

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Dacic, DB;Nesic, D

Title:

Simultaneous controller and protocol design for networked control systems with packet based communication

Date:

2006-01-01

Citation:

Dacic, D. B. & Nesic, D. (2006). Simultaneous controller and protocol design for networked control systems with packet based communication. PROCEEDINGS OF THE 45TH IEEE CONFERENCE ON DECISION AND CONTROL, VOLS 1-14, pp.508-513. IEEE.
<https://doi.org/10.1109/CDC.2006.377027>.

Persistent Link:

<https://hdl.handle.net/11343/299656>

Simultaneous Controller and Protocol Design for Networked Control Systems with Packet Based Communication

Dragan B. Dačić and Dragan Nešić

Abstract— We develop the necessary and sufficient conditions for quadratic stabilizability of linear networked control systems by communication protocols and output feedback controllers. These conditions are used to develop a computationally tractable design for simultaneous synthesis of controllers and protocols in terms of matrix inequalities. The obtained protocols do not require knowledge of controller's and plant's states but only of the discrepancies between current and the most recent signal values transmitted over the network, and are implementable on control area networks. We apply our design algorithm to a batch reactor and show that the obtained protocols and controllers ensure quadratic stability with a significantly smaller network bandwidth than required with previously available designs.

I. INTRODUCTION

In networked control systems (NCSs), a controller and spatially distributed sensors/actuators are grouped into network nodes and exchange information via a network. Data transmitted over the network is divided into packets consisting of a header used by the network to route the packet, and a data field for the destination node. NCSs have several advantages over classical control systems, such as reduced installation costs and improved flexibility, but network's presence in a feedback loop invalidates the classical framework for analysis and design of control systems. First, depending on the levels of network congestion and noise packets sent over a network may be lost and are subject to a variable delay [1]. Second, the network introduces quantization errors in signals transmitted over it due to its finite bandwidth. Significant research effort is directed at determining the minimal bandwidth required to stabilize an unstable plant [2]. Finally, shared communication resources among network nodes impose that only one node at a time can transmit its packet. Thus, NCSs are subject to communication constraints which raise the need for *communication scheduling*, of central interest in this paper.

There are two distinctive protocol classes for communication scheduling, commonly named static and dynamic protocols. In a static, that is, a round robin (RR) protocol the right to transmit is periodically granted to nodes in some predetermined order. Dynamic protocols grant network transmissions based on the system states. One such protocol is *maximum-error-first try-once-discard* (TOD) protocol [3]–[6], in which the next transmission is granted to the node with the largest discrepancy between the current and the most recent transmitted value of node's signals. We refer to these discrepancies as the *network induced errors*.

The effect of communication constraints imposed by the network on stability of NCSs is first analyzed by Walsh et

al. [3]–[6]. They assume that a stabilizing controller for the plant in the absence of the network is available, and treat the network as a perturbation of the stable closed-loop system whose magnitude is proportional to a transmission interval (TI) within which it successfully transmits packets. They show that stability of the NCS is preserved for a sufficiently small TI. The longest such TI which ensures stability of the NCS is called the maximum available transmission interval (MATI), and it is inversely proportional to the network bandwidth. This approach is equally applicable to linear [3] and nonlinear [4] NCSs, and to different protocols. It enables independent synthesis of the controller and the protocol thus reducing the design complexity. Its major drawback is that the obtained MATI estimates which guarantee stability are typically small.

A significant improvement of MATI estimates is achieved in [7]–[9]. Nešić et al. introduce and characterize a class of GAS protocols [7], [8] and show that both RR and TOD protocols belong to this class. They develop a novel model of NCSs using which they prove that if the closed loop system without the network is $\mathcal{L}_p$ stable, then $\mathcal{L}_p$ stability is preserved with the network for sufficiently small MATIs. Tabbara et al. [9] introduce persistently exciting (PE) protocols, and under milder assumptions show that a similar $\mathcal{L}_p$ stability property holds for PE protocols in wireless NCSs.

Given a linear time-invariant (LTI) plant and a *fixed* TI in this paper we develop several necessary and/or sufficient conditions for existence of an LTI output feedback controller and protocol which ensure quadratic stability (QS) of the resulting NCS. Using our sufficient conditions we provide a computationally tractable algorithm for *simultaneous* synthesis of controllers and protocols. The resulting protocols depend on the network induced errors, but can be further constrained to depend only on locally available network induced errors, thus simplifying their implementation. These information constraints are the key difference between our and previously available procedures for synthesis of dynamic protocols [10]–[12]. To address these constraints we introduce *weak partial state control Lyapunov functions* (WPSCFLs) which can be utilized in other control problems with information constraints.

In Section II we derive the model of linear NCSs (LNCSS). The notion of WPSCFLs is introduced in Section III and utilized in Section IV to derive the necessary and sufficient conditions for quadratic stabilizability of LNCSSs. Based on these conditions in Section V we develop a design algorithm for rendering LNCSSs quadratically stable. The implementation issues of the synthesized protocols are considered in Section VI. We illustrate our design procedure in Section VII and compare the obtained MATIs with MATIs obtained via procedures in [3], [7]. The proofs for most of the claims are omitted and can be found in [13].

This work is supported by the Australian Research Council under the Discovery Project and Australian Professorial Fellow program.

Authors are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010 Australia, d.dacic, d.nesic@ee.unimelb.edu.au

II. NCS MODEL

We perform our analysis in the sampled-data framework, that is, we consider *continuous-time* (CT) LTI plants controlled by a *discrete-time* (DT) LTI controllers and zero-order hold. We follow the approach in [7] and model a LNCS as a hybrid system with jumps. We decompose it into a continuous subsystem which governs the evolution of the LNCS between network transmissions $t \in [t_i, t_{i+1}]$, where t_i is the time of the i^{th} network transmission, and a jump subsystem which governs the jumps in states due to the network transmissions. To focus on the effects of communication constraints we assume that the transmissions are instantaneous, the communication channel is noiseless, and that there are no dropouts. We let the transmissions occur at the equidistant instants, $t_{i+1} - t_i = T$, in which case the constant T can be interpreted as TI.

We use two-dimensional time arguments (t, i) in our LNCS model [14], where the first argument denotes the continuous time $t \in \mathbb{R} \cup \{0\}$, while the second argument denotes the total number of jumps that occurred, $i \in \mathbb{N}$.

We describe the evolution of the continuous subsystem by

$$\tau \in [0, T] \quad \begin{cases} \dot{x}_p = A_p x_p + B_p \hat{u}, \\ \dot{x}_c = 0, \\ \dot{\hat{y}} = 0, \\ \dot{\hat{u}} = 0, \\ \dot{\tau} = 1, \end{cases} \quad (1)$$

while the jump subsystem is given by

$$\tau \in [T, \infty] \quad \begin{cases} x_p^+ = x_p, \\ x_c^+ = A_c x_c + B_c (\Pi_y \hat{y} + (I - \Pi_y) y), \\ \hat{y}^+ = \Pi_y \hat{y} + (I - \Pi_y) y, \\ \hat{u}^+ = \Pi_u \hat{u} + (I - \Pi_u) u, \\ \tau^+ = 0, \end{cases} \quad (2)$$

where $y = C_p x_p$ and $u = C_c x_c + D_c \hat{y}$.

