# Managing Quality-of-Control in Network-Based Control Systems by Controller and Message Scheduling Co-Design

Pau Martí, *Member, IEEE*, José Yépez, *Member, IEEE*, Manel Velasco, Ricard Villà, and Josep M. Fuertes, *Member, IEEE*

***Abstract*—In network-based control systems (NCSs), plant sensor–controller–actuator nodes in closed-loop operation drive principal network traffic. The quality-of-control (QoC) in an NCS, i.e., the performance delivered by each closed-loop operation, depends not only on the controller design but also on the message scheduling strategy. In this paper, we show that the co-design of adaptive controllers and feedback scheduling policies allows for the optimization of the overall QoC. First, we discuss the limitations of standard discrete-time control models for controllers of control loops that are closed over communication networks. Afterwards, we describe an approach to adaptive controllers for NCS that: 1) overcomes some of the previous restrictions by online adapting the control decisions according to the dynamics of both the application and executing platform and 2) offers capabilities for dynamic management of QoC through message scheduling.**

***Index Terms*—Communication systems, control systems, delay effects, discrete-time systems, quality control, real-time systems, scheduling.**

## I. Introduction

In the automation field, network-based control systems (NCSs) involve the cooperation of a set of processing devices (sensors, actuators, controllers, etc.) that run one or several control tasks, which communicate control data across a field-level communication network (fieldbus). The successful design and implementation of an NCS requires an appropriate integration of several disciplines (including control systems, real-time systems and communication systems) because the key for such systems is that almost no local action can be taken in isolation from the rest of the system. As we show in this paper, the quality-of-control (QoC) in an NCS, i.e., the performance delivered by each closed-loop operation that takes place over a communication network, can be managed by an approach based on the co-design of adaptive controllers and feedback scheduling algorithms.

In an NCS, the insertion of networks into control loops makes the analysis and design of control applications complex [1] because of the network time delays (or latencies) within related sampling and actuation instants. Note that depending on the network traffic, at each closed-loop operation, each control message may be subject to different delays, which at the end may degrade the performance of the closed-loop operation. The contribution of this paper is to present a method for the analysis and design of an NCS that allows us to manage the QoC of the NCS. It is based on considering the message latencies at the controller design stage and on providing mechanisms for QoC management by scheduling messages according to the control application demands, which may be expressed using a dynamic cost function (using the idea of feedback from the control application, as in [2]).

We start reviewing the effects of network time delays on the performance of closed-loop control systems for different control models. Next, we summarize a run-time control adaptive technique [3] that permits us to design controllers which are able to adjust their behavior according to network delays. Then, we show through simulation results that such controllers, which execute dynamically (with asynchronous actuation), can obtain better control performance that controllers executing statically (with synchronous actuation), as reported in [4].

The asynchronous actuation strategy for the NCS allows us to argue that the performance of the NCS depends not only on the control strategy but also on the message scheduling algorithm, as outlined also in [5]. We describe how adaptive controllers can be used to take advantage of the effects of time delays in system performance for QoC optimization. We formalize the notion of QoC (similar to [6]), and study which criterion must be used to assign the network to nodes (or messages), i.e., message scheduling, in order to optimize the QoC. We formulate a scheduling strategy that uses feedback information from the control application in order to schedule messages in such a way that the degrading effects of the message latencies are minimized, thus improving the overall QoC (see [2] for an earlier approximation of the scheduling strategy). We argue that the scheduling strategy is the solution of a constrained optimization problem specified in terms of a dynamic cost function that accounts for the state of each networked closed-loop system, subject to the communication bandwidth constraint.

The remainder of this paper is organized as follows. Section II reviews control models for the NCS and related performance issues. Section III summarizes the control technique that allows controllers to be adaptive according to network time delays. In Section IV we define the QoC metric related to time delays and present the scheduling strategy to be used for QoC optimiza-

Manuscript received August 29, 2003; revised May 31, 2004. Abstract published on the Internet September 10, 2004. This work was supported in part by the Spanish Ministerio de Ciencia y Tecnologia Project DPI2002-01621.

The authors are with the Department of Automatic Control and Computer Engineering, Technical University of Catalonia, 08028 Barcelona, Spain (e-mail: pau.marti@upc.es; jose.yepez@upc.es; manel.velasco@upc.es; ricard.villa@upc.es; josep.m.fuertes@upc.es).

Digital Object Identifier 10.1109/TIE.2004.837873

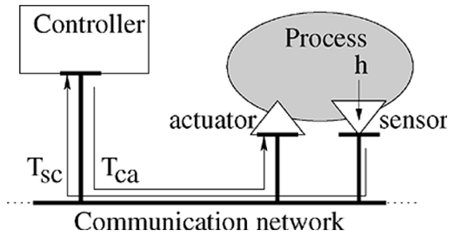


Fig. 1. Closed-loop operation in an NCS.

tion. Finally, Section V concludes and points out future research topics.

## II. Control Strategies for NCS

The functionality of a network-based closed-loop system can be described as the sequence of three main operations (sampling, computation, and actuation) that have to be repeatedly executed at every sampling period $h$, keeping a strict timing, in order to deliver the expected performance. If the networked computing platform does not give the expected determinism or the mathematical models do not accurately reflect the behavior of the computing platform, the system may fail. That is, system performance decreases.

### A. Mathematical Model of Sampling a System With Time Delay

A mathematical model that can be used in the analysis and design of an NCS (see Fig. 1) is described next. Note that for industrial processes such as in an NCS (where networks are used to carry feedback/feedforward signals for the closed-loop operation) it is common practice to include time delays in the mathematical model because the limited bandwidth induces unavoidable communication delays (sensor-to-controller delay $\tau_{\rm sc}$ and controller-to-actuator delay $\tau_{\rm ca}$, as illustrated in Fig. 1. Note that the time delay $\tau$ (time elapsed between related sampling and actuation instants) is given by $\tau = \tau_{\rm sc} + \tau_{\rm ca}$ ($\tau$ could also include any delays caused by computations on the sensor, controller, or actuator node).

