# Qualitative Analysis of a One-step Finite-Horizon Boundary for Event-driven Controllers

Manel Velasco, Pau Martí, José Yépez, Francisco J. Ruiz, Josep M. Fuertes and Enrico Bini

***Abstract*— Performance optimization for networked and embedded control systems refers to the ability of minimizing controllers' resource utilization and/or improving control performance. Event-driven control has been shown to be a promising technique for minimizing controllers' computational demands. However, optimization of control performance for event-driven control has not been fully addressed. For LTI plants, this paper presents a boundary for event-driven controllers that determines at each job execution when the next job execution should occur in order to minimize a continuous-time quadratic cost function while minimizing controllers' computational demand. Simulation results illustrate the qualitative shape of this boundary.**

## I. INTRODUCTION

Networked and embedded control systems are often designed under resource constraints such as processor capacity, communication bandwidth, battery life, etc. In order to make these systems cost-effective, it is mandatory to efficiently use the computational resources [1]. Since the computational load imposed by controllers depends on their rate of execution, it is of interest to study approaches to control systems design capable of producing controllers with low resource demands.

As repeatedly indicated in the research literature (e.g., [2] or [3]), although event-driven control systems lacks a system theory, it provides interesting benefits like reducing resource utilization. Although several theoretical results for event-driven control have recently appeared in the literature, (e.g., [4] or [5]), control performance guarantees for event-driven controllers have not been treated in the context of optimal control.

In [6] a preliminary attempt to jointly study resource utilization and control performance in an optimal control setting for event-driven controllers was made. The goal was to specify an event-driven control technique able to minimize cost while using the same amount of resources than in the periodic case. However, only a numerical receipt was provided. In [7] a comprehensive study was presented in terms of numerically assessing if non-periodic controller activation times could lead to a lower cost than in the periodic case. Finally, it is worth noting that in [8] an optimal control design methodology where adding the opportunity to update the control law at instants within the standard periodic ones was proved to only lead to an improvement of a quadratic cost function. Although the later approach results in an increase of resource utilization, it also combines non-periodic sampling and optimal control.

Inspired by the problem defined in [6] and the numerical results given in [7], this paper studies whether an optimal control performance problem that also takes into account minimization of resource utilization can be transformed into an event condition for event driven controllers. More specifically, and for the case of LTI plants, this paper presents a boundary for event-driven controllers, named "one-step" boundary, whose application at each controller execution tries to maximize the time when the next execution will occur while minimizing a finite-horizon quadratic cost function, and considering that the rest of controller executions will be periodic. Simulation results have been carried out to evaluate the one-step finite-horizon boundary controller in terms of resource utilization and control performance.

The rest of the paper is organized as follows. Section II formalizes the problem to be solved, whose solution is developed in Section III. Section IV presents detailed simulation results. Finally, Section V concludes this paper.

## II. PROBLEM TO BE SOLVED

We consider the control system

$$\begin{array}{rcl} \dot{x}(t) & = & A\,x(t) + B\,u(t) \\ y(t) & = & C\,x(t) \end{array} \tag{1}$$

with $x \in \mathbb{R}^{n\times 1}$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $u \in \mathbb{R}^{m\times 1}$, and $C \in \mathbb{R}^{1\times n}$.

Let

$$u(t) = u_k = L\,x(t_k) = L\,x_k \qquad \forall t \in [t_k, t_{k+1}) \tag{2}$$

be the control updates given by a linear feedback controller $L$ designed in the discrete-time domain using only samples of the state at discrete instants $t_0, t_1, \ldots, t_k, \ldots$. We refer to $t_k$ as the controller activation times. Between two consecutive control updates, $u(t)$ is held constant. In periodic sampling we have $t_{k+1} = t_k + h$, where $h$ is the period of the controller.

