

One-Step Finite Horizon Boundary with Varying Control Gain for Event-Driven Networked Control Systems

José Yépez, Manel Velasco, Pau Martí, Enric X. Martín and Josep M. Fuertes

Abstract—Recent research has shown that event-driven control requires less samples than time-driven (periodic) control. Consequently, event-driven control applied to Networked Control Systems (NCS) is a good approach for alleviating controllers bandwidth demands and reducing overall network traffic. Following this trend, this paper presents an execution rule for event-driven networked controllers that at each job execution aims at postponing the next job execution while applying an appropriated controller gain considering each varying sampling interval that applies at run time. This method permits to dynamically lower the generated traffic for each networked control loop while ensuring the same or better control performance than the achieved by the periodic case. In addition, an implementation strategy capable of dealing with network induced time delays is also presented. Simulation results illustrate the operation and benefits of the presented approach.

I. INTRODUCTION

In NCS, sensors, actuators and controllers are spatially distributed and exchange control data through a shared communication medium which inevitably limits the amount of traffic that can be dedicated to each networked control loop. NCS are normally implemented using periodic sampling, which implies that control messaging is periodic [1]. Periodic sampling allows for standard sampled-data control theory to be used, and the control loop stability can be easily guaranteed at the expenses of making a static use of the bandwidth: the given messages periodicity is imposed from the control design stage regardless of the current load in the network and/or changes in the plants that are being controlled. The standard approach for alleviating bandwidth consumption of periodic controllers is to decrease their periodicity rate, which in general terms implies a control performance degradation.

Lately, the research literature has provided diverse theoretical results suggesting that for several control loops sharing limited computing resources, a key design aspect is to efficiently select the controllers sampling periods. And a promising methodology is event-driven control (e.g., [2], [3], [4], [5]) because it has been shown that it may require less sampling operations than the periodic case to achieve the same control performance. Many of the results for event-driven control of embedded systems can

be extended to NCS. The basic idea is to sample, control and actuate, that is, to communicate, only when something significant has occurred in each controlled plant.

In fact, event-driven sampling for NCS has been shown to be much more efficient than time-driven sampling schemes in terms of bandwidth utilization in networked set-ups [6]. Recent research has applied event-triggering techniques to NCS (e.g., [7], [8], [9] or [10]). In [7] or [8] spare bandwidth is used for executing additional control updates apart from the periodic ones in such a way that control performance is increased. In [9] or [10] the approaches aim at minimizing network traffic.

In particular, the approach in [10] adapted the event-driven control method named one-step finite horizon boundary [5] to the case of NCS. The approach in [10] defines an execution rule (or boundary) 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. This is achieved by finding the maximum sampling interval that can elapse without requiring a new control update. This paper extends the theoretical result by deriving the form of the optimal controller gain that must apply considering each sampling interval. The approach in [10] discussed the application of the one-step boundary for NCS. However, time delays were not explicitly addressed. This paper presents an implementation strategy that is able of absorbing network induced time delays.

Overall, the paper presents the one-step finite-horizon boundary approach with varying control gains for event-driven controllers. Using this approach, each job execution determines both the occurrence of the next job execution and the appropriated control gain that must be applied. Simulation results show that controllers using this approach can provide better control performance than the periodic case while using the same amount of resources. In addition, simulations also show the effectiveness of the implementation strategy for dealing with varying time delays.

The rest of this paper is organized as follows. Section II introduces the preliminaries on NCS. Section III formalizes the problem that is solved in Section IV. Section V explains the implementation strategy for dealing with time delays. Section VI presents simulation results and Section VII concludes the paper.

II. PRELIMINARIES

The networked control system considered in this paper consists of $i = 1 \dots n$ control loops, each one formed by

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

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

a sensor, controller and actuator implemented in physically separated nodes and sharing a communication medium (network) to exchange the control data required for each control loop operation or job. Each plant is described by a continuous-time linear system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\quad (1)$$

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

$$u(t) = u_k = -L(t_{k+1} - t_k)x(t_k) = -L_k x_k \quad (2)$$

$\forall t \in [t_k, t_{k+1})$ (be the control updates given by a varying feedback controller L_k designed in the discrete-time domain using only samples of the state at discrete instants $t_0, t_1, \dots, t_k, \dots$. Times t_k are referred as the sampling 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, and thus the controller gain L_k becomes constant.

