Handbook of Networked and Embedded Control Systems, Birkhauser, Boston, 2005.

Feedback Control with Communication Constraints *

Dimitrios Hristu-Varsakelis

Department of Applied Informatics,
University of Macedonia, Thessaloniki, 54006, Greece
dcv@uom.gr

1 Introduction

One of systems theory's most useful and fundamental ideas is that of interconnecting simple systems in order to build complex ones. This is usually accomplished through the use of two important tools. One is a set of theoretical results that help predict the behavior and performance of the composed system given the properties of its components and the manner in which they are connected. The other is the ability to regard the interconnection as ideal in the sense that it neither corrupts nor delays data or—in situations where that is not the case—to "separate" its design from that of the other components (e.g., controllers).

The development in recent years of embedded and network technologies has given rise to the area of *Networked Control Systems* (NCSs), where sensors, actuators and computing elements are connected by means of a network or other shared medium. At the same time, the attempt to expand the scope of systems theory into this new domain has made the assumptions stated above increasingly difficult to justify. The goal of this chapter is to expose some of the complications that arise when a control system includes a network (taken to mean a shared communication medium in the most generic sense) and to introduce a small collection of basic results on the control of systems that operate under communication constraints.

The very technologies that enable one to construct NCSs impose limitations in communication that make the interconnection of components non-trivial from the point of view of control. Some of the issues that arise include

- Delays in transmitting information between components (e.g., from a sensor to a controller). These delays could be fixed or time varying (e.g., randomly distributed).

*This work was supported by the National Science Foundation under Grant No. EIA0088081 and by ARO ODDR&E MURI01 Grant No. DAAD19-01-1-0465 (Center for Communicating Networked Control Systems, through Boston University).

- The possibility of data failing to reach its destination (this is not only a function of the communication medium but also of the protocol being used; TCP/IP is a well-known example).
- Bottlenecks; they could occur because the shared network can only accommodate a limited number of simultaneous communications between components or has limited throughput. Bottlenecks could also occur because of computational constraints, e.g., the CPU on which the controller is implemented can only perform a limited amount of computation per unit time.

These constraints can be captured mathematically through a variety of techniques, some of which will be reviewed in the sequel. However, the existence of the constraints has the effect of complicating what are otherwise well-understood control problems (e.g., stabilization, estimation, linear quadratic regulator (LQR) tracking and others). The basic mechanism by which this occurs will become clear in the development; for now it could be summarized by stating that when a control system is subject to communication constraints, the policies that govern how the communication medium is used can have a direct effect on the design of the control policy and vice versa. In the same setting, optimal control must now be regarded jointly with optimal communication, and the goal is to simultaneously optimize the controller and communication policies governing the operation of an NCS, whenever possible. In cases where that may be difficult, one may attempt to make the problem easier to solve by assuming, for example, that the communication policy is fixed while designing a controller or vice versa.

Possible responses to these challenges include amending existing theoretical tools to apply to the new domain and developing new ones from first principles. Details such as the communication protocol and the operating system of the computer on which control is implemented can also influence the design of both control and communication policies. Here we will focus on how communication constraints affect the control and omit the implementation details, which are nevertheless discussed in other chapters of this book.

In the next sections we will give an overview of some of the available theoretical tools for addressing analysis and design problems involving NCSs, and for elucidating the interaction between control and communication decisions in systems with limited communication. We will focus mainly on stabilization and estimation. We begin by outlining a basic model for NCSs before going on to discuss (in Section 3) some feedback control problems for NCS. The basic viewpoint is that of sensor and actuator elements competing for the "attention" (in the form of time on the shared network) of a remote controller. The effects of transmission delays and dropped packets are outlined in Sections 3.3 and 3.4. Section 4 reviews basic results on feedback control and estimation of NCSs, this time emphasizing bit rates (instead of time) as the measure of "attention."

2 A Basic Model of Networked Control Systems

Fig. 1 depicts a generic NCS; the system consists of a plant, controller and network across which all sensor and actuator data must be sent. We use $u(\cdot) \in$

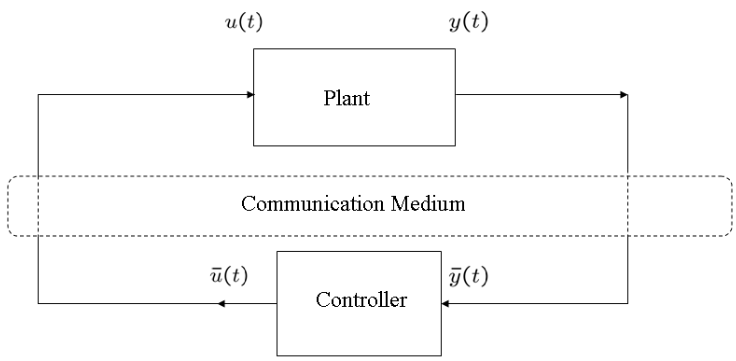

Fig. 1. A basic NCS, showing the underlying plant, its controller and the communication network that connects them

$\mathbb{R}^m$ and $y(\cdot) \in \mathbb{R}^p$ to denote the input and output of the plant, respectively. The quantities $\bar{y}(\cdot)$ and $\bar{u}(\cdot)$ denote the input and output of the controller, respectively. In general, these will differ from y and u because of the presence of the network. For example, u may be a delayed version of $\bar{u}$, if the network imposes only a delay. If the network cannot simultaneously carry signals for all m actuators, then some of the elements of u may be outdated compared to $\bar{u}$. Finally, if signals are quantized before being transmitted it may be that different elements of the vector u are quantized versions of the elements of $\bar{u}$ but with different accuracies. From a control design viewpoint, these considerations raise important questions like: "which sensor (actuator) should receive the most attention (in terms of time, frequency of communication or bit rate) by the controller?"

3 Modeling Medium Access Constraints

For now, we will ignore any transmission delays and quantization effects associated with controller/plant communication and focus instead on the bottlenecks created by the inability of the network to accommodate all sensors and actuators simultaneously. If transmission of a single sensor measurement takes t_s seconds, one may choose to "packetize" data from all sensors and transmit them every $p \cdot t_s$ seconds (what we refer to as *single-packet transmission*) or to sample one sensor at a time with frequency $1/t_s$ (*multiple packet transmission*). In the latter case, some sensors and actuators have access to the controller while others wait. This situation is illustrated in Fig. 2, where two sets of switches control access to the communication medium. Let the plant

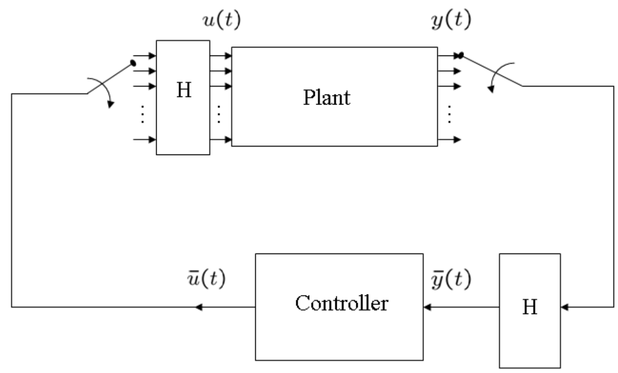

Fig. 2. Switch model

be linear time invariant (LTI), evolving in discrete time (the last assumption is not essential but it will simplify the discussion to follow):

$$x(k+1) = Ax(k) + Bu(k); \quad x \in \mathbb{R}^n, u \in \mathbb{R}^m \tag{1}$$

$$y(k) = Cx(k); \quad y \in \mathbb{R}^p. \tag{2}$$

Suppose that the communication medium connecting the sensors to the controller has n_σ $(1 \le n_\sigma < p)$ *output channels*. At any one time, only n_σ of p sensors can access these channels to send their output to the controller, while others have to wait. Likewise, actuators share n_ρ $(1 \le n_\rho < m)$ *input channels* to communicate with the controller, and at most n_ρ of them can do so simultaneously.

