A delay-dependent dual-rate PID controller over an Ethernet network

Ángel Cuenca, *Member, IEEE,* Julián Salt, *Member, IEEE,* Antonio Sala, *Member, IEEE,* and Ricardo Pizá

***Abstract*—In this paper, a methodology to design controllers able to cope with different load conditions on an Ethernet network is introduced. Load conditions induce time-varying delays between measurements and control. To face these variations an interpolated, delay-dependent gain scheduling law is used. The lack of synchronization is solved by adopting an event-based control approach. The dual-rate control action computation is carried out at a remote controller, whereas control actions and measurements are taken out locally at the controlled process site. Stability is proved in terms of probabilistic linear matrix inequalities. TrueTime simulations in an Ethernet case show the benefit of the proposal, which is later validated on an experimental test-bed Ethernet environment.**

***Index Terms*—Networked control system, network delay, multi-rate control systems, stability analysis, PID controller.**

I. Introduction

NETWORKED Control Systems (NCS) [1], [2], [3] are control systems in which different devices (sensor, actuator, controller) are connected by means of a shared communication medium. NCS can be found in several kinds of control applications: teleoperation, supervisory control, avionics, chemical plants, etc. Their main advantages are wiring reduction, easier and cheaper maintenance, and cost optimization. Their main drawbacks are existence of time-varying delays, lack of synchronization among devices, bandwidth limitations, and packet dropouts.

The problem under focus in this work is the loss of performance arising when nominal controllers (particularly, PID ones) are implemented in a networked setting subject to time-varying delays between measurements and actuation, so a network-aware probabilistic stability analysis is proposed.

Firstly, in this paper network considerations are faced. Indeed, the variation of the network features (number of nodes connected to the network, bandwidth occupied by each node, etc) affect the network load, and hence, the probabilistic distribution of the sensor-to-actuator delay. Truetime [4] is a simulation tool that enables to easily simulate event-based networked control systems with random time-varying delays, and it is becoming a standard simulation tool in this research field. Using the Truetime features, a simulation study of the control application can be done; in particular, probabilistic information (histograms) on the control-relevant network delays. Such a simulation can, of course, be replaced by actual measurements on a real network if the suitable hardware and software for network monitoring is available. In this work, not only simulation results but also experimental ones are provided in order to validate the control proposal.

In the examples in this paper, the network setup considers Ethernet as the shared medium, and a Direct Control topology [1], that is, a remote node (controller device) which is connected to the plant through the network, and one local node (with actuator and sensor devices) with direct communication to the plant and a local clock. In Truetime a set of so-called "interference nodes" are considered in order to modify the network load. In the practical application the network traffic is varied by executing different kind of programs in several Programmable Logic Controllers (PLCs) and computers connected to the network.

In order to apply controller gain-scheduling methodologies, the delay measurement must be carried out for each sample. One option is to synchronize nodes [5], [6]. If an accurate delay measurement is required, the synchronization protocol needs to send a huge quantity of special messages, increasing network load and hence delays. In the control structure in this paper, another option is to measure the round-trip time delay directly at the local side. Since sensor and actuator share the same local clock, no synchronization is required.

However, if no synchronization between remote and local nodes were considered and every device followed a time-driven policy, loss and reutilization of samples and control actions (what is known as message rejection and vacant sampling [2]) could appear. These issues may be a potential source of instability.

In the context of small delays these problems can be avoided when an event-based approach [7] is adopted at the controller and actuator devices. In this way, whereas the sensor device periodically samples the plant output, the controller and actuator devices are only triggered when data become available from the network. Indeed, with delays shorter than the sampling time (as assumed in this work) all events take place before next measurement sample.

Delay variations usually imply a control system performance degradation with respect to a *nominal* no-delay case. Thus, some control considerations must be introduced to face this problem:

- first, an adequate choice and design of the controller,

Á. Cuenca, J. Salt, A. Sala and R. Pizá are with Departamento de Ingenieria de Sistemas y Automatica, Instituto Universitario de Automatica e Informatica Industrial, Universidad Politecnica de Valencia, Camino de Vera s/n, 46022 Valencia (Spain) (e-mail: {acuenca, julian, asala, rpiza}@isa.upv.es).

The authors A.Cuenca, J. Salt and R. Pizá are grateful to grants PAID06-08 by Universidad Politecnica de Valencia, DPI2009-14744-C03-03 from Spanish Ministry of Education, and GV/2010/018 by Generalitat Valenciana. In addition, A. Cuenca is grateful to grant DPI2008-06737-C02-01 by Spanish Ministry of Education, and A. Sala is grateful to the financial support of Spanish Ministry of Education research grant DPI2008-06731-C02-01, and Generalitat Valenciana grant PROMETEO/2008/088

- second, how this controller must compensate the time-varying delays,
- and finally, to prove closed-loop stability under the expected network behavior.

Regarding the first subject, a dual-rate setup [8], [9], [10], [11] (considering that the control period is faster than the measurement period) is adopted in order to improve performance with relation to a single-rate approach, and to maintain an unloaded network [12], [8]. More concretely, due to its practical interest [13], [14], a dual-rate PID controller will be utilized.

With respect to compensating the time-varying delays, as previously commented, the value of the current round-trip time delay is known at the actuator device. Then, the remote controller has no information about it to generate the current control action. This problem can be faced via different control proposals (for instance: delay estimation and robust control [6], robust state-feedback and time-varying observers [15], [16], etc). In this work, a novel approach is proposed, which is based on interpolating at the actuator some information provided by a delay-dependent dual-rate control law implemented at the controller node. Due to the actuator interpolation, several control actions are sent through the network as a packet. Hence, a compromise arises between the higher network load involved and the benefits to be obtained. Note, however, that in many cases, the minimum packet size and network protocol bytes make the overhead for sending several control actions almost insignificant: for the same total information transmitted, packet-based control [17] involves a much lower network activity.

The proposed control strategy is based on a gain scheduling approach presented in [18]. Other gain scheduling strategies in the context of networked control systems can be found in [19], [20].

Regarding the third control aspect to be treated, as the problem becomes time-varying, stability for arbitrary time-varying network delays in different network load situations must be proved. For each network scenario, a performance objective can be defined. So, linear matrix inequalities (LMI) [21] will be used in a probabilistic Linear Time-Varying (LTV) setting [22]. Considering different network load scenarios, a multi-objective study can be carried out.

In summary, the main objective of this paper is to improve the preliminary study developed in [18], incorporating

- a remote controller structure, separated from both actuator and sensor, which nevertheless does not require the exchange of time-stamped data.
- probabilistic information of delays: considering only one LMI constraint (average decay) instead of one for each possible sampling period; in this way, temporal random increases of the Lyapunov function are tolerated as long as the average over time is decreasing.
- a multi-objective setting considering different persistent network load conditions

There are also alternative approaches to stability analysis of networked control systems which could be adapted to the proposed setup, see [23], [24], [25], [26], [27], [28], [29], [30], [31]; the main ideas behind some of them are briefly commented on Section III-D. The reader is particularly referred to [32] for a combination of gridding with non-optimistic overbounding in a stochastic control case.

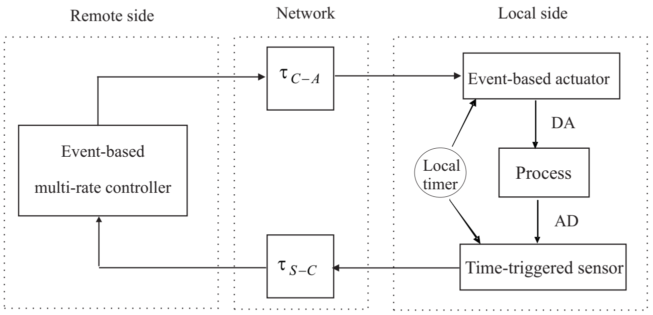

Fig. 1: Proposed networked control application

The structure of the paper is as follows. Firstly, in section II the methodology to study the network characteristics is introduced. In section III, the control considerations are presented (a review of the gain scheduling approach in III-A, a dual-rate PID controller realisation in III-B, how to interpolate the control signal in section III-C, and the control system stability in III-D). Section IV-A shows the expected improvements by means of simulation results and LMI analysis for the proposed example. A real implementation to validate the previous results is presented in section IV-B. Finally, section V enumerates the main conclusions.

