A Strategy for Dynamic Controller Emulation in Packet-Based Networked Control

Keywords: Networked control system; Packet-switching networks; Model-based control; Time-varying delays.

Abstract: The problem of the stabilisation of a nonlinear system via output-feedback dynamic control is addressed, under the assumption that every communication between the plant and the controller is subject to network-induced constraints. These constraints include variable transfer intervals; time varying, large communication delays; non-simultaneous access to the network. A control technique that copes with these constraints is presented which is based on recent results addressing the problem of static state-feedback. The stability of the resulting nonlinear networked control system is assessed. Network-in-the-loop experiment results are presented, which confirm that the proposed method is effective.

1 Introduction

The past decade has witnessed a dramatic growth in interest for control over distributed networked architectures, which have the strong potential to increase flexibility and scalability of a plant, while inducing a remarkable reduction of costs for both installation and maintenance. However, because of the use of the network and because of the system being distributed, some problems arise: e.g. bandwidth limitations, time-delays and packet losses, which cannot be ignored in the control design. The state-of-the-art is reported and discussed in (Heemels et al., 2010).

An essential aspect of many Networked Control Systems (NCS), such as those using Ethernet as a communication layer, is that they organise data transmission in packets. Such networks carry larger amount of data with less predictable rates with respect to circuit-switching communication channels. Packetised transmissions substantially alters the bandwidth/performance trade-off of traditional design. The potentially large size of packet payload can be exploited to reduce data transmissions without degrading the overall NCS performance. (Bemporad, 1998) pioneered the idea of sending feed-forward control sequences computed in advance on the basis of a model-based predictive (MBP) scheme to the aim of compensating for large delays in communication channels. The technique has been generalised to address time-varying delays and transfer intervals in (Polushin et al., 2008).

In (Greco et al., 2012) a control strategy (namely *Packet-Based Control*) for stabilising an uncertain nonlinear NCS affected by varying transmission intervals, varying large delays, and constrained access to the network is presented. A model of the plant is used to build a prediction of the control law. Feedback is provided by measuring the plant state. The state is measured by distributed sensors. A network protocol is in charge of deciding which sensor node can communicate at each instant. The control sequence sent by the remote controller is stored in an embedded memory on the plant side, a control command in the sequence is chosen based on the time stamp contained in the packet.

One major limitation of (Greco et al., 2012) is that it only applies to static-feedback controllers. The stabilisation of NCSs by means of dynamic controllers has been considered in (Nesic and Liberzon, 2009)–where it is addressed under the assumption of small delays–and in (Polushin et al., 2008)–where the authors solve the problem in the assumption that all the plant state is sent simultaneously. In this paper we aim at extending the *Packet-Based Control* to dynamic controllers, hence allowing for large delays and non-simultaneous transmissions. Indeed, a direct application of the aforementioned framework to the use a dynamic controller would require updating the internal state of both the system model and the controller by means of a protocol ensuring some nice error-decreasing properties.

We depart from the basic idea of updating the internal state of the dynamic controller in a way consistent with the behavior it would have had if it were directly connected to the plant. We then devise a method which produces the same effects as having a controller on the plant side which sends its internal state through the network towards the remote controller, the same way the state of the plant is sent. We consider the effects of our algorithm as virtual sendings. The only information we need to consistently run the controller is the history of the inputs to the controller. The virtual transmission of the controller state is hence realised by sending the history of the outputs of the plant and by using these outputs to feed the remote controller. The output history can be sent–again–by exploiting the large payload of packets. If output sensors are distributed, outputs are partitioned and the history of each sensor is sent according to a static protocol similar to Round Robin. The drawback of the virtual sendings–especially in the case of outputs partitioned over many nodes–is that a potentially large delay on the arrival of the virtual packets is introduced. Such a delay has to be directly taken into account in the conditions ensuring the stability of the overall system. We prove the exponential stability of the NCS over a prescribed basin of attraction, provided that some explicit bounds on the Maximum Allowable Delay (MAD (Heemels et al., 2010)) and on the Maximum Allowable Transfer Interval (MATI (Walsh et al., 1999)) are satisfied. We finally apply our technique to the control of a magnetic levitator involving an output-feedback dynamic controller. It will be shown that if the proposed technique is not used, the network strongly affects the behavior of the NCS. On the other hand, the presented algorithm closely mimic the ideal closed-loop behavior; in accordance with the paradigm adopted i.e. the presence of the network must be as transparent as possible to the designer of the stabilising controller.

2 System Description

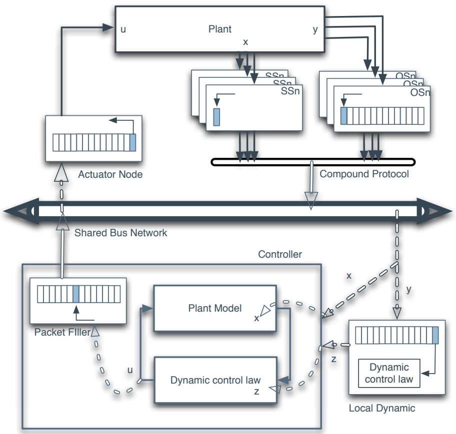


Figure 1: The proposed control architecture

In this section we provide the reader with all the characteristics we assume the controller, the network and the plant to have. Figure 1 shows the control architecture. The plant and the controller communicate via a shared-bus communication network. An output-feedback dynamic control law for the system is assumed to be available. The plant is equipped with network-enabled devices for actuation and sensing. The sensors measure the internal state of the system and its output. A protocol grants access to the network to one node at a time. Output-measuring sensors send the whole history of readings in tranches. State-measuring sensors send the current reading each time they are granted access to the network.

