2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Experimental study of two event-based PI controllers in a solar distributed collector field

J. Chacón[1], J. Sánchez[1], L. Yebra[3], A.Visioli[2], S.Dormido[1]

Abstract— **In this work we focus 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), which can be used to sample either the process variable or the control variable. We use an algorithm to calculate the limit cycles properties and then we compare the results obtained in simulations with experiments performed on a real plant, a distributed solar collector field at the Solar Platform of Almería (PSA, Spain). The process is identified as a first order plus time delay (FOPTD). Practical issues, such as the implementation of an anti-windup mechanism and a bumpless transfer between the manual and automatic mode are also addressed.**

Event-based, PI, experimental results

I. Introduction

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 [1], robot path planning [2] and engine control [3]. 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 (for example [4], [5], and [6]). Recently, event-based control is also being used in multi-agent [7] and distributed systems [8].

Though different approaches exist ([9], [10], [11]), 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 arise in a natural way because of the sensors, such as thermostats, bumpers, etc.

In recent times many authors have addressed the design and implementation of Proportional-Integral-Derivative (PID) controllers from an event-based perspective ([12], [13]). 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 on [14].

The 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 sampling is that a constant disturbance can lead to a sustained oscillation, or limit cycle. There are works in the literature that address the study of limit cycles in event-based control ([15], [16], [17]).

In a previous work [18], we characterized with a systematic approach the behaviour of two event-based control structures, based on the use of a PI controller and a send-on-delta sampler, with an integrator plus time delay process (IPTD). We analyzed 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. The control scheme, based on a level crossing sampling, 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 (Figure 2). Each case represents a configuration of a control scheme based on wireless transmissions, and has different properties.

We have developed an algorithm which allows us to analyze computationally the limit cycles that appear when an interator or self-regulating process is controlled by a send-on-delta PID controller. Now, our aim is to confirm that the limit cycles predicted by the theoretical analysis and the simulations performed in fact 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 experiments have been done with an equipment, known as Acurex, built in 1981 at the Almeria Solar Platform (PSA, Spain [19]). In this plant (Figure 1), two types of collecting systems were considered, a central receiver system (CRS) and a distributed colector 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 [20] and [21].

The organization of the paper is as follows. Section II presents the control schemes and describes the practical setup used in the experiments. Section III states the theoretical analysis of the limit cycles, and Section IV shows the ob-

[1]J. Chacón is with the Department of Computing Science and Automation, UNED, Juan del Rosal 16, Madrid, Spain jchacon at bec.uned.es

[2]L. Yebra is with the CIEMAT at the Solar Platform of Almería, Solar Platform of Almería lyebra at psa.es

[3]A. Visioli is with the Department of Electrical Engineering, Università degli studi di Brescia, Italy a.visioli at ing.ubr.it

Fig. 1: Distributed solar collector field at the Solar Platform of Almeria (PSA), Spain.

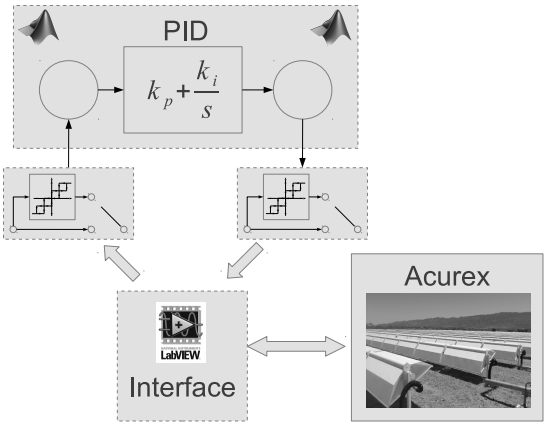


Fig. 2: Practical setup.

tained results. Finally, in Section V we give the conclusions and some future lines of work.

II. Practical setup

The control system is composed of the Acurex system, with its sensors measuring the state of the plant (temperatures, flows, etc.), the actuators (pumps), and two PCs. The PC connected to the plant has a SCADA software developed in LabVIEW that provides several functions such as the implementation of a local control system, monitoring capabilities or an interface to receive the state of the plant and send commands to the actuators. Finally, the other PC has the MATLAB software with the controller implementation. This setup is shown in Figure 2.

