Implementation and evaluation of event-based PID in the IEC-61499 standard

Oscar Miguel-Escrig
Department of System Engineering and Design.
Universitat Jaume I. Campus del Riu Sec
Castelló de la Plana. Spain.
omiguel@uji.es

Julio-Ariel Romero-Pérez
Department of System Engineering and Design.
Universitat Jaume I. Campus del Riu Sec
Castelló de la Plana. Spain.
romeroj@uji.es

***Abstract*—This paper presents a study about the performance of event-based PID controllers implemented in the standard IEC-61499. The implementation details are addressed and experimental results are given. The main goal is to evaluate the benefits and drawbacks of using the event-based controllers under this standard in substitution of the periodic controllers that has been used so far. Both the computational cost of the control algorithms and the close loop performance when using periodic and event based controllers are analyzed in order to compare these two approaches.**

***Index Terms*—PID, event-based, IEC 61499**

I. Introduction

In the last decade, numerous investigation about event-based control systems have been carried out. The main difference between periodic and event-based controllers (EBC) is its execution: in periodic controllers it is done periodically while in EBC is done only after the arrival of asynchronous events that indicate significant changes in the state of the system. Example of such asynchronous events are the level-crossing of the error signal (difference between set-point and controlled output), [1].

Simultaneously to the theoretical advances in EBC, a new standard, named IEC 61499 [2], has been developed for programming distributed control systems. This standard introduces new concepts in comparison to its predecessor, the IEC 61131 [3], widely used nowadays in programming of industrial automation systems; mainly, whereas the IEC 61131 is based in SCAN cycles, the IEC 61499, whose goal is to favor the design of distributed and reconfigurable control applications, is based on the management and treatment of events.

In this paper we present a study about the implementation issues and the performance of EBC algorithms using the standard IEC 61499. The main goal is to evaluate the benefits of using EBC under this standard in substitution of the periodic controllers that has been used so far. Both the computational cost of the control algorithms and the close loop performance when using periodic and event based controllers will be analyzed in order to compare these two approaches. The study of stability and robustness issues of EBC is out of the scope of this paper.

Among all the alternatives of event based controllers, the event-based PID has been selected for this study since PID algorithm is used in most of the industrial control applications. Furthermore, there are several works in which different versions of event based PID have been proposed, [4], [5], [6].

II. Brief description of IEC-61499 standard

The IEC 61499 standard is centered in the concept Function Block Model (FBM), where a function block (FB) is a functional software, with its own data structure that can be manipulated by one or more algorithms that gives functionality to a FB. A FB defines a *type*, from where one or more instances can be created.

In FBs converge two types of parameters, data and events which regulate the execution of the FB. A FB can receive and send this type of parameters and associate them so that the reception or the sent of an event corresponds with the refresh of the input or output data associated with it. The FB encapsulates a concrete functionality, this leads to internal variables within the FBs. As the IEC-61499 doesn't allow the existence of global variables for the application, the algorithms whithin a FB only have access to the inputs, outputs and internal variables, being all of them stored internally between calls to the FB.

In Figure 1 the elements of a FB described above are shown. In the upper part is represented the execution control that manages the entry and exit of events. In the inferior part is represented the functionality of the FB, which consist of the algorithms which process the data. When an input event arrives, the associated data is refreshed and the algorithms related to the event are executed. The association between events and data is shown as vertical lines and boxes that appear in the entry and exit of the FB. As a result of the execution of the algorithm, output events and data are generated and sent.

The Basic Function Blocks (BFB) have a characteristic structure and a behavior defined in function of input parameters. Internally, are constituted by a Execution Control Chart (ECC), in which its transitions are regulated by input events or by variables treated during its execution.

This work has been supported by MICINN project number TEC2015-69155-R from the Spanish government and project number P1·1B2015-42 from Universitat Jaume I.

