

Packet-based dynamic control for nonlinear networked systems

S. Falasca, M. Gamba, L. Greco, A. Chaillet and A. Bicchi

Abstract—The possibility to remotely control a plant through a communication network offers strong advantages in terms of installation, flexibility, and maintenance. These advantages come at the price of specific challenges due to the format of the transmitted data, the network nodes scheduling, and the inherent delays. To address these obstacles, we have recently developed a control methodology that exploits the packetized nature of the transmitted data by relying on a model-based prediction of the control signal to be applied. However, this methodology was limited to static controllers. Dynamic control laws indeed require a more careful synchronization between the plant and its remote model. This paper aims at filling this gap by exploiting the transmission of the measurements history, rather than their instantaneous value only.

I. INTRODUCTION

A distributed networked architecture has the strong potential to increase flexibility, scalability and robustness of a plant, while inducing a remarkable reduction of costs and delays for both installation and maintenance [2]. These advantages justify the increasing interest in control over networks (see for instance [3], [4]). On the other hand, the use of a network as a communication medium and the distributed nature of the system involve some problems like bandwidth limitations, quantization, time-delays and packet losses, which cannot be ignored in estimation and control design [5]. An excellent discussion of the state-of-the-art is reported in [6].

An essential aspect of many Networked Control Systems (NCS), such as those using Ethernet as a communication layer, is that they organize data transmission in packets (see e.g. [7], [8]). Such networks allow carrying larger amount of information but at less predictable rates compared with dedicated communication channels. The adoption of packetized transmissions substantially alter the bandwidth/performance trade-off of traditional design. However, the potentially large size of packet payload can be exploited to reduce data transmissions without degrading the overall NCS performance. In [9] the author pioneered the idea of sending feedforward control sequences, computed on the basis of a model-based predictive (MBP) scheme, with the aim of compensating large delays in communication channels.

Work supported by European Union under grant n°257462 HYCON2 and ERC AG Grant n°ERC-291166, "SoftHands".

S. Falasca and M. Gamba are with the Centro E. Piaggio, Università di Pisa, Pisa, IT, e-mail: stefano.falasca@centropiaggio.unipi.it, massimiliano.gamba@gmail.com.

L. Greco and A. Chaillet are with the L2S-Suplec – Univ. Paris Sud 11, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France, e-mail: luca.greco@lss.supelec.fr, antoine.chaillet@supelec.fr.

A. Bicchi is with the Centro E. Piaggio, Università di Pisa, Pisa, IT and with Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy, e-mail: bicchi@centropiaggio.unipi.it.

Similar MBP schemes are adopted in [10], [11] and [12] to counteract packet dropouts and to compensate communication delays. Following developments along these lines generalized the technique to address time-varying delays and transfer intervals [13]. Such a family of methods exploiting the packet payload are usually referred to as *Packet-Based Control* (PBC) (see [14], [15]).

In [16] we presented a PBC strategy ensuring the stability of an uncertain nonlinear NCS affected by varying transmission intervals, varying (and potentially large) delays, and constrained access to the network. An (imprecise) model of the closed-loop plant was used to build a prediction of the control law valid on a given time-horizon. The state of the model was asynchronously updated by means of the measurements of the plant state provided by sensors. In order to account for distributed sensors, we considered the access to the network to be ruled by a protocol deciding which sensor node can communicate at each instant. The control sequence sent by the remote controller was stored in an embedded memory on the plant side, and the right control command in the sequence was chosen by comparing the plant clock with the time stamp in the control sequence.

The present paper extends the framework presented in [16] by allowing the use of a dynamic control law. In order to adopt a dynamic controller, its internal state has to be consistently updated at each new control sequence generation according to the update of the state of the model used for the prediction. A direct application of the aforementioned framework would require the update of the internal state of each remote dynamic component (model + controller), by means of a protocol ensuring convenient error-decreasing properties.

We update the internal state of the dynamic controller in a way that is consistent with the behavior it would have if it were directly connected to the plant (without network). Such a method is equivalent to having a virtual controller on the plant side and emulating the sending of its state through the network towards the remote controller the same way the state of the plant is sent to update the remote model. The virtual sending of the controller state is realized by sending the history of the measured outputs of the plant and by using these output sequences to feed the remote controller. The knowledge of the history of plant outputs is indeed the unique information required to consistently run the controller. It is worth mentioning that the output history can be sent still exploiting the large payload of packets by slicing it in sequences over non overlapping time horizons. If output sensors are distributed, outputs are partitioned and the history of each sensor is assumed to be sent according

to a static protocol similar to Round Robin.

As a first step towards a comprehensive framework capable of encompassing both dynamic controllers and observers, we make here the simplifying hypothesis of not using an observer. Instead, we assume the plant state to be sent over the network by the distributed sensors. On the one hand, this choice allows us to approach the problem of the dynamic controller by means of the formalism in [16], on the other hand, however, we need to transmit both the instantaneous state measurements

By using the formalism of [16], we are able to prove the exponential stability of the NCS over a prescribed basin of attraction, provided that some explicit bounds on the Maximum Allowable Delay (MAD, [6]) and on the Maximum Allowable Transfer Interval (MATI, [17]) are satisfied. We finally apply our technique to the control of a Furuta pendulum involving an output-feedback dynamic controller.