In event-driven control, the sequence of activation times $t_k$ occur when the system trajectory crosses a boundary that specifies the tolerated threshold where the system trajectory can move without requiring control actions. Event conditions can be generalized by introducing a function $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ that defines a boundary measuring the tolerated error with

This work was partially supported the Spanish EVENTS project, CICYT DPI2010-18601, and by the EU NoE Design for Embedded Systems, IST-2008-214373

M. Velasco, P. Martí, J. Yépez, F.J. Ruiz and J.M. Fuertes are with the Automatic Control Department, Technical University of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain {manel.velasco, pau.marti, jose.yepez, francisco.javier.ruiz, josep.m.fuertes}@upc.edu

E. Bini is with Scuola Superiore Sant'Anna, Piazza Martiri della Libertà 33, 56127 Pisa, Italy e.bini@sssup.it

respect to the sampled state [9]. The condition that must be ensured is

$$f(x(t), x_k) \leq \eta \quad t \in [t_k, t_{k+1}) \tag{3}$$

where $\eta$ is the error tolerance. Typical prerequisites for $f$ are: $f$ be continuous, $f(x,x) = 0$.

Control performance is often measured by an infinite-horizon continuous-time quadratic cost function

$$J = \int_0^\infty \left[x^T(t)Q_c x(t) + u^T(t)R_c u(t) + 2x^T(t)N_c u(t)\right] \mathrm{d}t \tag{4}$$

where the weighting matrices $Q_c$ and $R_c$ are symmetric positive semidefinite matrices.

If the gain $L$ in (2) is designed using periodic linear quadratic (LQR) control for a given $h$, the minimum cost of (4) is [10]

$$J^* = x_0^T S(h) x_0 \tag{5}$$

where $S(h)$ is the solution to the discrete algebraic Riccati equation of the discretized system with period $h$, and $x_0$ is a given initial state.

Rather than applying periodic control with period $h$, we are interested in assessing whether a different sequence of activation times can demand less resource utilization than the periodic case without degrading the performance, assuming that the controller gain $L$ designed for the periodic scenario is left unchanged. Without loss of generality, let us set the latest sampling instant $t_0$ equal to 0 and $x_0$ the sampled state at $t_0 = 0$. We aim at computing the largest $\tau$ such that the sampling sequence

$$t_0 = 0, \quad t_1 = \tau, \quad t_k = \tau + (k-1)h$$

will provide a cost not larger than the cost $J^*$ of a periodic controller with period $h$ (of Equation (5)). In this problem, the time interval $\tau$ becomes the decision variable for achieving the desired operation at $\tau$. In addition, we need to make the cost (4) explicitly depending on $\tau$ and reflecting that after $\tau$ the controller activations will be periodic. To do so, we split cost (4) into two sub-costs $J_1$ and $J_2$. The first cost $J_1$ applies to the $\tau$ time interval, which is unknown and must be maximized, and the second cost applies from $\tau$ to $\infty$, where activations are expected to be periodic at the maximization time. Remembering that the controller gain $L$ is optimal for (4) given the sampling period $h$, cost (4) can be re-written as

$$J = J_1 + J_2 \tag{6}$$

where

$$J_1 = \int_0^\tau \left[x^T(t)Q_c x(t) + u_0^T R_c u_0 + 2x^T(t)N_c u_0\right] \mathrm{d}t \tag{7}$$

and

$$J_2 = x^T(\tau) S x(\tau). \tag{8}$$

Notice that in (7) the input is represented by $u_0$, since $u(t)$ is constant in $[0, \tau)$.

Hence, the boundary should permit to easy solve the optimization problem formally stated as

$$\begin{aligned} \text{maximize} \quad & \tau & (9)\\ \text{subject to} \quad & \frac{\partial J}{\partial \tau} = 0 & (10)\\ & J^* - J(\tau) > 0 & (11)\\ & \tau > 0 & (12)\\ & x(\tau) = \Phi(\tau)x_0 + \Gamma(\tau)u_0 & (13) \end{aligned}$$

where $u_0 = Lx_0$ and

$$\Phi(\tau) = e^{A\tau},\ \Gamma(\tau) = \int_0^\tau e^{As}\mathrm{d}sB,\ \tau \ \in [t_k, t_{k+1}). \tag{14}$$

Noting that the objective function (9) is continuous and the constraint set is compact (constraint (10) reduces the constraint set to a set of isolated points), the problem has solution.

