Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Design of Multiple Actuator-Link Control Systems with Packet Dropouts

Daniel E. Quevedo * **Eduardo I. Silva** * **Dragan Nešić** **

* *School of Electrical Engineering & Computer Science, The University of Newcastle, NSW 2308, Australia (e-mails: dquevedo@ieee.org, eduardo.silva@studentmail.newcastle.edu.au).*
** *Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC 3052 (e-mail: d.nesic@ee.unimelb.edu.au)*

Abstract: This paper presents a novel design strategy for networked control systems, where a centralized controller needs to divide its attention between various actuators. Communication is via an unreliable network affected by data-dropouts and which allows access to only one actuator node at a time. To achieve good performance, control and network protocol are co-designed and signal predictions are sent to buffered actuator nodes. By using methods from predictive control theory, we show how closed loop stability in the presence of data-dropouts can be ensured.

Keywords: Control over networks, predictive control, stabilizing controllers.

1. INTRODUCTION

In Networked Control Systems (NCS's), digital networks are used to transmit sensor and/or control signals. The use of general purpose network technology has numerous advantages, including low cost, high reliability, interoperability of devices and easy installation and maintenance.

Many interesting design challenges are associated with NCS's. For example, data needs to be quantized and coded prior transmission, see, e.g., Nair et al. (2007); transmission errors and time delays are often unavoidable, see, e.g., Schenato et al. (2007); if the communication medium is shared, then network protocol design (scheduling) plays a mayor role, see, e.g., Nešić and Teel (2004). Clearly, the communication technology used may constitute a bottleneck in the achievable control performance and, thus, should explicitly be taken into account when designing NCS's.

Perhaps unsurprisingly, Ethernet in its wired (hub-based and switched) and wireless forms (IEEE 802.11) is increasingly being adopted for low level control technology, see also Moyne and Tilbury (2007). A key feature of Ethernet is that the packet-structure contains large data fields, typically of the order of kilobytes. Thus, bit-rates are high and quantization issues will often play a minor role. However, time delays and packet dropouts are likely to occur.

As shown by Georgiev and Tilbury (2006), the packet structure of Ethernet networks can be used to send signal predictions, rather than only values to be implemented at the current time instant, without increasing the network load. Through buffering at the receiving end, these *packetized control* schemes will often give good performance in the presence of time delays and packet dropouts, see also Casavola et al. (2006); Tang and de Silva (2006); Liu et al. (2005); Quevedo et al. (2007).

In the present work, we focus on an NCS architecture where a centralized controller communicates with a set of actuator nodes through an Ethernet-like network, which is capable of carrying fairly large data packets. The network is such that only one packet can be sent at a time and each packet can be addressed to only one node. The network protocol, which regulates the medium access, allows for dynamic scheduling. Within this setup, we propose a network protocol and control co-design method which focuses on closed loop performance. For that purpose, at each time instant, it is decided, *which* actuator node to address and *what* to send. Whilst this type of communication constraints have also been studied, for example, in Gaid et al. (2006); Goodwin et al. (2004), a distinctive feature of our approach lies in how packet dropouts are handled. Here, the controller extends the single actuator-link *packetized control* methods referred to above and sends signal predictions to the (buffered) actuator nodes.

We will consider general constrained non-linear plant models, where we assume that the controller has direct access to the plant states. By adopting ideas from non-networked predictive control, see, e.g., Mayne et al. (2000); Goodwin et al. (2005), we establish sufficient conditions on tuning parameters for closed loop stability in the presence of packet dropouts.

Notation and basic definitions We will denote by I_m the $m \times m$ identity matrix, the zero element of $\mathbb{R}^{m\times 1}$ via $\bar{0}_m$, and define $0_m \triangleq 0 \cdot I_m$. The notation $\|v\|$ refers to the Euclidean norm $\sqrt{v^T v}$, where v is any vector. For given $N \in \mathbb{N}$, we define the *m-shift matrix* S_m via:

$$S_m \triangleq \begin{bmatrix} 0_m & I_m & 0_m & \dots & 0_m \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0_m & \dots & 0_m & I_m & 0_m \\ 0_m & \dots & \dots & 0_m & I_m \\ 0_m & \dots & \dots & \dots & 0_m \end{bmatrix} \in \mathbb{R}^{mN\times mN}. \quad (1)$$

2. MULTIPLE ACTUATOR-LINK NCS ARCHITECTURE

As foreshadowed in the introduction, we consider constrained discrete-time nonlinear multiple-input plant models with state $x(k)$ and input vector $u(k)$

$$x(k+1) = f(x(k), u(k)), \quad k \in \mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\}, \tag{2}$$

where

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

and $p \geq 1$, and $n \geq 1$.

The input vector is connected to a network via $R \leq p$ nodes according to the partition

$$u(k) = \begin{bmatrix} u_1(k) \\ \vdots \\ u_R(k) \end{bmatrix}. \tag{3}$$

In (3), each input $\{u_r(k)\}_{r \in \{1,2,\ldots,R\}}$ satisfies

$$u_r(k) \in \mathbb{U}_r \subseteq \mathbb{R}^{p_r}, \quad \forall k \in \mathbb{N}_0 \tag{4}$$

for some set $\{p_r : r = 1, 2, \ldots, R\}$ such that $\sum_{r=1}^{R} p_r = p$. Note that our model may refer to a single plant or also to a set of geographically separate plants.