For periodic sampling with (constant) sampling period $h$, a state-space discrete-time model of sampling a system with a constant time delay $\tau$ ($\tau$ is assumed to be less than or equal to the sampling period $h$, which implies that during each sampling period, two control signals, $u(kh-h)$ and $u(kh)$, will apply) can be described by (1) and (2) [7]

$$x(kh+h) = \Phi(h)x(kh) + \Gamma_0(h,\tau)u(kh) + \Gamma_1(h,\tau)u(kh-h) \tag{1}$$

$$y(kh) = Cx(kh) + Du(kh). \tag{2}$$

A state-space model of (1) and (2) is given by (3), where the past value of the control signal has been introduced as a new state variable ($z(kh) = u(kh-h)$)

$$\begin{bmatrix} x(kh+h) \\ z(kh+h) \end{bmatrix} = \begin{bmatrix} \Phi(h) & \Gamma_1(h,\tau) \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x(kh) \\ z(kh) \end{bmatrix} + \begin{bmatrix} \Gamma_0(h,\tau) \\ I \end{bmatrix} \cdot u(kh). \tag{3}$$

To meet the closed-loop system requirements, the system specified by (3) can be controlled using state feedback (4) where the gain matrix $L$ can be obtained by a design method such as pole placement or optimization approach [7]

$$u(kh) = -L(h,\tau) \begin{bmatrix} x(kh) \\ z(kh) \end{bmatrix}. \tag{4}$$

Equation (3) can be rewritten in terms of (4) as in (5)

$$\begin{bmatrix} x(kh+h) \\ z(kh+h) \end{bmatrix} = \left( \begin{bmatrix} \Phi(h) & \Gamma_1(h,\tau) \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \Gamma_0(h,\tau) \\ I \end{bmatrix} \cdot L(h,\tau) \right) \begin{bmatrix} x(kh) \\ z(kh) \end{bmatrix}. \tag{5}$$

At the end, the closed-loop time-invariant system is characterized by (5). The closed-loop matrix (6) described previously in (5) depends on $\Phi(h), \Gamma_0(h,\tau), \Gamma_1(h,\tau)$, and $L(h,\tau)$, which are constant matrices in terms of a constant sampling period $h$ and time delay $\tau$

$$\Phi_{cl}(h,\tau) = \begin{bmatrix} \Phi(h) & \Gamma_1(h,\tau) \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \Gamma_0(h,\tau) \\ I \end{bmatrix} \cdot L(h,\tau). \tag{6}$$

For discrete-time systems with a closed-loop matrix specified by (6), we can describe the closed-loop system evolution as in (7)

$$\begin{aligned} k=1 \quad & x(h) = \Phi_{cl}(h,\tau)x(0) \\ k=2 \quad & x(h+h) = \Phi_{cl}(h,\tau)x(h) \\ & = \Phi_{cl}(h,\tau)\Phi_{cl}(h,\tau)x(0) = \Phi_{cl}^2(h,\tau)x(0) \\ & \vdots \\ k+1 \quad & x(kh+h) \\ & = \Phi_{cl}(h,\tau)x(kh) = \Phi_{cl}(h,\tau)\Phi_{cl}^k(h,\tau)x(0) = \Phi_{cl}^{k+1}(h,\tau)x(0). \end{aligned} \tag{7}$$

Note that the closed-loop system evolution specified by (7) depends on a product sequence of $k$-equal closed-loop matrices. Note that the model assumes that the sampling is performed by the sampler at equidistant time instants given by the sampling period $h$, and the time delay $\tau$ is constant. Although these assumptions are the basis for systems with time delays in discrete-time control theory, when resources are limited, it is not possible to keep all of them in practice, as we discuss next.

### B. Effect of Time Delays in the Performance of Control Systems

The primary performance evaluation of closed-loop systems is concerned with meeting the closed-loop response performance specifications and stability (see, for example, [8]). Beyond these requirements, since controller designs attempt to minimize the system error to certain anticipated inputs, traditional performance criteria focus on the system error. The system error is defined as the difference between the desired response of the system and the actual response of the system. The smaller the difference, the better the performance.

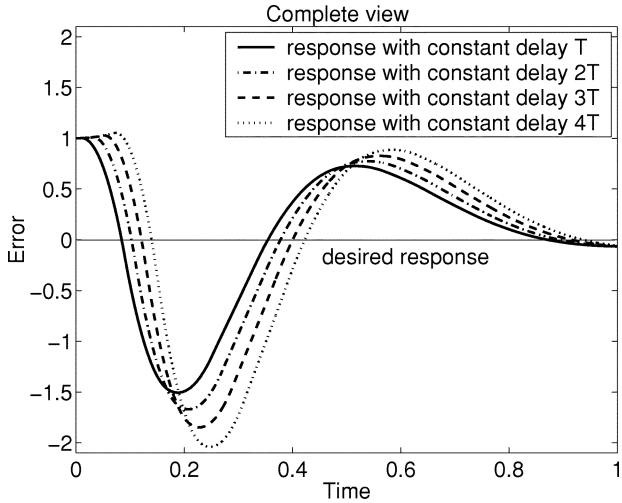


Fig. 2. Complete view of the performance of a generic controlled system for four different time delays.

