Collaborative Project
Small-medium-scale focused research project (STREP)

Grant Agreement n. 224168 FP7-ICT-2007-2

WIDE

Decentralized and Wireless Control of Large-Scale Systems

Starting date: 01 September 2008 Duration: 3 years

Deliverable number	**D4.1**
Title	**Analysis methods for networked control systems**
Work package	WP4 - Network-aware control and estimation
Due date	M12
Actual submission date	01/09/2009
Lead contractor for this deliverable	TU/e
Author(s)	W.P.M.H. Heemels <W.P.M.H.Heemels@tue.nl>
With the help of	N. van de Wouw <n.v.d.wouw@tue.nl>
Revision	v1.0 (September 1st, 2009)

Dissemination Level

→ **PU**	Public
PP	Restricted to other programme participants (including the Commission Services)
RE	Restricted to a group specified by the consortium (including the Commission Services)
CO	Confidential, only for members of the consortium (including the Commission Services)

Executive summary

The presence of a communication network in a control loop induces many imperfections such as varying transmission delays, varying sampling/transmission intervals, packet loss, communication constraints and quantization effects, which can degrade the control performance significantly and even lead to instability. Various techniques have been proposed in the literature for stability analysis and controller design for these so-called networked control systems. The aim of this report is to survey the main research lines in a comprehensive manner.

Contents

abstract

The presence of a communication network in a control loop induces many imperfections such as varying transmission delays, varying sampling/transmission intervals, packet loss, communication constraints and quantization effects, which can degrade the control performance significantly and even lead to instability. Various techniques have been proposed in the literature for stability analysis and controller design for these so-called networked control systems. The aim of this report is to survey the main research lines in a comprehensive manner.

1 Introduction

Networked control systems (NCSs) have received considerable attention in recent years. The interest for NCSs is motivated by many benefits they offer such as the ease of maintenance and installation, the large flexibility and the low cost. However, still many issues need to be resolved before all the advantages of wired and wireless networked control systems can be harvested. Part of the solution will be formed by improvements of the employed communication networks and protocols, resulting in increased reliability and reduction of the end-to-end latencies and packet dropouts. However, the solution can not be obtained in a (cost-effective) manner by only improving the communication infrastructure. It is important to take a systems perspective to overcome these problems and also develop control algorithms that can deal with communication imperfections and constraints. This latter aspect is recognized widely in the control community, as evidenced by the many publications appearing recently, see e.g. the survey papers [32, 74, 64, 72].

Roughly speaking, the network-induced imperfections and constraints can be categorized in five types:

(i) Variable sampling/transmission intervals;

(ii) Variable communication delays;

(iii) Packet dropouts caused by the unreliability of the network;

(iv) Communication constraints caused by the sharing of the network by multiple nodes and the fact that only one node is allowed to transmit its packet per transmission;

(v) Quantization errors in the signals transmitted over the network due to the finite word length of the packets.

Basically, the introduction of a communication network in a control loop (see Figure 1) modifies the external signals (u, y) of the plant and the controller due to these five effects. Indeed, the control input $\hat{u}$ going into the plant is no longer equal to the output u of the controller, while the measured output of the plant y is not exactly known by the controller that only has access to a 'networked' version $\hat{y}$ of this output. Each of the imperfections have their own particular effect on the network-induced differences $\hat{y} - y$ and $\hat{u} - u$. As a consequence, the presence of these network phenomena can degrade the performance of the control loop significantly and can even lead to instability, see e.g. [8, 12] for an illustrative example. Therefore, it is of importance to understand how these phenomena influence the closed-loop stability and performance properties, preferably in a quantitative manner. Since in any practical communication network all aforementioned network-induced imperfections are present, there is a need for analysis and synthesis methods including all these imperfections. This is especially of importance, because the design of a NCS often requires tradeoffs between the different types. For instance, reducing quantization errors (and thus transmitting larger or more packets) typically results in larger transmission delays. To support the designers in making these tradeoffs to design the total NCS (plant, controller and network) in an integral fashion, tools are needed that provide quantitative

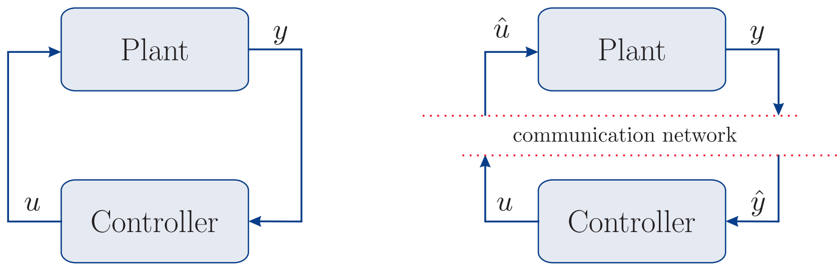

Figure 1: Introduction of a network in a control loop.

information on the consequences of each of the possible choices in plant, controller and network design.

Although the field on NCSs is relatively young, various major research lines are crystalizing out these days. However, much of the available literature on NCS considers only some of above mentioned types of network phenomena, while ignoring the other types. The available results need to be extended and integrated to obtain a framework in which all the network-induced imperfections can be studied simultaneously and tradeoffs can be made. This paper has the aim to provide an overview of the rapidly growing literature on NCS with a focus on methods for stability analysis that incorporate one or more of the above mentioned communication imperfections. To a lesser extent we will also discuss the stabilization problem. As such, this paper may form the basis for further research that eventually leads to a practically useful and complete analysis and design framework for control over communication networks.

2 Overview existing approaches

A categorization of the available literature on stability analysis of NCSs can be done, firstly, on the basis of the types of networked-induced imperfections considered (time-varying sampling intervals, time-varying delays, packet dropouts, communication constraints and quantisation), see Section 2.1, and, secondly, on the modeling and analysis approach adopted to study the stability of the NCS under these networked-induced imperfections, see Section 2.2.

Before categorizing the existing approaches, let us start by noting that two essentially different ways of modelling network-induced uncertainties, such as time-varying sampling intervals, time-varying delays and packet dropouts. Firstly, models that assume (deterministic) bounds on the delays, sampling intervals and the number of subsequent packet dropouts are often employed, without making any assumptions on the possibly random processes behind the generation of e.g. sequences of delays or packet drops. With some abuse of wording, we will call this the *deterministic* approach. Secondly, models exists in which information on the possibly stochastic nature of these variables is taken into account, provided this additional information is available, which we call the *stochastic* approach. In this overview, we focus mainly on *deterministic* approaches and refer the interested reader to one of the overview papers [32, 74, 64, 72] for stochastic approaches. One observation is that many of the stochastic approaches at present only can handle a *finite* or *countable* number of delays or sampling intervals, while in reality this is often not the case.

Another important distinction between existing network uncertainty models is whether only small delays or also large delays (delays smaller or larger, respectively, than the sampling interval) are considered. In this paper, we will consider methods dealing with both cases.

2.1 The types of networked-induced phenomena

Many systematic approaches that analyse stability of NCSs consider only one of these network-induced imperfections. Indeed, the effects of quantization are studied in [39, 63, 2, 48, 17, 26, 28], of packet dropouts in [60, 58], of time-varying transmission intervals and delays in [21, 42, 46], and [8, 12, 34, 37, 45, 73], respectively, and of communication constraints in [14, 1, 36, 57].

References that simultaneously consider two types of network-induced imperfections are given in Table 1. Moreover, [51] consider imperfections of type (i), (iv), (v), [44, 43, 7, 45, 9] study simultaneously type (i), (ii), (iii) and [52] focusses on type (i), (iii), (iv). Also [29, 31, 5, 18] studies three types, namely type (i), (ii), (iv). In addition some of the approaches mentioned in Table 1 that study varying sampling intervals and/or varying communication delays can be extended to include type (iii) phenomena as well by modeling dropouts as prolongations of the maximal sampling interval or delay (cf. also Remark 24 below). By recent unifications of the work in [51] and [29, 31] a framework is obtained in [27] that can model and analyze the five imperfections simultaneously. Although certain restrictive assumptions are adopted in [27] (e.g. the small delay case and the usage of particular quantizers), it is the first framework that includes all five of the mentioned network-induced imperfections.

Table 1: References that study NCS with two network-induced imperfections simultaneously.

&	(ii)	(iv)
(v)	[40]	
(iii)	[22, 25, 10, 41]	
(i)	[66, 65, 35]	[19, 3, 54, 69, 70, 62]

2.2 Different approaches in modelling/analysis of NCS

We distinguish three different approaches towards the modeling, stability analysis and controller synthesis for NCS:

1. Work on the discrete-time approach, see e.g [21, 22, 35, 12, 9, 66, 35, 74], has mainly focussed on linear NCS. The first step is to construct discrete-time representations of the sampled-data NCS system (which for linear systems can be done exactly), leading to an uncertain discrete-time system in which exponential uncertainties (due to network uncertainties) play a central role. The discrete-time modeling approaches can be further subcategorized by time-driven or event-driven models. In time-driven models the continuous-time model is integrated from sample/transmission time to the next sample/transmission time, while in event-driven models integration is done from each event time (being control updates times, sample times, etc), see e.g. [35] for the latter. Here we will mainly focus on time-driven linear NCS models, see Section 3.2. Next, to construct models suitable for stability analysis, polytopic overapproximation or embedding techniques are used to capture the exponential uncertainties. Various methods have been proposed to do this (some with fixed approximation error, others with tuning parameters to make the approximation more tight). The resulting polytopic models, possible including norm-bounded uncertainties, can then be used in a robust stability analysis (often based on linear matrix inequalities) to guarantee the stability of the discrete-time NCS model. The final step is to guarantee that also the intersampling behaviour is stable, such that stability of the true sampled-data NCS model can be concluded. This approach allows to consider discrete-time controllers, although by discretizing continuous-time controllers they can be incorporated as well in the analysis. Typically, this approach is applied to linear NCS since in that case *exact* discrete-time models can be derived. We will discuss this approach in more detail in Section 3.2;

2. A second approach is to model the NCS as a delay-impulsive differential equation to study the stability of the sampled-data NCS system *directly* [44, 45, 46, 66] (without considering any form of discretisation or emulation). These methods allow to consider discrete-time controllers and nonlinear plants; however, constructive stability conditions have only been obtained for linear NCS. We will discuss this approach in Section 3.3;

3. In the so-called emulation approach, see [70, 69, 52, 49, 30, 29, 14], a continuous-time controller is designed to stabilise the continuous-time plant in the absence of network-induced imperfections. Next, the stability analysis is based on a sampled-data model of the NCS (in the form of a hybrid system) and allows to quantify the level of network-induced uncertainty (e.g. the maximal allowable sampling/transmission interval and/or maximal allowable delay) for which the NCS inherits the stability properties of the closed-loop system without the network. This approach is applicable to a wide class of nonlinear NCS, since well-developed tools for the design of (nonlinear) controllers for nonlinear plants can be employed. A drawback is the fact that the controller is formulated in continuous time, whereas for NCS one typically designs the controller in discrete time. We will discuss this approach in detail in Section 4.1.

So, basically the discrete-time approach considers discrete-time controllers (or discretized continuous-time controllers) and a discrete-time NCS model, while the direct approach also considers discrete-time controllers, but has a continuous-time (sampled-data) NCS model. Finally, the emulation approach focusses on continuous-time controllers using a continuous-time (sampled-data) NCS model. Of course, within all these approaches many different techniques towards stability analysis, using common quadratic Lyapunov functions, parameter-dependent Lyapunov functions, Lyapunov-Krasovskii function(al)s, can be exploited.

3 NCS with delays, varying sampling intervals and packet loss

To capture the essence of the problems faced in networked control, we first study NCSs without the presence of communication constraints and quantization effects. In Section 4, we consider the case with communication constraints.

In Section 3.1, we discuss a general description of a single-loop NCS with time-varying sampling intervals, delays and packet dropouts. In Section 3.2, we discuss a discrete-time approach towards the modelling, stability analysis and controller design for these NCS. Finally, in Section 3.3, we present a continuous-time approach towards the modelling and stability analysis for these systems exploiting models in terms of delay-impulsive differential equations.

3.1 Description of the NCS

In this section, we present a fairly general description of a NCS including delays larger than the uncertain, and time-varying sampling interval and packet dropouts. It is based on the developments in [9] (see also [10, 12]). We choose this level of generality to show that the application of the stability techniques presented later can encompass all these types of networked-induced phenomena. Later we will focus on a simpler NCS setup (including time-varying delays only) for illustration and comparison purposes.

The NCS is depicted schematically in Figure 2. It consists of a linear continuous-time plant

$$\dot{x}(t) = Ax(t) + Bu^*(t) \tag{1}$$

with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, and a discrete-time static time-invariant controller, which are connected over a communication network that induces network delays (τ^{sc} and τ^{ca}). The state measure-

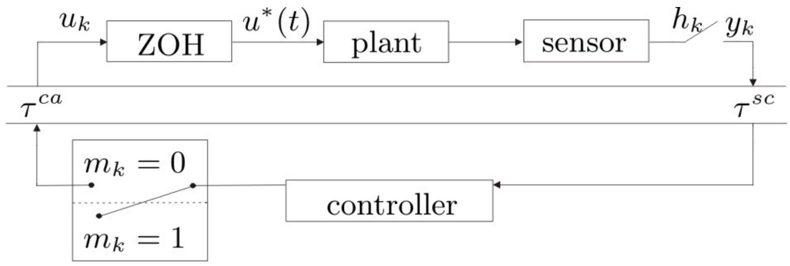

Figure 2: Schematic overview of the NCS with variable sampling intervals, network delays and packet dropouts.

ments ($y(t) = x(t)$) are sampled resulting in the sampling time instants s_k:

$$s_k = \sum_{i=0}^{k-1} h_i \quad \forall k \geq 1, \qquad s_0 = 0, \tag{2}$$

which are non-equidistantly spaced in time due to the time-varying sampling intervals $h_k > 0$. The sequence of sampling instants $s_0, s_1, s_2, \ldots$ is strictly increasing in the sense that $s_{k+1} > s_k$, for all $k \in \mathbb{N}$. We denote by $y_k := y(s_k)$ the k^{th} sampled value of y, by $x_k := x(s_k)$ the k^{th} sampled value of the state and by u_k the control value corresponding to $y_k = x_k$. Packet drops may occur (see Figure 2) and are modeled by the parameter m_k. This parameter denotes whether or not a packet is dropped:

$$m_k = \begin{cases} 0, & \text{if } y_k \text{ and } u_k \text{ are received} \\ 1, & \text{if } y_k \text{ and/or } u_k \text{ is lost.} \end{cases} \tag{3}$$

In (3), we make no distinction between packet dropouts that occur in the sensor-to-controller connection and the controller-to-actuator connection in the network. This can be justified by realizing that, for static controllers, the effect of the packet dropouts on the control updates implemented on the plant is the same in both cases. Indeed, for packet dropouts between the sensor and the controller no new control update is computed and thus no new control input is sent to the actuator. In the case of packet dropouts between the controller and the actuator no new control update is received by the actuator either. Finally, the zero-order-hold (ZOH) function (in Figure 2) is applied to transform the discrete-time control input u_k to a continuous-time control input $u^*(t)$ being the actual actuation signal of the plant.

