Quadratic Stabilization of Linear Networked Control Systems via Simultaneous Protocol and Controller Design ⋆

Dragan B. Dačić [a], Dragan Nešić [a],

[a] *Authors are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010 Australia.*

Abstract

We develop necessary and sufficient conditions for quadratic stabilizability of linear networked control systems by dynamic output feedback and communication protocols. 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 and plant states but only of the discrepancies between current and the most recently transmitted values of nodes' signals, and are implementable on control area networks. We demonstrate on a batch reactor example that our design guarantees quadratic stability with a significantly smaller network bandwidth than previously available designs.

Key words: Networked control systems, Quadratic stability, Protocol design, Partial state feedback stabilization

1 Introduction

In networked control systems (NCSs), a controller and spatially distributed sensors/actuators are grouped into network nodes and communicate by exchanging packet-based messages via a network. NCSs have several advantages over the classical control systems, such as reduced installation and maintenance costs, and are thus of large practical interest. However, NCSs require novel control designs to account for effects of the network's presence in the closed-loop. For example, depending on the level of congestion and noise, packets transmitted over the network are subject to variable delay and/or may be lost during a transmission, see for example [1] and references therein. Second, the network introduces quantization errors in signals transmitted over it due to a finite length of packets, see [2] and references therein. Finally, the network induces communication constraints because only one node per transmission is allowed to transmit its packet. We focus on this aspect of NCSs in this paper.

The communication constraints raise the issue of communication *scheduling* among the nodes. Algorithms for communication scheduling are referred to as *protocols*, and are divided in the literature into static and dynamic protocols. In a static protocol, such as round robin (RR), the network transmissions are periodically assigned to nodes in some prefixed order, see for example [3]-[5]. In contrast, dynamic protocols assign the transmissions to nodes based on the values of system states. One such protocol is *Try-Once-Discard* (TOD) protocol [6]-[9] which assigns a transmission to the node with the largest weighted discrepancy between the current and the most recently transmitted value of node's signals. Throughout of this paper we call these discrepancies the *network induced errors.*

We follow the approach to modelling of NCSs under the communication constraints introduced by Walsh et al. [6]-[9]. They assume that a stabilizing controller for the plant in absence of the communication constraints is known, and treat the constraints as a perturbation of the closed-loop system. It is shown that stability of the closed-loop system is preserved in presence of the communication constraints if the transmission interval (TI) within which the network successfully transmits packets is sufficiently small. The longest such TI that ensures stability is called the maximum allowable transmission interval (MATI), and it is inversely proportional to the network bandwidth. Inspired by the sampled-data literature, we refer to this approach to control design for NCSs as the *emulation* approach. The emulation approach is applicable to linear [6] and nonlinear [7] plants, as well as, to different protocols. Moreover, it enables independent controller and protocol synthesis, thus reducing design

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

Email addresses: d.dacic@ee.unimelb.edu.au (Dragan B. Dačić), d.nesic@ee.unimelb.edu.au (Dragan Nešić).

complexity. Its possible drawback is that the obtained MATI estimates may be small, that is, the estimated network bandwidth to ensure stability may be large.

Significant improvement of MATI estimates via the emulation approach is achieved in [10]-[12]. A class of GAS protocols is introduced in [10]-[11] and it is shown that both RR and TOD protocols belong to this class. A novel model of NCSs is then developed and used to prove that if the closed-loop system without the communication constraints is $\mathcal{L}_p$ stable with respect to disturbances, then $\mathcal{L}_p$ stability is preserved with the communication constraints for sufficiently small MATIs. MATI bounds that guarantee stability of NCSs in [6]-[12] are computed by analyzing a family of closed-loop systems parameterized by MATI. It is analytically shown on an example in [10]-[11] that for the same controller TOD protocols lead to larger MATI bounds than RR protocols. Moreover, it was observed both in experiments and simulations that (for the same controller and TI) TOD protocols provide better performance than RR protocols [6]-[9].

We depart from the emulation approach, and for a given linear time-invariant (LTI) plant and *fixed* TI derive necessary and sufficient conditions for existence of a LTI dynamic output feedback controller and a protocol that ensure quadratic stability (QS) of the resulting LNCS. Using our conditions we provide an algorithm for *simultaneous* design of controllers and protocols. The obtained protocols belong to the class of TOD protocols implying that a LNCS is quadratically stabilizable for a given LTI controller if and only if for this controller there exist weights such that the corresponding TOD protocol renders it QS. Our approach further motivates use of TOD protocols, and provides tools for computing weights in TOD protocols as a function of plant parameters and TI.

In our design we only consider protocols that depend on the network induced errors, but not on the controller and/or plant states. With such information constraints we ensure that the obtained protocols are implementable on control area networks (CANs) [6]-[9]. These constraints represent the key difference between our and other methods for design of dynamic protocols in the literature [13]-[16]. To address them we introduce the concept of *weak partial state control Lyapunov functions* (WPSCLFs) that can be utilized in other nonlinear control problems with partial state information.

In Section 2 we derive a model of linear NCSs (LNCSs). The notion of WPSCLFs is introduced in Section 3 and used in Section 4 to derive the necessary and sufficient conditions for quadratic stabilizability of LNCSs. Based on these conditions in Section 5 we develop an algorithm for simultaneous controller-protocol design, and consider the implementation issues of the designed protocols in Section 6. We illustrate our design in Section 7 on a linearized model of a batch reactor and compare the obtained MATI bounds with MATI bounds in [6,10]. Concluding remarks are given in Section 8.

2 NCS Model

We consider LNCSs that consist of a *continuous*-time (CT) LTI plant and a *discrete*-time (DT) LTI controller interconnected via a network. We follow the approach in [10] and model a LNCS as a hybrid system with jumps. We decompose it into a continuous subsystem that governs the evolution between the network transmissions and a jump subsystem that governs the jumps in its states due to the transmissions. To focus on the effects of communication scheduling, we assume that the transmissions are instantaneous, the communication channel is noiseless, and that there are no dropouts. Moreover, we let the transmissions occur at equidistant instants, $t_i \triangleq (i-1)T$, in which case the constant T can be interpreted as TI. We use two-dimensional time arguments (t,i) in our LNCS model, where the first argument denotes continuous time $t \geq 0$, while the second argument denotes the total number of jumps, $i \in \mathbb{N}$. For more details on this formalism see [17].

The continuous and jump subsystems are given by

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

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

respectively, where $y = C_p x_p$ and $u = C_c x_c + D_c \hat{y}$. The fifth line in (1) and (2) represents the timer whose variable τ determines which subsystem is currently active by measuring time since the last transmission. While $\tau \leq T$ continuous subsystem (1) is active, while when $\tau \geq T$ the next transmission occurs, that is, jump subsystem (2) is active.