On the controller side, upon reception of output data, the exact knowledge of the control law is exploited in order to infer a suitable value for the internal state of the controller at a given time. The so computed internal state of the controller, together with the received information about the internal state of the plant, is used to initialise a model for the *ideal closed loop* composed by a model for the plant dynamics and the control law. By means of simulating the *ideal closed loop* behavior, a sequence of control actions is computed, which is intended to be used in the future. The control actions are then sent to the plant; they will be received on the actuator side and used appropriately.

2.1 The Plant and the Controller

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

$$\dot{x}_p = f_p(x_p, u) \tag{1}$$
$$y = g_p(x_p), \tag{2}$$

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

$$\dot{x}_c = f_c(x_c, y) \tag{3}$$
$$u = g_c(x_c, x_p, y) \tag{4}$$

is available. Here $x_c : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_c}$ is the controller state, and $f_c : \mathbb{R}^{n_c} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_c}$ and $g_c : \mathbb{R}^{n_c} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ denote locally Lipschitz functions. Letting $x(t) \triangleq \left[x_p^T(t), x_c^T(t)\right]^T \in \mathbb{R}^{n_p+n_x} = \mathbb{R}^n$ and

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

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

$$\begin{aligned} \dot{x} &= f(x,u) && (5)\\ u &= g(x). && (6) \end{aligned}$$

We assume that the nominal controller (3)-(4) globally exponentially stabilises 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* $V:\mathbb{R}^n\to\mathbb{R}_{\geq 0}$ *and constants* $\underline{\alpha},\overline{\alpha},\alpha,d>0$ *such that the following conditions hold for all* $x\in\mathbb{R}^n$

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

Local Lipschitz constants are assumed to be available to the designer.

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

$$\|f(x_1,u_1)-f(x_2,u_2)\| \leq \lambda_f\left(\|x_1-x_2\|+\|u_1-u_2\|\right) \tag{7}$$

$$\|g(x_1)-g(x_2)\| \leq \lambda_\kappa\,\|x_1-x_2\|\,. \tag{8}$$

The control strategy analysed 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:

$$\begin{aligned} \dot{\hat{x}}_p &= \hat{f}_p(\hat{x}_p,\hat{u}) && (9)\\ \hat{y} &= \hat{g}_p(\hat{x}_p). && (10) \end{aligned}$$

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

$$\begin{aligned} \dot{\hat{x}} &= \hat{f}(\hat{x},\hat{u}) && (11)\\ \hat{u} &= \hat{g}(\hat{x}), && (12) \end{aligned}$$

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

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

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* $u\in B_{R_u}$*,*

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

2.2 The Network

Measurements are taken by distributed sensors and sent to the controller into packets. Sensors are assumed to be synchronised with each other. We assume that the measurement part of the network is partitioned in ℓ nodes and only a *unique* node at a time can send its information (i.e. only partial knowledge of the plant state is available at each time instant). The controller is seen as a unique node. The overall state of the system $x(t)\in\mathbb{R}^n$ is thus decomposed in $\ell+1$ nodes as $x(t)=[x_{p,1}^T(t),\ldots,x_{p,\ell}^T(t),x_c^T(t)]^T$ with $x_{p,i}(t)\in\mathbb{R}^{p_i}$ and $\sum_{i=1}^{\ell}p_i=n_p$.

Control sequences are sent as packets. An embedded control device receives, decodes, synchronises 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 $\{\tau_i^m+T_i^m\}$. In other words, $\{T_i^m\}$ denotes the sequence of (possibly time-varying) measurement data delays. 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*[1]*:*

- **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* $T_i^c\leq T^c$*,* $\forall i\in\mathbb{N}$*;*

2.3 The Network Protocol

The use of a *dynamic* controller imposes a careful update of the controller internal model in order to generate meaningful control sequences. We propose a strategy that consists in transmitting the measurement history of each output nodes over a prescribed time horizon, as well as the instantaneous value of the plant's state when access is granted to the network. The system thus involves two different kinds of sensor nodes: ℓ_y output-sending nodes (OSn) and ℓ state-sending nodes (SSn).

---

[1] It is apparent how the discussion carried out in this paper is, by its own nature, a worst-case analysis.

The access to the network is ruled by a protocol choosing, at each instant τ_i^m, which node communicates its data. In order to limit the cumulated delays induced by this approach, we assume that the nodes are granted access to the network according to the following rule: after each SSn access, all OSn are required to send their data according to a prescribed ordering (Round Robin). 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 $\{\tau_{o_i}^m\}$ and $\{\tau_{s_i}^m\}$. More precisely, we define two sequences $\{s_i\}$, $\{o_i\}$ having values in $\mathbb{N}$. Such sequences have the following meaning: at time $\tau_s^m, s \in \{s_i\}$ a SSn is granted access to the network; at time $\tau_o^m, o \in \{o_i\}$ an OSn has the ability to send. The policy is such that the two sequences exhibit the following properties

a) $\{s_i\} \cup \{o_i\} = \mathbb{N} - \{0\}$, $\{s_i\} \cap \{o_i\} = \emptyset$;

b) $s_i = 1 + (\ell_y + 1)(i - 1)$;

c) $o_{(i+1)} = \begin{cases} 1 + o_i & \text{if } 1 + o_i \notin \{s_i\} \\ 2 + o_i & \text{otherwise} \end{cases}$.

