Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Networked Predictive Control of Systems with Data Dropouts

Pablo Millán, Isabel Jurado, Carlos Vivas and Francisco R. Rubio

Abstract— **This paper presents a practical algorithm to design networked control systems able to cope with high data dropout rates. The algorithm is intended for application in packet based networks protocols (Ethernet-like) where data packets typically content large data fields. The key concept is using such packets to transmit not only the current control signal, but predictions on a finite horizon without significantly increasing traffic load. Thus, predictive control is used together with buffered actuators and a state estimator to compensate for eventual packet dropouts. Additionally, some ideas are proposed to decrease traffic load, limiting packet size and media access frequency. Simulation results on the control of a three-tank system are given to illustrate the effectiveness of the method.**

I. INTRODUCTION

In recent years, an increasing number of control applications with control loops closed via a shared communication network have been described, see for example [1], [2].

In these control systems, known as Networked Control Systems (NCS), serial communication networks are used to exchange system information and control signals between various physical components of the systems that may be physically distributed. Major advantages of NCS include low cost, reduced weight and power requirements, simple installation and maintenance, and high reliability.

Nonetheless, closing a control loop on a shared communication network introduces additional dynamics and constraints in the control problem. Typically, time-varying delays, data losses and data quantization effects, may degrade system performance and even unstabilize the system [1].

In recent years, a significant body of knowledge has been developed regarding the analysis of stability and robustness properties of NCS [3], [4], [5]. Beyond these remarkable analytic results, effective compensation strategies have also been devised to cope mainly with time-varying delays and data losses.

The random nature of transmission delays, makes it difficult to analyze stability and performance of NCS. It has been shown that, as far as the maximum time the systems operates in open-loop does not exceed certain bounds, stability is preserved. See for instance [5] and the notion of *maximum allowable transfer interval* in [6].

Random delays are nonetheless intimately related with the problem of data losses in NCS. In practical network protocols, information is transmitted bundled in packets through a number of intermediate nodes that account for packets routing, collisions detections as well as information reconstruction at destination.

That is the case of Ethernet networks, in its wired and wireless versions, which are becoming increasingly popular to implement networked control loops. The architecture of Ethernet networks is designed for integrity preservation of information, but it is not very appropriate in general for real-time control loops. The stringent bounds imposed on time-delays from closed-loop stability requirements, make that in practice, those packets arriving later than a maximum delay threshold must be discarded. In these sense, error free protocols like Ethernet, turn unreliable from the control point of view, and packet dropouts must be taken into account.

The packet structure of Ethernet networks has other important implications from the control point of view. As it is well known, performance of digital control systems asymptotically approaches that of the continuos time system, as sampling period decreases. Nonetheless, in networked controlled systems this would increase network congestion, increasing delays and data dropouts. Thus, tradeoff is compulsory for practical networked control design.

As some authors have proposed, one way to overcome this is to send fewer but more informative packets [7], [8], [9], [10]. Thus, large data packets in ethernet networks can be used to compute and send predictions on future control signals, without significantly increasing network load. This signals, appropriately buffered and scheduled at the actuator end, become a safeguard in case of delays or eventual packet dropouts. This concept naturally fits the predictive control paradigm, and so has been reported in the literature [11],[12].

Inspired in these results, this paper proposes a predictive control scheme focussing on the sensor/actuator vs. controller information interchange policy. We are concerned with the design of a strategy for networked linear systems with disturbances, with large data dropouts, retaining good performance. Additionally, limiting the amount of information transmitted in a Networked control system is a major concern. In this paper, we explore the effect of reducing the number of data packet exchanges between the controller and the actuator, while keeping an error threshold for the actuator control signals. This threshold allows us to limit the amount of information through the network, transmitting only when relevant information for control is needed.

The network model considered allows for packet dropouts in both links, controller-to-actuator and sensors-to-controller. This motivates the inclusion of detection and compensation of missing packets resorting to buffering and state estimator respectively. To show the behaviour of the proposed compensation strategy, simulation results are provided on the level

The authors would like to acknowledge CICYT for funding this work under grant DPI2007-64697

P. Millán, I. Jurado, C. Vivas and F.R. Rubio are with Dpto. de Ingeniería de Sistemas y Automática. Escuela Superior de Ingenieros. University of Seville (SPAIN) pmillan, ijurado, vivas, rubio@cartuja.us.es

control problem of a three-tank system.

