Stability and Stabilization of Networked Control Systems

W.P.M.H. Heemels and N. van de Wouw

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 chapter is to survey the main research lines in a comprehensive manner.

W.P.M.H. Heemels
Eindhoven University of Technology, Department of Mechanical Engineering, P.O. Box 513, 5600 MB Eindhoven, The Netherlands e-mail: M.Heemels@tue.nl

N. van de Wouw
Eindhoven University of Technology, Department of Mechanical Engineering, P.O. Box 513, 5600 MB Eindhoven, The Netherlands e-mail: N.v.d.Wouw@tue.nl

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 cannot 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 [39, 85, 98, 102].

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 Fig. 1) modifies the external signals (u,y) of the plant and the controller due to these five imperfections. Indeed, the control input $\hat{u}$ going into the plant is no longer equal to the output u of the controller, and 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 has its own particular effect on the network-induced differences $e_y := \hat{y} - y$ and $e_u := \hat{u} - u$. Obviously, the presence of these network phenomena can degrade the performance of the control loop significantly and can even lead to instability, see, e.g., [11, 14] 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, considering that 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 complete NCS (plant, controller and network) in an integral fashion,

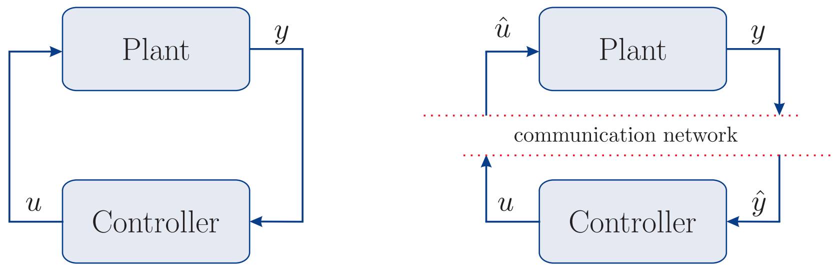

Fig. 1 Introduction of a network in a control loop.

tools are needed that provide quantitative information on the consequences of each of the possible choices in plant, controller and network design.

Although the NCS field is relatively young, various major research lines are already appearing 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 chapter has the aim to provide an overview of the rapidly growing literature on NCS with a focus on methods for stability analysis that incorporate several of the above mentioned communication imperfections. To a lesser extent we will also discuss the stabilization problem. As such, this chapter strives to form the basis for further research that eventually leads to a practically useful analysis and design framework for control over communication networks.

2 Overview of existing approaches

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

Before categorizing the existing approaches, let us start by noting that two essentially different ways exist to model network-induced uncertainties such as time-varying sampling intervals, time-varying delays and packet dropouts. The first class of models assumes (deterministic) bounds on the delays, sampling intervals and the number of subsequent packet dropouts, without adopting any further assumptions on the possibly random processes behind the generation of, e.g., sequences of delays or packet drops. With some abuse of terminology, we will call this the *deterministic* approach. A second class of models exist in which information about the 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 [49, 55, 76, 77, 91, 97] and the references therein for for stochastic approaches. One observation is that the cited references for the stochastic approach at present only can handle a *finite* or *countable* number of delays or sampling intervals, while in reality this is often not the case. Fortunately, recently results are appearing that more realistically consider delays and sampling intervals as continuous random variables taking possibly an uncountable number of values, see, for instance, [1, 2, 19, 54, 67, 82].

2.1 The types of network-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 [5, 18, 33, 35, 50, 61, 84], of packet dropouts in [78, 80], of time-varying transmission intervals and delays in [3, 25, 55, 59], and [11, 14, 22, 32, 41, 47, 58, 101], respectively, and of communication constraints in [4, 17, 44, 46, 71].

References that simultaneously consider two types of network-induced imperfections are given in Table 1. Moreover, [62] consider imperfections of type (i), (iv), (v), [9, 10, 56–58] study simultaneously type (i), (ii), (iii), [63] focuses on type (i), (iii), (iv), while [26] studies (ii), (iii) and (v). Also [7, 20, 36, 37] 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 18 below). Another subtle though important distinction between existing works incorporating varying delays is whether only small delays or also large delays (delays smaller or larger, respectively, than the sampling interval) are considered. In this chapter we will present methods dealing with both cases.

By recent unifications of the work in [62] and [36, 37] a framework is obtained in [34] that can model and analyze the five imperfections simultaneously. Although certain restrictive assumptions are adopted in [34] (regarding, 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)	(iii)	(iv)
(i)	[42, 93, 94]		[6, 21, 64, 83, 88, 89]
(iii)	[13, 27, 31, 52, 100]	–	
(iv)		[45]	-
(v)	[51]	[86]	

2.2 Different approaches in modeling/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 [10, 14, 22, 25, 27, 42, 70, 72, 90, 92, 93, 96, 101, 102], 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 the uncertainties appear in an exponential form (due to discretization). The discrete-time modeling approaches can be further categorized 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., [42] for the latter. Here we will mainly focus on time-driven linear NCS models. 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, possibly also having norm-bounded uncertainty, can then be used in a robust stability analysis, often based on linear matrix inequalities (LMIs), to guarantee the stability of the discrete-time NCS model.
   The final step is to guarantee that also the intersample behavior 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. Typically, this approach is applied to NCS with linear plants and controllers since in that case *exact* discrete-time models can be derived, although recently new results have been obtained that apply to NCS with nonlinear plants and controllers based on approximate discretizations, see [95]. We will discuss the approach for "linear NCS" in more detail in Sections 3.2 and 4.2;
2. The *sampled-data* approach uses continuous-time models that describe the sampled-data NCS dynamics in the continuous-time domain (so without exploiting any form of discretization) and perform stability analysis and controller synthesis based on these sampled-data NCS models directly. Fridman et al. [24] applied a descriptor system approach to model the sampled-data dynamics of systems with varying sampling intervals in terms of (infinite-dimensional) delay-differential equations (DDEs) and study their stability based on the Lyapunov-Krasovskii functional method. In [26, 99, 100], this approach is used for the stability analysis of NCSs with time-varying delays and constant sampling intervals, using (linear) matrix inequality-based techniques. The recent results in [26] show how varying delays, quantization and dropouts can be formulated in one framework based on DDEs, and stability analysis and H_∞

control design methods, based on LMIs, are presented. However, Mirkin [53] showed that the use of such an approach for digital control systems neglects the piecewise constant nature of the control signal due to the zero-order-hold mechanism thereby introducing conservatism when exploiting such modeling for stability analysis. More specifically, the conservatism is introduced by the fact that the zero-order hold and delay jointly introduce a particular linearly increasing time-varying delay within each control update interval (sometimes indicated by the sawtooth behavior of the delay), whereas in the modeling approach mentioned above it is replaced by an arbitrary bounded time-varying delay.

An alternative approach, proposed in [57–59, 93, 94], is based on impulsive DDEs and does take into account the piecewise constant nature of the control signal due to the zero-order-hold mechanism. It has been shown by [53] that this approach is less conservative than the descriptor approach. More specifically, the impulsive DDEs are based on introducing impulses (discontinuous updates) at the moment new information arrives at the controller or the plant. In this manner the true behavior of the underlying NCS is captured. As also noted in [59], the usage of infinite-dimensional DDE models and Lyapunov functionals to analyze the stability of essentially *finite-dimensional* sampled-data NCS does not seem to offer any advantage. The approach in [57, 93] is able to deal simultaneously with time-varying delays and time-varying sampling intervals but does not explicitly include packet dropouts in the model (although they might be considered as variations in the sampling intervals or delays). Here, we will focus mainly on the approach towards the modeling and stability analysis using impulsive DDEs. We call this the *sampled-data* approach, which allows the consideration of 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 *continuous-time* or *emulation* approach, see [17, 36, 37, 63, 64, 88, 89], a continuous-time controller is designed to stabilize 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 (in terms of, e.g., the maximal allowable sampling/transmission interval and/or maximal allowable delay) for which the NCS inherits the stability properties from 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.

Summarizing, the discrete-time approach considers discrete-time controllers (or discretized continuous-time controllers) and a discrete-time NCS model, while the sampled-data approach also considers discrete-time controllers, but has a continuous-time (sampled-data) NCS model. Finally, the continuous-time (emulation) approach focuses on continuous-time controllers using a continuous-time (sampled-data) NCS model. Within all these different approaches different techniques towards stability analysis are used. While the discrete-time approach uses common quadratic or parameter-dependent Lyapunov functions for the discrete-time model, the continuous-time (emulation) approach uses continuous-time Lyapunov functions constructed by combining separate Lyapunov functions for the network-free closed-loop system on the hand and the network protocol on the other hand (or, alternatively, adopting directly small gain arguments). The sampled-data approaches exploits extensions of Lyapunov-Krasovskii function(al)s. We will discuss these methods in details in the following sections, where we start with NCS without communication constraints and scheduling protocols (Section 3) and treat the case with communication constraints in Section 4. In this chapter, we will not pay any attention to quantization effects as these are extensively covered in the Chapter in this book written by Hideaki Ishii.

3 NCS with delays, varying sampling intervals and packet loss

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 modeling, stability analysis and controller design for these NCS. Finally, in Section 3.3, we present a continuous-time approach towards the modeling and stability analysis for these systems exploiting models in terms of impulsive DDEs.

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 intervals and packet dropouts. It is based on the developments in [10] (see also [13, 14]). We choose this level of generality for the reason that the application of the stability techniques presented later can encompass all these types of network-induced phenomena.

The NCS is depicted schematically in Fig. 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 measurements ($y(t) = x(t)$) are sampled resulting in the sampling time instants s_k given by:

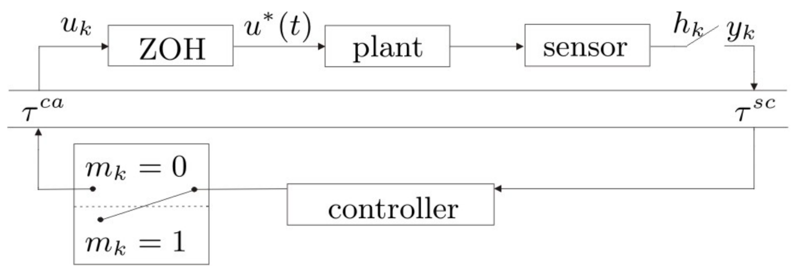

Fig. 2 Schematic overview of the NCS with variable sampling intervals, network delays and packet dropouts.

$$s_0 = 0 \text{ and } s_k = \sum_{i=0}^{k-1} h_i \text{ for } k \geq 1, \tag{2}$$

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 $k \in \mathbb{N}$. The notation $y_k := y(s_k)$ denotes the k-th sampled value of y, $x_k := x(s_k)$ the k-th sampled value of the state and u_k the control value corresponding to $y_k = x_k$. Packet drops may occur (see Fig. 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 Fig. 2) is applied to transform the discrete-time control value 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. The sensor is assumed to act in a time-driven fashion (i.e., sampling occurs at the times s_k defined in (2)) and 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 older arrives 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 [66, 102]. To include these effects in the continuous-time model, 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

$$\dot{x}(t) = Ax(t) + Bu^*(t) \tag{4a}$$

$$u^*(t) = u_{k^*(t)}. \tag{4b}$$

Here, we assume that the most recent control input remains active in the plant if a packet is dropped. Note that in some NCS setups one also use the different policy to set the control input to 0 in case dropout occurs instead of holding the previous control value, see, e.g., [74].

The delays are assumed to be bounded and contained in the interval $[\tau_{\min}, \tau_{\max}]$, the sampling interval are bounded and contained in the interval $[h_{\min}, h_{\max}]$ and the number of subsequent packet dropouts is upper bounded by $\overline{\delta}$. The latter means that

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

for all $k \in \mathbb{N}$ 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 $\mathscr{S}$ of admissible sequences $\{(s_k, \tau_k, m_k)\}_{k\in\mathbb{N}}$ can be described as follows:

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

which includes variable sampling intervals, small and large delays, and packet dropouts. Note that in this case we allow for large delays in the sense that τ_k might be larger than h_k.

3.2 Discrete-time modeling approaches

3.2.1 The exact discrete-time NCS model

To arrive at a discrete-time description of the NCS, the equation (4b) of the continuous-time control input $u^*(t)$ is reformulated to indicate explicitly which control inputs u_l 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 Fig. 3). Moreover, $0 = t_0^k \leq t_1^k \leq \ldots \leq t^k_{\overline{d}+\overline{\delta}-\underline{d}} \leq t^k_{\overline{d}+\overline{\delta}-\underline{d}+1} := h_k$.

Proof. The proof is given in [10], see also [9, 14]. ■

Note that the above lemma first of all indicates that the only control values that can be active in the interval $[s_k, s_{k+1}]$ are $u_{k-\overline{d}-\overline{\delta}}, \ldots, u_{k-\underline{d}}$. Secondly, (7) indicates that $u_{k+j-\overline{d}-\overline{\delta}}$ is active in $[s_k + t_j^k, s_k + t_{j+1}^k)$. Note that when $t_j^k = t_{j+1}^k$ this essentially means that the value $u_{k+j-\overline{d}-\overline{\delta}}$ is not active in the interval $[s_k, s_{k+1}]$ (e.g. due to a dropout or more recent information arriving earlier). The exact values of t_j^k are determined by the exact realization of the delays, sampling intervals and dropouts as present in the right-hand side of (8).

Based on Lemma 1, a discrete-time NCS model can be obtained now by exact integration of (4) leading to

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

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

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

Using now the lifted state vector

$$\xi_k = \left(x_k^T \; u_{k-1}^T \; \ldots \; u_{k-\overline{d}-\overline{\delta}}^T \right)^T$$

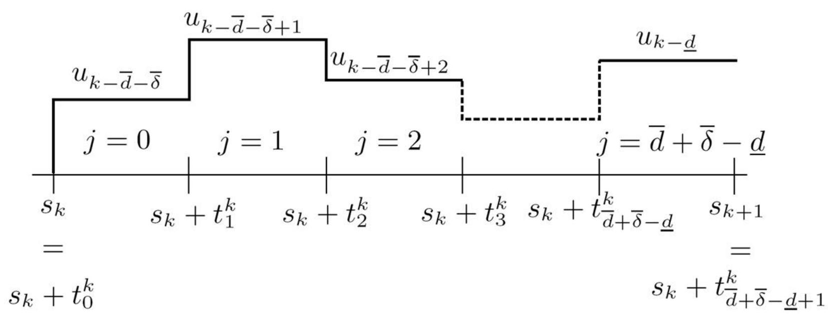

Fig. 3 Graphical interpretation of the actuation update instants t_j^k.

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

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

Remark 1. 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 expressions (8) for the actuation update times. 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 $m_{k-\bar{d}-\bar{\delta}}, \dots, m_{k-\underline{\delta}}$ 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 [75] (see also Remark 18).

