Implementation Considerations For Wireless Networked Control Systems

Payam Naghshtabrizi and João P. Hespanha

Abstract We show that delay impulsive systems are a natural framework to model wired and wireless NCSs with variable sampling intervals and delays as well as packet dropouts. Then, we employ discontinuous Lyapunov functionals to characterize admissible sampling intervals and delays such that exponential stability of NCS is guaranteed. These results allow us to determine requirements needed to establish exponential stability, which is the most basic Quality of Performance (QoP) required by the application layer. Then we focus on the question of whether or not the Quality of Service (QoS) provided by the wireless network suffices to fulfil the required QoP. To answer this question we employ results from real-time scheduling and provide a set of conditions under which the desired QoP can be achieved.

1 Introduction

Network Control Systems (NCSs) are spatially distributed systems in which the communication between sensors, actuators, and controllers occurs through a shared band-limited digital communication network, as shown in Fig. 1. There are two types of NCSs in terms of medium used at the physical layer: wired and wireless. Wired NCSs have been used widely in automotive and aerospace industry [14] to reduce weight and cost, increase reliability and connectivity. Particularly drive-by-wire and fly-by-wire systems have shown a high penetration rate in these industries. In wireless NCSs, the communication relies on the wireless technology and it has been finding applications in a broad range of areas that in which it is difficult or expensive to install wire, such as mobile sensor networks [17], HVAC systems [1], automated highway and unmanned aerial vehicles [18].

In both the wired and wireless domains, use of a shared network—in contrast to using several dedicated independent connections—introduces new challenges to

Payam Naghshtabrizi
Ford Motor Company, Dearborn MI, e-mail: pnaghsht@ford.com

João P. Hespanha
University of California, Santa Barbara, CA e-mail: hespanha@ece.ucsb.edu

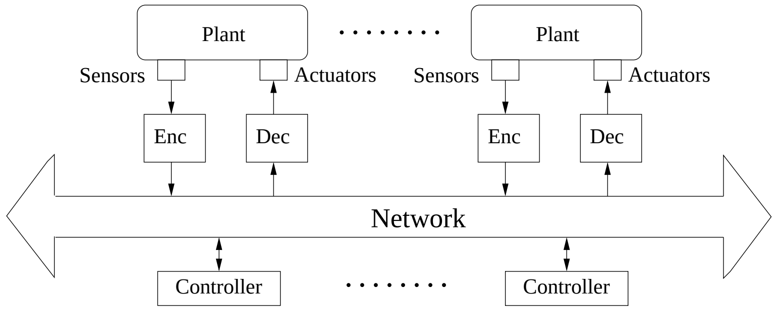

Fig. 1 General NCS architecture.

NCSs [7]. However, the reduced channel reliability and limited bandwidth that characterize the wireless domain require special care. In this paper we mainly focus on wireless NCSs, although most of the results presented are also applicable to wired NCSs. Traditional control theory assumes the feedback data to be accurate, timely, and lossless. These assumptions do not hold for wireless NCSs and the following factors have to be considered:

Sampling and Delay. To transmit a continuous-time signal over a network, the signal must be sampled, encoded in a digital format, transmitted over the network, and finally the data must be decoded at the receiver side. This process is significantly different from the usual periodic sampling in digital control. The overall *delay* between sampling and eventual decoding at the receiver can be highly variable because both the network access delays (i.e., the time it takes for a shared network to accept data) and the transmission delays (i.e., the time during which data are in transit inside the network) depend on highly variable network conditions such as congestion and channel quality. In some NCSs, the data transmitted are time-stamped, which means that the receiver may have an estimate of the delay's duration and take appropriate corrective action.

Packet dropout. Another significant difference between NCSs and standard digital control is the possibility that data may be lost while in transit through the network. Typically, *packet dropouts* result from transmission errors in physical network links (which is far more common in wireless than in wired networks) or from buffer overflows due to congestion. Long transmission delays sometimes result in packet re-ordering, which essentially amounts to a packet dropout if the receiver discards "outdated" arrivals.

Systems architecture. Fig. 1 shows the general architecture of an NCS. In this figure, the *encoder* blocks map measurements into streams of "symbols" that can be transmitted across the network. Encoders serve two purposes: they decide *when* to sample a continuous-time signal for transmission and *what* to send through the network. Conversely, the *decoder* blocks perform the task of mapping the streams of symbols received from the network into continuous actuation signals. One could also include in Fig. 1 encoding/decoding blocks to mediate the controllers' access to the network. Throughout this paper the encoder is simply a sampler and the de-

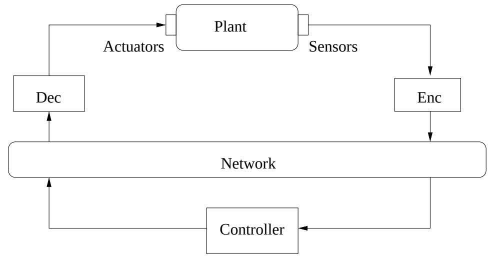

Fig. 2 Single-loop NCS.

coder is a hold. However, in Section 3.1.3 we will consider more sophisticated encoder/decoder pairs.

Most of the research on NCSs considers structures simpler than the general one depicted in Fig. 1. For example, some controllers may be collocated (and therefore can communicate directly) with the corresponding actuators. It is also often common to consider a single feedback loop as in Fig. 2. Although considerably simpler than the system shown in Fig. 1, this architecture still captures many important characteristics of NCSs such as bandwidth limitations, variable communication delays, and packet dropouts. In this paper we only consider linear plants and controller; however, some of the results can be extended to nonlinear systems [11].

In Sections 2 and 3, we show that delay impulsive systems provide a natural framework to model (wireless) NCSs with variable sampling intervals and delays as well as packet dropouts. Then, we employ discontinuous Lyapunov functionals to derive a condition that can be used to guarantee stability of the closed-loop NCS. This condition is expressed in the form of a set of LMIs that can be solved numerically using software packages such as MATLAB. By solving these LMIs, one can characterize admissible sampling intervals and delays for which exponential stability of the NCS is guaranteed.

From a networking perspective, the NCS is implemented using the usual layered architecture shown in Fig. 3 [10]. From this perspective, our goal is to determine under what conditions the network can provide stabilization, which is the most basic form of Quality of Performance (QoP). In essence, the stability conditions place requirements on the Quality of Service (QoS) that the lower layers need to provide to the application layer to obtain the desired QoP.

Section 4 is focused precisely on determining conditions under which the network provides a level of QoS that permits the desired application layer QoP. We review different real-time scheduling policies and identify the ones implementable on wireless NCSs. Among the discussed policies, the most desirable is Earliest Deadline First (EDF) because it has the advantage of being a dynamic algorithm that can adapt to changes in the wireless network. For EDF scheduling, we provide a set of conditions, often known as scheduling tests in real-time literature, that when satisfied, ensures the desired QoS of wireless NCS. Finally in Section 5 we address the question of how to implement EDF scheduling policy on LAN wireless NCSs.

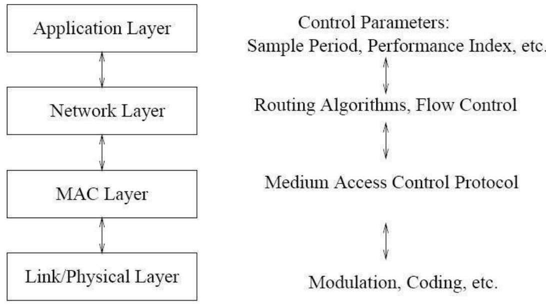

Fig. 3 Layered view of Wireless NCS.

Notation We denote the transpose of a matrix P by P'. We write $P > 0$ (or $P < 0$) when P is a symmetric positive (or negative) definite matrix and we write a symmetric matrix $\left[\begin{smallmatrix} A & B \\ B' & C \end{smallmatrix}\right]$ as $\left[\begin{smallmatrix} A & B \\ * & C \end{smallmatrix}\right]$. We denote the limit from below of a signal $x(t)$ by $x^-(t)$, i.e., $x^-(t) := \lim_{\tau \uparrow t} x(\tau)$. Given an interval $I \subset \mathbb{R}$, $B(I, \mathbb{R}^n)$ denotes the space of real functions from I to $\mathbb{R}^n$ with norm $\|\phi\| := \sup_{t \in I} |\phi(t)|, \phi \in B(I, \mathbb{R}^n)$ where $|.|$ denotes any one of the equivalent norms in $\mathbb{R}^n$. x_t denotes the function $x_t : [-r, 0] \to \mathbb{R}^n$ defined by $x_t(\theta) = x(t + \theta)$, and r is a fixed positive constant.

1.1 Related Work

To reduce network traffic in NCSs, significant work has been devoted to finding maximum allowable transmission interval τ_{MATI} that are not overly conservative (see [7] and references therein). First we review the related work in which there is no delay in the control loop. In [21], τ_{MATI} is computed for linear and nonlinear systems with Round-Robin (static) or Try-Once-Discard (TOD) (dynamic) protocols. Nesic et al. [15, 16] study the input-output stability properties of nonlinear NCSs based on a small gain theorem to find τ_{MATI} for NCSs. [6, 13, 24] consider linear NCSs and formulate the problem of finding τ_{MATI} by solving LMIs. In the presence of variable delays in the control loop, [5, 12, 25] show that for a given lower bound $\tau_{\min}$ on the delay in the control loop, stability can be guaranteed for a less conservative τ_{MATI} than in the absence of the lower bound.