Our interest lies in clock-driven Ethernet-like communication networks situated between controller output and plant inputs. Data is sent in large time-stamped packets. Thus, quantization is less of an issue. Nevertheless, time-delays and packet dropouts are likely to occur. We will here concentrate on the latter and, thus, model the network as a collection of R erasure channels, [1] see also Imer et al. (2006); Wu and Chen (2007). Only one of these links can be accessed at a time.

To summarize, the network model operates at the same sampling rate as the plant model (2) and has three defining properties:

(1) At any time instant, data can be transmitted to only one of the R actuator nodes (*scheduling constraint*).
(2) Transmission is affected by dropouts.
(3) Data-packets are sufficiently large to contain a sequence of control input values.

We will consider the situation where the network protocol is not pre-defined. Thus, at each time instant, one needs to decide *which* actuator-link to use and *what* data to send. In addition, the resulting NCS should be robust in the face of packet dropouts. In the following section we will present such a network protocol and control co-design strategy.

3. PACKETIZED CONTROL AND SCHEDULING

In our proposal, at each time instant, the controller sends a control packet to one of the R input nodes. The data sent contains possible control inputs for a finite number of N future time instants. To achieve good performance, despite the scheduling constraint and possible packet dropouts, buffers are located at the plant input side, see Fig. 1. We will first describe the buffering procedure and then propose a joint control and scheduling (network protocol) design method.

[1] Note that small time-delays up to a fixed threshold can be incorporated in the plant model (2). Signals, which are delayed more, are then considered as "lost."

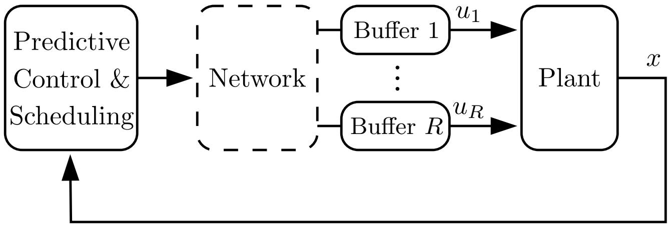

Fig. 1. Multiple Actuator-Link NCS with Packetized Control and Scheduling.

3.1 Plant Input Buffering

Each of the R plant input nodes is connected to a buffer, whose state is overwritten whenever a valid control packet arrives. Actuator values are then passed on to the plant sequentially until the next valid control packet arrives.

To be more precise, suppose that at time k, the controller sends the data packet $\boldsymbol{\mu}^\star(k) \in (\mathbb{U}_{r^\star})^N$ to node $r^\star$. The state of the $r^\star$-th buffer, say $\boldsymbol{b}_{r^\star}$, is then given by:

$$\boldsymbol{b}_{r^\star}(k) = \begin{cases} \boldsymbol{\mu}^\star(k) & \text{if } \boldsymbol{\mu}^\star(k) \text{ arrives at instant } k, \\ S_{p_{r^\star}} \boldsymbol{b}_{r^\star}(k-1) & \text{if } \boldsymbol{\mu}^\star(k) \text{ does not arrive at } k, \end{cases}$$

where $S_{p_{r^\star}}$ denotes the $p_{r^\star}$*-shift matrix*, see (1).

The buffer states at all other input nodes are updated via:

$$\boldsymbol{b}_r(k) = S_{p_r} \boldsymbol{b}_r(k-1), \quad \forall r \in \{1, 2, \ldots, R\},\ r \neq r^\star.$$

At all times, plant inputs stem from buffer states following:

$$u_r(k) = [I_{p_r}\ 0_{p_r}\ \ldots\ 0_{p_r}]\, \boldsymbol{b}_r(k), \quad \forall r \in \{1, 2, \ldots, R\}. \tag{5}$$

As in, for example, Quevedo et al. (2007), the buffering mechanisms amount to *parallel-in serial-out shift registers.* They act as safeguards against channel dropouts and also offer advantages in the presence of the scheduling constraints considered here. [2]

3.2 Predictive Control and Scheduling

We will next show how to choose actuator nodes $r^\star(k)$ and how to design the control sequences $\boldsymbol{\mu}^\star(k)$. For that purpose, we will concentrate on situations, where acknowledgments form part of the network protocol, such as TCP-like ones. Consequently, at time instant k the controller knows the previous buffer states, namely $\boldsymbol{b}_r(k-1)$, $\forall r \in \{1, 2, \ldots, R\}$. In addition, we will assume that the controller has exact knowledge of the plant state $x(k)$.

To achieve good performance in the presence of the scheduling constraints, we propose to use a finite horizon predictive optimal control framework, where at each time instant k, the following cost function is minimized:

$$V(\boldsymbol{\mu}', r'; k) \triangleq F(x'(k+N)) + \sum_{\ell=k}^{k+N-1} L(x'(\ell), u'(\ell)). \tag{6}$$

In (6), $\{x'(\cdot)\}$ and $\{u'(\cdot)\}$ denote predicted plant states and inputs, respectively. These obey the plant model, the buffering procedure and are penalized via the per-stage

[2] The choice in (1) corresponds to setting the buffer state to zero if no data is received over N consecutive instants at the node. Alternatively, if one wished to hold the latest value, one could set the "last" element of $S_{p_{r^\star}}$ equal to $I_{p_{r^\star}}$.

weighting function $L(\cdot,\cdot)$ and the terminal weighting $F(\cdot)$. (More details on how to choose the weighting functions are included in Section 4.) The decision variables r' and $\boldsymbol{\mu}'$ refer to the actuator node addressed at time k and to the associated control sequence.