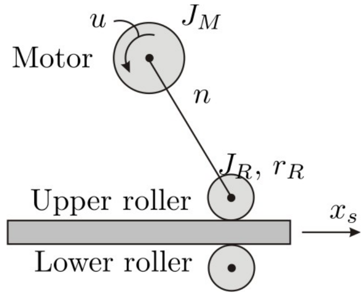

Fig. 4 Schematic overview of the motor-roller example.

Remark 2. In the above model set-up a time-driven modeling paradigm (exact integration from sample instant to sample instant) was adopted. An alternative discrete-time modeling approach was proposed in [42], which uses an event-driven paradigm (integrating from event instant to event instant, where the events include sampling, updating of control values, etc.).

To illustrate the developments so far, let us consider the following example.

Example 1. The example consists of a second-order motion control example, obtained from the document printing domain. In particular, a single motor driving a roller-pair is considered, as depicted in Fig. 4, which obeys the dynamics:

$$\ddot{x}_s = \frac{q r_R}{J_M + q^2 J_R} u, \tag{13}$$

with $J_M = 1.95 \cdot 10^{-5}\text{kgm}^2$ 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.

The continuous-time state-space representation of (13) is given by (1), with $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ b \end{bmatrix}$, with $b := \frac{q r_R}{J_M + q^2 J_R} = 126.7\ (\text{kgm})^{-1}$, and $x(t) = \begin{bmatrix} x_s(t) & \dot{x}_s(t) \end{bmatrix}^T$. For the sake of simplicity, consider the case that the sampling interval h is constant, the delays $\tau_k \in [\tau_{\min}, \tau_{\max}]$, $\forall k \in \mathbb{N}$, $\tau_{\min} = 0$ and $\tau_{\max} = h$ (i.e. the small delay case with $\overline{d} = 1$, $\underline{d} = 0$) and $\bar{\delta} = 0$ (no packet dropouts). The exact discrete-time model can be written in the form (11), where the extended discrete-time state consists of the state of continuous-time model and the previous control action (due to the small delay): $\xi_k = \begin{bmatrix} x_k^T & u_{k-1}^T \end{bmatrix}^T$. Moreover, the uncertain parameter $\theta_k = t_1^k = \tau_k$ (see (8)), since the small delay case is considered, a constant sampling interval and no packet dropouts. The (uncertain) matrices in (11) are given by

$$\begin{aligned} \Lambda = e^{Ah} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix},\ M_0(\tau_k) &= \begin{bmatrix} \frac{b}{2}\left(h^2 - (h-\tau_k)^2\right) \\ b\left(h - (h-\tau_k)\right) \end{bmatrix}, \\ M_1(\tau_k) = \int_0^h e^{As} ds B - M_0(\tau_k) &= \begin{bmatrix} \frac{1}{2} b (h-\tau_k)^2 \\ b(h-\tau_k) \end{bmatrix} \end{aligned} \tag{14}$$

and the overall model (11) reduces to

$$\begin{aligned}
\xi_{k+1} &= \tilde{A}(\theta_k)\xi_k + \tilde{B}(\theta_k)u_k && \text{(15a)}\\
&= \begin{pmatrix} \Lambda & M_0(\tau_k) \\ 0 & 0 \end{pmatrix} \xi_k + \begin{pmatrix} M_1(\tau_k) \\ 1 \end{pmatrix} u_k && \text{(15b)}\\
&= \begin{pmatrix} 1 & h & \frac{b}{2}\left(h^2-(h-\tau_k)^2\right) \\ 0 & 1 & b\left(h-(h-\tau_k)\right) \\ 0 & 0 & 0 \end{pmatrix} \xi_k + \begin{pmatrix} \frac{1}{2}b(h-\tau_k)^2 \\ b(h-\tau_k) \\ 1 \end{pmatrix} u_k. && \text{(15c)}
\end{aligned}$$

The latter equalities show clearly that we are dealing with a parameter-varying linear systems, in which one can observe the basic terms $h-\tau_k$ and $(h-\tau_k)^2$ as uncertainty terms. We will now present a general procedure how to find and overapproximate these uncertainty terms such that the system becomes amendable for stability analysis and controller synthesis.

3.2.2 The polytopic overapproximation

A first step towards the stability analysis is transforming the bounds on the delays, sampling intervals and dropouts ($\tau_{\min}$, $\tau_{\max}$, $h_{\min}$, $h_{\max}$ and $\bar{\delta}$) to upper and lower bounds on t_j^k. These computations are done in [9, 10] and lead to bounds $t_{j,\min}, t_{j,\max}$, i.e., $t_j^k \in [t_{j,\min}, t_{j,\max}]$ for all $k \in \mathbb{N}$, see [9, 10] for the exact expressions. Together with the fact that $h_k \in [h_{\min}, h_{\max}]$, one can define the uncertainty set

$$\Theta = \{\theta_k \in \mathbb{R}^{\overline{d}+\overline{\delta}-\underline{d}+1} \mid h_k \in [h_{\min}, h_{\max}], t_j^k \in [t_{j,\min}, t_{j,\max}], \\ 1 \le j \le \overline{d}+\overline{\delta}-\underline{d}, 0 \le t_1^k \le \ldots \le t^k_{\overline{d}+\overline{\delta}-\underline{d}} \le h_k\}, \quad (16)$$

such that $\theta_k \in \Theta$ for all $k \in \mathbb{N}$.

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 "larger" model that has useful structural properties such as discrete-time polytopic models with (or without) additional norm-bounded uncertainties. If robust stability (or other properties) can be proven for this polytopic overapproximation, then this also implies the robust stability of the original discrete-time NCS model. As these polytopic models are suitable for the application of available *robust* stability methods, this provides a means to tackle the NCS stability analysis problem.

In the literature, many different ways of constructing such polytopic embeddings of the uncertain system are proposed: overapproximation techniques are based on interval matrices [11], the real Jordan form [10, 12–14, 68], the Taylor series [41], gridding and norm-bounding [21, 25, 72, 79, 81], and the Cayley-Hamilton theorem [28, 29]. There are also some approaches that are related to gridding and norm-bounding such as [3, 22] in which essentially one grid point is taken corresponding to one nominal sampling interval or nominal delay and the variation of sampling intervals/delays is captured in the norm-bounded uncertainties. Typically, [21, 25, 72, 79, 81] use more grid points to reduce the size of the norm-bounded uncertainties and thereby the conservatism of the overapproximation and resulting stability conditions. 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 [10, 12–14]. For a comprehensive overview and comparison of these overapproximation techniques we refer the interested reader to [38].

Real Jordan form

To derive the stability analysis and control synthesis conditions, the model (11) is rewritten using the real Jordan form [43] of the continuous-time system matrix A. Basically, the state matrix is expressed as $A = TJT^{-1}$ with J the real Jordan form and T an invertible matrix. This leads to a generic model of the form

$$\xi_{k+1} = \left(F_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k) F_i \right) \xi_k + \left(G_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta_k) G_i \right) u_k \tag{17}$$

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, ν is the degree of the minimal polynomial $q_{\min}$ of A. Note that the minimal polynomial of A is the monic polynomial p of smallest degree that satisfies $p(A) = 0$. The minimal polynomial can be easily obtained [43] from the (complex) Jordan form. Actually, ν is equal to the sum of all the maximal dimensions of the *complex* Jordan blocks corresponding to all the *distinct* eigenvalues of A. Clearly, $\nu \leq n$, where n is the dimension of the state vector x. Note that $\nu = n$ 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 [38] or to Appendix B in [9].

Using bounds on the uncertain parameters $\theta_k = (h_k, t_1^k, \ldots, t_{\overline{d}+\overline{\delta}-\underline{d}}^k)$ described by the set Θ in (16) the set of matrix pairs

$$\mathscr{F}\mathscr{G} = \left\{ \left(F_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta) F_i, G_0 + \sum_{i=1}^{\zeta} \alpha_i(\theta) G_i \right) \mid \theta \in \Theta \right\} \tag{18}$$

can be formulated that contains all possible matrix combinations in (17) and thus also in (11). Based on this infinite set $\mathscr{F}\mathscr{G}$ of matrices, stability analysis (for a given controller) and the design of stabilizing controllers can be carried out for the NCS (4). To overcome the infinite dimension of the set $\mathscr{F}\mathscr{G}$, a *polytopic overapproximation* of the set is used. Denote the maximum and minimum value of $\alpha_i(\theta_k)$, respectively, by

$$\overline{\alpha}_i = \max_{\theta_k \in \Theta} \alpha_i(\theta_k), \ \underline{\alpha}_i = \min_{\theta_k \in \Theta} \alpha_i(\theta_k) \tag{19}$$

with Θ defined in (16). Then the set of matrices $\mathscr{F}\mathscr{G}$, given in (18), is a subset of $co(\mathscr{H}_{FG})$, where 'co' denotes the convex hull and $\mathscr{H}_{FG}$ is the *finite* set of matrix pairs given by

$$\mathscr{H}_{FG} = \left\{ \left((F_0 + \sum_{i=1}^{\zeta} \alpha_i F_i), (G_0 + \sum_{i=1}^{\zeta} \alpha_i G_i) \right) : \alpha_i \in \{\underline{\alpha}_i, \overline{\alpha}_i\}, i = 1, 2, \ldots, \zeta \right\}. \tag{20}$$

The set of vertices $\mathscr{H}_{FG}$ is written as $\mathscr{H}_{FG} = \{(H_{F,j}, H_{G,j}) \mid j = 1, 2, \ldots, 2^{\zeta}\}$ for enumeration purposes later. Hence, we have that

$$\mathscr{F}\mathscr{G} \subseteq co(\mathscr{H}_{FG}) := \{\sum_{j=1}^{\zeta} \beta_j (H_{F,j}, H_{G,j}) \mid \beta = (\beta_1, \ldots, \beta_{\zeta})^T \in \mathbb{B}\}, \tag{21}$$

where

$$\mathbb{B} := \{\beta \in \mathbb{R}^{\zeta} \mid \sum_{j=1}^{\zeta} \beta_j = 1 \text{ and } \beta_j \geq 0 \text{ for all } j = 1, \ldots, \zeta\}. \tag{22}$$

Hence, we obtain the polytopic system

$$\xi_{k+1} = (F_0 + \sum_{j=1}^{\zeta} \beta_j^k F_j)\xi_k + (G_0 + \sum_{j=1}^{\zeta} \beta_j^k G_j)u_k \tag{23}$$

with $\beta^k \in \mathbb{B}$ for each $k \in \mathbb{N}$. Due to (21) any input/state trajectory generated by (11) for some sequence $\{\theta_k\}_{k \in \mathbb{N}}$ with $\theta_k \in \Theta$, $k \in \mathbb{N}$ is also an input/state trajectory of (23) for some sequence $\{\beta^k\}_{k \in \mathbb{N}}$ with $\beta^k \in \mathbb{B}$, $k \in \mathbb{N}$.

Example 2. Revisit the motion control system as in Example 1. We take $h = 0.001$ constant, $\tau_{\min} = 0$, $\tau_{\max} = h$ and $\overline{\delta} = 0$, which leads to the exact discrete-time representation is given by (15). Let us now illustrate the procedure for convex overapproximation based on the real Jordan form, as explicated above, using this example. The exact discrete-time model (15) can be written in the form (17), with the uncertain functions $\alpha_1(\tau_k) = h - \tau_k$ and $\alpha_2(\tau_k) = (h - \tau_k)^2$, and

$$F_0 = \begin{bmatrix} 1 & h & \frac{b}{2}h^2 \\ 0 & 1 & bh \\ 0 & 0 & 0 \end{bmatrix}, \quad G_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$
$$F_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -b \\ 0 & 0 & 0 \end{bmatrix}, \quad G_1 = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \tag{24}$$
$$F_2 = \begin{bmatrix} 0 & 0 & -\frac{1}{2}b \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad G_2 = \begin{bmatrix} \frac{1}{2}b \\ 0 \\ 0 \end{bmatrix}.$$

Hence, the number of uncertain functions α_i is $\zeta = 2$.

In order to illustrate the conservatism introduced by the overapproximation of the set of matrices $\mathscr{F}\mathscr{G}$ in (18) by the convex hull of the set of matrices $\mathscr{H}_{FG}$ in (20), it is important to realize that the only uncertain matrix in the discrete-time system (15) is the matrix (2×1)-matrix $M_0(\tau_k)$ given in (14). Namely, $M_1(\tau_k)$ in (15) can be written as $M_1(\tau_k) = \int_0^h e^{As} ds B - M_0(\tau_k)$, see (14). The matrix $M_0(\tau_k)$ can be written as follows in terms of the uncertain functions:

$$M_0(\tau_k) = \begin{bmatrix} \frac{1}{2}b(h^2 - \alpha_2(\tau_k)) \\ b(h - \alpha_1(\tau_k)) \end{bmatrix}. \tag{25}$$

Now, the overapproximation of $\mathscr{F}\mathscr{G}$ by the convex hull of $\mathscr{H}_{FG}$ basically means overapproximating $\{M_0(\tau) \mid \tau \in [\tau_{\min}, \tau_{\max}]\}$, with $\tau_{\min} = 0$ and $\tau_{\max} = h$, by the convex hull of the following set of four ($2^\zeta = 4$) generators:

$$\begin{aligned}
M_{0,1} &= \begin{bmatrix} \frac{1}{2}b\left(h^2 - (h-\tau_{\max})^2\right) \\ b\left(h - (h-\tau_{\max})\right) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}bh^2 \\ bh \end{bmatrix} \\
M_{0,2} &= \begin{bmatrix} \frac{1}{2}b\left(h^2 - (h-\tau_{\min})^2\right) \\ b\left(h - (h-\tau_{\max})\right) \end{bmatrix} = \begin{bmatrix} 0 \\ bh \end{bmatrix} \\
M_{0,3} &= \begin{bmatrix} \frac{1}{2}b\left(h^2 - (h-\tau_{\max})^2\right) \\ b\left(h - (h-\tau_{\min})\right) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}bh^2 \\ 0 \end{bmatrix} \\
M_{0,4} &= \begin{bmatrix} \frac{1}{2}b\left(h^2 - (h-\tau_{\min})^2\right) \\ b\left(h - (h-\tau_{\min})\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\end{aligned} \tag{26}$$

The polytopic overapproximation is visualized in Fig. 5.