Ye et al. [23] introduced prioritized Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for mixed wireless traffic, in which some of the network capacity is devoted to real time control and monitoring. They introduced and validated several new algorithms for dynamically scheduling the traffic of wireless NCSs. We use similar MAC protocol for the wireless network (more precisely wireless LAN networks). Liu and Goldsmith [10] presented a cross layer codesign of network and distributed controllers and addressed the tradeoff between communication and con-

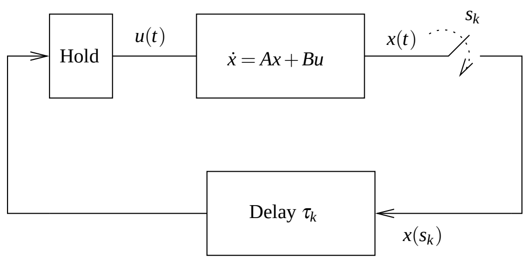

Fig. 4 An abstract system with delay τ_k where $u(t) = x(s_k)$ for $t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1})$
.

troller performance. The designed controller is robust and adaptive to the communication faults such as random delays and packet losses, while the network should be designed with the goal of optimizing the end-to-end control performance. Tabbara et al. [19] defined the notion of persistently exciting scheduling protocols and showed that it is a natural property to demand, especially for the design of wireless NCSs. Xia et al. [22] developed a cross layer adaptive feedback scheduling scheme to codesign control and wireless communications. The authors identified that the Deadline Miss Ratio (DMR) is an important factor to determine the sampling intervals. Consequently, the authors proposed a sampling algorithm that is the minimum of a function of DMR and maximum sampling period.

2 Delay Impulsive Systems: A Model For NCSs With Variable Sampling And Delay, SISO Case

Consider the system depicted in Fig. 4. The LTI process has a state space model of the form $\dot{x}(t) = Ax(t) + Bu(t)$, where x, u are the state and input of the process. At the sampling time s_k, $k \in \mathbb{N}$ the process state, $x(s_k)$ is sent to update the process input to be used as soon as it arrives and it should be kept constant until the next control command update. We denote the *k-th input update time* by t_k, which is the time instant at which the k-th sample arrives at the destination. In particular, denoting by τ_k the total delay that the k-th sample experiences in the loop, then $t_k := s_k + \tau_k$. The resulting closed-loop system can be written as

$$\dot{x}(t) = Ax(t) + Bx(s_k), \qquad t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1}), k \in \mathbb{N}, \tag{1}$$

We write the resulting closed-loop system (1) as an impulsive system of the form

$$\dot{\xi}(t) = F\xi(t), \qquad t \neq t_k, \forall k \in \mathbb{N} \tag{2a}$$

$$\xi(t_k) = \begin{bmatrix} x^-(t_k) \\ x(s_k) \end{bmatrix}, \qquad t = t_k, \forall k \in \mathbb{N}, \tag{2b}$$

where

$$F := \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}, \qquad \xi(t) := \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}.$$

The overall state of the system ξ is composed of the process state, x, and the *hold state*, z where $z(t) := x(s_k)$, $t \in [t_k, t_{k+1})$.

2.1 NCSs Modeled By Impulsive Systems

Equations (2) or (1) can be used to model NCSs in which a linear plant $\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t)$ where $x_p \in \mathbb{R}^n, u_p \in \mathbb{R}^m$ are the state and the input of the plant respectively, is in feedback with a static feedback gain K. At time s_k, $k \in \mathbb{N}$ the plant's state, $x(s_k)$, is sent to the controller and the control command $Kx(s_k)$ is sent back to the plant to be used as soon as it arrives and it should be kept constant until the next control command update. In particular, denoting by τ_k the total delay that the k-th sample experiences in the loop, then $t_k := s_k + \tau_k$. Then the closed-loop system can be written as (2) with $x := x_p$, $A := A_p$, $B := B_p K$.

Remark 1. Note that we only index the samples that reach the destination, which enables us to capture sample drops [24]. Consequently, even if the sampling intervals are constant, because of the sample drops the closed-loop should still be seen as a system with variable sampling intervals.

2.2 Exponential Stability Of SISO NCSs

In this section we provide conditions in terms of LMIs to guarantee exponential stability of the linear delay impulsive system in (2) which models the NCS described in section 2.1. The system (2) is said to be (globally uniformly) exponentially stable over a given set $\mathscr{S}$ of sampling-delay sequences, if there exist $c, \lambda > 0$ such that for every $(\{s_k\}, \{\tau_k\}) \in \mathscr{S}$ and every initial condition x_{t_0} the solution to (2) satisfies $|x(t)| \leq c \|x_{t_0}\| e^{-\lambda(t-t_0)}$, $\forall t \geq t_0$.

In this paper, we are mostly interested in class $\mathscr{S}$ of admissible sampling-delay sequences characterized by three parameters: The maximum interval of time τ_{MATI} between a signal is sampled and the *following* sample arrives at the destination; the minimum delay $\tau_{\min}$; and the maximum delay $\tau_{\max}$. Specifically, to be consistent with the results in [12], [25], and [5] we characterize the admissible set $\mathscr{S}$ of sampling-delay sequences $(\{s_k\}, \{\tau_k\})$ such that

$$s_{k+1} + \tau_{k+1} - s_k \leq \tau_{MATI}, \qquad \tau_{\min} \leq \tau_k \leq \tau_{\max}. \tag{3}$$

Although we adopt the above characterization, (3) is not in a convenient form to provide the sampling rule. Another characterization is the admissible set $\bar{\mathscr{S}}$ of sampling-delay sequences $(\{s_k\},\{\tau_k\})$ such that

$$s_{k+1}-s_k \leq \gamma_{\max}, \qquad \tau_{\min} \leq \tau_k \leq \tau_{\max}, \tag{4}$$

which provides an explicit bound on the sampling intervals. Note that if any sampling-delay sequence belongs to $\bar{\mathscr{S}}$, it necessarily belongs to $\mathscr{S}$ provided that $\gamma_{\max} := \tau_{MATI} - \tau_{\max}$.

The following theorem was proved in [11] based on the Lyapunov functional

$$\begin{aligned} V := {} & x'Px + \int_{t-\rho_1}^{t} (\rho_{1\max} - t + s)\dot{x}'(s)R_1\dot{x}(s)ds + \\ & \int_{t-\rho_2}^{t} (\rho_{2\max} - t + s)\dot{x}'(s)R_2\dot{x}(s)ds + \int_{t-\tau_{\min}}^{t} (\tau_{\min} - t + s)\dot{x}'(s)R_3\dot{x}(s)ds + \\ & \int_{t-\rho_1}^{t-\tau_{\min}} (\rho_{1\max} - t + s)\dot{x}'(s)R_4\dot{x}(s)ds + (\rho_{1\max} - \tau_{\min}) \int_{t-\tau_{\min}}^{t} \dot{x}'(s)R_4\dot{x}(s)ds + \\ & \int_{t-\tau_{\min}}^{t} x'(s)Zx(s)ds + (\rho_{1\max} - \rho_1)(x-w)'X(x-w), \end{aligned} \tag{5}$$

with $P, , X, Z, R_i, i = 1, .., 4$ positive definite matrices and

$$w(t) := x(t_k), \qquad \rho_1(t) := t - s_k, \qquad \rho_2(t) := t - t_k, \quad t_k \leq t < t_{k+1}, \tag{6}$$

$$\rho_{1\max} := \sup_{t\geq 0} \rho_1(t), \qquad\qquad \rho_{2\max} := \sup_{t\geq 0} \rho_2(t). \tag{7}$$

Theorem 1. *The system (2) is exponentially stable over* $\mathscr{S}$ *defined by* (3)*, if there exist positive definite matrices* $P, X, Z, R_i, i = 1, .., 4$ *and (not necessarily symmetric) matrices* $N_i, i = 1, .., 4$ *that satisfy the following LMIs:*

$$\begin{bmatrix} M_1+\tau_{MATI}(M_2+M_3) & \tau_{\max}N_1 & \tau_{\min}N_3 \\ * & -\tau_{\max}R_1 & 0 \\ * & * & -\tau_{\min}R_3 \end{bmatrix} < 0, \tag{8a}$$

$$\begin{bmatrix} M_1+\tau_{MATI}M_2 & \tau_{\max}N_1 & \tau_{\min}N_3 & \tau_{MATI}(N_1+N_2) & \tau_{MATI}N_4 \\ * & -\tau_{\max}R_1 & 0 & 0 & 0 \\ * & * & -\tau_{\min}R_3 & 0 & 0 \\ * & * & * & -\tau_{MATI}(R_1+R_2) & 0 \\ * & * & * & * & -\tau_{MATI}R_4 \end{bmatrix} < 0, \tag{8b}$$

where

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

$$M_2 := \bar{F}'(R_1+R_2+R_4)\bar{F}, \qquad M_3 := \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix} X\bar{F} + \bar{F}'X\begin{bmatrix} I & 0 & -I & 0 \end{bmatrix}.$$

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

If the LMIs in (8) are feasible for given $\tau_{MATI}, \tau_{\min}$, and $\tau_{\max}$, then there exists a $d_3 > 0$ such that $\frac{\mathrm{d}V(x_t,t)}{\mathrm{d}t} \leq -d_3|x(t)|^2$. It is straightforward to show that the Lyapunov functional (5) satisfies the remaining conditions of Theorem 15 in [11] that provides sufficient conditions for exponential stability of delay impulsive systems. Hence the NCS modeled by the delay impulsive system (2) is (globally uniformly) exponentially stable over $\mathscr{S}$ given by (3).