In the model, both the varying computation time (τ_k^c), needed to evaluate the controller, and the network-induced delays, i.e. the sensor-to-controller delay (τ_k^{sc}) and the controller-to-actuator delay (τ_k^{ca}), are taken into account. We assume that the sensor acts in a time-driven fashion (i.e. sampling occurs at the times s_k defined in (2)) and that both the controller and the actuator act in an event-driven fashion (i.e. responding instantaneously to newly arrived data). Furthermore, we consider that not all the data is used due to packet dropouts and message rejection, i.e. the effect that more recent control data is available before the older data is implemented and therefore the older data is neglected. Under these assumptions, all three delays can be captured by a single delay $\tau_k := \tau_k^{sc} + \tau_k^c + \tau_k^{ca}$, see also [55], [75]. To include these effects in the continuous-time model, let us define the parameter $k^*(t)$ that denotes the index of the most recent control input that is available at time t as $k^*(t) := \max\{k \in \mathbb{N} | s_k + \tau_k \leq t \wedge m_k = 0\}$. The continuous-time model of the plant of the NCS is then given by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu^*(t) \\ u^*(t) &= u_{k^*(t)} \end{aligned} \tag{4}$$

with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$. Here, we assume that the most recent control input remains active in the plant if a packet is dropped.

We assume that the variation in the delays is bounded by $\tau_{\min}$ and $\tau_{\max}$, the variation in the sampling interval is bounded by $h_{\min}$ and $h_{\max}$ and that the number of subsequent packet dropouts is upper bounded by $\overline{\delta}$. The latter means that

$$\sum_{v=k-\overline{\delta}}^{k} m_v \leq \overline{\delta},\ \forall k, \tag{5}$$

as this guarantees that from the control inputs $u_{k-\overline{\delta}}, u_{k-\overline{\delta}+1}, \ldots, u_k$ at least one is implemented. In summary, the class $\mathcal{S}$ of admissible sequences $\{(s_k, \tau_k, m_k)\}_{k\in\mathbb{N}}$ can be described as follows:

$$\begin{aligned} \mathcal{S} := \Big\{ &\{(s_k, \tau_k, m_k)\}_{k\in\mathbb{N}} \big|\ h_{\min} \leq s_{k+1} - s_k \leq h_{\max}, \\ &s_0 = 0,\ \tau_{\min} \leq \tau_k \leq \tau_{\max},\ \sum_{v=k-\overline{\delta}}^{k} m_v \leq \overline{\delta}, \forall k \in \mathbb{N} \Big\}, \end{aligned} \tag{6}$$

which includes variable sampling intervals, large delays, and packet dropouts.

3.2 Discrete-time modeling approaches

3.2.1 The exact discrete-time NCS model

To arrive at a discrete-time description, the general description of the continuous-time control input $u^*(t)$ in (4) is reformulated to indicate explicitly which control inputs u_k are active in the sampling interval $[s_k, s_{k+1})$. Such a reformulation is needed to derive the discrete-time NCS model, which will ultimately be employed in the stability analysis and controller synthesis methods.

Lemma 1 *Consider the continuous-time NCS as defined in* (4) *and the admissible sequences of sampling instants, delays, and packet dropouts as defined in* (6)*. Define* $\underline{d} := \lfloor \frac{\tau_{\min}}{h_{\max}} \rfloor$*, the largest integer smaller than or equal to* $\frac{\tau_{\min}}{h_{\max}}$ *and* $\overline{d} := \lceil \frac{\tau_{\max}}{h_{\min}} \rceil$*, the smallest integer larger than or equal to* $\frac{\tau_{\max}}{h_{\min}}$*. Then, the control action* $u^*(t)$ *in the sampling interval* $[s_k, s_{k+1})$ *is described by*

$$u^*(t) = u_{k+j-\overline{d}-\overline{\delta}} \text{ for } t \in [s_k + t_j^k, s_k + t_{j+1}^k), \tag{7}$$

where the actuation update instants $t_j^k \in [0, h_k]$ *are defined as*

$$\begin{aligned} t_j^k = \min\Big\{ &\max\{0, \tau_{k+j-\overline{d}-\overline{\delta}} - \sum_{l=k+j-\overline{\delta}-\overline{d}}^{k-1} h_l\} + m_{k+j-\overline{d}-\overline{\delta}} h_{\max}, \\ &\max\{0, \tau_{k+j-\overline{d}-\overline{\delta}+1} - \sum_{l=k+j+1-\overline{\delta}-\overline{d}}^{k-1} h_l\} + m_{k+j-\overline{d}-\overline{\delta}+1} h_{\max}, \\ &\ldots, \max\{0, \tau_{k-\underline{d}} - \sum_{l=k-\underline{d}}^{k-1} h_l\} + m_{k-\underline{d}} h_{\max}, h_k \Big\}, \end{aligned} \tag{8}$$

with $t_j^k \leq t_{j+1}^k$ *and* $j \in \{0, 1, \ldots, \overline{d} + \overline{\delta} - \underline{d}\}$ *(see Figure 3). Moreover,* $0 = t_0^k \leq t_1^k \leq \ldots \leq t_{\overline{d}+\overline{\delta}-\underline{d}}^k \leq t_{\overline{d}+\overline{\delta}-\underline{d}+1}^k := h_k$*.*

The proof is given in [9] (see also [12, 7]). ■

Based on Lemma 1, the discrete-time NCS model can be defined as

$$x_{k+1} = e^{Ah_k} x_k + \sum_{j=0}^{\overline{d}+\overline{\delta}-\underline{d}} \int_{h_k - t_{j+1}^k}^{h_k - t_j^k} e^{As} ds B u_{k+j-\overline{d}-\overline{\delta}} \tag{9}$$

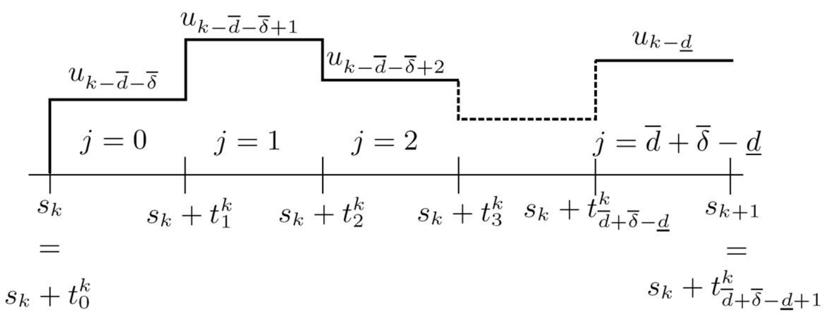

Figure 3: Graphical interpretation of the actuation update instants t_j^k.

with t_j^k as defined in Lemma 1.

Let θ_k denote the vector of uncertain parameters consisting of the sampling interval and the actuation update instants

$$\theta_k := (h_k, t_1^k, \dots, t^k_{\overline{d}+\overline{\delta}-\underline{d}}). \tag{10}$$

Remark 2 *Essentially, the uncertainty parameters $m_{k-\bar{d}-\bar{\delta}}, \dots, m_{k-\underline{\delta}}$ are included implicitly into the parameter θ_k using the formulas for the actuation update times* (8)*. When we will derive upper and lower bounds on t_j^k, this induces some conservatism if packet dropouts are present. However, the advantage of not including $m_{k-\bar{d}-\bar{\delta}}, \dots, m_{k-\underline{\delta}}$ them explicitly in θ_k is that the number of uncertainty parameters is smaller thereby reducing the complexity of the stability analysis. Alternative models for dropouts are discussed and compared in [68] (see also Remark 24 below).*

Using now the lifted state vector

$$\xi_k = \begin{pmatrix} x_k^T & u_{k-1}^T & \dots & u^T_{k-\overline{d}-\overline{\delta}} \end{pmatrix}^T$$

that includes the current system state and past system inputs, we obtain the lifted model

$$\xi_{k+1} = \tilde{A}(\theta_k)\xi_k + \tilde{B}(\theta_k)u_k, \tag{11}$$

where

$$\tilde{A}(\theta_k) = \begin{pmatrix} \Lambda(\theta_k) & M_{\overline{d}+\overline{\delta}-1}(\theta_k) & M_{\overline{d}+\overline{\delta}-2}(\theta_k) & \dots & M_1(\theta_k) & M_0(\theta_k) \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & I & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \ddots & 0 & 0 \\ 0 & \dots & \dots & 0 & I & 0 \end{pmatrix}$$

and

$$\tilde{B}(\theta_k) = \begin{pmatrix} M_{\overline{d}+\overline{\delta}}(\theta_k) \\ I \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

with $\Lambda(\theta_k) = e^{Ah_k}$ and

$$M_j(\theta_k) = \begin{cases} \displaystyle\int_{h_k - t^k_{j+1}}^{h_k - t_j^k} e^{As} ds B & \text{if } 0 \le j \le \overline{d} + \overline{\delta} - \underline{d}, \\ 0 & \text{if } \overline{d} + \overline{\delta} - \underline{d} < j \le \overline{d} + \overline{\delta}. \end{cases} \tag{12}$$

Remark 3 *In the above model set-up we adopt a time-driven modeling paradigm (exact integration from sample time to sample time). An alternative discrete-time modeling approach was proposed in [35], which uses an event-driven paradigm (integrating from event time to event time, where the event times include sampling times, update times of control values, etc.).*

3.2.2 The polytopic overapproximation

A first step towards the stability analysis is transforming the bounds on the delays, sampling intervals and dropouts (τ_{min}, τ_{max}, h_{min}, h_{max} and $\bar{\delta}$) to upper and lower bounds on t_j^k. These computations are done in [7, 9] and lead to bounds $t_{j,\min}, t_{j,\max}$, see [7, 9] for the exact expressions. Together with the fact that $h_k \in [h_{\min}, h_{\max}]$, we can define the uncertainty set

$$\begin{aligned} \Theta \;=\; \{\theta_k \in \mathbb{R}^{\overline{d}+\overline{\delta}-\underline{d}+1}\, | h_k \in [h_{\min}, h_{\max}], t_j^k \in [t_{j,\min}, t_{j,\max}], \\ 1 \leq j \leq \overline{d}+\overline{\delta}-\underline{d}, 0 \leq t_1^k \leq \ldots \leq t^k_{\overline{d}+\overline{\delta}-\underline{d}} \leq h_k\}, \end{aligned} \tag{13}$$

such that $\theta_k \in \Theta$, $\forall k$.

The stability analysis for the uncertain system (11) with the uncertainty parameter $\theta_k \in \Theta$ (given a discrete-time controller such as a lifted state feedback $u_k = -K\xi_k$) is now essentially a *robust* stability analysis problem. The obstruction to apply various robust stability techniques directly is that the uncertainty appears in an *exponential* fashion as observed from the form of $M_j(\theta_k)$ and $\Lambda(\theta_k)$. To render the formulation (11) amendable for robust stability analysis, overapproximation techniques can be employed to embed the original model (as tight as possible) in a model that has nice structural properties such as discrete-time polytopic models with (or without) additional norm-bounded uncertainties. These polytopic models are suitable for the application of available *robust* stability methods. In the literature, many different ways of constructing such polytopic embeddings of the uncertain system are proposed: overapproximation techniques are based on interval matrices [8], the real Jordan form [9, 12, 11, 10, 56], the Taylor series [34], gridding and norm-bounding [21, 61, 59, 19], and the Cayley-Hamilton theorem [23]. For the sake of brevity, we will only discuss only one of these overapproximation techniques to illustrate the ideas. We opt here to use the real Jordan form approach as adopted in [9, 12, 11, 10].

Real Jordan form To derive the stability analysis and control synthesis conditions, the model (11) is rewritten using the real Jordan form of the continuous-time system matrix A. Basically, we express the state matrix $A = TJT^{-1}$ with J the real Jordan form, and T an invertible matrix. This leads to a generic model of the form

$$\xi_{k+1} = \left(F_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k) F_i\right) \xi_k + \left(G_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k) G_i\right) u_k, \tag{14}$$

with θ_k defined in (10) and $\zeta = (\overline{d}+\overline{\delta}-\underline{d}+1)\nu$ the number of time-varying functions α_i. Here, $\nu \leq n$, where n is the dimension of the state vector x. We have $\nu = n$ when the geometric multiplicity of each distinct eigenvalue of A is equal to one and $\nu < n$ when the geometric multiplicity of an eigenvalue is larger than one. A typical function $\alpha_i(\theta_k)$ is of the form $(h_k - t_j^k)^l e^{\lambda(h_k - t_j^k)}$, when λ is a real eigenvalue of A, and of the form $(h_k - t_j^k)^l e^{a(h_k - t_j^k)} \cos(b(h_k - t_j^k))$ or $(h_k - t_j^k)^l e^{a(h_k - t_j^k)} \sin(b(h_k - t_j^k))$ when λ is a complex eigenvalue ($\lambda = a + bi$) of A with $l = 0, 1 \ldots, r_j$, where r_j is related to the size of the Jordan blocks corresponding to λ. For more details on the use of the real Jordan form to obtain the NCS model, the reader is referred to Appendix B in [7].

Using bounds on the uncertain parameters $\theta_k = (h_k, t_1^k, \ldots, t^k_{\overline{d}+\overline{\delta}-\underline{d}})$ described by the set Θ in (13) we can formulate the set of matrix pairs

$$\mathcal{FG} = \left\{ \left(F_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k) F_i, G_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k) G_i \right) \mid \theta_k \in \Theta \right\} \tag{15}$$

that contains all possible matrix combinations in (11) and in (14). Based on this infinite set $\mathcal{FG}$ of matrices we will perform stability analysis (for a given controller) and design stabilizing controllers for the NCS (4) below. To overcome the infinite dimension of the set $\mathcal{FG}$ a polytopic overapproximation of the set is used. Denote the maximum and minimum value of $\alpha_i(\theta_k)$, respectively, by

$$\overline{\alpha}_i = \max_{\theta_k \in \Theta} \alpha_i(\theta_k), \ \underline{\alpha}_i = \min_{\theta_k \in \Theta} \alpha_i(\theta_k) \tag{16}$$

with Θ defined in (13). Then the set of matrices $\mathcal{FG}$, given in (15), is a subset of $co(\mathcal{H}_{FG})$, where 'co' denotes the convex hull and $\mathcal{H}_{FG}$ is the *finite* set of matrix pairs given by

$$\mathcal{H}_{FG} = \left\{ \left((F_0 + \sum_{i=1}^{\zeta} \alpha_i F_i), (G_0 + \sum_{i=1}^{\zeta} \alpha_i G_i) \right) : \alpha_i \in \{\underline{\alpha}_i, \overline{\alpha}_i\}, i = 1, 2, \ldots, \zeta \right\}. \tag{17}$$

We will also write the set of vertices $\mathcal{H}_{FG}$ as $\mathcal{H}_{FG} = \{(H_{F,j}, H_{G,j}) \mid j = 1, 2, \ldots, 2^{\zeta}\}$ for enumeration purposes later.

