IFAC Conference on Advances in PID Control
PID'12
Brescia (Italy), March 28-30, 2012 ThB2.6

On event-based PI control of first-order processes

Ubaldo Tiberi, **José Araújo**, **Karl Henrik Johansson**

ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail:{ubaldot,araujo,kallej}@ee.kth.se)

Abstract: In this paper the design of an event-based proportional-integral (PI) control scheme for stable first-order processes is considered. A novel triggering mechanism which decides the transmission instants based on an estimate of the PI control signal is proposed. This mechanism addresses some side-effects that have been discovered in previous event-triggered PI proposals, which trigger on the process output. In the proposed scheme, the classic PI controller is further replaced with PIDPLUS, a promising version of PI controller for networked control systems. Although PIDPLUS has been introduced to deal with packet losses and time delays, and, to the best of our knowledge, a stability analysis of the closed-loop system where such a controller is used has never been performed, here the performance of such a controller in an event-based fashion are analyzed, and a stability analysis is further provided. The proposed event-based scheme ensures set-point tracking and disturbance rejection as in classic time-periodic implementations of PI controller, while greatly reducing the number of sensor transmissions. The theoretical results are validated by simulations, where the benefits in using PIDPLUS in combination with the proposed PI event-based triggering rule are shown.

Keywords: PI controller, Event-Based Control, Networked Control Systems (NCS), Adaptive Control.

1. INTRODUCTION

The proportional-integral-derivative (PID) controller has been applied to solve many control problems. Even though many controller choices are currently available, PID controllers are still by far the most widely used form of feedback control. In process industry it is know that more than 90% of the control loops are regulated by PID controllers, Åström and Hägglund [2006]. Most of such controllers are Proportional-Integral (PI), since the derivative part is usually not used in practice, Åström and Hägglund [2006]. In traditional control schemes, the implementation of PI controllers has always been performed by assuming that sensing, computation and actuation are performed periodically. However, with the introduction of networked control systems (NCSs), classic design techniques may no longer used. This originates from the fact that the network may introduce large communication delays and loss of information, which greatly influences the controller performance (see Eriksson [2008] for an overview of design methods of PID controllers for NCSs). Additionally, when the network is wireless, the control system designer should take into account bandwidth usage and energy consumption in the control loop design, Willig [2008]. Hence, new controller structures and PI tunings methods are required.

To cope with these problems, event-based techniques for control were recently introduced, Åström and Bernhardsson [1999], Tabuada [2007]. Such techniques allow an efficient utilization of the network resources, while ensuring a desired behavior of the closed-loop system. This is achieved by exchanging information between sensor, controllers and actuators only when relevant information is available. The use of these techniques has attracted much attention from the area NCSs, Wang and Lemmon [2011], and applied for PI or PID control Vasyutynskyy and Kabitzsch [2007], Rabi and Johansson [2008], Durand and Marchand [2009], Sánchez et al. [2011], Lehmann [2011]. However, no definite solution was yet achieved for eventual implementations in real industrial systems, because the utilization of such a control paradigm may lead to several problems as we discuss next.

The implementation of improper sampling techniques and controller structures may give rise to large oscillations of the process output, as observed by Årzen [1999], Cervin and Åström [2007], Vasyutynskyy and Kabitzsch [2007], Durand and Marchand [2009]. A common conclusion of previous studies has been that a large variation of integral component of the controller, due to long time-intervals between control updates, appear to be the cause of large oscillations. Another potential issue arising from the implementation of event-based PI controllers is that a *sticking* effect may occur. This effect is characterized by the absence of new events, even when the plant output is far from the desired set point, incurring in a non-zero steady-state error and no more controller updates.

Previous works have proposed methods to enhance event-based PI controllers performance. In Årzen [1999], Vasyutynskyy and Kabitzsch [2007], Rabi and Johansson [2008] and Durand and Marchand [2009] several sampling methods and controller adaptations were proposed to improve the transient performance of event-based PID controllers. Sánchez et al. [2011] and Lehmann [2011] proposed event-based PI controllers that rely on the knowledge of the plant model for sampling and control. However, in an industrial perspective, the derivation of an accurate mathematical model of the process may be expensive, thus such requirements may not be met in real implementations. Moreover, in all the aforementioned papers, the steady-state analysis has not been addressed. However, when an event-based control scheme is used, the design presents two degrees of freedom: one is represented by the choice of the sampling rule, and the other is represented by the choice of the controller.

In this paper, a novel event-based control scheme for stable first-order processes controlled by PI controllers is presented. The proposed scheme aim at canceling both the oscillations around the set-point and the sticking effect by jointly considering an appropriate event-based rule and an adaptive PI controller. More precisely, we consider PIDPLUS, Song et al.

[1] The work of the authors was supported by the by the VINNOVA project WiComPI, the Knut and Alice Wallenberg Foundation and the FeedNetBack and Hycon2 EU projects.