Of course, when a sensor (actuator) temporarily stops communicating with the controller (plant), the latter must decide how to handle the interruption. This takes place in the blocks denoted by H in Fig. 2. One option is for H to implement a zero-order hold (ZOH) so that the receiver uses the most recently transmitted value until communication is re-established. This has some appealing aspects but may increase the complexity of the control problem as we shall see. Another possibility is for the receiver to "ignore" the sensors or actuators that have gone off-line, in a way which will be made precise below.

The communication status of each sensor at time k can be encoded in the binary-valued function $\sigma_i(k), i = 1, ..., p$ with $\sigma_i(k) : \mathbb{Z} \mapsto \{0, 1\}$, where 1 means "accessing" and 0 means "not accessing". This leads to the following intuitive definition [8, 11].

Definition 1. *An m-to-n communication sequence is a map* $\sigma(k) : \mathbb{Z} \mapsto \{0, 1\}^m$*, satisfying* $\|\sigma(k)\|^2 = n$*,* $\forall k$*.*

The medium access status of the plant's sensors and actuators can then be represented by a pair of p-to-n_σ and m-to-n_ρ communication sequences, labeled σ and ρ, respectively. We will use σ (referred to as the *output communication sequence*) and ρ (the *input communication sequence*) to denote the sequences that govern the transmission of sensor and actuator data, respectively.

One is now faced with the problem of designing a pair of communication sequences and a controller that together achieve a desired control objective (e.g., stability). We will refer to the simultaneous selection of controller and communication sequence(s) for an NCS as the *joint problem.* We distinguish between two kinds of communication policies: *static* (or *fixed*), where a communication sequence is determined off-line, and *dynamic* (or *feedback based*), where communication decisions depend on the plant's outputs and on the access status of sensors and actuators.

Remark 1 (Selection of effective communication sequences). In general, the joint problem is difficult to solve when it comes to instances of typical NCS design problems, including stabilization and LQR tracking. When the joint problem is intractable, there are several alternatives:

- A typical approach is to postulate a communication sequence and then obtain a controller that satisfies the desired criteria. Such is the approach in Section 3.1 for example.
- Under some formulations, it is possible to narrow down the set of acceptable communication sequences and choose from that set. Sections 3.1 and 3.2 offer examples of this approach.
- Another alternative is to use heuristics or approximation methods in order to construct sequences that perform "sufficiently well" [18, 23].
- Finally, one could forgo the problem of choosing specific communication sequences and instead propose a policy for determining the communication on-line (as a function of time and sensor data, for example). We will discuss this further in Section 3.2.

3.1 Stability with a static communication sequence

We first consider the following problem

Problem 1. For an NCS whose plant is governed by (2) and whose controller can communicate with n_ρ and n_σ actuators and sensors respectively at any one time, find a pair of communication sequences $\sigma,\ \rho$, and a feedback controller $\bar{u}(k) = \Gamma(k)\bar{y}(k)$ so that the closed loop NCS is stable.

The solution to this problem is simplest if the controller and plant choose to "ignore" sensors and actuators which are not actively communicating, by assuming that the value of the corresponding output/input is simply zero. In that case, $\bar{y}(k)$, the output as seen by the controller is related to the actual output y by

$$\bar{y}(k) = \mathrm{diag}(\sigma(k)) \cdot y(k), \tag{3}$$

where for $v \in \mathbb{R}^n$, $\mathrm{diag}(v) \in \mathbb{R}^{n \times n}$ is the diagonal matrix formed using the elements of v. A similar relationship holds for $\bar{u}$, u and the input communication sequence ρ, so that from the point of view of the controller, the plant to be controlled is now time-varying:

$$x(k+1) = Ax(k) + B\mathrm{diag}(\rho(k))\bar{u}(k) \tag{4}$$

$$\bar{y}(k) = \mathrm{diag}(\sigma(k))Cx(k). \tag{5}$$

The stabilization problem can now be solved as follows:

- Restrict the solution to periodic communication sequences, so that the closed-loop dynamics (4) are periodic.
- Choose a periodic input (output) sequence that preserves the reachability (observability) of the plant. This is always possible if the original plant is controllable (observable) and A is invertible (as would be the case if (2) were obtained by discretizing a continuous time plant).
- Construct a periodic stabilizing feedback controller [24].

Theorem 1 ([30]). *Suppose A is invertible and the pair (A, B) of the plant (1) is reachable. For any integer $1 \leq n_\rho < m$, there exist integers $l, N > 0$ and an N-periodic p-to-n_ρ communication sequence ρ such that the extended plant (4) is l-step reachable, i.e., reachable on $[i, i+l]$ for any i.*

A communication sequence that preserves reachability can be easily constructed by examining the columns of

$$R = [A^{N-1}B \cdot \mathrm{diag}(\rho(0)),\ A^{N-2}B \cdot \mathrm{diag}(\rho(1)),\ \cdots\ B \cdot \mathrm{diag}(\rho(N-1))]. \tag{6}$$

An algorithm is given in [30]; similar statements hold for observability. If state feedback is available ($C = I$, so that we can write $\bar{x} = \bar{y}$ and $y = x$) then we have the following.

Theorem 2 ([30]). *Suppose that the extended plant (4) is l-step reachable and that A is invertible. Given constants $\alpha > 1$, $\eta > 1$ the feedback controller $u(k) = \Gamma(k)\bar{x}(k)$, with*

$$\Gamma(k) = -\bar{B}^T(k)(A^{-1})^T \mathcal{W}_{\eta\alpha}^{-1}(k, k+l), \tag{7}$$

is such that the closed loop NCS is uniformly exponentially stable [24] with rate α, where $\bar{B}(k) = B \cdot \mathrm{diag}(\rho(k))$ and
$\mathcal{W}_\alpha(k_0, k_f) = \sum_{j=k_0}^{k_f-1} \alpha^{4(k_0-j)} A^{k_0-j-1} \bar{B}(j) \bar{B}^T(j) (A^{k_0-j-1})^T$.

For the case of output feedback ($C \neq I$), the controller must be preceded by a state observer designed to reconstruct the plant's state from the intermittently arriving sensor data. The observer's state $\hat{x}(k)$ is then used in lieu of $\bar{x}(k)$ in the feedback controller. The (periodic) observer gains are selected using a procedure similar to that for selecting $\Gamma(k)$ (see [30] for details).

The effects of a ZOH

The stabilization problem becomes significantly more complicated if a ZOH is used when a sensor (actuator) relinquishes the network. In that case, the feedback controller has access only to $\bar{y}(k)$ (see Fig. 1), a vector composed of the most up-to-date sensor data available at the kth step. As we have mentioned, $\bar{y}(k) \neq y(k)$ because not all elements of $y(k)$ can be communicated to the controller at time k. A similar situation holds for u and the signal that actually arrives at the plant, $\bar{u}$. The communication sequences used at the input and output stages of the plant determine which components of u and y are updated at each time step. This leads to closed-loop dynamics of the form [9]

$$x(k+1) = Ax(k) + \sum_{i=0}^{2N-2} F_{ki} x(k-i), \tag{8}$$

where, assuming a constant feedback gain Γ ($\bar{u}(k) = \Gamma \bar{y}(k)$),

$$F_{ki} \triangleq B \sum_{j=\min(i,N-1)}^{\lfloor \frac{i}{N} \rfloor (i-N-1)} D_W(k,j) \Gamma D_R(k-j, i-j) C \tag{9}$$

and

$$D_R(k,i) \triangleq \begin{cases} \operatorname{diag}(\rho(k)) & i = 0 \\ \operatorname{diag}(\rho(k-i)) \prod_{j=0}^{i-1} M_R(k,j) & i > 0 \end{cases} \tag{10}$$

$$D_W(k,i) \triangleq \begin{cases} \operatorname{diag}(\sigma(k)) & i = 0 \\ \operatorname{diag}(\sigma(k-i)) \prod_{j=0}^{i-1} M_W(k,j) & i > 0 \end{cases} \tag{11}$$

with $M_R(k,j) \triangleq I - \operatorname{diag}(\rho(k-j))$, $M_W(k,j) \triangleq I - \operatorname{diag}(\sigma(k-j))$.