3.2.3 Stability analysis

In this section, we consider the stability analysis of the NCS (11) (or equivalently (14)) in closed-loop with a state-feedback controller. From the control design of view, when dealing with a system such as (11), it is natural to design a state feedback controller using the full state ξ_k of the underlying model (11), i.e.

$$u_k = -K\xi_k. \tag{18}$$

However, from the point of view of the NCS (4), this is equivalent to using a dynamical controller of the form

$$u_k = -K_0 x_k - K_1 u_{k-1} \ldots - K_{\overline{d}+\overline{\delta}} u_{k-\overline{d}-\overline{\delta}}.$$

The use of such a dynamic control law requires a reconsideration of the assumption made earlier to lump all the delays τ_k^{sc},τ_k^{c} and τ_k^{ca} in one parameter τ. It actually leads to very restrictive assumptions on the network setup: no packet dropout is allowed between the sensor and the controller and y_k should always arrive at the controller after the moment that u_{k-1} is sent to the actuator, i.e. $s_k + \tau_k^{sc} > s_{k-1} + \tau_{k-1}^{sc} + \tau_{k-1}^{c}$. For example, in the case of a packet dropout, it is possible that $y_k = x_k$ does not arrive at the controller and thus u_k cannot be computed with the consequence that the controller (18) cannot be updated beyond the k-th update. Therefore, a deadlock in the controller can occur and the worst case scenario would be not sending control updates at all to the actuator (although one could propose heuristic solutions to overcome this situation, which would complicate the structure of the controller and its analysis and synthesis). However, observe that a static state feedback of the form

$$u_k = -\bar{K} x_k = -\begin{bmatrix} \bar{K} & 0 \end{bmatrix} \xi_k =: -K\xi_k \tag{19}$$

does not suffer from such problems and these assumptions are not needed, which greatly enhances its applicability. For this reason, in the controller synthesis section we will focus on the design of a controller in the form (19), and we will provide references for the case of the lifted state feedback as in (18). Note that the state feedback in (19) requires the design of a *structured* feedback gain $K = \begin{bmatrix} \bar{K} & 0 \end{bmatrix}$, which is known to be a notoriously difficult problem. After the stability conditions derived next, we will provide a solution for this hard problem. In the stability analysis below we assume implicitly that in case the lifted state feedback controller is used the more restrictive assumptions on the network setup as mentioned above are satisfied.

The resulting closed-loop system (11), (18) can be formulated as follows:

$$\xi_{k+1} = \tilde{A}_{cl}(\theta_k)\xi_k \text{ with } \tilde{A}_{cl}(\theta_k) = \left(\tilde{A}(\theta_k) - \tilde{B}(\theta_k)K\right), \quad \theta_k \in \Theta, \tag{20}$$

or equivalently, after exploiting the real Jordan form as in (14), as

$$\xi_{k+1} = F_{cl}(\theta_k)\xi_k, \tag{21}$$

with

$$F_{cl}(\theta_k) = \left[(F_0 - G_0K) + \sum_{i=1}^{\zeta} \alpha_i(\theta_k)\,(F_i - G_iK)\right]\xi_k, \tag{22}$$

where $F_{cl}(\theta_k) \in \mathcal{F}_{cl}$, $k \in \mathbb{N}$, and

$$\mathcal{F}_{cl} = \left\{\left(F_0 - G_0K\right) + \sum_{i=1}^{\zeta} \alpha_i(\theta_k)\left(F_i - G_iK\right)\, |\theta_k \in \Theta\right\}. \tag{23}$$

Clearly, given the fact that $\mathcal{FG} \subseteq \mathcal{H}_{FG}$, with $\mathcal{FG}$ as in (15) and $\mathcal{H}_{FG}$ as in (17), we have that

$$\mathcal{F}_{cl} \subseteq co\,(\mathcal{H}_{F_{cl}}) \tag{24}$$

with

$$\mathcal{H}_{F_{cl}} = \left\{\left(F_0 - G_0K\right) + \sum_{i=1}^{\zeta} \alpha_i\left(F_i - G_iK\right) : \alpha_i \in \{\underline{\alpha}_i, \overline{\alpha}_i\}, i = 1, 2, \ldots, \zeta\right\}. \tag{25}$$

We will also write the set of vertices $\mathcal{H}_{F_{cl}}$ as $\mathcal{H}_{F_{cl}} = \{H_{F_{cl},j} \mid j = 1, 2, \ldots, 2^{\zeta}\}$ for enumeration purposes. Hence, $\mathcal{F}_{cl} \subseteq co\{H_{F_{cl},1}, \ldots, H_{F_{cl},2^{\zeta}}\}$. Using the finite set $\mathcal{H}_{F_{cl}}$ of 2^{ζ} vertices, a finite number of LMI-based stability conditions can be formulated using [13]. The resulting stability characterisation for the closed-loop system (20) using parameter-dependent Lyapunov functions is given in the following theorem.

Theorem 4 *Consider the discrete-time NCS model* (11) *and the state-feedback controller* (18)*, with the network-induced uncertainties $\theta_k \in \Theta$ and Θ defined in* (13)*. If there exist matrices $P_j = P_j^T \succ 0$, $j = 1, 2, \ldots, 2^{\zeta}$, that satisfy*

$$H_{F_{cl},j}^T P_l H_{F_{cl},j} - P_j \prec 0, \quad j, l \in \{1, 2, \ldots, 2^{\zeta}\}, \tag{26}$$

with $H_{F_{cl},j} \in \mathcal{H}_{F_{cl}}$, $j = 1, 2, \ldots, 2^{\zeta}$, and $\mathcal{H}_{F_{cl}}$ defined in (25)*, then the origin of the closed-loop NCS system* (11)*,* (18) *is a globally asymptotically stable equilibrium point.*

The proof is a direct consequence of the results in [33, 9, 12]. ■

Remark 5 *Using the results in [50], it can be shown that under the conditions of Theorem 4 also the intersample behaviour is stable (similar reasoning is used in [8, 7]). This also implies that the equilibrium point $x = 0$ of the sampled-data NCS* (4)*,* (7)*,* (8)*,* (18) *is globally asymptotically stable.*

Remark 6 *This theorem exploits the following function*

$$V(\xi_k) = \xi_k^T P(\mu_1^k, \mu_2^k, \ldots, \mu_{2^{\zeta}}^k)\xi_k \tag{27}$$

based on the polytopic overapproximation of (21)*, which can be rewritten as*

$$\xi_{k+1} = (\sum_{j=1}^{2^{\zeta}} \mu_j^k H_{F_{cl},j})\xi_k \tag{28}$$

with $\mu_j^k \geq 0$, $j = 1, \ldots, 2^\zeta$, and $\sum_{j=1}^{2^\zeta} \mu_j^k = 1$ for $k \in \mathbb{N}$. The parameter-dependent Lyapunov function V then has the form $P(\mu_1^k, \mu_2^k, \ldots, \mu_{2^\zeta}^k) = \sum_{j=1}^{2^\zeta} \mu_j^k P_j$. In [9] it shown that if the LMIs in the above theorem are satisfied then they imply the existence of a Lyapunov-Krasovskii function (LKF) of the form

$$V(x_k, \ldots, x_{k-\overline{d}-\overline{\delta}}) = \sum_{i=0}^{\overline{d}+\overline{\delta}} \sum_{j=0}^{\overline{d}+\overline{\delta}} x_{k-i}^T Q^{i,j}(\theta_k) x_{k-j}, \tag{29}$$

which is the most general LKF that can be obtained using quadratic forms. Notice that using this approach we avoid the conservative upper bounds in the difference of the LKF, which are usually encountered in the literature to arrive at LKF-based stability conditions in LMI form.

Remark 7 *The case of a common quadratic Lyapunov function (CQLF) $V(\xi) = \xi_k^T P \xi_k$ is a particular case of this theorem by taking $P_j = P, \;\; j = 1, \ldots, 2^\zeta$.*

3.2.4 Design of stabilizing controllers

As already quickly mentioned, the main difficulty to synthesize a state feedback (19) is that it results in a structured control synthesis problem, i.e. we need to design a control law (18) with a specific structure, $K = \begin{pmatrix} \overline{K} & 0_{m\times(\overline{d}+\overline{\delta})m} \end{pmatrix}$. A solution to this structured controller synthesis problem is to apply the approach presented in [16]. Moreover, as was already exploited for the stability analysis problem above, such an approach allows for the use of a parameter-dependent Lyapunov function [13] that might result in less conservative controller synthesis results than the use of a common quadratic Lyapunov function. LMI conditions for synthesis of state feedback controllers as in (19) are given in the next theorem.

Theorem 8 *Consider the NCS model* (4)*, and* (19)*, and its discrete-time representation* (11)*,* (19) *for sequences of sampling instants, delays, and packet dropouts $\sigma \in \mathcal{S}$ with $\mathcal{S}$ as in* (6)*. Consider the equivalent representation* (14) *based on the Jordan form of A and the set of vertices $\mathcal{H}_{FG}$ defined in* (17)*.*

If there exist symmetric positive definite matrices $Y_j \in \mathbb{R}^{(n+(\overline{d}+\overline{\delta})m)\times(n+(\overline{d}+\overline{\delta})m)}$, a matrix $\overline{Z} \in \mathbb{R}^{m\times n}$, matrices $X_j = \begin{pmatrix} X_1 & 0 \\ X_{2,j} & X_{3,j} \end{pmatrix}$, with $X_1 \in \mathbb{R}^{n\times n}$, $X_{2,j} \in \mathbb{R}^{(\overline{d}+\overline{\delta})m\times n}$, $X_{3,j} \in \mathbb{R}^{(\overline{d}+\overline{\delta})m\times(\overline{d}+\overline{\delta})m}$, $j = 1, 2, \ldots, 2^\zeta$, and a scalar $0 \leq \gamma < 1$ that satisfy

$$\begin{pmatrix} X_j + X_j^T - Y_j & X_j^T H_{F,j}^T - \begin{pmatrix} \overline{Z} & 0 \end{pmatrix}^T H_{G,j}^T \\ H_{F,j} X_j - H_{G,j} \begin{pmatrix} \overline{Z} & 0 \end{pmatrix} & (1-\gamma) Y_l \end{pmatrix} > 0, \tag{30}$$

for all $j, l \in \{1, 2, \ldots, 2^\zeta\}$, then the closed-loop NCS (4) *and* (19) *with $\overline{K} = \overline{Z} X_1^{-1}$ is globally asymptotically stable for sequences of sampling instants, delays, and packet dropouts $\sigma \in \mathcal{S}$.*

For the proof, see [9].

Note that here we formulated directly the stability of the continuous-time NCS model (4) and (19) given the bounds on delays, sampling intervals and dropouts is GAS as opposed to Theorem 4 in which we formulated GAS of the discrete-time NCS model, see also Remark 5.

Remark 9 *The case of a common quadratic Lyapunov function (CQLF) $V(\xi) = \xi_k^T P \xi_k$ is a particular case of this theorem by taking $Y_j = Y, \; \forall\, j = 1, \ldots, 2^\zeta$, with $P = Y^{-1}$.*

Remark 10 *If one is still interested in using an extended state feedback* (18) *despite the mentioned disadvantages, then Theorem 8 can be modified by replacing the matrices $X_j, \forall i \neq j$ with a constant matrix X without a specific structure and using Z instead of $\begin{pmatrix} \overline{Z} & 0 \end{pmatrix}$. The extended controller is obtained then by $K = Z X^{-1}$.*

Remark 11 *The derived type of discrete-time models based on polytopic overapproximations are suitable for model predictive control (MPC) as well. For instance, the MPC techniques in [38] can be used as was indicated in [23].*

Remark 12 *The design of output-based dynamic discrete-time controllers that result in stable closed-loop NCSs is at present an unsolved problem. Basically, the problem is due to the adopted polytopic overapproximations, since a controller design problem for polytopic systems is considered to be hard problem in the literature. The stability analysis for these type of controllers (under small delay assumptions) is solved, even in the presence of communication constraints, see Section 4.2 below.*

Remark 13 *Here, we only present results on the stability and stabilization of NCSs. However, extensions exist providing constructive LMI conditions guaranteeing input-to-state stability [66, 65]. In these references the input-to-state stability property is exploited to solve the (approximate) tracking problem for linear NCS with time-varying (small) delays and time-varying sampling intervals.*

3.3 Continuous-time modeling approaches

In this section, we discuss a modelling and analysis approach for NCS with (small) delays, time-varying sampling intervals and packet dropouts as developed in [47, 46, 45]. Herein, the sampled-data NCS model is formulated in terms of so-called delay-impulsive differential equations. Before going into details, we would like to make the following observations:

- This approach studies the stability of the sampled-data NCS without exploiting any form of discretisation of a continuous-time plant model;
- The model in terms of delay-impulsive differential equations shows great similarity with the modelling of the sampled-data NCS using the hybrid systems formalism, see e.g. [52, 49], as will be discussed in Section 4.1. However, the approach described in Section 4.1 is an emulation-type approach, where controllers are designed in continuous-time, whereas here *discrete-time* controllers are considered and included directly in the sampled-data NCS model.
- The modelling framework of delay-impulsive differential equations in principle allows to consider nonlinear systems for which stability results for nonlinear delay-impulsive differential equations have been presented e.g. in [46, 45]. However, only for the case of linear NCS constructive LMI-based stability conditions have been formulated in which the assumption that the lower bound on sampling intervals $h_{min} = 0$ has to be used.

Consider the linear continuous-time plant (1) and a discrete-time static state feedback controller as in (19), i.e. $u_k = -\bar{K}x_k$. The state measurements $x_k := x(s_k)$ are sampled at the sampling instants s_k satisfying (2), which are non-equidistantly spaced in time due to the time-varying sampling intervals $h_k > 0$, with $h_k \in [h_{min}, h_{max}]\,,\ k \in \mathbb{N}$. The sequence of sampling instants $s_0, s_1, s_2, \ldots$ is strictly increasing in the sense that $s_{k+1} > s_k$, for all $k \in \mathbb{N}$. As in Section 3.1, we assume that the sensor-to-controller delay, computational delay and controller-to-actuator delay can be lumped into a single delay τ_k, with $\tau_k \in [\tau_{min}, \min\{h_k, \tau_{max}\}]\,,\ \forall k$. In other words we consider the small delay case only. Resuming, we have that $\{(s_k, \tau_k)\}_{k\in\mathbb{N}} \in \bar{\mathcal{S}}$, where

$$\begin{aligned} \bar{\mathcal{S}} := \Big\{ &\{(s_k, \tau_k)\}_{k\in\mathbb{N}} |\ h_{\min} \leq s_{k+1} - s_k \leq h_{\max}, \\ &s_0 = 0,\ \tau_k \in [\tau_{min}, \min\{s_{k+1} - s_k, \tau_{max}\}), \forall k \in \mathbb{N} \Big\} \end{aligned} \tag{31}$$

represents the admissible sequences of sampling times and delays. Packet dropouts are not considered explicitly in this approach, but can be accounted for by considering packet drops as an elongation of the effective sampling interval, see also Remark 24.

Now, the sampled-data NCS system can be formulated as

$$\begin{cases} \dot{x} &= Ax + Bu^*(t), \quad x(0) = x_0 \\ u^*(t) &= u_k, \qquad\qquad r_k \leq t \leq r_{k+1}, \\ u_k &= -\bar{K}x_k, \end{cases} \tag{32}$$

where $r_k = s_k + \tau_k$, with $r_0 = \tau_0$, is the k-th control update instant. Alternatively, the sampled-data NCS system can more compactly be formulated as

$$\dot{x} = Ax - B\bar{K}x(s_k), \quad r_k \leq t \leq r_{k+1}, \tag{33}$$

with initial condition $\bar{x}(0) = \left[x_0^T, x^T(s_{-1})\right]^T$.