A. The Acurex system

Though the plant is complex and has many sensors, we are mainly interested in the temperature of the oil at the output of the receiver pipe, which is the controlled variable in our experiments. To control the heating of the oil, the flow through the pipes can be adjusted by a pump, which is able to provide a flow between 2 l/s and 9 l/s. The more flow the pump has, the least heat is transfer to the oil. This also imply that there is a change of sign in the response of the process variable to the control variable, which must be accounted for in the linear model.

B. The controller

The controller is a PID with a level crossing sampling strategy, where depending on the sampler location 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, i.e. $C(s) = k_p + \frac{k_i}{s}$, and the SOD sampler which generates the discrete events.

C. The SOD sampler

The SOD sampler is 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 $t_{k+1} = \inf\{t |\ t > t_k \wedge |u(t) - u(t_k)|\}$, except for t_0, which is assumed to be the time when the block is initialized as $u_{nl}(t_0) = u(t_0)$. 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 non-linearity resulting from the sampling of a continuous signal with the SOD sampler block is shown in Figure 3, where u_{nl} is plotted in the y axis, vs. u in the x axis.

D. 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 (we fixed it to $T_s = 15$s). This function receives the measures read from the sensors, updates the controller state and finally send the new control action to the actuators.

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

a) manual: In this mode, the control input is set manually by the operator in the MATLAB interface. This operation mode was used mainly to move the process to the operating process and to introduce steps in the input, during the identification phase.

b) SOD-PI: When the system is configured to work in *SOD-PI* mode, the SOD sampler block acts in the process variable. The control action is sent periodically to the actuators, but the information from the sensors is only received at event times.

c) PI-SOD: When the system is configured to work in *SOD-PI* mode, the SOD sampler block acts in the control variable. The information from the sensors is received periodically, but the control action is sent to the actuator only when an event occurs.

There are two important practical issues that must be handled: the *anti-windup* and the *bumpless transfer*. The following points describe the solution to these problems in the controller implementation.

1) Anti-windup: The integral windup is a well-known problem that may arise in PID controllers and which cause the loss of control capacity. Generaly it is due to physical limitations in the system which prevents the actuators to follow the demand of the controller, for example when the flow that a water pump must provide is higher than its maximum capacity. When that occurs, the system is irresponsive to changes in the error and, which is worst, the more time the integrator is in a windup state, the more time it takes to recover to a normal situation.

To cope with this problem, an anti-windup mechanism is frequently incorporated to practical implementations of the PID controller.

Though our control scheme is event-based, it is possible to incorporate an anti-windup mechanism similar to that used in discrete control systems as we show in the following lines.

2) Bumpless transfer: Another practical issue that must be accounted for is the switching between the automatic control mode and the manual control. Sometimes it is necessary to set the control input manually by the operator, thus bypassing temporarily the controller. However, when the controller is set again to work in automatic mode, it is possible to have an abrupt change in the control action which is suddenly introduced to the process, which might cause overshoot or even unstabilities. The mechanism that prevent this to occur is known as *bumpless transfer*.

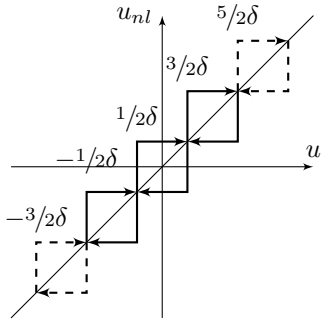


Fig. 3: Non-linearity resulting from the sampling of a continuos signal with the SOD sampler block.

Our approach to implement the bumpless transfer consists of two steps. First, when the controller change to manual mode the SOD sampler is still working. This means that events are generated even when the controller might ignore them. Finally, when the controller is set again to work in the automatic mode, at time t_k, the state of the integrator is reset to $x_c(t_k) = u(t_k)/k_i - (k_p/k_i)e(t_k)$, where $u(t_k)$ is the manual input and $e(t_k)$ is the control error.