More precisely, and in agreement with the buffering mechanisms described in Section 3.1, every chosen pair $(\boldsymbol{\mu}', r')$ yields predicted buffer states [3]

$$\boldsymbol{b}'_r(\ell+1) = S_{p_r}\boldsymbol{b}'_r(\ell), \quad \forall r,\ \forall \ell \in \{k,\dots,k+N-2\}$$

with initial condition:

$$\boldsymbol{b}'_r(k) = \begin{cases} \boldsymbol{\mu}' & \text{for } r = r', \\ S_{p_r}\boldsymbol{b}'_r(k-1) & \text{for all } r \neq r',\ r \in \{1,2,\dots,R\}. \end{cases}$$

The plant input and state predictions are then formed via

$$x'(\ell+1) = f(x'(\ell), u'(\ell)), \quad x'(k) = x(k),$$

where

$$u'(\ell) = \begin{bmatrix} u'_1(\ell) \\ \vdots \\ u'_R(\ell) \end{bmatrix}, \quad \ell \in \{k, k+1, \dots, k+N-1\},$$

with $u'_r(\ell) = [I_{p_r}\ 0_{p_r}\ \dots\ 0_{p_r}]\,\boldsymbol{b}'_r(\ell)$, $\forall r \in \{1,2,\dots,R\}$.

At each time instant k, the optimizer

$$(\boldsymbol{\mu}^\star(k), r^\star(k)) \triangleq \arg \min_{\substack{\boldsymbol{\mu}' \in (\mathbb{U}_{r'})^N \\ r' \in \{1,2,\dots,R\}}} V(\boldsymbol{\mu}', r'; k) \tag{7}$$

is determined and the sequence

$$\boldsymbol{\mu}^\star(k) = \begin{bmatrix} \mu^\star(k;k) \\ \vdots \\ \mu^\star(k+N-1;k) \end{bmatrix} \tag{8}$$

is sent through the network to the actuator node $r^\star(k)$. By construction, $\boldsymbol{\mu}^\star(k)$ contains possible values for

$$\{u_{r^\star(k)}(k), u_{r^\star(k)}(k+1), \dots, u_{r^\star(k)}(k+N-1)\}.$$

Following the buffering procedure, these values are implemented sequentially until some future (valid) control packet arrives at node $r^\star(k)$.

In the proposed method, scheduling and plant input design are done dynamically such as to optimize performance. Here it is important to note that whilst $\boldsymbol{\mu}^\star(k)$ and $r^\star(k)$ are found by evaluating open-loop predictions, the resultant joint scheduling and control policy is a closed loop one. Indeed, the loop is effectively closed at all successful transmission instants, i.e., whenever no dropouts occur. Since between successful transmission instants the plant operates in open loop, for the resultant NCS to exhibit good performance, not too many consecutive packet dropouts should occur.

For further reference, we will define the optimal value function at time k via:

$$V^\star(k) \triangleq V(\boldsymbol{\mu}^\star(k), r^\star(k); k), \quad k \in \mathbb{N}_0 \tag{9}$$

and denote the associated overall plant input and state predictions via:

$$\begin{aligned} \boldsymbol{u}(k) &= \{u(k;k), u(k+1;k), \dots, u(k+N-1;k)\}, \\ \boldsymbol{x}(k) &= \{x(k+1;k), x(k+2;k), \dots, x(k+N;k)\}. \end{aligned} \tag{10}$$

Note that the sequence $\boldsymbol{u}(k)$ is formed from $\boldsymbol{\mu}^\star(k)$ and from some of the buffer states at time $k-1$, which in turn originate from past optimizations.

From a computational viewpoint, it is worth emphasizing that $(\boldsymbol{\mu}^\star(k), r^\star(k))$ in (7) can be found by solving R smaller optimization problems, since

$$V^\star(k) = \min_{r' \in \{1,2,\dots,R\}} \left\{ \min_{\boldsymbol{\mu}' \in (\mathbb{U}_{r'})^N} V(\boldsymbol{\mu}', r'; k) \right\}. \tag{11}$$

Remark 1. The method presented is not equivalent to that which would be obtained if optimizing control sequences designed as in Goodwin et al. (2004); Gaid et al. (2006) were transmitted. [4] These sequences contain information for various actuator nodes and are, thus, unsuitable in the present NCS architecture. In contrast, the sequence $\boldsymbol{\mu}^\star(k)$ in (8) is sent through only one link, namely $r^\star(k)$. $\triangle$

Our control and network protocol co-design strategy is not only intuitively appealing but has also good stabilizing properties. Indeed, in the following section we will show how the prediction horizon N and the weighting functions $L(\cdot,\cdot)$ and $F(\cdot)$ in (6) can be utilized to guarantee stability of the closed loop in the presence of packet dropouts. Before doing so we will give a short illustrative example to clarify the scheduling aspect of the resultant NCS.

