

Preliminary Approach to Lyapunov Sampling in CAN-based Networked Control Systems

José Yépez, Camilo Lozoya, Manel Velasco, Pau Martí and Josep M. Fuertes

Automatic Control Department
Technical University of Catalonia
Pau Gargallo 5, 08028 Barcelona, Spain
Email: jose.yepez@upc.edu

Abstract—This paper presents a preliminary approach to networked control systems (NCS) that relies on an event-driven control method based on Lyapunov sampling. The goal is to study and develop approaches for NCS capable of offering controllers with low bandwidth demands.

The paper starts by presenting the theoretical framework required for applying Lyapunov sampling to a set of closed-loop systems that share a serial bus line. The discussion permits identifying which requirements must be fulfilled in order to guarantee overall stability. In addition, the feasibility of the theoretical framework in terms of implementation requires to address problems caused by the implicit distributed architecture of the NCS. In particular, an efficient implementation approach is described for the CAN network. Simulation results illustrate the operation and benefits of the presented approach.

I. INTRODUCTION

For networked control systems (NCS, [7]) it is important to identify methods for sensing and control capable of efficiently using the limited network bandwidth. It is well accepted that event-driven sampling methods offer controllers with a lower resource utilization than standard periodic discrete-time control laws. Recent and representative results on event-driven control can be found in [2], [3], [6], [12] and [9].

In particular, for a given closed-loop system, the event-driven control method presented in [9], called *Lyapunov sampling*, enforces control updates in such a way that the system energy expressed in terms of a Lyapunov function is decreased at a given rate. Control updates are triggered when a Lyapunov-based event-condition holds.

In this paper we explore the application of Lyapunov sampling [9] to a set of networked control loops. For each networked closed-loop system, sampling, control signal computation, and actuation is performed in separated nodes, namely sensor, controller and actuator respectively. In short, a control update for a given networked control system refers to the transmission of the two messages, sensor-to-controller and controller-to-actuator, required for each closed-loop operation.

The intuitive idea of our theoretical analysis we present is to treat a set of N networked closed-loop systems as an aggregated extended closed-loop system, namely N -NCS, under Lyapunov sampling. This requires to define a single event condition in terms of a Lyapunov function for the N -NCS, and to trigger all control updates each time the event-condition holds.

We show that the specification of the Lyapunov sampling for the set of networked closed-loop systems is theoretically feasible as long as some requirements regarding stability are met. We also address two main problems that appear in its deployment in a networked architecture. The first one is related to the distributed evaluation of the event-condition. The second one is related to the serialization of transmissions each time the event-condition holds, which is a main issue in NCS [5].

Event-driven control in NCS has been recently treated for example in [11] and [4]. We believe that our approach is complementary to their contributions in the sense that we also offer event-driven transmissions for each closed-loop operation. However, the difference relies in the fact that in our approach the event-condition that triggers control updates is specified in the system-energy domain rather than the state-space domain. Also, we address the feasibility of the implementation in the CAN [1] network.

The rest of this paper is organized as follows. Section II presents the networked control systems model and reviews prior work on Lyapunov sampling. In Section III the theoretical approach is developed. Section IV presents the strategies for a feasible implementation. Sections V and VI present the simulation set-up and simulation results, respectively. Finally, Section VII concludes the paper.

II. PRELIMINARIES

A. System Model

The theoretical set-up we consider is that N closed-loop systems, each one given by

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) \quad (1)$$

where $x_i \in \mathbb{R}^{n_i \times 1}$, $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times m_i}$, $u_i \in \mathbb{R}^{1 \times m_i}$, and $C_i \in \mathbb{R}^{1 \times n_i}$, share a serial bus that can only be used by a system at a time for transmitting a limited amount of data. In other words, at any instant t_k , only one control update of the form

$$\forall t \in [t_k, t_{k+1}) \quad u_i(t) = L_i x_i(t_k) \quad (2)$$

is executed.

Each closed-loop system specification (1)-(2) does not model the network delay. This is done on purpose to simplify the theoretical approach and notation. But including a time delay in the formulation would not alter the main results.