For each networked control loop, rather than applying periodic control, we are interested in achieving a non-periodic execution pattern based on

- enlarging at each job execution the current sampling interval
- and applying the appropriated controller gain considering the current sampling interval
- while providing the same or better control performance than the periodic case
- and considering that the forthcoming job activations after the current one will be periodic.

III. PROBLEM FORMULATION

Control performance is often measured by an infinite-horizon continuous-time quadratic cost function [11]

$$J = \int_0^\infty [x^T(t)Q_c x(t) + u^T(t)R_c u(t) + 2x^T(t)N_c u(t)] dt \quad (3)$$

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

If the gain L_k in (2) is designed using periodic linear quadratic (LQR) control for a given h , the minimum cost of (3) is

$$J^* = x_0^T S(h)x_0 \quad (4)$$

where $S(h)$ is the solution to the algebraic Riccati equation, and x_0 is a given initial state.

The problem to be solved is to determine the boundary with associated control signal that will permit the following operation: find at each activation time t_k the longest next activation time t_{k+1} and the control gain L_k that must apply such that minimization of the cost (3) with respect to the cost given by the LQR periodic controller is achieved, and considering that the forthcoming activations will be periodic at times $t_{k+1} + h, t_{k+1} + 2h, t_{k+1} + 3h, \dots$

After defining $\tau = t_{k+1} - t_k$, the problem requires making the cost (3) explicitly depending on τ and u , and reflecting

that after τ the controller activations will be periodic. To do so, we can split cost (3) into two sub-costs J_1 and J_2 . The first cost applies to the τ time interval, which is unknown and must be maximized. In this cost u is also unknown, and must be also selected to minimize the cost (3). The second cost applies from τ to ∞ , where activations are expected to be periodic at the optimization time. Hence, the cost (3) can be re-written as

$$J = J_1 + J_2 \quad (5)$$

where

$$J_1 = \int_0^\tau [x^T(t)Q_c x(t) + u^T(t)R_c u(t) + 2x^T(t)N_c u(t)] dt \quad (6)$$

and

$$J_2 = x^T(\tau)S(h)x(\tau) \quad (7)$$

The one-step finite horizon boundary with varying control gain problem can be formulated as finding τ and u as solutions of the optimization problem formally stated as

$$\text{maximize } \tau \quad (8)$$

$$\text{subject to } \frac{\partial J}{\partial \tau} = 0 \quad (9)$$

$$\frac{\partial J}{\partial u} = 0 \quad (10)$$

$$\tau > 0 \quad (11)$$

$$J^* - J \geq 0 \quad (12)$$

$$x(\tau) = \Phi(\tau)x(t_k) + \Gamma(\tau)u(t_k) \quad (13)$$

where $u(t_k) = -L_k x(t_k)$ and

$$\Phi(\tau) = e^{A\tau}, \Gamma(\tau) = \int_0^\tau e^{As} ds B, \tau \in [t_k, t_{k+1}). \quad (14)$$

Noting that the objective function (8) is continuous and the constraint set is compact (constraint (9) and (10) reduces the constraint set to a set of isolated points), the problem has solution. Note that the optimization problem here formulated adds restriction (10) with respect the problem formulated in [5].

IV. ONE-STEP FINITE HORIZON BOUNDARY WITH VARYING CONTROL GAIN

Solving problem (8)-(13) implies three main steps: a) solve the system of equations given by restrictions (9) and (10) and find the set of real positive isolated points τ_i and u_i , b) from those points find the set of τ_i and u_i that minimizes the cost function (5), and c) from those points, choose the longest τ_i and corresponding u_i whose cost evaluated using (5) is lower than the cost (4) provided by the periodic controller, that is, fulfilling restriction (12).