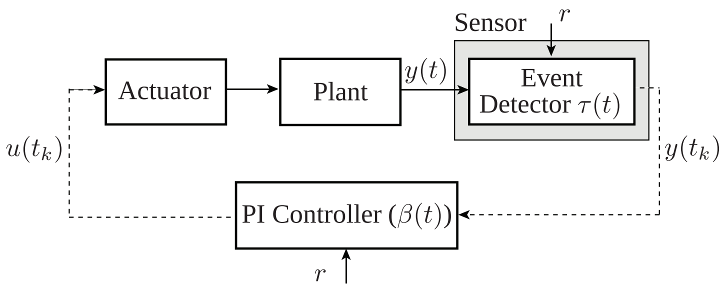


Fig. 1. The proposed event-based PI control system architecture

[2006] in combination with a PI-based triggering rule that schedule the measurement transmissions from the sensor to the controller. Although PIDPLUS was introduced to deal with packet losses and time delays, and its stability properties and performances are tested mainly by simulation or with experimental test-beds, Ungan [2010], Kaltiokallio et al. [2010], here we adopt such a controller in an event-based fashion, further providing stability conditions in a rigorous way.

The remainder of the paper is organized as follows. In the next section the control system architecture we consider is presented. In Section 3 potential issues in using event-based techniques in PI controlled systems are discussed. In Section 4 we introduce the proposed event-triggered PI controller, and in Section 5 we investigate its stability properties. In Section 6 the theoretical results are validated by simulation, and finally, a discussion in Section 7 concludes the paper.

2. EVENT-BASED PI CONTROL SYSTEM

The system architecture for the event-based PI control we consider is depicted in Fig. 1. It comprises a plant, a sensor, a PI controller and an actuator. The sensor continuously measures the plant output, and it has an event-detector implemented on-board. The event-detector decides when the plant output should be transmitted to the controller. The controller generates the input signal based on the received measurement from the sensor, which is then sent to the actuator and it is applied to the plant. We assume that a new control signal is computed and the actuator is updated at the same time and correspondently to the reception of a new output measurement.

The process is a first-order system of the form:

$$\begin{aligned} \dot{x}_p(t) &= a x_p(t) + b u(t)\,, \\ y(t) &= x_p(t)\,, \end{aligned} \tag{1}$$

where $x_p(t) \in \mathbb{R}$ is the process state, $u(t) \in \mathbb{R}$ is the control signal and $y(t) \in \mathbb{R}$ is the process output. We assume asymptotic stability of the process, i.e., $a < 0$. The controller is a PI controller, for which we consider its digital implementation given by

$$\begin{aligned} x_c(t_{k+1}) &= x_c(t_k) + \beta(t_k)(r - y(t_k))\,, \\ u(t_k) &= K_p\left((r - y(t_k)) + \frac{1}{T_i}x_c(t_k)\right)\,, \end{aligned} \tag{2}$$

where $x_c \in \mathbb{R}$ is the integrator state, $K_p \in \mathbb{R}$ is the proportional gain, $T_i \in \mathbb{R}$ is the integration time, $\beta(t_k)$ represents the integrator update rate, $r \in \mathbb{R}$ is the reference signal assumed to be constant and t_k is the time instant in which a new control input is computed. In traditional implementations, the parameter $\beta(t_k)$ is constant and equal to the sampling interval.

In event-based schemes, it is common to define an event as the violation of a triggering rule having the following form

$$\tau(t) \leq \delta\,,$$

where $\delta > 0$ and $\tau : \mathbb{R}^+ \to \mathbb{R}^+$ is a function that is reset to zero at any sampling-time, i.e. $\tau(t_k^+) = 0$, where t_k is the event instant defined as

$$t_k = \min\{t : t \geq t_{k-1} | \tau(t) > \delta\},$$

and where $h_k := t_{k+1} - t_k$ is the *inter-event time.* The function $\tau(t)$ generally depends on the accessible variables of the control system, like the output or the input signal, and the joint selection of $\tau(t)$ and δ encodes the desired behavior of the closed-loop system.

Given this setup, the problem we address in this paper is formalized as follows:

Problem 2.1. Given the process (1) controlled by a PI controller of the form (2), determine:

(1) an aperiodic event-based sampling rule $\tau(t) \leq \delta$;
(2) a dynamic integral update rule $\beta(t_k)$;

such that the closed-loop system exhibits zero steady-state error for constant reference signal and eventual constant external disturbances, while reducing the number of transmissions from the sensor node to the controller as much as possible. ◁

3. ISSUES WITH A NAIVE EVENT-BASED PI CONTROLLER

When PI controllers are used in event-based fashion, we may have the sticking effect or we may experience unacceptable oscillations of the output around the set-point. To illustrate these drawbacks we give an example. Consider the process (1) controlled by (2), with $a = -0.7, b = 1, K_p = 0.23, T_i = 3$, and consider the triggering implicitly defined by

$$\tau(t) = |y(t) - y(t_k)| \leq \delta\,, \tag{3}$$

where $\delta = 0.03$. We compare this event-based implementation with a periodic implementation of period $h = 0.3$ s.