II. Summary description of the networked-control setting

Consider a process (local node) and a remote controller interconnected by a shared communication network as described in Figure 1.

The local side includes:

- a time-triggered sensor (measuring at a period of NT where T is a basic sampling rate and N is a multiplicity factor),
- an event-based actuator, and
- the process to be controlled.

Both the sensor and the actuator are assumed to be connected to the process with a direct AD/DA link respectively.

The remote side includes an event-based multi-rate PID controller, which is connected to the plant via the network. The controller computation is triggered when a sample arrives from the network, and it sends a set of control actions to the event-based actuators (see next section).

At the local side, sensor and actuator are assumed to share a local timer [1]. So, both of them are perfectly synchronized, and the round-trip time delays can be measured when getting the "control-action-received" event time. For the k^{th} measurement y_k, the actual round-trip time delay is $\tau(k) = t_A(k) - t_S(k)$ where $t_S(k)$ is the latest sensor measurement time (a suitable multiple of NT) and $t_A(k)$ is the time of reception of the

[1] Note that, although a shared clock can be assumed in some applications, it might be hard to fulfill for other ones, for instance in multiple wireless sensor nodes that are physically distributed.

control-action packet at the actuator end. This delay is equal to

$$\tau(k) = \tau_{S-C}(k) + \tau_C(k) + \tau_{C-A}(k) \qquad (1)$$

where $\tau_{S-C}(k)$ is the sensor-to-controller delay, $\tau_{C-A}(k)$ is the controller-to-actuator delay, and $\tau_C(k)$ is a remote computation time delay.

As the controller structure chosen in this work (see section III) does not depend on the arrival time of information $\tau_{S-C}(k)$, the only relevant magnitude will be the lumped round-trip delay τ. This might not be the case for other control structures in which separate $\tau_{S-C}(k)$, $\tau_C(k)$, $\tau_{C-A}(k)$ might be relevant.

If the measured data is not time-stamped, the total round-trip delay can be computed at the local side by subtracting the latest measurement time $t_S(k)$ to $t_A(k)$; however, the lack of time-stamping and synchronisation makes the remote side unable to exploit this timing information. The subtraction procedure works under the assumption of no sample loss and $\tau(k) \in [0, \tau_{max}]$, being $\tau_{max} < NT$ (hence no packet disorder is assured). These conditions will be assumed to hold in the sequel.

If time-stamped data were transmitted in the same frame than sensor data, then $\tau_{S-C}(k)$ can be made available to the remote controller and, eventually, also $\tau_C(k-1)+\tau_{C-A}(k-1)$. This is the approach used in [15] where some gain-scheduling approaches were proposed on the remote side. The solution in this work will avoid the need of time-stamping data by performing gain-scheduling operations on the local side.

If the network link is shared between different processes and tasks, it will be a reasonable assumption that network delays may vary with time. This is the case in Ethernet networks [19], [33]. In other network protocols, such as Profibus, the sequence of communication operations can be made more predictable [34], [35] but, anyway, the delays will also change (although possibly predictably) if the measurement period NT is not a multiple of a basic network-polling period.

Irrespective of the nature of the underlying network, the procedures in this work change controller settings based on the measured delays. However, in the examples in later sections, the chosen network for simulation has been an Ethernet one, and the simulation tool has been Truetime [4].

In order to simulate different network load conditions when transmitting control information, the Truetime application has been suitably configured including additional network nodes inserting sporadic traffic, as discussed below.

A. Truetime configuration

Truetime is a well-known networked-control simulation application for Matlab/Simulink which has been chosen to demonstrate the proposals in this work. In order to simulate the above-proposed control structure (Figure 1), the needed components of the Truetime application appear depicted in Figure 2.

The above structure incorporates the local node, the remote controller node and two interference nodes generating network load. So, four parameters define the operation mode for each interference node:

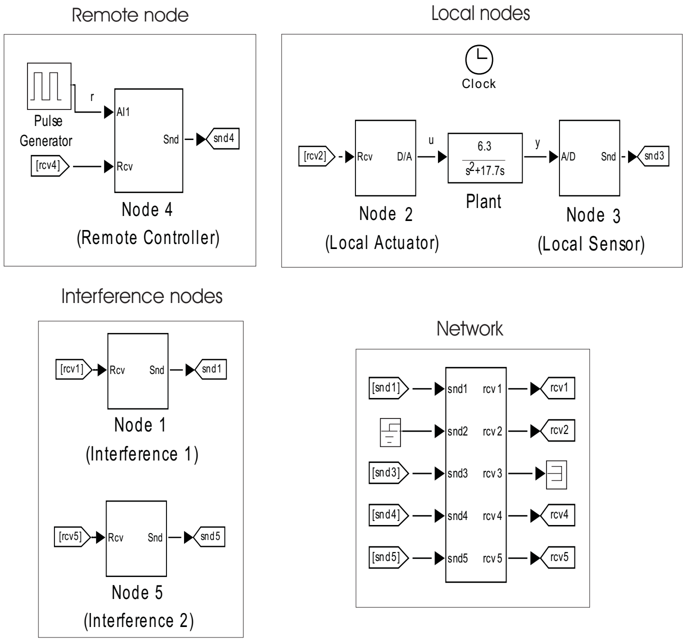

Fig. 2: View of the application developed in Truetime

- $Maxfs$, maximum frame size,
- $BWshare$, fraction of the network bandwidth occupied by the node ($0 \leq BWshare \leq 1$),
- BW, weighting index for the parameter $BWshare$. It is a random number defined as $0 \leq BW \leq 1$,
- $Nodes$, number of nodes emulated by this one.

In this way, each interference node can send a frame whose size will be $BW \cdot BWshare \cdot Maxfs$ bytes. If this action were repeated $Nodes$ times, the consequent effect would be equivalent to send a frame of $Nodes{\cdot}BW{\cdot}BWshare{\cdot}Maxfs$ bytes. As BW is a random parameter, the frame size will vary randomly in each iteration, and so, time-varying delays will appear in the network environment.

This application enables to use different load conditions for each data path, since each interference node can define its own value for the parameters $Nodes$ and $BWshare$. Even, the parameter BW could vary randomly in a different range. In addition, once the network bandwidth is defined in the Truetime tool, the parameter $Maxfs$ can be automatically updated. Thus, this application offers a straightforward way to analyze this kind of networked control systems.

With the above settings, it is easy to obtain histograms of the time-varying delays which can be used to approximate probability distributions.

