

Document downloaded from:

<http://hdl.handle.net/10251/65708>

This paper must be cited as:

Cuenca Lacruz, ÁM.; Ojha, U.; Salt Llobregat, JJ.; Chow, M. (2015). A non-uniform multi-rate control strategy for a Markov chain-driven Networked Control System. *Information Sciences*. 321:31-47. doi:10.1016/J.INS.2015.05.035.



The final publication is available at

<http://dx.doi.org/10.1016/j.ins.2015.05.035>

Copyright Elsevier

Additional Information

A non-uniform multi-rate control strategy for a Markov chain-driven Networked Control System

Ángel Cuenca, Unnati Ojha, Julián Salt, Mo-Yuen Chow¹

Abstract

In this work, a non-uniform multi-rate control strategy is applied to a kind of Networked Control System (NCS) where a wireless path tracking control for an Unmanned Ground Vehicle (UGV) is carried out. The main aims of the proposed strategy are to face time-varying network-induced delays and to avoid packet disorder. A Markov chain-driven NCS scenario will be considered, where different network load situations, and consequently, different probability density functions for the network delay are assumed. In order to assure mean-square stability for the considered NCS, a decay-rate based sufficient condition is enunciated in terms of probabilistic Linear Matrix Inequalities (LMIs). Simulation results show better control performance, and more accurate path tracking, for the scheduled (delay-dependent) controller than for the non-scheduled one (i.e. the nominal controller when delays appear). Finally, the control strategy is validated on an experimental test-bed.

Index terms: Networked control system, unmanned ground vehicle, network delay, packet disorder, multi-rate control systems, PID controller.

¹ Original Article Statement: a. This manuscript is the authors' original work and has not been published nor has it been submitted simultaneously elsewhere. b. All authors have checked the manuscript and have agreed to the submission. The corresponding author is Á. Cuenca. He is with Departamento de Ingeniería de Sistemas y Automática (DISA), Instituto Universitario de Automática e Informática Industrial (AI2), Universidad Politécnica de Valencia (UPV), and he was a Visiting Scholar at the Advanced Diagnosis, Automation and Control (ADAC) Laboratory, Electrical and Computer Engineering (ECE) Department, North Carolina State University (NCSSU) (e-mail: acuenca@isa.upv.es); U. Ojha and M-Y. Chow are with the ADAC Lab, ECE Dept., NCSSU (e-mails: uojha@ncsu.edu, chow@ncsu.edu); and J. Salt is with DISA, AI2, UPV (e-mail: julian@isa.upv.es).

1 Introduction

An NCS is a control system where different devices share a common communication link. Advantages such as cost reduction and easy maintenance have motivated a wide study about NCS in the last years (see, for instance, in [16], [48], [52]). Nevertheless, some drawbacks arise when using a shared link, being the fundamental one the existence of time-varying delays when transmitting information between devices (usually, from sensor to controller and from controller to actuator) [51], [47], [24], [7]. Also, in some cases, networks introduce packet dropouts [20], [46], [8], packet disorder [32], [8], [53], and bandwidth constraints [30], [31]. Dealing at the same time with all the drawbacks involved in an NCS becomes a complex problem, and hence, some simplifying assumptions are usually made.

The present work is focused on facing time-varying network-induced delays and avoiding packet disorder in a kind of NCS where the plant to be wirelessly controlled is a UGV. Most of papers treating this scenario (for instance, [30], [31], [43], [22], [44]) does not cope with the packet disorder phenomenon, and in addition, are characterized by:

- performing the main control tasks (that is, path tracking and delay compensation) at the remote side (with no direct communication to the plant). At the local side (close to the plant), a simple PID is located to control the instantaneous rotational velocity.
- compensating for round-trip time delays from estimated remote-to-local delays (since at the remote side only the current local-to-remote delay is known).
- requiring estimated UGV states so as to generate the proper path references to be followed when the packets reach the local side. At this moment, the references are compared to the current, instantaneous velocities, and the consequent error signal is controlled by the simple PID.

The main problem of the approach considered in these papers is working with estimated values, since the worse the estimation is, the higher performance degradation could be. In order to avoid this problem, a proposal which deals with actual values for the round-trip time delays and the UGV states is proposed in the present work. This approach considers a Multi-Rate Input Control (MRIC) strategy [18], [36], [6], [9] (i.e., the sensing rate is N times slower than the actuation rate), with the next features:

- the control tasks are separated into two control levels working at different rates: the path tracking control is located at the remote side generating slow-rate references from the actual UGV state, and the delay compensation is situated at the local side generating a fast-rate control signal from the actual

delay. Concretely, the path tracking controller implements a network-based adaptation of the Quadratic Curve (QC) algorithm by [49], and the delay compensation controller applies a version of the gain scheduling approach introduced in [35].

- the sensing period is chosen to be greater than the longest delay². In this way, the packet disorder problem is avoided. But, from a single-rate control framework, this period could then be too great to reach the desired control performance (the greater the sampling period is, the worse the control performance should be expected [28]). However, by adopting the MRIC strategy, since control actions are applied at N times faster rate than the sensing rate, the control performance can be maintained [3], [19].
- as a consequence of existing time-varying network delays, the fast-rate control signal will be non-uniformly applied, resulting in a non-uniform multi-rate control approach.
- actual round-trip time delays can be measured and compensated at the local side, since a local timer is shared by every local device. In addition, the remote controller needs no synchronization to develop its tasks, and hence no time-stamping techniques are required.

As a result of these features, our control proposal implies a straightforward implementation, being not required additional prediction stages for the delay and the UGV state (used in [30], [31], [43], [22], [44]), and the possibility of reducing the network utilization (since information travels through the network at the slower rate).

The present work contemplates a Markov chain-driven NCS [21], [10], [33], [38], [39], which is an special case of stochastic NCS [40], [25], [11]. Driving the NCS by a finite state Markov chain enables to consider not only states with different delay distributions (the stochastic case) but also transition probabilities among states. This kind of NCS model can be useful to be applied to most of Internet applications [26], [41]. In an Internet environment is usual to find different network load scenarios, where heavy traffic situations can appear (loaded network) and disappear (unloaded network) following some transition probabilities. This will be the case considered in the present work. However, Markov chains can also be used to model other aspects of the NCS. For example, to drive activation states of subsets of actuators [13], [14], or subsets of sensors and actuators [15], [23]. In any case, in order to assure stability in the jumping system, some conditions can be enunciated in terms of LMIs [5] under the consideration of transmission delays both in system output measurements and in control signals.

Summarizing, the paper includes the following sections. In section 2, the pro-

² From experimental or simulated data, network-induced delay distributions are assumed to be known.

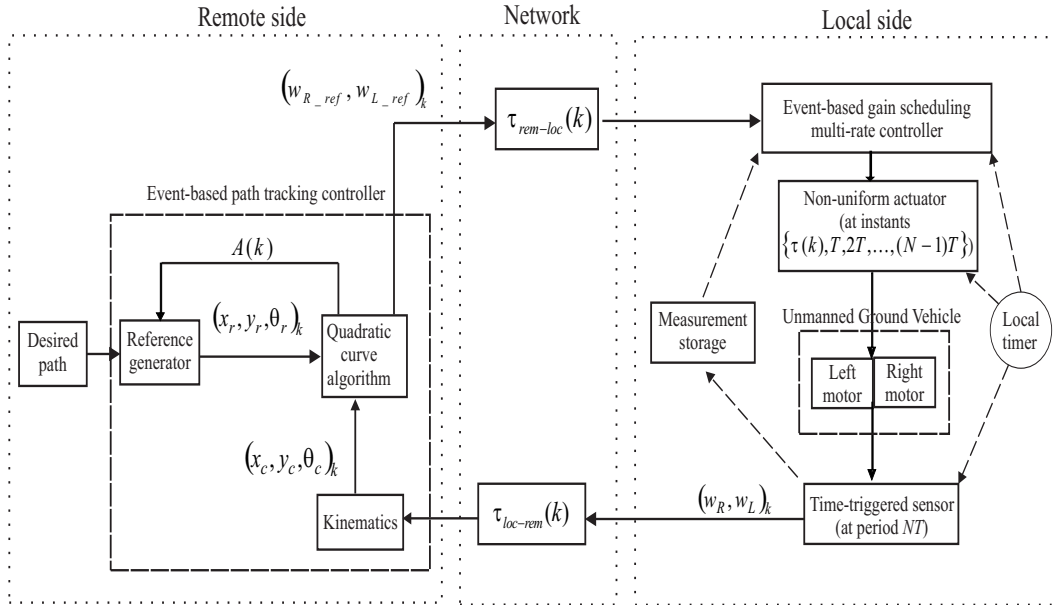


Fig. 1. NCS scenario

posed NCS is described. Section 3 presents the remote and local controllers. Then, in order to prove global mean-square asymptotical stability for the Markov chain-driven NCS, section 4 introduces a decay-rate based sufficient condition in terms of probabilistic LMIs. In addition, in this section, several cost functions are presented in order to be used when analyzing NCS performance. A simulation example in section 5 illustrates the main benefits of the control proposal, which is validated by means of an experimental test-bed in section 6. Finally, section 7 enumerates the main conclusions of the present work.