II. NETWORKED PREDICTIVE CONTROL ALGORITHM

A. Problem statement

In the present work, we focus on the design of a predictive control structure for a networked control system based on the above discussed paradigm.

Systems to be considered are unconstrained discrete-time linear multiple-inputs plants, under the effect of bounded disturbances as:

$$x(k+1) = Ax(k) + B_1 u(k) + B_2 \omega(k) \tag{1}$$

with $k \in \mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\}$ and

$$u(k) \in \mathbb{U} \subseteq \mathbb{R}^{p_1}, \quad x(k) \in \mathbb{X} \subseteq \mathbb{R}^{p_2}, \quad \forall k \in \mathbb{N}_0$$

disturbances, $\omega(k)$ are considered to be bounded as

$$\omega(k) \in \mathbb{W}, \quad \mathbb{W} = \{x \in \mathbb{R}^{p_2} / \|x\| < \delta\} \tag{2}$$

In our setup, the plant and controller are assumed to be linked through a communication network (see figure 1). Our interest lies in clock-driven Ethernet-like networks linking both, controller outputs to plant inputs, and plant outputs (sensors) to controller inputs. Data is sent in large packets, so that the relevant phenomena for control purposes are transmission delays and packet dropouts.

More precisely, only the problem of packet dropouts is addressed. Random delays are not a concern in this study, since small round-trip communication delays (the sum of delays from the sensor to the controller and from this to the actuator) are assumed, that is delays are considered negligible with respect to sample periods. Thus, in the event that data packets do not arrive, or arrive later than a certain threshold, they are considered as missing packets.

Our approach does not assume secured links in neither end of the communications chain. That is, packets can be dropped either in sensor to controller path, or in the controller to actuator one. This feature is particularly remarkable as usually dropouts are only considered in the controller to actuator path.

To this end, acknowledgment is assumed as part of the network protocol (TCP-like protocols), so that at any time instant k, the controller knows whether a control packet arrived at destination or not. Packets are also assumed to be time-stamped so they can be correctly sequenced at any point of the control loop.

To summarize, for the proposed control algorithm to work, all elements in the control loop are assumed to behave in a time-driven manner. Thus, the network model operates at the same sampling rate as the plant-controller model, with the following rules:

1) Time-driven sensors periodically sample plant outputs and states.
2) A time-driven predictive controller computes a control sequence at each sampling time.
3) A time-driven buffered actuator applies control signals at each sampling time.
4) Network is affected by dropouts at any point.
5) Delayed packets are taken as dropouts.

In order to achieve an appropriate performance level, this proposes the use of a finite horizon predictive optimal control framework.

The predictive controller has access to the plant states $x(k)$, and computes at every time instant k a finite horizon optimal control sequence $U_k \in (\mathbb{U})^{N_u}$ of length N_u, such that the following functional is minimized

$$V(U_k, k) = \sum_{i=k}^{k+N_u-1} \ell(x'(i), u'(i)) + F(x'(k+N_u)) \tag{3}$$

where $x'(\cdot)$ and $u'(\cdot)$ denote predicted plant states and outputs respectively. Also in (3), $\ell(\cdot)$ denotes the *stage cost* and $F(\cdot)$ is the *terminal cost*.

Assuming this setup, we will next show how this predictive control structure can be combined with an appropriate buffering and queuing strategy providing remarkable robustness to packet dropouts.

B. Packetized control and buffering strategy

In order to compensate for eventual packet dropouts and delays, one key feature of our proposed predictive control scheme is buffering control signals in the actuator side.

In this scheme, also exploited for instance in [7], [8], [13], the buffered signals act as a safeguard against packet dropouts. Thus, as depicts the proposed control structure in figure 1, the buffer stores a number of model-based predictions on future control actions, so the actuator can provide appropriate control in the event of dropouts.