Note that if the feasible domain given by restrictions (10)-(13) is an empty set, then we specify $\tau = h$.

## III. THEORETICAL APPROACH

Solving problem (9)-(13) implies two main steps: a) to find the set of positive isolated points $\tau_i$ that fulfill conditions (10) and (12) (among them, there will be the minimums of the cost function (6)); b) from those points, to choose the longest $\tau_i$ whose cost evaluated using (6) is lower than the cost (5) provided by the periodic controller, that is, fulfilling restriction (11).

### A. One-step Finite Horizon Optimal Boundary

The first step of the solution to the problem (9)-(13) is announced next.

*Proposition 1:* Given the closed-loop system (1)-(2) where $L$ is the discrete time LQR optimal gain minimizing (4) for a given sampling period $h$, the boundary condition that permits finding the set of isolated $\tau_i$ that fulfill constraint (10) for any arbitrary initial state $x_0$ considering that after $\tau$ time units the controller activation will be periodic is

$$\begin{bmatrix} x(\tau) & u_0 \end{bmatrix} \begin{bmatrix} \bar{Q} & \bar{N} \\ \bar{N}^T & \bar{R} \end{bmatrix} \begin{bmatrix} x(\tau) \\ u_0 \end{bmatrix} = 0 \tag{15}$$

where

$$\bar{Q} = \left[A^T S + SA + Q_c\right],\ \bar{N} = \left[N_c + SB\right],\ \bar{R} = R_c. \tag{16}$$

*Proof:* Since $J = J_1 + J_2$, we compute its derivative w.r.t. $\tau$, by computing the ones of $J_1$ and $J_2$ respectively.

$$\frac{\partial J_1}{\partial \tau} = x^T(\tau)Q_c x(\tau) + u_0^T R_c u_0 + 2x^T(\tau)N_c u_0 \tag{17}$$

$$\frac{\partial J_2}{\partial \tau} = \frac{\partial}{\partial \tau}x^T(\tau)Sx(\tau) + x^T(\tau)S\frac{\partial}{\partial \tau}x(\tau) \tag{18}$$

since $S$ does not depend on $\tau$. From the system dynamics (1), it follows that

$$\frac{\partial}{\partial \tau}x(\tau) = Ax(\tau) + Bu(\tau) = Ax(\tau) + Bu_0$$

Then it follows that

$$\begin{aligned}\frac{\partial J_2}{\partial \tau} &= x^T(\tau)A^TSx(\tau) + u_0^TB^TSx(\tau) \\ &\quad + x^T(\tau)SAx(\tau) + x^T(\tau)SBu_0 \\ &= x^T(\tau)(A^TS + SA)x(\tau) + 2x^T(\tau)SBu_0 \end{aligned} \tag{19}$$

Hence, by recalling that $J = J_1 + J_2$, we obtain

$$\begin{aligned}\frac{\partial J}{\partial \tau} &= x^T(\tau)Q_cx(\tau) + u_0^TR_cu_0 + 2x^T(\tau)N_cu_0 \\ &\quad + x^T(\tau)(A^TS + SA)x(\tau) + 2x^T(\tau)SBu_0 \\ &= x^T(\tau)\bar{Q}x(\tau) + u_0^T\bar{R}u_0 + 2x^T(\tau)\bar{N}u_0 \\ &= \left[\begin{array}{cc} x^T(\tau) & u_0 \end{array}\right] \left[\begin{array}{cc} \bar{Q} & \bar{N} \\ \bar{N}^T & \bar{R} \end{array}\right] \left[\begin{array}{c} x(\tau) \\ u_0 \end{array}\right] \end{aligned} \tag{20}$$

where

$$\bar{Q} = \left[A^TS + SA + Q_c\right], \ \bar{N} = \left[N_c + SB\right], \ \bar{R} = R_c.$$

From the constraint (10), by setting (20) equal to zero, we obtain (15). ■

Once the set of $\tau_i$ have been identified, the solution to the problem (9)–(13) requires selecting the longest $\tau_i$ that fulfills restriction (11). If non of them applies, then $\tau = h$.