2 Description of the NCS scenario

The proposed NCS is described in Figure 1, where the next devices are considered:

- a UGV as a process to be controlled, which includes two motors (for right and left wheels).
- a time-triggered sensor, working at period NT , to sample the process output (rotational velocity) for each motor $(w_R, w_L)_k$ (right and left motors, resp.³). This output is sent through the network, and saved (in the measurement storage block) to be used later by the local (dynamic) controller (described below).

³ Note, $k \in \mathbf{N}$ indicates the number of iteration at NT (being $N > 1$).

- an event-based path tracking controller (at the remote side) that implements the Quadratic Curve (QC) path tracking algorithm in order to generate slow-rate rotational velocity references for each motor $(w_{R.ref}, w_{L.ref})_k$ from the current UGV state $(x_c, y_c, \theta_c)_k$ and the reference UGV state $(x_r, y_r, \theta_r)_k$. The controller's tasks are triggered after receiving the process output $(w_R, w_L)_k$ from the sensor device, that is, after elapsing a local-to-remote delay $\tau_{loc-rem}(k)$. More details about the path tracking controller can be found in section 3.1.
- an event-based gain scheduling multi-rate controller (at the local side) that generates fast-rate control actions (to be applied to the UGV) from the slow-rate references $(w_{R.ref}, w_{L.ref})_k$. These references arrive to the local side after elapsing a remote-to-local delay $\tau_{rem-loc}(k)$. Then, the local (dynamic) controller algorithm is triggered in order to, firstly, calculate the round-trip time delay $\tau(k)$, and secondly, compensate for it following a gain scheduling approach (more details in section 3.2). The round-trip time delay $\tau(k)$ is defined as

$$\tau(k) = \tau_{rem-loc}(k) + \tau_C(k) + \tau_{loc-rem}(k) \quad (1)$$

where $\tau_C(k)$, a (negligible) computation time delay required by the different devices, is lumped together with the network-induced delays. $\tau(k)$ is assumed to be time-varying (and driven by a Markov chain, as later detailed) inside the range $[0, \tau_{max}]$, being $\tau_{max} < NT$ (hence, no packet disorder is assured).

In this case, the local (dynamic) controller is a multi-rate PID one, and its gains will be retuned according to $\tau(k)$ (more details in section 3.2).

- a non-uniform local actuator that applies the N fast-rate control actions, being the first of them injected after the round-trip time delay $\tau(k)$, and the rest ones applied at period T . Then, a non-uniform actuation pattern arises at instants $\{\tau(k), T, 2T, \dots, (N-1)T\}$ ⁴.
- a local timer shared by the local devices in such a way that all of them are perfectly synchronized. So, the round-trip time delay $\tau(k)$ can be measured subtracting packet sending and receiving times, not being needed time-stamping techniques.

Capturing different network loads in the described NCS, a Markov chain-driven network model with n_0 different states, r_i , can be generated [27]. A transition matrix Γ indicating the transition probabilities $q_{i,j}$ between two states will be considered, taking this form

$$\Gamma = \{q_{i,j}\}, \quad i, j \in \{1, \dots, n_0\} \quad (2)$$

where

$$q_{i,j} = P\{r_{k+1} = j \mid r_k = i\}, \quad i, j = 1, 2, \dots, n_0 \quad (3)$$

being $\sum_{j=1}^{n_0} q_{i,j} = 1, \forall i$.

⁴ Note, if the delay were $\tau(k) \geq dT, d \in \mathbf{N}^+$, the first d control actions would not be applied.

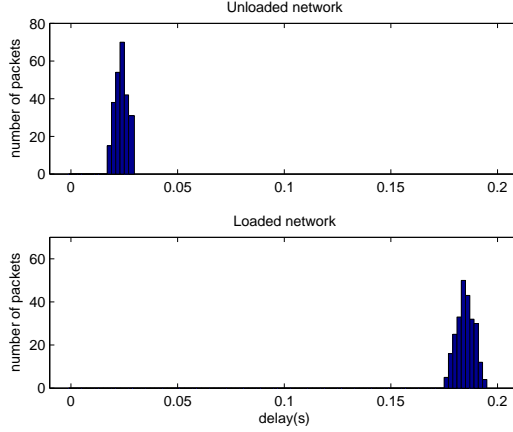


Fig. 2. Delay distributions for each network state

Let us propose a basic network model with two different states: one state represents an unloaded network r_1 , and the other state, a loaded network r_2 . In a two-state Markov chain, the transition matrix takes this form

$$\Gamma = \begin{pmatrix} q_{1,1} & 1 - q_{1,1} \\ 1 - q_{2,2} & q_{2,2} \end{pmatrix} \quad (4)$$

where, in this simple case, $q_{1,1}, q_{2,2}$ can be obtained from a priori experimental data, dividing the amount of time spent in each state by the total amount of time of the experiment.

Since, in the present work, a probabilistic model of the network delay $\tau(k)$ will be assumed, then a probability density function $p_i(\tau(k))$ for each state in the Markov chain is supposed to be known. Figure 2 shows an example of these functions.

Following with the two-state Markov chain, let us consider three different network contexts to be used later in the simulation example (section 5):

- $q_{1,1} = 1, q_{2,2} = 0$: unloaded network context (the network is always in the unloaded state r_1), presenting a lower average delay.
- $q_{1,1} = 0, q_{2,2} = 1$: loaded network context (the network is always in the loaded state r_2) that shows a higher average delay.
- $q_{1,1} = q_{2,2} = 0.5$: mix network context, where the network is switching between both states with the same probability.

Due to the time-varying nature of the network-induced delay, in order to assure stability in this Markov chain-driven NCS, some conditions will be enunciated in terms of probabilistic LMIs, in which both probability density functions $p_i(\tau(k))$ and transition probabilities $q_{i,j}$ will be taken into account

(more details in section 4).

3 Event-based remote and local controllers

3.1 Path tracking controller

After elapsing $\tau_{loc-rem}(k)$ time instants from the process output sampling $(w_R, w_L)_k$, this output arrives to the remote side. At this moment, the remote control algorithm is triggered to compute the slow-rate rotational velocity references $(w_{R.ref}, w_{L.ref})_k$, carrying out the next steps:

- kinematics computation, including:
 - the rotational (w) and translational (v) velocity calculation [12]:

$$\begin{pmatrix} v \\ w \end{pmatrix}_k = \begin{pmatrix} \rho/2 & \rho/2 \\ \rho/\varpi & -\rho/\varpi \end{pmatrix} \begin{pmatrix} w_R \\ w_L \end{pmatrix}_k \quad (5)$$

where ρ is the radius of the wheel, and ϖ is the width of the UGV base.

- the current UGV state $(x_c, y_c, \theta_c)_k$ calculation (being (x_c, y_c) the position, and $\theta_c(k)$ the posture angle):

$$\begin{aligned} x_c(k) &= x_c(k-1) + v(k)NT \cos(\theta_c(k-1) + w(k)NT/2) \\ y_c(k) &= y_c(k-1) + v(k)NT \sin(\theta_c(k-1) + w(k)NT/2) \\ \theta_c(k) &= \theta_c(k-1) + w(k)NT \end{aligned} \quad (6)$$

where $(x_c, y_c, \theta_c)_{k-1}$ must be initialized at the first iteration ($k=1$), $v(k)$ and $w(k)$ are obtained in (5), and NT is the slow sampling period.