The first lines in (1) and (2) describe respectively continuous and jump evolution of the plant, where $x_p \in \mathbb{R}_p^n$ are its states, $y \in \mathbb{R}^m$ is its measured output, and $\hat{u} \in \mathbb{R}^r$ is its control input. The plant is equipped with n_y sensors and n_u actuators grouped into respectively l_y and l_u nodes, $l_y \leq n_y$, $l_u \leq n_u$, $l \triangleq l_y + l_u$, based on their physical proximity. For simplicity we assume that node $j \in \{1,\ldots,l\} \triangleq \mathbb{N}_l$ contains either sensors or actuators but not both, and 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 governing 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 input $\hat{y}$ and output u are partitioned analogously to the partition of plant output y and input $\hat{u}$, $\hat{y} = [\hat{y}_1^T \dots \hat{y}_{l_y}^T]^T$, $\hat{y}_j \in \mathbb{R}^{m_j}$, $u = [u_1^T \dots u_{l_u}^T]^T$, $u_j \in \mathbb{R}^{r_j}$. We assume that the number of controller states n_c is given and consider controllers L 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 the sensors and actuators, and the controller is via the network. At each transmission we assume that *exactly one* node sends/receives the current value of its signals. The communication constraints induce a mismatch between plant's and controller's inputs and outputs which is governed by the third and forth lines in (1) and (2). Namely, $\hat{y}$ and $\hat{u}$ represent the most recent versions of plant and controller output, y and u respectively, transmitted via the network. The decision which node is to transmit is specified by the diagonal matrix $\Pi \triangleq \mathrm{diag}\{\Pi_y, \Pi_u\}$ which belongs to the set $\Pi \in \mathcal{P} \triangleq \{\Pi_1, \dots, \Pi_l\}$, where

$$\begin{aligned}\Pi_j &= \mathrm{diag}\{\Pi_{yj}, \Pi_{uj}\}\\ &= \mathrm{diag}\{(1-\delta(j-1))I_{m_1}, \dots, (1-\delta(j-l))I_{r_{l_u}}\},\end{aligned}$$

and $\delta : \mathbb{N} \cup \{0\} \to \{0, 1\}$ is Kronecker delta function. Suppose that at $(t_i, i-1)$ node $j^\star \in \mathbb{N}_{l_y}$ is assigned the i^{th} transmission. Then $\Pi = \Pi_{j^\star}$ and controller input component $\hat{y}_{j^\star}(t_i, i)$ is set to the current value of plant output component $y_{j^\star}(t_i, i-1)$, while the other controller/plant input components are kept constant. Analogously, if $j^\star \in \{l_y + 1, \dots, l\}$, then plant input component $\hat{u}_{j^\star}(t_i, i)$ is set to the current value of controller output component $u_{j^\star}(t_i, i-1)$.

By substituting the network induced errors

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

into LNCS (1)-(2), we derive the network-induced DT model that 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)

$$\begin{bmatrix} x^+ \\ e^+ \end{bmatrix} = \underbrace{\begin{bmatrix} A_T & B_T \\ C_T(I - A_T) & I - C_T B_T \end{bmatrix}}_{\hat{A}} \underbrace{\begin{bmatrix} I & 0 \\ 0 & \tilde{\Pi} \end{bmatrix}}_{\hat{\Pi}} \begin{bmatrix} x \\ e \end{bmatrix}, \tag{4}$$

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

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

While the plant matrices A_d, B_d, C_d are known, the controller matrices A_c, B_c, C_c, D_c are assumed to be unknown (unless the opposite is explicitly stated) and are to be designed. We treat the matrix $\Pi = \mathrm{diag}\{\Pi_y, \Pi_u\}$ as a control input and refer to an algorithm based on which it takes values in the set $\mathcal{P}$ as a *protocol*. We only consider protocols that depend on the network induced errors e, $\Pi : \mathbb{R}^{r+m} \to \mathcal{P}$.

Remark 1 Solutions of LNCS (4) and LNCS (1)-(2) from the same initial conditions $X(0)$ do not necessarily coincide at instants $(t_i, i-1)$, $i \in \mathbb{N}$, due to a possible initial discrepancy. Namely, the solutions of LNCS (4) first jump at $(t_1, 0)$, which is described by multiplication with matrix $\hat{\Pi}$, and then flow during the interval $[t_1, t_2]$, which is described by multiplication with matrix $\hat{A}$. In contrast, if $\tau(0) < T$ the solutions of LNCS (1)-(2) first flow according to (1) and then jump according to (2). However, stability of LNCS (1)-(2) is uniquely determined by stability of DT LNCS (4), see [18]. □

3 Partial State Control Lyapunov Functions

We introduce *weak partial state control Lyapunov functions* (WPSCLFs) which represent an extension of the control Lyapunov functions [19]-[20] to the case of partial state feedback stabilization and demonstrate their usefulness on protocol synthesis for NCSs. The motivation to use WPSCLFs for protocol synthesis stems from the fact that a protocol is implementable on CANs if it does not depend on plant or controller states, but only on the network induced errors [6]-[9].

We consider a DT system

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

where the set of input constraints $\mathcal{U}$ and C^0 functions $f_1, f_2 : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathcal{U} \to \mathbb{R}^{n_1}$ are such that origin $x_e \triangleq [0\ 0]$ 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 $f_2(0, 0, u^0) = 0$.

Definition 1 A C^0 function $V : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^+$ 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

- $\forall x,\ \alpha_1^V(\|x\|) \le V(x_1,x_2) \le \alpha_2^V(\|x\|)$,
- $\forall x_2, \exists u(x_2) \in \mathcal{U}$, such that $\sup_{x_1} \Delta V(x_1,x_2,u(x_2)) \le -\alpha_3^V(\|x_2\|)$, where $\Delta V(x_1,x_2,u) \triangleq V(x_1^+,x_2^+) - V(x_1,x_2)$. ■

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

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

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

$$\sigma(x_2) = \arg\min_u \sup_{x_1} \Delta V(x_1,x_2,u). \quad (7)$$

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

Example 1 Consider the second order system

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

which is neither stabilizable nor detectable from x_2. Thus, there does not exist a feedback law that renders x_e GAS. However, a function $V(x_1,x_2) = x_1^2 + x_2^2$ is a WPSCLF for system (8), since

$$\Delta V(x_1,x_2,u) = (2x_2+u)^2 - x_2^2,$$
$$\inf_u \sup_{x_1} \Delta V(x_1,x_2,u) = -x_2^2.$$

We note that partial state feedback law (7) derived from $V(x_1,x_2)$ is $\sigma(x_2) = -2x_2$ 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 that renders origin x_e GAS for system (5). Instead of imposing this detectability property directly on system (5), we require 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 function $W : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^+$ 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 $\gamma : \mathbb{R}^{n_2} \times \mathcal{U} \to \mathbb{R}$, $\gamma(0,u^0) = 0$, such that $\forall x \in \mathbb{R}^{n_1+n_2}$ and $\forall u \in \mathcal{U}$

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

Lemma 1 *Suppose that*

(1) system (5) possesses a WPSCLF V wrt states x_2,
(2) system (5) possesses a DF W wrt states x_1,
(3) there exists a function $\tilde{\gamma} \in \mathcal{K}^\infty$ such that feedback law (6) satisfies $\gamma(x_2,\sigma(x_2)) \le \tilde{\gamma}(\|x_2\|)$,
(4) $\limsup_{\|x_2\|\to\infty} \frac{\tilde{\gamma}(\|x_2\|)}{\alpha_3^V(\|x_2\|)} < \infty$.

Then there exists a Lyapunov function $\tilde{V} : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^+$ and functions $\alpha_1^{\tilde{V}}, \alpha_2^{\tilde{V}}, \alpha_3^{\tilde{V}} \in \mathcal{K}^\infty$ such that $\forall x$,

I. $\alpha_1^{\tilde{V}}(\|x\|) \le \tilde{V}(x_1,x_2) \le \alpha_2^{\tilde{V}}(\|x\|)$,
II. $\Delta\tilde{V}(x_1,x_2,\sigma(x_2)) \le -\alpha_3^{\tilde{V}}(\|x\|)$.

