Chapter 2
Role of Delays

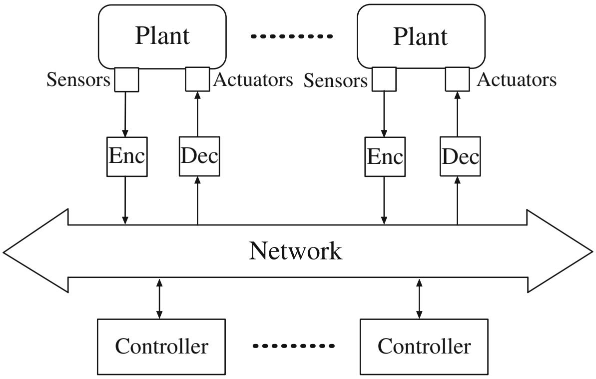
2.1 Introduction

Several key issues that make NCSs distinct from other control systems are:

1. *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 microactuators and sensors.
   Inspired by Shannons 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 [5, 7, 11, 22, 37, 41]. Recently, some progress has also been made in solving the finite-capacity stabilization problem for nonlinear systems [16, 23] and for linear systems with unknown parameters [36]. In [30], 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 [19], which obtains a general extension of Bodes integral inequality.
   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 asynchronous transfer mode (ATM) 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 [27, 33]. 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 Shannons point of view these channels have infinite capacity, the closed-loop system stability and performance is still limited by the network. This

M. S. Mahmoud, *Control and Estimation Methods over Communication Networks,* DOI 10.1007/978-3-319-04153-7_2,

Fig. 2.1 NCS architecture with encoders and decoders



perspective is prevalent in most of the results that we discuss in this survey, which generally neglect the quantization effects imposed by digital networks.

2. *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 delays 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.
3. *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 reordering, 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 retransmission of old data is generally not very useful.
4. *Systems Architecture:* Fig. 2.1 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

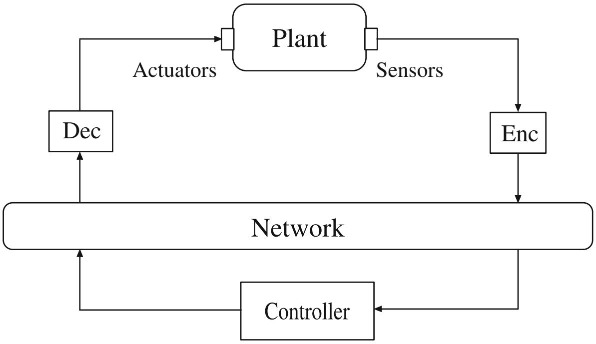


Fig. 2.2 Single loop NCS

> 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. 2.1. 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. 2.2. Although considerably simpler than the system shown in Fig. 2.1, this architecture still captures many important characteristics of NCSs such as bandwidth limitations, variable communication delays, and packet dropouts.

Recent technological advances have enabled distributed control systems (DCS) to be implemented via real-time networks. This permits feedback control loops to be closed over a shared communication channel. Thus real-time control networks (or networked control architectures) represent an emerging technology in industry and is receiving considerable attention. As available technologies (such as DeviceNet, Profitbus, FireWire, Ethernet, to name a few) become more mature and their prices decrease, they are being implemented in more real-time distributed control applications. These include, but not limited to, aircraft [32] and automobiles [28]. From a control-theoretic standpoint, the area of *networked control systems (NCS)* is rather new and starts to attract research investigations. One significant feature of NCS is that, instead of hardwiring, the control devices with point-to-point connections, sensors, actuators, and controllers are all connected to the network as nodes. Several advantages of this implementation include: reduced system wiring, plug and play devices, increased system agility, and ease of system diagnosis and maintenance.

2.2 Initial Design Methods

By and large, the insertion of the communication network in the feedback control loop makes the analysis and design of an NCS complex. Conventional control theories with several standing assumptions, including synchronized control and non-delayed sensing and actuation, must be reevaluated before they can be applied to NCSs. The following important issues stand out:

1) During data exchange among devices connected to the shared medium, time-delays occur between sensor to controller τ_{sc} and between controller to actuator τ_{ca}. Their sum $\tau_{in} = \tau_{sc} + \tau_{ca}$ represents the *network induced delay*. This delay, either constant (up to jitter) or time-varying, can degrade the performance of control systems designed without considering the delay. It can even destabilize the system.

2) Within NCS, the network can be viewed as a web of unreliable transmission paths. Some packets not only suffer *transmission delay* but, even worse, can be lost during transmission. The task to be considered is: *how such packet dropouts affect the performance of an NCS?*

3) There are cases due to bandwidth and packet size constraints where the plant output may be transmitted using multiple network packets. In such cases, the chances are that all/part/none of the packets could arrive in the required time. The purpose of this chapter is to provide a complete diagnostic profile of the role of delays in NCS. It analyzes the impact of delay sources on the stability and performance of NCS. A full attention is given to the network induced delay. It will be shown that the relationship between the sampling rate and the network induced delay can only be determined by simulation. An upper bound on the network delays using dynamic controllers will be developed in terms of an LMI-feasibility problem that can be readily solved using semi-definite programming tools.

In the literature, there have been two approaches to NCS design while taking into account the foregoing issues.