3.2.3 Stability analysis

In this section, we consider the stability analysis of the NCS (11) (or equivalently (17)) in closed-loop with a state feedback controller. From a control design perspective, when dealing with a system such as (11), it is natural to design a state feedback controller using the full lifted state variable ξ_k of the model (11), i.e.,

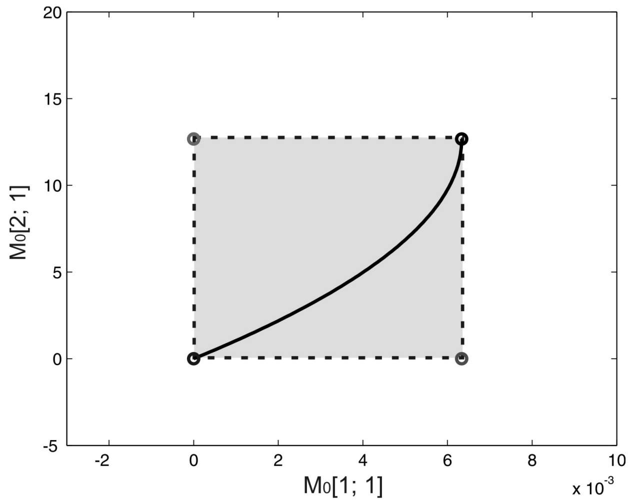

Fig. 5 Overapproximation of the set of 2×1-matrix $\{M_0(\tau) \mid \tau\in[\tau_{\min},\tau_{\max}]\}$ of exponential uncertainties ($M_0[1,1]$ on the horizontal axis and $M_0[2,1]$ on the vertical axis) for $h=0.001$ ms and $\tau_k\in[0,h]$. The circles indicate the vertices $M_{0,i}$, $i=1,2,3,4$, defined in (26) and the gray area the convex hull of these vertices.

$$u_k = -K\xi_k. \tag{27}$$

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

$$u_k = -K_0x_k - K_1u_{k-1}\ldots - K_{\overline{d}+\overline{\delta}}u_{k-\overline{d}-\overline{\delta}}. \tag{28}$$

The use of such a *dynamic* control law requires a reconsideration of the assumption made earlier that allowed, amongst others, to lump all the delays τ_k^{sc},τ_k^{c} and τ_k^{ca} in one parameter τ_k (see Section 3.1). Using a dynamic control law as in (28) actually leads to more restrictive assumptions on the network modeling setup as $y_k = x_k$ should always arrive at the controller after the moment that u_{k-1} is sent to the actuator, i.e. $s_k+\tau_k^{sc} > s_{k-1}+\tau_{k-1}^{sc}+\tau_{k-1}^{c}$ as otherwise special precautions are needed to handle out-of-order arrival of measured outputs resulting in longer delays. In addition, the adopted modeling setup and controller in (28) require that no packet dropouts occur between the sensors and the controller. Although modeling dropouts alternatively as prolongations of the sampling interval (see, e.g., the comparison in [75]) might alleviate these issues to some extent, dropouts in the channel between the controller and the actuators introduce similar complications in this case.

The mentioned issues do not occur for genuine static state feedbacks of the form

$$u_k = -\bar{K}x_k = -\begin{bmatrix}\bar{K} & 0\end{bmatrix}\xi_k =: -K\xi_k \tag{29}$$

and the more restrictive assumptions, as mentioned above, are not needed. This enhances its applicability. For this reason, in the controller synthesis section we will focus on the design of a controller in the form (29), and we will provide references for the design of lifted state feedbacks as in (27), see Remark 7. However, the design of state feedbacks as in (29) 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. A solution for this hard problem will be provided. However, we first derive stability conditions for the NCS given a state feedback as in (27) or (29). In the stability analysis it is assumed implicitly that in case the lifted state feedback controller (27) is used the more restrictive assumptions on the network setup, as mentioned above, are satisfied.

The closed-loop system resulting from interconnecting (11) and (27) can be formulated as follows:

$$\xi_{k+1} = \tilde{A}_{cl}(\theta_k)\xi_k \text{ with } \tilde{A}_{cl}(\theta_k) = \left(\tilde{A}(\theta_k) - \tilde{B}(\theta_k)K\right), \tag{30}$$

with $\theta_k \in \Theta$ for all $k \in \mathbb{N}$, or equivalently, after exploiting the real Jordan form as in (17), as

$$\xi_{k+1} = F_{cl}(\theta_k)\xi_k, \tag{31}$$

with

$$F_{cl}(\theta_k) = \left[(F_0 - G_0K) + \sum_{i=1}^{\zeta} \alpha_i(\theta_k)(F_i - G_iK)\right]\xi_k. \tag{32}$$

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

$$\mathscr{F}_{cl} = \left\{\left(F_0 - G_0K\right) + \sum_{i=1}^{\zeta} \alpha_i(\theta)\left(F_i - G_iK\right) | \theta \in \Theta\right\}. \tag{33}$$

Given the fact that $\mathscr{F}\mathscr{G} \subseteq co(\mathscr{H}_{FG})$ with $\mathscr{F}\mathscr{G}$ as in (18) and $\mathscr{H}_{FG}$ as in (20), it follows that

$$\mathscr{F}_{cl} \subseteq co\left(\mathscr{H}_{F_{cl}}\right) \tag{34}$$

with

$$\mathscr{H}_{F_{cl}} = \left\{\left(F_0 - G_0K\right) + \sum_{i=1}^{\zeta} \alpha_i\left(F_i - G_iK\right) : \alpha_i \in \{\underline{\alpha}_i, \overline{\alpha}_i\}, i = 1, 2, \dots, \zeta\right\}. \tag{35}$$

We will also write the set of vertices $\mathscr{H}_{F_{cl}}$ as $\mathscr{H}_{F_{cl}} = \{H_{F_{cl},j} \mid j = 1, 2, \dots, 2^{\zeta}\}$ for enumeration purposes. Hence,

$$\mathscr{F}_{cl} \subseteq co\{H_{F_{cl},1}, \dots, H_{F_{cl},2^{\zeta}}\}. \tag{36}$$

Using the finite set $\mathscr{H}_{F_{cl}}$ of 2^{ζ} vertices, a finite number of LMI-based stability conditions can be formulated using [15, 16]. The resulting stability characterization for the closed-loop system (30) using parameter-dependent Lyapunov functions is given in the following theorem in which we use the notation $\succ$ and $\prec$ to indicate positive definiteness and negative definiteness, respectively, of a matrix.

Theorem 1. *Consider the discrete-time NCS model* (11) *and the state feedback controller* (27)*, with the network-induced uncertainties* $\theta_k \in \Theta$*,* $\forall k \in \mathbb{N}$*, and* Θ *defined in* (16)*. If there exist matrices* $P_j = P_j^T \succ 0,\ j = 1,2,\dots,2^\zeta$*, that satisfy*

$$H_{F_{cl},j}^T P_l H_{F_{cl},j} - P_j \prec 0, \text{ for all } j,l \in \{1,2,\dots,2^\zeta\}, \tag{37}$$

with $H_{F_{cl},j} \in \mathscr{H}_{F_{cl}},\ j = 1,2,\dots,2^\zeta$*, and* $\mathscr{H}_{F_{cl}}$ *defined in* (35)*, then the origin of the closed-loop NCS system* (11), (27) *is a globally exponentially stable (GES) equilibrium point.*

Proof. The proof is a direct consequence of the results in [10, 14, 40]. ■

Remark 3. It can be shown that under the conditions of Theorem 1 also the intersample behavior is bounded, see e.g. [9–11, 14]. Using the results in [65], this also implies that the equilibrium point $x = 0$ of the sampled-data NCS (4), (7), (8), (27) is GES.

Remark 4. This theorem exploits the following Lyapunov function

$$V(\xi_k, \beta^k) = \xi_k^T P(\beta^k)\xi_k \tag{38}$$

based on the inclusion (36), which guarantees that (31) can be overapproximated by the polytopic system

$$\xi_{k+1} = (\sum_{j=1}^{2^\zeta} \beta_j^k H_{F_{cl},j})\xi_k \tag{39}$$

with $\beta^k \in \mathbb{B}$, $k \in \mathbb{N}$, and $\mathbb{B}$ defined in (22). The parameter-dependent Lyapunov function $V(\xi_k, \beta^k)$ is then given by $\xi_k^T P(\beta^k)\xi_k = \xi_k^T \sum_{j=1}^{2^\zeta} \beta_j^k P_j \xi_k$. In [10, 40] it is shown that if the LMIs in the above theorem are satisfied then they imply the existence of a Lyapunov-Krasovskii functional (LKF) of the form

$$V(x_k, \dots, x_{k-\overline{d}-\overline{\delta}}, \theta_k) = \sum_{i=0}^{\overline{d}+\overline{\delta}} \sum_{j=0}^{\overline{d}+\overline{\delta}} x_{k-i}^T Q^{i,j}(\theta_k) x_{k-j}, \tag{40}$$

which is the most general (discrete-time) LKF that can be obtained using quadratic forms. Notice that when using this approach based on parameter-dependent quadratic Lyapunov functions 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, are avoided.

Remark 5. The case of a common quadratic Lyapunov function (CQLF) $V(\xi_k) = \xi_k^T P \xi_k$ is a particular case of this theorem by taking $P_j = P,\ \ j = 1,\dots,2^\zeta$.

3.2.4 Design of stabilizing controllers

As already briefly mentioned, the main difficulty to synthesize a genuine state feedback (29) is that it results in a structured control synthesis problem, i.e., a control law (27) needs to be designed 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 [69]. Moreover, as was already exploited for the stability analysis problem above, such an approach allows for the use of a parameter-dependent Lyapunov function [15] 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 (29) are given in the next theorem.

Theorem 2. *Consider the NCS model* (4)*, and* (29)*, and its discrete-time representation* (11)*,* (29) *for sequences of sampling instants, delays, and packet dropouts* $\sigma \in \mathscr{S}$ *with* $\mathscr{S}$ *as in* (6)*. Consider the equivalent representation* (17) *based on the Jordan form of A and the set of vertices* $\mathscr{H}_{FG}$ *defined in* (20)*.*

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* $X_1 \in \mathbb{R}^{n\times n}$*,* $X_{2,j} \in \mathbb{R}^{(\overline{d}+\overline{\delta})m\times n}$*,* $X_{3,j} \in \mathbb{R}^{(\overline{d}+\overline{\delta})m\times(\overline{d}+\overline{\delta})m}$*,* $j = 1,2,\ldots,2^{\zeta}$*, that satisfy*

$$\begin{pmatrix} X_j + X_j^T - Y_j & X_j^T H_{F,j}^T - \begin{pmatrix}\overline{Z} & 0\end{pmatrix}^T H_{G,j}^T \\ H_{F,j}X_j - H_{G,j}\begin{pmatrix}\overline{Z} & 0\end{pmatrix} & Y_l \end{pmatrix} \succ 0, \tag{41}$$

for all $j,l \in \left\{1,2,\ldots,2^{\zeta}\right\}$*, then the closed-loop NCS* (4) *and* (29) *with* $\overline{K} = \overline{Z}X_1^{-1}$ *is globally exponentially stable (GES) for sequences of sampling instants, delays, and packet dropouts* $\sigma \in \mathscr{S}$*.*

Proof. For the proof, see [10].

Note that in the above theorem the stability is directly formulated for the continuous-time NCS model (4) and (29) using the ideas in Remark 3.

Remark 6. 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 $Y_j = Y$, for all $j = 1,\ldots,2^{\zeta}$, with $P = Y^{-1}$.

Remark 7. If one is still interested in using a lifted state feedback (27) despite the mentioned disadvantages, then Theorem 2 can be modified by replacing the matrices X_j $j = 1,2,\ldots,\zeta$ by a constant matrix X without a specific structure and replacing $\begin{pmatrix}\overline{Z} & 0\end{pmatrix}$ by Z. The extended state feedback controller is obtained then by $K = ZX^{-1}$.

Remark 8. The derived discrete-time models based on polytopic overapproximations as in (23) are suitable for control design using model predictive control (MPC) as well. For instance, the MPC techniques in [48] can be used for this purpose as was indicated in [28, 29].

Remark 9. The design of output-based dynamic discrete-time controllers that result in stable closed-loop NCSs is at present an unsolved problem. Due to the adopted polytopic overapproximations, the problem is basically a design problem for a robustly stabilizing output-based dynamic controller for polytopic systems, which 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 10. Here, we only presented results on the stability and stabilization of NCSs. However, extensions exist that provide constructive LMI conditions guaranteeing input-to-state stability, see [93, 94]. In [93, 94] 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.

Example 3. Consider again Example 1. As a first instance, assume that the sensor sampling interval $h = 1$ ms is constant and that the controller is given by (29) with $\bar{K} = \begin{pmatrix} 50 & K_2 \end{pmatrix}$. The controller gains K_2 that stabilize the system with time-varying delays $\tau_k \in [0, \tau_{\max}]$, with $\tau_{\max} \leq 2h$, are determined using Theorem 1 for the case of a common quadratic Lyapunov function, resulting in the gray area in Fig. 6. In other words for a fixed value of K_2 the NCS is stable for all $\tau_k \in [0, \tau_{\max}]$ as long as $\tau_{\max}$ lies in the gray area for the corresponding value of K_2. Clearly, the large delay case is considered here. To assess the conservatism of the computed stability region, the stability region for *constant* time-delays equal to $\tau_{\max}$ is depicted by the dash-dotted line in Fig. 6. This comparison reveals the fact that the stability bound is hardly conservative for this example, as the stability region for *time-varying* delays should always lie within the stability region for *constant* delays. In Fig. 6, also a

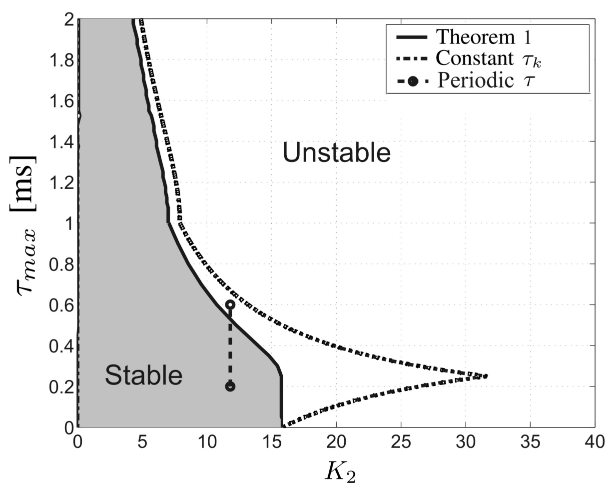

Fig. 6 Stability region in terms of K_2 and time-varying delays $\tau_k \in [0, \tau_{\max}]$ (for $h = 1$ms, $K_1 = 50$) for Theorem 1, with a common quadratic Lyapunov function, and for constant delays equal to $\tau_{\max}$.

periodic delay sequence τ_1, τ_2, τ_1, $\tau_2, \ldots$, with $\tau_1 = 0.2h$ and $\tau_2 = 0.6h$, is depicted which has been shown in [11, 14] to induce instability. The latter observation is another indicator for the fact that the stability boundary for uncertain, time-varying delays as in Figure 6 is hardly conservative.