Consequently, feedback law (6) renders origin x_e GAS for system (5). Moreover, 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 2 and 3 imply output-to-state stability (OSS) of system (5)-(6), [21]. Then combining OSS, the conditions 1 and 4, and Corollary 5.1 in [22], the claim of Lemma 1 follows. If the appropriate infimum is achieved, Lemma 1 is also true for feedback law (7) under the same conditions. ■

Remark 2 Feedback laws (6) and (7) may not be continuous due to taking the supremum over a noncompact set $\mathbb{R}^{n_1}$ and possibly finite cardinality of the set $\mathcal{U}$. Such continuity is typically assumed in similar situations [23], but instead we require a weaker property in the condition 3 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 3 We recently become aware that the notion of output CLFs, closely related to the notion of WPSCLFs, was introduced in [23] and used to obtain the necessary conditions for output feedback stabilization of continuous time nonlinear systems affine in control. □

4 Conditions for Protocol Quadratic Stabilizability of LNCSs

We now utilize WPSCLFs for design of protocols solely dependent on the network induced errors. We consider LTI plants and controllers, and are interested in QS of the resulting LNCS. 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 QS of LNCS (4). Since WPSCLFs in this case achieve inf-sup in Definition 1, they are thus replaced with min-max.

Definition 3 LNCS (4) is quadratically stable (QS) if for a given controller $L \in \mathcal{L}_{n_c}$ and protocol $\Pi = \Omega(e)$,

$\Omega : \mathbb{R}^{m+r} \to \mathcal{P}$, there exists a matrix $P = P^T > 0$ such that the Lyapunov function

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

satisfies $\forall X = [x^T\ e^T]^T \neq [0\ 0]^T$

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

where

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

LNCS (4) is *protocol quadratically stabilizable* (pQS) if for a given controller $L \in \mathcal{L}_{n_c}$ there exists a function $\Omega : \mathbb{R}^{m+r} \to \mathcal{P}$, such that LNCS (4) with 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 *LNCS (4) is pQS for a given controller $L \in \mathcal{L}_{n_c}$ if and only if (iff) there exists a matrix $P = P^T > 0$ such that corresponding quadratic function (9) is a WPSCLF. Then the protocol*

$$\Omega^\star(e) = \arg \min_{\Pi} \max_{x} \Delta V(x,e,\Pi), \tag{11}$$

renders LNCS (4) QS.

Proof: Suppose that LNCS (4) is pQS. Then there exist a matrix $P = P^T > 0$ and a protocol $\Omega : \mathbb{R}^{m+r} \to \mathcal{P}$ such that function (9) satisfies condition (10) for all $X = [x^T\ e^T]^T \neq 0$. Since $\Delta V(x,e,\Omega(e))$ is a negative quadratic function of states x it has a unique maximum in x. Then $\Delta V(x,e,\Omega(e)) < 0$ holds $\forall X \neq 0$ iff $\max_x \Delta V(x,e,\Omega(e)) < 0$. Consequently, Lyapunov function (9) is a WPSCLF for LNCS (4).

Suppose that for the matrix P function (9) is a WPSCLF for LNCS (4). We show that in this case the condition 1 of Lemma 1 implies the conditions 2-4 of Lemma 1. We use two observations. First, quadratic function (9) is a WPSCLF for LNCS (4) iff the quadratic function

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

is a WPSCLF as well, since $\max_x \Delta V(x,e,\Pi) = \max_x \Delta\overline{V}(x,e,\Pi)$. 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 get a Lyapunov inequality

$$A_T^T \hat{P}_{11} A_T - \hat{P}_{11} < -Z \leq 0, \tag{13}$$

where $Z \triangleq (I - A_T)^T C_T^T \hat{P}_{22}^T C_T (I - A_T)$, 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 (9). 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

$$\Delta W(x,e,\Pi) \leq -\frac{1}{2} z_m \|X\|^2 + (\tilde{\Pi} e)^T G \tilde{\Pi} e,$$

for suitably defined constant $z_m > 0$ and matrix $G = G^T > 0$. Then the condition 2 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 3 holds independently of the protocol, $\gamma(e,\Pi) \leq \tilde{\gamma}(\|e\|) \triangleq g^\star \|e\|^2$, $g^\star \triangleq \max_{\Pi \in \mathcal{P}} \|\tilde{\Pi}^T G \tilde{\Pi}\|$. The condition 4 is automatically satisfied because all bounding functions are quadratic. Hence, protocol (11), which corresponds to feedback law (7) in this case, renders LNCS (4) QS, that is, LNCS (4) is pQS. ■

Remark 4 Given a controller $L \in \mathcal{L}_{n_c}$, it follows from Theorem 1 that the necessary and sufficient condition for pQS of LNCS (4) is the existence of a matrix $P = P^T > 0$ for which corresponding function (9) and TOD protocol (11) along solutions of LNCS (4) satisfy

$$\forall e \neq 0,\ \max_x \Delta V(x,e,\Omega^\star(e)) < 0. \tag{14}$$

Condition (14) can thus be viewed as a characterization of WPSCLFs induced by their parametrization via the matrix P. Note that the class of TOD protocols (11) is natural when the goal is QS of LNCS (4). Namely, if QS can not be achieved via a protocol from this class then no other protocol independent of plant and controller states (such as a RR protocol) can achieve QS. □

Inequality (13) implies that LNCS (4) can be rendered QS by protocol (11) only if the controller L renders the plant QS without the network, that is, only if the matrix A_T is Schur. This is a consequence of constraining protocol (11) to depend only on the network induced errors.

Corollary 1 *Let a controller $L \in \mathcal{L}_{n_c}$ in LNCS (4) be given. Then LNCS (4) is pQS only if the matrix A_T is Schur.*

Remark 5 Synthesis of protocols by utilizing CLFs has already been proposed in [15]. In [15], 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 due to the communication constraints. □

Computation of a matrix P for which corresponding function (9) is a WPSCLF for LNCS (4), that is, which satisfies (14) is not tractable since it involves pointwise minimization of l quadratic functions dependent on the matrix P, see (11). We avoid this pointwise minimization by introducing a more flexible parametrization of WPSCLFs that leads to a more tractable computation. To that end we introduce sets $\mathcal{S}_j$, $j \in \mathbb{N}_l$,

$$\begin{aligned}
&\mathcal{S}_j \triangleq \{e : e^T(\hat{H}(\Pi_j,\mathbb{S}) - \hat{H}(\Pi_k,\mathbb{S}))e \le 0, \forall k \in \mathbb{N}_l\},\\
&\hat{H}(\Pi,\mathbb{S}) \triangleq \begin{bmatrix} H_{11}(\Pi,\mathbb{S}) & H_{12}(\Pi,\mathbb{S}) \\ H_{12}^T(\Pi,\mathbb{S}) & H_{22}(\Pi,\mathbb{S}) \end{bmatrix},\\
&H_{11}(\Pi,\mathbb{S}) \triangleq \Pi_y S_1 \Pi_y + \Pi_y S_2 + S_2^T \Pi_y,\\
&H_{12}(\Pi,\mathbb{S}) \triangleq \Pi_y S_3^T \Pi_u + \Pi_y S_4^T + S_5^T \Pi_u,\\
&H_{22}(\Pi,\mathbb{S}) \triangleq \Pi_u S_6 \Pi_u + \Pi_u S_7 + S_7^T \Pi_u,
\end{aligned} \tag{15}$$

parameterized by a collection of matrices $\mathbb{S} \in \mathcal{M} \triangleq \{\mathbb{S} = (S_1,\ldots,S_7) : S_1, S_2 \in \mathbb{R}^{m\times m}, S_1 = S_1^T, S_3, S_4, S_5 \in \mathbb{R}^{m\times r}, S_6, S_7 \in \mathbb{R}^{r\times r}, S_6 = S_6^T\}$. Note that sets defined in (15) satisfy $\cup_{j=1}^{l}\mathcal{S}_j = \mathbb{R}^{m+r}$.