We keep track of which OSn is granted access to the network at a given time by means of the definition of the sequence $\{\nu_i\}$ having values in $[1, \ell_y] \subset \mathbb{N}$ defined as

$$\nu_{i+1} = \begin{cases} 1 + \nu_i & \text{if } \nu_i < \ell_y \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

and we consider $\nu_0 = 1$ 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 $\tau_{o_i}^m$.

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

$$e_p(\tau_{s_i}^{m+}) = h_p\left(i, e_p(\tau_{s_i}^m)\right), \ \forall i \in \mathbb{N}. \tag{15}$$

where $h_p : \mathbb{N} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_p}$. 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 (Nesic and Teel, 2004) as given in (Greco et al., 2012).

Definition 1 *A function $h : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}^n$ is said to be an UGES protocol having parameters $\underline{a}, \overline{a}, \rho, c$ if there exists a function $W : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ locally Lipschitz in its second argument and there exist constants $\underline{a}, \overline{a} > 0; c > \underline{a}$ and $\rho \in [0, 1)$ such that the following conditions hold for the auxiliary discrete time system $\xi(i+1) = h(i, \xi(i))$:*

$$\begin{array}{c} \underline{a}\,\|\xi\| \leq W(i, \xi) \leq \overline{a}\,\|\xi\| \\ W(i+1, h(i, \xi)) \leq \rho W(i, \xi) \end{array} \tag{16}$$

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

$$\left\| \frac{\partial W}{\partial \xi}(i, \xi) \right\| \leq c \tag{17}$$

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

Assumption 5 *The protocol (15) is UGES with parameters $\underline{a}_p, \overline{a}_p, \rho_p, c_p$.*

Remark 1 *The UGES class is often used in the network control literature. Although it might seem to be very conservative, it has to be stressed out that it is not. Indeed, no limits are being posed on the rate of convergence of the error. The network protocol is used as a control-oriented description of the effects of the sending order (which is not necessarily pre-defined). UGES network protocols which are often used in practice are the Round Robin and the so called Try Once Discard.*

3 Algorithm Description

The algorithm we propose can be decomposed in different modules. At the plant side of the network, three kinds of *devices* are needed, namely the actuator node, the state-sending node and the output-sending node. The controller is divided into two modules: the first one–namely *Local Dynamics*–is in charge of converting the output data it receives from the plant into information about the internal state the controller is supposed to have; the second one has state-related information as an input (either received from the network or produced by the *Local Dynamics*); based on its input it computes the controls to be sent to the actuator node.

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

3.1 The Plant

3.1.1 Actuator Node

Such node is in charge of receiving, decoding and re-synchronise packets sent by the controller. Each received packet contains a timestamp and a certain number of control values which are stored in a local buffer. Upon choosing which one of the control values is the most appropriate, it actuates the plant. More precisely, the actuator node compares the timestamp of the last packet it received with its internal clock and moves within the control sequence it received up to the corresponding starting point.

3.1.2 State-Sensor Node

When such a node is granted access to the network (see section 2.3 for a description of the policy used to take this decision), it encodes the sensed values into a network packet, timestamps it and sends it to the controller.

3.1.3 Output-Sensor Node

An output-sensor node continuously monitors the sensed outputs, storing the readings into a buffer. The buffer content is timestamped and encoded into packets sent to the controller when the network is available. Upon sending, data is discarded. The sending of tranches of data is consistent with how control data is sent in (Greco et al., 2012).

Remark 2 *It is tacitly assumed that the effects of sending discrete-time values instead of continuous function to the controller and to the actuator node can be neglected.*

3.2 The Controller

3.2.1 Local Dynamics

By exploiting output readings it is possible to mathematically consider the controller as being executed on the plant side and sending its internal state over the network. We will define the sequence of time-instants at which such virtual-sendings happen and the delay each of these packets incurs into. Finally, we will use such virtual packets, together with the state packets, in order to design an overall UGES protocol acting on the complete system as expressed in (5).

First of all, we need give some definitions.

Definition 2 *Given $T \in \mathbb{R}_{\geq 0}$, we will say that $y_{[0,\tau]\,l}$ is known at time $\tau + T$ if there exist $j \in \{o_i\}, k \in \mathbb{N}$ such that:*

a) $\tau_j^m \geq \tau, \tau_j^m + T_j^m \leq \tau + T$;

b) $o_k = j, \nu_k = l$.

Moreover, we will say that $y_{[0,\tau]}$ is known at time $\tau + T$ if $\forall l \in [1, \ell_y] \subset \mathbb{N}$ we have that $y_{[0,\tau]\,l}$ is known at time $\tau + T$.

This definition reflects the natural notion of the controller having already received a packet related to y_l which was sent no later than time τ.

Definition 3 *Given $T \in \mathbb{R}_{\geq 0}$, we will say that $x_c(\tau)$ is known at time $\tau + T$ if either $\tau = 0$ or $y_{[0,\tau]}$ is known to the controller at time $\tau + T$.*

This definition formalises the fact that given σ such that $x_c(\sigma)$ is known at time $\tau + T$, it is possible to use the solution for the differential equation $\phi_{f_c}^{[\sigma,\tau]}$: $[\sigma,\tau] \times [\sigma,\tau] \times \mathbb{R}^{n_c} \times \mathbb{R}^{n_y}_{[\sigma,\tau]} \to \mathbb{R}^{n_c}$ in order to compute $x(\tau)$, since

$$x_c(\tau) = \phi_{f_c}^{[\sigma,\tau]}\left(\tau, \sigma, x_c(\sigma), y_{[\sigma,\tau]}\right) \tag{18}$$

With the previous definitions in mind we can state the following.