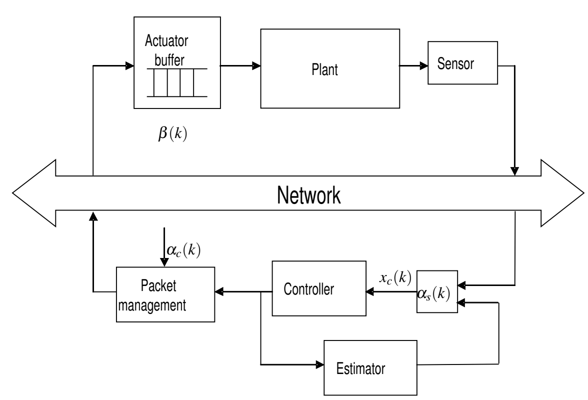

Fig. 1. NCS proposed scheme

The buffering policy is designed such that whenever new packets arrive, buffer is overwritten. The actuator is then sequentially feeded with the information in the buffer until new packets arrive. This corresponds to the intuitively appealing idea of *"Use the most recent control sequence if available. If not, use predictions from the buffer."*

The amount of consecutive dropouts this strategy can compensate for, is obviously equal to the buffer length. In this sense, the buffer can be reasonably dimensioned to store as many control actions as the prediction horizon N_u[1], and so is the maximum consecutive dropouts allowed by the proposed control structure.

This simple idea can be formalized as:

Let's represent the state of the buffer at a given time instant k as $\beta(k) \in (\mathbb{U})^{N_u}$. Then, the dynamics of the buffer can be expressed as the recursive rule

$$\beta(k) = \alpha_c(k)U_k + (1-\alpha_c(k))S\beta(k-1) \tag{4}$$

where matrix $S \in \mathbb{R}^{p_1 N_u \times p_1 N_u}$ is a shift matrix defined as the block matrix

$$S_{i,j} = \delta_{i+1,j} \cdot I_{p_1}; \quad i,j = 1,...,N_u$$

with $\delta_{i,j}$ the Kronecker delta symbol.

In (4), $\alpha_c(k) \in \{0,1\}$ is a signal accounting for reception acknowledgment in the controller to actuator link, such that

$$\alpha_c(k) = \begin{cases} 1 & \text{if packet } U_k \text{ arrives to buffer at time } k \\ 0 & \text{if packet } U_k \text{ does not arrive to buffer at time } k \end{cases}$$

With this description the control action $u(k)$ released from the buffer at instant k can be expressed as

$$u(k) = \begin{bmatrix} I_{p_1} & 0_{p_1} & \dots & 0_{p_1} \end{bmatrix} \beta(k)$$

This basic mechanism implicitly assumes that the controller computes and sends a whole control sequence of length N_u at every sampling time. This, as it has been discussed, does not significantly increase network load, as information is bundled in rather lengthy packets.

Nonetheless, specific situations in networked control systems suggest to reduce network access to its minimum. That is the case for instance of wireless sensor networks where typically energy saving is a major concern. In this kind of systems, it is advisable to design network protocols that avoid unnecessary network use, for example transmitting data packets of minimum length and only when relevant information for control is available.

In this sense, a further refinement can be introduced in our scheme in order to alleviate network load to a greater extent. The key idea here is comparing at every time instant k the control sequence in the buffer, $\beta(k)$, and the current controller sequence in the actuator, $U(k)$. This comparison is performed in the controller, so if both sequences match up to a certain degree, only the relevant changes are sent, or even no sequence might need to be sent at all.

Note that, as an acknowledgment signal $\alpha_c(k)$ is assumed part of the protocol, the controller has full access to the buffer state, $\beta(k)$, at every time instant k. That is, the buffer dynamics can be accurately reproduced at the controller side.

It is a natural assumption that the buffer's most distant predictions in the future, should be those differing to a greater extent with the more recently computed sequences in the controller. This intuitive idea suggests reducing packet size to form a *trimmed packet* by sending just the last few components that differ more than a certain threshold from the buffer state.

This *packet management* policy can be formalized as:

Consider $\beta(k)$ the buffer state, and $U(k)$ the computed control sequence at time instant k. Denote $\beta_j(k)$ and $U_j(k)$ as the j-th component of the corresponding sequences at time instant k.

The length of a trimmed packet N_T can be determined according to buffer at instant k as

$$N_T = N_u - \arg \min_{j \in \{1,\dots,N_u\}} \|U_j(k) - \beta_j(k)\| > \varepsilon$$

That is, only the last N_T components of sequence U_k need to be sent to the buffer, as the first $N_u - N_T$ match those in the buffer up to a certain tolerance ε.