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
```

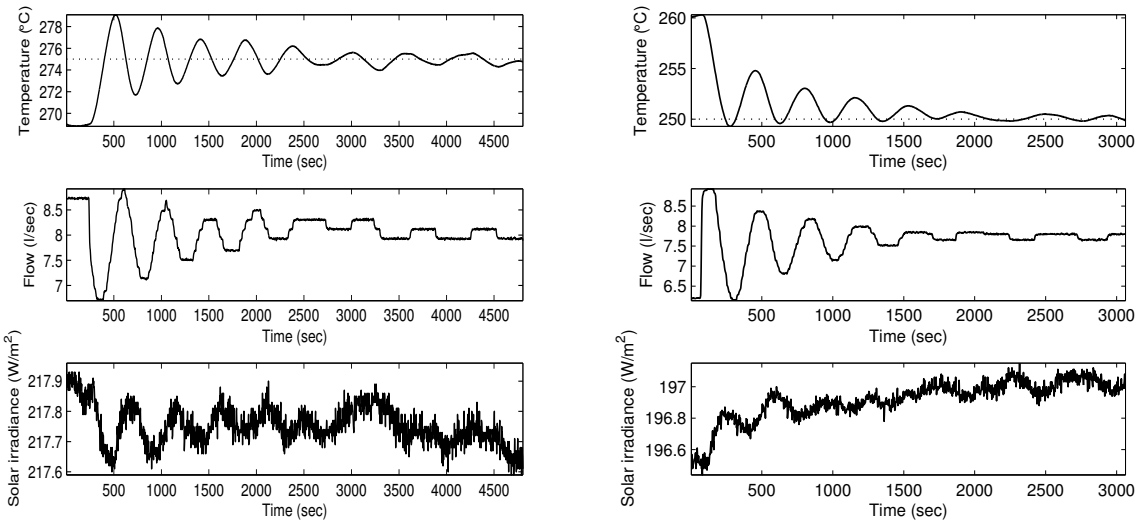
Listing 2: Code of the controller with the sampler at the controller output.

III. Analysis of the limit cycles

Assume that the process $P(s)$ is described by the time-invariant state-space system

$$P(s) \sim \begin{cases} \dot{x}_p(t) = A_p x_p(t) + B_p u_d(t) \\ y(t) = C_p x_p(t), \end{cases} \tag{1}$$

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



(a) (b)

Fig. 4: Responses of the Acurex system (solid line) to a step change in the setpoint, (a) with the event-based sampler placed at the controller output, and (b) with the sampler placed at the process output.

by a PI with proportional gain k_p and integral gain k_i, which is modeled as

$$C(s) \sim \begin{cases} \dot{x}_c(t) = A_c x_c(t) + B_c y_s(t) \\ u(t) = C_c x_c(t) + D_c y_s(t), \end{cases} \tag{2}$$

where $A_c = 0$, $B_c = 1$, $C_c = -k_i$, $D_c = -k_p$ and $y_s(t)$ 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

$$G_{ps}(s) \sim \begin{cases} \dot{X}_{ps}(t) = A_{ps} X_{ps}(t) + B_{ps} U_{ps}(t) \\ Y_{ps}(t) = C_{ps} X_{ps}(t) + D_{ps} U_{ps}(t), \end{cases} \tag{3}$$

where $X_{ps} = [x_p \ x_c]^T$, $Y_{ps} = [y \ u]^T$, $U_{ps} = [y_s \ d]^T$, and

$$A_{ps} = \begin{bmatrix} A_p & -k_i B_p \\ 0 & 0 \end{bmatrix} \quad B_{ps} = \begin{bmatrix} -k_p B_p & B_p \\ 1 & 0 \end{bmatrix}$$
$$C_{ps} = \begin{bmatrix} C_p & 0 \\ 0 & -k_i \end{bmatrix} \quad D_{ps} = \begin{bmatrix} 0 & 0 \\ -k_p & 0 \end{bmatrix}.$$

Sampling the control variable, the resulting system is

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

where $X_{cs} = [x_p \ x_c]^T$, $Y_{cs} = [y \ u]^T$, $U_{cs} = [u_s \ d]^T$,

$$A_{cs} = \begin{bmatrix} A_p & 0 \\ C_p & 0 \end{bmatrix} \quad B_{cs} = \begin{bmatrix} B_p & B_p \\ 0 & 0 \end{bmatrix}$$
$$C_{cs} = \begin{bmatrix} C_p & 0 \\ -k_p C_p & -k_i \end{bmatrix}$$

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 (4) from $t = 0$ to $t = t_i$. It gives

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

where $\Phi(t) = \mathrm{e}^{At}$ 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

$$\begin{cases} X_2 = \Phi(t_1) X_1 + \Gamma_1(t_1) U_n + \Gamma_0(t_1) U_1 \\ \vdots \\ X_n = \Phi(t_{n-1}) X_{n-1} + \Gamma_1(t_{n-1}) U_{n-2} + \Gamma_0(t_{n-1}) U_{n-1} \\ X_1 = \Phi(t_n) X_n + \Gamma_1(t_n) U_{n-1} + \Gamma_0(t_n) U_n \end{cases}.$$

Substituting recursively, we get the following expression,

$$[I - \Phi_n ... \Phi_1] X_n = \sum_{i=1}^{2n-1} \Phi_{2n-1} ... \Phi_{i+1} [\Gamma_1(t_i) U_{i-1} + \Gamma_0(t_i) U_i]. \tag{6}$$

First, we define the functions

$$f_i(X_i, t_1, ..., t_{2n}) = [I - \Phi_{2n} ... \Phi_1] X_i - \sum_{i=1}^{2n-1} \Phi_{2n-1} ... \Phi_{i+1} [\Gamma_1(t_i) U_{i-1} + \Gamma_0(t_i) U_i] \tag{7}$$

where the X_i appear as unknowns. Considering the switching conditions in the sampler, which fix the values of either the process output or the control input at the event times, the complete set of equations that must be solved to obtain the features of the limit cycle is

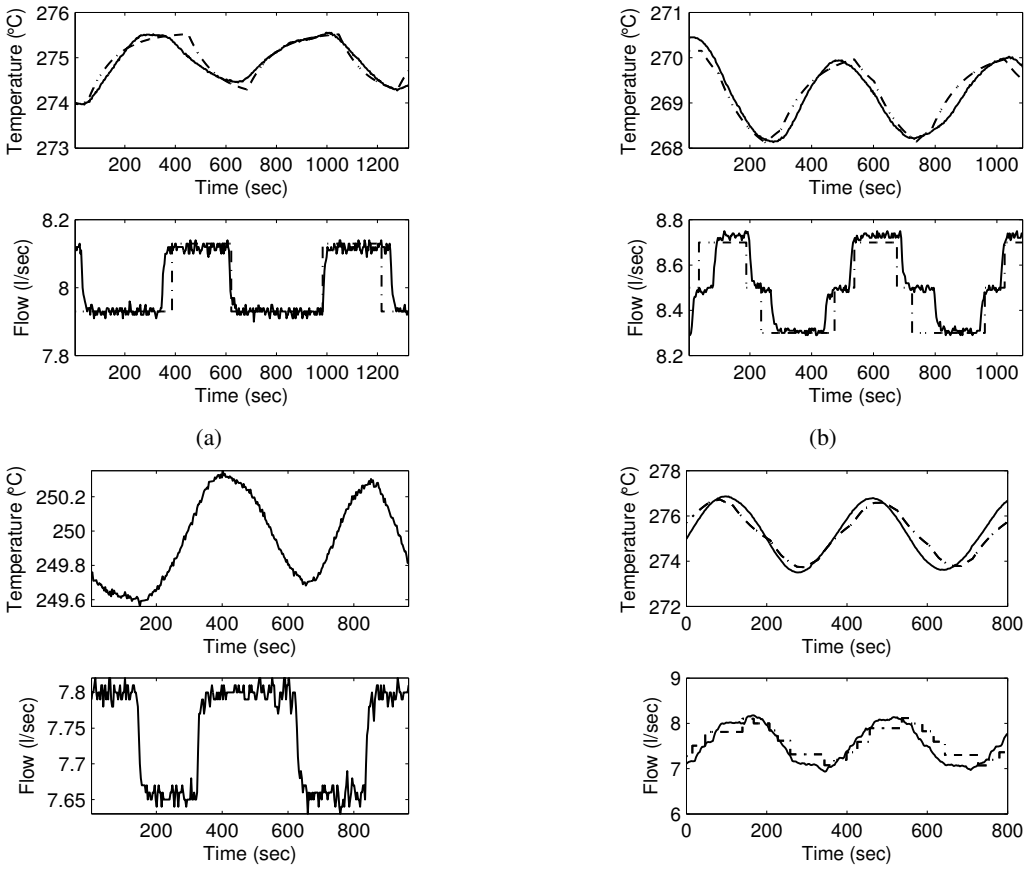