NOTATION

Given a set $A \subset \mathbb{R}$ and $a \in A$, $A_{\geq a}$ denotes the set $\{s \in A \mid s \geq a\}$. Given $R \geq 0$, B_R denotes the closed ball of radius R centered in zero: $B_R \triangleq \{x \in \mathbb{R}^n \mid |x| \leq R\}$. We denote with \mathbb{N} the set of positive integer numbers. We use mod to denote the modulo operator, i.e. given $m, n \in \mathbb{Z}_{\geq 0}$, $m \text{ mod } n = p$ if and only if there exists $r \in \mathbb{Z}_{\geq 0}$ such that $m = rn + p$ with $p < n$. We define the ceiling function $\lceil \cdot \rceil : \mathbb{R} \rightarrow \mathbb{Z}$ as $\lceil x \rceil \triangleq \min\{m \in \mathbb{Z} \mid m \geq x\}$. Given $t \in \mathbb{R}$ and a piecewise continuous function $f : \mathbb{R} \rightarrow \mathbb{R}^n$, we use the notation $f(t^+) := \lim_{s \rightarrow t, s > t} f(s)$

II. SYSTEM DESCRIPTION

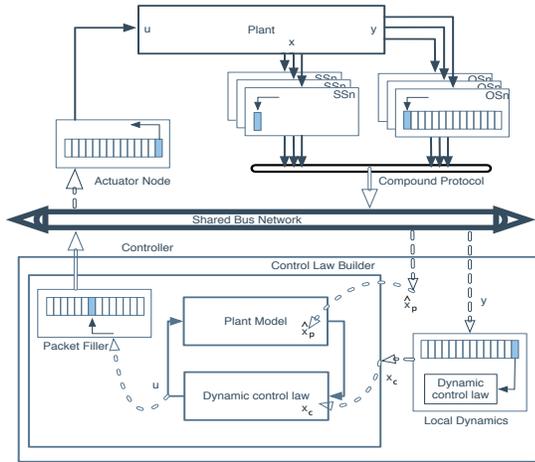


Fig. 1. The proposed control architecture

This section provides a detailed description of the architecture (controller, network and plant) of the NCS analyzed in the present work. Consider Figure 1 as a pictorial reference.

A. The Plant and the Controller

We address the stabilization of a nonlinear continuous-time system of the form

$$\dot{x}_p = f_p(x_p, u) \quad (1)$$

$$y = g_p(x_p), \quad (2)$$

where $x_p : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n_p}$ is the plant state, $y : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n_y}$ is the output, $u : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n_u}$ represents the control input, and $f_p : \mathbb{R}^{n_p} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_p}$ and $g_p : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_y}$ denote locally Lipschitz functions. For this system, we assume that a nominal dynamic feedback controller of the form

$$\dot{x}_c = f_c(x_c, y) \quad (3)$$

$$u = g_c(x_c, y) \quad (4)$$

has been developed by neglecting the network effects. Here $x_c : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n_c}$ is the controller state, and $f_c : \mathbb{R}^{n_c} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_c}$ and $g_c : \mathbb{R}^{n_c} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_u}$ denote locally Lipschitz functions. Letting $x(t) \triangleq [x_p^T(t), x_c^T(t)]^T \in \mathbb{R}^{n_p+n_c} = \mathbb{R}^n$ and

$$f(x, u) \triangleq \begin{bmatrix} f_p(x_p, u) \\ f_c(x_c, g_p(x_p)) \end{bmatrix}$$

$$g(x) \triangleq g_c(x_c, g_p(x_p)),$$

the closed-loop system (1)-(4) in the absence of network effects simply reads

$$\dot{x} = f(x, u) \quad (5)$$

$$u = g(x). \quad (6)$$

We assume that the nominal controller (3)-(4) globally exponentially stabilizes the plant (1)-(2) in the absence of network effects.

Assumption 1. (Nominal GES) *The origin of the system (1)-(2) in closed-loop with (3)-(4) is globally exponentially stable (GES), and there exists a differentiable function $V : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ and constants $\underline{\alpha}, \bar{\alpha}, \alpha, d > 0$ such that the following conditions hold for all $x \in \mathbb{R}^n$*

$$\begin{aligned} \underline{\alpha} \|x\|^2 &\leq V(x) \leq \bar{\alpha} \|x\|^2 \\ \frac{\partial V}{\partial x}(x) f(x, g(x)) &\leq -\alpha \|x\|^2 \\ \left\| \frac{\partial V}{\partial x}(x) \right\| &\leq d \|x\|. \end{aligned}$$

We also assume that local Lipschitz constants are available to the designer.

Assumption 2. (Local Lipschitz) *Given $R_x, R_u > 0$, there exist some constants $\lambda_f, \lambda_\kappa > 0$ such that, for all $x_1, x_2 \in B_{R_x}$ and all $u_1, u_2 \in B_{R_u}$, the following inequalities hold*

$$\|f(x_1, u_1) - f(x_2, u_2)\| \leq \lambda_f (\|x_1 - x_2\| + \|u_1 - u_2\|) \quad (7)$$

$$\|g(x_1) - g(x_2)\| \leq \lambda_\kappa \|x_1 - x_2\|. \quad (8)$$

The control strategy analyzed in this paper aims at compensating the network-induced effects by relying on a prediction of the plant behavior. To that aim, we assume that a model for (1)-(2) is known:

$$\dot{\hat{x}}_p = \hat{f}_p(\hat{x}_p, \hat{u}) \quad (9)$$

$$\hat{y} = \hat{g}_p(\hat{x}_p). \quad (10)$$

This model in closed loop with the nominal controller (3)-(4) reads

$$\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}) \quad (11)$$

$$\hat{u} = \hat{g}(\hat{x}), \quad (12)$$

where $\hat{x} \triangleq (\hat{x}_p^T, \hat{x}_c^T)^T : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ and

$$\begin{aligned} \hat{f}(\hat{x}, \hat{u}) &\triangleq \begin{bmatrix} \hat{f}_p(\hat{x}_p, \hat{u}) \\ f_c(\hat{x}_c, \hat{g}_p(\hat{x}_p)) \end{bmatrix} \\ \hat{g}(\hat{x}) &\triangleq g_c(\hat{x}_c, \hat{g}_p(\hat{x}_p)). \end{aligned}$$

The plant-model inaccuracy is assumed to be sector-bounded.