- QC algorithm, developing (for details see in [49]):
 - computation of the error between the reference UGV state $(x_r, y_r, \theta_r)_k$ and the current UGV state $(x_c, y_c, \theta_c)_k$ (by transforming from the world coordinate to the robot coordinate):

$$\begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix}_k = \begin{pmatrix} \cos \theta_c & \sin \theta_c & 0 \\ -\sin \theta_c & \cos \theta_c & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{pmatrix}_k \quad (7)$$

where $(x_r, y_r, \theta_r)_k$ is provided by the reference generator (described below).

- calculation of the quadratic curve coefficients $A(k)$ and $K(k)$:

$$A(k) = \text{sign}(e_x(k)) \frac{e_y(k)}{e_x^2(k)} \quad (8)$$

being $y_M=A(k)x_M^2$ the quadratic curve in the robot coordinate, where (x_M, y_M) is the position in this coordinate.

$$K(k) = \text{sign}(e_x(k)) \frac{\alpha}{1 + |A(k)|} \quad (9)$$

in such a way that x_M at $kNT \leq t < (k+1)NT$ is given by $x_M=K(k)(t - kNT)$, and $\alpha > 0$ is the speed of the robot.

- rotational and translational velocity reference generation:

$$v_{ref}(k) = K(k) \quad (10)$$

$$w_{ref}(k) = 2A(k)K(k) \quad (11)$$

which are widely-used approximations from the original equations when the period of time $t - kNT$ is very small.

- generation of the slow-rate rotational velocity reference for each motor (the right and left motor references):

$$\begin{pmatrix} w_{R.ref} \\ w_{L.ref} \end{pmatrix}_k = \begin{pmatrix} 1/\rho & \varpi/2\rho \\ 1/\rho & -\varpi/2\rho \end{pmatrix} \begin{pmatrix} v_{ref} \\ w_{ref} \end{pmatrix}_k \quad (12)$$

- reference generator, which includes two computations:
 - calculation of the length $d_0(k)$ between the reference point $(x_r, y_r, \theta_r)_k$ and the UGV state $(x_c, y_c, \theta_c)_k$:

$$d_0(k) = \frac{d_{max}}{1 + \beta|A(k)|} \quad (13)$$

where d_{max} is the maximum distance between the reference point and the UGV, β is some positive constant, and $A(k)$ is provided by the QC algorithm (see in (8)).

- generation of the next reference point $(x_r, y_r, \theta_r)_{k+1}$ from $d_0(k)$ and taking into account the desired path. Note, the first reference value $(x_r, y_r, \theta_r)_1$ must be initialized.

3.2 Gain scheduling multi-rate PID controller

After elapsing $\tau(k)$ time instants from the process output sampling $(w_R, w_L)_k$, the rotational velocity references for each motor $(w_{R.ref}, w_{L.ref})_k$ (remember (12)) arrive to the local side. This event triggers the local (dynamic) control algorithm, where a multi-rate PID controller for each motor is implemented. In this way, the N delay-dependent control actions⁵ $\bar{U}_k=(u_1(\tau(k)), u_2(\tau(k)), \dots,$

⁵ Where, on the sequel, $(.)^T$ means transpose function.

$u_N(\tau(k))_k^T$ to be non-uniformly applied to every motor are generated. The design of \bar{U}_k is performed by means of this causal⁶ lifted discrete state-space realization $\Sigma_R = (A_R, B_R, C_R, D_R)$ [17]:

$$\begin{aligned}\psi(k+1) &= A_R\psi(k) + B_R e(k) \\ \bar{U}_k &= C_R\psi(k) + D_R e(k)\end{aligned}\tag{14}$$

being in this work

$$\begin{aligned}A_R &= \begin{pmatrix} 1 & 0 \\ 0 & f(\tau(k)) \end{pmatrix} \\ B_R &= \begin{pmatrix} K_i(\tau(k)) \\ 1 - f(\tau(k)) \end{pmatrix} \\ C_R &= \begin{pmatrix} 1 & -K_d(\tau(k)) \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \end{pmatrix} \\ D_R &= \begin{pmatrix} K_p(\tau(k)) + K_d(\tau(k)) \\ K_p(\tau(k)) \\ \vdots \\ K_p(\tau(k)) \end{pmatrix}\end{aligned}\tag{15}$$

and where $\psi(k)$ is the state vector, $e(k) = (w_{R.ref} - w_R)_k$ is the error signal⁷ for the right motor (it can be analogously expressed for the left motor), and $K_p(\tau(k)), K_i(\tau(k)), K_d(\tau(k)), f(\tau(k))$ are, respectively, the proportional, integral and derivative gains, and a derivative noise-filter pole. All of these controller parameters can be denoted as the delay-dependent gain vector $\theta(\tau(k)) = (K_i(\tau(k)), K_p(\tau(k)), K_d(\tau(k)), f(\tau(k)))^T$, which will be computed by means of the following gain scheduling approach:

$$\theta(\tau(k)) = \theta(0) + M\tau(k)\tag{16}$$

⁶ To fulfill causality constraints [50], [45], [37], D_R must be a block lower triangular matrix. Concretely, in our case, D_R is a column matrix.

⁷ Note, as in this approach the round-trip time delay is measured and compensated at the local side, the sampled rotational velocity is considered to compute the error signal instead of the instantaneous velocity like in [30], [31], [43], [22], [44].

where the new variables $\theta(0)$ and M are

- $\theta(0) = (K_i(0), K_p(0), K_d(0), f(0))^T$ is the no-delay, nominal gain vector, which can be designed via classical procedures such as direct design [28], time response optimization [4], or discretization of continuous PID controllers [4].
- M is denoted as the scheduling vector, which is deduced after solving a least-square problem on the minimization of the first-order Taylor term of $\|\pi(\tau(k), \theta(\tau(k))) - \pi(0, \theta(0))\|$, being $\pi(\tau(k), \theta(\tau(k)))$ a performance vector defined by closed-loop poles. The solution of the proposed problem yields

$$M = -(\Delta^T W^T W \Delta)^{-1} W^T \Delta^T \delta_\tau \quad (17)$$

where W is a weighting filter (to give priority to dominant closed-loop poles), Δ is a Jacobian matrix that includes the derivatives $\frac{\partial \pi}{\partial \theta_i}$ evaluated at the nominal point ($\tau = 0$, $\theta(\tau(k)) = \theta(0)$) for each controller parameter θ_i , and δ_τ is the derivative related to the delay, $\delta_\tau = \frac{\partial \pi}{\partial \tau}$, evaluated at the same nominal point.

The scheduling law in (16) tries to maintain the no-delay, nominal control performance despite network delays.

4 Stability and performance analysis

From a causal⁸ lifted plant model $\Sigma_P = (A_P, B_P, C_P, D_P)$ [17]

$$\begin{aligned} \tilde{x}(k+1) &= A_P \tilde{x}(k) + B_P \bar{U}_k \\ w_R(k) &= C_P \tilde{x}(k) + D_P \bar{U}_k \end{aligned} \quad (18)$$

where $\tilde{x}(k)$ is the plant state vector⁹, it is well-known that the feedback connection of an output feedback controller $\Sigma_R = (A_R, B_R, C_R, D_R)$ and a process $\Sigma_P = (A_P, B_P, C_P, D_P)$ in a lifted framework results in a dynamical system governed by [2]:

$$\bar{x}(k+1) = \begin{pmatrix} \psi_{k+1} \\ \tilde{x}_{k+1} \end{pmatrix} = \begin{pmatrix} A_R & -B_R C_P \\ B_P C_R & A_P - B_P D_R C_P \end{pmatrix} \begin{pmatrix} \psi_k \\ \tilde{x}_k \end{pmatrix} = \bar{A}_{cl} \bar{x}(k) \quad (19)$$

⁸ Causality constraints are automatically satisfied when the lifted model is derived from a continuous-time model [37].

⁹ In (18) the right motor output $w_R(k)$ is presented, but analogously the left motor (with the output $w_L(k)$) could have been considered.

