and delays obtained make sense from the previous published works in this field. Notice the minus sign in (31), which represents the inverse response of the plant, i.e. a positive change in the flow produces a negative change in the temperature.

7.3. Implementation

The Acurex system has a SCADA software developed in LabVIEW. This software provides the user with an interface to execute its own controller implementation in MATLAB code. Thus, one must write a MATLAB callback function which is invoked with a configurable sampling period (it was fixed to $T_s = 15$ s). This function receives the measures from the sensors, updates the controller state and, finally, sends the new control action to the actuators.

The controller implementation can be configured to work in three modes, namely,

- *manual*, the control input is set manually,
- *SOD-PI*, the sampler is at the process output, and
- *PI-SOD*, the sampler is at the controller output.

A summary of the controller is listed, in Listing 1 and 2.

```
% Sampling at the process output
  e = setpoint - output;
  if(~((u_prev >= umax && e > 0) || (u_prev <= umin && e < 0)))
    I = I + e_prev*Ts;
  end

  % event detection
  if (abs(e - e_prev) > delta)
    levels = floor(abs(e - e_prev) / delta);
    e_prev = e_prev + sign(e - e_prev)*delta*levels;
    u_prev = sat(Kp*e_prev + Ki*I, umin, umax);
    I = (u_prev - Kp *e_prev) / Ki;
  end
```

Listing 1: Code of the controller with the sampler at the process output.

```
% Sampling at the controller output
  e = setpoint - output;

  % anti-windup
  if(~((u_prev >= umax && e > 0) || (u_prev <= umin && e < 0)))
    I = I + e*Ts;
    u = Kp*e + Ki*I;
  else
    u = u_prev;
  end

  % event detection
  if (abs(u - u_prev) > delta)
    levels = floor(abs(u - u_prev) / delta);
    u_prev = sat(u_prev + sign(u - u_prev)*delta, umin, umax);
  end
```

Listing 2: Code of the controller with the sampler at the controller output.

7.3.1. Controller sampling

The first set of tests presented were carried out with the controller in the *PI-SOD* mode, with the purpose of reproducing the limit cycles obtained in simulation composed of two levels (with $\alpha = 0.5$) and of three levels (with $\alpha = 0.0$). The procedure followed is the same for each test, first the system temperature is moved to an operating point, and next when the process reaches a steady state a set-point step change is introduced into the controller.

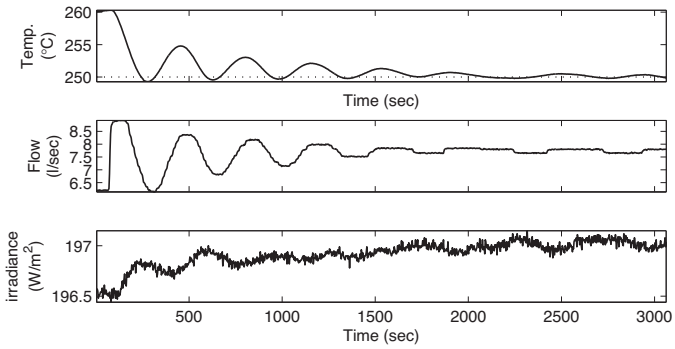


Fig. 9. Response of the Acurex system (solid line) to a step change in the setpoint, with the event-based sampler placed at the process output.

The response is shown in Fig. 8, where the oil temperature, the flow, and the solar irradiance during the test are plotted. It can be seen that the system goes into a limit cycle composed of two levels, which is similar to the results obtained in simulation with the FOPTD model.

Fig. 10a and b shows the comparison of the limit cycles obtained in simulation with the model and the results obtained with the Acurex system. The results are similar both qualitatively (limit cycles with two states), and quantitatively (the period and amplitude are approximately equal).

