https://doi.org/10.1109/ACC.2007.4282630

Packetized Predictive Control over Erasure Channels

Daniel E. Quevedo, *Member, IEEE,* Eduardo I. Silva, and Graham C. Goodwin, *Fellow, IEEE*

***Abstract*— In digital Networked Control Systems links between controller and plant are not transparent, but are affected by time-delays, data-dropouts and quantization. An important observation is that, in contemporary communication networks, such as those employing Ethernet, data is sent in large packets. This motivates the development of networked control schemes where signal predictions are sent as packets. In the present work we present such a strategy. We focus on a configuration where the controller output is connected to the plant input via a network which we assume is prone to transmission errors. By using methods from predictive control theory, we show how closed loop stability can be ensured directly in the design.**

I. Introduction

In a Networked Control System (NCS), plant and controller are typically connected via a communication network which may be shared with other applications. When compared with direct point-to-point analog wired connections, the sharing of a network simplifies the cabling and, thus, increases overall system reliability. On the other hand, the transmitted data needs to be quantized and may be affected by time delays and data-dropouts. Thus, in a NCS, linkages are not transparent, often constituting a significant bottleneck in the achievable performance, see, e.g., [1], [2].

Traditionally NCS's have relied upon special purpose network protocols such as FIP, Profibus, CAN and variants thereof. However, to increase portability, interoperability, flexibility and maintainability, there has been a growing trend to move towards general purpose protocols and technologies. Indeed, TCP/IP over (wired) Ethernet has become the most widely used network technology in industry. Also, wireless protocols have been studied; see, e.g., [3].

The direct use of general purpose network platforms in a control loop presents some serious challenges, since they were not originally designed for real-time control, but for data communications without critical timing requirements. Thus, particularly in a wide area network and also in wireless applications, time delays and data-loss will, in general be unavoidable. Given the susceptibility of control systems to time delays and, more significantly, to data-dropouts, the need to design networked controllers which give adequate performance arises.

One interesting feature of modern communication protocols is that data is sent in large packets. For example, in Ethernet the frame format allows for a data-packet of 46-1500 bytes, the overhead being 26 bytes. For IEEE 802.11 the data-packet size in each frame is up to 2312 bytes. This opens the possibility to conceive control schemes which operate at a network level and in which *packets of data*, rather than individual values, are sent through the network. In this context it makes sense to send signal predictions, which are calculated at the transmission side, to compensate for time delays and data-dropouts. Through buffering and appropriate selection logic at the receiver node, only the latest relevant value is used. This idea is related to that used in predictive interfaces and was proposed in [4] for the teleoperation of prestabilized systems. The concept also underlies more recent NCS configurations described, for instance, in [5]–[11]. We note that, within this context, predictive control methods are a natural choice, since they inherently provide signal predictions.

Whilst experimental results of NCS's which use packetized data are promising, see [5]–[9], there exist few supporting theoretical results. For example, methods for the stability analysis of packetized NCS's over channels affected by time delays have treated only the case of unconstrained linear plants, see, e.g., [9], [11]. To the best of the authors' knowledge, there exist no general methodologies which ensure closed loop stability for nonlinear plants controlled over networks affected by data-loss.

In the present work we present a packetized predictive networked control scheme in which an optimizing sequence of control inputs is sent over a communication network affected by data-loss. We will show that closed loop stability of the resulting NCS can be imposed directly in the design through rather mild conditions on the associated tuning parameters. A key aspect of our work lies in the fact that we will treat general nonlinear systems, which are subject to input and/or state constraints.

II. Brief Review of Predictive Control

The packetized predictive control algorithm under study uses predictive control ideas. Before presenting our results, we will first briefly recall basic elements of predictive control algorithms, following essentially as in [12], [13].

We consider the discrete-time nonlinear plant model:

$$x(k+1) = f(x(k), u(k)), \tag{1}$$

where the plant input and state are constrained according to:

$$u(k) \in \mathbb{U} \subseteq \mathbb{R}^p, \quad x(k) \in \mathbb{X} \subseteq \mathbb{R}^n, \qquad \forall k \in \mathbb{N} \cup \{0\}. \tag{2}$$

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

Predictive control algorithms are based on the minimization, at each time instant k and for a given plant state $x(k)$, of a cost function which uses predicted plant behaviour over

The authors are with the School of Electrical Engineering & Computer Science, The University of Newcastle, NSW 2308, Australia; e-mails: dquevedo@ieee.org, eduardo.silva@studentmail.newcastle.edu.au, graham.goodwin@newcastle.edu.au.