As the network delay $\tau(k)$ can vary from sensing period to sensing period, then \bar{A}_{cl} in (19) must be replaced by $\bar{A}_{cl}(\tau(k))$ to represent a discrete LTV system

$$\bar{x}(k+1) = \bar{A}_{cl}(\tau(k))\bar{x}(k) \quad (20)$$

4.1 Mean-square stability for a Markov chain-driven NCS.

Theorem The NCS defined by the closed-loop state in (20), where the network-induced delay $\tau(k)$ is driven by a finite state Markov chain with state-space $\{r_i\}, i \in \{1, 2, \dots, n_0\}$ and transition probability matrix Γ with elements $q_{i,j}$, being each network state r_i described by a probability density function of the network delay $p_i(\tau(k))$, is globally mean-square asymptotically stable around the null initial state if there exist positive-definite matrices $Q(1), Q(2), \dots, Q(n_0)$ such that

$$\sum_{j=1}^{n_0} q_{i,j} \left(\sum_{m=1}^l p_i(\vartheta_m) \bar{A}_{cl}(\vartheta_m)^T Q(j) \bar{A}_{cl}(\vartheta_m) \right) - \alpha^2 Q(i) < 0, \quad (21)$$

$$\forall i, j = 1, \dots, n_0$$

where $0 < \alpha < 1$ is the performance objective (an average decay rate), $\vartheta_m, m \in \{1, \dots, l\}$ is a dummy parameter ranging in a set $\Theta = [0, \tau_{\max}]$ where the time-varying parameter $\tau(k)$ is assumed to take values in (usually, $\vartheta_1 = 0, \vartheta_l = \tau_{\max}$, and the rest of values ϑ_m are l -equally spaced), and matrices $Q(i)$ are composed of decision variables to be found by a semi-definite programming solver [42].

Proof Assuming the random variation of the network delay, stability of the closed-loop system will be analyzed in the mean-square sense by means of a quadratic Lyapunov function

$$V(\bar{x}(k), r_k) = \bar{x}(k)^T Q(r_k) \bar{x}(k) \quad Q(r_k) > 0 \quad (22)$$

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

$$E[V(\bar{x}(k+1), r_{k+1})] \leq E[V(\bar{x}(k), r_k)] \quad (23)$$

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

$$E[V(\bar{x}(k+1), r_{k+1})] \leq \alpha^2 E[V(\bar{x}(k), r_k)] \quad (24)$$

If the closed-loop equations in (22) are replaced in (24), the expected value of the difference between two consecutive samples yields:

$$\begin{aligned}
& E[\Delta V \mid \bar{x}(k) = \bar{x}, r_k = i] = \\
& = E[V(\bar{x}(k+1), r_{k+1}) - \alpha^2 V(\bar{x}(k), r_k) \mid \bar{x}(k) = \bar{x}, r_k = i] = \\
& = E[\bar{x}(k+1)^T Q(r_{k+1}) \bar{x}(k+1) \mid \bar{x}(k) = \bar{x}, r_k = i] - \bar{x}^T \alpha^2 Q(i) \bar{x} = \\
& = E[\bar{x}^T \bar{A}_{cl}(\tau(k))^T Q(r_{k+1}) \bar{A}_{cl}(\tau(k)) \bar{x} \mid r_k = i] - \bar{x}^T \alpha^2 Q(i) \bar{x} = \\
& = \sum_{j=1}^{n_0} q_{i,j} (\bar{x}^T \bar{A}_{cl}(\tau(k))^T Q(j) \bar{A}_{cl}(\tau(k)) \bar{x}) - \bar{x}^T \alpha^2 Q(i) \bar{x} = \\
& = \bar{x}^T \left(\sum_{j=1}^{n_0} q_{i,j} (\bar{A}_{cl}(\tau(k))^T Q(j) \bar{A}_{cl}(\tau(k))) - \alpha^2 Q(i) \right) \bar{x}
\end{aligned} \tag{25}$$

The LMI gridding in [34] can be extended to a probabilistic case where the network-induced delay $\tau(k)$:

- is driven by a finite state Markov chain, being each state described by a probability density function $p_i(\tau(k))$.
- is assumed to take values in a bounded set $\Theta = [0, \tau_{\max}]$.

In addition, if a dense enough set of l -equally spaced values $\vartheta_m, m \in \{1, \dots, l\}$ is used as a dummy parameter ranging in Θ , then the previous Lyapunov decrecence condition (25) can be rewritten as (21).

4.2 Mean-square stability for a stochastic NCS (with no Markov chain), and robust stability

If no transition probabilities $q_{i,j}$ were known, the proposed Markov chain model could not be used. Under this consideration, stability could be analyzed either following an stochastic framework (with only the probability density functions $p_i(\tau(k))$) or even following a robust approach (with no statistical information). For the first case, following similar steps to those presented in the previous subsection, the next LMI can be deduced in order to prove stochastic stability:

$$\sum_{m=1}^l p(\vartheta_m) \bar{A}_{cl}(\vartheta_m)^T Q \bar{A}_{cl}(\vartheta_m) - \alpha^2 Q < 0 \tag{26}$$

For the second case, stability must be ensured for any arbitrary delay change (with unknown probability) inside the interval $\Theta = [0, \tau_{\max}]$, resulting the worst-case performance. Similarly to the previous subsection, the next LMI can be deduced in order to prove robust stability:

$$\bar{A}_{cl}(\vartheta_m)^T Q \bar{A}_{cl}(\vartheta_m) - \alpha^2 Q < 0, \quad \forall \vartheta_m \in \Theta \tag{27}$$

4.3 Cost functions

To ensure stability in the previous sections, the decay rate is utilized as a performance objective. In order to carry out a more exhaustive evaluation of the performance for the proposed control strategy, three cost functions will be used. The first one, J_1 , is based on the ℓ_2 -norm, and its goal is to provide a measure about how accurate the path is followed:

$$J_1 = \sqrt{\sum_{k=1}^l (x_c - x_r)_k^2 + (y_c - y_r)_k^2} \quad (28)$$

where k are iterations at NT (as previously used), being l of them required by the UGV to reach the final point of the path. The rest of variables were defined in section 3.1.

To know the maximum difference between the desired path and the current UGV position, a second cost function J_2 is defined, which is based on the ℓ_∞ -norm:

$$J_2 = \max_{1 \leq k \leq l} \{|(x_c - x_r)_k|, |(y_c - y_r)_k|\} \quad (29)$$

And finally, the third cost function J_3 measures the total amount of time (in seconds) elapsed to arrive at the final destination:

$$J_3 = lNT \quad (30)$$

5 Simulation results

In this section, a simulation example is presented in order to illustrate the main benefits of the control proposal. The section is split into three parts. In the first subsection, the selection of the dual-rate local (dynamic) controller is justified to be the nominal, no-delay one (i.e. to be later retuned when delays are taken into account); in the second subsection, the LMIs introduced in section 4 will be analyzed in order to ensure stability in the proposed time-varying NCS; and the third subsection exploits a Truetime [29] application in order to evaluate the cost functions exposed in 4.3 and other control performance aspects.

5.1 Selection of the nominal local controller

The UGV to be controlled can be represented by the next transfer function

$$G(s) = \frac{w(s)}{V(s)} = \frac{30.5627}{(s + 15.97)} \quad (31)$$

being $V(s)$ the applied voltage in V, and $w(s)$ the rotational velocity for the right or the left motor, indistinctly, in rad/s.

In order to justify the selection of the dual-rate PID controller as the nominal one, a comparison between single-rate and dual-rate PID controllers in the no-delay situation is carried out. In the case of the dual-rate controller, two different values for N ($N = 2$ and $N = 4$) will be studied.

In the next subsections, when coping with delays, the example will consider the Markov chain-driven NCS described in section 2 with probability density functions depicted in Figure 2, which shows a maximum network delay $\tau_{max} = 0.195$ s. Then, in order to deal with this largest delay, the output sampling time will be 0.2s (assuring $\tau_{max} < NT$). So now, the comparison for the no-delay context takes this sensing period so as to treat three different design options, being the main goal of them to achieve a response as fast as possible:

- single-rate PID controller ($T = 0.2$ s) which is fine-tuned by hand from an initial root-locus design. It is the bilinear discretisation of

$$K_p \left(1 + K_i \frac{1}{s} + K_d \frac{s}{f s + 1} \right)$$

for the values: $K_p = 0.3$, $K_i = 10.5$, $K_d = 0.04$, $f = 0.5$.

- dual-rate PID controller with $N = 2$ ($T = 0.1$ s, $NT = 0.2$ s), which corresponds to the realization (15) with values: $K_p(0) = 0.5$, $K_i(0) = 0.5$, $K_d(0) = 0.04$, $f(0) = 0.25$.
- the same previous dual-rate PID controller but with $N = 4$ ($T = 0.05$ s, $NT = 0.2$ s).