For more examples, illustrating both Theorems 1 and 2 including the case including packet dropouts and the exploitation of parameter-dependent Lyapunov functions, we refer the interested reader to [9, 10, 13, 75].

3.3 Sampled-data modeling approaches

In this section, we discuss a modeling and analysis approach for NCS with delays, time-varying sampling intervals and packet dropouts as developed in [57–59]. Herein, the sampled-data NCS model is formulated in terms of so-called impulsive delay-differential equations (DDEs). 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 discretization of a continuous-time plant model as in the discrete-time approach;
- The model in terms of impulsive DDEs shows great similarity with the modeling of the sampled-data NCS using the hybrid systems formalism, see, e.g., [6, 34, 36, 37, 63, 64], as will be discussed in Section 4.1. However, the continuous-time approach described in Section 4.1 is an emulation-type approach, where controllers are designed in continuous-time, whereas here *discrete-time* (state feedback) controllers are considered and included directly in the sampled-data NCS model. Also the sampled-data approach using impulsive DDEs has not incorporated the presence of communication constraints and the resulting scheduling protocols as has been done in the continuous-time approach, see [6, 34, 36, 37, 63, 64] and Section 4.1.
- The modeling framework of impulsive DDEs in principle allows to consider nonlinear systems for which stability results for nonlinear impulsive DDEs have been presented, e.g., in [58, 59]. However, only for the case of *linear* NCS constructive LMI-based stability conditions have been formulated. The continuous-time approach in Section 4.1 (see [6, 34, 36, 37, 63, 64]) has stability conditions for nonlinear NCSs as well.

Consider the linear continuous-time plant (1) and a discrete-time static state feedback controller as in (29), 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 $h_k \in [h_{\min}, h_{\max}]$ for all $k \in \mathbb{N}$. 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, it is assumed that the sensor-to-controller delay, computational delay and controller-to-actuator delay can be lumped into a single delay τ_k, with $\tau_k \in [\tau_{\min}, \tau_{\max}]$ for all $k \in \mathbb{N}$. Summarizing, $\{(s_k, \tau_k)\}_{k\in\mathbb{N}} \in \bar{\mathscr{S}}$, where

$$\bar{\mathscr{S}} := \left\{ \{(s_k, \tau_k)\}_{k\in\mathbb{N}} |\, h_{\min} \le s_{k+1} - s_k \le h_{\max}, s_0 = 0, \tau_k \in [\tau_{\min}, \tau_{\max}) \text{ for all } k \in \mathbb{N} \right\} \tag{42}$$

represents the admissible sequences of sampling times and delays. The uncertainty set $\bar{\mathscr{S}}$ represents a subset of the uncertainties characterised by $\mathscr{S}$ in (6), in which also packet dropouts where accounted for. 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 18. Let us denote by $r_k = s_k + \tau_k$ the k-th control update instant with $r_0 = \tau_0$. Both the small and large delay cases can be considered, where we allow the delays τ_k to be larger than the sampling intervals h_k with the understanding that the sequence of input update times $\{r_0, r_1, r_2, \ldots\}$ remains strictly increasing. In essence, this means that if a sample arrives at the destination in an out-of-order fashion (i.e., an old sample arrives the destination after the most recent one), it should be rejected (and is effectively deleted from the sequence s_k).

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

$$\begin{cases} \dot{x} &= Ax + Bu^*(t), \;\; x(0) = x_0 \\ u^*(t) &= u_k, \qquad\qquad r_k \le t \le r_{k+1}, \; . \\ u_k &= -\bar{K}x_k, \end{cases} \tag{43}$$

Alternatively, the sampled-data NCS system can more compactly be formulated as

$$\dot{x} = Ax - B\bar{K}x(s_k), \quad r_k \le t \le r_{k+1}, \tag{44}$$

with initial condition given by x_0 and $x(s_{-1})$.

Let us introduce the definition $v_1(t) := x(s_k)$ for $t \in [r_k, r_{k+1})$, where $v_1(t)$ represents a piece-wise constant signal reflecting a delayed version the most recently sampled state (that is not rejected). Moreover, when we introduce $\zeta(t) := \left[x^T(t)\; v_1^T(t)\right]^T$, we can write the dynamics of the NCS (43) (or (44)) in the form of an impulsive DDE of the form

$$\dot{\zeta}(t) = F\zeta(t), \qquad t \in [r_k, r_{k+1}) \tag{45a}$$

$$\zeta(r_{k+1}) = \begin{bmatrix} x(r_{k+1}) \\ x(s_{k+1}) \end{bmatrix}, \qquad k \in \mathbb{N} \tag{45b}$$

with $\zeta(t)$ right-continuous, the initial condition $\zeta(0) := \left[x^T(0)\; x^T(s_{-1})\right]^T$, and

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

Consider the following positive-definite candidate Lyapunov functional

$$
\begin{aligned}
V :=& x^T P x + \int_{t-\rho_1}^{t} (\rho_{1max} - t + s)\dot{x}^T(s) R_1 \dot{x}(s) ds \\
&+ \int_{t-\rho_2}^{t} (\rho_{2max} - t + s)\dot{x}^T(s) R_2 \dot{x}(s) ds + \int_{t-\tau_{\min}}^{t} (\tau_{\min} - t + s)\dot{x}^T(s) R_3 \dot{x}(s) ds \\
&+ \int_{t-\rho_1}^{t-\tau_{\min}} (\rho_{1max} - t + s)\dot{x}^T(s) R_4 \dot{x}(s) ds + (\rho_{1max} - \tau_{\min}) \int_{t-\tau_{\min}}^{t} \dot{x}^T(s) R_4 \dot{x}(s) ds \\
&+ \int_{t-\tau_{\min}}^{t} x^T(s) Z x(s) ds + (\rho_{1max} - \rho_1)(x - v_2)^T X (x - v_2)
\end{aligned} \tag{46}
$$

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

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

and

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

Note that $\rho_1(t)$ and $\rho_2(t)$ are sawtooth-like functions of time representing, within a control update interval, the elapsed time since the last (not rejected) sampling instant and the elapsed time since the last (not rejected) control update, respectively. The evolution of this Lyapunov functional is discontinuous at the control update times r_k, due to the jump in ζ in (45b), but a decrease of V over the jump is guaranteed by construction.

The next theorem formulates LMI-based conditions for global exponential stability of the NCS (45) for any sequence of sampling instants and delays taken from the class $\bar{\mathscr{S}}$ as in (42).

Theorem 3. *[57, 58] If there exist positive definite matrices P, X, Z, R_i, $i = 1, \dots, 4$, and not necessarily symmetric matrices N_i, $i = 1, \dots, 4$, satisfying the LMIs*

$$
\begin{bmatrix} M_1 + (\beta - \tau_{\min})(M_2 + M_3) & \tau_{\max} N_1 & \tau_{\min} N_3 \\ * & -\tau_{\max} R_1 & 0 \\ * & * & -\tau_{\max} R_3 \end{bmatrix} \prec 0, \tag{47a}
$$

$$
\begin{bmatrix} M_1 + (\beta - \tau_{\min}) M_2 & \tau_{\max} N_1 & \tau_{\min} N_3 & (\beta - \tau_{\min})(N_1 + N_2) & (\beta - \tau_{\min}) N_4 \\ * & -\tau_{\max} R_1 & 0 & 0 & 0 \\ * & * & -\tau_{\min} R_3 & 0 & 0 \\ * & * & * & -(\beta - \tau_{\min})(R_1 + R_2) & 0 \\ * & * & * & * & -(\beta - \tau_{\min}) R_4 \end{bmatrix} \prec 0, \tag{47b}
$$

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

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

then, system (45) *is globally exponentially stable for any sequence of delays and sampling instants taken from the class $\bar{\mathscr{S}}$ as in* (42).

Proof. For the proof, see [58].

Remark 11. The proof of Theorem 3 exploits stability results for nonlinear impulsive DDEs as presented in [58, 59].

Remark 12. The conditions in Theorem 3 do not explicitly depend on the values of $h_{\min}$. Consequently, this approach towards modeling NCSs may result in more conservative conditions in comparison to those obtained using the discrete-time approach discussed in Section 3.2, when $0 \ll h_{\min} \simeq h_{\max}$. The reason is that the discrete-time approach can actually exploit the knowledge that $h_{\min} > 0$.

Remark 13. When considering the control synthesis problem, i.e. when the control gain $\bar{K}$ is considered unknown, the LMIs in Theorem 3 generally become bilinear matrix inequalities (BMIs). However, for the case without delays in [59, 60] LMI-based control synthesis conditions for static state feedback controllers have been proposed.

Remark 14. In [57], results on the stability analysis of linear NCS with continuous-time, dynamic output feedback controllers are presented. Herein, it is assumed that these continuous-time controllers can be evaluated exactly on the sampling instants (by exact discretization and some form of time-stamping of the sampled measurements).

Example 4. We now reconsider the motion control example of Example 1 and use it to compare the discrete-time and sampled-data approach.

First consider the case of a constant sampling interval $h = 0.005$ s, but with time-varying and uncertain delays in the set $[0, \tau_{\max}]$. Applying Theorem 3 in a slightly modified form (for the special case in which a Lyapunov functional (46) with $Z = R_i = 0$, $i = 1, \ldots, 4$, is exploited) leads to stability guarantee for the NCS for a maximal delay up to $\tau_{\max} = 0.33h$, see [93, 94]. Comparing this with the discrete-time approach using Theorem 1 (for the special case of a common quadratic Lyapunov function) shows 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., $h_k \in [h_{\min}, h_{\max}]$, $k \in \mathbb{N}$ 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 1 (for the special case of a common quadratic Lyapunov function), stability can be assured 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 1 are not conservative in this example. Using the impulsive DDE approach, stability can only be guaranteed up to $h_{\max} = 9 \times 10^{-3}$ s. Hence, for this motion control example the discrete-time approach clearly outperforms the sampled-data approach as far as the characterisation of stability is concerned.

Remark 15. In [93, 94], an extension of Theorem 3 (for the special case that $Z = R_i = 0$, $i = 1, \ldots, 4$), guarantees input-to-state stability in the face of perturbations. This extension 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 impulsive DDE modeling approach are much tighter than those obtained using the discrete-time modeling and analysis approach as shown in [93, 94]. The conservatism in the estimates of the input-to-state stability gain using the discrete-time approach are mainly due to the conservative upperbounding of the intersample behavior. In this respect it seems that the impulsive DDE approach is beneficial in studying such performance related issues.

4 NCS including communication constraints

In this section we will discuss stability analysis approaches that incorporate communication constraints. Specifically, communication is constrained in the sense that the number of control inputs and measured outputs that can be transmitted over a network is limited. 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. Which node obtains access is determined by a scheduling protocol. As we will see this complicates the description and the analysis of the NCS considerably. The communication constraints and protocol will actually introduce (additional) discrete effects in the problem, which will require modeling and stability analysis techniques from the hybrid systems domain [30, 87].

We will present a continuous-time/emulation approach in Section 4.1 and a discrete-time approach in Section 4.2. 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 [34] and in Remark 18, but for the ease of exposition we will not consider them below. The model that we discuss in this section was derived in [36, 37] and forms an extension of the NCS models used before in [63] that were motivated by the work in [89]. The emulation approach is characterized by the design procedure that , first, a stabilizing continuous-time controller for the continuous-time plant is designed (ignoring any network effects). Next, we study under which network effects (level of delays, size of sampling interval lengths, type of protocol used for the communication scheduling) the NCS inherits the stability properties from the network-free continuous-time closed-loop system.

4.1.1 Description of the NCS

Consider the continuous-time plant

$$\dot{x}_p = f_p(x_p, \hat{u}), \qquad y = g_p(x_p) \tag{48}$$

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

$$\dot{x}_c = f_c(x_c, \hat{y}), \qquad u = g_c(x_c), \tag{49}$$

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 $0 \leq t_0 < t_1 < t_2 < \ldots$. Even stronger, we assume that 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 channel. They will arrive after a transmis-

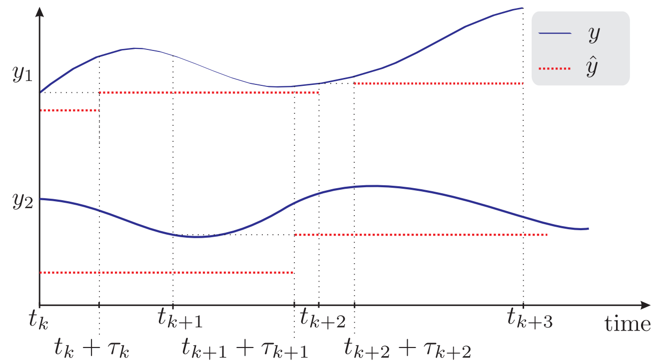

Fig. 7 Illustration of a typical evolution of y and $\hat{y}$ for 2 nodes.

sion delay of τ_k time units at the controller or actuator, respectively. This results in updates of the corresponding entries in $\hat{y}$ or $\hat{u}$ at the arrival times $t_k + \tau_k$, $k \in \mathbb{N}$, which were denoted by r_k in the previous section. The situation described above is illustrated for y and $\hat{y}$ in Fig. 7 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). In particular, we will use the following standing assumption in the sequel.

Assumption 4 *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.*

This assumption implies that each transmitted packet arrives before the next sample is taken meaning that only the small delay case is considered here[1]. In various situations $\tau_k \leq h_k := t_{k+1} - t_k$, $k \in \mathbb{N}$ is a realistic assumption. Indeed, if two or more nodes are sharing one communication channel and one of the nodes is transmitting its data, the channel is busy and hence other nodes cannot access the network, which guarantees $\tau_k \leq h_k = t_{k+1} - t_k$, $k \in \mathbb{N}$.

Remark 16. Compared to the notation in the previous section we have that $h_{\max} = h_{mati}$ and $h_{\min} = \delta$, which can actually be chosen arbitrarily close to 0. For the delays we have that $\tau_{\min} = 0$ and $\tau_{\max} = \tau_{mad}$. We used here the terms MATI and MAD (h_{mati} and τ_{mad}) as done in the literature [6, 34, 36, 37, 63, 64] on the continuous-time approach.