The steps of the solution to the problem (8)-(13) are announced next.

Proposition 1: Given the closed-loop system (1)-(2), the boundary condition that permits finding the set of isolated τ_i that fulfill constraint (9) and (10) for any arbitrary initial

state $x(t_k)$ considering that after τ time units the controller activation will be periodic is

$$x^T(t_k) [\bar{Q}(\tau) + \bar{N}(\tau)L(\tau) + L^T(\tau)\bar{M}(\tau)] x(t_k) + x^T(t_k) [L^T(\tau)\bar{R}(\tau)L(\tau)] x(t_k) = 0 \quad (15)$$

where

$$\bar{Q}(\tau) = \Phi^T(\tau) (2SA + Q_c) \Phi(\tau) \quad (16)$$

$$\begin{aligned} \bar{N}(\tau) &= \Phi^T(\tau) (2SA + Q_c) \Gamma(\tau) \\ &+ 2\Phi^T(\tau)(N_c + SB) \end{aligned} \quad (17)$$

$$\bar{M}(\tau) = \Gamma^T(\tau) (2SA + Q_c) \Phi(\tau) \quad (18)$$

$$\begin{aligned} \bar{R}(\tau) &= \Gamma^T(\tau) (2SA + Q_c) \Gamma(\tau) + R_c \\ &+ 2\Gamma^T(\tau)(N_c + SB) \end{aligned} \quad (19)$$

and the varying gain is given by

$$L(\tau) = \hat{R}^{-1}(\tau)\hat{N}^T(\tau) \quad (20)$$

Proof: From (13) (system dynamics), the integral into the sum of the cost function (5) can be rewritten as

$$J = x^T(t_k)\hat{Q}(\tau)x(t_k) + 2x^T(t_k)\hat{N}(\tau)u(t_k) + u^T(t_k)\hat{R}(\tau)u(t_k) \quad (21)$$

where

$$\hat{Q}(\tau) = \int_0^\tau \Phi^T(t)Q_c\Phi(t)dt + \Phi^T(\tau)S\Phi(\tau) \quad (22)$$

$$\begin{aligned} \hat{N}(\tau) &= \int_0^\tau [\Phi^T(t)Q_c\Gamma(t) + \Phi^T(t)N_c]dt \\ &+ \Phi^T(\tau)S\Gamma(\tau) \end{aligned} \quad (23)$$

$$\begin{aligned} \hat{R}(\tau) &= \int_0^\tau [\Gamma^T(t)Q_c\Gamma(t) + 2\Gamma^T(t)N_c]dt \\ &+ \tau R_c + \Gamma^T(\tau)S\Gamma(\tau) \end{aligned} \quad (24)$$

Looking at the optimization problem (8)-(13), the first step is to find the critical points of $J(\tau)$, that is, to solve constraint (9) and (10). From the equation (21), the constraint (9) can be expanded as

$$\begin{aligned} \frac{\partial J}{\partial \tau} &= x^T(t_k)\frac{\partial \hat{Q}(\tau)}{\partial \tau}x(t_k) + 2x^T(t_k)\frac{\partial \hat{N}(\tau)}{\partial \tau}u(t_k) \\ &+ u^T(t_k)\frac{\partial \hat{R}(\tau)}{\partial \tau}u(t_k) \end{aligned} \quad (25)$$

Taking into account the Leibniz integration rule and since the matrix S is symmetric, the equation (9) is given by

$$\begin{aligned} \frac{\partial J}{\partial \tau} &= x^T(t_k)\bar{Q}(\tau)x(t_k) + x^T(t_k)\bar{N}(\tau)u(t_k) \\ &+ u^T(t_k)\bar{M}(\tau)x(t_k) + u^T(t_k)\bar{R}(\tau)u(t_k) \end{aligned} \quad (26)$$

where $\bar{Q}(\tau)$, $\bar{N}(\tau)$, $\bar{M}(\tau)$ and $\bar{R}(\tau)$ are given by the equations (16), (17), (18) and (19) respectively.