When the load in the network is low and the computation delays are small, the total end-to-end delay in the loop is dominated by transmission and propagation delays which can be small. This motivates a closer examination of the case $\tau_{\min} = 0$, which is the subject of the following result. The conditions in the theorem that follows can be derived from the conditions in Theorem 1 for the case when $\tau_{\min} \to 0$ or they can be directly derived employing the following Lyapunov functional

$$V := x'Px + \int_{t-\rho_1}^{t} (\rho_{1\max} - t + s)\dot{x}'(s)R_1\dot{x}(s)ds + \int_{t-\rho_2}^{t} (\rho_{2\max} - t + s)\dot{x}'(s)R_2\dot{x}(s)ds + (\rho_{1\max} - \rho_1)(x-w)'X(x-w).$$

Theorem 2. *The system (2) is exponentially stable over* $\mathscr{S}$ *defined by* (3) *with* $\tau_{\min} = 0$, *if there exist positive definite matrices* P, X, R_1, R_2 *and (not necessarily symmetric) matrices* N_1, N_2 *that satisfy the following LMIs:*

$$\begin{bmatrix} M_1+\tau_{MATI}(M_2+M_3) & \tau_{\max}N_1 \\ * & -\tau_{\max}R_1 \end{bmatrix} < 0, \tag{9a}$$

$$\begin{bmatrix} M_1+\tau_{MATI}M_2 & \tau_{\max}N_1 & \tau_{MATI}(N_1+N_2) \\ * & -\tau_{\max}R_1 & 0 \\ * & * & -\tau_{MATI}(R_1+R_2) \end{bmatrix} < 0, \tag{9b}$$

where

$$\begin{aligned} M_1 :=& \bar{F}'\begin{bmatrix} P & 0 & 0 \end{bmatrix} + \begin{bmatrix} P \\ 0 \\ 0 \end{bmatrix}\bar{F} - \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix} X \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix}' - N_1\begin{bmatrix} I & -I & 0 \end{bmatrix} - \begin{bmatrix} I \\ -I \\ 0 \end{bmatrix} N_1' \\ & - N_2\begin{bmatrix} I & 0 & -I \end{bmatrix} - \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix} N_2' \\ M_2 :=& \bar{F}'(R_1+R_2+R_4)\bar{F}, \\ M_3 :=& \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix} X\bar{F} + \bar{F}'X\begin{bmatrix} I & 0 & -I \end{bmatrix}. \end{aligned} \tag{10}$$

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

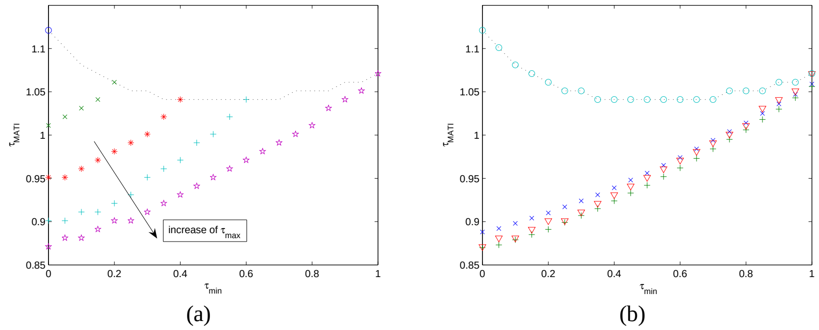

Fig. 5 Fig. 5(a) shows the plot of τ_{MATI} versus $\tau_{\min}$ for $\tau_{\max}$ equal to $0, .2, .4, .6, 1$ based on Theorem 1. The dashed line is the same as the one in Fig.5(b). Fig. 5(b) shows the plot τ_{MATI} versus $\tau_{\min}$ where $\tau_{\max} = \tau_{\min}$ from [12] ('$+$') and [25] ('$\times$'), the worse case where $\tau_{\max} = \tau_{MATI}$ ('∇') and the best case where $\tau_{\max} = \tau_{\min}$ ('o') from Theorem 1.

2.3 Example

Consider the state space plant model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u,$$

with state feedback gain $K = -\begin{bmatrix} 3.75 \; 11.5 \end{bmatrix}$, for which we have

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -0.1 \end{bmatrix}, \qquad B = -\begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \times \begin{bmatrix} 3.75 & 11.5 \end{bmatrix}.$$

[2]. By checking the condition $eig(\left[\begin{smallmatrix} I & 0 \\ I & 0 \end{smallmatrix}\right] e^{Fh}) < 0$ on a tight grid of h, we can show that the closed-loop system remains stable for any *constant* sampling interval smaller than 1.7, and becomes unstable for larger constant sampling intervals. On the other hand, when the sampling interval approaches zero, the system is essentially described by a Delay Differential Equation (DDE) and we can find the maximum constant delay for which stability is guaranteed by looking at the roots of the characteristic function $\det(sI - A - Be^{-\tau_0 s})$. Using the Pade approximation $e^{-\tau_0 s} = \frac{1 - s\tau_0/2}{1 + s\tau_0/2}$ to compute the determinant polynomial, we conclude by the Routh-Hurwitz test that the system is stable for any constant delay smaller than 1.36. Comparing these numbers with the maximum variable sampling interval 1.1137 and the maximum variable delay 1.0744 both obtained using Theorem 1 (see below) reveals that this result is not very conservative.

No-delay and variable sampling. When there is no delay but the sampling intervals are variable, τ_{MATI} determines an upper bound on the variable sampling intervals $s_{k+1} - s_k$. The upper bound given by [6, 12, 24] (when $\tau_{\min} = 0$) is 0.8696

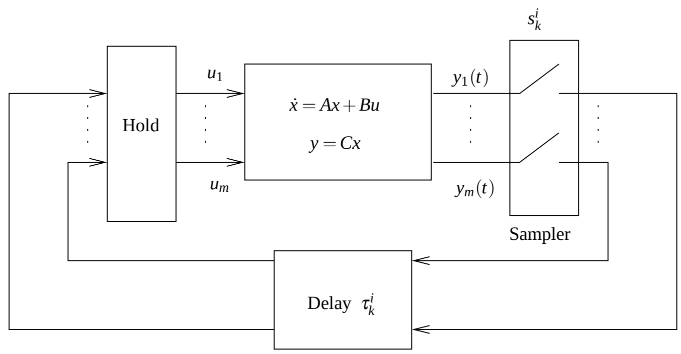

Fig. 6 MIMO system with variable sampling intervals and delays where $u_i(t) = y_i(s_k^i)$ for $t \in [t_k^i, t_{k+1}^i), \forall i \in \{1, \cdots, m\}$ where $t_k^i := s_k^i + \tau_k^i$.

which is improved to 0.8871 in [25]. Theorem 1 and [13] gives the upper bound equal to 1.1137.

Variable-delay and sampling. Fig. 5(a) shows the value of τ_{MATI} obtained from Theorem 1, as a function of $\tau_{\min}$ for different values of τ_{max}. The dashed curves in Fig. 5(a) and Fig. 5(b) are the same and correspond to of the largest τ_{MATI} for different values of $\tau_{\max}$. Fig. 5(b) shows τ_{MATI} versus τ_{min} where the results from [12], [25] are shown by $+$, $\times$ respectively. The values of τ_{MATI} given by [5] lie between the "$+$" and "$\times$" in Fig. 5(b) and we do not show them. In Theorem 1, τ_{MATI} is a function of $\tau_{\min}$ and $\tau_{\max}$. To be able to compare our result to the others we consider two values for $\tau_{\max}$ and we obtain τ_{MATI} as a function of $\tau_{\min}$ based on Theorem 1. First we consider $\tau_{\max} = \tau_{\min}$, which is the case that the delay is constant and equal to the value of $\tau_{\min}$. The largest τ_{MATI} for a given $\tau_{\min}$ provided by Theorem 1 is shown using an "o" in Fig. 5(b). The second case is when $\tau_{\max} = \tau_{MATI}$, which is the case where there can be very large delays in the loop in comparison to the sampling intervals. The largest τ_{MATI} for a given $\tau_{\min}$ for this case provided by Theorem 1 is shown using a "∇" in Fig. 5(b). One can observe that when the delays in the control loop are small, our method shows a good improvement in comparison to the other results in the literature.

3 Delay Impulsive Systems: A Model For NCSs With Variable Sampling And Delay, MIMO Case

We now consider the MIMO system depicted in Fig. 6. The input is partitioned as $u := [u_1' \cdots u_m']'$ where $u_i \in \mathbb{R}^{q_i}$, $i \in \{1, \cdots, m\}$ and $\sum_{i=1}^m q_i = q$ and the output is partitioned similarly with $y := [y_1' \cdots y_m']'$ where $y_i \in \mathbb{R}^{q_i}$, $i \in \{1, \cdots, m\}$. At time s_k^i, $i \in \{1, \cdots, m\}, k \in \mathbb{N}$ the $i-$th output of the system, $y_i(t)$ is sampled and $y_i(s_k^i)$ is sent through the network to update u_i, to be used as soon as it arrives until the next update arrives. The total delay in the control loop that the k-th sample of y_i experiences is denoted by τ_k^i where $\tau_{i\min} \le \tau_k^i \le \tau_{i\max}$, $\forall k \in \mathbb{N}, i \in \{1,\cdots,m\}$. We define $t_k^i := s_k^i + \tau_k^i$ which is the time instant that the value of u_i is updated. The overall system can be written as an impulsive system of the form

$$\dot{\xi}(t) = F\xi(t), \qquad t \neq t_k^i, \forall k \in \mathbb{N}, i \in \{1,\cdots,m\} \tag{11a}$$

$$\xi(t_k) = \begin{bmatrix} x^-(t_k^i) \\ \cdots\cdots \\ z_1^-(t_k^i) \\ \vdots \\ y_i(s_k^i) \\ \vdots \\ z_m^-(t_k^i) \end{bmatrix}, \qquad t = t_k^i, \forall k \in \mathbb{N}, i \in \{1,\cdots,m\}, \tag{11b}$$

where

$$F := \begin{bmatrix} A & \vdots & B \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix}, \qquad \xi(t) := \begin{bmatrix} x(t) \\ \cdots\cdots \\ z(t) \end{bmatrix}, \qquad z(t) := \begin{bmatrix} z_1(t) \\ \vdots \\ z_m(t) \end{bmatrix},$$

so we have $z_i(t) := y_i(s_k^i), t \in [t_k^i, t_{k+1}^i)$.

