A SURVEY OF RECENT RESULTS IN NETWORKED CONTROL SYSTEMS

João P. Hespanha Payam Naghshtabrizi Yonggang Xu

Department of Electrical and Computer Engineering,
Univ. of California, Santa Barbara, CA 93106-9560
Tel: +1 (805) 893-7067 Fax: +1 (805) 893-3262

Abstract

Networked Control Systems (NCSs) are spatially distributed systems for which the communication between sensors, actuators, and controllers is supported by a shared communication network. In this paper we review several recent results on estimation, analysis, and controller synthesis for NCSs. The results surveyed address channel limitations in terms of packet-rates, sampling, network delay and packet dropouts. The results are presented in a tutorial fashion, comparing alternative methodologies.

I. 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.

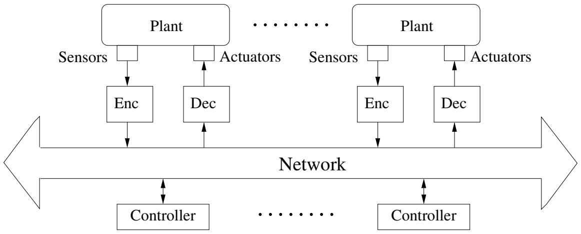


Fig. 1. General NCS architecture.

The use of a multi-purpose shared network to connect spatially distributed elements results in flexible architectures and generally reduces installation and maintenance costs. Consequently, NCSs have been finding application in a broad range of areas such as mobile sensor networks [52], remote surgery [33], haptics collaboration over the Internet [17, 19, 59], and automated highway systems and unmanned aerial vehicles [57, 58]. However, the use of a shared network—in contrast to using several dedicated independent connections—introduces new challenges and Murray et al. [39] identify *control over networks* as one of the key future directions for control.

NCSs lie at the intersection of control and communication theories. Traditionally, control theory focuses on the study of interconnected dynamical systems linked through "ideal channels", whereas communication theory studies the transmission of information over "imperfect channels". A combination of these two frameworks is needed to model NCSs. This survey is primarily written from a controls perspective and attempts to systematically address several key issues that make NCSs distinct from other control systems.

a) Band-limited channels: Any communication network can only carry a finite amount of information per unit of time. In many applications, this limitation poses significant constraints on the operation of NCSs. Examples of NCSs that are afflicted by severe communication limitations include unmanned air vehicles (UAVs) due to stealth requirements, power-starved vehicles such as planetary rovers, long-endurance energy-limited systems such as sensor networks, underwater vehicles, and large arrays of micro-actuators and sensors.

Inspired by Shannon's results on the *maximum bit-rate* that a communication channel can carry reliably, a significant research effort has been devoted to the problem of determining the *minimum bit-rate* that is needed to stabilize a linear system through feedback over a finite capacity channel [71, 18, 41, 9, 66, 4] (c.f., the survey [67] in this special issue). Recently, some progress has also been made in solving the finite-capacity stabilization problem for nonlinear systems [42, 24] and for linear systems with unknown parameters [64]. In [56], stability conditions are derived based on *anytime information*, which quantifies the "time value" of data bits. Performance limitations of feedback over finite capacity memory-less channels are addressed in [31], which obtains a general extension of Bode's integral inequality.

This material is based upon work supported by the National Science Foundation under Grants No. CCR-0311084 and ECS-0242798.

Most of the results discussed in this survey are motivated by the observation that, in most digital networks, data is transmitted in atomic units called *packets* and sending a single bit or several hundred bits consumes the same amount of network resources. For example, every fixed-size ATM (Asynchronous Transfer Mode) cell consists of a 40-bit header and a 384-bit data field, an Ethernet IEEE 802.3 frame has a 112 or 176-bit header and a data field that must be at least 368-bit long, and each Bluetooth time slot carries 625 bits leaving at least 499 bits for data payload [62, 54]. This observation leads to an alternative view of band-limited channels, in which a channel can transmit a finite number of packets per unit of time (packet-rate), but each packet can carry a large number of bits (possibly infinitely many). Although from Shannon's point of view these channels have infinite capacity, the closed-loop system stability and performance is still limited by the network. This perspective is prevalent in most of the results that we discuss in this survey, which generally neglect the quantization effects imposed by digital networks.

b) 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. A significant number of results have attempted to characterize a maximum upper-bound on the sampling interval for which stability can be guaranteed. These results implicitly attempt to minimize the *packet-rate* that is needed to stabilize a system through feedback.

c) 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. Reliable transmission protocols, such as TCP, guarantee the eventual delivery of packets. However, these protocols are not appropriate for NCSs since the re-transmission of old data is generally not very useful.

d) Systems architecture: Figure 1 shows the general architecture of an NCS. In this figure, *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, *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. We do not explicitly represent these blocks because the boundaries between a digital controller and encoder/decoder blocks are often blurry.

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.

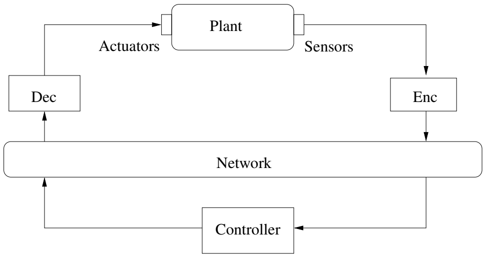


Fig. 2. Single-loop NCS.

Paper organization

Section II addresses the problem of state estimation over imperfect communication channels. Section III reviews a collection of results to determine the closed-loop stability of NCSs in the presence of network sampling, delays, and packet dropouts. Section IV addresses control synthesis methods for NCSs. To provide a unified view, we formulate all the results surveyed with consistent terminology and notation. Consequently, the statements of the theorems found in this survey often differ from the ones in the original papers. In some cases, we also combine results from multiple papers into single theorems.

Notation and basic definitions

Throughout the paper, $\mathbb{R}$ stands for real numbers and $\mathbb{N}$ for nonnegative integers. For a given matrix $A \in \mathbb{R}^{n\times n}$ and vector $x \in \mathbb{R}^n$, $\|x\| := \sqrt{x'x}$ denotes the Euclidean norm of x, $\|A\|$ the corresponding induced matrix norm, and $\lambda(A)$ the set of eigenvalues of A. Random variables are generally denoted in boldface. For a random variable $\mathbf{y}$, $\mathrm{E}[\mathbf{y}]$ stands for the expectation of $\mathbf{y}$ [53]. For deterministic systems we use standard notions of stability, e.g., from [21]. General stability notions for stochastic systems can be found in [22], and our definitions are compatible with [10]: We say that a random process $\mathbf{x}_k$, $k \in \mathbb{N}$ is *almost surely stable* if $\Pr\left(\sup_{k\in\mathbb{N}} \|\mathbf{x}_k\| < \infty\right) = 1$; that it is *stable in the m-th moment* if $\sup_{k\in\mathbb{N}} \mathrm{E}\,\|\mathbf{x}_k\|^m < \infty$; that it is *asymptotically stable in the m-th moment* if $\lim_{k\to\infty} \mathrm{E}\,\|\mathbf{x}_k\|^m = 0$; and that it is *exponentially stable in the m-th moment* if there exist constants $\alpha, \beta > 0$ such that $\mathrm{E}\,\|\mathbf{x}_k\|^m < \alpha\,\mathrm{E}\,\|\mathbf{x}_0\|^m e^{-\beta k}$, $\forall k \in \mathbb{N}$. When $m = 2$, we simply say that the random process is (asymptotically/exponentially) *mean-square stable*.

II. Estimation over lossy networks

This section addresses the problem of estimating the state of a remote plant based on measurements carried through a lossy network. State estimation over networks is important in applications such as remote sensing, space exploration, and sensor networks. It is also a crucial component of certainty equivalence NCS controllers that construct control signals based on state estimates of a remote plant. In several NCS scenarios, certainty equivalence controllers are not optimal, but are still of a great practical interest due to the difficulty in designing optimal controllers.

In this and the subsequent sections, we assume that the network can be viewed as a channel that can carry real numbers without distortion, but that some of the messages can be lost. This is appropriate when the number of bits in each data packet is sufficiently large so that quantization effects can be ignored, but packet dropouts cannot.

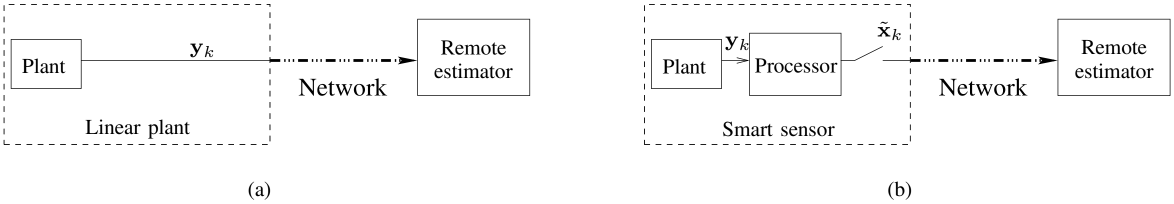


Fig. 3. Scenarios for state estimation over a network. In (a) the raw sensor measurements are sent to a remote estimator, whereas in (b) the measurements are processed locally before transmission.

We consider two scenarios for state-estimation over networks. In the one depicted in Fig. 3(a), every raw sensor measurement $\mathbf{y}_k$ is sent to the remote estimator via the network, but may not arrive there if there is a packet dropout [32, 61, 60]. Alternatively, in the scenario shown in Fig. 3(b), the raw sensor measurements may be processed locally and at each time instant a decision is made on whether or not it is worth to send data to the remote estimator [72, 75, 74]. This option is motivated by the desire to reduce network traffic and to make the estimate robust with respect to packet dropouts.

We restrict out attention to linear time-invariant (LTI) plants with Gaussian measurement noise and disturbance:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{w}_k, \qquad \mathbf{y}_k = C\mathbf{x}_k + \mathbf{v}_k, \qquad \forall k \in \mathbb{N},\ \mathbf{x}_k, \mathbf{w}_k \in \mathbb{R}^n,\ \mathbf{y}_k, \mathbf{v}_k \in \mathbb{R}^p, \tag{1}$$

where the initial state $\mathbf{x}_0$ is zero-mean Gaussian with covariance matrix Σ, and the zero-mean Gaussian white noises $\mathbf{w}_k$ and $\mathbf{v}_k$ are mutually independent with covariance matrices $R_w \geq 0$ and $R_v > 0$, respectively. It is assumed that (C, A) is detectable, and (A, R_w) is stabilizable.

A. *Optimal estimation for Bernoulli dropouts*

Consider the architecture in Fig. 3(a) and a lossy channel modeled by a stochastic process $\boldsymbol{\theta}_k \in \{0, 1\}$, $\forall k \in \mathbb{N}$ with the understanding that $\boldsymbol{\theta}_k = 1$ signifies that the measurement $\mathbf{y}_k$ sent at time k reaches its destination and that $\boldsymbol{\theta}_k = 0$ when it does not. It is generally assumed that the dropout process $\boldsymbol{\theta}_k$ is statistically independent of $\mathbf{x}_0$, $\mathbf{w}_k$, and $\mathbf{v}_k$.

The optimal estimate of $\mathbf{x}_k$, $k \in \mathbb{N}$ given all the data $\{\boldsymbol{\theta}_\ell, \forall\, \ell \leq k-1\} \cup \{\mathbf{y}_\ell : \boldsymbol{\theta}_\ell = 1, \forall\, \ell \leq k-1\}$ that is known to the remote estimator at time k is given by

$$\hat{\mathbf{x}}_{k|k-1} = \mathrm{E}\left[\mathbf{x}_k \;\middle|\; \boldsymbol{\theta}_\ell, \forall\, \ell \leq k-1;\ \mathbf{y}_\ell,\ \forall \ell \leq k-1 \text{ s.t. } \boldsymbol{\theta}_\ell = 1\right]. \tag{2}$$

This estimate can be computed recursively using the following time-varying Kalman filter (TVKF) [20]:

$$\hat{\mathbf{x}}_{0|-1} = 0, \tag{3a}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \boldsymbol{\theta}_k F_k(\mathbf{y}_k - C\hat{\mathbf{x}}_{k|k-1}), \qquad \forall k \in \mathbb{N}, \tag{3b}$$

$$\hat{\mathbf{x}}_{k+1|k} = A\hat{\mathbf{x}}_{k|k}, \tag{3c}$$

with the gain matrix F_k calculated recursively as follows

$$\begin{aligned} P_0 &= \Sigma \\ F_k &= P_k C'(CP_kC' + R_v)^{-1}, & \forall k \in \mathbb{N}, \\ P_{k+1} &= AP_kA' + R_w - AF_k(CP_kC' + R_v)F_k'A'. \end{aligned}$$

Each P_k corresponds to the estimation error covariance matrix

$$P_k = \mathrm{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})'], \quad \forall k \in \mathbb{N}.$$

Sinopoli et al. [60] study the performance of this Kalman filter when $\boldsymbol{\theta}_k$ is a Bernoulli process with probability of dropout ($\boldsymbol{\theta}_k = 0$) equal to $p \in [0, 1)$. They show the existence of a critical value p_c for the dropout rate p, above which the estimation error covariance becomes unbounded:

Theorem 1 ([60]): Assume that the dropout process $\boldsymbol{\theta}_k \in \{0, 1\}$ is Bernoulli with dropout probability

$$p := \Pr[\boldsymbol{\theta}_k = 0] \in [0, 1), \quad \forall k \in \mathbb{N}.$$

There exists a critical value $p_c \in (0, 1]$ with the property that

1) for every $p \geq p_c$, there is some $P_0 \geq 0$ for which $\mathrm{E}[P_k]$, $k \in \mathbb{N}$ is unbounded along solutions to the TVKF (3);
2) for every $p < p_c$ and every $P_0 \geq 0$, $\mathrm{E}[P_k]$, $k \in \mathbb{N}$ remains uniformly bounded along solutions to the TVKF (3).

Furthermore, the critical value p_c satisfies $\underline{p} \leq p_c \leq \bar{p}$, where the upper-bound is given by

$$\bar{p} = \frac{1}{(\max\{|\lambda(A)|\})^2}, \tag{4}$$

and the lower-bound is given by the solution to the following (quasi-convex) optimization problem

$$\underline{p} = \max\{p \geq 0 : \Psi_p(Y, Z) > 0,\ 0 \leq Y \leq I \text{ for some } Y, Z\},$$

where

$$\Psi_p(Y, Z) = \begin{bmatrix} Y & \sqrt{1-p}(YA+ZC) & \sqrt{p}YA \\ \sqrt{1-p}(A'Y+C'Z') & Y & 0 \\ \sqrt{p}A'Y & 0 & Y \end{bmatrix}, \quad \forall Y, Z \in \mathbb{R}^{n\times n}. \qquad \square$$

In some special cases (such as when the matrix C is invertible) the upper-bound in (4) is tight in the sense that $p_c = \bar{p}$ but in general this may not be the case.

Liu and Goldsmith [29] consider a setting similar to that of [60], but extend the results to allow partial observation losses. In [29], each $\mathbf{y}_k$ is split into two elements that are encoded separately and sent over different (wireless) channels in which packets may be dropped independently.

B. Multi-sensor plants

Several variations of the problem formulated above have been considered. Matveev and Savkin [32] consider N sensors, each independently sending its measurements to the estimator with some delay. In practice, this corresponds to the following plant model

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{w}_k, \qquad \mathbf{y}_{\nu,k} = C_\nu \mathbf{x}_k + \mathbf{v}_{\nu,k},\ \ \nu \in \{1, \cdots, N\}, \qquad \forall k \in \mathbb{N}, \tag{5}$$

where $\mathbf{y}_{\nu,k}$ denotes the measurement collected by sensor ν at time k. Assuming that the measurement $\mathbf{y}_{\nu,\ell}$ suffers a random delay of $\boldsymbol{\tau}_\nu(\ell)$, the optimal estimate of $\mathbf{x}_k$ that the receiver can construct at time k is given by

$$\hat{\mathbf{x}}_k = \mathrm{E}\left[\mathbf{x}_k \;\middle|\; \boldsymbol{\theta}_{\nu,\ell}, \forall\, \ell \leq k - \boldsymbol{\tau}_\nu(\ell);\ \mathbf{y}_{\nu,\ell},\ \forall \ell \leq k - \boldsymbol{\tau}_\nu(\ell) \text{ s.t. } \boldsymbol{\theta}_{\nu,\ell} = 1\right],$$

where $\boldsymbol{\theta}_{\nu,\ell} = 1$ if $\mathbf{y}_{\nu,\ell}$ reaches its destination, and $\boldsymbol{\theta}_{\nu,\ell} = 0$ otherwise. Matveev and Savkin [32] derive a recursive Kalman filter and provide conditions under which the estimation error process is almost surely stable. These conditions are given in terms of the observability of $\mathbf{x}_k$ for specific realizations of the process $\boldsymbol{\theta}_k$.