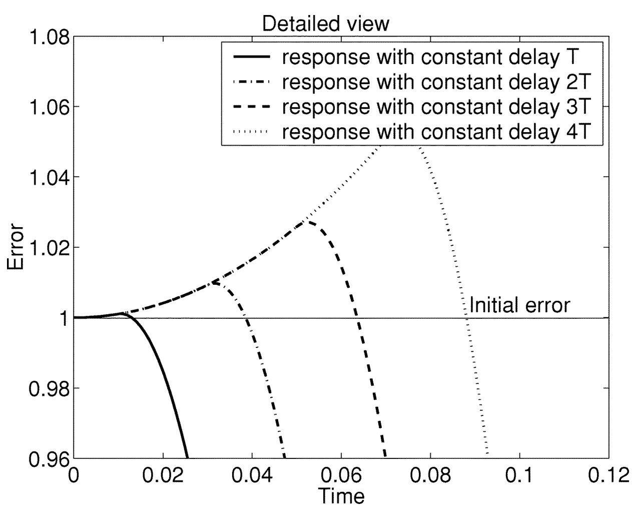


Fig. 3. Detailed view of the performance of a generic controlled system for four different time delays.

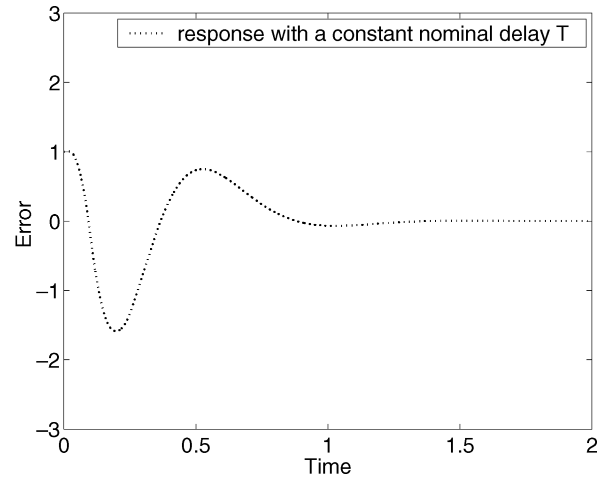


Fig. 4. System response with a constant time delay.

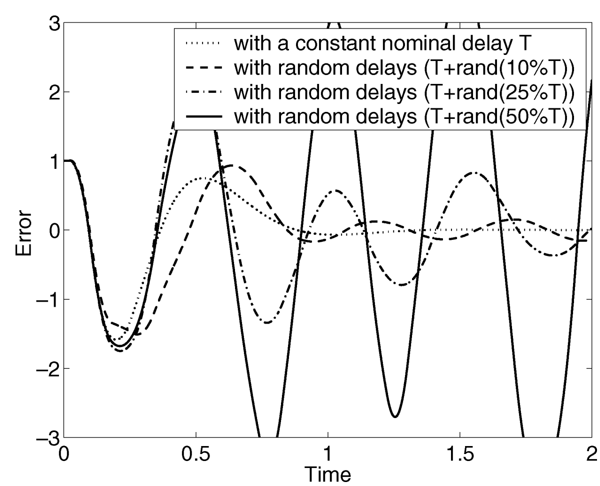


Fig. 5. Degradation on the system response due to varying time delays.

Using the model for systems with time delays (of Section II-A) we describe the effects of network time delays on the performance of closed-loop control systems. Recall that we are evaluating time delays in the sense of the time elapsed from sampling to actuation. To do so, we first focus on constant time delays and later on varying time delays.

*1) Effect of Constant Time Delays:* For illustrative purposes, if we experiment with constant sampling-to-actuation delays (multiples of a nominal delay $\tau$) and include them into the controller algorithm that is executing in a controller node of an NCS, we can summarize the effect they have in the closed-loop operation as shown in Figs. 2 and 3. Note that the system we consider is an NCS where nodes (sensor, controller, and actuator) communicate data across a network which introduces a constant time delay for the sensor-to-controller and controller-to-actuator messages. In Figs. 2 and 3 we show the complete view and the detailed view of four responses of a generic controlled system affected by a perturbation (of magnitude 1 in the error axis) for four different constant sampling-to-actuation delays ($\tau$, $2\tau$, $3\tau$, and $4\tau$). Although the controller has been designed to account for the delays, there will always be degrading effects on the controlled system response. The degrading effects on system performance can be summarized as follows.

- *Delayed Response*: The first effect is that a sampling-to-actuation delay delays the application of each control action to the plant.
- *Increased Error*: The second effect is that the later the control action is applied to the process (the longer the delay), the bigger is the error that the closed-loop system response suffers (see in Fig. 3 the relation between each delayed response and the difference between when each curve starts to correct the deviation and the initial error).

Although the degrading effects, it has to be pointed out that time delays can be assumed in closed-loop systems (using the model reviewed in Section II-A) as far as they are constant, known at design stage, and that the degradation they introduce can be made to conform to the performance specifications (for further reading on this topic, see, for example, [9]).

*2) Effect of Varying Time Delays:* Alternatively, if the communication delays are not constant but exhibit fluctuations, care must be taken because the system performance may drastically decrease as we show next. In Figs. 4 and 5, we illustrate the effects of varying communication time delays in system performance. The system considered is an NCS that consists of a set of control and noncontrol nodes that communicate data across a communication network. The control nodes act as before. The noncontrol nodes increase the traffic load and introduce unexpected longer latencies, inducing varying time delays for the sensor-to-controller and controller-to-actuator messages.

The generic system response obtained by the distributed controller if it would execute in isolation (on a networked architecture but with no competitors, thus, with a constant time delay) in the presence of a perturbation can be seen in Fig. 4. The previous response and the response obtained by the distributed controller executing in the networked architecture and suffering varying time delays due to the competitor's interferences is shown in Fig. 5. As can be seen in Fig. 5, the response suffers different degrees of degradation depending on the randomly induced communication delays. Concretely, in Fig. 5 we show four curves: the dotted line corresponds to the ideal response of the system, with a constant time delay $\tau$ (same as in Fig. 4). The other three curves correspond to three responses obtained by three controllers, each one suffering random delays of different magnitudes (delays longer than the nominal $\tau$ but bounded by 10%, 25%, or 50% of $\tau$). The longer the random generated delay, the bigger the degradation. Note that for the case with longest delays (delays bounded by 50% of $\tau$), the system becomes unstable.

