Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

An experimental framework to analyze limit cycles generated by event-based sampling

J. Chacón * M. Beschi ** J. Sánchez * A. Visioli * S. Dormido ***

* *Dpto. de Informática y Automática, UNED, Madrid, Spain. (jchacon@bec.uned.es, jsanchez@dia.uned.es, sdormido@dia.uned.es)*
** *Dipartimento d'Ingegneria dell'Informazione, Università degli Studi di Brescia, Italy. (manuel.beschi@ing.unibs.it)*
*** *Dipartimento d'Ingegneria Meccanica e Industriale, Università degli Studi di Brescia, Italy. (antonio.visioli@ing.unibs.it)*

Abstract: In this work a remote lab developed to explore the properties of event-based controllers is presented. The architecture is based on the decomposition of the system in three tiers. The server layer is directly connected to the plant, the client side implements the controller and the graphical interface, and the middle-tier acts as interface between client and server. The implementation of the remote lab allows the control of a Coupled Tank plant through a generic network. The platform is used to evaluate the effectiveness of an algorithm that allows the user to find numerically the limit cycles in control schemes where the feedback is done through a level crossing sampling, and which can be applied to LTI systems with delay.

Keywords: Laboratory, education, event-based, limit cycles, experimental framework.

1. INTRODUCTION

In recent times many authors have addressed the design and implementation of Proportional-Integral-Derivative (PID) controllers from an event-based perspective (Årzén, 1999; Vasyutinskyy and Kabitzsch, 2006). This is because PID controllers are widespread in industry, mainly due to their satisfactory performance for many process and because they are relatively easy to design and tune. A comprehensive survey of the different methods proposed in the literature for event-based PID control can be found in (Sánchez et al., 2012).

Event-based control is nowadays an active research field in control engineering. However, the use of event-based strategies is not new, and in fact they have been used for long time in areas such as control of industrial processes (Kwon et al., 1999), robot path planning (Tarn et al., 1996), and engine control (Hendricks et al., 1994). Until recent times, 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 (see (Åström, 2008; Åström and Bernhardsson, 2002; Hespanha et al., 2007)). Recently, event-based control is also being used in multi-agent (Demir and Lunze, 2012) and distributed systems (Weimer et al., 2012).

Though different approaches exist (Yu and Antsaklis, 2011; Tabuada, 2007; Almeida et al., 2010), they share in common that sampling times are not uniformly distributed in time. Thus, they have the advantage that it is possible to react to changes in the plant more rapidly and only when it is really necessary, as opposed to time-triggered systems, where it is possible to have periods of time when nothing happens, but the control system is still wasting resources. Frequently, the main motivation of event-based control is to optimize the use of the resources, either in terms of power consumption, the use of a shared communication channel, or both. But there also exist systems where the event-based technique arises in a natural way because of the nature of the sensors, such as thermostats, bumpers, etc. Event-based control introduces non-linearities in the system, which can provoke phenomena which do not exist in linear systems. For example, a typical situation in a PID controller with send-on-delta (SOD) sampling is that a constant disturbance can lead to a sustained periodic oscillation, or limit cycle. There are several works in the literature that address the study of limit cycles in event-based control (Cervin and Åström, 2007; Åström, 1995; Gonçalves, 2000).

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. There are several reasons to justify the study of these limit cycles. First, 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, limit cycles mean oscillations, which, depending on the process, may be more or less problematic. For example, high frequency oscillations may wear out the actuators. Also, the study of limit cycles can be used for identification purposes. For example, the relay auto tuning method (Åström and Hägglund, 1984) is based on the properties of the limit cycle that appears 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 determine the stability of these limit cycles.

In this paper, the control scheme is based on level crossing sampling and considers two possible structures, the first one is when the sampler is located after the process output and the second one after the controller output (Beschi et al., 2012). Each case represents a configuration of a control scheme based on wireless transmissions, and has different properties.

