Analysis and Design of Real-Time Systems with Random Delays

Nilsson, Johan

1996

Document Version:
Publisher's PDF, also known as Version of record

Department of Automatic Control
Lund Institute of Technology
Box 118
S-221 00 Lund Sweden

Document name
LICENTIATE THESIS

Date of issue
May 1996

Document Number
ISRN LUTFD2/TFRT--3215--SE

Author(s)
Johan Nilsson

Supervisor
Bo Bernhardsson and Björn Wittenmark

Sponsoring organisation
Swedish National Board for Industrial and Technical Development (NUTEK).

Title and subtitle
Analysis and Design of Real-Time Systems with Random Delays

Abstract

Control loops that are closed over a communication network get more and more common. A problem with such systems is that the transfer delays will be varying with different characteristics depending on the network hardware and software. To analyze control systems with network delays in the loop we have to model these. The network delay is typically varying due to varying network load, scheduling policies in the network and the nodes, and due to network failures. Three network models of different complexity are studied:

Constant delay
Random delay, which is independent from transfer to transfer
Random delay, with probability distributions governed by a Markov chain

To design a controller for a distributed digital control system it is important to know how to analyze such systems. In standard computer control theory it is assumed that the closed loop system is time-invariant. In a system with varying delays this is not true. In the work it is shown how to analyze stability and expected performance of linear controllers where the network delays are described by one of the three developed network models above. Methods to evaluate a quadratic cost function are developed. Through the same analysis we find criteria for mean square stability of the closed loop for the different network models.

The Linear Quadratic Gaussian (LQG) optimal controller is developed in the case of random delays that are independent from transfer to transfer. The derived controller uses knowledge of old time delays. These can be calculated using "time-stamping" of messages in the network. "Time-stamping" means that every transfered signal is marked with the time of generation. The receiving node can then calculate how long the transfer delay was by comparing the time-stamp with the node's internal clock.

Key words
Delay compensation, Distributed computer control systems, Real-time systems, Stochastic control, Stochastic parameters, Jump linear systems.

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
84

Recipient's notes

Security classification

Analysis and Design of Real-Time Systems with Random Delays

Analysis and Design of Real-Time Systems with Random Delays

Johan Nilsson

Department of Automatic Control
Lund Institute of Technology
Lund, May 1996

Department of Automatic Control
Lund Institute of Technology
Box 118
S-221 00 LUND
Sweden

ISSN 0280–5316
ISRN LUTFD2/TFRT–3215–SE

Printed in Sweden by Reprocentralen, Lunds Universitet.
Lund 1996

Contents

Preface

My involvement in real-time systems with random delays started in late 1994. The work was initialized by the DICOSMOS research project, which is a joint project between Department of Automatic Control at Lund Institute of Technology, DAMEK Mechatronics Division at the Royal Institute of Technology, and Department of Computer Engineering at Chalmers Institute of Technology. The aim of the project is to investigate theories and design rules needed when applying distributed computer solutions for controlling complex machinery such as industrial robots, production machines, vehicles, aircraft etc.

I would like to thank my supervisors Doctor Bo Bernhardsson and Professor Björn Wittenmark for all help and encouragement they have given me. I would also like to thank Professor Jan Sternby for reading and giving valuable comments on the thesis.

The work has been supported by NUTEK, Swedish National Board for Industrial and Technical Development, Project Dicosmos, 93-3485.

1

Introduction

Control loops that are closed over a communication network get more and more common as the hardware devices for network and network nodes become cheaper. A control system communicating with sensors and actuators over a communication network will be called a *distributed control system*. In distributed control systems, see Figure 1.1, data are sent and received by network nodes of different kind and manufacturer. Network nodes that are of specific interest for distributed control are sensor nodes, actuator nodes, and controller nodes. Sensor nodes measure process values and transmit these over the communication network. Actuator nodes receive new values for the process inputs over the communication network and apply these on the process input. Controller nodes read process values from sensor nodes. Using a control algorithm control signals are calculated and sent to the actuator nodes. The system setup with a common communication network reduces cost of cabling, and offers modularity and flexibility in system design. The distributed control setup is powerful, but some caution must be taken. Communication networks inevitably introduce delays, both due to limited bandwidth, but also due to overhead in the communicating nodes and in the network. The delays will in many systems be varying in a random fashion. From a control perspective the control system with varying delays will no longer be time-invariant. As an effect of this the standard computer control theory can not be used in analysis and design of distributed control systems. The thesis addresses the problem of analysis and design of control systems when the communication delays are varying in a random fashion. Models

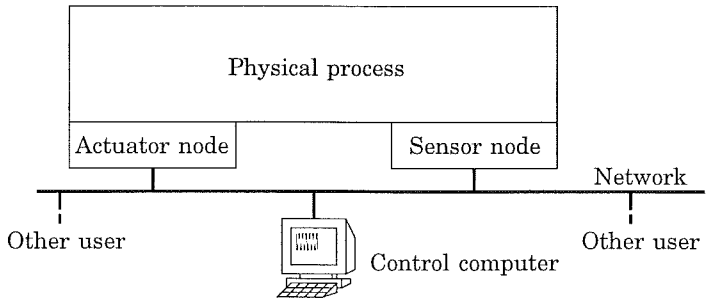

Figure 1.1 Distributed control system with sensor node, controller node, and actuator node. The communication network is also used for other applications in the system.

for communication network delays are developed. The most advanced model has an underlying Markov chain, which generates the probability distributions of the time delays. For the different network models closed loop stability and evaluation of a quadratic cost function are analyzed. The LQG-optimal controller is derived in a setup where the time delays are independent from sample to sample. The derived controller uses knowledge of old time delays. This can be achieved by so called "time-stamping", all transfered signals are marked with the time they were generated. By comparing the "time-stamp" with the internal clock of the controller the time delay can be calculated. It is shown that the optimal controller is the combination of an LQ-controller and a Kalman filter, i.e. the separation principle applies.

Outline of the Thesis

The contents of the thesis are as follows:

Chapter 2: Problem Formulation This chapter gives an introduction to the problem formulation. A short review of clock synchronization and networks for distributed control is also presented. The chapter is concluded with a summary of work related to this thesis.

Chapter 3: Modeling of Network Delays Models for the network induced delays are developed. The models are well suited for

analysis and design of distributed control systems.

Chapter 4: Analysis of Control Laws Results are developed to determine system stability and values of quadratic cost functions given a proposed controller. The analysis is made for the network models developed in Chapter 3. Some examples are given.

Chapter 5: Optimal Stochastic Control The LQG-controller is derived for systems with time delays which are modeled to be independent from transfer to transfer. A design example comparing four controller designs is given.

Chapter 6: Conclusions and Future Work In the last chapter future work and extensions are discussed.

The thesis is concluded with two appendices, one on Kronecker products, and one containing some results from probability theory.

2

Problem Formulation

2.1 Distributed Control

We will study the closed loop system depicted in Figure 2.1. The actuators and sensors are connected to a communication network. These units receive respectively send control information to the centralized controller. The centralized controller is connected to the network, and communicates with sensors and actuators by sending messages over the network. Sending a message over a network typically takes some time. Depending on the network and scheduling policy in the system this transfer time can have different characteristics. The transfer time can in some setups be nearly constant, but in many cases it is varying in a random fashion. The length of the transfer delay can, for instance, depend on the network load, priorities of the other ongoing communications, and electrical disturbances, Ray (1987). Depending on how the sensor, actuator, and controller nodes are synchronized several setups can be found. Several previous authors have suggested such control schemes with slightly different timing setups. The different setups come from whether a node is event-driven or clock-driven. By event-driven we mean that the node starts its activity when an event occurs, for instance, when it receives information from another node over the data network. Clock-driven means that the node starts its activity at a prespecified time, for instance, the node can run periodically. There are essentially three kinds of computer delays in the

system, see Figure 2.1:

- Communication delay between the sensor and the controller, τ^{sc}.
- Computational delay in the controller, τ^{c}.
- Communication delay between the controller and the actuator, τ^{ca}.

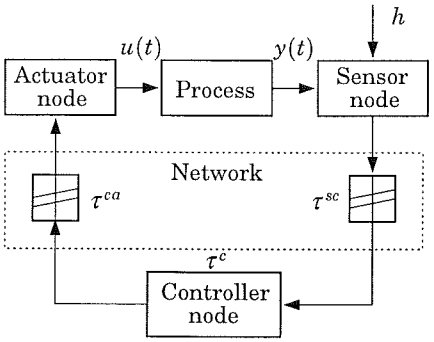

Figure 2.1 Distributed digital control system with induced delays, τ^{sc} and τ^{ca}. The computational delay in the controller node, τ^{c}, is also indicated.

The *control delay* for the control system, the time from when a measurement signal is sampled to when it is used in the actuator, equals the sum of these delays.

One important problem in this control system setup is the delays, which are varying in a random fashion. This makes the system time-varying and theoretical results for analysis and design for time-invariant systems can not be used directly. One way to get rid of the time variations is to introduce clocked buffers on the input in the controller node and the actuator node. If these buffers are chosen large enough, larger than the worst case delay, the delay for a transfer between two nodes is deterministic. This scheme was proposed in e.g. Luck and Ray (1990). Introduction of buffers in the loop means that we sometimes are using older information than we need to. It is shown in Chapter 5 that this can lead to a degradation of performance in comparison with an event-driven setup.

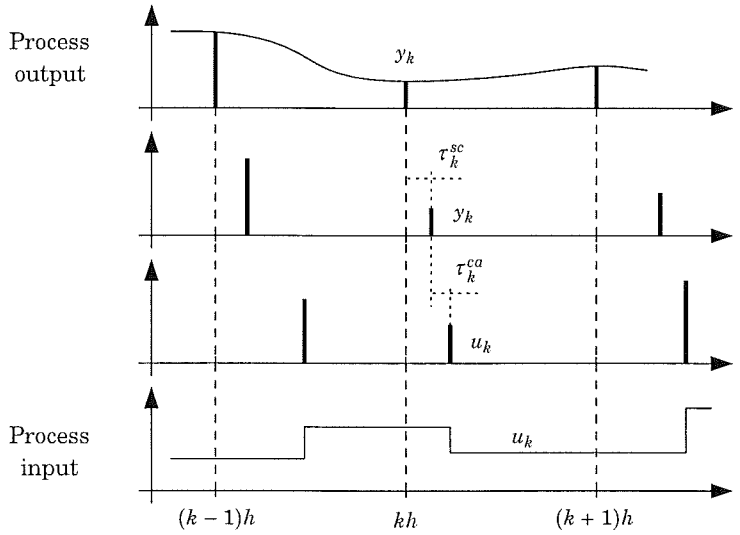

Figure 2.2 Timing of signals in the control system. The first diagram illustrates the process output and the sampling instants, the second diagram illustrates the signal into the controller node, the third diagram illustrates the signal into the actuator node, and the fourth diagram illustrates the process input, compare with Figure 2.1.

From a sampled data control perspective it is natural to sample the process output equidistantly with a sample period of h. It is also natural to keep the control delay as short as possible. The reason is that time delays give rise to phase lag, which often degenerate system stability and performance. This motivation suggests a system setup with event-driven controller node and event-driven actuator node, which means that calculation of the new control signal respectively D/A-conversion of the new control signal takes place as soon as the new information arrives from the sensor node and the controller node respectively. In Figure 2.2 the timing in such a system is illustrated. A drawback with this setup is that the system becomes time-varying. This is seen from Figure 2.2 in that the control signal is changed at irregular times.

In the subsequent chapters we will analyze and design controllers

with equidistantly sampling sensor node and event-driven controller and actuator node. We will also make the assumption that the control delay is less than the sampling period. This can be motivated in several ways. From a control point of view, a normal design usually has $0.2 \leq \omega h \leq 0.6$, where h is the sampling period and ω is the natural frequency, see Åström and Wittenmark (1990). With a delay equal to one sample this design has a phase lag induced by the controller, ϕ_{lc}, of $11° \leq \phi_{lc} \leq 34°$. An even larger phase lag would make many processes hazardous to control. If we have a larger control delay than the sampling period, h, samples may arrive in a non-chronological order at the actuator-node. This would make both implementation of algorithms and system analysis much harder. The condition that the control delay is less than h can be lighten to that the control delay may not vary more than h, which also guarantees that samples arrive in chronological order. We will in the following only look at the influences from τ^{sc} and τ^{ca}. The effect of τ^{c} can be embedded in τ^{ca}. We will also assume that we have the knowledge of how large the previous transfer-delays in the loop were. Ways to implement this feature is discussed in the sequel.

Knowledge of Old Time Delays – Time Stamps

Using synchronized clocks in the nodes, delay information can be extracted by including the time of generation to every message. Clock synchronization is further discussed in Section 2.3. In most networks the extra network load introduced by the time stamp is negligible in comparison with message and network overhead. The delay information is used in the controller node. The controller node can easily calculate τ^{sc} by comparing the time stamp of the sensed signal with the controller node's internal clock. The controller can also obtain information about the last τ^{ca}. One way to achieve propagation of old τ^{ca} to the controller is to immediately send a message back from the actuator-node to the controller containing the transfer time τ^{ca}. It is however not certain that the controller will have received this message when the next control signal is to be calculated. The timing for the control system between two control signal calculations is shown in Figure 2.3. We have here introduced the transfer delay τ^{ca-b} for the transfer-time to send back information about the length of τ^{ca} to the controller. In some setups the controller will immediately know when the control

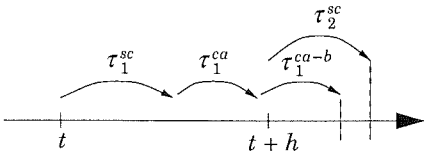

Figure 2.3 Timing plot showing delays during a clock cycle.

signal arrived to the actuator. In this case τ^{ca-b} is automatically 0. The condition that the old τ^{ca} is known when we calculate the control signal can in the general case be written as

$$\tau_1^{sc} + \tau_1^{ca} + \tau_1^{ca-b} < \tau_2^{sc} + h. \tag{2.1}$$

We will assume that the sum of the time delays in one control cycle, the control delay, is less than the sampling interval h. As the control delay is the sum of τ^{sc} and τ^{ca}, it is reasonable to assume that each of the delays be distributed on $[0, \alpha h]$, where $\alpha < 0.5$. The problem of guaranteed knowledge of old time delays will be analyzed for some special cases of communication network behavior.

Total randomness If the communication network gives time delays that are random and independent it is seen from (2.1) that the condition on the distribution of the time delays is

$$\alpha < \frac{1}{3}. \tag{2.2}$$

Total order If we add the requirement that network messages are transmitted in the order they were generated, the message containing the length of τ^{ca} will always be sent before the next measurement if

$$\alpha < \frac{1}{2}. \tag{2.3}$$

An implementation of such a system would require something like a global queue for messages to be sent in the network.

Priority Some communication networks, for instance CAN, see Section 2.2, have the possibility to give messages priorities and guarantee that waiting messages are sent in priority order. In such a system we can give the message containing τ^{ca} a higher priority than the message containing the new measurement. This guarantees knowledge of the old τ^{ca} if

$$\alpha < \frac{1}{2}. \tag{2.4}$$

Summary of Assumptions

We will make the following assumptions about the control system:

- The sensor node is time-driven. The output of the process is sampled periodically without any scheduling disturbances. The sampling period is h.
- The controller node is event-driven. The control signal is calculated as soon as the sensor data arrives at the controller node.
- The actuator node is event-driven. The control signal is applied to the process as soon as the data arrives at the actuator node.
- The communication delays τ^{sc} and τ^{ca} are randomly varying with known stochastic properties. The total time delay $\tau^{sc} + \tau^{ca}$ have arbitrary distribution, but is always less than one sampling period.
- The lengths of the past time delays are known to the controller.

2.2 Networks

Communication networks were introduced in control systems in the 70's. At that time the driving force was the car industry. The motives for introducing communication networks were reduced cost for cabling, modularization of systems, and flexibility in system setup. Since then, several types of communication networks have been developed. Communication protocols can be grouped into *fieldbuses* (e.g. FIP and PROFIBUS), *automotive buses* (e.g. CAN), "other" *machine buses* (e.g 1553B and the IEC train communication network), *general purpose networks* (e.g.

IEEE LAN's and ATM-LAN) and a number of *research protocols* (e.g. TTP), see Törngren (1995). A short summary is now given of some of the most used communication protocols, see also Olsson and Piani (1992) and Tindell and Hansson (1995).

FIP (Factory Instrumentation Protocol)

FIP was developed by a group of French, German, and Italian companies. FIP uses a twisted pair conductor and the transmission speeds are from 31.25 kbit/s up to 2.5 Mbit/s, depending on the spatial dimension of the bus. For a transmission speed of 1 Mbit/s the maximum length of the bus is 500 m. The maximum number of nodes in a FIP network is 256.