Let us now write the dynamics of the NCS (32) (or (33)) in the form of a delay-impulsive differential equation of the form

$$\dot{\zeta}(t) = F\zeta(t), \qquad t \in [r_k, r_{k+1}) \tag{34a}$$

$$\zeta(r_{k+1}) = \begin{bmatrix} x(r_{k+1}) \\ x(s_{k+1}) \end{bmatrix}, \qquad k \in \mathbb{N} \tag{34b}$$

with the initial condition $\zeta(0) := \begin{bmatrix} x^T(0) & x^T(s_{-1}) \end{bmatrix}^T$, $\zeta(t) := \begin{bmatrix} x^T(t) & v_1^T(t) \end{bmatrix}^T$, $v_1(t) := x(s_k)$, for $t \in [r_k, r_{k+1})$, and

$$F := \begin{bmatrix} A & -B\bar{K} \\ 0 & 0 \end{bmatrix}.$$

Consider the following positive Lyapunov functional

$$\begin{aligned} V :=& x^T P x + \int_{t-\rho_1}^{t} (\rho_{1max} - t + s)\dot{x}^T(s)R_1\dot{x}(s)ds \\ &+ \int_{t-\rho_2}^{t} (\rho_{2max} - t + s)\dot{x}^T(s)R_2\dot{x}(s)ds + \int_{t-\tau_{min}}^{t} (\tau_{min} - t + s)\dot{x}^T(s)R_3\dot{x}(s)ds \\ &+ \int_{t-\rho_1}^{t-\tau_{min}} (\rho_{1max} - t + s)\dot{x}^T(s)R_4\dot{x}(s)ds + (\rho_{1max} - \tau_{min})\int_{t-\tau_{min}}^{t} \dot{x}^T(s)R_4\dot{x}(s)ds \\ &+ \int_{t-\tau_{min}}^{t} x^T(s)Zx(s)ds + (\rho_{1max} - \rho_1)(x - v_2)^T X(x - v_2), \end{aligned} \tag{35}$$

with P, X, Z, R_i, $i = 1, \ldots, 4$, positive definite matrices,

$$v_2(t) := x(r_k), \; \rho_1(t) := t - s_k, \; \rho_2(t) := t - r_k, \; \text{ for } \; r_k \leq t < r_{k+1},$$

and

$$\rho_{1max} := \sup_{t \geq 0} \rho_1(t), \quad \rho_{2max} := \sup_{t \geq 0} \rho_2(t).$$

The evolution of this Lyapunov functional is discontinuous at the control update times r_k, due to the jump in ζ in (34b), but a decrease of V over the jump is guaranteed by construction.

The next theorem formulates LMI-based conditions for global asymptotic stability of the NCS (34) for a sequence of sampling instants and delays taken from the class $\bar{\mathcal{S}}$ as in (31).

Theorem 14 ([47, 45]) *If there exist positive definite matrices P, X, Z, R_i, $i = 1, \ldots, 4$, and not necessarily symmetric matrices N_i, $i = 1, \ldots, 4$, satisfying the LMIs*

$$\begin{bmatrix} M_1 + (\beta - \tau_{min})(M_2 + M_3) & \tau_{max}N_1 & \tau_{min}N_3 \\ * & -\tau_{\max}R_1 & 0 \\ * & * & -\tau_{\max}R_3 \end{bmatrix} < 0, \tag{36a}$$

$$\begin{bmatrix} M_1 + (\beta - \tau_{min})M_2 & \tau_{max}N_1 & \tau_{min}N_3 & (\beta - \tau_{min})(N_1 + N_2) & (\beta - \tau_{min})N_4 \\ * & -\tau_{max}R_1 & 0 & 0 & 0 \\ * & * & -\tau_{min}R_3 & 0 & 0 \\ * & * & * & -(\beta - \tau_{min})(R_1 + R_2) & 0 \\ * & * & * & * & -(\beta - \tau_{min})R_4 \end{bmatrix} < 0, \tag{36b}$$

where $\beta := h_{max} + \tau_{max}$, $\bar{F} := \begin{bmatrix} A & -B\bar{K} & 0 & 0 \end{bmatrix}$,

$$\begin{aligned}
M_1 :=& \bar{F}^T \begin{bmatrix} P & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} P \\ 0 \\ 0 \\ 0 \end{bmatrix} \bar{F} + \tau_{min} F^T (R_1 + R_3) F - \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix} X \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix}^T + \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix} Z \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \\
& - \begin{bmatrix} 0 \\ 0 \\ 0 \\ I \end{bmatrix} Z \begin{bmatrix} 0 \\ 0 \\ 0 \\ I \end{bmatrix}^T - N_1 \begin{bmatrix} I & -I & 0 & 0 \end{bmatrix} - \begin{bmatrix} I \\ -I \\ 0 \\ 0 \end{bmatrix} N_1^T - N_2 \begin{bmatrix} I & 0 & -I & 0 \end{bmatrix} - \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix} N_2^T \\
& - N_3 \begin{bmatrix} I & 0 & 0 & -I \end{bmatrix} - \begin{bmatrix} I \\ 0 \\ 0 \\ -I \end{bmatrix} N_3^T - N_4 \begin{bmatrix} 0 & -I & 0 & I \end{bmatrix} - \begin{bmatrix} 0 \\ -I \\ 0 \\ I \end{bmatrix} N_4^T, \\
M_2 :=& \bar{F}^T (R_1 + R_2 + R_4) \bar{F}, \\
M_3 :=& \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix} X \bar{F} + \bar{F}^T X \begin{bmatrix} I & 0 & -I & 0 \end{bmatrix},
\end{aligned}$$

then, system (34) *is globally exponentially stable for any sequence of delays and sampling instants taken from the class* $\bar{S}$ *as in* (31)*.*

For the proof, we refer to [45].

Remark 15 *The proof of Theorem 14 exploits stability results for nonlinear delay-impulsive differential equations as presented in [46, 45].*

Remark 16 *We note that the conditions in Theorem 14 do not explicitly depend on the values of* $h_{\min}$*. Consequently, this approach towards modelling NCSs may result in more conservative conditions in comparison to those obtained using the discrete-time approach discussed in Section 3.2, when* $0 \ll h_{\min} \simeq h_{\max}$*.*

Remark 17 *When considering the control synthesis problem, i.e. when the control gain* $\bar{K}$ *is considered a priori unknown, the LMIs in Theorem 14 generally become bilinear matrix inequalities. However, for the case without delays in [46] LMI-based control synthesis conditions have been proposed.*

Remark 18 *In [66, 65] extensions of Theorem 14 (for the special case that* $Z = R_i = 0$*,* $i = 1, \dots, 4$*) have been proposed that guarantee input-to-state stability of the sampled-data NCS system in the face of perturbations.*

3.4 Comparing the discrete-time and direct approach on an example

We consider an example of a motion control system from the document printing domain. We limit ourselves to one single motor driving one roller-pair, as depicted in Figure 4, which obeys the dynamics:

$$\ddot{x}_s = \frac{q r_R}{J_M + q^2 J_R} u, \tag{37}$$

with $J_M = 1.95 \cdot 10^{-5} \mathrm{kgm}^2$ the inertia of the motor, $J_R = 6.5 \cdot 10^{-5}$ kgm^2 the inertia of the roller-pair, $r_R = 14 \cdot 10^{-3}$ m the radius of the roller, $q = 0.2$ the transmission ratio between motor and upper roller, x_s the sheet position and u the motor torque.

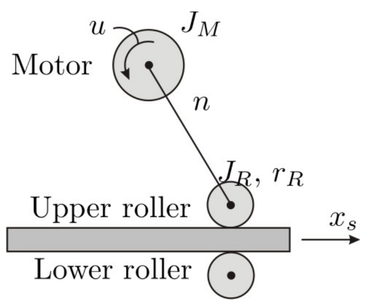

Figure 4: Schematic overview of the motor-roller example.

The continuous-time state-space representation of (37) is given by (1), with $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ b \end{bmatrix}$, with $b := \frac{qr_R}{J_M + q^2 J_R}$, and $x(t) = \begin{bmatrix} x_s(t) & \dot{x}_s(t) \end{bmatrix}^T$. We adopt a feedback controller of the form $u_k = -\bar{K}x_k$ with $\bar{K} = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$, $K_1 = 50$ and $K_2 = 1.18$.

Let us first consider the case of a constant sampling interval $h = 0.005$ s, but with time-varying and uncertain delays in the set $[0, \tau_{\max}]$. Now, we applied a variant of Theorem 14 (for the special case in which we exploit a Lyapunov functional (35) with $Z = R_i = 0$, $i = 1, \ldots, 4$) and Theorem 4 for the discrete-time approach. In [66, 65] it has been shown that this variant of Theorem 14 can be used to guarantee up to maximal delays of $\tau_{max} = 0.33h$, whereas Theorem 4 (for the special case of a common quadratic Lyapunov function) can be used to show that stability can be guaranteed up to a maximal delay of $\tau_{max} = 0.94h$.

Next, we consider the case in which the sampling interval is variable, i.e., $h \in [h_{\min}, h_{\max}]$, and the delay is zero. More specifically, we take $h_{\min} = h_{\max}/1.5$, so $h_{\min} \neq 0$. Using the discrete-time approach in Theorem 4 (for the special case of a common quadratic Lyapunov function), we can assure stability almost up to $h_{\max} = 1.34 \times 10^{-2}$ s, which is the sampling interval for which the system with a constant sampling interval (and no delay) becomes unstable. This fact shows that the proposed discrete-time stability conditions as in Theorem 4 are not conservative in this example. Using the delay impulsive approach, stability can only be guaranteed up to $h_{\max} = 9 \times 10^{-3}$ s.

Remark 19 *In [66, 65], an extension of Theorem 14 (for the special case that $Z = R_i = 0$, $i = 1, \ldots, 4$), guaranteeing input-to-state stability in the face of perturbations, is exploited to solve the (approximate) tracking problem for NCS with time-varying delays and sampling intervals. It is important to note that the input-to-state stability gains from additive perturbations to the states of the NCS provided by the delay-impulsive modelling approach are much tighter than those obtained using the discrete-time modelling and analysis approach as shown in [66, 65]. The conservatism in the input-to-state stability gain estimates in the discrete-time approach are mainly due to the conservative upperbounding of the intersample behaviour. In this respect it seems that the delay-impulsive approach is beneficial in studying such performance related issues.*

4 NCS including communication constraints

In this section we discuss stability analysis approaches that incorporate communication constraints in the sense that at a transmission time not all the information of the measured outputs of the plants and the corresponding control updates can be sent at once. At each transmission time only one of the nodes consisting of particular actuators and/or sensors will obtain access to the network to communicate its data. As we will see this complicates the description and the analysis of the NCS considerably. The communication constraint will actually introduce an (additional) discrete nature to the problem, which will require a modeling and stability analysis from a hybrid systems perspective [67].

We will discuss two approaching being distinguished by continuous-time (Section 4.1) versus discrete-time (Section 4.2) modeling as above. Both approaches have their own advantages and disadvantages as we will conclude at the end.

4.1 Continuous-time (emulation) approaches

In this section, we introduce the continuous-time model that will be used to describe NCSs including communication constraints as well as varying transmission intervals and transmission delays. Dropouts and quantization effects can be included as discussed in [27] and in Remark 24 below. The model that we discuss in this section was derived in [29, 31] and forms an extension of the NCS models used before in [52] that were motivated by the work in [70]. The basic idea behind this approach is that of emulation. In such an emulation approach, first, a stabilizing continuous-time controller for the continuous-time plant (ignoring any network effects). Next, we study under which network effects (level of delays, sampling interval lengths, type of protocol used for the communication scheduling) the networked sampled-data system inherits the stability properties from the continuous-time closed-loop system is designed.

4.1.1 Description of the NCS

We consider the continuous-time plant

$$\dot{x}_p = f_p(x_p, \hat{u}), \qquad y = g_p(x_p) \tag{38}$$

that is sampled. Here, $x_p \in \mathbb{R}^{n_p}$ denotes the state of the plant, $\hat{u} \in \mathbb{R}^{n_u}$ denotes the most recent control values available at the plant and $y \in \mathbb{R}^{n_y}$ is the output of the plant. The controller is given by

$$\dot{x}_c = f_c(x_c, \hat{y}), \qquad u = g_c(x_c), \tag{39}$$

where the variable $x_c \in \mathbb{R}^{n_c}$ is the state of the controller, $\hat{y} \in \mathbb{R}^{n_y}$ is the most recent output measurement of the plant that is available at the controller and $u \in \mathbb{R}^{n_u}$ denotes the control input. At times t_k, $k \in \mathbb{N}$, (parts of) the input u at the controller and/or the output y at the plant are sampled and transmitted over the network. The transmission times satisfy $0 \leq t_0 < t_1 < t_2 < \ldots$ and there exists a $\delta > 0$ such that the transmission intervals $t_{k+1} - t_k$ satisfy $\delta \leq t_{k+1} - t_k \leq h_{mati}$ for all $k \in \mathbb{N}$, where h_{mati} denotes the maximally allowable transmission interval (MATI). At each transmission time t_k, $k \in \mathbb{N}$, the protocol determines which of the nodes $j \in \{1, 2, \ldots, N\}$ is granted access to the network. Each node corresponds to a collection of sensors or actuators. The sensors/actuators corresponding to the node, which is granted access, collect their values in $y(t_k)$ or $u(t_k)$ that will be sent over the communication network. They will arrive after a transmission delay of τ_k time units at the controller or actuator. This results in updates of the corresponding entries in $\hat{y}$ or $\hat{u}$ at times $t_k + \tau_k$, $k \in \mathbb{N}$. The situation described above is illustrated for y and $\hat{y}$ in Fig. 5 for the situation that there are two nodes and for which the nodes get access to the network in an alternating sequence.

It is assumed that there are bounds on the maximal delay in the sense that $\tau_k \in [0, \tau_{mad}]$, $k \in \mathbb{N}$, where $0 \leq \tau_{mad} \leq h_{mati}$ is the maximally allowable delay (MAD). To be more precise, we adopt the following standing assumption.

Assumption 20 *The transmission times satisfy $\delta \leq t_{k+1} - t_k < h_{mati}$, $k \in \mathbb{N}$ and the delays satisfy $0 \leq \tau_k \leq \min\{\tau_{mad}, t_{k+1} - t_k\}$, $k \in \mathbb{N}$, where $\delta \in (0, h_{mati}]$ is arbitrary.*

The latter condition also implies that each transmitted packet arrives before the next sample is taken. Clearly, this assumption implies that only the small delay case is considered here. Extensions of this emulation approach including large delays do not exist to this date. The updates satisfy

$$\hat{y}((t_k + \tau_k)^+) = y(t_k) + h_y(k, e(t_k)) \tag{40a}$$
$$\hat{u}((t_k + \tau_k)^+) = u(t_k) + h_u(k, e(t_k)) \tag{40b}$$

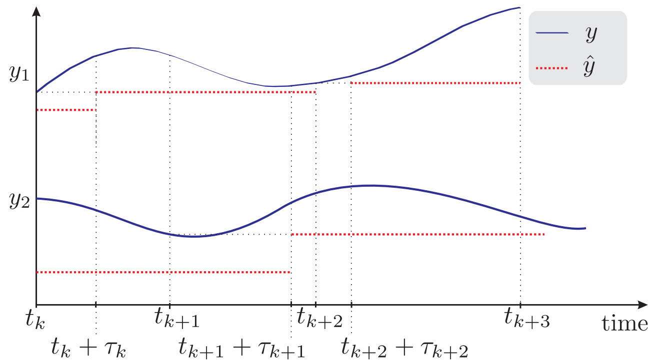

Figure 5: Illustration of a typical evolution of y and $\hat{y}$ for 2 nodes.

at $t_k + \tau_k$, where e denotes the vector (e_y, e_u) with $e_y := \hat{y} - y$ and $e_u := \hat{u} - u$. Hence, $e \in \mathbb{R}^{n_e}$ with $n_e = n_y + n_u$. If the NCS has N nodes, then the error vector e can be partitioned as $e = (e_1^T, e_2^T, \ldots, e_N^T)^T$. The functions h_y and h_u are now update functions that are related to the protocol, but typically when the j-th node gets access to the network at some transmission time t_k we have that the corresponding part in the error vector has a jump at $t_k + \tau_k$. In most situations, the jump will actually be to $y(t_k)$ (or $u(t_k)$), since we assume that the quantization effects are negligible. For instance, when y_j is transmitted at time t_k, we have that $h_{y,j}(k, e(t_k)) = 0$. However, we allow for more freedom in the protocols by allowing general functions h. Two well-known examples are the Round Robin (RR) protocol, which is given for 2 nodes by

$$h(k,e) = \begin{cases} \begin{pmatrix} 0 \\ e_2 \end{pmatrix}, & \text{if } k = 0, 2, 4, 6, \ldots \\ \begin{pmatrix} e_1 \\ 0 \end{pmatrix}, & \text{if } k = 1, 3, 5, 7, \ldots \end{cases}$$

Hence, the two nodes get access to the network in an alternating fashion: When the transmission counter is even the first node gets access, when the counter is odd the second node. The RR protocol is a *static* protocol in the sense that the order of the nodes is fixed. In contrast, there are also *dynamic* scheduling protocols such as that the Try-Once-Discard (TOD) protocol (sometimes also called the maximum-error-first protocol), which is given for two nodes by

$$h(k,e) = \begin{cases} \begin{pmatrix} 0 \\ e_2 \end{pmatrix}, & \text{if } |e_1| \geq |e_2| \\ \begin{pmatrix} e_1 \\ 0 \end{pmatrix}, & \text{if } |e_2| > |e_1|, \end{cases}$$

Here $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^n$ and we will use $\langle \cdot, \cdot \rangle$ for the corresponding inner product. Extensions of these protocols to more than 2 nodes are straightforward.