Fig. 5: Limit cycles in the Acurex system (solid line) and in the simulated model (dashed line) controlled by a PI with SOD sampling placed after the controller output, with (a) two levels and (b) three levels, and with the sampler placed after the process output, with (c) two levels and (d) eight levels.

$$\begin{cases} f_1(X_1^*, t_1^*, ..., t_{2n}^*) = 0 \\ \vdots \\ f_{2n}(X_{2n}^*, t_1^*, ..., t_{2n}^*) = 0 \end{cases} \quad \begin{cases} CX_1^* - d_1 = 0 \\ \vdots \\ CX_{2n}^* - d_{2n} = 0 \end{cases} \tag{8}$$

IV. Experimental results

In this section, we show experimental results which clearly evidence the existence in a real system of the results derived in the previous sections. Therefore the experiences we carried out were focused on finding limit cycles in the Acurex system in order to compare with the results obtained by simulation. As we show in the following paragraphs, it is worth to note that even when the model we use 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 that predicted in theory.

A. The model

The plant was identified as a FOTPD transfer function, where the process input is the oil flow (m^3/sec) in the pipes and the process output is the temperature of the oil after the heaters (°C). There are unmodeled dynamics that we consider as external disturbances, such as the temperature of the oil 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, 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,

$$P(s) = -\frac{6.07(\pm 0.1)}{103.27(\pm 3.8)s + 1} \mathrm{e}^{-67.50(\pm 0.7)s}. \tag{9}$$

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. Also, the time constants and delays obtained make sense from the previous published works in this field. Notice the minus sign in (9), which represents the inverse response of the plant, i.e. a positive change

in the flow produces a negative change in the temperature.

The model identified corresponds to a linearization around the operating point, and thus it is only valid in a limited range.

B. Results

1) Controller Sampling: The first set of tests we present were carried out with the controller in *PI-SOD* mode, with the purpose of reproducing the limit cycles obtained in simulation composed of two levels (with $\alpha = 0.5$) and 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 in the controller.

The response is shown in Figure 4.a, where we show the oil temperature, the flow and the solar irradiance during the test. 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.

Figure 5.a and 5.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).

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 (Figure 5.c), we searched for the existence of a more complex limit cycle. Increasing the proportional gain we found a limit cycle with eight different levels, which is shown in Figure 5.d. 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.

V. Conclusions

We have implemented and tested an event-based PI controller with a send-on-delta sampler in an experimental platform which consists of a distributed solar collector field. The controller admits two possible configurations, one with the sampler placed after the controller output and the other with the sampler placed at the process output.