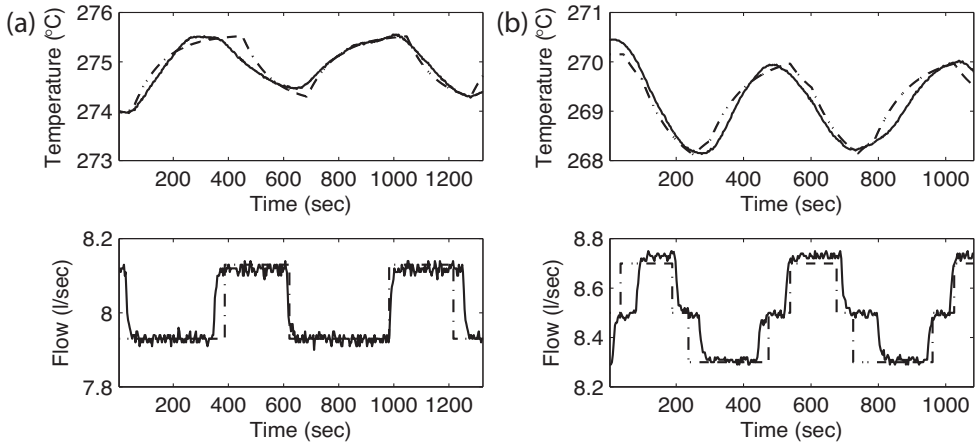


Fig. 10. Limit cycles in the Acurex system (solid line) and in the simulated model (dashed line) controlled by a PI with SOD sampling placed after the controller output. (a) with two levels and (b) with three levels.

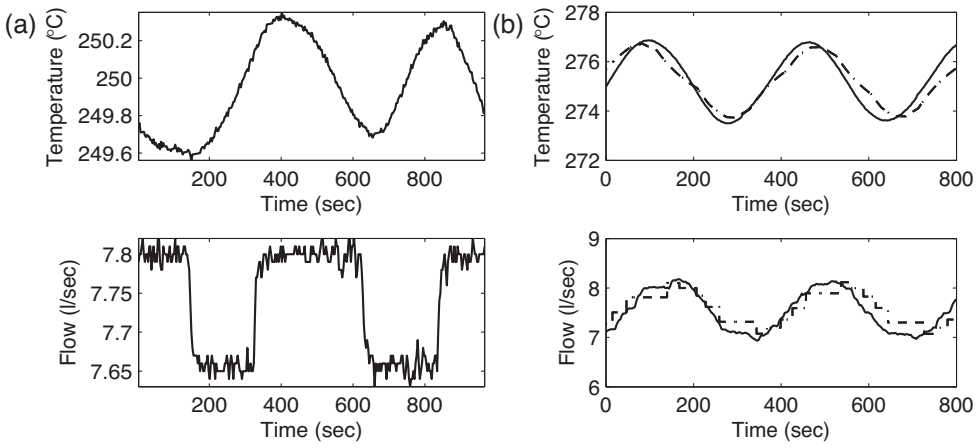


Fig. 11. Limit cycles in the Acurex system (solid line) and in the simulated model (dashed line) controlled by a PI with SOD sampling placed after the process output. (a) with two levels, and (b) with eight levels.

7.3.2. Process sampling

The second set of tests were carried out with the sampler placed at the process output. After verifying that, as in the previous section, the system enters into a limit cycle of two levels (Fig. 9), the existence of more complex limit cycles was investigated. Increasing the proportional gain, a limit cycle with eight different levels was found, which is shown in Fig. 11. It is remarkable that even in this case, the comparison between the experimental data and the simulated process shows that there are no significative differences in the behaviour.

8. Conclusions

In this work the behaviour of a control system based on the use of a level crossing sampling either in the process output or in the control output has been studied. Limit cycles are of particular interest since they are associated to oscillations in processes, and therefore it is worth to have knowledge about them in order to avoid them when possible or to assure that they are not problematic.

When trying to find properties about the limit cycles, it is common to have system of equations which involve transcendent function and thus it is not possible in general to find closed-form solutions. Moreover, due to the combinatorial explosion, it can be computationally expensive to find these solutions, and it becomes harder when higher order process model are considered.

Therefore, an algorithm to analyze the properties of the limit cycles has been proposed; it allows us to introduce some knowledge in the problem statement, so that the complexity can be reduced.