With this definition, a trimmed packet of length $N_T \leq N_u$, $U^*(k) \in (\mathbb{R})^{N_T}$, can be built as

$$U^*(k) = \begin{bmatrix} 0_{p_1 N_T \times p_1 (N_u - N_T)} & I_{p_1 N_T} \end{bmatrix} U(k)$$

Thus, the buffer dynamics in (4) can be trivially modified to deal with trimmed packets $U^*(k)$ as

$$\begin{aligned} \beta(k) &= \alpha_c(k) \left[\beta_1^T(k-1), \dots, \beta_{N_u - N_T}^T(k-1), (U_1^*(k))^T, \dots, (U_{N_T}^*(k))^T \right]^T \\ &+ (1-\alpha_c(k))S\beta(k-1) \end{aligned} \tag{5}$$

The proposed networked control structure in this work also considers the possibility of missing data packets in the sensor to actuator path. This issue, not treated in most previous works, is specially relevant to take into account realistic networked control problems, as plant and controller are usually physically distributed.

To deal with eventual missing state measures, this work resorts to a model-based estimator that approximates plant states when no updated information from the sensors is received. The estimator takes the form

$$\hat{x}(k+1) = \alpha_s(k)x(k) + (1-\alpha_s(k))f(\hat{x}(k),u(k)) \tag{6}$$

where $\hat{x}(k) \in \mathbf{R}^n$ is the estimated plant state at instant k, and $f(\hat{x}(k),u(k))$ is an open-loop approximation of the plant dynamics. Considering the plant model (1), $f(\hat{x}(k),u(k)) = A\hat{x}(k) + B_1 u(k)$ can be taken.

As in equation (4), the estimator (6) makes use of a signal $\alpha_s(k)$ accounting in this case for the acknowledgment of reception of packets sent in the link from sensors to actuator. In a similar fashion

$$\alpha_s(k) = \begin{cases} 1 & \text{if packet } U_k \text{ arrives to actuator at time } k \\ 0 & \text{if packet } U_k \text{ does not arrive to actuator at time } k \end{cases}$$

As can be easily interpreted from the estimator equation (6), the estimated state $\hat{x}(k)$ is updated with the measured state $x(k)$ when data packet from the sensor arrives, otherwise the plant state is estimated from the plant model.

It is worth to mention that signal $\alpha_s(k)$ is not directly provided by the network protocol, as packet reception is acknowledged at the controller side. Nonetheless, $\alpha_s(k)$ can be synthesized from the packet time stamps arriving from the sensor. Since clock-driven networking protocol is assumed, a simple procedure consists in checking the arrival time of every packet, so that only those arriving within the current

[1]Note that a larger buffer size is useless as the buffer receives at most N_u control predictions. There is also little point in using a smaller buffer since the last few predictions of every received sequence would be lost.

sample period, are considered as valid states measures, otherwise dropout is assumed.

The addition of the estimator in the control scheme allows the controller to be feeded with the plant states at each sampling time, regardless of packet dropouts. This input to the controller can be measured or estimated depending on the arrival of the most recent sensor packet.

This estimation procedure at the controller together with the packet management policy above discussed, constitute the basic predictive networked control scheme proposed in this work. It is worth to remember that the network predictive controller is not required to satisfy network constraints of any kind, as it is designed regardless of the underlying network structure. From this point of view, the proposed methodology can be regarded as network compensation technique rather than a control methodology by itself.

Remark: (Stability)
Stability of the proposed compensation methodology can be ensured as far as a number of mild requirements are satisfied.

Let $u(k)$ be the stabilizing predictive control action for system (1) computed at time instant k without network. As it has been discussed, packet dropouts associated to the inclusion of a network in the control structure, implies that there is no guaranty that signal $u(k)$ is accurately applied at every time instant k. Instead, the presented methodology computes a compensated control, $u_c(k)$, based on buffered predictions and state estimations.

From this point of view, the inclusion of the network, together with the proposed compensation scheme, amounts to introducing an additional disturbance term on system (1), as the following decomposition suggest

$$\begin{aligned} x(k+1) &= Ax(k)+B_1u(k)+B_1(u_c(k)-u(k))+B_2\omega(k) \\ &= Ax(k)+B_1u(k)+B_1\omega_u(k)+B_2\omega(k) \end{aligned} \tag{7}$$

where $\omega_u(k) = u_c(k)-u(k)$ represents the network effect on the predictive control structure.