Example 2. (Scheduling and packet dropouts). Suppose we wish to control a plant having $R = 4$ inputs with horizon $N = 8$ and that the buffer states at time $\ell = 1$ are:

$$\boldsymbol{b}_r(1) = \begin{bmatrix} b_{r1} \\ \vdots \\ b_{r8} \end{bmatrix}, \quad r \in \{1,2,3,4\}.$$

If the on-line optimization yields the schedule

$$\{r^\star(1), r^\star(2), \dots, r^\star(10)\} = \{2,1,4,3,1,3,4,2,1,3\}$$

and if packet dropouts occur at $\ell = 6$ and $\ell = 9$, then the plant inputs at $\ell = 1,2,\dots,10$ are characterized via:

$$\begin{aligned} \{u(1), u(2), \dots, u(10)\} = \{&u(1;1), u(2;2), u(3;3), u(4;4), \\ &u(5;5), u(6;5), u(7;7), u(8;8), u(9;8), u(10;10)\} \end{aligned}$$

$$= \left\{ \begin{bmatrix} b_{11} \\ \mu^\star(1;1) \\ b_{31} \\ b_{41} \end{bmatrix}, \begin{bmatrix} \mu^\star(2;2) \\ \mu^\star(2;1) \\ b_{32} \\ b_{42} \end{bmatrix}, \begin{bmatrix} \mu^\star(3;2) \\ \mu^\star(3;1) \\ b_{33} \\ \mu^\star(3;3) \end{bmatrix}, \right.$$

$$\begin{bmatrix} \mu^\star(4;2) \\ \mu^\star(4;1) \\ \mu^\star(4;4) \\ \mu^\star(4;3) \end{bmatrix}, \begin{bmatrix} \mu^\star(5;5) \\ \mu^\star(5;1) \\ \mu^\star(5;4) \\ \mu^\star(5;3) \end{bmatrix}, \begin{bmatrix} \mu^\star(6;5) \\ \mu^\star(6;1) \\ \mu^\star(6;4) \\ \mu^\star(6;3) \end{bmatrix}, \begin{bmatrix} \mu^\star(7;5) \\ \mu^\star(7;1) \\ \mu^\star(7;4) \\ \mu^\star(7;7) \end{bmatrix},$$

$$\left. \begin{bmatrix} \mu^\star(8;5) \\ \mu^\star(8;8) \\ \mu^\star(8;4) \\ \mu^\star(8;7) \end{bmatrix}, \begin{bmatrix} \mu^\star(9;5) \\ \mu^\star(9;8) \\ \mu^\star(9;4) \\ \mu^\star(9;7) \end{bmatrix}, \begin{bmatrix} \mu^\star(10;5) \\ \mu^\star(10;8) \\ \mu^\star(10;10) \\ \mu^\star(10;7) \end{bmatrix} \right\}$$

4. STABILITY WITH PACKET DROPOUTS

We will next show how closed loop stability of the NCS can be ensured even in the presence of packet dropouts. As in many approaches to establish stability of receding horizon control, see, e.g., Mayne et al. (2000); Goodwin et al. (2005), we will introduce additional ingredients in the on-line optimization.

[3] Note that in our approach, delayed packets are not used further, see also Footnote 1. Accordingly, the buffer state prediction model only considers $\boldsymbol{\mu}'$.

[4] Note that these works do not use buffering at receiving nodes.

Firstly, we will impose that the predicted plant state at time $k+N$ satisfies the *terminal state constraint*: [5]

$$x'(k+N) \in \mathbb{X}_f \subseteq \mathbb{X}, \tag{12}$$

where $\mathbb{X}_f$ is a given set containing the origin.

Secondly, we will assume that in the cost function of (6), the weighting functions $F(\cdot)$ and $L(\cdot,\cdot)$ satisfy:

$$F(x) \geq 0, \quad \forall x \in \mathbb{X}_f, \qquad F(0) = 0, \tag{13}$$

$$L(0,0) = 0,\ L(x,u) \geq \alpha(\|x\|),\ \forall x \in \mathbb{X}_N,\ \forall u \in \mathbb{U}, \tag{14}$$

where $\alpha(\cdot)\colon [0,\infty) \to [0,\infty)$ is a continuous, nondecreasing, unbounded function such that $\alpha(0) = 0$ and $\alpha(\rho) > 0$, for all $\rho > 0$. In (14),

$$\mathbb{X}_N \subseteq \mathbb{X}$$

denotes the set of all *feasible* initial states, i.e., states $x(k)$ such that there exists a pair $(\boldsymbol{\mu}', r)$ which is compatible with the optimization (7) and satisfies the constraint (12).

To establish our result, it is convenient to denote the time instants where no packet dropouts occur via $\{k_i\}_{i\in\mathbb{N}}$ and to define: [6]

$$m_i \triangleq k_{i+1} - k_i, \quad i \in \mathbb{N}.$$

We furthermore introduce the sets:

$$\mathbb{V}_r \triangleq \bar{0}_{p_1} \times \cdots \times \bar{0}_{p_{r-1}} \times \mathbb{U}_r \times \bar{0}_{p_{r+1}} \times \ldots \bar{0}_{p_R},\ r \in \{1,\ldots,R\}$$

These are clearly related to the scheduling constraint. Nevertheless, it should be noted that, due to the buffering procedure, we will often have $u(k) \notin \mathbb{V}_r$, $\forall r \in \{1,2,\ldots,R\}$.