In a FIP-network one node acts as *bus arbitrator*. The *bus arbitrator* cyclically polls all nodes in the network to broadcast its data on the network. The inactive nodes listen to the communication and recognizes when data of interest to the node is sent. The FIP-network can be seen as a distributed database, where the database is updated periodically.

PROFIBUS (Process Fieldbus)

PROFIBUS was developed by a group of German companies and is now a German standard. A screened twisted pair is used as conductor. The transfer speed can be from 9.6 kbit/s to 500 kbit/s. The maximum length of the bus is 1200 m. Up to 127 stations can be connected to the network. PROFIBUS messages can be up to 256 bytes long. PROFIBUS is a token-passing network. The nodes are divided into *active* and *passive* nodes. The node which holds the token has the permission to send data on the network. The token is passed around in the network between the *active* nodes. *Active* nodes can transmit when they hold the token. *Passive* nodes need to be addressed by an *active* node to be allowed to send data on the network.

CAN (Controller Area Network)

CAN was developed by the German company Bosch for the automation industry. CAN is defined in the ISO standards 11898 and 11519-1. The transfer speed can be 1 Mbit/s if the bus is no longer than 50 m, and 500 kbit/s if the bus is longer than 50 m. There is no limit on the

number of nodes. A node can start transmitting at any time if the bus is silent. If several nodes are trying to transmit an arbitration starts. The node trying to send the message with highest priority gets the right to use the bus. There are 2032 priority levels. The possibility to give messages different priorities can be used to assure that the message containing the delay time from controller node to actuator node, τ_k^{ca}, reaches the controller node before the next sample value as discussed in Section 2.1.

2.3 Clock Synchronization

Clock synchronization is a research area in itself. The purpose of clock synchronization is to make the internal clock of two or more nodes to have corresponding values. We will only consider software synchronization, i.e. the synchronization signals are sent over the communication network. Hardware synchronization is also a possibility, for instance, using special wiring just to distribute a global clock signal in the system. As all communication between nodes is over the data network, all messages between nodes will be subject to random delays. Several schemes for synchronization are available in the literature, see Christian and Fetzer (1994),van Oorschot (1993).

Most schemes build on the concept of estimating the difference between the clocks of two nodes by sending clock-requests back and forth as described in the following. Let S be the node which wants to estimate its clock difference to node R. Let the absolute time be t_i, let the local time in node S be t_i^S, and let the local time in node R be t_i^R. The local clocks in S and R have a skew to the absolute time such that

$$t_i^S = t_i + \delta^S \tag{2.5}$$
$$t_i^R = t_i + \delta^R, \tag{2.6}$$

where δ^S and δ^R are the clock mismatches. We define the clock offset, δ, as

$$\delta = \delta^R - \delta^S. \tag{2.7}$$

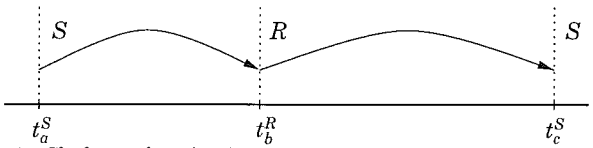

Figure 2.4 Clock synchronization sequence. First a request is sent from node S to node R, then R responds by sending the value of its local clock to S.

From (2.5) and (2.6) it follows that

$$t_i^S = t_i^R - \delta. \tag{2.8}$$

The clock offset will have a drift in time due to inaccuracies in the local clocks. For the moment we assume that δ is constant. The synchronization sequence starts with a clock-read request from node S to node R, this message is sent at time t_a^S, see Figure 2.4. As node R receives the message from node S it immediately sends a message back containing the local clock value t_b^R. This message arrives at node S at time t_c^S. Introduce T_{SR} and T_{RS} as the transfer time for the from S to R, and from R to S respectively. The transfer times can be written as

$$T_{SR} = t_b^S - t_a^S = (t_b^R - \delta) - t_a^S \tag{2.9}$$
$$T_{RS} = t_c^S - t_b^S = t_c^S - (t_b^R - \delta). \tag{2.10}$$

Assuming that $\mathrm{E}(T_{SR} - T_{RS}) = 0$ we find that

$$\delta = \mathrm{E}\left\{\frac{2t_b^R - t_a^S - t_c^S}{2}\right\}, \tag{2.11}$$

where E denotes the expectation operator. By repeating the clock synchronization experiment we can find an accurate estimate of δ by (2.11). There are other clock synchronization algorithms not depending on the assumption $\mathrm{E}(T_{SR} - T_{RS}) = 0$, see van Oorschot (1993). There are also algorithms addressing fault-tolerant synchronization. These faults can be failures in the network, in a local clock etc., see Christian and Fetzer (1994). If the clock offset, δ, is drifting due to inaccuracies in the local clocks, resynchronization must be done after a while. The drift

in a clock is often defined as clock drift ρ. If $H(t)$ is the value of the clock at time t, the following holds for a time interval $[s,t]$:

$$(1-\rho)(t-s) \leq H(t) - H(s) \leq (1+\rho)(t-s). \tag{2.12}$$

It is claimed in Christian and Fetzer (1994) that clocks in modern computers have ρ of the order 10^{-5} or 10^{-6}, and high precision quartz clocks have ρ of the order 10^{-7} or 10^{-8}. With $\rho = 10^{-6}$ we would over one hour have

$$1\text{h} - 3.6\text{ms} \leq H(t+1\text{h}) - H(t) \leq 1\text{h} + 3.6\text{ms}. \tag{2.13}$$

These drift values have to be compared with the time scales in the actual system to determine how often resynchronization must be done to keep an acceptable clock synchronization in the system.

2.4 Related Work

Some work has been done on setups related to the one described in Section 2.1. No work by other authors is know on the setup in Section 2.1. In this section some of the related work is described.

Make the System Time-Invariant

In Luck and Ray (1990) the closed loop system is made time-invariant by introduction of buffers at the controller and actuator nodes as illustrated in Figure 2.5. All nodes are clocked and act synchronized. By making the buffers longer than the worst case delay the process state can be written as

$$x_{k+1} = Ax_k + Bu_{k-\Delta_1} \tag{2.14}$$

$$y_k = Cx_k, \tag{2.15}$$

where Δ_1 is the length in samples of the buffer at the actuator node. If the buffer at the controller node is assumed to have the length Δ_2 samples, the process output available for the controller at time k is $w_k = y_{k-\Delta_2}$. The design problem is now reformulated as a standard

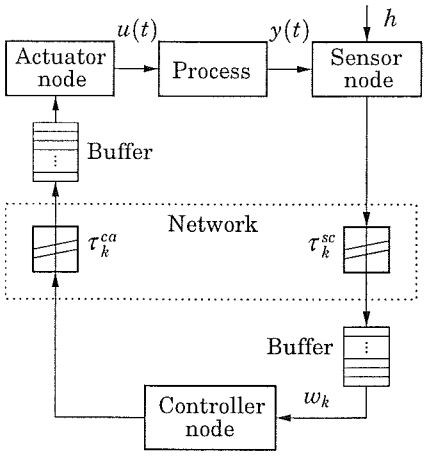

Figure 2.5 In Luck and Ray (1990) buffers are introduced after the varying communication delays to make the system time-invariant. The buffers must be longer than the worst case communication delay.

sampled data control problem. The information set available for calculation of u_k is

$$\mathcal{W}_k = \{w_k, w_{k-1}, \dots\}. \tag{2.16}$$

In Luck and Ray (1990) the LQG-optimal controller,

$$u_k = \xi(\mathcal{W}_k), \tag{2.17}$$

is derived. An advantage with the method is that it handles control delays that are longer than the sampling period. As will be shown in Chapter 5 performance can be increased by having event-driven controller and actuator node, which makes the control delay smaller.

Stochastic Approaches

In Liou and Ray (1991) a scheme with time-driven sensor, time-driven controller, and event-driven actuator, is studied. The sensor and the controller is started with a time skew of Δ_s. The probability that the

new sensor value has reached the controller when the control signal is calculated, $P(\tau_k^{sc} < \Delta_s)$, is known. If $\tau_k^{sc} > \Delta_s$ the new control signal is calculated without knowledge of the new measurement signal. The actuator node D/A-converts the new control signal as soon as it is transmitted to the actuator node. A discrete time *augmented plant model* is derived by introducing the delayed signals as states in the augmented plant model. The *augmented plant model* is

$$x_{k+1} = A_k x_k + B_k u_k, \tag{2.18}$$

where A_k and B_k are stochastic matrices due to the random communication delays. The LQ-optimal controller is solved for the stated problem setup. It is also discussed how to construct a state estimator in the case when all states are not measured. It is claimed that time-stamping of signals is important for estimation of process state. For the problem setup in Liou and Ray (1991) it is not known if the combination of optimal controller and the proposed state estimator is the optimal output feedback controller, i.e. if the *separation principle* applies.

The LQ-controller of Liou and Ray (1991) is used in Ray (1994) together with a stochastic state estimator. The timing setup is the same as in Liou and Ray (1991). The estimator is designed to minimize the variance of the state prediction errors. The combination of the LQ-controller and the minimum variance estimator is introduced as the *DCLQG*-controller, delay compensated LQG. It is stressed that the separation principle does not hold for the *DCLQG*-controller, i.e. *DCLQG*-controller is a suboptimal control scheme.

In Krtolica *et al.* (1994) control systems with random communication delays are studied. The delays, from sensor to controller and from controller to actuator, are modeled as being generated from a Markov chain, see Figure 2.6. Only one of the β_i coefficients in Figure 2.6 are one, all the others are zero. Notice that the delay must be a multiple of the sampling period, i.e. all nodes are clock driven. It is shown that the closed loop system can be written as

$$z_{k+1} = H_k z_k, \tag{2.19}$$

where H_k depends on the state of the delay elements depicted in Figure 2.6. The sequence of β_i is generated by a Markov chain. Necessary

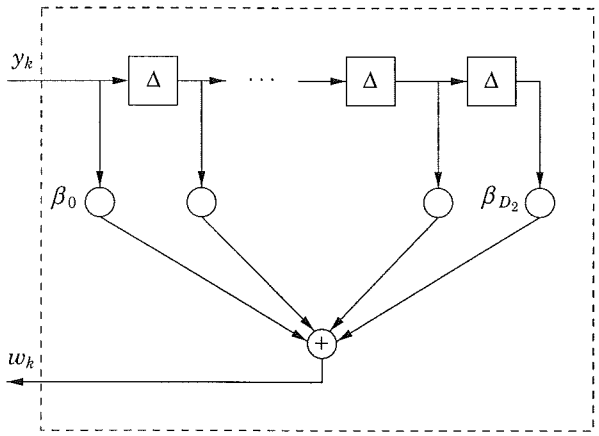

Figure 2.6 The network model used in Krtolica *et al.* (1994). The sampled signal, y_k, is delayed a number of samples due to communication delay. The controller reads the signal w_k. Notice that only one of the β_i coefficients is one, the others are zero.

and sufficient conditions are found for zero-state mean-square exponential stability. The stability criterion is that the solution of two coupled (Lyapunov-like) equations need to be positive definite for stability of the closed loop.

In Chan and Özgüner (1995) a setup using the *Ford SCP Multiplex Network hardware* is studied. The communication delay is modeled as in Figure 2.7. There is a queue of unsent sensor readings at the sensor node. A simple form of time-stamping is done by appending the size of the queue to every message that is sent from the sensor node to the controller node. It is shown that by this method the controller node can reduce its uncertainty about which sensor reading it is using down to two possible. By knowing the probability for the two possible cases of delay, a state estimator is constructed. The sensor node and the controller node are both time-driven with a skew of Δ_{sp}. It is also shown how pole placement can be done for the described setup.

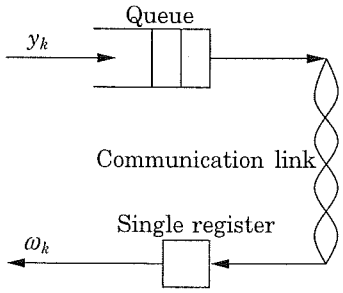

Figure 2.7 Block diagram of the transmission from the sensor node to the controller node in Chan and Özgüner (1995). The sampled signal y_k is delayed during the transmission to the controller node. The controller reads the sensor value ω_k from a register in the controller node. A simple form of time-stamping is done by appended every message with the size of the queue when the message was sent.

Jump Linear Systems

Jump linear systems can in discrete time be written as

$$x_{k+1} = A(r_k)x_k + B(r_k)u_k, \tag{2.20}$$

where $A(r_k)$ and $B(r_k)$ are real-valued matrix functions of the random process $\{r_k\}$, see Ji *et al.* (1991). An interesting special case of $\{r_k\}$-process is to let $\{r_k\}$ be a time homogeneous Markov chain taking values in a finite set $\{1,\ldots,s\}$. The Markov chain has the transition probabilities

$$P(r_{k+1} = j \mid r_k = i) = p_{ij} \geq 0, \tag{2.21}$$

where

$$\sum_{j=1}^{s} p_{ij} = 1. \tag{2.22}$$

If $\{r_k\}$ is generated by a time homogeneous Markov chain the system is called a *discrete-time Markovian jump linear system*. As will be discussed in Chapter 3 this is an attractive model for control systems with induced network delays.

Jump systems in continuous time was introduced in the 60's by Krasovskii and Lidskii (1961). The LQ-problem was solved with finite and infinite time horizon by Sworder (1969) and Wonham (1971).

The discrete-time jump LQ-problem was solved for a finite-horizon by Blair and Sworder (1975). Extensions of this problem, such as control of jump linear systems with Gaussian input and measurement noise, has been done in Ji and Chizeck (1990). Different notations of stability of jump linear systems were compared in Ji *et al.* (1991).

All the above mentioned work rely on the availability of the value r_k at time k. If this is not the case, r_k has to be estimated. Less work has been done in this area, see Ji *et al.* (1991) for a discussion.

An interesting extension of jump linear systems is to let the Markov chain postulate the distribution of the system matrices A, B etc. instead of values for these. As an example the states of the Markov chain could be "Low network load", "Medium network load" and "High network load". This is further discussed in Chapter 3.

3

Modeling of Network Delays

Network delays, or network transfer times, have different characteristics depending on the network hardware and software. To analyze control systems with network delays in the loop we have to model these. The network delay is typically varying due to varying network load, scheduling policies in the network and the nodes, and due to network failures. We will use three models of the network delay:

- Constant delay
- Random delay, which is independent from transfer to transfer
- Random delay, with probability distributions governed by an underlying Markov chain

The control loop usually also contains computational delays. The effect of these can be embedded in the network delays.

3.1 Network Modeled as Constant Delay

The simplest model of the network delay is to model it as being constant for all transfers in the communication network. This can be a good model even if the network has varying delays, for instance, if the time scale in the process is much larger than the delay introduced by the communication. In this case the mean value or maybe the worst case delay can be used in the analysis. If this is not the case wrong conclusions can be drawn regarding system stability and performance.

One way to achieve constant delays is by introduction of timed buffers after each transfer. By making this buffer longer than the worst case delay time the transfer time can be viewed as being constant. This method was proposed in Luck and Ray (1990). A drawback with this method is that the control delay becomes longer than necessary. This can lead to decreased performance as shown in Chapter 5.

3.2 Network Modeled as Consecutive Delays Being Independent

To take the randomness of the network delays into account in the model, the time delays can be modeled as being taken from a probabilistic distribution. To keep the model simple to analyze one can assume the transfer delay to be independent of previous delay times. In a real communication system the transfer time will however usually be correlated with the last transfer delay. For example the network load, which is one of the factors affecting the delay, is typically varying with a slower time constant than the sampling period in a control system, i.e. the time between two transfers.

3.3 Network Modeled Using Markov Chain

One way to model dependence between samples is by letting the distribution of the network delays be governed by the state of an underlying Markov chain. Effects such as varying network load can be modeled by making the Markov chain do a transition every time a transfer is done in the communication network. A short discussion of the theory of Markov chains is given in Appendix B.