The first lines in (1) and (2) describe respectively continuous and jump evolution of the plant, where $x_p \in \mathbb{R}^{n_p}$ are plant's states, $y \in \mathbb{R}^m$ is the measured output, and $\hat{u} \in \mathbb{R}^r$ is the control input. The plant is equipped with n_y sensors and n_u actuators grouped into respectively l_y and l_u nodes, $l_y \leq n_y$, $l_u \leq n_u$, $l \triangleq l_y + l_u$, typically based on their physical proximity. We assume that this grouping is fixed, and for simplicity, that node $j \in \{1, \dots, l\} \triangleq \mathbb{N}_l$ contains either sensors or actuators. Without loss of generality we let nodes $j \in \mathbb{N}_{l_y}$ contain sensors and nodes $j + l_y$, $j \in \mathbb{N}_{l_u}$, contain actuators. We denote the output components measured in node $j \in \mathbb{N}_{l_y}$ with $y_j \in \mathbb{R}^{m_j}$, $\sum_j m_j = m$, $y = [y_1^T \dots y_{l_y}^T]^T$, and input components driving actuators in node $l_y + j$, $j \in \mathbb{N}_{l_u}$, with $\hat{u}_j \in \mathbb{R}^{r_j}$, $\sum_j r_j = r$, $\hat{u} = [\hat{u}_1^T \dots \hat{u}_{l_u}^T]^T$.

The second lines in (1) and (2) describe continuous and jump evolution of the controller L , where $x_c \in \mathbb{R}^{n_c}$ are its states, $\hat{y} \in \mathbb{R}^m$ is its input and $u \in \mathbb{R}^r$ is its output. Controller's input and output are partitioned analogously to the partition of plant's output and input, $\hat{y} = [\hat{y}_1^T \dots \hat{y}_{l_y}^T]^T$, $\hat{y}_j \in \mathbb{R}^{m_j}$, $u = [u_1^T \dots u_{l_u}^T]^T$, $u_j \in \mathbb{R}^{r_j}$. We assume that the number of controller states n_c is given and restrict the controller L to be parameterized by matrices A_c, B_c, C_c, D_c which are to be computed, that is, $L \in \mathcal{L}_{n_c} \triangleq \{(A_c, B_c, C_c, D_c) : A_c \in \mathbb{R}^{n_c \times n_c}, B_c \in \mathbb{R}^{n_c \times m}, C_c \in \mathbb{R}^{r \times n_c}, D_c \in \mathbb{R}^{r \times m}\}$.

Communication between plant's sensors and actuators, and the controller is via the network. At each network transmission

we assume that *exactly one* node sends/receives the current value of its associated signals. Communication constraints of the network induce a mismatch between plant's inputs/outputs and controller's outputs/inputs which is governed by the third and forth lines in (1) and (2). In other words, $\hat{y}$ and $\hat{u}$ represent the most recent versions of plant's and controller's outputs, y and u respectively, transmitted via the network. The decision which node is to transmit at $(t_i, i - 1)$ is specified by the diagonal matrix $\Pi \triangleq \text{diag}\{\Pi_y, \Pi_u\}$ which belongs to the set $\mathcal{P} \triangleq \{\Pi_1, \dots, \Pi_l\}$, where $\Pi_j = \text{diag}\{\Pi_{y_j}, \Pi_{u_j}\}$, $\Pi_{y_j} = \text{diag}\{\delta(j-1)I_{m_1}, \dots, \delta(j-l_y)I_{m_{l_y}}\}$, $\Pi_{u_j} = \text{diag}\{\delta(j-l_y-1)I_{r_1}, \dots, \delta(j-l)I_{r_{l_u}}\}$, and $\delta : \mathbb{N} \cup \{0\} \rightarrow \{0, 1\}$ is Cronecker delta function. If at $(t_i, i - 1)$ node $j^* \in \mathbb{N}_{l_y}$ is granted the right to access the network then $\Pi = \Pi_{j^*}$, and controller's input component $\hat{y}_{j^*}$ is set to the actual value of plant's output component y_{j^*} , while the other controller's and plant's input components are kept constant. Analogous situation occurs for $j^* \in \{l_y + 1, \dots, l\}$, except then plants's input component $\hat{u}_{j^*}$ is set to u_{j^*} . We treat the matrix Π as a control input and refer to an algorithm according to which it takes values in the set $\mathcal{P}$ as a *protocol*.

By substituting the network induced errors

$$e(t, i) \triangleq \begin{bmatrix} e_y(t, i) \\ e_u(t, i) \end{bmatrix} \triangleq \begin{bmatrix} \hat{y}(t, i) - y(t, i) \\ \hat{u}(t, i) - u(t, i) \end{bmatrix} \quad (3)$$

into LNCS (1)-(2), we derive the network-induced DT model which relates the states $X \triangleq [x^T \ e^T]^T \triangleq [x_p^T \ x_c^T \ e_y^T \ e_u^T]^T$ at the instants $(t_i, i - 1)$ and (t_{i+1}, i)

$$X^+ = \hat{A} \hat{\Pi} X, \quad (4)$$

where $X \triangleq X(t_i, i - 1)$, $X^+ \triangleq X(t_{i+1}, i)$, $A_d = e^{A_p T}$, $B_d = \int_0^T e^{A_p t} dt B_p$, $C_d = C_p$, and

$$\begin{aligned} \hat{A} &\triangleq \begin{bmatrix} A_T & B_T \\ C_T(I - A_T) & I - C_T B_T \end{bmatrix}, \quad \hat{\Pi} \triangleq \begin{bmatrix} I & 0 \\ 0 & \tilde{\Pi} \end{bmatrix}, \\ A_T &\triangleq \begin{bmatrix} A_d + B_d D_c C_d & B_d C_c \\ B_c C_d & A_c \end{bmatrix}, \quad B_T \triangleq \begin{bmatrix} B_d D_c & B_d \\ B_c & 0 \end{bmatrix}, \\ C_T &\triangleq \begin{bmatrix} C_d & 0 \\ 0 & C_c \end{bmatrix}, \quad \tilde{\Pi} \triangleq \begin{bmatrix} \Pi_y & 0 \\ D_c(I - \Pi_y) & \Pi_u \end{bmatrix}. \end{aligned}$$

III. PARTIAL STATE CONTROL LYAPUNOV FUNCTIONS

We introduce *weak partial state control Lyapunov functions* (WPSCLFs) which represent an extension of the control Lyapunov functions [15] to the case of partial state feedback stabilization. We demonstrate their usefulness on protocol synthesis for NCSs and believe that WPSCLFs can be exploited for other control problems with information constraints, such as output feedback stabilization. The motivation to use WPSCLFs for protocol synthesis stems from the fact that a protocol is practically implementable if it does not depend on *any* plant and/or controller states, but only on the network induced errors [3]–[6]. We consider a DT system

$$\begin{aligned} x_1^+ &= f_1(x_1, x_2, u), \quad x \triangleq [x_1^T \ x_2^T]^T \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}, \\ x_2^+ &= f_2(x_1, x_2, u), \quad u \in \mathcal{U}, \end{aligned} \quad (5)$$

where the set of input constraints $\mathcal{U}$ and C^0 functions $f_1, f_2 : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathcal{U} \rightarrow \mathbb{R}^{n_1}$ are such that the origin $x_e \triangleq [0 \ 0]^T \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ is an equilibrium of system (5), that is, there exists $u^0 \in \mathcal{U}$ such that $f_1(0, 0, u^0) = 0$ and $f_2(0, 0, u^0) = 0$. We

particularly consider the case when the set $\mathcal{U}$ contains finitely many elements which happens for protocol synthesis in NCSs.