If the communication is periodic in k then so are the parameters F_{ki}, and (8) can be written in first-order form as [9, 10]

$$\chi(k+1) = \mathcal{F}_k \chi(k), \tag{12}$$

where $\chi = [x^T_{(k-2N+1)} \ \cdots \ x^T_{(k)} \ x^T_{(k+1)}]^T \in \mathbb{R}^{(2N-1)n}$. Equation (12) is linear time-varying, and describes the state evolution of the computer-controlled system under output feedback and N-periodic communication. The new state vector χ now includes past state values up to two communication periods. The periodic form (12) can be rewritten as a time-invariant system of higher dimension (equal to $N(2N-1)n$) to obtain what is known as the "extensive form" [9, 10] of the original system:

$$\mathcal{X}_e(k+1) = \mathcal{A} \mathcal{X}_e(k); \qquad \mathcal{X}_e(k) \in \mathbb{R}^{(2N^2-N)n}, \tag{13}$$

where $\mathcal{A}$ is affine in the entries of the feedback gain Γ. For fixed Γ, the problem of selecting gains to guarantee stability is non-deterministic polynomial-time hard (NP-hard) [2,10], even for a fixed communication sequence. The work in [10] describes a numerical approach to the problem, using simulated annealing to choose Γ so that the eigenvalues of $\mathcal{A}$ are enclosed in a circle with the smallest possible radius.

If we allow for time-varying feedback gains and assume state feedback, then stabilizing gains can be designed for the periodic form of the NCS (12) using results from linear periodic systems [15, 24]. On the other hand, the output feedback case, as well as the problem of simultaneously designing the communication sequences and controller, is not easy to approach. Some interesting special cases include [14]; that work discusses the stabilization of NCSs with time-varying decentralized controllers and gives criteria for stabilizability and rules for sequence design, although the latter problem becomes complex as the length of the sequence and number of possible interconnections grows.

3.2 Feedback-based communication

Feedback-based communication offers a sometimes attractive alternative to the problem of selecting communication sequences for NCSs. The idea is to let the position of the switches in Fig. 2 be determined by the state (or output) of the NCS by defining a suitable mapping

$$\sigma(x,t) \in \{0,1\}^m, \quad \sigma : \mathbb{R}^n \times \mathbb{R}_+ \to \{0,1\}^m \tag{14}$$

for the output sequence σ, and another for the input sequence ρ. In contrast to Section 3.1 where the controller and plant poll each other's outputs, here communication is interrupt driven. Such a choice has an obvious potential advantage: if the policy σ is chosen carefully, the controller may be able to respond immediately to changes in a sensor's output if they are deemed important. Under static communication, that sensor would have to wait for its turn, which could come much later, depending on the particular communication sequence chosen. On the other hand, static communication can guarantee that every sensor and actuator will be polled. This offers a robustness advantage, because it makes a "dead" sensor easy to detect, for example. Next, we give two examples of dynamic communication policies.

A block-diagonal NCS

Consider a collection of continuous-time linear time-invariant (LTI) systems

$$\begin{aligned} &\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t); \quad i = 1, \ldots, N \\ &x_i(t) \in \mathbb{R}^n, u_i(t) \in \mathbb{R}^m \end{aligned} \tag{15}$$

whose open-loop dynamics are unstable ($\mathrm{Re}\{\lambda(A_i)\} > 0, \quad i = 1, \ldots, N$). Each system communicates with a remotely located controller over an idealized

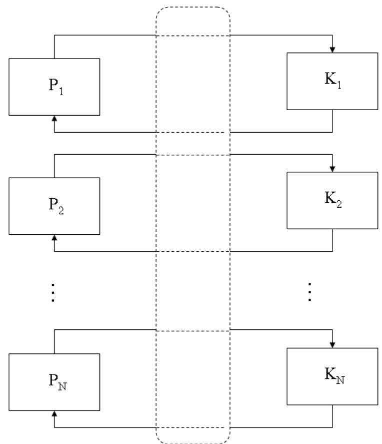

Fig. 3. A collection of NCSs $G_i(s) = I(sI - A_i)^{-1}B_i$ driven by static feedback controllers K_i via a network. Only k of N switches s_i can be closed at any one time [12].

shared network, according to the static state feedback law[1] $u_i(t) = K_i x_i(t)$ (see Fig. 3).

The gains K_i are *designed a priori* so that $\mathrm{Re}\{\lambda(A_i + B_i K_i)\} < 0$, $i = 1, \ldots, N$; i.e., each system is stabilized in the absence of communication constraints. Controller-plant communication is limited in the sense that a maximum of $C < N$ plants may close their feedback loops at any one time. We note that although there are no coupling terms in (15), the dynamics of the systems *are* coupled because of the presence of the communication constraint; if a system monopolizes the network others may not be stabilizable.

Problem 2. Find a *feedback-based* policy for establishing and terminating communication between each system and its controller in a way that stabilizes all systems in the collection.

To proceed, write the dynamics of each system in the collection as

$$\dot{x}_i(t) = A_{s_i(t)} x_i(t); \quad i = 1, \ldots, N, \tag{16}$$

where $A_{s_i(t)} \in \{A_i^o, A_i^c\}$, $A_i^o \stackrel{\triangle}{=} A_i$ and $A_i^c \stackrel{\triangle}{=} A_i + B_i K_i$ denote the open- and closed-loop dynamics of the collection, and $s_i(t) \in \{0, 1\}$ are piecewise constant functions that indicate when the ith loop is closed ($s_i(t) = 1$).

The following result [11] gives a sufficient condition for the existence of communication sequences that simultaneously stabilize the collective.

Theorem 3 ([11]). *Consider the collection of networked LTI systems in (16) and assume that at most C out of N systems are allowed to close their feedback*

[1]The discussion applies in the case of output feedback as well.

loops at any one time. For $i = 1, ..., N$, let $V_i^c(x_i) = x_i^T P_i x_i$, $P_i = P_i^T > 0$ be Lyapunov functions for the closed-loop systems, satisfying $(A_i^c)^T P_i + P_i A_i^c < \lambda_i P_i < 0$ when communication is available (feedback loop closed) and $(A_i^o)^T P_i + P_i A_i^o < \mu_i P_i$ otherwise (for some $\lambda_i < 0$, $\mu_i > 0$). Then, for any $T > 0$, there exists a T-periodic communication sequence that stabilizes all N systems if

$$\sum_{i=1}^{N} \frac{\mu_i}{\mu_i - \lambda_i} < C. \tag{17}$$

See also [4] for a condition based on rate-monotonic scheduling.

The parameters μ_i, λ_i in (17) are not unique but can be optimized to yield a less conservative bound. The optimization involves solving a set of bilinear matrix inequalities (see [11] for details).

For simplicity, assume from now on that $C = 1$, i.e., only one system can close its feedback loop at any one time, and consider the following communication policy [12].