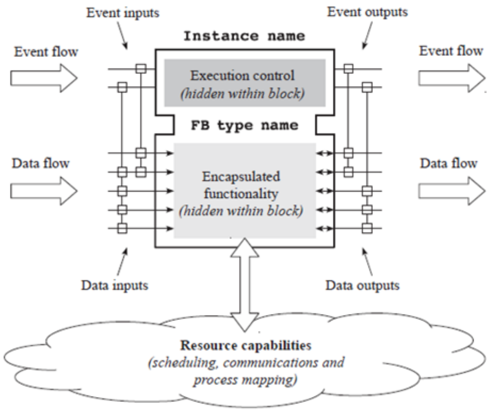


Fig. 1. FB structure, [2].

FBs are grouped in a model that configures an application of the IEC-61499 standard (called Application Model). An application is defined by the different instances of the FB that composes it (there could be multiple instances of the same FB type) and the interconnection of them. As the full functionality of the program is defined with the FB and its connections, global variables aren't need.

III. EVENT-BASED PID CONTROLLERS

One of the first contributions to the development of the event-based PID controllers was introduced by Årzén in [4] as a way of reducing the CPU usage of the control systems based in computer without affecting in a significant way the performance of the control loop. In his work, Årzén spotted some of the principal considerations for the EBC. Among them it can be highlighted the errors due to the calculus of the integral and derivative terms when the time between samples increases significantly. Several works have been done afterwards in order to correct the errors spotted, the most remarkable are those in relation with the corrections to the computation of the integral term, in particular, the works done by Durand [5], [7] and Vasyutynskyy [6], [8].

In particular, the Årzén's algorithm, which is presented in Listing 1, implements a periodic call to the controller, but the calculus execution isn't periodic. Årzén proposes an event generation strategy based on the difference between the actual error and the last execution's error: if this difference is bigger than a preset threshold, then a new execution event is generated. Additionally, the controller output is also updated if the time without execution reaches a maximum given value. These two conditions for execution correspond to line 4 in Listing 1. A significant difference from the PID controllers based in SCAN cycles, which pre-calculates some coefficients, is that in Årzén's algorithm some of them must be re-calculated at each execution provided they depend on the time between executions, lines 6-8 in Listing 1.

IV. EBC IMPLEMENTATION WITHIN IEC-61499

The development of controllers using the standard IEC-61499 is carried out by the implementation of new FBs, because these are the programming basic unit of this standard. In this study, a classic periodic controller is implemented and compared with Årzén's EBC, developed in [4].

Listing 1. The event-based PID algorithm proposed by Årzéns [4].

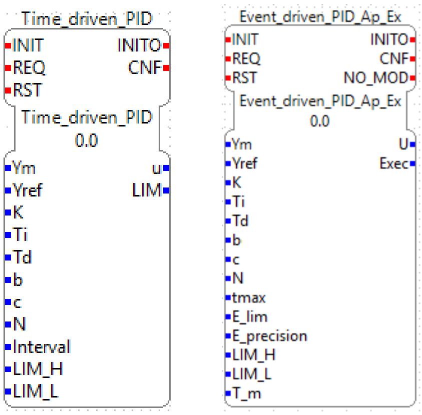
```
1    % detection
     e = ysp - y;
3    hact = hact + hnom;
     if abs(e - e_old) > elim || hact > hmax
5            % discrete PID coefficients
             ad=Td/(N*hact+Td);
7            bd=K*Td*N/(Td+N*hact);
             bi=K*hact/Ti;
9            % control algorith of discrete PID
             up= K*(b*ysp - y);
11           ud=ad*ud + bd*(c*(ysp-ysp_old)-(y-y_old));
             ui=ui+bi*(ysp - y);
13           u=up + ui + ud;
             if (u<u_sat_min)
15                   u=u_sat_min;
                     if (ysp-y<0)
17                           ui=ui-bi*(ysp - y);
                     end
19           end
             if (u>u_sat_max)
21                   u=u_sat_max;
                     if (ysp-y>0)
23                           ui=ui-bi*(ysp - y);
                     end
25           end
             y_old=y;
27           ysp_old=ysp;
             hact = 0;
29           event=1;
     end
31   u_out = u;
```

In order to implement and evaluate the behavior of the controllers the software 4DIAC has been used, which is an *open-source* development environment. An alternative more focused on the industrial field would be the software NxtOne, which in addition of developing IEC-61499 applications, supports hardware from different industrial manufacturers such as BECKHOFF, WAGO or SIEMENS.