Gupta et al. [14] also consider the state-estimation of the plant (5), but they impose the constraint that only one sensor can use the channel at each time step. They develop a stochastic sensor selection algorithm that decides how to schedule the sensors' access to the network to minimize the error covariance. They also compute upper- and lower-bounds on the achievable error covariance matrices.

C. Reduced-computation estimation

Even though the plant (1) is time-invariant, the matrix gain F_k and the covariance matrix P_k of the optimal TVKF (2) do not converge to steady-state values as $k \to \infty$. Moreover, these matrices cannot be computed off-line because they depend on the whole dropout history $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_k\}$. Smith and Seiler [61] avoid this difficulty by pre-computing a finite set of gains to be selected according to the dropout history in the last τ time-steps. The resulting estimator is called a *finite loss history estimator* (FLHE) and has the following form:

$$\begin{aligned} \hat{\mathbf{x}}_{0|-1} &= 0, \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + F_{\boldsymbol{\eta}_k}(\mathbf{y}_k - C\hat{\mathbf{x}}_{k|k-1}), \qquad \forall k \in \mathbb{N}, \\ \hat{\mathbf{x}}_{k+1|k} &= A\hat{\mathbf{x}}_{k|k}, \end{aligned}$$

where $\boldsymbol{\eta}_k$ is an integer from 1 to 2^τ that encodes the dropout history $\{\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k-1}, \dots, \boldsymbol{\theta}_{k-\tau+1}\}$ in the last τ time-steps and $F_1, F_2, \dots, F_{2^\tau}$ are appropriately selected matrix gains with $F_{\eta_k} = 0$ whenever $\boldsymbol{\theta}_k = 0$. Smith and Seiler [61] model the evolution of $\boldsymbol{\eta}_k$ as a Markov chain, which allows them to consider correlated (bursty) dropouts and is an improvement over the Bernoulli drop model. For this model, they provide an optimal synthesis method for the gains $F_1, F_2, \dots, F_{2^\tau}$ and conditions for stability of the error process in terms of the convergence of a Riccati iteration. Although the FLHE is not optimal, simulations show that the variance of the estimation error is comparable to that of the optimal TVKF when τ is sufficiently large.

D. Estimation with local computation

The encoder-estimator scheme in Fig. 3(b) is motivated by the growing number of *smart sensors* with embedded processing units that are capable of local computation. In this context, Xu and Hespanha [74] investigate the benefits of pre-processing the measurements before transmission to the network. For the LTI plant (1), the smart sensor computes locally an optimal state estimate $\tilde{\mathbf{x}}_{k|k} = \mathrm{E}[\mathbf{x}_k | \mathbf{y}_\ell, \ell \le k]$ using the following stationary Kalman filter:

$$\tilde{\mathbf{x}}_{0|0} = 0, \qquad \tilde{\mathbf{x}}_{k+1|k+1} = A\tilde{\mathbf{x}}_{k|k} + F(\mathbf{y}_{k+1} - CA\tilde{\mathbf{x}}_{k|k}), \quad \forall k \in \mathbb{N}, \tag{6}$$

where $F := PC'(CPC' + R_v)^{-1}$ and $P > 0$ is the solution to the discrete-time Algebraic Riccati Equation

$$P = APA' + R_w - APC'(CPC' + R_v)^{-1}CPA'.$$

The smart sensor then transmits the local estimates $\tilde{\mathbf{x}}_{k|k}$ (instead of the raw measurements $\mathbf{y}_k$), which are used by the remote estimator to compute the optimal estimate $\hat{\mathbf{x}}_{k|k-1}$ of $\mathbf{x}_k$, $k \in \mathbb{N}$ given all the data $\{\boldsymbol{\theta}_\ell, \forall\, \ell \le k-1\} \cup \{\tilde{\mathbf{x}}_{\ell|\ell} : \boldsymbol{\theta}_\ell = 1, \forall\, \ell \le k-1\}$ successfully received up to time k:

$$\hat{\mathbf{x}}_{k|k-1} = \mathrm{E}\left[\mathbf{x}_k \mid \boldsymbol{\theta}_\ell, \forall\, \ell \le k-1;\ \tilde{\mathbf{x}}_{\ell|\ell}, \forall\, \ell \le k-1 \text{ s.t. } \boldsymbol{\theta}_\ell = 1\right].$$

This remote estimate can be computed recursively by

$$\hat{\mathbf{x}}_{0|-1} = 0, \tag{7a}$$

$$\hat{\mathbf{x}}_{k|k} = (1 - \boldsymbol{\theta}_k)\hat{\mathbf{x}}_{k|k-1} + \boldsymbol{\theta}_k\tilde{\mathbf{x}}_{k|k}, \quad k \in \mathbb{N}, \tag{7b}$$

$$\hat{\mathbf{x}}_{k+1|k} = A\hat{\mathbf{x}}_{k|k}. \tag{7c}$$

The main advantage of this solution is that each message $\tilde{\mathbf{x}}_{k|k}$ that successfully reaches the remote estimator encodes all the relevant information that can be extracted *from every raw measurement collected up to time k.* In general, this permits the stability of the error process for larger drop rates p than those permitted by the architecture in Fig. 3(a), in which the raw measurements are sent over the network.

Theorem 2 ([74]): Assume that $\boldsymbol{\theta}_k$ is a Bernoulli process with probability of dropout $p \in [0, 1)$. For every $m \ge 2$ and

$$p < \frac{1}{(\max\{|\lambda(A)|\})^m}, \tag{8}$$

the (remote) estimation error $\mathbf{e}_k := \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}$ computed by (7) is stable in the m-th moment. □

The bound in (8) is tight in the sense that the estimation error is not stable in the m-th moment for every $p > \frac{1}{(\max\{|\lambda(A)|\})^m}$. The critical value in (8) for mean-square stability ($m = 2$) matches the upper bound in (4) that is only known to be tight when the matrix C is invertible. Theorem 2 shows that this critical value can be achieved even when C is not full rank, as long as one uses the architecture in Fig. 3(b). This is obtained at the expense of shifting some computation to the smart sensor, which now needs an embedded processor. Moreover, each network packet now carries $\tilde{\mathbf{x}}_{k|k}$ instead of the raw measurement $\mathbf{y}_k$, which will generally require larger packets.

E. Estimation with controlled communication

To actively reduce network traffic, sensor measurements may not be sent to the remote estimator at every time step. This is known as *controlled communication* and has been investigated by Yook et al. [75], Xu and Hespanha [73, 74] to explore the trade-off between communication and estimation performance. It is also related to the concept of Lebesgue sampling introduced by Åström and Bernhardsson [1]. Controlled communication requires local processing at the sensor so we will focus our attention on the architecture in Fig. 3(b).

Yook et al. [75], Xu and Hespanha [73, 74] construct within the smart sensor the stationary Kalman filter (6) as well as a copy of the remote estimator (7). This implicitly assumes an erasure channel for which the smart sensor receives immediate feedback regarding the success of the transmission, which is required to be able to use $\boldsymbol{\theta}_k$ in implementing (7). To decide whether or not the local estimate $\tilde{\mathbf{x}}_{k|k}$ should be sent to the network, the smart sensor compares $\tilde{\mathbf{x}}_{k|k}$ with the (local copy) of the remote estimate $\hat{\mathbf{x}}_{k|k-1}$. Xu and Hespanha [74] make this decision stochastically, by selecting

$$\boldsymbol{\theta}_k = \begin{cases} 1 & \text{with prob. } \Lambda_k(\tilde{\mathbf{e}}_k) \\ 0 & \text{with prob. } 1 - \Lambda_k(\tilde{\mathbf{e}}_k) \end{cases} \tag{9}$$

where the random variable $\boldsymbol{\theta}_k$ is now determined by a random draw *inside the smart sensor* and specifies whether or not $\tilde{\mathbf{x}}_{k|k}$ is sent to the remote estimator. Its distribution is a function $\Lambda_k(\cdot) \in [0, 1]$ of the difference $\tilde{\mathbf{e}}_k := \tilde{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}$ between the local and the remote estimators. The function $\Lambda_k(\cdot)$ should be chosen so that it takes values close to one when the error $\tilde{\mathbf{e}}_k$ is large because this is an indication that the remote state estimate can be significantly improved by sending $\tilde{\mathbf{x}}_{k|k}$.

When the network introduces dropouts with probability $p \in [0, 1)$, independent of the plant state, one should replace (9) by

$$\boldsymbol{\theta}_k = \begin{cases} 1 & \text{with prob. } (1-p)\Lambda_k(\tilde{\mathbf{e}}_k) \\ 0 & \text{with prob. } 1 - (1-p)\Lambda_k(\tilde{\mathbf{e}}_k), \end{cases} \tag{10}$$

to account for the fact that a message only reaches the remote estimator if (i) the smart sensor decides to send it [which occurs with probability $\Lambda_k(\tilde{\mathbf{e}}_k)$] and (ii) the packet that carries the message is not dropped by the network, which happens with probability $1 - p$.

Unlike the "uncontrolled" Bernoulli dropout process discussed in the previous sections, neither the controlled process $\boldsymbol{\theta}_k$ in (9) nor the mixed controlled/uncontrolled process $\boldsymbol{\theta}_k$ in (10) is independent of the plant's state $\mathbf{x}_k$. In fact, combining (1), (6), (7) and (10), we conclude that the difference $\tilde{\mathbf{e}}_k := \tilde{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}$ between the local and the remote estimators evolves according to

$$\tilde{\mathbf{e}}_{k+1} = \begin{cases} \mathbf{d}_k & \text{with prob. } (1-p)\Lambda_k(\tilde{\mathbf{e}}_k) \\ A\tilde{\mathbf{e}}_k + \mathbf{d}_k & \text{with prob. } 1 - (1-p)\Lambda_k(\tilde{\mathbf{e}}_k), \end{cases}$$

where $\mathbf{d}_k := FCA(\mathbf{x}_k - \tilde{\mathbf{x}}_{k|k}) + F\mathbf{w}_k + \mathbf{v}_{k+1}$ is a zero-mean Gaussian process. The following discrete-time result is adapted from those in [74] for continuous-time processes.

Theorem 3: Assume that $\boldsymbol{\theta}_k$ is generated by (10) with $\Lambda_k(e) \to 1$ as $\|e\| \to \infty$. For every $m \geq 2$ and drop probability p satisfying (8), the (remote) estimation error $\mathbf{e}_k := \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}$ computed by (7) is stable in the m-th moment. □

Xu and Hespanha [73] consider the problem of *optimal* controlled communication for a network without packet dropouts. They make the simplifying assumption that the smart sensor can measure the whole state $\mathbf{x}_k$ and therefore it does not need to build a local estimate $\tilde{\mathbf{x}}_{k|k}$. The objective is to find the optimal "communication policy" that minimizes the following long-term average cost

$$J := \lim_{K \to \infty} \frac{1}{K} \operatorname{E}\left[\sum_{k=0}^{K-1} \|\mathbf{e}_{k+1}\|^2 + \lambda \boldsymbol{\theta}_k\right], \quad \lambda > 0, \tag{11}$$

which penalizes a linear combination of the remote estimation error variance $\operatorname{E}[\|\mathbf{e}_{k+1}\|^2]$ and the average packet sending rate $\operatorname{E}[\boldsymbol{\theta}_k]$. In this context, a *communication policy* should be understood as a rule that selects a distribution for the $\boldsymbol{\theta}_k$ based on all the measurements $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ available to the smart sensor, i.e., a family of functions $\bar{\Lambda}_k(\cdot)$ for which

$$\boldsymbol{\theta}_k = \begin{cases} 1 & \text{with prob. } \bar{\Lambda}_k(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k) \\ 0 & \text{with prob. } \bar{\Lambda}_k(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k), \end{cases}$$

Finding the optimal $\bar{\Lambda}_k(\cdot)$ that minimize (11) can be done using dynamic programming and value iteration. The optimal policy turns out to be stationary and it depends on the measurements only through the local estimation error $\tilde{\mathbf{e}}_k$. In particular, it can be expressed as in (9), but with

$$\Lambda_k(e) = \begin{cases} 0 & e \in \Omega \\ 1 & \text{otherwise}, \end{cases} \qquad \forall k \in \mathbb{N},\ e \in \mathbb{R}^n$$

for an appropriately chosen set $\Omega \subset \mathbb{R}^n$, which contains the origin. This corresponds to a deterministic controlled communication scheme, in which data communication is inhibited while $\tilde{\mathbf{e}}_k$ remains inside Ω and is triggered when the error exits this set. Xu and Hespanha [73] consider a more general case of networks that introduce a τ-step delay, but this does not significantly change the nature of the results.

III. Stability of NCSs with Sampling, Delay, and Packet Dropouts

This section addresses the stability of feedback loops that are closed over a network. This problem is motivated by scenarios in which sensors, controllers, and actuators are not co-located and use a shared network to communicate. As before, we ignore quantization and focus our attention on the effects of data sampling, network delay, and packet dropouts on the stability of the resulting closed-loop NCSs.

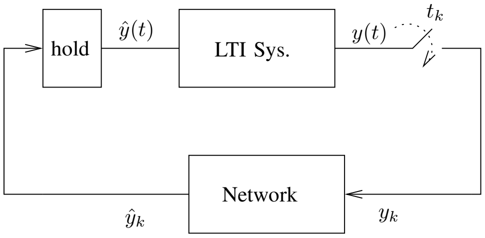


Fig. 4. One-channel feedback NCS with LTI plant/controller

A. Sampling and delay

The *one-channel feedback NCS* in Fig. 4 has been commonly used to investigate the effects of sampling and delay in the stability of NCSs. The LTI system encapsulates a linear time-invariant plant/controller pair modeled by the following continuous-time system

$$\dot{x} = Ax + B\hat{y}, \qquad\qquad y = Cx. \tag{12}$$

This one-channel feedback NCS can capture several NCS configurations. The signal y can be regarded as a vector of sensor measurements and $\hat{y}$ as the input to a continuous-time controller collocated with the actuators, as in Fig. 5(a) [57, 36]. Alternatively, $\hat{y}$ can be viewed as the input to the actuators and y as the desired control signal computed by a controller collocated with the sensors, as in Fig. 5(b). In either case, x would include the states of the plant and the controller. The block diagram in Fig. 4 also captures the case of a static controller that is not collocated with the sensors nor with the actuators as in Fig. 5(c), because a memoryless controller could be moved next to the actuators, without affecting the stability of the closed loop [3, 81].