Definition 1: A C^0 function $V : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^+$ is a WPSCLF for system (5) with respect to (wrt) states x_2 if there exist functions $\alpha_1^V, \alpha_2^V, \alpha_3^V \in \mathcal{K}^\infty$ such that :

- a) $\alpha_1^V(\|x\|) \leq V(x_1, x_2) \leq \alpha_2^V(\|x\|)$,
- b) $\inf_{u \in \mathcal{U}} \sup_{x_1 \in \mathbb{R}^{n_1}} \Delta V(x_1, x_2, u) \leq -\alpha_3^V(\|x_2\|)$, where $\Delta V(x_1, x_2, u) \triangleq V(x_1^+, x_2^+) - V(x_1, x_2)$. ■

A consequence of Definition 1 is that existence of a WPSCLF implies existence of a partial state feedback law $u = \sigma(x_2)$, $\sigma : \mathbb{R}^{n_2} \rightarrow \mathcal{U}$ which renders x_e globally stable (GS) for system (5). For example, any feedback law satisfying

$$\sigma(x_2) \in \{u \in \mathcal{U} : \sup_{x_1} \Delta V(x_1, x_2, u) \leq -\frac{1}{2}\alpha_3^V(\|x_2\|)\}, \quad (6)$$

renders x_e GS, while if the infimum over the set $\mathcal{U}$ is achieved then feedback law (6) can be replaced by

$$\sigma(x_2) = \arg \min_{u \in \mathcal{U}} \sup_{x_1} \Delta V(x_1, x_2, u). \quad (7)$$

However, existence of a WPSCLF for system (5) does not imply existence of a partial state feedback law which renders x_e globally asymptotically stable, as illustrated in Example 1.

Example 1: Consider the second order system

$$x_1^+ = x_1, \quad x_2^+ = 2x_2 + u, \quad x_1, x_2, u \in \mathbb{R}, \quad (8)$$

which is neither stabilizable nor detectable from x_2 . Consequently there does not exist any feedback law which renders x_e globally asymptotically stable (GAS). However, a function $V(x_1, x_2) = x_1^2 + x_2^2$ is a WPSCLF for system (8), since $\inf_u \sup_{x_1} \Delta V(x_1, x_2, u) = -x_2^2$. We note that the partial state feedback law derived from $V(x_1, x_2)$ is $\sigma(x_2) \triangleq \arg \min_u \sup_{x_1} \Delta V(x_1, x_2, u) = -2x_2$ and renders x_e GS. □

Example 1 reveals the necessity of detectability of states x_1 from states x_2 for existence of a partial state feedback law which renders the origin x_e GAS for system (5). Rather than imposing this detectability property directly on system (5), we require the existence of an additional *detectability* function (DF) whose increments along solutions of system (5) contain a negative definite term and a possibly positive term in the known states x_2 and control u .

Definition 2: A C^0 function $W : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^+$ is a DF for system (5) wrt states x_1 if there exist functions $\alpha_1^W, \alpha_2^W, \alpha_3^W \in \mathcal{K}^\infty$, and a C^0 function $\gamma : \mathbb{R}^{n_2} \times \mathcal{U} \rightarrow \mathbb{R}$, $\gamma(0, u^0) = 0$, such that

- a) $\alpha_1^W(\|x\|) \leq W(x_1, x_2) \leq \alpha_2^W(\|x\|)$,
- b) $\Delta W(x_1, x_2, u) \leq -\alpha_3^W(\|x\|) + \gamma(x_2, u)$. ■

Lemma 1: Suppose that

- a) system (5) possesses a WPSCLF V wrt states x_2 ,
- b) system (5) possesses a DF W wrt states x_1 ,
- c) there exists a function $\tilde{\gamma} \in \mathcal{K}^\infty$ such that *partial state* feedback law (6) satisfies $\gamma(x_2, \sigma(x_2)) \leq \tilde{\gamma}(\|x_2\|)$,
- d) $\limsup_{\|x_2\| \rightarrow \infty} \frac{\tilde{\gamma}(\|x_2\|)}{\alpha_3^V(\|x_2\|)} < \infty$.

Then feedback law (6) renders the origin x_e GAS for system (5), and there exists a Lyapunov function $\tilde{V} : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^+$ and functions $\alpha_1^{\tilde{V}}, \alpha_2^{\tilde{V}}, \alpha_3^{\tilde{V}} \in \mathcal{K}^\infty$ such that

- i) $\alpha_1^{\tilde{V}}(\|x\|) \leq \tilde{V}(x_1, x_2) \leq \alpha_2^{\tilde{V}}(\|x\|)$,
- ii) $\Delta \tilde{V}(x_1, x_2, \sigma(x_2)) < -\alpha_3^{\tilde{V}}(\|x\|)$.

Furthermore, if the functions V, W and their bounds α_i^V, α_i^W are quadratic, then so is the function $\tilde{V}$ and its bounds $\alpha_i^{\tilde{V}}$, $i = 1, 2, 3$, and the condition **d)** is automatically satisfied. ■

Proof: The conditions **b)** and **c)** imply output-to-state stability (OSS) of system (5) and (6), [16]. Combining OSS, the conditions **a)** and **d)**, and Corollary 5.1 in [17], the claim of Lemma 1 follows. Lemma 1 is true for feedback law (7) under the same conditions, if the appropriate infimum is achieved. ■

Remark 1: Feedback laws (6) and (7) may not be continuous due to taking the supremum over a noncompact set $\{x_1 : x_1 \in \mathbb{R}^{n_1}\}$ and possibly finite cardinality of the set $\mathcal{U}$. Such continuity is typically assumed in similar situations [18], but instead we require a weaker property in the condition c) of Lemma 1. This condition holds, for example, when feedback laws (6) and (7) are continuous or when the set $\mathcal{U}$ is finite. ■

Remark 2: We recently became aware that output CLFs, closely related to WPSCLFs, were introduced in [18] and used to obtain the necessary conditions for output feedback stabilization of CT nonlinear systems affine in control. ■

IV. NECESSARY AND SUFFICIENT CONDITIONS FOR QUADRATIC STABILITY OF LNCS

Here we use WPSCLFs for synthesis of protocols which are solely dependent on network induced errors. We consider NCSs with LTI plants and controllers, and are interested in quadratic stability. Moreover, the number of elements in the set of input constraints $\mathcal{P}$ for protocol is equal to the number of network nodes. For this case we strengthen Lemma 1 and show that existence of a quadratic WPSCLF is the necessary and sufficient condition for quadratic stability of LNCS (4). We note that in this case WPSCLFs achieve inf and sup in Definition 1, which are thus replaced with min and max.