From (21), the one-step optimal control (10) is given by

$$\frac{\partial J}{\partial u(t_k)} = 2x^T(t_k)\hat{N}(\tau) + 2u^T(t_k)\hat{R}(\tau) = 0 \quad (27)$$

Since R is positive definite from the hypothesis, then it is invertible. Hence from equation (27) it follows that

$$u^*(t_k) = -\hat{R}^{-1}(\tau)\hat{N}^T(\tau)x(t_k) \quad (28)$$

which can be rewritten as

$$u^*(t_k) = -L_k x(t_k) = -L(\tau)x(t_k) \quad (29)$$

That is the optimal control gain L_k to apply at t_k is a state feedback control with gain

$$L(\tau) = \hat{R}^{-1}(\tau)\hat{N}^T(\tau) \quad (30)$$

Substituting the optimal control (28) in (26) we obtain

$$\begin{aligned} x^T(t_k) [\bar{Q}(\tau) + \bar{N}(\tau)L(\tau) + L^T(\tau)\bar{M}(\tau)] x(t_k) \\ + x^T(t_k) [L^T(\tau)\bar{R}(\tau)L(\tau)] x(t_k) \end{aligned} \quad (31)$$

Recovering (9), by making (31) equal to zero, we obtain (15). \blacksquare

Once the set of τ_i and u_i have been identified, the second step of the solution to the problem (8)-(13) requires selecting the set of τ_i and u_i that minimizes the cost function (5). Noting that u_i is characterized by $L(\tau_i)$ (30), which is a parametric form depending on τ_i , the selection restricts to the set of τ_i . From those points, the third step of the solution to the problem (8)-(13) requires selecting the longest τ_i that fulfills restriction (12). Note that other policies could be applied at the third step. For example, rather than maximizing τ , that is, rather than reducing bandwidth utilization, another policy could be to select a τ such that the cost difference (12) is maximized. Such alternatives policies are left for future work.

The optimal performance is obtained by replacing u^* optimal (29) in (21)

$$J^* = x^T(t_k)\hat{S}(\tau)x(t_k) \quad (32)$$

where

$$\hat{S}(\tau) = [\hat{Q}(\tau) + 2\hat{N}L(\tau) + L^T(\tau)\hat{R}L(\tau)] \quad (33)$$

It is interesting to note that the time interval τ^* generated by the one-step finite horizon boundary approach with varying control u^* of each closed loop system only uses local information, and therefore, it can be easily computed in any node. And the computation of the next activation time t_{k+1} requires solving (15) to obtain the candidate points τ_i and then choose the longest one fulfilling (12). To do so, $x(\tau)$ must be made explicitly depending on τ , and then solve for τ . As shown in [5], a general approach can be to use a n -order approximation of $x(\tau)$ around the $x(t_k)$ if an exact expression for $x(\tau)$ does not exist.

However, at each closed loop execution, performing the three steps to determine τ and u may be expensive in terms of computational overhead. In particular, the application of the execution rule requires solving equation (15) at each closed loop execution. Note that (15) and $u(t_k)$ given (28) depend on $x(t_k)$. Therefore, the only parameter of (15) that changes at each execution is $x(t_k)$, that is, the sampled state. However, as indicated in [12], for execution rules having the structure specified in (15), the solution to (15) remain the same for states lying in the same direction, that is, having the same orientation. This property permits an offline approach when applying the set of steps at each control loop execution.