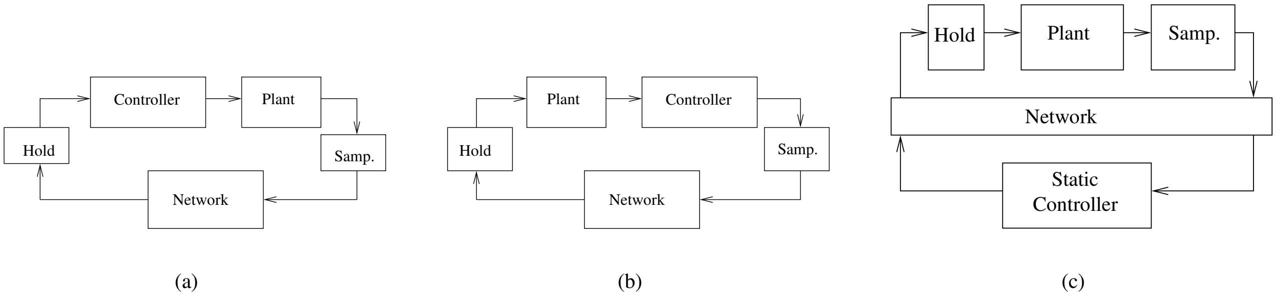


Fig. 5. NCS control architectures captured by the one-channel feedback NCS in Fig. 4

In the one-channel feedback NCS in Fig. 4, the signal $y(t)$ is sampled at times $\{t_k : k \in \mathbb{N}\}$ and the samples $y_k := y(t_k)$, $\forall k \in \mathbb{N}$ are sent through the network. In a lossless network, we have

$$\hat{y}_k = y_k, \quad \forall k \in \mathbb{N}, \tag{13}$$

but the samples only arrive at the destination after a (possibly variable) delay of $\tau_k \geq 0$. At these times $\hat{y}(t)$ is updated, leading to

$$\hat{y}(t) = \begin{cases} \hat{y}_{k-1} & t \in [t_k, t_k + \tau_k) \\ \hat{y}_k & t \in [t_k + \tau_k, t_{k+1}) \end{cases} \tag{14}$$

where we assume that the network delays are always smaller than one sampling interval, i.e., that $t_k + \tau_k < t_{k+1}$, $\forall k \in \mathbb{N}$ (cf. Fig 6). Defining $x_k := x(t_k)$, $\forall k \in \mathbb{N}$ and applying the variation of constants formula to (12) and (14) we conclude that

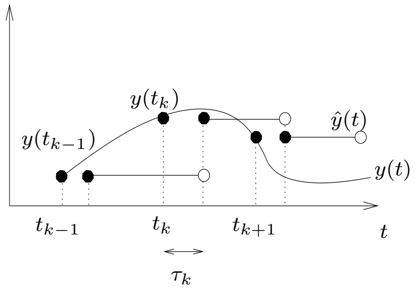


Fig. 6. The piecewise constant signal $\hat{y}(t)$ defined by (14).

$$x_{k+1} = e^{A(t_{k+1}-t_k)}x_k + e^{A(t_{k+1}-t_k-\tau_k)}\Gamma(\tau_k)B\hat{y}_{k-1} + \Gamma(t_{k+1} - t_k - \tau_k)B\hat{y}_k, \tag{15}$$

where $\Gamma(s) := \int_0^s e^{Az}dz$, $\forall s \in \mathbb{R}$. To analyze this NCS, one can define an augmented discrete-time state $z'_k := \begin{bmatrix} x(t_k)' & \hat{y}'_{k-1} \end{bmatrix}$, $\forall k \in \mathbb{N}$ and conclude from (13) and (15) that it evolves according to

$$z_{k+1} = \Phi(t_{k+1} - t_k, \tau_k)z_k, \quad \forall k \in \mathbb{N}, \tag{16}$$

where

$$\Phi(a, b) := \begin{bmatrix} e^{Aa} + \Gamma(a - b)BC & e^{A(a-b)}\Gamma(b)B \\ C & 0 \end{bmatrix}, \quad \forall a, b \in \mathbb{R}.$$

In the absence of delay (i.e., when $\tau_k = 0$, $\forall k \in \mathbb{N}$), the state z_k does not need to include $\hat{y}_{k-1}$ and therefore the function $\Phi(\cdot)$ in (16) can simply be defined by

$$\Phi(a, 0) := e^{Aa} + \Gamma(a)BC, \quad \forall a > 0. \tag{17}$$

When the plant (12) is open-loop unstable, (16) will generally be unstable if the interval between sampling times becomes very large. In view of this, significant work has been devoted to finding upper bounds on $t_{k+1} - t_k$, $\forall k \in \mathbb{N}$ for which stability can be guaranteed. These upper bounds are sometimes called the *maximum allowable transfer interval* (MATI) [69].

Delays longer than one sampling interval may result in more than one $\hat{y}_k$ (or none) arriving during a single sampling interval, making the derivation of recursive formulas like (16) difficult. All results reviewed in this section are based on a reduction of the NCS to some form of discrete-time system such as (16), for simplicity we will therefore implicitly assume delays smaller than one sampling interval. This restriction will be lifted in Section III-C.

Periodic sampling and constant delay: When $y(t)$ is sampled periodically and the delay is constant, the discrete-time system (16) is time-invariant and it is straightforward to establish its stability:

Theorem 4 ([3]): Assuming that there exist constants $h > \tau \geq 0$ such that

$$t_{k+1} - t_k = h, \quad \tau_k = \tau, \quad \forall k \in \mathbb{N},$$

the NCS (12)–(14) in Fig. 4 is exponentially stable if and only if $\Phi(h, \tau)$ is Schur (i.e., all its eigenvalues have magnitude strictly less than one). □

Remark 1: Defining the alternative augmented state $\bar{z}'_k := \begin{bmatrix} x(t_k + \tau_k)' & x(t_k)' \end{bmatrix}$, $\forall k \in \mathbb{N}$ we obtain

$$\bar{z}_{k+1} = \bar{\Phi}(t_{k+1} - t_k - \tau_k, \tau_{k+1})\bar{z}_k, \ \ \forall k \in \mathbb{N}, \qquad \bar{\Phi}(a, b) := \begin{bmatrix} e^{A(a+b)} & \Gamma(a + b)BC \\ e^{Aa} & \Gamma(a)BC \end{bmatrix}, \ \ \forall a, b \in \mathbb{R}, \tag{18}$$

from which stability of the NCS can also be deduced. Zhang et al. [81] use results from [2] on the stability of nonlinear hybrid systems to conclude that Schurness of $\bar{\Phi}(h, \tau)$ is a sufficient condition for stability of the NCS in the time-invariant case. From (18) one can see that this condition is also necessary. □

While some network protocols guarantee constant delay, such as the Controller Area Network (CAN) protocol [23], most protocols introduce delays that can vary significantly from message to message. Variable delays can be equalized by introducing a buffer at the receiver, where data packets can be held so that all packets appear to have the same delay from the perspective of the NCS [30]. However, the downside of delay equalization is that all packets will appear to have a delay as large as the worst-case delay that the network can introduce.

Periodic sampling and variable delay: Suppose that the sampling intervals are constant and equal to h and the delay takes values equal to $\kappa \frac{h}{N}$ where $\kappa \in 0, 1, \cdots, D_{\max}$ and $D_{\max} \leq N \in \mathbb{N}$. This situation happens when computation and transmission delays are negligible and access delays serve as the main source of delays in NCS [26, 25, 27]. Under these assumptions the closed-loop system (16) can be written as a discrete-time switched system with $D_{\max}+1$ modes as follows

$$z_{k+1} = A_{\sigma_k} z_k, \qquad \forall k \in \mathbb{N},$$

where the *switching signal* σ_k takes values from $\{0, 1, \cdots D_{\max}\}$ at each time step and, when $\sigma_k = \kappa$,

$$A_\kappa := \Phi\big(h, \kappa \frac{h}{N}\big) \qquad \kappa \in \{0, 1, \cdots D_{\max}\}$$

Lin et al. [26] assume that for the case of no delay or small delays ($\kappa \leq N_0$), the corresponding state matrix, A_κ, is Schur stable while for the case of large delay ($\kappa > N_0$), A_κ is not Schur stable. Using *average dwell time* results for discrete switched systems [79] provides conditions such that NCS stability is guaranteed. Also the authors consider robust disturbance attenuation analysis for this class of NCSs.

Remark 2: One Packet dropout can be modeled as an extra mode where $\kappa = N$. The authors extended the results for the case of consecutive packet dropouts in [25].

Variable sampling and delay: When the network delay is not constant or when the signal $y(t)$ is sampled in a non-periodic fashion, (16) is not time-invariant and one needs a Lyapunov-based argument to prove its stability. The following result is adapted from [80][1] and expresses a sufficient condition for $V(z) := z'Pz$ to be a Lyapunov function for (16), from which stability of the NCS can be deduced:

Theorem 5: Assume that there exist constants $h_{\min}, h_{\max}, \tau_{\min}, \tau_{\max}$ such that

$$0 \leq h_{\min} \leq t_{k+1} - t_k \leq h_{\max}, \qquad 0 \leq \tau_{\min} \leq \tau_k \leq \tau_{\max}, \quad \forall k \in \mathbb{N}.$$

The NCS (12)–(14) in Fig. 4 is exponentially stable if there exists a symmetric matrix P such that

$$P > 0, \qquad \Phi(h,\tau)' P \Phi(h,\tau) - P < 0, \qquad \forall h \in [h_{\min}, h_{\max}],\ \tau \in [\tau_{\min}, \tau_{\max}]. \tag{19}$$

□

From a numerical perspective, it is generally not simple to find a matrix P that satisfies (19) for *all* values of h and τ in the given intervals. However, testing the existence of a matrix P that satisfies (19) for values of h and τ on a *finite grid* leads to a finite set of Linear Matrix Inequalities (LMIs) that is easy to solve. Zhang and Branicky [80] propose a randomized algorithm to find the largest value of $h_{\max}$ for which stability can be guaranteed when $h_{\min} = \tau_{\min} = \tau_{\max} = 0$.

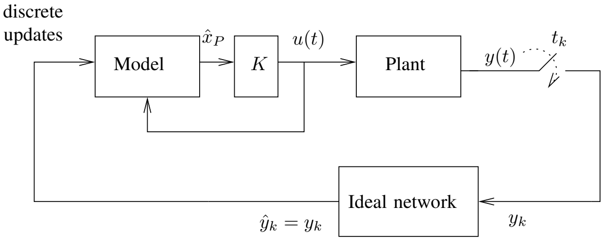


Fig. 7. State feedback model-based NCS.

Model-based controller: Montestruque and Antsaklis [34, 35, 36, 37, 38] consider the model-based one-channel feedback NCS in Fig. 7, in which the signal y transmitted across the network is the state of an LTI plant

$$\dot{x}_P = A_P x_P + B_P u, \qquad y = x_P \tag{20}$$

whose input u is generated by an estimator-based controller collocated with the actuators. In general, an exact model of the plant is not available and therefore the controller must construct an estimate $\hat{x}_P$ of x_P based on the following approximate plant model

$$\dot{\hat{x}}_P = \hat{A}_P \hat{x}_P + \hat{B}_P u, \tag{21}$$

which is instantaneously updated at the sampling times $\{t_k : k \in \mathbb{N}\}$ using the true value of x_P coming from the network. The key difference between the NCSs in Figures 4 and 7 is that in the former the data coming from the network is held constant

[1] A special case of Theorem 5 with $h_{\min} = \tau_{\min} = \tau_{\max} = 0$ and the matrix $\Phi(\cdot)$ given by (17) can be found in [80].

between sampling times, whereas in the later this data is used to instantaneously update the state of the controller. Assuming that the network delay is negligible, the controller updates its state estimate according to

$$\hat{x}_P(t_k) = y(t_k) = x_P(t_k), \quad \forall k \in \mathbb{N}. \tag{22}$$

It then uses a certainty equivalence control law of the form

$$u = K\hat{x}_P, \tag{23}$$

with the matrix K chosen so that $\hat{A}_P + \hat{B}_P K$ is Hurwitz (i.e., all its eigenvalues have strictly negative real part). Note that the matrices in the plant model (20) and the estimator model (21) generally do not match due to parametric uncertainty.

Because of (22), we conclude that the state estimation error $e := x_P - \hat{x}_P$ is reset to zero at sampling times and therefore its dynamics are defined by

$$\dot{e} = (\tilde{A}_P + \tilde{B}_P K)x_P + (\hat{A}_P - \tilde{B}_P K)e, \qquad\qquad e(t_k) = 0,$$

where the matrices $\tilde{A} := A_P - \hat{A}_P$ and $\tilde{B}_P := B_P - \hat{B}_P$ represent the difference between the actual plant and the model used to build the estimator. Defining $z' := \begin{bmatrix} x'_P & e' \end{bmatrix}$, we conclude that the overall closed-loop evolves according to the following impulsive system

$$\dot{z} = \Lambda z, \quad z(t_k)' = \begin{bmatrix} x'_P(t_k^-) & 0 \end{bmatrix}, \tag{24}$$

where $x_P(t_k^-)$ denotes the limit from below of $x_P(\tau)$ as $\tau \uparrow t_k$ and

$$\Lambda := \begin{bmatrix} A_P + B_P K & -B_P K \\ \tilde{A}_P + \tilde{B}_P K & A_P - \tilde{B}_P K \end{bmatrix}. \tag{25}$$

Defining the discrete-time state $z_k := z(t_k)$, $\forall k \in \mathbb{N}$, we obtain the following model for its evolution

$$z_{k+1} = M(t_{k+1} - t_k)z_k, \ \ \forall k \in \mathbb{N}, \qquad\qquad M(a) := \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda a} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \ \ \forall a > 0.$$

The following result is adapted from [36] and follows from standard results on the stability of discrete-time systems:

Theorem 6: The following two results hold for the NCS (20)–(23) in Fig. 7:

1) When there exists a constant h such that $t_{k+1} - t_k = h$, $\forall k \in \mathbb{N}$, the NCS is globally exponentially stable if and only if $M(h)$ is Schur.
2) When there exist constants $h_{\min}$ and $h_{\max}$ such that $0 \leq h_{\min} \leq t_{k+1} - t_k \leq h_{\max}$, $\forall k \in \mathbb{N}$, the NCS is exponentially stable if there exists a symmetric matrix P such that

$$P > 0, \qquad\qquad M(h)PM(h) - P < 0, \qquad\qquad \forall h \in [h_{\min}, h_{\max}]. \qquad \square$$

For periodic sampling, Montestruque and Antsaklis [34, 35, 36] use a similar approach to determine the maximum value of $h := t_k - t_{k-1}$, $\forall k \in \mathbb{N}$ for which the NCS is stable, both under state and output feedback. The same authors [37, 38] also consider (non-constant) stochastic[2] sampling intervals $\mathbf{h}_k := \mathbf{t}_k - \mathbf{t}_{k-1}$, $\forall k \in \mathbb{N}$. The following result addresses the case of independent identically distributed (i.i.d.) $\mathbf{h}_k$.

Theorem 7 ([38]): Assume that the sampling intervals $\mathbf{h}_k := \mathbf{t}_k - \mathbf{t}_{k-1}$, $\forall k \in \mathbb{N}$ are i.i.d. random variables and that the following two conditions hold

$$E\big[(e^{2\sigma_{\max}(\Lambda)\mathbf{h}_k} - 1)^{1/2}\big] < \infty, \qquad\qquad E\big[\sigma_{\max}(M(\mathbf{h}_k))\big] < 1, \qquad\qquad \forall k \in \mathbb{N},$$

where the matrix Λ is defined by (25) and $\sigma_{\max}(\Lambda)$ denotes its largest singular value. Then the NCS (20)–(23) in Fig. 7 is almost surely stable in the sense that for every initial condition $z(t_0)$ to (24) and for every $\epsilon > 0$, we have that

$$\lim_{\bar{\delta} \to \infty} \Pr\left\{ \sup_{k \geq \bar{\delta}} \left(\int_{\mathbf{t}_k}^{\mathbf{t}_{k+1}} \|z(\xi)\|^2 d\xi \right)^{1/2} > \epsilon \right\} = 0. \qquad \square$$

Montestruque and Antsaklis [38] further generalize these results to sampling intervals driven by Markov chains, which can be used to model correlated inter-sampling intervals $\mathbf{h}_k$.

[2] To emphasize the fact that the sampling times are now random variables, we represent them in bold-face.