Definition 3: LNCS (4) is quadratically stable (QS) if for a given controller $L \in \mathcal{L}_{n_c}$ and protocol $\Pi = \Omega(e)$, $\Omega : \mathbb{R}^{m+r} \rightarrow \mathcal{P}$, there exists a matrix $P = P^T > 0$ such that the Lyapunov function

$$V(x, e) = X^T P X \triangleq \begin{bmatrix} x \\ e \end{bmatrix}^T \begin{bmatrix} \hat{P}_{11} & \hat{P}_{12} \\ \hat{P}_{12}^T & \hat{P}_{22} \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}, \quad (9)$$

satisfies for all $X \triangleq [x^T \ e^T]^T \neq X_e \triangleq [0 \ 0]^T$

$$\Delta V(x, e, \Omega(e)) = X^T (\hat{\Pi}^T(e) Q \hat{\Pi}(e) - P) X < 0, \quad (10)$$

$$Q \triangleq \hat{A}^T P \hat{A} = \begin{bmatrix} \hat{Q}_{11} & \hat{Q}_{12} \\ \hat{Q}_{12}^T & \hat{Q}_{22} \end{bmatrix}, \quad \hat{\Pi}(e) \triangleq \hat{\Pi}|_{\Pi=\Omega(e)}, \quad \hat{\Pi}(e) \triangleq \hat{\Pi}|_{\Pi=\Omega(e)}.$$

LNCS (4) is *protocol quadratically stabilizable* (pQS) if for a given controller $L \in \mathcal{L}_{n_c}$ there exists a function $\Omega : \mathbb{R}^{m+r} \rightarrow \mathcal{P}$, such that LNCS (4) controlled by the protocol $\Pi = \Omega(e)$ is QS. LNCS (4) is *protocol and controller quadratically stabilizable* (pcQS) if there exists a controller $L \in \mathcal{L}_{n_c}$ such that LNCS (4) is pQS. ■

Theorem 1: A quadratic function (9) is a WPSCLF wrt the network induced errors e for LNCS (4) *if and only if* (iff) the protocol

$$\Omega^V(e) \triangleq \arg \min_{\Pi \in \mathcal{P}} \sup_{x \in \mathbb{R}^{n_p+n_c}} \Delta V(x, e, \Pi), \quad (11)$$

renders LNCS (4) QS. ■

Proof: Necessity is a straightforward consequence of Definitions 1 and 3. Here we show sufficiency, that is, we show

that for LNCS (4) the condition **a)** in Lemma 1 implies the conditions **b)-d)**. Toward this end we utilize two observations. First, quadratic function (9) is a WPSCLF for LNCS (4) iff the quadratic function

$$\bar{V}(x, e) \triangleq \begin{bmatrix} x \\ e \end{bmatrix}^T \begin{bmatrix} \hat{P}_{11} & -\hat{P}_{12} \\ -\hat{P}_{12}^T & \hat{P}_{22} \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} \quad (12)$$

is a WPSCLF, since $\max_x \Delta V(x, e, \Pi) = \max_x \Delta \bar{V}(x, e, \Pi)$. Second, condition (10) implies $\hat{Q}_{11} < \hat{P}_{11}$, which for WPSCLFs (9) and (12) respectively amounts to

$$\begin{aligned} & A_T^T \hat{P}_{11} A_T + (I - A_T)^T C_T^T \hat{P}_{22}^T C_T (I - A_T) \\ & + A_T^T \hat{P}_{12} C_T (I - A_T) + (I - A_T)^T C_T^T \hat{P}_{12}^T A_T < \hat{P}_{11}, \\ & A_T^T \hat{P}_{11} A_T + (I - A_T)^T C_T^T \hat{P}_{22}^T C_T (I - A_T) \\ & - A_T^T \hat{P}_{12} C_T (I - A_T) - (I - A_T)^T C_T^T \hat{P}_{12}^T A_T < \hat{P}_{11}. \end{aligned}$$

Adding these inequalities we obtain the Lyapunov inequality

$$A_T^T \hat{P}_{11} A_T - \hat{P}_{11} < -(I - A_T)^T C_T^T \hat{P}_{22}^T C_T (I - A_T) \leq 0, \quad (13)$$

which implies that the matrix A_T is Schur. Now, there are multiple ways to construct a DF for LNCS (4) in terms of WPSCLF V . For example, it can be shown that the function $W(x, e) \triangleq x^T \hat{P}_{11} x + e^T \hat{P}_{22} e$ is one such DF, since it satisfies $\Delta W(x, e, \Pi) < -\frac{1}{2} h_m \|X\|^2 + (\tilde{\Pi} e)^T G \tilde{\Pi} e$, where $h_m > 0$ is the minimal eigenvalue of $H \triangleq \hat{P}_{11} - A_T^T \hat{P}_{11} A_T - (I - A_T)^T C_T^T \hat{P}_{22}^T C_T (I - A_T)$, and $G_{xe} \triangleq A_T^T \hat{P}_{11} B_T^T + (I - A_T)^T C_T^T \hat{P}_{22} (I - C_T B_T)$, $G_{ee} \triangleq B_T^T \hat{P}_{11} B_T + (I - C_T B_T)^T \hat{P}_{22} (I - C_T B_T)$, $G \triangleq \frac{1}{2} h_m + G_{ee} + \frac{2}{h_m} G_{xe}^T G_{xe}$. Then the condition **b)** in Lemma 1 holds with $\gamma(e, \Pi) = (\tilde{\Pi} e)^T G \tilde{\Pi} e$. Since the set $\mathcal{P}$ is finite the condition **c)** holds independently of the protocol $\Pi = \Omega(e)$, $\gamma(e, \Pi) \leq \tilde{\gamma}(\|e\|) \triangleq g^* \|e\|^2$, $g^* \triangleq \max_{\Pi \in \mathcal{P}} \|\tilde{\Pi}^T G \tilde{\Pi}\|$. The condition **d)** is true automatically because all bounding functions are quadratic. Hence, protocol (11), which is derived by applying feedback law (7) to LNCSs, renders LNCS (4) QS. ■

Note that Lyapunov inequality (13) implies that LNCS (4) is pQS only if the controller renders the plant QS without the network, that is, only if the matrix A_T is Schur.