In between the updates of the values of $\hat{y}$ and $\hat{u}$, the network is assumed to operate in a zero-order-hold (ZOH) fashion, meaning that the values of $\hat{y}$ and $\hat{u}$ remain constant in between the updating times $t_k + \tau_k$ and $t_{k+1} + \tau_{k+1}$:

$$\dot{\hat{y}} = 0, \qquad \dot{\hat{u}} = 0. \tag{41}$$

To compute the resets of e at the update times $\{t_{s_i}+\tau_k\}_{k\in\mathbb{N}}$, we proceed as follows:

$$\begin{aligned}
&e_y((t_k+\tau_k)^+) = \hat{y}((t_k+\tau_k)^+) - y(t_k+\tau_k)\\
&= y(t_k) + h_y(k, e(t_k)) - y(t_k+\tau_k)\\
&= h_y(k,e(t_k)) + \underbrace{y(t_k)-\hat{y}(t_k)}_{-e(t_k)} + \underbrace{\hat{y}(t_k+\tau_k)-y(t_k+\tau_k)}_{e(t_k+\tau_k)}\\
&= h_y(k,e(t_k)) - e(t_k) + e(t_k+\tau_k).
\end{aligned}$$

In the third equality we used that $\hat{y}(t_{s_i}) = \hat{y}(t_{s_i}+\tau_k)$, which holds due to the ZOH character of the network.

A similar derivation holds for e_u, leading to the following model for the NCS:

$$\left.\begin{array}{rcl}\dot{x}(t) &=& f(x(t),e(t))\\ \dot{e}(t) &=& g(x(t),e(t))\end{array}\right\} \quad t\in[t_k, t_k+\tau_k) \tag{42a}$$

$$e((t_k+\tau_k)^+) = h(k,e(t_k)) - e(t_k) + e(t_k+\tau_k), \tag{42b}$$

where $x=(x_p,x_c)\in\mathbb{R}^{n_x}$ with $n_x = n_p+n_c$, f, g are appropriately defined functions depending on f_p, g_p, f_c and g_c and $h=(h_y,h_u)$. See [52] for the explicit expressions of f and g.

Remark 21 *The model* (42) *reduces to the model used in [52, 54] in absence of delays, i.e.* $\tau_k = 0$ *for all* $k\in\mathbb{N}$. *Indeed, then* (42) *becomes*

$$\left.\begin{array}{rcl}\dot{x}(t) &=& f(x(t),e(t))\\ \dot{e}(t) &=& g(x(t),e(t))\end{array}\right\} \quad t\in[t_{s_i}, t_{s_i}+\tau_k) \tag{43a}$$

$$e(t_k^+) = h(k,e(t_k)). \tag{43b}$$

Assumption 22 f *and* g *are continuous and* h *is locally bounded.* ■

Observe that the system $\dot{x}=f(x,0)$ is the closed-loop system (38)-(39) without the network.

The stability problem that is considered is formulated as follows.

Problem 23 *Suppose that the controller* (39) *was designed for the plant* (38) *rendering the continuous-time closed loop* (38)-(39) *(or equivalently,* $\dot{x}=f(x,0)$*) stable in some sense. Determine the value of* h_{mati} *and* τ_{mad} *so that the NCS given by* (42) *is stable as well when the transmission intervals and delays satisfy Standing Assumption 20.* ■

It is in this problem statement that we clearly recognize the fact that an emulation approach is taken here.

Remark 24 *Of course, there are certain extensions that can be made to the above setup. The inclusion of packet dropouts is relatively easy, if one models them as prolongations of the transmission interval. Indeed, if we assume that there is a bound* $\bar{\delta}\in\mathbb{N}$ *on the maximum number of successive dropouts, the stability bounds derived below are still valid for the MATI given by* $h'_{mati} := \frac{h_{mati}}{\bar{\delta}+1}$, *where* h_{mati} *is the obtained value for the dropout-free case.*

4.1.2 Reformulation in a hybrid system framework

To facilitate the stability analysis, we transform the above NCS model into the hybrid system framework as developed in [24]. To do so, we introduce the auxiliary variables $s\in\mathbb{R}^n$, $\kappa\in\mathbb{N}$, $\tau\in\mathbb{R}_{\geq 0}$ and $\ell\in\{0,1\}$ to reformulate the model in terms of so-called flow equations and reset equations. The variable s is an auxiliary variable containing the memory in (42b) storing the value $h(k,e(t_k))-e(t_k)$

for the update of e at the update instant $t_k + \tau_k$, κ is a counter keeping track of the number of the transmission, τ is a timer to constrain both the transmission interval as well as the transmission delay and ℓ is a Boolean keeping track whether the next event is a transmission event or an update event. To be precise, when $\ell = 0$ the next event will be related to transmission and when $\ell = 1$ the next event will be an update (note that here we make explicit use of the fact that only small delays are considered).

The hybrid system Σ_{NCS} is given by the flow equations

$$\left.\begin{array}{rcl} \dot{x} &=& f(x,e) \\ \dot{e} &=& g(x,e) \\ \dot{s} &=& 0 \\ \dot{\kappa} &=& 0 \\ \dot{\tau} &=& 1 \\ \dot{\ell} &=& 0 \end{array}\right\} \begin{array}{l} (\ell = 0 \wedge \tau \in [0, h_{mati}])\vee \\ \vee(\ell = 1 \wedge \tau \in [0, \tau_{mad}]) \end{array} \tag{44}$$

and the reset equations are obtained by combining the "transmission reset relations," active at the transmission instants $\{t_k\}_{k\in\mathbb{N}}$, and the "update reset relations", active at the update instants $\{t_k + \tau_k\}_{k\in\mathbb{N}}$, given by

$$(x^+, e^+, s^+, \tau^+, \kappa^+, \ell^+) = G(x, e, s, \tau, \kappa, \ell), \text{ when } \\ (\ell = 0 \wedge \tau \in [\delta, h_{mati}]) \vee (\ell = 1 \wedge \tau \in [0, \tau_{mad}]) \tag{45}$$

with G given by the transmission resets (when $\ell = 0$)

$$G(x, e, s, \tau, \kappa, 0) = (x, e, h(\kappa, e) - e, 0, \kappa + 1, 1) \tag{46}$$

and the update resets (when $\ell = 1$)

$$G(x, e, s, \tau, \kappa, 1) = (x, s + e, -s - e, \tau, \kappa, 0). \tag{47}$$

4.1.3 Lyapunov-based stability analysis

We are going to construct a Lyapunov function for Σ_{NCS} based on the following conditions for the reset part (the protocol) and the flow part of the system.

Conditions on the reset part

Condition 25 *The protocol given by h is UGES (uniformly globally exponentially stable), meaning that there exists a function $W : \mathbb{N} \times \mathbb{R}^{n_e} \to \mathbb{R}_{\geq 0}$ that is locally Lipschitz in its second argument such that*

$$\underline{\alpha}_W |e| \leq W(\kappa, e) \leq \overline{\alpha}_W |e| \tag{48a}$$

$$W(\kappa + 1, h(\kappa, e)) \leq \lambda W(\kappa, e) \tag{48b}$$

for constants $0 < \underline{\alpha}_W \leq \overline{\alpha}_W$ and $0 < \lambda < 1$. ■

Additionally we assume here that

$$W(\kappa + 1, e) \leq \lambda_W W(\kappa, e) \tag{49}$$

for some constant $\lambda_W \geq 1$ and that for almost all $e \in \mathbb{R}^{n_e}$ and all $\kappa \in \mathbb{N}$

$$\left|\frac{\partial W}{\partial e}(\kappa, e)\right| \leq M_1 \tag{50}$$

for some constant $M_1 > 0$. For all protocols discussed in [70, 69, 52, 3] such Lyapunov functions and corresponding constants exist. For instance, if N is the number of nodes in the network, for the RR protocol $\lambda_{RR} = \sqrt{\frac{N-1}{N}}$, $\underline{\alpha}_{W_{RR}} = 1$, $\overline{\alpha}_{W_{RR}} = \sqrt{N}$, $\lambda_{W_{RR}} = \sqrt{N}$, $M_{1,RR} = \sqrt{N}$ and for the TOD protocol $\lambda_{TOD} = \sqrt{\frac{N-1}{N}}$, $\underline{\alpha}_{W_{TOD}} = \overline{\alpha}_{W_{TOD}} = 1$, $\lambda_{W_{TOD}} = 1$, $M_{1,TOD} = 1$. In particular $W_{TOD}(i,e) = |e|$. See [31, 52] for the proofs.

Conditions on the flow part We also assume the following growth condition on the flow of the NCS model (42)

$$|g(x,e)| \leq m_x(x) + M_e|e|, \tag{51}$$

where $m_x : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ and $M_e \geq 0$ is a constant. Moreover, we use additionally the following.

Condition 26 *There exists a locally Lipschitz continuous function $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ satisfying the bounds*

$$\underline{\alpha}_V(|x|) \leq V(x) \leq \overline{\alpha}_V(|x|) \tag{52}$$

for some $\mathcal{K}_\infty$-functions $\underline{\alpha}_V$ and $\overline{\alpha}_V$, and the condition

$$\langle \nabla V(x), f(x,e) \rangle \leq -m_x^2(x) - \rho(|x|) + (\gamma^2 - \varepsilon) W^2(\kappa, e) \tag{53}$$

for almost all $x \in \mathbb{R}^{n_x}$ and all $e \in \mathbb{R}^{n_e}$ with $\rho \in \mathcal{K}_\infty$.

The constants in (53) satisfy $0 < \varepsilon < \max\{\gamma^2, 1\}$, where $\varepsilon > 0$ is sufficiently small.

Stability result Lumping the above parameters into four new ones given by

$$L_0 = \frac{M_1 M_e}{\underline{\alpha}_W};\ L_1 = \frac{M_1 M_e \lambda_W}{\lambda \underline{\alpha}_W};\ \gamma_0 = M_1 \gamma;\ \gamma_1 = \frac{M_1 \gamma \lambda_W}{\lambda} \tag{54}$$

we can provide the following conditions on MAD and MATI to guarantee stability of Σ_{NCS}.

Consider now the differential equations

$$\dot{\phi}_0 = -2L_0\phi_0 - \gamma_0(\phi_0^2 + 1) \tag{55a}$$

$$\dot{\phi}_1 = -2L_1\phi_1 - \gamma_0(\phi_1^2 + \frac{\gamma_1^2}{\gamma_0^2}). \tag{55b}$$

Observe that the solutions to these differential equations are strictly decreasing as long as $\phi_\ell(\tau) \geq 0$, $\ell = 0, 1$.

Define the equilibrium set as $\mathcal{E} := \{(x, e, s, \kappa, \tau, \ell) \mid x = 0,\ e = s = 0\}$

Theorem 27 *Consider the system Σ_{NCS} such that Assumptions 20 and 22 are satisfied. Let Condition 25 with* (49) *and* (50) *and Condition 26 with* (51) *hold. Suppose $h_{mati} \geq \tau_{mad} \geq 0$ satisfy*

$$\phi_0(\tau) \geq \lambda^2 \phi_1(0) \text{ for all } 0 \leq \tau \leq h_{mati} \tag{56a}$$

$$\phi_1(\tau) \geq \phi_0(\tau) \text{ for all } 0 \leq \tau \leq \tau_{mad} \tag{56b}$$

for solutions ϕ_0 and ϕ_1 of (55) *corresponding to certain chosen initial conditions $\phi_\ell(0) > 0$, $\ell = 0, 1$, with $\phi_1(0) \geq \phi_0(0) \geq \lambda^2 \phi_1(0) \geq 0$ and $\phi_0(h_{mati}) > 0$. Then for the system Σ_{NCS} the set $\mathcal{E}$ is uniformly globally asymptotically stable (UGAS).* ■

The proof is based on constructing Lyapunov functions $U(\xi)$ for Σ_{NCS}, using the solutions ϕ_0 and ϕ_1 to (55), that satisfy $U(\xi^+) \leq U(\xi)$ at reset times and $\dot{U}(\xi) < 0$ during flow. See [31] for the proof and the exact definition of UGAS.

From the above theorem quantitative numbers for h_{mati} and τ_{mad} can be obtained by constructing the solutions to (55) for certain initial conditions. By computing the τ value of the intersection of ϕ_0 and the constant line $\lambda^2\phi_1(0)$ provides h_{mati} according to (56a), while the intersection of ϕ_0 and ϕ_1 gives a value for τ_{mad} due to (56b). In Figure 6 this is illustrated. Different values of the initial conditions $\phi_0(0)$ and $\phi_1(0)$ lead, of course, to different solutions ϕ_0 and ϕ_1 of the differential equations (55) and thus different h_{mati} and τ_{mad}. As a result, tradeoff curves between h_{mati} and τ_{mad} can be obtained that indicate when stability of the NCS is still guaranteed. This will be illustrated below for the benchmark example of the batch reactor. Below we provide a systematic procedure to determine these tradeoff curves.

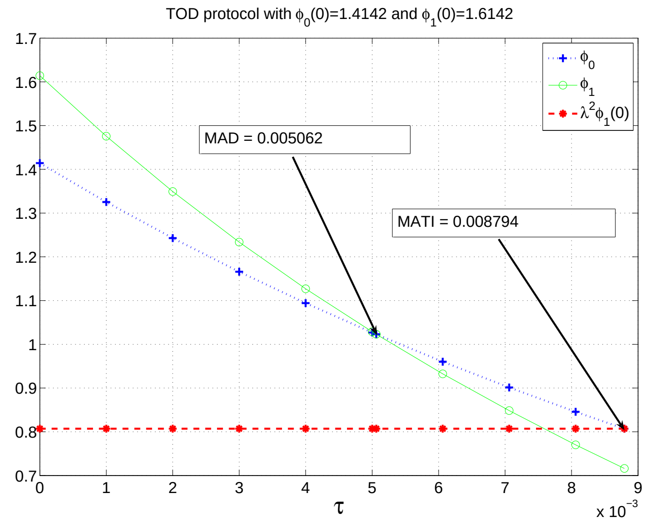

Figure 6: Illustration of how the solutions ϕ_ℓ, $\ell = 0, 1$, lead to MAD and MATI.

Systematic procedure for the determination of MATI and MAD Below we indicate the main steps in the procedure to compute the tradeoff curves between MATI and MAD.