The degradation caused by varying time delays can be explained as follows. From a control perspective, the control system with varying delays is no longer time invariant. Therefore, the standard computer control theory (targeted for time-invariant systems) cannot be used in analysis and design of NCSs with varying time delays.

### C. Related Work

The problem of analysis and design of control systems when the communication delays are varying in a random fashion is complex and it is still an open research field (see [10] for an introductory tutorial). Following the seminal work on communication and control [11], an approach to overcome these problems is to model these delays as probabilistic distributions and to treat them in the controller design stage. For example, [1] presents various probabilistic communication delays models and accordingly solves an LQG optimal control problem for them. Another approach is to assume a constant sampling-actuation delay, forcing a synchronous actuation (synchronous with respect to sampling) [12]. This can be achieved by forcing the actuation to occur at equidistant times, given by the longest possible delay. However, this may unnecessarily impose a longer time delay than the ones that appear at run time, thus forcing a pessimistic timing behavior in the closed-loop system that may result in a graceful but unnecessary system performance degradation, as pointed out in [4].

## III. An Adaptive Control Technique for Controllers Subject to Varying Time Delays

In this section we review a control-based technique (see [3]) that allows the design of controllers that are able to adjust their behavior according to the varying network delays.

### A. Problem Description

Looking at the system equations given by (1), (2), and (4), in an NCS, the controller will fail for two major reasons. First, the state feedback control law has been designed assuming to be constant but in reality the sampling-to-actuation delay will vary at each control loop execution. Secondly, the control algorithm mandates that the sampling period and the sampling-actuation delays must be known at the beginning of each controller execution. The sampling period is chosen at the design stage, and, since in our architecture we assume that the sampler node is performing strictly periodic sampling, $h$ is known and constant. However, the varying sampling-actuation delays introduced by the network cannot be completely known at the beginning of each controller execution because unfortunately, controller-to-actuator delay is unknown (sensor-to-controller delay can be known, for example, using time stamping). The only thing we can guarantee is that this former delay will be bounded. We assumed bounded time delays because, for our systems, we consider industrial communication networks with real-time capabilities, that is, supporting timeliness guarantees (see [13] for a discussion on existing fieldbus protocols and timeliness guarantees). Consequently, although delays may vary, they will always vary below a threshold guaranteed by the particular communication network, that we call "fieldbus worst case message latency."

In order to overcome this problem and to meet the required implementation timing constraint, what we need to know at the beginning of each controller execution is the exact sampling-actuation delay. This means, for example, that if the controller is executed immediately before the output is sent to the plant, we will have the information we need. To achieve this requirement, what we can do is to change the architecture of our distributed control loop. If we move the control computation from the controller node to the actuator node and remove the controller node, the problem is solved (see further details in [14]). However, further analysis is needed in the controller design, because it was designed for constant time delays and in this new architecture the system is still suffering varying but known time delays. The control design strategy must take into account this variation. What we know is that this variation will be limited, that is, $\tau$ will take values within a known discrete rank, which can be determined at the design stage (offline) according to the fieldbus characteristics that are being used. Therefore, at each $j$-execution of the control computation, the sampling-actuation delay ($\tau_j$) will take a finite number of values within the generalized rank described by $0 \leq \tau_\sigma \leq$ fieldbus worst case message latency, $\tau_\sigma \in \{\tau_j : j \in J\}$ where the index set $J$ is a finite subset of $\mathbb{N}$, and $\sigma = \sigma(k) : \mathbb{N} \rightarrow J$ is a map.

### B. Method Formulation

Knowing this, at each execution, the control law can be updated at run time for each delay if it implements the sampled-data system with time delays described in (1), (2), and (4) formulated as (8)–(10) (where $z(kh) = u(kh - h)$, and matrices $\Phi(h)$, $\Gamma_0(h, \tau_{\sigma(k)})$, $\Gamma_1(h, \tau_{\sigma(k)})$, and $L(h, \tau_{\sigma(k)})$ are obtained at each controller execution for each sampling-to-actuation delay, $\tau_{\sigma(k)} = \tau_j$)

$$x(kh+h) = \Phi(h)x(kh) + \Gamma_0(h,\tau_{\sigma(k)})u(kh) + \Gamma_1(h,\tau_{\sigma(k)})u(kh-h) \quad (8)$$

$$y(kh) = Cx(kh) + Du(kh) \quad (9)$$

$$u(kh) = -L(h,\tau_{\sigma(k)})\begin{bmatrix} x(kh) \\ z(kh) \end{bmatrix}. \quad (10)$$

Consequently, the discrete-time system is no longer time invariant. The state feedback controller $L(h,\tau_j)$ in (10) is obtained at each controller execution using the same control design approach used in (4) but for the specific delays $\tau_j$. To do so, the controller can be either computed at run time if the overhead is negligible, or otherwise computed offline and stored for run-time table look up.

Therefore, the closed-loop matrix that applies at the $k$th closed-loop execution specified in (11) depends on $\Phi(h)$, $\Gamma_0(h,\tau_j)$, $\Gamma_1(h,\tau_j)$, and $L(h,\tau_j)$, which are varying matrices in terms of each delay $\tau_{\sigma(k)} = \tau_j$, and using the same map as before ($\Phi_\sigma \in \Omega = \{\Phi_j : j \in J\}$)

$$\Phi_{\sigma(k)}(h,\tau_{\sigma(k)}) = \begin{bmatrix} \Phi(h) & \Gamma_1(h,\tau_{\sigma(k)}) \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \Gamma_0(h,\tau_{\sigma(k)}) \\ I \end{bmatrix} \cdot L(h,\tau_{\sigma(k)}). \quad (11)$$