3.1 *Sticking effect*

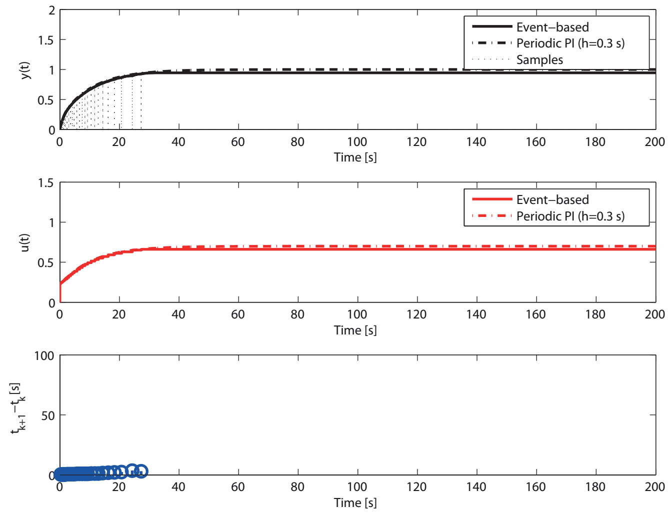
Figure 2(a) represents the step response and the inter-event times. After $t = 25$ s, the controller is no longer updated, despite there is a steady-state error of $\simeq 0.15$, and the system gets stuck. The benefit of achieving zero steady state error for constant references offered by continuous or periodic discrete-time PI controllers are clearly lost. This is due because the process is stable and the constant control inputs applied between sensor transmissions are not strong enough to fulfill condition (3).

In the sequel we refer the problem of having steady-state error in addition to do not performing any control updates as *sticking*. As the reader may argue, a trivial method to avoid sticking is to add a time out to the sampling rule, so that the sensor is enforced to send a new measurement to the controller whenever the closed-loop system gets stuck. Nonetheless, several problems arise also by adopting this simple trick, as we discuss next.

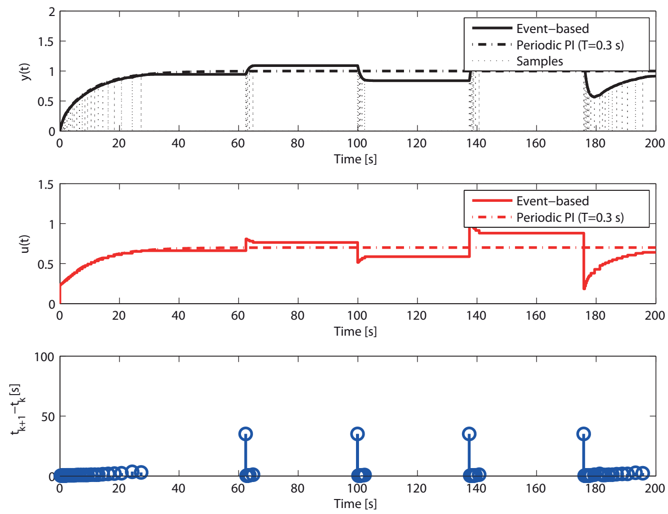
3.2 *Oscillatory behavior*

Consider again the previous example, and, in addition to the event rule defined by (3), consider a time-out that enforce the sensor to send a packet if no events are detected for $h_{\max}$ units of time. Such time-out is added to the sampling rule (3) to avoid sticking. In the example we set $h_{\max} = 35$ s. The output response is depicted in Figure 2(b). By considering the event-based rule (3) plus the time-out, we obtain a large output oscillations around the set-point.

Such oscillations are due to the large value of $h_{\max}$. Because the integrator update rate $\beta(t_k)$ is equal to the inter-event times h_k,



(a) Sticking effect



(b) Oscillations around the set-point

Fig. 2. Issues with a naive event-based PI controller.

if $h_{\max}$ is too large so it is $\beta(t_k)$. Then, the control input input applied at time $t_{k+1} = t_k + h_{\max}$ may be too aggressive and it potentially triggers an oscillatory behavior of the output. On the other hand, by choosing small values for $h_{\max}$ to reduce the strong variation of the integral state, we may lose the benefits of using an event-based scheme since more transmissions are required.

The design of an event-based PI controller presents then two degrees of freedom: one degree of freedom is represented by the selection of an appropriate design of the integrator update rate $\beta(t_k)$, and the other concerns the choice of a suitable event-based rule $\tau(t) \leq \delta$.

4. PROPOSED EVENT-BASED PI CONTROL SCHEME

To solve the sticking problem, we need to enforce the sensor to send a new measurement to the controller whenever the system gets stuck, and we have to adapt the controller to avoid oscillations around the set-point. These are the arguments of the following two sections.