EXAMPLE 3.1—SIMPLE NETWORK MODEL
To get a simple network model we can let the network have three states, one for low network load, one for medium network load, and one for high network load. In Figure 3.1 the transitions between different states in the communication network are modeled with a Markov chain. Together with every state in the Markov chain we have a corre-

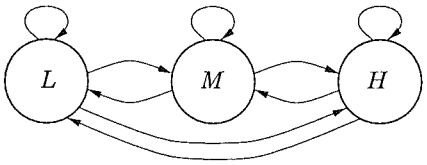

Figure 3.1 An example of a Markov chain modeling the state in a communication network. L is the state for low network load, M the state for medium network load, and H is the state for high network load. The arrows shows possible transitions in the system.

sponding delay distribution modeling the delay for that network state. These distributions could typically look like the probabilistic distributions in Figure 3.2. □

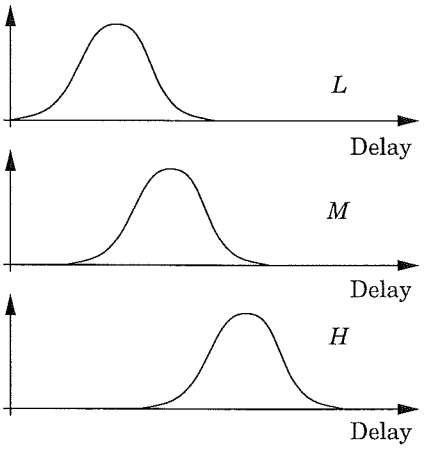

Figure 3.2 The delay distributions corresponding to the states of the Markov chain in Figure 3.1. L is the state for low network load, M the state for medium network load, and H is the state for high network load.

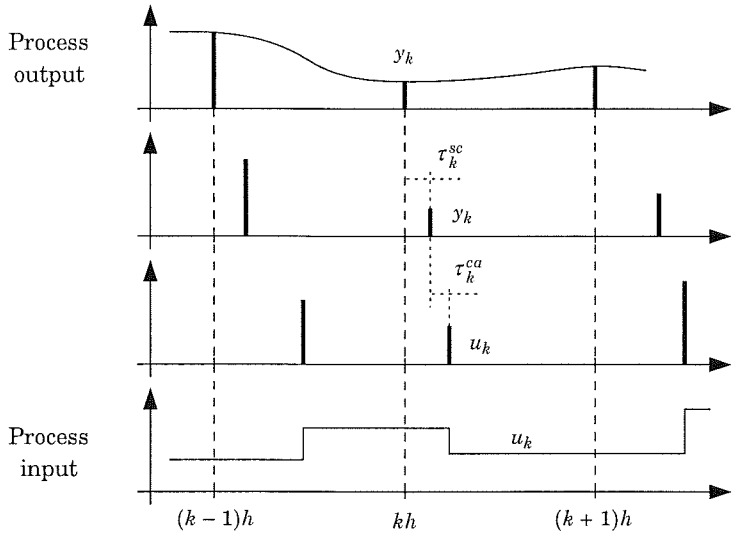

Figure 3.3 Timing of signals in the control system. The first diagram illustrates the process output and the sampling instants, the second diagram illustrates the signal into the controller node, the third diagram illustrates the signal into the actuator node, and the fourth diagram illustrates the process input.

3.4 Sampling of Systems with Network Delays

In continuous-time time-delays are infinite dimensional systems. A finite dimensional description of the control loop can be formulated by sampling of the continuous-time process. Let the controlled process be

$$\frac{dx}{dt} = Ax(t) + Bu(t) + v(t), \tag{3.1}$$

where $x(t) \in \mathcal{R}^n$, $u(t) \in \mathcal{R}^m$ and $v(t) \in \mathcal{R}^n$. A and B are matrices of appropriate sizes. $u(t)$ is the controlled input and $v(t)$ is white noise with zero mean and incremental covariance R_v. The timing of the signals in the system is shown in Figure 3.3. Notice that the control signal segments are active a varying time. Assume that the delay from sensor to actuator is less than the sampling period h, i.e. $\tau_k^{sc} + \tau_k^{ca} < h$.

Integration of (3.1) over a sampling interval gives

$$x_{k+1} = \Phi x_k + \Gamma_0(\tau_k^{sc}, \tau_k^{ca})u_k + \Gamma_1(\tau_k^{sc}, \tau_k^{ca})u_{k-1} + v_k, \tag{3.2}$$

where

$$\Phi = e^{Ah} \tag{3.3}$$

$$\Gamma_0(\tau_k^{sc}, \tau_k^{ca}) = \int_0^{h-\tau_k^{sc}-\tau_k^{ca}} e^{As} ds B \tag{3.4}$$

$$\Gamma_1(\tau_k^{sc}, \tau_k^{ca}) = \int_{h-\tau_k^{sc}-\tau_k^{ca}}^{h} e^{As} ds B. \tag{3.5}$$

The state noise v_k has zero mean and the variance

$$R_1 = \mathrm{E}\{v_k v_k^T\} = \int_0^h e^{A(h-s)} R_v e^{A^T(h-s)} ds. \tag{3.6}$$

This is a standard result on sampling of systems with time-delays, see, for instance, Åström and Wittenmark (1990). The infinite dimensional continuous-time system has now been reformulated to the time-varying, finite-dimensional, discrete-time system (3.2). The drawback is that we do not have direct control over intersample behavior using the discrete-time model. This is however easy to study if needed. Some early work on modeling of imperfections in sampled data systems, such as random sampling and imperfect hold etc., was made in Kalman and Bertram (1959).

4

Analysis of Control Laws

In order to design a controller for a distributed digital control system it is important to know how to analyze such systems. We will assume that the sensor node is sampled regularly at a constant sampling period h. The actuator node is assumed to be event driven, i.e. the control signal will be used as soon as it arrives. We will analyze the different models for the communication network described in Chapter 3. In Figure 4.1 the control system is illustrated in a block diagram. In this chapter we will analyze given linear control laws. Optimal control laws are calculated in Chapter 5. The controlled process is assumed to be

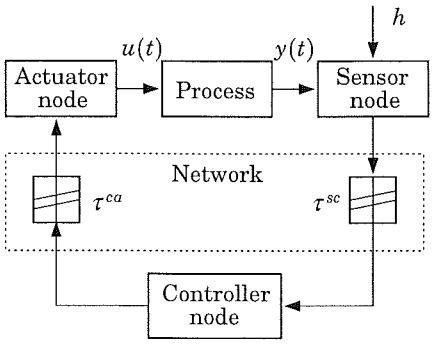

Figure 4.1 Distributed digital control system with induced delays, τ^{sc} and τ^{ca}.

$$\frac{dx}{dt} = Ax(t) + Bu(t) + v(t), \tag{4.1}$$

where $x(t) \in \mathcal{R}^n$, $u(t) \in \mathcal{R}^m$ and $v(t) \in \mathcal{R}^n$. A and B are matrices of appropriate sizes. $u(t)$ is the controlled input and $v(t)$ is white noise with zero mean and covariance R_v. We will assume that the delay from sensor to actuator is less than the sampling period h, i.e. $\tau_k^{sc} + \tau_k^{ca} < h$. If this condition is not satisfied control signals may arrive at the actuator in corrupted order, which makes the analysis much harder. The influence from the network is collected in the variable τ_k. For instance τ_k can be a vector with the delays in the loop, i.e. $\tau_k = [\tau_k^{sc}, \tau_k^{ca}]^T$. Discretizing (4.1) in the sampling instants, see Chapter 3, gives

$$x_{k+1} = \Phi^p x_k + \Gamma_0^p(\tau_k)u_k + \Gamma_1^p(\tau_k)u_{k-1} + v_k. \tag{4.2}$$

The output equation is

$$y_k = C^p x_k + w_k, \tag{4.3}$$

where $y_k \in \mathcal{R}^p$. The stochastic processes v_k and w_k are uncorrelated white noise with zero mean and covariance matrices R_1 and R_2 respectively.
A linear controller for this system can be written as

$$x_{k+1}^c = \Phi^c(\tau_k)x_k^c + \Gamma^c(\tau_k)y_k \tag{4.4}$$

$$u_k = C^c(\tau_k)x_k^c + D^c(\tau_k)y_k, \tag{4.5}$$

where appearance of τ_k in Φ^c, Γ^c, C^c or D^c, means that the controller knows the network delays completely or partly. Examples of such controllers are given in Krtolica *et al.* (1994), Ray (1994), and Nilsson *et al.* (1996).

From (4.2) – (4.5) we see that the closed loop system can be written as

$$z_{k+1} = \Phi(\tau_k)z_k + \Gamma(\tau_k)e_k, \tag{4.6}$$

where

$$z_k = \begin{bmatrix} x_k \\ x_k^c \\ u_{k-1} \end{bmatrix}, \tag{4.7}$$

$$\Phi(\tau_k) = \begin{bmatrix} \Phi^p + \Gamma_0^p(\tau_k)D^c(\tau_k)C^p & \Gamma_0^p(\tau_k)C^c(\tau_k) & \Gamma_1^p(\tau_k) \\ \Gamma^c(\tau_k)C^p & \Phi^c(\tau_k) & 0 \\ D^c(\tau_k)C^p & C^c(\tau_k) & 0 \end{bmatrix}, \tag{4.8}$$

$$e_k = \begin{bmatrix} v_k \\ w_k \end{bmatrix}, \tag{4.9}$$

and

$$\Gamma(\tau_k) = \begin{bmatrix} I & \Gamma_0^p(\tau_k)D^c(\tau_k) \\ 0 & \Gamma^c(\tau_k) \\ 0 & D^c(\tau_k) \end{bmatrix}. \tag{4.10}$$

The variance R of e_k is

$$R = \mathrm{E}(e_k e_k^T) = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}.$$

The rest of this chapter investigates properties of the closed loop system (4.6). The analysis is made for the network models described in Chapter 3. Section 4.1 studies systems with network transfers modeled as constant delays. Section 4.2 analyses loops with delays modeled as being independent from transfer to transfer, and Section 4.3 transfer delays modeled with an underlying Markov chain. Section 4.4 describes how to analyze these systems using computer simulations.

4.1 Network Modeled as Constant Delay

The simplest model of the communication delay in a data network is to model it as being constant for all transfers, see Chapter 3. If we

make the assumption that τ_k in (4.6) is constant for all k, the closed loop system can be written as

$$z_{k+1} = \Phi z_k + \Gamma e_k, \tag{4.11}$$

where Φ and Γ are constant matrices. We can then use the standard tools from the theory of linear time-invariant discrete time systems to analyze stability, variances of signals etc., see Åström and Wittenmark (1990).

The control scheme with buffers, see Chapter 2, makes the closed loop system time invariant by introducing buffers at the controller and actuator node. By adding these buffers as states in the description of the closed loop system the system can be written on the form (4.11). This analysis is also straightforward to do in the case when the network delay is longer than the sampling period, i.e. when $\tau^{ca} + \tau^{sc} > h$.

4.2 Network Modeled as Consecutive Delays Being Independent

As described in Chapter 3, communication delays in a data network usually vary from transfer to transfer. In this situation the standard methods from linear time-invariant discrete time systems cannot be applied. There are examples where the closed loop system is stable for all constant delays, but give instability when the delay is varying. This section develops some analysis tools for systems where consecutive delays are random and independent. First it is investigated how to calculate covariances of signals generated by (4.6), this naturally leads to a stability criterion for systems with random and independent delays.

Evaluation of Covariance

Let the closed loop system be given by (4.6), where $\{\tau_k\}$ is a random process uncorrelated with $\{e_k\}$. The form of $\Phi(\tau_k)$ and $\Gamma(\tau_k)$ is determined by the process, the communication network, and the controller structure. τ_k can be a vector consisting of the delay from sensor to controller, τ_k^{sc}, and the delay from controller to actuator, τ_k^{ca}. We assume that τ_k has known distribution, and that τ_k is independent from

sample to sample. This can be an unrealistic assumption and will be discussed in Section 4.3. To keep track of the noise processes we collect the random components up to time k in

$$\mathcal{Y}_k = \{\tau_0, ..., \tau_k, e_0, ..., e_k\}.$$

Introduce the state covariance P_k as

$$P_k = \underset{\mathcal{Y}_{k-1}}{\mathrm{E}}\,(z_k z_k^T), \tag{4.12}$$

where the expectation is calculated with respect to noise in the process and randomness in the communication delays. By iterating (4.12) we get

$$\begin{aligned} P_{k+1} &= \underset{\mathcal{Y}_k}{\mathrm{E}}(z_{k+1} z_{k+1}^T) \\ &= \underset{\mathcal{Y}_k}{\mathrm{E}}\left((\Phi(\tau_k) z_k + \Gamma(\tau_k) e_k)(\Phi(\tau_k) z_k + \Gamma(\tau_k) e_k)^T\right) \\ &= \underset{\mathcal{Y}_k}{\mathrm{E}}\left((\Phi(\tau_k) z_k z_k^T \Phi(\tau_k)^T + \Gamma(\tau_k) e_k e_k^T \Gamma(\tau_k)^T\right) \\ &= \underset{\tau_k}{\mathrm{E}}\left((\Phi(\tau_k) P_k \Phi(\tau_k)^T + \Gamma(\tau_k) R \Gamma(\tau_k)^T\right). \end{aligned}$$

Here we have used that τ_k, z_k and e_k are independent, and that e_k has mean zero. This is crucial for the applied technique to work and indirectly requires that τ_k and τ_{k-1} are independent. Using Kronecker products, see Appendix A, this can be written as

$$\begin{aligned} \mathrm{vec}(P_{k+1}) &= \underset{\tau_k}{\mathrm{E}}(\Phi(\tau_k) \otimes \Phi(\tau_k))\,\mathrm{vec}(P_k) + \mathrm{vec}\,\underset{\tau_k}{\mathrm{E}}(\Gamma(\tau_k) R \Gamma(\tau_k)^T) \\ &= \mathcal{A}\,\mathrm{vec}(P_k) + \mathcal{G}, \end{aligned} \tag{4.13}$$

where

$$\mathcal{A} = \underset{\tau_k}{\mathrm{E}}(\Phi(\tau_k) \otimes \Phi(\tau_k)) \qquad \mathcal{G} = \underset{\tau_k}{\mathrm{E}}(\Gamma(\tau_k) \otimes \Gamma(\tau_k))\,\mathrm{vec}(R).$$

From (4.13) we see that stability in the sense of $\mathrm{E}(z_k^T z_k) < \infty$, i.e. second moment stability, is guaranteed if $\rho(\mathrm{E}(\Phi(\tau_k) \otimes \Phi(\tau_k))) < 1$, where $\rho(A)$ denotes the spectral radius of the matrix A. This stability condition appeared in Kalman (1962) in a slightly different setting. For a discussion of the connection between second moment stability and other stability concepts such as mean square stability, stochastic stability and exponential mean square stability see Ji *et al.* (1991).

Calculation of Stationary Covariance

If the recursion (4.13) is stable, $\rho(\mathrm{E}(\Phi(\tau_k) \otimes \Phi(\tau_k))) < 1$, the stationary covariance

$$P^{\infty} = \lim_{k \to \infty} P_k \tag{4.14}$$

can be found from the unique solution of the linear equation

$$\mathrm{vec}(P^{\infty}) = \mathrm{E}(\Phi(\tau_k) \otimes \Phi(\tau_k))\,\mathrm{vec}(P^{\infty}) + \mathrm{vec}\,\mathrm{E}(\Gamma(\tau_k) R \Gamma(\tau_k)^T). \tag{4.15}$$

Calculation of Quadratic Cost Function

In LQG-control it is of importance to evaluate quadratic cost functions like $\mathrm{E}\, z_k^T S(\tau_k) z_k$. This can be done as

$$\mathop{\mathrm{E}}_{\mathcal{Y}_k} z_k^T S(\tau_k) z_k = \mathrm{tr} \mathop{\mathrm{E}}_{\mathcal{Y}_k} z_k^T S(\tau_k) z_k = \mathrm{tr}(\mathop{\mathrm{E}}_{\tau_k} S(\tau_k) \mathop{\mathrm{E}}_{\mathcal{Y}_{k-1}} z_k z_k^T), \tag{4.16}$$

which as $k \to \infty$ gives

$$\lim_{k \to \infty} \mathrm{E}\, z_k^T S(\tau_k) z_k = \mathrm{tr}(\mathop{\mathrm{E}}_{\tau_k} S(\tau_k) P^{\infty}). \tag{4.17}$$

This quantity can now be calculated using the previous result.

Normally we want to calculate a cost function of the form $\mathrm{E}(x_k^T S_{11} x_k + u_k^T S_{22} u_k)$. As u_k is not an element of the vector z_k, see (4.7), this cost function can not always directly be cast into the formalism of (4.16). A solution to this problem is to rewrite u_k of (4.5) using the output equation (4.3) as

$$\begin{aligned} u_k &= C^c(\tau_k) x_k^c + D^c(\tau_k)(C^p x_k + w_k) \\ &= \begin{bmatrix} D^c(\tau_k) C^p & C^c(\tau_k) & 0 \end{bmatrix} z_k + D^c(\tau_k) w_k. \end{aligned}$$

Noting that τ_k and w_k are independent, and that w_k has zero mean, the cost function can be written as

$$\mathrm{E}(x_k^T S_{11} x_k + u_k^T S_{22} u_k) = \mathrm{E}(z_k^T S(\tau_k) z_k) + J_1,$$

where

$$S(\tau_k) = \begin{bmatrix} S_{11} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} (D^c(\tau_k)C^p)^T \\ C^c(\tau_k)^T \\ 0 \end{bmatrix} S_{22} \begin{bmatrix} D^c(\tau_k)C^p & C^c(\tau_k) & 0 \end{bmatrix}$$

$$J_1 = \operatorname{tr}\left(\mathrm{E}\{D^c(\tau_k)^T S_{22} D^c(\tau_k)\} R_2\right),$$

where the first part again is on the form of (4.16).

