Stabilization of Networked Control Systems with Large Delays and Packet Dropouts

M.B.G. Cloosterman, N. van de Wouw, W.P.M.H. Heemels and H. Nijmeijer

Abstract— We consider the stabilization problem for Networked Control Systems (NCSs) with uncertain, time-varying network-induced delays and a bounded number of subsequent packet dropouts. A discrete-time model, describing a NCS with packet dropouts and time-varying delays, that can be both smaller and larger than the sampling interval, is presented. Based on this NCS model sufficient LMI conditions are proposed for the stability analysis and controller synthesis problem for two different controllers, i.e. a feedback controller that depends on both the state and the past control inputs and a state-feedback controller. The applicability of both controllers is compared. Moreover, the stability and controller synthesis LMIs allow for a performance analysis in terms of a lower bound for the transient decay rate of the response. The results are illustrated by application to a typical motion control example.

I. INTRODUCTION

Networked Control Systems (NCSs) are systems where the control loop is closed over a communication channel. NCSs have received increasing attention in recent years [1]–[4]. Advantages are, e.g. low cost and flexible architectures. The disadvantages are time-delays and packet dropouts that are caused by the unreliability and shared use of the network. The nature of the time-delays and the possibility of packet dropouts depends on the chosen communication protocol and network [5]–[7]. Note that all communication networks suffer from some delay, depending on the amount of network traffic, the device delays and the network data-rate. Packet dropouts are known to occur in wireless Ethernet, but also in other Ethernet solutions if many collisions take place.

An understanding of the effects of packet dropouts and time-varying delays on the stability (or stabilization) and performance of the NCS is needed to use NCSs in an industrial environment. In the NCS literature, it is assumed that the time-varying delays are either bounded (often by the sampling interval or a multiple of it), without any knowledge on the variation of the delay within these bounds, or modeled stochastically with a Bernoulli distribution or a Markov chain, see e.g. [4], [6]. Packet dropouts are either modeled deterministically [1] or modeled as a Markov chain, see [4] and the references therein. Most of the literature that deals with stability and stabilization of NCSs focuses either on packet dropouts [1], [8], [9] or on time-varying delays (see e.g. [1], [10], [11]). In [12] and [13] stability

and stabilization of NCSs with packet dropouts and constant delays are investigated, based on a discrete-time NCS representation. Continuous-time modeling approaches, based on (impulsive) delay-differential equations, including packet dropouts and time-varying delays, are described in [14]–[16]. For these models, both stability analysis and controller synthesis methods are proposed and time-varying delays both smaller and larger than the sampling interval are included. In [17], the stability and disturbance attenuation of a NCS with time-varying delays and packet dropouts are investigated, based on a switched system approach. In their model and analysis, it is assumed that the delays take values in a finite set, that is upper-bounded by the sampling interval.

In this paper, we obtain stability and stabilizability conditions in terms of LMIs for a NCS with packet dropouts and time-varying delays that may be larger than the sampling interval, based on a discrete-time NCS representation. Our conditions are an alternative approach to the approaches in [14]–[16] and an extension to the discrete-time approaches that consider only constant delays or time-varying delays smaller than the sampling interval and packet dropouts. Moreover, we assume that the delays take values from a bounded set, containing an infinite number of values (i.e. $\tau_k \in [\tau_{\min}, \tau_{\max}]$), while in [17], it is assumed that $\tau_k \in \{0, 0.1h, 0.2h, \ldots, h\}$. Additionally, in this paper, we use the controller synthesis results to obtain a performance bound on the transient behavior of the NCS.

The outline of the paper is as follows. In Section II, a NCS model with time-delays and packet dropouts is proposed. In Section III, solutions to the stability analysis and controller synthesis problem are proposed. In Section IV, an illustrative example is given dealing with the stability analysis and controller synthesis results and the performance bound in terms of convergence of the transient behavior.

Notation: We denote the transpose of a matrix A by A^T and we write $P > 0$ (or $P < 0$) for a positive (or negative) definite matrix. With $*$ we denote the symmetric part of a matrix and with $\dim(J)$ the dimension of the square matrix J, i.e. if $J \in \mathbb{R}^{m \times m}$, then $\dim(J) = m$. The diagonal operator is denoted as $diag(A_1, A_2) = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$. $\lfloor f \rfloor := \max\{k \in \mathbb{N} | k \leq f\}$ denotes the floor function of f and $\lceil f \rceil := \min\{k \in \mathbb{N} | k \geq f\}$ the ceil function of f.

II. NCS MODEL

In this section, we will derive a discrete-time NCS model that includes a bounded number of subsequent packet dropouts and time-delays where the variation can be larger
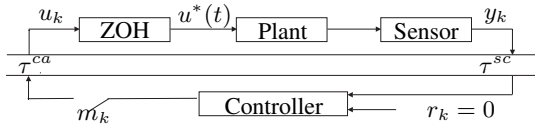
Fig. 1. Schematic overview of the networked control system with packet dropouts and network delays.

than the sampling interval. This discrete-time model originates from the discrete-time NCS description in [1] and [18]. The NCS is depicted schematically in Figure 1. It consists of a continuous-time plant that is connected over a network, where packet may be dropped (denoted by the variable m_k), to a discrete-time controller. The output measurements are sampled with a fixed sampling interval $h > 0$. In the model, the possibility that packets are lost, the computation time (τ^c) and the networked induced delays, i.e. the sensor-to-controller delay (τ^{sc}) and the controller-to-actuator delay (τ^{ca}) are taken into account. If the sensor acts in a time-driven fashion (i.e. sampling at the times $s_k = kh$, $k \in \mathbb{N}$), the controller and actuator act in an event-driven fashion (i.e. respond instantaneously to newly arrived data) and the controller is static and time-invariant, all three delays can be represented by a single delay $\tau_k := \tau_k^{sc} + \tau_k^c + \tau_k^{ca}$ [1]. In our model, without an observer, the loss of a packet between the sensor and controller results in no new control update being sent, which is similar to the loss of a packet between the controller and actuator.