4.1 PI-based triggering rule

Inspired by the deadband triggering-rule based on the *output* signal Otanez et al. [2002], our intuition is to consider the deadband triggering rule on an appropriate *filtered version of the output* signal. For instance, by considering the control input as a filtered version of the output, our idea is to consider a deadband sampling on a input-like signal. By denoting

$$\tilde{u}(t) = \tilde{K}_p \left((r - y(t)) + \frac{1}{\tilde{T}_i} \int_{t_k}^{t} (r - y(s))\, ds + \tilde{x}_c(t_k) \right), \tag{4}$$

where $\tilde{K}_p, \tilde{T}_i \in \mathbb{R}$ are two sampling parameters, and $\tilde{x}_c \in \mathbb{R}$ is the state of the integrator implemented on the sensor, the event-based rule we propose is implicitly defined by the condition

$$\tau(t) = \left| \tilde{u}(t_k) - \tilde{K}_p \Big((r - y(t)) - \frac{1}{\tilde{T}_i} \int_{t_k}^{t} (r - y(s))\, ds - \tilde{x}_c(t_k) \Big) \right| \leq \delta, \tag{5}$$

We denote (5) as *PI-based triggering rule*. Whenever the system gets stuck, the integral term in (5) grows unbounded, enforcing the sensor to send a new measurement to the controller, and the sticking is avoided. Moreover, the controller is no longer updated if and only if $y_{k^*} = r$, that yields $\tilde{u}(t_{k+1}) = \tilde{u}(t_k)$ for all k greater than k^*. That way, when the system gets stuck, the sampling rule imposes a time-out that depends on the distance of y_k from the desired set-point r.

Notice that by using such a triggering rule, the sensor can potentially compute the new control input $\tilde{u}(t_{k+1})$ and send this information straight to the actuator. However, whenever the utilization of (5) cancels the sticking problem, the controller would be updated with $\tilde{u}(t_{k+1}) = \tilde{u}(t_k) \pm \delta$. This control update rule leads to limit cycles that may generate unacceptable oscillations of the output. To avoid such oscillations, we let the sensor to verify when (5) is violated, and we let it to transmit to the controller the value of $y_p(t_k)$ instead of the value of $\tilde{u}(t_{k+1})$. The controller updates the input signal according to the received measurement $y_p(t_k)$ and to the elapsed inter-event times.

Remark 4.1. In general, it is possible to use completely different tuning of the parameters $\tilde{K}_p$ and $\tilde{T}_i$ at the sensor and K_p and T_i at the controller. This fact should not provide any concerns, because constraining the PI sampler and the PI controller parameters to have the same tuning, then the sensor manufactures would be constrained to produce ad-hoc sensors depending on which controller is used by a certain customer. ◁

4.2 Integrator update rate adaptation

The integration update rate $\beta(t_k)$ that we use is the same as PIDPLUS Song et al. [2006]. After some calculation, we get that the PIDPLUS can be rewritten in the form (2), where

$$\beta(t_k) = -T_i (1 - e^{\frac{h_{k-1}}{T_i}}). \tag{6}$$

In the sequel we consider the formulation of the PIDPLUS as (2), with $\beta(t_k)$ defined as in (6), and we will show that to guarantee asymptotic stability of the closed-loop system a condition on the controller's proportional gain K_p must be fulfilled.

5. STABILITY ANALYSIS

In this section we study the stability property of the controlled process when $\beta(t_k)$ is chosen according to (6), and $\tau(t) \leq \delta$ is chosen according to (5). Before addressing the general case of aperiodic controller updates, we first show how the adaptation (6) ensures asymptotic stability of the controlled-process for any fixed controller update rate of period $h > 0$.

Lemma 5.1. Consider the system (1) controller by (2), where $\beta(t_k)$ is given by (6). Then, the controlled system is asymptotically stable for any constant sampling interval $h > 0$ if, and only if $0 < -bK_p/a < 1$. ◁

Proof: We start the proof by assuming $r = 0$. To capture all the model details of the closed loop system, let us denote $x = [x_p \quad x_c]^T$, and consider the hybrid model, Goebel et al. [2009]

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + w\,, \text{ if } x \in \mathcal{C}\,, \\ x(t_{k+1}) &= A_d(\beta(t_k))x(t_k)\,, \text{ if } x \in \mathcal{D}\,, \end{aligned} \tag{7}$$

where

$$A_c = \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix}, w = \begin{pmatrix} -bK_p x_p(t_k) \\ 0 \end{pmatrix} \tag{8}$$

$$A_d(\beta(t_k)) = \begin{pmatrix} 1 & 0 \\ -\dfrac{K_p}{T_i}\beta(t_k) & 1 \end{pmatrix}, \tag{9}$$