3.1 *NCSs Modeled By MIMO System* (11)

The impulsive system (11) can be used to represent the distributed sensors/actuators configurations shown in Figs. 7 and 8. We now consider an LTI plant

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t), \qquad y_p(t) = C_p x_p(t), \tag{12}$$

where $x_p \in \mathbb{R}^{n_p}, u_p := [u'_{p1} \cdots u'_{pm_c}]' \in \mathbb{R}^{m_c}$, and $y_p := [y'_{p1} \cdots y'_{pm_p}] \in \mathbb{R}^{m_p}$ are the state, input and output of the plant and matrices, respectively. At time $s_k^i, i \in \{1,\cdots,m_p\}$, sensor i sends $y_{pi}(s_k^i)$ to the controller, which arrives at the destination at time t_k^i. When a new measurement of the sensor i arrives at the controller side, the corresponding value at the hold block, z_i, is updated and held constant until another measurement of the sensor i arrives (all other hold values remain unchanged when the value of hold i is updated). Hence $u_{ci}(t) = z_i(t) = y_{pi}(s_k^i)$, $t \in [t_k^i, t_k^{i+1})$ for $\forall i \in \{1,\cdots,m_p\}$. An output feedback controller (or a state feedback controller) uses the measurements to construct the control signal. The controller has the state space of the form

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t), \qquad y_c(t) = C_c x_c(t) + D_c u_c(t), \tag{13}$$

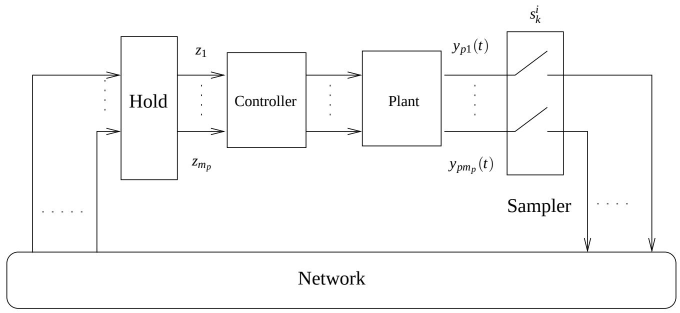

Fig. 7 One channel NCSs with the plant (12) and the controller (13).

where $x_c \in \mathbb{R}^{n_c}, u_c := \left[u'_{c1} \cdots u'_{cm_p} \right]' \in \mathbb{R}^{m_p}, y_c := \left[y'_{c1} \cdots y'_{cm_c} \right] \in \mathbb{R}^{m_p \times 1} \in \mathbb{R}^{m_c \times 1}$ are the state, input and output of the controller, respectively. The main difference between the NCS configurations discussed below is related to the control signal construction.

3.1.1 One-Channel NCS With Dynamic Feedback Controller

Fig. 7 shows a one-channel NCS consisting of a plant (12) in feedback with a dynamic output controller (13). In this one-channel NCS, the controller is directly connected to the plant and only the measurements of the plant are sent through the network. To match the system in Fig. 7 with the system in Fig. 6 $y_i(t) := y_{pi}(t)$, $m := m_p$, and $x := \left[\begin{smallmatrix} x_p \\ x_c \end{smallmatrix}\right]$. This closed-loop system can be written as the impulsive system (11) where

$$A := \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix}, \quad B := \begin{bmatrix} B_p D_c \\ B_c \end{bmatrix}, \quad C := \begin{bmatrix} C_p & 0 \end{bmatrix}. \tag{14}$$

One-channel NCSs may look artificial since the controller and the plant appear to be collocated and one may suggest that there is no need to send the output of the plant through the network. It turns out that there are important cases of NCSs that can be modeled as a one-channel NCSs. One example is controlling a robot using cameras that are not mounted on the robot, which provide the global image of the field. In this case, position of the robot is "measured" by cameras and the measurements are sent through the network to the robot to be used to compute the control commands by the local controller of the robot.

3.1.2 Two-Channel NCS With Non-Anticipative Controller

Fig. 8 shows a two-channel NCS consisting of a plant (12) in feedback with a *non-anticipative* controller (13) where $D_c = 0$. Non-anticipative controllers are simply output-feedback controllers for which a single value control command is calculated.

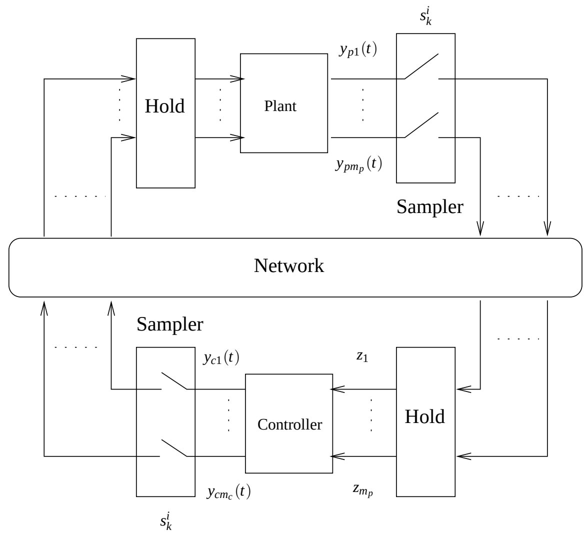

Fig. 8 Two channel NCS with the plant (12) and anticipative or non-anticipative controller (13).

Now the controller is located away from the actuators and the control commands also need to be sent through the network. The control signal for the actuator i, $y_{ci}(t)$, is sampled at times s_k^i, , $i \in [m_p+1, \cdots, m_p+m_c]$, and the samples get to the actuator i at times $t_k^i := s_k^i + \tau_k^i$. The non-anticipative control unit sends a single-value control command to be applied to the actuator i of the plant and held until the next control update of the actuator i arrives (all other actuator values remain unchanged while the value of actuator i is updated). Hence $u_{pi}(t) = z_i(t) = y_{ci}(s_k^i)$, $t \in [t_k^i, t_{k+1}^i)$ for $m_p + 1 \leq i \leq m_p + m_c$. So, in this case we can model the closed loop as the impulsive system (11) in Fig. 6, with

$$y_i := \begin{cases} y_{pi}, & 1 \leq i \leq m_p \\ y_{ci}, & m_p + 1 \leq i \leq m_p + m_c \end{cases},$$

$m := m_p + m_c$, $x := \left[\begin{smallmatrix} x_p \\ x_c \end{smallmatrix}\right]$ and

$$A := \begin{bmatrix} A_p & 0 \\ 0 & A_c \end{bmatrix}, \qquad B := \begin{bmatrix} 0 & B_p \\ B_c & 0 \end{bmatrix}, \qquad C := \begin{bmatrix} C_p & 0 \\ 0 & C_c \end{bmatrix}. \tag{15}$$

3.1.3 Two-channel NCS With Anticipative Controller

Fig. 8 can also represent a two-channel NCS consisting of a plant with the state-space (12) in feedback with an *anticipative* controller with state-space (13). Anticipative controller attempts to compensate the sampling and delay introduced by the actuation channels. For simplicity, we assume that the actuation channels are sampled with constant sampling intervals $h = s^i_{k+1} - s^i_k$, and that the delay in the actuation channels is constant and equal to $\tau = \tau^i_k \, \forall k \in \mathbb{N}, i \in \{m_p+1, \cdots, m_p+m_c\}$. At each sampling time s^i_k, $i \in [m_p+1, \cdots, m_p+m_c]$ the controller sends a time-varying control signal $y_{ci}(\cdot)$ to the actuator i. This control signal should be used from the time $s^i_k + \tau$ at which it arrives until the time $s^i_k + h + \tau$ at which the next control update of the actuator i will arrive. This leads to $u_{pi}(t) = y_{ci}(t), \quad \forall t \in [s^i_k + \tau, s^i_k + h + \tau)$. However, the prediction of control signal $y_{ci}(t)$ needed in the interval $[s^i_k + \tau, s^i_k + h + \tau)$ must be available at the transmission time s^i_k, which requires the control unit to calculate the control signal up to $h + \tau$ time units into the future.

Anticipative controllers send actuation signals to be used during time intervals of duration h, therefore the sample and hold blocks in Fig. 8 should be understood in a broad sense. In practice, the sample block would send over the network some parametric form of the control signal $u_{ci}(\cdot)$ (e.g., the coefficients of a polynomial approximation to this signal).