We can now state sufficient conditions for convergence of plant state trajectories in the presence of packet dropouts:

Theorem 3. (Stability). Consider the plant model (2), controlled via the packetized controller and network protocol described in Section 3.2, subject to (12)–(14). Suppose that the horizon N in (6) is chosen such that

$$N \geq m_i, \quad \forall i \in \mathbb{N} \tag{15}$$

and that there exist R *control policies*

$$\kappa_r\colon \mathbb{X}_f \to \mathbb{V}_r, \quad r \in \{1,2,\ldots,R\} \tag{16}$$

such that $\forall \xi \in \mathbb{X}_f$ and $\forall r \in \{1,2,\ldots,R\}$ it holds that:

$$F(f(\xi,\kappa_r(\xi))) - F(\xi) + L(\xi,\kappa_r(\xi)) \leq 0, \tag{17}$$

$$\kappa_r(\xi) \in \mathbb{V}_r, \tag{18}$$

$$f(\xi,\kappa_r(\xi)) \in \mathbb{X}_f. \tag{19}$$

We then have:

$$\lim_{k\to\infty} \|x(k)\| = 0, \tag{20}$$

for all initial states $x(0) \in \mathbb{X}_N$.

Proof. The proof is included in the appendix.

The above result allows one to synthesize the proposed controller and network protocol such that *stabilizing* scheduling and control sequences are obtained, provided the maximum number of consecutive dropouts is bounded.

Remark 4. The scheduling constraint of the multiple actuator-channel NCS architecture makes Theorem 3 more demanding than the sufficient conditions for stability obtained in Quevedo et al. (2007) for the one actuator-channel case. Indeed, conditions (17)–(19) need to hold for all R single actuator-link policies κ_r and, thus, also involve the plant evolution, when inputs are set to zero. Interestingly, it is sufficient to inspect only the case of one channel dropout, rather than all possible scenarios $m_i \leq N$. △

Not surprisingly, Theorem 3 is, in general, more restrictive than stability results for non-networked receding horizon control as summarized in Mayne et al. (2000). Nevertheless, it holds that if stability of receding horizon control is established via a terminal control policy which maps the entire state space to the origin, then this same policy can also be used to establish stability in the present networked case. This aspect is illustrated in the following example:

Example 5. (Closed loop stability). Consider a stable linear time invariant plant (or a set of plants) with no state constraints ($\mathbb{X} = \mathbb{R}^n$) and a quadratic cost function, i.e.,

$$\begin{aligned} &f(x,u) = Ax + Bu, && (21)\\ &L(x,u) = x^TQx + u^T\Lambda u,\ F(x) = x^TPx, \quad P,Q,\Lambda > 0 \end{aligned}$$

If we choose $\mathbb{X}_f = \mathbb{R}^n$ and

$$\kappa_r(\xi) = \bar{0}_p, \quad \forall \xi \in \mathbb{R}^n,\ \forall r \in \{1,2,\ldots,R\},$$

see (16), then $\mathbb{X}_N = \mathbb{R}^n$ and (17) reduces to:

$$\begin{aligned} &\xi^TA^TPA\xi - \xi^TP\xi + \xi^TQ\xi \\ &\qquad = \xi^T\left(A^TPA - P + Q\right)\xi \leq 0, \quad \forall \xi \in \mathbb{R}^n. \end{aligned}$$

Theorem 3 guarantees that the plant state will asymptotically tend to the origin, provided that the prediction horizon N satisfies (15) and that P is designed to satisfy the Lyapunov inequality: [7]

$$A^TPA + Q - P \leq 0. \tag{22}$$

This result applies to any LTI stable plant with an input constraint set containing the origin, such as (some) quantization constraints and convex constraint sets. △

5. SIMULATION STUDIES

To illustrate performance aspects of our NCS co-design method, we will next examine two NCS's with $R = 2$. We use the cost functional (6) with $N = 3$ and

$$L(x,u) = x^Tx + 0.1u^Tu, \quad F(x) = 0. \tag{23}$$

In all simulations we incorporate disturbances and white Gaussian zero mean state measurement noise of variance 0.01. The channels drop packets with probability 0.2.

Example 6. We first consider two non interacting unstable non-linear plant models, described via:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k)^{1.2} + 0.5x_1(k)u_1(k), \\ -0.5x_2(k) + (0.8x_2(k)+2)u_2(k) \end{bmatrix} + \delta, \tag{24}$$

where $\delta = [0\ \ 0.5]^T$ is unknown to the controller. In (24), the scalar inputs are constrained according to $u_1(k), u_2(k) \in [-1.5, 1.5]$. Furthermore, plant states are corrupted by step-like additive disturbances, denoted via $d_1(k)$ and $d_2(k)$. The initial state is $x(0) = [2\ -2]^T$.