EXAMPLE 4.1—CONSTANT VS RANDOM DELAY
Consider the control system in Figure 4.2 with one delay τ_k in the loop. The process output is sampled with the sampling period h. We will consider two cases for the delay.

- The delay is constant with the value $\tau_k = h/2$.
- The delay is uniformly distributed on the interval $[0, h]$.

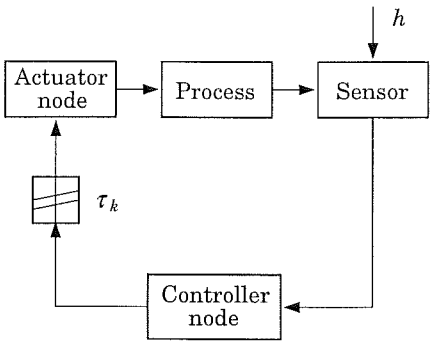

Figure 4.2 Digital control system with induced delay.

Let the process be, this can for instance be an inverted pendulum,

$$Y(s) = \frac{1}{s^2 - 1} U(s).$$

Discretizing this in the sampling instants and assuming that white noise v_k affects the state gives

$$x_{k+1} = \Phi^p x_k + \Gamma_0^p(\tau_k) u_k + \Gamma_1^p(\tau_k) u_{k-1} + \sqrt{h} I v_k, \tag{4.18}$$

where v_k is white noise with variance I and zero mean. The $\sqrt{h}$-factor comes from sampling of continuous time white noise with the sampling period h. We will control the process with an LQ-controller which minimizes

$$J = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} (y_k^2 + u_k^2).$$

We let the control design take the nominal delay into account by designing the controller for (4.18) with $\tau_k = h/2$. This gives a control law

$$u_k = -L \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix},$$

where we partition L as $L = [l_x, l_u]$. Assuming that we can measure the process state the closed loop system can be written as (4.7), where

$$\Phi = \begin{bmatrix} \Phi^p - \Gamma_0^p(\tau_k) l_x & \Gamma_1^p(\tau_k) - \Gamma_0^p(\tau_k) l_u \\ -l_x & -l_u \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} I \\ 0 \end{bmatrix},$$

and $z_k = [\, x_k \quad u_{k-1} \,]^T$.

In Figure 4.3 the value of the cost function J is plotted for the two models of the delay for $h \in [0, 1]$. It is seen that the controller is most successful in minimizing J when the delay is constant instead of being spread over an interval. This is not surprising since the controller was designed assuming a constant delay. From calculations using (4.13) it is seen that the controller fails to stabilize the process for $h > 0.785$ if the time delays are randomly varying. □

4.3 Network Modeled Using Markov Chain

As described in Chapter 3 a more realistic model for communication delays in data networks is to model the delays as being random with the distribution selected from an underlying Markov chain. In this section some analysis tools for these systems are developed. First variances of

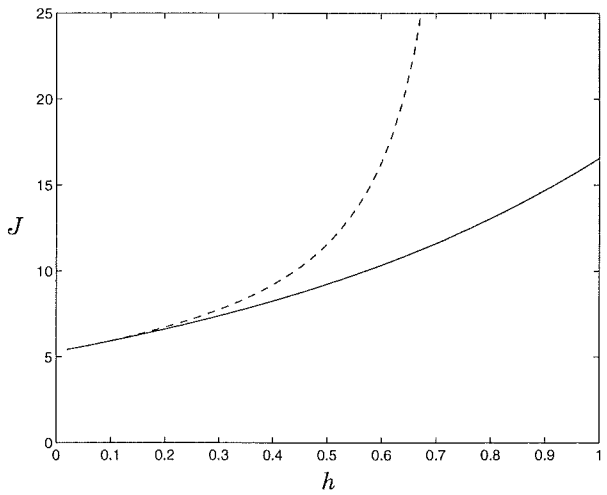

Figure 4.3 Values of the cost function J for constant delay (full line) and for uniformly distributed delay (dashed line) vs sampling period.

signals and stability of the closed loop are studied for a system with a Markov chain which makes one transition every sample. These results are then generalized to the case when the Markov chain makes two transitions every sample, this to allow for the state of the Markov chain to change both when sending measurement and control signals.

Evaluation of Covariance

Let the closed loop system be described by (4.6), where τ_k is a random variable with probability distribution given by the state of a Markov chain. The Markov chain has the state $r_k \in \{1, ..., s\}$ when τ_k is generated. The Markov chain then makes a transition between k and $k+1$. The transition matrix for the Markov chain is $Q = \{q_{ij}\}, i,j \in \{1, ..., s\}$, where

$$q_{ij} = P(r_{k+1} = j \mid r_k = i).$$

The Markov chain is assumed to be stationary and regular, see Ap-

pendix B. Introduce the Markov state probability

$$\pi_i(k) = P(r_k = i), \tag{4.19}$$

and the Markov state distribution

$$\pi(k) = [\,\pi_1(k) \quad \pi_2(k) \quad \dots \quad \pi_s(k)\,].$$

The probability distribution for r_k is given by the recursion

$$\begin{aligned} \pi(k+1) &= \pi(k)Q \\ \pi(0) &= \pi^0, \end{aligned}$$

where π^0 is the probability distribution for r_0. The state noise e_k is assumed to be white with unit variance. The random components up to time k are collected in

$$\mathcal{Y}_k = \{e_0, \dots, e_k, \tau_0, \dots, \tau_k, r_0, \dots, r_k\}.$$

Introduce the conditional state covariance as

$$P_i(k) = \mathop{\mathrm{E}}_{\mathcal{Y}_{k-1}} \left(z_k z_k^T \mid r_k = i\right),$$

and

$$\widetilde{P}_i(k) = \pi_i(k) P_i(k).$$

The following relationship now holds for the state covariance $P(k)$:

$$P(k) = \sum_{i=1}^{s} \pi_i(k) P_i(k) = \sum_{i=1}^{s} \widetilde{P}_i(k). \tag{4.20}$$

In Section 4.2 we obtained a recursive equation for $P(k)$. In this section we will instead obtain a recursion for $\widetilde{P}_i(k)$. The following theorem gives an algorithm to evaluate $\widetilde{P}_i(k)$. See Appendix A for the Kronecker and vec notation.

THEOREM 4.1
The vectorized state covariance matrix $\widetilde{\mathbf{P}}(k)$ satisfies the recursion

$$\widetilde{\mathbf{P}}(k+1) = (Q^T \otimes I)\,\mathrm{diag}(\mathcal{A}_i)\widetilde{\mathbf{P}}(k) + (Q^T \otimes I)(\mathrm{diag}(\pi_i(k)) \otimes I)\mathbf{G}. \tag{4.21}$$

where

$$\mathcal{A}_i = \mathop{\mathrm{E}}_{\tau_k}(\Phi(\tau_k) \otimes \Phi(\tau_k) \mid r_k = i) \qquad \mathcal{G}_i = \mathop{\mathrm{E}}_{\tau_k}(\Gamma(\tau_k) R \Gamma^T(\tau_k) \mid r_k = i)$$

$$\widetilde{\mathbf{P}}(k) = \begin{bmatrix} \mathrm{vec}\,\widetilde{P}_1(k) \\ \mathrm{vec}\,\widetilde{P}_2(k) \\ \vdots \\ \mathrm{vec}\,\widetilde{P}_s(k) \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} \mathrm{vec}\,\mathcal{G}_1 \\ \mathrm{vec}\,\mathcal{G}_2 \\ \vdots \\ \mathrm{vec}\,\mathcal{G}_s \end{bmatrix}.$$

□

The proof of Theorem 4.1 is given in the following section.

From (4.21) it is seen that the closed loop will be stable, in the sense that the covariance is finite, if the matrix $(Q^T \otimes I)\,\mathrm{diag}(\mathcal{A}_i)$ has all its eigenvalues in the unit circle.

Proof of Theorem 4.1

We will need the following property of τ_k and z_k.

LEMMA 4.1—CONDITIONAL INDEPENDENCE LEMMA
The random variables τ_k and z_k are conditionally independent relative r_k, i.e.

$$\mathop{\mathrm{E}}_{\mathcal{Y}_k}(f(\tau_k)g(z_k) \mid r_k = i) = \mathop{\mathrm{E}}_{\mathcal{Y}_k}(f(\tau_k) \mid r_k = i)\mathop{\mathrm{E}}_{\mathcal{Y}_k}(g(z_k) \mid r_k = i),$$

for all measurable functions $f(.)$ and $g(.)$. □

Proof The proof is based on Chapter 9.1 of Chung (1974), especially Theorem 9.2.1. A short summary of the used results are given

in Appendix B. Conditional independence follows by Theorem 9.2.1 of Chung (1974) and from the fact that

$$P(\tau_k \in E \mid z_k, r_k = i) = P(\tau_k \in E \mid r_k = i).$$

This is a consequence of (4.6) and the fact that z_k is measurable with respect to $\mathcal{Y}_k$. □

Remark: z_k and τ_k are not independent.

We will also need the following Markov property.

LEMMA 4.2
Under the stated Markov assumptions it holds that

$$\underset{\mathcal{Y}_k}{\mathrm{E}}(f(z_k, \tau_k) \mid r_{k+1} = j, r_k = i) = \underset{\mathcal{Y}_k}{\mathrm{E}}(f(z_k, \tau_k) \mid r_k = i),$$

where $f(.)$ is a measurable function. □

Proof From Theorem 9.2.1 of Chung (1974), see Appendix B, it follows that an equivalent condition is

$$P(r_{k+1} = j \mid f(z_k, \tau_k), r_k = i) = P(r_{k+1} = j \mid r_k = i).$$

The validity of this condition follows directly from the Markov property. □

Introduce $\tilde{q}_{ij}(k)$, the state transition probability for the reversed Markov chain, as

$$\tilde{q}_{ij}(k) = P(r_{k-1} = i \mid r_k = j). \tag{4.22}$$

From (4.19) and (4.22) it follows that

$$\tilde{q}_{ij}(k) = \frac{q_{ij}\pi_i(k-1)}{\sum_{l=1}^{s} q_{lj}\pi_l(k-1)} \tag{4.23}$$

and

$$\pi_j(k) = \sum_{l=1}^{s} q_{lj}\pi_l(k-1). \tag{4.24}$$

Combining (4.23) and (4.24) we get

$$\widetilde{q}_{ij}(k)\pi_j(k) = q_{ij}\pi_i(k-1)$$

The iteration of $\widetilde{P}$ now becomes

$$\begin{aligned}
\widetilde{P}_j(k+1) &= \pi_j(k+1)P_j(k+1) = \pi_j(k+1)\mathop{\mathrm{E}}_{\mathcal{Y}_k}\left(z_{k+1}z_{k+1}^T \mid r_{k+1}=j\right) \\
&= \pi_j(k+1)\mathop{\mathrm{E}}_{\mathcal{Y}_k}\left(\Phi(\tau_k)z_k z_k^T\Phi^T(\tau_k)\right. \\
&\qquad \left. + \Gamma(\tau_k)e_k e_k^T\Gamma^T(\tau_k) \mid r_{k+1}=j\right) \\
&= \pi_j(k+1)\sum_{i=1}^{s}\widetilde{q}_{ij}(k+1)\mathop{\mathrm{E}}_{\mathcal{Y}_k}\left(\Phi(\tau_k)z_k z_k^T\Phi^T(\tau_k)\right. \\
&\qquad \left. + \Gamma(\tau_k)e_k e_k^T\Gamma^T(\tau_k) \mid r_k=i, r_{k+1}=j\right) \\
&= \sum_{i=1}^{s} q_{ij}\pi_i(k)\mathop{\mathrm{E}}_{\mathcal{Y}_k}\left(\Phi(\tau_k)z_k z_k^T\Phi^T(\tau_k)\right. \\
&\qquad \left. + \Gamma(\tau_k)e_k e_k^T\Gamma^T(\tau_k) \mid r_k=i\right).
\end{aligned}$$

The fourth equality follows from that $P(A|C) = \sum_B P(A|B,C)P(B|C)$. In the last equality we have used the Markov property of Lemma 4.2. By vectorizing $\widetilde{P}_j(k+1)$ and using the Conditional independence lemma, Lemma 4.1, we get

$$\operatorname{vec}\widetilde{P}_j(k+1) = \sum_{i=1}^{s} q_{ij}\mathcal{A}_i \operatorname{vec}\widetilde{P}_i(k) + \sum_{i=1}^{s} q_{ij}\pi_i(k)\operatorname{vec}\mathcal{G}_i,$$

where $\mathcal{A}_i$ and $\mathcal{G}_i$ are as stated in the theorem. Rewriting the sums as matrix multiplications and using Kronecker products, see Appendix A, the recursion can be written as the linear recursion

$$\widetilde{\mathbf{P}}(k+1) = (Q^T \otimes I)\operatorname{diag}(\mathcal{A}_i)\widetilde{\mathbf{P}}(k) + (Q^T \otimes I)(\operatorname{diag}(\pi_i(k)) \otimes I)\mathbf{G}, \tag{4.25}$$

which completes the proof. □

This result generalize the results in Ji *et al.* (1991) and Gajic and Qureshi (1995) in the sense that we let the Markov chain postulate

the distribution of $\Phi(\tau_k)$ and $\Gamma(\tau_k)$, while Ji *et al.* (1991) and Gajic and Qureshi (1995) let the Markov chain postulate a deterministic $\Phi(\tau_k)$ and $\Gamma(\tau_k)$ for every Markov state. The results in Gajic and Qureshi (1995) are for the continuous time case.

Calculation of Stationary Covariance

In the stable case the recursion (4.21) will converge as $k \to \infty$,

$$\widetilde{\mathbf{P}}^{\infty} = \lim_{k\to\infty} \widetilde{\mathbf{P}}(k).$$

As the Markov chain is regular the stationary distribution π^{∞} of the Markov chain is given uniquely by

$$\pi^{\infty} Q = \pi^{\infty}.$$

Since (4.21) is a linear difference equation it follows that $\widetilde{\mathbf{P}}^{\infty}$ will be the unique solution of the linear equation

$$\widetilde{\mathbf{P}}^{\infty} = (Q^T \otimes I)\,\mathrm{diag}(\mathcal{A}_i)\widetilde{\mathbf{P}}^{\infty} + (Q^T \otimes I)(\mathrm{diag}(\pi_i^{\infty}) \otimes I)\mathbf{G}.$$

It then follows that the stationary value of $\mathrm{E}\, z_k z_k^T$ is given by

$$P^{\infty} = \lim_{k\to\infty} \mathrm{E}(z_k z_k^T) = \lim_{k\to\infty} \sum_{i=1}^{s} \mathrm{E}(z_k z_k^T \mid r_k = i) P(r_k = i) = \sum_{i=1}^{s} \widetilde{P}_i^{\infty},$$

where $\widetilde{P}_i^{\infty}$ is the corresponding part of $\widetilde{\mathbf{P}}^{\infty}$.

Two Transitions Every Sample

To model the situation in the communication network more closely we now let the Markov chain make two transitions between successive samples, one transition for each message being sent in the communication network. Let the Markov chain be divided into two groups of states, one group R^{sc} which generates the probability distributions for delays from sensor to controller, and one group R^{ca} of states that generates the probability distributions for delays from controller to actuator. The transitions between states are then such that the state of

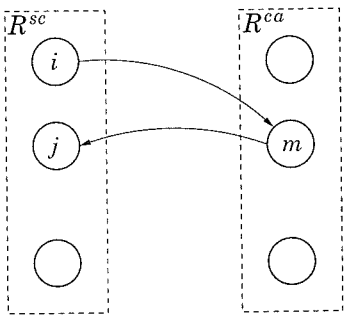

Figure 4.4 Transitions between the two sets of states. First a transition is made from state $i \in R^{sc}$ to state $m \in R^{ca}$, then a transition is made from $m \in R^{ca}$ to $j \in R^{sc}$.

the Markov chain will alter between the two groups of states, see Figure 4.4. When we transmit the data package containing y_k the state of the Markov chain will be r_k^{sc}, and when u_k is transmitted the state will be r_k^{ca}. The probability distribution of τ_k^{sc} is given by r_k^{sc}, and the distribution of τ_k^{ca} by r_k^{ca}. Let the closed loop system be (4.6) with

$$\tau_k = [\, \tau_k^{sc} \quad \tau_k^{ca} \,].$$

To simplify writing we introduce the notation

$$\begin{aligned} \Phi_k &= \Phi(\tau_k) \\ \Gamma_k &= \Gamma(\tau_k), \end{aligned}$$

for the matrices in the closed loop system. Introduce the Markov state probabilities

$$\begin{aligned} \pi_j^{sc}(k) &= P(r_k^{sc} = j) \\ \pi_j^{ca}(k) &= P(r_k^{ca} = j). \end{aligned}$$

Collect the random components up to time k in

$$\mathcal{Y}_k = \{e_0, ..., e_k, r_0^{sc}, ..., r_k^{sc}, \tau_0^{sc}, ..., \tau_k^{sc}, r_0^{ca}, ..., r_k^{ca}, \tau_0^{ca}, ..., \tau_k^{ca}\}.$$

Introduce the conditional state covariance

$$P_i(k) = \underset{\mathcal{Y}_{k-1}}{\mathrm{E}} \left(z_k z_k^T \mid r_k^{sc} = i \right)$$

and

$$\widetilde{P}_i(k) = \pi_i^{sc}(k) P_i(k).$$

The following Theorem gives a procedure to evaluate $\widetilde{P}_i(k)$.