Note that the event condition (15) will have solution if the matrix characterized by $\bar{Q}$, $\bar{N}$, $\bar{N}^T$ and $\bar{R}$ has positive and negative eigenvalues.

### B. Implementation Issues

The implementation of the one-step boundary event-driven controller requires a dedicated hardware that at each control update continuously checks if condition (15) with the selected $\tau$ is fulfilled. Hence, at each control update, the optimization problem (9)-(13) must be solved to obtain $\tau$ and then the dedicated hardware must be adjusted with this value.

To avoid using dedicated hardware, a self-triggered implementation may be desirable (e.g., [4] or [5]). A self-triggered implementation requires computing at each control update when the next controller activation will occur, and then programming a timer with this value. In the one-step boundary, this again requires solving the optimization problem (9)-(13) to obtain $\tau$.

Therefore, using either the dedicated hardware approach or the self-triggered approach, computing the solution to the optimization problem (9)-(13) is always required. The following subsection tackles this problem.

### C. Computation of the Next Activation Time

The computation of the next activation time requires solving (15) to obtain the candidate points $\tau_i$ and then choose the longest one fulfilling (11). To do so, the dependency of $x(\tau)$ on $\tau$ must be made explicit, and then solve for $\tau$. A general approach can be to use a $n$-order approximation of $x(\tau)$ if an exact expression for $x(\tau)$ does not exist.

First of all, $\Phi(\tau)$ and $\Gamma(\tau)$ in (13) can be written in terms of $\Psi(\tau)$ as [10]

$$\Phi(\tau) = e^{A\tau} = I + A\Psi(\tau) \tag{21}$$

$$\Gamma(\tau) = \int_0^\tau e^{As}B\mathrm{d}s = \Psi(\tau)B \tag{22}$$

where $\Psi(\tau)$ is

$$\Psi(\tau) = \int_0^\tau e^{As}\mathrm{d}s = \sum_{i=0}^{\infty} \frac{A^i\tau^{i+1}}{(i+1)!}. \tag{23}$$

Substituting (21) and (22) in (13) we obtain

$$\begin{aligned} x(\tau) &= \Phi(\tau)x_0 + \Gamma(\tau)u_0 \\ &= [I + A\Psi(\tau)]x_0 + \Psi(\tau)Bu_0 \\ &= [I + A\Psi(\tau)]x_0 + \Psi(\tau)BLx_0 \\ &= x_0 + A\Psi(\tau)x_0 + \Psi(\tau)BLx_0. \end{aligned} \tag{24}$$

Commuting matrices $\Psi(\tau)$ and $A$, (24) can be written as

$$x(\tau) = x_0 + \Psi(\tau)(A + BL)x_0. \tag{25}$$

Finally, we approximate $\Psi(\tau)$ by a Taylor series of $n$-order in a neighborhood of $\tau = 0$ in (25), thus obtaining

$$\begin{aligned} x(\tau) = x_0 + (A + BL)x_0\tau + \frac{A(A + BL)}{2!}x_0\tau^2 \\ + \frac{A^2(A + BL)}{3!}x_0\tau^3 + \ldots \end{aligned} \tag{26}$$

Using (21), (22), (23) and a convenient approximation given by (26), eq. (15) can be solved for $\tau$. Note that using a $n$-order approximation will result in a $2n$-order equation. Among all solutions, the next activation time will be the biggest real solution (i.e., longest $\tau$) providing a cost (6) smaller than the one delivered by the periodic LQR controller (5), that is, fulfilling constraint (11).

## IV. SIMULATION RESULTS

In order to qualitatively evaluate the effectiveness of an event-driven controller based on the one-step finite-horizon boundary, a detailed simulation analysis has been carried out.