For instance, in later examples in this work, an Ethernet network working at 10Mbps is considered. With this bandwidth, the maximum frame size is $Maxfs = 1518$ bytes [36]. Two situations are studied: an unloaded network considering $BWshare = 0.5$, and a loaded network with $BWshare = 0.94$. In both cases, for simplicity, $Nodes = 175$ for both paths, and an estimated A/D-D/A conversion time of 0.01s is considered. In this way, for a simulation of 3.5 hours

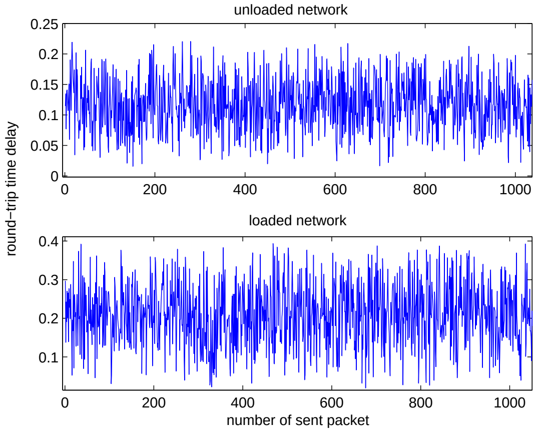

Fig. 3: Round-trip time delays $\tau(k)$

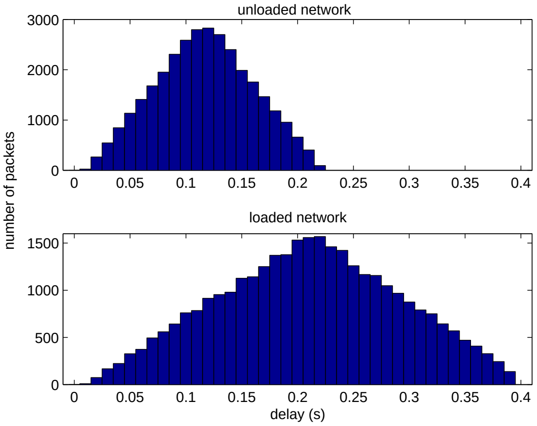

Fig. 4: Simulation delay histograms

(> 30000 transmitted packets), figure 3 shows the first round-trip time delays measured for each case. Figure 4 shows the histograms for the previous delay distributions. The simulation seems to suggest $NT = 0.4$ s, so that there is no loss of sensor samples. Although Gaussian distribution approximation for the Ethernet network delay is suggested in some references ([19], [33]) Truetime seems to simulate a triangular-shaped density probability function. Anyway, the later stability analysis may accommodate any probability distribution so the discussion on the reasons because of which the above distributions are obtained is not relevant.

From these histograms, a probabilistic LMI study (to be presented in section III-D) can be carried out. Performance analysis can be developed separately for each network load case, or in a multi-objective setting. The details on such procedures appear in the next sections.

III. Control considerations

A. Controller gain scheduling approach

In the proposed approach, a plant $\mathcal{P}$ will receive a set of control actions which depend on the network delay, $C(\tau)$; the controller will have adjustable parameters θ so the control action will be obtained from $u = C(\tau, \theta)e$, being $e = S(r - y)$. S denotes an event-driven sampling operation at regular instants. Note that, as the timing information τ is not known to the remote controller, the actual gain scheduling must be carried out at the local side, as discussed later in this section.

The aim of this approach is to introduce a controller retuning law $\theta(\tau)$ in order to maintain nominal (no-delay) control performance despite network delays.

In this work, the closed-loop poles will be chosen as a sensible criterion to define control performance. So, a so-called *performance vector* $\pi(\tau, \theta)$ will take the form

$$\pi(\tau, \theta) = Eig(Q(\tau, \theta)) \tag{2}$$

where θ are the design parameters (includes the PID controller parameters), τ is the network delay, Q represents the closed-loop operator $Sy = Q(Sr)$, and the operator Eig represents the eigenvalue computation of the resulting state-space matrix. In principle, a time-invariant nature for τ will be considered for design (otherwise, poles do not exist). Later on, time-variant stability analysis will be used to check the obtained results.

Thus, the nominal situation to be kept will be defined by $\pi(0, \theta_{nom})$, being θ_{nom} the nominal controller parameters designed for a delay-free case. If a delay $\tau(k)$ appears, the values for the performance vector will change with respect to $\pi(0, \theta_{nom})$. But retuning θ according to the delay, $\theta(\tau(k))$, one can try to make this difference as small as possible. The following linear scheduling law is proposed:

$$\theta(\tau(k)) = \theta_{nom} + M\tau(k) \tag{3}$$

where M is denoted as *scheduling vector*, which is deduced after solving a least-square problem on the minimization of the first-order Taylor term of $\|\pi(\tau, \theta) - \pi(0, \theta_{nom})\|$ (details are omitted for brevity, more information in [18]). The result takes the form

$$M = (\Delta^T W^T W \Delta)^{-1} W^T \Delta^T \delta_\tau \tag{4}$$

being W a weighting filter (to give priority to dominant poles), $\delta_\tau = \frac{\partial \pi}{\partial \tau}$ evaluated at $\tau = 0$, $\theta = \theta_{nom}$, and Δ is the vector of derivatives with respect to parameters θ evaluated at the same nominal point.

B. A dual-rate controller realisation

The chosen controller structure will be a dual-rate PID [18]:

$$\begin{aligned}
\psi(k+1) &= \begin{pmatrix} 1 & 0 \\ 0 & f(\tau(k)) \end{pmatrix} \psi(k) + \begin{pmatrix} K_i(\tau(k)) \\ 1 - f(\tau(k)) \end{pmatrix} e(k) \\
U_k(\tau(k)) &= \begin{pmatrix} u_1(\tau(k)) \\ u_2(\tau(k)) \\ \vdots \\ u_N(\tau(k)) \end{pmatrix} = \begin{pmatrix} 1 & -K_d(\tau(k)) \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \end{pmatrix} \psi(k) + \\
&\quad + \begin{pmatrix} K_p(\tau(k)) + K_d(\tau(k)) \\ K_p(\tau(k)) \\ \vdots \\ K_p(\tau(k)) \end{pmatrix} e(k)
\end{aligned} \tag{5}$$

where, as previously used, k are iterations at period NT, $\tau(k)$ is the current network delay and $e(k)$ is the error signal. The controller parameters are $K_p(\cdot)$, $K_i(\cdot)$, $K_d(\cdot)$, i.e., the proportional, integral and derivative gains, respectively, and $f(\cdot)$ is a derivative noise-filter pole.

The nominal dual-rate PID controller gains $\theta_{nom} = \theta(0)$ (that is, $K_p(0)$, $K_i(0)$, $K_d(0)$, $f(0)$) can be designed by means of classical procedures such as direct design [37], time response optimization [14], or discretization of continuous PID controllers [38]. Afterwards, such parameters are adjusted via the scheduling approach proposed above.

This realisation assumes that the proportional and integral actions are generated at the slow period, and the derivative one, which has relationship with anticipation and high-frequency behavior, is concentrated in the first sample. In [18], some implementation alternatives (as the noise filter operating at a fast rate) are discussed.

C. Control signal interpolation

As commented in section II, measurements are carried out with period NT. From non-conventional sampling literature, NT will be also denoted as global period or metaperiod, being T the control period, and $N \in \mathbf{N}^+$. Then, a dual-rate controller generates N control actions in every metaperiod.

As event-based devices, when a packet reaches either the controller or the actuator, they start to run a sequence of activities, as discussed below:

Controller activities: When a packet reaches the controller (the measurement-received event), the controller must compute the control action sequence (at fast period) to be sent to the actuator. However, as such a sequence depends on the total round-trip delay (which is unknown when the sensor packet reaches the controller), the controller will compute a sequence of actions to be sent as a lookup table to be interpolated in the actuator:

- First, a set of candidate delays $\gamma_1 = 0, \dots \gamma_l = \tau_{max}$ are chosen as interpolation points and the PID parameters are computed for them according to the linear gain scheduling formula (3).
- Then, for each of the above interpolation points, the parameters are plugged into the realization (5) and a sequence of control actions is obtained.
- A matrix $\bar{U}_k$ with the N control actions for each interpolation point γ_i is formed and sent to the actuator over the network. For instance, for two interpolation points, $\bar{U}_k$ takes the form

$$\begin{aligned} \bar{U}_k &= \begin{pmatrix} U_k(0) \\ U_k(\tau_{max}) \end{pmatrix} = \\ &= \begin{pmatrix} u_1(0) & u_2(0) & \dots & u_N(0) \\ u_1(\tau_{max}) & u_2(\tau_{max}) & \dots & u_N(\tau_{max}) \end{pmatrix} \end{aligned} \tag{6}$$

Figure 5 summarizes these activities by means of a flow chart.