A remote laboratory that was previously developed to allow experiments with event-based control structures, has been extended to implement the control loops studied in this work (Figure 1). With this implementation, it is possible to have the controller physically separated from the plant, and thus the communication can be done over a generic network, thus making the use of event-based structures sensible.

The focus of the work is two-fold. On one side it proposes a new experimental framework for the development of remote laboratories for control applications. The traditional approach has been to place the controller and the plant in the server-side. The user with this structure can only modify the controller settings from the client side, and make changes to the set point. This paper proposes a new architecture where the controller is placed on the client side. In this way the closed loop control system contains the network inside. This new approach fits like a glove in the philosophy of event based control systems. On the other side the paper presents a new algorithm which allows analyzing numerically the limit cycles that appear when a self-regulating process is controlled by a send-on-delta PID controller. The aim is to confirm that the limit cycles predicted by the theoretical analysis and the simulations performed appear in a real system, and compare the properties obtained from the study of the model with the properties of the limit cycles in the real system.

The organization of the paper is as follows. Section 2 presents the control schemes and describes the practical setup used in the experiments. Section 3 provides the mathematical context for the theoretical analysis of the limit cycles, and Section 4 shows the obtained results. Finally, in Section 5 we give the conclusions and future lines of work.

2. THE EXPERIMENTAL SETUP

2.1 The Plant

The platform used to obtain the experimental data is a remote laboratory built with two identical *Coupled Tank* plants by *Quanser* (Qua, 2013), situated in the Laboratory of Automatic Control at the Università degli Studi di Brescia. Each plant consists of two tanks and a water pump. One of the tanks is placed at the bottom, and the other at the top. The top tank has a valve whose output goes to the first tank. Thus, the system admits configurations of different complexities (Johansson, 2000).

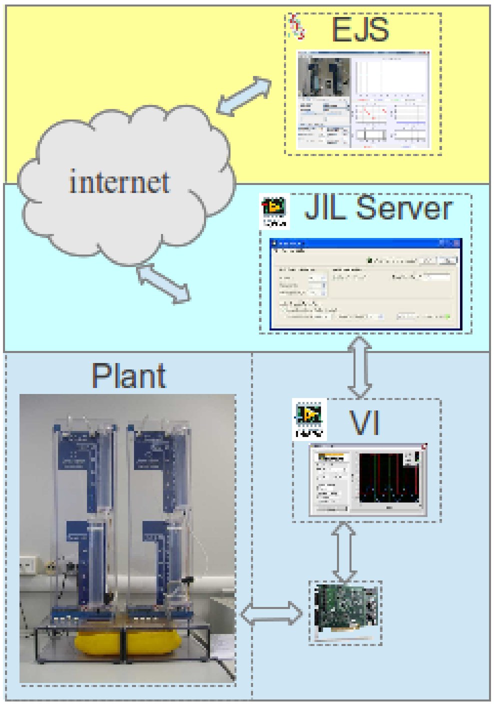


Fig. 1. The architecture of the control system: the plant is connected to a PC with a Data Acquisition Board. The PC executes a LabVIEW Virtual Instrument which exchanges the measures of the sensors and the control actions directly with the plant, and the JIL server enables the system to communicate with other PCs via Internet. The controller, implemented in *Easy Java Simulations*, runs in another PC.

2.2 The Controller

The controller is implemented in the client side running at the UNED, Madrid. It is a PID controller with a level crossing sampling strategy where, depending on the sampler location, either the sensor sends information to the controller only when the observed 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 to be the event that triggers the capture and the sending of a new sample. Thus, the controller can be divided into two parts, the continuous transfer function which corresponds to the PID (where actually the derivative part has not been employed), i.e. $C(s) = k_p + \frac{k_i}{s}$, and the SOD sampler which generates the discrete events.