For discrete-time systems with a closed-loop matrix specified by (11), we can describe the closed-loop system evolution as in (12)

$$\begin{aligned} k=1 \quad & x(h) = \Phi_{\sigma(1)}(h,\tau_{\sigma(1)})x(0) \\ k=2 \quad & x(h+h) = \Phi_{\sigma(2)}(h,\tau_{\sigma(2)})x(h) \\ & = \Phi_{\sigma(2)}(h,\tau_{\sigma(2)})\Phi_{\sigma(1)}(h,\tau_{\sigma(1)})x(0) \\ & \vdots \\ k{+}1 \quad & x(kh+h) \\ & = \Phi_{\sigma(k+1)}(h,\tau_{\sigma(k+1)})x(kh) = \\ & = \Phi_{\sigma(k+1)}(h,\tau_{\sigma(k+1)})\Phi_{\sigma(k)}(h,\tau_{\sigma(k)}) \\ & \quad \ldots \Phi_{\sigma(2)}(h,\tau_{\sigma(2)})\Phi_{\sigma(1)}(h,\tau_{\sigma(1)})x(0). \end{aligned} \quad (12)$$

Since the closed-loop system evolution with varying sampling-actuation delays (12) is different from that of classic discrete-time systems with constant time delay (7), in the controller design method we have to analyze the stability and response of these new systems.

### C. Stability Analysis

Systems with closed-loop evolution given by (12) will be characterized by an infinite product of a finite number of matrices $\prod_i \Phi_{\sigma(i)}$ taken from $\Omega$.

For the stability analysis of such systems [15], in Corollary 2, is given the necessary and sufficient stability condition (where inequalities are in the sense of positive or negative definiteness [16]), as in (13)

$$\Omega \text{ asymptotically stable} \leftrightarrow \exists P > 0 : \Phi_j^T \cdot P \cdot \Phi_j - P < 0 \quad \forall \Phi_j \in \Omega^k, \quad k \geq 1. \quad (13)$$

Note that the application of this condition is not easy in terms of computability. To make it easier to analyze the stability of such systems, (14) and (15) list two sufficient although not necessary stability conditions that can also be applied. The first (14) was presented in [15], Corollary 1, and the second (15) was presented in [17], which is the simplest to apply. However, it is more conservative than the first

$$\text{If } \exists P > 0 : \forall \Phi_j \in \Omega, \Phi_j^T \cdot P \cdot \Phi_j - P < 0 \Rightarrow \Omega \text{ asymptotically stable} \quad (14)$$

$$\text{If } \forall \Phi_j \in \Omega,\ \Phi_j^T \cdot \Phi_j - I < 0, \Rightarrow \Omega \text{ asymptotically stable.} \quad (15)$$

Using these conditions, the closed-loop system stability can be assessed.

### D. Response Analysis

In this section, we first derive the mathematical tools for the evaluation of the response of a system specified as in (8)–(10) (that is, an adaptive discrete-time system where the controller gain is updated according to each time delay $\tau_j$). Afterwards, we analyze the type of response we obtain using a specific example.

In order to evaluate the response, we want to examine whether the adaptive controller strategy appropriately minimizes the closed-loop system error. To do so, we use the integral of the absolute error (IAE) performance criterion, which is defined by (16) where $e(t)$ is the system error and $|\cdot|$ denotes an appropriate norm. The integral upper limit could be any time $T$ marking the evaluation time interval

$$\mathrm{IAE} = \int_0^\infty |e(t)|\,dt. \quad (16)$$

If we assume the equilibrium point to be 0 (without losing generality), then the system error is the same as the system output $y(t)$, and (16) can be rewritten as in (17)

$$\mathrm{IAE} = \int_0^\infty |y(t)|\,dt. \quad (17)$$

Taking into account that the output (9) of our system model is given at each sampling period, (17) can be expressed as an infinite (or finite, if an evaluation time interval as been specified) sum of all the IAE values of the system output at time $k+1$, for each period $h$, as in (18) with $0 < t < h$

$$\mathrm{IAE} = \sum_{k=0}^{\infty} \int_0^h |y(kh+t)|\,dt. \quad (18)$$

For the sake of simplicity, assuming $D = 0$ in (9), and substituting the output $y(kh+t)$ in (18) by the system evolution determined by each closed-loop matrix (11), we obtain (19) (where $\Phi_{\sigma(k)}(t,\tau_{\sigma(k)})$ is the closed-loop matrix that applies at

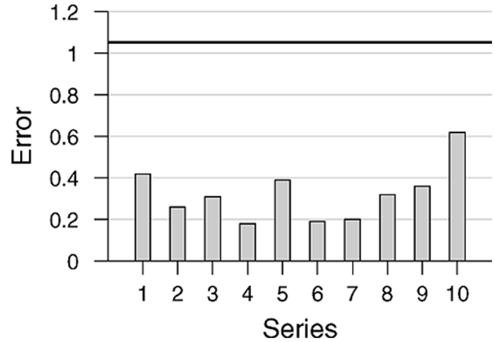


Fig. 6. Performance evaluation of the adaptive control technique.

the $k$th closed-loop execution (as in (11)), depending on each time delay, but specified for any time $0 < t < h$)

$$\mathrm{IAE} = \sum_{k=0}^{\infty} \int_0^h \left| C\Phi_{\sigma(k)}(t, \tau_{\sigma(k)}) \begin{bmatrix} x(kh) \\ z(kh) \end{bmatrix} \right| dt. \tag{19}$$

Using (19), the response of the adaptive discrete-time system specified by (8)–(10) can be evaluated. To analyze the type of response given by this adaptive control technique, we will compare the responses of an inverted pendulum (see [14] for more details on the model) that is controlled on an NCS, with a sampling period of 100 ms, and considering the following patterns for the delay:

- a constant but longer delay (thus being able to use the standard procedure given by (1), (2), and (4)) of 80 ms, which results in a synchronous actuation;
- varying but shorter delays (thus using the novel procedure given by (8)–(10)), $20 < \tau_j < 80$ ms, which results in an asynchronous actuation.