Proposition 4 (Virtual packets) *Given $i \geq \ell_y - 1, i \in \mathbb{N}$ the value $x_c\left(\tau^m_{o_{(i-(\ell_y-1))}}\right)$ is known at time $\tau^m_{o_{(i-(\ell_y-1))}} + \ell_y \tau^m + T^m$.*

This means that we can consider that at time $\tau^m_{o_{(i-(\ell_y-1))}}$ a packet containing $x_c\left(\tau^m_{o_{(i-(\ell_y-1))}}\right)$ is sent by the plant; such a virtual packet incurs a delay which is no longer than $\ell_y \tau^m + T^m$.

We will consider those virtual packets in conjunction with the packets containing $x_s(\tau^m_{s_i})$, which arrive at the controller at time $\tau^m_{s_i} + T^m_{s_i}$. The sequence of time instants at which such packets are sent is $\{\tau_i^m\}$. As for the delays, we define a new sequence $\left\{T_i^f\right\}$.

Definition 5 (Sequence of state-sending delays) *The sequence of state-sending delays $\left\{T_i^f\right\}, T_i^f \in \mathbb{R}_{\geq 0}$ such that:*

$$T_i^f = \begin{cases} T_i^m & \text{if} \exists k : i = s_k \\ \min\left\{\mathcal{K}_{\tau_i^m}\right\} & \text{otherwise} \end{cases} \tag{19}$$

where

$$\mathcal{K}_\tau \triangleq \{T : x_c(\tau) \textit{ is known at time } \tau + T\}. \tag{20}$$

By virtue of Proposition 4, the following inequality holds:

$$T_i^f \leq \ell_y \tau^m + T^m = T^f. \tag{21}$$

From now on we will consider the packets containing the state of the system and the virtual packets containing the state of the controller. The information they gather will be used in order to design a protocol which acts on the error $e(t) = (e_p^T(t), e_c^T(t))^T \triangleq \hat{x}(t) - x(t)$, where $e_c(t) \in \mathbb{R}^{n_c} : e_c(t) \triangleq \hat{x}_c(t) - x_c(t)$. We will show that, provided that an UGES protocol acting on e_p is in use (see Assumption 5), the designed protocol is UGES.

Proposition 6 (Compound protocol) *The function $h : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}^n$*

$$\left(i, \left[\xi_p^T, \xi_c^T\right]^T\right) \mapsto \begin{bmatrix} \begin{cases} h_p(k, \xi_p) & \text{if} \exists k : s_k = i \\ \xi_p & \textit{otherwise} \end{cases} \\ \begin{cases} \xi_c & \text{if} \exists k : s_k = i \\ 0 \in \mathbb{R}^{n_c} & \textit{otherwise} \end{cases} \end{bmatrix} \tag{22}$$

defines an UGES protocol. Here $\xi_p \in \mathbb{R}^{n_p}, \xi_c \in \mathbb{R}^{n_c}$.

3.2.2 Computing the Control Law

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

$$T_0^p \geq T^c + T^f + \tau^m + \tau^c \tag{23}$$

by numerically running the model (11)-(12). Such computation generates values for the function $\hat{u}(t)$ (cf. equation (12)) which are then coded, marked with the appropriate timestamp, and put in a single packet which is sent at the next network access .

3.3 The Overall Model

The loop composed of the system (1)-(2) and the controller node which executes the algorithms described in Section 3.2 can be modelled by means of the following equations (see also (Greco et al., 2012)).

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 connected to the outputs of the plant. Moreover, N vectors of additional state variables are used for modelling 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. It is defined as

$$N \triangleq \left\lceil \frac{T_0^p - \tau^m}{\varepsilon^m} \right\rceil + 1. \tag{24}$$

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

$$\dot{x} = F(t, \bar{x}, e) \tag{25a}$$
$$\dot{e} = G(t, \bar{x}, e) \tag{25b}$$
$$e(\tau_i^{m+}) = H(i, e(\tau_i^m)), \tag{25c}$$

where

$$F(t, \bar{x}, e) = f(x, \upsilon(t, e + \bar{x})) \tag{26a}$$

$$G(t, \bar{x}, e) = \begin{bmatrix} \hat{f}(e_1 + x, \hat{g}(e_1 + x)) - f(x, \upsilon(t, e + \bar{x})) \\ \vdots \\ \hat{f}(e_N + x, \hat{g}(e_N + x)) - f(x, \upsilon(t, e + \bar{x})) \end{bmatrix} \tag{26b}$$

$$H(i, e) = \begin{bmatrix} e_1 + (h(i, e_N) - e_1)\eta(i, 1) \\ e_2 + (h(i, e_1) - e_2)\eta(i, 2) \\ \vdots \\ e_N + (h(i, e_{N-1}) - e_N)\eta(i, N) \end{bmatrix}, \tag{26c}$$