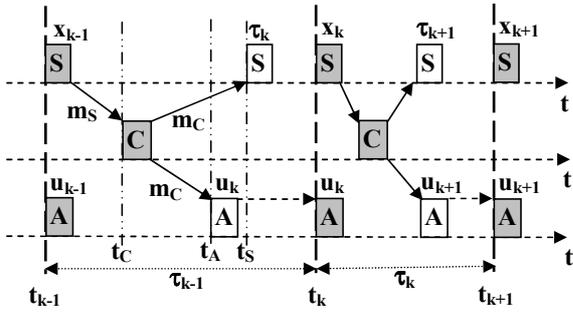


Fig. 1: Operation of the distributed control algorithm

In this case, for each networked closed loop system, in the controller node, the information associated to each state orientation, that is, the corresponding τ and u , can be stored in memory for run-time table look up. Once the state is sampled and its orientation determined, the table will be accessed to determine τ and u . The size of the table will depend on the granularity of states orientations specified in the interval $[0, 2\pi]$. In fact, the linear systems symmetry property [12] permits to shrink the interval to $[0, \pi]$. Hence, its implementation is easy and the computational overhead is negligible. The memory storage overhead depends on the granularity chosen for the table, which at the end will affect the accuracy of the selected τ .

V. IMPLEMENTATION STRATEGY FOR DEALING WITH TIME DELAYS

The operation of the one-step finite horizon boundary approach with varying controller gain may suffer important control performance degradation if network-induced time delays are not accounted for. The system model in section II assumed zero time delay between sensing and actuation. However, in a networked set-up, such assumption does not longer holds. To cope with time delays, an implementation strategy named *time delay compensation* is presented.

The basic idea of the time delay compensation is to force each updated actuation at the next sampling instant. To do so, the controller node, after receiving the sampled state (taken at time t_{k-1} at the sensor node), namely x_{k-1} , predicts the state \hat{x}_k at the next sampling instant t_k knowing when it will occur, which is at $t_k = t_{k-1} + \tau_{k-1}$. Hence, the controller uses the τ_{k-1} solution of the optimization problem (8)-(13) computed in the previous controller job execution. Then using the sampled state, as well as the previous applied control signal u_{k-1} , the prediction is computed by

$$\hat{x}_k = \Phi(\tau_{k-1})x_{k-1} + \Gamma(\tau_{k-1})u_{k-1}. \quad (34)$$

Using the predicted state \hat{x}_k , the updated control signal can be computed as

$$u_k = -L_k \hat{x}_k \quad (35)$$

where L_k is given by the solution of the optimization problem (8)-(13) having \hat{x}_k as a input. The same solution gives also τ_k , that will be used in the next controller job execution. These two informations, τ_k and u_k are sent to the

Algorithm 1: Sensor (triggered by interrupt every τ)

```

1 begin
2    $(x_{k-1}, t_{k-1}) := \text{sample}()$ 
3    $\text{send\_sensor\_message}(x_{k-1}, t_{k-1})$ 
4    $\tau_k = \text{receive\_control\_message}()$ 
5    $\text{wait}(t_{k-1} + \tau_{k-1} - t_S)$ 
6    $\tau_{k-1} = \tau_k$ 
7 end

```

Algorithm 2: Controller (triggered on message reception)

```

1 begin
2    $x_{k-1} := \text{receive\_sensor\_message}()$ 
3    $\hat{x}_k = \Phi(\tau_{k-1})x_{k-1} + \Gamma(\tau_{k-1})u_{k-1}$ 
4    $(\tau_k, u_k) := \text{compute\_}\tau\text{\_and\_}u(\hat{x}_k)$ 
5    $\text{send\_controller\_message}(\tau_k, u_k, t_{k-1})$ 
6    $\tau_{k-1} = \tau_k$ ;
7 end

```

Algorithm 3: Actuator (triggered on message reception)

```

1 begin
2    $(\tau_k, u_k, t_{k-1}) := \text{receive\_controller\_message}()$ 
3    $\text{wait}(t_{k-1} + \tau_{k-1} - t_A)$ 
4    $\text{apply\_control\_signal}(u_k)$ 
5    $\tau_{k-1} = \tau_k$ ;
6 end

```

Fig. 2: Nodes pseudo-codes with the novel approach

sensor and actuator nodes, respectively. The sensor node will use this information to program the sampling corresponding to the time $t_k + \tau_k$. The actuator will use both informations to delay the application of u_k up to the next sampling instant, at time t_k , which is computed using the previous received τ_{k-1} .