Theorem 2 *LNCS (4) is a pQS for a given controller $L \in \mathcal{L}_{n_c}$ iff there exist a matrix $P = P^T > 0$ and a collection of matrices $\mathbb{S} \in \mathcal{M}$ such that $\forall j \in \mathbb{N}_l$ quadratic function (9) and sets (15) satisfy*

$$\forall e \in \mathcal{S}_j,\ e \ne 0,\ \max_x \Delta V(x,e,\Pi_j) < 0. \tag{16}$$

Then the protocol

$$\Omega^{\mathbb{S}}(e) = \arg\min_{\Pi} e^T \hat{H}(\Pi,\mathbb{S})e \tag{17}$$

renders LNCS (4) QS.

Proof: The sufficiency is a straightforward consequence of Definition 3 and $\cup_{j=1}^{l}\mathcal{S}_j = \mathbb{R}^{m+r}$, and we only show the necessity. Suppose that LNCS (4) is pQS. From Theorem 1 it follows that LNCS (4) possesses a quadratic WPSCLF (9) defined by a matrix P, and protocol (11) renders LNCS (4) QS, that is, condition (14) holds.

Let $\mathcal{S}_j^\star \triangleq \{e : \max_x \Delta V(x,e,\Pi_j) \le \max_x \Delta V(x,e,\Pi_k)$, $\forall k \in \mathbb{N}_l\}$, $j \in \mathbb{N}_l$, be the set of network induced errors for which the minimum of $\Pi \in \mathcal{P}$ in (11) is equal to Π_j. By maximizing function $\Delta V(x,e,\Pi)$ wrt to x in (10), these sets can be written as $\mathcal{S}_j^\star = \{e : e^T(\hat{H}(\Pi_j,\mathbb{S}^\star) - \hat{H}(\Pi_k,\mathbb{S}^\star))e \le 0, \forall k \in \mathbb{N}_l\}$, where explicit dependence of the matrix collection $\mathbb{S}^\star$ in (15) on the plant matrices A_d, B_d, C_d, the matrix P and the controller L is given in the Appendix. Using sets $\mathcal{S}_j^\star$ protocol (11) can be represented by $\Omega^\star(e) = \Pi_{j^\star}$, for $e \in \mathcal{S}_j^\star$. Combining this expression with (14) we deduce that condition (16) holds for the matrix P and the collection $\mathbb{S}^\star$.

Note that protocol (17) renders LNCS (4) QS since $\forall j \in \mathbb{N}_l$ and $\forall e \in \mathcal{S}_j$ it holds that $\max_x \Delta V(x,e,\Pi_j) = \max_x \Delta V(x,e,\Omega^{\mathbb{S}}(e)) < 0$. ■

We note that intersection of sets $\mathcal{S}_j$ may not be empty, since they may have a common boundary. For $e \in \mathcal{S}_j \cap \mathcal{S}_k$, $j \ne k$, protocol (17) does not uniquely assign a transmission. We avoid this ambiguity by assigning the transmission to the node with the smallest identifier $j^\star$ whose set $\mathcal{S}_{j^\star}$ contains the current network induced error e, $\Omega^{\mathbb{S}}(e) = \Pi_{j^\star}$ where $j^\star = \min\{j : e \in \mathcal{S}_j\}$. The same issue arises for all protocols designed here and they are always implemented in conjunction with this rule, even if this is not explicitly stated.

5 Conditions for pcQS of LNCSs

Using Theorem 2 in this Section we derive sufficient conditions for pcQS of LNCS (4) in terms of matrix inequalities (MIs).

Corollary 2 *LNCS (4) is pcQS if there exist a matrix $P = P^T > 0$, a controller $L \in \mathcal{L}_{n_c}$, a collection of matrices $\mathbb{S} \in \mathcal{M}$, and parameters $\tau_{jk} \ge 0$, $j,k \in \mathbb{N}_l$, such that $\forall j \in \mathbb{N}_l$*

$$\begin{bmatrix} P + \sum_{k=1}^{l} \tau_{jk}\left(\tilde{H}(\Pi_j,\mathbb{S}) - \tilde{H}(\Pi_k,\mathbb{S})\right) & \star \\ P\hat{A}_j & P \end{bmatrix} > 0, \tag{18}$$

where $\tilde{H}(\Pi_j,\mathbb{S}) \triangleq \mathrm{diag}\{0, \hat{H}(\Pi_j,\mathbb{S})\}$, and $\hat{A}_j \triangleq \hat{A}\hat{\Pi}|_{\Pi=\Pi_j}$. The resulting protocol is then given by (17).

Proof: From Theorem 2 and Definition 3 we conclude that LNCS (4) is pcQS iff there exists a controller $L \in \mathcal{L}_{n_c}$, a matrix $P = P^T > 0$, and a collection $\mathbb{S} \in \mathcal{M}$ for which condition (16) holds. Applying then the $\mathcal{S}$-procedure [24] on (16), we get that existence of parameters $\tau_{jk} \ge 0$, $j,k \in \mathbb{N}_l$, such that

$$\begin{aligned}
&\max_x \Delta V(x,e,\Pi_j) - \\
&\qquad \sum_{k=1}^{l} \tau_{jk} e^T\left(\hat{H}(\Pi_j,\mathbb{S}) - \hat{H}(\Pi_k,\mathbb{S})\right)e < 0,
\end{aligned} \tag{19}$$

holds for $\forall j \in \mathbb{N}_l$, $\forall x \ne 0$, and $\forall e \ne 0$, implies pcQS of LNCS (4). Suppressing the dependence on e, and applying the Schur complement [24], we get

$$
\begin{aligned}
&e^T(\hat{Q}_{12}\tilde{\Pi}_j - \hat{P}_{12})^T(\hat{P}_{11} - \hat{Q}_{11})^{-1}(\hat{Q}_{12}\tilde{\Pi}_j - \hat{P}_{12})e+\\
&e^T(\tilde{\Pi}_j^T\hat{Q}_{22}\tilde{\Pi}_j - \hat{P}_{22})e - \textstyle\sum_{k=1}^{l}\tau_{jk}e^T\Big(\hat{H}(\Pi_j,\mathbb{S})-\\
&\hat{H}(\Pi_k,\mathbb{S})\Big)e < 0 \;\Leftrightarrow\\
&\hat{\Pi}_j^T Q\hat{\Pi}_j - P - \textstyle\sum_{k=1}^{l}\tau_{jk}\Big(\tilde{H}(\Pi_j,\mathbb{S}) - \tilde{H}(\Pi_k,\mathbb{S})\Big) =\\
&\hat{A}_j^T P\hat{A}_j - P - \textstyle\sum_{k=1}^{l}\tau_{jk}\Big(\tilde{H}(\Pi_j,\mathbb{S}) - \tilde{H}(\Pi_k,\mathbb{S})\Big) < 0.
\end{aligned}
$$

Applying again the Schur complement on the above inequality, the claim of Corollary 2 follows. We note that application of the $\mathcal{S}$-procedure leads to the loss of necessity of condition (18) for pcQS of LNCS (4), except in the special case when $l = 2$. ■

We now show that there is no loss of generality in restricting the controller L to be in the observable canonical form, that is, $L \in \tilde{\mathcal{L}}_{n_c} \triangleq \{\tilde{L} \in \mathcal{L}_{n_c} : C_c = [I_r\ 0]\}$. Note that in that case matrices $\hat{A}_j$, $j \in \mathbb{N}_l$, are linear functions of unknown controller matrices A_c, B_c, D_c.