a finite horizon of length N. The following cost function encompasses many alternatives:

$$V(x(k), \vec{u}'(k)) \triangleq F(x'(k+N)) + \sum_{\ell=k}^{k+N-1} L(x'(\ell), u'(\ell)), \tag{3}$$

where

$$x'(\ell+1) = f(x'(\ell), u'(\ell)), \quad x'(k) = x(k) \tag{4}$$

describes the predicted quantities. The decision variables are contained in

$$\vec{u}'(k) \triangleq \{u'(k), u'(k+1), \ldots, u'(k+N-1)\}.$$

Both, the predicted state trajectory and the decision variables are constrained in accordance with (2), i.e.,

$$u'(\ell) \in \mathbb{U},\ x'(\ell) \in \mathbb{X}, \quad \forall \ell \in \{k, k+1, \ldots, k+N-1\}. \tag{5}$$

In addition, $x'(k+N)$ is typically required to satisfy a given terminal state constraint:

$$x'(k+N) \in \mathbb{X}_f \subseteq \mathbb{X}, \tag{6}$$

where $\mathbb{X}_f$ is a set containing the origin. In the cost function of (3), $F(\cdot)$ and $L(\cdot,\cdot)$ are assumed to satisfy:

$$F(x) \geq 0, \quad \forall x \in \mathbb{X}_f, \qquad F(0) = 0, \tag{7}$$

$$L(0,0) = 0,\ L(x,u) \geq \alpha(\|x\|),\ \forall x \in \mathbb{X}_N,\ \forall u \in \mathbb{U}, \tag{8}$$

where $\alpha(\cdot)\colon [0,\infty) \to [0,\infty)$ is a continuous, nondecreasing, unbounded function such that $\alpha(0) = 0$ and $\alpha(r) > 0$, for all $r > 0$. In (8),

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

denotes the set of all *feasible* initial states, i.e., states $x(k)$ such that there exists $\vec{u}'(k)$ which is compatible with (4)–(6).

Constrained minimization of $V(\cdot,\cdot)$ as in (3) gives the optimizing control sequence at time k and for state $x(k)$:

$$\vec{u}(k) \triangleq \{u(k;k), u(k+1;k), \ldots, u(k+N-1;k)\}, \tag{9}$$

the associated optimizing state predictions:[1]

$$\vec{x}(k) \triangleq \{x(k+1;k), x(k+2;k), \ldots, x(k+N;k)\},$$

and the optimal value function:

$$V^\star(x(k)) \triangleq V(x(k), \vec{u}(k)). \tag{10}$$

Despite the fact that $\vec{u}(k)$ contains feasible plant inputs over the entire horizon, in standard predictive control algorithms, only the first element is used, i.e., the plant input is set to $u(k) = u(k;k)$. Following the receding horizon optimization paradigm, at the next sampling step, i.e., at time $k+1$ and given $x(k+1)$, the horizon is shifted by one and another optimization is carried out. This yields $u(k+1) = u(k+1;k+1)$, etc.

The prediction horizon N, the terminal constraint set $\mathbb{X}_f$, and the functions $L(\cdot,\cdot)$ and $F(\cdot)$ are design variables which can be utilized to influence stability and performance of the closed loop. The following theorem, adapted from [13] (see also [12]), summarizes many existing results.

[1]In the sequel we will restrict our attention to state trajectories in $\mathbb{X}_N$.

Theorem 1: Suppose that there exists a *terminal control law* $\kappa_f\colon \mathbb{X}_f \to \mathbb{U}$ such that, for all $\xi \in \mathbb{X}_f$:

$$F(f(\xi, \kappa_f(\xi))) - F(\xi) + L(\xi, \kappa_f(\xi)) \leq 0, \tag{11}$$

$$\kappa_f(\xi) \in \mathbb{U}, \tag{12}$$

$$f(\xi, \kappa_f(\xi)) \in \mathbb{X}_f. \tag{13}$$

Then, the closed loop

$$x(k+1) = f(x(k), u(k;k))$$

has a fixed point at the origin which is globally attractive in $\mathbb{X}_N$. If, in addition, the origin belongs to the interior of $\mathbb{X}_N$ and $V^\star(\cdot)$ in (10) is continuous in a neighbourhood of the origin, then the origin is asymptotically stable in $\mathbb{X}_N$.

Proof: A detailed proof can be found in [12], [13]. ■

Predictive control methods, as described above, have been widely used in several application areas. In particular, they have been utilized in the context of NCS's with different types of communication constraints; see, e.g., [7], [14], [15].