It is important to mention that when the actuation is synchronous, we are forcing a longer delay than the actual ones which implies to allow an unnecessarily degradation in the system. The performance criterion (20) we use is a measure of the system error, where $y_{\mathrm{nom}}$ is the nominal system response (obtained when assuming constant nominal time delay of 20 ms) and $y_{\mathrm{act}}$ is the current system response

$$V_a = \int_0^{\infty} |e(t)|\, dt = \int_0^{\infty} |y_{\mathrm{nom}}(t) - y_{\mathrm{act}}(t)|\, dt. \tag{20}$$

Fig. 6 shows the performance evaluation results. Notice that the error of the system response of the asynchronous approach (we run ten series, represented by the ten bars in Fig. 6, where delays randomly vary from 20 to 80 ms) is always smaller than the synchronous case (represented by the solid horizontal line in Fig. 6, where the delay is constant at 80 ms). Consequently, we can conclude that, for the inverted pendulum example, the performance of the control loop on an NCS with controllers that adjust their control decisions according to varying delays (asynchronous actuation) is better than compensating for a constant but longer delay (synchronous actuation). For further details, see [4].

## IV. Managing QoC in an NCS

For an NCS, our adaptive controllers (the asynchronous approach) offer the possibility to the network scheduler of taking message scheduling decisions based on control information for each closed-loop execution. This can be done by actively forcing different delays for each closed-loop execution, depending on the overall control application dynamics. Our adaptive controllers allow us to choose specific values for each message delay at each closed-loop execution. This will result in different levels of performance for each closed-loop system.

### A. Notion of QoC

We define the QoC metric in terms of the closed-loop system error. Since the aim of controllers is to minimize the error (the deviation that the controlled system response is subject to, due to perturbations), we define that better QoC will correspond to smaller errors (deviations). That is, there is an inverse relationship between the IAE index [8] and the QoC. For that reason, we define the QoC metric (21) in terms of: 1) the controlled system response error given by the IAE index and 2) a sequence of delays in the closed-loop operation (similarly as it was defined in [6])

$$\mathrm{QoC}(y_{\mathrm{act}} : \mathrm{seq}\langle \tau_{\sigma(k)} \rangle) = \frac{\frac{1}{IAE(y_{\mathrm{act}}:\mathrm{seq}\langle \tau_{\sigma(k)} \rangle)} - \frac{1}{IAE(y_{\mathrm{act}}:\mathrm{seq}\langle \max(\tau_\sigma) \rangle)}}{\frac{1}{IAE(y_{\mathrm{act}}:\mathrm{seq}\langle \min(\tau_\sigma) \rangle)} - \frac{1}{IAE(y_{\mathrm{act}}:\mathrm{seq}\langle \max(\tau_\sigma) \rangle)}}. \tag{21}$$

In (21) the IAE error evaluation time interval is the time elapsed from the time of occurrence of the perturbation to the settling time and $y_{\mathrm{act}} : \mathrm{seq}\langle \tau_\sigma \rangle$ denotes that the actual system response ($y_{\mathrm{act}}$) has been obtained in a closed-loop with a specific sequence of message delays ($\mathrm{seq}\langle \tau_\sigma \rangle = \tau_{\sigma(1)}, \tau_{\sigma(2)}, \ldots, \tau_{\sigma(k)}$). Note that $y_{\mathrm{act}} : \mathrm{seq}\langle \min(\tau_\sigma) \rangle$ and $y_{\mathrm{act}} : \mathrm{seq}\langle \max(\tau_\sigma) \rangle$ denote the actual system response if the control task runs always with the shortest or longest delay.

Note that the QoC values will fall in the range of [0,1] (due to the normalization), where 0 is equivalent to the lowest QoC and 1 is the best QoC. From the study of Section II-B, and using the new terminology, the following holds: *the shorter the delay (although constant for all its execution) for a closed-loop operation, the better the QoC (the smaller the error).*

### B. Influence of Different Sequences of Delay Values on the QoC

Using the QoC metric, now we are specifically interested in the influence of different delay values orderings on the QoC of the controlled system in the presence of perturbations. We divide the study into several cases in order to cover all relevant situations when evaluating the influence of different delays values orderings on the QoC. For the simulation study we use the same setup described in Section III-D.

Figs. 7 and 8 give representative graphs of theses cases (in each graph, each point along the $x$ axis represents a sequence of delays values, the corresponding QoC value plotted on the $y$ axis; the sequence is given as a column of values from top to

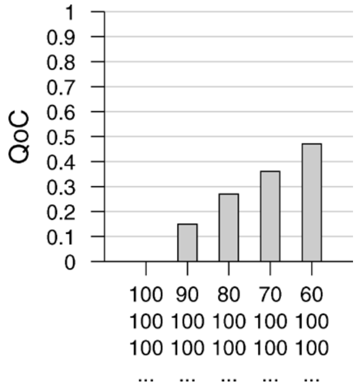


Fig. 7. Effect of the first delay value of a sequence of delays in the QoC.

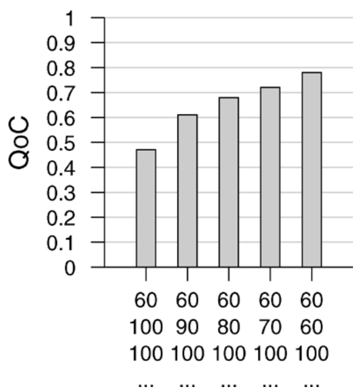


Fig. 8. Effect of the second delay value of a sequence of delays in the QoC.

bottom). We focus on a set of delays values obtained in the control analysis for the inverted pendulum problem as before. The closed-loop operation is executing with the longest delay because the pendulum is initially in equilibrium. In each case, after the perturbation arrival, we study the effects of the following.