Corollary 3 *Let $n_c \geq r$. Then there exist a matrix $P = P^T > 0$, controller $L \in \mathcal{L}_{n_c}$, a collection $\mathbb{S} \in \mathcal{M}$, and parameters $\tau_{jk} \geq 0$, $k, j \in \mathbb{N}_l$ for which MI (18) holds iff for the same collection $\mathbb{S}$ and parameters τ_{jk}, there exist a matrix $P_T = P_T^T > 0$ and a controller $\tilde{L} \in \tilde{\mathcal{L}}_{n_c}$ for which MI (18) holds.*

Proof: We define $\hat{T} \triangleq \mathrm{diag}\{I_{n_p}, T, I_{m+r}\}$, where the matrix $T \in \mathbb{R}^{n_c \times n_c}$ is nonsingular. Multiplying MI (18) from the right with the nonsingular matrix $\mathrm{diag}\{\hat{T}, \hat{T}\}$, and from the left with its transpose, we get that it is equivalent to the following inequality

$$
\begin{bmatrix} \hat{T}^T(P + \sum_k \tau_{jk}(\tilde{H}(\Pi_j,\mathbb{S}) - \tilde{H}(\Pi_k,\mathbb{S})))\hat{T} & \star \\ \hat{T}^T P\hat{A}_j\hat{T} & \hat{T}^T P\hat{T} \end{bmatrix} =
$$
$$
\begin{bmatrix} P_T + \sum_k \tau_{jk}(\tilde{H}(\Pi_j,\mathbb{S}) - \tilde{H}(\Pi_k,\mathbb{S})) & \star \\ P_T\hat{T}^{-1}\hat{A}_j\hat{T} & P_T \end{bmatrix} > 0
$$

where $P_T \triangleq \hat{T}^T P\hat{T}$, since $\hat{T}^T\tilde{H}(\Pi_j,\mathbb{S})\hat{T} = \tilde{H}(\Pi_j,\mathbb{S})$. The matrix $\hat{A}_{Tj} \triangleq \hat{T}^{-1}\hat{A}_j\hat{T}$ is equal to

$$
\hat{A}_{Tj} = \begin{bmatrix}
A_d + B_dD_cC_d & B_d\tilde{C}_c & B_dD_c & B_d\Pi_{uj} \\
\tilde{B}_cC_d & \tilde{A}_c & \tilde{B}_d\Pi_{yj} & 0 \\
C_d(I - A_d - B_dD_cC_d) & -C_dB_d\tilde{C}_c & \Pi_{yj} - C_dB_dD_c & -C_dB_d\Pi_{uj} \\
-\tilde{C}_c\tilde{B}_cC_d & \tilde{C}_c(I - \tilde{A}_c) & D_c(I - \Pi_{yj}) - \tilde{C}_c\tilde{B}_c\Pi_{yj} & \Pi_{uj}
\end{bmatrix}
$$

where $\tilde{A}_c \triangleq T^{-1}A_cT$, $\tilde{B}_c \triangleq T^{-1}B_c$ and $\tilde{C}_c = C_cT$. Thus, there exist a matrix P, a controller $L = (A_c, B_c, C_c, D_c)$, a collection $\mathbb{S}$, and parameters τ_{jk} for which MI (18) holds, iff MI (18) holds for the matrix P_T, the controller $\tilde{L} = (T^{-1}A_cT, T^{-1}B_c, C_cT, D_c) = (\tilde{A}_c, \tilde{B}_c, \tilde{C}_c, D_c)$, and the same collection $\mathbb{S}$ and parameters τ_{jk}. By selecting the matrix T to satisfy $\tilde{C}_c = C_cT = [I_r\ 0]$, we have that $\tilde{L} \in \tilde{\mathcal{L}}_{n_c}$ and the claim of Corollary 3 follows. We note that such matrix T exists if the matrix C_c has full row rank, which is only possible if the number of controller states n_c is not smaller then the number of its outputs r, $n_c \geq r$. However, our results apply to the static output feedback case, $n_c = 0$, in which case the second row and column in the matrix $\hat{A}_{Tj}$ are omitted and the matrix $\tilde{C}_c$ is set to zero. ■

Since matrices $\hat{A}_j$, $j \in \mathbb{N}_l$ are linear functions of the unknown controller matrices, A_c, B_c, D_c, sufficient condition (18) for pcQS of LNCS (4) represents a bilinear matrix inequality (BMI) in the variables $P = P^T > 0$, $\tilde{L} = (A_c, B_c, D_c) \in \tilde{\mathcal{L}}_{n_c}$, $\mathbb{S} \in \mathcal{M}$, and $\tau_{jk} \geq 0$, due to products $\tau_{jk}(\tilde{H}(\Pi_j,\mathbb{S}) - \tilde{H}(\Pi_k,\mathbb{S}))$ and $P\hat{A}_j$. There exist several algorithms in the literature for solving BMIs [25]-[27], but their computational complexity is larger than complexity of LMI algorithms. Moreover, the ability to solve BMIs often depends on the initial guess of the variables. We solve BMI (18) by modifying the algorithm in [27] that 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 number of controller states $n_c \geq r$, $0 < \delta a \ll 1$, and $0 < \epsilon \ll 1$. Set initial controller matrices to zero, $\tilde{L}^0 = (0,0,0) \in \tilde{\mathcal{L}}_{n_c}$, $P^0 = I$, $\mathbb{S}^0 = (0,\ldots,0) \in \mathcal{M}$, and $\tau_{jk}^0 = 1$. Compute $\hat{A}_j^0 \triangleq \hat{A}_j(\tilde{L}^0)$, $a^0 = \max_j \|\hat{A}_j^0\|$, and $\tilde{H}(\Pi_j, \mathbb{S}^0)$, $j, k \in \mathbb{N}_l$.

S2: Solve the LMI problem in variables $\delta P = \delta P^T$, $\delta L = (\delta A_c, \delta B_c, \delta D_c) \in \tilde{\mathcal{L}}_{n_c}$, $\delta\mathbb{S} = (\delta S_1, \ldots, \delta S_7) \in \mathcal{M}$, $\delta\tau_{jk}$, $j, k \in \mathbb{N}_l$,

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

where $b_i = \frac{1}{2}\epsilon + \kappa_i - 1$, $\kappa^i = 1 + (a^i - \delta a)^2$ for $a^i \geq \delta a$ and $\kappa^i = 1$ for $a^i < \delta$, $W_j^i \triangleq \sum_{k=1}^{l}((\tau_{jk}^i + \delta\tau_{jk})(\tilde{H}(\Pi_j, \mathbb{S}^i) -$

$\tilde{H}(\Pi_k, \mathbb{S}^i)) + \tau_{jk}^i(\tilde{H}(\Pi_j, \delta\mathbb{S}) - \tilde{H}(\Pi_k, \delta\mathbb{S}))), \hat{A}_j^i \triangleq \hat{A}_j(\tilde{L}^i)$,

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

and $\delta Z_j \triangleq \delta D_c(I - \Pi_{yj}) - C_c \delta B_c \Pi_{yj}$.

S3: Stop iterating if the LMI problem in S2 is not feasible. Otherwise, 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 $\mathbb{S}^{i+1} = \mathbb{S}^i + \delta\mathbb{S}$ (where the last addition is component-wise). Set $i := i+1$. Stop if $a^i < 0$, otherwise go to S2.

Remark 6 The parameter κ^i represents a measure of instability of LNCS (4) with the controller $\tilde{L}^i$. We initially set κ^0 to the value which corresponds to instability level of uncontrolled LNCS (4) and in each iteration we update the controller to reduce this level with the ultimate goal of achieving $\kappa^i = 1$. If this is achieved we say that the LMBIA terminates successfully. Note that κ^0 is chosen such that the LMIs in S2 are feasible for $\delta\hat{A}_j = 0$.