General nonlinear case: Consider a nonlinear plant and remote controller with exogenous disturbances of the following form:

$$\dot{x}_P = f_P(x_P, \hat{u}, w), \qquad\qquad y = g_P(x_P), \tag{26a}$$

$$\dot{x}_C = f_C(x_C, \hat{y}, w), \qquad\qquad u = g_C(x_C), \tag{26b}$$

where x_P and x_C are the states of the plant and the controller; $\hat{u}$ and y the plant's input and output; $\hat{y}$ and u the controller's input and output; and w an exogenous disturbance. The plant and the controller are connected through a *two-channel feedback NCS* as in Fig. 2. Ignoring network delay, between the sampling times $\{t_k : k \in \mathbb{N}\}$ both $\hat{u}$ and $\hat{y}$ are held constant:

$$\hat{u}(t) = \hat{u}(t_k^+), \qquad \hat{y}(t) = \hat{y}(t_k^+), \qquad \forall t \in (t_k, t_{k+1}],\ k \in \mathbb{N}, \tag{27}$$

where $u(t_k^+)$ and $y(t_k^+)$ denote the limit from above of $u(\tau)$ and $y(\tau)$, respectively, as $\tau \downarrow t_k$. The signals $u(t)$ and $y(t)$ are not necessarily both sampled and sent to the network at every sampling time and therefore

$$\hat{u}(t_k^+) = \begin{cases} u(t_k) & u \text{ sampled at time } t_k \\ \hat{u}(t_k) & u \text{ not sampled at time } t_k \end{cases} \qquad \hat{y}(t_k^+) = \begin{cases} y(t_k) & y \text{ sampled at time } t_k \\ \hat{y}(t_k) & y \text{ not sampled at time } t_k \end{cases} \qquad \forall k \in \mathbb{N}. \tag{28}$$

The sampling model (28) can be written compactly as

$$\hat{u}(t_k^+) = u(t_k) + h_u\big(k, e(t_k)\big), \qquad \hat{y}(t_k^+) = y(t_k) + h_y\big(k, e(t_k)\big), \qquad \forall k \in \mathbb{N}, \tag{29}$$

where

$$e = \begin{bmatrix} e_y \\ e_u \end{bmatrix} := \begin{bmatrix} \hat{y} - y \\ \hat{u} - u \end{bmatrix} \in \mathbb{R}^{n_e},$$

and

$$h_u(k, e) := \begin{cases} 0 & u \text{ sampled at time } t_k \\ e_u & u \text{ not sampled at time } t_k \end{cases} \qquad h_y(k, e) := \begin{cases} 0 & y \text{ sampled at time } t_k \\ e_y & y \text{ not sampled at time } t_k \end{cases} \qquad \forall k \in \mathbb{N},\ e \in \mathbb{R}^{n_e}.$$

These definitions allow us to write the following "reset-map" for the error e at the sampling times:

$$e(t_k^+) = \begin{bmatrix} \hat{y}(t_k^+) - y(t_k) \\ \hat{u}(t_k^+) - u(t_k) \end{bmatrix} = \begin{bmatrix} h_u(k, e) \\ h_y(k, e) \end{bmatrix}, \tag{30}$$

where we used (29) and the fact that both y and u are continuous functions of time. The fact that the neither the state of the process nor the state of the controller appear in (30) will be very convenient for the analysis. It is also interesting to observe that the error reset map in (30) does not depend on the process/controller dynamics, but only on the "protocol" used to decide which variables should be sampled at each sampling time.

Walsh et al. [70], Nesic and Teel [43] actually consider a sampling model more general than (28), as they allow for only some entries of u and y to be transmitted through the network at each sampling time. In practice, this means that only some entries of $h_u(\cdot)$ and $h_y(\cdot)$ may be equal to zero at each sampling time. To capture this, Nesic and Teel [43] generalize (30) to

$$e(t_k^+) = h\big(k, e(t_k)\big), \qquad \forall t \in (t_k, t_{k+1}],\ k \in \mathbb{N}, \tag{31}$$

where $h(k, \cdot)$ specifies which entries of the error are reset to zero at the kth sampling time. This function can be regarded as implementing a *network access protocol* that decides which input/output signals should be sampled at each time t_k, $k \in \mathbb{N}$. When this decision is based on the current mismatches between u and $\hat{u}$ and/or between y and $\hat{y}$, we have a *dynamic protocol*, such as the Try-Once-Discard protocol in [70, 43]. Otherwise, we have a *static protocol*, such as the Round-robin protocol in [81, 70, 43]. Since the current mismatches may not always be available, Tabbara et al. [63] propose an alternative formulation in which the decision to sample a signal is based on an estimate of the mismatch for that signal.

Defining $x := \begin{bmatrix} x_P' & x_C' \end{bmatrix}'$, the NCS described by (26), (27), and (31) can be modeled by an impulsive system of the form

$$\dot{x} = f(x, e, w), \qquad \forall t \geq 0, \qquad x \in \mathbb{R}^{n_x},\ w \in \mathbb{R}^{n_w}, \tag{32a}$$

$$\dot{e} = g(x, e, w), \qquad \forall t \in (t_k, t_{k+1}], \qquad e \in \mathbb{R}^{n_e}, \tag{32b}$$

$$e(t_k^+) = h\big(k, e(t_k)\big), \qquad \forall k \in \mathbb{N}, \tag{32c}$$

where $e(t_k^+)$ denotes the limit from above of $e(\tau)$ as $\tau \downarrow t_k$ and all limits from below of $e(t)$, $t \geq 0$ coincide with the value of the function at the same time. This is a generalization of the NCS model in (24), in which signals were assumed continuous from above instead of continuous from below.

The following result is adapted from [43] and can be used to establish the stability of (32):

Theorem 8: Suppose that the following conditions hold:

1) There exists a function $W : \mathbb{N} \times \mathbb{R}^{n_e} \to [0, \infty)$ and constants $\rho \in [0, 1)$, $a_1, a_2 > 0$ such that

$$a_1\|e\| \leq W(k,e) \leq a_2\|e\|, \qquad W(k+1, h(k,e)) \leq \rho W(k,e), \qquad \forall k \in \mathbb{N},\ e \in \mathbb{R}^{n_e}.$$

2) There exists a function $H : \mathbb{R}^{n_x} \to [0, \infty)$ and a constant L such that

$$\frac{\partial W(k,e)}{\partial e} \cdot g(x,e,w) \leq L\, W(k,e) + H(x) + \|w\|, \qquad \forall k \in \mathbb{N},\ x \in \mathbb{R}^{n_x},\ e \in \mathbb{R}^{n_e},\ w \in \mathbb{R}^{n_w}.$$

3) There exists a class $\mathcal{KL}$ function[3] β_1 and a positive constant $\gamma_1 > 0$ such that

$$H\big(x(t)\big) \leq \beta_1\big(\|x(t_0)\|, t - t_0\big) + \gamma_1 \operatorname*{ess\,sup}_{\tau \in (t_0, t)} \Big(\|e(\tau)\| + \|w(\tau)\|\Big), \qquad \forall t \geq t_0 \geq 0$$

along solutions to (32a).

4) There exists a class $\mathcal{KL}$ function β_2 and a class $\mathcal{K}$ function γ_2 such that

$$\|x(t)\| \leq \beta_2\big(\|x(t_0)\|, t - t_0\big) + \operatorname*{ess\,sup}_{\tau \in (t_0, t)} \gamma_2\Big(H\big(x(\tau)\big) + \|e(\tau)\| + \|w(\tau)\|\Big), \qquad \forall t \geq t_0 \geq 0$$

along solutions of (32a).

5) There exists a positive constant ξ (called the *maximum allowable transfer interval*, MATI) such that

$$0 < t_{k+1} - t_k \leq \xi < \frac{1}{L} \ln\Big(\frac{L + \gamma_1}{\rho L + \gamma_1}\Big), \qquad \forall k \in \mathbb{N}.$$

Then the NCS modeled by (32) is input-to-state stable from the disturbance input w to its state (x, e). □

Condition 1 should be viewed as a requirement on the network access protocol specified by the function $h(\cdot)$. In practice, this condition requires the protocol to define an exponentially stable auxiliary discrete-time system

$$z_{k+1} = h(k, z_k)$$

with a decay rate of $\rho < 1$. In view of this, Nesic and Teel [43] introduce the terminology "uniformly exponentially stable protocol" to denote any protocol that satisfies condition 1.

For linear systems, the remaining assumptions of Theorem 8 are fairly mild. They basically require a growth for the error dynamics no faster than exponential and appropriate disturbance rejection properties of the "closed-loop" system (32), with respect to the inputs e and w. However, for nonlinear systems these assumptions may be difficult to verify. In either case, Nesic and Teel [43, 44, 45] show that the MATI condition 5 in Theorem 8 is less conservative than the ones in [69, 70, 68].

B. Packet dropouts

Packet dropouts can be modeled either as stochastic or deterministic phenomena. The simplest stochastic model assumes that dropouts are realizations of a Bernoulli process [60, 65]. Finite-state Markov chains can be used to model correlated dropouts [61] and Poisson processes can be used to model stochastic dropouts in continuous time [74]. Deterministic models for dropouts have also been proposed, either specified in terms of time-averages [81] or in terms of worst-case bounds on the number of consecutive dropouts [78, 40]. We defer the study of worst-case dropout models to Section III-C.

Consider again the one-channel feedback NCS in Fig. 4, with a plant/controller pair (12), for which the signal y is sampled at times $\{t_k : k \in \mathbb{N}\}$ and the samples $y_k := y(t_k)$ are sent through the network. When packets are dropped, the network model in (13) must be changed. It is often assumed that when the packet containing the sample y_k is dropped, the NCS utilizes the previous value of $\hat{y}_k$ [81, 57]. This corresponds to replacing the lossless network model (13) by

$$\hat{y}_k = \theta_k y_k + (1 - \theta_k)\hat{y}_{k-1} = \begin{cases} y_k & \theta_k = 1 \text{ (no packet dropout)} \\ \hat{y}_{k-1} & \theta_k = 0 \text{ (packet dropout)} \end{cases} \qquad \forall k \in \mathbb{N}, \tag{33}$$

where $\theta_k = 0$ when there is a packet dropout at time k and $\theta_k = 1$ otherwise. Hadjicostis and Touri [15] assume instead that $\hat{y}_k$ is set to zero when the packet containing y_k is dropped, i.e., $\hat{y}_k = \theta_k y_k$, $\forall k \in \mathbb{N}$.

Assuming that the delay[4] τ_k experienced by the kth packet is smaller than the corresponding sampling interval, the continuous-time signal $\hat{y}(t)$ is still updated according to (14). For simplicity, we assume periodic sampling and constant network delay,

[3] Recall that a function $\alpha : [0, \infty) \to [0, \infty)$ is said to be of class $\mathcal{K}$ if it is continuous, strictly increasing, and $\alpha(0) = 0$. A function $\beta : [0, \infty) \times [0, \infty) \to [0, \infty)$ is said to be of class $\mathcal{KL}$ if $\beta(\cdot, t)$ is of class $\mathcal{K}$ for each fixed $t \geq 0$ and $\beta(r, t)$ decreases to 0 as $t \to \infty$ for each fixed $r \geq 0$.

[4] When the kth packet is dropped the value of τ_k is of no consequence and can be assumed zero.

i.e., $t_{k+1} - t_k = h$, $\tau_k = \tau$, $\forall k \in \mathbb{N}$. To analyze this NCS, once again we define an augmented discrete-time state $z'_k := \begin{bmatrix} x(t_k)' & \hat{y}'_{k-1} \end{bmatrix}$. From (15) and (33) we now conclude that

$$z_{k+1} = \Phi_{\theta_k} z_k, \tag{34}$$

where

$$\Phi_\theta := \begin{bmatrix} e^{Ah} + \theta\Gamma(h-\tau)BC & e^{A(h-\tau)}\Gamma(\tau)B + (1-\theta)\Gamma(h-\tau)B \\ \theta C & (1-\theta)I \end{bmatrix}, \quad \forall \theta \in \{0,1\}. \tag{35}$$

Deterministic dropouts: Zhang et al. [81] consider a deterministic dropout model, with packet dropouts occurring at an asymptotic rate defined by the following time-average

$$r := \lim_{T\to\infty} \frac{1}{T} \sum_{k=k_0}^{k_0+T-1} (1-\theta_k), \quad \forall k_0 \in \mathbb{N}, \tag{36}$$

which implicitly assumes that the limit exists. Under this dropout model, the system (34) falls under the class of *asynchronous dynamical systems* (ADSs). These are hybrid systems whose continuous dynamics are governed by differential or difference equations and the discrete dynamics are governed by finite automata. In ADSs, the finite automata are driven asynchronously by external events that occur at pre-specified rates. The ADSs of interest to us are defined by a difference equation such as (34), where θ_k takes values in some index set $\{0, 1, \dots, N\}$ and the rate at which the event $\theta_k = j$ occurs is defined by the following time average

$$r_j := \lim_{T\to\infty} \frac{1}{T} \sum_{k=k_0}^{k_0+T-1} \delta_{j\,\theta_k}, \qquad \forall k_0 \in \mathbb{N},\ j \in \{0, 1, \dots, N\}, \tag{37}$$

where $\delta_{j\,\theta_k} = 1$ when $\theta_k = j$ and zero otherwise. When all the limits exist, we have $\sum_{j=0}^{N} r_j = 1$. The following result is adapted from Hassibi et al. [16] and uses a quadratic Lyapunov function of the form $V(z) := z'Pz$ to establish the asymptotic stability of the ADS system (34) with rates (37).

Theorem 9 ([16]): Assume that there exist a symmetric matrix $P > 0$ and scalars $\alpha, \alpha_0, \alpha_1, \dots, \alpha_N$ such that

$$\alpha_0^{r_0}\alpha_1^{r_1}\cdots\alpha_N^{r_N} > \alpha > 1, \qquad \Phi'_j P \Phi_j \le \alpha_j^{-2} P, \quad \forall j \in \{0, 1, \dots, N\},$$

then the ADS (34) is exponentially stable in the sense that $\lim_{k\to\infty} \alpha^k \|z_k\| = 0$ for every sequence $\delta_{j\,\theta_k}$ for which (37) holds. □

The following result is obtained by applying Theorem 9 to our NCS with constant sampling interval h and constant delay τ:

Corollary 10: Assuming that there exist a symmetric matrix $P > 0$ and scalars $\alpha, \alpha_0, \alpha_1$ such that

$$\alpha_0^r \alpha_1^{1-r} > \alpha > 1, \qquad \Phi_0(h,\tau)' P \Phi_0(h,\tau) \le \alpha_0^{-2} P, \qquad \Phi_1(h,\tau)' P \Phi_1(h,\tau) \le \alpha_1^{-2} P, \tag{38}$$

then the NCS (34) is exponentially stable in the sense that $\lim_{k\to\infty} \alpha^k \|z_k\| = 0$ for every sequence θ_k for which (36) holds. □

The main difficulty in applying this result is that the set of matrix inequalities that appears in (38) is bilinear in the unknowns P, α_j and therefore generally non-convex. However, one can use a "line-search" procedure over the two scalars α_0, α_1 to determine the feasibility of (38).

Remark 3: One can also express (33) as $\hat{y}_k = C\hat{x}_k$, $\forall k \in \mathbb{N}$, with $\hat{x}_k$ defined by

$$\hat{x}_k := \theta_k x_k + (1-\theta_k)\hat{x}_{k-1} = \begin{cases} x_k & \theta_k = 1 \text{ (no packet dropout)} \\ \hat{x}_{k-1} & \theta_k = 0 \text{ (packet dropout)} \end{cases} \qquad \forall k \in \mathbb{N}.$$

When $\tau = 0$ one can analyze the system's stability using the discrete-time state $\bar{z}'_k := \begin{bmatrix} x'_k & \hat{x}'_k \end{bmatrix}$, which evolves according to

$$\bar{z}_{k+1} = \bar{\Phi}_{\theta_{k+1}} \bar{z}_k, \qquad \bar{\Phi}_\theta := \begin{bmatrix} e^{Ah} & \Gamma(h)BC \\ \theta e^{Ah} & \theta\Gamma(h)BC + (1-\theta)I \end{bmatrix}, \qquad \forall k \in \mathbb{N}.$$

In their work, Zhang et al. [81] considered this discrete-time system instead of (34). □

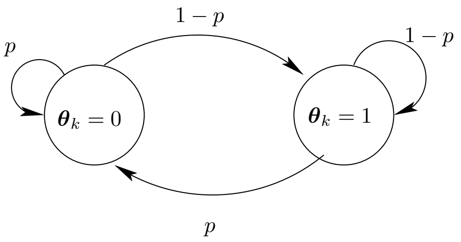


Fig. 8. Bernoulli drops with probability p modeled as a two-state discrete-time Markov chain.