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

$$\eta(i, r) \triangleq \begin{cases} 1 & \text{if } \mu(i) = r \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

and $\mu : \mathbb{N} \to \{1, \ldots, N\}$ is defined as

$$\mu(i) \triangleq ((i-1) \bmod N) + 1 . \tag{28}$$

The control signal υ in (26a) and (26b) is defined as $\upsilon : \mathbb{R}_{\geq 0} \times \mathbb{R}^{Nn} \to \mathbb{R}^{n_u}$,

$$(t, \left[\xi_1^T, \ldots, \xi_N^T\right]^T) \mapsto \sum_{k=1}^{N} \hat{g}(\xi_k)\zeta(t, k) \tag{29}$$

where $\xi_i \in \mathbb{R}^n$ and $\zeta : \mathbb{R}_{\geq 0} \times \{1, \ldots, N\} \to \{0, 1\}$ is the map

$$(t, k) \mapsto \begin{cases} 1 & \begin{array}{l} \text{if } \exists j \in \mathbb{N} \text{ s.t. } \mu(\gamma(j)) = k \text{ and} \\ t \in (\tau_j^c + T_j^c, \tau_{j+1}^c + T_{j+1}^c] \end{array} \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

and $\gamma : \mathbb{N} \to \mathbb{N}$

$$j \mapsto \max\left\{ i \in \mathbb{N} \mid \tau_i^m + T_i^f < \tau_j^c \right\} \tag{31}$$

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

4 Main Result

Theorem 7 *Assume that assumptions 1, 4, 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 $\underline{a}_p, \overline{a}_p, \rho_p, c_p$ $\underline{\alpha}, \overline{\alpha}, \alpha, d, \lambda_{f\hat{f}}, \lambda_f$ and λ_k be generated by these assumptions. Pick*

$$\underline{a} = \underline{a}_p \rho_p, \quad \overline{a} = \overline{a}_p, \quad \rho = \rho_p^{\frac{1}{\ell_y + 1}}, \quad c = c_p \tag{32}$$

and define $a_H \triangleq \overline{a}$,

$$a_L \triangleq \begin{cases} \underline{a} & \text{if} N = 1 \\ \frac{\underline{a}}{N} \min\left\{1, \left(\frac{\underline{a}}{\overline{a}}\right)^2 \frac{1}{\rho}\right\} & \text{otherwhise} \end{cases} . \tag{33}$$

Assume that the following conditions on $\tau^m, T^f, \tau^c, T^c, \varepsilon^m$ hold:

$$\begin{array}{l} \tau^m \in [\varepsilon^m, \tau^{m*}),\ \tau^{m*} \triangleq \frac{1}{L} \log\left(\frac{H\gamma_2 + a_L L}{H\gamma_2 + a_L \rho L}\right) \\ N = \left\lceil \frac{T^c + T^f + \tau^c}{\varepsilon^m} \right\rceil + 1 \end{array} \tag{34}$$

where

$$\begin{array}{rl} L \triangleq & \frac{c}{a_L}\Big((1 + \lambda_k)\sqrt{N}\lambda_{f\hat{f}} + \\ & \sqrt{N}\lambda_f + \left(\sqrt{N-1} + N - 1\right)\lambda_f \lambda_k\Big) \\ M \triangleq & (1 + \lambda_k) c N \lambda_{f\hat{f}} \\ \gamma_2 \triangleq & \frac{d}{\alpha}\sqrt{\frac{\overline{a}}{\underline{a}}}\lambda_f \lambda_k \end{array} . \tag{35}$$

Then the origin of the NCS (25) *is exponentially stable with radius of attraction*

$$\tilde{R} \triangleq \frac{R}{K} \tag{36}$$

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

Conditions expressed in (34) establish a relation between the relevant parameters, namely $\varepsilon^m, T^c, T^f, \tau^c$ and τ^m. Notice that (21) can be used to express such a relation in terms of T^m and ℓ_y instead of T^f. Note that since Theorem 7 guarantees only local properties, Assumption 1 could be relaxed to *local* exponential stability of the nominal plant, over a sufficiently large domain.

Remark 3 *The presented formulation of the MATI and the expression for the radius of convergence are based on (Greco et al., 2012) where examples showing that the MATI constitutes an improvement over the previously existing state-of-the-art can be found. It is easily seen that due to the high number of variables involved in the presented expressions, it is impractical to actually compute the MATI and the radius of convergence for a real plant. In fact, the same can be said for most of the similar results which can be found in literature. In this case the presented theorem can be seen as an* existence *result: it assesses that it is possible to stabilise the system by means of the presented architecture.*

5 Network-in-the-loop Experiments

In this section, we provide results of the experiments carried out for the networked control of a magnetic levitator. The setup uses two computers, one for the controller and the other for simulating the plant with sensors and actuators. The computers are connected through a real Ethernet link. The experimental network setup is such that

$$\begin{array}{c} 1\times 10^{-3}\,[\mathrm{s}] \le \tau^m_{i+1} - \tau^m_i \le 5\times 10^{-3}\,[\mathrm{s}] \\ 1\times 10^{-3}\,[\mathrm{s}] \le \tau^c_{i+1} - \tau^c_i \le 5\times 10^{-3}\,[\mathrm{s}] \end{array}. \tag{37}$$

Figure 2 shows the measured round-trip time. Based on the measurements, we can consider the maximum delays[2] to be $T^m, T^c = \frac{\max \mathrm{RTT}}{2} \approx 26\times 10^{-3}\,[\mathrm{s}]$.