Corollary 1: For a given controller $L \in \mathcal{L}_{n_c}$ LNCS (4) is pQS only if the matrix A_T is Schur. ■

Remark 3: Synthesis of protocols by utilizing CLFs has already been proposed in [11]. In [11], however, the constructed protocols depend on the plant states, and the presence of network induced errors is avoided by setting to zero all actuator and controller inputs which can not be updated. ■

Now we analyze protocol (11) in more detail. We define the set $\mathcal{S}_j$, $j \in \mathbb{N}_l$, to be the set of all network induced errors e for which the minimum of $\Pi \in \mathcal{P}$ in (11) is equal to Π_j ,

$$\mathcal{S}_j \triangleq \{e : \max_x \Delta V(x, e, \Pi_j) \leq \max_x \Delta V(x, e, \Pi_k), \forall k \in \mathbb{N}_l\}.$$

By maximizing function $\Delta V(x, e, \Pi)$ in (10) it can be shown that the sets $\mathcal{S}_j$ can equivalently be represented by

$$\begin{aligned} \mathcal{S}_j &= \{e : e^T (\hat{H}(\Pi_j, M_1) - \hat{H}(\Pi_k, M_1)) e \leq 0, \forall k \in \mathbb{N}_l\}, \\ \hat{H}(\Pi, M_1) &= \begin{bmatrix} H_{11}(\Pi, \tilde{M}_1) & H_{12}(\Pi, \tilde{M}_1) \\ H_{12}^T(\Pi, \tilde{M}_1) & H_{22}(\Pi, \tilde{M}_1) \end{bmatrix}, \end{aligned} \quad (14)$$

where $H_{11} \triangleq \Pi_y S_1 \Pi_y + \Pi_y S_2 + S_2^T \Pi_y$, $H_{12} \triangleq \Pi_y S_3^T \Pi_u + \Pi_y S_4^T + S_5^T \Pi_u$, $H_{22} \triangleq \Pi_u S_6 \Pi_u + \Pi_u S_7 + S_7^T \Pi_u$, and

$M_1 \in \mathcal{M}_1$ is a collection of matrices, $\mathcal{M}_1 \triangleq \{(S_1, \dots, S_7) : S_1, S_2 \in \mathbb{R}^{m \times m}, S_3, S_4, S_5 \in \mathbb{R}^{m \times r}, S_6, S_7 \in \mathbb{R}^{r \times r}, S_1 = S_1^T, S_6 = S_6^T\}$. The dependence of matrices S_i on the plant parameters A_d, B_d, C_d , and the unknown parameters P, L is omitted due to the space constraints. Sets (14) are closed, satisfy $\cup_{j=1}^l \mathcal{S}_j = \mathbb{R}^{m+r}$, and have the property that $\Pi = \Pi_j$ only if $e \in \mathcal{S}_j$. However, the intersection of two such sets may not be empty and for $e \in \mathcal{S}_j \cap \mathcal{S}_k$, $j \neq k$, protocol (11) can not uniquely assign the next transmission. We avoid this ambiguity by assigning the next transmission to the node with the smallest identifier j^* whose set $\mathcal{S}_{j^*}$ contains the current e

$$\Omega^V(e) = \Pi_{j^*}, \quad j^* = \min\{j \in \mathbb{N}_l : e \in \mathcal{S}_j\}. \quad (15)$$

Similar situation arises for all protocols synthesized here and they are always assumed to be implemented in conjunction with (15), even if this is not explicitly stated. The following Corollary follows from Definition 3, Theorem 1, and (14).

Corollary 2: LNCS (4) is pcQS iff there exists a controller $L \in \mathcal{L}_{n_c}$ and a matrix $P = P^T > 0$ such that quadratic function (9) satisfies $\forall j \in \mathbb{N}_l$,

$$\forall e \in \mathcal{S}_j, \quad e \neq 0, \quad \max_{x \in \mathbb{R}^{n_c+n_p}} \Delta V(x, e, \Pi_j) < 0. \quad (16)$$

V. COMPUTATIONALLY TRACTABLE SUFFICIENT CONDITION FOR QUADRATIC STABILITY OF LNCS

The computation of the unknown parameters P and L for which condition (16) holds is not tractable because sets (14) are specified as the pointwise minimum of l quadratic functions dependent on P and L . We avoid this difficulty by introducing new sets

$$\tilde{\mathcal{S}}_j \triangleq \{e : e^T (\hat{H}(\Pi_j, \tilde{M}) - \hat{H}(\Pi_k, \tilde{M})) e \leq 0, \forall k \in \mathbb{N}_l\},$$

$\tilde{M} = (\tilde{S}_1, \dots, \tilde{S}_7) \in \mathcal{M}_1$, specified by independent matrices $\tilde{S}_i$, rather than by matrices $S_i(P, L)$ derived from the parameters P, L . Due to this more flexible parameterization, it can be shown that for any l -tuple of sets $\mathcal{S}_j$ specified by P and L via (14), there exists $\tilde{M} \in \mathcal{M}_1$ such that sets $\tilde{\mathcal{S}}_j$ satisfy $\tilde{\mathcal{S}}_j = \mathcal{S}_j$, $\forall j \in \mathbb{N}_l$. This property of sets $\tilde{\mathcal{S}}_j$ preserves the necessity of the conditions in Theorem 2 for pcQS of LNCS (4).

Theorem 2: LNCS (4) is pcQS iff there exist a matrix $P = P^T > 0$, a controller $L \in \mathcal{L}_{n_c}$, and sets $\tilde{\mathcal{S}}_j$ parameterized by $\tilde{M} \in \mathcal{M}_1$, such that quadratic function (9) satisfies $\forall j \in \mathbb{N}_l$

$$\forall e \in \tilde{\mathcal{S}}_j, \quad e \neq 0, \quad \max_{x \in \mathbb{R}^{n_p+n_c}} \Delta V(x, e, \Pi_j) < 0. \quad (17)$$

Furthermore, the synthesized protocol $\Pi = \tilde{\Omega}(e)$ is given by

$$\tilde{\Omega}(e) = \arg \min_{\Pi \in \mathcal{P}} e^T \hat{H}(\Pi, \tilde{M}) e. \quad (18)$$

To derive conditions for pcQS of LNCS (4) in terms of matrix inequalities, we next apply the $\mathcal{S}$ -procedure [19] on condition (17). However, use of the $\mathcal{S}$ -procedure leads to loss of necessity of the obtained conditions for pcQS of LNCS (4).