- *First Delay Value*: Fig. 7 shows that the first value of each sequence has an important influence on the QoC of the system. If the control task can arbitrarily choose any value for the delay after the perturbation arrival, the smaller the chosen value, the better the QoC.
- *Second Delay Value*: Simulations have shown (Fig. 8) that whatever the first value is, the next delay value of each sequence also has an influence on the QoC of the system. As before, the smaller the value for the second delay value, the better the QoC.
- *Successive Delay Values*: Simulations show that whatever the first and second delay values are, the following delays values also have an influence on the QoC of the system in the sense that the smaller the value, the better the QoC.

The influence of different delay value orderings on the QoC of the controlled system response can be summarized as follows: *the shorter and earlier, although varying, delay values we have for messages of a closed-loop configuration, the better the QoC*.

### C. QoC Optimization as a Message Scheduling Problem

The previous study allows us to formulate the QoC optimization of an NCS as a message scheduling problem [2]. We mandate the following behavior for each closed-loop system. During

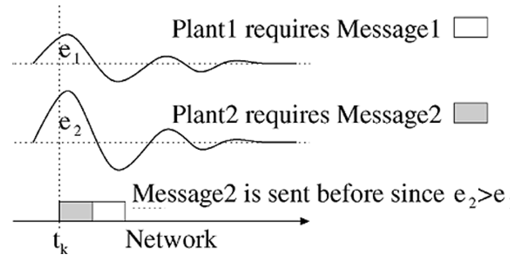


Fig. 9. Message scheduling using control performance information.

the time the controlled system is in equilibrium, the latencies for the controlling messages should have the longest possible value ($\max(\tau_\sigma)$). This way, the nodes in the closed-loop operation with the controlled system in equilibrium will give priority (i.e., bandwidth) to other nodes requiring shorter delays. Upon detection of a perturbation, high QoC must be achieved to counteract the perturbation, that is, by trying to shorten the latencies of the involved messages. The scheduling problem is illustrated in Fig. 9 and explained next.

Let us assume we have two plants that are controlled over a communication network. At each time $t_k$, a node acting as scheduler scans the error of each plant ($e_1$ and $e_2$ in Fig. 9) and assigns the network to the node with the highest error. At time $t_k$, the network is assigned to the node controlling $Plant_2$, because $e_2 > e_1$. This implies to decrease the corresponding message latency, thus improving the QoC. Consequently, the message sent at time $t_k$ is $Message_2$ (it includes all messages from the sensor of $Plant_2$ to the controller of $Plant_2$, and from the controller of $Plant_2$ to the actuator of $Plant_2$). Then, $Message_1$ is sent.

The main advantage of such new paradigm is the fact that it allows a dynamic adjustment of each closed-loop operation according to the application characteristics (i.e., the states of the controlled plants). This adjustment, which implies increasing or decreasing the messaging over the network, can be formulated as a constrained optimization problem where the goal is to maximize the QoC delivered by each networked closed-loop system, taking into account the state of each system, subject to the communication bandwidth constraint. Formally, QoC scheduling can be seen as a bandwidth allocation problem whose solution implies solving the optimization problem given by (22) and (23)

$$\text{maximize} \quad g(s_i(t), f_i(b_i)) \tag{22}$$

$$\text{subject to} \quad \sum_{i=1}^{n} b_i \leq B_d. \tag{23}$$

In (22), $b_i$ is the bandwidth share to be assigned to each closed loop, $f_i$ is the function that relates benefit (e.g., QoC) for a given amount of bandwidth, $s_i(t)$ is the run-time weighting factor that adjusts the relation between the amount of allocated bandwidth and the control application dynamics (e.g., $s_i(t) = |x(t)|$), and $g$ is the objective function to be maximized that links all benefits (e.g., sum of all QoCs). In (23), $B_d$ is the allowed (or desired) communication bandwidth for all messaging of the $n$ networked closed-loop systems. Although it is out of the scope of this paper, by studying functions $g$, $s_i$, and $f_i$, we can assess whether at run time to find the vector $\vec{b} = [b_1, b_2, \ldots, b_n]$ that maximizes $g$ is feasible or not.

## V. Conclusion

In this paper we have shown that managing QoC in an NCS can be carried out by jointly designing the networked controllers and the message scheduling strategy. When the communication resources are limited, sensors, controllers, and actuators, along with other nodes, have to compete for accessing the network. By using an adaptive technique for controllers, we have been able to choose when, according to the application dynamics, to send the required control messages for each closed-loop operation. This gave us the possibility of controlling the message latencies that affect each closed-loop operation, thus allowing us to mange the QoC of each controlled system.

In addition, we have shown how controllers can be dynamically executed to improve the QoC. This allowed us to formulate the QoC management as a message scheduling problem. We have shown that candidate scheduling strategies for QoC optimization rely on the idea of feedback, that is, on the idea of taking scheduling decisions by evaluating information (the state of the controlled plants) obtained from the control application. The optimization problem behind the QoC scheduling strategy has been formulated and discussed.

## References


[1] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, no. 1, pp. 57–64, 1998.

[2] J. Yépez, P. Martí, and J. M. Fuertes, "Control loop scheduling paradigm in distributed control systems," in *Proc. IEEE IECON'03*, Roanoke, VA, Nov. 2–6, 2003, pp. 1441–1446.

[3] P. Martí, R. Villà, J. M. Fuertes, and G. Fohler, "On real-time control tasks schedulability," in *Proc. IFAC Eur. Control Conf.*, Porto, Portugal, Sept. 4–7, 2001, pp. 2227–2232.

[4] J. Yépez, P. Martí, and J. M. Fuertes, "Control loop performance analysis over networked control systems," in *Proc. IEEE IECON'02*, vol. 4, Seville, Spain, Nov. 5–8, 2002, pp. 2880–2885.

[5] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Contr. Syst. Mag.*, vol. 21, pp. 57–65, Nov. 2001.

[6] P. Martí, J. M. Fuertes, G. Fohler, and K. Ramamritham, "Improving Quality-of-Control using flexible timing constraints: Metric and scheduling issues," in *Proc. 23rd IEEE Real-Time Systems Symp.*, Austin, TX, Dec. 3–5, 2002, pp. 91–100.