2.3 The SOD Sampler

The SOD sampler is a block which has a continuous signal $v(t)$ as input and generates a sampled signal $v_{nl}(t)$ as output, which is a piecewise constant signal with $v_{nl}(t) = v(t_k), \forall t \in [t_k, t_{k+1})$. Each t_k is denoted as *event time*, and it holds $t_{k+1} = \inf\{t | \ t > t_k \wedge |v(t) - v(t_k)| \geq \delta\}$, except for t_0, which is assumed to be the time instant when the block is initialized as $v_{nl}(t_0) = v(t_0)$. Depending on the initial value, the non-linearity introduced could have an offset with respect to the origin, $\alpha = v(t_0) - i\delta$, where $i = \lfloor v(t_0)/\delta \rfloor$.

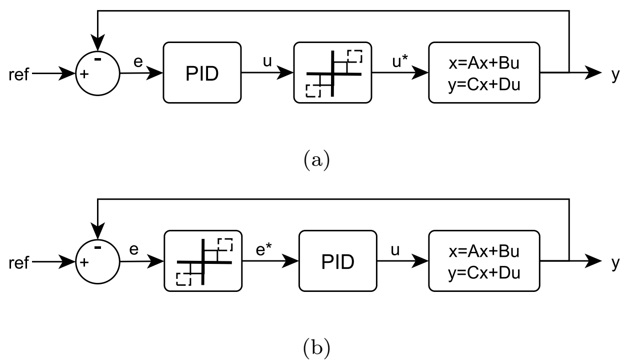


Fig. 2. Two control loops with different location of the send-on-delta sampler. (a) The error signal at the output of the controller is sampled, and (b) the process output is sampled.

2.4 The Remote Lab

The remote lab has been developed with the software tools Easy Java Simulations (EJS) (Christian and Esquembre, 2007; EJS, 2013), the JIL (Java-Internet-LabVIEW) Server (Vargas, 2010), and LabVIEW, that are combined to allow the interaction with the plant over the network. The controller is entirely in the client side, thus the event-based schemes are adequate because they allow the reduction of the data transmissions, using more efficiently the network resources. The remote lab is based on a three-layers architecture. In the server side, there is a PC connected to the plant through a Data Acquisition Board (DAQ). This PC runs a LabVIEW Virtual Instrument (VI) which implements monitoring funtions and acts as an interface with the plant, i.e. it allows to obtain the readings from the sensors and send the control action to the pumps. Also, there is a webcam connected which transmits a real-time video and audio streaming of the plant, to allow users to feel more like if they were in a real lab, even if they are connected from remote. The middle-layer is the JIL server, which publishes the variables (controls and indicators) of the VI to make them available over a network connection. Further, the third layer is the EJS application in the client side, which is not only the graphical interface to configure the control system and/or monitor the plant, but it also contains the controller implementation itself.

With regard to the communications, from an abstract point of view each node is composed of two components: a signal-generator and an event-generator. For example, for a sensor node the signal generator can be a zero-order hold that builds the signal from the periodic sensor readings, and the event-generator is the sampling scheme that decides when to communicate the data to another nodes. Note that since the event generator can also be configured to emulate a periodic sampling, this approach is also valid to represent a discrete control system.

From the point of view of the control system, the two control loops depicted in Figure 2 are considered. In the first configuration, the sampler is placed at the output of the controller, and in the second one it is situated after the process output.

The user interface (Figure 3) has been implemented in EJS, based on the use of *elements* (ready-to-use components which implement a functionality), namely the LabVIEW Connector Element, which encapsulates the connection with JIL Server, and the Process Control Library, which provides a set of control components, such as the implementation of the PID controller and of the SOD sampler that has been mentioned above. The use of these elements allows to facilitate the building of the lab and to assure its reliability.