We note that, when dealing with NCS's, the plant model (1) is to be controlled over a digital network. Thus, the set $\mathbb{U}$ will typically be finite; see, e.g. [14]. However, if the network allows for a sufficiently large data-packet, then quantization effects will often be negligible. In this situation, $\mathbb{U}$ can be regarded as a polytope, or, depending on the situation, even as $\mathbb{U} = \mathbb{R}^p$.

III. Packetized Predictive Networked Control

A shortcoming of applying predictive control (as described in the previous section) to NCS's lies in the fact that the resulting closed loop is susceptible to time delays and, especially, data-loss. These are inevitable in a NCS which uses general purpose network technology.

In the sequel, we will present a NCS strategy aimed at overcoming this deficiency. The method consists of two modules: a *Packetized Predictive Controller* (PPC) which generates data-packets; and, at the plant input side, appropriate selection logic. We will assume that the plant state is available to the controller and, thus, focus our attention on an architecture where the network is located between controller output and plant input, see Fig. 1.

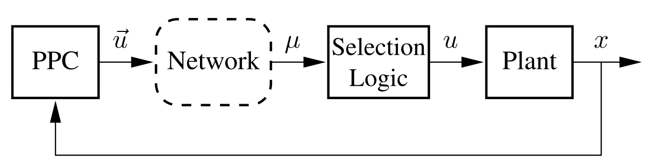

Fig. 1. Packetized Predictive Networked Control System.

A. Packetized Predictive Controller

The PPC embellishes the predictive control idea described in Section II. As before, at each sampling instant k and given $x(k)$, $V(\cdot,\cdot)$ in (3) is minimized subject to (5) and (6). The main novelty lies in that, in PPC not only $u(k;k)$ is used, but the entire optimizing sequence $\vec{u}(k)$, see (9). As depicted in Fig. 1, this sequence is sent through the network to the actuator node at each time instant k. At time $k+1$, the horizon is shifted by one and the sequence $\vec{u}(k+1)$ is transmitted. This procedure is repeated *ad infinitum*.

B. Network Model

In the present work we are interested in wired and wireless Ethernet-like channels, where data-packets are large, so that quantization is less of an issue. However, time-delays and transmission errors are likely to occur.

From a closed loop control perspective, transmission errors are the most serious network effect. This motivates us to model the network as an erasure channel, see also [7], [16]. To be more precise, we utilize a *Gilbert-Elliot model*; see, e.g., [17], [18]. In this model, network congestion is modeled by a two state Markov chain with states HIGH and LOW, corresponding to high and low congestion conditions, respectively. Associated with each state are data-drop probabilities, say, p_H for the HIGH state and p_L for the LOW state. Within each state, data-dropouts are independent and identically distributed (i.i.d.). The state transition probabilities are:

$$p_{HL} \triangleq \mathcal{P}\left\{p(k+1) = \text{LOW} \,|\, p(k) = \text{HIGH}\right\},$$
$$p_{LH} \triangleq \mathcal{P}\left\{p(k+1) = \text{HIGH} \,|\, p(k) = \text{LOW}\right\},$$

where $p(k)$ is the Markov chain state at time instant k.

If we define the discrete random process $\{d_r(k)\}_{k\in\mathbb{N}_0}$ via

$$d_r(k) = \begin{cases} 1 & \text{if data-dropout occurs at instant } k, \\ 0 & \text{if data dropout does not occur at instant } k, \end{cases}$$

then this network congestion model implies that, within each Markov chain state, $\{d_r(k)\}_{k\in\mathbb{N}_0}$ is i.i.d. Bernoulli.

The channel output at time k is given by:

$$\mu(k) = (1 - d_r(k))\vec{u}(k) \tag{14}$$

For further reference, we will denote the time instants where no data-dropouts occur via $\{k_i\}_{i\in\mathbb{N}_0}$ and define

$$m_i \triangleq k_{i+1} - k_i, \quad i \in \mathbb{N}. \tag{15}$$

Note that $m_i \geq 1$, $\forall i \in \mathbb{N}$, with equality if and only if no data-dropouts occur between k_i and k_{i+1}.

When data is lost, the NCS operates in open-loop. Thus, it is reasonable to expect that, to ensure desirable properties of the NCS, the number of consecutive data-dropouts should be limited. This will become apparent in Section IV.

C. Selection Logic

At the actuator node, the latest control value from the received data sequence, i.e., μ, is selected, see Fig. 1. This is achieved through buffering, where old data is overwritten by new data.

For example, if at some time instant k, $\mu(k) = \vec{u}(k)$, then $u(k;k)$ is implemented, i.e., $u(k) = u(k;k)$, as in standard predictive control. However, if $\vec{u}(k)$ is lost, then the latest data-packet which contains a possible control input corresponding to time k is used. That is, if $u(k;k-1)$ is available, then $u(k) = u(k;k-1)$; otherwise, $u(k) = u(k;k-2)$, etc. Finally, if no $u(k;\ell)$ is available, then the current plant input is held.