Stochastic dropouts: Seiler and Sengupta [57, 58] consider stochastic losses. In their formulation, $\boldsymbol{\theta}_k$ is a Bernoulli process[5] with probability of dropout (i.e., $\boldsymbol{\theta}_k = 0$) equal to $p \in [0, 1)$. Under this dropout model, the system (34) is a special case of a discrete-time *Markovian jump linear system* (MJLS). In general MJLSs, the index $\boldsymbol{\theta}_k$ in (34) would be the state of a discrete-time Markov chain with a finite number of states and a given transition probability matrix. For Bernoulli drops, the Markov chain only has two states and the transition probability from any state to the *dropout-state* $\boldsymbol{\theta}_k = 0$ is equal to p and the transition probability from any state to the state $\boldsymbol{\theta}_k = 1$ is equal to $1 - p$, as shown in Fig 8. The stability of discrete-time MJLSs can be established using results from [5] (cf. [6] for continuous-time MJLs), leading to the following theorem:

Theorem 11 ([57]): The NCS (34) with dropout probability p (Bernoulli) is exponentially mean-square stable if there exists a symmetric matrix $Z > 0$ such that[6]

$$\begin{bmatrix} Z & \sqrt{p}(\Phi_0 Z)' & \sqrt{1-p}(\Phi_1 Z)' \\ * & Z & 0 \\ * & * & Z \end{bmatrix} > 0. \tag{39}$$

□

When the controller is collocated with the actuators, Seiler and Sengupta [57, 58] suggest that the control law can adapt to the occurrence of dropouts by allowing the controller to use different gains at different time instants k, based on the value of $\theta_k \in \{0, 1\}$. In this case, the matrices A, B, C in (35) depend on θ, but one still gets a system of the form (34) and one can use similar tools to analyze its stability. We will return to this issue in Section IV when we discuss the design of NCS controllers.

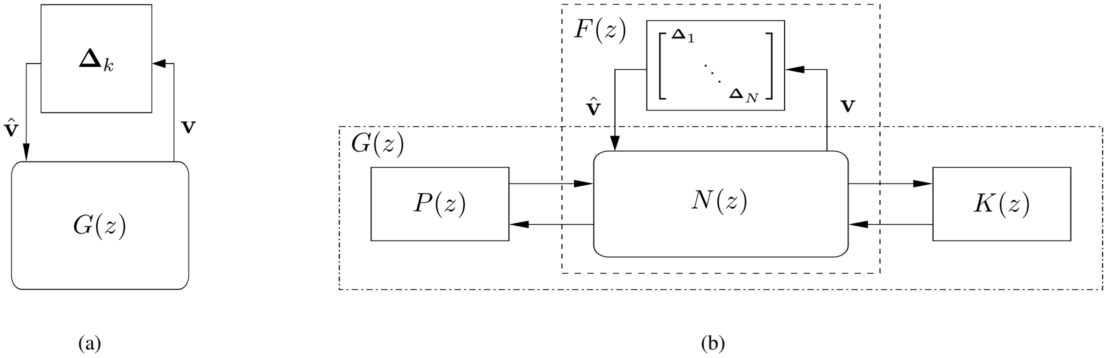


Fig. 9. Fading networks: (a) Nominal deterministic system with a stochastic perturbation. (b) Fading network representation of a general NCS.

Fading networks: Elia [8] models NCSs with LTI plants and controllers as deterministic "nominal" discrete-time systems connected to zero-mean stochastic structured uncertainty as in Fig. 9(a). To see how this can be done, consider the one-channel feedback NCS in Fig. 4, with a SISO plant/controller pair (12) with periodic sampling at times $t_k := kh$, $\forall k \in \mathbb{N}$ and constant delay $\tau < h$. We have seen that if the previous value of $\hat{y}_k$ is used when the packet containing the sample y_k is dropped, this NCS can be modeled by the discrete-time system (34)–(35), where $\boldsymbol{\theta}_k \in \{0, 1\}$ is a Bernoulli process with probability of dropout ($\boldsymbol{\theta}_k = 0$) equal to $p \in [0, 1)$. Defining

$$\boldsymbol{\Delta}_k := \frac{\boldsymbol{\theta}_k}{1-p} - 1, \quad \forall k \in \mathbb{N}, \tag{40}$$

[5] To emphasize the fact that now the $\boldsymbol{\theta}_k$, $k \in \mathbb{N}$ are random variables, we denote them in boldface.

[6] Matrix entries denotes by '*' are implicitly defined by the fact that the matrix is symmetric.

this variable can be viewed as a (SISO) stochastic perturbation with zero mean and variance

$$\sigma^2 := \mathrm{E}[\mathbf{\Delta}_k^2] = \frac{p}{1-p}.$$

The corresponding nominal system is obtained using the fact that $\boldsymbol{\theta}_k = (1-p)(1+\mathbf{\Delta}_k)$ to re-write (34)–(35) as in Fig. 9(a):

$$\begin{cases} \mathbf{z}_{k+1} = \bar{A}\mathbf{z}_k + \bar{B}\hat{\mathbf{v}}_k \\ \mathbf{v}_k = \bar{C}\mathbf{z}_k \end{cases} \qquad \hat{\mathbf{v}}_k = \mathbf{\Delta}_k \mathbf{v}_k, \tag{41}$$

where

$$\bar{A} := \begin{bmatrix} e^{Ah} + (1-p)\Gamma(h-\tau)BC & e^{A(h-\tau)}\Gamma(\tau)B + p\Gamma(h-\tau)B \\ (1-p)C & pI \end{bmatrix}, \quad \bar{B} := \begin{bmatrix} (1-p)\Gamma(h-\tau)B \\ (1-p)I \end{bmatrix}, \quad \bar{C} := \begin{bmatrix} C & -I \end{bmatrix}.$$

Much more general NCSs can be modeled in this fashion. Fig. 9(b), depicts a general model for NCSs in this framework. In this figure, $P(z)$ denotes a discretized model of the plant(s) with associated sample and hold blocks and $K(z)$ a discrete-time model of the controller(s). These two blocks are interconnected through a *fading network* denoted by $F(z)$, which is depicted inside the dashed box. This network consists of a deterministic discrete-time model $N(z)$ called the *mean network* and a zero-mean stochastic block-diagonal multiplicative block $\mathbf{\Delta}$ called the *stochastic perturbation*. In the example above, the fading network essentially corresponds to (33), which can be re-written as follows in terms of the stochastic perturbation (40):

$$\begin{cases} \hat{y}_k = p\hat{y}_{k-1} + (1-p)y_k + (1-p)\hat{\mathbf{v}}_k \\ \mathbf{v}_k = y_k - \hat{y}_{k-1} \end{cases} \qquad \hat{\mathbf{v}}_k = \mathbf{\Delta}_k \mathbf{v}_k,$$

with the mean network on the left-hand-side and the stochastic perturbation on the right-hand-side. Figure 9(b) can be viewed as a special case of Fig. 9(a) by associating the system inside the dash-dot box in Fig. 9(b) with $G(z)$ in Fig. 9(a).

To study the stability of the NCSs in Fig. 9, Elia [8] considers the interconnection of a discrete-time LTI system with a transfer function $G(z)$ and a stabilizable and detectable realization[7]

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\hat{\mathbf{v}}_k, \qquad \mathbf{v}_k = C\mathbf{x}_k, \qquad k \in \mathbb{N},\ \mathbf{x}_k \in \mathbb{R}^n,\ \mathbf{v}_k, \hat{\mathbf{v}}_k \in \mathbb{R}^m, \tag{42}$$

in feedback with a multiplicative stochastic perturbation of the form

$$\hat{\mathbf{v}}_k = \begin{bmatrix} \mathbf{\Delta}_k^1 & 0 & \cdots & 0 \\ 0 & \mathbf{\Delta}_k^2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Delta}_k^m \end{bmatrix} \mathbf{v}_k, \qquad \forall k \in \mathbb{N}, \tag{43}$$

where all the $\mathbf{\Delta}_k^i$ are i.i.d. random variables with zero mean and the same variance $\sigma^2 := \mathrm{E}[(\mathbf{\Delta}_k^i)^2]$, $\forall i, k$. It turns out that the stability of this interconnection is independent of the distribution of the $\mathbf{\Delta}_k^i$ and only depends on their variance σ^2. The following theorem characterizes the stability of the interconnection in terms of LMIs.

Theorem 12 ([8]): Suppose that the matrix A is Schur and that the initial state $\mathbf{x}_0$ has finite variance and is independent of the $\mathbf{\Delta}_k^i$. The system (42)–(43) is mean-square stable if and only if there exists a symmetric matrix $Q > 0$ and scalars $\alpha_1, \dots, \alpha_m > 0$ such that

$$AQA' + \sum_{j=1}^{m} b_j \alpha_j b_k' < Q, \qquad \sigma^2 c_i Q c_i' < \alpha_i, \ \ \forall i \in \{1, \dots, m\},$$

where b_j denotes the jth column of B and c_i the ith row of C. □

We leave to the reader the formulation of a necessary and sufficient condition for the stability of the NCS (41) in Fig. 4 based on Theorem 12.

The *mean-square structured norm* of (42)–(43) is defined to be

$$\mu_{MS}(G, \mathbf{\Delta}) = \frac{1}{\sup\{\sigma^2 : \text{(42)–(43) is mean-square stable}\}}.$$

The quantity $1/\mu_{MS}(G, \mathbf{\Delta})$ can be viewed as a stability margin that measures how much (structured) stochastic uncertainty the system G can tolerate, since it is the largest value of σ^2 for which the interconnection is mean-square stable. The following

[7] Elia [8] actually considers more general LTI systems, which can have a strictly triangular direct feed-through term.

result relates the mean-square structured norm of a system with the mean-square norm of its transfer function. We recall that the *mean-square norm* of the transfer function $T(s)$ is given by

$$\|T(s)\|_{MS} := \max_{i\in\{1,\dots,m\}} \Big(\sum_{j=1}^{m} \|T_{ij}(s)\|_2^2\Big)^{\frac{1}{2}}, \qquad\qquad \|T_{ij}(s)\|_2^2 := \frac{1}{2\pi}\int_{-\infty}^{\infty} |T_{ij}(j\omega)|^2\, d\omega.$$

Theorem 13 ([8]): Under the assumptions of Theorem 12,

$$\mu_{MS}(G,\mathbf{\Delta}) = \inf_{D>0,\ D \text{ diagonal}} \|D^{-1}G(z)D\|_{MS},$$

where the infimum is taken over all positive definite $m \times m$ diagonal matrices □.

This result is used in Section IV for NCSs controller synthesis.

C. NCSs as delayed differential equations (sampling, delays, and dropouts)

Once again we go back to the one-channel feedback NCS in Fig. 4, with a plant/controller pair (12), for which the signal $y(t)$ is sampled at times $\{t_k : k \in \mathbb{N}\}$. In a lossless network, all the samples $\hat{y}_k = y_k = Cx(t_k)$ arrive at the destination with a (possibly variable) delay of $\tau_k \geq 0$, which leads to

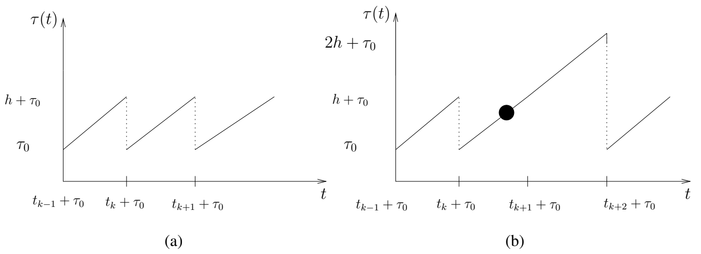
$$\hat{y}(t) = Cx(t_k), \quad \forall t \in [t_k + \tau_k, t_{k+1} + \tau_{k+1}),\ k \in \mathbb{N}. \tag{44}$$

In the previous sections, we proceeded by deriving discrete-time models for the evolution of the state of the NCS at sampling times. Instead, Yu et al. [76] propose to regard (44) as a delayed equation

$$\hat{y}(t) = Cx\big(t - \tau(t)\big), \quad \forall t \geq 0, \tag{45}$$

in which the delay $\tau(t)$ is time-varying and evolves according to

$$\tau(t) = t - t_k, \qquad\qquad \forall t \in [t_k + \tau_k, t_{k+1} + \tau_{k+1}),\ k \in \mathbb{N}.$$



(a) (b)

Fig. 10. Variable delay in an NCS for constant network delay $\tau_k = \tau_0$, $\forall k \in \mathbb{N}$ and periodic sampling with period $t_{k+1} - t_k = h$, $\forall k \in \mathbb{N}$ for (a) no packet dropout and (b) one packet dropout.

Fig. 10(a) shows the evolution of the time-varying delay $\tau(t)$ in (45) for the case of periodic sampling with period $h = t_{k+1} - t_k$, $\forall k \in \mathbb{N}$ and constant network delay $\tau_0 = \tau_k$, $\forall k \in \mathbb{N}$. In view of (45), this approach regards the overall NCS as a continuous-time delayed differential equation (DDE) of the form

$$\dot{x}(t) = Ax(t) + BCx\big(t - \tau(t)\big), \quad \forall t \geq 0, \tag{46}$$

where the time-varying delay $\tau(t)$ satisfies

$$\tau(t) \in [\tau_{\min}, \tau_{\max}), \quad \dot{\tau} = 1, \quad \forall t \geq 0,\ \textit{a.e.} \tag{47}$$

where

$$\tau_{\min} := \min_{k\in\mathbb{N}}\{\tau_k\}, \qquad\qquad \tau_{\max} := \max_{k\in\mathbb{N}}\{t_{k+1} - t_k + \tau_{k+1}\}. \tag{48}$$

An important advantage of characterizing an NCS as in (46)–(48) is that these equations are valid even when the delay exceeds the sampling interval. So in this section we shall not restrict our attention to delays smaller than one sampling interval.

As illustrated in Fig. 10(b), we can also view packet dropouts as a delay $\tau(t)$ that grows beyond the maximum in (48). This means that an NCS with a maximum number of consecutive dropouts equal to m is still a DDE like (46)–(47), but with

$$\tau_{\min} := \min_{k\in\mathbb{N}}\{\tau_k\}, \qquad\qquad \tau_{\max} := \max_{k\in\mathbb{N}}\{t_{k+1+m} - t_k + \tau_{k+1+m}\}.$$

The Lyapunov-Krasovskii and the Razumikhin Theorems [13, 55, 11] are the two main tools available to study the stability of DDEs of the form (46)–(47). However, the Lyapunov-Krasovskii Theorem generally leads to less conservative results. To formulate this theorem we need following notation: Given a constant $\tau_{\max} > 0$, a continuous signal $x : (-\tau_{\max}, \infty) \to \mathbb{R}^n$, and some time $t \in \mathbb{R}$, we denote by $x_t : [-\tau_{\max}, 0] \to \mathbb{R}^n$ the restriction of x to the interval $[t - \tau_{\max}, t]$ translated to $[-\tau_{\max}, 0]$, i.e., $x_t(s) = x(t+s)$, $\forall s \in [-\tau_{\max}, 0]$. The function x_t is an element of the Banach space $\mathcal{C}([-\tau_{\max}, 0], \mathbb{R}^n)$ of continuous functions from $[-\tau_{\max}, 0]$ to $\mathbb{R}^n$.

Theorem 14 (Lyapunov-Krasovskii [13]): The DDE

$$\dot{x}(t) = Ax(t) + A_d x\big(t - \tau(t)\big), \quad \forall t \geq 0,$$

with $\tau(t) \in [0, \tau_{\max}]$, $\forall t \geq 0$ and initial condition $x(t) = \psi(t)$, $\forall t \in [-\tau_{\max}, 0]$ is asymptotically stable if there exists a bounded quadratic *Lyapunov-Krasovskii functional* $V : \mathcal{C}([-\tau_{\max}, 0], \mathbb{R}^n) \to \mathbb{R}$ and a positive constant ϵ for which

$$V(\phi) \geq \epsilon \|\phi(0)\|^2, \qquad \frac{\mathrm{d}V(x_t)}{\mathrm{d}t}\Big|_{x_t = \phi} \leq -\epsilon \|\phi(0)\|^2, \qquad \forall \phi \in \mathcal{C}([-\tau_{\max}, 0], \mathbb{R}^n).$$

where the (total) derivative is taken along solutions to (46). □

To study the stability of (46), Yue et al. [78] use the following Lyapunov-Krasovskii functional

$$V(x_t) = x(t)' P x(t) + \int_{t-\tau_{\max}}^{t} \int_{s}^{t} \dot{x}(v)' T \dot{x}(v) dv ds,$$

with symmetric matrices $P > 0$ and $T > 0$. They show that the derivative of $V(x_t)$ is negative along solutions to (46) if

$$\begin{bmatrix} N_1 + N_1' - M_1 A - A' M_1' & N_2' - N_1 - A' M_2' - M_1 BK & N_3' - A' M_3' + M_1 + P & \tau_{\max} N_1 \\ * & -N_2 - N_2' - M_2 BK - K' B' M_2' & -N_3' + M_2 - K' B' M_3' & \tau_{\max} N_2 \\ * & * & M_3 + M_3' + \tau_{\max} T & \tau_{\max} N_3 \\ * & * & * & -\tau_{\max} T \end{bmatrix} < 0, \tag{49}$$

where N_i, M_i, $i \in \{1, 2, 3\}$ are slack matrix variables. This leads to the following result:

Theorem 15 ([78]): For a given scalar $\tau_{\max} > 0$, suppose that there exist square matrices N_i, M_i, $i \in \{1, 2, 3\}$ and symmetric matrices $P, T > 0$ such that (49) holds. Then the NCS (46)–(47) in Fig 4 is asymptotically stable as long as

$$t_{k+1+m} - t_k + \tau_{k+1+m} \leq \tau_{\max}, \quad \forall k \in \mathbb{N},$$

where m denotes the maximum number of consecutive dropouts. □

Yu et al. [76] also model one-channel feedback NCSs as DDEs, but they study their stability using the Razumikhin Theorem, which generally leads to more conservative results.