Assumption 3. (Sector-Bounded Model Inaccuracy) *Given $R_x, R_u > 0$, there exists a constant $\lambda_{f\hat{f}} \geq 0$ such that, for all $x \in B_{R_x}$ and all $u \in B_{R_u}$,*

$$\|f(x, u) - \hat{f}(x, u)\| \leq \lambda_{f\hat{f}} (\|x\| + \|u\|). \quad (13)$$

B. The Network

Control sequences are sent as packets. An embedded control device receives, decodes, synchronizes these packets and applies control commands to the plant. We consider that measurements are taken and sent at instants $\{\tau_i^m\}$, and are received by the remote controller at instants $\{\tau_i^m + T_i^m\}$, $i \in \mathbb{N}$. In other words, $\{T_i^m\}$ denotes the sequence of (possibly time-varying) measurement data delays. These delays cover both processing time and transmission delays on the measurement chain. Similarly, control commands are sent over the network at time instants $\{\tau_i^c\}$. They reach the plant at instants $\{\tau_i^c + T_i^c\}$, where $\{T_i^c\}$ denotes the sequence of delays accounting for both the computation time and the transmission delay from the remote controller to the plant.

Assumption 4. (Network) *The communication network satisfies the following properties:*

- i) **(MATI)** *There exist two constants $\tau^m, \tau^c \in \mathbb{R}_{\geq 0}$ such that $\tau_{i+1}^m - \tau_i^m \leq \tau^m$ and $\tau_{i+1}^c - \tau_i^c \leq \tau^c$, $\forall i \in \mathbb{N}$;*
- ii) **(mTI)** *There exist constants $\varepsilon^m, \varepsilon^c \in \mathbb{R}_{\geq 0}$ such that $\varepsilon^m \leq \tau_{i+1}^m - \tau_i^m$ and $\varepsilon^c \leq \tau_{i+1}^c - \tau_i^c$, $\forall i \in \mathbb{N}$;*
- iii) **(MAD)** *There exist two constants $T^m, T^c \in \mathbb{R}_{\geq 0}$ such that $T_i^m \leq T^m$ and $T_i^c \leq T^c$, $\forall i \in \mathbb{N}$.*

Properties *i*) and *ii*) state that the inter-sending time is lower and upper bounded both on the control side and on the measurement side of the network. The minimum transfer interval (mTI) is needed in order to avoid Zeno phenomena, but will also be used for design purposes. The upper bound in *i*) is referred to as maximum allowable transfer interval (MATI). Property *iii*) states that the delays are bounded.

C. The Network Protocol: Physical Layer

The system involves two different kinds of sensor nodes sending measurements: ℓ_x state-sending nodes (SSn) and ℓ_y output-sending nodes (OSn). Sensors are assumed to be distributed and synchronized with each other. Among the $\ell = \ell_x + \ell_y$ nodes, only one node at a time can send its information (as a consequence, only partial knowledge of the plant state is available at each time instant).

1) *Overall protocol:* The access to the network is ruled by an overall protocol choosing, at each instant τ_i^m , which node communicates its data. In order to limit the cumulated delays on the output measurements, we assume that the nodes

access the network as follows: after each SSn access, all OSn are required to send their data. Then access is again granted to a SSn, and so on. This rule can be formally stated by extracting from the sequence of access times $\{\tau_i^m\}$ two subsequences $\{\tau_{s_i}^m\}$ and $\{\tau_{o_i}^m\}$. More precisely, we define two sequences $\{s_i\}_{i \in \mathbb{N}}$, $\{o_i\}_{i \in \mathbb{N}}$ having values in \mathbb{N} . Such sequences have the following meaning: at time $\tau_{s_i}^m$, $s \in \{s_i\}$, a SSn is granted access to the network; at time $\tau_{o_i}^m$, $o \in \{o_i\}$, an OSn has the ability to send. The policy is such that the two sequences exhibit the following property: $\{s_i\} \cup \{o_i\} = \mathbb{N}$, $\{s_i\} \cap \{o_i\} = \emptyset$.

2) *State protocol:* The SSn are granted access to the network according to a protocol ruled by the map involving the estimation error $e_p(t) \in \mathbb{R}^{n_p}$ defined as $e_p(t) \triangleq \hat{x}_p(t) - x_p(t)$:

$$e_p(\tau_{s_i}^{m+}) = h_p(i, e_p(\tau_{s_i}^m)), \quad \forall i \in \mathbb{N}, \quad (14)$$

where $h_p : \mathbb{N} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_p}$. This protocol is assumed to induce an exponential decrease of the error e_p when the inter-sample dynamics are neglected; i.e. we are interested in UGES protocols. We recall here a slightly modified version of the definition in [18] as given in [16].

Definition 1 (UGES protocol). A function $h : \mathbb{N} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is said to be a UGES protocol with parameters $\underline{a}, \bar{a}, \rho, c$ if there exist a function $W : \mathbb{N} \times \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ locally Lipschitz in its second argument and constants $\underline{a}, \bar{a}, c \in \mathbb{R}_{>0}$ and $\rho \in [0, 1)$ such that:

$$\begin{aligned} \underline{a} \|e\| &\leq W(i, e) \leq \bar{a} \|e\| \\ W(i+1, h(i, e)) &\leq \rho W(i, e) \end{aligned} \quad (15)$$

for all $e \in \mathbb{R}^n$ and all $i \in \mathbb{N}$, and

$$\left\| \frac{\partial W}{\partial e}(i, e) \right\| \leq c \quad (16)$$

for almost all $e \in \mathbb{R}^n$ and all $i \in \mathbb{N}$.

Assumption 5. (UGES of SSn protocol) *The protocol (14) is UGES with parameters $\underline{a}_p, \bar{a}_p, \rho_p, c_p$.*

3) *Output protocol:* The OSn are granted access to the network according to the Round Robin protocol. We keep track of which OSn transmits at a given time by means of the sequence $\{\nu_i\}_{i \in \mathbb{N}}$ having values in $[1, \ell_y] \subset \mathbb{N}$ defined as

$$\nu_{i+1} = (\nu_i \bmod \ell_y) + 1 \quad (17)$$

and we consider $\nu_1 = 1$ to express the fact that the OSn number 1 sends first. The ν_i -th OSn is thus granted the access to the network at time $\tau_{o_i}^m$.