Approach I *Design the control system without any regard to the packet delay and loss and construct a communication protocol that minimizes the likelihood of these events and to gain better performance.*

Approach II *Consider the network protocol and traffic as given conditions and design control strategies based on delay-differential equations [21] that explicitly treat the foregoing issues.*

In both approaches, analysis and design strategies for both sources of delays and packet loss are examined. Previous related work is found in [1, 9, 10, 13–15, 17, 25, 29, 34, 38, 43] where discrete-time design approach was developed using integrated communication and control system (ICCS) and representing the network delays in a random fashion. Scheduling algorithms are constructed to handle the flow of messages among networks nodes. Appropriate measures were introduced to assess the system performance with particular emphasis on **maximum allowable transfer interval (MATI)** τ_m to denote an upper bound time separation between sensor messages. Explicit expression of τ_m was developed in [4, 39] in terms of sufficient conditions on the network sampling rate to guarantee that the original non-networked system remains stable when the control-loop is closed by the network.

2.2.1 Network Induced 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 delays 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.

The network induced delay (NID) occurs in NCS when sensors, actuators, and controllers exchange data across the network. Depending on the medium access control (MAC) protocol of the control network, NID can be constant, time-varying, or even random. MAC protocols are divided into:

1) *Random Access*, including carrier sense multiple access (CSMA), and

2) *Scheduling*, including token passing and time division multiple access (TDMA)

Control networks using CSMA protocols include DeviceNet and Ethernet. Various possible situations are depicted in Fig. 2.3 for the case of two nodes continually transmitting messages. Observe that the transmission begins when the network is not busy. Indeed possible collision can take place and its avoidance is protocol dependent (DeviceNet employs bit-wise arbitration to introduce prioritization whereas Ethernet incorporates collision detection portion).

In a similar way, Fig. 2.4 shows a timing diagram for TDMA. In this case, the protocols eliminate the contention of the shared network medium by allowing each node on the network to transmit according to a predetermined schedule where the token is passed around a logical or physical ring. Note that in scheduling networks, periodic transmission of messages can be arranged.

2.2.2 NCS Model

Let the plant be given by:

$$\begin{aligned} \dot{x}_p(t) &= A_p x_p(t) + B_p u_p(t) \\ y_p(t) &= C_p x(t) \end{aligned} \tag{2.1}$$

and the discrete controller is

$$u_p(kh) = -\ K\ x_p(kh) \quad , k = 0, 1, 2 \ldots \tag{2.2}$$

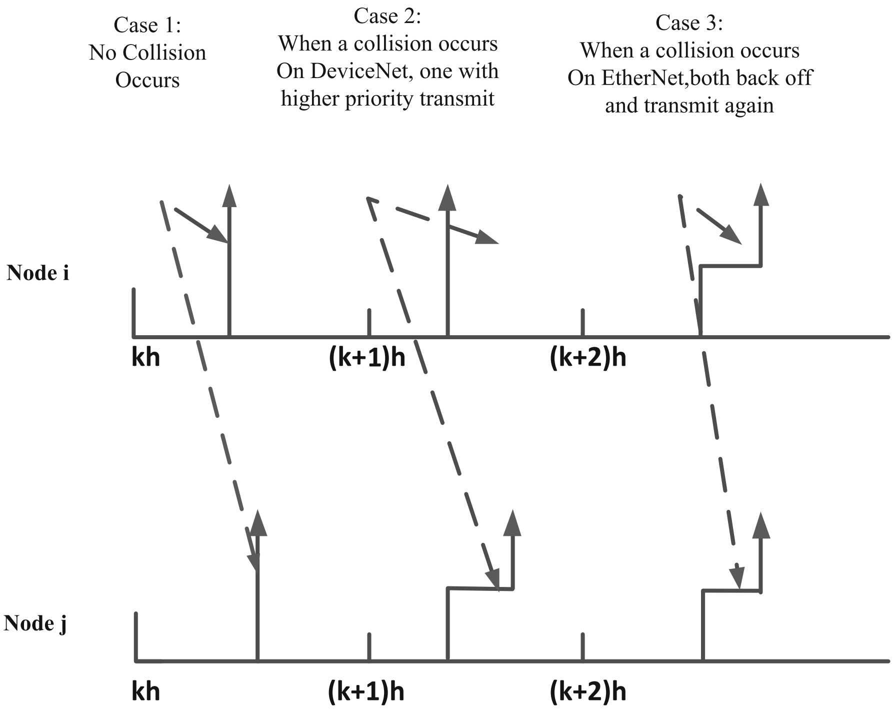


Fig. 2.3 Time diagram 1

where x_p, u_p, y_p are the state, control, and output vectors and h is the sampling period. Consider a setup having:

1. A fixed control law which is used to imply that the sensor-to-controller delay and controller-to-actuator delay can be lumped together as τ_{in},
2. Clock-driven sensors that sample the plant outputs periodically at sampling instants,
3. An event-driven controller which can be implemented by an external event interrupt mechanism and calculates the control signal as soon as the sensor data arrives, and
4. An event-driven actuators meaning that the plant inputs are changed as soon as the data become available.

In the case of $\tau_{in} < h$, a timing of the signals is shown in Fig. 2.5. The corresponding timing diagram is depicted in Fig. 2.6.