The implementation strategy is illustrated in Figure 1, where the sensor node is triggered according to τ , which varies at each sensor execution. Grey boxes with A, S or C represent the execution of the code in the actuator, sensor or controller nodes, respectively. White boxes with A or S represent data updates in the actuator or sensor, respectively. And m_s and m_c represent the messages required at each closed loop operation. The basic functionality, illustrated also in the pseudocodes of Figure 2, is:

- At time t_{k-1} it is assumed that all nodes know the length of the current sampling interval τ_{k-1} . At this time two actions are carried out. First, the actuator node applies the control signal u_{k-1} (previously obtained) to the plant. Second, the sensor samples the plant and sends the time-stamped sample (x_{k-1}, t_{k-1}) to the controller (message m_s).
- Upon reception of this message, the controller estimates the plant state \hat{x}_k that will apply at t_k , and computes τ_k and u_k solving the optimization problem (8)-(13) using this prediction. Values τ_k , u_k , and t_{k-1} are sent to the

τ	u	J
0.3953	-2.7127	0.0701
0.2461	-2.7401	0.0702
-3.2550	0.2858	-0.4715
-0.2043	-2.4120	0.0709
-3.4282	0.2858	-0.4715
0.0754	-2.6879	0.0701
-0.15 ± 1.65	$0.63 \pm 1.13i$	$-1.51 \pm 0.57i$
0.03 ± 1.65	$0.63 \pm 1.11i$	$-1.51 \pm 0.57i$

TABLE I: Set of possible values for τ and u

actuator and sensor nodes (message m_c).

- Upon reception of message m_c at time t_A , the actuator node programs the actuation time to be at $t = t_{k-1} + \tau_{k-1} - t_A$. When it fires, it applies u_k to the plant.
- When the message m_c is received by the sensor node at time t_S , it programs the next sampling time to be at $t = t_{k-1} + \tau_{k-1} - t_S$.
- All nodes update the next sampling interval

It is important to stress that a) transmission times have been omitted for the sake of clarity and b) the presented pseudocodes assume global time within nodes. Transmission times can be easily added in the computations of the sampling and actuation instants, and synchronized sampling and actuation operations can be achieved without assuming global time using the approach presented in [13].

VI. SIMULATIONS

A. Simulation Settings

The networked control system consists on two networked control loops. The LTI plant for each networked control is a double integrator system given by

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

with initial conditions, $x_1(0) = [\sqrt{2}/2 \ \sqrt{2}/2]^T$ and $x_2(0) = [0.9 \ 1.3]^T$

For both control loops, the cost function (5) to be minimized is characterized by

$$Q_c = \begin{bmatrix} 1.6242 & 0.0363 \\ 0.0363 & 0.0008 \end{bmatrix}, \quad N_c = \begin{bmatrix} 0.4738 \\ 0.0106 \end{bmatrix},$$

$$R_c = 0.1382$$

and

$$S = \begin{bmatrix} 0.1095 & 0.00004 \\ 0.00004 & 0.0308 \end{bmatrix}$$

Simulations are based on different scenarios for NCS: centralized control system (which simulates no delays) and distributed control system. The later has two flavors depending on whether the compensation technique explained in section V is applied. The performance of the presented approach is compared to the LQR optimal periodic controller over NCS with a sampling period of $h = 0.3s$. The corresponding optimal LQR gain for (2) is $L = [-3.1232 \ -0.7488]$.

B. Details on the Computation of τ and u

To illustrate the procedure for computing the sampling interval and the control signal described in Section IV, we will study in detail the first system with initial condition $x_1(0) = [\sqrt{2}/2 \ \sqrt{2}/2]^T$. In this case the double integrator system has an exact closed expression given by

$$x(\tau, u(t_k)) = \begin{bmatrix} \frac{u(t_k)\tau^2 + \sqrt{2}\tau + \sqrt{2}}{2} \\ \tau u(t_k) + \frac{\sqrt{2}}{2} \end{bmatrix}$$