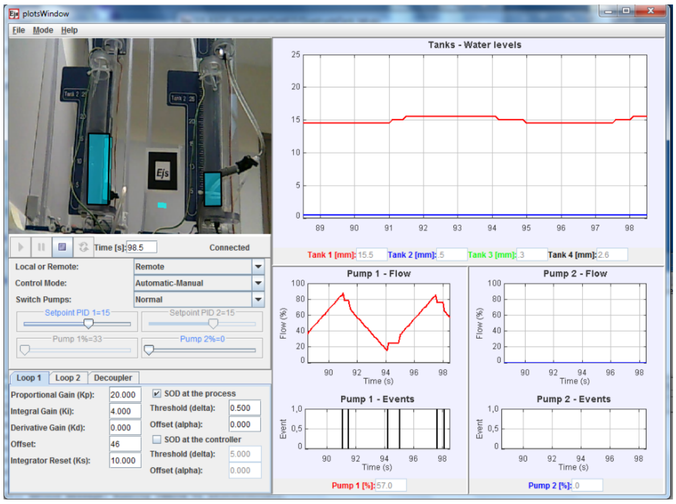


Fig. 3. The interface of the Remote Lab has been implemented in EJS. The state of the plant is shown by means of the plots at the right, and the image obtained from the webcam with augmented reality at the top-left part of the window. At the bottom-left, the user can configure the control system.

3. BACKGROUND

The state-space representation of the system depends on where the sampler is placed. Let the transfer function of the system be $P(s) = \frac{k}{\tau s+1}$. Defining the normalized system variables as $t' = \frac{t}{\tau}$, $k_p' = kk_p$, $k_i' = kk_i$, $\delta' = \frac{k}{\tau}\delta$, $u'(t) = \frac{k}{t}u(t)$, and $d' = \frac{d}{k_p}$, the dynamics of the system, if the sampler is at the process variable, can be written as,

$$\begin{aligned} \dot{x}(t') &= \begin{bmatrix} -1 & -k_i' \\ 0 & 0 \end{bmatrix} x(t') + \begin{bmatrix} -k_p' & k_p' \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_{nl} \\ d' \end{bmatrix} \\ \begin{bmatrix} y(t) \\ u'(t) \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & -k_i' \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ -k_p' & 0 \end{bmatrix} \begin{bmatrix} v_{nl} \\ d \end{bmatrix} \end{aligned} \tag{1}$$

where $v_{nl} = (j_k + \alpha)\delta$, and it has been assumed that $r = n\delta$, and thus the sampling offset is not affected by the setpoint.

Note that, to completely determine the behaviour of the system, the effect of the SOD sampler must be accounted for. This is done by considering that, at the inter-event times, the sampled signal is $v_{nl} = x_p(t_k)$, where $x_p(t)$ is the process output, and t_k is the time of the last sample.

On the contrary, if the sampler is at the control variable the representation of the system is,

$$\begin{aligned} x(t') &= \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} x(t') + \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{nl} \\ d \end{bmatrix} \\ \begin{bmatrix} y(t) \\ u'(t) \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ -k_p' & -k_i' \end{bmatrix} x(t) \end{aligned} \tag{2}$$

and, at the sampling times, $v_{nl} = [-k'_p \ -k'_i]$, $x = (j_k - \alpha)\delta$.

Note that in both cases the system can be represented as a Piecewise Linear System (PLS),

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_k \\ y(t) &= Cx(t) + D_k \\ S_k &= \{x | Cx + D_k - (j_k - \alpha)\delta = 0\} \end{aligned} \tag{3}$$

where $B_k = B(v_{nl} \ d)^T$ and $D_k = D(v_{nl} \ d)^T$. S_k are the switching surfaces, which correspond to the regions of the state-space where the dynamics can switch from one system to another.

3.1 Limit Cycles

In (Beschi et al., 2012), authors show that under certain conditions, a first-order system with time delay controlled by a PI with SOD sampler is asimptotically stable, and the system cannot have a limit cycle. In particular, the SOD must be symmetric without sampling offset $\alpha = 0$. Once provided that, it is possible to define a region of the controller parameters (k_p, k_i) where the stability is assured.