Actuator activities: When $\bar{U}_k$ reaches the actuator, the following steps are triggered:

- First, the reception time $t_A(k)$ is taken. Then, as $t_S(k)$ was previously taken, (1) can be computed, and $\tau(k)$ obtained.
- Secondly, from the received subset of control actions $\bar{U}_k$, the array of control actions to be finally applied to the process
$$\hat{U}_k(\tau(k)) = [\hat{u}_1(\tau(k)), \hat{u}_2(\tau(k)), ..., \hat{u}_N(\tau(k))]$$
will be generated by means of a linear interpolating function
$$\hat{u}_j(\tau(k)) = u_j(0) - \tau(k)\frac{(u_j(0) - u_j(\tau_{max}))}{\tau_{max}}, j \in [1, N] \tag{7}$$
- Finally, $\hat{u}_1(\tau(k))$ will be applied at the time of arrival of the packet ($\tau(k)$ time units after the measurement was taken), and the other $\hat{u}_i(\tau(k))$, as they are not influenced by the network delay, will be applied at their respective trigger times $(kNT + (i-1)T)$. Note that if the delay were
$$\tau(k) \geq dT, d \in \mathbf{N}^+ \tag{8}$$
the first d control actions would not be applied.

Figure 6 depicts the activities for the general case by means of a flow chart.

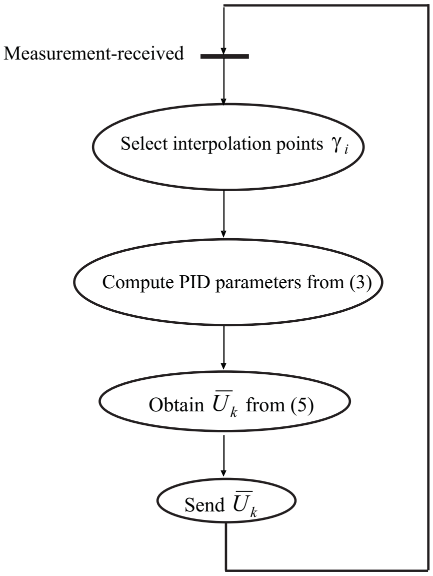
Measurement-received
Select interpolation points γ_i
Compute PID parameters from (3)
Obtain $\bar{U}_k$ from (5)
Send $\bar{U}_k$

Fig. 5: Flow chart for the controller activities

D. Stability analysis

It is well-known that the feedback connection of a generic controller and a process in a lifted framework [39] can be expressed as (see details, for example, in [13], [18])

$$\bar{x}(k+1) = \bar{A_{cl}}\bar{x}(k) \tag{9}$$

where the augmented state $\bar{x}$ contains both the controller states and the plant state variables.

Due to network characteristics, network delay $\tau(k)$ can vary from metaperiod to metaperiod. Then, $\bar{A_{cl}}$ in (9) must be replaced by $\bar{A_{cl}}(\tau(k))$ to represent a discrete LTV system

$$\bar{x}(k+1) = \bar{A_{cl}}(\tau(k))\bar{x}(k) \tag{10}$$

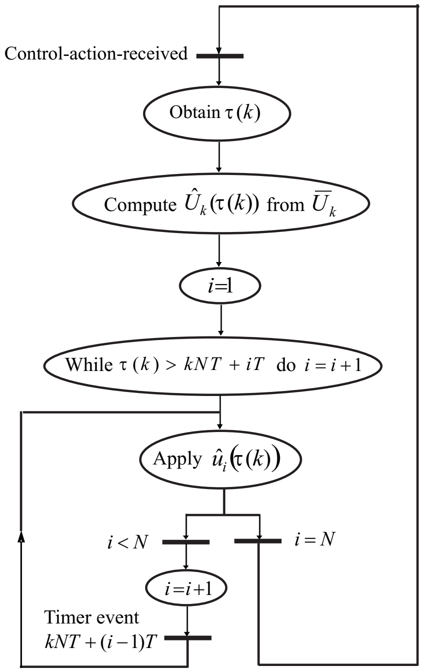
Control-action-received
Obtain τ(k)
Compute $\hat{U}_k(\tau(k))$ from $\bar{U}_k$
i=1
While τ(k) > kNT + iT do i = i+1
Apply $\hat{u}_i(\tau(k))$
i < N
i = N
i=i+1
Timer event
kNT + (i − 1)T

Fig. 6: Flow chart for the actuator activities

In the present work, a probabilistic model of the network delay τ will be assumed so a probability density function $p(\tau)$ is assumed known. Indeed, experimental or simulated data from Ethernet-based control have been fitted in literature to, for instance, exponential or Gaussian distributions [19], [33]. Also, as previously discussed, the Truetime software package allows obtaining simulation-based histograms for τ.

Assuming, then, the random variation of the network delay, stability of the closed-loop system will be analyzed in the mean square sense [22] (a particular case of the Martingales convergence theorem [40]), by means of a quadratic Lyapunov function

$$V(\bar{x}) = \bar{x}^T \mathrm{Q} \bar{x} \quad \mathrm{Q} > 0 \tag{11}$$

which will be shown to decrease in average, so $E[V(x)]$ will tend to zero hence the state will converge to zero with probability one ($E[\cdot]$ denotes the statistical expectation). The average descent to prove will be expressed as:

$$E[V(x(k+1))] \leq E[V(x(k))] \tag{12}$$

or, considering an average decay rate $0 < \alpha < 1$, the descent expression yields:

$$E[V(x(k+1))] \leq \alpha^2 E[V(x(k))] \tag{13}$$

Replacing the closed-loop equations in (11), the Lyapunov decrescence condition (13) can be written as the following probabilistic linear matrix inequality:

$$\int p(\vartheta)(\bar{A_{cl}}(\vartheta)^T \mathrm{Q} \bar{A_{cl}}(\vartheta) - \alpha^2 \mathrm{Q}) d\vartheta < 0 \tag{14}$$

where ϑ is a dummy parameter ranging in a set Θ where the time-varying parameter $\tau(k)$ is assumed to take values in, and matrix Q is composed of decision variables to be found by a semi-definite programming solver.

For a generic probability distribution, working with the above integral may be cumbersome. For bounded Θ a dense enough gridding in ϑ must be set up in order to approximately check for the above conditions. This procedure extends the LMI gridding in [41], [15] to a probabilistic case. In this work, Θ will be an interval $[0, \tau_{\max}]$. Choosing a set of l equally-spaced values ϑ_j, $j = 1, \ldots, l$ so that $\vartheta_1 = 0$, $\vartheta_l = \tau_{\max}$, (14) can be approximately rewritten as

$$\sum_{j=1}^{l} p(\vartheta_j) \bar{A_{cl}}(\vartheta_j)^T \mathrm{Q} \bar{A_{cl}}(\vartheta_j) - \alpha^2 \mathrm{Q} < 0 \tag{15}$$

which is a standard LMI to be solved by widely-known methods [21], [42].

Note that the above results are more relaxed than those in the previous related work by the authors [18]. Indeed, in a probabilistic case there is only one LMI constraint (average decay) instead of one for each possible sampling period in [18]. In this way, temporal random increases of the Lyapunov function are tolerated as long as the average over time is decreasing. Hence, better results in stability analysis can be obtained; however, the gridding approach leaves intermediate points out of the analysis so, in rigor, the results are not valid unless the grid is very fine. Some comments and alternative approaches are later discussed.

Approximation of probability distribution: In section II, histograms for different load conditions on an Ethernet framework have been simulated (see figure 4). From these histograms, an approximated probability distribution $p(\vartheta_j)$ for each network situation can be obtained, taking into account that the larger the simulation time is, the more representative the histograms and hence the probability distribution will be [43], [44].