Definition 2 (CLS-ϵ). *Let $i_*(t)$ denote the index of the system whose feedback loop is closed at time t.*

- *1. Let t_0 denote the current time. Set $i_*(t_0) = \arg\max(\|x_i(t_0)\|)$.*
- *2. When $\|x_{i_*}(t)\| = \epsilon > 0$, repeat from step 1.*

This policy, which seeks to "Contain the Largest State" (CLS-ϵ), can be viewed as the analog of the "Clear the Largest Buffer" policy, originally introduced in the study of distributed manufacturing systems [22]. CLS-ϵ chooses the system with the largest state and steers it near the origin, before selecting again. We note that such a policy cannot stabilize the collection; at best, it may guarantee that the systems are *ϵ-captured*, i.e., the $\|x_i\|$ will be arbitrarily close to ϵ as $t \to \infty$.

If the systems under consideration have *scalar dynamics*, we can obtain a necessary and sufficient condition for ϵ-capture.

Theorem 4 ([12]). *Consider the collection of networked LTI systems described in (16) with $A_{s_i(t)} \in \{A_i^o, A_i^c\}$, $A_i^o > 0$, $A_i^c < 0$, where at most $C = 1$ out of N systems are allowed to close their feedback loops at any one time and where the binary-valued $s_i(t)$ are determined by CLS-ϵ for any fixed $\epsilon > 0$. Then, all $|x_i(t)|$ will approach ϵ if and only if [2] $\phi \stackrel{\triangle}{=} \sum_{i=0}^{N} \frac{A_i^o}{A_i^o - A_i^c} < 1$. Furthermore, if $\phi > 1$ then there exists no stabilizing communication sequence.*

CLS-ϵ can also be used to drive the systems to the origin by gradually decreasing the value of ϵ. Under CLS-ϵ, the switching rate is not bounded. It is possible however to slightly modify the switching policy so that the switching rate is bounded above by $\frac{1}{\tau}$ [12]. The "minimum waiting" time $\tau > 0$ will

[2] For $k > 1$, replace the right-hand side of the inequality with k.

be the analog of the "setup time" in [22]. In that case (which will not be discussed here due to space constraints) the states will remain bounded.

If the systems of (16) are multivariable, then it is possible that CLS may fail to stabilize the collection but that there are other communication sequences that result in stability. In fact, there are well-known examples of switched systems for which there exists a stabilizing switching sequence, even when A^c and A^o are both unstable [3]. This suggests that, unlike the scalar case, there may be no necessary condition for stability based solely on the eigenvalues of the systems. However, sufficient conditions for stability or ϵ-capture can be obtained if we are willing to make switching decisions based not on the norms $\|x_i\|$ but rather on the exponential curves that bound the Lyapunov functions from Theorem 3, or on the Lyapunov functions V_i themselves [12]. In the latter case, one typically obtains a less conservative switching policy.

Theorem 5. *The collection of systems in (16) will be ϵ-captured under the interrupt-based communication policy obtained by replacing $\|x_i(t)\|$ by $V_i(x_i(t))$ in the CLS-ϵ algorithm, if $\phi = \sum_{i=1}^{N} \frac{\mu_i}{\mu_i - \lambda_i} < 1$, where λ_i and μ_i are obtained by solving Problem 1.*

In the latter case, the $V_i(x_i(t))$ are not pure exponentials and in fact may not be monotonic between switching times; therefore the state whose Lyapunov function is largest at a given switching time t may not always correspond to the system whose envelope function is largest at t.

We note that in Theorems 4 and 5, the CLS-ϵ policy must continuously attend to the states x_i in order to decide when a switching must take place. It is possible to modify matters so that making network access decisions requires only intermittent feedback (sampling of the $\|x_i\|$) or no feedback at all. In those cases, switching decisions are made based on a set of piecewise exponential curves that bound the Lyapunov functions V_i [12].

Fully coupled NCS

Consider now an NCS where the plant is the following controllable LTI system:

$$\dot{x} = Ax + Bu; \quad x(0) = x_0 \tag{18}$$

$$y = Cx; \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m,\ y \in \mathbb{R}^p. \tag{19}$$

For now, assume a state feedback controller ($y(t) = x(t)$) and that only one of the n sensors can communicate with the controller while others must wait. At the input side of the plant, m actuators share a single input channel to communicate with the controller (what is discussed below can be easily extended to the multiple-access case).

This time, define a communication sequence as the continuous-time analog of Definition 1, namely $\sigma(t) : \mathbb{R} \mapsto \{0,1\}^M$, with $\|\sigma(t)\|^2 = N$, $\forall t$, so that a given output, say $x_i(t)$, is available to the controller only when $\sigma_i(t) = 1$;

otherwise, we assume (as in Section 3.1) that a *zero* value will be used by the controller for that sensor to generate the control signals, while the actual output $x_i(t)$ will be ignored due to its being unavailable [31].

The state x and its value $\bar{x}$ as seen from the controller (Fig. 1) are now related similarly to those in Section 3.1 so that under static feedback, $\bar{u}(t) = K \cdot \bar{x}(t)$, the closed-loop dynamics of the NCS are

$$\dot{x}(t) = (A + B \cdot \mathrm{diag}(\rho(t)) \cdot K \cdot \mathrm{diag}(\sigma(t)))\, x(t). \tag{20}$$

The medium access constraints are captured by cascading the plant with a pair of time-varying operators which are obtained directly from the input and output communication sequences. The stabilization problem can now be solved in a straightforward way, in contrast to the case when a ZOH was used between the communication medium and the plant.

By definition, $\rho(t)$ can only have m possible values and $\sigma(t)$ can only have n possible values. Hence the closed loop NCS (20) is essentially a switched system with $m \cdot n$ possible dynamics[3]:

$$\dot{x} = \mathcal{A}_{s(t)} x \tag{21}$$

where $s(t)$ defines a switching rule, $s(t) : \mathbb{R} \mapsto \{1, \ldots, m\} \times \{1, \ldots, n\}$ and $\mathcal{A}_{s(t)}$ takes values on the set $\{\mathcal{A}_{ij} : i = 1, \ldots, m;\ j = 1, \ldots, n\}$, where $\mathcal{A}_{ij}$ denotes the closed-loop dynamics when actuator i and sensor j are accessing the communication medium.

A stabilizing gain and communication policy can now be determined by the following algorithm [31], using a result [6] from switched systems:

- Choose $\Gamma \in \mathbb{R}^{p \times m}$ so that $A + B\Gamma$ is stable.
- Choose scalars $\alpha_{ij} > 0$, for $1 \le i \le m$, $1 \le j \le p$ so that $\sum \alpha_{ij} = 1$.
- Write $\Gamma = \sum_{i,j} \alpha_{ij} K_{ij}$ where K_{ij} are $p \times m$ basis matrices, whose (i,j) entry is the real variable k_{ij} and all other entries are zero.
- Notice that $A + B\Gamma = A + B \sum K_{ij} = \sum \mathcal{A}_{ij}$.
- The communication policy selects at any time t the sensor and actuator corresponding to the indices

  $$i^*(t), j^*(t) = \arg\min_{i,j}\ x^T(t)(\mathcal{A}_{ij}^T P + P\mathcal{A}_{ij})x(t),$$

  where P is such that $(A + B\Gamma)^T P + P(A + B\Gamma) = -Q$, for some $Q = Q^T > 0$.
- The corresponding stabilizing feedback gain K is obtained by solving $\Gamma = \sum \alpha_{ij} K_{ij}$ for the K_{ij} and setting $K = \sum_{ij} K_{ij}$.

[3]When the communication medium has n_ρ $(1 < n_\rho < m)$ input channels and n_σ $(1 < n_\sigma < m)$ output channels, then $\rho(t)$ and $\sigma(t)$ will have $\binom{m}{n_\rho}$ and $\binom{n}{n_\sigma}$ possible values, respectively. The closed-loop system will then switch between $\binom{m}{n_\rho} \cdot \binom{n}{n_\sigma}$ possible dynamics.