B. Lyapunov Sampling Revisited

In this section we review the Lyapunov sampling mechanism introduced in [9].

Consider a single continuous linear time-invariant control system given by (1). Let $V_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$

$$V_i = x_i^T P_i x_i \quad (3)$$

be a (local or global) Lyapunov function for the system in the classical sense, i.e., continuous and positive definite. The Lyapunov sampling method mandates to issue the control updates by a linear state feedback controller (2) using only samples of the state at discrete instants t_k given by the Lyapunov sampling triggering mechanism

$$V_i(x_i(t_{k+1})) = \eta_i V_i(x_i(t_k)) \quad , \quad \eta_i \in \mathbb{R}^+, \quad (4)$$

where

$$0 < \eta_i^* < \eta_i < 1, \quad (5)$$

$$\eta_i^* = \max_{x_i(0)} \frac{V_i^*(x_i(0))}{V_i(x_i(0))}, \quad (6)$$

and

$$V_i^*(x_i(0)) = \min_t V_i(x_i(t, x_i(0))) \quad \forall t \geq 0 \quad (7)$$

being $V_i(x_i(t, x_i(0)))$ the solution of (1)-(2) when $u_i(t) = L_i x_i(0)$ for a given initial condition $x_i(0)$.

The application of this triggering mechanism ensures stability of the closed-loop system in the Lyapunov sense. In particular, the stability is guaranteed by the construction of the event-condition (4) as long as η_i fulfils (5), where η_i^* is defined by (6)-(7). And in terms of resource demand, it produces an aperiodic sequence of control updates whose average frequency depends on the system energy decay ratio given by η_i , which becomes a design parameter in terms of trading off resource usage and control performance. Noting that different values of $V_i(x_i(t_k))$ in (4) define a set of contour curves in the system energy domain, a value of η_i near to 1 specifies closer contour curves, that is, more frequent control updates.

III. THEORETICAL APPROACH

In this section we present the theoretical approach of Lyapunov sampling to the set of networked closed loop systems. First, we present the N -NCS model under Lyapunov sampling, and discuss stability issues. Second, to solve the bus contention problem that the theoretical framework introduces, we specify which rule is enforced in order to have only one control update at a time. This requires to study the N -NCS energy variation in terms of each networked closed-loop system.

A. N -NCS Model

Let

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (8)$$

be the N -NCS that includes the set of N decoupled systems, where

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_N \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_N \end{bmatrix}, \quad \mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \end{bmatrix}.$$

And let $\mathcal{V} : \mathbb{R}^{\sum_i n_i} \rightarrow \mathbb{R}$ defined as

$$\mathcal{V} = \mathbf{x}^T \mathcal{P} \mathbf{x} \quad (9)$$

be a Lyapunov function for (8) where

$$\mathcal{P} = \begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_N \end{bmatrix}. \quad (10)$$

Function \mathcal{V} can be understood as a common Lyapunov function for the N networked closed loop systems.

Lyapunov sampling as described in Section II-B applied to the N -NCS would specify that control updates

$$\mathbf{u}(t) = \mathbf{L}\mathbf{x}(t) = \begin{bmatrix} L_1 & 0 & \cdots & 0 \\ 0 & L_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_N \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix} \quad (11)$$

should be enforced when the Lyapunov sampling condition

$$\mathcal{V}(\mathbf{x}(t_{k+1})) = \eta_k \mathcal{V}(\mathbf{x}(t_k)), \quad \eta_k \in \mathbb{R}^+, \quad (12)$$

holds.

We conjecture that considering (8)-(11) and \mathcal{V} , the energy decay ratio η_k in (12) can be similarly defined as η_i in (4). In other words, we leave for future work the theoretical analysis required for proving that system (8)-(11) with \mathcal{V} where control updates are triggered by (12) is stable by construction when some given requirements for η_k hold. Note that for the N -NCS under Lyapunov sampling, we leave the door open to having a varying decay ratio η_k . However, for the simulation results presented at the end of this paper, η_k has been heuristically selected constant.