Multi-objective analysis: The above idea can be extended to considering several possible network states, say n_o, with different performance objectives. Each network state will be described by a probability density of the network delay $p_i(\tau)$, and a performance objective α_i, $i = 1, \ldots, n_o$. For example, two objectives can be considered: the first one can be defined by the probability function for an unloaded network (with a probability distribution around a "short" delay mean), and the other objective for a saturated network case (with a larger delay mean).

Then, the Lyapunov decrescence conditions can be written as the following probabilistic linear matrix inequalities:

$$\int p_i(\vartheta)(\bar{A_{cl}}(\vartheta)^T \mathrm{Q} \bar{A_{cl}}(\vartheta) - \alpha_i^2 \mathrm{Q}) d\vartheta < 0 \quad \forall i \tag{16}$$

or, for computation, being replaced by its discrete approximation (15).

For simplicity, the decay-rate conditions (15) will be later used. Other well-known LMI conditions could be set up (see [21], [45] for details) for pole-region of norm-bound performance measures.

It is well known that the optimal result of multi-objective analysis will be a Pareto front with the optimal performance α_i' for a particular i', being the rest fixed. If the performance bounds on the rest of constraints are made more restrictive, the resulting $\alpha_{i'}$ decreases. The reader is referred to [46] for basic ideas on multi-objective optimization.

Note that the use of the shared Lyapunov function in (16) proves stability of the networked control system for any probabilistic mixture of the considered network states (i.e., a network state whose probability density can be expressed as a convex combination of those in LMIs (16)).

Limiting worst-case performance: In a realistic situation, actual probability distributions for delays might substantially differ from those in the above stability analysis, at least in some situations (faults, unpredicted overloads, etc.). Hence, it is advised to include a "backup" performance analysis considering the case of arbitrary delay changes with unknown probability. A way of solving the issue would be using an LMI gridding approach [15], [18] requiring the LMIs below to hold for some degraded performance measure α_0, and for a representative enough grid of delay values $\vartheta_i \in \Theta$:

$$\bar{A_{cl}}(\vartheta_i)^T \mathrm{Q}' \bar{A_{cl}}(\vartheta_i) - \alpha_0^2 \mathrm{Q}' < 0 \tag{17}$$

where, for instance $\alpha_0 = 1$ would amount to requiring stability as the minimum performance requirement on any unexpected situations. Note that a different Lyapunov function Q' has been used, as this worst-case analysis does not need to consider the switching between it and the previously considered situations.

Alternative approaches for a more precise analysis: other alternative techniques for stability analysis may be used in the proposed setup, both in the robust case (arbitrary probability distribution) [18] or in the stochastic framework. As this paper focuses on the dual-rate PID controller structures, the reader is referred to works such as [15], [26], [23], [28] for contributions regarding state-feedback designs, etc., in some cases for delays larger than one sampling period (via augmented realizations, see [27]). Also, networked control systems may be considered a particular case of stochastic hybrid systems [31].

Regarding [18], it is clear that gridding gives somehow optimistic performance bounds by neglecting intermediate values to those in the grid. Several alternatives to overcome such limitation can be considered in the framework of uncertain systems. In [24], a grid generation algorithm is provided to remove the above issue; in [30], a Lipschitz-like condition $\|\bar{A}_c l(\vartheta_i) - \bar{A}_c l(\vartheta)\| \leq L(\vartheta_i - \vartheta)$ is used; some convex bounds of the relevant set of matrices are suggested in [25] (via Taylor expansions) and [26] (based on the exponential of the real Jordan form). In [29], multiple loops sharing a network are considered; depending on which node has the network access, a different model is obtained; the networked control system is considered to be an uncertain discrete switched system for LMI analysis.

Regarding the stochastic-stability approach, many sampled-data networked-control systems may be regarded as a particular case of "impulsive renewal systems" [28] ($\dot{x(t)} = \bar{A}x(t)$, $x(t_k) = \bar{J}x(t_k^-)$) where the sampling periods $h_k = t_{k+1} - t_k$ are independent identically-distributed random variables. The continuous state includes the zero-order hold state (i.e., $\dot{u} = 0$). LMI characterizations of mean-square stability are given [28, Theorem 2] which reduce to (14) in some particular cases. Some extensions of the LQR control to the renewal-system case are also presented.

Finally, the paper [32] combines both the polytopic overapproximation idea and the discretization of probability density functions, by carrying out a polytopic approximation in each of the elements of a triangulation of the sampling rate/ delay space and obtaining the discrete probability of each triangle. In this way, probabilistic information is not lost and the optimistic gridding is avoided, solving open issues of the presented methodology.

IV. Application example: control of a crane platform

In this section, a test-bed Ethernet environment is used to implement the proposed NCS (see figure 7), which includes the following devices:

- an industrial crane platform (to be controlled) equipped with three cc motors (to actuate each axis: X, Y, Z) and five encoders (to sense the three axis and two different angles). The motors are controlled by an analog signal in the range ±1V. The encoders provide a position measurement of 1V/m. In this application only the X axis is actuated and sensed. Details on the crane characteristics can be obtained at http://www.inteco.com.pl (3D crane apparatus).
- a local computer which is connected to the platform by means of a DAQ board,
- two PLCs and one computer working as interference nodes in order to introduce different load scenarios,
- a switch shared by the previous devices to connect them to Ethernet,
- a remote laptop computer where the event-based controller is implemented.

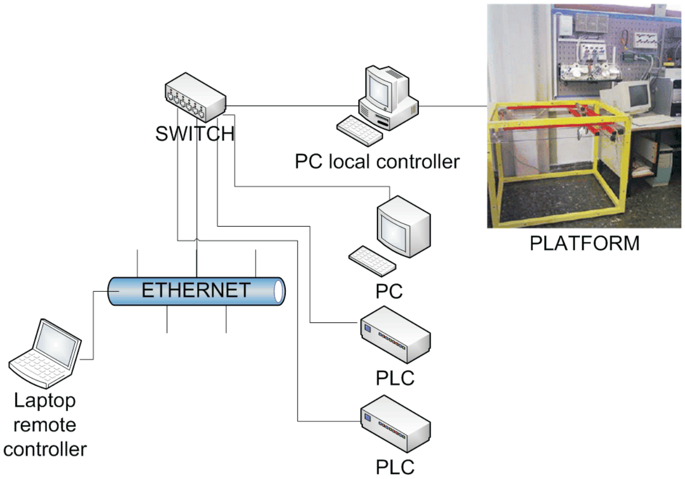

Fig. 7: Test-bed Ethernet environment

The network configuration will be the one in section II, whose probability histograms and configuration parameters have been already described.

The objective of the methodology of this paper is being able to design a working network-based controller from Truetime simulation and LMI tools. Hence, the section will be divided into a first *a priori* control design phase with LMI and simulation studies and, later, an experimental verification of the applicability of the obtained results.

A. Model-based design and simulation study

In this section, the following DC motor plant of the X axis of an industrial crane will be used as the controlled plant at the local side:

$$\mathcal{P}(s) = \frac{6.3}{s(s+17.7)}$$

The main aim of this example is to show the possibilities of the event-based dual-rate PID controller when it is retuned according to Ethernet network delays.

First of all, the dual-rate PID controller choice is justified. Then, the probabilistic LMI analysis is performed, comparing the nominal controller with the scheduled one. Finally, time response for both controllers is simulated by means of the Truetime application.

Controller choice: A comparison, for the no-delay case, between single-rate regulators and a dual-rate PD regulator is carried out.

The output sampling time for the dual-rate case will be $NT = 0.4$s, since, as shown in figure 4, the largest delay obtained in the simulations is 0.39s (delay bound: $\tau_{max} = 0.39$s), and then the requirement commented after (1), that is $\tau_{max} < NT$, is fulfilled.

Since the motor plant has an integrator, a suitable control can be achieved with a PD controller.