[2]The measurements include both the network-induced delays and some additional delays which have been added via software in order to simulate the effects of additional traffic. The program `tc` has been used on both `Linux` hosts to provide additional sending delays, which have a normal probability distribution of $\mathcal{N}(15,5)$. Hence, the mean value of the added round trip time is 30ms. Additional delays account for the larger portion of the overall measured delay.

The experiments have been carried out by means of a software for networked control systems based on the one presented in (Falasca et al., 2010) which allows for network-in-the-loop tests to be carried. For this purpose a module implementing the *local dynamic* algorithm has been designed.

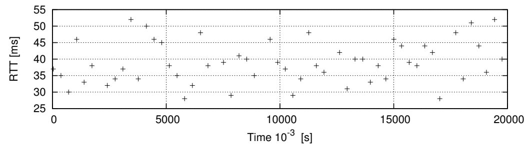
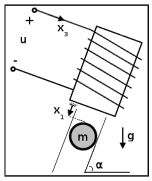


Figure 2: Round Trip Time



$$\dot{x} = \begin{bmatrix} x_2 \\ \sin(\alpha)\,g - \frac{c}{m}\left(\frac{x_3}{x_1}\right)^2 \\ -\frac{R}{L}x_3 + \frac{2c}{L}\frac{x_2 x_3}{x_1^2} + \frac{1}{L}u \end{bmatrix}$$
$$y = x_1$$

Figure 3: Magnetic levitator and its model

The plant parameters (equations are shown in Figure 3) are $\alpha = \frac{\pi}{6}$, $g = 9.8\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$, $m = 0.05\,[\mathrm{Kg}]$, $c = 0.5\,[\mathrm{Hm}]$, $L = 1\,[\mathrm{H}]$, $R = 10\,[\Omega]$.

Following the philosophy of section 2.1, we assume that a stabilising controller is given for the nominal plant. In our case, the nominal controller is the result of the straightforward application of numerical self-tuning routines in Matlab, and has the transfer function:

$$C(s) = -\frac{a_2 s^2 + a_1 s + a_0}{s^2(s+b)} \tag{38}$$

where $a_2 = 1418, a_1 = 767, a_0 = 377$ and $b = 29$.

The plant model used in the controlling computer, is subject to parametric uncertainties. The parameters for the model it uses are very different from the real ones, i.e. $\alpha = \frac{\pi}{2}$, $g = 9.8\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$, $m = 1\,[\mathrm{Kg}]$, $c = 1\,[\mathrm{Hm}]$, $L = 5\,[\mathrm{H}]$, $R = 0.1\,[\Omega]$

Figure 4 shows the results of the experiments for a reference signal $x_d = 0.05\,[\mathrm{m}]$. One of the trajectories shows the behavior of the ideal closed-loop; the second one shows the networked system. Finally, the behavior of the networked system when the algorithm taking into account local dynamics (cf. section 3.2.1) is not used is shown, i.e. the protocol $h : \mathbb{N}\times\mathbb{R}^n \to \mathbb{R}^n$

$$\left(i, \left[\xi_p^T, \xi_c^T\right]^T\right) \mapsto \begin{bmatrix} \begin{cases} h_p(k,\xi_p) & \text{if } \exists k : s_k = i \\ \xi_p & \text{otherwise} \end{cases} \\ \xi_c \end{bmatrix}$$

is used. Experiments show that the proposed algorithm manages to produce results resembling the ideal closed loop behavior. If the proposed algorithm is not used, the behavior is altered; for instance–for the given example–the steady state error is not zero.

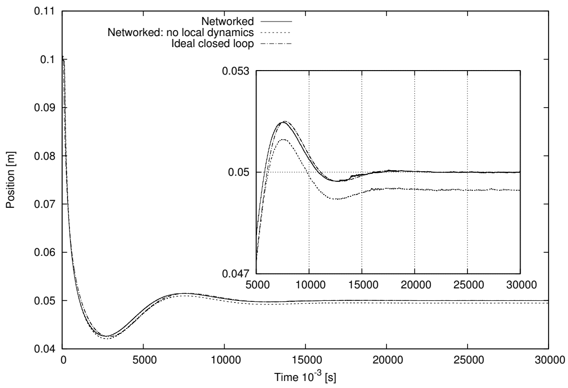


Figure 4: Trajectory $x_{1p}(t)$

6 Proofs

We start by giving a proof for Proposition 4.

Proof 1 (Virtual packets) *Pick $i \in \mathbb{N}, i \geq \ell_y - 1$ and assume that $x_c(\tau^m_{o_{i-(\ell_y-1)}})$ is not known at time $T = \tau^m_{o_{i-(\ell_y-1)}} + \ell_y\tau^m + T^m$. Then there exists $l \in [1, \ell_y]$: $y_{\left[0,\tau^m_{o_{i-(\ell_y-1)}}\right] l}$ is not known at time T. It follows that $\nu_{i-(\ell_y-1)} \neq l$ and that there exists $j < \ell_y$: $\nu_{i-(\ell_y-1)+j} = l$. Hence $y_{\left[0,\nu_{i-(\ell_y-1)+j}\right] l}$ is known at time $\tau^m_{o_{i-(\ell_y-1)+j}} + T^m_{o_{i-(\ell_y-1)+j}}$. But since $\tau^m_{o_{i-(\ell_y-1)+j}} + T^m_{o_{i-(\ell_y-1)+j}} \leq \tau^m_{o_{i-(\ell_y-1)+j}} + T^m \leq \tau^m_{o_{i-(\ell_y-1)}} + j\tau^m + T^m \leq \tau^m_{o_{i-(\ell_y-1)}} + \ell_y\tau^m + T^m = T$ we can conclude that $y_{\left[0,\nu_{i-(\ell_y-1)+j}\right] l}$ is known at time T; which is an absurd.*

In order to prove Proposition 6 we need to state some preliminary results.