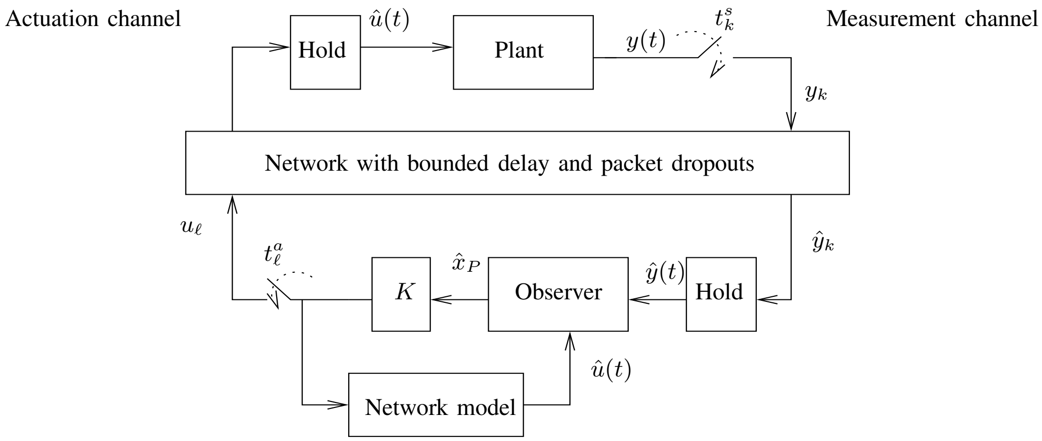


Fig. 11. Two channel feedback NCS with observer-based controller.

Naghshtabrizi and Hespanha [40] consider the two-channel feedback NCS in Fig. 11, in which a known plant

$$\dot{x}_P = A_P x_P + B_P \hat{u}, \qquad y = C_P x_P \tag{50}$$

is controlled by a remote observer-based controller that receives sensor data through a *sensor channel* and sends control signals to the actuators through an *actuation channel*. The output signal $y(t)$ is sampled at times t_k^s, $k \in \mathbb{N}$ and the samples $y_k := y(t_k^s)$, $k \in \mathbb{N}$ are sent through the sensor channel suffering a (possibly variable) delay of $\tau_k^s \geq 0$. In a lossless network, $\hat{y}(t)$ is therefore updated according to

$$\hat{y}(t) = y_k = C_P x_P(t_k^s), \quad \forall t \in [t_k^s + \tau_k^s, t_{k+1}^s + \tau_{k+1}^s), \ k \in \mathbb{N},$$

and if $m^s > 0$ sensor channel packets are dropped after the kth packet, the above equation holds $\forall t \in [t^s_k + \tau^s_k, t^s_{k+1+m^s} + \tau^s_{k+1+m^s})$.

Two alternative observer-based controllers are proposed: non-anticipative and anticipative. *Non-anticipative controllers* construct an estimate $\hat{x}_P$ of the plant' state x_P using

$$\dot{\hat{x}}_P(t) = A_P \hat{x}_P(t) + B_P \hat{u}(t) + L\big(\hat{y}(t) - C_P \hat{x}_P(t^s_k)\big), \quad \forall t \in [t^s_k + \tau^s_k, t^s_{k+1} + \tau^s_{k+1}),\ k \in \mathbb{N}, \tag{51}$$

for which the innovation term $\hat{y} - C_P \hat{x}_P(t^s_k)$ was chosen so that it is equal to zero whenever $x_P(t^s_k) = \hat{x}_P(t^s_k)$. Control updates $u_\ell := K \hat{x}_P(t^a_\ell)$ are sent through the actuation channel to the actuators at times t^a_ℓ, $\ell \in \mathbb{N}$ suffering a (possibly variable) delay of $\tau^a_\ell \geq 0$. In a lossless network, the control signal $\hat{u}(t)$ is therefore updated according to

$$\hat{u}(t) = u_\ell = K \hat{x}_P(t^a_\ell), \quad \forall t \in [t^a_\ell + \tau^a_\ell, t^a_{\ell+1} + \tau^a_{\ell+1}),\ \ell \in \mathbb{N},$$

which would hold over a longer interval if actuation channels packets were dropped.

Anticipative controllers attempt to compensate the sampling and delay introduced by the actuation channel. For simplicity, we assume that the actuation channel is sampled with period $h^a = t^a_{\ell+1} - t^a_\ell$, $\forall \ell \in \mathbb{N}$ and that its delay is constant and equal to $\tau^a = \tau^a_\ell$, $\forall \ell \in \mathbb{N}$. At each sampling time $t^a_\ell = \ell h^a$, $\ell \in \mathbb{N}$ the controller sends a time-varying control signal $u_\ell(\cdot)$ that should be used from the time $\ell h^a + \tau^a$ at which it arrives until the time $(\ell + 1)h^a + \tau^a$ at which the next control update will arrive. This leads to

$$\hat{u}(t) = u_\ell(t), \quad \forall t \in [\ell h^a + \tau^a, (\ell+1)h^a + \tau^a),\ \ell \in \mathbb{N}.$$

To stabilize (50), $u_\ell(t)$ should be equal to $-K\hat{x}_P(t)$, where $\hat{x}_P(t)$ is an estimate of $x_P(t)$. However, the estimates $\hat{x}_P(\cdot)$ needed in the interval $[\ell h^a + \tau^a, (\ell+1)h^a + \tau^a)$ must be available at the transmission time ℓh^a, which requires the control unit to estimate the plant' state up to $h^a + \tau^a$ time units into the future. In this case, the estimator (51) is of no use. Instead, an estimate $z(t)$ of $x_P(t + h^a + \tau^a)$ is constructed as follows:

$$\dot{z}(t) = A_P z(t) + B_P \hat{u}(t + h^a + \tau^a) + L\big(\hat{y}(t) - C_P z(t^s_k - h^a - \tau^a)\big), \quad \forall t \in [t^s_k + \tau^s_k, t^s_{k+1} + \tau^s_{k+1}),\ k \in \mathbb{N},$$

for which the innovation term $\hat{y} - C_P z(t^s_k - h^a - \tau^a)$ was chosen so that it is equal to zero whenever $x_P(t^s_k) = z\big(t^s_k - h^a - \tau^a\big)$. The signal $u_\ell(\cdot)$ sent at time ℓh^a {and to be used during the interval $[\ell h^a + \tau^a, (\ell+1)h^a + \tau^a)$} is then given by

$$u_\ell(t) = -Kz(t - h^a - \tau^a), \quad \forall t \in [\ell h^a + \tau^a, (\ell+1)h^a + \tau^a),\ \ell \in \mathbb{N},$$

which only requires knowledge of $z(\cdot)$ in $[(\ell-1)h^a, \ell h^a)$ and is therefore available at the transmission time ℓh^a. For anticipative controllers to be able to compensate for packet dropouts in the actuation channel, z would have to estimate x_P further into the future. Anticipative controllers send through the actuation channel actuation signals to be used during time intervals of duration h^a, therefore for these controllers the sample and hold blocks in Fig. 11 should be understood in a broad sense. In practice, the sample block would send over the network some parametric form of the control signal $u_\ell(\cdot)$ (e.g., the coefficients of a polynomial approximation to this signal).

Naghshtabrizi and Hespanha [40] write the closed-loop NCSs as DDEs for both the anticipative and the non-anticipative controllers. For an anticipative controller with no dropouts, this leads to

$$\dot{x}(t) = \begin{bmatrix} A_P - B_P K & 0 \\ 0 & A_P \end{bmatrix} x(t) + \begin{bmatrix} 0 & LC_P \\ 0 & -LC_P \end{bmatrix} x(t - \tau), \quad \forall t \geq 0, \tag{52}$$

where $x'(t) := \begin{bmatrix} z(t) & x_P(t + h^a + \tau^a) - z(t) \end{bmatrix}$, $\forall t \geq 0$ and

$$\tau(t) := t - t^s_k + h^a + \tau^a, \quad \forall t \in [t^s_k + \tau^s_k, t^s_{k+1+m} + \tau^s_{k+1+m}),\ k \in \mathbb{N}.$$

Moreover, if $m^s > 0$ sensor channel packets are dropped after the kth packet, this equation holds over the interval $[t^s_k + \tau^s_k, t^s_{k+1+m^s} + \tau^s_{k+1+m^s})$.

The "triangular" structure of (52) is unique to the anticipate controller. With this type of controller, if we choose K so that $A_P - B_P K$ is Hurwitz, asymptotic stability of the NCS is equivalent to the asymptotic stability of the (decoupled) dynamics of the error $e(t) := x_P(t + h^a + \tau^a) - z(t)$, $\forall t \geq 0$, which is given by the following DDE

$$\dot{e}(t) = A_P\, e(t) - LC_P\, e\big(t - \tau(t)\big), \quad t \geq 0,$$

with $\tau(t) \in [\tau_{\min}, \tau_{\max})$, $\dot{\tau} = 1$, $\forall t \geq 0$, a.e., where

$$\tau_{\min} := \min_{k \in \mathbb{N}} \{\tau^s_k + h^a + \tau^a\}, \qquad \tau_{\max} := \max_{k \in \mathbb{N}} \{t^s_{k+1+m^s} - t^s_k + \tau^s_{k+1+m^s} + h^a + \tau^a\},$$

where m^s denotes the maximum number of consecutive packet dropouts in the sensor channel. Naghshtabrizi and Hespanha [40] use the following Lyapunov-Krasovskii functional to analyze this system

$$V(e_t) = e(t)'P_1 e(t) + \int_{t-\tau_{\max}}^{t} \int_{s}^{t} \dot{e}'(v) R \dot{e}(v) dv ds + \int_{t-\tau_{\min}}^{t} e'(s) S e(s) ds,$$

where $P_1 > 0$, $R > 0$, $S > 0$. This leads to the following result:

Theorem 16 ([40]): Suppose that there exist symmetric matrices $P_1, S, R > 0$, square matrices P_2, P_3, Z_1, Z_2, and a (non-square) matrix T such that

$$\begin{bmatrix} \Psi & P'\begin{bmatrix} 0 \\ -LC_P \end{bmatrix} - T' \\ * & -S \end{bmatrix} < 0, \qquad \begin{bmatrix} R & \begin{bmatrix} 0 & -(LC_P)' \end{bmatrix} P \\ * & Z_2 \end{bmatrix} > 0, \qquad \begin{bmatrix} R & T \\ * & Z_1 \end{bmatrix} > 0, \tag{53}$$

where

$$P := \begin{bmatrix} P_1 & 0 \\ P_2 & P_3 \end{bmatrix}, \quad \Psi := P'\begin{bmatrix} 0 & I \\ A_P & -I \end{bmatrix} + \begin{bmatrix} 0 & I \\ A_P & -I \end{bmatrix}' P + \begin{bmatrix} S & 0 \\ 0 & \tau_{\max} R_i \end{bmatrix} + \tau_{\min} Z_1 + (\tau_{\max} - \tau_{\min}) Z_2 + \begin{bmatrix} T \\ 0 \end{bmatrix} + \begin{bmatrix} T \\ 0 \end{bmatrix}'. \tag{54}$$

Then the NCS with the anticipative controller (52), (47) is asymptotically stable as long as there are no dropouts in the actuation channel and

$$\tau_k^s + h^a + \tau^a \geq \tau_{\min}, \qquad t_{k+1+m^s}^s - t_k^s + \tau_{k+1+m^s}^s + h^a + \tau^a \leq \tau_{\max}, \qquad \forall k \in \mathbb{N},$$

where m^s denotes the maximum number of consecutive dropouts in the sensor channel. □

The reader is referred to [40] for an analogous result with a non-anticipative controller.

IV. Controller Synthesis

In this section, we discuss the design of feedback controllers for NCSs. Some of these results stem directly from the analysis methods presented in Section III.

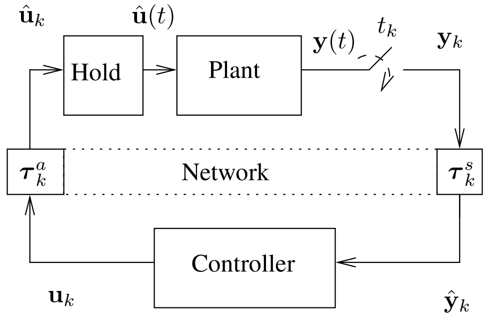


Fig. 12. Two-channel feedback NCS considered by Nilsson [46].

A. Sampling and delay

Nilsson et al. [47, 49, 48, 51, 50, 46] consider the two-channel feedback NCS in Fig. 12. The plant is an LTI system with the following state-space model

$$\dot{\mathbf{x}} = A\mathbf{x} + B\hat{\mathbf{u}} + \mathbf{w}, \qquad \mathbf{y} = C\mathbf{x} + \mathbf{v}, \tag{55}$$

where $\mathbf{v}$ and $\mathbf{w}$ are zero-mean uncorrelated white noise processes. The output signal $\mathbf{y}(t)$ is sampled periodically at times $t_k := kh, \forall k \in \mathbb{N}$ and the samples $\mathbf{y}_k := \mathbf{y}(t_k), \forall k \in \mathbb{N}$ are sent through the network. After a (possibly varying) delay of $\boldsymbol{\tau}_k^s \geq 0$, these samples reach a remote controller that immediately computes control updates $\mathbf{u}_k$ and sends them to the network. These updates reach the actuators after a (possibly varying) delay of $\boldsymbol{\tau}_k^a \geq 0$. Assuming that there are no packet dropouts in either of the network channels, this leads to

$$\hat{\mathbf{u}}(t) = \begin{cases} \mathbf{u}_{k-1} & t \in [t_k, t_k + \boldsymbol{\tau}_k^s + \boldsymbol{\tau}_k^a) \\ \mathbf{u}_k & t \in [t_k + \boldsymbol{\tau}_k^s + \boldsymbol{\tau}_k^a, t_{k+1}) \end{cases}, \tag{56}$$

where we assumed that the total delay is smaller than one sampling interval, i.e., that $\boldsymbol{\tau}_k^s + \boldsymbol{\tau}_k^a < h$, $\forall k \in \mathbb{N}$. Any computation time needed by the remote controller can be incorporated in the network delay $\boldsymbol{\tau}_k^a$.

The delays $\boldsymbol{\tau}^s_k$ and $\boldsymbol{\tau}^a_k$ are assumed to be independent random variables with known probability distributions. Moreover, it is assumed that all data sent through the network is time-stamped, which means that the controller knows the value of the delay $\boldsymbol{\tau}^s_k$ when the kth measurement $\hat{\mathbf{y}}_k$ arrives.