It is intuitively clear that the (time-varying and nonlinear) control algorithm which results from combining PPC and the selection logic has the potential to make the resulting NCS robust with respect to packet loss. Indeed, as we will see in Section IV, closed loop stability can be ensured by appropriate selection of tuning parameters.

D. Relationship to Previous Schemes

The idea of sending sequences rather than individual values is not new. For example, it has been used in [8], [11] for the networked control of unconstrained single-input single-output linear time invariant (LTI) systems, i.e., where

$$f(x,u) = Ax + Bu, \quad u \in \mathbb{R},\ x \in \mathbb{R}^n, \tag{16}$$

via an unconstrained LTI controller.

Also, the authors of [6], [7], [9] consider reference tracking for unconstrained LTI systems with a packetized optimization based predictive controller. Here, for zero reference signal, a quadratic cost function of the form

$$L(x,u) = x^T Q x + u^T R u \tag{17}$$

is utilized with no terminal constraints, i.e., $\mathbb{X}_f = \mathbb{X} = \mathbb{R}^n$ and $F(\cdot) = 0$. Also, particular choices of $Q > 0$ and $R \geq 0$ are used.

The early paper [4] considers reference tracking for a (prestabilized) general constrained nonlinear system and proposes the use of a packetized predictive controller with a quadratic cost function. More recently, in [5], this idea was applied to the teleoperation of LTI systems affected by bounded disturbances. Moreover, additional intelligence was proposed at the actuator side to ensure closed loop stability.

The main focus of the above work has been on demonstrating experimental performance of packetized controllers for NCS's affected by time-delays. Stabilizing properties have been studied only for given designs, i.e., a posteriori. Indeed, save for the approach presented in [5], there seem not to exist simple guidelines for choosing the tuning parameters so as to ensure closed loop stability of the resulting NCS.

In the following section, we will use tools from the predictive control framework to show how one can choose the design parameters in the PPC so as to ensure closed loop stability.

IV. Stability of Packetized Predictive NCS's with Erasure Channels

Our first observation is that, if there are no data-dropouts, i.e., if $m_i = 1$, for all $i \in \mathbb{N}$, see (15), then $u(k) = u(k;k)$ and the PPNCS reduces to a standard (i.e, non-networked) predictive control loop, whose stabilizing properties have already been characterized in Theorem 1.

We will next analyze the networked situation. For that purpose, we will assume that the number of consecutive data-dropouts is uniformly bounded such that m_i in (15) satisfies:

$$m_i \leq m^{\max}, \quad \forall i \in \mathbb{N}. \tag{18}$$

Our main result is stated in Theorem 2 given below. It uses the following technical lemma:

Lemma 1: Suppose that there exists a terminal control law $\kappa_f(\cdot)\colon \mathbb{X}_f \to \mathbb{U}$, such that, for all $\xi \in \mathbb{X}_f$, conditions (11)–(13) of Theorem 1 are satisfied. Then, for every $m \in \mathbb{N}$ and for every $\xi \in \mathbb{X}_f$, the sequence

$$\vec{v} = \{v(0), v(1), \dots, v(m-1)\} \in \mathbb{U}^m$$

defined via:

$$v(\ell) = \kappa_f(\xi(\ell)), \quad \ell \in \{0, 1, \dots, m-1\}, \tag{19}$$
$$\xi(\ell+1) = f(\xi(\ell), v(\ell)), \quad \xi(0) = \xi \tag{20}$$

yields:

$$\xi(\ell) \in \mathbb{X}_f, \quad \forall \ell \in \{1, 2, \dots, m\}, \quad \text{and} \tag{21}$$

$$\Omega(\xi, \vec{v}) \triangleq F(\xi(m)) - F(\xi(0)) + \sum_{\ell=0}^{m-1} L(\xi(\ell), v(\ell)) \leq 0. \tag{22}$$

Conversely, if, for every $m \in \mathbb{N}$ and for every $\xi \in \mathbb{X}_f$, there exists $\vec{v} \in \mathbb{U}^m$ which satisfies (19)–(22), then there exists $\kappa_f(\cdot)\colon \mathbb{X}_f \to \mathbb{U}$ such that conditions (11)–(13) of Theorem 1 are fulfilled.