Moreover, it can be checked that this additional term $\omega_u(k)$ is bounded. Notice that, as new packets arrive, the buffer and estimator are reset to match the computed sequence, hence $\omega_u(k) = 0$. As by assumption the number of consecutive network dropouts is limited, the difference between the compensated control, $u_c(k)$, and the computed control, $u(k)$, can only grow between valid packets, thus it is bounded.

Also by assumption the term $\omega(k)$ is bounded, thus the overall disturbance term $\Omega(k) = B_1\omega_u(k)+B_2\omega(k)$ is also bounded.

As discussed in [14] a stabilizing predictive controller can always be found under appropriate conditions for the unperturbed system (7).

Recalling results in [15]:

(A1) Let $x(k+1) = F(x(k))$ be the closed loop dynamics of the unperturbed system (7), with the origin being a fixed point.

(A2) Let $V(x)$ a Lyapunov function of the system Lipschitz in a neighborhood of the origin $\Lambda_r = \{x \in \mathbb{R}^n / V(x) \leq r\}$ such that

$$\begin{aligned} a\cdot\|x\|^p \leq V(x) \leq b\cdot\|x\|^p \\ V(F(x))-V(x) \leq -c\cdot\|x\|^p \end{aligned} \tag{8}$$

where a, b, c are positive constants and $p > 1$.

Then there exits a constant $\mu > 0$ such that for all disturbances $\Omega(k) \in B_\mu = \{\Omega(k) \in \mathbb{R}^n / \|\Omega(k)\| < \mu\}$ the perturbed system $x(k+1) = F(x(k))+\Omega(k)$ is asymptotically ultimately bounded $\forall x(0) \in \Lambda_r$.

To conclude stability of the proposed control methodology, notice that conditions (A1) and (A2) are satisfied for system (7) taking $p = 2$, and considering a Lyapunov function of the form $V(x) = x^TPx$, which is trivially Lipschitz in a neighborhood of the origin Λ_r for arbitrarily large values of r.

III. PROCESS MODELING

The proposed algorithm has been tested on a level control model as depicted in figure 2. The system is composed of three tanks, with the control problem consisting in tracking a reference level in the last one, acting on the flow poured in the first one. The model of the process can be easily obtained from a mass balance as:

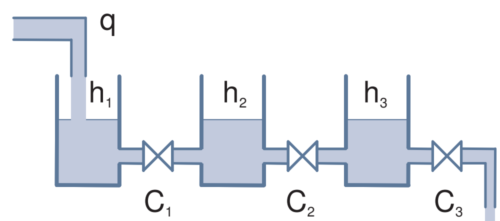

Fig. 2. Three tank system

$$\begin{aligned} \frac{dh_1}{dt} &= \frac{1}{S}\,q-\frac{1}{S}\,C_1\sqrt{h_1-h_2} \\ \frac{dh_2}{dt} &= \frac{1}{S}\,C_1\sqrt{h_1-h_2}-\frac{1}{S}\,C_2\sqrt{h_2-h_3} \\ \frac{dh_3}{dt} &= \frac{1}{S}\,C_2\sqrt{h_2-h_3}-\frac{1}{S}\,C_3\sqrt{h_3} \end{aligned} \tag{9}$$

where h_i represent the level of tank i.

The system is linearized to apply the proposed control structure around a trimming point H_1, H_2, H_3 and Q. Thus we have:

$$\begin{aligned} h_1 = H_1+\Delta H_1, \quad h_2 = H_2+\Delta H_2 \\ h_3 = H_3+\Delta H_3, \quad q = Q+\Delta Q \end{aligned}$$

yielding the linear equation:

$$\Delta\dot{H} = L\Delta H+M\Delta Q \tag{10}$$

where:

$$\Delta H = \begin{bmatrix} \Delta H_1 & \Delta H_2 & \Delta H_3 \end{bmatrix}^T$$