To derive the NCS model, firstly, we define the parameter m_k that denotes whether or not a packet is lost:

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

The model has to describe both packet dropouts and message rejection, being the effect that more recent control data becomes available before the older data is implemented and therefore the older data is neglected. Therefore, the model should be able to describe situations in which not all data is used. Next, the basic continuous-time model of sampled-data systems [20], with a piecewise constant control input $u(t)$, will be adapted. Let us define $k^*(t) := \max\{k \in \mathbb{N} | (kh + \tau_k \leq t) \wedge (m_k = 0)\}$, which denotes the index of the most recent control input that is available at time t. The continuous-time NCS model is then given by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu^*(t) \\ u^*(t) &= u_{k^*(t)}, \text{ for } t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1}), \end{aligned} \quad (2)$$

with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ the system matrices, $x(t) \in \mathbb{R}^n$ the state at time $t \in \mathbb{R}$, $u^*(t) \in \mathbb{R}^m$ the continuous-time control input, $u_{k^*(t)} \in \mathbb{R}^m$ the discrete-time control input and τ_k the time-delay. Moreover, we consider full state-feedback ($y_k = x_k$) and if packet dropouts occur, it is assumed that the most recent control input remains active.

We assume that at maximum $\overline{\delta} \in \mathbb{N}$ subsequent packet dropouts occur. Then, it holds for m_k in (1) that:

$$\sum_{v=k-\overline{\delta}}^{k} m_v \leq \overline{\delta}, \quad (3)$$

which guarantees that from the control inputs $u_{k-\overline{\delta}}, u_{k-\overline{\delta}+1}, \ldots, u_k$ at least one control input is implemented. Let us introduce the class $\mathcal{S}$ of admissible sequences $\{(\tau_k, m_k)\}_{k \in \mathbb{N}}$ as follows:

$$\mathcal{S} := \big\{\{(\tau_k, m_k)\}_{k \in \mathbb{N}} : \tau_{\min} \leq \tau_k \leq \tau_{\max}, \\ \textstyle\sum_{v=k-\overline{\delta}}^{k} m_v \leq \overline{\delta}, \forall k \in \mathbb{N}\}, \quad (4)$$

that allows for both the occurrence of large delays and packet dropout. For the sake of brevity, we will use $\sigma := \{(\tau_k, m_k)\}_{k \in \mathbb{N}} \in \mathcal{S}$.

In Lemma 1, the general NCS description (2) is reformulated to make explicit which control inputs are active in the sampling interval $[s_k, s_{k+1})$. Such a formulation is needed to derive the discrete-time NCS model for large delays (incorporating message rejection) and packet dropouts that will be used for stability analysis and controller synthesis.

Lemma 1: Consider the continuous-time NCS as defined in (2) and the admissible sequences in $\mathcal{S}$. Define $\underline{d} := \lfloor \frac{\tau_{\min}}{h} \rfloor$, $\overline{d} := \lceil \frac{\tau_{\max}}{h} \rceil$. Then, the control action $u^*(t)$ in the sampling interval $[s_k, s_{k+1})$ is described by

$$u^*(t) = u_j \text{ for } t \in [s_k + t_j^k, s_k + t_{j+1}^k), \quad (5)$$

where t_j^k is defined as:

$$\begin{aligned} t_j^k = \min[&\max\{0, \tau_j - (k-j)h\} + m_j h, \\ &\max\{0, \tau_{j+1} - (k-j-1)h\} + m_{j+1}h, \ldots, \\ &\max\{0, \tau_{k-\underline{d}} - \underline{d}h\} + m_{k-\underline{d}}h, h], \end{aligned} \quad (6)$$

with $t_j^k \leq t_{j+1}^k$ and $j \in \{k - \overline{d} - \overline{\delta}, k - \overline{d} - \overline{\delta} + 1, \ldots, k - \underline{d}\}$. Moreover, $0 = t_{k-\overline{d}-\overline{\delta}}^k \leq t_{k-\overline{d}-\overline{\delta}+1}^k \leq \ldots \leq t_{k-\underline{d}}^k \leq t_{k-\underline{d}+1}^k := h$.

Proof: The proof is given in the appendix. ∎

For $\sigma \in \mathcal{S}$, we can exploit Lemma 1 to define the discrete-time NCS model as follows:

$$x_{k+1} = e^{Ah}x_k + \sum_{j=k-\overline{d}-\overline{\delta}}^{k-\underline{d}} \int_{h-t_{j+1}^k}^{h-t_j^k} e^{As}ds Bu_j, \quad (7)$$

with t_j^k defined in (6). Namely, according to Lemma 1, this model contains all possible control inputs that can be active during the sampling interval $[s_k, s_{k+1})$. Note that $t_j^k = t_{j+1}^k$ corresponds to the situation that the integral related to u_j in (7) is zero, which corresponds to an inactive control input u_j during the sampling interval $[s_k, s_{k+1})$, which allows for message rejection and packet dropouts.