However, it is not always the case where these requirements can be fulfilled. As explained before, the structure of the control system must require that the sampler be at the controller variable. In this case, and considering that the system can be exposed to an external disturbance, in order to avoid limit cycles it is necessary to exactly estimate the disturbance, which is not possible in practice. Thus, it is of interest to characterize the limit cycles that can arise. In this section, an algorithm to obtain numerically the period of a limit cycle and the intermediate switching times is outlined. Consider an initial state $x(t_0) = (-n + \alpha)\delta$, and a limit cycle which crosses $2n$ levels, and let $t_0 = 0$ be the time when the process output crosses the minimum level of the limit cycle, with positive derivative. It is assumed that, in each semiperiod, the sign of derivative does not change, and thus, the process or the controller output at the switching times is an ordered sequence, i.e. $Cx(t_k) = (j_k + \alpha)\delta$, where $j_k = -n + k$, $k = 0, ..., 2n - 1$ in the first semiperiod, and $j_k = n - k$ in the second semiperiod.

With regard to the sampling offset, two cases are considered, $\alpha = 0$ and $\alpha = 0.5$. In both cases, due to the symmetry of the sampled signal around the origin, the equations can be simplified considering only one semiperiod.

Proposition 1. Consider the PLS in (3), with a nonlinearity defined by the switching surfaces $S_i = \{X \mid CX - (j_i + \alpha)\delta = 0\}$, 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. Define

$$f_k(t_1^*, \ldots, t_n^*) = C(I - e^{AT})^{-1} \left[\sum_{i=1}^{n-1} \Phi_{n-1} \cdots \Phi_{i+1} \Lambda_i \right] - E_k \tag{4}$$

where $\Phi_i = \Phi(t_i) = e^{At_i}$, $\Lambda_i = [\Gamma_1(t_i)U_{i-1} + \Gamma_0(t_i)U_i]$, and $E_k = (j_k + \alpha)\delta$. Then, the following conditions hold

$$\begin{cases} f_1(t_1^*, t_2^*, \ldots, t_n^*) = 0 \\ f_2(t_1^*, t_2^*, \ldots, t_n^*) = 0 \\ \vdots \\ f_n(t_1^*, t_2^*, \ldots, t_n^*) = 0 \end{cases} \tag{5}$$

and

$$\begin{array}{lll} E_i \le CX_i(t) < E_{i+1} & \text{for } 0 \le t < t_i^* & i = 1 \ldots n-1 \\ CX_n(t) \ge E_{n+1} & \text{for } 0 \le t < t_n^* & \\ E_i \ge CX_i(t) > E_{i+1} & \text{for } 0 \le t < t_i^* & i = n+1 \ldots 2n \end{array} \tag{6}$$

where

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

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

$$X_0^* = (I - e^{AT})^{-1} \left[\sum_{i=1}^{2n-1} \Phi_{2n-1} \cdots \Phi_{i+1} \Lambda_i \right]. \tag{7}$$

Proof. A similar proof can be found in (Gonçalves, 2000).

3.2 Algorithm

An algorithm to find the limit cycles is presented hereafter. It can be implemented either in a symbolic or in a numerical computation tool. Note that, though Proposition 1 considers only symmetric limit cycles, the definition of the algorithm has been done to cope with non-symmetric limit cycles, to be more general.

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

- 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 $v_{nl_i} := (j_i + \alpha)\delta$, and $x_{c_i} = -\frac{v_{nl_i} + k_p x_{p_i}}{k_i}$ (controller) or $x_{p_i} = v_{nl_i}$ (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, $\Delta_c = C_c[X(t_{max}) - X(t_{min})]$.