Norms of variables in LMI problem (20) are bounded by a function of κ^i and the parameter ϵ to ensure that the linearization of original BMI (18) is valid for $\kappa^i \approx 1$. It can be shown that if the LMBIA terminates successfully, that is, if LMIs (20) are feasible for $\kappa^i = 1$ then BMIs (18) are feasible for matrix $P^i + \delta P$, controller $\tilde{L}^i + \delta\tilde{L}$, matrices $\mathbb{S}^i + \delta\mathbb{S}$, and parameters $\tau_{jk}^i + \delta\tau_{jk}$. □

Remark 7 The LMBIA is sensitive to the choice of the initial condition for the matrix P^0. Through extensive computation we observed that $P^0 = I$ performs well. For successful termination of the LBMIA it is beneficial to select small δa, but this increases the computational effort for design of controllers and protocols. □

6 Implementation Issues

Here we discuss implementation issues of protocol (17). In NCSs implemented via control area network the nodes exchange packets that consist of a header used by the network to rout a packet and a data field containing the current value of nodes' signals. At each transmission all nodes simultaneously attempt to transmit their packets. The network compares the transmission priorities, denoted by $TP(e, j)$, $j \in \mathbb{N}_l$, in packets headers', assigns the transmission to the node whose packet has the highest priority, and disables transmissions of all other nodes. Thus, protocol (17) can be implemented by manipulating transmission priorities as proposed in [6]-[9]. In so far, we assumed that each node has the knowledge of *all* networked induced errors but not of plant and/or controller states. Then protocol (17) can be implemented by setting the transmission priorities to

$$TP(e, j) = p(e) - e^T\hat{H}(\Pi_j, \mathbb{S})e, \tag{21}$$

where $p : \mathbb{R}^{m+r} \to \mathbb{R}$ is any function of the network induced errors independent of the matrix Π. Note that the node selected by $\arg\max_j TP(e, j)$ coincides with the node selected by protocol (17).

However, there are NCSs in which node j can only rely on its own network induced error e_j to compute its transmission priority, where $e = [e_1^T \ \dots \ e_l^T]^T$. Then $TP(e, j)$ defined by (21) can not be computed at node j. For such NCSs we impose a particular structure of the matrix $\hat{H}$ and select the function $p(e)$ such that $TP(e, j)$ in (21) only depends on e_j. Namely, by setting

$$\hat{H}(\Pi, \mathbb{Q}) \triangleq \Pi \operatorname{diag}\{Q_1, \dots, Q_l\}\Pi, \ p(e) \triangleq e^T\hat{H}(I, \mathbb{Q})e,$$

where $\mathbb{Q} \in \mathcal{Q} = \{(Q_1, \dots, Q_l) : Q_j = Q_j^T > 0\}$, formula (21) reduces to $TP(e, j) = e_j^T Q_j e_j$, which can be computed at node j with available signals.

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

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

To implement the resulting protocol

$$\Pi = \Pi_{j^\star}, \quad j^\star = \arg\max_{j \in \mathbb{N}_l} e_j^T Q_j e_j, \tag{22}$$

node j only requires its own network induced error e_j.

Besides a different dependance of matrix $\tilde{H}$ on protocol parameters, $\mathbb{Q} \in \mathcal{Q}$ instead of $\mathbb{S} \in \mathcal{M}$, no other changes of the LBMIA are required.

Remark 8 In some situations it may be useful to beforehand fix protocol (22), that is, matrices $\mathbb{Q}^\star \in \mathcal{Q}$. That simplifies the computation in S2 of the LBMIA, because it makes the term $\sum_k \tau_{jk}(\tilde{H}(\Pi_j, \mathbb{Q}^\star) - \tilde{H}(\Pi_k, \mathbb{Q}^\star))$ linear in the variables. Then S2 can be replaced with $\hat{\text{S}}2$.

$\hat{\text{S}}2$: Solve the LMI problem in decision variables δP, $\delta\tilde{L} \in \tilde{\mathcal{L}}_{n_c}$, τ_{jk}, $j, k \in \mathbb{N}_l$,

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

where $W_j \triangleq \sum_{k=1}^{l} \tau_{jk} \left(\tilde{H}(\Pi_j, \mathbb{Q}^\star) - \tilde{H}(\Pi_k, \mathbb{Q}^\star) \right)$. $\square$

7 Case Study: Batch Reactor

We illustrate the LMBIA on a linearized batch reactor model considered in [6,10]

$$\dot{x}_p = A_p x_p + B_p \hat{u},\ y = C_p x_p, \tag{23}$$

where $x_p \in \mathbb{R}^4$, $\hat{u}, y \in \mathbb{R}^2$, and

$$A_p = \begin{bmatrix} 1.38 & -0.208 & 6.715 & -5.676 \\ -0.581 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix},$$
$$B_p^T = \begin{bmatrix} 0 & 5.679 & 1.136 & 1.136 \\ 0 & 0 & -3.146 & 0 \end{bmatrix},\ C_p = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

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 batch reactor (23) is unstable. Our goal is to design controllers and protocols which allow that the TI be as large as possible, hence reducing the required network bandwidth. We start with $T^0 = 0.01s$, $k = 0$, and compute the discrete time model of batch reactor (23)

$$x_p^+ = A_d(T^k) x_p + B_d(T^k)\hat{u},\ \ y_p = C_p x_p, \tag{24}$$

where $A_d(T^k) \triangleq e^{A_p T^k}$ and $B_d(T^k) \triangleq \int_0^{T^k} e^{A_p t} dt B_p$. If the LBMIA successfully terminates for discrete time plant (24) 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 $T^\star$ of TI beyond which the LBMIA fails to terminate successfully. This value of TI is interpreted as a MATI bound and its existence is verified by numeric computation.

We consider two scenarios. 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 vicinity of the actuators. Then the set of protocol matrices is given by $\mathcal{P}_2 = \{\mathrm{diag}\{0,1\}, \mathrm{diag}\{1,0\}\}$. In the second scenario the network has an additional node, $l = 3$, which contains two actuators governing the control inputs. Then the set of protocol matrices is given by $\mathcal{P}_3 = \{\mathrm{diag}\{0,1,1,1\}, \mathrm{diag}\{1,0,1,1\}, \mathrm{diag}\{1,1,0,0\}\}$. The purpose of the first scenario is to compare MATI bounds obtained via the LMBIA with MATI bounds in the literature, while the purpose of the second scenario is to illustrate the influence of increasing the number of network nodes on MATI bounds.

Table 1
Batch reactor: Comparison of MATI bounds for different controllers and protocols

Controller	Protocol	l	MATI Bound [s]
(25)	RR	2	[6]: 10^{-5}
(25)	TOD	2	[6]: 10^{-5}
(25)	RR	2	[10]: 0.0082
(25)	TOD	2	[10]: 0.01
LBMIA: via S3	$\mathbb{S} \in \mathcal{M}$	2	0.75
LBMIA: via S3	$\mathbb{Q} \in \mathcal{Q}$	2	0.75
LBMIA: via Ŝ3	fixed $\mathbb{Q}^\star \in \mathcal{Q}$	2	0.81
LBMIA: via S3	$\mathbb{S} \in \mathcal{M}$	3	0.067
LBMIA: via S3	$\mathbb{Q} \in \mathcal{Q}$	3	0.061
LBMIA: via Ŝ3	fixed $\mathbb{Q}^\star \in \mathcal{Q}$	3	0.033