and where the flow and the jump sets are defined respectively as $\mathcal{C} = \{x \in \mathbb{R}^2 : \tau(t) \le \delta\}$ and $\mathcal{D} = \{x \in \mathbb{R}^2 : \tau(t) > \delta\}$, where $\tau(t)$ and δ are defined in (5). Given the system (7), we want to study the stability property of the origin. Under the assumption $a < 0$, and because the input w is constant and it acts only on the asymptotically stable part of (7), i.e. it acts only on the dynamics of x_p, then $\|x(t)\| \le \alpha(\|x(t_k)\|)$ for all $t \in (t_k, t_{k+1})$, and for a certain $\mathcal{K}$-class function $\alpha(\cdot)$ [2], see Fig. 3. To achieve asymptotic stability of the origin, it is then enough to show the convergence to zero of the sequence $x(t_k)$. Because it holds $x_p(t_{k+1}^+) = x_p(t_k)$ and $\dot{x}_c(t) = 0$, we can study the behavior of $x(t_k)$ only correspondently to the transmissions times $t = t_k$. By observing the particular structure of (7), it is enough to study the stability properties of the following system

$$x(t_{k+1}) = \Phi x(t_k)\,, \tag{10}$$

where the matrix

$$\Phi = \begin{pmatrix} e^{ah} - K_p\dfrac{b}{a}(e^{ah}-1) & \dfrac{b}{a}(e^{ah}-1) \\ -\dfrac{K_p}{T_i}\beta(t_k) & 1 \end{pmatrix}, \tag{11}$$

is obtained by considering the exact discretization of the continuous time process (1) controlled by (2) and by considering constant time intervals of the form $[t_k, t_{k+1})$. Note that under constant sampling, the matrix Φ is time invariant. The polynomial characteristic of Φ is given by

$$p(\lambda) = \lambda^2 - \mathrm{tr}(\Phi)\lambda + \det(\Phi)\,,$$

where $\mathrm{tr}(\Phi)$ and $\det(\Phi)$ denote the trace and the determinant of the matrix Φ respectively. By applying the Jury criterion we get the following necessary and sufficient conditions for asymptotic stability

1. $|\det(\Phi)| < 1$,
2. $1 - \mathrm{tr}(\Phi) + \det(\Phi) > 0$,
3. $1 + \mathrm{tr}(\Phi) + \det(\Phi) > 0$.

By using PIDPLUS, the above conditions become

1. $|e^{ah} - \frac{b}{a}K_p e^{\frac{h}{T_i}}(e^{ah}-1)| < 1$,
2. $\frac{b}{a}K_p(e^{ah}-1)(1-e^{\frac{h}{T_i}}) > 0$,
3. $2(1+e^{ah}) - \frac{b}{a}K_p(e^{ah}-1)(1-e^{\frac{h}{T_i}}) > 0$,

[2] A continuous function $\alpha : [0, a) \to \mathbb{R}_{\ge 0}, a > 0$, is said to be of class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$, Khalil [2002].

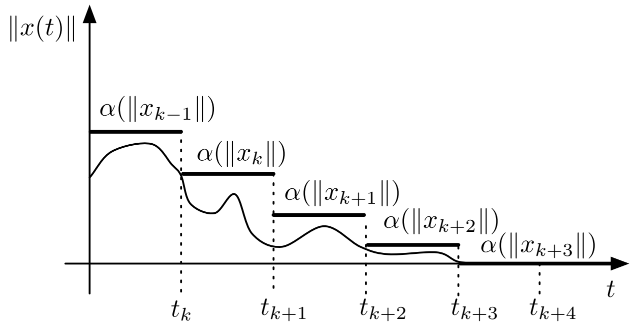


Fig. 3. Stability analysis of the closed-loop system. The value of $\|x(t)\|$ is bounded by a decreasing bound $\alpha(\|x_k\|)$ in any time interval $[t_k, t_{k+1})$.

Condition 2 is always verified, while conditions 1–3 are verified for all $h > 0$ if, and only if $0 < -bK_p/a < 1$. Then, by assumption, we have that Φ is Schur for any constant time $h > 0$. The Schur property of Φ implies the convergence to zero of the sequence $x(t_k)$, and since $\|x(t)\| \le \alpha(\|x(t_k)\|)$ for $t \in (t_k, t_{k+1})$, then asymptotic stability of the origin follows.

In the case $r \ne 0$, we have that the equilibrium point of the controlled system is in $x_{eq} = [r \quad -aT_i/(K_p b)\cdot r]^T$. Hence, the set-point tracking can be studied by considering a coordinate change that translates x_{eq} into the origin, and then by studying the stability of the origin in the new coordinates as done in the case $r = 0$. When there are external disturbances, it is possible to proceed in the same way by considering $x_{eq} = [r \quad -aT_i/(K_p b)\cdot r - d]^T$ as equilibrium point. ◇

It is well known that while a common method to test instability is to verify the position of the eigenvalues of the discretized controlled process outside the unit circle, to prove stability of the continuous-time process, this method should be used with caution because no information is given on what occurs between consecutive discretization instants. It can happen for example that at every discretization instant it holds $x_p(t_k) = 0$ but the output is oscillating between the discretization points. However, by resorting to hybrid models, we proved that between two consecutive sampling instants, the continuous-time dynamics are also upper bounded with a class $\mathcal{K}$ function $\alpha(\cdot)$, and such a bound converges to zero, see Fig. 3.