B. Bus Contention, Energy and Execution Rule

By construction, and assuming that the previous conjecture holds, the N -NCS under Lyapunov sampling (8), (11) and (12) is stable if all control signals are updated each time the event condition holds. However, this approach would rise transmission conflicts: messages would collide in the bus of the networked architecture.

In order to solve this problem, we study the N -NCS energy variation, \mathcal{V} . From this study, we establish a rule that

permits updating only one control signal at a time without compromising stability.

The following proposition states how the N -NCS energy varies after the event-condition (12) holds.

Proposition 1: Consider the N -NCS (8) with its Lyapunov function \mathcal{V} defined in (9). The energy contribution v_i of each i^{th} closed-loop system in $\dot{\mathcal{V}}$ is

$$v_i = \frac{\partial \mathcal{V}}{\partial x_i} (A_i x_i + B_i u_i) \quad (13)$$

Proof: The energy variation is given by

$$\begin{aligned} \dot{\mathcal{V}} &= \frac{d\mathcal{V}}{dt} \\ &= \frac{\partial \mathcal{V}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} \\ &= \left[\frac{\partial \mathcal{V}}{\partial x_1} \quad \frac{\partial \mathcal{V}}{\partial x_2} \quad \cdots \quad \frac{\partial \mathcal{V}}{\partial x_n} \right] (\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)) \\ &= \frac{\partial \mathcal{V}}{\partial x_1} (A_1 x_1 + B_1 u_1) + \frac{\partial \mathcal{V}}{\partial x_2} (A_2 x_2 + B_2 u_2) \\ &\quad + \cdots + \frac{\partial \mathcal{V}}{\partial x_N} (A_N x_N + B_N u_N). \end{aligned} \quad (14)$$

It is interesting to note that the energy contribution (13) of each closed loop system to the global energy variation only uses local information, and therefore, it can be easily computed at each sampler node. That is, for each i^{th} closed-loop system, the energy contribution v_i only depends on its own state vector x_i and control signal u_i . Note that if system (8)-(11) is not decoupled, the computation of v_i would require using others nodes' information. The latter case is left for future work.

When the event-condition (12) holds, by definition $\dot{\mathcal{V}} \leq 0$. By assuming that η_k can be appropriately chosen in such a way that at each occurrence of the event condition we have i) $\dot{\mathcal{V}} < 0$, and ii) $\forall i, i = 1, 2, \dots, N, v_i < 0$, then we specify that the only control update that should be issued at each event condition corresponds to the specific networked closed-loop system providing the *weakest energy contribution*, which can be defined as

$$w = \arg \max_i \frac{\partial \mathcal{V}}{\partial x_i} (A_i x_i + B_i u_i). \quad (15)$$

We call *weak system* the w^{th} closed-loop system providing the weakest energy contribution.

IV. FEASIBLE IMPLEMENTATION STRATEGY

The previous theoretical analysis indicates that the application of the Lyapunov sampling requires updating only the control signal of the weakest closed-loop system without compromising overall stability. Therefore, at each event condition, we need to identify which is the weakest system. In addition, all sensor nodes must be able to detect the event-condition.

A. Distributed Detection of the Event Condition

The event condition (12) is defined in terms of all N -NCS states. However, it has to be evaluated at each sampler node in order to decide whether the control update for each given

networked closed-loop system should be triggered by sending the sensor-to-controller message.