4DIAC allows the creation of FBs, systems, applications and all the models of the IEC-61499 standard. Applications developed with 4DIAC are interpreted by the runtime FORTE that must be executing in the control device. This runtime can be downloaded from 4DIAC's page.

In Figure 2 the external structure of two controllers is shown. On the left is presented the periodic controller and on the right Årzén's controller. In this case both are BFBs, so they are internally constituted by an ECC.

Årzén's controller ECC is shown in Figure 3. This ECC presents an initialization and a reset algorithm, but more interesting is its execution, once the event REQ is received, the algorithm passes to *Check_point* state, where the time between executions is calculated in the algorithm *Check_time*. Once this calculus has been done, the conditions on the transitions are evaluated, which are the conditions that Årzén imposes for the execution of his controller (*abs* function return an integer, that's why we must include some significant digits with



(a) (b)

Fig. 2. Function Blocks of periodic PID controller (a) and event-based controller (b).

E_precision). If the conditions aren't satisfied, the algorithm enters in the state SENDNOMOD and NO_MOD event is sent. Otherwise, algorithm *Req* is executed, where the calculus of the new control output is computed.

Finally, an application with different FBs is modeled for the correct execution of the control algorithm. Several FBs are included, such as XENO_RT_E_CYCLE, used for starting the execution of the FB chain periodically, BBBio_AIn, that provides the measured variable, F_RT_CLOCK_NS, that provides the processor's time to the controller FB, and an additional FB for refreshing the output PWM, BBBio_PWM. This model is shown in Figure 4.

The execution of the application is developed in two phases. In the first phase an initialization is done, an event is sent to the INIT entry of the FB Cycle from the resource that contains the application. Once this FB has been initialized, all the chain is also initialized until arriving to the PWM FB, being this FB who starts the execution phase with the event to the START entry in the FB Cycle, starting the periodic execution of the control algorithm, executing or not the controller depending on the conditions stated previously.

V. Execution Times Modeling

As it has been said before, one of the main objectives of this work is the evaluation of EBC according to IEC-61499 standard. In this evaluation it will be treated the behavior of the control system regarding to its response, and also the computational cost of this type of controllers. It will be effectuated a comparison with a periodic controller.

First of all, we can characterize the time of carrying out Q execution of the periodic controller as the sum of the execution times of each of the FBs that forms its execution chain as is stated in the equation (1):

$$t_{PIDtb} = Q(t_{AIN} + t_{PID} + t_{PWM}) \tag{1}$$

where t_{AIN} is the execution time of the FB that reads the analogue input (BBBio_AIN), t_{PID} is the execution time of the PID FB and t_{PWM} is the execution time of refreshing the output (BBBio_PWM).

In the case of event-based PID, its equation must take into account the percentage of generated events, that we call η, that is, when is the condition for the calculation of a new control output satisfied, thus, the control algorithm must be fully executed. The execution time is described by the equation (2) obtained from the model presented in Figure 4. In this case, t_{PID_e} is the time consumed by the PID FB when a new control output must be calculated, $t_{PID_{ne}}$ is the time spent by the PID FB when no control output has to be calculated and t_{CLK} is the time consumed by F_RT_CLOCK_NS FB, that obtains the processor time.

$$\begin{split} t_{PIDeb} = Q(\eta(t_{AIN} + t_{PID_e} + t_{PWM} + t_{CLK}) + \\ (1-\eta)(t_{AIN} + t_{PID_{ne}} + t_{CLK})) \end{split} \tag{2}$$

Using equations (1) and (2) the time percentage needed by the EBC with respect to the periodic controller can be calculated, in function of the event generation percentage (η). This relation is given by the following quotient:

$$\begin{split} \frac{t_{PIDeb}}{t_{PIDtb}} = \frac{Q(\eta(t_{AIN} + t_{PID_e} + t_{PWM} + t_{CLK}))}{t_{AIN} + t_{PID} + t_{PWM}} + \\ \frac{(1-\eta)(t_{AIN} + t_{PID_{ne}} + t_{CLK})}{t_{AIN} + t_{PID} + t_{PWM}} \end{split} \tag{3}$$