For comparison we use MATI bounds obtained in [6,10] using the emulation approach. (For a short description of the emulation approach and its differences with respect to our approach see Section 1.) Controller proposed in [6,10] to globally exponentially stabilize reactor (23) in absence of the communication constraints is given by

$$\dot{x}_c = A_{cc} x_c + B_{cc}\hat{y},\ u = C_{cc} x_c + D_{cc}\hat{y}, \tag{25}$$

where $A_{cc} = \mathrm{diag}\{0,0\}$, $C_{cc} = \mathrm{diag}\{-2,8\}$, and

$$B_{cc} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ D_{cc} = -\begin{bmatrix} 0 & 2 \\ -5 & 0 \end{bmatrix}.$$

With the LBMIA we compute controllers with the same number of states as controller (25), $n_c = 2$, and investigate the effect of different protocols on MATI bounds. We consider protocols obtained by: a) computing $\mathbb{S} \in \mathcal{M}$, b) computing $\mathbb{Q} \in \mathcal{Q}$, c) beforehand fixing $\mathbb{Q}^\star \in \mathcal{Q}$. For the case c) we select the parameters $\mathbb{Q}^\star = \mathrm{diag}\{1,1\}$ for $l = 2$, and $\mathbb{Q}^\star = \mathrm{diag}\{1,1,I_2\}$ for $l = 3$, that correspond to the original TOD protocol, and we modify the LBMIA by replacing S3 with Ŝ3, see Remark 9. The values of the remaining parameters in the LMBIA 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 $l = 3$. The obtained MATI bounds are summarized in Table 1.

In the first scenario the obtained MATI bounds are 75 times larger than the MATI bounds obtained in [10] and 7500 times than the MATI bounds obtained in [6]. It can be shown that in this scenario both protocol parameterizations, $\mathbb{S} \in \mathcal{M}$ and $\mathbb{Q} \in \mathcal{Q}$, are equivalent, hence the resulting MATI bounds are equal. It is however 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 Ŝ3 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 an additional network induced error. In contrast to the first scenario, the largest MATI bound here is obtained with the most general protocol parametrization $\mathbb{S} \in \mathcal{M}$.

Remark 9 We assumed in this paper that the data received from the network is buffered in each node, which is described by $\dot{\hat{y}} = 0$ and $\dot{\hat{u}} = 0$ in (1). We expect that a significant increase in MATI bounds can be achieved by using dynamic predictors for $\hat{y}$ and $\hat{u}$ in each node, so that nodes have better estimates of current values of $y(t)$ and $u(t)$. In other words, we think that there is a trade-off between the required network bandwidth to stabilize the NCS and computational capabilities of the network nodes. Similar observation was made in [28]. □

In Figs. 1-2 we show the typical closed-loop behavior of reactor (23) in the first and the second scenario respectively, for $T = 0.03s$, initial conditions $[x_p^T(0)\ x_c^T(0)]^T = [2\ 0\ 0\ 0\ 0\ 0]^T$, $\hat{u}(0) = [0\ 0]^T$, $\hat{y}(0) = [0\ 0]^T$ and controller and protocol designed via the LBMIA. Due to the space limitation we only give the controller and protocol in the first scenario for case a). The resulting controller $L = (A_c, B_c, D_c) \in \tilde{\mathcal{L}}_2$ is

$$A_c = \begin{bmatrix} -0.0346 & -0.0013 \\ -0.0014 & 0.002 \end{bmatrix},\ B_c = \begin{bmatrix} 0.2399 & -0.7612 \\ 0.0123 & 0.0202 \end{bmatrix},$$
$$D_c = \begin{bmatrix} 0.149 & -0.307 \\ 1.7847 & 1.5501 \end{bmatrix},$$

while the resulting protocol is $\Omega^{\mathbb{S}}(e) = \Pi_1$ if $e \in \mathcal{S}_1 = \{e : e^T \mathrm{diag}\{-0.959, 1.216\} e \le 0\}$, and $\Omega^{\mathbb{S}}(e) = \Pi_2$ if $e \in \mathcal{S}_2 = \{e : e^T \mathrm{diag}\{-0.959, 1.216\} e \ge 0\}$.

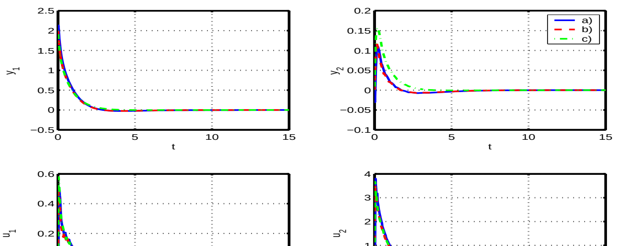

Fig. 1. Input and output trajectories of reactor (23) in the first scenario: a) $\mathbb{S} \in \mathcal{M}$, b) $\mathbb{Q} \in \mathcal{Q}$, c) fixed $\mathbb{Q}^\star \in \mathcal{Q}$.

Through our extensive simulations of batch reactor (23) with different protocols and controllers, we observed that for the same TI the 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 increasing the number of network nodes in a NCS, results in its slower response due to reduction of

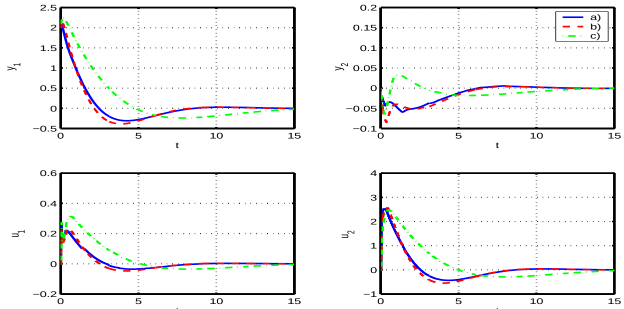

Fig. 2. Input and output trajectories of reactor (23) in the second scenario: a) $\mathbb{S} \in \mathcal{M}$, b) $\mathbb{Q} \in \mathcal{Q}$, c) fixed $\mathbb{Q}^\star \in \mathcal{Q}$.

the resulting controller gain, making it thus more sensitive to disturbances.

8 Conclusion

In this paper we considered LNCSs in which the network transmissions are assigned via TOD protocols that depend on the network induced errors but not on controller and/or plant states. We derived the necessary and sufficient conditions for *protocol quadratic stabilizability* of LNCSs via such protocols by utilizing *weak partial state control Lyapunov functions*. Using the obtained conditions we provided sufficient conditions for *protocol and controller quadratic stabilizability* of LNCSs in terms of BMIs. Applying the path-following method for solving BMIs, we developed a computationally tractable algorithm for simultaneous controller and protocol design that relies on solving a sequence of LMIs. Finally, we illustrated our design on a linearized model of a batch reactor and showed that the designed controllers and protocols guarantee QS with significantly smaller network bandwidth than required with previous designs.