The updates of $\hat{y}$ and $\hat{u}$ satisfy

$$\hat{y}((t_k + \tau_k)^+) = y(t_k) + h_y(k, e(t_k)) \tag{50a}$$
$$\hat{u}((t_k + \tau_k)^+) = u(t_k) + h_u(k, e(t_k)) \tag{50b}$$

[1] Extensions of this continuous-time approach including large delays do not exist to this date.

at $t_k+\tau_k$, where e denotes the vector $(e_y,e_u) := (e_y^T,e_u^T)^T$ with $e_y := \hat{y}-y$, $e_u := \hat{u}-u$ and h_u and h_y are update functions related to the protocol. 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 update functions h_y and h_u are related to the protocol of which we give two well-known examples below. Typically when the j-th node gets access to the network at some transmission time t_k the corresponding values in $\hat{y}$ and $\hat{u}$ have a jump at $t_k+\tau_k$ to the corresponding transmitted values in $y(t_k)$ and $u(t_k)$, since the quantization effects are assumed to be negligible. For instance, when y_j is transmitted at time t_k, it holds that $h_{y,j}(k,e(t_k)) = 0$ meaning that $\hat{y}_j((t_k+\tau_k)^+) = y_j(t_k)$. However, for reasons of generality, more freedom is allowed in the protocols given by $h = (h_y,h_u) := (h_y^T,h_u^T)^T$. Two well-known examples are the Round Robin (RR) protocol the Try-Once-Discard (TOD) protocol (sometimes also called the maximum-error-first protocol). For 2 nodes the RR protocol is given 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 can send its data. As such, the RR protocol is a *static* protocol in the sense that the order of the nodes is fixed. In contrast, the TOD protocol is a *dynamic* scheduling 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 later we will also use $\langle \cdot,\cdot \rangle$ as the corresponding inner product. Hence, the TOD protocol gives access to the node with the largest difference between the latest transmitted value of the corresponding inputs/outputs and the current value of these inputs/outputs. Indeed, the node with the largest network-induced error e_i is allowed to transmit its signal values. 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 $t_{k+1}+\tau_{k+1}$:

$$\dot{\hat{y}} = 0, \quad \dot{\hat{u}} = 0. \tag{51}$$

To compute the resets of e at the update or arrival times $\{t_i+\tau_k\}_{k\in\mathbb{N}}$, 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 it is 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:

$$
\left.\begin{aligned}
\dot{x}(t) &= f(x(t),e(t)) \\
\dot{e}(t) &= g(x(t),e(t))
\end{aligned}\right\} \quad t \in [t_k, t_k+\tau_k) \tag{52a}
$$

$$
e((t_k+\tau_k)^+) = h(k,e(t_k)) - e(t_k) + e(t_k+\tau_k), \tag{52b}
$$

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 [63] for the explicit expressions of f and g.

Remark 17. The model (52) reduces to the model used in [63, 64] in absence of delays, i.e. $\tau_k = 0$ for all $k \in \mathbb{N}$. Indeed, then (52) becomes

$$
\left.\begin{aligned}
\dot{x}(t) &= f(x(t),e(t)) \\
\dot{e}(t) &= g(x(t),e(t))
\end{aligned}\right\} \quad t \in [t_{s_i}, t_{s_i}+\tau_k) \tag{53a}
$$

$$
e(t_k^+) = h(k,e(t_k)). \tag{53b}
$$

Assumption 5 *f and g are continuous and h is locally bounded.* ■

Observe that the system $\dot{x} = f(x,0)$ is the closed-loop system (48)-(49) without the network ($e = 0$).

The stability problem that is considered is formulated as follows.

Problem 1. Suppose that the controller (49) was designed for the plant (48) rendering the continuous-time closed loop (48)-(49) (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 (52) is stable as well when the transmission intervals and delays satisfy Assumption 4. ■

Remark 18. 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 MATI for the dropout-free case.

Remark 19. In case $h(k,e) = 0$ for all $k \in \mathbb{N}$ and $e \in \mathbb{R}^{n_e}$, the above model essentially reduces to a sampled-data systems (without communication constraints) with a continuous-time controller. In this particular case the impulsive DDE in the sampled-data approach of Section 3.3 (see Remark 14) and the continuous-time NCS model as presented here are related.

4.1.2 Reformulation in a hybrid system framework

To facilitate the stability analysis, the above NCS model is transformed into the hybrid system framework as developed in [30]. To do so, the auxiliary variables $s \in \mathbb{R}^n$, $\kappa \in \mathbb{N}$, $\tau \in \mathbb{R}_{\geq 0}$ and $\ell \in \{0,1\}$ are introduced 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 (52b) 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 explicit use is made of the fact that only small delays are considered.

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

$$\left.\begin{aligned} \dot{x} &= f(x,e) \\ \dot{e} &= g(x,e) \\ \dot{s} &= 0 \\ \dot{\kappa} &= 0 \\ \dot{\tau} &= 1 \\ \dot{\ell} &= 0 \end{aligned}\right\} (\ell = 0 \wedge \tau \in [0, h_{mati}]) \vee (\ell = 1 \wedge \tau \in [0, \tau_{mad}]) \tag{54}$$

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

$$\begin{aligned} (x^+, e^+, s^+, \tau^+, \kappa^+, \ell^+) = G(x,e,s,\tau,\kappa,\ell), \text{ when} \\ (\ell = 0 \wedge \tau \in [\delta, h_{mati}]) \vee (\ell = 1 \wedge \tau \in [0, \tau_{mad}]) \end{aligned} \tag{55}$$

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

$$G(x,e,s,\tau,\kappa,0) = (x, e, h(\kappa,e) - e, 0, \kappa + 1, 1) \tag{56}$$

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

$$G(x,e,s,\tau,\kappa,1) = (x, s + e, -s - e, \tau, \kappa, 0). \tag{57}$$

4.1.3 Lyapunov-based stability analysis

A Lyapunov function for Σ_{NCS} will be constructed based on the following conditions for the reset part (the protocol) and the flow part of the system.

Conditions on the reset part

Condition 6 *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*

$$\underline{\alpha}_W |e| \leq W(\kappa, e) \leq \overline{\alpha}_W |e| \tag{58a}$$
$$W(\kappa+1, h(\kappa, e)) \leq \lambda W(\kappa, e) \tag{58b}$$

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

Additionally it is assumed here that

$$W(\kappa+1, e) \leq \lambda_W W(\kappa, e) \tag{59}$$

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

$$\left| \frac{\partial W}{\partial e}(\kappa, e) \right| \leq M_1 \tag{60}$$

for some constant $M_1 > 0$. For all protocols discussed in [6, 63, 88, 89] 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 [37, 63] for the proofs.

Conditions on the flow part

The following growth condition on the flow of the NCS model (52) is used:

$$|g(x, e)| \leq m_x(x) + M_e |e|, \tag{61}$$

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

Condition 7 *There exists a locally Lipschitz continuous function* $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ *satisfying the bounds*

$$\underline{\alpha}_V(|x|) \leq V(x) \leq \overline{\alpha}_V(|x|) \tag{62}$$

for some $\mathcal{K}_\infty$*-functions*[2] $\underline{\alpha}_V$ *and* $\overline{\alpha}_V$*, and the condition*

[2] A function $\alpha : \mathbb{R}_+ \to \mathbb{R}_+$ is called a $\mathcal{K}$-function, if it is continuous, strictly increasing and $\alpha(0) = 0$. A $\mathcal{K}$-function α is called a $\mathcal{K}_\infty$-function if $\alpha(s) \to \infty$ if $s \to \infty$. Examples of $\mathcal{K}_\infty$-functions are $\alpha(s) = cs^\lambda$ for some $c > 0$ and $\lambda > 0$.

$$\langle \nabla V(x), f(x,e)\rangle \leq -m_x^2(x) - \rho(|x|) + (\gamma^2 - \varepsilon)W^2(\kappa, e) \tag{63}$$

for almost all $x \in \mathbb{R}^{n_x}$ *and all* $e \in \mathbb{R}^{n_e}$ *with* $\rho \in \mathscr{K}_\infty$*, for some* $\gamma > 0$ *and* $0 < \varepsilon < \max\{\gamma^2, 1\}$.

Essentially, the condition above is a Lyapunov-based formulation for the system $\dot{x} = f(x,e)$ to have an $\mathscr{L}_2$ gain [73] from $W^2(\kappa, e)$ to $m_x^2(x)$ strictly smaller than γ together with global asymptotic stability in case $e = 0$.

Stability result

Lumping the above parameters into four new ones given by

$$L_0 = \frac{M_1 M_e}{\underline{\alpha}_W};\ L_1 = \frac{M_1 M_e \lambda_W}{\lambda\, \underline{\alpha}_W};\ \gamma_0 = M_1\gamma;\ \gamma_1 = \frac{M_1 \gamma \lambda_W}{\lambda} \tag{64}$$

we can provide the following conditions on MAD and MATI to guarantee stability of Σ_{NCS}. Indeed, consider now the differential equations

$$\dot{\phi}_0 = -2L_0\phi_0 - \gamma_0(\phi_0^2 + 1) \tag{65a}$$

$$\dot{\phi}_1 = -2L_1\phi_1 - \gamma_0(\phi_1^2 + \frac{\gamma_1^2}{\gamma_0^2}). \tag{65b}$$

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

$$\mathscr{E} := \{(x, e, s, \kappa, \tau, \ell) \mid x = 0,\ e = s = 0\}.$$

Theorem 8. *Consider the system* Σ_{NCS} *such that Assumptions 4 and 5 are satisfied. Let Condition 6 with* (59) *and* (60) *and Condition 7 with* (61) *hold. Suppose* $h_{mati} \geq \tau_{mad} \geq 0$ *satisfy*

$$\phi_0(\tau) \geq \lambda^2 \phi_1(0) \text{ for all } 0 \leq \tau \leq h_{mati} \tag{66a}$$

$$\phi_1(\tau) \geq \phi_0(\tau) \text{ for all } 0 \leq \tau \leq \tau_{mad} \tag{66b}$$

for solutions ϕ_0 *and* ϕ_1 *of* (65) *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* $\mathscr{E}$ *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 (65), that satisfy $U(\xi^+) \leq U(\xi)$ at reset times and $\dot{U}(\xi) < 0$ during flow. See [37] for the proof and the exact definition of UGAS of $\mathscr{E}$, which implies (next to Lyapunov stability of $\mathscr{E}$) that $x(t) \to 0$, $e(t) \to 0$ and $s(t) \to 0$, when $t \to \infty$.

From the above theorem quantitative numbers for h_{mati} and τ_{mad} can be obtained by constructing the solutions to (65) for certain initial conditions. Computing the τ

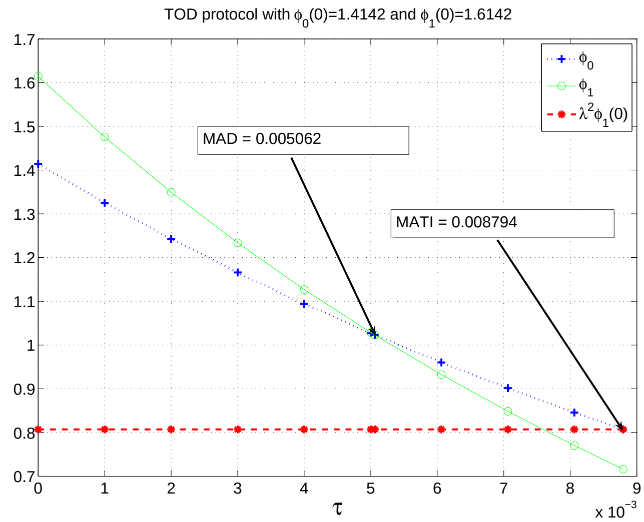

Fig. 8 Illustration of how the solutions ϕ_ℓ, $\ell = 0,1$, lead to MAD and MATI.

value of the intersection of ϕ_0 and the constant line $\lambda^2\phi_1(0)$ provides h_{mati} according to (66a), while the intersection of ϕ_0 and ϕ_1 gives a value for τ_{mad} due to (66b). In Fig. 8 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 (65) 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. Before showing the example, a systematic procedure to determine these tradeoff curves is provided.

Systematic procedure for the determination of MATI and MAD

The main steps in the procedure to compute the tradeoff curves between MATI and MAD are given as follows:

Procedure 9 *Given Σ_{NCS} apply the following steps:*

1. *Construct a Lyapunov function W for the UGES protocol as in Condition 6 with the constants $\underline{\alpha}_W$, $\overline{\alpha}_W$, λ, λ_W and M_1 as in* (58), (59) *and* (60). *Suitable Lyapunov functions and the corresponding constants are available for many protocols in the literature [37, 63, 64];*
2. *Compute the function m_x and the constant M_e as in* (61) *bounding g as in* (52)*;*
3. *Compute for $\dot{x} = f(x,e)$ in the NCS model* (52) *the $\mathcal{L}_2$ gain from $W(\kappa,e)$ to $m_x(x)$ in the sense that* (62)-(63) *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*

LMIs. Of course, here the 'emulated' controller should guarantee that such a property is satisfied;

4. *Use now* (64) *to obtain* L_0, L_1, γ_0 *and* γ_1*;*
5. *For initial conditions* $\phi_0(0)$ *and* $\phi_1(0)$ *with* $\lambda^2\phi_1(0) \le \phi_0(0) < \phi_1(0)$ *compute (numerically) the solutions* ϕ_0 *and* ϕ_1 *to* (65) *and find (the largest values of)* h_{mati} *and* τ_{mad} *such that* (66) *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 thereby obtaining various combinations of* h_{mati} *and* τ_{mad} *leading to tradeoff curves.*

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

Delay-free results

In the above setting taken from [36, 37] both varying h_k and τ_k are allowed. The case without delays ($\tau_{mad} = 0$) has been treated in the earlier works [6, 63, 64, 88, 89]. Basically, the least conservative of them, being [6], uses slightly weaker versions of Condition 6, Condition 7 and

$$\left\langle \frac{\partial W}{\partial e}(\kappa, e), g(x, e) \right\rangle \le LW(\kappa, e) + m_x(x) \tag{67}$$

for all $\kappa \in \mathbb{N}$ and almost all $e \in \mathbb{R}^{n_e}$. Instead of four parameters as in (64) they only have the parameters γ and L next to λ to determine h_{mati} (as $\tau_{mad} = 0$). Also the two differential equations that are formulated in (65) reduce to only one differential equation given by

$$\dot{\phi} = -2L\phi - \gamma(\phi^2 + 1) \tag{68}$$

and they choose the initial condition $\phi(0) = \lambda^{-1}$. The conditions (66) reduce to $\phi(\tau) \ge \lambda$ for all $0 \le \tau \le h_{mati}$ 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

$$h_{mati} = \begin{cases} \frac{1}{Lr}\arctan\left(\frac{r(1-\lambda)}{2\frac{\lambda}{1+\lambda}(\frac{\gamma}{L}) + 1 + \lambda}\right), & \gamma > L \\ \frac{1-\lambda}{L(1+\lambda)}, & \gamma = L \\ \frac{1}{Lr}\operatorname{arctanh}\left(\frac{r(1-\lambda)}{2\frac{\lambda}{1+\lambda}(\frac{\gamma}{L}) + 1 + \lambda}\right), & \gamma < L, \end{cases} \tag{69}$$

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

Application to the benchmark example of the batch reactor

In this part the discussed results are applied to the case study of the batch reactor, which has developed over the years as a benchmark example in NCSs [6, 63, 89]. The functions in the NCS (52) for the batch reactor are given by the linear functions $f(x,e,w) = A_{11}x + A_{12}e$ and $g(x,e,w) = A_{21}x + A_{22}e$ in which the numerical values for A_{ij}, $i, j = 1,2$, are provided in [63, 89] and given by

$$A_{11} = \begin{pmatrix} 1.3800 & -0.2077 & 6.7150 & -5.6760 & 0 & 0 \\ -0.5814 & -15.6480 & 0 & 0.6750 & -11.3580 & 0 \\ -14.6630 & 2.0010 & -22.3840 & 21.6230 & -2.2720 & -25.1680 \\ 0.0480 & 2.0010 & 1.3430 & -2.1040 & -2.2720 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 1.0000 & -1.0000 & 0 & 0 \end{pmatrix};$$

$$A_{12} = \begin{pmatrix} 0 & 0 \\ 0 & -11.3580 \\ -15.7300 & -2.2720 \\ 0 & -2.2720 \\ 0 & 1.0000 \\ 1.0000 & 0 \end{pmatrix};$$

$$A_{21} = \begin{pmatrix} 13.3310 & 0.2077 & 17.0120 & -18.0510 & 0 & 25.1680 \\ 0.5814 & 15.6480 & 0 & -0.6750 & 11.3580 & 0 \end{pmatrix};$$

$$A_{22} = \begin{pmatrix} 15.7300 & 0 \\ 0 & 11.3580 \end{pmatrix}.$$

The batch reactor, which is open-loop unstable, has $n_u = 2$ inputs, $n_y = 2$ outputs, $n_p = 4$ plant states and $n_c = 2$ controller states and $N = 2$ nodes (only the outputs are assumed to be sent over the network). See [63, 89] for more details on this example.