A set of simulation results illustrates the behaviour of the controllers with a set of processes models used very frequently in industrial context, which are the IPTD, the FOPTD, and the SOPTD. Also, this behaviour has been tested and verified in the Acurex Field of the Solar Platform of Almeria, Spain. The experiments performed confirm that the simulation results can be extrapolated to real cases, obviously with the divergences due to unmodeled dynamics of the process, disturbances, etc.

As future lines of research, the effort will be mainly focused on two issues, the study of the global stability of the controller, and the optimization of the algorithm proposed to find the properties of the limit cycles.

A simulation software tool have been developed to simulate the behaviour of the two control schemes proposed with all the process models analyzed in this work. This software can be freely downloaded from http://www.dia.uned.es/SOD-Simulator.jar.

Acknowledgements

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

Appendix A. Proofs of the Propositions

Proposition 1. Consider that the sampler is placed at the process output. Then, the state-space representation is as follows,

$$G_{cs}(s)\sim \begin{cases} \begin{bmatrix}\dot{x}_p\\ \dot{x}_c\end{bmatrix} = \begin{bmatrix}A_p & -k_iB_p\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_p\\ x_c\end{bmatrix} + \begin{bmatrix}-k_pB_p & k_pB_p & B_p\\ 1 & -1 & 0\end{bmatrix}\begin{bmatrix}u_{nl}\\ r\\ d\end{bmatrix} \\ \begin{bmatrix}y\\ u\end{bmatrix} = \begin{bmatrix}C_p & 0\\ 0 & -k_i\end{bmatrix}\begin{bmatrix}x_p\\ x_c\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\ -k_p & k_p & 0\end{bmatrix}\begin{bmatrix}u_{nl}\\ r\\ d\end{bmatrix} \end{cases} \quad (A.1)$$

Note that the system can be equivalently written as

$$G_{cs}(s)\sim \begin{cases} \begin{bmatrix}\dot{x}_p\\ \dot{x}_c\end{bmatrix} = \begin{bmatrix}A_p & -k_iB_p\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_p\\ x_c\end{bmatrix} + \begin{bmatrix}-k_pB_p & B_p\\ 1 & 0\end{bmatrix}\begin{bmatrix}\tilde{u}_{nl}\\ d\end{bmatrix} \\ \begin{bmatrix}y\\ u\end{bmatrix} = \begin{bmatrix}C_p & 0\\ 0 & -k_i\end{bmatrix}\begin{bmatrix}x_p\\ x_c\end{bmatrix} + \begin{bmatrix}0 & 0\\ -k_p & 0\end{bmatrix}\begin{bmatrix}\tilde{u}_{nl}\\ d\end{bmatrix} \end{cases}, \quad (A.2)$$

where $\tilde{u}_{nl} = u_{nl} - r$, and the switching surfaces are $S_i = \{CX - \tilde{\delta}_i\}$, where $\tilde{\delta}_i = \delta_i - r$.

Now, consider that the sampler is placed at the controller output.

$$G_{cs}(s)\sim \begin{cases} \begin{bmatrix}\dot{x}_p\\ \dot{x}_c\end{bmatrix} = \begin{bmatrix}A_p & 0\\ C_p & 0\end{bmatrix}\begin{bmatrix}x_p\\ x_c\end{bmatrix} + \begin{bmatrix}B_p & B_p & 0\\ 0 & 0 & -1\end{bmatrix}\begin{bmatrix}u_{nl}\\ d\\ r\end{bmatrix} \\ \begin{bmatrix}y\\ u\end{bmatrix} = \begin{bmatrix}C_p & 0\\ -k_pC_p & -k_i\end{bmatrix}\begin{bmatrix}x_p\\ x_c\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & k_p\end{bmatrix}\begin{bmatrix}u_{nl}\\ d\\ r\end{bmatrix} \end{cases}, \quad (A.3)$$