We consider that all sampler nodes oversample each plant at a fast rate given by a short sampling period h . And at each oversample, the i -sampler node of the i^{th} -system obtains the n_i states, either measured or observed. In order to obtain the states of the other closed-loop systems, a predictor is implemented in the i -sampler. By having the N -NCS model (8), by knowing the N -NCS initial state $\mathbf{x}(0)$ and \mathbf{L} in (11), the open loop predictor

$$\mathbf{x}(k+1) = (\Phi(h) + \Gamma(h)\mathbf{L})\mathbf{x}(k), \quad (16)$$

where $\Phi(t) = e^{\mathbf{A}t}$ and $\Gamma(t) = \int_0^t e^{\mathbf{A}s} d\mathbf{s}\mathbf{B}$, can be used by any sampler node to obtain the rest of states at each sample. In fact, a closed-loop predictor could be a better approach in order to obtain the same unknown states. This alternative approach would remove the assumption of knowing the initial state, would be robust against perturbations and noise, and would be able to take advantage of the state information that may be enforced to be traveling in the bus. However, for paper readability, we will leave for future work the formulation of the closed-loop predictor and we will spend more space in the simulation results for illustrating the operation of the presented approach.

It is important to note that due to the discretization introduced by (16), the event condition (12) must be slightly changed to

$$\mathcal{V}(\mathbf{x}(t_{k+1})) \leq \eta_k \mathcal{V}(\mathbf{x}(t_k)), \quad \eta_k \in \mathbb{R}^+. \quad (17)$$

In fact, in any implementation, the inequality in (17) rather than the equality in (12) must be used also due to quantization effects or noise.

B. Selection of the Weakest Closed-loop System

As outlined in Section III-B, each energy contribution v_i can be computed locally, at each sampler node. Moreover, rather than updating all control loops when the event-condition holds, we can update only the weakest system and still ensure overall stability. Therefore, the problem of selecting which networked closed-loop system to update among all of them requires computing the global function *weakest energy contribution* (15) that takes information, *energy contributions* (13), from the different and physically distributed sampler nodes. This function has to be computed efficiently and with a small (and bounded) number of transmissions.

The key idea of the solution adopted in this paper, and inspired in the approaches presented by [10] or [8], is to observe that CAN permits to schedule messages on a priority based semantics, and that a feasible implementation for selecting the node with weakest energy could take advantage of this property provided that nodes are perfectly synchronized. Hence, we propose to encode the energy contribution v_i of each networked control system into each sensor-to-controller message identifier in order to have the solution implemented at the CAN bitwise arbitration resolution. Then, each time the event-condition for the N -NCS holds, all sampler nodes

try to send the sensor-to-controller message at the same time instant. However, the distributed bitwise arbitration of CAN only grants bus access to the sampler-to-controller message with highest priority, that is, the sampler message with the weakest contribution. Then, the controller node, after receiving a sampler-to-controller message, computes the control signal using L_i and the i^{th} state (also coded in the incoming message), and sends it to the corresponding actuator.

Note that in the set-up we consider, after a i^{th} -sensor-to-controller message has been sent, no other messages will appear in the bus before the i^{th} -controller-to-actuator is sent. Another message could enter the bus if the time spent by the i^{th} -controller would be longer than the next event condition to hold, which is unlikely to happen. Time intervals between event conditions are of the order of milliseconds while controller computations take few microseconds in current micro-controllers. Future work will further study this aspect.

V. SIMULATION SET-UP

We present a simple simulation set-up consisting of a set of $N = 2$ networked control loops to illustrate the operation and the benefits of the presented approach. In fact, two similar approaches are presented:

- 2-NCS-1-LF: this corresponds to the 2-NCS under Lyapunov sampling approach defined in Section III. Only one control update belonging in this case to one of the two closed loop systems will be triggered each time the event condition (17) defined on a *single* Lyapunov function \mathcal{V} holds.
- 2-NCS-2-LF: this approach is introduced for comparative performance evaluation. Each networked control loop triggers its control updates when its own event condition of the form (4) (with inequality), defined in terms of its particular Lyapunov function, holds. This would be the direct but unrealistic application of the Lyapunov sampling approach [9] to NCS, unrealistic in the sense that no bus contention problems are considered. Although it is not realistic, it provides interesting numbers for baseline comparisons.