An estimate $\hat{x}_c(t)$ of $x_c(t+h+\tau)$ can be constructed as $\dot{\hat{x}}_c(t) = A_c\hat{x}_c(t) + B_c u_c(t)$, where

$$u_{ci}(t) = y_{pi}(s^i_k), \forall t \in [s^i_k + \tau, s^i_k + \tau + h), \tag{16}$$

for $i \in \{1, \cdots, m_p\}$. To compensate for time varying delays and sampling intervals in the actuation channels, $\hat{x}_c$ would have to be calculated further into the future. Hence the assumptions of constant delay and sampling interval for actuation channel can be relaxed by predicting x_c further into the future.

With the controller state prediction (16), the signal $y_{ci}(t)$ sent at times s^i_k, to be used in the interval $\left[s^i_k + \tau, s^i_k + h + \tau\right)$ is then given by

$$y_{ci}(t) = C_{ci}\hat{x}_c(t-h-\tau) + D_c u_{ci}(t), \forall t \in [s^i_k + \tau, s^i_k + h + \tau), \tag{17}$$

which only requires the knowledge of $\hat{x}_c(.)$ in the interval $t \in \left[s^i_k - h, s^i_k\right)$, available at the transmission times s^i_k. The closed-loop system can be written as

$$\begin{bmatrix} \dot{\hat{x}}_p(t) \\ \dot{\hat{x}}_c(t) \end{bmatrix} = \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} \hat{x}_p(t) \\ \hat{x}_c(t) \end{bmatrix} + \begin{bmatrix} B_p D_c \\ B_c \end{bmatrix} u_c(t), \tag{18}$$

where $\hat{x}_p(t) := x_p(t+h+\tau)$ and the elements of $u_c(t)$ are defined by (16). Hence in this case $y_i(t) := y_{pi}(t), \quad m := m_p$ and the closed-loop system with state $x := \left[\begin{smallmatrix} \hat{x}_p \\ \hat{x}_c \end{smallmatrix}\right]$ can be written as the impulsive system (11) where

$$A := \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix}, \quad B := \begin{bmatrix} B_p D_c \\ B_c \end{bmatrix}, \quad C := \begin{bmatrix} C_p & 0 \end{bmatrix}. \tag{19}$$

The equation (14), which represents a one-channel NCS with dynamic output feedback is similar to equation (19) that represents two-cannel NCS with anticipative controller. Consequently for analysis purpose one can model a two-cannel NCS with anticipative controller as a one-channel NCS with *"fictitious"* delays equal to the sum of the delay in the sensor to actuator channels, the delay in the actuator to sensor channels, and the sampling of the actuator channel.

Remark 2. Anticipative controllers are similar to model predictive controllers in the sense that both calculate future control actions. However in model predictive controllers only the most recent control prediction is applied. Anticipative controllers are predictive controllers that send a control prediction for a certain duration. At the expense of sending more information to the actuators in each packet, one expects that fewer packets need to be transmitted to stabilize the system.

3.2 Exponential stability of MIMO NCSs

In this section we provide exponential stability conditions for the linear impulsive system (11) that can model both one-channel and two-channel NCSs with anticipative or non-anticipative controller, as described in Section 3.1.

Since the minimum of delay in the network is typically small, for simplicity of derivations, we assume that $\tau_{i\min} = 0, 1 \le i \le m$. We now present two theorems for the stability of the system (11). These stability tests are generalizations of the result in Theorem 1. The first theorem is less conservative; however, the number of LMIs grows exponentially with m. The second stability condition is based on the feasibility of a single LMI, but its size grows only linearly with m. For small m the first stability test is more adequate because it leads to less conservative results, but the second stability test is more desirable for large m. We present our results for $m = 2$, but deriving the stability conditions for other values of m is straightforward by following the same steps.

Inspired by the Lyapunov functional (5), we employ the Lyapunov functional

$$V := V_1 + V_2 + V_3 + V_4, \tag{20}$$

$$\begin{aligned}
V_1 &:= x'Px,\\
V_2 &:= \sum_{i=1}^{2} \int_{t-\rho_i}^{t} (\rho_{i\max} - t + s)\dot{y}_i'(s) R_{1i} \dot{y}_i(s) ds,\\
V_3 &:= \sum_{i=1}^{2} \int_{t-\sigma_i}^{t} (\sigma_{i\max} - t + s)\dot{y}_i'(s) R_{2i} \dot{y}_i(s) ds,\\
V_4 &:= \sum_{i=1}^{2} (\rho_{i\max} - \rho_i)(y_i - w_i)' X_i (y_i - w_i),
\end{aligned}$$

with P, R_{1i}, R_{2i}, X_i symmetric positive definite matrices and

$$\rho_i(t) := t - s_k^i,\ \sigma_i(t) := t - t_k^i,\ w_i(t) := y_i(t_k^i), \quad t \in [t_k^i, t_{k+1}^i),$$
$$\rho_{i\max} := \sup_{t\geq 0} \rho_i(t), \quad \sigma_{i\max} := \sup_{t\geq 0} \sigma_i(t),$$

for $i = 1,2$. The next theorem characterizes the admissible set $\mathscr{S}_i, i = 1,2,$ of sampling-delay sequences $(\{s_k^i\},\{\tau_k^i\})$ such that

$$s_{k+1}^i + \tau_{k+1}^i - s_k^i \leq \rho_{i\max}, \qquad 0 \leq \tau_k^i \leq \tau_{i\max}, \quad \forall k \in \mathbb{N}. \tag{21}$$

Theorem 3. *The system* (11) *is exponentially stable over* $\mathscr{S}$ *defined by* (21)*, if there exist symmetric positive definite matrices* P, R_{1i}, R_{2i}, X_i *and (not necessarily symmetric) matrices* N_{1i}, N_{2i} *that satisfy the following LMIs:*

$$\begin{bmatrix} M_1+\rho_{1\max}(M_{21}+M_{31})+\rho_{2\max}(M_{22}+M_{32}) & \tau_{1\max}N_{11} & \tau_{2\max}N_{12} \\ * & -\tau_{1\max}R_{11} & 0_{q_1q_2} \\ * & * & -\tau_{2\max}R_{12} \end{bmatrix} < 0, \tag{22a}$$

$$\begin{bmatrix} M_1+\rho_{1\max}M_{21}+\rho_{2\max}(M_{22}+M_{32}) & \tau_{1\max}N_{11} & \tau_{2\max}N_{12} & G_{11} \\ * & -\tau_{1\max}R_{11} & 0_{q_1q_2} & 0_{q_1q_1} \\ * & * & -\tau_{2\max}R_{12} & 0_{q_2q_1} \\ * & * & * & G_{21} \end{bmatrix} < 0, \tag{22b}$$

$$\begin{bmatrix} M_1+\rho_{1\max}(M_{21}+M_{31})+\rho_{2\max}M_{22} & \tau_{1\max}N_{11} & \tau_{2\max}N_{12} & G_{12} \\ * & -\tau_{1\max}R_{11} & 0_{q_1q_2} & 0_{q_1q_2} \\ * & * & -\tau_{2\max}R_{12} & 0_{q_2q_2} \\ * & * & * & G_{22} \end{bmatrix} < 0, \tag{22c}$$

$$\begin{bmatrix} M_1+\rho_{1\max}M_{21}+\rho_{2\max}M_{22} & \tau_{1\max}N_{11} & \tau_{2\max}N_{12} & G_{11} & G_{12} \\ * & -\tau_{1\max}R_{11} & 0_{q_1q_2} & 0_{q_1q_1} & 0_{q_1q_2} \\ * & * & -\tau_{2\max}R_{12} & 0_{q_2q_1} & 0_{q_2q_2} \\ * & * & * & G_{21} & 0_{q_1q_2} \\ * & * & * & * & G_{22} \end{bmatrix} < 0, \tag{22d}$$

where $\bar{F} := \begin{bmatrix} A & B & 0_{nq} \end{bmatrix}$ *and*

$$\begin{aligned} M_1 &:= \bar{F}' \begin{bmatrix} P & 0_{nq} & 0_{nq} \end{bmatrix} + \begin{bmatrix} P \\ 0_{qn} \\ 0_{qn} \end{bmatrix} \bar{F} - T_1'X_1T_1 - T_2'X_2T_2 - N_{11}T_3 \\ &- T_3'N_{11}' - N_{21}T_1 - T_1'N_{21}' - N_{12}T_4 - T_4'N_{12}' - N_{22}T_2 - T_2'N_{22}', \\ M_{2i} &:= \bar{F}'C_i'(R_{1i}+R_{2i})C_i\bar{F}, \qquad M_{3i} := T_i'X_iC_i\bar{F} + \bar{F}'C_i'X_iT_i, \\ G_{1i} &:= \rho_{i\max}(N_{1i}+N_{2i}), \qquad G_{2i} := \rho_{i\max}(R_{1i}+R_{2i}) \end{aligned} \tag{23}$$

with

$$\begin{aligned} &C_1 := \begin{bmatrix} I_{q_1} & 0_{q_1q_2} \end{bmatrix} C, \quad C_2 := \begin{bmatrix} 0_{q_2q_1} & I_{q_2} \end{bmatrix} C, \quad T_1 := \begin{bmatrix} C_1 & 0_{q_1q} & -\bar{I}_1 \end{bmatrix}, \ T_2 := \begin{bmatrix} C_2 & 0_{q_2q} & -\bar{I}_2 \end{bmatrix}, \\ &T_3 := \begin{bmatrix} C_1 & -\bar{I}_1 & 0_{q_1q} \end{bmatrix}, \quad T_4 := \begin{bmatrix} C_2 & -\bar{I}_2 & 0_{q_2q} \end{bmatrix}, \quad \bar{I}_1 := \begin{bmatrix} I_{q_1} & 0_{q_1q_2} \end{bmatrix}, \quad \bar{I}_2 := \begin{bmatrix} 0_{q_2q_1} & I_{q_2} \end{bmatrix} \end{aligned} \tag{24}$$

□

It is possible to generalize Theorem 3 for an arbitrary m. However, the number of LMIs that one needs to solve equal to 2^m and the size of LMIs and the number of scalar variables increases linearly, which limits the application of this result for large values of m. The next theorem, also provided in [11], presents another stability test, which is more conservative, but only requires the solution of a single LMI.