Note that system can be written equivalently as

$$G_{cs}(s)\sim \begin{cases} \begin{bmatrix}\dot{\tilde{x}}_p\\ \dot{x}_c\end{bmatrix} = \begin{bmatrix}A_p & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\tilde{x}_p\\ x_c\end{bmatrix} + \begin{bmatrix}-k_pB_p & B_p\\ 1 & 0\end{bmatrix}\begin{bmatrix}\tilde{u}_{nl}\\ \tilde{d}\end{bmatrix} \\ \begin{bmatrix}y\\ u\end{bmatrix} = \begin{bmatrix}C_p & 0\\ -k_pC_p & -k_i\end{bmatrix}\begin{bmatrix}\tilde{x}_p\\ x_c\end{bmatrix}, \end{cases} \quad (A.4)$$

where $\tilde{x}_p = x_p - (C_p^T/\|C_p\|^2)r$, $\tilde{y} = y + r$, and $\tilde{d} = d + (B_p^TA_pC_p^T)/(\|B_p\|^2\|C_p\|^2)r$, and the switching surfaces are $S_i = \{CX - \tilde{\delta}_i\}$, where $\tilde{\delta}_i = \delta_i - r$.

Proposition 2. Assuming that $t_i > \tau$, where t_i is the time elapsed between the crossing of the switching surfaces i and $i+1$, then the state is obtained by integrating (8) from $t=0$ to $t=t_i$. It gives

$$X_{i+1} = \Phi(t_i)X_i + \Gamma_1(t_i)U_{i-1} + \Gamma_0(t_i)U_i, \quad (A.5)$$

where $\Phi(t) = e^{At}$ is the state transition matrix, $\Gamma_0 = \int_0^{t-\tau}\Phi(s)ds$ accounts for the effect of the input, and $\Gamma_1 = \int_{t-\tau}^{t}\Phi(s)ds$ is a term which represents the effect of the input because of the time delay of the system.

Then, for a limit cycle involving n switching times, we have a system of equations described by

$$\begin{aligned} &\Phi(t_1)X_1 + \Gamma_1(t_1)U_n + \Gamma_0(t_1)U_1 - X_2 = 0\\ &\qquad\cdots\\ &\Phi(t_{n-1})X_{n-1} + \Gamma_1(t_{n-1})U_{n-2} + \Gamma_0(t_{n-1})U_{n-1} - X_n = 0\\ &\Phi(t_n)X_n + \Gamma_1(t_n)U_{n-1} + \Gamma_0(t_n)U_n - X_1 = 0 \end{aligned}$$

Substituting recursively, we get the following expression,

$$\begin{aligned}[I - \Phi(t_n)\dots\Phi(t_1)]X_n &= \Phi(t_n)\dots\Phi(t_2)(\Gamma_1(t_1)U_n + \Gamma_0(t_1)U_1)\\ &+ \Phi(t_n)\dots\Phi(t_3)(\Gamma_1(t_2)U_1 + \Gamma_0(t_2)U_2) + \dots + \Phi(t_n)(\Gamma_1(t_{n-1})U_{n-2}\\ &+ \Gamma_0(t_{n-1})U_{n-1}) + \Gamma_1(t_n)U_{n-1} + \Gamma_0(t_n)U_n. \end{aligned} \quad (A.6)$$

The previous expression can be written in a more compact form as,

$$[I - \Phi_n\dots\Phi_1]X_n = \sum_{i=1}^{2n-1}\Phi_{2n-1}\dots\Phi_{i+1}[\Gamma_1(t_i)U_{i-1} + \Gamma_0(t_i)U_i]. \quad (A.7)$$

If we assume that the system matrix A is nonsingular, then the integrals Γ_1 and Γ_0 can be explicitly computed and the state X_n can be solved, yielding the expression of the initial state (12). Since we know that at the switching times the state must be at the switching surface, we can combine the previous expression with the switching conditions to get the set of equations given by (10). Finally, the conditions (11) hold because the state does not cross the switching surface in the interval between two switchings.