A. Plants, Controllers and Event Conditions

Let us consider that each plant is a double integrator system, where

$$A_1 = A_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_1 = B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

We arbitrarily select the gains

$$L_1 = L_2 = [225 \quad 30].$$

The initial conditions for both plants are slightly different and defined as

$$x_1(0) = \begin{bmatrix} 1 \\ -3 \end{bmatrix} \quad \text{and} \quad x_2(0) = \begin{bmatrix} 1 \\ -4 \end{bmatrix}.$$

We call *first system* to $(A_1, B_1, L_1, x_1(0))$ and *second system* to $(A_2, B_2, L_2, x_2(0))$.

For the 2-NCS-1-LF, the event condition (17) depends on the Lyapunov function

$$\mathcal{V}(\mathbf{x}) = 3.416x_{1,1}^2 + 0.0089x_{1,1}x_{1,2} + 0.0168x_{1,2}^2 + 3.416x_{2,1}^2 + 0.0089x_{2,1}x_{2,2} + 0.0168x_{2,2}^2$$

characterized by

$$\mathcal{P} = \begin{bmatrix} 3.4167 & 0.0044 & 0 & 0 \\ 0.0044 & 0.0168 & 0 & 0 \\ 0 & 0 & 3.4167 & 0.0044 \\ 0 & 0 & 0.0044 & 0.0168 \end{bmatrix},$$

with $\eta_k = 0.9$, where $x_{i,j}$ denotes the j^{th} state of the i^{th} control loop. Whenever the event condition holds, we only update the weak control loop. To do so, we evaluate for both systems $i = 1, 2$

$$\begin{aligned} v_i &= \frac{\partial \mathcal{V}}{\partial x_i} (A_i x_i + B_i u_i) \\ &= 6.833x_{i,1}x_{i,2} + 0.0089x_{i,2}^2 \\ &\quad + 0.0089u_i x_{i,1} + 0.0336u_i x_{i,2} \end{aligned}$$

and update the control signal whose v_i is maximum.

For the 2-NCS-2-LF, for each i^{th} control loop, $i = 1, 2$, the same event condition (4) (with inequality) has been defined, depending on a Lyapunov function

$$V_i(x_i) = 3.416x_{i,1}^2 + 0.0089x_{i,1}x_{i,2} + 0.0168x_{i,2}^2$$

characterized by

$$P_i = \begin{bmatrix} 3.4167 & 0.0044 \\ 0.0044 & 0.0168 \end{bmatrix},$$

with $\eta_i = 0.9$. For each system, whenever the event-condition holds, the corresponding control update is triggered.

Each sensor node oversamples each plant with $h = 0.001s$.

VI. SIMULATION RESULTS

Figure 1 shows the systems' responses, that is, the states evolution $x_{i,1}$ and $x_{i,2}$ for both systems. In addition, it plots on top of the states evolution the triggering of the control updates: circles for the first system and squares for the second system. As it can be seen in the figure, both responses, starting from the different initial conditions, are very similar. The interesting observation is that control updates for each system occurs during different time intervals, which means that during those intervals, the system with control updates was the weak system. In Figure 1 and also in the following figures, we plot for both systems a control update at time zero or at the beginning of the simulation run. This has been done on purpose only for fair comparative analysis. Hence, it should be ignored, and the first *real* control update always occurs at time $t > 0$.

The following list explains the first steps taken in the operation of the 2-NCS-1-LF approach illustrated in Figure 1:

- 1) starting from $\mathbf{x}(t_k = 0) = [1 \quad -3 \quad 1 \quad -4]^T$, the 2-NCS system energy is $\mathcal{V}(\mathbf{x}(t_k)) = 7.19$, data that is known at each sensor node;
- 2) then each sensor node has to detect when $\mathcal{V}(\mathbf{x}(t_{k+1}))$ will be less or equal than $0.9 \cdot \mathcal{V}(\mathbf{x}(t_k)) = 6.47$;