Simulation results for every option are depicted in Figure 3. These are the main conclusions:

- the three responses are able to approximately achieve the same settling time ($\simeq 0.4$ s) but, while the dual-rate controllers present no overshoot, the single-rate controller shows an overshoot of almost 20%.
- increasing N in the dual-rate strategy does not imply an important control performance improvement.¹⁰

¹⁰ As a rule of thumb, the higher N is, the better control performance may be, although the improvement may not always be significant.

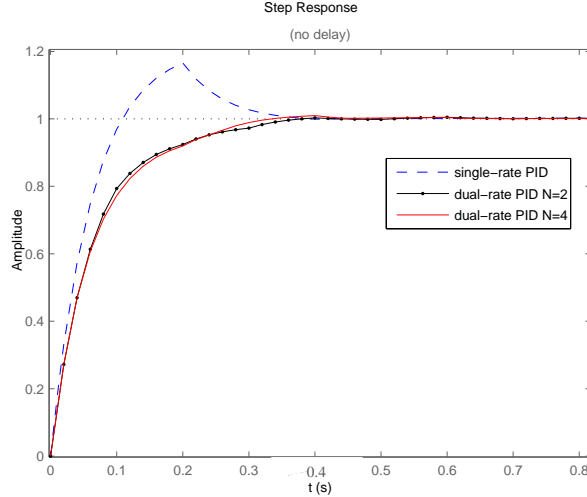


Fig. 3. Single vs. dual-rate comparison (continuous inter-sample response) for the no-delay case

From these conclusions, the dual-rate PID controller with $N = 2$ is chosen as the nominal local controller. Then, summarizing, the case study to be developed in the next subsections will consider:

- the UGV described in (31).
- the Markov chain-driven NCS described in section 2, with probability density functions depicted in Figure 2, and with the transition matrix defined in (4). In addition, the three network contexts presented at the end of section 2 (unloaded, loaded, and mix) will be analyzed.
- sampling and actuation periods: $NT = 0.2\text{s}$, and $T = 0.1\text{s}$ ($N = 2$).
- no-delay, nominal gain vector: $\theta(0) = (K_i(0), K_p(0), K_d(0), f(0))^T = (0.50, 0.50, 0.04, 0.25)^T$, in order to achieve the dynamic control performance previously defined (settling time $t_s \approx 0.4\text{s}$, and no overshoot).
- scheduling vector (from (17); details omitted for brevity): $M = (K_i, K_p, K_d, f)^T = (0.9401, -0.3075, -0.1259, 0.1306)^T$.

5.2 Stability study

From the considered Markov chain-driven NCS, the LMI in (21) is evaluated. Table 1 points out the consequent results for each control strategy (scheduled, non-scheduled) in each network context (unloaded, loaded, mix). The main conclusions are:

- all the decay-rate values are less than 1, indicating stable control systems.
- when the network is unloaded ($q_{1,1} = 1, q_{2,2} = 0$), both multi-rate PID controllers (the scheduled one and the non-scheduled one) behave in a similar

Table 1
LMI decay-rate (with Markov chain)

network context	scheduled	non-scheduled
unloaded	0.29	0.27
loaded	0.79	0.88
mix	0.62	0.75

way.

- when the network is loaded ($q_{1,1} = 0, q_{2,2} = 1$), the scheduled controller presents a better decay-rate value (around 10% less) than the non-scheduled one, and hence a better performance is expected for the former controller.
- when considering the mix context ($q_{1,1} = q_{2,2} = 0.5$), intermediate figures are obtained, and then both controllers improve their behavior with respect to the loaded context. In this case, the scheduled controller presents a 20% lower decay-rate than the non-scheduled controller.

In order to perform a wider study for both control strategies, decay-rate values for other transition probabilities will be depicted by means of surfaces. In Figure 4, the surfaces for every control strategy are together shown to better appreciate the performance improvement (lower decay-rates) in the scheduled approach. The following concerns can be stated:

- when $q_{1,1} = q_{2,2} = 1$, the worst case in Table 1 is reached, that is, the decay-rate of the loaded network state (0.79 and 0.88, resp., for scheduled¹¹ and non-scheduled controller).
- when $q_{1,1} = q_{2,2} = 0$, the network is always switching between both states, and an intermediate performance is achieved (0.65 and 0.81, resp., for scheduled and non-scheduled controller¹²).
- both surfaces are inclined to the values $q_{1,1} = 1, q_{2,2} = 0$, since the lower $q_{2,2}$ and the higher $q_{1,1}$ are, the better (lower) decay-rate will be. Thus, the less the network is loaded, the better performance can be guaranteed. As previously commented, only this case (with $q_{1,1} = 1, q_{2,2} = 0$) presents similar values for both controllers.
- the results obtained for the unloaded ($q_{1,1} = 1, q_{2,2} = 0$) and loaded ($q_{1,1} = 0, q_{2,2} = 1$) contexts become lower and upper bounds, resp., in the stability analysis.

Finally, two last cases will be treated. In both cases, statistical information is not known, and hence, the proposed Markov chain cannot be used to prove

¹¹ This value cannot be observed in the figure but it coincides with that obtained when $q_{1,1} = 0, q_{2,2} = 1$.

¹² These values are slightly worse ($\simeq 5\%$ for the scheduled controller, and $\simeq 10\%$ for the non-scheduled controller) than those obtained for the mix context (with $q_{1,1} = q_{2,2} = 0.5$) in Table 1.

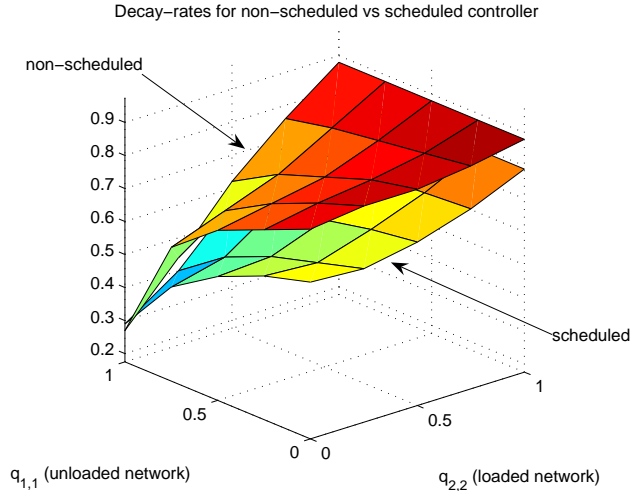


Fig. 4. Decay-rate comparison

stability. These are the two cases:

- if no transition probability $q_{i,j}$ is known (that is, only probability density functions of the network delay $p_i(\tau(k))$ are considered), a probabilistic LMI analysis (26) can be separately carried out for each network state (that is, either for the unloaded network context or for the loaded network context¹³). As expected, the decay-rate values obtained for this analysis coincide with those for the unloaded and loaded network contexts presented in Table 1.
- if no statistical information is known (that is, neither transition probabilities nor probability density functions), a robust LMI analysis (27) can be carried out. As probability density functions are not known, no difference between network contexts can be contemplated. The results shown in Table 2 yield the worst-case performances for the scheduled and non-scheduled approaches. Note that for the non-scheduled strategy, the resultant decay-rate is $\alpha = 1.04$, which is out of the range $0 < \alpha < 1$ required to ensure stability. Then, under this condition, the non-scheduled control system becomes unstable for some delay combination.

Table 2

LMI decay-rate (with no statistical information)

scheduled	non-scheduled
0.91	1.04

¹³ Since no transition probabilities are known, the drawback of this analysis is not being able to treat mix contexts such as in the Markov chain approach.

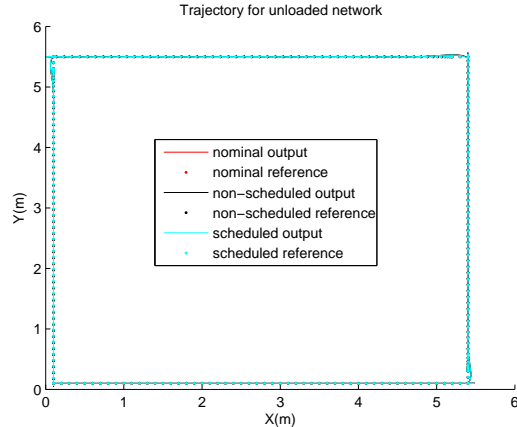


Fig. 5. XY path for unloaded network

5.3 Truetime implementation. Evaluation of cost functions.

The Truetime application is developed considering a Wireless Local Area Network (WLAN). In order to generate different load conditions in the Truetime application framework, and hence different network contexts, a set of interference nodes has been added to the proposed NCS.