### A. Simulation Settings

The LTI plant for simulation is the double integrator system given by

$$\dot{x} = \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \end{array}\right] x + \left[\begin{array}{c} 0 \\ 1 \end{array}\right] u$$

.

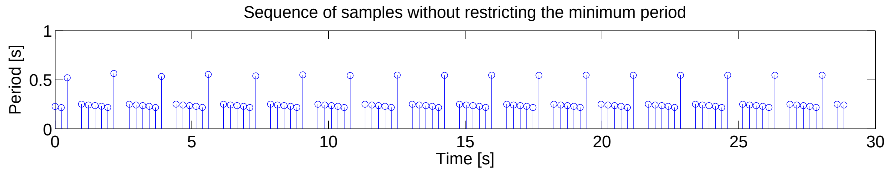
The cost function (6) to be minimized is characterized by

$$Q_c = \left[\begin{array}{cc} 1.6242 & 0.0363 \\ 0.0363 & 0.0008 \end{array}\right], \quad N_c = \left[\begin{array}{c} 0.4738 \\ 0.0106 \end{array}\right],$$
$$R_c = 0.1382$$

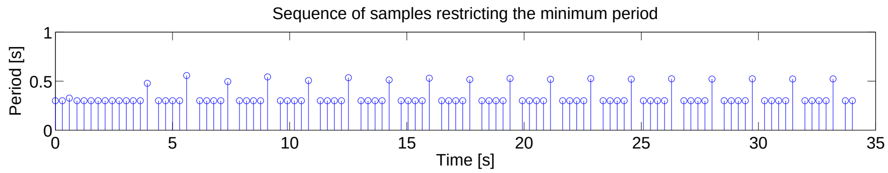
and

$$S = \left[\begin{array}{cc} 0.1095 & 0.0000 \\ 0.0000 & 0.0308 \end{array}\right]$$

for a sampling period of $h = 0.3$s. The optimal LQR gain for (4) with $h = 0.3$s is $L = \left[\begin{array}{cc} -3.1232 & -0.7488 \end{array}\right]$.




(a) One-step boundary




(b) One-step boundary with restriction of a minimum period

Fig. 2: Sampling intervals.

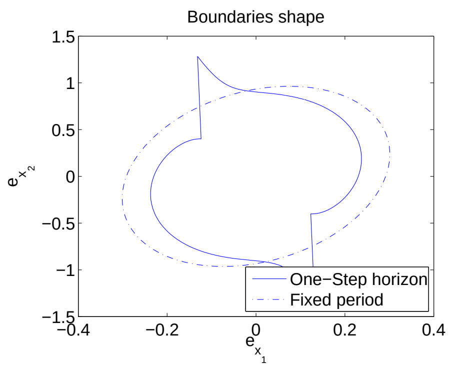



Fig. 1: Periodic and one-step boundary

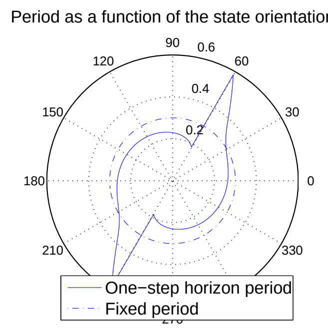



Fig. 3: Comparative of sampling intervals

The evaluation of the one-step boundary event-driven controller is compared to the LQR optimal periodic controller. In [11] the boundary in the form of (3) is given for periodic discrete-time systems, producing the periodic activation of control jobs. The periodic boundary of the LQR optimal controller is used for comparative purposes.

Figure 1 shows the boundary shapes for both the one-step boundary controller and the periodic LQR controller, given the specific simulation settings. For both boundaries, this figure illustrates the allowed distance (error) that, from any given state (located in the center of the boundary), the closed-loop system trajectory will move without requiring a control update. The control signal will be updated when the trajectory will hit the solid line for the one-step boundary controller or the dashed line for the periodic controller. While the periodic boundary has an expected circle shape, the one-step boundary (plotted considering always the longest $\tau$) offers an irregular shape that indicates that depending on the trajectory direction, more or less space and time will be covered without requiring a control update.