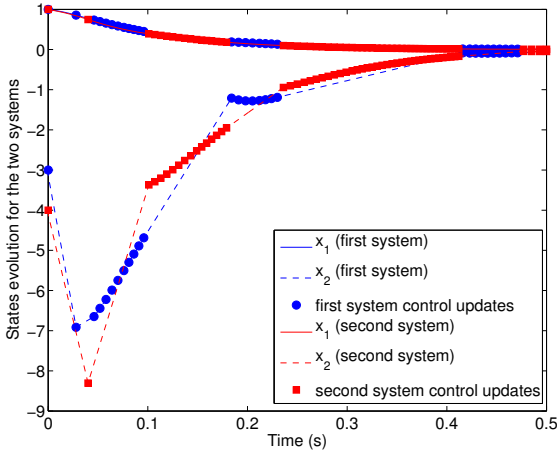


Fig. 1. States evolution for the 2-NCS-1-LF

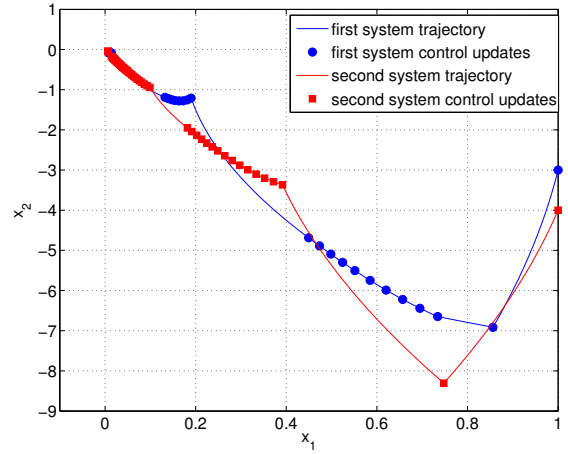


Fig. 2. Trajectories for the 2-NCS-1-LF in the phase portrait

- 3) to do so, each sensor node, at each millisecond, measures its own state variables, observes the state variables of the other system, and checks whether the event condition holds;
- 4) and this happens at time $t_{k+1} = 0.024s$, when $\mathbf{x}(0.024) = [0.85 \quad -6.91 \quad 0.83 \quad -7.04]^T$, because $\mathcal{V}(\mathbf{x}(0.024)) = 6.44$;
- 5) then, both sensor nodes compute their energy contribution, $v_1 = -9.66$ and $v_2 = -15.89$, and send the sensor-to-controller message with identifier codifying v_i , as explained in Section IV-B;
- 6) the CAN bitwise arbitration grants the bus access only to the sensor node sending v_1 , because it provides the weakest energy contribution. This message triggers the control signal computation u_1 in the controller node. Then, this node sends the controller-to-actuator message, finishing the control update for the first system;
- 7) afterward, the process starts again. That is, step 2) and 3) have to be repeated to check whether $\mathcal{V}(\mathbf{x}(t_{k+1})) \leq 0.9 \cdot \mathcal{V}(\mathbf{x}(0.024)) = 5.80$;
- 8) similar to step 4) this happens at time $t_{k+1} = 0.040s$, when $\mathbf{x}(0.040) = [0.77 \quad -6.73 \quad 0.74 \quad -8.30]^T$, because $\mathcal{V}(\mathbf{x}(0.040)) = 5.78$;
- 9) similar to step 5), then, both sensor nodes compute v_i , $v_1 = -38.49$ and $v_2 = -13.19$, and send the sensor-to-controller message with identifier codifying v_i ;
- 10) in this case, the sensor sending v_2 , the weakest energy contribution, wins bus access. This message triggers the control signal computation u_2 in the controller node. Then, this node will send the controller-to-actuator message, finishing the control update for the second system;

After performing step 6), it can be identified (and also observed in Figure 1) that the first control update for the first system occurs at $0.024s$. And after performing step 10), it can be identified that the first control update for the second system occurs at $0.040s$. Afterward, by looking at Figure 1, it can be