From the equation (21), the constraints (9) and (10) are given by

$$\begin{aligned} \frac{\partial J}{\partial \tau} &= 2.297\tau^3 u_k + 3.07\tau^2 u_k + 0.8121\tau^2 \\ &+ 2.065\tau u_k + 1.77\tau + 0.1382u_k^2 \\ &+ 1.457u_k + 0.9583 = 0 \end{aligned} \quad (36)$$

$$\begin{aligned} \frac{\partial J}{\partial u} &= 0.3643\tau + 0.1382\tau u_k + 0.04144\tau^2 u_k \\ &+ 0.1582\tau^3 u_k + 0.03645\tau^4 u_k + 0.0812\tau^5 u_k \\ &+ 0.2194\tau^2 + 0.243\tau^3 + 0.1436\tau^4 = 0 \end{aligned} \quad (37)$$

The set of possible values to τ and u are obtained from solution of equations (36) and (37). The table I shows the set of possible values for τ and u , and their cost (5).

As indicated in the solution of optimization problem (8)-(13), the procedure is performed. First, the set of possible values of τ_i is restricted to the set of real positive. Therefore the candidate values are $\tau = \{\tau_1, \tau_3, \tau_6\}$. Second, the set of τ_i that minimize the cost function (5) is selected. This restriction reduces the candidate values to $\tau = \{\tau_1, \tau_6\}$. Knowing that the cost of the periodic LQR controller with $h = 0.3s$ is $J^* = 0.0702$, it is easy to assert that the candidate values for the sampling interval are $\tau = \{\tau_1, \tau_6\}$ 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 $\tau^* = \tau_1 = 0.3953s$. Note that it provides a lower cost than the periodic case with a resource utilization factor less than the periodic case.

C. Main Results

Figure 3 shows the control performance evaluation of several networked controllers. The cost is evaluated using (3) with the specific matrices given in sub-section VI-A. Hence, the lower the curve, the better the performance. Three strategies have been tested. The centralized control does not suffer delays, and therefore represents the ideal situation where delays does not exist. The two other strategies are affected by delays, and one of them uses the presented compensation technique.

From the same initial condition and a varying time delay generated by a normal distribution with $\mu = 0$ and $\sigma = \tau/10$, the networked one-step finite horizon boundary approach with varying controller gain and with delay compensation (dash-dotted line) has the same performance than the centralized controller (solid line). This result corroborates that the presented implementation strategy is effective at coping with

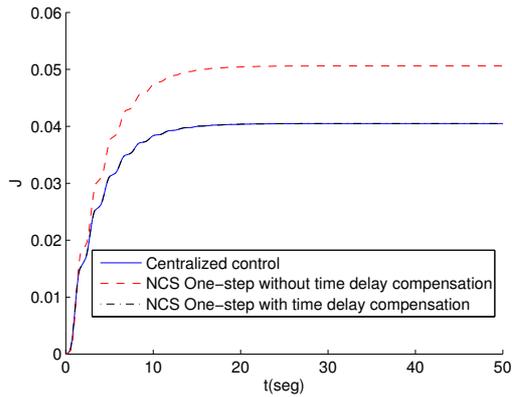


Fig. 3: Control performance evaluation for different NCS scenarios

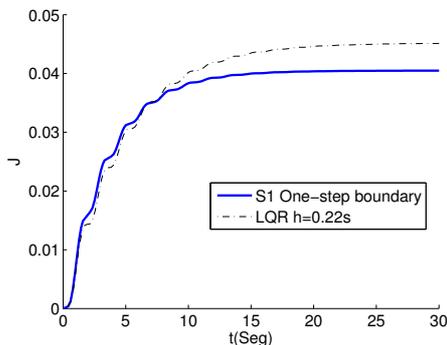


Fig. 4: Control performance evaluation of the event-driven approach with respect to the periodic one

network induced time delays. The dashed line corresponds to same event-driven networked strategy without compensating for time delays. It can be seen that delays introduce certain degree of degradation since the curve is above the previous ones.