THEOREM 4.2
The vectorized state covariance matrix $\widetilde{\mathbf{P}}(k)$ satisfies the recursion

$$\begin{aligned} \widetilde{\mathbf{P}}(k+1) = {} & (Q^T \otimes I)\, \mathrm{block}_{ij}(q_{ji}\mathcal{A}_{ji})\widetilde{\mathbf{P}}(k) \\ & + (Q^T \otimes I)\, \mathrm{block}_{ij}(q_{ji}\pi_i^{sc}(k)\mathcal{G}_{ji}), \end{aligned} \quad (4.26)$$

where

$$\begin{aligned} \mathcal{A}_{im} &= \underset{\tau_k^{sc},\tau_k^{ca}}{\mathrm{E}} \left(\Phi_k \otimes \Phi_k \mid r_k^{ca} = m, r_k^{sc} = i \right), \\ \mathcal{G}_{im} &= \underset{\tau_k^{sc},\tau_k^{ca}}{\mathrm{E}} \left(\Gamma_k R \Gamma_k^T \mid r_k^{ca} = m, r_k^{sc} = i \right), \\ \widetilde{\mathbf{P}}(k) &= \begin{bmatrix} \mathrm{vec}\, \widetilde{P}_1(k) \\ \mathrm{vec}\, \widetilde{P}_2(k) \\ \vdots \\ \mathrm{vec}\, \widetilde{P}_s(k) \end{bmatrix}, \end{aligned}$$

and $\mathrm{block}_{ij}(A(i,j))$ means the block matrix with elements $A(i,j)$ in block position (i,j) for $i,j \in \{1,...,s\}$. □

The proof of Theorem 4.2 is given in the following section.

From (4.26) it is seen that the closed loop will be stable, in the sense that the covariance is finite, if the matrix $(Q^T \otimes I)\, \mathrm{block}_{ij}(q_{ji}\mathcal{A}_{ji})$ has all its eigenvalues inside the unit circle.

Proof of Theorem 4.2

Introduce the transition probabilities for the reversed Markov chain

$$\begin{aligned}\widetilde{q}^{sc}_{mj}(k) &= P(r^{ca}_{k-1} = m \mid r^{sc}_k = j)\\ \widetilde{q}^{ca}_{im}(k) &= P(r^{sc}_k = i \mid r^{ca}_k = m).\end{aligned}$$

It is easily seen that

$$\pi^{sc}_j(k+1)\widetilde{q}^{sc}_{mj}(k+1) = q_{mj}\pi^{ca}_m(k)$$

and

$$\pi^{ca}_m(k)\widetilde{q}^{ca}_{im}(k) = q_{im}\pi^{sc}_i(k).$$

Using the Markov property lemma, Lemma 4.2, we get

$$\begin{aligned}\widetilde{P}_j(k+1) &= \pi^{sc}_j(k+1)\underset{\mathcal{Y}_k}{\mathrm{E}}(z_{k+1}z^T_{k+1} \mid r^{sc}_{k+1} = j)\\ &= \pi^{sc}_j(k+1)\underset{\mathcal{Y}_k}{\mathrm{E}}\left(\Phi_k z_k z^T_k \Phi^T_k + \Gamma_k e_k e^T_k \Gamma^T_k \mid r^{sc}_{k+1} = j\right)\\ &= \pi^{sc}_j(k+1)\sum_{m=1}^{s}\widetilde{q}^{sc}_{mj}(k+1)\underset{\mathcal{Y}_k}{\mathrm{E}}\left(\Phi_k z_k z^T_k \Phi^T_k\right.\\ &\qquad \left.+\Gamma_k e_k e^T_k \Gamma^T_k \mid r^{sc}_{k+1} = j, r^{ca}_k = m\right)\\ &= \pi^{sc}_j(k+1)\sum_{m=1}^{s}\widetilde{q}^{sc}_{mj}(k+1)\sum_{i=1}^{s}\widetilde{q}^{ca}_{im}(k)\underset{\mathcal{Y}_k}{\mathrm{E}}\left(\Phi_k z_k z^T_k \Phi^T_k\right.\\ &\qquad \left.+\Gamma_k e_k e^T_k \Gamma^T_k \mid r^{sc}_{k+1} = j, r^{ca}_k = m, r^{sc}_k = i\right)\\ &= \sum_{m=1}^{s} q_{mj}\sum_{i=1}^{s} q_{im}\pi^{sc}_i(k)\underset{\mathcal{Y}_k}{\mathrm{E}}\left(\Phi_k z_k z^T_k \Phi^T_k\right.\\ &\qquad \left.+\Gamma_k e_k e^T_k \Gamma^T_k \mid r^{sc}_{k+1} = j, r^{ca}_k = m, r^{sc}_k = i\right)\\ &= \sum_{m=1}^{s} q_{mj}\sum_{i=1}^{s} q_{im}\pi^{sc}_i(k)\underset{\mathcal{Y}_k}{\mathrm{E}}\left(\Phi_k z_k z^T_k \Phi^T_k\right.\\ &\qquad \left.+\Gamma_k e_k e^T_k \Gamma^T_k \mid r^{ca}_k = m, r^{sc}_k = i\right).\end{aligned}$$

Vectorizing $\widetilde{P}_j(k+1)$ and using the conditional independence lemma, Lemma 4.1, we get

$$\operatorname{vec}\widetilde{P}_j(k+1) = \sum_{m=1}^{s} q_{mj} \sum_{i=1}^{s} q_{im}\mathcal{A}_{im} \operatorname{vec}\widetilde{P}_i(k) + \sum_{m=1}^{s} q_{mj} \sum_{i=1}^{s} q_{im}\pi_i^{sc}(k) \operatorname{vec}\mathcal{G}_{im}.$$

Using matrix notation and Kronecker products, see Appendix A, this linear recursion can be written as the Lyapunov recursion (4.26), which completes the proof. □

EXAMPLE 4.2—VARIABLE DELAY

Consider the closed loop system in Figure 4.5. Assume that the dis-

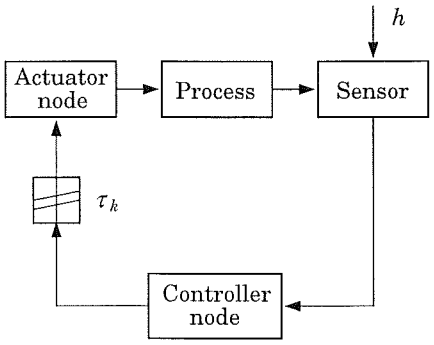

Figure 4.5 Digital control system with induced delay. The time-delay τ_k is determined by the state r_k of the Markov chain in Figure 4.6

tribution of the communication delay τ_k from controller to actuator is given by the state r_k of a Markov chain. The Markov chain has two states, see Figure 4.6. The delay is

$$\tau_k = \begin{cases} 0 & \text{if } r_k = 1, \\ \operatorname{rect}(d-a, d+a) & \text{if } r_k = 2, \end{cases} \tag{4.27}$$

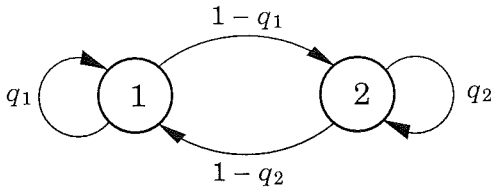

Figure 4.6 Markov chain with two states. State 1 corresponds to no delay, and state 2 corresponds to a time-delay in the interval $[d-a, d+a]$, see Equation (4.27).

where rect$(d-a, d+a)$ denotes a uniform distribution on the interval $[d-a, d+a]$. It is also assumed that $d-a > 0$ and $d+a < h$. The controlled process is

$$\begin{cases} \dot{x} = x + u + e \\ y = x \end{cases}$$

Let the control strategy be given by $u_k = -Lx_k$. Discretizing the process in the sampling instants determined by the sensor we get

$$x_{k+1} = \Phi x_k + \Gamma_0(\tau_k)u_k + \Gamma_1(\tau_k)u_{k-1} + \Gamma_e e_k$$

where

$$\begin{aligned} \Phi &= e^{Ah} = e^h, \\ \Gamma_0(\tau_k) &= \begin{cases} \int_0^h e^{As} ds B = e^h - 1, & \text{if } r_k = 1, \\ \int_0^{h-d} e^{As} ds B = e^{h-d} - 1, & \text{if } r_k = 2. \end{cases} \\ \Gamma_1(\tau_k) &= \begin{cases} 0, & \text{if } r_k = 1, \\ \int_{h-d}^h e^{As} ds B = e^{h-d}(e^d - 1), & \text{if } r_k = 2. \end{cases} \end{aligned}$$

Introduce the closed loop state z_k as

$$z_k = \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}.$$

The closed loop system can then be written as

$$z_{k+1} = A(\tau_k)z_k + \Gamma(\tau_k)e_k,$$

where

$$A(\tau_k) = \begin{bmatrix} \Phi - \Gamma_0(\tau_k)L & \Gamma_1(\tau_k) \\ -L & 0 \end{bmatrix} \qquad \Gamma(\tau_k) = \begin{bmatrix} \Gamma_e \\ 0 \end{bmatrix}.$$

Stability of the closed loop system is determined by the spectral radius of $(Q^T \otimes I)\,\mathrm{diag}(\mathcal{A}_i)$, where

$$\begin{aligned} \mathcal{A}_1 &= A(0) \otimes A(0), \\ \mathcal{A}_2 &= \mathop{\mathrm{E}}_{\tau_k}\{A(\tau_k) \otimes A(\tau_k) | r_k = 2\} \end{aligned}$$

and the transition matrix for the Markov chain is

$$Q = \begin{bmatrix} q_1 & 1-q_1 \\ 1-q_2 & q_2 \end{bmatrix}.$$

Figure 4.7 shows the stability region in the $q_1 - q_2$ space for $h = 0.3$, $d = 0.8h$, $a = 0.1h$ and $L = 4$. This corresponds to a control close to deadbeat for the nominal case. In Figure 4.7 the upper left corner ($q_1 = 1$ and $q_2 = 0$) corresponds to the nominal system, i.e. a system without delay. The lower right corner ($q_1 = 0$ and $q_2 = 1$) corresponds to the system with a delay uniformly distributed on $[d - a, d + a]$. As seen from Figure 4.7 the controller does not stabilize the process in this case. When $q_1 = q_2$ the stationary distribution of the state in the Markov chain is $\pi_1 = \pi_2 = 0.5$. In Figure 4.7 this is a line from the lower left corner to the upper right corner. Note that if the Markov chain stays a too long or a too short short time in the states (i.e. if $q_1 = q_2 \approx 1$ or $q_1 = q_2 \approx 0$) the closed loop is not stable, but for a region in between the closed loop is stable (i.e. if $q_1 = q_2 \approx 0.5$). If $q_1 = 1$ and $q_2 < 1$ the system is stable. If $q_1 = 1$ and $q_2 = 1$ stability depends on the initial state for the Markov chain. □

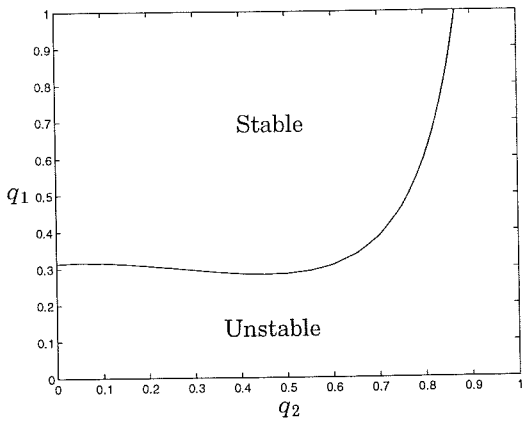

Figure 4.7 Stability region in $q_1 - q_2$ space for the system in Example 4.2.

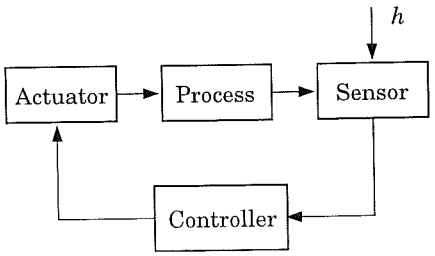

Figure 4.8 Digital control system.

EXAMPLE 4.3—VACANT SAMPLING - TWO HEURISTIC CONTROL STRATEGIES
Consider the digital control system in Figure 4.8. Due to sensor failure or communication problem samples can be lost. This is called vacant sampling. Vacant sampling is modeled with a Markov chain, see Figure 4.6. If the Markov chain is in State 2 the sample is lost, and State 1

corresponds to normal operation. Let the process be

$$\begin{cases} \dfrac{dx}{dt} = x + u + e \\ y = x. \end{cases}$$

Discretizing the process in the sampling instants we get

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Gamma_e e_k.$$

We will now compare two heuristic control strategies. The first is given by $u_k = -Lx_k$ if we get a new sample, and $u_k = u_{k-1}$ if the sample is lost. The closed loop system will be

$$z_{k+1} = A(r_k)z_k + B(r_k)e_k,$$

where

$$\begin{aligned} z_k &= \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} \\ A(r_k) &= \begin{cases} \begin{bmatrix} \Phi - \Gamma L & 0 \\ -L & 0 \end{bmatrix} & \text{if } r_k = 1, \\ \begin{bmatrix} \Phi & \Gamma \\ 0 & I \end{bmatrix} & \text{if } r_k = 2 \end{cases} \\ B(r_k) &= \begin{bmatrix} \Gamma_e \\ 0 \end{bmatrix}. \end{aligned}$$

The sampling period is chosen to $h = 0.3$, and the noise variance such that $\Gamma_e = 1$. The feedback is designed to minimize the nominal LQ cost function, i.e. assuming $r_k \equiv 1$,

$$J = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} (x_k x_k^T + u_k u_k^T).$$

This gives $L = 1.776$. Using (4.21) we can determine stability and calculate stationary performance. The stability region and level curves

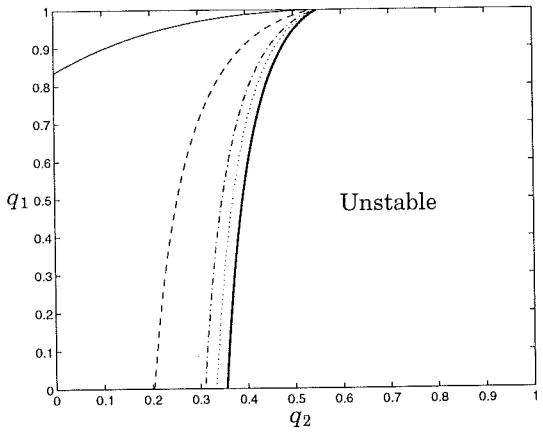

Figure 4.9 Stability region and level curves for J in the $q_1 - q_2$ space. The level curves are plotted for $J = \{10(-), 20(--), 50(-.), 100(..), \infty(thick)\}$.

for J in the $q_1 - q_2$ space are shown in Figure 4.9. Note that we fail to stabilize the system if we have large periods with lost samples (large q_2).
The second choice of controller is to use state feedback when the sensor is working, and in case of sensor failure use feedback from an estimated state $\hat{x}_k$. An estimate of x_k can be formed as

$$\hat{x}_{k+1} = \begin{cases} x_{k+1} & \text{if } r_k = 1, \\ \Phi\hat{x}_k + \Gamma u_k & \text{if } r_k = 2. \end{cases}$$