Defining $\mathbf{x}_k := \mathbf{x}(t_k)$, $\forall k \in \mathbb{N}$ and applying the variation of constants formula to (55)–(56), we conclude that

$$\begin{aligned}\mathbf{x}_{k+1} &= e^{Ah}\mathbf{x}_k + e^{A(h-\boldsymbol{\tau}^s_k-\boldsymbol{\tau}^a_k)}\Gamma(\boldsymbol{\tau}^s_k+\boldsymbol{\tau}^a_k)B\mathbf{u}_{k-1} + \Gamma(h-\boldsymbol{\tau}^s_k-\boldsymbol{\tau}^a_k)B\mathbf{u}_k + \boldsymbol{v}_k,\\ \mathbf{y}_k &= C\mathbf{x}_k + \mathbf{w}_k,\end{aligned}$$

where $\Gamma(s) := \int_0^s e^{Az}dz$, $\forall s \in \mathbb{R}$ and $\mathbf{v}_k$, $\mathbf{w}_k$ are uncorrelated zero-mean white noise processes.

Assuming state-feedback (i.e., $\mathbf{y}_k = \mathbf{x}_k$, $\forall k \in \mathbb{N}$), Nilsson et al. [51, 49] show that the optimal control $\mathbf{u}_k$ that minimizes

$$J_k = \mathrm{E}\left\{\mathbf{x}'_N Q_N \mathbf{x}_N + \sum_{j=k}^{N-1} \begin{bmatrix}\mathbf{x}_j\\ \mathbf{u}_j\end{bmatrix}' Q \begin{bmatrix}\mathbf{x}_j\\ \mathbf{u}_j\end{bmatrix}\right\},$$

with

$$Q_N \geq 0, \qquad Q := \begin{bmatrix} Q_{11} & Q_{12}\\ * & Q_{22}\end{bmatrix} \geq 0, \quad Q_{22} > 0,$$

is of the form

$$\mathbf{u}_k = -L_k(\boldsymbol{\tau}^s_k)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_{k-1}\end{bmatrix}, \quad \forall k \in \mathbb{N}. \tag{57}$$

Hence the optimal controller with full state information is a linear but $\boldsymbol{\tau}^s_k$-dependent function of the current state and previous control signal. The computation of the matrix gain $L_k(\boldsymbol{\tau}^s_k)$ requires the solution of a backwards-in-time Riccati equation that involves the computation of expectations with respect to the random variables $\boldsymbol{\tau}^s_k$ and $\boldsymbol{\tau}^a_k$.

In practice, the delays $\boldsymbol{\tau}^s_k, \boldsymbol{\tau}^a_k$, $\forall k \in \mathbb{N}$ are often correlated because they depend on the network load, which typically varies at time scales slower than the sampling interval h. To account for this, Nilsson and Bernhardsson [48] consider three alternative distributions for the delay and model the transitions between the distributions using a three-state Markov chain. Each state of the Markov chain would correspond to a particular network load (low, medium, or high). In this case the optimal control strategy is of the form

$$\mathbf{u}_k = -L_k(\boldsymbol{\tau}^s_k, \mathbf{r}_k)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_{k-1}\end{bmatrix}, \tag{58}$$

where now the matrix gain $L_k(\boldsymbol{\tau}^s_k, \mathbf{r}_k)$ depends both on the delay $\boldsymbol{\tau}^s_k$ and the current state $\mathbf{r}_k$ of the Markov chain. To implement this control law, the remote controller must know the current value of $\mathbf{r}_k$.

The main difficulty in using the optimal controllers (57), (58) is the computation of the matrix gains $L_k(\cdot)$. However, when stationary values for these gains exist, they can be computed offline and stored in a table, which is indexed in real time by the current value of the delay $\boldsymbol{\tau}^s_k$ and network state $\mathbf{r}_k$. Nilsson et al. [51, 49] also propose to use suboptimal controllers that are more attractive from a computational perspective. The same authors [51, 49, 48] extended this work for the output feedback case. They showed that the separation principle holds and that the optimal control can be obtained by replacing $\mathbf{x}_k$ in (57) and (58) by an estimate $\hat{\mathbf{x}}$ computed using a time-varying Kalman filter.

Nilsson et al. [50] further extended this work by considering non-periodic sampling, timeouts on the period during which a controller waits for new measurements before sending a new control command, and asynchronous loops in which the clocks used for time-stamping are not synchronized and run at different speeds. Lincoln and Bernhardsson [28] also extended these results to situations in which the sum of the delays exceeds one sampling interval but remains bounded.

B. Packet dropout

Deterministic dropout rates: Yu et al. [77] consider the one-channel feedback NCS shown in Fig. 13 with a LTI plant

$$\dot{x} = Ax + Bu, \qquad\qquad y = Cx,$$

whose output is sampled periodically at times $\{t_k := kh : k \in \mathbb{N}\}$ and the samples $y_k := y(t_k)$ are sent through the network. It is assumed that the delay introduced by the network is negligible but packets may be dropped. The network output $\hat{y}_k$ is kept equal to its previous value when the packet containing the sample y_k is dropped as in (33). Denoting by κ_j, $j \in \mathbb{N}$ the indexes of the packets that are *not dropped*, $\hat{y}_k$ remains equal to y_k from $k = \kappa_j$ until $k = \kappa_{j+1} - 1$, i.e.,

$$\hat{y}_k = y_{\kappa_j}, \quad \forall k \in \{\kappa_j, \kappa_j + 1, \ldots, \kappa_{j+1} - 1\}.$$

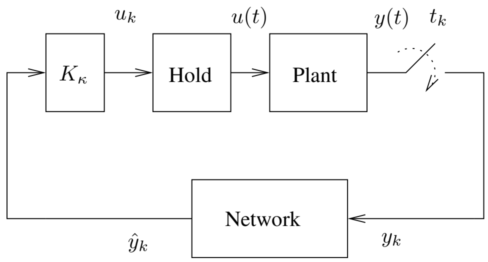


Fig. 13. One-channel feedback NCS with switching controller.

Yu et al. [77] use a static output-feedback controller, whose gain changes depending on whether or not a packet is dropped. More precisely, they use

$$u_k = K_{k-\kappa_j}\hat{y}_k = K_{k-\kappa_j} y_{\kappa_j}, \quad \forall k \in \{\kappa_j, \kappa_j + 1, \dots, \kappa_{j+1} - 1\},$$

where the matrix gain K_0 is used when the sample y_k has not been dropped, K_1 is used when y_k has been dropped but y_{k-1} has not, K_2 is used when y_k and y_{k-1} have been dropped but y_{k-2} has not, and so on. The control signal $u(t)$ is kept constant between samples:

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}), \quad \forall k \in \mathbb{N}.$$

Defining $z_j := x(t_{\kappa_j})$, $\forall j \in \mathbb{N}$ and using the variation of constants formula, we conclude that

$$z_{j+1} = A_{\kappa_{j+1}-\kappa_j-1} z_j, \qquad A_d := e^{Ah(d+1)} + \sum_{i=0}^{d} e^{Ah(d-i)}\Gamma(h)BK_iC, \quad \forall d \in \mathbb{N}, \tag{59}$$

where $\Gamma(s) := \int_0^s e^{Az}dz$, $\forall s \in \mathbb{R}$. Assuming that the maximum number of consecutive dropouts is equal to m, we have $0 \le \kappa_{j+1} - \kappa_j - 1 \le m$, $\forall j \in \mathbb{N}$ and we can view (59) as a linear system that switches among the matrices $A_0, \dots, A_m$. The stability of such system can be established using a common quadratic Lyapunov function $V(z) := z'S^{-1}z$, leading to the following theorem:

Theorem 17 ([77]): Suppose that there exists matrices M, Y_i, $i \in \{0, 1, \dots, m\}$ and a symmetric matrix $S > 0$ such that[8] $MC = CS$ and

$$\begin{bmatrix} -S & * \\ e^{Ah(d+1)}S + \sum_{i=0}^{d} e^{Ah(d-i)}\Gamma(h)BY_iC & -S \end{bmatrix} < 0, \quad \forall d \in \{0, 1, \dots, m\}.$$

Then the NCS (59) in Fig. 13 is exponentially stable for the controller gain $K_i = Y_iM^{-1}$, $\forall i \in \{0, 1, \dots, m\}$. □

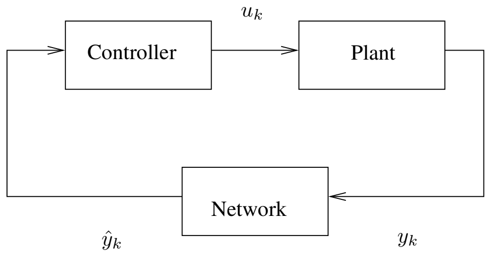


Fig. 14. One-channel feedback NCS with a discrete-time plant and controller.

Stochastic (Markovian) dropouts: Seiler and Sengupta [57] consider the design of a switching controller for the one-channel feedback NCSs in Fig. 14. The plant and the controller are modeled by discrete-time LTI systems of the form

$$x_{k+1} = Ax_k + Bu_k, \qquad y_k = Cx_k, \tag{60a}$$

$$\bar{x}_{k+1} = A_{\boldsymbol{\theta}_k}\bar{x}_k + B_{\boldsymbol{\theta}_k}\hat{y}_k, \qquad u_k = C_{\boldsymbol{\theta}_k}\hat{x}_k, \tag{60b}$$

where y_k and $\hat{y}_k$ are the input and the output of the network. These variables are related by (33), where $\boldsymbol{\theta}_k$ is a Bernoulli process with probability of dropout (i.e., $\boldsymbol{\theta}_k = 0$) equal to $p \in [0, 1)$. They allow the controller state-space model to adapt to the occurrence of dropouts, hence its matrix gains depend on the value of $\boldsymbol{\theta}_k \in \{0, 1\}$.

[8] In the state feedback case, C is the identity matrix and we simply have $M = S$.

Defining $z'_k := \begin{bmatrix} x'_k & \bar{x}'_k & u'_k \end{bmatrix}$, the system can be written in compact form as (34), with

$$\Phi_\theta := \begin{bmatrix} A & BC_\theta & 0 \\ B_\theta C & A_\theta & (1-\theta)B_\theta \\ \theta C & 0 & (1-\theta)I \end{bmatrix}, \quad \forall \theta \in \{0,1\}.$$

Since this system is an MJLS, the inequality (39) in Theorem 11 provides a necessary and sufficient condition for the exponentially mean-square stability of the NCS. For given controller matrices $A_\theta, B_\theta, C_\theta$, $\theta \in \{0,1\}$, the inequality (39) is a (convex) LMI on the unknown Z. However, if the controller matrices are also taken as unknowns we have a bilinear matrix inequality (BMI), leading to a feasibility problem that is generally not convex and therefore not numerically tractable. However, it is possible to obtain an equivalent LMI that leads to a tractable feasibility problem.

Theorem 18 ([57]): Suppose that there exist matrices $L_\theta, F_\theta, W_\theta$, $\theta \in \{0,1\}$, $Y = Y'$, $X = X'$ such that

$$\begin{bmatrix} \left[\begin{smallmatrix} Y & I \\ I & X \end{smallmatrix}\right] & \sqrt{p} \left[\begin{smallmatrix} Y\bar{A}_0 + L_0\bar{C}_0 & W_0 \\ \bar{A}_0 & \bar{A}_0 X + \bar{B}F_0 \end{smallmatrix}\right]' & \sqrt{1-p} \left[\begin{smallmatrix} Y\bar{A}_1 + L_1\bar{C}_1 & W_1 \\ \bar{A}_1 & \bar{A}_1 X + \bar{B}F_1 \end{smallmatrix}\right]' \\ * & \left[\begin{smallmatrix} Y & I \\ I & X \end{smallmatrix}\right] & \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right] \\ * & * & \left[\begin{smallmatrix} Y & I \\ I & X \end{smallmatrix}\right] \end{bmatrix} > 0,$$

where

$$\bar{A}_\theta := \begin{bmatrix} A & 0 \\ \theta C & (1-\theta)I \end{bmatrix}, \qquad \bar{B}_\theta := \begin{bmatrix} B \\ 0 \end{bmatrix}, \qquad \bar{C}_\theta := \begin{bmatrix} \theta C & (1-\theta)I \end{bmatrix}, \qquad \forall \theta \in \{0,1\}.$$

Then the NCS (60) in Fig. 14 is exponentially mean-square stable for the following controller realizations:

$$A_\theta = -Y^{-1}\big(Y\bar{A}_\theta X + Y\bar{B}_\theta F_\theta + L_\theta \bar{C}_\theta X - W_\theta\big)\big(Y^{-1} - X\big)^{-1}, \quad B_\theta = Y^{-1}L_\theta, \quad C_\theta = F_\theta\big(Y^{-1} - X\big)^{-1}, \quad \forall \theta \in \{0,1\}.$$

□

Seiler and Sengupta [58] further extended these results to the synthesis of an H_∞ controller that rejects plant disturbances. De Farias et al. [6] consider mixed H_∞ / H_2 synthesis for MJLSs in a continuous-time setting.

Fading networks: Within the fading networks formulation, Elia [8] proposes to design controllers for NCSs that minimize the mean-square structured norm of the nominal system $G(z)$ in Fig. 9(b). Such controllers would maximize the NCS's tolerance with respect to packet dropouts.

Elia [8] shows that minimum achievable mean-square structure norm is the solutions to either of the following optimizations

$$\mu^*_{MS} = \inf_K \inf_{D>0,\ D \text{ diagonal}} \|D^{-1}G(P,K)D\|^2_{MS} \tag{61a}$$

$$= \inf_{D>0,\ D \text{ diagonal}} \inf_K \|D^{-1}G(P,K)D\|^2_{MS}, \tag{61b}$$

where the infimum over K is taken over the set of LTI stabilizing controllers; the infimum over δ is taken over all positive diagonal matrices; and $G(P,K)$ denotes the interconnection of the plant P, the controller K, and the mean network N shown in Fig. 9(b). As in many structured robust control problems, the search for the optimal controller that minimizes (61a) is not convex on the diagonal matrix D. However, for every fixed D, the controller optimization in (61b) is convex and can be reduced to an LMI [8, Theorem 6.6]. Heuristic methods such as the D-K iteration can explore this to find suboptimal solutions to (61a) [7].

C. NCSs as delay differential equations (sampling, delays, and dropouts)

In Section III-C we presented matrix inequalities from which one could conclude stability for NCSs modeled as delay differential equations. For given controllers gains K and L, the inequalities (49) and (53) in Theorems 15 and 16, respectively, are linear on all the matrices that do not depend on the (known) plant model. The stability of the NCS can therefore be verified by studying the feasibility of a (convex) LMI. However, if the controllers gains are also taken as unknowns, we obtain a BMI and therefore these matrix inequalities are not directly suitable for controller synthesis.

Yue et al. [78] utilize Theorems 15 as the basis for a numerically tractable controller synthesis procedure. They require the matrices M_1, M_2, M_3 in (49) to further satisfy

$$M_1 = M'_1 > 0, \qquad M_2 = \rho_2 M_1, \qquad M_3 = \rho_3 M_1, \tag{62}$$

for some constants $\rho_2, \rho_3 \in \mathbb{R}$. They then make the (bijective) change of variables

$$X := M^{-1}, \quad Y := KX', \quad \tilde{P} := XPX', \quad \tilde{T} := XTX', \quad \tilde{N}_i := XN_iX', \quad \tilde{M}_i := XM_iX', \quad \forall i \in \{1,2,3\}.$$

Pre- and post-multiplying (49) by $\operatorname{diag}(X\,X\,X\,X)$ and its transpose, respectively, yields

$$\begin{bmatrix} \tilde{N}_1 + \tilde{N}_1' - AX' - XA' & \tilde{N}_2' - \tilde{N}_1 - \rho_2 XA' - BY & \tilde{N}_3' - \rho_3 XA' + X' + \tilde{P} & \tau_{\max}\tilde{N}_1 \\ * & -\tilde{N}_2 - \tilde{N}_2' - \rho_2 BY - \rho_2 Y'B' & -\tilde{N}_3' + \rho_2 X' - \rho_3 Y'B' & \tau_{\max}\tilde{N}_2 \\ * & * & \rho_3 X + \rho_3 X' + \tau_{\max}\tilde{T} & \tau_{\max}\tilde{N}_3 \\ * & * & * & -\tau_{\max}\tilde{T} \end{bmatrix} < 0,$$

which is an LMI on the unknowns $X, Y, \tilde{P}, \tilde{T}$ and $\tilde{N}_i$, $i \in \{1,2,3\}$. Moreover, the controller gain can be recovered using $K = YX^{-1}$. This procedure introduces some conservativeness because it will not find controller gains K for which (49) holds for matrices M_2 and M_3 that are not scalar multiples of M_1, as in (62).