Figure 4 illustrates the capacity of the presented event-driven approach at minimizing control performance with respect to the periodic approach. The cost is evaluated using (3), and again, the lower the line, the better the approach. The average sampling interval for one of the two networked controllers used in the previous simulation is 0.22s. Hence, it is interesting to compare that event-driven controller against a LQR periodic controller with a constant period of 0.22s (dashed line in Figure 4). Both controllers have the same resource demands in average. However, the one-step finite horizon boundary controller with varying gain delivers better control performance than the periodic case.

VII. CONCLUSIONS

This paper has presented a novel approach to event-driven networked control systems. The main properties of this approach is that the defined execution rule permits at each control loop operation to maximize the length of the current sampling interval and obtaining the appropriated control signal that must be applied. And the computation of

both parameters is subject to the restriction that the delivered cost must be not worse than the one achieved by the periodic LQR controller using a similar amount of network traffic. A close form for the boundary acting as execution rule has been derived, and a general procedure for solving the computation of the length of the current sampling interval and associated controller gain has been presented.

In addition, an implementation strategy has been presented to absorb the negative effects that network-induced delays have in NCS. The technique is based on synchronizing each actuation operation with each next sampling operation, and applying the control signal computed using an estimation of the state at the time when the actuation is performed.

Simulation results have shown that the application of the implementation strategy removes the degrading effects of time delays, and moreover, that using the same amount of resources, better control performance can be obtained using the presented event-driven sampling approach than the standard periodic approach.

Similar to model predictive control, the length of the finite-horizon determines the goodness of the controller. Therefore future work will focus on investigating whether a boundary in closed form can be obtained such that its application would result in optimizing a finite-horizon cost function for more than just one step.

REFERENCES

- [1] D. Hristu-Varsakelis and W. S. Levine. *Handbook of networked and embedded control systems* Birkhäuser Boston, June, 2008.
- [2] K.J. Åström and B.M. Bernharsson. "Comparison of Riemann and Lebesgue sampling for first order stochastic systems". In *Proceedings of the Conference on Decision and Control*, 2011-2016, 2002.
- [3] P. Tabuada, Event-triggered real-time scheduling of stabilizing control tasks, *IEEE Trans. on Automatic Control*, 52(9), 2007, pp. 1680-1685
- [4] 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.
- [5] M. Velasco, P. Martí, J. Yépez, F. Ruiz, J.M. Fuertes and E. Bini, "Qualitative Analysis of a One-step Finite-Horizon Boundary for Event-driven Controllers", in *50th IEEE Conference on Decision and Control*, December 2011.
- [6] A. Cervin and T. Henningsson, "Scheduling of event-triggered controllers on a shared network," *47th IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 2008.
- [7] A. Anta and P. Tabuada, "On the benefits of relaxing the periodicity assumption for networked control systems over CAN," *Real Time Systems Symposium*, December 2009.
- [8] P. Martí, A. Camacho, M. Velasco, M. Ben Gaid "Run-Time Allocation of Optional Control Jobs to a Set of CAN-based Networked Control Systems," *IEEE Transactions on Industrial Informatics*, vol.6, no. 4, November 2010.
- [9] X. Wang and M.D. Lemmon, "Event-triggering in distributed networked control systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp 586-601, March 2011.
- [10] P. Martí, M. Velasco, J. Yépez and E. Martín, "Lowering Traffic without Sacrificing Performance in Networked Control Systems", in *16th Emerging Technology and Factory Automation*, September 2011.
- [11] K.J. Åström and B. Wittenmark, *Computer controlled systems*, Prentice Hall, 1997.
- [12] M. Velasco, P. Martí and E. Bini, "Equilibrium sampling interval sequences for event-driven controllers," *European Control Conference 2009*, Budapest, Hungary, August 2009.
- [13] P. Martí, A. Camacho, M. Velasco, P. Marés and J.M. Fuertes, "Synchronizing Sampling and Actuation in the Absence of Global Time in Networked Control Systems," In *15th IEEE International Conference on Emerging Technologies and Factory Automation*, September, 2010.