Procedure 28 *Given Σ_{NCS} apply the following steps:*

1. *Construct a Lyapunov function W for the UGES protocol as in Condition 25 with the constants $\underline{\alpha}_W$, $\overline{\alpha}_W$, λ, λ_W and M_1 as in* (48), (49) *and* (50). *Suitable Lyapunov functions and the corresponding constants are available for many protocols in the literature [54, 52, 31];*

2. *Compute the function m_x and the constant M_e as in* (51) *bounding g as in* (42)*;*

3. *Compute for $\dot{x} = f(x, e)$ in the NCS model* (42) *the $\mathcal{L}_2$ gain from $W(\kappa, e)$ to $m_x(x)$ in the sense that* (52)*-*(53) *is satisfied for a (storage) function V for some small $0 < \varepsilon < \max\{\gamma^2, 1\}$ and $\rho \in \mathcal{K}_\infty$. When f is linear, this can be done using linear matrix inequalities (LMIs). Of course, here the 'emulated' controller should guarantee that such property is satisfied;*

4. *Use now* (54) *to obtain L_0, L_1, γ_0 and γ_1;*

5. *For initial conditions $\phi_0(0)$ and $\phi_1(0)$ with $\lambda^2\phi_1(0) \leq \phi_0(0) < \phi_1(0)$ compute (numerically) the solutions ϕ_0 and ϕ_1 to* (55) *and find (the largest values of) h_{mati} and τ_{mad} such that* (56) *are satisfied. The largest values can be found by determining the intersection of ϕ_0 and ϕ_1 (giving*

τ_{mad}) and the intersection of ϕ_0 with $\lambda^2\phi_1(0)$ (giving h_{mati}). Repeat this step for various values of the initial conditions giving various combinations of h_{mati} and τ_{mad} leading to tradeoff curves.*

This procedure is systematic in nature and can be applied in a straightforward manner.

Delay-free results The case $\tau_{mad} = 0$ has been treated before in [70, 69, 54, 52, 3]. Basically, in the least conservative of them, being [3], one uses the conditions as in (25), (49), (52), (53) and

$$\left\langle \frac{\partial W}{\partial e}(\kappa, e), f(x, e) \right\rangle \leq LW(\kappa, e) + M_1 m_x(x) \tag{57}$$

for all $\kappa \in \mathbb{N}$ and almost all $e \in \mathbb{R}^{n_e}$. Instead of four parameters as in (54) they only have the parameters γ and L next to λ to determine h_{mati} (as $\tau_{mad} = 0$). Instead of the two differential equations that are formulated in (55) there is only one

$$\dot{\phi} = -2L\phi - \gamma(\phi^2 + 1) \tag{58}$$

and they choose the initial condition $\phi(0) = \lambda^{-1}$. The conditions (56) reduce to $\phi(\tau) \geq \lambda$ for all $0 \leq \tau \leq \tau_{mad}$ to guarantee stability of Σ_{NCS}. Hence, the value of τ for which $\phi(\tau) = \lambda$ determines the h_{mati} that can be guaranteed. Interestingly, due to the fact that there is only one differential equation h_{mati} can be analytically computed and results in

$$h_{mati} = \begin{cases} \frac{1}{L_0 r} \arctan\left(\frac{r(1-\lambda)}{2\frac{\lambda}{1+\lambda}(\frac{\gamma_0}{L_0})+1+\lambda}\right), & \gamma_0 > L_0 \\ \frac{1-\lambda}{L_0(1+\lambda)}, & \gamma_0 = L_0 \\ \frac{1}{L_0 r} \text{arctanh}\left(\frac{r(1-\lambda)}{2\frac{\lambda}{1+\lambda}(\frac{\gamma_0}{L_0})+1+\lambda}\right), & \gamma_0 < L_0, \end{cases} \tag{59}$$

where $r = \sqrt{|(\frac{\gamma_0}{L_0})^2 - 1|}$.

Application to the benchmark example of the batch reactor In this part we show the potential of the discussed results for the case study of the batch reactor, which has developed over the years as a benchmark example in NCSs, see [3, 70, 52] for all the details on this example. We refer for all the technical details of the application of the above procedure to [29, 31] and show only the outcomes here.

Figure 7 shows the stability regions in terms of MAD and MATI for the TOD and the RR protocol for the batch reactor as can be proven on the basis of the above results. Interestingly, this shows tradeoff curves between MAD and MATI: a larger MAD requires a smaller MATI in order to guarantee stability. In addition, we recover exactly the delay-free results as also obtained in [3] (improving the earlier bounds in [52]), which amount for the TOD protocol to $\tau_{mad} = 0$ and $h_{mati} = 0.0108$ and for the RR protocol to $\tau_{mad} = 0$ and $\tau_{masi} = 0.0090$. Next to finding tradeoffs between MAD and MATI, different protocols can be compared with respect to each other. In Fig. 7, it is seen that for the task of stabilization of the unstable batch reactor the TOD protocol outperforms the RR protocol in the sense that it can allow for larger delays and larger transmission intervals.

Extension of these results to include guarantees on disturbance attenuation properties in the sense of $\mathcal{L}_p$ gains from disturbance inputs to certain to be controlled outputs are reported as well in [31]. In case of the batch reactor this would yield results as depicted in Figure 8 for the $\mathcal{L}_2$ gain. This picture shows tradeoffs between the network properties MAD and MATI on the one hand and control performance in terms of $\mathcal{L}_2$ gain from a specific disturbance input to a controlled output variable. These tradeoff curves are very useful for control and network designers to make well founded design decisions.

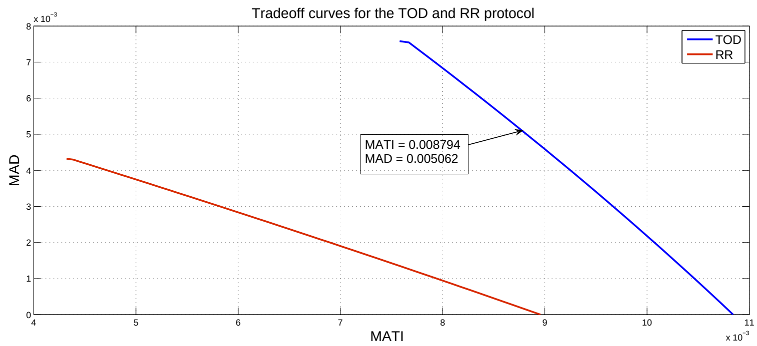

Figure 7: Tradeoff curves between MATI and MAD.

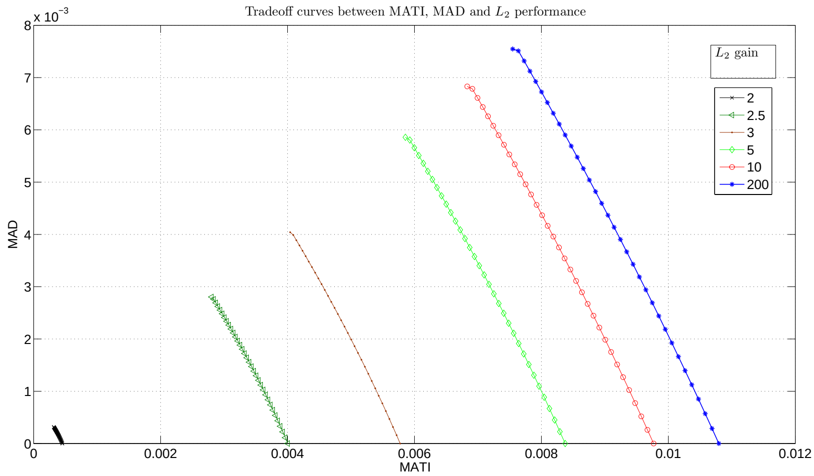

Figure 8: Tradeoff curves between MATI and MAD for various levels of the $\mathcal{L}_2$ gain of the NCS with the TOD protocol.

4.2 Discrete-time approach

The continuous-time (emulation) approach as presented in Section 4.1 applies to general *continuous-time* nonlinear plants and controllers. However, it does not include the possibility of allowing the controller to be formulated in discrete time. The case of discrete-time controllers has been considered in [15], where however, a fixed transmission interval and no delay are assumed. Another feature of the continuous-time approach is that the lower bounds on the transmission intervals h_k and delays τ_k are always equal to zero (i.e., $h_k \in (0, h_{\text{mati}}]$, $\tau_k \in (0, \tau_{\text{mad}}]$). The ability to handle discrete-time controllers and nonzero lower bounds on the transmission intervals and delays is highly relevant from a practical point of view, because controllers are typically implemented in a digital and, thus, discrete-time form. Furthermore, finite communication bandwidth always introduces nonzero lower bounds on the transmission intervals and transmission delays. The discrete-time approach surveyed here (see [19, 18]) studies these highly relevant situations as well, although be it in a *linear* context. The linearity property is exploited in the stability analysis and leads to less conservative results than the continuous-time approach. However, note that the continuous-time approach can accommodate for NCSs based on nonlinear plants and controllers and general (UGES) protocols, a feature that the discrete-time approach does not offer.

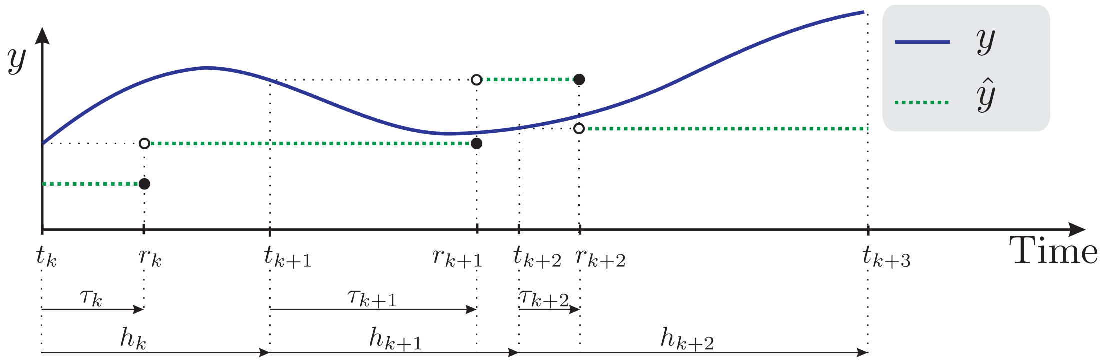

Figure 9: Illustration of a typical evolution of y and $\hat{y}$.

4.2.1 The exact discrete-time NCS model

As mentioned, the discrete-time approach applies in a *linear* context, which means that we replace (38) by the linear time-invariant (LTI) continuous-time plant given by

$$\begin{aligned}\dot{x}^p(t) &= A^p x^p(t) + B^p \hat{u}(t)\\ y(t) &= C^p x^p(t),\end{aligned} \tag{60}$$

where $x^p \in \mathbb{R}^{n_p}$ denotes the state of the plant, $\hat{u} \in \mathbb{R}^{n_u}$ the most recently received control variable, $y \in \mathbb{R}^{n_y}$ the (measured) output of the plant and $t \in \mathbb{R}^+$ the time. The controller, also an LTI system, is assumed to be given in either continuous time by

$$\begin{aligned}\dot{x}^c(t) &= A^c x^c(t) + B^c \hat{y}(t)\\ u(t) &= C^c x^c(t) + D^c \hat{y}(t),\end{aligned} \tag{61a}$$

or in discrete time by

$$\begin{aligned}x^c_{k+1} &= A^c x^c_k + B^c \hat{y}_k\\ u(t_k) &= C^c x^c_k + D^c \hat{y}(t_k).\end{aligned} \tag{61b}$$

In parallel with Section 4.1 (only subscripts becoming superscripts), $x^c \in \mathbb{R}^{n_c}$ denotes the state of the controller, $\hat{y} \in \mathbb{R}^{n_y}$ the most recently received output of the plant and $u \in \mathbb{R}^{n_u}$ denotes the controller output. At transmission instant $t_k, k \in \mathbb{N}$, (parts of) the outputs of the plant $y(t_k)$ and controller $u(t_k)$ are sampled and are transmitted over the network. We assume that they arrive at instant $r_k = t_k + \tau_k$, called the arrival instant, where τ_k denotes the communication delay. The situation described above is illustrated in Fig. 9. In the case we have a discrete-time controller (61b), the states of the controller x^c_{k+1} are updated using $\hat{y}_k := \lim_{t \downarrow r_k} \hat{y}(t)$, directly after $\hat{y}$ is updated. Note that in this case, the update of x^c_{k+1} in (61b) has to be performed in the time interval $(r_k, t_{k+1}]$.

Let us now explain in more detail the functioning of the network and define these 'most recently received' $\hat{y}$ and $\hat{u}$ exactly. As in the continuous-time (emulation) approach in Section 4.1, the plant is equipped with sensors and actuators that are grouped into N nodes. At each transmission instant t_k, $k \in \mathbb{N}$, one node, denoted by $\sigma_k \in \{1, \ldots, N\}$, obtains access to the network and transmits its corresponding values. These transmitted values are received and implemented on the controller or the plant at arrival instant r_k. As was assumed in Section 4.1, a transmission only occurs after the previous transmission has arrived, i.e., $t_{k+1} > r_k \geqslant t_k$, for all $k \in \mathbb{N}$. In other words, also here we consider the small delay case in the sense that the delay is smaller than the transmission interval $\tau_k \leq h_k := t_{k+1} - t_k$. After each transmission and reception, the values in $\hat{y}$ and $\hat{u}$ are updated, while the other values in $\hat{y}$ and $\hat{u}$ remain the same. This leads to the constrained data exchange expressed as

$$\begin{cases}\hat{y}(t) = \Gamma^y_{\sigma_k} y(t_k) + (I - \Gamma^y_{\sigma_k})\hat{y}(t_k)\\ \hat{u}(t) = \Gamma^u_{\sigma_k} u(t_k) + (I - \Gamma^u_{\sigma_k})\hat{u}(t_k)\end{cases} \tag{62}$$

for all $t \in (r_k, r_{k+1}]$, where $\Gamma_{\sigma_k} := \operatorname{diag}\left(\Gamma^y_{\sigma_k}, \Gamma^u_{\sigma_k}\right)$ is a diagonal matrix, which for each $k \in \mathbb{N}$, is taken from the set $\mathcal{G} = \{\Gamma_1, \ldots, \Gamma_N\}$ with

$$\Gamma_i = \operatorname{diag}\left(\gamma_{i,1}, \ldots, \gamma_{i,n_y+n_u}\right). \tag{63}$$

In (63), the elements $\gamma_{i,j}$, with $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, n_y\}$, are equal to one, if plant output y^j is in node i, elements $\gamma_{i,j+n_y}$, with $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, n_u\}$, are equal to one, if controller output u^j is in node i, and are zero elsewhere. Note that (62) is directly related to (40) in the continuous-time approach with $h_y(k, e_y(t_k)) = (I - \Gamma^y_{\sigma_k})e(t_k)$, $h_u(k, e(t_k)) = (I - \Gamma^u_{\sigma_k})e_u(t_k)$, $e_y(t) = \hat{y}(t) - y(t)$ and $e_u(t) = \hat{u}(t) - u(t)$.

The value of $\sigma_k \in \{1, \ldots, N\}$ in (62) indicates which node is given access to the network at transmission instant t_k, $k \in \mathbb{N}$. Indeed, (62) reflects that the values in $\hat{u}$ and $\hat{y}$ corresponding to node σ_k are updated just after r_k, with the corresponding transmitted values at time t_k, while the others remain the same. A scheduling protocol determines the sequence $(\sigma_0, \sigma_1, \ldots)$ such as the Round Robin and Try-Once-Discard protocols discussed earlier.