To make the model of (7) suitable for the stability analysis and controller synthesis, we rewrite it in a state-space notation:

$$\xi_{k+1} = M\xi_k + Nu_k, \quad (8)$$

with $\xi_k = \begin{pmatrix} x_k^T & u_{k-1}^T & u_{k-2}^T & \cdots & u_{k-\overline{d}-\overline{\delta}}^T \end{pmatrix}^T$,

$$M = \begin{pmatrix} e^{Ah} & \tilde{B}_1 & \tilde{B}_2 & \ldots & \tilde{B}_{\overline{d}+\overline{\delta}} \\ 0 & 0 & 0 & \ldots & 0 \\ 0 & I & 0 & \ldots & 0 \\ \vdots & & \ddots & & \ldots \\ 0 & \ldots & 0 & I & 0 \end{pmatrix}, N = \begin{pmatrix} \tilde{B}_0 \\ I \\ 0 \\ \vdots \\ 0 \end{pmatrix} \text{ and}$$

$$\tilde{B}_\rho = \begin{cases} \int_{h-t_{k-\rho+1}^k}^{h-t_{k-\rho}^k} e^{As}ds B, & \text{if } \rho \geq \underline{d} \\ 0 & \text{if } \rho < \underline{d}, \end{cases}$$

$\rho \in \{0, 1, ..., \overline{d}+\overline{\delta}\}$ and $t_{k-\rho}^k$ defined in (6), with $j = k - \rho$.

4992

III. STABILIZATION OF NCSs

Similar as in [18], we use the (real) Jordan form of the system matrix $A := QJQ^{-1}$ [21], with A defined in (2), for the stability analysis and controller synthesis. The general notation of the NCS model is then given by:

$$\xi_{k+1} = \left(F_0 + \sum_{i=1}^{\beta} \alpha_i(\cdot)F_i\right)\xi_k + \left(G_0 + \sum_{i=1}^{\beta} \alpha_i(\cdot)G_i\right)u_k, \tag{9}$$

with $F_i, G_i, i = 1, 2, \ldots, \beta$, constant matrices, $\alpha_i(\cdot), i = 1, 2, \ldots, \beta$, a time-varying parameter that depends on one value of t_j^k, defined in (6), and the parameter β representing the total number of uncertain parameters. To illustrate this, we give an example where A has two real eigenvalues:

Example Consider $A = QJQ^{-1}$, with $J = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix}$ and λ_1 a real non-zero eigenvalue. It holds that $e^{Ah} = Qe^{Jh}Q^{-1} = Q\left(e^{\lambda_1 h}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{h}{1!}e^{\lambda_1 h}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\right)Q^{-1}$. Solving the integrals in (7) gives: $\int_{h-t_{j+1}^k}^{h-t_j^k} e^{As}ds =$

$$Q\left(\frac{e^{\lambda_1(h-t_j^k)}}{\lambda_1} - \frac{e^{\lambda_1(h-t_{j+1}^k)}}{\lambda_1}\right)S_1 Q^{-1} +$$

$$Q\left(\frac{h-t_j^k}{\lambda_1}e^{\lambda_1(h-t_j^k)} - \frac{h-t_{j+1}^k}{\lambda_1}e^{\lambda_1(h-t_{j+1}^k)}\right)S_2 Q^{-1},$$

with $S_1 = \begin{pmatrix} 1 & -\frac{1}{\lambda_1} \\ 0 & 1 \end{pmatrix}$ and $S_2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$. If $\tau_{\min} = 0, \tau_{\max} = h$ and $\overline{\delta} = 0$, the time-varying parameters in (9) are given by: $\alpha_1(\tau_k) = \frac{1}{\lambda_1}e^{\lambda_1(h-\tau_k)}$, $\alpha_2(\tau_k) = \frac{h-\tau_k}{\lambda_1}e^{\lambda_1(h-\tau_k)}$ and the constant matrices in (9) are given by: $F_0 = \begin{pmatrix} Qe^{Jh}Q^{-1} & Q(S_1 + hS_2)\frac{e^{\lambda_1 h}}{\lambda_1}Q^{-1}B \\ 0 & 0 \end{pmatrix}$, $G_0 = \begin{pmatrix} -QS_1\frac{1}{\lambda_1}Q^{-1}B \\ I \end{pmatrix}$, $F_{\hat{i}} = \begin{pmatrix} 0 & -QS_{\hat{i}}Q^{-1}B \\ 0 & 0 \end{pmatrix}$, $G_{\hat{i}} = \begin{pmatrix} QS_{\hat{i}}Q^{-1}B \\ 0 \end{pmatrix}$ and $\hat{i} = 1, 2$. For other combinations of $\tau_{\min}, \tau_{\max}$ and $\overline{\delta}$ the matrices and parameters can be derived in a similar manner. For more details, see [19]. ∎

In general, the parameter β in (9) is defined as: $\beta = (\overline{d} + \overline{\delta} - \underline{d})\nu$, with ν the number of time-varying parameters for one value of t_j^k, obtained from the continuous-time system matrix A. To define ν, we use $A = QJQ^{-1}$, with $J = diag(J_1, J_2, \ldots, J_p)$ the Jordan form of A that consists of p different Jordan blocks $J_{\tilde{i}}, \tilde{i} = 1, 2, \ldots, p$, where $p \in \mathbb{N}$ denotes the number of distinct eigenvalues. The Jordan block $J_{\tilde{i}}$ consists of different Jordan blocks if $g_{\tilde{i}} > 1$, with $g_{\tilde{i}} \in \mathbb{N}$ the geometric multiplicity of the $\tilde{i}^{th}$ eigenvalue: $J_{\tilde{i}} = diag(J_{\tilde{i},1}, J_{\tilde{i},2}, \ldots, J_{\tilde{i},g_{\tilde{i}}}), \tilde{i} = 1, 2, \ldots, p$. Then, it holds that $\nu = \sum_{\tilde{i}=1}^{p} \max_{\tilde{j} \in \{1,2,\ldots,g_{\tilde{i}}\}}(\dim J_{\tilde{i},\tilde{j}})$. In the example it holds that $p = 1, g_1 = 1, \nu = 2$ and $\beta = 2$.