A simple but conservative way to make the matrix inequalities in Theorem 16 suitable for controller synthesis consists of requiring that

$$P_2 > 0, \qquad\qquad P_3 = \rho P_2,$$

for some positive constant $\rho > 0$ and making the (bijective) change of variables $Y = P_2 L$, which transforms (53) into

$$\begin{bmatrix} \Psi & -\begin{bmatrix} YC_P \\ \rho Y C_P \end{bmatrix} - T' \\ * & -S \end{bmatrix} < 0, \qquad \begin{bmatrix} R & -\begin{bmatrix} C_P'Y' & \rho C_P'Y' \end{bmatrix} \\ * & Z_2 \end{bmatrix} > 0, \qquad \begin{bmatrix} R & T \\ * & Z_1 \end{bmatrix} > 0,$$

with Ψ given by (54). This inequality is linear in the unknowns $P_1, P_2, S, R, Z_1, Z_2, T, Y$ and can therefore be solved using efficient numerical algorithms. The observer gain is found using $L = P_2^{-1}Y$. This procedure introduces some conservativeness because it restricts P_3 to be a scalar multiple of P_2. Naghshtabrizi and Hespanha [40] use the linear cone complementarity algorithm introduced by Ghaoui et al. [12] to design the controller gains L and K for the anticipative or non-anticipative controllers in Section III-C. The use of the cone complementarity algorithm avoids introducing additional conservativeness in going from a matrix inequality that is only appropriate for analysis to another matrix inequality that is appropriate for controller synthesis.

V. Conclusion and future direction of research

We presented several recent results on estimation, analysis, and controller synthesis for NCSs. The materials surveyed address different aspects of the limitations imposed by the use of communication channels to connect elements of NCSs. We focused on limitations in terms of packet-rate, sampling, network delay, and packet-dropout. In Section II, we addressed the state estimation of a remote plant over a channel in which some of the packets are lost in a random fashion. We considered both single and multiple sensors. Section III covered a collection of results to determine the closed-loop stability of NCSs in the presence of network sampling, delays, and packet dropouts. We considered a variety of assumptions on the plant and network effects, which led to closed-loop models represented by Markovian jump linear systems (MJLSs), linear time-varying systems, switched systems, nonlinear systems with resets, asynchronous dynamical systems (ADSs), linear time-invariant systems with stochastic structured uncertainty, and linear systems with delayed inputs. Many of the results presented rely on Lyapunov-based techniques, and only provide sufficient conditions for stability of the NCS. For linear systems, these results usually translate into linear matrix inequalities (LMIs), which generally become bilinear matrix inequalities (BMIs) in design problems for which the controller needs to be viewed as an unknown (cf. Section IV).

This survey did not addressed a few important issues in NCSs, such as bit-rate and quantization. The problem of determining the minimum bit-rate needed for stabilization has been solved exactly for linear plants, but only conservative results have been obtained for nonlinear plants. Quantization becomes especially important for networks designed to carry very small packets with little overhead, because for such networks one can save bandwidth by encoding measurements or actuation signals with a small number of bits.

NCS have been attracting significant interest in the past few years and will continue to do so for the years to come. With the advent of cheap, small, and low-power processors with communication capabilities, it has become possible to endow sensor and actuators with processing power and the ability to communicate with remove controllers through multi-purpose networks. In view of this, we conjecture that in the near future NCSs will become the norm, replacing the current fixed-rate digital control systems that rely on dedicated connections between sensors, controllers and actuators.

Results are still lacking to overcome several of the challenges raised by NCSs. Among these we highlight the following: There has been significant work on NCSs with variable sampling rate, but most results investigate the stability for a given wost-case interval between consecutive sampling times. This generally leads to conservative results that could be improved by taking into account a stochastic characterization for the inter-sampling times. Most work has been devoted to determining the stability of NCSs, whereas issues related to performance have been somewhat neglected. The design of controllers for NCSs has also been overlooked, as many researchers start with a controller that has been designed ignoring the challenges introduced by NCSs and then investigate to what extent such controllers can guarantee stability in spite of the network.

References

[1] K. Åström and B. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proc. of the 41th Conf. on Decision and Contr.*, volume 2, pages 2011–2016, Dec. 2002.

[2] M. S. Branicky. Stability of hybrid systems: State of the art. In *Proc. of the 36th Conf. on Decision and Contr.*, volume 1, pages 120–125, Dec. 1997.

[3] 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.

[4] J. H. Braslavsky, R. H. Middleton, and J. S. Freudenberg. Feedback stabilization over signal-to-noise ratio constrained channels. In *Proc. of the 2004 Amer. Contr. Conf.*, pages 4903–4908, June 2004.

[5] O. L. V. Costa and M. D. Fragoso. Stability results for discrete-time linear systems with Markovian jumping parameters. *Journal of mathematical analysis and applications*, 179(1):154–178, Oct 1993.

[6] D. P. De Farias, J. C. Geromel, J. B. R. Do Val, and O. L. V. Costa. Output feedback control of markov jump linear systems in continuous-time. *IEEE Trans. on Automat. Contr.*, 45(5):944 – 949, May 2000.

[7] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory*. Number 36 in Texts in Applied Mathematics. Springer, New York, 1999.

[8] N. Elia. Remote stabilization over fading channels. *Syst. & Contr. Lett.*, 54(3):237–249, Mar. 2005.

[9] N. Elia and S. K. Mitter. Stabilization of linear systems with limited information. *IEEE Trans. on Automat. Contr.*, 46 (9):1384–1400, Sept. 2001.

[10] Y. Fang and K. A. Loparo. Stochastic stability of jump linear systems. *IEEE Trans. on Automat. Contr.*, 47(7):1204–1208, July 2002.

[11] E. Fridman and U. Shaked. A descriptor system approach to H_∞ control of linear time-delay systems. *IEEE Trans. on Automat. Contr.*, 47(2):253 – 270, Feb. 2002.

[12] L. E. Ghaoui, F. Oustry, and M. AitRami. A cone complementarity linearization algorithm for static output-feedback and related problems. *IEEE Trans. on Automat. Contr.*, 42(8):1171–1176, Aug. 1997.

[13] K. Gu, V. L. Kharitonov, and J. Chen. *Stability of Time-delay Systems*. Birkhauser, 1st edition, 2003.

[14] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, Feb. 2006.

[15] C. N. Hadjicostis and R. Touri. Feedback control utilizing packet dropping network links. In *Proc. of the 41th Conf. on Decision and Contr.*, volume 2, pages 1205–1210, Dec. 2002.

[16] A. Hassibi, S. P. Boyd, and J. P. How. Control of asynchronous dynamical systems with rate constraints on events. In *Proc. of the 38th Conf. on Decision and Contr.*, volume 2, pages 1345–1351, Dec. 1999.

[17] J. P. Hespanha, M. L. McLaughlin, and G. Sukhatme. Haptic collaboration over the Internet. In *Proc. of the 5th Phantom Users Group Workshop*, Oct. 2000.

[18] J. P. Hespanha, A. Ortega, and L. Vasudevan. Towards the control of linear systems with minimum bit-rate. In *Proc. of the Int. Symp. on the Mathematical Theory of Networks and Syst.*, Aug. 2002.

[19] K. Hikichi, H. Morino, I. Arimoto, K. Sezaki, and Y. Yasuda. The evaluation of delay jitter for haptics collaboration over the Internet. In *IEEE Global Telecomm. Conf. (GLOBECOM)*, volume 2, pages 1492 – 1496, Nov. 2002.

[20] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice-Hall, Upper Saddle River, New Jersey, 2000.

[21] H. K. Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle River, NJ, 1996.

[22] F. Kozin. A survey of stability of stochastic systems. *Automatica*, 5:95–112, 1969.

[23] F.-L. Lian. *Analysis, design, modeling, and control of networked control systems*. PhD thesis, University of Michigan, 2001.

[24] D. Liberzon and J. P. Hespanha. Stabilization of nonlinear systems with limited information feedback. *IEEE Trans. on Automat. Contr.*, 50(6):910–915, June 2005.

[25] H. Lin and P. J. Antsaklis. Persistent disturbance attenuation properties for networked control systems. In *Proc. of the 43th Conf. on Decision and Contr.*, volume 1, pages 953–958, Dec 2004.

[26] H. Lin, G. Zhai, and P. J. Antsaklis. Robust stability and disturbance attenuation analysis of a class of networked control systems. In *Proc. of the 42th Conf. on Decision and Contr.*, volume 2, pages 1182–1187, Dec. 2003.

[27] H. Lin, G. Zhai, L. Fang, and P. J. Antsaklis. Stability and $\mathcal{H}_\infty$ performance preserving scheduling policy for networked control systems. In *Proc. of the 16th World Congress of Int. Federation of Automat. Contr.*, July 2005.

[28] B. Lincoln and B. Bernhardsson. Optimal control over networks with long random delays. In *Proc. of the 14th Int. Symp. of Mathematical Th. of Networks and Syst.*, 2000.

[29] X. Liu and A. Goldsmith. Kalman filtering with partial observation losses. In *Proc. of the 43th Conf. on Decision and Contr.*, pages 4180–4186, dec 2004.

[30] R. Luck and A. Ray. An observer-based compensator for distributed delays. *Automatica*, 26(5):903–908, Sept. 1990.

[31] N. C. Martins and M. A. Dahleh. Feedback control in the presence of noisy channels: “Bode-like” fundamental limitations of performance. Submitted for publication, Oct. 2004.

[32] A. Matveev and A. Savkin. The problem of state estimation via asynchronous communication channels with irregular transmission times. *IEEE Trans. on Automat. Contr.*, 48(4):670–676, Apr. 2003.

[33] C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian. Remote surgery case: robot-assisted teleneurosurgery. In *IEEE Int. Conf. on Robot. and Auto. (ICRA'04)*, volume 1, pages 819–823, Apr. 2004.

[34] L. A. Montestruque and P. J. Antsaklis. Model-based networked control systems: Necessary and sufficient conditions for stability. In *10th Mediterranean Contr. Conf. and Auto., Lisbon, Portugal*, 2002.

[35] L. A. Montestruque and P. J. Antsaklis. State and output feedback control in model-based networked control systems. In *Proc. of the 41th Conf. on Decision and Contr.*, volume 2, pages 1620 – 1625, Dec. 2002.

[36] L. A. Montestruque and P. J. Antsaklis. On the model-based control of networked systems. *Automatica*, 39(10):1837–1843, Oct. 2003.

[37] L. A. Montestruque and P. J. Antsaklis. Stochastic stability for model-based networked control systems. In *Proc. of the 2003 Amer. Contr. Conf.*, volume 5, pages 4119 – 4124, June 2003.

[38] L. A. Montestruque and P. J. Antsaklis. Stability of model-based networked control systems with time-varying transmission times. *IEEE Trans. on Automat. Contr.*, 49(9):1562–1572, Sept. 2004.

[39] R. M. Murray, K. J. Åström, S. P. Boyd, R. W. Brockett, and G. Stein. Control in an information rich world. *IEEE Contr. Syst. Mag.*, 23(2):20–33, Apr. 2003.

[40] P. Naghshtabrizi and J. 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.

[41] G. N. Nair and R. J. Evans. Exponential stabilisability of finite-dimensional linear systems with limited data rates. *Automatica*, 39(4):585–593, Apr. 2003.

[42] G. N. Nair, R. J. Evans, I. M. Y. Mareels, and W. Moran. Topological feedback entropy and nonlinear stabilization. *IEEE Trans. on Automat. Contr.*, 49(9):1585–1597, Sept. 2004.

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

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

[45] D. Nesic and A. R. Teel. A note on input-to-state stability of networked control systems. In *Proc. of the 43th Conf. on Decision and Contr.*, volume 5, pages 4613 – 4618, Dec 2004.

[46] J. Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, Jan. 1998.

[47] J. Nilsson and B. Bernhardsson. Analysis of real-time control systems with time delays. In *Proc. of the 35th Conf. on Decision and Contr.*, volume 3, pages 3173–3178, Dec. 1996.

[48] J. Nilsson and B. Bernhardsson. LQG control over a Markov communication network. In *Proc. of the 36th Conf. on Decision and Contr.*, volume 5, pages 4586–4591, Dec. 1997.

[49] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. In *Proc. of the 13th World Congress of Int. Federation of Automat. Contr.*, 1996.

[50] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Some topics in real-time control. In *Proc. of the 1998 Amer. Contr. Conf.*, volume 4, pages 2386–2390, June 1998.

[51] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1):57–64, Jan. 1998.

[52] 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.

[53] B. Oksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, Berlin, 2000.

[54] L. L. Peterson and B. S. Davie. *Computer Networks, A System Approach*. Morgan Kaufmann Publishers, 3 edition, 2003.

[55] J. P. Richard. Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39(10): 1667–1694, Oct. 2003.

[56] A. Sahai. *Anytime Information Theory*. PhD thesis, MIT, Cambridge, MA, 2001.

[57] P. Seiler and R. Sengupta. Analysis of communication losses in vehicle control problems. In *Proc. of the 2001 Amer. Contr. Conf.*, volume 2, pages 1491–1496, June 2001.

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

[59] S. Shirmohammadi and N. H. Woo. Evaluating decorators for haptic collaboration over the Internet. In *The 3rd IEEE Int. Workshop on Haptic, Audio and Visual Env. and their Applic.*, pages 105–109, Oct. 2004.

[60] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Trans. on Automat. Contr.*, 49(9):1453–1464, Sept. 2004.

[61] S. C. Smith and P. Seiler. Estimation with lossy measurements: jump estimators for jump systems. *IEEE Trans. on Automat. Contr.*, 48(12):2163–2171, Dec. 2003.

[62] W. Stallings. *Data and Computer Communications*. Prentice Hall, 6th edition, 2000.

[63] 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.

[64] G. Tang and L. Guo. Adaptive stabilization of unknown linear systems under communication constraints. Submitted to the Chinese Control Conference (in Chinese), Mar. 2005.

[65] S. Tatikonda. *Control Under Communication Constrains*. PhD thesis, MIT, Cambridge, MA, 2000.

[66] S. Tatikonda and S. Mitter. Control under communication constraints. *IEEE Trans. on Automat. Contr.*, 49(7):1056–1068, July 2004.

[67] TBA. TBA. To appear.

[68] G. C. Walsh, O. Beldiman, and L. G. Bushnell. Asymptotic behavior of nonlinear networked control systems. *IEEE Trans. on Automat. Contr.*, 46(7):1093–1097, July 2001.

[69] 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.

[70] 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 2001.

[71] W. S. Wong and R. W. Brockett. Systems with finite communication bandwidth constraints–II: Stabilization with limited information feedback. *IEEE Trans. on Automat. Contr.*, 44(5), May 1999.

[72] Y. Xu and J. P. Hespanha. Communication logics for networked control systems. In *Proc. of the 2004 Amer. Contr. Conf.*, volume 1, pages 572–577, June 2004.

[73] Y. Xu and J. P. Hespanha. Optimal communication logics for networked control systems. In *Proc. of the 43th Conf. on Decision and Contr.*, pages 3527–3532, Dec. 2004.

[74] Y. Xu and J. P. Hespanha. Estimation under uncontrolled and controlled communications in networked control systems. In *Proc. of the 44th Conf. on Decision and Contr.*, pages 842–847, June 2005.

[75] 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. Contr. Syst. Technol*, 10(4):503–518, July 2002.

[76] M. Yu, L. Wang, T. Chu, and F. Hao. An LMI approach to networked control systems with data packet dropout and transmission delays. *J. of Hybrid Syst.*, 3(11), Nov. 2004.

[77] M. Yu, L. Wang, G. Xie, and T. Chu. Stabilization of networked control systems with data packet droupout via switched system approach. In *IEEE Int. symp. on computer aided control systems design, Taipei, Taiwan*, pages 362–367, Sept. 2004.

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

[79] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel. Qualitative analysis of discrete-time switched systems. In *Proc. of the 2002 Amer. Contr. Conf.*, volume 3, pages 1880–1885, June 2002.

[80] W. Zhang and M. S. Branicky. Stability of networked control systems with time-varying transmission period. In *Allerton Conf. of Communication, contr. and computing, Urbana, IL*, Oct. 2001.

[81] W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Contr. Syst. Mag.*, 21(1): 84–99, Feb. 2001.