Corollary 3: LNCS (4) is pcQS if there exist a matrix $P = P^T > 0$, a controller $L \in \mathcal{L}_{n_c}$, sets $\tilde{\mathcal{S}}_j$ parameterized by $\tilde{M} \in \mathcal{M}_1$, and parameters $\tau_{jk} \geq 0$, $k \in \mathbb{N}_l$, such that $\forall j \in \mathbb{N}_l$,

$$\begin{bmatrix} P + \sum_{k=1}^l \tau_{jk} (\tilde{H}(\Pi_j, \tilde{M}) - \tilde{H}(\Pi_k, \tilde{M})) & \star \\ P \hat{A}_j & P \end{bmatrix} > 0. \quad (19)$$

where $\hat{A}_j \triangleq \hat{A}\hat{\Pi}_j$, $\hat{\Pi}_j \triangleq \hat{\Pi}|_{\Pi=\Pi_j}$, and $\tilde{H}(\Pi_j, \tilde{M}) \triangleq \text{diag}\{0, \tilde{H}(\Pi_j, \tilde{M})\}$. Furthermore, if $l = 2$ then condition (19) is also necessary for pcQS of LNCS (4). ■

Dynamics of LNCS (4) are a bilinear function of controller's parameters. When the number of controller states n_c is not smaller than the number of controller outputs r , $n_c \geq r$, without loss of generality it can be assumed that the controller $\tilde{L}$ is in the observable canonical form, $\tilde{L} \in \tilde{\mathcal{L}}_{n_c} \triangleq \{L \in \mathcal{L}_{n_c} : C_c = [I_r \ 0]\}$, in which case dynamics of LNCS (4) become linearly dependent on $\tilde{L}$. However, our results also apply to the static output feedback case, $n_c = 0$, by setting the matrices A_c, B_c, C_c to zero in LNCS (4).

Corollary 4: There exist a matrix $P = P^T > 0$, a controller $L \in \mathcal{L}_{n_c}$, $n_c \geq r$, and parameters $\tilde{M} \in \mathcal{M}_1$, $\tau_{jk} \geq 0$, $k \in \mathbb{N}_l$, such that condition (19) holds, iff for the same parameters $\tilde{M}$ and τ_{jk} there exist a matrix $P_T = P_T^T > 0$ and a controller $\tilde{L} \in \tilde{\mathcal{L}}_{n_c}$ such that condition (19) holds. ■

Sufficient condition (19) for pcQS of LNCS (4) is a bilinear matrix inequality (BMI) in the unknowns $\tilde{L} \in \tilde{\mathcal{L}}_{n_c}$, P , $\tilde{M}$, and τ_{jk} 's due to products $\tau_{jk}(\tilde{H}(\Pi_j, \tilde{M}) - \tilde{H}(\Pi_k, \tilde{M}))$ and $P\hat{A}_j$. Computational complexity of solving a BMI is greater than solving a linear matrix inequality (LMI), but there exist several algorithms for solving BMIs in the literature. We solve BMI (19) by modifying the path-following algorithm proposed in [20] which relies on linearizing a BMI, and solving a sequence of the corresponding LMIs whose solution converges to the solution of the original BMI.

Linearized BMI Algorithm (LBMIA)

S1: Select the order of the dynamic controller $n_c \geq r$, and set the initial controller to zero, $\tilde{L}^0 = (0, 0, 0) \in \tilde{\mathcal{L}}_{n_c}$.

S2: Let $i := 0$. Choose $0 < \delta a \ll 1$, $P^0 = I$, $\tilde{M}^0 = (0, \dots, 0) \in \mathcal{M}_1$, $\tau_{jk}^0 = 1$, $k, j \in \mathbb{N}_l$, and compute $\hat{A}_j^0 \triangleq \hat{A}_j(\tilde{L}^0)$, $a^0 = \max_j \|\hat{A}_j^0\|$, $\tilde{H}(\Pi_j, \tilde{M}^0)$, $j \in \mathbb{N}_l$.

S3: Solve the feasibility LMI problem in decision variables δP , $\delta \hat{A}_j(\delta L)$, $\delta L \in \tilde{\mathcal{L}}_{n_c}$, $\delta \tilde{M} = (\delta S_1, \dots, \delta S_7) \in \mathcal{M}_1$, $\delta \tau_{jk}$,

$$\begin{aligned} P^i + \delta P > 0, \quad \|\delta P\| \leq b^i, \quad \tau_{jk}^i + \delta \tau_{jk} \geq 0, \quad \|\delta \hat{A}_j\| \leq b^i, \\ \|\tilde{H}(\Pi_j, \delta \tilde{M}) - \tilde{H}(\Pi_k, \delta \tilde{M})\| \leq \frac{b^i}{\sqrt{l-1}}, \quad |\delta \tau_{jk}| \leq \frac{b^i}{\sqrt{l-1}}, \\ \begin{bmatrix} \kappa^i(P^i + \delta P) + W_j^i & \star \\ P^i(\delta \hat{A}_j + \hat{A}_j^i) + \delta P \hat{A}_j^i & P^i + \delta P \end{bmatrix} > \epsilon I, \quad j, k \in \mathbb{N}_l, \end{aligned} \quad (20)$$

where $W_j^i \triangleq \sum_{k=1}^l ((\tau_{jk}^i + \delta \tau_{jk})(\tilde{H}(\Pi_j, \tilde{M}^i) - \tilde{H}(\Pi_k, \tilde{M}^i)) + \tau_{jk}^i(\tilde{H}(\Pi_j, \delta \tilde{M}) - \tilde{H}(\Pi_k, \delta \tilde{M})))$, $b_i = \sqrt{\frac{1}{2}\epsilon + \kappa_i - 1}$,

$$\delta \hat{A}_j \triangleq \begin{bmatrix} B_d \delta D_c C_d & 0 & B_d \delta D_c & 0 \\ \delta B_c C_d & \delta A_c & \delta B_c \Pi_{yy} & 0 \\ -C_d B_d \delta D_c C_d & 0 & -C_d B_d \delta D_c & 0 \\ -C_c \delta B_c C_d & -C_c \delta A_c & \delta D B_j & 0 \end{bmatrix},$$

$\delta D B_j \triangleq \delta D_c(I - \Pi_{yy}) - C_c \delta B_c \Pi_{yy}$, $0 < \epsilon \ll 1$, and $\kappa^i = 1 + (a^i - \delta a)^2$ for $a^i \geq \delta a$ and $\kappa^i = 1$ for $a^i < \delta a$.

S4: If the LMI problem in S3 is feasible, let $a^{i+1} = a^i - \delta a$, $P^{i+1} = P^i + \delta P$, $\hat{A}_j^{i+1} = \hat{A}_j^i + \delta \hat{A}_j$, $\tau_{jk}^{i+1} = \tau_{jk}^i + \delta \tau_{jk}$, and $\tilde{M}^{i+1} = \tilde{M}^i + \delta \tilde{M}$, where the last addition is component-wise.

S5: Set $i := i + 1$. Stop if $a^i < 0$, otherwise go to S3.