This policy and gain will stabilize the NCS (20). It is easy to see why: given that $A + B\Gamma$ is stable by choice of Γ, there exist positive definite matrices P, Q such that $(A + B\Gamma)^T P + P(A + B\Gamma) = -Q < 0$, and for all $x(t) \neq 0$,

$$\sum_{i,j} \alpha_{ij} x^T(t)(\mathcal{A}_{ij}^T P + P\mathcal{A}_{ij})x(t) = -x^T(t)Qx(t) < 0.$$

Because $\alpha_{ij} > 0$ it follows that for all $x(t) \neq 0$ there always exist indices $i(x) \in \{1, \ldots, m\}$ and $j(x) \in \{1, \ldots, n\}$ such that $x^T(t)(\mathcal{A}_{i(x)j(x)}^T P + P\mathcal{A}_{i(x)j(x)})x(t) < 0$, which immediately gives us a choice of communication policy that keeps the Lyapunov function $V(t) = x^T(t)Px(t)$ always decreasing.

We note that for the same choice of $\mathcal{A}$, different choices of α_{ij}'s result in different values of the feedback gain K. A larger α_{ij} leads to a smaller k_{ij}. This fact gives us additional freedom in the design of K. By properly choosing α_{ij}'s we can make the controller K meet certain optimization or design criteria, or force the communication policy to pay more "attention" to certain sensors and actuators. A general communication policy might take the form [31]

Definition 3 (Weighted Fastest Decay (WFD)). *For all t, let $s(t) = (i(t), j(t))$ be determined by*

$$s(t) = \arg\min_{i,j} \alpha_{ij} \mathbf{x}^T(t)[\mathcal{A}_{ij}^T P + P\mathcal{A}_{ij}]\mathbf{x}(t), \tag{22}$$

where the coefficients α_{ij} act as weights associated with the dynamics $\mathcal{A}_{ij}$.

This class of policies is stabilizing, provided that K has been designed according to the algorithm given above. Modifications can also be made to ensure that the switching rate is bounded [31].

Definition 4 ([26]). *The system (21) is said to be quadratically stable if there exists a positive definite quadratic function $V(x) = \mathbf{x}^T P\mathbf{x}$, a positive number ϵ and a switching rule $s(t)$ such that $\frac{d}{dt}V(\mathbf{x}) < -\epsilon\mathbf{x}^T\mathbf{x}$ for all trajectories $\mathbf{x}$ of the system (21).*

Theorem 6 ([31]). *If $\mathcal{A}$ is stable, system (21) is quadratically stable under the switching rule WFD.*

The output feedback case can be handled by inserting a state observer between the communication medium and the feedback controller (see [30] for details in the discrete-time case).

3.3 The effects of transmission delays

We now discuss some of the effects of transmission delays on the stability of NCSs. We begin with the structure illustrated in Fig. 4, where sensor data are delayed by τ_s while actuator data are delayed by τ_a units of time. In practice,

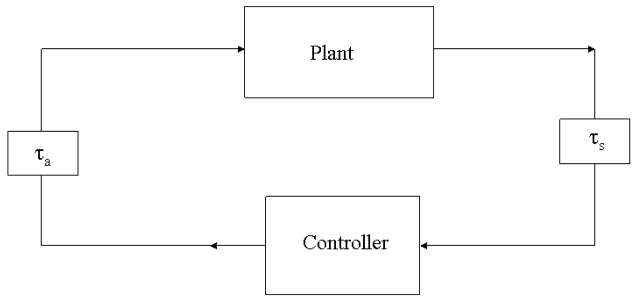

Fig. 4. NCS subject to transmission delays

these delays are induced not only by the finite speed at which data travel inside the communication medium, but also by the details of the communication protocol (e.g., single- or multiple-packet transmissions). For the remainder of this section we will assume single-packet transmission, i.e., all elements of $y(t)$ are transmitted together.

Let the continuous-time LTI plant evolve according to (18). The signal available to the controller is $x(kh-\tau_s)$. The (discrete-time) controller is given by

$$\bar{u}(t) = -Kx(kh - \tau_s); \quad k = 0, 1, 2, ..., \quad t \in [kh + \tau_s, (k+1)h + \tau_s)$$

and arrives at the plant τ_a seconds later. Thus, from the point of view of the plant, the total delay around a *constant-gain* feedback loop is the sum $\tau = \tau_s + \tau_a$, so that the plant is driven with a piecewise constant input which is obtained from the delayed sensor data:

$$u(t) = -Kx(kh - \tau); \quad t \in [kh + \tau, (k+1)h + \tau),\ k = 0, 1, 2, ... \tag{23}$$

The question for this type of NCS is the following.

Question 1. Given K such that $A+BK$ is stable, what is the maximum delay τ_{max} which can be tolerated before the NCS becomes unstable?

Constant delay

If τ is constant, then

Theorem 7 ([32]). *The NCS with constant delay (23) is stable if the eigenvalues of*

$$H = \left[\begin{array}{c|c} e^{Ah} & \int_0^h e^{A(h-s)} ds BK \\ \hline e^{A(h-\tau)} & -e^{A\tau}\left(\int_0^h e^{A(h-s)} ds - \int_0^\tau e^{A(h-s)} ds\right) BK \end{array}\right]$$

lie inside the unit disc.

Time-varying delays

If the loop delay τ_k varies from one transmission to the next, then the rate at which new inputs arrive at the plant is not fixed. The system can be analyzed by augmenting the state to include $x(kh)$ as well as all inputs that the plant receives in the interval $[kh, kh+\tau_k)$. The augmented state is $z(kh) = [x^T(kh), u^T((k-l)h), ..., u^T((k-1)h)]^T$, where the integer l is such that $(l-1)h < \tau_k < lh$ for all k. The stability of the (linear) dynamics of z is equivalent to the stability of the original NCS.

For example, if $l = 1$, then one must check the stability of the time-varying linear system [32] $z((k+1)h) = \Phi(k)z(kh)$ with $z(kh) = [x^T(kh), u^T((k-1)h)]^T$ and

$$\Phi(k) = \left[\begin{array}{c|c} e^{Ah} - \int_0^{h-\tau_k} e^{As} B ds K & \int_{h-\tau_k}^{h} e^{As} B ds \\ \hline -K & 0 \end{array}\right]. \tag{24}$$

As the discussion above suggests, the problem becomes significantly more complicated if τ_k varies in such a way that the integer l is not fixed.

We remark that it is sometimes possible to compensate for some of the effects of transmission delays. If the plant has the ability to "time-stamp" its sensor data before transmission, then the controller can use the time information to compensate for the delay τ_s, assuming that plant and controller have synchronized their clocks. The controller can then estimate the current state of the plant by propagating (according to the dynamics (18)) the state data it receives from the sensors, for an amount of time equal to that of the sensor-to-controller delay. If the controller-to-actuator delay τ_a is constant, then the controller can also compensate for the amount of time its data will take to reach the plant. For further details, including the construction of a delay-compensating state observer, see [32] and references therein. See also [16, 21] for discussions of NCS stability under random delays.

3.4 The effects of unreliable communication links

We now turn our attention to the possibility that the communication between plant and controller is unreliable, in the sense that sensor/actuator data may fail to reach their destination. This situation, where data packets are "dropped," can arise because of network congestion, unreliable hardware, or because of the transmission protocol used (the transmission control protocol (TCP) and user datagram protocol (UDP) are two well-known examples).