III. ALGORITHM DESCRIPTION

At the plant side of the network, three kinds of devices are needed, namely the actuator node (ACn), the SSn and the OSn. The controller, instead, is decomposed into two modules: the *Local Dynamics* and the *Control Law Builder*.

In this section we describe the role of each module and provide a model for the overall closed-loop system.

A. The Plant

1) *Actuator Node*: The ACn is in charge of receiving, decoding and re-synchronizing packets sent by the controller and then actuating the plant. Each received packet contains a time-stamp and a certain number of control values which are stored in a local buffer. The actuator node compares the time-stamp of the latest packet it received with its internal clock and moves within the control sequence up to the corresponding control value to be applied.

2) *State-Sending Node*: The state $x_p(t) \in \mathbb{R}^{n_p}$ of the system (1) is partitioned in ℓ_x components as $x_p(t) = [x_{p,1}^T(t), \dots, x_{p,\ell_x}^T(t)]^T$ with $x_{p,l}(t) \in \mathbb{R}^{p_l}$ and $\sum_{l=1}^{\ell_x} p_l = n_p$. When at a time instant τ_s^m a SSn $l \in \{1, \dots, \ell_x\}$ is granted access to the network, it encodes the instantaneously sensed value $x_{p,l}(\tau_s^m)$ into a packet, timestamps it and sends it to the controller.

3) *Output-Sending Node*: The output $y(t) \in \mathbb{R}^{n_y}$ of the system (1) is partitioned in ℓ_y components as $y(t) = [y_1^T(t), \dots, y_{\ell_y}^T(t)]^T$ with $y_l(t) \in \mathbb{R}^{n_{y_l}}$ and $\sum_{l=1}^{\ell_y} n_{y_l} = n_y$. An OSn $l \in \{1, \dots, \ell_y\}$ continuously monitors the sensed output $y_l(t)$, storing the readings into a buffer. When at a time instant $\tau_{o_k}^m$ the l -th OSn is granted access to the network (i.e. $\nu_k = l$), it timestamps and encodes the output history stored into the buffer and sends it to the controller. After the sending, the buffer is flushed. According to (17), the latest time the l -th OSn sent its output history was $\tau_{o_{k-\ell_y}}^m$. We assume conventionally that $\tau_{o_{k-\ell_y}}^m = 0$ if at time $\tau_{o_k}^m$ the l -th OSn makes its first transmission (i.e. if $k < \ell_y$). Therefore, the time horizon of the l -th output history is $[\tau_{o_{k-\ell_y}}^m, \tau_{o_k}^m]$ and we will write it as follows: $y_l[\tau_{o_{k-\ell_y}}^m, \tau_{o_k}^m]$.

B. The Controller

1) *Local Dynamics*: The output histories reach the Local Dynamics module, where they are stored and used to reconstruct the state of the controller. As discussed in the Introduction, the controller state reconstruction is necessary to consistently initialize the dynamic controller before running the prediction and computing a new control packet. Assuming that the controller state $x_c(\sigma)$ is known at some time σ , then the state $x_c(\tau)$ at a time $\tau > \sigma$ can be reconstructed by integrating the controller dynamics (3), provided that the output history $y_{[\sigma, \tau]}$ is known. In other words, the reconstruction of the controller state requires the histories of all the ℓ_y OSn to be available to the Local Dynamics module. Moreover, all the histories should have an overlapping time interval. As a consequence, the Local Dynamics cumulates a delay on the reconstruction of the controller state that is quantified in the following proposition.

Proposition 2 (Virtual packets' delay). *For each $k \in \mathbb{N}$, the controller state $x_c(\tau_{o_k}^m)$ is reconstructed by the Local Dynamics module at time $\tau \leq \tau_{o_k}^m + T^f$ with $T^f \triangleq \ell_y \tau^m + T^m$.*

For sake of simplicity let us assume $\nu_k = 1$, that is at time $\tau_{o_k}^m$ the OSn 1 is sending the output history $y_1[\tau_{o_{k-\ell_y}}^m, \tau_{o_k}^m]$. We use again the convention that $\tau_{o_{k-\ell_y}}^m = 0$ if $k < \ell_y$.

Such an output history reaches the Local Dynamics module after a delay $T_{o_k}^m \leq T^m$. Similarly, the output history $y_2[\tau_{o_{k+1-\ell_y}}^m, \tau_{o_{k+1}}^m]$ sent by the OSn 2 at time $\tau_{o_{k+1}}^m$ reaches the Local Dynamics after a delay $T_{o_{k+1}}^m \leq T^m$. Hence, for the Local Dynamics to receive both the output histories, a time $\max\left(T_{o_k}^m, \tau_{o_{k+1}}^m - \tau_{o_k}^m + T_{o_{k+1}}^m\right) \leq \tau^m + T^m$ is required. The same argument holds for all the ℓ_y OSn. Moreover, the entire cycle of the output sendings is interleaved with a state sending, thus adding a further delay to the reconstruction which is upper bounded by τ^m . Finally, an upper bound for the time required to receive all the histories is given by $(\ell_y - 1)\tau^m + T^m + \tau^m = \ell_y \tau^m + T^m$. We stress that the Local Dynamics module can store the past histories, therefore we can assume they all have the same starting time, for instance $\tau_{o_{k-1}}^m$. Since the controller state $x_c(\tau_{o_{k-1}}^m)$ is known by previous reconstructions (or by initialization if $\tau_{o_{k-\ell_y}}^m = 0$), the state $x_c(\tau_{o_k}^m)$ can be reconstructed at $\tau_{o_k}^m + T^f$, by integrating the controller dynamics (3) with the output history $y_{[\tau_{o_{k-1}}^m, \tau_{o_k}^m]}$.