Proof: The proof follows using induction. ∎

Theorem 2: Suppose that N in (3) is chosen according to:

$$N \geq m^{\max} \tag{23}$$

and that there exists $\kappa_f(\cdot)\colon \mathbb{X}_f \to \mathbb{U}$, such that, for all $\xi \in \mathbb{X}_f$, conditions (11)–(13) of Theorem 1 are satisfied.

Then, the PPNCS has a fixed point at the origin which is globally attractive in $\mathbb{X}_N$. If, in addition, the origin belongs to the interior of $\mathbb{X}_N$ and $V^\star(\cdot)$ is continuous in a neighbourhood of the origin, then it is asymptotically stable.

Proof: The proof is included in the appendix. ∎

Thus, we can directly design the PPNCS such that stability can be ensured, provided that the number of consecutive packet dropouts is bounded. This stands in contrast to existing results on packetized predictive NCSs, where closed loop stability has been studied only for given designs.

Remark 1: One might have expected that the conditions for stability of PPNCS would be stronger than those needed for standard non-networked predictive control. However, the above theorem shows that having a stable non-networked predictive control loop is generally sufficient to ensure stability of the associated PPNCS.

Remark 2: It is worth emphasizing that $\kappa_f(\cdot)$ in Theorem 2 (as well as in Theorem (1)) can be regarded as a locally stabilizing control law. A key point here is that it is not really implemented on the plant. It is simply a construct needed in the proof of stability.

Remark 3: It would be useful to remove the restriction on the horizon N, see (23). However dropping this requirement is not trivial. Indeed, since, during periods of data-loss, the plant is operated in open-loop, one cannot ensure stability for unbounded $m^{\max}$ in a general case. One particular exception corresponds to the case, where there exists $\kappa_f(\cdot)\colon \mathbb{X}_f \to \mathbb{U}$, which satisfies (11)–(13) for all $\xi \in \mathbb{X}_f$ and which can be computed at the plant input side *without explicit knowledge of the current plant state*. Here one can conceive a networked control scheme, where, if $k_{i+1} > k_i + N$, the values $\kappa_f(x(k_i + N + \ell))$, $\ell \geq 0$, are actually implemented at the plant input. Stability can then be ensured even for unbounded m_i. In particular, if $f(\cdot, 0)$ has a stable equilibrium point at the origin and $\mathbb{X}_f$ lies within its basin of attraction, then one can simply set $\kappa_f(\xi) = 0$, $\forall \xi \in \mathbb{X}_f$.

V. Example

Consider a stabilizable LTI plant with convex constraint sets $\mathbb{X}$ and $\mathbb{U}$ and a quadratic cost function, i.e.,

$$\begin{aligned} &f(x, u) = Ax + Bu, \\ &L(x, u) = x^T Q x + u^T R u, \ \ F(x) = x^T P x, \quad P, Q, R > 0. \end{aligned} \tag{24}$$

To ensure stability of the PPNCS, one can choose:

$$\kappa_f(\xi) = -K\xi, \quad K \triangleq (R + B^T P B)^{-1} B^T P A,$$

where P satisfies the algebraic Riccati equation:

$$P = A^T P A + Q - K^T (R + B^T P B) K. \tag{25}$$

Inequality (11) now becomes:

$$\xi^T \left((A - BK)^T P (A - BK) - P + Q + K^T R K \right) \xi \leq 0,$$

which, given (25), holds for every $\xi \in \mathbb{R}^n$ with equality.

If one chooses:

$$\begin{aligned} \mathbb{X}_f = &\left\{ x \in \mathbb{X}\colon K(A - BK)^\ell x \in \mathbb{U}, \ \forall \ell \in \mathbb{N}_0 \right\} \\ &\cap \left\{ x \in \mathbb{X}\colon (A - BK)^\ell x \in \mathbb{X}, \ \forall \ell \in \mathbb{N}_0 \right\}, \end{aligned}$$

then (12) and (13) are also satisfied. Theorem 2 now yields that, in the PPNCS, the origin is attractive, provided that $N \geq m^{\max}$. The corresponding result for non-networked predictive control can be found, e.g., in [12], [19], [20].

As an illustration, consider the plant model:

$$x(k+1) = \begin{bmatrix} 1.5357 & -1.9171 \\ 0.4344 & 0.4699 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k), \tag{26}$$

where $|u(k)| \leq 0.2$. Note that (26) has unstable oscillatory modes located at $1.2468\, e^{\pm 0.6363 j}$. The network is characterized via the model of Section III-B with parameters $p_H = 80\%$, $p_L = 0\%$, $p_{LH} = 5\%$ and $p_{HL} = 50\%$.