The transmission instants t_k, as well as the arrival instants r_k, $k \in \mathbb{N}$ are not necessarily distributed equidistantly in time. Hence, both the transmission intervals $h_k := t_{k+1} - t_k$ and the transmission delays $\tau_k := r_k - t_k$ are varying in time, as is also illustrated in Fig. 9. We assume that the variations in the transmission interval and delays are bounded and are contained in the sets $[\underline{h}, \overline{h}]$ and $[\underline{\tau}, \overline{\tau}]$, respectively, with $\overline{h} > \underline{h} > 0$ and $\overline{\tau} > \underline{\tau} \geq 0$. Since we assumed that each transmission delay τ_k is smaller than the corresponding transmission interval h_k, we have that $(h_k, \tau_k) \in \Psi$, for all $k \in \mathbb{N}$, where

$$\Psi := \left\{(h, \tau) \in \mathbb{R}^2 \mid h \in [\underline{h}, \overline{h}], \tau \in [\underline{\tau}, \min\{h, \overline{\tau}\})\right\}. \tag{64}$$

Note that in comparison with Section 4.1, h_{mati} would correspond to $\bar{h}$ and τ_{mad} to $\bar{\tau}$. However, in Section 4.1 it was assumed that $\underline{\tau} = 0$ and $\underline{h} = 0$ due to the emulation type of approach, while that is not the case here. Therefore, here the different notation is used.

To analyse stability of the NCS described above, we transform it into a discrete-time model. In this framework, we need a discrete-time equivalent of (60) and also of (61a) in case a continuous-time controller is used. To arrive at this description, let us first define the network-induced error as

$$\begin{cases} e^y(t) := \hat{y}(t) - y(t) \\ e^u(t) := \hat{u}(t) - u(t). \end{cases} \tag{65}$$

By exact discretization of (60) and/or (61a) a discrete-time switched uncertain system can be obtained that describes the evolution of the states between t_k and $t_{k+1} = t_k + h_k$. In order to do so, we define $x^p_k := x^p(t_k)$, $u_k := u(t_k)$, $\hat{u}_k := \lim_{t \downarrow r_k} \hat{u}(t)$ and $e^u_k := e^u(t_k)$. This results in three different models each describing a particular NCS. The first and the second model cover the situation where both the plant and the controller outputs are transmitted over the network, differing by the fact that the controller is given by (61a) and (61b), respectively. In the third model, it is assumed that the controller is given by (61a) and that only the plant outputs y are transmitted over the network and u are sent continuously via an ideal nonnetworked connection. We include this particular case, because it is often used in examples in NCS literature (e.g. for the benchmark example of the batch reactor [19, 53, 30, 71, 4, 15]), see also Section 4.1.

The NCS model with controller (61a) For an NCS having controller (61a), the complete NCS model is obtained by combining (62), (65) with exact discretisations of plant (60) and controller (61a) and defining

$$\bar{x}_k := \begin{bmatrix} x^{p\top}_k & x^{c\top}_k & e^{y\top}_k & e^{u\top}_k \end{bmatrix}^\top. \tag{66}$$

This results in the discrete-time model

$$\bar{x}_{k+1} = \underbrace{\begin{bmatrix} A_{h_k} + E_{h_k}BDC & E_{h_k}BD - E_{h_k-\tau_k}B\Gamma_{\sigma_k} \\ C(I - A_{h_k} - E_{h_k}BDC) & I - D^{-1}\Gamma_{\sigma_k} + C(E_{h_k-\tau_k}B\Gamma_{\sigma_k} - E_{h_k}BD) \end{bmatrix}}_{=:\tilde{A}_{\sigma_k,h_k,\tau_k}} \bar{x}_k \tag{67}$$

in which $\tilde{A}_{\sigma_k,h_k,\tau_k} \in \mathbb{R}^{n\times n}$, with $n = n_p + n_c + n_y + n_u$, and

$$A_{h_k} := \operatorname{diag}(e^{A^p h_k}, e^{A^c h_k}), \qquad B := \begin{bmatrix} 0 & B^p \\ B^c & 0 \end{bmatrix}, \tag{69a}$$

$$C := \operatorname{diag}(C^p, C^c), \qquad D := \begin{bmatrix} I & 0 \\ D^c & I \end{bmatrix}, \tag{69b}$$

$$E_\rho := \operatorname{diag}(\int_0^\rho e^{A^p s} ds, \int_0^\rho e^{A^c s} ds), \quad \rho \in \mathbb{R}. \tag{69c}$$

The NCS model with controller (61b) For an NCS having controller (61b), the complete NCS model is obtained by combining (61b), (62), (65), and an exact discretisation of the continuous-time plant (60), also resulting in (67), in which now

$$A_{h_k} := \operatorname{diag}(e^{A^p h_k}, A^c), \qquad B := \begin{bmatrix} 0 & B^p \\ B^c & 0 \end{bmatrix}, \tag{70a}$$

$$C := \operatorname{diag}(C^p, C^c), \qquad D := \begin{bmatrix} I & 0 \\ D^c & I \end{bmatrix}, \tag{70b}$$

$$E_\rho := \operatorname{diag}(\int_0^\rho e^{A^p s} ds, I), \quad \rho \in \mathbb{R}. \tag{70c}$$

The NCS model if only y is transmitted over the network In this case we assume that only the outputs of the plant are transmitted over the network and the controller communicates its values continuously and without delay. We therefore have that $u(t) = \hat{u}(t)$, for all $t \in \mathbb{R}^+$, which allows us to combine (60) and (61a), yielding

$$\begin{bmatrix} \dot{x}^p(t) \\ \dot{x}^c(t) \end{bmatrix} = \begin{bmatrix} A^p & B^pC^c \\ 0 & A^c \end{bmatrix} \begin{bmatrix} x^p(t) \\ x^c(t) \end{bmatrix} + \begin{bmatrix} B^pD^c \\ B^c \end{bmatrix} \hat{y}(t). \tag{71}$$

Since $\hat{y}$ is still updated according to (62), we can describe the evolution of the states between t_k and $t_{k+1} = t_k + h_k$ also by exact discretization. This allows us to write the complete NCS model by defining

$$\bar{x}_k := \begin{bmatrix} x_k^{p\top} & x_k^{c\top} & e_k^{y\top} \end{bmatrix}^\top, \tag{72}$$

resulting in (67), in which

$$A_{h_k} := e^{\left[\begin{smallmatrix} A^p & B^pC^c \\ 0 & A^c \end{smallmatrix}\right] h_k}, \qquad B := \begin{bmatrix} B^pD^c \\ B^c \end{bmatrix}, \tag{73a}$$

$$C := \begin{bmatrix} C^p & 0 \end{bmatrix}, \qquad D := I, \tag{73b}$$

$$E_\rho := \int_0^\rho e^{\left[\begin{smallmatrix} A^p & B^pC^c \\ 0 & A^c \end{smallmatrix}\right] s} ds, \quad \rho \in \mathbb{R}. \tag{73c}$$

Protocols as a Switching Function Based on the previous modeling steps, the NCS is formulated as a discrete-time switched uncertain system (67). In this framework, protocols are considered as the switching function determining σ_k. We consider the two protocols mentioned before, namely the Try-Once-Discard (TOD) and the Round-Robin (RR) protocol, and generalise these into the classes of 'quadratic' and 'periodic' protocols.

A *quadratic* protocol is a protocol, for which the switching function can be written as

$$\sigma_k = \arg \min_{i=1,\ldots,N} \bar{x}_k^\top P_i \bar{x}_k, \tag{74}$$

where P_i, $i \in \{1, \ldots, N\}$, are certain given matrices. In fact, the TOD protocol belongs to this class of protocols, see [19, 18].

A *periodic* protocol is a protocol that satisfies for some $\tilde{N} \in \mathbb{N}$

$$\sigma_{k+\tilde{N}} = \sigma_k \tag{75}$$

for all $k \in \mathbb{N}$. $\tilde{N}$ is then called the period of the protocol. Clearly, the RR protocol belongs to this class.

The above modeling approach now provides a description of the NCS system in the form of a *discrete-time switched linear uncertain system* given by (67) and one of the protocols, characterised by (74) or (75). The system switches between N linear uncertain systems and the switching is due to the fact that only one node accesses the network at each transmission instant. The uncertainty is caused by the fact that the transmission intervals and the transmission delays $(h_k, \tau_k) \in \Psi$ are varying over time.

Remark 29 *If there is only one node $N = 1$ and $\Gamma_i = I$, we recover the setting of Section 3.2 for the case of small delays. If in addition, $\bar{\tau} = 0$ and $\bar{h} = \underline{h}$, then the NCS model reduces to a standard sampled data system (in case of a continuous-time plant and discrete-time controller), see e.g. [20, 6].*

4.2.2 The polytopic overapproximation

As in the case without communication constraints (see Section 3.2), also in the NCS models derived in the previous section the uncertainty appears in an exponential manner (see terms A_{h_k}, $A_{h_k-\tau_k}$, E_{h_k} and $E_{h_k-\tau_k}$ in (67)). To convert these descriptions into a suitable form for robust stability analysis we exploit a polytopic overapproximation method. Basically any of the mentioned ones can be applied, but in [18], in which the above modeling is presented, a combination of gridding and norm-bounding is combined into an efficient method. The gridding method in [18] has the advantage that, as is formally proven, it is non-conservative in the sense that if the exact discrete-time model is asymptotically stable proven by a parameter-dependent quadratic Lyapunov function [13], then the overapproximated polytopic system corresponding to a sufficiently refined grid of Ψ also has a parameter-dependent quadratic Lyapunov function and the corresponding LMIs are feasible. In other words, if the exact NCS system is "quadratically stable," then the LMIs derived in [18] will prove this for a sufficiently fine gridding. We refer the reader for full details to [19, 18].

4.2.3 Application to the batch reactor

We will analyse the exact same setup as for the continuous-time approach in Subsection 4.1.3 and focus on the TOD and RR protocol and assume that the controller is directly connected to the actuator, i.e., only the (two) outputs are transmitted via the network. Using the LMIs as in [18] we try to obtain combinations of $\bar{h}$ and $\bar{\tau}$ for which the NCS is stable, and we assume that $\underline{\tau} = 0$, and we take $\underline{h} = 10^{-4}$. This results in tradeoff curves, as shown in Fig. 10. These tradeoff curves can be used to

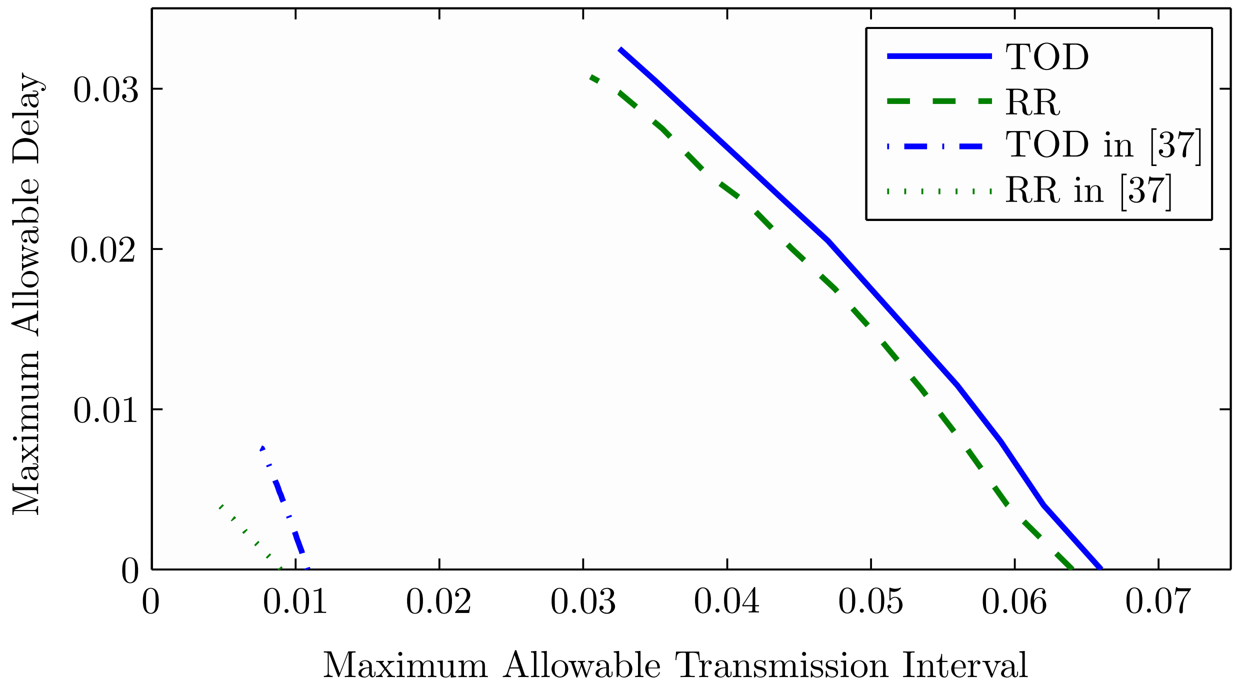

Figure 10: Tradeoff curves between allowable transmission intervals and transmission delays for two different protocols, where [37] refers to the approach in Section 4.1, while the other curves refer to the discrete-time approach as discussed here.

impose or select a suitable network with certain communication delay and bandwidth requirements. Note that bandwidth is inversely proportional with the maximum transmission interval.

Moreover, in Fig 10, also the tradeoff curves as obtained for the continuous-time approach (Subsection 4.1.3) are given. We conclude that the discrete-time approach is less conservative than the continuous-time one (at least for this example). More interestingly, in case there is no delay, i.e., $\underline{\tau} = \overline{\tau} = 0$, the maximum allowable transmission interval $\overline{h}$ obtained in [4], which provide the least conservative results known in literature so far, was $\overline{h} = 0.0108$, while we obtain $\overline{h} = 0.066$. In [71], $\overline{h}$ was estimated (using simulations) to be between 0.06 and 0.08 for the TOD protocol. Furthermore, for the RR protocol, [4] provides the bound $\overline{h} = 0.009$ in the delay-free case, while we obtain $\overline{h} = 0.064$. Also in [71], for a constant transmission interval, i.e. $\underline{h} = \overline{h}$, the bound 0.0657 was obtained for the RR protocol. The case where the transmission interval is constant, provides an upper bound on the true maximum allowable transmission interval (MATI). We can therefore conclude that for this example, the discrete-time methodology reduces conservatism significantly in comparison to existing methodologies and even approximates known estimates of the true MATI closely. In addition, the discrete-time approach applies to situations (non-zero lower bounds and discrete-time controllers, see [18] for examples) that cannot be handled by the continuous-time methodologies.

4.3 Comparison of discrete-time and continuous-time approaches

Interestingly, both the discrete-time and the continuous-time approaches exploit a NCS model that is intrinsically of a hybrid nature. The continuous-time approach results in *hybrid inclusions* with flows and resets [24], while the discrete-time approach uses *uncertain switched linear systems* that are overapproximated by *uncertain switched polytopic systems*.

There are some clear (dis)advantages of both methods. The continuous-time approach as presented in Subsection 4.1 applies to general *continuous-time* nonlinear plants and controllers. However, it does not allow for discrete-time controllers and cannot handle nonzero lower bounds on the transmission intervals h_k and delays τ_k. However, note that the continuous-time approach can accommodate for NCSs based on general (continuous-time) nonlinear plants and controllers and (UGES) protocols.