Consider an NCS, where the connections from controller to plant and plant to controller (referred to as *uplink* and *downlink*, respectively) are unreliable, in the sense that transmitted data may occasionally fail to reach its destination. To make things precise, let the plant be described by the discrete-time LTI system

$$x(k+1) = Ax(k) + \alpha(k)Bu(k), \qquad (25)$$
$$y(k) = \beta(k)x(k),$$

where $\alpha(k), \beta(k) \in \{0, 1\}$ indicate whether at time k the control (measurement) signal reaches the plant (controller) or not. The assumption here is that the vectors u and y are sent in single-packet transmissions, and that the sequences $\{\alpha(k)\}$, $\{\beta(k)\}$ are i.i.d. Bernoulli, with $Pr[\alpha(k) = 0] = \alpha$ and $Pr[\beta(k) = 0] = \beta$ being the link failure probabilities.

One can then pose the following problem.

Problem 3. For the system of (25) and given the link failure rates α, β, find a control policy that minimizes

$$J = \mathcal{E}\Big\{\sum_{k=0}^{\infty} x(k)^T Q x(k) + \alpha(k) u(k)^T R u(k)\Big\},$$

where $\mathcal{E}\{\cdot\}$ denotes expected value.

This LQR problem (and its finite-horizon version) are discussed in [13]. A related problem is the following.

Problem 4. For the system (25), what are the maximum link failure rates α, β for which a stabilizing controller exists?

In [13] it is shown that the controller that minimizes J is given by a feedback law similar to that for the standard LQR problem:

$$u^*(k) = G(k)\hat{x}(k); \quad \hat{x}(k) \stackrel{\triangle}{=} \mathcal{E}\{x(k)\}, \qquad (26)$$

where $\hat{x}(k)$ is an estimate of the state x obtained by

$$\hat{x}(k) = \begin{cases} A\hat{x}(k) + \alpha(k-1)Bu(k-1), & \beta(k) = 0 \\ x(k), & \beta(k) = 1 \end{cases}, \qquad (27)$$

$G(k) = -(R + B^T K(k+1)B)^{-1} B^T K(k+1)A$, and $K(k)$ is determined by the following recursive matrix equations:

$$P(k) = (1-\alpha)A^T K(k+1)B(R + B^T K(k+1)B)^{-1} B^T K(k+1)A \qquad (28)$$
$$K(k) = A^T K(k+1)A - P(k) + Q. \qquad (29)$$

The solution to the last set of equations as well as the answer to Problem 4 depend strongly on whether the communication protocol includes "acknowledgment" (ACK) packets that allow the sender to know whether its transmission was received or not. If all receptions are acknowledged and ACK packets are always received by the transmitter, then the separation principle [7] holds, and the controller and estimator can be designed separately. If the protocol does not support acknowledgment (e.g., UDP), then the controller does not know the "state of the channel," i.e., whether $\alpha(k)$ was 0 or 1 and thus has no way of knowing whether the sensor output it receives at time $k+1$ includes the effect of $u(k)$.

In the case of the infinite-horizon LQR problem with acknowledgments, the stabilizing controller and maximum link failure rate are given by the following theorem.

Theorem 8 ([13]). *Let B be square and full rank, and let $(A, Q^{1/2})$ be observable. Suppose*

$$\max_i |\lambda_i(A)| < \min \left\{ \frac{1}{\sqrt{\alpha}}, \frac{1}{\beta} \right\}.$$

Then (i) $K(k)$ converges to the positive definite solution of

$$K = A^T K A + Q - (1-\alpha) A^T K B (R + B^T K B)^{-1} B^T K A$$

and (ii) the closed-loop system is stable.

Without ACK packets the estimator is again given by (27). However, the separation principle does not hold and the maximum tolerable link failure rate is slightly reduced, as given the following theorem.

Theorem 9 ([13]). *Let B be square and full rank, and let $(A, Q^{1/2})$ be observable. Suppose*

$$\max_i |\lambda_i(A)| < \min \left\{ \sqrt{\frac{1+\alpha\beta}{\alpha+\alpha\beta}}, \frac{1}{\beta} \right\}.$$

Then (i) there exist $K > 0$, $P > 0$, such that for $P(0) = 0$ and all $K(0) > 0$, the Riccati equations (29) converge to the positive definite solutions of

$$P = (1-\alpha) A^T K B (R + B^T K B)^{-1} B^T K A \tag{30}$$

$$K = A^T K A - P + Q \tag{31}$$

and (ii) the closed-loop system is stable.

The case where the system is subject to actuator noise and transmissions are multiple-packet is discussed in [1]. In that work acknowledgment packets can also be "dropped," so that the separation principle does not hold. Thus one has a system whose dynamics switch randomly between eight possible dynamics, depending on whether the transmitted data (from either controller or plant) arrived at its destination, and whether an acknowledgment failed to arrive back to the sender. One can design a suboptimal controller/estimator pair (by insisting on separation) and arrive at a set of necessary and sufficient linear matrix inequality(LMI)-based conditions that relate the stability of the closed-loop NCS (under the proposed controller/estimator structure) to the link failure rates.

For the special case where only the downlink is subject to unreliable communication under single-packet transmission, and the controller $u(k) = K\hat{x}(k)$ uses ZOH,

$$\hat{x}(k) = \begin{cases} x(k) & \text{if } \beta(k) = 1, \\ \hat{x}(k-1) & \text{if } \beta(k) = 0 \end{cases} \tag{32}$$

a bound for the maximum allowable link failure rate is given in [32].

Theorem 10 ([32]). *Consider the system of (25) where $\alpha(k) = 1$ for all k, i.e., only the downlink is subject to failures, with rate $(1-r)$, $0 < r \leq 1$. If the controller K is such that $A + BK$ is stable, then the closed-loop system is exponentially stable for all*

$$\frac{1}{1 - \log\left(\lambda_{\max}^2(A+BK)\right) / \log\left(\lambda_{\max}^2(A)\right)} < r \leq 1.$$

3.5 Communication sequences: Beyond stability-related problems

In addition to the stabilization problems discussed in the previous sections, communication sequences have been used to capture communication constraints in problems related to tracking, optimal control and robust control. For example, the work in [23] discusses LQR problems with communication constraints, and [8] addresses least-squares output tracking for NCS. As we have mentioned, the problem of finding optimal communication sequences is typically a difficult one. Interesting heuristics that attempt to find near-optimal communication sequences are explored in [18] (H_2 and H_∞ control for NCS) and [17] (optimizing communication in LQR problems).

4 A Complementary Viewpoint: Control with Limited Bit Rate

Up to now, we have concentrated on time-division based models for capturing communication constraints and have treated the communication channel as being able to transmit signals with infinite precision. This assumption works well when channel throughput is high enough so that performance is not significantly affected by quantization errors. Aside from the fact that realistic channels can only accommodate a finite number of bits per second, re-examining limited communication control where the limited resource is *bits* as opposed to time can yield valuable insights as to how one could design controllers for NCSs, and what data rates are required for adequate performance. The rest of this section reviews some of the fundamentals when the feedback loop is closed via digital channels which are subject to data rate limitations.

4.1 State estimation and stabilization with limited bit rate

Fig. 5 illustrates an NCS whose feedback loop is closed over a digital channel. The channel can support a maximum rate of R bits per second (it takes $\delta = 1/R$ seconds to transmit a single bit). Assume that the plant is continuous-

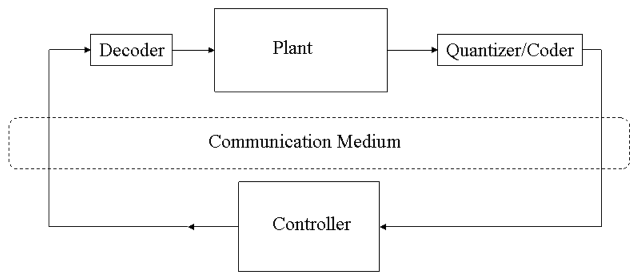

Fig. 5. A feedback loop which is closed over a communication channel with limited bit rate. The controller acts on coded versions of the sensor data and produces inputs that must be decoded before being applied to the plant

time LTI (18). Digital communication means that sensor and actuator data must be sampled with finite precision (quantized) and coded in a string of bits (or more generally, symbols from a D-ary alphabet). The coded observation is sent to the controller, which decodes it and computes a new plant input, which is again coded before being transmitted to the plant. There are various choices with respect to the coding scheme used, including fixed- and variable-length codes. A particularly useful subset of the latter category are *prefix codes* [5], which allow the receiver to immediately recognize the end of a code word.