The closed loop system will be

$$z_{k+1} = A(r_k)z_k + B(r_k)e_k,$$

where

$$z_k = \begin{bmatrix} x_k \\ \hat{x}_k \end{bmatrix}$$

$$A(r_k) = \begin{cases} \begin{bmatrix} \Phi - \Gamma L & 0 \\ \Phi - \Gamma L & 0 \end{bmatrix} & \text{if } r_k = 1, \\ \begin{bmatrix} \Phi & -\Gamma L \\ 0 & \Phi - \Gamma L \end{bmatrix} & \text{if } r_k = 2 \end{cases}$$

$$B(r_k) = \begin{bmatrix} \Gamma_e \\ 0 \end{bmatrix}.$$

The stability region and level curves for J in the $q_1 - q_2$ space are shown in Figure 4.10. In Figure 4.9 and Figure 4.10 the upper left corner corresponds to the nominal case without lost samples. The lower right corner corresponds a system where all samples are lost, of course the controllers fail to stabilize the system in this region. It can be seen from Figure 4.9 and Figure 4.10 that the stability region of the closed

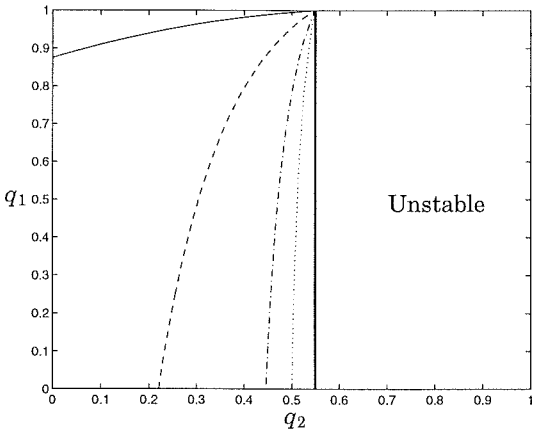

Figure 4.10 Stability region and level curves for J in the $q_1 - q_2$ space with controller using prediction. The level curves are plotted for $J = \{10(-), 20(--), 50(-.), 100(..), \infty(thick)\}$. Compare with Figure 4.9. Note that the stability region is increased using the second control with an estimator.

loop system is increased using the second control with an estimator. There are however regions in the $q_1 - q_2$ space where the first controller is better. In Figure 4.11 the area where the controller without prediction outperforms the controller using prediction is shown. This is not surprising, although the controller using prediction has a larger stability area it is not an optimal controller. □

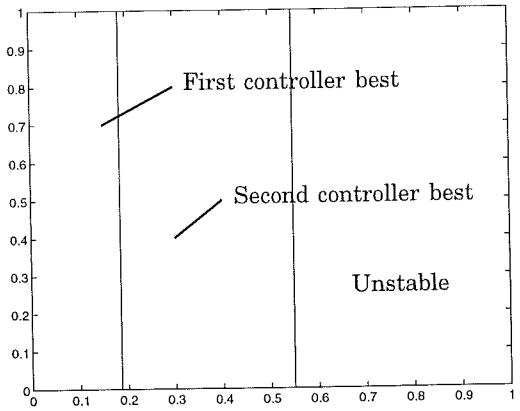

Figure 4.11 Regions in the $q_1 - q_2$ space where the two controllers are performing best. Remember that none of the controllers are optimal.

4.4 Simulation of Systems with Network Delays

An alternative to the algebraic analysis of control systems with network induced delays is to use simulation. This can also be useful in cases when there are no analytic results available. Typical cases are time domain responses such as step responses, noise amplification, response to load disturbances etc. Simulation can also be used when we want to study effects of nonlinear elements in the control loop, for example actuator saturation. An alternative way to evaluate a cost function is to do Monte Carlo simulations and calculate the mean value of the cost function, one such example is given in Chapter 5. A problem with this approach is the large number of samples needed to get

a small confidence interval for the cost function. Although this can be lowered using variance reduction techniques, see Morgan (1984).

5

Optimal Stochastic Control

This chapter deals with controller design for the distributed digital control system in Figure 5.1. The controlled process is assumed to be

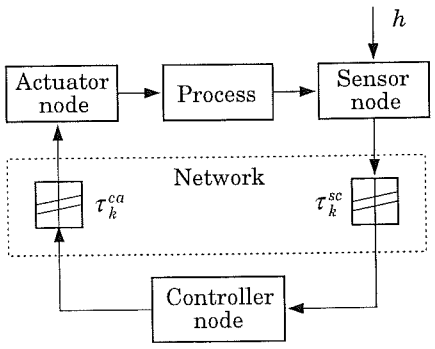

Figure 5.1 Distributed digital control system with induced delays.

$$\frac{dx}{dt} = Ax(t) + Bu(t) + v(t), \tag{5.1}$$

where $x(t) \in \mathcal{R}^n$, $u(t) \in \mathcal{R}^m$ and $v(t) \in \mathcal{R}^n$. A and B are matrices of appropriate sizes, $u(t)$ is the controlled input and $v(t)$ is white noise with zero mean and covariance R_v. The influence from the network is collected in the variables τ_k^{sc} and τ_k^{ca}, which is the delay from sensor to controller and from controller to actuator. We will assume that

the delay from sensor to actuator is less than the sampling period h, i.e. $\tau_k^{sc} + \tau_k^{ca} < h$. We also assume that old time delays are known when the control signal is calculated, i.e. when we calculate u_k the set $\{\tau_0^{sc}, \ldots, \tau_k^{sc}, \tau_0^{ca}, \ldots, \tau_{k-1}^{ca}\}$ of time delays is known. The knowledge of old time delays can for instance be solved by clock synchronization and time stamping as described in Chapter 2. Discretizing (5.1) in the sampling instants determined by the sensor node, see Chapter 2, gives

$$x_{k+1} = \Phi x_k + \Gamma_0(\tau_k^{sc}, \tau_k^{ca})u_k + \Gamma_1(\tau_k^{sc}, \tau_k^{ca})u_{k-1} + v_k. \tag{5.2}$$

The output equation is

$$y_k = Cx_k + w_k, \tag{5.3}$$

where $y_k \in \mathcal{R}^p$. The stochastic processes v_k and w_k are uncorrelated white noise with zero mean and covariance matrices R_1 and R_2 respectively.
If the network delays can be assumed to be constant, (5.2) degenerates to the standard process used in sampled data control theory, and thus the standard design tools can be used for synthesis, see Åström and Wittenmark (1990).

In this chapter we will assume that consecutive network delays are stochastically independent, see Chapter 3. Section 5.1 solves the LQ-problem for (5.2) with full state information. In Section 5.2 the optimal state estimator is derived. The LQG-problem with output feedback is solved in Section 5.3. Through a separation theorem it is seen that the optimal controller is the combination of the LQ-controller and the optimal state estimator. The parameters of the optimal controller can be precalculated and interpolated from a tabular. The proof of the separation property follows the lines for the delay-free case in Åström (1970). Section 5.4 discusses complexity and implementation of the optimal controller. A suboptimal scheme is also proposed. Section 5.5 compares the different control schemes in an example of distributed digital control.

5.1 Optimal State Feedback

In this section we solve the control problem set up by the cost function

$$J_N = x_N^T Q_N x_N + \mathrm{E} \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \tag{5.4}$$

where Q is symmetric with the structure

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix}. \tag{5.5}$$

Here Q is positive semi-definite and Q_{22} is positive definite. The solution of this problem follows by the same technique as for the standard LQG problem. We have the following result:

THEOREM 5.1—OPTIMAL STATE FEEDBACK
Given the plant (5.2), with noise free measurement of the state vector x_k, i.e. $y_k = x_k$. The control law that minimizes the cost function (5.4) is given by

$$u_k^* = -L(\tau_k^{sc}) \begin{bmatrix} x_k \\ u_{k-1}^* \end{bmatrix} \tag{5.6}$$

where

$$\begin{aligned}
L(\tau_k^{sc}) &= (Q_{22} + \tilde{S}_{k+1}^{22})^{-1} [\, Q_{12}^T + \tilde{S}_{k+1}^{21} \quad \tilde{S}_{k+1}^{23} \,] \\
\tilde{S}_{k+1}(\tau_k^{sc}) &= \mathop{\mathrm{E}}_{\tau_k^{ca}} \{ G^T(\tau_k^{sc}, \tau_k^{ca}) S_{k+1} G(\tau_k^{sc}, \tau_k^{ca}) \} \\
G(\tau_k^{sc}, \tau_k^{ca}) &= \begin{bmatrix} \Phi & \Gamma_0(\tau_k^{sc}, \tau_k^{ca}) & \Gamma_1(\tau_k^{sc}, \tau_k^{ca}) \\ 0 & I & 0 \end{bmatrix} \\
S_k &= \mathop{\mathrm{E}}_{\tau_k^{sc}} \{ F_1^T(\tau_k^{sc}) Q F_1(\tau_k^{sc}) + F_2^T(\tau_k^{sc}) \tilde{S}_{k+1}(\tau_k^{sc}) F_2(\tau_k^{sc}) \} \\
F_1(\tau_k^{sc}) &= (Q_{22} + \tilde{S}_{k+1}^{22})^{-1} \begin{bmatrix} (Q_{22} + \tilde{S}_{k+1}^{22}) I & 0 \\ -(Q_{12}^T + \tilde{S}_{k+1}^{21}) & -\tilde{S}_{k+1}^{23} \end{bmatrix} \\
&= \begin{bmatrix} I & 0 \\ -L(\tau_k^{sc}) \end{bmatrix}
\end{aligned}$$

$$F_2(\tau_k^{sc}) = (Q_{22} + \tilde{S}_{k+1}^{22})^{-1} \begin{bmatrix} (Q_{22} + \tilde{S}_{k+1}^{22})I & 0 \\ -(Q_{12}^T + \tilde{S}_{k+1}^{21}) & -\tilde{S}_{k+1}^{23} \\ 0 & (Q_{22} + \tilde{S}_{k+1}^{22}) \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ -L(\tau_k^{sc}) \\ 0 & I \end{bmatrix}$$

$$S_N = \begin{bmatrix} Q_N & 0 \\ 0 & 0 \end{bmatrix}.$$

$\tilde{S}_k^{ij}$ is block (i,j) of the symmetric matrix $\tilde{S}_k(\tau^{sc})$, and Q_{ij} is block (i,j) of Q. □

Proof Introduce a new state variable $z_k = \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}$. Using dynamic programming with S_k the cost to go at time k, and with α_k the part of the cost function that cannot be affected by control, gives

$$z_k^T S_k z_k + \alpha_k = \min_{u_k} \mathrm{E}_{\tau_k^{sc}, \tau_k^{ca}, v_k} \left\{ \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix} + z_{k+1}^T S_{k+1} z_{k+1} \right\} + \alpha_{k+1}$$

$$= \mathrm{E}_{\tau_k^{sc}} \min_{u_k} \mathrm{E}_{\tau_k^{ca}, v_k} \left\{ \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix} + z_{k+1}^T S_{k+1} z_{k+1} \middle| \tau_k^{sc} \right\} + \alpha_{k+1}$$

$$= \mathrm{E}_{\tau_k^{sc}} \min_{u_k} \left\{ \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} x_k \\ u_k \\ u_{k-1} \end{bmatrix}^T \tilde{S}_{k+1} \begin{bmatrix} x_k \\ u_k \\ u_{k-1} \end{bmatrix} \right\}$$

$$+ \alpha_{k+1} + \operatorname{tr} S_{k+1}^{11} R_1.$$

The second equality follows from the fact that τ_k^{sc} is known when u_k is determined. The third equality follows from independence of $\begin{bmatrix} x_k \\ u_k \end{bmatrix}$ and τ_k^{ca}, and from the definition of $\tilde{S}_{k+1}$. The resulting expression is a quadratic form in u_k. Minimizing this with respect to u_k gives the optimal control law (5.6). From the assumption that Q is symmetric it follows that S_k and $\tilde{S}_k$ are symmetric. □

Theorem 5.1 states that the optimal controller with full state information is a linear τ_k^{sc}-depending feedback from the state and the previous control signal

$$u_k = -L(\tau_k^{sc}, S_{k+1}) \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}.$$

The equation involved in going from S_{k+1} to S_k is a stochastic Riccati equation evolving backwards in time. Each step in this iteration will contain expectation calculations with respect to the stochastic variables τ_k^{sc} and τ_k^{ca}. Under reasonable assumptions, that we will not discuss here, a stationary value S_∞ of S_k can be found by iterating the stochastic Riccati equation. In practice a tabular for $L(\tau_k^{sc}, S_\infty)$ can then be calculated to get a control law on the form

$$u_k = -L(\tau_k^{sc}) \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix},$$

where $L(\tau_k^{sc})$ is interpolated from the tabular values of $L(\tau_k^{sc}, S_\infty)$ in real-time.

5.2 Optimal State Estimate

It is often impossible to get full state information. A common solution to this is to construct a state estimate from the available data. In our setup there is the problem of the random time delays which enter (5.2) in a nonlinear fashion. The fact that the old time delays up to time $k-1$ are known at time k, however, allows the standard Kalman filter of the process state to be optimal. This is because x_k only depend on delays in the set $\{\tau_0^{sc}, ..., \tau_{k-1}^{sc}, \tau_0^{ca}, ..., \tau_{k-1}^{ca}\}$, as seen from (5.2).

When we are to calculate an estimate of x_k we assume that we know old values of the process output and process input. These can simply be stored in the controller for later use. We also assume that old values of the transfer delays for process output measurements and control signals are known. One way to achieve this is by time stamping of signals transferred in the control system, see Chapter 2. Denote the

information available when the control signal u_k is calculated by $\mathcal{Y}_k$. This has the structure

$$\mathcal{Y}_k = \{y_0, ..., y_k, u_0, ..., u_{k-1}, \tau_0^{sc}, ..., \tau_k^{sc}, \tau_0^{ca}, ..., \tau_{k-1}^{ca}\}.$$

Notice that the sensor to controller delay τ^{sc} at time k and older are available, but the controller to actuator delays τ^{ca} are only known up to time $k-1$.

The state estimator that minimizes the error covariance is given in the following theorem.

THEOREM 5.2—OPTIMAL STATE ESTIMATE
Given the plant (5.2)–(5.3). The estimator

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \overline{K}_k(y_k - C\hat{x}_{k|k-1}) \tag{5.7}$$

with

$$\begin{aligned}
\hat{x}_{k+1|k} &= \Phi\hat{x}_{k|k-1} + \Gamma_0(\tau_k^{sc}, \tau_k^{ca})u_k + \Gamma_1(\tau_k^{sc}, \tau_k^{ca})u_{k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \\
\hat{x}_{0|-1} &= \mathrm{E}(x_0) \\
P_{k+1} &= \Phi P_k\Phi^T + R_1 - \Phi P_k C^T[CP_kC^T + R_2]^{-1}CP_k\Phi \\
P_0 &= R_0 = \mathrm{E}(x_0x_0^T) \\
K_k &= \Phi P_k C^T[CP_kC^T + R_2]^{-1} \\
\overline{K}_k &= P_kC^T[CP_kC^T + R_2]^{-1}
\end{aligned}$$

minimizes the error variance $\mathrm{E}\{[x_k - \hat{x}_k]^T[x_k - \hat{x}_k] \mid \mathcal{Y}_k\}$. Note that the filter gains K_k and $\overline{K}_k$ do not depend on τ^{sc} and τ^{ca}. Moreover, the estimation error is Gaussian with zero mean and covariance $P_{k|k} = P_k - P_kC^T[CP_kC^T + R_2]^{-1}CP_k$. □

Proof Note that the random matrices in the process (5.2), $\Gamma_0(\tau_k^{sc}, \tau_k^{ca})$ and $\Gamma_1(\tau_k^{sc}, \tau_k^{ca})$, are known when the estimate $\hat{x}_{k|k-1}$ is calculated. This simply follows from the assumption that old time delays are known when we make the estimate. By this we know how the control signal enters x_{k+1}, and the optimality of the estimator can be proved in the same way as the standard Kalman filter for time-varying, linear systems, see Anderson and Moore (1979). See also Chen *et al.* (1989). □

5.3 Optimal Output Feedback

The following theorem justifies use of the estimated state in the optimal controller.

THEOREM 5.3—SEPARATION PROPERTY
Given the plant (5.2)–(5.3), with $\mathcal{Y}_k$ known when the control signal is calculated. The controller that minimizes the cost function (5.4) is given by

$$u_k^* = -L(\tau_k^{sc}) \begin{bmatrix} \hat{x}_{k|k} \\ u_{k-1}^* \end{bmatrix} \tag{5.8}$$

with

$$L(\tau_k^{sc}) = (Q_{22} + \tilde{S}_{k+1}^{22})^{-1} [\, Q_{12}^T + \tilde{S}_{k+1}^{21} \quad \tilde{S}_{k+1}^{23} \,], \tag{5.9}$$

where $\tilde{S}_k$ is calculated as in Theorem 5.1, and $\hat{x}_{k|k}$ is the minimum variance estimate from Theorem 5.2. □

To prove Theorem 5.3 we will need some lemmas. The first lemma is from Åström (1970).