In figure 8, three different design options are compared:

- a single-rate ($T = 0.4$) cancellation controller with the fastest possible poles at $z = 0$
- a conventional PD controller, at the same rate, and
- a dual-rate PD controller ($T = 0.2$, $NT = 0.4$, $N = 2$).

Results show that the inter-sample behavior of the cancellation controller has almost 11% overshoot and settling time of 0.75 seconds.

From an initial root-locus design, the single-rate PD is fine-tuned by hand to achieve a response as fast as possible. It is the bilinear discretisation of:

$$K_p(1 + K_d\frac{s}{fs+1})$$

for the values: $K_p = 9.5, K_d = 0.019, f = 0.1$. Although the settling time achieved is similar to that of the cancellation controller (0.85 seconds), the overshoot is drastically worse (35%).

The dual-rate PD corresponds to the realization (5), with values:

$$K_p = 6.95, K_i = 0, K_d = 2.2, f = 0.1$$

It achieves the least overshoot (8%) and the least settling time (0.65 seconds).

As the dual-rate setup obtains a better response in the proposed scenario, the dual-rate PD has been chosen as the nominal controller.

From the nominal parameters, the gain scheduling approach achieves a delay-dependent law. To do it, firstly, the parameters δ_τ and Δ in (4) must be obtained (steps omitted for brevity, see [18] for details). Then, considering W as the identity matrix, the least-squares solution produced the following scheduling law as a function of the measured network delay $\tau(k)$:

$$\left(\begin{array}{ccc} K_p & K_d & f \end{array} \right)^T = \left(\begin{array}{ccc} 11.4123 & -43.8824 & 1.5514 \end{array} \right)^T \tau(k)$$

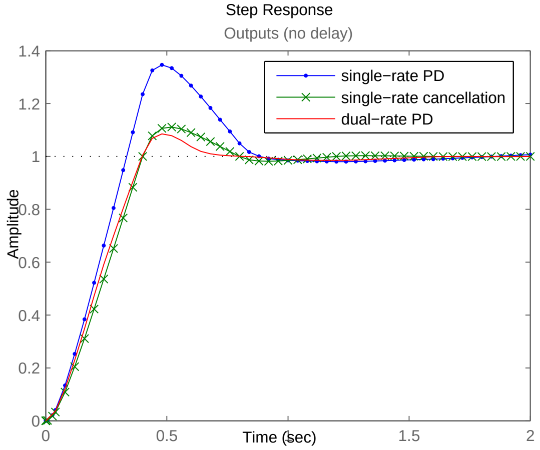

Fig. 8: Single vs. dual-rate comparison (continuous inter-sample response)

Mean-square stability analysis: The closed-loop poles as a function of $\tau(k)$ are an optimistic estimate of the actual performance for varying $\tau(k)$. Then, in order to assess the stability of the setup in probabilistic time-varying delays, an LMI gridding has been carried out computing the closed-loop realization for the delay bound τ_{max}=0.39s. Thus, the parameter space is Θ=[0,0.39], and l=40 grid points for the probability density approximation (15) were taken. In order to have a reasonable estimation of the probability distributions, the network has been simulated for a time interval H=3.5 hours (as shown in section II), such that the probability density estimations for H and those for $H/2$ differ less than 0.0039 for all cells.

Two cases are analyzed. In the first one, considering each network situation separately (a different Lyapunov function for each scenario), the LMI in (15) is applied to each situation to obtain the minimum α for which a feasible solution Q exists. In the second case, a multi-objective analysis is performed by considering a unique Lyapunov function for both network load scenarios. The second case is more conservative but allows for stability guarantees for mixtures and random switching between both scenarios. The two proposed cases are somehow extreme situations from which would happen in a practical situation. If each of the network behaviours is very likely to remain active for a dwell-time significantly longer than the loop's settling time, then assumptions in case 1 will be closer to reality. If arbitrary, fast, network load changes were expected, then case 1 would be too optimistic and the analysis in case 2 would be recommended.

Regarding the first analysis, results are presented in Table I, both for the scheduled PD and for the nominal one.

TABLE I: LMI decay-rate for the dual-rate PD controller (with probability information)

network context	scheduled PD	nominal PD
only unloaded	0.50	0.65
only loaded	0.68	0.83

In conclusion, the less the network is loaded, the better worst-case performance can be guaranteed. In addition, the scheduled approach shows better behavior than the nominal one.

Now, the second (multi-objective) study is carried out. Figure 9 shows a Pareto front that summarizes the analysis, which is developed by setting the decay-rate of one objective and optimizing the other's one. As depicted, the decay-rates obtained in the previous study for the unloaded network case (here, the first objective) can not be now achieved, despite considering the highest decay-rate for the second objective ($\alpha_2 = 0.99$), that is, the loaded network scenario. However, if $\alpha_1 = 0.99$, the previous decay-rates for the loaded network case can be achieved. Finally, the figure reveals the scheduled approach outperforms again the nominal one.

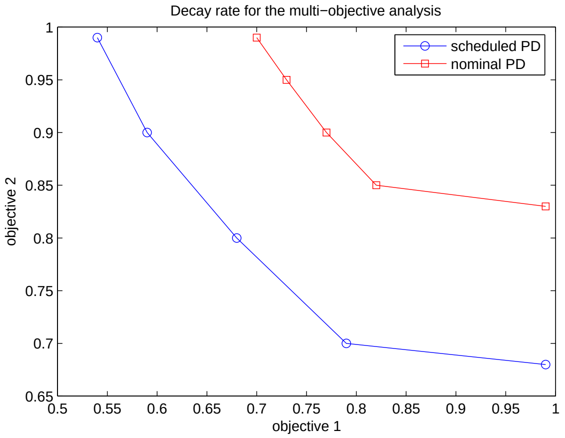

Fig. 9: LMI decay-rate for the multi-objective study

Worst-case stability: In order to assess the worst-case behaviour of the proposed PID regulator, the LMIs in (17) have been tested for different maximum delay bounds. The results appear on Table II.

TABLE II: LMI decay-rate for the dual-rate PD controller (worst-case, unknown probability distribution)

Max. Delay bound	scheduled PD	nominal PD
0.1	0.42	0.59
0.15	0.63	0.73
0.20	0.84	0.84
0.30	1	0.99

As a conclusion, the proposed gain scheduled regulator improves worst-case performance for small delays (up to 0.2), as expected from the Taylor-series argumentation used in its design. In large delays, the linearised approximation of the performance loses precision and results are similar (marginally worse, in fact) than those of a non-scheduled regulator. In fact, there are delay distributions involving delays larger than 0.3 which might render the system unstable.

In summary, from the analysis of both tables we may conclude that the use of probabilistic information indicates that the Taylor-series design seems a sensible practical procedure if the probability of large delays is low, as intuitively expected, because average performance is improved. If, however, no likelihood of (transient) unstability is required, then the network must be reconfigured so the maximum delay does not exceed 0.3 seconds or the initial controller specifications must be changed (reducing gains to improve robustness).

Truetime simulation: Since the previous figures indicate only stability and decay-rate, it is interesting to simulate the control system time response in order to observe other performance differences such as overshoot. In order to better compare the time response for the scheduled and the nominal (no delay, no scheduling) strategies, the TrueTime simulation program described in section II is executed with the two different network configurations, which are used in the obtention of the probability histograms.

As previously discussed, the proposed network configuration is the same as in section II. Of course, each different simulation run involves a different realisation of the random delay which might not have the same empirical histogram. This might lead to some variations between the predicted performance and the actually achieved one in a short simulation run.

In figure 10 the unloaded network case is shown. Considering a step generator as a reference, the scheduled PD achieves better control performance than the nominal one. Depending on the network delay which is measured when the reference changes, the performance indexes vary. Anyway, the scheduled regulator reduces the overshoot at least a 10% and up to a 40%, and the settling time is reduced up to a 60%, with a 30% on the average.