For all the technical details of the application of Procedure 9 to this benchmark example the reader is referred to [36, 37]. Here we show only the outcomes. Fig. 9 shows the stability regions in terms of MAD and MATI for the TOD and the RR protocols 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, the delay-free results as obtained in [6], which improved the earlier bounds in [63], are exactly recovered. These delay-free results amount for the TOD protocol to $\tau_{mad} = 0$ and $h_{mati} = 0.0108$ and for the RR protocol to $\tau_{mad} = 0$ and $h_{mati} = 0.0090$. Next to finding tradeoffs between MAD and MATI, different protocols can be compared with respect to each other. In Fig. 9, it is apparent 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. Note that in Fig. 9 the particular combination $\tau_{mati} = 0.008794$ and $\tau_{mad} = 0.005062$ corresponding to Fig. 8 is highlighted.

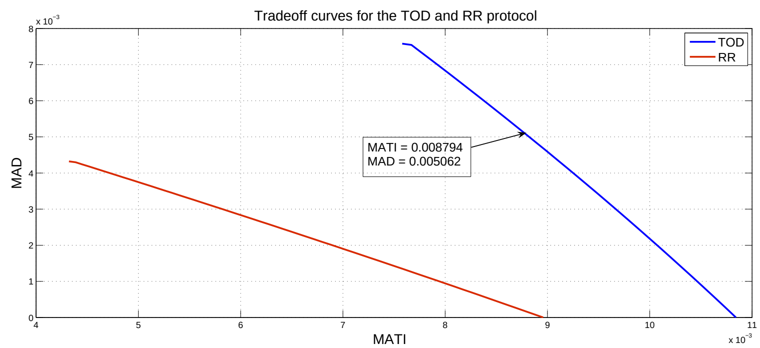

Fig. 9 Tradeoff curves between MATI and MAD.

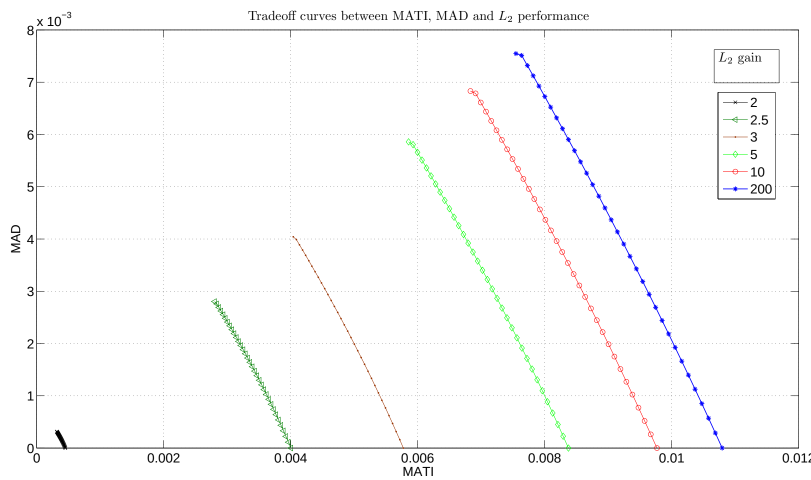

Fig. 10 Tradeoff curves between MATI and MAD for various levels of the $\mathscr{L}_2$ gain of the NCS with the TOD protocol.

Extension of these results to include guarantees on disturbance attenuation properties in the sense of $\mathscr{L}_p$ gains from certain disturbance inputs to to-be-controlled outputs are reported as well in [36, 37]. In case of the batch reactor this would yield results as depicted in Fig. 10 for the $\mathscr{L}_2$ gain. This picture shows tradeoffs between the network properties MAD and MATI on the one hand and control performance in terms of $\mathscr{L}_2$ 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.

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 [17], 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., $h_k \in (\delta, h_{\text{mati}}]$, $\tau_k \in [0, \tau_{\text{mad}}]$, where δ could be chosen arbitrarily close to 0). 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 introduces nonzero lower bounds on the transmission intervals and transmission delays. The discrete-time approach surveyed here (see [20, 21]) 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.

4.2.1 The exact discrete-time NCS model

As mentioned, the discrete-time approach applies in a *linear* context, which means that (48) is replaced by the linear time-invariant (LTI) continuous-time plant given by

$$\begin{aligned} \dot{x}^p(t) &= A^p x^p(t) + B^p \hat{u}(t) \\ y(t) &= C^p x^p(t), \end{aligned} \tag{70}$$

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

$$\begin{aligned} \dot{x}^c(t) &= A^c x^c(t) + B^c \hat{y}(t) \\ u(t) &= C^c x^c(t) + D^c \hat{y}(t), \end{aligned} \tag{71a}$$

or in discrete time by

$$\begin{aligned} x^c_{k+1} &= A^c x^c_k + B^c \hat{y}_k \\ u(t_k) &= C^c x^c_k + D^c \hat{y}(t_k). \end{aligned} \tag{71b}$$

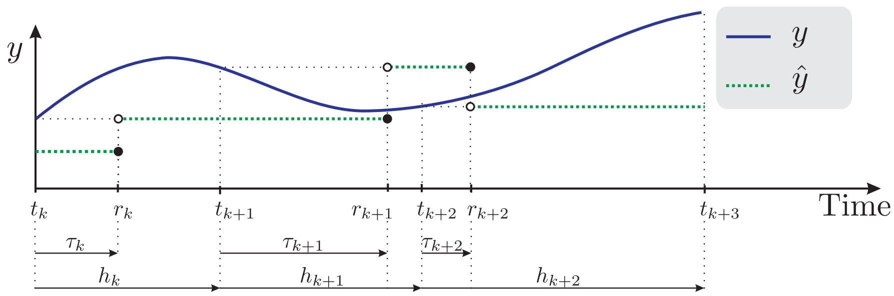

Fig. 11 Illustration of a typical evolution of y and $\hat{y}$.

In parallel with Section 4.1 (only subscripts becoming superscripts, as subscripts are used to indicate the counter k of the discrete-time step), $x^c \in \mathbb{R}^{n_c}$ 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. It is assumed 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. 11. In the case of a discrete-time controller (71b), 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 (71b) has to be performed in the time interval $(r_k, t_{k+1}]$.

The functioning of the network will now be explained in more detail by defining 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 the small delay case is treated 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 using the newly received values, while the other values in $\hat{y}$ and $\hat{u}$ remain the same, as no additional information has been received for them. This leads to the constrained data exchange expressed as

$$\begin{cases} \hat{y}(t) = \Gamma^y_{\sigma_k} y(t_k) + (I - \Gamma^y_{\sigma_k})\hat{y}(t_k) \\ \hat{u}(t) = \Gamma^u_{\sigma_k} u(t_k) + (I - \Gamma^u_{\sigma_k})\hat{u}(t_k) \end{cases} \tag{72}$$

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 the diagonal matrix given by

$$\Gamma_i = \operatorname{diag}\left(\gamma_{i,1}, \ldots, \gamma_{i,n_y+n_u}\right), \tag{73}$$

where $\sigma_k = i$ and the elements $\gamma_{i,j}$, $j \in \{1,\ldots,n_y\}$, are equal to one, if plant output y^j is in node i, elements $\gamma_{i,j+n_y}$, $j \in \{1,\ldots,n_u\}$, are equal to one, if controller output u^j is in node i, and are zero elsewhere. Note that (72) is directly related to (50) 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 (72) indicates which node is given access to the network at transmission instant t_k, $k \in \mathbb{N}$. Indeed, (72) 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. 11. It is assumed that the variations in the transmission interval and delays are bounded and are contained in the sets $[h_{\min}, h_{\max}]$ and $[\tau_{\min}, \tau_{\max}]$, respectively, with $h_{\max} \geq h_{\min} \geq 0$ and $\tau_{\max} \geq \tau_{\min} \geq 0$. Since it is assumed that each transmission delay τ_k is smaller than the corresponding transmission interval h_k, it holds that $(h_k, \tau_k) \in \Psi$, for all $k \in \mathbb{N}$, where

$$\Psi := \left\{ (h,\tau) \in \mathbb{R}^2 \mid h \in [h_{\min}, h_{\max}], \tau \in [\tau_{\min}, \min\{h, \tau_{\max}\}) \right\}. \tag{74}$$

Note that in comparison with Section 4.1, h_{mati} would correspond to $h_{\max}$ and τ_{mad} to $\tau_{\max}$. However, in Section 4.1 it was assumed that $\tau_{\min} = 0$ and $h_{\min} = \delta$, where δ could be chosen arbitrarily small, due to the emulation type of approach, while that is not the case here. Therefore, here the different notation using $h_{\min}$, $h_{\max}$, $\tau_{\min}$ and $\tau_{\max}$ is used.

To analyse stability of the NCS described above, it is transformed into a discrete-time model. In this framework, a discrete-time equivalent of (70) is needed. Additionally, when a continuous-time controller is used, also a discretization of (71a) is needed. To arrive at this description, define the network-induced error as

$$\begin{cases} e^y(t) := \hat{y}(t) - y(t) \\ e^u(t) := \hat{u}(t) - u(t). \end{cases} \tag{75}$$

By exact discretization of (70) and/or (71a) 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, 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 (71a) and (71b), respectively. In the third model, it is assumed that the controller is given by (71a) and that only the plant outputs y are transmitted over the network and u are sent continuously via an ideal nonnetworked connection. This particular case is included, because it is often used in examples in NCS literature, e.g., for the benchmark example of the batch reactor as discussed before in Section 4.1.

The NCS model with continuous-time controller (71a)

For an NCS having continuous-time controller (71a), the complete NCS model is obtained by combining (72), (75) with exact discretizations of plant (70) and controller (71a) and defining

$$\bar{x}_k := \begin{bmatrix} x_k^{p\top} & x_k^{c\top} & e_k^{y\top} & e_k^{u\top} \end{bmatrix}^\top. \tag{76}$$

This results in the discrete-time model

$$\bar{x}_{k+1} = \underbrace{\begin{bmatrix} A_{h_k} + E_{h_k}BDC & E_{h_k}BD - E_{h_k-\tau_k}B\Gamma_{\sigma_k} \\ C(I - A_{h_k} - E_{h_k}BDC) & I - D^{-1}\Gamma_{\sigma_k} + C(E_{h_k-\tau_k}B\Gamma_{\sigma_k} - E_{h_k}BD) \end{bmatrix}}_{=:\tilde{A}_{\sigma_k,h_k,\tau_k}} \bar{x}_k \tag{77}$$

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

$$A_{h_k} := \mathrm{diag}(e^{A^p h_k}, e^{A^c h_k}), \qquad B := \begin{bmatrix} 0 & B^p \\ B^c & 0 \end{bmatrix}, C \qquad := \mathrm{diag}(C^p, C^c), \tag{79a}$$

$$D := \begin{bmatrix} I & 0 \\ D^c & I \end{bmatrix},\ E_\rho := \mathrm{diag}(\int_0^\rho e^{A^p s} ds, \int_0^\rho e^{A^c s} ds), \quad \rho \in \mathbb{R}. \tag{79b}$$

The NCS model with discrete-time controller (71b)

For an NCS having controller (71b), the complete NCS model is obtained by combining (71b), (72), (75), and an exact discretization of the continuous-time plant (70), also resulting in (77), in which now

$$A_{h_k} := \mathrm{diag}(e^{A^p h_k}, A^c), \qquad B := \begin{bmatrix} 0 & B^p \\ B^c & 0 \end{bmatrix}, C \qquad := \mathrm{diag}(C^p, C^c), \tag{80a}$$

$$D := \begin{bmatrix} I & 0 \\ D^c & I \end{bmatrix},\ E_\rho := \mathrm{diag}(\int_0^\rho e^{A^p s} ds, I), \quad \rho \in \mathbb{R}. \tag{80b}$$

The NCS model if only y is transmitted over the network

In this case it is assumed that only the outputs of the plant are transmitted over the network and the controller communicates its values continuously and without delay. Therefore it holds that $u(t) = \hat{u}(t)$, for all $t \in \mathbb{R}^+$, which allows the combination of (70) and (71a) into

$$\begin{bmatrix} \dot{x}^p(t) \\ \dot{x}^c(t) \end{bmatrix} = \begin{bmatrix} A^p & B^p C^c \\ 0 & A^c \end{bmatrix} \begin{bmatrix} x^p(t) \\ x^c(t) \end{bmatrix} + \begin{bmatrix} B^p D^c \\ B^c \end{bmatrix} \hat{y}(t). \tag{81}$$

Since $\hat{y}$ is still updated according to (72), the evolution of the states between t_k and $t_{k+1} = t_k + h_k$ can also be described by exact discretization. In this case, (76) reduces to

$$\bar{x}_k := \begin{bmatrix} x_k^{p\top} & x_k^{c\top} & e_k^{y\top} \end{bmatrix}^\top, \tag{82}$$

resulting in (77), in which

$$A_{h_k} := e^{\begin{bmatrix} A^p & B^p C^c \\ 0 & A^c \end{bmatrix} h_k}, \qquad B := \begin{bmatrix} B^p D^c \\ B^c \end{bmatrix}, \qquad C := \begin{bmatrix} C^p & 0 \end{bmatrix}, \tag{83a}$$

$$D := I, \; E_\rho := \int_0^\rho e^{\begin{bmatrix} A^p & B^p C^c \\ 0 & A^c \end{bmatrix} s} ds, \quad \rho \in \mathbb{R}. \tag{83b}$$

Protocols as a Switching Function

Based on the previous modeling steps, the NCS is reformulated as the discrete-time switched uncertain system (77). In this framework, protocols are considered as the switching function determining σ_k. The two protocols mentioned before, namely the Try-Once-Discard (TOD) and the Round-Robin (RR) protocol, are considered and generalized into the classes of ‘quadratic’ and ‘periodic’ protocols, respectively.