We use $L(x, u)$ as in (24) and synthesize a PPC with horizon $N = 3$, and weighting matrices $R = 0.5$, $Q = I_2$. Following Theorem 2, P is chosen as in (25). In addition, we synthesize a PPC with $F(x) = 0$, $\forall x$.

Time-domain simulations were carried out, including zero-mean Gaussian i.i.d. state measurement noise with covariance matrix $0.08 \cdot I_2$. The initial plant state was randomly chosen as $x(0) = \begin{bmatrix} 0.723 & 0.040 \end{bmatrix}^T$ and the simulation length was 1000 samples.

Fig. 2 shows typical trajectories of the plant input and norm of the state. For comparison purposes, we have also included trajectories corresponding to the non-networked case, i.e., when the associated standard predictive controller (see Section II) is used with an error free (perfect) down-link. As can be seen in this figure, the PPNCS with stabilizing P matrix yields performance which is close to that achievable

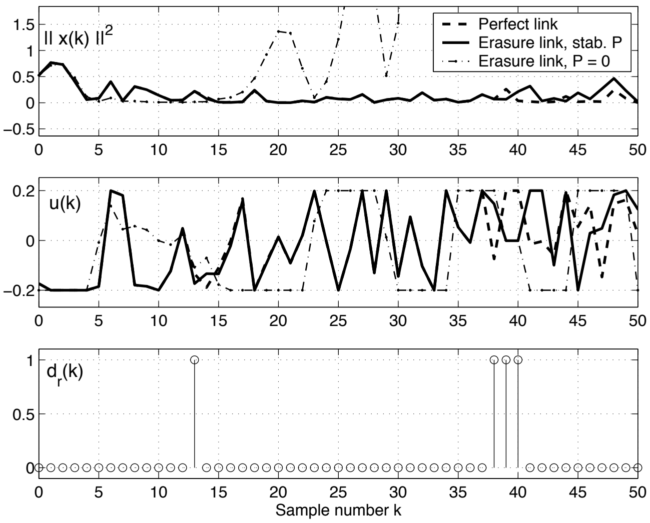

Fig. 2. Performance of the PPNCS: ideal non-networked situation (dashed), with Erasure channel and $P = 0$ (dash-dotted), and with Erasure channel and stabilizing P which solves (25) (solid).

if no data-dropouts occur. Indeed, in the present situation, the sample variance of $\|x(k)\|^2$ for the PPNCS with stabilizing P matrix is 0.0851, whilst in the non-networked case it is 0.0796. In contrast, the PPNCS with $F(\cdot) = 0$ (and also the associated standard predictive controller) fail to stabilize the plant over the network model used.

It is interesting to note that our results, namely Theorem 2, only guarantee closed loop stability in the noise free case and provided N is larger than the maximum number of consecutive data-dropouts. Despite the fact that these conditions were not satisfied in the simulation[2], the PPNCS designed according to Theorem 2 gave a stable closed loop.

VI. Conclusions

This paper has established sufficient conditions for stability of a packetized predictive controller used with erasure channels. It has been shown that, provided the number of consecutive packet losses is bounded, closed loop stability can be ensured by appropriate choice of tuning parameters.

An example has been given where, although the sufficient conditions for stability are not satisfied, the packetized networked control scheme is nonetheless stable for tuning parameters recommended by the results. Thus, the scheme may have additional robustness properties beyond those studied here. The latter topic is the subject of current research.

Appendix
Proof of Theorem 2

The key element of the proof is to show that the sequence of optimal value functions *at the time instants* $\{k_i\}$, namely $\{V^\star(x(k_i))\}_{i\in\mathbb{N}}$ constitutes a Lyapunov function for the PPNCS. For sake of clarity, we distinguish two cases, namely, $m_i \leq N-1$ and $m_i = N$.

[2]The maximum number of consecutive dropouts was $m^{\max} = 6$.

1) $m_i \leq N-1$: This is equivalent to $k_{i+1} \leq k_i + N - 1$.[3] In the PPNCS, and provided the plant model (1) is exact, then, beginning at time k_i the first m_i elements of $\vec{u}(k_i)$ are implemented. To be more precise, it follows that the optimizing sequence at time k_i (see (9)) satisfies:

$$\begin{aligned}\vec{u}(k_i) = \big\{&u(k_i), u(k_i+1), \dots, u(k_{i+1}-1),\\ &u(k_{i+1};k_i), u(k_{i+1}+1;k_i) \dots, u(k_i+N-1;k_i)\big\},\end{aligned}$$

where, in accordance with (1), $u(k)$ denotes the actual plant input at time instant k. Similarly,

$$\begin{aligned}\vec{x}(k_i) = \big\{&x(k_i+1), x(k_i+2), \dots, x(k_{i+1}),\\ &x(k_{i+1}+1;k_i), x(k_{i+1}+2;k_i) \dots, x(k_i+N;k_i)\big\}.\end{aligned}$$