Remark 5.1. The reader may argue that since PIDPLUS ensures asymptotic stability for any constant sampling period, a natural method to reduce the amount of communication between the sensor and the controller would be to use a large constant sampling period. However, if a disturbance suddenly enters the system, the performance may drastically deteriorate since it will be detected only at the next sampling instant that may be far, while an event-based control scheme would react immediately. ◁

In the case of event-based control, the stability analysis is more involved, since the inter-event times are varying, namely the sampling intervals h_k are not constant. This implies that the matrices Φ are time-varying, and the controlled system can be rewritten as

$$x(t_{k+1}) = \Phi(h_k, h_{k-1})x(t_k)\,, \tag{12}$$

where

$$\Phi(h_k, h_{k-1}) = \begin{pmatrix} e^{ah_k} - K_p\dfrac{b}{a}(e^{ah_k}-1) & \dfrac{b}{a}(e^{ah_k}-1) \\ -K_p\Big(1 - e^{\frac{h_{k-1}}{T_i}}\Big) & 1 \end{pmatrix}. \tag{13}$$

However, by using PIDPLUS, it is still possible to verify the stability condition under any event-based rule, as stated in the next result.

Theorem 5.1. Consider the system (12), and assume $0 < -bK_p/a < 1$. Let $0 < h_{\min} < h_{\max}$ two arbitrary positive constants. If there exists a matrix $P = P^T > 0$ that sastisfies

$$\Phi(h_{\min}, h_{\min}) P \Phi^T(h_{\min}, h_{\min}) - P < 0, \quad (14)$$

$$\Phi(h_{\min}, h_{\max}) P \Phi^T(h_{\min}, h_{\max}) - P < 0, \quad (15)$$

$$\Phi(h_{\max}, h_{\min}) P \Phi^T(h_{\max}, h_{\min}) - P < 0, \quad (16)$$

$$\Phi(h_{\max}, h_{\max}) P \Phi^T(h_{\max}, h_{\max}) - P < 0, \quad (17)$$

then, the origin is asymptotically stable for any $h_{\min} \leq h_k \leq h_{\max}$. ◁

Proof: The system (12) can be viewed as a discrete-time system with time-varying uncertainty. Let the set $\mathcal{A} := \{\Phi(h_k, h_{k-1}) | h_{\min} \leq h_k \leq h_{\max}\}_{k \in \mathbb{N}}$. It is easy to verify that every matrix that belongs to $\mathcal{A}$ can be expressed as convex combination of the four matrices $\Phi(h_k, h_{k-1})$ obtained when h_k, h_{k-1} are equal either to $h_{\min}$ or $h_{\max}$, and then the uncertainty is polytopic. Hence, asymptotic stability of (12) is achieved if there exists a matrix P that satisfies (14)–(17), see for example Amato [2006]. ◇

The previous theorem states that the PIDPLUS controller ensures asymptotic stability of the closed-loop system, no matter when the controller is updated, provided that $h_{\min} \leq h_k \leq h_{\max}$. Then, the set-point is asymptotically tracked and eventual undesirable oscillations around it are canceled if the controller receives an infinity number of measurements. Hence, according to Theorem 5.1, we have to slightly modify the sampling rule (5) by using $h_k = h_{\min}$ or $h_k = h_{\max}$ if the inter-sampling times given by (5) are too short or too long respectively.

The fact that an infinite number of measurements are required to achieve asymptotic tracking of the set-point should not mislead about the efficiency of the proposed control scheme. The efficiency of the proposed method relies on the fact that the transmissions can be performed at any time, provided that $h_{\min} \leq h_k \leq h_{\max}$, where $h_{\max}$ can be very large.

6. SIMULATION RESULTS

We illustrate the performance of the proposed scheme when controlling a first-order plant of the form (1). Then, just for the sake of investigation, we simulate the case in which there is a time delay between the controller and the actuator. For steady-state condition analysis we look at the number of transmitted packets N from the sensor to the controller. Moreover, we use the Integral of the Absolute Error (IAE) parameter as a general indicator for both transient and steady-state performance. The IAE value is calculated as

$$\text{IAE} = \int_0^\infty |r - y(s)| ds. \quad (18)$$

The simulations were performed using Simulink in combination with Truetime, Cervin et al. [2003].

6.1 Example 1: First-order process

We consider the same setup as in Section 3 and we compare the performance of the proposed scheme with a periodic implementation of period $h = 0.3$ s. The PI-based sampling scheme is set with K_p and T_i as for the controller. Note that the assumptions of Lemma 5.1 are satisfied.