For the stability analysis and controller synthesis, we consider two types of controllers. First, we consider an extended state-feedback controller that depends on the state x_k and the previous control inputs $u_{k-1}, u_{k-2}, \ldots, u_{k-\overline{d}-\overline{\delta}}$:

$$u_k = -K\xi_k \tag{10}$$

with $K \in \mathbb{R}^{m \times ((\overline{d}+\overline{\delta})m+n)}$. A similar structure was proposed in [10] for time-delays smaller than the sampling interval

and without packet dropouts. Second, we consider a state-feedback controller that depends only on the state x_k:

$$u_k = -\overline{K}x_k \tag{11}$$

with $\overline{K} \in \mathbb{R}^{m \times n}$. Note that this control law is a special case of (10) with $K = \begin{pmatrix} \overline{K} & 0_{m,(\overline{d}+\overline{\delta})m} \end{pmatrix}$.

In the case of packet dropout related to u_k, i.e. $m_k = 1$, two situations are possible:

1) $y_k = x_k$ does not arrive at the controller and thus u_k can not be computed,
2) $y_k = x_k$ arrives at the controller, thus u_k is computed, but does not arrive at the actuator.

In the second case, in principle, ξ_{k+1} can be computed, because u_k is available in the implementation of the controller. In the first case, the controller (10) can not be updated anymore, due to an unknown u_k and a deadlock in the controller will occur for u_{k+1}. Of course, one might modify the controller using a heuristic manner (e.g. set $u_k := u_{k-1}$ if $m_k = 1$), but this results in a controller that depends explicitly on m_k and would require a separate analysis for this switching controller. For the sake of brevity, this additional analysis will not be considered. Therefore, the observation of a possible deadlock means that controller (10), as proposed in [10] for delays smaller than the sampling interval, does not function properly in the case of packet dropouts between the sensor and the controller (first case). The second controller (11) does not suffer from this problem as it only depends on the state x_k (and not on the previous control inputs). A similar reasoning holds for message rejection between the sensor and the controller. Therefore, to avoid additional complexity, in the case of controller (10), we adopt the following assumption:

Assumption 2: There is no packet dropout between the sensor and the controller and y_k always arrives at the controller after the moment that u_{k-1} is sent to the actuator, i.e. $kh + \tau_k^{sc} > (k-1)h + \tau_{k-1}^{sc} + \tau_{k-1}^c$.

Note that this assumptions does not need to be considered for the state-feedback controller.

A. Stability Analysis

From (9), we can obtain a set of matrices that describe all possible matrix combinations (F_i, G_i), resulting from the sequences $\sigma \in \mathcal{S}$.

$$\mathcal{FG} = \left\{\left(F_0 + \sum_{i=1}^{\beta} \alpha_i(t_j^k)F_i, G_0 + \sum_{i=1}^{\beta} \alpha_i(t_j^k)G_i\right) : t_j^k \in [t_{j,\min}^k, t_{j,\max}^k]\right\},$$

with $k - \overline{d} - \overline{\delta} < j \leq k - \underline{d}$ and

$$t_{j,min}^k = \begin{cases} \tau_{\min} - \underline{d}h & \text{if } j = k - \underline{d} \\ 0 & \text{if } j < k - \underline{d}, \end{cases}$$

$$t_{j,max}^k = \begin{cases} h & \text{if } j > k - \overline{d} - \overline{\delta} + 1 \\ \tau_{\max} - (\overline{d}-1)h & \text{if } j = k - \overline{d} - \overline{\delta} + 1. \end{cases} \tag{12}$$

To study the stability of the fixed point $\xi = 0$ of system (8), with a given controller (10) or (11), for $\sigma \in \mathcal{S}$ we will use a common quadratic Lyapunov function: $V(\xi_k) = \xi_k^T P \xi_k$. Note that, due to the time-varying behavior of τ_k, in each sampling interval the values of t_j^k will be different. Therefore, the matrices M and N will have different values within the

4993

different sampling intervals. This observation shows that (8), (10) or (8), (11) represent a switching discrete-time system. Then, the existence of a common quadratic Lyapunov function is indeed sufficient for the stability of (8), (10) or of (8), (11) for which the same reasoning holds. If there exists a matrix $P \in \mathbb{R}^{(n+(\overline{d}+\overline{\delta})m)\times(n+(\overline{d}+\overline{\delta})m)}$ satisfying:

$$
\begin{aligned}
P = P^T &> 0 \\
C(\sigma)^T P C(\sigma) - P &< -\gamma P, \forall \sigma \in \mathcal{S},
\end{aligned}
\tag{13}
$$

with $C(\sigma) = F_0 - G_0 K + \sum_{i=1}^{\beta} \alpha_i(t_j^k)(F_i - G_i K)$ and $0 \le \gamma < 1$ then stability is guaranteed. Due to the definition of $\mathcal{S}$, (13) contains an infinite number of LMIs. Based on an overapproximation of α_i, we will derive a finite number of LMIs that guarantee stability.