Fig. 2 illustrates the results. Clearly, the proposed NCS co-design method gives good performance, despite the scheduling constraint, data dropouts, un-modelled disturbances, and noise. In particular, δ is compensated and

[5] In the sequel, we will assume that the constraint sets $\{\mathbb{U}_r\}$ and $\mathbb{X}$ contain the origin (of their respective spaces).

[6] Note that $m_i \geq 1$, $\forall i \in \mathbb{N}$, with equality if and only if no packet dropouts occur between time instants k_i and k_{i+1}.

[7] Note that, since the plant model is assumed to be open loop stable, A in (21) is Hurwitz and, thus (22) has a positive definite solution.

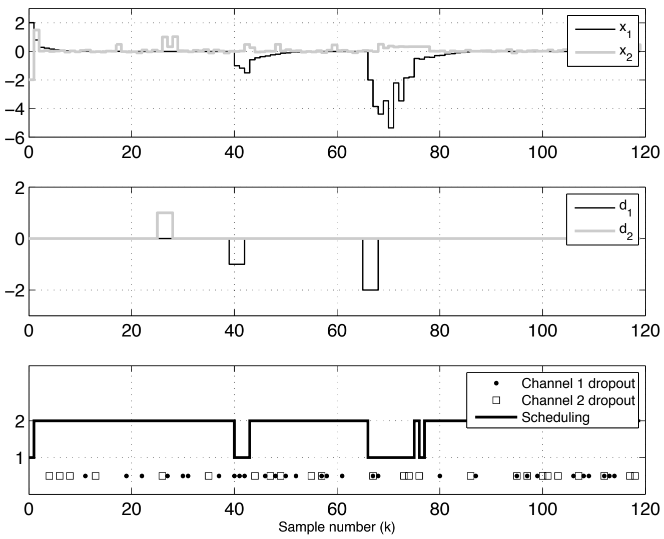

Fig. 2. Packetized Control and Scheduling of (24).

dropouts have no serious consequences on the state trajectories. A key feature is that scheduling is dynamic. Indeed, in Fig. 2 it can be appreciated that the second plant receives most of the attention (mainly due to the fact that δ must be compensated). However, whenever $d_1(k) \neq 0$, the attention is focused on the first plant, so as to compensate d_1. We note that this type of behavior cannot be achieved with static scheduling policies. △

Example 7. We next consider the unstable linear model

$$x(k+1) = \begin{bmatrix} 1.1 & 0.5 \\ 0 & 2 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u(k). \qquad (25)$$

Each component of $u(k)$ is constrained to lie in $[-2.5, 2.5]$.

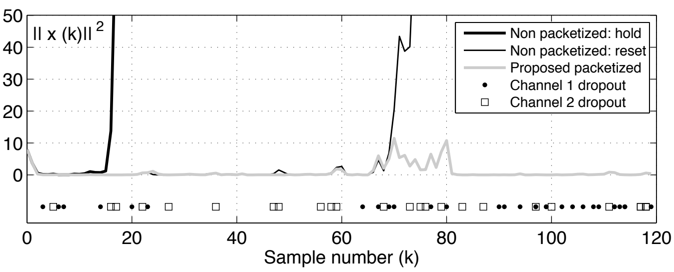

Fig. 3. Control of (25): Proposed and previous strategies.

Fig. 3 illustrates the performance achieved, when a unit magnitude step-like disturbance is added to the first state for $k \in \{66, 67, \dots, 80\}$. In Fig. 3, we have also included results obtained by using the non-packetized controllers of Gaid et al. (2006), see also Goodwin et al. (2004). These strategies incorporate scheduling constraints, but not data dropouts. [8] For simulation of these methods, whenever no data is received at some actuator node, we studied two alternatives, namely, hold the current value, and reset the value to zero. The results show that, for this example, sending properly designed packets instead of single control values is fundamental to preserve closed loop stability in the face of disturbances. △

[8] The computational load incurred by these non-packetized designs is larger than that of the present proposal. More precisely they rely upon solving R^N rather than R small optimization problems, see (11).

6. CONCLUSIONS

We have presented a network protocol and control co-design method for NCS's with multiple actuator-links. A key aspect lies in the use of buffering and predicted signals. Closed loop stability despite data dropouts can, at times, be ensured directly. Simulation results document that good performance can often be achieved even in more general situations.

REFERENCES

A. Casavola, E. Mosca, and M. Papini. Predictive teleoperation of constrained dynamic systems via internet-like channels. *IEEE Trans. Contr. Syst. Technol.*, 14(4): 681–694, July 2006.

M.-M. Ben Gaid, A. Çela, and Y. Hamam. Optimal integrated control and scheduling of networked control systems with communication constraints: Applications to a car suspension system. *IEEE Trans. Contr. Syst. Technol.*, 14(4):776–787, July 2006.

D. Georgiev and D. M. Tilbury. Packet-based control: The H_2-optimal solution. *Automatica*, 42:137–144, 2006.

G. C. Goodwin, H. Haimovich, D. E. Quevedo, and J. S. Welsh. A moving horizon approach to networked control system design. *IEEE Trans. Automat. Contr.*, 49(9): 1427–1445, September 2004.

G. C. Goodwin, M. M. Serón, and J. A. De Doná. *Constrained Control & Estimation – An Optimization Perspective.* Springer-Verlag, London, 2005.