Remark 4: The parameter a^i represents a level of instability of LNCS (4) with the current controller $\tilde{L}^i$. We initially set a^0 to the level of instability of uncontrolled LNCS (4) and in each iteration we modify the controller to reduce this level

for δa , with the ultimate goal of achieving $a^i < 0$. Moreover, norms of the unknowns in the LMI problem S3 are bounded by a function of a^i and ϵ to ensure that feasibility of LMI (20) for $a^i < \delta a$ implies sufficient condition (19). ■

VI. IMPLEMENTATION ISSUES

In NCSs implemented via control area network, at each transmission instant all nodes simultaneously attempt to transmit a packet containing the current value of their signals. The network compares transmission priorities (TP_j) in packets headers' and assigns the next transmission to the node with the highest priority, while disabling all other nodes. Thus, protocol can be implemented by manipulating transmission priorities as proposed in [3]-[6]. We assumed so far that each node has the knowledge of *all* networked induced errors but not of plant's and/or controller's states. Then protocol (18) can be implemented by setting the transmission priority for node j to

$$TP_j = p(e) - e^T \hat{H}(\Pi_j, \tilde{M})e, \quad (21)$$

where $p: \mathbb{R}^{m+r} \rightarrow \mathbb{R}$ is any function of e independent of the matrix Π , in which case nodes selected by $\arg \max_j TP_j$ and protocol (18) coincide. However, in some NCSs node j can only use its own network induced error e_j , $e = [e_1^T \dots e_l^T]^T$, to compute its transmission priority TP_j , hence TP_j defined by (21) can not be computed locally at node j . For such NCSs we select the function $p(e)$ and impose a particular structure on the matrix $\hat{H}(\Pi, \tilde{M})$ such that TP_j (21) is solely a function of e_j . Namely, if $\hat{H}(\Pi, \tilde{M})$ and $p(e)$ are defined by

$$\hat{H}(\Pi, \tilde{M}) \triangleq \Pi \text{diag}\{Q_1, \dots, Q_l\} \Pi, \quad p(e) \triangleq e^T \hat{H}(\Pi, \tilde{M})e,$$

where $\tilde{M} \in \mathcal{Q} = \{(Q_1, \dots, Q_l) : Q_j = Q_j^T > 0, j \in \mathbb{N}_l\}$, then (21) reduces to $TP_j = e_j^T Q_j e_j$, which can be computed locally at node j . The sets $\tilde{\mathcal{S}}_j$ corresponding to parametrization $\tilde{M} \in \mathcal{Q}$ are $\tilde{\mathcal{S}}_j = \{e : e_j^T Q_j e_j \geq e_k^T Q_k e_k, \forall k \in \mathbb{N}_l\}$, $j \in \mathbb{N}_l$.

Corollary 5: LNCS (4) is pcQS if there exist a matrix $P = P^T > 0$, a controller $\tilde{L} \in \tilde{\mathcal{L}}_{n_c}$, and parameters $\tilde{M} \in \mathcal{Q}$, $\tau_{jk} \geq 0$, $k, j \in \mathbb{N}_l$, such that $\forall j \in \mathbb{N}_l$,

$$\begin{bmatrix} P + \sum_{k=1}^l \tau_{jk} (\tilde{H}(\Pi_j, \tilde{M}) - \tilde{H}(\Pi_k, \tilde{M})) & \star \\ P \hat{A}_j & P \end{bmatrix} > 0.$$

Furthermore, to implement the resulting protocol $\Pi = \tilde{\Omega}(e) \triangleq \arg \max_{j \in \mathbb{N}_l} e_j^T Q_j e_j$, node j requires only its own network induced error e_j . ■

Protocol $\Pi = \tilde{\Omega}(e)$ corresponds to a scaled TOD protocol, [3]-[8]. Besides a different parametrization of the sets $\tilde{\mathcal{S}}_j(\tilde{M})$, $\tilde{M} \in \mathcal{Q}$ instead of $\tilde{M} \in \mathcal{M}_1$, no other changes of the LBMIA are required. In some situations it may be useful to beforehand fix the protocol $\Pi = \tilde{\Omega}(e)$, that is, parameters $\tilde{M}^* \in \mathcal{Q}$. That simplifies the computation in the step S3 of the LBMIA because it makes the term $\sum_{k=1}^l \tau_{jk}(\tilde{H}(\Pi_j, \tilde{M}^*) - \tilde{H}(\Pi_k, \tilde{M}^*))$ linear in the unknowns. Then this step can be replaced with the following, simpler one.

S3: Solve the following feasibility LMI problem in decision variables δP , $\delta A_c, \delta B_c, \delta D_c, \tau_{jk}$,

$$\begin{aligned} a) \quad & P^i + \delta P > 0, \quad \tau_{jk} \geq 0, \quad \|\delta P\| \leq \sqrt{b^i}, \quad \|\delta \hat{A}_j\| \leq \sqrt{b^i}, \\ b) \quad & \begin{bmatrix} \kappa^i(P^i + \delta P) + W_j & \star \\ P^i(\delta \hat{A}_j + \hat{A}_j^i) + \delta P \hat{A}_j^i & P^i + \delta P \end{bmatrix} > \epsilon I, \quad k, j \in \mathbb{N}_l, \end{aligned}$$

where $W_j \triangleq \sum_{k=1}^l \tau_{jk}(\tilde{H}(\Pi_j, \tilde{M}^*) - \tilde{H}(\Pi_k, \tilde{M}^*))$.

VII. CASE STUDY: BATCH REACTOR

We apply LBMIA to a linearized model of a batch reactor

$$\dot{x}_p = A_p x_p + B_p \hat{u}, \quad y = C_p x_p, \quad (22)$$

$x_p \in \mathbb{R}^4$, $y, \hat{u} \in \mathbb{R}^2$, where the matrices A_p, B_p, C_p are given in [7], Section IX. The eigenvalues of matrix A_p are $\lambda_1 = 1.991$, $\lambda_2 = 0.063$, $\lambda_3 = -5.065$, and $\lambda_4 = -8.666$, hence open-loop batch reactor (22) is unstable. Our goal is to synthesize controllers and protocols which allow that the transmission interval (TI) between two consecutive network transmissions T be as large as possible, hence reducing the required network bandwidth. We start with $T^k = 0.01s$, $k = 0$, and compute DT model of batch reactor (22) $x_p^+ = A_d(T^k)x_p + B_d(T^k)\hat{u}$, $y_p = C_d(T^k)x_p$, $A_d(T^k) \triangleq e^{A_p T^k}$, $B_d(T^k) \triangleq \int_0^{T^k} e^{A_p t} dt B_p$, and $C_d(T^k) \triangleq C_p$. If the LBMIA successfully terminates for DT model specified by T^k , then in the next iteration, $k := k + 1$, we increment TI for a fixed amount, $T^k := T^{k-1} + \delta T$ and repeat the above procedure. We expect that there exists a critical value of TI, T^* , beyond which the LBMIA fails to terminate successfully. This value of TI, T^* , is interpreted as a MATI bound and its existence is verified by numeric computation.

We consider two network architectures. In the first scenario the network has two nodes, $l = 2$, each containing a sensor measuring one output component, while the controller is located in the vicinity of the actuators. In the second scenario the network has an additional node, $l = 3$, which contains two actuators governing the control inputs. The purpose of the first scenario is to compare MATI bounds obtained via the LBMIA with the ones in the literature, while the purpose of the second scenario is to illustrate the influence of increasing the number of network nodes on a MATI bound.