LEMMA 5.1
Let $\mathrm{E}[\cdot \mid y]$ denote the conditional mean given y. Assume that the function $f(y,u) = \mathrm{E}[l(x,y,u) \mid y]$ has a unique minimum with respect to $u \in \mathcal{U}$ for all $y \in \mathcal{Y}$. Let $u^0(y)$ denote the value of u for which the minimum is achieved. Then

$$\min_{u(y)} \mathrm{E}\, l(x,y,u) = \mathrm{E}\, l(x,y,u^0(y)) = \mathop{\mathrm{E}}_{y}\{\min_{u} \mathrm{E}[l(x,y,u) \mid y]\}, \tag{5.10}$$

where $\mathop{\mathrm{E}}_{y}$ denotes the mean value with respect to the distribution of y. □

Proof This is Lemma 3.2 in Chapter 8 of Åström (1970). □

LEMMA 5.2
With the notation in (5.7) and under the conditions for Theorem 5.2 the following holds.

$$\begin{aligned}
&\mathop{\mathrm{E}}_{\tau_k^{ca},v_k,w_{k+1}}\left\{\begin{bmatrix}\hat{x}_{k+1|k+1}\\ u_k\end{bmatrix}^T S_{k+1}\begin{bmatrix}\hat{x}_{k+1|k+1}\\ u_k\end{bmatrix}\Big|\,\mathcal{Y}_k\right\}\\
&\quad=\begin{bmatrix}\hat{x}_{k|k}\\ u_k\\ u_{k-1}\end{bmatrix}^T \widetilde{S}_{k+1}(\tau_k^{sc})\begin{bmatrix}\hat{x}_{k|k}\\ u_k\\ u_{k-1}\end{bmatrix}+\operatorname{tr}(R_1C^T\overline{K}_{k+1}^T S_{k+1}^{11}\overline{K}_{k+1}C)\\
&\qquad+\operatorname{tr}(R_2\overline{K}_{k+1}^T S_{k+1}^{11}\overline{K}_{k+1})+\operatorname{tr}(P_{k|k}\Phi^T C^T\overline{K}_{k+1}^T S_{k+1}^{11}\overline{K}_{k+1}C\Phi),
\end{aligned}$$

where S_k^{11} is block $(1,1)$ of the matrix S_k. □

Proof The calculations are similar to those in Theorem 5.1. In Theorem 5.2 the state estimate recursion is written as a recursion in $\hat{x}_{k|k-1}$. This can by use of the equations in Theorem 5.2 be rewritten as a recursion in $\hat{x}_{k|k}$.

$$\begin{aligned}
\hat{x}_{k+1|k+1} &= (I-\overline{K}_{k+1}C)\hat{x}_{k+1|k}+\overline{K}_{k+1}y_{k+1}\\
&= (I-\overline{K}_{k+1}C)\left\{\Phi\hat{x}_{k|k}+\Gamma_0(\tau_k^{sc},\tau_k^{ca})u_k+\Gamma_1(\tau_k^{sc},\tau_k^{ca})u_{k-1}\right\}\\
&\quad+\overline{K}_{k+1}\{C(\Phi x_k+\Gamma_0(\tau_k^{sc},\tau_k^{ca})u_k+\Gamma_1(\tau_k^{sc},\tau_k^{ca})u_{k-1}+v_k)\\
&\quad+w_{k+1}\}. \qquad (5.11)
\end{aligned}$$

By introducing the estimation error $\tilde{x}_k = x_k - \hat{x}_{k|k}$, which we know is orthogonal to $\hat{x}_{k|k}$ from Theorem 5.2, (5.11) can be written as

$$\begin{aligned}
\hat{x}_{k+1|k+1} &= \Phi\hat{x}_{k|k}+\Gamma_0(\tau_k^{sc},\tau_k^{ca})u_k+\Gamma_1(\tau_k^{sc},\tau_k^{ca})u_{k-1}\\
&\quad+\overline{K}_{k+1}C\Phi\tilde{x}_k+\overline{K}_{k+1}Cv_k+\overline{K}_{k+1}w_{k+1}. \qquad (5.12)
\end{aligned}$$

From this it follows that

$$\begin{bmatrix}\hat{x}_{k+1|k+1}\\ u_k\end{bmatrix}=G(\tau_k^{sc},\tau_k^{ca})\begin{bmatrix}\hat{x}_{k|k}\\ u_k\\ u_{k-1}\end{bmatrix}+H\begin{bmatrix}\tilde{x}_k\\ v_k\\ w_{k+1}\end{bmatrix}, \qquad (5.13)$$

where

$$G(\tau_k^{sc}, \tau_k^{ca}) = \begin{bmatrix} \Phi & \Gamma_0(\tau_k^{sc}, \tau_k^{ca}) & \Gamma_1(\tau_k^{sc}, \tau_k^{ca}) \\ 0 & I & 0 \end{bmatrix} \tag{5.14}$$

$$H = \begin{bmatrix} \overline{K}_{k+1} C\Phi & \overline{K}_{k+1} C & \overline{K}_{k+1} \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.15}$$

The sought equality can now be written as

$$\begin{aligned} &\mathop{\mathrm{E}}_{\tau_k^{ca}, v_k, w_{k+1}} \left\{ \begin{bmatrix} \hat{x}_{k+1|k+1} \\ u_k \end{bmatrix}^T S_{k+1} \begin{bmatrix} \hat{x}_{k+1|k+1} \\ u_k \end{bmatrix} \mid \mathcal{Y}_k \right\} \\ &\quad = \begin{bmatrix} \hat{x}_{k|k} \\ u_k \\ u_{k-1} \end{bmatrix}^T \mathop{\mathrm{E}}_{\tau_k^{ca}} \left\{ G^T(\tau_k^{sc}, \tau_k^{ca}) S_{k+1} G(\tau_k^{sc}, \tau_k^{ca}) \right\} \begin{bmatrix} \hat{x}_{k|k} \\ u_k \\ u_{k-1} \end{bmatrix} \\ &\qquad + \mathop{\mathrm{E}}_{v_k, w_{k+1}} \left\{ \begin{bmatrix} \tilde{x}_k \\ v_k \\ w_{k+1} \end{bmatrix}^T H^T S_{k+1} H \begin{bmatrix} \tilde{x}_k \\ v_k \\ w_{k+1} \end{bmatrix} \mid \mathcal{Y}_k \right\} \\ &= \begin{bmatrix} \hat{x}_{k|k} \\ u_k \\ u_{k-1} \end{bmatrix}^T \widetilde{S}_{k+1}(\tau_k^{sc}) \begin{bmatrix} \hat{x}_{k|k} \\ u_k \\ u_{k-1} \end{bmatrix} + \operatorname{tr}(P_{k|k} \Phi^T C^T \overline{K}_{k+1}^T S_{k+1}^{11} \overline{K}_{k+1} C\Phi) \\ &\qquad + \operatorname{tr}(R_2 \overline{K}_{k+1}^T S_{k+1}^{11} \overline{K}_{k+1}) + \operatorname{tr}(R_1 C^T \overline{K}_{k+1}^T S_{k+1}^{11} \overline{K}_{k+1} C), \end{aligned} \tag{5.16}$$

where

$$\widetilde{S}_{k+1}(\tau_k^{sc}) = \mathop{\mathrm{E}}_{\tau_k^{ca}} \left\{ G^T(\tau_k^{sc}, \tau_k^{ca}) S_{k+1} G(\tau_k^{sc}, \tau_k^{ca}) \right\}. \tag{5.17}$$

The first part of the first equality follows from that $\hat{x}_{k|k}$, u_k, and u_{k-1} are independent of $\tilde{x}_k$, τ_k^{ca}, v_k, and w_{k+1}. The second part of the first equality follows from that H is independent of τ_k^{ca}. The second equality follows from that $\tilde{x}_k$ is independent of v_k and w_{k+1}. □

Proof of Theorem 5.3 By repeated use of Lemma 5.1, and the knowledge that $\hat{x}_{k|k}$ is a sufficient statistic for the conditional distribu-

tion of x_k given $\mathcal{Y}_k$, we find the functional equation

$$W(\hat{x}_{k|k},k) = \mathop{\mathrm{E}}_{\tau_k^{sc}} \min_{u_k} \mathrm{E}\left\{ \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix} + W(\hat{x}_{k+1|k+1},k+1) \,\middle|\, \mathcal{Y}_k \right\}$$

$$= \mathop{\mathrm{E}}_{\tau_k^{sc}} \min_{u_k} \mathrm{E}\left\{ \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix} + W(\hat{x}_{k+1|k+1},k+1) \,\middle|\, \hat{x}_{k|k},\, \tau_k^{sc} \right\}. \quad (5.18)$$

The initial condition for the functional (5.18) is

$$W(\hat{x}_{N|N},N) = \mathrm{E}\left\{ x_N^T Q_N x_N \mid \hat{x}_{N|N} \right\}. \quad (5.19)$$

In (5.18) $\mathop{\mathrm{E}}_{\tau_k^{sc}}$ is brought outside the minimization using Lemma 5.1, i.e. τ_k^{sc} is known when we calculate the control signal. We will now show that the functional (5.18) has a solution which is a quadratic form

$$W(\hat{x}_{k|k},k) = \begin{bmatrix} \hat{x}_k \\ u_{k-1} \end{bmatrix}^T S_k \begin{bmatrix} \hat{x}_k \\ u_{k-1} \end{bmatrix} + s_k, \quad (5.20)$$

and that the functional is minimized by the controller of Theorem 5.1 with x_k replaced by $\hat{x}_{k|k}$. Using Theorem 5.2 we can rewrite the initial condition (5.19) as

$$W(\hat{x}_{N|N},N) = \hat{x}_{N|N}^T Q_N \hat{x}_{N|N} + \mathrm{tr}(Q_1 P_{N|N}), \quad (5.21)$$

which clearly is on the quadratic form (5.20). Proceeding by induction we assume that (5.20) holds for $k+1$ and we will then show that it also holds for k. We have that

$$W(\hat{x}_{k|k},k) = \mathop{\mathrm{E}}_{\tau_k^{sc}} \min_{u_k} \left\{ \begin{bmatrix} \hat{x}_{k|k} \\ u_k \end{bmatrix}^T Q \begin{bmatrix} \hat{x}_{k|k} \\ u_k \end{bmatrix} + \mathrm{tr}(P_{k|k}Q_1) \right.$$
$$+ \begin{bmatrix} \hat{x}_{k|k} \\ u_k \\ u_{k-1} \end{bmatrix}^T \widetilde{S}_{k+1}(\tau_k^{sc}) \begin{bmatrix} \hat{x}_{k|k} \\ u_k \\ u_{k-1} \end{bmatrix} + \mathrm{tr}(R_1 C^T \overline{K}_{k+1}^T S_{k+1}^{11} \overline{K}_{k+1} C)$$
$$\left. + \mathrm{tr}(R_2 \overline{K}_{k+1}^T S_{k+1}^{11} \overline{K}_{k+1}) + \mathrm{tr}(P_{k|k} \Phi^T C^T \overline{K}_{k+1}^T S_{k+1}^{11} \overline{K}_{k+1} C \Phi) + s_{k+1} \right\}, \quad (5.22)$$

where we have used Lemma 5.2 to rewrite $\mathrm{E}\, W(\hat{x}_{k+1|k+1}, k+1)$. Comparing (5.22) with the quadratic form in the proof of Theorem 5.1 we see that it is minimized by the control law

$$u_k^* = -(Q_{22} + \tilde{S}_{k+1}^{22})^{-1} \begin{bmatrix} Q_{12}^T + \tilde{S}_{k+1}^{21} & \tilde{S}_{k+1}^{23} \end{bmatrix} \begin{bmatrix} \hat{x}_{k|k} \\ u_{k-1}^* \end{bmatrix}, \tag{5.23}$$

where $\tilde{S}_{k+1}$ is as stated in Theorem 5.1. Using the optimal control in (5.22) and applying $\mathrm{E}_{\tau_k^{sc}}$, which can be moved inside $[\hat{x}_{k|k}^T \; u_{k-1}^T]$, we find that $W(\hat{x}_{k|k}, k)$ is on the quadratic form (5.20). The induction is thus completed and the criterion is minimized by the controller stated in the theorem. □

5.4 A Suboptimal Scheme

A small drawback with the optimal scheme is the need to form a tabular for the state feedback matrix $L(\tau_k^{sc})$. This is because it is typically too time-consuming to compute $L(\tau_k^{sc})$ from the Riccati equation in real-time. A solution to this problem is to precalculate $L(\tau_k^{sc})$ for some values of τ_k^{sc} and then in real-time interpolate from the precalulated values. Note that the tabular is one-dimensional and is computed off-line.

Another approach is to use a suboptimal control schemes. An alternative to the optimal controller is the suboptimal controller

$$u_k = -L \begin{bmatrix} \Phi_k^{pred} & \Gamma_k^{pred} \end{bmatrix} \begin{bmatrix} \hat{x}_{k|k} \\ u_{k-1} \end{bmatrix}, \tag{5.24}$$

where

$$\Phi_k^{pred} = e^{A(\tau_k^{sc} + \mathrm{E}\,\tau_k^{ca})},$$
$$\Gamma_k^{pred} = \int_0^{\tau_k^{sc} + \mathrm{E}\,\tau_k^{ca}} e^{As} ds B,$$

and L is the optimal state feedback vector in the delay-free setup. Here $\mathrm{E}\,\tau_k^{ca}$ is the mean value of τ_k^{ca}. The operation $\Phi_k^{pred}\hat{x}_{k|k} + \Gamma_k^{pred} u_{k-1}$

can be seen as a prediction from the state estimate at time kh to a state estimate when the control signal is applied at the actuator. This controller requires less real-time computations than the optimal controller in Section 5.3. In Section 5.5 this controller is compared with the optimal controller in a numerical example.

5.5 Example

Consider the following plant, both plant and design specifications are taken from Doyle and Stein (1979),

$$\begin{aligned} \frac{dx}{dt} &= \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 35 \\ -61 \end{bmatrix} \xi \\ y &= \begin{bmatrix} 2 & 1 \end{bmatrix} x + \eta, \end{aligned} \tag{5.25}$$

where $\mathrm{E}(\xi(t)) = \mathrm{E}(\eta(t)) = 0$ and $\mathrm{E}[\xi(t_1)\xi(t_2)] = \mathrm{E}[\eta(t_1)\eta(t_2)] = \delta(t_1 - t_2)$. The control objective is to minimize the cost function

$$J = \mathrm{E} \lim_{T\to\infty} \frac{1}{T} \int_0^T (x^T H^T H x + u^2) dt,$$

where $H = 4\sqrt{5}[\sqrt{35} \quad 1]$. The sampling period for the controller is chosen as $h = 0.05$. This is in accordance with the rule of thumb that is given in Åström and Wittenmark (1990). The time delays, τ_k^{sc} and τ_k^{ca}, are assumed to be uniformly distributed on the interval $[0, \alpha h/2]$, where α is a parameter in $[0, 1]$, the effect of which will be studied in the sequel.

The stationary cost function will be evaluated and compared for four different schemes, an LQG-controller neglecting the time delays, the scheme with buffers proposed in Luck and Ray (1990), the optimal controller derived in Section 5.3, and the suboptimal controller in Section 5.4.

The first design is done without taking any time delays into account. The process and the cost function are sampled to get discrete time equivalents, and the standard LQG-controller is calculated. This gives

the design

$$L = \begin{bmatrix} 38.911 \\ 8.094 \end{bmatrix}^T, \quad K = \begin{bmatrix} 2.690 \\ -4.484 \end{bmatrix}, \quad \overline{K} = \begin{bmatrix} 2.927 \\ -5.012 \end{bmatrix}.$$

This choice of L, K and $\overline{K}$ gives the following closed loop poles

$$\begin{aligned} sp(\Phi - \Gamma L) &= \{0.700 \pm 0.0702i\} \\ sp(\Phi - KC) &= \{0.743, 0.173\}. \end{aligned}$$

Even if these looks reasonable a Nyquist plot of the loop transfer function reveals a small phase margin, $\phi_m = 10.9°$. The small phase margin indicates that there could be problems to handle unmodeled time delays. Numerical evaluation of (4.13) gives the stability limit $\alpha_{crit} = 0.425$ for the controller neglecting the time delays.

The scheme by *Luck and Ray*, see Luck and Ray (1990), eliminates the randomness of the time delays by introduction of timed buffers. This will, however, introduce extra time delay in the loop. The design for this scheme is done in the same way as in the standard LQG-problem.

The third scheme we will compare is the optimal controller described in Section 5.3. Notice that the optimal state estimator gains $\overline{K}$ and K will be the same for the optimal controller as if the time delays were neglected. The feedback from the estimated state will have the form

$$u_k = -L(\tau_k^{sc}) \begin{bmatrix} \hat{x}_k \\ u_{k-1} \end{bmatrix}.$$

The feedback vector $L(\tau_k^{sc})$ is precomputed for 21 values of τ_k^{sc}, which then are used for interpolation in the controller implementation. The used interpolation method is linear interpolation for each element of $L(\tau_k^{sc})$.