9 Appendix

Here we give dependence of the collection of matrices $\mathbb{S}^\star = (S_1^\star, \dots, S_7^\star)$ on the plant matrices A_d, B_d, C_d, controller matrices $L = (A_c, B_c, C_c, D_c) \in \mathcal{L}_{n_c}$ and the matrix P used in the proof of Theorem 2. We first refine the partitions of matrices P and $Q = \hat{A}^T P \hat{A}$ by introducing

$$\hat{P}_{11} \triangleq P_{11},\ \ \hat{P}_{12}^T \triangleq \begin{bmatrix} P_{12}^T \\ P_{13}^T \end{bmatrix},\ \hat{P}_{22} \triangleq \begin{bmatrix} P_{22} & P_{23} \\ P_{23}^T & P_{33} \end{bmatrix},$$
$$\hat{Q}_{11} \triangleq Q_{11},\ \hat{Q}_{12}^T \triangleq \begin{bmatrix} Q_{12}^T \\ Q_{13}^T \end{bmatrix},\ \hat{Q}_{22} \triangleq \begin{bmatrix} Q_{22} & Q_{23} \\ Q_{23}^T & Q_{33} \end{bmatrix},$$

where $P_{11}, Q_{11} \in \mathbb{R}^{n\times n}$, $P_{12}, Q_{12} \in \mathbb{R}^{n\times m}$, $P_{13}, Q_{13} \in \mathbb{R}^{n\times r}$, $P_{22}, Q_{22} \in \mathbb{R}^{m\times m}$, $P_{23}, Q_{23} \in \mathbb{R}^{m\times r}$, $P_{33}, Q_{33} \in \mathbb{R}^{r\times r}$, and $n \triangleq n_p + n_c$. Then the matrices $S_j^\star$, $j \in \mathbb{N}_7$, are given by

$$S_1^\star = Q_{12}^T (P_{11} - Q_{11})^{-1} Q_{12} - Q_{12}^T (P_{11} - Q_{11})^{-1} Q_{13} D_c$$

$$
\begin{aligned}
&- D_c^T Q_{13}^T (P_{11} - Q_{11})^{-1} Q_{12} - D_c^T Q_{23} - Q_{23} D_c \\
&+ D_c^T Q_{13}^T (P_{11} - Q_{11})^{-1} Q_{13} D_c + Q_{22} + D_c^T Q_{33} D_c, \\
S_2^\star &= Q_{23} D_c - D_c^T Q_{33} D_c + Q_{12}^T (P_{11} - Q_{11})^{-1} Q_{13} D_c \\
&+ D_c^T Q_{13}^T (P_{11} - Q_{11})^{-1} P_{12} - Q_{12}^T (P_{11} - Q_{11})^{-1} P_{12} \\
&- D_c^T Q_{13}^T (P_{11} - Q_{11})^{-1} Q_{13} D_c, \\
S_3^\star &= Q_{23}^T - Q_{33} D_c + Q_{13}^T (P_{11} - Q_{11})^{-1} Q_{12} \\
&- Q_{13}^T (P_{11} - Q_{11})^{-1} Q_{13} D_c, \\
S_4^\star &= P_{13}^T (P_{11} - Q_{11})^{-1} Q_{13} D_c - P_{13}^T (P_{11} - Q_{11})^{-1} Q_{12}, \\
S_5^\star &= Q_{13}^T (P_{11} - Q_{11})^{-1} Q_{13} D_c - Q_{13}^T (P_{11} - Q_{11})^{-1} P_{12} \\
&+ Q_{33} D_c, \\
S_6^\star &= Q_{33} + Q_{13}^T (P_{11} - Q_{11})^{-1} Q_{13}, \\
S_7^\star &= -Q_{13}^T (P_{11} - Q_{11})^{-1} P_{13}.
\end{aligned}
$$

References

[1] J.P. Hespanha, P. Naghshtabrizi and Y. Xu, "Networked Control Systems: Analysis and Design", to appear in IEEE Proceedings, special issue on Network Control Technology, 2007.

[2] G.N. Nair, F. Fagnani, S. Zampieri and R.J. Evans, "Feedback Control under Data Rate Constraints: an Overview", to appear in IEEE Proceedings, special issue on Network Control Technology, 2007.

[3] D. Hristu-Varsakelis, "Feedback Control Systems as Users of a Shared Network: Communication Sequences that Guarantee Stability", in Proc. of 40^{th} Conf. on Decision and Control, 2001.

[4] M.S. Branicky, S.M. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems", in Proc. of 41^{st} Conf. on Decision and Control, 2002.

[5] S. Lall and G. Dullerud, "An LMI Solution to the Robust Synthesis Problem for Multi-rate Sampled-data Systems", Automatica, Vol. 37, pp. 1909-1922, 2001.

[6] G.C. Walsh, O. Belidman, and L.G. Bushnell, "Stability Analysis of Networked Control Systems", IEEE Trans. on Control System Technology, Vol. 10, pp. 438 - 446, March 2002.

[7] G.C. Walsh, H. Ye and L.G. Busnell, "Asymptotic Behavior of Nonlinear Networked Control Systems", IEEE Trans. on Automatic Control, Vol. 46, pp. 1093-1097, July 2001.

[8] G.C. Walsh and H. Ye, "Scheduling of Networked Control Systems", IEEE Control System Magazine, Vol. 21, pp. 57-65, February 2001.

[9] G.C. Walsh, O. Belidman and L.G. Bushnell, "Error Encoding Algorithms for Networked Control Systems", Automatica, Vol. 38, pp. 261-267, 2002.

[10] D. Nešić and A.R. Teel, "Input-output Stability Properties of Networked Control Systems", IEEE Trans. on Automatic Control, Vol. 49, pp. 1650 - 1667, 2004.

[11] D. Nešić and A.R. Teel, "Input to State Stability of Networked Control Systems", Automatica, vol. 40, pp. 2121-2128, 2004.

[12] M. Tabbara, D. Nešić and A.R. Teel, "Input-output Stability of Wireless Networked Control Systems", in Proc. of the joint 44^{th} Conf. on Decision and Control and European Control Conf., 2005.

[13] L. Palopoli, A. Bicchi and A.S. Vincentelli, "Numerically Efficient Control of Systems with Communication Constraints", in Proc. of 41^{st} Conf. on Decision and Control, 2002.

[14] H. Rehbinder and M. Sanfridson, "Scheduling of a Limited Communication Channel for Optimal Control", Automatica, vol. 40, pp. 491-500, March 2004.

[15] L.Zhang and D. Hristu-Varsakelis, "Stabilization of Networked Control Systems under Feedback based Communication", in Proc. of American Control Conf., 2005.

[16] M.M. Ben Gaid and A. Çela, "Model Predictive Control of Systems with Communication Constraints", in Proc. of the 16^{th} IFAC World Congress, 2005.

[17] R. Goebel, J.P. Hespanha, A.R. Teel, C. Cai and R. Sanfelice, "Hybrid Systems: Generalized Solutions and Robust Stability", in Proc. of 6^{th} NOLCOS, 2004.

[18] D. Nešić, A.R. Teel, and E.D. Sontag, "Formulas relating KL-stability estimates of discrete-time and sampled-data nonlinear systems", Systems and Control Letters, vol. 38, pp. 49-60, 1999.

[19] Z. Artstein, "Stabilization with Relaxed Controls", Nonlinear Analysis, vol. 7, pp. 1163-1173, 1983.

[20] E.D. Sontag, "A 'universal' Construction of Artstein's Theorem on Nonlinear Stabilization", Systems and Control Letters, vol. 13, pp. 117-123, 1989.

[21] E.D. Sontag and Y. Wang, "Output-to-state Stability and Detectability of Nonlinear Systems", Systems and Control Letters, vol. 29, pp. 279-290, 1997.

[22] 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.

[23] C. Prieur and L. Praly, "A Tentative Direct Design of Output Feedbacks", in Proc. of 6^{th} NOLCOS, 2004.

[24] 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.

[25] K.C. Goh, M.G. Safonov and G.P. Papavassilopoulos, "A Global Optimization Approach for the BMI Problem", in Proc. of 33^{rd} Conf. on Decision and Control, 1994.

[26] E. Beran, L. Vanderberghe and S. Boyd, "A Global BMI Algorithm based on the Generalized Benders Decomposition", in Proc. of European Control Conf., 1997.

[27] A. Hassibi, J. How, and S. Boyd, "A Path-following Method for Solving BMI Problems in Control", in Proc. of American Control Conf., USA, 1999.

[28] J.K. Yook, D.M. Tilbury, and N.R. Soparkar, "Trading computation for bandwidth: reducing communication in distributed control systems using state estimators", IEEE Trans. on Control Systems Technology, Vol. 10, pp. 503-518, 2002.