[7] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems. Theory and Design*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1997.

[8] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 7th ed. Reading, MA: Addison-Wesley, 1995.

[9] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "A design methodology for distributed control systems to optimize performance in the presence of time delays," in *Proc. American Control Conf.*, vol. 3, Chicago, IL, June 28–30, 2000, pp. 1959–1964.

[10] M.-Y. Chow and Y. Tipsuwan, "Network-based control systems: A tutorial," in *Proc. IEEE IECON'01*, vol. 3, Denver, CO, Nov. 29 –Dec. 2, 2001, pp. 1593–1602.

[11] Y. Halevi and H. Ray, "Analysis of integrated communication and control system networks," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 112, pp. 365–372, 1990.

[12] B. Wittenmark, B. Bastian, and J. Nilsson, "Analysis of time delays in synchronous and asynchronous control loops," in *Proc. 37th IEEE Conf. Decision and Control*, vol. 1, Tampa, FL, 1998, pp. 283–288.

[13] L. Almeida, P. Pedreiras, and J. A. G. Fonseca, "The FTT-CAN protocol: Why and how," *IEEE Trans. Ind. Electron.*, vol. 49, pp. 1189–1201, Dec. 2002.

[14] P. Martí, J. M. Fuertes, and G. Fohler, "An integrated approach to real-time distributed control systems over fieldbuses," in *Proc. 8th IEEE Int. Conf. Emerging Technologies and Factory Automation*, vol. 1, France, Oct. 15–18, 2001, pp. 177–182.

[15] M. Dogruel and U. Özgüner, "Stability of a set of matrices: A control theoretic approach," in *Proc. 34th IEEE Conf. Decision and Control*, vol. 2, Dec. 13–15, 1995, pp. 1324–1329.

[16] G. Strang, *Linear Algebra and its Applications*, 2nd ed. New York: Academic, 1980.

[17] P. Martí, R. Villà, J. M. Fuertes, and G. Fohler, "Stability of on-line compensated real-time scheduled control tasks," in *Proc. IFAC Conf. New Technologies for Computer Control*, Hong Kong, China, Nov. 19–22, 2001.



**Pau Martí** (M'02) was born in Catalonia, Spain, in 1969. He received the degree in computer science and the Ph.D. degree in automatic control from the Technical University of Catalonia, Barcelona, Spain, in 1996 and 2002, respectively.

Since 1996, he has been an Assistant Professor in the Department of Automatic Controland Computer Engineering, Technical University of Catalonia. From 1999 to 2002, he spent several months as a Visiting Student at Mälardalen University, Västerås, Sweden. From 2003 to 2004, he held a Research Fellow appointment in the Computer Science Department, University of California at Santa Cruz, where he was involved in research on soft real-time systems. His research interests are real-time control systems, with emphasis on the interaction and integration of control systems, real-time systems, and communications systems.



**José Yépez** (M'02) was born in Mérida, Venezuela, in 1968. He received the Systems Engineering degree and the M.Sc. degree in control engineering from the Universidad de Los Andes, Mérida, Venezuela, in 1994 and 1997, respectively. He is currently working toward the Ph.D. degree in automatic control at the Technical University of Catalonia, Barcelona, Spain.

During 1997–1998, he was an Assistant Professor in the Department of Control Systems of the Universidad de Los Andes. His interests lie in the field of real-time distributed control systems, with emphasis on the interaction and integration of control design, communication, and real-time scheduling.



**Manel Velasco** was born in Catalonia, Spain, in 1973. He received the degree in maritime engineering in 1999 from the Technical University of Catalonia, Barcelona, Spain, where he is currently working toward the Ph.D. in automatic control.

Since 2002, he has been an Assistant Professor in the Department of Automatic Control and Computer Engineering, Technical University of Catalonia. He was involved in research on artificial intelligence from 1999 to 2002. Since 2000, he has been involved in research on the impact of real-time systems on controllers and control systems. His research interests are artificial intelligence, real-time control systems, and collaborative control systems, in particular, redundant controllers and multiple controllers with self-interaction systems.

**Ricard Villà** was born in Catalonia, Spain, in 1944. He graduated in electrical engineering from the Technical University of Barcelona, Barcelona, Spain, in 1970, and received the Ph.D. degree in electrical engineering from the Technical University of Catalonia, Barcelona, Spain, in 1983.

In 1977, he become an Assistant Professor at the Technical University of Catalonia, where, since 1984, he has been an Associate Professor. He has been involved in industrial electronics, astronautical mission planning, and control systems.

**Josep M. Fuertes** (S'74–M'96) was born in Barcelona, Spain. He received the Ph.D. degree from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1986.

He became a Permanent Professor at the Technical University of Catalonia (UPC), Barcelona, Spain, in 1987. He was a Researcher (1975–1986) at the Institut de Cibernètica (Spanish Consejo Superior de Investigaciones Científicas). In 1987, he spent the year at the Lawrence Berkeley Laboratory working as a Visiting Scientific Fellow on the design of the Active Control System of the W. M. Keck 10-m segmented telescope (Hawaii). From 1990 to 1992, he was esponsible for the doctoral program of the Department of Automatic Control and Computer Engineering at UPC. From 1992 to 1999, he was the Director of the University Research Line in Advanced Control Systems and, since 2001, he has been the Director of the Department of Automatic Control and Computer Engineering, UPC. Since 1989, he has coordinated projects at the national and international levels related to his areas of expertise. He has authored or coauthored more than 80 papers on distributed control systems and applications.

Dr. Fuertes has collaborated as an organizer (IEEE-WFCS'97, IEEE-ETFA'99), chairman, session organizer, and member of program committee for several international conferences. He acted as the Spanish Representative at the Council of the European Union Control Association (EUCA). He was also an AdCom Member of the IEEE Industrial Electronics Society, and is a member of several other scientific and technical societies.