The discrete-time approach as discussed in Subsection 4.2 can allow for both *continuous-time* and *discrete-time* controllers and *non-zero* lowerbounds on delays and transmission intervals. However, it applies to the case of *linear* plants and controllers and specific protocols (periodic and quadratic protocols) only, although it can do this in a significantly less conservative manner as the ("general-purpose") continuous-time approach.

5 Conclusions

In this overview we summarized various approaches to the stability analysis and stabilizing controller synthesis of NCSs with varying delays, varying transmission intervals, packet dropouts and communication constraints. Three main lines of modelling and stability analysis methods can be distinguished, namely the discrete-time approach (discrete-time NCS model, which can be used for both discrete-time and continuous-time controllers), emulation approach (continuous-time sampled-data NCS models with continuous-time controllers) and the direct approach using impulsive-delay differential equations (continuous-time sampled-data NCS models with discrete-time controllers). The controller synthesis methods are rather limited still and deserve some proper and useful extensions. In the discrete-time case ordinary and lifted state feedback controllers could be designed using LMI conditions, while in the emulation approach, continuous-time controllers are synthesized based on the network-free nonlinear system (using some arbitrary method for the design of stabilizing controllers for nonlinear systems, which is in general not trivial). As the emulation design does not incorporate any information on the network, it is hard to design controllers that are stabilizing and performing for sufficiently long delays and transmission intervals, although one can aim at obtaining favourable characteristics through the presented stability conditions. In addition, one might wonder if a continuous-time controllers are useful at all practical problems as most NCS setups will require digital discrete-time controllers that are tailored towards non-zero lower bounds on delays and transmission intervals. Of course, one can implement the continuous-time controller using numerical integration schemes. Constructive design conditions for the direct approach seem to be missing in the literature. Also the design of output-based dynamic controllers in the discrete-time context is unsolved at present and it seems that this problem reduces to output-based dynamic control design for polytopic systems, which is known to be a hard problem. Observer-based control design (possibly using MPC controllers) might offer attractive advantages, especially since the observer might also be used to compensate for delays, varying sampling times and packet losses. Some of the research of WIDE will be directed in this direction. Also the stability analysis framework will be extended to include *all* five of the network-induced communication imperfections. Initial results obtained within WIDE [27] provide the first framework that actually can do this. Implementing the stability analysis and controller design methodologies in an efficient Matlab toolbox will be very useful as well, as such tools are currently not available in the control community.

References

[1] R. Brockett. Stabilization of motor networks. In *Proc. of the 34th IEEE Conf. on Decision and Control*, volume 2, pages 1484–1488, 1995.

[2] R.W. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Trans. Automat. Control*, 45:1279–1289, 2000.

[3] D. Carnevale, A.R. Teel, and D. Nešić. A Lyapunov proof of improved maximum allowable transfer interval for networked control systems. *IEEE Trans. Aut. Control*, 52:892–897, 2007.

[4] D. Carnevale, A.R. Teel, and D. Nešić. A Lyapunov proof of improved maximum allowable transfer interval for networked control systems. *IEEE Trans. Autom. Control*, 52(5):892–897, 2007.

[5] A. Chaillet and A. Bicchi. Delay compensation in packet-switching networked controlled sytems. In *Submitted for publication*.

[6] T. Chen and B.A. Francis. *Optimal Sampled-data Control Systems*. Springer-Verlag, London, UK, 1995.

[7] M. Cloosterman. *Control over Communication Networks: Modeling, Analysis, and Synthesis*. PhD thesis Eindhoven University of Technology, 2008.

[8] M. Cloosterman, N. van de Wouw, W.P.M.H. Heemels, and H. Nijmeijer. Robust stability of networked control systems with time-varying network-induced delays. In *cdc*, pages 4980–4985, San Diego, USA, December 2006.

[9] M.B.G. Cloosterman, L. Hetel, N. van de Wouw, W.P.M.H. Heemels, J. Daafouz, and H. Nijmeijer. Controller synthesis for networked control systems. *submitted*, 2009.

[10] M.B.G. Cloosterman, N. van de Wouw, W.P.M. Heemels, and H. Nijmeijer. Stabilization of networked control systems with large delays and packet dropouts. In *American Control Conference*, pages 4991 – 4996, 2008.

[11] M.B.G. Cloosterman, N. van de Wouw, W.P.M.H. Heemels, and H. Nijmeijer. Stability of networked control systems with large delays.

[12] M.B.G. Cloosterman, N. van de Wouw, W.P.M.H. Heemels, and H. Nijmeijer. Stability of networked control systems with uncertain time-varying delays. *IEEE Trans. Autom. Control*, 54(7):1575–1580, 2009.

[13] J. Daafouz and J. Bernussou. Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & Control Letters*, 43:355–359, 2001.

[14] D.B. Dacic and D. Nesic. Quadratic stabilization of linear networked control systems via simultaneous protocol and controller design. *Automatica*, 43:1145–1155, 2007.

[15] D.B. Dačić and D. Nešić. Quadratic stabilization of linear networked control systems via simultaneous protocol and controller design. *Automatica*, 43(7):1145–1155, 2007.

[16] M. C. de Oliveira, J. Bernussou, and J. C. Geromel. A new discrete-time robust stability condition. *Systems & Control Letters*, 37:261–265, 1999.

[17] D.F. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Trans. Automat. Contr.*, 35:916–924, 1990.

[18] M.C.F. Donkers, W.P.M.H. Heemels, L. Hetel, N. van de Wouw, and M. Steinbuch. Stability analysis of networked control systems using a switched linear systems approach. *Submitted for journal publication*.

[19] M.C.F. Donkers, L. Hetel, W.P.M.H. Heemels, N. van de Wouw, and M. Steinbuch. Stability analysis of networked control systems using a switched linear systems approach. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, pages 150–164. Springer Verlag, 2009.

[20] Gene F. Franklin, J. David Powell, and M.L. Workman. *Digital control of dynamic systems*. Addison-Wesley Pub. Co. Inc., Reading, USA, 1990.

[21] H. Fujioka. Stability analysis for a class of networked/embedded control systems: A discrete-time approach. In *Proc. of the American Control Conf.*, pages 4997–5002, 2008.

[22] M. Garcia-Rivera and A. Barreiro. Analysis of networked control systems with drops and variable delays. *Automatica*, 43:2054–2059, 2007.

[23] R. Gielen, S. Olaru, and M. Lazar. On polytopic embeddings as a modeling framework for networked control systems. In *Proc. 3rd Int. Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control., Pavia, Italy*, 2008.

[24] R. Goebel, R. Sanfelice, and A.R. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, 2009.

[25] V. Gupta, T.H. Chung, B. Hassibi, and R.M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, 2006.

[26] W.P.M.H. Heemels, R.J.A. Gorter, A. van Zijl, P.P.J. v.d. Bosch, S. Weiland, W.H.A. Hendrix, and M.R. Vonder. Asynchronous measurement and control: a case study on motor synchronisation. *Control Engineering Practice*, 7(12):1467–1482, 1999.

[27] W.P.M.H. Heemels, D. Nešić, A.R. Teel, and N. van de Wouw. In *Proc. Joint 48th IEEE Conference on Decision and Control (CDC) and 28th Chinese Control Conference, Shanghai, China*, 2009.

[28] W.P.M.H. Heemels, H. Siahaan, A. Juloski, and S. Weiland. Control of quantized linear systems: an l_1-optimal control approach. In *Proc. American Control Conference*, pages 3502–3507, Denver, USA, 2003.

[29] W.P.M.H. Heemels, A.R. Teel, N. van de Wouw, and D. Nešić. Networked control systems with communication constraints: tradeoffs between sampling intervals and delays. In *Proc. European Control Conference in Budapest, Hungary*, 2009.

[30] W.P.M.H. Heemels, A.R. Teel, N. van de Wouw, and D. Nešić. Networked control systems with communication constraints: Tradeoffs between sampling intervals, delays and performance. *IEEE Trans. Autom. Control*, 2009 (accepted).

[31] W.P.M.H. Heemels, A.R. Teel, N. van de Wouw, and D. Nešić. Networked control systems with communication constraints: Tradeoffs betweens sampling intervals, delays and performance. *IEEE Trans. Autom. Control*, submitted.

[32] J.P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proc. of the IEEE*, pages 138–162, 2007.

[33] L. Hetel, M.B.G. Cloosterman, N. van de Wouw, W.P.M.H. Heemels, J. Daafouz, and H. Nijmeijer. Comparison of stability characterisations for networked control systems. In *Proc. Joint 48th IEEE Conference on Decision and Control (CDC) and 28th Chinese Control Conference, Shanghai, China*, 2009.

[34] L. Hetel, J. Daafouz, and C. Iung. Stabilization of arbitrary switched linear systems with unknown time-varying delays. *IEEE Trans. Autom. Control*, 51(10):1668–1674, 2006.

[35] L. Hetel, J. Daafouz, and C. Iung. Analysis and control of LTI and switched systems in digital loops via an event-based modeling. *International Journal of Control, To appear*, 2008.

[36] D. Hristu and K. Morgansen. Limited communication control. *Systems & Control Letters*, 37(4):193–205, July 1999.

[37] C.-Y. Kao and B. Lincoln. Simple stability criteria for systems with time-varying delays. *Automatica*, 40:1429–1434, 2004.

[38] M. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32:1361–1379, 1996.

[39] D. Liberzon. On stabilization of linear systems with limited information. *IEEE Trans. Aut. Control*, 48(2):304–307, 2003.

[40] D. Liberzon. Quantization, time delays, and nonlinear stabilization. *IEEE Trans. Automat. Contr.*, 51(7):1190–1195, 2006.

[41] A.S. Matveev and A.V. Savkin. The problem of state estimation via asynchronous communication channels with irregular transmission times. *IEEE Trans. Autom. Control,*, 48(4):670–676, 2003.

[42] L.A. Montestruque and P. Antsaklis. Stability of model-based networked control systems with time-varying transmission times. *IEEE Trans. Autom. Control*, 49(9):1562–1572, 2004.

[43] P. Naghshtabrizi and J.P. Hespanha. Designing an observer-based controller for a network control system. In *Proc. of the 44th Conference on Decision and Control, and the European Control Conference 2005*, pages 848–853, Seville, Spain, December 2005.

[44] P. Naghshtabrizi and J.P. Hespanha. Stability of network control systems with variable sampling and delays. In *Proc. of the Forty-Fourth Annual Allerton Conf. on Communication, Control, and Computing*, 2006.

[45] P. Naghshtabrizi, J.P. Hespanha, and A.R. Teel. Stability of delay impulsive systems with application to networked control systems. In *Proc. American Control Conference*, pages 4899–4904, New York, USA, 2007.

[46] P. Naghshtabrizi and A.R. Teel J.P. Hespanha. Exponential stability of impulsive systems with application to uncertain sampled-data systems. *Systems & Control Letters*, 57(5):378–385, 2008.

[47] Payam Naghshtabrizi and João Pedro Hespanha. Stability of network control systems with variable sampling and delays. In Andrew Singer and Christoforos Hadjicostis, editors, *Proc. of the Forty-Fourth Annual Allerton Conf. on Commun., Control, and Comp.*, September 2006.

[48] G.N. Nair and R.J. Evans. Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM J. Control Optim.*, 43:413–436, 2004.

[49] D Nesic and A Teel. Input-to-state stability of networked control systems. *Automatica*, 40(12):2121–2128, 2004.

[50] D. Nesic, A. R. Teel, and E. D. Sontag. Formulas relating kl-stability estimates of discrete-time and sampled-data nonlinear systems. *Systems & Control Letters*, 38(1):49–60, 1999.

[51] D. Nešić and D. Liberzon. A unified framework for design and analysis of networked and quantized control systems. *IEEE Trans. Autom. Control*, 54(4):732–747, 2009.

[52] D. Nešić and A.R. Teel. Input-output stability properties of networked control systems. *IEEE Trans. Aut. Control*, 49(10):1650–1667, 2004.

[53] D. Nešić and A.R. Teel. Input-output stability properties of networked control systems. *IEEE Trans. Autom. Control*, 49(10):1650–1667, 2004.

[54] D. Nešić and A.R. Teel. Input-to-state stability of networked control systems. *Automatica*, 40:2121–2128, 2004.

[55] Johan Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Dept. of Automatic Control, Lund Inst. of Techn., Lund, Sweden, 1998.

[56] S. Olaru and S.-I. Niculescu. Predictive Control for Linear Systems with Delayed Input Subject to Constraints. In *Proceedings of 17th IFAC World Congress 2008 17th IFAC World Congress 2008*, pages CD–ROM, Seul, Korea, 2008.

[57] H. Rehbinder and M. Sanfridson. Scheduling of a limited communication channel for optimal control. *Automatica*, 40(3):491–500, March 2004.

[58] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *IEEE Trans. Autom. Control*, 49(9):1453–1464, 2004.

[59] J. Skaf and S. Boyd. Analysis and synthesis of state-feedback controllers with timing jitter. *IEEE Transactions on Automatic Control*, 54(3), 2009.

[60] S.C. Smith and P. Seiler. Estimation with lossy measurements: jump estimators for jump systems. *IEEE Trans. Autom. Control*, 48(12):2163–2171, 2003.

[61] Y.S. Suh. Stability and stabilization of nonuniform sampling systems. *Automatica*, 44:3222–3226, 2008.

[62] M. Tabbara, D. Nešić, and A.R. Teel. Stability of wireless and wireline networked control systems. 52(9):1615–1630, 2007.

[63] S. Tatikonda and S.K. Mitter. Control under communication constraints. *IEEE Trans. Automat. Control*, 49:1056–1068, 2004.

[64] Y. Tipsuwan and M.-Y. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11:1099–1111, 2003.

[65] N. van de Wouw, P. Naghshtabrizi, M. Cloosterman, and J. P. Hespanha. Tracking control for networked control systems. In *Proc. of the 46th IEEE Conference on Decision and Control*, pages 4441–4446, New Orleans, LA, USA, December 2007.

[66] N. van de Wouw, P. Naghshtabrizi, M.B.G. Cloosterman, and J.P. Hespanha. Tracking control for sampled-data systems with uncertain sampling intervals and delays. *Intern. Journ. Robust and Nonlinear Control*, 2009 (accepted).

[67] A.J. van der Schaft and J.M. Schumacher. *An Introduction to Hybrid Dynamical Systems*, volume 251 of *LNCIS*. Springer-Verlag, London, 2000.

[68] J.J.C. van Schendel, M.C.F. Donkers, W.P.M.H. Heemels, and N. van de Wouw. On dropout modelling for stability analysis of networked control systems. In *Submitted for publication*.

[69] G.C. Walsh, O. Belidman, and L.G. Bushnell. Asymptotic behavior of nonlinear networked control systems. *IEEE Trans. Automat. Contr.*, 46:1093–1097, 2001.

[70] G.C. Walsh, O. Belidman, and L.G. Bushnell. Stability analysis of networked control systems. *IEEE Trans. Control Systems Technology*, 10:438–446, 2002.

[71] G.C. Walsh, Hong Ye, and L.G. Bushnell. Stability analysis of networked control systems. *IEEE Trans. on Control Systems Technology*, 10(3):438–446, 2002.

[72] T. C. Yang. Networked control system: a brief survey. *IEE Proc.-Control Theory Appl.*, 153(4):403–412, July 2006.

[73] L. Zhang, Y. Shi, T. Chen, and B. Huang. A new method for stabilization of networked control systems with random delays. *IEEE Trans. on Autom. Control*, 50(8):1177–1181, Aug. 2005.

[74] W. Zhang, M.S. Branicky, and S.M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, 2001.

[75] Wei Zhang, Michael S. Branicky, and Stephen M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, February 2001.