Theorem 3: Consider the NCS of (8), with (10), including Assumption 2, or (11) with $K = \begin{pmatrix} \overline{K} & 0_{m,(\overline{d}+\overline{\delta})m} \end{pmatrix}$ for $\sigma \in \mathcal{S}$. Define the set of matrices $\mathcal{H}_{FG}$:

$$
\mathcal{H}_{FG} = \left\{ \left(\bar{F}_0 + \sum_{i=1}^{\beta} \delta_i \bar{F}_i, \bar{G}_0 + \sum_{i=1}^{\beta} \delta_i \bar{G}_i \right) : \right.
$$
$$
\left. \delta_i \in \{0,1\}, i = 1,2,...,\beta \right\},
\tag{14}
$$

with $\bar{F}_0 = F_0 + \sum_{i=1}^{\beta} \underline{\alpha}_i F_i$, $\bar{F}_i = (\overline{\alpha}_i - \underline{\alpha}_i)F_i$, $\bar{G}_0 = G_0 + \sum_{i=1}^{\beta} \underline{\alpha}_i G_i$, $\bar{G}_i = (\overline{\alpha}_i - \underline{\alpha}_i)G_i$, and $\overline{\alpha}_i = \max_{t_j^k \in \Gamma} \alpha_i(t_j^k)$, $\underline{\alpha}_i = \min_{t_j^k \in \Gamma} \alpha_i(t_j^k)$, the maximum and minimum value of $\alpha_i(t_j^k)$, respectively, with $\Gamma = [t_{j,\min}^k, t_{j,\max}^k]$ and $t_{j,\min}^k$, $t_{j,\max}^k$ defined in (12).

If there exist a matrix $P \in \mathbb{R}^{(n+(\overline{d}+\overline{\delta})m)\times(n+(\overline{d}+\overline{\delta})m)}$ and a scalar $0 \le \gamma < 1$, such that the following LMI conditions are satisfied:

$$
\begin{pmatrix} (1-\gamma)P & (H_{F,s} - H_{G,s}K)^T P \\ P(H_{F,s} - H_{G,s}K) & P \end{pmatrix} > 0,
\tag{15}
$$

for all $(H_{F,s}, H_{G,s}) \in \mathcal{H}_{FG}$, $s = 1,2,\ldots,2^{\beta}$, then (2), (5), (6), (10) or (11) is globally asymptotically stable (GAS) for any sequence of time-varying delays and packet dropouts $\sigma \in \mathcal{S}$. Note that in the case of controller (10), Assumption 2 should hold.

Proof: The proof is analogous to the proof of Theorem 1 in [18]. It consists of a convex overapproximation of the set $\mathcal{FG}$, based on generators and an investigation of the intersample behavior that shows that if the discrete-time system satisfies (15), then the continuous-time system is GAS as well. ∎

B. Controller Synthesis

First, we consider the controller synthesis problem for the NCS described by (8), (10). Based on a similar overapproximation as exploited in Theorem 3, sufficient conditions for the GAS of system (2), (10) are given in Theorem 4.

Theorem 4: Consider the NCS model (2) and its discrete-time representation (9) for sequences of delays and packet dropouts $\sigma \in \mathcal{S}$, with $\mathcal{S}$ as in (4). Consider the set of matrices $\mathcal{H}_{FG}$ defined in (14).

If there exist matrices $Y \in \mathbb{R}^{(n+(\overline{d}+\overline{\delta})m)\times(n+(\overline{d}+\overline{\delta})m)}$, $\mathbb{Z}^{m\times(n+(\overline{d}+\overline{\delta})m)}$ and a scalar $0 \le \gamma < 1$ such that the following LMIs are satisfied:

$$
\begin{pmatrix} (1-\gamma)Y & YH_{F,s}^T - Z^T H_{G,s}^T \\ H_{F,s}Y - H_{G,s}Z & Y \end{pmatrix} > 0,
\tag{16}
$$

for all $(H_{F,s}, H_{G,s}) \in \mathcal{H}_{FG}$, $s = 1,2,\ldots,2^{\beta}$, then $x = 0$ is a GAS equilibrium point of the the closed-loop NCS (2), (5), (6), (10) with $K = ZY^{-1}$.

Proof: Pre- and postmultiplying (16) with $diag(Y^{-1}, Y^{-1})$ gives (15), after the linearizing change of variables $Y^{-1} = P$ and $ZY^{-1} = K$. ∎

Remark Due to the overapproximation of α_i, $i = 1,2,\ldots,\beta$, and the minimum and maximum values of t_j^k defined in (12), the values of $\tau_{\max}$ and $\overline{\delta}$ are interchangeable as long as their summation remains constant. E.g. the cases $\tau_{\max} = h, \overline{\delta} = 0$ and $\tau_{\max} = 0, \overline{\delta} = 1$ give the same range of stabilizing controllers for Theorem 3 (or the same controller for Theorem 4), independent on the chosen value of $\tau_{\min}$.

Remark For the state-feedback controller (11), Theorem 4 holds as well, if $Y = \begin{pmatrix} Y_1 & 0 \\ 0 & Y_2 \end{pmatrix}$ is used, with a symmetric matrix $Y_1 \in \mathbb{R}^{n\times n}$ and a diagonal matrix $Y_2 \in \mathbb{R}^{(\overline{d}+\overline{\delta})m\times(\overline{d}+\overline{\delta})m}$ and $Z = \begin{pmatrix} Z & 0_{m,(\overline{d}+\overline{\delta})m} \end{pmatrix}$.

To obtain a performance measure in terms of the transient behavior of the closed-loop system, we determine the convergence bound based on the obtained $P = Y^{-1}$ and corresponding value γ in Theorem 4. Recall that, if the conditions of Theorem 3 or 4 are satisfied, then (13) holds; i.e. it holds that $\Delta V < -\gamma V$. We adopt the notation $|\xi_k|_P^2 = |\xi_k^T P \xi_k|$. Using the fact that $\lambda_{\min}(P)|\xi_k|^2 \le |\xi_k|_P^2 \le \lambda_{\max}(P)|\xi_k|^2$ and $|x_k|^2 \le \|C_x P^{-\frac{1}{2}}\|^2|\xi_k|_P^2$, with C_x obtained from $x_k = C_x \xi_k$, we can derive a lower bound for the transient decay rate of the discrete-time state x_k as:

$$
|x_k|^2 \le (1-\gamma)^k \|C_x P^{-\frac{1}{2}}\|^2 |\xi_0|_P^2.
\tag{17}
$$

It is obvious that the lower bound on the decay rate depends both on P and γ. An optimization algorithm that derives a control gain K according to (16), with the fastest decay rate as defined in (17), results in a control design tool that combines stability and transient performance (settling-time) in the face of time-varying delays and packet dropouts $\sigma \in \mathcal{S}$.