2) *Control Law Builder*: When a new measurement is received, the remote controller uses the new data in order to update an estimate of the state of the plant. The controller then computes a prediction of the control signal over a fixed time horizon

$$T_0^p \geq T^c + T^f + \tau^m + \tau^c, \quad (18)$$

where τ^m , τ^c and T^c are given by Assumption 4 and T^f is defined in Proposition 2. This is done by numerically running the model (11)-(12) based on the latest state reconstruction. Such computation generates values for the function $\hat{u}(t)$ (see (12)) over the horizon T_0^p , which are then coded, marked with the appropriate time-stamp, and put in a single packet which is sent at the next network access.

C. The Network Protocol: Virtual Layer

Section II-C was devoted to the description of the physical protocol used to arbitrate the node scheduling. In this section, instead, we build a virtual layer upon the physical layer in order to account for the controller state reconstruction described in Section III-B.1. The virtual state protocol coincides with the physical state protocol described in Section II-C.2, whilst the virtual output protocol and the overall protocol (now named *Compound Protocol*) are defined in the following sections.

1) *The virtual output protocol*: The controller state reconstruction performed by the Local Dynamics (see Section III-B.1) allows us to consider the controller as if it were executed on the plant side and its internal state sent over the network. This is tantamount to considering that at time $\tau_{o_k}^m$ a virtual packet containing $x_c(\tau_{o_k}^m)$ is sent by the plant and that such a packet incurs a delay no larger than T^f (see Proposition 2). Therefore, while the sequence $\{\tau_i^m\}$ of sending times is unchanged, we need to define a new sequence $\{T_i^f\}$ to account both for the delays affecting the physical packets and those affecting the virtual packets. The

sequence of state-sending delays $\{T_i^f\}_{i \in \mathbb{N}}$, $T_i^f \in \mathbb{R}_{\geq 0}$ is defined as:

$$T_i^f \triangleq \begin{cases} T_i^m & \text{if } i \in \{s_k\}_{k \in \mathbb{N}} \\ \min \{\mathcal{T}_{\tau_i^m}\} & \text{otherwise,} \end{cases} \quad (19)$$

where $\mathcal{T}_\tau \triangleq \{T \mid x_c(\tau) \text{ is reconstructed at } \tau + T\}$.

By virtue of Proposition 2, the following inequality holds:

$$T_i^f \leq T^f, \forall i \in \mathbb{N}. \quad (20)$$

The virtual output protocol can be described as a discrete-time map similar to that given for the state protocol in (14). Given the estimation error $e_c(t) \in \mathbb{R}^{n_c}$, defined as $e_c(t) \triangleq \hat{x}_c(t) - x_c(t)$, we have

$$e_c(\tau_{o_i}^{m+}) = h_c(i, e_c(\tau_{o_i}^m)), \forall i \in \mathbb{N}, \quad (21)$$

where $h_c : \mathbb{N} \times \mathbb{R}^{n_c} \rightarrow \mathbb{R}^{n_c}$. When an OSn transmits its history, the state of the controller is perfectly known (even if with a large delay). Therefore, the function h_c takes in our case the null value and we simply write

$$e_c(\tau_{o_i}^{m+}) = 0, \forall i \in \mathbb{N}. \quad (22)$$

The protocol (22) is clearly UGES.

2) *The compound protocol:* We now gather together the physical packets, containing the state of the system, and the virtual packets, containing the state of the controller. They are used to design a protocol which acts on the compound error $(e_p^T(t), e_c^T(t))^T$. In the following proposition we prove that, if the protocol updating e_p is UGES (see Assumption 5), then the compound protocol is UGES as well.

Proposition 3 (Compound protocol). *The function $h : \mathbb{N} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$*

$$h(i, [e_p^T, e_c^T]^T) \triangleq \begin{bmatrix} h_s(i, e_p) \\ h_o(i, e_c) \end{bmatrix} \quad (23)$$

with

$$h_s(i, e_p) \triangleq \begin{cases} h_p(q_s(i), e_p) & \text{if } i \in \{s_k\}_{k \in \mathbb{N}} \\ e_p & \text{otherwise} \end{cases} \quad (24)$$

$$h_o(i, e_c) \triangleq \begin{cases} e_c & \text{if } i \in \{s_k\}_{k \in \mathbb{N}} \\ h_c(q_o(i), e_c) & \text{otherwise,} \end{cases} \quad (25)$$

where $h_p : \mathbb{N} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_p}$ and $h_c : \mathbb{N} \times \mathbb{R}^{n_c} \rightarrow \mathbb{R}^{n_c}$ are defined in (14) and (21)-(22), respectively; and the functions $q_s : \mathbb{N} \rightarrow \mathbb{N}$, $q_o : \mathbb{N} \rightarrow \mathbb{N}$ are given by

$$q_s(i) \triangleq \begin{cases} 1 & \text{if } i = 1 \\ \max \{k \in \mathbb{N} \mid s_k \leq i - 1\} & \forall i \in \mathbb{N}_{\geq 2} \end{cases} \quad (26)$$

$$q_o(i) \triangleq \begin{cases} 1 & \text{if } i = 1 \\ \max \{k \in \mathbb{N} \mid o_k \leq i - 1\} & \forall i \in \mathbb{N}_{\geq 2}, \end{cases} \quad (27)$$

defines a UGES protocol with parameters $\rho_p^{1-1/\ell_y} \underline{\alpha}_p$, $\bar{\alpha}_p$, ρ_p^{1/ℓ_y} and c_p .

D. The Overall Model

The loop composed of the system (1)-(2) and the controller node which executes the algorithms described in Section III-B can be summarized as follows (see also [16]).