$$L = \begin{bmatrix} \frac{-C_1}{2S\sqrt{H_1-H_2}} & \frac{C_1}{2S\sqrt{H_1-H_2}} & 0 \\ \frac{C_1}{2S\sqrt{H_1-H_2}} & \frac{-C_1}{2S\sqrt{H_1-H_2}} - \frac{C_2}{2S\sqrt{H_2-H_3}} & \frac{C_2}{2S\sqrt{H_2-H_3}} \\ 0 & \frac{C_2}{2S\sqrt{H_2-H_3}} & \frac{-C_2}{2S\sqrt{H_2-H_3}} - \frac{C_3}{2S\sqrt{H_3}} \end{bmatrix}$$

$$M = \begin{bmatrix} \frac{1}{S} & 0 & 0 \end{bmatrix}^T$$

A discrete model is then easily obtained from (10) as

$$x(k+1) = Ax(k) + B_1 u(k) \tag{11}$$

IV. APPLICATION TO A THREE-TANK SYSTEM

A number of simulations for different network operational conditions have been performed, taking as system parameters $S = 0.16m^2$, $C_1 = C_2 = 0.0256\,\frac{m^3}{sm^{1/2}}$ and $C_3 = 0.0251\,\frac{m^3}{sm^{1/2}}$, with an operation point $H_1 = 1m$, $H_2 = 0.7m$, $H_3 = 0.4m$, $Q = 0.014m^3/h$.

As an standard tool to compare performance results, the *integral square error* (ISE) measure has been employed.

First, the proposed strategy for reducing network traffic is compared with the conventional case where the entire control sequence is sent over the network at each sampling time. The influence of data dropout rate p, and allowed error ε is shown in figure 3. This plot represents the average ISE performance index for a number of experiments taking a step-like sequence with period $T = 2000s$ as reference. and a simulation time of 5000 s.

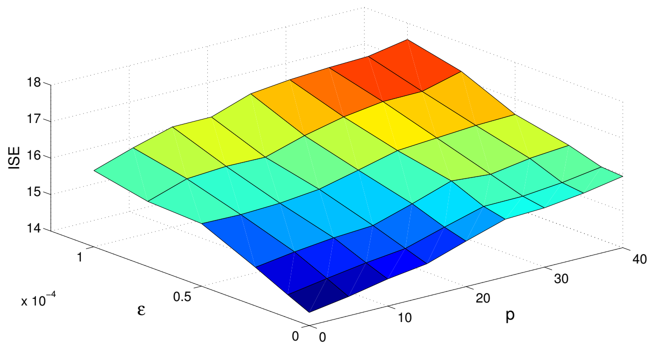

Fig. 3. Influence of allowed error

In figure 4 the percentage reduction of controller-to-actuator transmissions is shown. This reduction is computed as the amount of information transmitted with the proposed queueing/buffering scheme with respect to the full information transmission case. It can be observed that savings above 85% can be obtained for sufficiently high allowed error ε. Nonetheless, from figure 3, it is clear that there is little point in taking excessively high values of ε, as ISE performance starts degrading faster than transmission saving. For instance, in view of figure 3 and figure 4, by selecting $\varepsilon = 0.2 \cdot 10^{-4}$ a reduction of 70% is reached without worsening significantly the system response.

Figures 5 and 6 show the system response with a remarkable 30% packet dropout probability. It can be observed that