We consider the case in which the delay of each sample τ_k is less than one sampling period h. The system model becomes:

$$\begin{aligned}
\dot{x}_p(t) &= A_p x(_p t) + B_p u_p(t) \quad t \in [kh_{\tau_k}, (k+1)h + \tau_k] \\
y_p(t) &= C_p x_p(t) \\
u_p(t^*) &= -K x_p(t - \tau_k) \; t \in \{kh_{\tau_k}, k = 0, 1, 2 \ldots\}
\end{aligned} \tag{2.3}$$

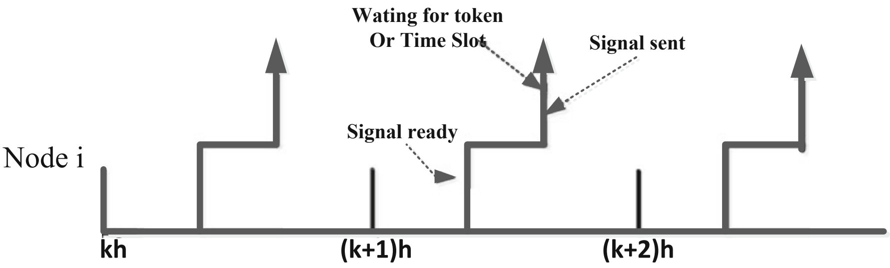


Fig. 2.4 Time diagram 2

A sampled form of system (2.3) is given by

$$\begin{aligned} x_p[(k+1)h] &= \Phi_p x_p[kh] + \Psi_p[\tau_k]u_p[kh] \\ &\quad + \Upsilon_p[\tau_k]u_p[(k-1)h] \\ y_p[kh] &= C_p x_p[kh] \end{aligned} \tag{2.4}$$

where

$$\begin{aligned} \Phi_p &= e^{Ah}\ ,\ \Psi_p = \int_0^{h-\tau_k} e^{Ah} B ds \\ \Upsilon_p &= \int_{h-\tau_k}^{h} e^{Ah} B ds \end{aligned} \tag{2.5}$$

In terms of $z[kh] = \left[u_p^t[(k-1)h] \quad [x_p^t[kh] \right]^t$, we express the controlled system as

$$\begin{aligned} z[(k+1)h] &= \Xi_p z[kh] \\ \Xi_p &= \begin{bmatrix} 0 & -K \\ \Phi_p - \Psi_p[\tau_k]K & \Upsilon_p[\tau_k] \end{bmatrix} \end{aligned} \tag{2.6}$$

If NID is constant ($\tau_k = d\ \forall k$), the stability of system (2.6) can be readily inferred from the roots of

$$\left| sI - K(sI - \Upsilon_p)^{-1}(\Phi_p - \Psi_p K) \right|$$

This is the case of static scheduling network protocol. The size of delay tolerated by the system can be estimated by the relative magnitude of Υ_p and Ψ_p to guarantee stability. Observe that in the special case $A = 0\ ,\ B = 1$, the requirement of stability entails that

$$\max\left[\frac{1}{2} - \frac{1}{Kh}, 0\right] < \frac{\tau}{\mathrm{h}} < \min\left[\frac{1}{Kh}, 1\right]$$

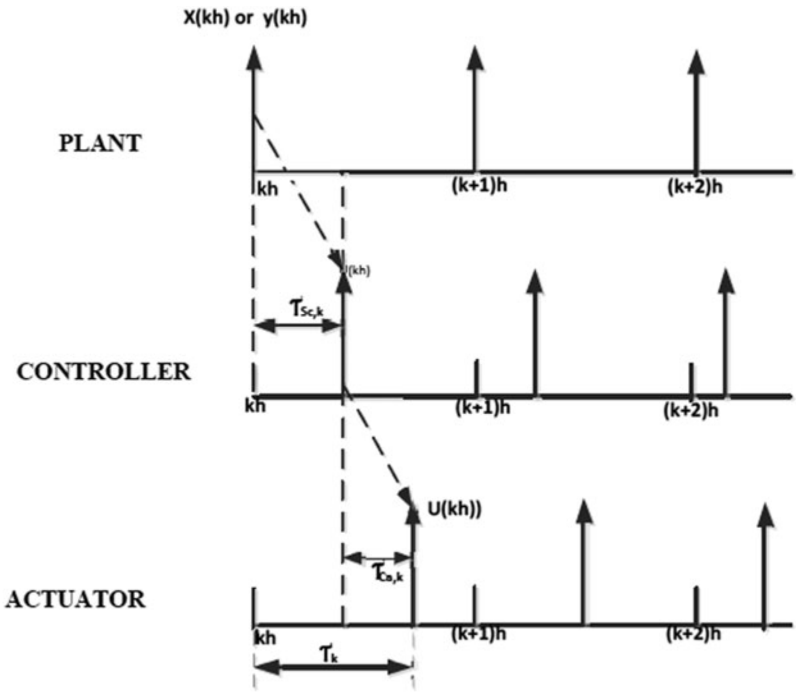


Fig. 2.5 Time diagram 3

which can determined graphically. On the other hand, letting $A = a$, $B = 1$, we obtain

$$\Xi_p = \begin{bmatrix} 0 & -K \\ e^{ah} - \frac{K}{a}(e^{a(h-\tau)} - 1) & \frac{1}{a}e^{ah}(1 - e^{-a\tau}) \end{bmatrix} \tag{2.7}$$

in which case the stability region can be determined by computer simulation.

In general, in the more practical forms of NCS with NID stability, analysis should be undertaken by dedicated software. One possible remedy to the foregoing difficulties is to employ feedback compensation for the non-delayed parts of NCS [26].