Defining $\gamma = \frac{t_{PIDeb}}{t_{PIDtb}}$ and simplifying, we obtain the relation between γ and η given in equation (4), that corresponds to the equation of a line whose coefficients depend on the execution times of each of the FB involved in the application model.

$$\begin{split} \gamma = \eta \frac{t_{PID_e} - t_{PID_{ne}} + t_{PWM}}{t_{AIN} + t_{PID} + t_{PWM}} + \\ \frac{t_{PID_{ne}} + t_{AIN} + t_{CLK}}{t_{AIN} + t_{PID} + t_{PWM}} \end{split} \tag{4}$$

From the equation above some interesting conclusions can be deduced about the time spent by an EBC regarding to the periodic controller in function of the execution percentage of the EBC. For example, evaluating the equation (4) for $\gamma = 1$, we obtain the event generation percentage η for which the EBC has the same average execution time that the periodic controller. That value is given by the following equation:

$$\eta = \frac{t_{PID} - t_{PID_{ne}} + t_{PWM} - t_{CLK}}{t_{PID_e} - t_{PID_{ne}} + t_{PWM}} \tag{5}$$

Evaluating the equation (4) for $\eta = 0$, we obtain the average time spent by the EBC regarding to periodic controller for an event generation percentage equal to zero. This condition is fulfilled when the controller is operating in a stable state, if no events are generated. This equation would be:

$$\gamma = \frac{t_{PID_{ne}} + t_{AIN} + t_{CLK}}{t_{AIN} + t_{PID} + t_{PWM}} \tag{6}$$

Finally, if we evaluate the equation (4) for $\eta = 1$, we obtain the average time consumed by the event based controller in relation to the periodic controller for a 100% rate of event generation. This behavior occurs in the transitory of

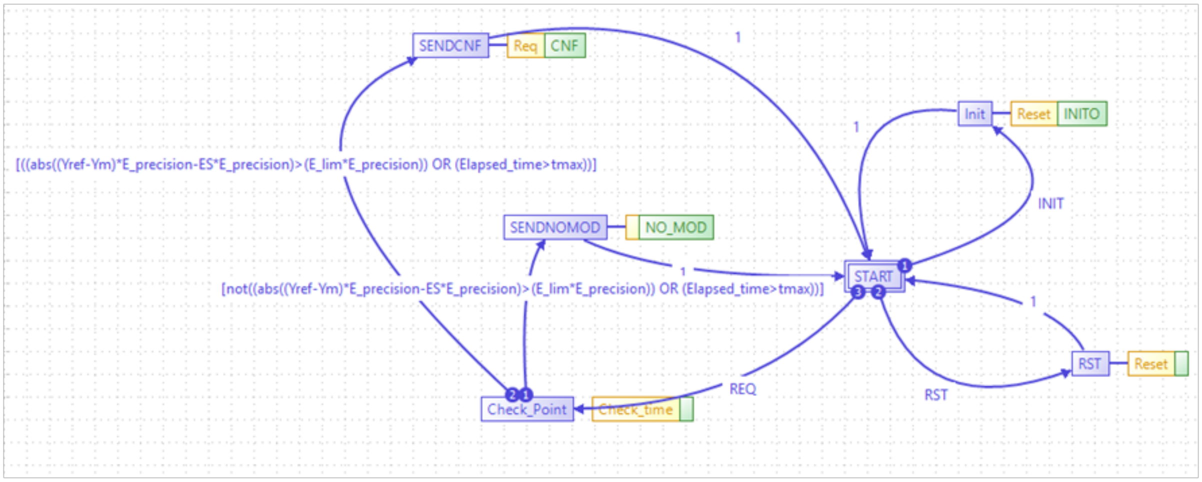

SENDCNF
Req
CNF
1
Init
Reset
INITO
1
INIT
[((abs((Yref-Ym)*E_precision-ES*E_precision)>(E_lim*E_precision)) OR (Elapsed_time>tmax))]
SENDNOMOD
NO_MOD
1
START
1
3
2
[not((abs((Yref-Ym)*E_precision-ES*E_precision)>(E_lim*E_precision)) OR (Elapsed_time>tmax))]
1
RST
Reset
REQ
RST
2
1
Check_Point
Check_time

Fig. 3. Årzén's event based PID controller, proposed in [4], ECC.