For this system we find the matrix $P = \begin{pmatrix} 0.0956 & -0.090 \\ -0.090 & 0.1944 \end{pmatrix}$ to fulfill Theorem 5.1 for $h_{\min} = 0.3$ s and $h_{\max} = 10^{10}$ s. The results are depicted in Figure 4. By comparing Figure 4 with Figure 2, we can appreciate how there are no oscillations of the output, and how the system does not stick. Within this simulation we obtained $N = 28$ number of transmissions and IAE=13.45 for the event-based implementation, while the periodically sampled PI controller generated $N = 1500$ transmissions and IAE=8.97. In order to generate the same number of transmissions as the event-based, the periodically sampled PI controller requires a transmission period $h = 16$ s which would render the closed-loop system unstable. However, if using the PIDPLUS with period $h = 16$ s, the system is stable according to Lemma 5.1, but its transient response is slower than the proposed scheme, with an IAE=22.31 s.

We also test the disturbance rejection of the proposed scheme. The simulation result is illustrated in Figure 5, where a disturbance of amplitude 0.2 is added to the input of the plant at $t = 200$ s. As it can be seen, the proposed scheme efficiently rejects the disturbance with a small number of samples. With our method we experienced $N = 47$ transmissions with an IAE=23.08, while the periodically sampled PI with $h = 0.3$ achieves an IAE=15.4 with $N = 1500$ number of transmissions. The PIDPLUS with period $h = 16$ s is able to reject the disturbance, but its response is slow, where the disturbance being only detected at approximately $t = 205$ s, 5 s after it occurred. Moreover, it generated $N = 28$ transmissions with an IAE=36.40.

6.2 Example 2: First-order process with delay

We now evaluate our event-based scheme for the control of a first-order process with delay. We remark that the stability analysis provided in Section 5 are no longer valid for a plant with delay. This example serves the purpose of demonstrating the robustness of the proposed control scheme to delays.

We consider the same example as above, with the addition of an actuation delay of 5 s. The results are depicted in Figure 6. As it can be seen, the response becomes oscillatory with the introduction of the actuation delay, whereas the proposed control scheme successfully tracks the set-point and rejects the disturbance. Moreover, the performance is very close to the periodically sampled PI controller with $h = 0.3$, where our scheme provided IAE=26.41, while the periodic PI provided IAE=26.83. As an added benefit, the number of required transmissions is significantly reduced to 62.

7. CONCLUSIONS

When event-based techniques are used in PI control scheme, drawbacks as sticking or output oscillation may arise. To cope with these problems we proposed a novel event-based scheme that provides a PI-based triggering used in combination with PIDPLUS. Despite PIDPLUS was introduced to deal with network imperfections like packet losses and time delays, here we used such a controller in an event-based scheme, further analyzing the stability property of the closed-loop system. Simulations results show how the utilization of PIDPLUS in combination with the PI-based triggering rule is capable to achieve asymptotic set-point tracking and disturbance rejection as classic PI controller, while drastically reducing the number of transmissions from the sensor to the controller.

Future work include the extension to processes with delay and the extension to multi dimensional systems. Moreover, the effect of the derivative part of the PIDPLUS when used in an event-based scheme is worth of investigation. Finally, the optimal choice of δ in the PI-based triggering rule to achieve a trade-off between performance of the closed-loop system and number of transmission is another future research topic.

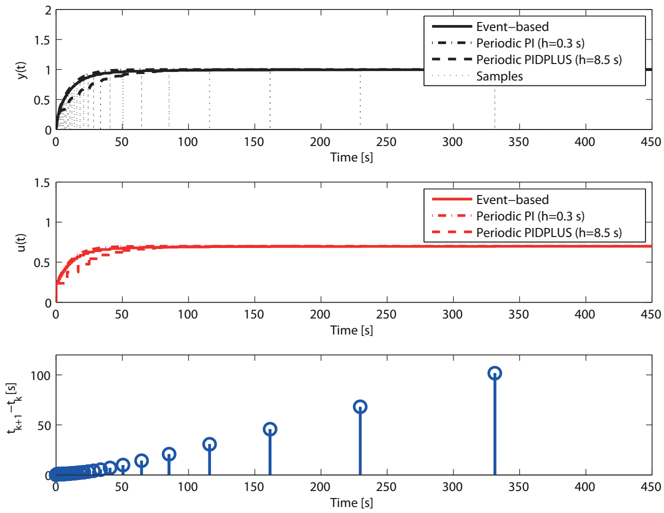


Fig. 4. Simulation result of a PIDPLUS controller with PI sampling in Example 1. The performance of the event-based controller is compared to a periodic implementation of a classic PI controller with period $h = 0.3$s and a PIDPLUS controller with period $h = 16$s.