### B. Resource Utilization Analysis

Figures 2 and 3 illustrate the computational demand of the one-step boundary controller. In particular, Figure 3 plots for both the one-step boundary controller and the LQR periodic controller the expected sampling interval given the orientation of the current state. For example, for any state orientation, the sampling period of the periodic controller (dashed line) is $0.3$s. However, the period (separation between two consecutive controller activations) for the one-step boundary controller varies depending on the orientation of the current step (see [12] for a deeper analysis of activation patterns for certain types of event-driven controllers). For example, if the current state orientation is $90°$, the figure tells that the next controller activation will occur after $0.23$s approximately. And if the current state orientation is $59°$, the next controller activation time would be after $0.6s$ approximately. Hence, from Figure 3 it can be observed the type of sampling intervals that the one-step boundary will produce compared to the periodic boundary. In some cases, sampling intervals will be longer than the periodic case ($h = 0.3$s) and in other cases they will be shorter.

Sub-figure 2a shows the activation pattern for the one-step boundary for a given initial condition, $x_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. The $x$-axis is simulation time, and the $y$-axis is the sampling interval also in seconds. Each job activation time is represented by a vertical line, whose height indicates the next job expected execution time (the next sampling interval). As it can be seen in the figure, an oscillatory pattern occurs, alternating a long sampling interval of $0.55s$ with several short intervals ranging from $0.25$s to $0.21$s.

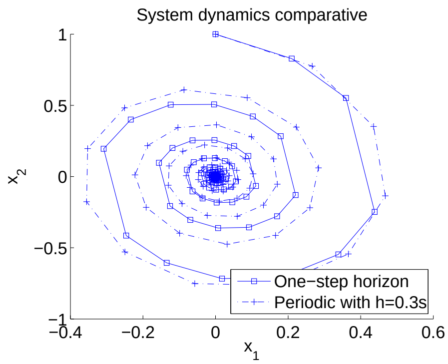



Fig. 4: Event-driven dynamics

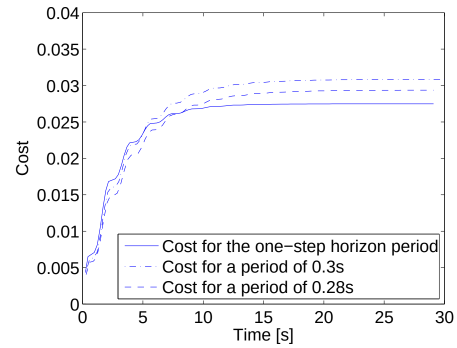



Fig. 5: Control performance evaluation

Sub-figure 2b shows a similar sequence of activations than the case illustrated in sub-figure 2a but for the case of restricting the execution to a minimum period. Since the one-step boundary can produce shorter periods than the periodic case, it is also of interest to restrict the timing of the one-step boundary controller to the case where longer periods than $0.3$s are allowed while shorter periods than $0.3$s are saturated to $0.3$s. This will permit a more accurate analysis of the control performance of several control strategies, which is presented next.

### C. *Control Performance Analysis*

Figure 4 shows the closed-loop system dynamics for both the one-step boundary controller and the periodic controller for the same initial condition used for generating the activation patterns shown in Figure 2. As it can be seen, the one-step boundary is capable of driving the state toward zero quicker than the optimal LQR periodic case. This is an expected result observing the activation pattern of the one-step boundary (sub-figure 2a) and the fact the the period of the LQR controller is $0.3$s. Note that the one-step boundary has an average sampling interval of $0.28$s, and therefore it seems reasonable that the dynamics progress quicker than for the periodic case.

It is also interesting to observe in Figure 4 the next sampling interval that occurs for the state $\begin{bmatrix} -0.24 & -0.41 \end{bmatrix}^T$ belonging to the one-step boundary trajectory, and for the state $\begin{bmatrix} -0.24 & -0.52 \end{bmatrix}^T$ belonging to the periodic trajectory. Both states have a similar orientation, which approximately is $239°$ and $245°$, respectively. However, although having a similar orientation, the next sampling for the one-step boundary will occur at $\begin{bmatrix} -0.30 & 0.19 \end{bmatrix}^T$ after almost $0.6$s. For the periodic case, to reach a similar state, $\begin{bmatrix} -0.35 & 0.19 \end{bmatrix}^T$, it is required to apply an extra control update, each one of $0.3$s. Hence, depending on the state orientation, the sampling intervals that apply can be very different for both controllers.