IV. ILLUSTRATIVE EXAMPLE

In this section, we apply the proposed results to a second-order motion control example, obtained from the document printing domain. We limit ourselves to one single motor driving one roller pair that is used to transport a sheet through part of the printer paperpath. The continuous-time motor-roller model is given by (2), with $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ \frac{nr_R}{J_M+n^2 J_R} \end{pmatrix}$ and $x = \begin{pmatrix} x_s \\ \dot{x}_s \end{pmatrix}$ the state vector, which contains the sheet position and velocity, $J_M = 1.95 \cdot 10^{-5} \text{kgm}^2$ the inertia of the motor, $J_R = 6.5 \cdot 10^{-5} \text{kgm}^2$ the inertia of the roller, $r_R = 14 \cdot 10^{-3}$ m the radius of the roller, $n = 0.2$ the transmission ratio between motor and roller and u the motor torque.

a) Stability Analysis: Here, we limit ourselves to an example where a different number of subsequent packet dropouts can occur, in combination with a time-varying delay that is upper bounded by the sampling interval ($h = 1$ ms, $\tau_{\min} = 0, \tau_{\max} \le h$). The controller has the specific form
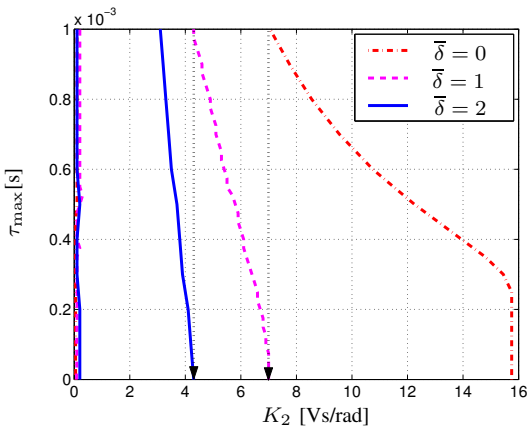
4994

Fig. 2. Upper and lower bounds of the stability region for $\overline{K} = [50\ K_2]$ (11) for different values of $\overline{\delta}$ and $\tau \in [0, \tau_{\max}]$.
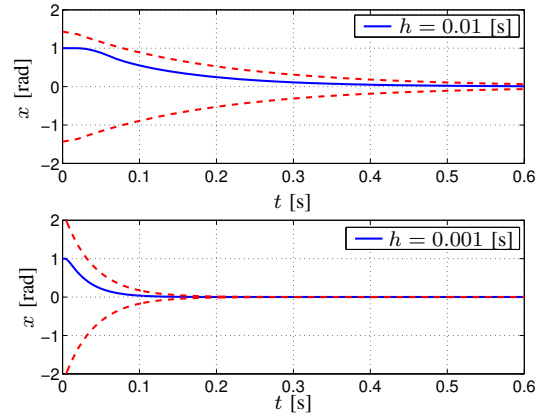

Fig. 3. Time-response (solid line) and lower bound for transient decay rate (dashed line) obtained from (16) and (17), respectively, for $h = 0.01$ s, $\gamma = 0.1$ and $h = 0.001$ s, $\gamma = 0.15$.

of (11), with $\overline{K} = \begin{pmatrix} 50 & K_2 \end{pmatrix}$. We are interested in the stability region, which gives all values of K_2 that guarantee stability for each combination of delay $\tau_k \in [0, \tau_{\max}]$ and packet dropout $\overline{\delta}$. Based on the LMIs of Theorem 3, we obtain the stability region, given by an upper and lower bound for K_2, for $\overline{\delta} = 0$, i.e. no packet dropouts, $\overline{\delta} = 1$ and $\overline{\delta} = 2$ in Figure 2. Clearly, packet dropouts decrease the allowable controller gains that stabilize the NCS. Note that the allowable controller gain K_2 for a time-varying delay upper bounded by the sampling interval without packet dropouts is the same as the allowable controller gain K_2 for one subsequent packet dropout, without delays. The same holds for one packet dropout in combination with a time-varying delay upper bounded by the sampling interval and two packet dropouts without delays. This similarity is evident to the analysis as presented in Theorem 3 and 4.

b) Controller Synthesis and Convergence: To determine the applicability of Theorem 4 for controller synthesis and the estimation of the transient decay rate in (17), we obtain controllers that guarantee stability for $\tau_{\min} = 0$, $\tau_{\max} = 2h$ and $\overline{\delta} = 1$, for the constant sampling intervals $h = 0.01$s and $h = 0.001$s and for a given value of γ. For the sampling interval $h = 0.01$ s and $\gamma = 0.1$, we obtain the controller $K = \begin{pmatrix} 2.8248 & 0.4115 & 0.1647 & 0.2467 & 0.0216 \end{pmatrix}$. The corresponding time-response of the position signal is depicted in the upper plot of Figure 3. To obtain this response a random time-varying delay is chosen and the moment of packet dropout is determined in a random fashion, but it is guaranteed that at maximum $\overline{\delta}$ subsequent packet dropouts occur. For $h = 0.001$ s and $\gamma = 0.15$, we obtain the controller $K = \begin{pmatrix} 42.2313 & 1.4462 & 0.2091 & 0.2243 & 0.1417 \end{pmatrix}$. The corresponding time-response is depicted in the lower plot of Figure 3. The estimated lower bounds for the decay rate of both controllers, based on (17), are given by the dashed lines in the same figure. This shows that the estimated lower bound of the decay rate is not an overly conservative prediction of the decay rate of the time-response itself. We remark that comparable results can be obtained for other values of the sampling interval and the convergence parameter γ. Therefore, relation (17) is useful to predict