2.2.3 *Maximum Allowable Delay Bound*

For a single loop of NEC (see Fig. 2.2), consider the plant and control model as:

$$\begin{aligned} \dot{x}_p(t) &= A_p x_p(t) + B_p u_p(t) \\ y_p(t) &= C_p x(t) \end{aligned} \tag{2.8}$$

$$\begin{aligned} \dot{x}_c(t) &= A_c x(t) + B_c u_c(t) \\ y_c(t) &= C_c x_c(t - \tau_c) + E_c u_c(t - \tau_c) \end{aligned} \tag{2.9}$$

The communication delays are modeled as

$$u_c(t) = y_p(t - \tau_{sc}) \ , \ u_p(t) = y_c(t - \tau_c) \tag{2.10}$$

Defining $z(t) = \begin{bmatrix} x_p^t(t) & [x_c^t(t) \end{bmatrix}^t$, we manipulate (2.9)–(2.10) to reach

$$\dot{z}(t) = F_p x(t) + \sum_{k=1}^{3} F_k x_p(t - \tau_k) \tag{2.11}$$

$$F_p = \begin{bmatrix} A_p & 0 \\ 0 & A_c \end{bmatrix}, \ F_1 = \begin{bmatrix} 0 & 0 \\ B_c C_p & 0 \end{bmatrix}$$
$$F_2 = \begin{bmatrix} 0 & B_p C_c \\ 0 & 0 \end{bmatrix}, \ F_3 = \begin{bmatrix} B_p E_c C_p & 0 \\ 0 & 0 \end{bmatrix}$$

$$\tau_1 = \tau_{sc} \ , \ \tau_2 = \tau_{ca} + \tau_c \ , \ \tau_3 = \tau_{in} + \tau_c \tag{2.12}$$

System (2.11) is a multi-time-delay system [21] with $\tau_1 < \tau_2 < \tau_3$ for which an upper bound of the time delays can be computed by the solving the quasi-convex optimization problem

$$\max \ \tau_{\mathrm{M}} \quad \textit{subject to}$$
$$\begin{bmatrix} X(F_p^t + \sum_{k=1}^{3} F_k)+ \\ (F_p + \sum_{k=1}^{3} F_k^t)X+ & \sum_{k=1}^{3} \tau_k F_k \\ \sum_{k=1}^{3} 4\tau_k X \\ \sum_{k=1}^{3} F_k^t \tau_k & -\tau_k W_k \end{bmatrix} < 0$$
$$\begin{bmatrix} X & X F_j^t \\ F_j X & P_k \end{bmatrix} \geq 0$$
$$X > 0 \ , \ P_{kj} > 0$$
$$k = 1, \ldots, 3 \ , \ j = 0, \ldots, 3 \ , \ W_k = \sum_{j=0}^{3} P_{kj}$$

using the software environment MATLAB-LMI [8]. Observe that in this way, the variability of the induced and computational delays can be readily taken care of.

To close up, we emphasize that a model for networked control systems comprising clock-driven sensors and even-driven controller, and actuators has been examined with a view to disclose the impact of network induced delays on the stability and performance of NCS. It has been shown that the relationship between the sampling rate and the network induced delay can only be determined by simulation. An upper bound on the network delays using dynamic controllers has been developed in terms of an LMI-feasibility problem that can be readily solved using semi-definite programming tools

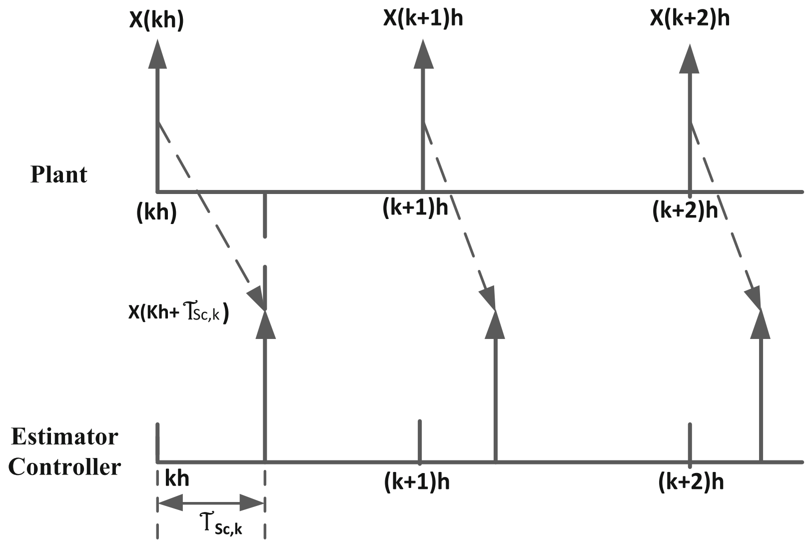


Fig. 2.6 Time diagram 4

2.3 Problems in Networks and Control

It has been a prevailing trend in modern, industrial, and commercial systems to integrate computing, communication, and control (for brevity C^3) into different levels of machine/factory operations and information processes. Networks and control is widely recognized as a very challenging and promising research field, particularly NCSs.

Control loops that are closed over a communication network have become more and more common as the hardware devices for networks and network nodes have become cheaper. The traditional communication architecture for control systems, which has been successfully implemented in industry for decades, is point-to-point; that is, a wire connects the central control computer with each sensor or actuator point. However, expanding physical setups and functionality is pushing the limits of the point-to-point architecture. Hence, a traditional centralized point-to-point control system is no longer suitable to meet new requirements, such as modularity, decentralization of control, integrated diagnostics, quick and easy maintenance, and low cost. The introduction of common-bus network architectures can improve the efficiency, flexibility, and reliability of these integrated applications through reduced wiring and distributed intelligence, and so reduce the installation, reconfiguration, and maintenance time and costs.

The change of communication architecture from point-to- point to common bus, however, introduces different forms of time-delay uncertainty between sensors, actuators, and controllers. These time delays come from the timesharing of the communication medium as well as the computation time required for physical signal coding and communication processing.