The desired path to be tracked by the UGV is approximately a 30m^2 looped, rectangle path, where the starting and goal points are located near the coordinate origin $(0,0)$ ¹⁴ (see in Figure 5). The path consists of straight lines, which are gone across following an anti-clockwise movement. In this experiment, the next parameters are considered¹⁵ (remember section 3.1): $\rho=0.0275$, $\varpi=0.16$, $d_{max}=0.1$, $\alpha=\beta=0.5$. Figures 6, 7 and 8 show a zoom of the path tracking, where both the current UGV state (output) $(x_c, y_c, \theta_c)_k$ and the reference UGV state $(x_r, y_r, \theta_r)_k$ reached by each controller (the nominal controller, the non-scheduled controller, and the scheduled controller) are depicted. As a result, the next aspects can be observed:

- for every evaluated case (including the nominal one), some spike is observed when arriving to the corner. This spike appears as a consequence of the values selected for the controller's parameters when designing the PID controller. So, if K_p were reduced around 20% and K_i around 40%, the spike would be practically eliminated, but the settling time would be approximately worsened 100% (see Figure 6).
- in the unloaded context (Figure 6), the non-scheduled and the scheduled controllers behave in a similar way (outputs are practically overlapped), and they follow accurately the nominal performance. This conclusion can be confirmed by computing the cost functions presented in section 4.3. For the

¹⁴ Concretely, $(x_r, y_r, \theta_r)_1=(0.1,0.1,0)$, and $(x_c, y_c, \theta_c)_0=(0.09,0.1,0)$.

¹⁵ All of them in SI units.

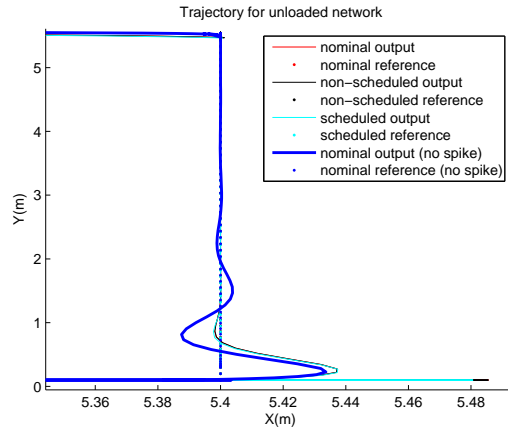


Fig. 6. XY path for unloaded network (zoom)

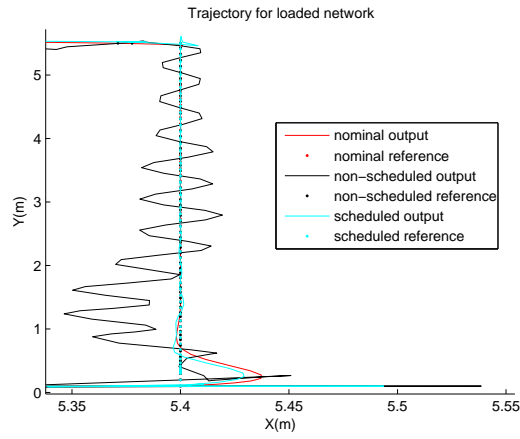


Fig. 7. XY path for loaded network (zoom)

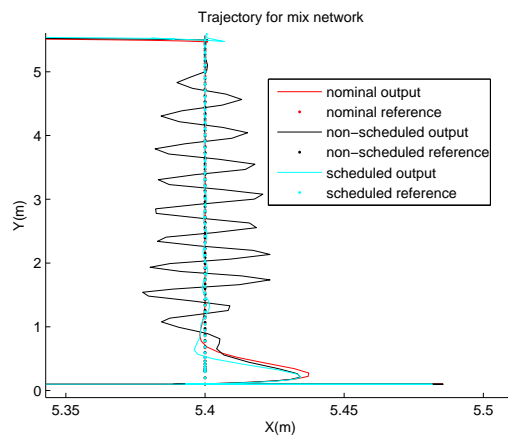


Fig. 8. XY path for mix network (zoom)

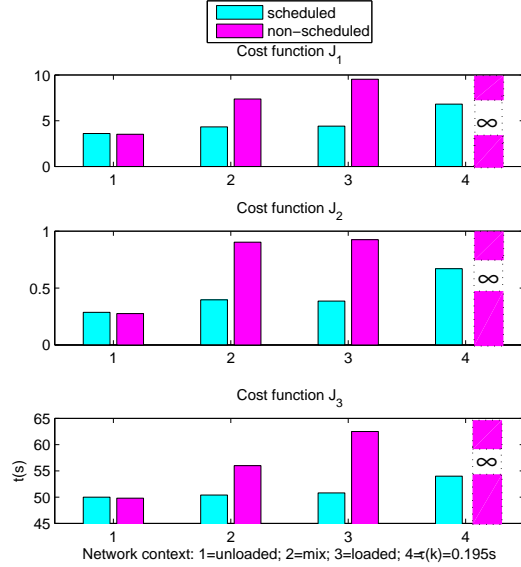


Fig. 9. Cost indexes J_1, J_2, J_3

nominal controller, the cost functions present these values: $J_1 = 3.52$, $J_2 = 0.27$, $J_3 = 49.6s$. As shown in Figure 9, in this unloaded context, similar values to the previous ones are achieved both for the non-scheduled controller and for the scheduled one.

- in the loaded context (Figure 7), the non-scheduled controller worsens dramatically its behavior, whereas the scheduled controller tries to keep the nominal specification. As shown in Figure 9, with respect to the unloaded context, evident increments of the cost functions for the non-scheduled controller, but slight ones for the scheduled strategy, validate this fact. Concretely, for the non-scheduled controller, J_1 and J_2 are incremented around 200%, and J_3 around 25%; nevertheless, the scheduled controller experiments less increments ($\simeq 20\%$ for J_1 , $\simeq 30\%$ for J_2 , and $\simeq 2\%$ for J_3).
- in the mix context (Figure 8), once the UGV overtakes the bottom corner, the Markov chain switches from the unloaded network state to the loaded one, being patent a path tracking worsening for the non-scheduled controller. The behavior in this case is retrieved when approaching to the top corner, since at this moment the chain switches to the unloaded network state. Nevertheless, the behavior obtained from the scheduled controller is slightly affected by the switching. Once again, this aspect can be checked by observing the values obtained for each cost function (Figure 9). On one hand, for the non-scheduled controller, whereas J_1 and J_3 present intermediate values between the two previous contexts (now, regarding the unloaded case, J_1 worsens around 90%, and J_3 around 15%), J_2 takes a similar value with respect to the loaded network framework (200% greater than in the unloaded context). On the other hand, the scheduled controller reaches cost indexes that are similar to those obtained in the loaded context, which are a bit worse than in the unloaded network situation (as previously com-

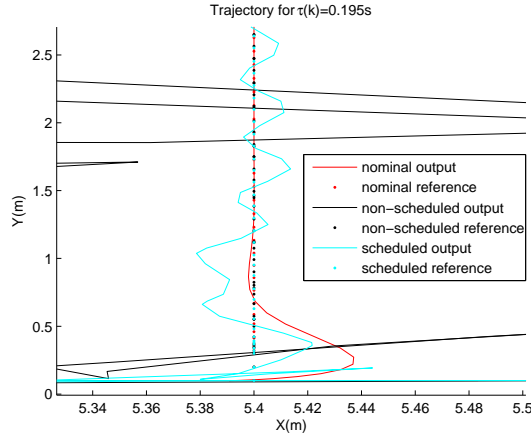


Fig. 10. XY path for $\tau(k)=0.195s$ (zoom)

mented).

- all of these conclusions confirm those based on decay-rate figures, which were presented in Table 1.