Figure 11 shows how the nominal regulator worsens drastically its previous control system performance, due to the existence of larger delays in a loaded network. Again, the scheduled PD achieves a better behavior.

Although the scheduled proposal does not exactly follow the no-delay response (the Taylor-series scheduling only keeps performance for small delays near the nominal conditions), simulation results show that this proposal always seems to outperform the unscheduled nominal design (under the proposed probability distributions of the network delay), confirming the previous LMI analysis theoretical results.

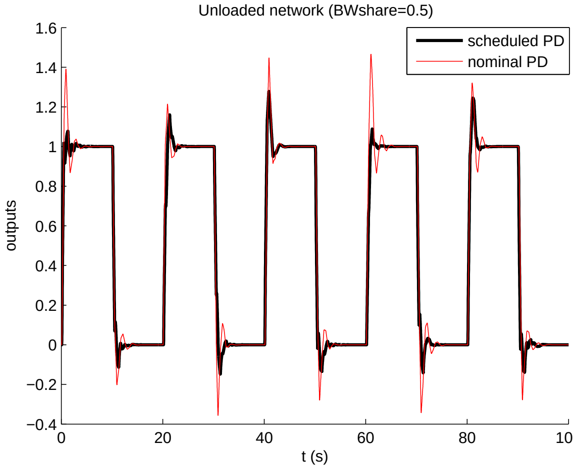

Fig. 10: Closed loop output (unloaded network, simulation)

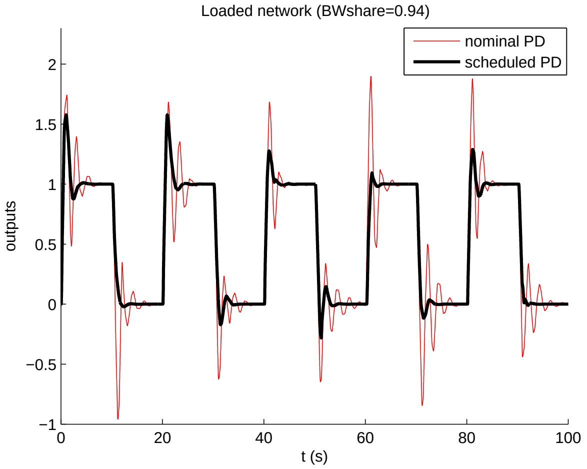

Fig. 11: Closed loop output (loaded network, simulation)

B. Experimental results

Once the simulation studies have been deemed satisfactory, the controller parameters obtained in the previous section have been applied directly to control the real experimental system.

Several experiments are carried out, where the number and complexity of the tasks developed by the interference nodes is modified in order to obtain different load scenarios ranging between the two extreme histograms on Figure 12.

All the experiments present the same trend observed in the Truetime simulation, that is, the scheduled PD controller point out a better behavior than the nominal PD controller. For brevity, only the results of one of the experiments are shown. Figure 12 presents the histograms for each network load case, and a similar shape to those shown in figure 4 can be stated. Note, since these practical experiences are based on a short run, few packets are transmitted and hence histograms present lower resolution than those obtained in simulation (figure 4). Finally, the position output for the X axis depicted in figures 13 and 14 can be respectively compared to those in figures 10 and 11, validating the control proposal. Note that at some transients in figure 13, both approaches present a similar performance. This is due to existing, casually, short network delays at those instants (for example, around t=15s). In other transients (around t=0s, t=10s and t=40s), random delays are bigger and, hence, a larger difference between both control strategies appears.

From the experimental data, the integral of the absolute error (IAE) performance criterion has also been computed, its results appearing on Table III. The non-scheduled PID worsens IAE around 10% in the unloaded-network case; in the loaded case (bigger likelihood of larger delays) the nominal PID has 70% worse IAE.

TABLE III: experimental IAE performance criterion for the dual-rate PD controller

network context	scheduled PD	nominal PD
unloaded	116.0	127.3
loaded	112.8	189.5

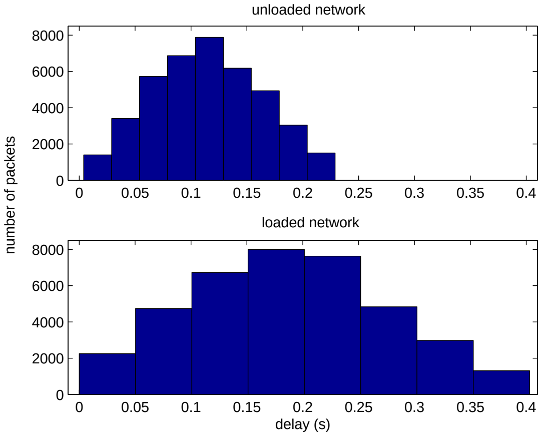

Fig. 12: Experimental delay histograms

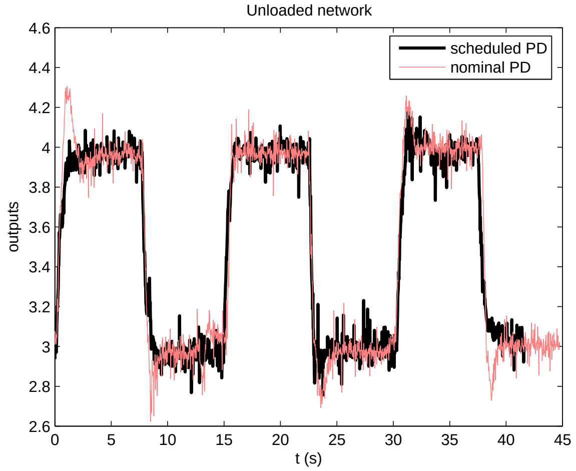

Fig. 13: Closed loop output (unloaded network, experimental)

V. Conclusions

In this paper, a delay-dependent dual-rate gain-scheduled controller structure is proposed. Its stability analysis exploits probabilistic time-varying delay information. Such information can be obtained from a network with either direct monitoring or suitable simulations (for instance, with the Truetime package in this case).

The network setup considers two different sides: the remote one (controller) and the local one (plant, sensor and actuator). The network is assumed to be shared with other nodes generating a randomly-varying load on it.

An event-based dual-rate PID controller is chosen in order to reduce network traffic, to improve control performance, and to avoid clock synchronization problems. Its control actions are generated using an interpolating function based on a delay-dependent gain scheduling approach. So, by actuating in a time-varying scheduled way, control performance degrades less than when using the nominal no-delay regulator parameters.

An LMI analysis enables to study the closed-loop system's mean-square stability. LMI results point out a better behavior for the scheduled controller than for the nominal one.

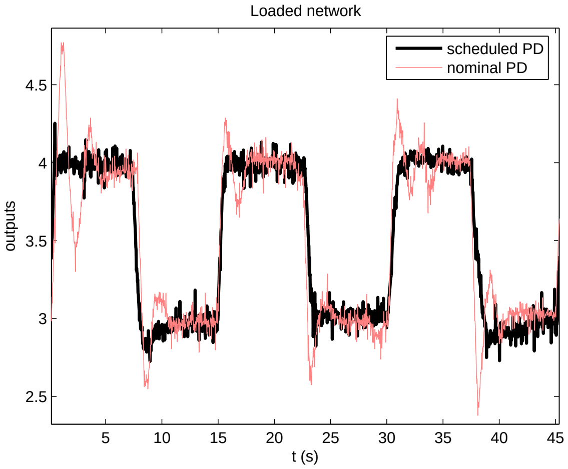

Fig. 14: Closed loop output (loaded network, experimental)

The experimental validation on an Ethernet test-bed shows a delay histogram which is reasonably close to the TrueTime simulations and also a closed-loop control response which agrees with the calculations and simulations in the design phase. Hence, the experimental results seem to validate the methodology.