Figure 5 shows the control performance evaluation of several controllers. The cost is evaluated using (4) with the specific matrices given in sub-section IV-A. Four strategies have been tested, although only three are plotted. From the same initial condition, the one-step boundary controller provides the best cost (solid line) compared to the LQR periodic controller with $h = 0.3$s (dash-dotted line). This was a known result already stated before and illustrated in Figure 4.

As we pointed out before, the average sampling period for the one-step boundary controller is $0.28$s. Hence, it seems interesting to compare the one-step boundary controller against a LQR periodic controller with a period of $0.28$s (dashed line in Figure 5). Both controllers have the same resource demands. However, the one-step controller still delivers better control performance.

Finally, it is worth mentioning that the one-step controller with restricted minimum period, whose activation pattern was shown in Figure 2b, has also been evaluated. Note that this one consumes less resources than the periodic with $h = 0.3s$ because from time to time it applies a long sampling interval near $0.6$s. Regarding the cost, the restricted one-step boundary controller provides the same cost than the periodic case. The curve was omitted in Figure 5 for the sake of clarity.

### D. *Next Activation Time for Self-triggered Implementation*

This section illustrates the computation of the next activation time described in sub-section III-C if a self-triggered implementation is pursued.

With the simulation settings presented in sub-section IV-A, the one-step boundary matrix in (15) is

$$\begin{bmatrix} \bar{Q} & \bar{N} \\ \bar{N}^T & \bar{R} \end{bmatrix} = \begin{bmatrix} 1.6240 & 0.1458 & 0.4738 \\ 0.1458 & 0.0009 & 0.0414 \\ 0.4738 & 0.0414 & 0.1382 \end{bmatrix}.$$

To better illustrate the procedure for computing the next activation time, we will use an initial condition giving a longer next activation time than $0.3$s that would be the case for the periodic LQR controller. Looking at Figure 3, we outlined that for states with orientation $60°$, the next sampling interval is long. Hence, we take as initial condition $x_0 = \begin{bmatrix} 1/2 & 1/2\sqrt{3} \end{bmatrix}^T$. In addition, for the double integrator system, $x(\tau)$ has an exact closed expression given by

$$x(\tau) = \begin{bmatrix} x_1(\tau) \\ x_2(\tau) \end{bmatrix} = \begin{bmatrix} 0.5000 - 1.105\,\tau^2 + 0.8660\,\tau \\ -2.210\,\tau + 0.8660 \end{bmatrix}.$$

And recalling that $u_0 = Lx_0 = -2.2101$, then eq. (15) reduces to

$$1.983\,\tau^4 - 2.396\,\tau^3 + 0.9043\,\tau^2 - 0.1095\,\tau + 0.00218 = 0.$$

The solutions $\tau_i$ to the previous equation are given next, as well as their cost (6),

$$\tau_i = \begin{bmatrix} 0.5790s \\ 0.4229s \\ 0.1813s \\ 0.02518s \end{bmatrix}, \; J_i = \begin{bmatrix} 0.05053 \\ 0.05072 \\ 0.05038 \\ 0.05057 \end{bmatrix}.$$

because the specific value of $\tau$ will be chosen to be the longest period giving a cost lower than the periodic case, as indicated in the optimization problem (9)-(13). By observing that the cost of the periodic LQR controller with $h = 0.3$s is $J^* = 0.05054$, it is easy to assert that the candidate values for $\tau$ are $\tau_1$ and $\tau_3$ because they will deliver a smaller cost than $J^*$. Since among these two candidates we are interested in the one minimizing resource utilization, we pick for the next activation time the longest sampling interval, that is

$$\tau_1 = 0.5790\text{s}.$$

Note that it provides a lower cost than the periodic case while being almost twice than the period of the periodic case.