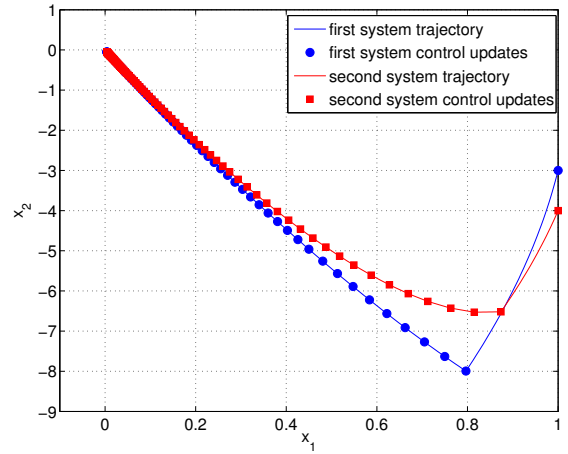


Fig. 3. Trajectories for the 2-NCS-2-LF in the phase portrait

also observed that 10 consecutive control updates occur for the first system, then 15 for the second system, etc...

Figure 2 gives a complimentary view of Figure 1, plotting both systems' trajectories in the phase portrait, as well as, the occurrence of control updates. For comparison purposes, Figure 3 gives the same information than figure 2 but for the 2-NCS-2-LF approach. In this case, for each closed-loop system, the sensor triggers the control update each time the "individual" event condition holds.

Comparing Figures 2 and 3 it can be observed that the 2-NCS-1-LF approach has the advantage of reducing the overall number of control updates. In particular, during the $0.5s$ simulation run, the 2-NCS-1-LF demands 87 control updates, 30 for the first system and 57 for the second system. Meanwhile, the 2-NCS-2-LF demands 182 control updates, 91 for each system. On the other hand, comparing both figures it can be observed that the trajectories achieved by the 2-NCS-1-LF are not as smooth as those obtained by 2-NCS-2-LF, thus indicating lower control performance.