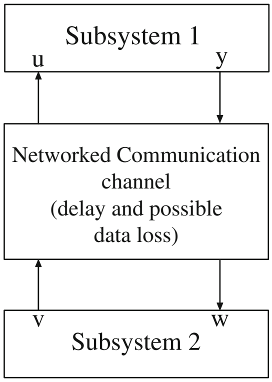


Fig. 2.7 A framework for networks and control

Time delay arising within the plant itself is a feature of many feedback control systems. This type of delay can be called natural delay or plant delay. There has been considerable research on the control of systems having plant delays, for example, see Chen et al. [6] and Manutch [20]. The control-induced or computational delay, that is, the delay caused by the control scheme adopted, was first studied in the 1970s, when digital controllers were introduced to replace analogue controllers. An analysis and example presented in Yang [42] demonstrate that computational delay in a digital controller can cause system instability. Digital controller design, taking into account the computational delay, has also been extensively studied, for example, see Astrom and Wittenmark [2]. With the rapid development in electronics, cheap and fast digital processors have become widely available, and the issue of computational delay in a digital controller has now become less important, in most applications. In an NCS, however, there is time delay caused by the signal transfer method adopted. This kind of control induced delay is commonly known in the literature as network induced delay. Network induced delays, because of their discrete and distributed nature, are quite different from the plant delays and computational delays that have been studied in the past. However, in some cases, it is possible to use tools for linear sampled-data systems for the analysis and design of certain classes of linear NCSs [24].

2.4 Classes of Problems

A representative framework for the study of networks and control is depicted in Fig. 2.7. One major view is that an NCS can be considered as having two subsystems interacting with each other through networked communication channels. Here, it is worthwhile pointing out that Fig. 2.7 can also be used to study active queue management for feedback internet congestion control. Needless to emphasize, this is a typical problem in the area of control over networks. For feedback internet congestion control, Subsystem 1 represents routers with their buffers and Subsystem 2 represents internet TCP/IP traffic sources. Subsystem 1, through the networked communication channel, the internet, sends the congestion measurement signals to traffic sources. These sources send packets through the internet, at rates according to the congestion signals and TCP algorithms adapted, to the routers. If the notations used in [42] are applied to the feedback internet congestion control problem, y, w, v and u in the figure correspond to p, q, x and y, respectively.

Using the proposed general framework for networks and control, the study of NCSs can be divided into two classes of problem: Class I and Class II.

2.4.1 Class I Problems

For Class I problems, Subsystem 1 represents a distributed plant to be controlled with sensors and actuators. Subsystem 1 : $\Sigma_1 = (A, B, C, D)$ is continuous and has state variable x, output vector y, and input vector u

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.13}$$

Subsystem 2 represents a digital controller $\Sigma_2 = \Sigma_O + \Sigma_C$, where Σ_O and Σ_C represent the dynamics of an observer and a controller, respectively. In principle, the separation principle can be assumed to be valid in NCS. It is proved that the separation theorem still holds in networked control system in which network induced delay is longer than the sampling period. If there is no observer, or if the observer design is not explicitly concerned, then $\Sigma_2 = \Sigma_C = (E, F, G, H)$ is a discrete-time system with state variable z, such that

$$\begin{aligned} z_{k+1} &= Ez_k + Fw_k \\ v_k &= Gz_k + Hw_k \end{aligned} \tag{2.14}$$

The input to the controller, $w_k = w(kT)$, is a delayed version of the sensor output $y(t)$ at some sampling instant. For a particular element i in w_k, w_k^i

$$w_k^i = y^i(t - \tau_{sc}^i(t)) \tag{2.15}$$

where $t_{sc}^i(t)$ is the sensor controller time delay for the ith sensor signal $y^i(t)$. (In some literatures, sensor controller delay is also called feedback-path delay or down-link delay.) Let

$$0 < \tau_{sc}^i(t) \le \tau_{sc\max} \qquad \forall i \tag{2.16}$$

where $\tau_{sc\max}$ is the upper bound of $\tau_{sc}^i(t)$, which may be greater than the sampling period T. Similarly, $u(t)$ in Σ_1 is a delayed version of the controller output v_k. This is noted as controller–actuator delay $\tau_{ca}^j(t)$ (also called forward-path delay or down-link delay), and one can set

$$0 < \tau_{sc}^j(t) \le \tau_{ca\max} \qquad \forall j \tag{2.17}$$

where $\tau_{ca\max}$ is the upper bound of $\tau_{ca}^{j}(t)$, which may also be greater than T. These network induced delays are sums of several small delays, for example, medium access delay, transportation delay, congestion delay, and so on, and their values depend on the network architectures, protocols, operating conditions, and so on.

Some general assumptions for Class I problem are as follows:

1. The controller is time driven, with a sampling period of T. Time driven means that the controller starts each calculation at the time instant kT.
2. There are N sensors in the plant, each sampling with a period of $T^{pi} (1 \leq i \leq N)$ and

$$T^{pi} = \beta_i T \tag{2.18}$$

   where β_i can be any positive real number. In a simple case, all β_i are integers. This can be further simplified to let all β_i be equal to one, that is, all sensors have the same sampling periods as that of the controller.
3. The controller clock and sensor clocks are not synchronised. There is a fixed time skew gi between the controller clock and the ith sensor clock when β_i is an integer. However, measurement signals are indexed: $y_l^i, y_{l-1}^i, y_{l-2}^i, \ldots$, where y_l^i is the current sampled signal; y_{l-1}^i is the signal sampled one sampling period before and so on. The controller always uses the latest signal, that is, the signal has the largest index l. Because of the randomness of $\tau_{sc}^i(t)$ and the possibility of packet loss, it may be possible that

$$w_{k+1}^i = w_k^i = y_l^i(t - \tau_{sc}^i(t)) \tag{2.19}$$

   That is, at the controller site with a sampling period of T, the signal wi used as the controller input at two (or more) consecutive sampling instances is the same (y_l^i) even when $T = T^{pi}$.
4. There is a zero-order hold for each actuator. Therefore, the plant input signals $u(t)$ in (2.13) are piecewise constants. Actuators are event driven, that is, whenever a newest v_l^j signal is available to the jth actuator, its corresponding $u^j(t)$ is updated to this value immediately. Here, the newest is in the same sense as that used in assumption (2.15). Again because of the randomness of $\tau_{ca}^j(t)$ and the possibility of packet loss, a newly arrived signal is not necessarily a newest signal. Therefore, it may be possible that $u^j(t)$ is holding an unchanged value for a period longer than T.