References

[1] Y. Tipsuwan and M. Chow, "Control methodologies in networked control systems," *Control Eng. Practice*, vol. 11, no. 10, pp. 1099–1111, 2003.

[2] Y. Halevi and A. Ray, "Integrated communication and control systems. I- Analysis," *ASME, Trans., J. Dynamic Syst., Meas. and Control*, vol. 110, pp. 367–373, Dec. 1988.

[3] T. Yang, "Networked control system: a brief survey," *IEE Proc. Control Theory Appl.*, vol. 153, no. 4, pp. 403–412, Jul. 2006.

[4] M. Ohlin, D. Henriksson, and A. Cervin, *TrueTime 1.5 Reference Manual*.

[5] S. Johannessen, "Time synchronization in a local area network," *IEEE Control Syst. Mag.*, vol. 24, no. 2, pp. 61–69, Apr. 2004.

[6] N. Vatansky and al. al. al. al., "Networked control with delay measurement and estimation," *Control Eng. Practice*, 2008. [Online]. Available: doi:10.1016/j.conengprac.2008.07.004

[7] K. Astrom, "Event based control," *Analysis and Design of Nonlinear Control Systems*, 2007.

[8] D. Li, S. Shah, and T. Chen, "Analysis of dual-rate inferential control systems," *Automatica*, vol. 38, no. 6, pp. 1053–1059, Jun. 2002.

[9] Á. Cuenca, J. Salt, and P. Albertos, "Implementation of algebraic controllers for non-conventional sampled-data systems," *Real-Time Syst.*, vol. 35, no. 1, pp. 59–89, Jan. 2007.

[10] S. Lall and G. Dullerud, "An LMI solution to the robust synthesis problem for multi-rate sampled-data systems," *Automatica*, vol. 37, no. 12, pp. 1909–1922, Dec. 2001.

[11] J. Salt and P. Albertos, "Model-based multirate controllers design," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 6, pp. 988–997, Nov. 2005.

[12] M. Araki, "Recent developments in digital control theory," in *Proc. IFAC World Congr.*, vol. 9, 1993, pp. 951–960.

[13] P. Albertos and A. Sala, *Multivariable Control Systems: An Engineering Approach*. Springer, 2004.

[14] K. Astrom and T. Hagglund, "PID Controllers: Theory, Design, and Tuning," *Instrum. Soc. of America*, pp. 80–81, 1995.

[15] A. Sala, "Computer control under time-varying sampling period: An LMI gridding approach," *Automatica*, vol. 41, no. 12, pp. 2077–2082, Dec. 2005.

[16] Á. Cuenca, P. García, K. Årzén, and P. Albertos, "A Predictor-Observer for a Networked Control System with time-varying delays and non-uniform sampling," in *Proc. Eur. Control Conf.*, 2009.

[17] D. Georgiev and D. Tilbury, "Packet-based control: The H2-optimal solution," *Automatica*, vol. 42, no. 1, pp. 137–144, 2006.

[18] A. Sala, Á. Cuenca, and J. Salt, "A retunable PID multi-rate controller for a networked control system," *Inform. Sci.*, vol. 179, no. 14, pp. 2390–2402, Jun. 2009.

[19] Y. Tipsuwan and M. Chow, "Gain scheduler middleware: a methodology to enable existing controllers for networked control and teleoperation-part I: networked control," *IEEE Trans. Ind. Electron.*, vol. 51, no. 6, pp. 1218–1227, Dec. 2004.

[20] H. Li, Z. Sun, B. Chen, H. Liu, and F. Sun, "Intelligent Scheduling Control of Networked Control Systems with Networked-induced Delay and Packet Dropout," *Int. J. Control, Autom., and Syst.*, vol. 6, no. 6, pp. 915–927, Dec. 2008.

[21] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Soc. Ind. Math., 1994.

[22] L. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1562–1572, Sep. 2004.

[23] E. Fridman and U. Shaked, "Stability and guaranteed cost control of uncertain discrete delay systems," *Int. J. Control*, vol. 78, no. 4, pp. 235–246, 2005.

[24] H. Fujioka, "A discrete-time approach to stability analysis of systems with aperiodic sample-and-hold devices," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2440–2445, 2009.

[25] L. Hetel, J. Daafouz, and C. Iung, "Stabilization of arbitrary switched linear systems with unknown time-varying delays," *IEEE Trans. Autom. Control*, vol. 51, no. 10, pp. 1668–1674, 2006.

[26] M. Cloosterman, N. van de Wouw, W. Heemels, and H. Nijmeijer, "Stability of networked control systems with uncertain time-varying delays," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1575–1580, 2009.

[27] M. Cloosterman, L. Hetel, N. van de Wouw, W. Heemels, J. Daafouz, and H. Nijmeijer, "Controller synthesis for networked control systems," *Automatica*, doi:10.1016/j.automatica.2010.06.017, 2010.

[28] D. Antunes, J. Hespanha, and C. Silvestre, "Control of Impulsive Renewal Systems: Application to Direct Design in Networked Control," in *Proc. Joint 48th IEEE Conf. on Decision and Control (CDC) and 28th Chinese Control Conf.*, 2009.

[29] M. Donkers, L. Hetel, W. Heemels, N. van de Wouw, and M. Steinbuch, "Stability analysis of networked control systems using a switched linear systems approach," in *Hybrid Systems: Computation and Control (LNCS)*. Springer, 2009, pp. 150–164.

[30] J. Skaf and S. Boyd, "Analysis and synthesis of state-feedback controllers with timing jitter," *IEEE Trans. Autom. Control*, vol. 54, no. 3, pp. 652–657, 2009.

[31] D. Antunes, J. Hespanha, and C. Silvestre, "Stochastic hybrid systems with renewal transitions," in *Proc. American Control Conf. (ACC)*, 2010, pp. 3124–3129.

[32] M. Donkers, W. Heemels, D. Bernardini, A. Bemporad, and V. Schneer, "Stability analysis of stochastic networked control systems," in *Proc. American Control Conf. (ACC)*, 2010, pp. 3684–3689.

[33] J. Park and J. Lee, "Transmission modeling and simulation for Internet-based control," in *Proc. Conf. IEEE Ind. Electron. Soc.*, vol. 1, 2001.

[34] K. Lee, S. Lee, and M. Lee, "Remote fuzzy logic control of networked control system via Profibus-DP," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 784–792, Aug. 2003.

[35] ——, "QoS-based remote control of networked control systems via Profibus token passing protocol," *IEEE Trans. Ind. Informat.*, vol. 1, no. 3, pp. 183–191, Aug. 2005.

[36] A. Tanenbaum, *Computer networks*. Prentice Hall, 2003.

[37] K. Ogata, *Discrete-time control systems*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1987.

[38] J. Salt, A. Cuenca, V. Casanova, and V. Mascaros, "A PID Dual Rate Controller Implementation over a Networked Control System," in *Proc. IEEE Int. Conf. Control Appl.*, 2006, pp. 1343–1349.

[39] P. Khargonekar, K. Poolla, and A. Tannenbaum, "Robust control of linear time-invariant plants using periodic compensation," *IEEE Trans. Autom. Control*, vol. 30, no. 11, pp. 1088–1096, Nov. 1985.

[40] M. Rao, *Foundations of stochastic Analysis*. Academic Press, 1981.

[41] P. Apkarian and R. Adams, "Advanced gain-scheduling techniques for uncertain systems," *IEEE Trans. Control Syst. Technol.*, vol. 6, no. 1, pp. 21–32, 1998.

[42] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1, pp. 625–653, 1999.

[43] S. Dasgupta, "Learning probability distributions," Ph.D. dissertation, Univ. of California, 2000.

[44] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning theory*. Springer, 2001.

[45] R. Sanchez-Pena and M. Sznaier, *Robust Systems Theory and Application*. John Wiley & Sons, New York, 1998.

[46] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of multiobjective optimization*. Academic Press, 1985.