Theorem 4. *The system* (11) *is exponentially stable over* $\mathscr{S}_i$, $i = 1,2$, *defined by* (21), *provided that there exist symmetric positive definite matrices* P, R_{1i}, R_{2i} *and (not necessarily symmetric) matrices* N_{1i}, N_{2i} *that satisfy the following LMIs:*

$$\begin{bmatrix} \bar{M}_1+\rho_{1\max}\bar{M}_{21}+\rho_{2\max}\bar{M}_{22} & \tau_{1\max}N_{11} & \tau_{2\max}N_{12} & G_{11} & G_{12} \\ * & -\tau_{1\max}R_{11} & 0_{q_1q_2} & 0_{q_1q_1} & 0_{q_1q_2} \\ * & * & -\tau_{2\max}R_{12} & 0_{q_2q_1} & 0_{q_2q_2} \\ * & * & * & G_{21} & 0_{q_1q_2} \\ * & * & * & * & G_{22} \end{bmatrix} < 0, \tag{25}$$

where $\bar{F} := \begin{bmatrix} A & B & 0_{nq} \end{bmatrix}$ *and*

$$\begin{aligned} \bar{M}_1 :=& \bar{F}' \begin{bmatrix} P & 0_{nq} & 0_{nq} \end{bmatrix} + \begin{bmatrix} P \\ 0_{qn} \\ 0_{qn} \end{bmatrix} \bar{F} - N_{11}T_3 - T_3'N_{11}' - N_{21}T_1 \\ & -T_1'N_{21}' - N_{12}T_4 - T_4'N_{12}' - N_{22}T_2 - T_2'N_{22}', \\ \bar{M}_{2i} :=& \bar{F}'C_i'(R_{1i}+R_{2i})C_i\bar{F}, \quad G_{1i} := \rho_{i\max}(N_{1i}+N_{2i}), \quad G_{2i} := \rho_{i\max}(R_{1i}+R_{2i}). \end{aligned}$$

with the matrix variables defined in (24). □

By solving the LMIs in Theorems 3 or 4 one can find positive constants $\rho_{i\max}$, $i = 1,2$, that determine upper bounds between the *consecutive samples of channel i* for which the stability of the closed-loop system is guaranteed, for a given lower and upper bound on the total delay in the loop i.

Most of the work in the literature has been devoted to finding a single constant τ_{MATI} ([7, 15, 16, 20, 21] and references therein) that provides an upper bound between *any consecutive sampling instances* for which the stability of the closed-loop system is guaranteed. It is thus not surprising that having m distinct constants $\rho_{i\max}$, $1 \le i \le m$ instead of one single constant τ_{MATI}, reveals more information about the system and permits less conservative results.

3.3 Example

This example appeared in [8, 15, 19, 21] and considers a linearized model of the form (12) for a two-input, two-output batch reactor where

$$A_p := \begin{bmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix}, \quad B_p := \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}, \quad C_p := \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

This system is controlled by a PI controller of the form (13) where

$$A_c := \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad B_c := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad C_c := \begin{bmatrix} -2 & 0 \\ 0 & 8 \end{bmatrix}, \quad D_C := \begin{bmatrix} 0 & -2 \\ 5 & 0 \end{bmatrix}.$$

Following the assumptions of [8, 15, 19, 21], we assume that only the outputs of

	no drop
Maximum deterministic time interval between samples from [19]	0.0123
Maximum stochastic arbitrary inter-sampling time distribution from [8]	0.0279
Maximum stochastic uniform inter-sampling time distribution from [8]	0.0517
Maximum deterministic time interval between samples from Theorem 3	0.0405
Maximum deterministic time interval between samples from Theorem 4	0.029

Table 1 Comparison of τ_{MATI} for Example 3.3 when there is no delay.

the plant are transmitted over the network, there are no dropouts and the outputs are sent in a round-robin fashion and consecutively. We compare τ_{MATI} of this example given by the stability results in [8, 15, 19, 21], where the effect of the delay was ignored. From Theorem 3 we compute $\rho_{1\max} = 0.081, \rho_{2\max} = 0.113$ when there is no delay. We can also show that if the upper bound between any consecutive sampling, τ_{MATI}, is smaller than $\frac{1}{2}\min(\rho_{1\max}, \rho_{2\max}) = .0405$, then the upper bound between the samples of y_{p1} or y_{p2} are smaller than $\min(\rho_{1\max}, \rho_{2\max})$ and the system is stable. Table 1 shows the less conservative results in the literature and our τ_{MATI} for comparison. Note that the value of τ_{MATI} for a (stochastic) uniform inter-sampling time distribution given by [8] is less conservative than τ_{MATI} given by Theorem 3. However, for a fair comparison our result should be compared to the stochastic arbitrary inter-sampling time distribution given by [8]. If we can send the measurements of y_{1p} and y_{2p} in one packet then $\tau_{MATI} = \min(\rho_{1\max}, \rho_{2\max}) = 0.081$, because the requirement for the stability that the consecutive samplings of y_{1p} and y_{2p} are smaller than $\rho_{1\max}$ and $\rho_{2\max}$ respectively. As expected, in the presence of delays the value of τ_{MATI} decreases and, for example, when the maximum delay is 0.03 then $\rho_{1\max} = 0.058$ and $\rho_{2\max} = 0.087$.

4 Wireless NCS QoS

The framework that we have developed so far, provides conditions to guarantee QoP of the control system in terms of exponential stability. We now focus on the QoS that the network should provide to obtain the desired QoP at the application layer.

Consider again the system depicted in Fig. 1. We assume that the overall system consists of ℓ connections, where a *connection* is a path between a sampler and its corresponding hold. From the results in previous sections, we can find constants $\gamma_{i\max}, \tau_{i\max}, i \in \{1,2,...\}$ such that[1]

$$s^i_{k+1} - s^i_k \leq \gamma_{i\max}, \qquad \tau^i_k \leq \tau_{i\max}, \tag{26}$$

[1] The results in this chapter require explicit bounds on the sampling intervals. Hence, we use the definition of sampling-delay sequences as in (4) for SISO and MIMO cases. Note that $\rho_{i\max} = \gamma_{i\max} + \tau_{i\max}$. Moreover, for simplicity we assume the lower bound on the delay is zero.

for which exponential stability of all subsystems is guaranteed. Suppose now that the upper bounds on the sampling intervals of all connections, $\gamma_{i\max}, i \in \{1, \cdots \ell\}$, are given, then using Theorem 1 or Theorem 2 for the SISO case and Theorem 3 or 4 for the MIMO case (with $\tau_{i\min} = 0$), we can find upper bounds $\tau_{i\max}$ for the total delay in each connection so that exponential stability can be guaranteed. The main question addressed in this section is whether or not the network can deliver all packets for all connections before their *deadlines* $\tau_{i\max}$.

To answer this question we will employ results in real-time scheduling [4]. In real-time scheduling, different jobs are released periodically or lower bounds between release times are given. In the most basic setting, one shared resource services different jobs and servicing a job takes a certain amount of time. Each job should be completed before a deadline and if all the timing requirements can be met, then the set of jobs is said to be *schedulable*. In the context of real-time computation, typically a processor is a shared computation resource and jobs are corresponds to computing tasks. While computation resources can also be shared in NCSs; for example a processor can be used to implement two controllers [9], this will not pursue here and we assume that the computation delay is negligible. In the context of NCSs, the shared resource typically is a network shared between different nodes.

Job i refers to transmitting a packet from the source to the destination of connection i and the time required to service a job is the time needed to transfer a packet, which we call the transmission time. Suppose that $\gamma_{i\max}, \tau_{i\max}$ are given such that the stability LMIs are satisfied. If the set of "jobs" are schedulable with deadlines $\tau_{i\max}$, and release times greater than $\gamma_{i\min}$ for every $i \in \{1, \cdots, \ell\}$, then the completion of the job i is guaranteed before $\tau_{i\max}$. Hence the corresponding sampling-delay sequence satisfies (26) and consequently stability of all subsystems connected to the network is guaranteed.

4.1 Types Of Real-Time Scheduling

Two main types of scheduling can be found in the literature: non-preemptive and preemptive. In non-preemptive scheduling, a running service will not be interrupted to service a higher priority job. On the contrary, in preemptive scheduling, as soon as a job with a higher priority is released, the shared resource is allocated to the higher priority job and the current job with lower priority is interrupted. Preemptive scheduling is suitable for computation resource sharing, but cannot be used on communication networks for which access to the network cannot be granted to a new transmission until the current transmission is completed.

There are two main priority assignment schemes: static and dynamic. A static priority is fixed and set a priori, so it can be stored in a table. Static scheduling is simple, yet it is very inflexible to changes, failures, and often it under-utilizes the shared resources [4]. When scheduling decisions are based on the current decision variables, we have what is called a dynamic scheduling. Dynamic priority assignment is more difficult to implement because priorities change over time and they should be computed online; however, a dynamic priority assignment is generally more flexible and efficient. In the following we summarize the most common scheduling policies and we refer the readers to [4] for more details.