O. C. Imer, S. Yüksel, and T. Başar. Optimal control of LTI systems over unreliable communication links. *Automatica*, 42(9):1429–1439, September 2006.

G. P. Liu, Y. Xia, and D. Rees. Predictive control of networked systems with random delays. In *Proc. IFAC World Congr.*, Prague, Czech Republic, 2005.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Optimality and stability. *Automatica*, 36(6):789–814, June 2000.

J. R. Moyne and D. M. Tilbury. The emergence of industrial control networks for manufacturing control, diagnostics, and safety data. *Proc. IEEE*, 95(1):29–47, January 2007.

G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans. Feedback control under data rate constraints: An overview. *Proc. IEEE*, 95(1):108–137, January 2007.

D. Nešić and A. R. Teel. Input-output stability properties of networked control systems. *IEEE Trans. Automat. Contr.*, 49(10):1650–1667, October 2004.

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

L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry. Foundations of control and estimation over lossy networks. *Proc. IEEE*, 95(1):163–187, January 2007.

P. L. Tang and C. W. de Silva. Compensation for transmission delays in an Ethernet-based control network using variable-horizon predictive control. *IEEE Trans. Contr. Syst. Technol.*, 14(4):707–718, July 2006.

J. Wu and T. Chen. Design of networked control systems with packet dropouts. *IEEE Trans. Automat. Contr.*, 52(7):1314–1319, July 2007.

Appendix A. PROOF OF THEOREM 3

As in the stability proofs in Quevedo et al. (2007), the key idea consists in showing that the sequence of optimal costs at the time instants where no packet dropouts occur, namely $\{V^\star(k_i)\}$, is decreasing. We will distinguish the two possible cases $m_i \leq N-1$ and $m_i = N$, separately.

1) $m_i \leq N-1$. By virtue of the buffering procedure, the first m_i elements of $\boldsymbol{\mu}^\star(k_i)$, see (8), are implemented at node $r^\star(k_i)$, i.e., we have:

$$\begin{aligned}\boldsymbol{\mu}^\star(k_i) = \{&u_{r^\star(k_i)}(k_i), u_{r^\star(k_i)}(k_i+1), \dots, u_{r^\star(k_i)}(k_{i+1}-1),\\ &\mu^\star(k_{i+1};k_i), \mu^\star(k_{i+1}+1;k_i), \dots, \mu^\star(k_i+N-1;k_i)\},\end{aligned}$$

see (8). Thus, the first m_i elements of $\boldsymbol{u}(k_i)$ and of $\boldsymbol{x}(k_i)$, see (10), correspond to inputs and states of (2). Consequently, the optimal value function at time k_i satisfies:[9]

$$\begin{aligned}V^\star(k_i) &= F(x(k_i+N;k_i)) + \sum_{\ell=k_i}^{k_{i+1}-1} L(x(\ell), u(\ell))\\ &+ L(x(k_{i+1}), u(k_{i+1};k_i)) + \sum_{\ell=k_{i+1}+1}^{k_i+N-1} L(x(\ell;k_i), u(\ell;k_i)).\end{aligned}$$

We next consider time instant $k_{i+1} = k_i + m_i$ and a pair $(\boldsymbol{\mu}^\sharp, r^\sharp)$, where $r^\sharp = r^\star(k_i)$,

$$\begin{aligned}\boldsymbol{\mu}^\sharp \triangleq \{&\mu^\star(k_{i+1};k_i), \mu^\star(k_{i+1}+1;k_i), \dots, \mu^\star(k_i+N-1;k_i),\\ &\mu^\sharp(k_i+N), \mu^\sharp(k_i+N+1), \dots, \mu^\sharp(k_{i+1}+N-1)\},\end{aligned}$$

with

$$\mu^\sharp(k_i+N+j) \in \mathbb{U}_{r^\sharp}, \quad \forall j \in \{0,1,\dots,m_i-1\}. \tag{A.1}$$

Since between k_i and k_{i+1} no valid control packets arrive at any of the R nodes, the plant input sequence, say $\boldsymbol{u}^\sharp$, which would result if $\boldsymbol{\mu}^\sharp$ was implemented at node $r^\sharp$ is:

$$\begin{aligned}\boldsymbol{u}^\sharp = \{&u(k_{i+1};k_i), u(k_{i+1}+1;k_i), \dots, u(k_i+N-1;k_i),\\ &u^\sharp(k_i+N), u^\sharp(k_i+N+1), \dots, u^\sharp(k_{i+1}+N-1)\},\end{aligned} \tag{A.2}$$

where

$$u^\sharp(k_i+N+j) = \begin{bmatrix} u_1^\sharp(k_i+N+j) \\ \vdots \\ u_R^\sharp(k_i+N+j)\end{bmatrix}, \ \forall j \in \{0,1,\dots,m_i-1\}$$

and where $\{u_r^\sharp(k_i+N+j)\}$, $r \in \{1,2,\dots,R\}$ satisfy:

$$u_r^\sharp(k_i+N+j) = \begin{cases} 0 \in \mathbb{R}^{p_r \times 1}, & \text{if } r \neq r^\star \\ \mu^\sharp(k_i+N+j), & \text{if } r = r^\star.\end{cases}$$