Finally, let us simulate the NCS behavior when no Markov chain is considered, that is, when delays can arbitrarily change (with no statistical information¹⁶) inside the interval $\Theta = [0, \tau_{\max}]$. Different sequences of network-induced delays could be treated, but let us consider a long time-invariant network-induced delay $\tau(k)=\tau_{\max}=0.195s$. For this delay, whereas the non-scheduled NCS becomes unstable (and then the UGV is not able to follow the desired path), the scheduled controller tries to track the path but worsening the performance with respect to the Markov chain framework (see in Figure 10). This conclusion confirms that deduced in the previous section where, for some combination of the network-induced delay, a decay-rate degradation could be expected for the scheduled control strategy, and even instability, for the non-scheduled one (remember Table 2). Analyzing cost functions (see in Figure 9), now the scheduled controller presents greater values with respect to those obtained in the Markov chain case, corroborating the previous conclusions. As, in this case, the non-scheduled approach becomes unstable, its cost indexes tend to infinity (marked as ∞ in Figure 9).

¹⁶In this case, neither probability density functions nor transition probabilities are known. Considering only the former ones is equivalent to separately treat either the unloaded or the loaded network context in the Markov chain framework (as previously carried out).



Fig. 11. Mobile robot used in the experimental test-bed

6 Experimental results

Once the simulation studies have been deemed satisfactory, the controller parameters obtained in the previous section have been directly applied to control an experimental test-bed, which includes the following devices:

- one computer acting as the event-based path tracking controller located at the remote side.
- one mobile robot (UGV), which is built from the LEGO Mindstorms NXT hardware [1] resulting in a classical tribot (see in Figure 11), and whose input/output behavior complies with (31). In the LEGO NXT "intelligent" brick, the event-based gain-scheduling multi-rate controller located at the local side is implemented.

In the implementation, some specific time delay injection code is included in order to obtain different load scenarios (acting as the interference nodes in the Truetime simulation). After carrying out several experiments, the same trend seen in the simulation is now observed, that is, the scheduled controller points out a better behavior than the non-scheduled controller as soon as delays are longer.

Next, for the sake of simplicity, let us only show one of these experiments in which the worst case (loaded network context) is treated. Figure 12 depicts the histogram for the round-trip time delays appeared in the experience, stating a similar shape to that shown in Figure 2 for the loaded network. Figure 13 shows a zoom of the experimental path tracking, where both the current UGV state (output) $(x_c, y_c, \theta_c)_k$ and the reference UGV state $(x_r, y_r, \theta_r)_k$ reached by each control strategy¹⁷ are illustrated. Comparing Figure 13 to Figure 7, the control proposal seems to be validated, since the scheduled controller follows

¹⁷ Obviously, since delays are inherent to the experimental platform, now it is not possible to get the nominal, no-delay output as obtained in simulation.

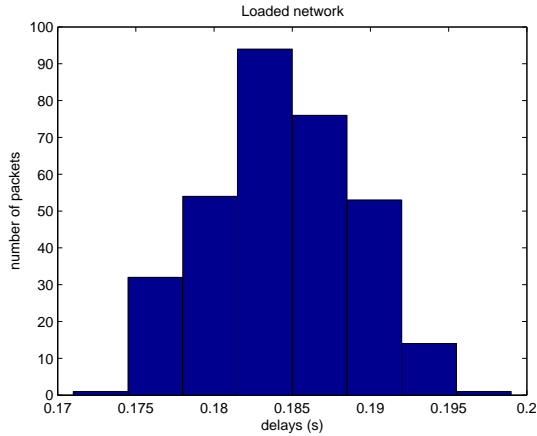


Fig. 12. Experimental delay distribution for loaded network

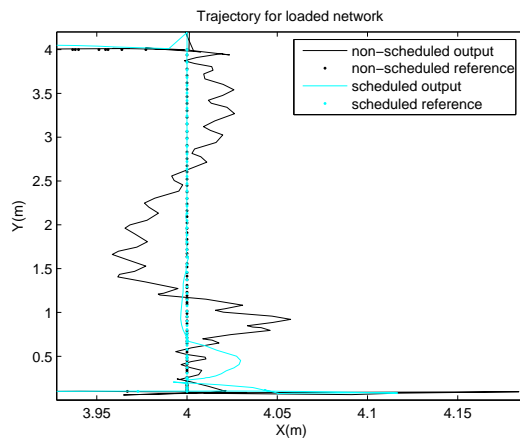


Fig. 13. Experimental XY path for loaded network (zoom)

the path better than the non-scheduled controller, and those oscillations detected in the simulation for the non-scheduled strategy are also visible in the experience. Cost functions confirm the previous conclusion (Figure 14), since all of them appear incremented in the non-scheduled case (as previously noted in Figure 9 for the loaded network context).

7 Conclusions

In this work, a Markov chain-driven NCS is presented, where the plant to be wirelessly controlled is a UGV. The output sampling period can be chosen to be greater than the longest delay, avoiding the packet disorder phenomenon. Consequently, to improve dynamic control performance, multi-rate control techniques are used in such a way that the UGV is sensed at the long period (slow rate) and actuated at an N times shorter period (fast rate).

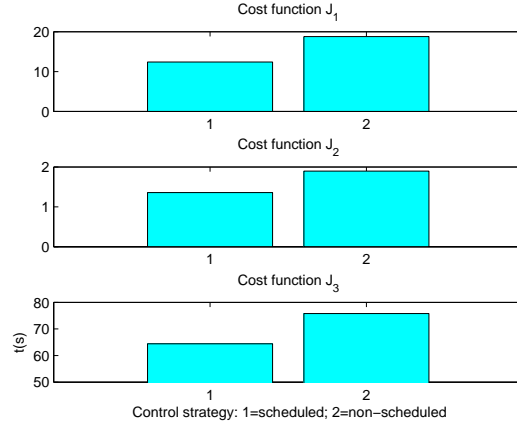


Fig. 14. Cost indexes J_1, J_2, J_3 for the experiment

The non-uniform multi-rate control system is decomposed into two sides: a remote side carrying out the path tracking control and generating slow-rate velocity references from UGV states, and a local side compensating for time-varying network-induced delays and converting the slow-rate references into fast-rate control actions, which are non-uniformly applied to the UGV. Event-based controllers are implemented at both sides. Interesting advantages of the control proposal are the consideration of actual (not estimated) values for network delays and UGV states, a possible reduction of the network utilization, and a straightforward implementation.

Control system stability for the Markov chain-driven NCS is assured in terms of probabilistic LMIs. Simulation results point out the benefits of the proposed control strategy, which is validated on an experimental test-bed.

8 Acknowledgements

This work was supported in part by grants TEC2012-31506 from the Spanish Ministry of Education, DPI2011-28507-C02-01 by the Spanish Ministry of Economy, and PAID-00-12 from Technical University of Valencia (Spain). In addition, this research work has been developed as a result of a mobility stay funded by the Erasmus Mundus Programme of the European Commission under the Transatlantic Partnership for Excellence in Engineering (TEE Project).

References

- [1] *LabVIEW LEGO Mindstorms NXT Module Programming Guide*. National Instruments Corporation, [online] www.ni.com/pdf/manuals/372574c.pdf, 2009.
- [2] P. Albertos and A. Sala. *Multivariable Control Systems: An Engineering Approach*. Springer, 2004.
- [3] M. Araki. Recent developments in digital control theory. In *Proc. IFAC World Congr.*, volume 9, pages 951–960, 1993.
- [4] K.J. Astrom and T. Hagglund. *PID Controllers: Theory, Design, and Tuning*. *Instrum. Soc. of America*, 1995.
- [5] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Soc. Ind. Math., 1994.
- [6] M. Cimino and P.R. Pagilla. Design of linear time-invariant controllers for multirate systems. *Automatica*, 46(8):1315–1319, 2010.
- [7] M.B.G. Cloosterman, N. van de Wouw, W. Heemels, and H. Nijmeijer. Stability of networked control systems with uncertain time-varying delays. *IEEE Trans. Autom. Control*, 54(7):1575–1580, 2009.
- [8] A. Cuenca, P. García, P. Albertos, and J. Salt. A non-uniform predictor-observer for a networked control system. *Int. J. Control, Automation and Systems*, 9(6):1194–1202, 2011.
- [9] A. Cuenca and J. Salt. RST controller design for a non-uniform multi-rate control system. *J. Process Control*, 22(10):1865–1877, 2012.
- [10] H. Dong, Z. Wang, and H. Gao. Distributed H_∞ filtering for a class of Markovian jump nonlinear time-delay systems over lossy sensor networks. *IEEE Trans. Ind. Electron.*, 60(10):4665–4672, 2013.
- [11] M. Donkers, W. Heemels, D. Bernardini, A. Bemporad, and V. Schnerer. Stability analysis of stochastic networked control systems. In *Proc. American Control Conf.*, pages 3684–3689, 2010.
- [12] T. Fukao, H. Nakagawa, and N. Adachi. Adaptive tracking control of a nonholonomic mobile robot. *IEEE Trans. Robotics and Automation*, 16(5):609–615, 2000.
- [13] G. Guo. Linear systems with medium-access constraint and markov actuator assignment. *IEEE Trans. Circuits and Systems I: Regular Papers*, 57(11):2999–3010, 2010.
- [14] G. Guo and Z. Lu. Markov actuator assignment for networked control systems. *European Journal of Control*, 18(4):323–330, 2012.
- [15] G. Guo, Z. Lu, and Q-L. Han. Control with markov sensors/actuators assignment. *IEEE Trans. Autom. Control*, 57(7):1799–1804, 2012.