4. RESULTS

To identify the system as a first-order model, it has been assumed that the operating point is of 15mm for the water level, which corresponds to an input of around 50% of the maximum pump flow. A batch of step tests were introduced as input, thus obtaining a set of experimental data which was divided into two subsets, one of them

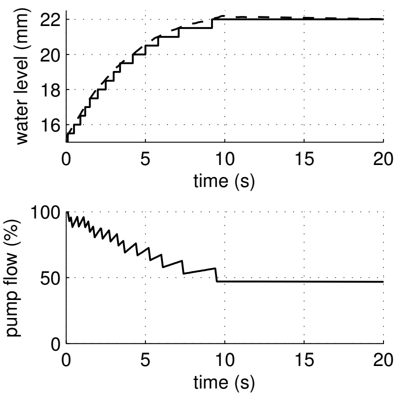


Fig. 4. Step response of the process, controlled by a PI with $k_p = 20$ and $k_i = 0.1$, and with the sampler at the process variable, with $\delta = 0.5$ and $\alpha = 0.5$. The top plot shows the process output measured at the process (solid line) and the send-on-delta sampled signal received by the controller. The bottom plot shows the output of the controller.

used for the identification and the other one for the validation of the model. Figure 4 shows the response of the system to a step input and the response of the linearized model obtained in the identification. The transfer function identified is ,

$$P(s) = \frac{0.63}{26.6s + 1} \tag{8}$$

The model was identified with a minimum squares method. The percentage of the process output variation explained by the model is around 90%.

An important practical issue is how to choose δ. As mentioned before, in theory the value of δ does not affect to the limit cycle properties. On the other hand, a value excessively small of δ can provoke unwanted events due to the signal noise, while a high value of the threshold can make the control system irresponsive. A rule of thumb is to choose a value to have around 10 samples in a step change. Since the step considered in the experiments ranges from 3 to 7 cm, choosing $\delta = 0.5$ seems reasonable. When the sampler is placed at the controller output, a value of $\delta = 5$ has been selected.

The first case considered is the PI controller with SOD sampler at the process output. The sampler parameters are $\alpha = 0$ and $\delta = 0.5$ The parameters have been tuned as $k_p = 20$, $k_i = 0.1$. Since these values are inside the stability region, the system does not present limit cycles. This can be seen in Figure 4, which shows the response of the system to a step change in the set-point.

Increasing the integral gain progressively, it can be observed how the trajectories of the system tend to a limit cycle with different number of levels. It is worth to note that, depending on the value of α, the number of level varies. In particular, if $\alpha = 0$ the number of levels that are crossed by the sampled variable in the limit cycle must be odd, and if $\alpha = 0.5$ it must be even.

Figure 5 shows this behaviour. As an example, for the particular case $k_i = 0.4$, the system presents a limit cycle with two levels. Solving the equations of a limit cycle with two levels, the solutions yields a period of $T \approx 9.96$, while

	T_m(s)	T_p(s)	A_m(cm)	A_p(cm)
$k_i = 0.2, \alpha = 0.5$	10.2	9.96	0.6	0.5
$k_i = 0.4, \alpha = 0.0$	9.2	9.02	1.5	1.43
$k_i = 0.8, \alpha = 0.5$	8.9	8.34	1.6	1.51
$k_i = 0.2, \alpha = 0.5$	15.3	15.02	0.44	0.52
$k_i = 1.0, \alpha = 0.5$	8.8	9.4	0.6	0.58
$k_i = 2.0., \alpha = 0.0$	8.2	8.5	0.7	0.64

Table 1. Comparison of the measured periods and amplitudes with the values computed with the algorithm.

the real limit cycle has a measured period of $T \approx 10.2s$. This means that the error in the prediction is around 3%. The comparison of the measured periods and amplitudes with the values computed with the algorithm is shown in Table 1, for all the different considered cases.

5. CONCLUSIONS

In this paper a new paradigm of remote labs for control education has been presented. The main characteristic is to place the controller in the client side and the plant in the server-side. This architecture allows exploring in an experimental way the properties of event-based controllers. A second result of this work has been to propose an algorithm to find numerically the limit cycles that can appear in a first-order system, demonstrating that these limit cycles can be generated in real systems. In particular, the experiments carried out with the experimental plant have reproduced a wide range of these limit cycles as predicted by the theory. Though the algorithm has been used for a first-order model, it is more general. It can be applied to a LTI system controlled by a general linear controller. The effect of the sampler in the system can be analyzed by considering a modification in the matrices of the state-space representation. The implementation of the control system has been done by using an architecture that has been proven to be adequate for remote laboratories. It is based on the use of EJS, JIL server, and LabVIEW. This platform allows the controller to be physically separated from the plant, and communicating with it through a network connection.