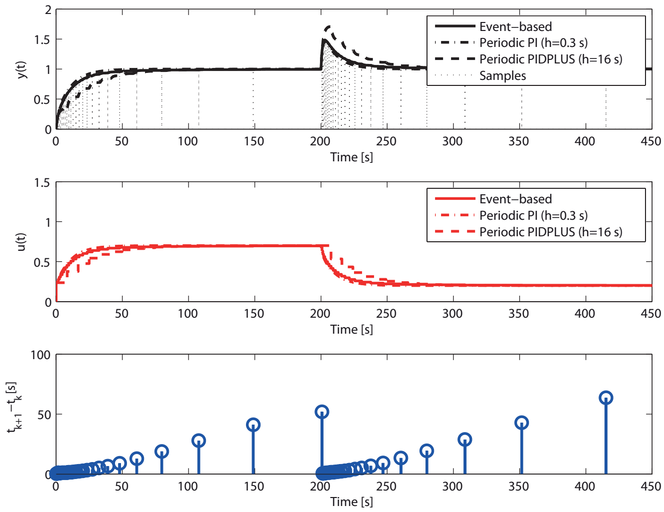


Fig. 5. Simulation result of a PIDPLUS controller with PI sampling for disturbance rejection. Performance comparisons as in Figure 4.

REFERENCES

F. Amato. *Robust Control of Linear Systems subject to Uncertain Time-varying Parameters*. Springer, 2006.

K. E. Årzen. A simple event-based PID controller. In *IFAC World Congress*, pages 423–428, Beijing, China, 1999.

K. J. Åström and B. Bernhardsson. Comparison of periodic and event-based sampling for first-order stochastic systems. In *IFAC World Congress*, 1999.

K.J. Åström and T. Hägglund. *Advanced PID Control*. ISA, NC, 2006.

A. Cervin and K.J. Åström. On limit cycles in event-based control systems. In *46th IEEE CDC*, pages 3190 –3195, Dec. 2007.

A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzen. How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime. *Control Systems, IEEE*, 23(3):16 – 30, june 2003. ISSN 1066-033X.

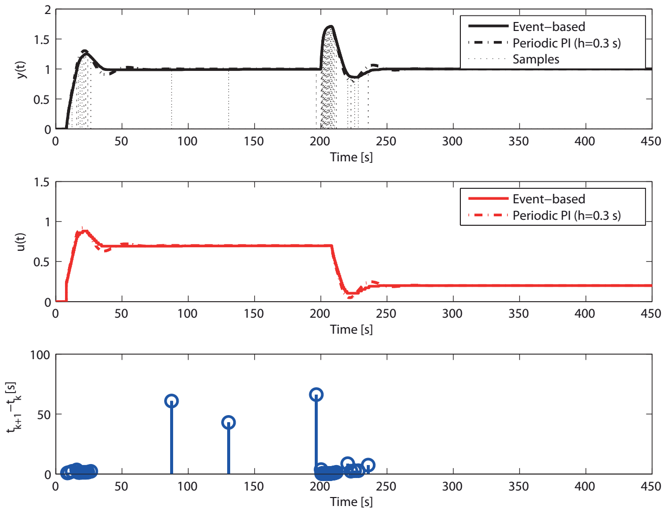


Fig. 6. Simulation result of a PIDPLUS controller with PI sampling for disturbance rejection under an actuation delay $L = 8$ s.

S. Durand and N. Marchand. Further Results on Event-Based PID Controller. In *Proceedings of ECC 2009*, pages 1979–1984, Budapest, Hongrie, August 2009.

L. Eriksson. *PID Controller Design and Tuning in Networked Control Systems*. PhD thesis, Helsinki University of Technology, 2008.

R. Goebel, R. Sanfelice, and A. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, 2009.

O. Kaltiokallio, L. M. Eriksson, and M. Bocca. On the performance of the pidplus controller in wireless control systems. In *18th MED*, pages 707 –714, Jun. 2010.

H. Khalil. *Nonlinear Systems*. Pretience Hall, 3^{rd} edition, 2002.

D. Lehmann. *Event-based state-feedback control*. PhD thesis, Ruhr-Universitat Bochum, 2011.

P. G. Otanez, J. R. Moyne, and D. M. Tilbury. Using deadbands to reduce communication in networked control systems. In *Americal Control Conference (ACC)*, volume 4, pages 3015–3020, 2002.

M. Rabi and K. H. Johansson. Event-triggered strategies for industrial control over wireless networks. In *International Conference on Wireless Internet*, pages 1–7, 2008.

J. Sánchez, A. Visioli, and S. Dormido. A two-degree-of-freedom pi controller based on events. *Journal of Process Control*, 21(4):639 – 651, 2011. ISSN 0959-1524.

J. Song, A. K. Mok, D. Chen, M. Nixon, T. Blevins, and W. Wojsznis. Improving pid control with unreliable communications. In *ISA EXPO Technical Conference*, 2006.

P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52, 2007.

V. Ungan. Networked PID controllers for wireless systems. Master's thesis, Royal Institute of Technology (KTH), November 2010.

V. Vasyutynskyy and K. Kabitzsch. Simple pid control algorithm adapted to deadband sampling. In *IEEE ETFA*, pages 932 –940, Sep. 2007.

X. Wang and M.D. Lemmon. Event-triggering in distributed networked control systems. *Automatic Control, IEEE Transactions on*, 56(3):586 –601, march 2011. ISSN 0018-9286.

A. Willig. Recent and emerging topics in wireless industrial communication. *IEEE Transactions on Industrial Informatics*, 4(2):102–124, 2008.