The corresponding optimal value function at time k_i is:[4]

$$\begin{aligned}V^\star(x(k_i)) &= F(x(k_i+N;k_i)) + \sum_{\ell=k_i}^{k_{i+1}-1} L(x(\ell), u(\ell))\\ &+ L(x(k_{i+1}), u(k_{i+1};k_i)) + \sum_{\ell=k_{i+1}+1}^{k_i+N-1} L(x(\ell;k_i), u(\ell;k_i)).\end{aligned} \tag{27}$$

We next consider k_{i+1} and the feasible control sequence:

$$\begin{aligned}\vec{u}^\sharp \triangleq \big\{&u(k_{i+1};k_i), u(k_{i+1}+1;k_i), \dots, u(k_i+N-1;k_i),\\ &u^\sharp(k_i+N), u^\sharp(k_i+N+1), \dots, u^\sharp(k_{i+1}+N-1)\big\}.\end{aligned} \tag{28}$$

The first m_i elements of $\vec{u}^\sharp$ are equal to the last elements of $\vec{u}(k_i)$. The sequence of remaining elements, i.e.,

$$\vec{u}^r \triangleq \{u^\sharp(k_i+N), \dots, u^\sharp(k_{i+1}+N-1)\} \in \mathbb{U}^{m_i}$$

satisfies (recall the notation in (22)):

$$\Omega(x(k_i+N;k_i), \vec{u}^r) \leq 0, \tag{29}$$

where:

$$\begin{aligned}x^\sharp(\ell+1) &= f(x^\sharp(\ell), u^\sharp(\ell)) \in \mathbb{X}_f,\\ &\qquad \ell \in \{k_i+N, \dots, k_{i+1}+N-1\},\\ x^\sharp(k_i+N) &= x(k_i+N;k_i),\end{aligned}$$

but is otherwise arbitrary.[5]

The cost associated to $\vec{u}^\sharp$ is given by:

$$\begin{aligned}V(x(k_{i+1}), \vec{u}^\sharp) &= F(x^\sharp(k_{i+1}+N))\\ &+ L(x(k_{i+1}), u(k_{i+1};k_i)) + \sum_{\ell=k_{i+1}+1}^{k_i+N-1} L(x(\ell;k_i), u(\ell;k_i))\\ &+ L(x(k_i+N;k_i), u^\sharp(k_i+N)) + \sum_{\ell=k_i+N+1}^{k_{i+1}+N-1} L(x^\sharp(\ell), u^\sharp(\ell)).\end{aligned} \tag{30}$$

[3]Note that, due to (15), this also restricts $N \geq 2$.

[4]In the sequel we use the convention $\sum_{\ell=p_1}^{p_2} g(\ell) = 0$, whenever $p_2 < p_1$ and irrespective of $g(\cdot)$.

[5]Note that $x(k_i+N;k_i) \in \mathbb{X}_f$, since it arises from the constrained minimization at time k_i. Thus, existence of $\vec{u}^r$ is guaranteed by virtue of Lemma 1; e.g., set $\vec{u}^\sharp(k_i+N+\ell) = \kappa_f(x^\sharp(k_i+N+\ell))$, $\forall \ell$.

We note that the incremental costs from $\ell = k_{i+1}$ to $\ell = k_i + N - 1$ in (30) coincide with the corresponding terms in $V^\star(x(k_i))$, see (27). Thus, we can rewrite (30) as:

$$V(x(k_{i+1}), \vec{u}^\sharp) = V^\star(x(k_i)) - \left(\sum_{\ell=k_i}^{k_{i+1}-1} L(x(\ell), u(\ell)) \right) + \Omega(x(k_i + N; k_i), \vec{u}^r). \quad (31)$$

On the other hand, the candidate $\vec{u}^\sharp$, though feasible by construction, is not necessarily the optimizing sequence at time k_{i+1}. Thus, $V^\star(x(k_{i+1})) \leq V(x(k_{i+1}), \vec{u}^\sharp)$ and (31) leads to:

$$V^\star(x(k_{i+1})) \leq V^\star(x(k_i)) - \left(\sum_{\ell=k_i}^{k_{i+1}-1} L(x(\ell), u(\ell)) \right) + \Omega(x(k_i + N; k_i), \vec{u}^r).$$

Since (29) holds, we have:

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

where we have used (8) and (29). Since $V^\star(x(k_i)) \geq 0$ for all i, the sequence $\{V^\star(x(k_i))\}_{i \in \mathbb{N}}$ is convergent and

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

from where attractiveness of the origin follows. Asymptotic stability of the origin can be proved mirroring the method described in Chapter 4 of [12].