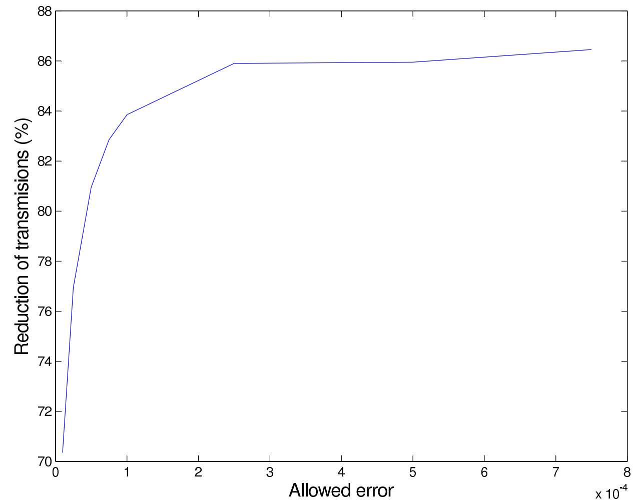

Fig. 4. Reduction of transmitted data

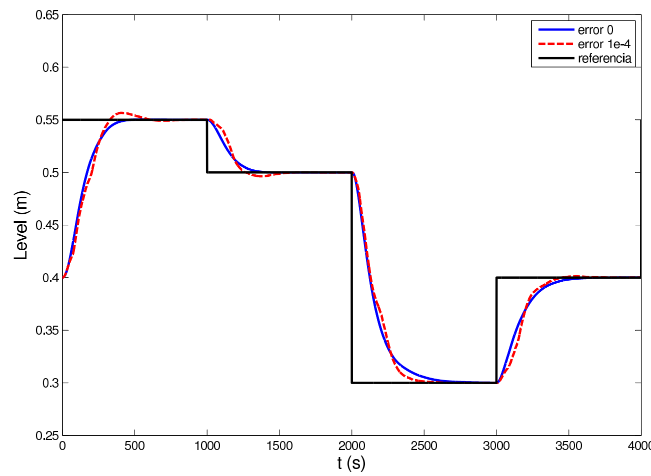

Fig. 5. Step Response

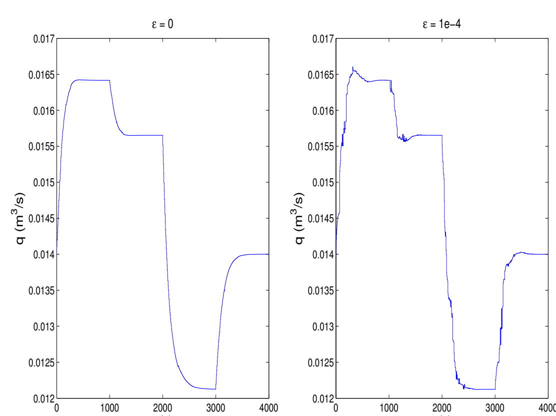

Fig. 6. Tracking of references

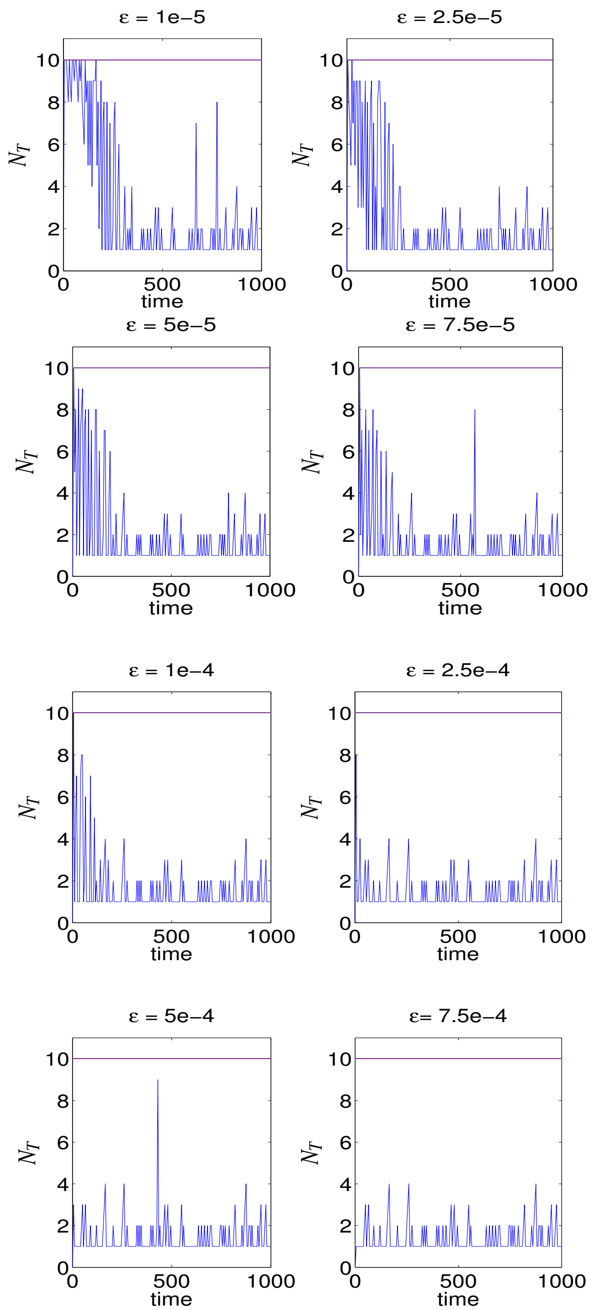

Fig. 7. Number of components of $U^*(k)$ sent

the algorithm retains good performance even when with high dropout probability. Nonetheless, in some cases, depending on the random dropouts, the response may exhibit small overshoot.