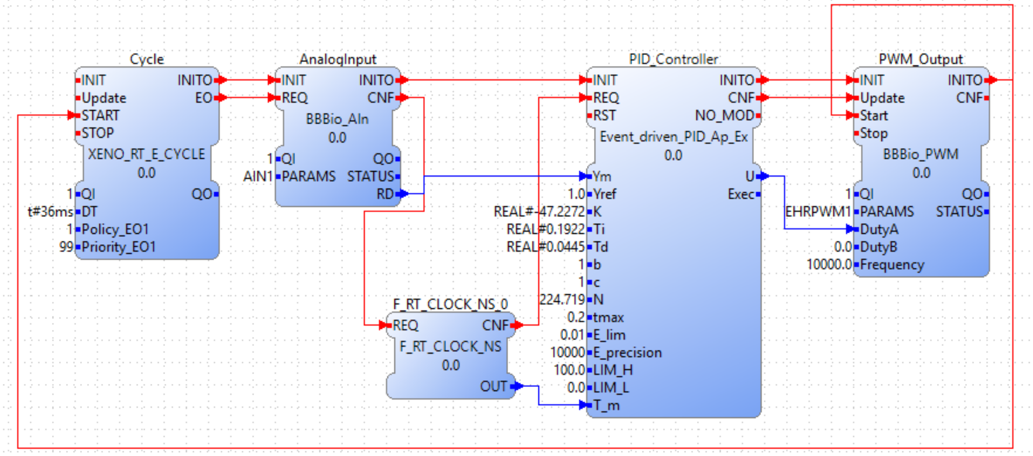


Fig. 4. Control application with an event-based PID controller using IEC-61499 standard structure, developed in 4DIAC.

the system, when a change in the reference is produced or a perturbation of a significant value arrives.

VI. EXPERIMENTAL PLATFORM AND RESULTS

The experimental platform for the evaluation of the behavior of EBC is composed by a card BeagleBone Black (BBB) and the systems to control. This card has several analog inputs and PWM outputs, which make this card a suitable option for our experiments. Other alternatives such as Raspberry Pi have also been evaluated.

Focusing on the simplicity and versatility of the experimental platform, we chose to define the systems to control with electronic circuits whose dynamics is determined by their passive components. In particular, the chosen circuit is a second order low-pass filter configured with a Rauch structure. Its transfer function is defined by its components as it is shown in equation (7).

$$H(s) = -\frac{(R^2C_1C_2)^{-1}}{s^2 + 3(RC_2)^{-1}s + (R^2C_1C_2)^{-1}} \tag{7}$$

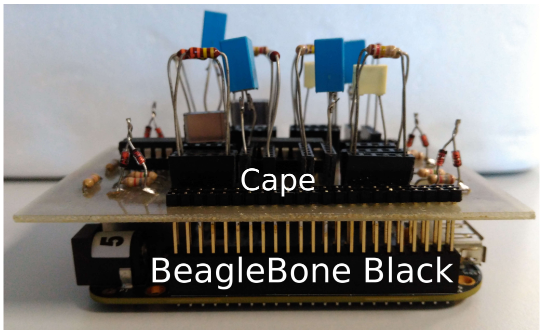


Fig. 5. Set of BeagleBone Black and cape with the systems to control.

Four of this circuits have been included in an electronic card that can be connected directly to the BBB as a cape, as it is shown in Figure 5. The components that define the dynamics of the circuit can be easily replaced, having the possibility of testing with systems with different dynamics.

For programming the BBB card and making the experiments, the card must be connected to the PC and via a SSH connection, the runtime FORTE (the version of FORTE must have our FB compiled) is executed. Afterwards, the application is downloaded to the BBB with 4DIAC and the execution begins automatically.