   If the controller computational delay is included in $\tau_{ca}^j(t)$, this will not affect system analysis and simulation. Therefore, from now on, it is assumed that $\tau_{ca}^j(t) = \tau_{canet}^j(t) + \tau_{cacomp}^j(t)$, where $\tau_{canet}^j(t)$ is the network induced delay as discussed before and $\tau_{cacomp}^j(t)$ is the computational delay. With this definition, the controller can be considered as an ideal controller without computational delay.

Pertinent comments on the formulation of Class I problems are in order:

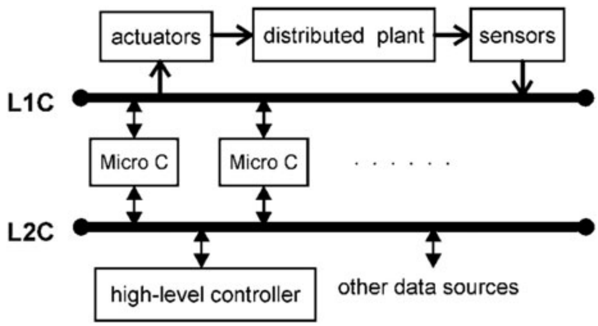
- A simple way of clock synchronisation between the controller and the plant is possible. The controller sends a message to the plant and the plant will send its clock reading back to the controller. The controller records the clock offset and the *round-trip time (RTT)*. The measurement is carried out many times and the controller uses the clock offset with the minimum RTT. Due to the randomness of sensor controller delays, although clock synchronisation makes system analysis easier, it has little benefit for real-system performance—it is time delays and data loss, not clock time-skew, which cause deterioration in system performance.

- For the benefit of system performance, event-driven actuators are preferred to time-driven actuators to eliminate the extra waiting time required, that is, the waiting for the next actuator sampling instance to arrive, before an input signal to the actuator can be updated. (In fact, time-driven actuators can be considered as a special case, with events happening at regular sampling instances, of event-driven actuators.) However, the controller, because it receives signals from multisensors, must be time driven.
- Only the newest data are used, and a long-delayed packet, even when it finally arrives, will be ignored. Packet loss can be considered as a packet having an infinite delay. Therefore in these formulations, there is no need to treat data packet loss as a separate issue.
- In most analytical studies, it is assumed that all bi are equal to one, that is, all sensors have the same sampling periods as that of the controller. This makes system analysis easier. However, multi-rate-sampling is quite common in many applications. For example, in a vehicle engine management system, the exhaust lambda value is sampled more frequently than engine temperature. Reducing sampling rates for slowly changing variables can reduce CPU and network load.
- In typical networks, in particular, those designed for real-time control applications, such as ControlNet, DeviceNet and so on, controller–actuator delays can be made very small and deterministic, for example, by scheduling or giving priority to control signals over other traffic in a network. Therefore, the issue of controller–actuator delay in these networks can be reduced to an issue of ordinary deterministic time delay. In some other applications, by using smart actuators that can compute their own control signals, there is no time delay (apart from computational delay) between controller and actuator. In such applications, the remaining sensor–controller delay can sometimes be included in the plant model for the purposes of system analysis and design, and many results for plant time-delayed systems can be applied.
- An observer in many designs is an integral part of the controller. To take this into account, w_k in Fig. 2.7 should be considered as the input to the general controller $\Sigma_2 = \Sigma_O + \Sigma_C$, and the output v_k should be considered as the output from this general subsystem of Σ_2. More detailed equations for Σ_O and Σ_C should then be used to replace (2.14). With these modifications, the general formulation can still be used.

2.4.2 Class II Problems

To describe Class II problems, a two-level communication model is presented in Fig. 2.8. In this model, microcontrollers communicate with the sensors and actuators, that are attached to the plant through a communication network, typically some

Fig. 2.8 NCS with two-level communications



kind of field bus dedicated to real-time control. This communication can be called level 1 communication (L1C). Microcontrollers also communicate with a high-level computer system, which typically functions as an adaptive or parameter-optimisation controller, through another communication network. This network is typically some kind of existing local area network, wide area network (WAN), or possibly the internet. This communication can be called level 2 communication (L2C). Control signal traffic at level 2 shares the available network bandwidth with other data communications.

The foregoing model can be used for many industrial distributed plants for the study of NCSs. For example, an industrial process plant where a number of embedded controllers (microcontrollers) through L1C control a number of sub-processes within the plant, and supervisory computers through L2C coordinate various low-level control actions carried out by local controllers.