ACKNOWLEDGEMENTS

This work has been funded by the National Plan Projects DPI2011-27818-C02-02 and DPI2012-31303 of the Spanish Ministry of Science and Innovation and FEDER funds.

REFERENCES

(2013). *http://fem.um.es/Ejs.*

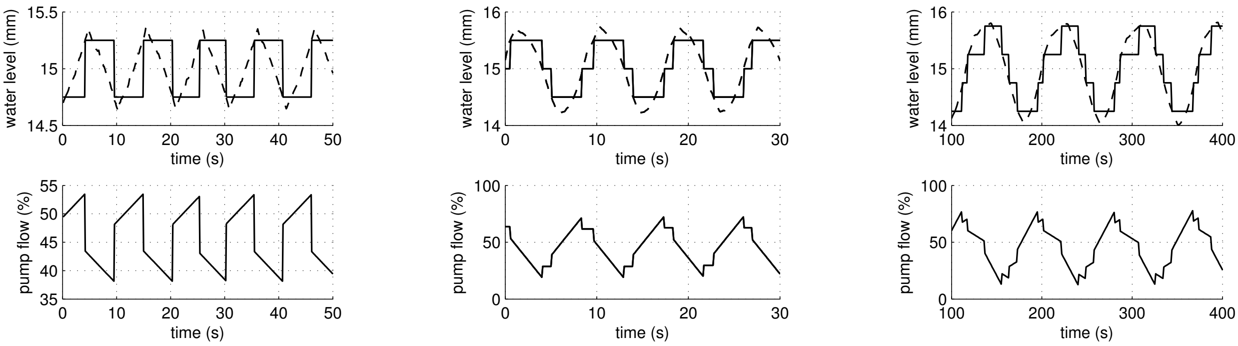
(2013). *http://www.quanser.com/.*

Årzén, K.E. (1999). A simple event-based PID controller. In *14th IFAC World Congress*. Beijing, P. R. China.

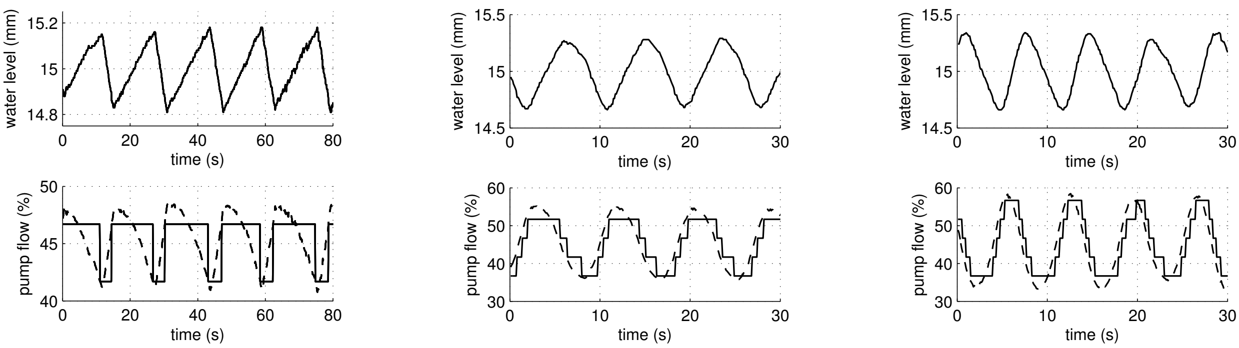
Åström, K.J. (1995). Oscillations in systems with relay feedback. In *Adaptive Control, Filtering, and Signal Processing, IMA Volumes in Mathematics and its Applications*, volume 74. Springer-Verlag.