Question 2. What is a necessary (sufficient) bit rate to guarantee the existence of a coding scheme and estimator for the state x?

Question 3. What is a necessary (sufficient) bit rate to guarantee the existence of a coder, decoder and controller that stabilize the NCS?

Estimation

We begin by addressing Question 2. A “good” estimate of the state x seems necessary in order for any controller to be effective. Under the assumptions given in the previous section, the time required to transmit a measurement of the state grows linearly with the number of bits sent. Therefore, precision becomes counterproductive after a certain point: if one chooses to describe a sensor reading too precisely, the controller will receive the digital description later and the data will be outdated.

To illustrate this trade-off, assume that fixed-length coding is used to send data from the plant to the estimator, and that the plant is discrete time, with no process or measurement noise:

$$x(k+1) = f_k(x(k)), \quad x(k) \in \mathbb{R}^n, \quad x(0) \in S_0 \subset \mathbb{R}^n, \quad k = 0, 1, 2, \dots \, . \tag{33}$$

The initial state $x(0)$ is assumed to be drawn from a probability density with compact support (e.g., a compact set S_0) and the functions f_k are Lipschitz:

$$\|f_k(x) - f_k(y)\| \leq M_k \|x - y\| \quad \forall x, y \in \mathbb{R}^n.$$

At each time step k, an estimator receives R bits of information on the state $x(k)$ (propagation delays are ignored) and must produce an estimate $\hat{x}(k+1)$, based on all *past transmissions*. One way to do this is to begin by partitioning S_0 into 2^R disjoint regions and transmit the index of the region that contains $x(0)$. The estimator can then set the auxiliary variable $z(0)$ to be any point in that region (e.g., its centroid) and then propagate $z(0)$ ahead according to the dynamics (33) to generate the current estimate, i.e., at time k, set $\hat{x}(k+1) = f_k(z(k))$. The key point is that if the error $e(k) = x(k) - \hat{x}(k)$ does not grow too rapidly with k (equivalently, if the Lipschitz constants M_k are not too large), the estimator can know with certainty that the state $x(k+1)$ is contained in a subset $S(k+1)$ whose size (in terms of diameter or Lesbegue measure) is smaller than that of $S(k)$. The new $S(k+1)$ is then repartitioned in 2^R subsets and the procedure is repeated, further "narrowing down" the expected value of the error. The following is a restatement of the main result in [19].

Theorem 11 ([19]). *Let the distribution of $x(0)$ have compact support. Then, there exist a coder and estimator (based on the successive partitioning idea described above) such that*

$$\mathcal{E}\|x(k+1) - \hat{x}(k+1)\|^2 \leq \phi^2 \left(\frac{\prod_0^k M_i}{2^{Rk/n}} \right),$$

where ϕ is a constant that depends on the state dimension n, the bit rate R and the size of the compact set that contains the initial state.

The last inequality gives a sufficient condition for the estimation error to converge to zero. If the plant is linear (35), then the same condition specializes to

$$R > n \log_2(\max_i \lambda_i(A)).$$

The work in [19] includes extensions of the last theorem in the case where the distribution of $x(0)$ is not compact. For additional insights into the problem of state estimation under process and measurement noise, including an explicit consideration of transmission delays, see [27].

Stabilization: Explicit consideration of transmission delays

We now turn to the problem of stabilizing the NCS (Fig. 5) under the bit rate constraints discussed in the previous section. We will assume for the moment that controls are applied for arbitrarily short time intervals. The finite delay between when y (or x) is measured and u is applied means that the system is essentially uncontrolled for some time initially and cannot be asymptotically stabilized unless the initial state and start time are known precisely. However, one can ask for a slightly weaker type of stability.

Definition 5. *An NCS is containable if for any ball N around the origin there exists an open neighborhood of the origin M and coding and feedback control laws such that any trajectory started in M remains in N for all time.*

Question 3 can be answered (in the context of containability) by assuming first that $x(0)$ lies in some Lesbegue-measurable set S_0. Suppose the plant is continuous-time LTI (18). If the plant is unstable, then any uncertainty in one's knowledge of $x(0)$ will be "amplified" as time goes on. On the other hand, the coder must balance speed with precision (i.e., giving an "answer" quickly versus transmitting more bits) when providing information on x.

We begin by asking how many bits it takes to "narrow down" the set that contains x over consecutive transmissions, given the bit rate of $1/\delta$ [28]. After applying a control u to the plant over some interval $[0, k\delta]$, and in the absence of any observations, the state must lie somewhere in the set

$$S_1(S_0, u, T) = e^{At} S_0 + g(u),$$

where $g(u)$ is some vector in $\mathbb{R}^n$ and $e^{At} S \stackrel{\triangle}{=} \{e^{At} x : x \in S\}$.

If we let $\mu(S)$ denote the n-dimensional "volume" (more precisely the Lesbegue measure) of the set S, we have

$$\mu(K_1(S_0, u, T)) = \det(e^{At}) \mu(S_0) = e^{tr(A)t} \mu(S_0).$$

As before, consider decomposing the set S_0 into $S_0 = \cup_{i=0}^{N} K_i$ where the subsets K_i are such that all elements of K_i correspond to a unique code word c_i. The coder checks to see which of the K_i contains the state and transmits the corresponding code word (using c_i bits) to the controller, which in turn sends d_i bits to the plant. The c_i, d_i are assumed to be prefix codes. Then, the set that contains the state at the end of the first transmission satisfies

$$\mu(S_1) = e^{tr(A)(c_i + d_i)\delta} \mu(K_i).$$

If the system is to be containable, then $\mu(S_1) < \sum \mu(K_i)$ because $S_1 \subset S_0 = \cup_i K_i$. Summing over all K_i leads to

$$\sum_{0}^{N} \frac{1}{e^{\delta(c_i + d_i) trA}} \geq 1 \tag{34}$$

as a necessary condition for containability. Moreover, if the same set of code words is used for both observation and control, then the following can be shown [28].

Theorem 12 ([28]). *The NCS is containable only if* $e^{2tr(A)\delta} \leq D$.

In the case where y, x and u all have the same dimension n, one can obtain a sufficient condition as well.

Theorem 13 ([28]). *If* (A, B) *is controllable and* C *is invertible, then the NCS (with binary code words) is containable if* $\max_{\|x\|_\infty} \|e^{\delta A} x\|_\infty^{2^n+1} < 2$.

Stabilization with "instantaneous" transmissions

The situation discussed in the last section makes a very useful connection between the bit rate supported by the channel and the size of the alphabet in which data is sent. It also shows explicitly that containability will be violated if one chooses to be too precise about measurements (34) and that—as expected—increasing the size of the alphabet improves the bound because it increases the amount of information carried by each bit around the loop. A similar but slightly "tighter" condition can be obtained for the discrete-time counterpart of (18) [20]:

$$x(k+1) = Ax(k) + Bu(k); \quad x(0) \in K_0 \subset \mathbb{R}^n \tag{35}$$

$$y(k) = Cx(k), \tag{36}$$

where we assume that the plant and controller are co-located, i.e., there is no need for coding/decoding actuator data and only sensor measurements must be transmitted through a digital channel at a rate of R bits per time step.