BBB cards have installed by default a Linux-based operative system that doesn't allow the execution of real-time applications. This supposes a big problem since the execution of each FB with this OS takes 5 ms independently of the execution time of each one. With the aim of improving the performance of the BBB, Xenomai has been installed in it. This provides an application management in real-time and reduces significantly the card response time.

In order to apply the equation (4) to our periodic control application and event-based control application, some measures on each FB used in this applications have been made with the BBB. It has been measured the execution times using an oscilloscope, measuring the time between the activation of a digital output (placed before the measured FB) and the deactivation of this digital output (placed after the measured FB). A preliminary test has been made in which a digital output was activated and deactivated in order to correct the times obtained with the characterization of the other FBs. The time spent in activating and deactivating a digital output is around 4 μs. This value has been subtracted to the time obtained for each FB. The gathered results of this study are shown in Table I.

TABLE I
AVERAGE EXECUTION TIMES OBTAINED EXPERIMENTALLY.

Measured time	Value (μs)
t_{AIN}	8
t_{PWM}	12.7
t_{PID}	8.5
t_{PID_e}	10.5
$t_{PID_{ne}}$	5.5
t_{CLK}	2

Using data from Table I in the equation 4, the line showed in Figure 6 is obtained, from where some important results can be extracted:

- The average execution time of the EBC is equal to the time required for the periodic controller if the event generation percentage (η) is 77.12%.
- If the event generation rate is equal to 100%, then the average execution time spent by the EBC surpasses the periodic controller execution time by a 13.8%.
- At last but not least, if no events are generated, the average execution time of the EBC spends the 53.45% of the time spent by the periodic controller.

In relation with this last point, it must be noticed that, for the controller proposed by Årzén, an event generation rate equal to zero doesn't exist because the algorithm fixes a maximal time (hmax) that if it is surpassed an event will be generated. Then, if the periodic call to the algorithm is done in a nominal time (hnom), the event generation rate for this time constraint will be hnom/hmax. This would be the minimum event generation rate that can be reached by Årzén's algorithm.

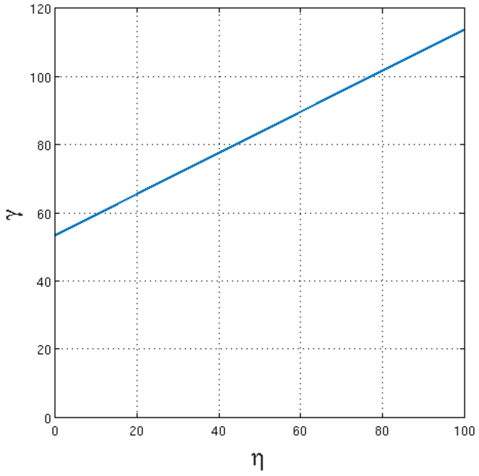


Fig. 6. Relation between η y γ.

For the study of the behavior of the control system it has been considered the circuit mentioned above, whose transfer function is given by the equation (8) and it is function of its components.

$$G(s) = \frac{0.612}{s^2 + 9s + 34} \tag{8}$$

The tunning of the parameters of the PID controller has been done using the AMIGO method presented in [9]. For the design we have used a maximum value for the sensibility function of Ms=1.4 and a filtering coefficient of the derivative N=10. The parameters obtained are: Kc=47.2, Ti=0.19, Td=0.044. This parameters have been used both for the periodic PID controller and for the event-based controller. The sample time in both cases has been fixed to hnom=36ms and the reference and derivative term weighting coefficients were considered b=1 y c=1. Furthermore, the maximum and minimum saturation values for the control output were: LIM_H=5 y LIM_L=0. For the event-based controller it has been considered a threshold of E_lim=0.01 and a maximum time without execution hmax=5hnom.

The Figure 7(a) shows the response obtained with the periodic controller and with the EBC proposed by Årzén facing three changes in the reference in $t = 15$, 20 and 25 seconds. It can be observed that the response of the periodic PID is similar in all the reference changes, whereas the EBC has significant differences in the third change of reference. This is a consequence of the casuistry in the event generation because the response of the EBC algorithms strongly depends on the arrival of events. It can be observed that the control output of the EBC is more abrupt than the control output from the periodic controller. This is due to the effect of the integral term, as it has been noted by Durand and his collaborators in [7].