Proposition 3. Consider a trajectory with initial condition $X(0) = X_0^*$. The solution is $X(t) = \Phi(t)X_0^* + \Gamma_1(t)u_{nl_{n-1}} + \Gamma_0(t)u_{nl_n}$. When X reaches the switching surface at time $t_1^* + \delta_1t_1^*$, we have,

$$\begin{aligned} X(t_1^* + \delta_1t_1^*) &= \Phi(t_1^* + \delta_1t_1^*)(X_0^* + \delta X_0^*) + \Gamma_1(t_1^*\\ &+ \delta_1t_1^*)u_{nl_{n-1}} + \Gamma_0(t_1^* + \delta_1t_1^*)u_{nl_n}. \end{aligned}$$

The series expansion in $\delta_1t_1^*$ and $\delta_1X_0^*$ is,

$$X(t_1^* + \delta_1t_1^*) = X_1^* + \nu_1\delta_1t_1^* + \Phi(t_1^*)\delta_1X_0^* + O(\delta_1^2)$$

where $\nu_1 = AX_1^* + B_1 = A[\Phi(t_1^*)X_0^* + \Gamma_1(t_1^*)u_{nl_{n-1}} + \Gamma_0(t_1^*)u_{nl_n}] + Bu_{nl_n}$.

Since the solution is on the switching surface at $t_1^* + \delta_1t_1^*$, we have

$$C_1X(t_1^* + \delta_1t_1^*) + d_1 = C_1X_1^* + C_1\nu_1\delta_1t_1^* + C_1\Phi(t_1^*)\delta_1X_0^* + d_1 = 0$$

and, therefore, the following equality holds,

$$\delta_1t_1^* = -\frac{C_1\Phi(t_1^*)}{C_1\nu_1}\delta_1X_0^*.$$

The rest of the proof follows as in [27]. The state after the first switch is

$$x(t_1^* + \delta_1t_1^*) = x_1^* + (I - \tfrac{\nu_1C_1}{C_1\nu_1})\Phi(t_1^*)\delta_1X_0 + O(\delta_1^2) = x_1^* + W_1\delta_1x_0^* + O(\delta_1^2). \quad (A.8)$$

Taking as initial condition $X_1^* + \delta_2X_1^*$ and neglecting the $O(\delta^2)$ term we have $X(t_2^* + \delta_2t_2^*) = X_2^* + W_2\delta_2X_1^*$. Combining with (A.8) yields $\delta_2X_1^* = W_1\delta_1X_0^*$. Replacing in the previous expression and applying successively for k eventually leads to

$$X(t_k^* + \delta_kt_k^*) = X_0^* + W_kW_{k-1}\dots W_1\delta_1X_0^* + O(\delta_1^2).$$

References

[1] K.J. Åström, B. Wittenmark, Computer-controlled Systems, Theory and Design, Prentice-Hall, 1996.
[2] W.H. Kwon, Y.H. Kim, S.J. Lee, K.-N. Paek, Event-based modeling and control for the burnthrough point in sintering processes, in: IEEE Transactions on Control Systems Technology, vol.7, 1999, pp. 31–41.
[3] T.-J. Tarn, N. Xi, A. Bejczy, Path-based approach to integrated planning and control for robotic systems, Automatica 32 (12) (1996) 1675–1687.
[4] E. Hendricks, A. Jensen, A. Chevalier, A. Vesterholm, Problems in event based engine control, in: American Control Conference, Baltimore, Maryland, USA, 1994, pp. 1585–1587.
[5] K.J. Åström, Event-based Control, Springer-Verlag, Berlin, 2008.
[6] K.J. Åström, B. Bernhardsson, Comparison of Riemann and Lebesgue sampling for first order stochastic systems, in: 41st IEEE conference on Decision and Control, Las Vegas, NV, 2002, pp. 2011–2016.
[7] J.P. Hespanha, P. Naghshtabrizi, Y. Xu, A survey of recent results in networked control systems, in: Proceedings of the IEEE, volume 95, 2007, pp. 138–162.
[8] O. Demir, J. Lunze, Event-based synchronisation of multi-agent systems, in: 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12), Eindhoven, The Netherlands, 2012.
[9] J. Weimer, J. Araújo, K.-H. Johansson, Distributed event-triggered estimation in networked systems, in: 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12), Eindhoven, The Netherlands, 2012.
[10] H. Yu, P.J. Antsaklis, Event-triggered output feedback control for networked control systems using passivity: time-varying network induced delays, in: 50th

IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, December 2011.
[11] P. Tabuada, Event-triggered real-time scheduling of stabilizing control tasks, IEEE Transactions on Automatic Control 52 (9) (2007) 1680–1685.
[12] J. Almeida, C. Silvestre, A.M. Pascoal, Self-triggered state feedback control of linear plants under bounded disturbances, in: 49th IEEE Conference on Decision and Control (CDC), Atlanta, Georgia, USA, December 2010, pp. 7588–7593.
[13] W.P.M.H. Heemels, M.C.F. Donkers, Model-based periodic event-triggered control for linear systems. Automatica, in press
[14] W.P.M.H. Heemels, M.C.F. Donkers, Periodic event-triggered control for linear systems. IEEE Transactions on Automatic Control, in press
[15] M. Mazo Jr., Ming Cao, Decentralized event-triggered control with one bit communications, in: 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12), Eindhoven, The Netherlands, 2012.
[16] D. Lehmann, J. Lunze, K.H. Johansson, Comparison between sampled-data control, deadband control and model-based event-triggered control, in: 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12), Eindhoven, The Netherlands, 2012.
[17] U. Tiberi, K.H. Johansson, A simple self-triggered sampler for nonlinear systems, in: 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12), Eindhoven, The Netherlands, 2012.
[18] A. Anta, P. Tabuada, Isochronous manifolds in self-triggered control, in: 48th IEEE Conference on Decision and Control, Shanghai, China, 2009, pp. 3194–3199.
[19] A. Anta, P. Tabuada, To sample or not to sample: self-triggered control for nonlinear systems, IEEE Transactions on Automatic Control 55 (9) (2010) 2030–2042.
[20] X. Wang, M. Lemmon, Self-triggered feedback systems with state-independent disturbances, in: American Control Conference, Riverfront, St. Louis, MO, USA, 2009.
[21] K.-E. Årzén, A simple event-based PID controller, in: 14th IFAC World Congress, Beijing, PR China, 1999.
[22] V. Vasyutinskyy, K. Kabitzsch, Implementation of PID controller with send-on-delta sampling, in: ICC'2006, International Conference on Control, Glasgow, Scotland, 2006.
[23] J. Sánchez, A. Visioli, S. Dormido, Event-based PID control, in: PID Control in the Third Millennium, Advances in Industrial Control, Springer London, 2012, pp. 495–526.
[24] K.J. Åström, Oscillations in systems with relay feedback, in: Adaptive Control, Filtering, and Signal Processing, IMA Volumes in Mathematics and its Applications, vol. 74, Springer-Verlag, 1995.
[25] A. Cervin, K.J. Åström, On limit cycles in event-based control system, in: 46th Conference on Decision and Control, pp. 3190-3195., December 2007.
[26] D. Lehmann, K.H. Johansson, Event-triggered PI control subject to actuator saturation, in: IFAC Conference on Advances in PID Control, Brescia, Italy, 2012.
[27] J.M. Gonçalves, Constructive Global Analysis of Hybrid Systems. Ph.D. thesis, Massachussetts Institute of Technology, url: http://hdl.handle.net/1721.1/8980, 2000.
[28] 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.
[29] M. Beschi, S. Dormido, J. Sánchez, A. Visioli, Tuning rules for event-based SSOD-PI controllers, in: 20th Mediterranean Conference on Control and Automation, Barcelona, 2012.
[30] F.A. Schraub, H. Dehne, Electric generation system design: management, startup and operation of IEA distributed collector solar system in Almería, Spain, Solar Energy 31 (4) (1983) 351–354.
[31] E.F. Camacho, F.R. Rubio, M. Berenguel, L. Valenzuela, A survey on control schemes for distributed solar collector fields. Part I: modeling and basic control approaches, Solar Energy 81 (10) (October 2007) 1240–1251.
[32] E.F. Camacho, F.R. Rubio, M. Berenguel, L. Valenzuela, A survey on control schemes for distributed solar collector fields. Part II: advanced control approaches, Solar Energy 81 (10) (October 2007) 1252–1272.
[33] E.F. Camacho, M. Berenguel, F.R. Rubio, D. Martínez, Control of Solar Energy Systems, Springer, 2012.