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

$$\sigma_k = \arg \min_{i=1,\dots,N} \bar{x}_k^\top P_i \bar{x}_k, \tag{84}$$

where P_i, $i \in \{1, \dots, N\}$, are certain given matrices. In fact, the TOD protocol belongs to this class of protocols, see [20, 21].

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

$$\sigma_{k+\tilde{N}} = \sigma_k \tag{85}$$

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 (77) and one of the protocols, characterized by (84) or (85). 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 20. If there is only one node $N = 1$ and $\Gamma_i = I$, the setting of Section 3.2 for the case of small delays is recovered. If in addition, $\tau_{\max} = 0$ and $h_{\min} = h_{\max}$, then the NCS reduces to a standard sampled-data system (in case of a continuous-time plant and discrete-time controller), see, e.g., [8, 23].

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 the terms $A_{h_k}, A_{h_k-\tau_k}, E_{h_k}$ and $E_{h_k-\tau_k}$ in (77)). To convert these descriptions into a suitable form for robust stability analysis, a polytopic overapproximation method is exploited. Essentially any of the available overapproximation methods (e.g., the one based on the real Jordan form discussed before) can be applied. However, in [20], in which the above modeling is presented, a combination of gridding and norm-bounding is combined into a new and efficient method. The gridding method in [20] has the advantage that if the exact discrete-time model (so before the overapproximation) is exponentially stable proven by a parameter-dependent quadratic Lyapunov function [15], then the overapproximated polytopic system with a sufficiently dense set of grid points in Ψ also has a parameter-dependent quadratic Lyapunov function. The Lyapunov function for the polytopic system can then be found by LMIs. In other words, if the original NCS system is "quadratically stable," then the LMIs derived in [20] will prove this for a sufficiently dense gridding. The reader is referred to [20, 21] for full details. Actually, in a recent paper [38] a comparison was made between the various polytopic overapproximation methods and it was argued that the gridding method of [20, 21] has the most favorable properties of the studied methods.

4.2.3 Application to the batch reactor

The exact same setup will be analyzed 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 [20] the aim is to construct combinations of $h_{\max}$ and $\tau_{\max}$ for which the NCS is stable, and it is assumed that $\tau_{\min} = 0$ and $h_{\min} = 10^{-4}$. This results in tradeoff curves, as shown in Fig. 12. These tradeoff curves can be used to impose or select a suitable network with certain communication delay and bandwidth requirements. Note that bandwidth is inversely proportional with the transmission interval.

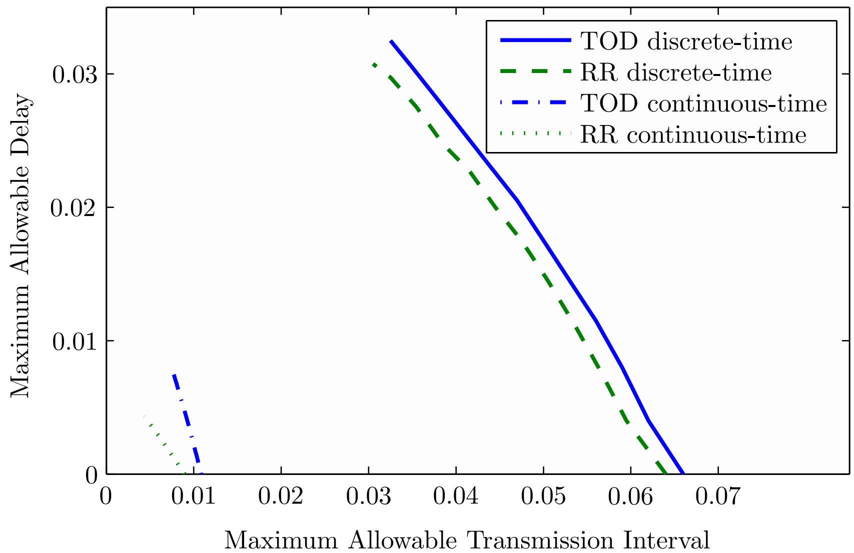

Fig. 12 Tradeoff curves between allowable transmission intervals and transmission delays for two different protocols, where "*continuous* − *time*″ refers to the approach in Section 4.1, while the other curves refer to the discrete-time approach as discussed in Section 4.2.

Moreover, in Fig 12, also the tradeoff curves as obtained for the continuous-time approach (Subsection 4.1.3) are given. It can be observed that the discrete-time approach provides less conservative results than the continuous-time approach (at least for this example). More interestingly, in case there is no delay, i.e., $\tau_{\min} = \tau_{\max} = 0$, the maximum allowable transmission interval $h_{\max}$ obtained in [6], which provide the least conservative results known in literature so far, was $h_{\max} = 0.0108$, while the discrete-time approach results in $h_{\max} = 0.066$. In [89], the largest $h_{\max}$ for which stability can be guaranteed was estimated (using simulations) to be approximately 0.08 for the TOD protocol, so the result of $h_{\max} = 0.066$ as found here already approaches this value accurately. Furthermore, for the RR protocol, [6] provides the bound $h_{\max} = 0.009$ in the delay-free case, while the discrete-time approach yields stability for $h_{\max} = 0.064$. Also in [89], for a constant transmission interval, i.e. $h_{\min} = h_{\max}$, the bound 0.0657 was obtained for the RR protocol. Obviously, the case where the transmission interval is constant, provides an upper bound on the true maximum allowable transmission interval (MATI). Therefore, one can conclude that for this example, the discrete-time methodology reduces conservatism significantly in comparison to existing approaches including the continuous-time approach discussed in Section 4.1. Furthermore, even approximates known estimates of the true MATI ($h_{mati} = h_{\max}$) closely. In addition, the discrete-time approach applies to situations (non-zero lower bounds and discrete-time controllers, see [20] 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 (emulation) approach results in *hybrid inclusions* with flows and resets [30], 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 (emulation) approach as presented in Subsection 4.1 applies to general *continuous-time* nonlinear plants and controllers, and general (UGES) protocols. In addition, in the continuous-time approach $\mathscr{L}_p$ gain analysis of the original continuous-time NCS can be done in a straightforward manner (see [37]), while this not true for the discrete-time approach. However, the continuous-time approach does not allow for discrete-time controllers and cannot handle nonzero lower bounds on the transmission intervals h_k and delays τ_k. The discrete-time approach as discussed in Subsection 4.2 can allow for both *continuous-time* and *discrete-time* controllers and *nonzero* 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 discussed various approaches to the modeling, stability analysis and stabilizing controller synthesis of NCSs with varying delays, varying transmission intervals, packet dropouts and communication constraints. The methods discussed in this chapter all assume hard bounds on the size of the varying delays, transmission intervals and the maximal number of subsequent dropouts. Roughly speaking, three main lines can be distinguished, as summarized below.

(i) The discrete-time approach is based on a discrete-time NCS model, which can be used for both discrete-time and continuous-time linear controllers and linear plants. LMI-based stability conditions are derived by using common or parameter-dependent quadratic Lyapunov function for an overapproximated polytopic model. Different methods are available for performing the polytopic overapproximation of the discretized NCS model in which the delay and sampling interval uncertainties appear in an exponential fashion, see [38]. Within this research line the largest number of network-induced imperfections are treated in [20] that includes varying sampling intervals and delays, dropouts and communication constraints. Both discrete-time and continuous-time controllers can be handled. However, in [20] only delays smaller than the sampling interval are considered. For the most comprehensive discrete-time approach including large delays, but without communication constraints, we refer to [10, 14].

(ii) The sampled-data approach uses (impulsive) delay-differential equations describing continuous-time sampled-data NCS models with discrete-time controllers. Extensions of the classical Lyapunov-Krasovskii functionals are exploited to give stability guarantees for *linear* plants and controllers. The conditions also result in LMIs. Communication constraints have not (yet) been treated using (impulsive) delay-differential equations.

(iii) The continuous-time (emulation) approach studies continuous-time sampled-data NCS models with continuous-time controllers. Continuous-time Lyapunov functions are constructed by combining individual Lyapunov functions of the network-free closed-loop system and the network protocol (or adopting directly small gain arguments) to assess stability of the NCS. Typically only continuous-time controllers are treated at present, however both controller and plant can be nonlinear.

Although already three main research lines of NCSs exist for stability analysis (where we did not even discuss the stochastic approaches) and many papers are written, at present there are almost no complete frameworks that can handle all the 5 mentioned network-induced imperfections: (i) Variable sampling/transmission intervals; (ii) variable communication delays; (iii) Packet dropouts (iv) Communication constraints; (v) Quantization errors. Only recently one approach was presented in [34] that includes all 5 imperfections under quite restricted conditions (small delay case, particular quantizers, continuous-time controllers, etc.) The availability of a complete analysis and design frameworks based on either one of the main lines as discussed in this chapter or possibly on a completely new line, would be desirable. At present such a framework is not available, certainly not with the analysis and design techniques implemented in suitable software tools. One of the goals of this chapter was to survey the main techniques and discuss the open problems, which will hopefully stimulate further research in this direction in order to develop this envisioned framework and toolset in the near future.

Particular attention should be given to controller design methods, considering that for the three main lines these methods are still rather limited and extensions are needed. Indeed, most of the design techniques lead to static feedback controllers, while in industrial practice there is a strong need for output-based dynamic controllers. In the discrete-time approach 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 any available method for the design of stabilizing controllers for nonlinear systems, which is in general a nontrivial task. 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 favorable properties through the presented stability conditions. In addition, one might wonder if continuous-time controllers are useful in 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 option is to implement the continuous-time controller using numerical integration schemes, however it is unclear if these controllers will have the desired properties in the end. A clear advantage of the emulation-based results is the fact that the results can employed to tackle nonlinear NCSs, whereas the results for the discrete-time approach are generally limited to the case of linear plants. Given the benefits of the discrete-time approach in terms of designing discrete-time controllers tailored to deal with non-zero sampling intervals an delays, it would be interesting to pursue the development of a discrete-time framework for nonlinear NCSs as well. Also for the sampled-data approach constructive design methods seem to be missing in the literature, certainly in an efficient form. Although in the discrete-time efficient LMI-based synthesis conditions for state feedbacks were given, as mentioned before, the design of output-based dynamic controllers is still an open problem. Observer-based control design might provide attractive solutions, especially since the observer might also be used to compensate for delays, varying sampling times and packet losses. In summary, one can state that the controller design and controller/protocol co-design techniques for NCSs are still in their infancy and deserve a lot of attention in the years to come.

6 Acknowledgement

The authors are grateful to João Hespanha, Nick Bauer and Tijs Donkers for giving detailed comments on an earlier version of this manuscript.

References

1. Antunes, D., Hespanha, J., Silvestre, C.: Control of impulsive renewal systems: Application to direct design in networked control. In: IEEE Conference on Decision and Control, pp. 6882–6887 (2009)
2. Antunes, D., Hespanha, J., Silvestre, C.: Stability of impulsive systems driven by renewal processes. In: Proc. Amer. Contr. Conf., pp. 4032–4037 (2009)
3. Balluchi, A., Murrieri, P., Sangiovanni-Vincentelli, A.L.: Controller synthesis on non-uniform and uncertain discrete-time domains. In: The 8^{th} International Workshop on Hybrid Systems: Computation and Control(HSCC05). Zurich, Switzerland (2005)
4. Brockett, R.: Stabilization of motor networks. In: Proc. of the 34th IEEE Conf. on Decision and Control, vol. 2, pp. 1484–1488 (1995)
5. Brockett, R., Liberzon, D.: Quantized feedback stabilization of linear systems. IEEE Trans. Autom. Control **45**, 1279–1289 (2000)
6. Carnevale, D., Teel, A., Nešić, D.: A Lyapunov proof of improved maximum allowable transfer interval for networked control systems. IEEE Trans. Aut. Control **52**, 892–897 (2007)
7. Chaillet, A., Bicchi, A.: Delay compensation in packet-switching networked controlled sytems. In: IEEE Conf. Decision and Control, pp. 3620–3625 (2008)
8. Chen, T., Francis, B.: Optimal Sampled-data Control Systems. Springer-Verlag, London, UK (1995)
9. Cloosterman, M.: Control over Communication Networks: Modeling, Analysis, and Synthesis. PhD thesis Eindhoven University of Technology (2008)