2) $m_i = N$: Here, $k_{i+1} = k_i + N$ and the sequences $\vec{u}(k_i)$ and $\vec{x}(k_i)$ contain only plant inputs and states, respectively. Furthermore, we have:

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

Similarly to the previous case, we consider a sequence:

$$\vec{u}^\sharp \triangleq \left\{ u^\sharp(k_{i+1}), u^\sharp(k_{i+1}+1), \ldots, u^\sharp(k_{i+1}+N-1) \right\} \in \mathbb{U}^N,$$

which satisfies $\Omega(x(k_{i+1}), u^\sharp) \leq 0$, and provides

$$\vec{x}^\sharp \triangleq \left\{ x^\sharp(k_{i+1}+1), x^\sharp(k_{i+1}+2), \ldots, x^\sharp(k_{i+1}+N) \right\},$$

where:

$$x^\sharp(\ell+1) = f(x^\sharp(\ell), u^\sharp(\ell)) \in \mathbb{X}_f, \quad \ell \in \{k_{i+1}, \ldots, k_{i+1}+N-1\},$$
$$x^\sharp(k_{i+1}) = x(k_{i+1}).$$

Direct calculation now gives:

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

As before, $V(x(k_{i+1}), \vec{u}^\sharp)$ is related to $V^\star(x(k_i))$. Indeed, $k_i + N = k_{i+1}$ and, therefore:

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

The remainder of the proof now follows as in the previous case.

References

[1] P. Antsaklis and J. Baillieul, "Guest editorial special issue on networked control systems," *IEEE Trans. Automat. Contr.*, vol. 49, pp. 1421–1423, Sept. 2004.

[2] M.-Y. Chow, "Guest editorial special issue on distributed network-based control systems and applications," *IEEE Trans. Ind. Electron.*, vol. 51, p. 1126, Dec. 2004.

[3] D. Hristu-Varsakelis and W. S. Levine, eds., *Handbook of Networked and Embedded Systems*. Boston, MA: Birkhäuser, 2005.

[4] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Proc. IEEE Conf. Decis. Contr.*, (Tampa, Florida), pp. 2133–2138, 1998.

[5] A. Casavola, E. Mosca, and M. Papini, "Predictive teleoperation of constrained dynamic systems via internet-like channels," *IEEE Trans. Contr. Syst. Technol.*, vol. 14, pp. 681–694, July 2006.

[6] 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.*, vol. 14, pp. 707–718, July 2006.

[7] D. Srinivasagupta, H. Schättler, and B. Joseph, "Time-stamped model predictive control: an algorithm for control of processes with random delays," *Comp. Chem. Eng.*, vol. 28, pp. 1337–1246, 2004.

[8] K. Ji and W.-J. Kim, "Real-time control of networked control systems via Ethernet," *Int. J. Contr., Automation, and Syst.*, vol. 3, pp. 591–600, Dec. 2005.

[9] G. P. Liu, J. X. Mu, D. Rees, and S. C. Chai, "Design and stability analysis of networked control systems with random communication time delay using the modified MPC," *Int. J. Contr.*, vol. 79, pp. 288–297, Apr. 2006.

[10] M.-M. B. 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.*, vol. 14, pp. 776–787, July 2006.

[11] 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.

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

[13] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Optimality and stability," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[14] 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.*, vol. 49, pp. 1427–1445, Sept. 2004.

[15] D. E. Quevedo and G. C. Goodwin, "An improved architecture for networked control systems," in *Proc. 16th IFAC World Congress*, (Prague, Czech Republic), 2005.

[16] O. C. Imer, S. Yüksel, and T. Başar, "Optimal control of LTI systems over unreliable communication links," *Automatica*, vol. 42, pp. 1429–1439, Sept. 2006.

[17] E. N. Gilbert, "Capacity of a burst-noise channel," *The Bell Syst. Tech. J.*, vol. 39, pp. 1253–1265, Sept. 1960.

[18] E. O. Elliot, "Estimates of error rates for codes on burst-noise channels," *The Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, Sept. 1963.

[19] M. Sznaier and M. J. Damborg, "Suboptimal control of linear systems with state and control inequality constraints," in *Proc. IEEE Conf. Decis. Contr.*, pp. 761–762, 1987.

[20] P. O. M. Scokaert and J. B. Rawlings, "Constrained linear quadratic regulation," *IEEE Trans. Automat. Contr.*, vol. 43, pp. 1163–1169, Aug. 1998.