Not surprisingly, performance degrades as either the allowed error ε, or the data dropout rate p, increase. Remarkably, the controller can cope with data dropout rates above 40%.

In figure 7, the transmission profile for a step tracking experiment with different values of ε are shown. It can be observed that, as expected, an intense transmission pattern is observed for the first instants of simulation, corresponding to the transient regime. As the system approaches steady state, traffic load is drastically reduced.

V. CONCLUSIONS

This work has presented a methodology to compensate for data dropouts and delays in networked control systems. The methodology takes advantage of the intrinsic computation of future control signals in predictive control, to cope with eventual data dropouts. A key aspect is the inclusion of a buffering strategy together with a model based plant estimator that, under certain conditions, ensure stability of the controlled system.

Simulation results show that remarkable data dropout rates up to 40% can be achieved without significant performance degradation, as well as traffic load alleviation up to 85% with respect to conventional buffered predictive control systems.

REFERENCES

[1] W. Zhang, M. S. Branicky, S. M. Phillips, ”Stability of Networked Control Systems”, *in IEEE Control Systems Magazine*, 2001.

[2] J.P. Hespanha, P. Naghshtabrizi, X. Yonggang, ”A Survey of Recent Results in Networked Control Systems”,*Proceedings of the IEEE*, vol 95, n. 1, pp.138-162, 2007

[3] Tatikonda, S. ”Control Under Communication Constraints”. *PhD thesis, Massachusetts Institute of Technology*, 2000

[4] D. Liberzon, ”On stabilization of linear systems with limited information”. *IEEE Transaction on Automatic Control*, vol 48(2), pp 304-307, 2003.

[5] J. Hespanha, A. Ortega, L. Vasudevan,”Towards the control of linear systems with minimum bit-rate”, *Proc. of the Int. Symp. on the Mathematical Theory of*, 2002

[6] G.C. Walsh, H. Ye, and L. Bushnell, ”Stability analysis of networked control systems”, in *Proc. Amer. Control Conf.*, pp. 2876-2880, San Diego, June 1999.

[7] Yang, Can and Zhu, Shan-An and Kong, Wan-Zeng and Lu, Li-Ming, ”Application of generalized predictive control in networked control system”, *in Journal of Zhejiang University: Science*, 2006, vol. 7(2), pp. 225-233.

[8] H. Wenshan, G.P. Liu, D. Rees, ”Networked predictive control system with data compression”,*IEEE International Conference on Networking, Sensing and Control, ICNSC'07*, pp. 52 - 57, 2007

[9] Rehbinder H., Sanfridson M., ”Scheduling of a limited communication channel for optimal control”. *Proceedings of the 39th IEEE conference on decision and control*, 2000, pp. 10111016, Vol. 1.

[10] G. Walsh and Y. Hong, ”Scheduling of networked control systems”, *IEEE Control Systems Magazine*, 2001, v. 21 , pp. 5765

[11] A. Casavola, E. Mosca, and M. Papini. ”Predictive teleoperation of constrained dynamic systems via internetlike channels”. *IEEE Trans. Contr. Syst. Technol.*, 2006, 14(4), pp. 681694

[12] D. E. Quevedo, E. I. Silva, and G. C. Goodwin. ”Packetized predictive control over erasure channels”. *In Proc. Amer. Contr. Conf.*,2007, New York N.Y.

[13] D. Muñoz, C. Panagiotis D., ”Estimation-Based Networked Predictive Control of Nonlinear Systems”, *in Dynamics of Continuous Discrete and Impulsive Systems*, 2007, vol. 14(1), pp. 52-58.

[14] D.Q. Mayne, J.B. Rawlings, C.V. Rao. ”Constrained model predictive control: Stability and optimality” , *Automatica*, 2000, v 36, n 6, p 789-814

[15] D. Limon, T. Alamo, E.F. Camacho, ”Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of MPC” , *Proc. American Control Conference*, 2002.