The experiences carried out with the system have demonstrated that the limit cycles that have been predicted for the proposed control scheme by the theoretical analysis and simulations in fact appear in real systems. Moreover, the period and amplitude of the limit cycles estimated by the FOPTD model were close to that of the real system.

Several implementation issues such as the integrator windup of the PI controller and the bumpless transfer from manual to automatic mode have been also address successfully.

As future lines of work, it can be interesting to compare the obtained results with the predictions of a more complex model of the system, for example a SOPTD, to see whether the properties obtained in theory are closer to the real process. A possible improvement of the controller is the addition of auto-tuning capabilities, for example reducing the gains of the controller when the process enter into a limit cycle. Also, another improvement can be to incorporate a disturbance estimator to the control scheme.

References

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

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

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

[4] K. J. Åström. *Event-based control*. Springer-Verlag, Berlin, 2008.

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

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

[7] O. Demir and 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.

[8] J. Weimer, J. Araújo, and 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.

[9] H. Yu and 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)*, December 2011.

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

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

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

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

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

[15] A. Cervin and K. J. Åström. On limit cycles in event-based control system. In *46th Conference opn Decision and Control*, pages 3190–3195, December 2007.

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

[17] Jorge M. Gonçalves. *Constructive Global Analysis of Hybrid Systems*. PhD thesis, Massachussetts Institute of Technology, 2000.

[18] J. Chacón, J. Sánchez, A. Visioli, and S. Dormido. Analysis of the limit cycles in the PI control of IPD processes with send-on-delta sampling. In *IEEE International Conference on Control Applications, Dubrovnik (HR)*, pages 1609–1614, 2012.

[19] F. A. Schraub and H. Dehne. Electric generation system design: management, startup and operation of IEA distributed collector solar system in Almería, Spain. *Solar Energy*, 31(4):351–354, 1983.

[20] E. F. Camacho, F. R. Rubio, M. Berenguel, and L. Valenzuela. A survey on control schemes for distributed solar collector fields. Part I: Modeling and basic control approaches. *Solar Energy*, 81(10):1252–1272, October 2007.

[21] E. F. Camacho, F. R. Rubio, M. Berenguel, and L. Valenzuela. A survey on control schemes for distributed solar collector fields. Part II: Advanced control approaches. *Solar Energy*, 81(10):1252–1272, October 2007.