TABLE II
IAE AND TV INDEXES OF THE EXPERIMENTAL RESPONSE OF BOTH CONTROLLERS.

Index	Experiment	Periodic	Årzén
IAE	15sec	0.103	0.0948
	20sec	0.1035	0.0969
	25sec	0.1031	0.0792
TV	15sec	0.6378	0.6431
	20sec	0.6106	0.6747
	25sec	0.6264	0.5323

Figure 7(b) shows the behaviour of γ during the control experiments presented in Figure 7(a). It is worth noticing that the value of γ reach its maximum (113.8%) during the first part of the response to step like changes in the set-point. However, most of the time γ is clearly lower than 100%, and keeps the minimum value (66%) during the steady state of the system, which means that during that time the average execution time of Årzén controller takes the 66% of the average execution time that takes the periodic controller. This minimum value rise from the selection of hnom=5hmin used in this experiment.

With the objective of quantifying the difference between both responses, the IAE index of the output and the TV of the control output for each controller has been calculated. The values obtained are shown in Table II, where it can be observed that it doesn't exist a significant difference between both controllers. These results confirm the variability in the response of the EBC with respect to the periodic controller: for the third change in the reference signal the values obtained for the EBC are different from those obtained for $t = 15$ y $t = 20$ seconds.

VII. CONCLUSION

In this paper it has been presented the implementation of event based controllers using the IEC-61499 standard for programming distributed control systems. Furthermore, it has been developed a study where it has been taken into account aspects related to the computational cost of algorithms and also to the response of the control system.

Regarding to the computational cost, it has been obtained a model that allows to compare systematically event based controllers with periodic controllers. This model, relates the event generation rate of an event-based controller with the execution time percentage of a periodic controller used by the time-based algorithm. This model, allow us to know the savings in terms of execution time that is obtained for a low event generation rate or the temporal overcharge for a high event generation rate, taking as a reference the execution time of a periodic controller.

It has been demonstrated experimentally the viability of the IEC-61499 standard for implementing event-based control algorithms. To that end, a platform for carrying out experiments considering systems with different characteristics in its dynamic response has been developed.

REFERENCES

[1] S. Dormido, J. Sánchez, and E. Kofman, "Muestreo, control y comunicación basados en eventos," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 5, no. 1, pp. 5–26, 2008.

[2] A. Zoitl and R. Lewis, *Modelling control systems using IEC 61499*. IET, 2014, vol. 95.

[3] K.-H. John and M. Tiegelkamp, *IEC 61131-3: programming industrial automation systems: concepts and programming languages, requirements for programming systems, decision-making aids*. Springer Science & Business Media, 2010.

[4] K.-E. Årzén, "A simple event-based pid controller," in *Proc. 14th IFAC World Congress*, vol. 18, 1999, pp. 423–428.