Åström, K.J. (2008). *Event-based control.* Springer-Verlag, Berlin.

Åström, K.J. and Bernhardsson, B. (2002). Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *41st IEEE conference on Decision and Control*, 2011–2016. Las Vegas NV.



(a) $k_p = 20$, $k_i = 0.2$, $\delta = 0.5$, $\alpha = 0.5$ (b) $k_p = 20$, $k_i = 0.4$, $\delta = 0.5$, $\alpha = 0$ (c) $k_p = 20$, $k_i = 0.8$, $\delta = 0.5$, $\alpha = 0.5$



(d) $k_p = 20$, $k_i = 0.2$, $\delta = 0.5$, $\alpha = 0.5$ (e) $k_p = 20$, $k_i = 1.0$, $\delta = 0.5$, $\alpha = 0.5$ (f) $k_p = 20$, $k_i = 2.0$, $\delta = 0.5$, $\alpha = 0$

Fig. 5. Step response of the process, controlled by a PI with the sampler at the process variable (top plots) and at the controller variable (bottom plots). The plots show the process and the controller output measured locally at their corresponding node (dashed line) and the send-on-delta sampled signal (solid line).

Åström, K. and Hägglund, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20, 645–651.

Almeida, J., Silvestre, C., and Pascoal, A.M. (2010). Self-triggered state feedback control of linear plants under bounded disturbances. In *49th IEEE Conference on Decision and Control (CDC)*, 7588–7593.

Beschi, M., Dormido, S., Sanchez, J., and Visioli, A. (2012). Characterization of symmetric send-on-delta PI controllers. *Journal of Process Control*, 22(10), 1930–1945.

Cervin, A. and Åström, K.J. (2007). On limit cycles in event-based control system. In *46th Conference on Decision and Control*, 3190–3195.

Christian, W. and Esquembre, F. (2007). Modeling Physics with Easy Java Simulations. *The Physics Teacher*, 45(8), 475–480.

Demir, O. and Lunze, J. (2012). Event-based synchronisation of multi-agent systems. In *4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12)*. Eindhoven, The Netherlands.

Gonçalves, J.M. (2000). *Constructive Global Analysis of Hybrid Systems*. Ph.D. thesis, Massachussetts Institute of Technology.

Hendricks, E., Jensen, A., Chevalier, A., and Vesterholm, A. (1994). Problems in event based engine control. In *American Control Conference*, 1585–1587.

Hespanha, J.P., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. In *IEEE*, volume 95.

Johansson, K. (2000). The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *Control Systems Technology, IEEE Transactions on*, 8(3), 456–465. doi:10.1109/87.845876.

Kwon, W.H., Kim, Y.H., Lee, S.J., and Paek, K.N. (1999). Event-based modeling and control for the burnthrough point in sintering processes. In *IEEE Transactions on Control Systems Technology*.

Sánchez, J., Visioli, A., and Dormido, S. (2012). *PID Control in the Third Milennium*, chapter Event-based PID control, 495–526. Springer.

Tabuada, P. (2007). Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9), 1680–1685.

Tarn, T.J., Xi, N., and Bejczy, A. (1996). Path-based approach to integrated planning and control for robotic systems. *Automatica*, 32(12), 1675–1687.

Vargas, H. (2010). *An Integral Web-based Environment for Control Engineering Education*. Ph.D. thesis, Dptm. Informática y Automática, UNED, Madrid, Spain.

Vasyutinskyy, V. and Kabitzsch, K. (2006). Implementation of PID controller with send-on-delta sampling. In *ICC'2006, International Conference on Control*. Glasgow, Scotland.

Weimer, J., Araújo, J., and Johansson, K.H. (2012). Distributed event-triggered estimation in Networked Systems. In *4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12)*. Eindhoven, The Netherlands.

Yu, H. and Antsaklis, P.J. (2011). 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)*.