Lemma 8 (Sum of Protocols) *Given two UGES protocols $h_s : \mathbb{N} \times \mathbb{R}^{n_s} \to \mathbb{R}^{n_s}$ and $h_p : \mathbb{N} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_p}$ with relative constants and functions as contained in Definition 1, distinguished by the subscript s and p, let define $e \triangleq \left[s^T, p^T\right]^T$, with $s \in \mathbb{R}^{n_s}, p \in \mathbb{R}^{n_p}$. The protocol:*

$$h(i,e) \triangleq \begin{bmatrix} h_s(i,s) \\ h_p(i,p) \end{bmatrix} \tag{39}$$

is UGES with parameters $\underline{a} = \min\left\{\underline{a}_p, \underline{a}_s\right\}$, $\overline{a} = \max\left\{\overline{a}_s, \overline{a}_p\right\}$, $\rho = \max\left\{\rho_s, \rho_p\right\}$ and $c = \max\left\{c_s, c_p\right\}$.

Proof 2 *With the parameters defined in the lemma and the Definition 1 in mind, we define the function $W : \mathbb{N} \times \mathbb{R}^{(n_s+n_p)} \to \mathbb{R}_{\geq 0}$ as*

$$W(i,e) = \sqrt{W_s(i,s)^2 + W_p(i,p)^2} \tag{40}$$

Hence

$$\underline{a}^2 \|e\|^2 = \underline{a}^2 \left(\|s\|^2 + \|p\|^2\right) \leq W(i,e)^2 \leq \overline{a}^2 \left(\|s\|^2 + \|p\|^2\right) = \overline{a}^2 \|e\|^2$$

$$W(i+1, h(i,e))^2 \leq \rho^2 \left(W_s(i,s)^2 + W_p(i,p)^2\right).$$

Furthermore

$$\left\|\frac{\partial W}{\partial e}\right\| = \frac{1}{W} \left\|\left[W_s \frac{\partial W_s}{\partial s},\, W_p \frac{\partial W_p}{\partial p}\right]\right\| = \frac{1}{W} \sqrt{\left\|W_s \frac{\partial W_s}{\partial s}\right\|^2 + \left\|W_p \frac{\partial W_p}{\partial p}\right\|^2} \leq \frac{1}{W} \sqrt{W_s^2 c_s^2 + W_p^2 c_p^2} \leq \frac{c}{W} \sqrt{W_s^2 + W_p^2} = c,$$

by means of which the theorem is proven.

Lemma 9 ($\mathbb{N}$-dilation of a protocol) *Let us consider a UGES protocol $h : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}^n$ and a not-decreasing surjective function $k : \mathbb{N} \to \mathbb{N}$, the protocol defined as:*

$$h_D(i,e) = \begin{cases} h(k(i),e) & \text{if } k(i+1) \neq k(i) \\ e & \text{otherwise} \end{cases} \tag{41}$$

is UGES, provided that $\exists \bar{\imath} : k(i) \neq k(i+\bar{\imath}) \forall i$. In particular, if h is UGES with parameters $\underline{a}, \overline{a}, \rho, c$, h_D is UGES with parameters $\rho\underline{\alpha}, \overline{\alpha}$, $\rho^{\frac{1}{\bar{\imath}}}$ and c.

Proof 3 *Given the function W associated with the UGES protocol h, we define a function W_D*

$$W_D(0,e) = W(0,e)$$

$$W_D(i+1,e) = \begin{cases} W(k(i+1),e) & \text{if } k(i+1) \neq k(i) \\ \rho^{\frac{1}{\bar{\imath}}} W_D(i,e) & \text{otherwise} \end{cases} \tag{42}$$

The following conditions hold:

$$\rho\underline{a}\|e\| \leq W_D(i,e) \leq \overline{a}\|e\|.$$

Furthermore, when $k(i+1) = k(i)$

$$W_D(i+1, h_D(i,e)) \leq \rho^{\frac{1}{\bar{\imath}}} W_D(i,e).$$

Consider now the case $k(i+1) \neq k(i)$. Take $n \leq \bar{\imath}$: $k(i) = k(i-n+1) \neq k(i-n)$. We have:

$$W_D(i-n+1,e) = W(k(i-n+1),e) \tag{43}$$

and

$$W_D(i,e) = \rho^{\frac{n-1}{\bar{\imath}}} W(k(i),e) \geq \rho^{\frac{\bar{\imath}-1}{\bar{\imath}}} W(k(i),e). \tag{44}$$

Therefore:

$$\rho W(k(i),e) \le \rho^{\frac{1}{\iota}} W_D(i,e). \tag{45}$$

We can then write $W_D(i+1, h_D(i,e)) = W(k(i+1), h(k(i),e)) \le \rho W(k(i),e) \le \rho^{\frac{1}{\iota}} W_D(i,e)$.

Since $\rho < 1$*, it is apparent from the very definition of* W_D *that*

$$\left\| \frac{\partial W(i,e)}{\partial e} \right\| \le c \Rightarrow \left\| \frac{\partial W_D(i,e)}{\partial e} \right\| \le c,$$

which concludes the proof.