First-Come First-Serve (FCFS) scheduling. This policy serves the oldest request first so that resource allocation is based on the order of request arrivals. This policy is generally not suitable for control application because it may serve a job with longer deadline over a job with shorter deadline.

Round-Robin (RR) scheduling. This is a static algorithm in which a fixed time slot is dedicated to each node. This policy is simple and effective when: all nodes are synchronized, they have data most of the time, and the network structure is fixed so that no new node joins after the time slots are assigned to the nodes. If, for any reason, a node loses its turn, no matter how close its deadline is, it should wait until its next allocated slot.

Deadline Monotonic (DM) scheduling. This *static* policy allocates the resource to nodes according to their deadlines. A task with the *shortest deadline*, (smallest $\tau_{i\max}$) is assigned the highest priority. For example if $\tau_{1\max} = 3$ and $\tau_{2\max} = 4$ then jobs of source one will always have higher priority over jobs of source two.

Earliest Deadline First (EDF) scheduling. EDF is a dynamic algorithm that assigns priorities to jobs according to their *absolute deadlines*, which are the times remaining to miss the deadline. A job with the earliest absolute deadline, $(t_{il} + \tau_{i\max} - t)$ will have the highest priority, where t_{il} is the last sampling time of connection i and t is the current time. Again consider job one with $\tau_{1\max} = 3, t_{1l} = 2$ and job two with $\tau_{2\max} = 4, t_{2l} = 0$, which means that job one must be completed before time 5 and job two must be completed before time 4. In this case, if both nodes have a job ready to be service at time 3, then job two will be served because its absolute deadline is smaller (in spite of the fact that its $\tau_{i\max}$ is larger).

Each of these scheduling policies can be easily implemented on computation resources, but scheduling policies for shared communication resources depend on the specific network. FCFS is not easily implementable on wireless networks; however, RR, DM, and EDF are implementable on particular wireless networks as we shall see in Section 5. Of all the policies discussed, the most desirable policy is EDF because, as a dynamic algorithm, is most adaptable to network conditions. The disadvantage of EDF is that the priorities are a function of time and should be updated periodically, which require spending additional computation [11].

4.2 Scheduling Test To Guarantee QoS

The core of scheduling analysis is a scheduling test, which determines if a particular scheduling policy can guarantee that the tasks will be serviced, even under worst case condition. When this happens, we say that the tasks are *schedulable under the policy*. Our focus here will be on EDF scheduling. The deadline to finish job $i \in \{1, \cdots, n\}$ is denoted by D_i, the lower bound between consecutive job release times is denoted by T_i, and the time to service job i is denoted by C_i. If the conditions in the next theorem hold for a given set of jobs, then the set of jobs is schedulable under EDF policy. This means that the maximum delay experienced by job i, from the time it is released to the resource until the time that it is serviced, is always smaller than its deadline D_i. This delay consists of the wait time until the jobs gets service plus the service time. Note that the wait time depends greatly on the scheduling policy.

Theorem 5 ([26]). *A set of connections $(T_i, C_i, D_i), i \in \{1, \cdots, n\}$ is schedulable over a network under a (non-preemptive) EDF scheduling policy if and only if the following inequalities hold:*

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1, \tag{27}$$

$$\sum_{i=1}^{n} \lfloor \frac{t - D_i}{T_i} \rfloor^{+} C_i + C_{\max} \leq t, \qquad \forall t \in \cup_{i=1}^{n} S_i, \tag{28}$$

where $C_{\max} := \max_i C_i$

$$S_i := \left\{ D_i + hT_i : h = 0, 1, \cdots, \lfloor \frac{d_{\max} - d_i}{T_i} \rfloor \right\},$$
$$d_{\max} := \max \left\{ D_1, \cdots, D_n, \frac{\sum_{i=1}^{n} (1 - D_i/T_i) C_i + C_{\max}}{1 - \sum_{i=1}^{n} C_i/T_i} \right\},$$

$\lfloor . \rfloor$ *is a floor function,* $\lfloor x \rfloor^{+} := \lfloor x + 1 \rfloor$ *for* $x \geq 0$ *and zero otherwise.* □

To apply the results in Section 2 or 3 to our problem, we associate a job i with a packet transmission in connection i. We set $D_i = \tau_{i\max}$ to be the deadline for the packets in connection i, $T_i = \gamma_{i\min}$ to be the minimum time between consecutive transmissions, and C_i to be the transmission time in that same connection.

Corollary 1. *Assume that the sampling intervals satisfy $\gamma_{i\min} \leq s^i_{k+1} - s^i_k \leq \gamma_{i\max}$, and* (27)-(28) *hold for $D_i = \tau_{i\max}$, $T_i = \gamma_{i\min}$, and C_i equal to the transmission time of the connection i. If $\gamma_{i\max}$ and $\tau_{i\max}$ satisfy the stability conditions in Section 2 or 3 (where $\rho_{i\max} = \gamma_{i\max} + \tau_{i\max}$), then all subsystems connected to the network are exponentially stable.*

This corollary formally verifies the design specification that end to end delays must be smaller than $\tau_{i\max}$. Without this type of scheduling analysis, one has to rely on extensive testing to find rare events that may destabilize the system.

5 Implementation Considerations

In this section, we discuss the implementation of EDF scheduling on wireless NCSs. In particular, we consider wireless LAN networks governed by the IEEE 802.11 set of standards. These standards use a distributed coordination function (DCF) or a point coordination function (PCF) for Medium Access Protocol (MAC). Based on CSMA/CA protocol, DCF uses random backoff in the event of a collision. In wireless NCSs, short and periodic packets are sent frequently, so DCF is not suitable since its throughput is high only with light bursty traffic. However, PCF can be implemented to ensure high throughput by a polling mechanism to eliminate collisions. A node called the Access Point (AP), grants permission to nodes to transmit. There are three priority levels of mixed traffic [23]:

- **Level 1.** contains time critical aperiodic data that is bursty and cannot bear any loss. Retransmission is required when the former transmission is unsuccessful.
- **Level 2.** includes time critical periodic data that can tolerate some loss.
- **Level 3.** consists of noncritical data, which require data integrity, i.e., no loss is allowed. Retransmission is always implemented when there is data loss.

Level 1 data has the highest and level 3 has the lowest priority. NCS measurement and control packets are level 2 data and the focus of this paper. For simplicity, we assume that all data is level 2; however, the extension to the general case is straightforward. For implementation, Ye et al. [23] proposed a Centrally Maintained Polling Table (CMPT) to poll stations to schedule the level 2 data. The table maintains globally known network induced errors[2] for each connection which is defined as $e_i := z_i(t) - y_i(t)$. The CMPT knows $z_i(t)$ since it was broadcasted on the network, but not $y_i(t)$. The authors proposed to estimate $y_i(t)$ if $t \neq s^i_k$, $\forall k \in \mathbb{N}$ and otherwise update it with the true broadcasted value. Ye et al. [23] proposed several algorithms to handle scheduling and grant node permission to send data.

The implementation of our method can be similar to the method proposed by Ye et al. [23]; however, instead of error, our method employs *timers*. For each connection i, we employ a timer r_i that keeps track of how much time has elapsed since the last time permission was granted to node i and transmission was successful. Note that since the receiver node confirms a successful transmission by sending acknowledgment (ACK), the AP node is aware of possible packet dropout. As soon as the AP node grants permission to a node, its corresponding timer is reset to zero. The timer values are maintained in CMPT and are globally known. When the AP senses that the wireless network is idle, it gives permission to the node whose timer is closest to its deadline $\gamma_{i\max}$ (if $r_i \geq \gamma_{i\min}$).

In case of packet dropout, the AP does not get an ACK packet from the destination node, and again grants the permission to the same node. The only required assumption is that the network designer must know an upper bound on the maximum number of consecutive packet dropouts in the network. With that knowledge, it is possible to find the upper bound for the sampling intervals (see Remark 1 and also the example in Section 5.1).

[2] In Section 3 we defined $z_i := y_i(s^i_k)$, $t \in [t^i_k, t^i_{k+1})$, but since Ye et al. [23] assume that the delay is negligible, this would correspond to $z_i(t) := y_i(s^i_k)$, $t \in [s^i_k, s^i_{k+1})$.

5.1 Example

We consider the example of a motion control system for sheet control in a printer paper path from [3]. The system consists of several pinches or rollers, driven by motors, to move papers through the printer. Motor controllers are implemented on the AP node and the position and velocity measurements are sent through a wireless LAN network. The motors are directly connected to the AP node. Each subsystem (a single motor-roller pair) can be modeled as $\ddot{x}_s = \frac{nr_P}{J_M+n^2J_P}u$, where $J_M = 1.95 \times 10^5$ kg/m^2 is the inertia of the motor, $J_P = 6.5 \times 10^5$ kg/m^2 is the inertia of the pinch, $r_P = 14 \times 10^{-3}$ m is the radius of the pinch, $n = 0.2$ is the transmission ratio between motor and pinch and u is the motor torque. Each subsystem can be presented with the state space of the form $\dot{x} = Ax + B_u u$ with

$$x = \begin{bmatrix} x_s \\ \dot{x}_s \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 \\ \frac{nr_P}{J_M+n^2J_P} \end{bmatrix}. \tag{29}$$

We use the state feedback control $K = -\begin{bmatrix} 50 & 1.18 \end{bmatrix}$ to control the motors. We assume that for each subsystem, the AP node needs 0.1 ms to read a measurement packet from its buffer, decode the data, calculate the control command, and apply it to the motor. Moreover, we assume that it takes 1 ms between the AP consecutive sent permissions to nodes (for this example, sending ACK is not necessary because the receiver of all measurement packets is the AP node itself and it knows if the data was dropped or corrupted).