Remark 2.1 *Using this two-level communication model, we can say that a Class I problem is a basic control problem, with stabilization, regulation and so on, achieved via L1C, whereas a Class II problem is a controller parameter adaptation problem, achieved via L2C. Therefore, for Class II problems, Subsystem 1 in Fig. 2.7 represents a controlled plant and Subsystem 2 represents a controller parameter adaptation algorithm. Now in Fig. 2.7, $w(t)$, the delayed signals of $y(t)$, are signals required for the parameter adaptation algorithm used; and $u(t)$, the delayed signals of $v(t)$, are parameters to be changed in the Subsystem 1 controller. The detailed mathematical Eqs. (2.13) and (2.14), describing the dynamics of the two subsystems, will depend on the specific problem under study. With a slightly different interpretation, the other equations and the general formulation for Class I problems presented there, are also valid for Class II problems.*

Remark 2.2 *By and large, it is proposed to develop an NCS simulation package based on the general framework and formulations proposed in this chapter. In this package, MATLAB/Simulink blocks will be used to simulate the dynamics of Subsystems 1 and 2 in Fig. 2.7. Some further code will be developed to simulate the exact characteristics of the time delay and data loss in the communication network, but without the need to specify the details of the network protocols, operating conditions, and so on. The package will be run on a single PC and will differ from the TrueTime toolbox [76] that*

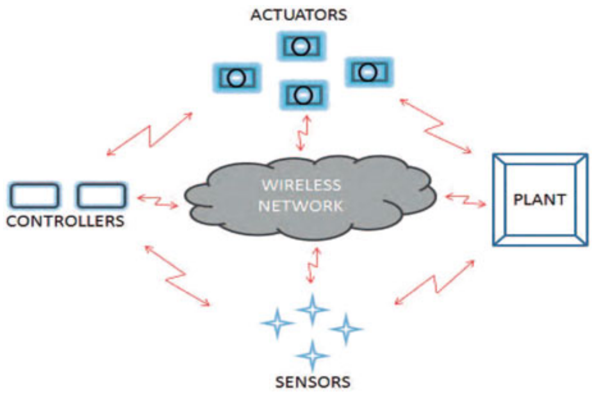


Fig. 2.9 A generic model of NCS

is specifically designed for the study of integrated control and real-time scheduling [77] (for a brief description of the True Time toolbox [77]).

2.5 Notes

This chapter paves the way to discuss numerous concepts pertaining to networked control systems. The references [16, 22, 23, 36, 37, 41] are recommended to consult for further reading. Additional challenging issues were raised in [1, 4, 13, 14, 17, 34, 38–40, 43]. More open problems were addressed [12, 31], which should motivate further research investigations.

2.6 Suggested Problems

In the following set of problems, reference is made to the generic prototype of wireless networked control system (WNCS) depicted in Fig. 2.9.

Problem 2.1 Adopting an information processing viewpoint, identify the possible routes in Fig. 2.9 and examine possible blockage of these routes.

Problem 2.2 Consider that the plant in Fig. 2.9 is described by the linear state-space model

$$\begin{aligned}\dot{x}_p(t) &= A_p x_p(t) + B_p u_p(t) + \Gamma_p w_p(t)\\ y_p(t) &= C_p x_p(t)\end{aligned} \tag{2.20}$$

Design a class of controllers (for example, output feedback, adaptive, model-predictive) to regulate the dynamic behavior of system (2.20)

Problem 2.3 Consider that the plant in Fig. 2.9 is described by the linear state-space model 2.20 with $w_p(t) \equiv 0$. Let the control law be of hold-input mechanism and expressed as linear function of the state and let the sensor-to-controller route and the controller-to-actuator route be subject to randomly occurring packet losses. Put the system in a discrete-time format using an appropriate sampling interval. Build up the resulting closed-loop system and derive conditions under which it exhibits mean-square exponentially stability.

Problem 2.4 The system matrices of a two-axis milling machine tool of the type 2.20 is given by

$$A_p(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -18.18 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -17.86 \end{bmatrix}, \quad B_p(t) = \begin{bmatrix} 0 & 0 \\ 515.38 & 0 \\ 0 & 0 \\ 0 & 517.07 \end{bmatrix}$$

Using a sampling interval of $T_s = 0.1s$, apply the results of **Problem 2.3** with appropriate rates of randomly occurring packet losses.

References

1. P. Antsaklis, J. Baillieul, *Special Issue on Technology of Networked Control Systems. Proceedings the IEEE*, vol. 95, no. 1 (2007)
2. K.J. Astrom, B. *Wittenmark, Computer Controlled Systems: Theory and Design* (Prentice-Hall, 1997)
3. J. Baillieul, P. Antsaklis, Control and Communication Challenges in Networked Real-time Systems. *Proceedings the IEEE*, vol. 95, no. 1 (2007)
4. M.S. Branicky, S.M. Phillips, W. Zhang, Stability of Networked Control Systems: Explicit Analysis of Delay. *Proceedings American Control Conference*. (Chicago 2000), pp. 2352–2357
5. J.H. Braslavsky, R.H. Middleton, J.S. Freudenberg, Feedback Stabilization Over Signal-to-noise Ratio Constrained Channels. *Proceedings Amer. Contr. Conf.*, 4903–4908 June 2004
6. J. Chen, K. Gu, V. Kharitonov, *Stability of Time-Delay Systems*. (Birkhauser, Boston, 2002)
7. N. Elia, S.K. Mitter, Stabilization of Linear Systems with Limited Information. *Proceedings IEEE Transtion Automatic Controle*, vol. 46, no. 9, 1384–1400, Sept 2001
8. P. Gahinet, A. Nemirovski, A.L. Laub, M. Chilali, LMI Control Toolbox, The Math Works, Inc., (Boston, 1995)
9. K. Gu, V.L. Kharitonov, J. Chen, *Stability of Time-Delay Systems*. (Birkhauser, 2003)
10. Y. Halevi, A. Ray, Integrated communication and control systems: Part I-analysis. J. Dynamic. Syst. Measure. Contr. **110**, 367–373 (1988)
11. J.P. Hespanha, A. Ortega, L. Vasudevan, Towards the Control of Linear Systems with Minimum Bit-rate. *Proceedings Int. Symp. Mathematical Theory of Networks and Syst.* Aug 2002
12. W.P.M.H. Heemels, R. Andrew N. Teel, van de Wouw, D. Nesic, Networked control systemsWith communication constraints: Tradeoffs between transmission intervals, delays and performance, IEEE Trans. Automatic Cont. **55**(8), 1781–1796 (2010)
13. J.P. Hespanha, P Naghshtabrizi, Y. Xu, A Survey of Recent Results in Networked Control Systems. *Proceedings of the IEEE*, vol. 95, no. 1, (2007)
14. H. Ishii, Bruce A. Francis, Stabilization with control networks. Automatica. **38**, 1745–1751, (2002)