The NCS model has a state $x(t)$ which models the internal state of the plant as well as the state of the controller as it would act if it were directly connected to the output of the plant. Moreover, N vectors of additional state variables are used to model the estimations of the vector \hat{x} . N represents the number of packets, either real or virtual, that can be received by the controller during the time T_0^p . In view of Assumption 4, it is defined as

$$N \triangleq \left\lceil \frac{T_0^p - \tau_i^m}{\varepsilon^m} \right\rceil + 1. \quad (28)$$

By means of $\bar{x}(t), e(t) \in \mathbb{R}^{Nn}$ defined as $\bar{x}(t) \triangleq [x^T(t), \dots, x^T(t)]^T$ and $e(t) \triangleq [e_1^T(t), \dots, e_N^T(t)]^T$, $e_i(t) \in \mathbb{R}^n$, the closed-loop dynamics of the NCS can be compactly written as

$$\dot{x} = F(t, \bar{x}, e) \quad (29a)$$

$$\dot{e} = G(t, \bar{x}, e) \quad (29b)$$

$$e(\tau_i^{m+}) = H(i, e(\tau_i^m)), \quad (29c)$$

where

$$F(t, \bar{x}, e) = f(x, v(t, e + \bar{x})) \quad (30a)$$

$$G(t, \bar{x}, e) = \begin{bmatrix} \hat{f}(e_1 + x, \hat{g}(e_1 + x)) - f(x, v(t, e + \bar{x})) \\ \vdots \\ \hat{f}(e_N + x, \hat{g}(e_N + x)) - f(x, v(t, e + \bar{x})) \end{bmatrix} \quad (30b)$$

$$H(i, e) = \begin{bmatrix} e_1 + (h(i, e_N) - e_1) \eta(i, 1) \\ e_2 + (h(i, e_1) - e_2) \eta(i, 2) \\ \vdots \\ e_N + (h(i, e_{N-1}) - e_N) \eta(i, N) \end{bmatrix}, \quad (30c)$$

where $\eta : \mathbb{N} \times \{1, \dots, N\} \rightarrow \{0, 1\}$ identifies the index of the relevant state estimate

$$\eta(i, r) \triangleq \begin{cases} 1 & \text{if } \mu(i) = r \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

and $\mu : \mathbb{N} \rightarrow \{1, \dots, N\}$ is defined as the following periodic inspection $\mu(i) \triangleq ((i-1) \bmod N) + 1$. The control signal v in (30a) and (30b) is defined as the emulation of (12) based on the available state estimate:

$$v(t, [\hat{x}_1^T, \dots, \hat{x}_N^T]^T) \triangleq \sum_{k=1}^N \hat{g}(\hat{x}_k) \zeta(t, k) \quad (32)$$

where $\hat{x}_i \in \mathbb{R}^n$ and $\zeta : \mathbb{R}_{\geq 0} \times \{1, \dots, N\} \rightarrow \{0, 1\}$ is defined as

$$\zeta(t, k) \triangleq \begin{cases} 1 & \text{if } \exists j \in \mathbb{N} \text{ s.t. } \mu(\gamma(j)) = k \\ & \text{and } t \in (\tau_j^c + T_j^c, \tau_{j+1}^c + T_{j+1}^c] \\ 0 & \text{otherwise} \end{cases} \quad (33)$$

and $\gamma : \mathbb{N} \rightarrow \mathbb{N}$, defined as

$$\gamma(j) \triangleq \max \left\{ i \in \mathbb{N} \mid \tau_i^m + T_i^f < \tau_j^c \right\}, \quad (34)$$

denotes the index of the latest measure received before τ_j^c .

IV. MAIN RESULT

We now have all the ingredients to state our main result.

Theorem 4. *Assume that Assumptions 1, 4 and 5 hold. Given some $R > 0$, fix $R_x = R$ and $R_u = \lambda_k R$ and suppose that Assumptions 2 and 3 hold with these constants. Let $\underline{a}_p, \bar{a}_p, \rho_p, c_p, \underline{\alpha}, \bar{\alpha}, \alpha, d, \lambda_{f\hat{f}}, \lambda_f$ and λ_k be generated by these assumptions. Pick $\underline{a} = \underline{a}_p \rho_p^{1-1/\ell_y}$, $\bar{a} = \bar{a}_p, \rho = \rho_p^{1/\ell_y}, c = c_p$ and define $a_H \triangleq \bar{a}$, $a_L \triangleq \frac{\bar{a}}{N} \min \left\{ 1, \left(\frac{\bar{a}}{\underline{a}} \right)^2 \frac{1}{\rho} \right\}$. Assume that the following conditions on $\tau^m, T^f, \tau^c, \varepsilon^m$ hold:*

$$\begin{aligned} \tau^m \in [\varepsilon^m, \tau^{m*}), \tau^{m*} &\triangleq \frac{1}{L} \log \left(\frac{M\gamma_2 + a_L L}{M\gamma_2 + a_L \rho L} \right) \\ N &= \left\lceil \frac{T^c + T^f + \tau^c}{\varepsilon^m} \right\rceil + 1 \end{aligned} \quad (35)$$

where

$$\begin{aligned} L &\triangleq \frac{c}{a_L} \left((1 + \lambda_k) \sqrt{N} \lambda_{f\hat{f}} + \right. \\ &\quad \left. \sqrt{N} \lambda_f + (\sqrt{N-1} + N - 1) \lambda_f \lambda_k \right) \\ M &\triangleq (1 + \lambda_k) c N \lambda_{f\hat{f}} \quad \gamma_2 \triangleq \frac{d}{\alpha} \sqrt{\frac{\bar{a}}{\underline{a}}} \lambda_f \lambda_k. \end{aligned} \quad (36)$$

Then the origin of the NCS (29) is exponentially stable with radius of attraction

$$\tilde{R} \triangleq \frac{R}{K} \quad (37)$$

where $K \triangleq \frac{\sqrt{2}}{1-\gamma_1\gamma_2} \max \{ (1+\gamma_1)k_2, (1+\gamma_2)k_1 \}$, $\gamma_1 \triangleq \frac{\exp(L\tau^m)-1}{a_L L(1-\rho \exp(L\tau^m))} M$, $k_1 \triangleq \frac{\alpha_H}{\rho a_L}$ and $k_2 \triangleq \sqrt{\frac{\bar{\alpha}}{\underline{\alpha}}}$.