the decay rate of the system. Moreover, solving Theorem 4 for a maximum γ is useful to design a controller that, firstly, guarantees stability for time-varying delays and packet dropouts and, secondly, has an optimal time-response in terms of the transient decay rate. Note that, in general, the maximum value of γ will result in infinite controller gains, therefore an extension to optimal control (see e.g. [6]), where the control input is weighted is advisable.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a discrete-time model for NCSs with time-varying delays, that can be both smaller and larger than the sampling interval, and packet dropouts, for which an upper bound on the number of subsequent packet dropouts that can occur is assumed. Based on this model, stability and controller synthesis conditions (in the form of linear matrix inequalities) that guarantee global asymptotic stability of the NCSs are derived. These stability conditions are not overly conservative for the presented example. For the stability analysis, two controllers are investigated, i.e. a state-feedback controller and an extended state feedback controller that depends on both the state vector and past control inputs. The extended state-feedback controller is only valid if no packet dropouts and message rejection between the sensor and controller occur, because of its dependence on past control inputs. For the state-feedback controller this restriction is not needed. For both controllers, controller synthesis conditions that guarantee stability for bounded time-varying delays and bounded packet dropouts are derived based on a common quadratic Lyapunov approach. Moreover, these conditions also guarantee a lower bound on the transient decay rate of the time-response of the system. This lower bound gives a useful estimation of the decay rate of the system, which makes it a useful design tool to derive controllers that guarantee stability and have to satisfy demands with respect to the transient response.

Future work deals with the consequences when Assumption 2 is not valid for the extended state-feedback controller.

APPENDIX

To prove that $u_{k-\overline{d}-\overline{\delta}}$ is the oldest input that can be active during the sampling interval $[s_k, s_{k+1})$ we consider,

4995

firstly, the case without packet dropouts and, secondly, the case with packet dropouts. From the definition of $\overline{d}$ in Lemma 1, we have that the control input $u_{k-\overline{d}}$ is always available before or exactly at $t = s_k := kh$, provided $u_{k-\overline{d}}$ is not dropped, as $s_{k-\overline{d}} + \tau_{k-\overline{d}} \leq s_{k-\overline{d}} + \tau_{\max} \leq s_k$. Hence, in the case that $u_{k-\overline{d}}$ is not dropped, no control inputs u_j with $j < k - \overline{d}$ will be active in $[s_k, s_{k+1})$. To prove that newer inputs are not necessarily available before s_k, we determine the latest time at which $u_{k-\overline{d}+1}$ might be implemented, which is equal to $s_{k-\overline{d}+1} + \tau_{\max}$. Based on the definition of $\overline{d}$ it holds that $\overline{d}h - \tau_{\max} \in [0, h)$, i.e. $\tau_{\max} > (\overline{d} - 1)h$. Using this fact and that fact that $s_{k-\overline{d}+1} = s_k - (\overline{d} - 1)h$ gives: $s_{k-\overline{d}+1} + \tau_{\max} > s_k$. This proves that $u_{k-\overline{d}+1}$ might be implemented after s_k, implying that an older input $u_{k-\overline{d}}$ might indeed be active in $[s_k, s_{k+1})$. Next, we consider the case with packet dropouts. Note that, from (3), it follows that at least one of the control inputs $u_{k-\overline{d}-\overline{\delta}}, u_{k-\overline{d}-\overline{\delta}+1}, \ldots, u_{k-\overline{d}}$ is not lost. If $u_{k-\overline{d}+1}$ is indeed implemented after s_k (which is possible as just shown), then at least one of the inputs $u_{k-\overline{d}-\overline{\delta}}, u_{k-\overline{d}-\overline{\delta}+1}, \ldots, u_{k-\overline{d}}$ will be active in the sampling interval $[s_k, s_{k+1})$. The fact that the maximum number of subsequent packet dropouts equals $\overline{\delta}$ implies that $u_{k-\overline{d}-\overline{\delta}}$ is the oldest control input that might be implemented in the sampling interval $[s_k, s_{k+1})$.

From the definition of $\underline{d}$ in the lemma, it follows that the input $u_{k-\underline{d}}$ represents the most recent control input that might be implemented during the sampling interval $[s_k, s_{k+1})$. Indeed, as $s_{k-\underline{d}} + \tau_{\min} < s_{k+1}$ the input $u_{k-\underline{d}}$ might be available for implementation before time s_{k+1}. To show that there is no more recent control input that might be active in the interval $[s_k, s_{k+1})$, consider the control input u_j, for some $j > k - \underline{d}$. From the definition of $\underline{d}$, we have that: $s_j + \tau_j \geq s_j + \tau_{\min} \geq s_{k+1} \ \forall j > k - \underline{d}$. Therefore, the control input u_j, $j > k - \underline{d}$, can not be implemented in the sampling interval $[s_k, s_{k+1})$. Hence, the control inputs $u_{k-\overline{d}-\overline{\delta}}, \ldots, u_{k-\underline{d}}$ are the only control inputs that can be active in the sampling interval $[s_k, s_{k+1})$.