15. F.L. Lian, J. Moyne, D. Tilbury, Network design consideration for distributed control systems, IEEE Trans. Control Syst. Technol. **10**(2), (2002)
16. D. Liberzon, J.P. Hespanha, Stabilization of nonlinear systems with limited information feedback, IEEE Trans. Automat. Contr. **50**(6), 910–915 (2005)
17. I. Lopez, J.L. Piovesan, C.T. Abdallah, D. Lee, O. Martinez, M. Spong, Practical Issues in Networked Control Systems. *Proceedings American Control Conference*, (2006) p. 4201–4206
18. S.H. Low, F. Paganini, J.C. Doyle, Internet congestion control. IEEE Control Syst. Mag. **22**, 2843, (2002)
19. M.S. Mahmoud, *Robust Control and Filtering of Time-Delay Systems*. (North-Holland, 2000)
20. N.C. Martins, M.A. Dahleh, Feedback control in the presence of noisy channels: 'Bode-like' fundamental limitations of performance, (2004)
21. M.S. Mahmoud, *Robust Control and Filtering for Time-Delay Systems*. (Dekker, New York, 2000)
22. G.N. Nair, R.J. Evans, Exponential stabilizability of finite-dimensional linear systems with limited data rates. Automatica. **39**(4), 585–593 (2003)
23. G.N. Nair, R.J. Evans, I.M.Y. Mareels, W. Moran, Topological feedback entropy and nonlinear stabilization, IEEE Trans. Automat. Contr. **49**(9), 1585–1597 (2004)
24. D. Nesic, A.R. Teel, Input-output stability properties of networked control systems. IEEE Trans. Autom. Control. **49**, 1650–1667 (2004)
25. J. Nilsson, B. Bernhardsson, B. Wittenmark, Stochastic analysis of control of realtime systems with random time delays. Automatica. **34**, 57–64 (1998)
26. H.S. Park, Y.H. Kim, D.S. Kim, W.H. Kwon, A scheduling method for networkbased control systems, IEEE Trans. Control Syst. Technol. **10**, 319–330 (2002)
27. L.L. Peterson, B.S. Davie, *Computer Networks, A System Approach*, 3rd edn. (Morgan Kaufman, New York, 2003)
28. U. Ozguner, H. Goktag, H. Chan, Automotive suspension control through a computer communication network, *Proceedings IEEE Conf. Control Applications* (1992)
29. A. Ray, Distributed data communication networks for real-time process control. Chem. Eng. Commun. **65**, 139–154 (1988)
30. A. Sahai, Anytime Information Theory, Ph.D. dissertation, MIT, Cambridge (2001)
31. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, S.S. Sastry, Foundations of control and estimation over lossy networks, *Proceedings the IEEE*, vol. 95, no. 1 (2007)
32. J.A. Sparks, Low cost technologies for aerospace applications. Microprocess. Microsyst. **20**, 449–454 (1987)
33. W. Stallings, *Data and Computer Communications*, 6th edn. (Prentice Hall, Englewood Cliffs, 2000)
34. D.J. Stilwell, B.E. Bishop, Platoons of underwater vehicles, IEEE Control Syst. **20**(6), 45–52 (2000)
35. M. Tabbara, D. Nesic, Stability of networked control systems with stochastic protocols. Proceedings 2007 American Control Conference. (2007), pp. 1717–1722
36. G. Tang, L. Guo, Adaptive stabilization of unknown linear systems under communication constraints, (in Chinese), *Proceedings Chinese Control Conf.*, March (2005)
37. S. Tatikonda, S. Mitter, Control under communication constraints, IEEE Trans. Automat. Contr. **49**(7) 1056–1068 (2004)
38. Y. Tipsuwan, M.Y. Chow, Control methodologies in networked control systems. Control Eng. Pract. **11**, 1099–1111 (2003)
39. G.C. Walsh, H. Ye, L. Bushnell, Stability analysis of networked control systems. *Proceedings American Control Conference*, San Diego, 1999, pp. 2876–2880
40. G.C. Walsh, H.Ye, L. Bushnell, Stability analysis of networked control systems, *Proceedings American Control Conference*, San Diego, 1999, pp. 2876–2880
41. W.S. Wong, R.W. Brockett, Systems with finite communication bandwidth constraints-II: Stabilization with limited information feedback. IEEE Trans. Automat. Contr. 44(5), (1999)
42. T.C. Yang, Computational delay in digital and adaptive controllers. IEE Control 90 Conference, Brighton 1990

43. T.C. Yang, Networked control system: A brief survey. IEE *Proceedings Control Theory Appl.* **153**(4), (2006)
44. S. Yian, Li Yu, W. A. Zhang, A Switched Systemapproach to Networked H Filtering with Packet Losses. Circuits, Systems & Signal Process, January (2011)
45. S. Zampieri, Trends in Networked Control Systems. *Proceedings the 17th World Congress. Seoul, Korea* 2886–2894 (2008)

http://www.springer.com/978-3-319-04152-0

Control and Estimation Methods over Communication Networks
Mahmoud, M.S.
2014, XXI, 521 p. 191 illus., 89 illus. in color., Hardcover
ISBN: 978-3-319-04152-0