10. Cloosterman, M., Hetel, L., van de Wouw, N., Heemels, W., Daafouz, J., Nijmeijer, H.: Controller synthesis for networked control systems. Automatica (2010)
11. Cloosterman, M., van de Wouw, N., Heemels, W., Nijmeijer, H.: Robust stability of networked control systems with time-varying network-induced delays. In: Proc. IEEE Conf. on Decision and Control, pp. 4980–4985. San Diego, USA (2006)
12. Cloosterman, M., van de Wouw, N., Heemels, W., Nijmeijer, H.: Stability of networked control systems with large delays. In: 46th IEEE Conference on Decision and Control, pp. 5017 – 5022 (2007)
13. Cloosterman, M., van de Wouw, N., Heemels, W., Nijmeijer, H.: Stabilization of networked control systems with large delays and packet dropouts. In: American Control Conference, pp. 4991 – 4996 (2008)
14. Cloosterman, M., van de Wouw, N., Heemels, W., Nijmeijer, H.: Stability of networked control systems with uncertain time-varying delays. IEEE Trans. Autom. Control **54**(7), 1575–1580 (2009)
15. Daafouz, J., Bernussou, J.: Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. Systems & Control Letters **43**, 355–359 (2001)
16. Daafouz, J., Riedinger, P., Iung, C.: Stability analysis and control synthesis for switched systems: A switched Lyapunov function approach. IEEE Transactions on Automatic Control **47**, 1883–1887 (2002)
17. Dačić, D., Nešić, D.: Quadratic stabilization of linear networked control systems via simultaneous protocol and controller design. Automatica **43**, 1145–1155 (2007)
18. Delchamps, D.: Stabilizing a linear system with quantized state feedback. IEEE Trans. Autom. Control **35**, 916–924 (1990)
19. Donkers, M., Heemels, W., Bernardini, D., Bemporad, A.: Stability analysis of stochastic networked control systems. In: Proc. Amer. Contr. Conf. (2010)
20. Donkers, M., Heemels, W., Hetel, L., van de Wouw, N., Steinbuch, M.: Stability analysis of networked control systems using a switched linear systems approach. Submitted for journal publication
21. Donkers, M., Hetel, L., Heemels, W., van de Wouw, N., Steinbuch, M.: Stability analysis of networked control systems using a switched linear systems approach. In: Hybrid Systems: Computation and Control, Lecture Notes in Computer Science, pp. 150–164. Springer Verlag (2009)
22. Dritsas, L., Tzes, A.: Robust stability analysis of networked systems with varying delays. International Journal of Control (2009). URL DOI 10.1080/00207170903061705
23. Franklin, G.F., Powell, J.D., Workman, M.: Digital control of dynamic systems. Addison-Wesley Pub. Co. Inc., Reading, USA (1990)
24. Fridman, E., Seuret, A., Richard, J.: Robust sampled-data stabilization of linear systems: an input delay approach. Automatica **40**, 1441–1446 (2004)
25. Fujioka, H.: Stability analysis for a class of networked/embedded control systems: A discrete-time approach. In: Proc. of the American Control Conf., pp. 4997–5002 (2008)
26. Gao, H., Chen, T., Lam, J.: A new delay system approach to network-based control. Automatica **44**(1), 39–52 (2008)
27. Garcia-Rivera, M., Barreiro, A.: Analysis of networked control systems with drops and variable delays. Automatica **43**, 2054–2059 (2007)
28. Gielen, R., Olaru, S., Lazar, M.: 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)
29. Gielen, R., Olaru, S., Lazar, M., Heemels, W., van deWouw, N., Niculescu, S.I.: On polytopic approximations as a modelingframework for systems with time-varying delays. Automatica (2010)
30. Goebel, R., Sanfelice, R., Teel, A.: Hybrid dynamical systems. IEEE Control Systems Magazine **29**(2), 28–93 (2009)
31. Gupta, V., Chung, T., Hassibi, B., Murray, R.: On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. Automatica **42**(2), 251–260 (2006)

32. Hea, Y., Wang, Q.G., C. Lin, M.W.: Delay-range-dependent stability for systems with time-varying delay. Automatica **43**, 371376 (2007)
33. Heemels, W., Gorter, R., van Zijl, A., v.d. Bosch, P., Weiland, S., Hendrix, W., Vonder, M.: Asynchronous measurement and control: a case study on motor synchronisation. Control Engineering Practice **7**(12), 1467–1482 (1999)
34. Heemels, W., Nešić, D., Teel, A., van de Wouw, N.: Networked and quantized control systems with communication delays. In: Proc. Joint 48th IEEE Conference on Decision and Control (CDC) and 28th Chinese Control Conference, Shanghai, China, pp. 7929–7935 (2009)
35. Heemels, W., Siahaan, H., Juloski, A., Weiland, S.: Control of quantized linear systems: an l_1-optimal control approach. In: Proc. American Control Conference, pp. 3502–3507. Denver, USA (2003)
36. Heemels, W., Teel, A., van de Wouw, N., Nešić, D.: Networked control systems with communication constraints: tradeoffs between sampling intervals and delays. In: Proc. European Control Conference in Budapest, Hungary (2009)
37. Heemels, W., Teel, A., van de Wouw, N., Nešić, D.: Networked control systems with communication constraints: Tradeoffs betweens transmission intervals, delays and performance. IEEE Trans. Autom. Control (2010)
38. Heemels, W., van de Wouw, N., Gielen, R., Donkers, M., Hetel, L., Olaru, S., Lazar, M., Daafouz, J., Niculescu, S.: Comparison of overapproximation methods for stability analysis of networked control systems. 13th International Workshop on Hybrid Systems: Computation and Control, Stockholm, Sweden (2010)
39. Hespanha, J., Naghshtabrizi, P., Xu, Y.: A survey of recent results in networked control systems. Proc. of the IEEE pp. 138–162 (2007)
40. Hetel, L., Cloosterman, M., van de Wouw, N., Heemels, W., Daafouz, J., Nijmeijer, H.: 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, pp. 7911–7916 (2009)
41. Hetel, L., Daafouz, J., Iung, C.: Stabilization of arbitrary switched linear systems with unknown time-varying delays. IEEE Trans. Autom. Control **51**(10), 1668–1674 (2006)
42. Hetel, L., Daafouz, J., Iung, C.: Analysis and control of LTI and switched systems in digital loops via an event-based modeling. International Journal of Control **81**(7), 1125–1138 (2008)
43. Horn, R., Johnson, C.R.: Matrix Analysis. Cambridge University Press (1985)
44. Hristu, D., Morgansen, K.: Limited communication control. Systems & Control Letters **37**(4), 193–205 (1999)
45. Ishii, H.: H-infty control with limited communication and message losses. Systems and Control Letters **57**, 322–331 (2008)
46. Ishii, H., Francis, B.: Stabilization with control networks. Automatica **38**, 1745–1751 (2002)
47. Kao, C.Y., Lincoln, B.: Simple stability criteria for systems with time-varying delays. Automatica **40**, 1429–1434 (2004)
48. Kothare, M., Balakrishnan, V., Morari, M.: Robust constrained model predictive control using linear matrix inequalities. Automatica **32**, 1361–1379 (1996)
49. Krtolica, R., Ozguner, U., Chan, H., Goktas, H., Winkelman, J., Liubakka, M.: Stability of linear feedback systems with random communication delays. In: Proc. American Control Conf., pp. 2648–2653 (1991)
50. Liberzon, D.: On stabilization of linear systems with limited information. IEEE Trans. Autom. Control **48**(2), 304–307 (2003)
51. Liberzon, D.: Quantization, time delays, and nonlinear stabilization. IEEE Trans. Autom. Control **51**(7), 1190–1195 (2006)
52. Matveev, A., Savkin, A.: The problem of state estimation via asynchronous communication channels with irregular transmission times. IEEE Trans. Autom. Control **48**(4), 670–676 (2003)
53. Mirkin, L.: Some remarks on the use of time-varying delay to model sample-and-hold circuits. IEEE Transactions on Automatic Control **52**(6), 1109–1112 (2007). DOI 10.1109/TAC.2007.899053

54. Montestruque, L., Antsaklis, P.: Stochastic stability for model-based networked control systems. In: Proc. American Control Conf., pp. 4119 – 4124 (2003)
55. Montestruque, L., Antsaklis, P.: Stability of model-based networked control systems with time-varying transmission times. IEEE Trans. Autom. Control **49**(9), 1562–1572 (2004)
56. Naghshtabrizi, P., Hespanha, J.: 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, pp. 848–853. Seville, Spain (2005)
57. Naghshtabrizi, P., Hespanha, J.: 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). URL http://www.ece.ucsb.edu/ hespanha/published
58. Naghshtabrizi, P., Hespanha, J., Teel, A.: Stability of delay impulsive systems with application to networked control systems. In: Proc. American Control Conference, pp. 4899–4904. New York, USA (2007)
59. Naghshtabrizi, P., Hespanha, J., Teel, A.: Exponential stability of impulsive systems with application to uncertain sampled-data systems. Systems & Control Letters **57**(5), 378–385 (2008)
60. Naghshtabrizi, P., Hespanha, J.P., Teel, A.R.: On the robust stability and stabilization of sampled-data systems: A hybrid system approach. In: Proc. of the 45th Conf. on Decision and Contr. (2006)
61. Nair, G., Evans, R.: Stabilizability of stochastic linear systems with finite feedback data rates. SIAM J. Control Optim. **43**, 413–436 (2004)
62. Nešić, D., Liberzon, D.: A unified framework for design and analysis of networked and quantized control systems. IEEE Trans. Autom. Control **54**(4), 732–747 (2009)
63. Nešić, D., Teel, A.: Input-output stability properties of networked control systems. IEEE Trans. Autom. Control **49**(10), 1650–1667 (2004)
64. Nešić, D., Teel, A.: Input-to-state stability of networked control systems. Automatica **40**, 2121–2128 (2004)
65. Nešić, D., Teel, A.R., Sontag, E.D.: Formulas relating KL-stability estimates of discrete-time and sampled-data nonlinear systems. Systems & Control Letters **38**(1), 49–60 (1999)
66. Nilsson, J.: Real-Time Control Systems with Delays. PhD thesis, Dept. of Automatic Control, Lund Inst. of Techn., Lund, Sweden (1998)
67. Nilsson, J., Bernhardsson, B., Wittenmark, B.: Stochastic analysis and control of real-time systems with random time delays. Automatica **34**, 57 – 64 (1998)
68. Olaru, S., Niculescu, S.I.: Predictive Control for Linear Systems with Delayed Input Subject to Constraints. In: Proceedings of 17th IFAC World Congress 2008 17th IFAC World Congress 2008, pp. CD–ROM. Seul, Korea (2008)
69. de Oliveira, M.C., Bernussou, J., Geromel, J.C.: A new discrete-time robust stability condition. Systems & Control Letters **37**, 261–265 (1999)
70. Pan, Y.J., Marquez, H.J., Chen, T.: Stabilization of remote control systems with unknown time varying delays by LMI techniques. Int. Journal of Control **79**(7), 752–763 (2006)
71. Rehbinder, H., Sanfridson, M.: Scheduling of a limited communication channel for optimal control. Automatica **40**(3), 491–500 (2004)
72. Sala, A.: Computer control under time-varying sampling period: An LMI gridding approach. Automatica **41**(12), 2077–2082 (2005)
73. van der Schaft, A.: L2-Gain and Passivity in Nonlinear Control. Springer, 2nd edition (1999)
74. Schenato, L.: To zero or to hold control inputs with lossy links? IEEE Trans. Autom. Control (To appear)
75. van Schendel, J., Donkers, M., Heemels, W., van de Wouw, N.: On dropout modelling for stability analysis of networked control systems. In: Proc. American Control Conf. (2010)
76. Seiler, P., Sengupta, R.: An $\mathscr{H}_\infty$ approach to networked control. IEEE Trans. Autom. Control **50**, 356 – 364 (2005)
77. Shi, Y., Yu, B.: Output feedback stabilization of networked control systems with random delays modeled by markov chains. IEEE Trans. Autom. Control **54**, 1668 – 1674 (2009)
78. Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M., Sastry, S.: Kalman filtering with intermittent observations. IEEE Trans. Autom. Control **49**(9), 1453–1464 (2004)

79. Skaf, J., Boyd, S.: Analysis and synthesis of state-feedback controllers with timing jitter. IEEE Transactions on Automatic Control **54**(3) (2009)
80. Smith, S., Seiler, P.: Estimation with lossy measurements: jump estimators for jump systems. IEEE Trans. Autom. Control **48**(12), 2163–2171 (2003)
81. Suh, Y.: Stability and stabilization of nonuniform sampling systems. Automatica **44**, 3222–3226 (2008)
82. Tabbara, M., Nešić, D.: Input-output stability of networked control systems with stochastic protocols and channels. IEEE Trans. Autom. Control **53**, 1160–1175 (2008)
83. Tabbara, M., Nešić, D., Teel, A.: Stability of wireless and wireline networked control systems. IEEE Trans. Autom. Control **52**(9), 1615–1630 (2007)
84. Tatikonda, S., Mitter, S.: Control under communication constraints. IEEE Trans. Autom. Control **49**, 1056–1068 (2004)
85. Tipsuwan, Y., Chow, M.Y.: Control methodologies in networked control systems. Control Engineering Practice **11**, 1099–1111 (2003)
86. Tsumura, K., Ishii, H., Hoshina, H.: Tradeoffs between quantization and packet loss in networked control of linear systems. Automatica (2009). URL doi:10.1016/j.automatica.2009.09.030
87. van der Schaft, A., Schumacher, J.: An Introduction to Hybrid Dynamical Systems, *LNCIS*, vol. 251. Springer-Verlag, London (2000)
88. Walsh, G., Belidman, O., Bushnell, L.: Asymptotic behavior of nonlinear networked control systems. IEEE Trans. Automat. Contr. **46**, 1093–1097 (2001)
89. Walsh, G., Ye, H., Bushnell, L.: Stability analysis of networked control systems. IEEE Trans. on Control Systems Technology **10**(3), 438–446 (2002)
90. Wang, Y., Yang, G.: Multiple communication channels-based packet dropout compensation for networked control systems. IET Control Theory and Applications **2**, 717–727 (2008)
91. Wei, G., Wang, Z., He, X., Shu, H.: Filtering for networked stochastic time-delay systems with sector nonlinearity. IEEE Trans. Circuits and Systems II: Express Briefs **56**, 71–75 (2009)
92. Wittenmark, B., Nilsson, J., Törngren, M.: Timing problems in real-time control systems. In: Proc. of the Amer. Control Conf., pp. 2000–2004. Seattle, USA (1995)
93. van de Wouw, N., Naghshtabrizi, P., Cloosterman, M., Hespanha, J.: Tracking control for sampled-data systems with uncertain sampling intervals and delays. Intern. Journ. Robust and Nonlinear Control **20**(4), 387–411 (2010)
94. van de Wouw, N., Naghshtabrizi, P., Cloosterman, M., Hespanha, J.P.: Tracking control for networked control systems. In: Proc. of the 46th IEEE Conference on Decision and Control, pp. 4441–4446. New Orleans, LA, USA (2007)
95. van de Wouw, N., Nešić, D., Heemels, W.: A discrete-time framework for stability analysis of nonlinear networked control systems. Submitted (2010)
96. Xie, G., Wang, L.: Stabilization of networked control systems with time-varying network-induced delay. In: Proc. of the 43rd IEEE Conference on Decision and Control, pp. 3551–3556. Atlantis, Paradise Island, Bahamas (2004)
97. Yang, F., Wang, Z., Hung, Y., Gani, M.: $\mathscr{H}_\infty$ control for networked systems with random communication delays. IEEE Trans. Autom. Control **51**, 511 – 518 (2006)
98. Yang, T.C.: Networked control system: a brief survey. IEE Proc.-Control Theory Appl. **153**(4), 403–412 (2006)
99. Yu, M., Wang, L., Chu, T.: Sampled-data stabilization of networked control systems with nonlinearity. IEE Proc.-Control Theory Appl. **152**(6), 609–614 (2005)
100. Yue, D., Han, Q.L., Peng, C.: State feedback controller design of networked control systems. IEEE Trans. on circuits and systems **51**(11), 640–644 (2004)
101. Zhang, L., Shi, Y., Chen, T., Huang, B.: A new method for stabilization of networked control systems with random delays. IEEE Trans. Autom. Control **50**(8), 1177–1181 (Aug. 2005)
102. Zhang, W., Branicky, M., Phillips, S.: Stability of networked control systems. IEEE Control Systems Magazine **21**(1), 84–99 (2001)