The times t_j^k with $j \in [k - \overline{d} - \overline{\delta}, \ldots, k - \underline{d}]$ will be constructed in such a manner that $s_k + t_j^k$ is the time at which the control input u_j becomes active in the sampling interval $[s_k, s_{k+1})$. Hence, $t_{k-\underline{d}}^k$ is given by:

$$t_{k-\underline{d}}^k = \min[h, \tau_{k-\underline{d}} - \underline{d}h + m_{k-\underline{d}}h]. \tag{18}$$

Indeed, if $m_{k-\underline{d}} = 0$, then $s_k + \tau_{k-\underline{d}} - \underline{d}h$ is the time at which $u_{k-\underline{d}}$ is available at the plant. If $\tau_{k-\underline{d}} - \underline{d}h > h$, then $u_{k-\underline{d}}$ might be active after s_{k+1}, but not in $[s_k, s_{k+1})$. Since we are only interested in the interval $[s_k, s_{k+1})$, we take the minimum of this value and h. Note that, by definition, $\tau_{k-\underline{d}} - \underline{d}h \geq 0$. Finally, if $u_{k-\underline{d}}$ is lost, i.e. $m_{k-\underline{d}} = 1$, then the expression for $t_{k-\underline{d}}^k$ in (18) becomes h, which means that the input is not used in $[s_k, s_{k+1})$. Next, as $u_{k-\underline{d}-1}$ can only be active before $u_{k-\underline{d}}$ is available, $t_{k-\underline{d}-1}^k$ is given by: $t_{k-\underline{d}-1}^k = \min[t_{k-\underline{d}}^k, \max\{0, \tau_{k-\underline{d}-1} - (\underline{d} + 1)h\} + m_{k-\underline{d}-1}h]$. Similarly to $t_{k-\underline{d}}^k$, if $\max\{0, \tau_{k-\underline{d}-1} - (\underline{d}+1)h\} + m_{k-\underline{d}-1}h \in [0, t_{k-\underline{d}}^k)$ then $s_k + \tau_{k-\underline{d}-1} - (\underline{d}+1)h$ is the time at which $u_{k-\underline{d}-1}$ is available at the actuator.

If $\tau_{k-\underline{d}-1} - (\underline{d} + 1)h < 0$, then $u_{k-\underline{d}-1}$ might already be active before s_k. Since, we are only interested, here, in the interval $[s_k, s_{k+1})$, we take the maximum of this value and 0. For the other values of t_j^k, the recursion can be derived similarly as: $t_j^k = \min[t_{j+1}^k, \max\{0, \tau_j - (k - j)h\} + m_j h]$, for $k - \overline{d} - \overline{\delta} \leq j \leq k - \underline{d}$, m_j satisfying (3) and with $t_{k-\underline{d}+1}^k := h$. The elaboration of this recursive relation yields the characterization of (6).

REFERENCES

[1] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.

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

[3] T. C. Yang, "Networked control system: a brief survey," *IEE Proc.-Control Theory Appl.*, vol. 153, no. 4, pp. 403–412, 2006.

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

[5] J. R. Moyne and D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proc. of the IEEE*, pp. 29–47, 2007.

[6] J. Nilsson, *Real-Time Control Systems with Delays*. Lund, Sweden: PhD thesis, Dep. of Automatic Control, Lund Inst. of Techn., 1998.

[7] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 66–83, 2001.

[8] H. Ishii, "Stabilization under shared comminication with message losses and its limitations," in *Proc. of the 45th IEEE Conf. on Decision and Control*, San Diego, CA, USA, 2006, pp. 4974–4979.

[9] S. Hu and W.-Y. Yan, "Stability robustness of networked control systems with respect to packet loss," *Automatica*, no. 7, pp. 1243–1248, 2007.

[10] L. Hetel, J. Daafouz, and C. Iung, "Stabilization of arbitrary switched linear systems with unknown time-varying delays," *IEEE Transactions on Automatic Control*, vol. 51, no. 10, pp. 1668 – 1674, 2006.

[11] Y.-J. Pan, H. J. Marquez, and T. Chen, "Stabilization of remote control systems with unknown time varying delays by LMI techniques," *Int. Journal of Control*, vol. 79, no. 7, pp. 752–763, 2006.

[12] J. Xiong and J. Lam, "Stabilization of linear systems over networks with bounded packet loss," *Automatica*, vol. 43, pp. 80–87, 2007.

[13] M. Yu, L. Wang, T. Chu, and G. Xie, "Stabilization of networked control systems with data packet dropout and network delays via switching system approach," in *Proc. of the 43rd Conf. on Decision and Control*, Atlantis, Bahamas, 2004, pp. 3539–3544.

[14] ——, "An LMI approach to networked control systems with data packet dropout and transmission delays," in *Proc. of the 43rd Conf. on Decision and Control*, Atlantis, Bahamas, 2004, pp. 3545–3550.

[15] D. Yue, Q.-L. Han, and C. Peng, "State feedback controller design of networked control systems," *IEEE Trans. on circuits and systems*, vol. 51, no. 11, pp. 640–644, 2004.

[16] P. Naghshtabrizi and J. P. Hespanha, "Stability of network control systems with variable sampling and delays," in *Proc. 44th Annual Allerton Conf. on Communication, Control, and Computing*, 2006.

[17] H. Lin and P. J. Antsaklis, "Stability and persistent disturbance attenuation properties for a class of networked control systems: switched system approach," *Int. J. of Control*, vol. 78, no. 18, pp. 1447–1458, 2005.

[18] M. B. G. Cloosterman, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, "Stability of networked control systems with large delays," in *Proc. of the 46th IEEE Conf. on Decision and Control*, New Orleans, USA, 2007, pp. 5017–5022.

[19] ——, "Robust control of networked control systems with uncertain time-varying delays," *DCT internal report 2006-121*, 2006, Eindhoven University of Technology, the Netherlands.

[20] K. J. Åström and B. Wittenmark, *Computer controlled systems, theory and design*. Englewood Cliffs: Prentice-Hall International, Inc, 1990.

[21] C. Moler and C. van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM review*, vol. 20, pp. 801–836, 1978.