Theorem 14 ([20]). *Assume that the system (35) is reachable and observable and that its initial state* $x(0)$ *is random, with a distribution which is absolutely continuous with respect to the Lesbegue measure on* $\mathbb{R}^n$ *and has finite* $(r+\epsilon)$*-th absolute moment* $\mathcal{E}\|x_0\|^{r+\epsilon} < \infty$ *for some* $r, \epsilon > 0$. *Then, for a given data rate* R *(bits per step* k*), a coder/controller that exponentially stabilizes the NCS with rate* ρ*, i.e.,*

$$\lim_{k \to \infty} \rho^{-kr} \mathcal{E}\|x(k)\|^r = 0,$$

exists if and only if

$$R > \sum_{|\lambda_i| \geq \rho} \log_2 \left| \frac{\lambda_i(A)}{\rho} \right|, \tag{37}$$

where $\lambda_i(A)$ *are the eigenvalues of* A.

The same condition on R is thoroughly explored in [25] in the context of observability, controllability and exponential stability of the NCS (35). See also [20, 29] for additional discussions of (37), including details of how various choices of coding/quantization schemes affect the bounds on the bit rate necessary for stability.

If $C = I$, the condition (37) is necessary and sufficient for the existence of a stabilizing encoder/decoder/controller. In the case where the system (35) is subject to bounded input noise,

$$x(k+1) = Ax(k) + Bu(k) + w(k); \quad \|w(k)\| < M \tag{38}$$

$$y(k) = x(k), \tag{39}$$

then (37) is necessary and sufficient for the existence of a coder, decoder and controller for which the estimation error $\limsup_{k\to\infty} \|x(k) - \hat{x}(k)\|$ is bounded, where $\hat{x}(k)$ is the output of the decoder. Here, the existence of the encoder is asserted over all encoders that have access to all past observations and controls; both encoder and decoder are assumed to have knowledge of the dynamics of the plant as well as one another.

5 Beyond this Introduction

This chapter explored some of the important results in the area of control with limited communication, focusing mainly on stabilization and estimation problems. Our goal was to give the reader a "flavor" of how communication constraints enter into the solution of control problems and to describe some of the available tools for designing (jointly when possible) effective controllers and communication policies. We explored various types of communication constraints, including transmission delays, unreliable communication links, and what could be termed "bottlenecks." The latter were due either to the limited number of channels available for controller-plant communication or to the limited throughput of a single channel shared by all sensors or actuators.

The area of NCSs is at the interface of several "core" fields within systems and control. Some of the results discussed here were built on previous contributions in switched and hybrid systems (see related chapters in this book), as well as results from more "mature" areas such as periodic systems and information theory. An excellent collection of NCS-related references can be found online at http://home.cwru.edu/ncs/allpubs.htm. Other areas that may offer the reader additional useful viewpoints on the interplay of control and communication, but were not mentioned in this chapter, include multirate systems, scheduling, systems with delays and quantized control.

References

1. B. Azimi-Sadjadi. Stability of networked control systems in the presence of packet losses. In *Proceedings, 42nd IEEE Conference on Decision and Control*, pages 676–81, Dec. 2003.

2. V. Blondell and J. Tsitsiklis. NP hardness of some linear control design problems. *SIAM Journal on Control and Optimization*, 35:2118–2127, 1997.
3. M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. on Automatic Control*, 43(4):475–82, April 1998.
4. Michael S. Branicky, Stephen M. Phillips, and Wei Zhang. Scheduling and feedback co-design for networked control systems. In *Proceedings of the 38th Conference on Decision and Control*, pages 1211–1217, December 2002.
5. T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, New York, 1991.
6. Eric Feron. Quadratic stabilization of switched systems via state and output feedback. Technical Report CICS-P-468, Center for intelligent control systems, MIT, Cambridge MA, 1996.
7. B. Friedland. *Control System Design: An Introduction to State-Space Methods.* McGraw-Hill, New York, 1986.
8. D. Hristu. Generalized inverses for finite-horizon tracking. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 2, pages 1397–402, 1999.
9. D. Hristu. Stabilization of LTI systems with communication constraints. In *American Control Conference*, volume 4, pages 2342–6, June 2000.
10. D. Hristu and K. Morgansen. Limited communication control. *Systems and Control Letters*, 37(4):193–205, July 1999.
11. D. Hristu-Varsakelis. Feedback control systems as users of a shared network: Communication sequences that guarantee stability. In *IEEE Conference on Decision and Control*, pages 3631–6, Dec. 2001.
12. D. Hristu-Varsakelis and P. R. Kumar. Interrupt-based feedback control over a shared communication medium. In *Proc. of the IEEE Conference on Decision and Control*, pages 3223–8, Dec. 2002.
13. O. C. Imer, S. Yuksel, and T. Basar. Optimal control of dynamical systems over unreliable communication links. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Sept. 2004.
14. H. Ishii and B. Francis. Stabilization with control networks. *Automatica*, 38:1745–51, 2002.
15. P. T. Kabamba. Monodromy eigenvalue assignment in linear periodic systems. *IEEE Trans. on Automatic Control*, 31(10):950–2, Oct. 1986.
16. B. Lincoln. Jitter compensation in digital control systems. In *Proceedings of the American Control Conference*, 2002.
17. B. Lincoln and B. Bernhardsson. Efficient pruning of search trees in LQR control of switched linear systems. In *Proceedings of the* 39*th IEEE Conference on Decision and Control*, volume 2, pages 1828–33, Dec. 2000.
18. L. Lu., X. Lihua, and M. Fu. Optimal control of networked systems with limited communication: A combined heuristic search and convex optimization approach. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 1194–9, Dec. 2003.
19. G. N. Nair and R. J. Evans. State estimation via a capacity-limited communication channel. In *Prof. of the 36th IEEE Conference on Decision and Control*, pages 866–71, Dec. 1997.
20. G. N. Nair and R. J. Evans. Exponential stabilizability of finite-dimensional linear systems with limited data rates. *Automatica*, 39:585–93, 2003.

21. J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 23(1):57–64, 1998.
22. J. Perkins and P. R. Kumar. Stable distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems. *IEEE Trans. on Automatic Control*, 34(2):139–48, Feb. 1989.
23. H. Rehbinder and M. Sanfridson. Scheduling of a limited communication channel for optimal control. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 1, pages 1011–16, Dec. 2000.
24. W. J. Rugh. *Linear Systems Theory*. Prentice-Hall, Englewood Cliffs, NJ, 1996.
25. S. Tatikonda and S. Mitter. Control under communication constraints. *IEEE Trans. Automatic Control*, 49(7):1056–68, Jul. 2004.
26. M. Wicks, P. Peleties, and R. DeCarlo. Switched controller synthesis for the quadratic stabilization of a pair of unstable linear systems. *European Journal of Control*, 4:140–7, 1998.
27. W. S. Wong and R. W. Brockett. Systems with finite communication bandwidth constraints–Part I: State estimation problems. *IEEE Transactions on Automatic Control*, 42(9):1294–1299, 1997.
28. W. S. Wong and R. W. Brockett. Systems with finite bandwidth constraints–Part II: Stabilization with limited information feedback. *IEEE Transactions on Automatic Control*, 42(5):1049–1052, 1999.
29. S. Yuksel and T. Basar. Quantization and coding for decentralized LTI systems. In *Proceedings of IEEE Conference on Decision and Control*, pages 2847–52, Dec. 2003.
30. L. Zhang and D. Hristu-Varsakelis. Stabilization of networked control systems: Communication and controller co-design. In *subm. to the IFAC World Congress*, 2005.
31. L. Zhang and D. Hristu-Varsakelis. Stabilization of networked control systems under feedback-based communication. In *subm. to the American Control Conference*, 2005.
32. W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, Feb. 2001.

Note

For additional material and references see "Supplementary Information and Updates to the First Printing" at `www.birkhauser.com/978-0-8176-3239-7`.