We can now give a proof for Proposition 6.

Proof 4 *Define*

$$h_{Dp}(i,e_p) = \begin{cases} h_p(k,e_p) & \text{if } \exists k : s_k = i \\ e_p & \text{otherwise} \end{cases} \tag{46}$$

it is easily seen that h_{Dp} *can be obtained by applying Lemma 9 to the protocol* $h_p(i,e_p)$ *by defining*

$$k_p(i+1) = \begin{cases} k_p(i)+1 & \text{if } \exists k : s_k = i \\ k_p(i) & \text{otherwise} \end{cases}, \tag{47}$$

$k_p(0) = 0$. *From the definition of* s_i*, it follows that* $k_p(i+\ell_y+1) \neq k_p(i) \forall i \in \mathbb{N}$. *Hence* $h_{Dp}(i,e_p)$ *is UGES with parameters* $\underline{a}_{Dp} = \underline{a}_p \rho_p$, $\overline{a}_{Dp} = \overline{a}_p$, $\rho_{Dp} = \rho_p^{\frac{1}{\ell_y+1}}$ *and* $c_{Dp} = c_p$.

Define now $h_c(i,e_c) = 0$*, which is an UGES protocol having parameters* $\underline{a}_c = \overline{a}_c = c_c = \underline{a}_p$ *and* $\rho_c = \rho_p$[3]. *If we define*

$$h_{Dc}(i,e_c) = \begin{cases} e_c & \text{if } \exists k : s_k = i \\ 0 & \text{otherwise} \end{cases} \tag{48}$$

it is easily seen that h_{Dc} *can be obtained by applying Lemma 9 to the protocol* $h_c(i,e_c)$ *by defining*

$$k_c(i+1) = \begin{cases} k_c(i) & \text{if } \exists k : s_k = i \\ k_c(i)+1 & \text{otherwise} \end{cases}, \tag{49}$$

$k_c(0) = 0$. *From the definition of* o_i *and* s_i*, it follows that* $k_c(i+2) \neq k_c(i) \forall i \in \mathbb{N}$. *Hence* $h_{Dc}(i,e_c)$ *is UGES with parameters* $\underline{a}_{Dc} = \underline{a}_p \rho_p$, $\overline{a}_{Dc} = \underline{a}_p$, $\rho_{Dc} = \rho_p^{\frac{1}{2}}$ *and* $c_{Dc} = \underline{a}_p$.

Now, the protocol in Proposition 6 can be obtained by applying Lemma 8 to the protocols h_{Dp} *and* h_{Dc}*, hence it is UGES with parameters* $\underline{a} = \underline{a}_p \rho_p$, $\overline{a} = \overline{a}_p$, $\rho = \rho_p^{\frac{1}{\ell_y+1}}$ *and* $c = c_p$.

In order to prove Theorem 7, we are now going to invoke theorem 1 from (Greco et al., 2012). The assumption on the network made there are satisfied by means of (21) which follows from our Assumption 4 in conjunction with the definition of the sequence of state-sending times and delays. The assumption regarding the protocol being UGES is here satisfied by means of Proposition 6. The ideal closed-loop system has the nominal GES property, as stated in Assumption 1. Moreover, Assumptions 2 and 3 are the same required in (Greco et al., 2012).

Theorem 7 is a rewriting of Theorem 1 in (Greco et al., 2012) in terms of the quantities involved in the writing of system (25).

[3] The function $W_c(i,e_c) = \underline{a}_c \|e_c\|$ can be used to show this.

7 Conclusions

The networked stabilisation of a nonlinear plant via an output-feedback dynamic controller has been considered. An algorithm is proposed which exploits the packet-based nature of the considered network. Sufficient condition for the local exponential stability of the resulting system are given. The stabilisation of a magnetic levitator is presented as an example. Network-in-the loop experiment results 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 behavior of the controlled system.

REFERENCES

Bemporad, A. (1998). Predictive control of teleoperated constrained systems with unbounded communication delays. In *Proc. IEEE Int. Conf. on Decision and Control*, volume 2, pages 2133–2138, Tampa, FL, USA.

Falasca, S., Belsito, C., Quagli, A., and Bicchi, A. (2010). A Modular and Layered Cosimulator for Networked Control Systems. *Proc. IEEE Mediterranean Conference on Control*.

Greco, L., Chaillet, A., and Bicchi, A. (2012). Exploiting packet size in uncertain nonlinear networked control systems. *Automatica*, 48:2801–2811.

Heemels, W. P. M. H., Teel, A. R., van de Wouw, N., and Nešić, D. (2010). Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *IEEE Trans. on Automat. Contr.*, 55(8):1781–1796.

Nesic, D. and Liberzon, D. (2009). A unified framework for design and analysis of networked and quantized control systems. *Automatic Control, IEEE Transactions on*, 54(4):732 –747.

Nesic, D. and Teel, A. (2004). Input-output stability properties of networked control systems. *Automatic Control, IEEE Transactions on*, 49(10):1650 – 1667.

Polushin, I. G., Liu, P. X., and Lung, C.-H. (2008). On the model-based approach to nonlinear networked control systems. *Automatica*, 44(9):2409–2414.

Walsh, G. C., Ye, H., and Bushnell, L. (1999). Stability analysis of networked control systems. In *Proc. American Control Conf.*