The conditions expressed in (35) establish a relation among all the relevant parameters, namely $\varepsilon^m, T^c, T^f, \tau^c$ and τ^m . Notice that (20) can be used to express such a relation in terms of the measurement MAD T^m and the number of OSn ℓ_y instead of T^f . Note that since Theorem 4 guarantees only local properties, Assumption 1 could be relaxed to *local* exponential stability of the nominal plant, over a sufficiently large domain.

The presented formulation of the MATI and the expression for the radius of convergence are based on [16] where examples showing that the MATI constitutes an improvement over the previously existing state-of-the-art can be found.

V. NETWORK-IN-THE-LOOP EXPERIMENTS

In this section, we address a network-in-the-loop experiment and show that if the state of the dynamic controller is not reconstructed, the control law is not able to stabilize the plant.

The plant, namely a Furuta pendulum [19] is depicted in Figure 2 and its parameters are listed in Table I.

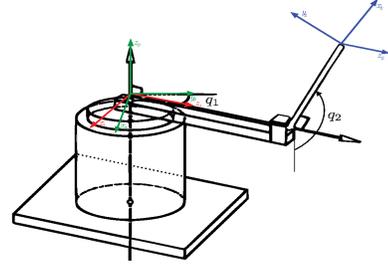


Fig. 2. Schematic representation of Furuta pendulum

TABLE I
PARAMETERS OF FURUTA PENDULUM

Physical quantity	Symbol	Value	Units
Arm mass	m_1	200×10^{-3}	kg
Pendulum mass	m_2	72×10^{-3}	kg
Arm length	L_1	224×10^{-3}	m
Arm COM	l_1	144×10^{-3}	m
Pendulum COM	l_2	106×10^{-3}	m
Arm z_0 inertia	J_{z_0}	0.9×10^{-3}	kg m ²
Pendulum x_2 inertia	J_{x_2}	1.65×10^{-6}	kg m ²
Pendulum y_2 inertia	J_{y_2}	2.7×10^{-4}	kg m ²
Pendulum z_2 inertia	J_{z_2}	2.71×10^{-4}	kg m ²
Arm friction	c_1	0.9×10^{-2}	N m s
Pendulum friction	c_2	2.71×10^{-7}	N m s
Motor torque constant	K	2.2274	N m A ⁻¹
Motor inductance	L_a	0.044	H
Motor resistance	R_a	1.9	Ω

The vector of state variables is represented by $q = [q_1, q_2]^T$, where q_1 is the angular position of the arm and q_2 is the angular position of the pendulum. Only the arm joint is actuated by means of the torque τ . The dynamics of the nonlinear plant is given by:

$$\begin{aligned} &\begin{bmatrix} \pi_1 + \pi_2 \sin^2 q_2 + \pi_3 \cos^2 q_2 & \pi_4 \cos q_2 \\ \pi_4 \cos q_2 & \pi_7 \end{bmatrix} \ddot{q} + \\ &\begin{bmatrix} \pi_6 + \pi_5 \dot{q}_2 \sin 2q_2 & -\pi_4 \dot{q}_2 \sin q_2 + \pi_5 \dot{q}_1 \sin 2q_2 \\ -\pi_5 \sin(2q_2) \dot{q}_1 & \pi_8 \end{bmatrix} \dot{q} + \\ &\begin{bmatrix} 0 \\ \pi_9 \sin q_2 \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \end{aligned} \quad (38)$$

where the quantities π_i represent the dynamic parameters of the system, which are defined, according to the mechanical parameters in Table I, as follows:

$$\begin{aligned} \pi_1 &= J_{z_0} + m_1 l_1^2 + m_2 L_1^2 & \pi_2 &= J_{y_2} + m_2 l_2^2 & \pi_3 &= J_{x_2} \\ \pi_5 &= \frac{1}{2} (J_{y_2} - J_{x_2} + m_2 l_2^2) & \pi_4 &= m_2 L_1 l_2 & \pi_6 &= c_1 \\ \pi_7 &= J_{z_2} + m_2 l_2^2 & \pi_8 &= c_2 & \pi_9 &= m_2 l_2 g \end{aligned}$$

where g is the gravity.

The torque τ is generated by the following first order linear dynamics, representing the model of a DC motor:

$$L_a \dot{\tau} = KV - R_a \tau - K^2 \dot{q}_1 \quad (39)$$

where V is the voltage applied to the motor and K, L_a, R_a are the motor parameters described in Table I.

The adopted dynamic control law is:

$$C(s) = \begin{bmatrix} \frac{-4696.5(s-2000)(s+2011)(s+0.01525)}{(s+1300)(s^2+4612s+6.329 \cdot 10^6)} \\ \frac{-3169.4(s-2000)(s-225.6)(s+9.466)}{(s+1300)(s^2+4612s+6.329 \cdot 10^6)} \end{bmatrix}^T, \quad (40)$$

whose input is the vector $[q_1, q_2 - \pi]^T$.

The experiment setup uses two computers, one for the controller and the other for simulating the plant with sensors and actuators. The same dynamics are used for the model and the plant (perfect model hypothesis). The computers are connected through a real Ethernet link. The experimental network setup is such that $1 \times 10^{-3} \text{ s} \leq \tau_{i+1}^m - \tau_i^m \leq 10 \times 10^{-3} \text{ s}$ and $1 \times 10^{-3} \text{ s} \leq \tau_{i+1}^c - \tau_i^c \leq 10 \times 10^{-3} \text{ s}$. Based on the measurements, we can consider the maximum delays to be $T^m, T^c = \frac{\max \text{RTT}}{2} \approx 8 \times 10^{-3} \text{ s}$, where RTT is the packet round-trip-time. The delays are induced by the Ethernet network and by computation overhead, no probabilistic characterization is assumed. The experiments have been carried out by means of a software for networked control systems developed in [20]. For this purpose a module implementing the virtual layer, as in Sections III-B and III-C, has been designed.