- [16] R. A. Gupta and M-Y. Chow. Networked control system: Overview and research trends. *IEEE Trans. Ind. Electron.*, 57(7):2527–2535, 2010.
- [17] P. Khargonekar, K. Poolla, and A. Tannenbaum. Robust control of linear time-invariant plants using periodic compensation. *IEEE Trans. Autom. Control*, 30(11):1088–1096, Nov. 1985.
- [18] G.M. Kranc. *Multi-rate Sampled Systems*. Dept. Electrical Engineering, Electronics Research Laboratories, Columbia University Engineering Center, 1956.
- [19] D. Li, S.L. Shah, and T. Chen. Analysis of dual-rate inferential control systems. *Automatica*, 38(6):1053–1059, Jun. 2002.
- [20] H. Li and Y. Shi. Network-based predictive control for constrained nonlinear systems with two-channel packet dropouts. *IEEE Trans. Ind. Electron.*, 61(3):1574–1582, 2014.
- [21] M. Liu, D. Ho, and Y. Niu. Stabilization of markovian jump linear system over networks with random communication delay. *Automatica*, 45(2):416–421, 2009.
- [22] C. Lozoya, P. Martí, M. Velasco, J.M. Fuertes, and E.X. Martín. Simulation study of a remote wireless path tracking control with delay estimation for an autonomous guided vehicle. *Int. J. Advanced Manufacturing Technology*, 52(5):751–761, 2011.
- [23] Z. Lu and G. Guo. Control with sensors/actuators assigned by markov chains: transition rates partially unknown. *IET Control Theory & Applications*, 7(8):1088–1097, 2013.
- [24] X. Luan, P. Shi, and C-L. Liu. Stabilization of networked control systems with random delays. *IEEE Trans. Ind. Electron.*, 58(9):4323–4330, 2011.
- [25] L.A. Montestruque and P.J. Antsaklis. Stochastic stability for model-based networked control systems. In *Proc. American Control Conf.*, volume 5, pages 4119–4124. IEEE, 2003.
- [26] L. Muscariello, M. Mellia, M. Meo, M. Ajmone Marsan, and R. Lo Cigno. Markov models of internet traffic and a new hierarchical MMPP model. *Computer Communications*, 28(16):1835–1851, 2005.
- [27] J. Nilsson. *Real-time control systems with delays*. Dept. Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998.
- [28] K. Ogata. *Discrete-time control systems*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1987.
- [29] M. Ohlin, D. Henriksson, and A. Cervin. *TrueTime 1.5 Reference Manual*.
- [30] U. Ojha and M-Y. Chow. Realization and validation of delay tolerant behavior control based adaptive bandwidth allocation for networked control system. In *Int. Symp. on Ind. Electron. (ISIE)*, pages 2853–2858. IEEE, 2010.

- [31] U. Ojha, B.R. Klingenberg, and M-Y. Chow. Dynamic bandwidth allocation based on curvature and velocity in a fleet of unmanned ground vehicles. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.
- [32] C. Peng, M-R. Fei, E. Tian, and Y-P. Guan. On hold or drop out-of-order packets in networked control systems. *Information Sciences*, 268:436–446, 2014.
- [33] L. Qiu, C. Liu, F. Yao, and G. Xu. Analysis and design of networked control systems with random markovian delays and uncertain transition probabilities. In *Abstract and Applied Analysis*, volume 2014. Hindawi Publishing Corp., 2014.
- [34] A. Sala. Computer control under time-varying sampling period: An LMI gridding approach. *Automatica*, 41(12):2077–2082, Dec. 2005.
- [35] A. Sala, Á. Cuenca, and J. Salt. A retunable PID multi-rate controller for a networked control system. *Information Sciences*, 179(14):2390–2402, Jun. 2009.
- [36] J. Salt, A. Cuenca, F. Palau, and S. Dormido. A multirate control strategy to the slow sensors problem: An interactive simulation tool for controller assisted design. *Sensors*, 14(3):4086–4110, 2014.
- [37] J. Sheng, T. Chen, and S.L. Shah. Generalized predictive control for non-uniformly sampled systems. *Journal of Process Control*, 12(8):875–885, 2002.
- [38] Y. Shi and B. Yu. Output feedback stabilization of networked control systems with random delays modeled by markov chains. *IEEE Trans. Autom. Control*, 54(7):1668–1674, 2009.
- [39] Y. Shi and B. Yu. Robust mixed H_2/H_∞ control of networked control systems with random time delays in both forward and backward communication links. *Automatica*, 47(4):754–760, 2011.
- [40] H. Shousong and Z. Qixin. Stochastic optimal control and analysis of stability of networked control systems with long delay. *Automatica*, 39(11):1877–1884, 2003.
- [41] D. Shukla and S. Thakur. Index based Internet traffic sharing analysis of users by a Markov chain probability model. *Karpagam J. Computer Science*, 4(3):1539–1545, 2010.
- [42] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1):625–653, 1999.
- [43] Y. Tipsuwan and M.Y. Chow. Gain scheduler middleware: a methodology to enable existing controllers for networked control and teleoperation-part I: networked control. *IEEE Trans. Ind. Electron.*, 51(6):1218–1227, Dec. 2004.
- [44] R. Vanijirattikhan, M-Y. Chow, P. Szemes, and H. Hashimoto. Mobile agent gain scheduler control in inter-continental Intelligent Space. In *Proc. Int. Conf. on Robotics and Automation (ICRA)*, pages 1115–1120. IEEE, 2005.
- [45] P.G. Voulgaris and B. Bamieh. Optimal H_∞ control of hybrid multirate systems. In *Proc. IEEE Conf. on Decision and Control*, pages 457–462. IEEE, 1992.

- [46] D. Wang, J. Wang, and W. Wang. Output feedback control of networked control systems with packet dropouts in both channels. *Information Sciences*, 221:544–554, 2013.
- [47] Y. Xia, W. Xie, B. Liu, and X. Wang. Data-driven predictive control for networked control systems. *Information Sciences*, 235:45–54, 2013.
- [48] T.C. Yang. Networked control system: a brief survey. *IEE Proc. Control Theory Appl.*, 153(4):403–412, Jul. 2006.
- [49] K. Yoshizawa, H. Hashimoto, M. Wada, and S. Mori. Path tracking control of mobile robots using a quadratic curve. In *Proc. IEEE Intelligent Vehicles Symposium*, pages 58–63. IEEE, 1996.
- [50] H. Zhang, Y. Shi, and J. Wang. On energy-to-peak filtering for nonuniformly sampled nonlinear systems: a markovian jump system approach. *IEEE Trans. Fuzzy Systems*, 22(1):212–222, 2014.
- [51] J. Zhang, J. Lam, and Y. Xia. Output feedback delay compensation control for networked control systems with random delays. *Information Sciences*, 265:154–166, 2014.
- [52] L. Zhang, H. Gao, and O. Kaynak. Network-induced constraints in networked control systems: a survey. *IEEE Trans. Ind. Informat.*, 9(1):403–416, 2013.
- [53] Y-B. Zhao, G-P. Liu, and D. Rees. Actively compensating for data packet disorder in networked control systems. *IEEE Trans. Circuits and Systems II: Express Briefs*, 57(11):913–917, 2010.