The cost associated with $\boldsymbol{u}^\sharp$ is related to $V^\star(k_i)$. Indeed, direct algebraic manipulations yield that

$$\begin{aligned}V(\boldsymbol{\mu}^\sharp, r^\sharp; k_{i+1}) &= V^\star(k_i) - \sum_{\ell=k_i}^{k_{i+1}-1} L(x(\ell), u(\ell))\\ &+ F(x^\sharp(k_{i+1}+N)) - F(x^\sharp(k_i+N))\\ &+ \sum_{\ell=k_i+N}^{k_{i+1}+N-1} L(x^\sharp(\ell), u^\sharp(\ell)),\end{aligned} \tag{A.3}$$

where:

$$x^\sharp(\ell+1) = f(x^\sharp(\ell), u^\sharp(\ell)), \ \ell \in \{k_i+N, \dots, k_{i+1}+N-1\},$$

with initial condition

$$x^\sharp(k_i+N) = x(k_i+N;k_i) \in \mathbb{X}_f,$$

see (12).

Whilst (A.3) holds for all sequences which obey (A.2), it is convenient to choose $\{u^\sharp(k_i+N+j)\}$ via:[10]

$$u^\sharp(k_i+N+j) = \kappa_{r^\sharp}(x^\sharp(k_i+N+j)) \in \mathbb{V}_{r^\sharp}.$$

Property (17) then amounts to:

$$F(x^\sharp(\ell+1)) - F(x^\sharp(\ell)) + L(x^\sharp(\ell), u^\sharp(\ell)) \leq 0,$$

which summed for $\ell = k_i+N$ to $\ell = k_{i+1}+N-1$ yields:

$$\begin{aligned}F(x^\sharp(k_{i+1}+N)) - F(x^\sharp(k_i+N))&\\ + \sum_{\ell=k_i+N}^{k_{i+1}+N-1} L(x^\sharp(\ell), u^\sharp(\ell)) &\leq 0.\end{aligned} \tag{A.4}$$

Use of (A.4) in (A.3) now gives:

$$V(\boldsymbol{\mu}^\sharp, r^\sharp; k_{i+1}) \leq V^\star(k_i) - \sum_{\ell=k_i}^{k_{i+1}-1} L(x(\ell), u(\ell)). \tag{A.5}$$

Whilst $(\boldsymbol{\mu}^\sharp, r^\sharp)$ is feasible at time k_{i+1}, it is not necessarily optimal. Thus, it holds that

$$V^\star(k_{i+1}) \leq V(\boldsymbol{\mu}^\sharp, r^\sharp; k_{i+1}).$$

Expression (A.5) then yields that the differences in the optimal value functions at the time instants where no packet dropouts occur satisfy:

$$\begin{aligned}V^\star(k_{i+1}) - V^\star(k_i) &\leq -\left(\sum_{\ell=k_i}^{k_{i+1}-1} L(x(\ell), u(\ell))\right)\\ &\leq -\left(\sum_{\ell=k_i}^{k_{i+1}-1} \alpha(\|x(\ell)\|)\right) \leq 0, \ \forall i \in \mathbb{N},\end{aligned}$$

where we have used (14). Since $V^\star(k_i) \geq 0$ for all i, it follows that the sequence $\{V^\star(k_i)\}_{i\in\mathbb{N}}$ is convergent so that

$$\lim_{i\to\infty} \sum_{\ell=k_i}^{k_{i+1}-1} \alpha(\|x(\ell)\|) = 0 \Longrightarrow \lim_{k\to\infty} \alpha(\|x(k)\|) = 0,$$

from where (20) follows.

2) $m_i = N$. With $m_i = N$, i.e., $k_{i+1} = k_i + N$, we have:

$$V^\star(k_i) = F(x(k_i+N)) + \sum_{\ell=k_i}^{k_{i+1}-1} L(x(\ell), u(\ell)).$$

We will next examine time k_{i+1}. Unlike in the previous case, we will consider any $r^\sharp \in \{1,2,\dots,R\}$ and the associated sequence

$$\boldsymbol{u}^\sharp = \{u^\sharp(k_{i+1}), u^\sharp(k_{i+1}+1), \dots, u^\sharp(k_{i+1}+N-1)\}, \tag{A.6}$$

where $\{u^\sharp(k_i+N+j)\}$ satisfy:

$$u^\sharp(k_i+N+j) = \kappa_{r^\sharp}(x^\sharp(k_i+N+j)) \in \mathbb{V}_{r^\sharp}$$

with

$$x^\sharp(\ell+1) = f(x^\sharp(\ell), u^\sharp(\ell)), \ \ell \in \{k_{i+1}, \dots, k_{i+1}+N-1\},$$

with initial condition $x^\sharp(k_{i+1}) = x(k_{i+1}) \in \mathbb{X}_f$.

The choice $\boldsymbol{u}^\sharp$ in (A.6) is feasible, though not optimal. The remainder of the proof now follows along similar lines as those in the previous case. □

[9] In the sequel we use the convention $\sum_{\ell=j_1}^{j_2} g(\ell) = 0$, whenever $j_2 < j_1$ and irrespective of $g(\cdot)$.

[10] This implicitly defines the values $\mu^\sharp(k_i+N+j)$ in (A.1).