The goal of the control is to keep the pendulum rod in the upright position ($q_2 = 2k\pi \text{ rad}$, $k \in \mathbb{Z}$ in Figure 3). The initial condition for the plant is near the equilibrium point, i.e. 10° from the upright position and all the other state variables set to zero.

Figure 3 shows the results of the experiments, the two trajectories represent the behavior of the controlled pendulum rod with and without controller state reconstruction. The absence of the controller state reconstruction means that no correction of the controller state error is done, or, equivalently, that the following compound protocol $h : \mathbb{N} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is used:

$$h \left(i, [e_p^T, e_c^T]^T \right) = \begin{bmatrix} h_s(i, e_p) \\ e_c \end{bmatrix}.$$

Experiments show that if the proposed algorithm is not used, the pendulum rod cannot be stabilized around the vertical equilibrium and it starts oscillating.

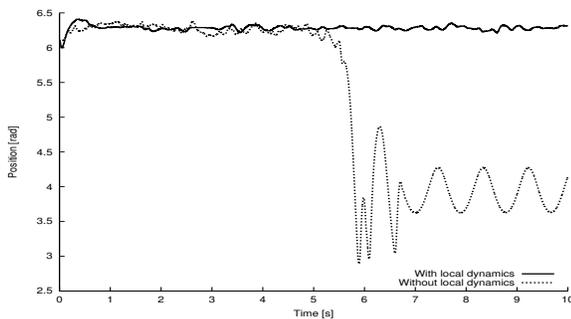


Fig. 3. Trajectories of the pendulum rod

VI. CONCLUSIONS

The problem of stabilizing a networked nonlinear plant via an output-feedback dynamic controller has been considered. Differences and difficulties in using a dynamic controller w.r.t. a static one have been underlined and an algorithm exploiting the packet-based nature of the network has been proposed. Sufficient condition for the local exponential stability of the resulting system are given. A Furuta pendulum is used to illustrate the effectiveness of the presented method. Network-in-the loop experiments show that the resulting network controlled system closely mimics the behavior of the ideal closed-loop system. If, on the contrary, the proposed algorithm is not used, the network is shown to strongly affect the performance and stability of the controlled system.

REFERENCES

- [1] R. Zurawski, "Special issue on industrial communication systems," *Proc. of IEEE*, vol. 93, no. 6, 2005.
- [2] P. Antsaklis and J. Baillieul, "Special issue on networked control systems," *IEEE Trans. on Automat. Contr.*, vol. 49, no. 9, 2004.
- [3] F.-Y. Wang and D. Liu, *Networked Control Systems: Theory and Applications*, Springer, Ed., 2008.
- [4] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. of IEEE*, vol. 95, no. 1, pp. 138–162, 2007.
- [5] W. P. M. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nešić, "Networked control systems with communication constraints: Trade-offs between transmission intervals, delays and performance," *IEEE Trans. on Automat. Contr.*, vol. 55, no. 8, pp. 1781–1796, 2010.
- [6] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Contr. Syst. Mag.*, vol. 21, no. 1, pp. 57–65, 2001.
- [7] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," *IEEE Contr. Syst. Mag.*, vol. 21, no. 1, pp. 66–83, 2001.
- [8] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Proc. IEEE Int. Conf. on Decision and Control*, vol. 2, Tampa, FL, USA, 1998, pp. 2133–2138.
- [9] D. E. Quevedo and D. Nešić, "Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts," *IEEE Trans. on Automat. Contr.*, vol. 56, no. 2, pp. 370–375, 2011.
- [10] D. Muñoz de la Peña and P. D. Christofides, "Lyapunov-based model predictive control of nonlinear systems subject to data losses," *IEEE Trans. on Automat. Contr.*, vol. 53, no. 9, pp. 2076–2089, 2008.
- [11] R. Findeisen and P. Varutti, "Stabilizing nonlinear predictive control over nondeterministic communication networks," in *Nonlinear Model Predictive Control*, ser. LNCIS. Springer Berlin / Heidelberg, 2009, vol. 384, pp. 167–179.
- [12] I. G. Polushin, P. X. Liu, and C.-H. Lung, "On the model-based approach to nonlinear networked control systems," *Automatica*, vol. 44, no. 9, pp. 2409–2414, 2008.
- [13] D. Georgiev and D. Tilbury, "Packet-based control," in *Proc. American Control Conf.*, 2004, pp. 329–336.
- [14] Y.-B. Zhao, G.-P. Liu, and D. Rees, "Design of a packet-based control framework for networked control systems," *IEEE Trans. Contr. Syst. Tech.*, vol. 17, no. 4, pp. 859–865, Jul. 2009.
- [15] L. Greco, A. Chaillet, and A. Bicchi, "Exploiting packet size in uncertain nonlinear networked control systems," *Automatica*, vol. 48, no. 11, pp. 2801–2811, 2012. [Online]. Available: <http://hal-supelec.archives-ouvertes.fr/hal-00583525/fr/>
- [16] G. C. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," in *Proc. American Control Conf.*, 1999.
- [17] D. Nešić and A. R. Teel, "Input-output stability properties of networked control systems," *IEEE Trans. on Automat. Contr.*, vol. 49, no. 10, pp. 1650–1667, 2004.
- [18] Y. Xu, M. Iwase, and K. Furuta, "Time optimal swing-up control of single pendulum," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 3, pp. 518–527, 2001.
- [19] S. Falasca, C. Belsito, A. Quagli, and A. Bicchi, "A modular and layered cosimulator for networked control systems," in *Proc. IEEE Med. Conf.*, 2010.