The suboptimal controller (5.24) uses L, K, and $\overline{K}$ from the standard LQG-controller.

The stationary cost function has been evaluated for the four schemes by solving (4.15). For comparison the stationary cost has also been evaluated by Monte Carlo simulation, which is made by calculating

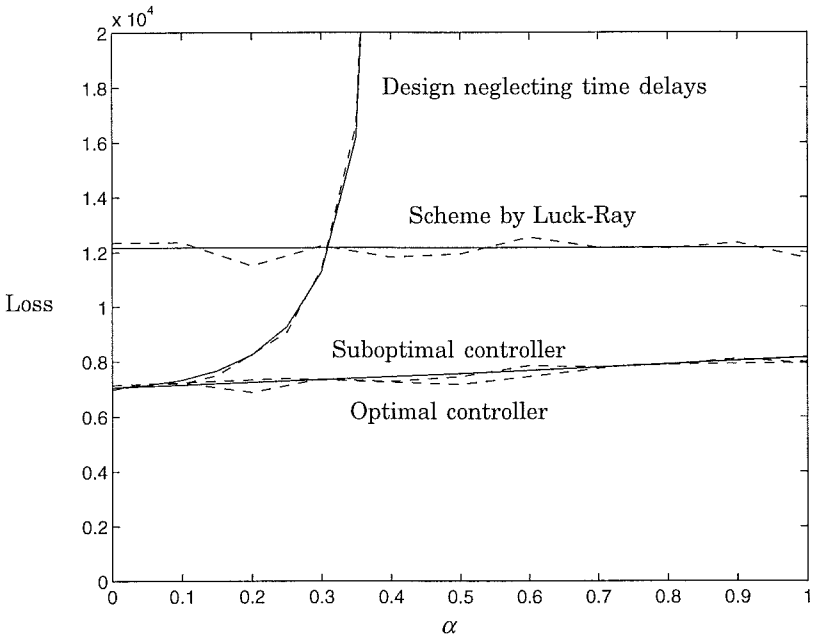

Figure 5.2 Exact calculated performance (solid lines) of the four schemes, and simulated performance (dashed lines) of system (5.25) as a function of the amount of stochastics in the time-delays. The time-delays are uniformly distributed on $[0, \alpha h/2]$. Notice the small difference between the suboptimal controller and the optimal controller. For $\alpha > 0.425$ the controller neglecting the time delays fails to stabilize the process.

the mean cost during $2 \cdot 10^4$ simulated samples. The results agree very well, see Figure 5.2. From Figure 5.2 it is seen that the controller neglecting the time delays fails to stabilize the process for $\alpha > \alpha_{crit}$. The optimal controller and the proposed suboptimal scheme outperforms the scheme proposed in Luck and Ray (1990). Note that for this example the cost is just slightly higher with the suboptimal controller than with the optimal controller.

6

Conclusions and Future Work

Conclusions

This thesis has presented a control problem that arises when control loops are closed over a communication network, as is being more and more common. The communication network introduces time delays in the control loop. The network induced time delays can have effect on system stability and performance. As a first step in this work three models for the communication delays were discussed:

- Constant delay
- Random delay, which is independent from transfer to transfer
- Random delay, with probability distributions governed by an underlying Markov chain

The model including a Markov chain has the attractive property to make modeling of trends in the network delays possible. These trends can, for instance, arise due to varying network load. Using a linear controller it was concluded that the closed loop system can be written on the form

$$z_{k+1} = \Phi(\tau_k)z_k + \Gamma(\tau_k)e_k, \tag{6.1}$$

where the stochastic properties of τ_k depend on the network model. For the different network models methods to evaluate a quadratic cost

function were developed. Through the same analysis we found criteria for mean square stability of (6.1) for the different network models. The LQG-optimal controller was developed in the case of random delays that are independent from transfer to transfer. The derived controller uses knowledge of old time delays. These can be calculated using "time-stamping" of messages in the network. The solution was found by combining the LQ-controller with a Kalman filter. It was shown that a separation principle holds.

Future Work

Future work will include studies of

- Optimal controllers when the distributions of the network delays are generated from a Markov chain. In this work it will also be natural to study methods to estimate the state of the Markov chain from the observed time delays.
- Experimental verification of the theoretical results for systems with network delays.
- Classification of which system setups that are sensitive to network delays. In the best case such an analysis could result in a rule of thumb for deciding which setups need special control schemes.
- Analysis and control strategies when the control delay can be larger than the sampling interval. A problem that occurs in this case is that there are then no guarantee that the samples arrive at the controller and the actuator in the order they were sent.
- Another interesting problem is how to control when we have one or several lost samples, so called *vacant sampling*? What can be done if we have a network *black out*, the network malfunctioning for period of time?

7

References

ANDERSON, B. and J. MOORE (1979): *Optimal filtering.* Prentice-Hall, Englewood Cliffs, N.J.

ÅSTRÖM, K. J. (1970): *Introduction to Stochastic Control Theory.* Academic Press, New York. Translated into Russian, Japanese and Chinese.

ÅSTRÖM, K. J. and B. WITTENMARK (1990): *Computer Controlled Systems—Theory and Design.* Prentice-Hall, Englewood Cliffs, New Jersey, second edition.

BERMAN, A. and R. J. PLEMMONS (1969): *Theory of Matrices.* Academic Press, New York.

BLAIR, W. P. and D. D. SWORDER (1975): "Feedback control of a class of linear discrete systems with jump parameters and quadratic cost criteria." *Int. J. Control*, **21:5**, pp. 833–841.

CHAN, H. and Ü. ÖZGÜNER (1995): "Closed-loop control of systems over a communications network with queues." *Int. J. Control*, **62:3**, pp. 493–510.

CHEN, H.-F., P. R. KUMAR, and J. H. VAN SCHUPPEN (1989): "On kalman filtering for conditionally gaussian systems with random matrices." *Systems & Control Letters*, pp. 397–404.

CHRISTIAN, F. and C. FETZER (1994): "Probabilistic internal clock synchronization." In *Proceedings of the Tirteenth Symposium on Reliable Distributed Systems.*

CHUNG, K. L. (1974): *A Course in Probability Theory.* Academic Press, New York.

DOYLE, J. C. and G. STEIN (1979): "Robustness with observers." *IEEE Trans. Automat. Contr.*, **AC-24:4**, pp. 607–611.

ELLIOT, J. E., L. AGGOUN, and J. B. MOORE (1995): *Hidden Markov models, estimation and control.* Springer-Verlag.

GAJIC, Z. and M. T. J. QURESHI (1995): *Lyapunov matrix equation in system stability and control.* Academic Press.

HALMOS, P. R. (1958): *Finite-Dimensional Vector Spaces.* D. Van Nostrand Company, Inc.

JI, Y. and H. J. CHIZECK (1990): "Controllability, stabilizability, and continuous-time markovian jump linear quadratic control." *IEEE Transactions on Automatic Control*, **35:7**, pp. 777–788.

JI, Y., H. J. CHIZECK, X. FENG, and K. A. LOPARO (1991): "Stability and control of discrete-time jump linear systems." *Control-Theory and Advanced Applications*, **7:2**, pp. 447–270.

KALMAN, R. E. (1962): "Control of randomly varying linear dynamical systems." *Proceedings of symposia in applied mathematics*, **13**, pp. 287–298.

KALMAN, R. E. and J. E. BERTRAM (1959): "A unified approach to the theory of sampling systems." *Journal of the Franklin Institute*, **267:5**, pp. 405–436.

KRASOVSKII, N. N. and E. A. LIDSKII (1961): "Analytic design of controllers in systems with random attributes, I, II, III." *Automation and Remote Control*, **22:9–11**, pp. 1021–1025, 1141–1146, 1289–1294.

KRTOLICA, R., Ü. ÖZGÜNER, H. CHAN, H. GÖKTAS, J. WINKELMAN, and M. LIUBAKKA (1994): "Stability of linear feedback systems with random communication delays." *Int. J. Control*, **59:4**, pp. 925–953.

LANCASTER, P. (1969): *Theory of Matrices.* Academic Press, New York.

LIOU, L.-W. and A. RAY (1991): "A stochastic regulator for integrated communication and control systems: Part I - Formulation of control law." *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, **113**, pp. 604–611.

LUCK, R. and A. RAY (1990): "An observer-based compensator for distributed delays." *Automatica*, **26:5**, pp. 903–908.

MORGAN, B. J. T. (1984): *Elements of Simulation.* Chapman and Hall.

NILSSON, J., B. BERNHARDSSON, and B. WITTENMARK (1996): "Stochastic analysis and control of real-time systems with random time delays." *Proceedings of the 13th International Federation of Automatic Control World Congress.*

OLSSON, G. and G. PIANI (1992): *Computer Systems for Automation and Control.* Prentice-Hall, Englewood Cliffs, New Jersey.

RAY, A. (1987): "Performance evaluation of medium access control protocols for distributed digital avionics." *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, **109**, December, pp. 370–377.

RAY, A. (1994): "Output feedback control under randomly varying distributed delays." *Journal of Guidance, Control, and Dynamics*, **17:4**, pp. 701–711.

SWORDER, D. D. (1969): "Feedback control of a class of linear systems with jump parameters." *IEEE Transactions on Automatic Control*, **14:1**, pp. 9–14.

TINDELL, K. and H. HANSSON (1995): "Real time systems and fixed priority scheduling." Technical Report, Department of Computer Systems, Uppsala University.

TÖRNGREN, M. (1995): *Modelling and design of distributed real-time control applications.* PhD thesis, Royal Institute of Technology, KTH, Sweden.

VAN OORSCHOT, J. (1993): *Measuring and Modeling Computer Networks.* PhD thesis, Delft University of Technology.

WONHAM, W. M. (1971): "Random differential equations in control theory." *Probabilistic Methods in Applied Mathematics*, pp. 131–212.

A

Kronecker Products

This appendix contains the definition of *Kronecker product* and some results for calculation with Kronecker products. For a more thorough discussion, see Halmos (1958) and Lancaster (1969).

A.1 Definitions

Let $A \in \mathcal{R}^{m\times n}$ and $B \in \mathcal{R}^{p\times q}$. The *Kronecker product* $A \otimes B \in \mathcal{R}^{mp\times nq}$ is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}, \tag{A.1}$$

where a_{ij} are the elements of A. Let $X \in \mathcal{R}^{m\times n}$, with the structure

$$X = [X_1 \quad X_2 \quad \dots \quad X_n]. \tag{A.2}$$

The vectorized form of X, $\text{vec}\{X\} \in \mathcal{R}^{mn\times 1}$, is defined by stacking the columns into a vector as

$$\text{vec}\{X\} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}. \tag{A.3}$$

A.2 Basic Rules of Calculation

The Kronecker product fulfills the following rules of calculation:

$$\begin{aligned}
\alpha A \otimes \beta B &= \alpha\beta(A \otimes B), \quad \alpha, \beta \in \mathcal{R} && \text{(A.4)}\\
(A + B) \otimes C &= A \otimes C + B \otimes C && \text{(A.5)}\\
A \otimes (B + C) &= A \otimes B + A \otimes C && \text{(A.6)}\\
A \otimes (B \otimes C) &= (A \otimes B) \otimes C && \text{(A.7)}\\
(A \otimes B)^T &= A^T \otimes B^T && \text{(A.8)}\\
(A \otimes B)(C \otimes D) &= AC \otimes BD && \text{(A.9)}\\
(A \otimes B)^{-1} &= A^{-1} \otimes B^{-1}. && \text{(A.10)}
\end{aligned}$$

The proofs follow directly from the definition of Kronecker products.

LEMMA A.1

$$\text{vec}\{AXB\} = (B^T \otimes A)\,\text{vec}\{X\}. \qquad \text{(A.11)}$$

□

For a proof see Lancaster (1969).

B

Some Results from Probability Theory

B.1 Markov Chains

This section presents some results on Markov chains. See Berman and Plemmons (1969), and Elliot *et al.* (1995) for a more thorough discussion on Markov chains.

The Markov process is characterized by that if its state is given at a time t, its future evolution is independent of the history that gave the state at time t.

DEFINITION B.1—MARKOV PROCESS
A sequence of random variables $x(t)$ is said to be a *Markov process* or to posses the *Markov property* if the associated probability measure has the property

$$P(x(t_n + h) \mid x(t_1), x(t_2), \ldots, x(t_n)) = P(x(t_n + h) \mid x(t_n)), \qquad \text{(B.1)}$$

where $t_i < t_n$ for $i = 1, ..., n - 1$. □

DEFINITION B.2—MARKOV CHAIN
A finite *Markov chain* is a Markov process that takes values $\{r_k\}$ in a

finite set $\mathcal{S} = \{1, 2, \ldots, s\}$, with transition probabilities

$$P(r_{k+1} = j \mid r_k = i) = q_{ij}. \tag{B.2}$$

The *transition probabilities*, q_{ij}, fulfill $q_{ij} \geq 0$ for all $i, j \in \mathcal{S}$, and

$$\sum_{j=1}^{s} q_{ij} = 1. \tag{B.3}$$

□

Introduce the Markov state probability distribution

$$\pi(k) = [\pi_1(k) \quad \pi_2(k) \quad \ldots \quad \pi_s(k)], \tag{B.4}$$

where $\pi_i(k)$ is the probability that the Markov chain state at time k is i. The probability distribution for r_k is given by

$$\pi(k+1) = \pi(k)Q \tag{B.5}$$

$$\pi(0) = \pi^0, \tag{B.6}$$

where π^0 is the distribution for r_0.

A Markov chain is said to be regular if the transition matrix Q is a primitive matrix. A primitive matrix fulfill $Q^k \gg 0$ for a positive integer k, $A \gg B$ denotes that the matrix elements satisfies $a_{ij} > b_{ij}$. That a Markov chain is regular means that all states will be possible to reach in the future, there are no "dead ends" in the Markov chain.

If a Markov chain is primitive the stationary probability distribution $\pi^\infty = \lim_{k\to\infty} \pi(k)$ is given uniquely by

$$\pi^\infty = \pi^\infty Q, \tag{B.7}$$

where π^∞ is a probability distribution.

B.2 Conditional Independence

Stochastic *independence* of two events X and Y is usually defined by

$$P(X\&Y) = P(X)P(Y), \tag{B.8}$$

where $P(X)$ is the probability that event X occurs. A weaker condition on two events is *conditional independence.*

DEFINITION B.3—CONDITIONAL INDEPENDENCE
X and Y are said to be *conditional independent relative to* Z if

$$P(X\&Y \mid Z) = P(X \mid Z)P(Y \mid Z). \tag{B.9}$$

□

The following theorem is often useful when conditional independence of two events are to be shown.

THEOREM B.1
The following three conditions for X and Y being conditional independent relative to Z are equivalent

1. $P(X\&Y \mid Z) = P(X \mid Z)P(Y \mid Z)$
2. $P(X \mid Y\&Z) = P(X \mid Z)$
3. $P(Y \mid X\&Z) = P(Y \mid Z)$

□

Proof We will use the following three equalities in the proof.

$$\begin{aligned} P(X\&Y\&Z) &= P(X \mid Y\&Z)P(Y\&Z) \\ &= P(X \mid Y\&Z)P(Y \mid Z)P(Z) && \text{(B.10)} \\ P(X\&Y\&Z) &= P(Y \mid X\&Z)P(X\&Z) \\ &= P(Y \mid X\&Z)P(X \mid Z)P(Z) && \text{(B.11)} \\ P(X\&Y\&Z) &= P(X\&Y \mid Z)P(Z) && \text{(B.12)} \end{aligned}$$

$1 \Rightarrow 2$: Use Condition 1 in (B.12), and compare with (B.10).

$2 \Rightarrow 3$: Use Condition 2 in (B.11), and compare with (B.10).

$3 \Rightarrow 1$: Use Condition 3 in (B.11), and compare with (B.12).

□

Theorem B.1 can also be formulated using random variables and expected values.

THEOREM B.2
Let x and y be random variables which are conditionally independent relative to z. The following relations hold for all measurable functions $f(.)$ and $g(.)$.

1. $\mathrm{E}(f(x)g(y) \mid z) = \mathrm{E}(f(x) \mid z)\,\mathrm{E}(g(y) \mid z)$
2. $\mathrm{E}(f(x) \mid y,z) = \mathrm{E}(f(x) \mid z)$
3. $\mathrm{E}(f(y) \mid x,z) = \mathrm{E}(f(y) \mid z)$

□

Proof This is proven by use of Theorem B.1 and the technique in Chapter 9.1 of Chung (1974). □