### *E. Discussion*

From the previous performance analysis, it is worth commenting several issues.

First of all, from the performance analysis shown in Figure 5, an interesting conclusion can be drawn. Given an optimal LQR periodic controller, better control performance (evaluated using the same cost function) can be obtained if the same amount of resources used by the periodic controller are redistributed following a non-periodic pattern. In particular, the one-step boundary presented in Section III can provide such a non-periodic pattern that consuming the same resources that the periodic controller, is able to provide better control performance.

Second, in the execution of the one-step boundary controller, the controller gain that applies is always the same, which is the LQR gain for the periodic controller. Current work is analyzing the effectiveness of the one-step boundary controller in two different scenarios: when the controller gain changes according to each sampling interval that will apply, and when the control signal $u_0$ becomes also a decision variable of the optimization problem (9)-(13) together with $\tau$.

Third, similar to model predictive control, the length of the finite-horizon determines the goodness of the controller. Current work is also investigating whether a boundary in closed form can be obtained such that its application would result in optimizing a finite-horizon cost function for two or more steps rather than just one.

## V. CONCLUSIONS

This paper has presented the one-step boundary for event-driven controllers. With this boundary, at each sampled state, the next activation time becomes the decision variable to be maximized considering also that the delivered cost must be not worse than the one achieved by the periodic LQR controller. A close form for the boundary has been derived, and a general procedure for solving the computation of the next activation time has been presented. Detailed simulation results have evaluated the resource utilization and control performance of the one-step boundary controller compared to an optimal LQR periodic controller. It has been shown that, using the same amount of resources, better control performance can be obtained using aperiodic sampling than the standard periodic approach. Hence, “clever” redistribution of resources is the key for improving control performance.

## REFERENCES


[1] G. Buttazzo, ”Research Trends in Real-Time Computing for Embedded Systems”, *ACM SIGBED Review*, Vol. 3, No. 3, 2006.

[2] K.-E. Årzén, ”A Simple Event-Based PID Controller”, *in 14th World Congress of IFAC*, January, 1999.

[3] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. van den Bosch, ”Analysis of event-driven controllers for linear systems,” *International Journal of Control*, 81(4), 2008, pp. 571-590.

[4] X. Wang and M. Lemmon, Self-triggered Feedback Control Systems with Finite-Gain L2 Stability, *IEEE Transactions on Automatic Control*, vol. 45, n. 3, pp. 452-467, March 2009.

[5] M. Jr. Mazo, A. Anta, P. Tabuada, ”An ISS self-triggered implementation of linear controllers”, *Automatica*, Vol. 46, N. 8, 2010

[6] P. Martí M. Velasco and E. Bini, ”The optimal boundary and regulator design problem for event-driven controllers”, *in 12th International Conference on Hybrid Systems: Computation and Control*, April 2009

[7] E. Bini, ”Design of Optimal Control Systems”, Master thesis, Scuola Superiore Sant’Anna, Pisa, Italy, January 2010.

[8] M. Ben Gaid, D. Simon and O. Sename, ”A Design Methodology for Weakly-Hard Real-Time Control,” *in 17th IFAC World Congress on Automatic Control*, Seoul, Korea, July 2008.

[9] M. Velasco, P. Martí and E. Bini, ”Control-driven Tasks: Modeling and Analysis”, *in 29th IEEE Real-Time Systems Symposium*, Dec. 2008

[10] K.J. Åström and B. Wittenmark, *Computer controlled systems*, Prentice Hall, 1997.

[11] M. Velasco, P. Martí and C. Lozoya, ”On the timing of discrete events in event-driven control systems”, *in 11th International Conference on Hybrid Systems: Computation and Control*, April 2008

[12] M. Velasco, P. Martí and E. Bini, ”Equilibrium sampling interval sequences for event-driven controllers”, *in European Control Conference 2009*, Budapest, Hungary, August 2009.