Depending on the protocol (TOD or RR) and analysis tools used, for a dynamic output feedback controller L^* with two states, $n_c = 2$, given in [3], [7] MATI bounds which guarantee GES of batch reactor (22) are given in Table 1.

In the LBMIA we fix $n_c = 2$ and investigate the effect of different protocols on MATI bounds. We consider protocols obtained by: a) computing $\tilde{M} \in \mathcal{M}_1$, b) computing $\tilde{M} \in \mathcal{Q}$, c) beforehand fixing $\tilde{M}^* \in \mathcal{Q}$. For the case c) we select the parameters $\tilde{M}^* = \text{diag}\{1, 1\}$ for $l = 2$, and $\tilde{M}^* = \text{diag}\{1, 1, I_2\}$ for $l = 3$, which correspond to the original TOD protocol and we modify the LBMIA by replacing S3 with $\hat{S}3$. The values of the remaining parameters are: $\epsilon = 0.001$, $\delta a = 0.04$ and $\delta T = 0.01$ for $l = 2$, and $\delta a = 0.01$ and $\delta T = 0.001$ for $l = 3$. In the first scenario the obtained MATI bounds are 75 times larger than the MATI bounds in [7] and 7500 times larger than the MATI bounds in [3]. In this scenario both protocol parameterizations, $\tilde{M} \in \mathcal{M}_1$ and $\tilde{M} \in \mathcal{Q}$, are equivalent, hence the resulting bounds are equal. However, it is surprising that the largest MATI bound is obtained in the case c) for a beforehand fixed protocol. We believe that this is a consequence of a simpler computation in $\hat{S}3$ versus S3, and on the other hand, comparable degrees of freedom in this scenario for the cases a), b), and c). In the second scenario the obtained MATI bounds are significantly smaller due to the presence of additional network induced errors. In contrast to the first scenario, here the largest MATI bound is obtained with the most general protocol parametrization $\tilde{M} \in \mathcal{M}_1$.

TABLE I

MATI BOUNDS FOR DIFFERENT CONTROLLERS AND PROTOCOLS

Controller	Protocol	l	Bound on MATI [s]
L^*	RR	2	[3]: 10^{-5}
L^*	TOD	2	[3]: 10^{-5}
L^*	RR	2	[7]: 0.0082
L^*	TOD	2	[7]: 0.01
LBMIA via S3	$\tilde{M} \in \mathcal{M}_1$	2	0.75
LBMIA via S3	$\tilde{M} \in \mathcal{Q}$	2	0.75
LBMIA via $\hat{S}3$	fixed $\tilde{M}^* \in \mathcal{Q}$	2	0.81
LBMIA via S3	$\tilde{M} \in \mathcal{M}_1$	3	0.067
LBMIA via S3	$\tilde{M} \in \mathcal{Q}$	3	0.061
LBMIA via $\hat{S}3$	fixed $\tilde{M}^* \in \mathcal{Q}$	3	0.033

Through extensive simulations of batch reactor (22) we observed that for the same TI closed-loop trajectories converge faster to the origin in the first scenario, while the level of control signals is lower in the second scenario. This suggests that by increasing the number of network nodes, the NCS's response becomes more sluggish due to reduction of resulting controller gain, making it thus more sensitive to disturbances.

REFERENCES

- [1] J.P. Hespanha, P. Naghshtabrizi and Y. Xu, "Networked Control Systems: Analysis and Design", submitted to IEEE special issue on Network Control Technology, 2005.
- [2] G.N. Nair, F. Fagnani, S. Zampieri and R.J. Evans, "Feedback Control under Data Rate Constraints: an Overview", submitted to IEEE special issue on Networked Control Technology, 2005.
- [3] G.C. Walsh, O. Belidman, and L.G. Bushnell, "Stability Analysis of Networked Control Systems", IEEE TCST, Vol. 10, pp. 438 - 446, 2002.
- [4] G.C. Walsh, H. Ye and L.G. Busnell, "Asymptotic Behavior of Nonlinear Networked Control Systems", IEEE TAC, Vol. 46, pp. 1093-1097, 2001.
- [5] G.C. Walsh and H. Ye, "Scheduling of Networked Control Systems", IEEE Control System Magazine, Vol. 21, pp. 57-65, February 2001.
- [6] G.C. Walsh, O. Belidman and L.G. Bushnell, "Error Encoding Algorithms for Networked Control Systems", Automatica, Vol. 38, pp. 261-267, 2002.
- [7] D. Nešić and A.R. Teel, "Input-output Stability Properties of Networked Control Systems", IEEE TAC, Vol. 49, pp. 1650 - 1667, 2004.
- [8] D. Nešić and A.R. Teel, "Input to State Stability of Networked Control Systems", Automatica, vol. 40, pp. 2121-2128, 2004.
- [9] M. Tabbara, D. Nešić and A.R. Teel, "Input-output Stability of Wireless Networked Control Systems", in Proc. of the joint 44th IEEE CDC and ECC, Sevilla, Spain, December 2005.
- [10] L. Palopoli, A. Bicchi and A. Sangiovanni Vincentelli, "Numerically Efficient Control of Systems with Communication Constraints", in Proc. of 41st IEEE CDC, Las Vegas, NV, December 2002.
- [11] L. Zhang and D. Hristu-Varsakelis, "Stabilization of Networked Control Systems under Feedback-based Communication", in Proc. of ACC, Portland, OR, June 2005.
- [12] M.M. Ben Gaid and A. Čela, "Model Predictive Control of Systems with Communication Constraints", in Proc. of the 16th IFAC World Congress, Prague, Check Republic, July 2005.
- [13] D.B. Dačić and D. Nešić, "Quadratic Stabilization of Linear Networked Control Systems via Simultaneous Protocol and Controller Synthesis", submitted to Automatica, February 2006.
- [14] R. Goebel, J.P. Hespanha, A.R. Teel, C. Cai and R. Sanfelice, "Hybrid Systems: Generalized Solutions and Robust Stability", in Proc. of 6th NOLCOS, Stuttgart, Germany, September 2004.
- [15] E.D. Sontag, "A 'universal' Construction of Artstein's Theorem on Nonlinear Stabilization", SCL, vol. 13, pp. 117-123, 1989.
- [16] E.D. Sontag and Y. Wang, "Output-to-state Stability and Detectability of Nonlinear Systems", SCL, vol. 29, pp. 279-290, 1997.
- [17] D.S. Laila and D. Nesic, "Changing supply rates for input-output to state stable discrete-time nonlinear systems with applications", Automatica, vol. 39, pp. 821-835, 2001.
- [18] C. Prieur and L. Praly, "A Tentative Direct Design of Output Feedbacks", in Proc. of 6th NOLCOS, Stuttgart, Germany, September 2004.
- [19] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, "Linear Matrix Inequalities in System and Control Theory", Studies in Applied Mathematics, Vol. 15, SIAM, 1994.
- [20] A. Hassibi, J. How, and S. Boyd, "A Path-following Method for Solving BMI Problems in Control", in Proc. of ACC, San Diego, California, 1999.