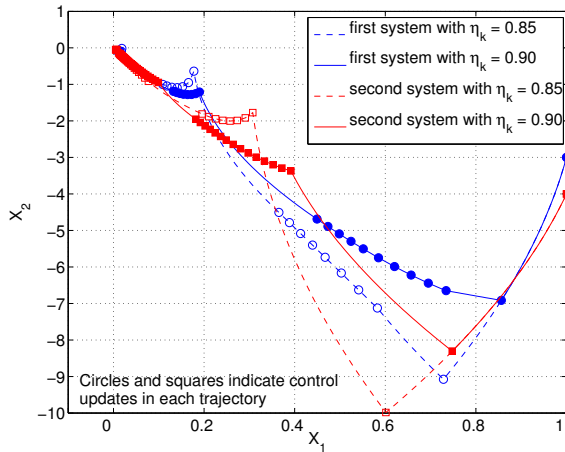


Fig. 4. 2-NCS-1-LF trajectories and control updates with a different η_k

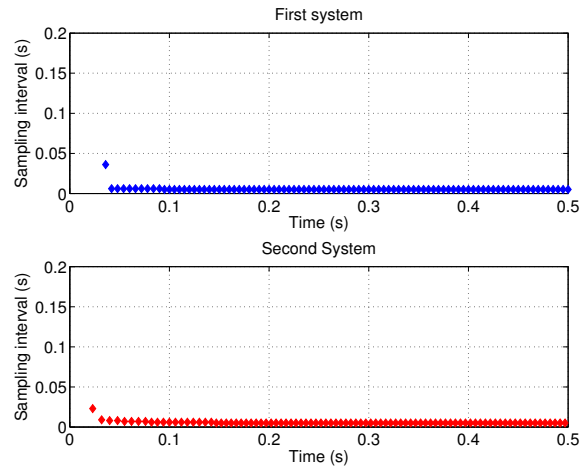


Fig. 6. 2-NCS-2-LF sampling intervals

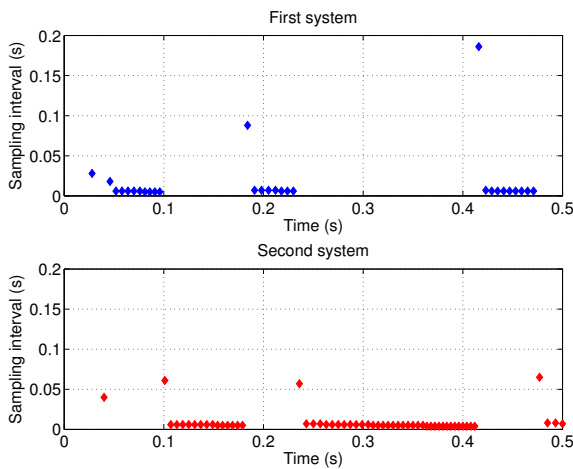


Fig. 5. 2-NCS-1-LF sampling intervals

In addition, if in 2-NCS-1-LF we change the decay ratio $\eta_k = 0.9$ to 0.85 , the number of control updates is reduced, as shown in Figure 4. However, the shape of the system trajectories indicate further control performance degradation.

Figures 5 and 6 provide complimentary information on the resource demands of the networked controllers in terms of the sampling intervals. As it can be seen in Figure 5, when one system uses the network, the other one does not. Therefore, serialization of transmissions is achieved. In addition, comparing Figures 5 and 6, it can also be seen that sampling intervals for 2-NCS-1-LF are clearly longer than the ones obtained by 2-NCS-2-LF, hence, demanding less network bandwidth.

VII. CONCLUSIONS

This paper has presented an approach to networked control systems where control updates are triggered i) according to a Lyapunov-based event-condition and ii) in such a way that transmission are serialized. It has been shown to be beneficial in terms of reducing transmissions. Future work will focus

on a deeper study of the Lyapunov-based event-condition for the N -NCS in order to identify which properties can ensure overall stability when control updates are serialized. A physical implementation of the approach is also planned.

ACKNOWLEDGMENT

This work was partially supported by NoE ARTISTDESIGN IST-FP7-2008-214373, and by Spanish Ministerio de Educación y Ciencia Project ref. CICYT DPI2007-61527.

REFERENCES

- [1] CAN Specification version 2.0. Robert Bosch GmbH, Postfach 30 02 40, D-70442 Stuttgart, 1991.
- [2] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. van den Bosch, "Analysis of event-driven controllers for linear systems," *Int. Journal of Control*, 81(4), 571–590, 2008.
- [3] T. Henningson, E. Johansson, A. Cervin, "Sporadic Event-Based Control of First-Order Linear Stochastic Systems," *Automatica*, vol. 44, n. 11, pp. 2890–2895, Nov. 2008
- [4] M. Mazo and Paulo Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks", in Proceedings of the 47th Conference on Decision and Control, 2008.
- [5] D. Nesić, and D. Liberzon, "A Unified Framework for Design and Analysis of Networked and Quantized Control Systems", *IEEE Transactions on Automatic Control*, vol. 54, n. 4, pp. 732–747, April 2009.
- [6] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, n. 9, pp. 1680–1685, 2007.
- [7] Y. Tipsuwan and M. Y. Chow, "Control methodologies in networked control systems", *Control Engineering Practice*, Vol. 11, 2003.
- [8] M. Velasco, P. Martí, R. Castañé, J. Guardia and J.M. Fuertes, "A CAN Application Profile for Control Optimization in Networked Embedded Systems", 32th Annual Conference of the IEEE Industrial Electronics Society, 2006.
- [9] M. Velasco, P. Martí and E. bini, "On Lyapunov Sampling for Event-driven Controllers", submitted to 48th IEEE Conference on Decision and Control, 2009. Available only for reviewing purposes at <http://paginaspersonals.upcnet.es/~pmc16/09CDCEvents.pdf>
- [10] G.C. Walsh, and Y. Hong, "Scheduling of networked control systems," *IEEE Control Systems Magazine*, vol. 21, n. 1, pp. 57–65, Feb. 2001
- [11] X. Wang and M. Lemmon, "Event-triggered Broadcasting across Distributed Networked Control Systems", American Control Conference, 2008.
- [12] X. Wang, and M. Lemmon, "Self-triggered Feedback Control Systems with Finite-Gain \mathcal{L}_2 Stability" *IEEE Transactions on Automatic Control*, Vol. 53, N. 3, pp 452-467, March 2009.