In traditional control system design, one often ignores the effect of network delays and selects a constant sampling times that are sufficiently fast so that sampling can be ignored. By checking the condition

$$eig(\begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix} e^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} h}) < 1, \quad B := B_u \times K, \tag{30}$$

on a tight grid of h, we can show that the closed loop system remains stable for any *constant* sampling interval smaller than 48 ms, and becomes unstable for larger constant sampling intervals. So a designer who follows traditional design guidelines may choose the sampling interval equal 12 ms, which is 4 times faster than the threshold beyond which stability is lost. A key question at this point is: how many motors can be controlled given this architecture. The answer to this question depends on the designer experience and judgment. A conservative designer would choose $n = 6$ to guarantee a total bus load equal to 50% (the total bus load is defined as $\sum_i \frac{C_i}{T_i} \times 100\%$) whereas an aggressive designer would choose up to $n = 11$ so that the bus load remains strictly lower than 100% (91.7% in this case).

In the following, we present our proposed systematic design process. The closed loop subsystems can be modeled as a SISO delay impulsive system given by (2) with A, B defined in (29) and (30). Then we determine the set of pairs $(\gamma_{i\max}, \tau_{i\max})$ for which the system would be exponentially stable, based on Theorem 1. This set is

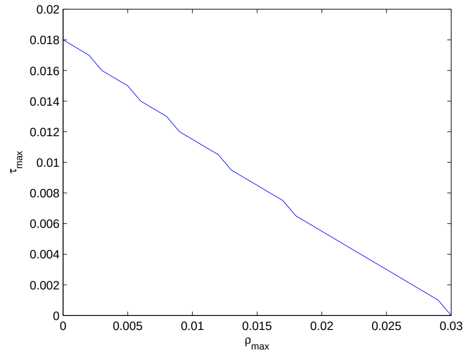

Fig. 9 Admissible set of variable sampling-delay sequences for a single motor-pinch subsystem is any sampling interval and delay sequence that belong to the triangle consists of the horizontal and vertical axis and the blue line.

shown in Fig. 9. To compare with the first approach, we choose the sampling times constant and equal to 12 ms.

We consider two scenarios. First we assume there is no packet dropout so $\gamma_{i\,\mathrm{min}} = \gamma_{i\,\mathrm{max}} = 12$ ms. Based on Fig. 9, for this choice, stability of the subsystems are guaranteed for delays smaller than 10 ms. At the scheduling level, we determine how many subsystems can share the network so that the total delay in each loop remains smaller than 10 ms. To do so, we test the conditions in Theorem 5 with $T_i = 12$, $C_i = 1$, $D_i = 10 - 0.1 = 9.9$ for different values of n. It turns out that the conditions are satisfied for up to $n = 9$. This result indicates that 9 pinch-motor subsystems can share the network while the stability of all subsystems is guaranteed. Note that for $n = 10, 11$ the delay can be larger than 10 ms for some corner cases that may not be easily captured by simulation and testing (the worse case delay occurs when all the sensors send data at the same time). By following the proposed design procedure, we can avoid very conservative choices (e.g., n=6) or choices that lead to unsafe behavior of the subsystems.

In a second scenario, we assume that at most 3 consecutive packet dropout are possible. For this case $\gamma_{i\,\mathrm{min}} = 12$, and $\gamma_{i\,\mathrm{max}} = 15$. Based on the analysis results depicted in Fig. 9, for this choice, stability of the subsystems are guaranteed for delays smaller than 8 ms. By testing the conditions in Theorem 5 with $T_i = 15$, $C_i = 1$, $D_i = 8 - 0.1 = 7.9$ for different values of n, it turns out that the conditions are satisfied for up to $n = 7$.

6 Conclusions And Future Work

We showed that delay impulsive systems are a natural framework to model wired or wireless NCSs with variable sampling intervals and delays and possible packet dropouts. We employed discontinuous Lyapunov functionals to characterize admissible sampling intervals and delays such that exponential stability of wireless NCS is guaranteed. We defined exponential stability as a minimal QoP, and we found the requirements to guarantee QoP at the application level of wireless NCSs. Then we provided a set of conditions for EDF scheduling, that if satisfied, ensures the desired QoS for the wireless network required to provide QoP. We also discussed implementation considerations to implement EDF scheduling (or other dynamic scheduling policies, on a wireless network.

An important topic for future research is the controller and network codesign. The framework we developed for the analysis of wireless NCS can provide the foundation for the codesign of the network and controller. It is also important to consider other QoP metrics such as robust design as measured by the H_∞ norm or the H_2 norm. In general, faster sampling improves QoP of the control systems; however, QoS of the network may decrease due to higher traffic of the network. The tradeoff between the performance of the control systems at the application layer and the behavior at other layers of wireless NCS is another important topic.

References

[1] T. Arampatzis, J. Lygeros, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proc. of the 13th Mediterranean Conf. on Control and Automation*, volume 3, pages 719–724, 2005.

[2] M. S. Branicky, S. M. Phillips, and W. Zhang. Stability of networked control systems:explicit analysis of delay. In *Proc. of the 2000 Amer. Contr. Conf.*, volume 4, pages 2352–2357, June 2000.

[3] M. Cloosterman, N. van de Wouw, W. Heemels, and H. Nijmeijer. Robust stability of networked control systems with time-varying network-induced delays. In *Proc. of the 45th Conf. on Decision and Contr.*, Dec. 2006.

[4] F. Cottet, J. Delacroix, C. Kaiser, and Z. Mammeri. *Scheduling in real-time systems*. Willy, 2002.

[5] E. Fridman. A new Lyapunov technique for robust control of systems with uncertain non-small delays. *J. of Math. Contr. and Info.*, 22(3), Nov. 2005.

[6] E. Fridman, A. Seuret, and J. P. Richard. Robust sampled-data stabilization of linear systems: an input delay approach. *Automatica*, 40(8):1441–1446, Aug. 2004.

[7] J. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proc. of the IEEE*, 95(1):138 – 162, Jan 2007.

[8] J. P. Hespanha and A. R. Teel. Stochastic impulsive systems driven by renewal processes. *Proc. of the 17th Int. Symp. on the Mathematical Theory of Networks and Syst.*, pages 606–618, July 2006.

[9] F.-L. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks. *IEEECSM*, 21(1):66–83, Feb. 2001.

[10] X. Liu and A. Goldsmith. Wireless network design for distributed control. In *Proc. of the 43th Conf. on Decision and Contr.*, volume 3, pages 2823– 2829, Dec. 2005.

[11] P. Naghshtabrizi. *Delay impulsive systems: A framework for modeling networked control systems*. PhD thesis, University of California at Santa barbara, 2007.

[12] P. Naghshtabrizi and J. P. Hespanha. Designing observer-based controller for network control system. In *Proc. of the 44th Conf. on Decision and Contr.*, volume 4, pages 2876–2880, June 2005.

[13] P. Naghshtabrizi, J. P. Hespanha, and A. R. Teel. On the robust stability and stabilization of sampled-data systems: A hybrid system approach. In *Proc. of the 45th Conf. on Decision and Contr.*, pages 4873–4878, 2006.

[14] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilvert. Trends in automotive communication systems. *Proc. of the IEEE*, 93(6):1204–1223, June 2005.

[15] D. Nesic and A. R. Teel. Input-output stability properties of networked control systems. *IEEE Trans. on Automat. Contr.*, 49(10):1650–1667, Oct. 2004.

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

[17] P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Trans. on Automat. Contr.*, 49(8):1292–1302, Aug. 2004.

[18] P. Seiler and R. Sengupta. An H_∞ approach to networked control. *IEEE Trans. on Automat. Contr.*, 50(3):356–364, Mar. 2005.

[19] M. Tabbara, D. Nesic, and A. R. Teel. Input- output stability of wireless networked control systems. In *Proc. of the 44th Conf. on Decision and Contr.*, pages 209–214, dec 2005.

[20] G. C. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. In *Proc. of the 1999 Amer. Contr. Conf.*, volume 4, pages 2876–2880, June 1999.

[21] G. C. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *IEEE Trans. on Contr. Syst. Tech.*, 10(3):438–446, May 2002.

[22] F. Xia, L. Ma, C. Peng, Y. Sun, and J. Dong. Cross-layer adaptive feedback scheduling of wireless control systems. 8(7):4265 – 4281, 2008.

[23] H. Ye, G. C. Walsh, and L. G. Bushnell. Real-time mixed -traffic wireless networks. *IEEE Trans. on Industrial Electronics.*, 48(5):883–890, Oct 2001.

[24] D. Yue, Q.-L. Han, and C. Peng. State feedback controller design for networked control systems. *IEEE Trans. on Automat. Contr.*, 51(11):640–644, Nov. 2004.

[25] D. Yue, Q. L. Han, and J. Lam. Network-based robust H_∞ control of systems with uncertainty. *Automatica*, 41(6):640–644, June 2005.

[26] Q. Zheng and K. G. Shin. On the ability of establishing real-time channels in point-to-point packet-switched networks. *IEEE Trans. on Communications*, 42(2/3/4):1096–1105, Feb. 1994.