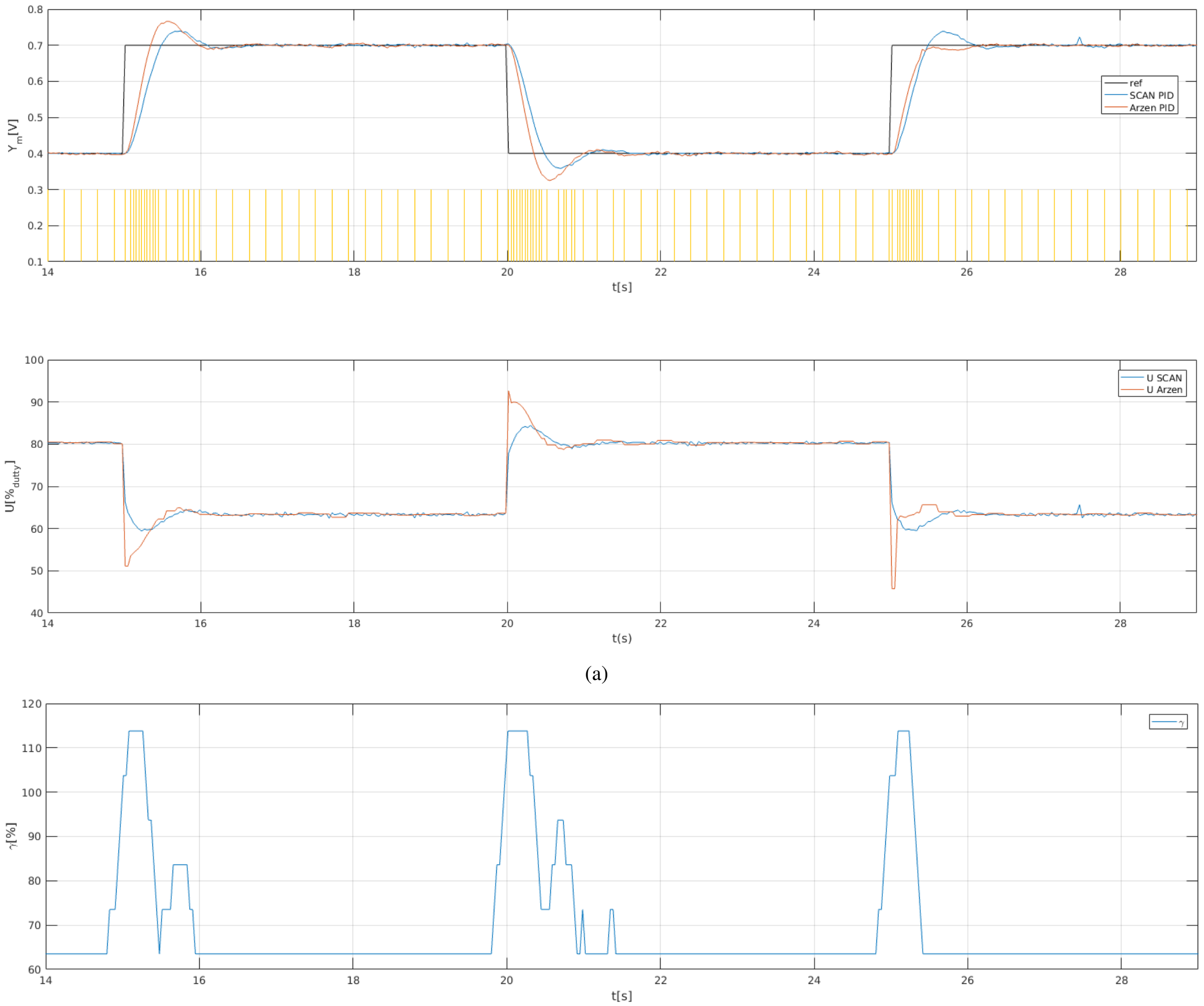
[5] S. Durand and N. Marchand, "An event-based pid controller with low computational cost," in *8th International Conference on Sampling Theory and Applications (SampTA'09)*, 2009, pp. Special–session.

[6] V. Vasyutynskyy and K. Kabitzsch, "A comparative study of pid control algorithms adapted to send-on-delta sampling," in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 3373–3379.

[7] S. Durand and N. Marchand, "Further results on event-based pid controller," in *Control Conference (ECC), 2009 European*. IEEE, 2009, pp. 1979–1984.

[8] V. Vasyutynskyy and K. Kabitzsch, "Time constraints in pid controls with send-on-delta," *IFAC Proceedings Volumes*, vol. 42, no. 3, pp. 48–55, 2009.

[9] K. Åström and T. Hägglund, "Revisiting the Ziegler?Nichols step response method for PID control," *Journal of Process Control*, vol. 14, no. 6, pp. 635–650, sep 2004.



(a)

(b)

Fig. 7. Figure (a) shows the response of the system (upper graph) and control output (lower graph) facing changes in the reference controlled with a periodic PID and with the EBC PID proposed by Årzén. Vertical yellow lines in upper graph represent generated events by Årzén's controller. Figure (b) shows the ratio between the time consumed by the and the Årzén's algorithm and the periodic PID (γ).